

К Н И Г А І

**ПЛАНУВАННЯ, МОДЕЛЮВАННЯ, ВЕРИФІКАЦІЯ ТА УПРАВЛІННЯ
В КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ СИСТЕМАХ**

КНИГА І ПЛАНУВАННЯ, МОДЕЛЮВАННЯ, ВЕРИФІКАЦІЯ ТА УПРАВЛІННЯ В КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ СИСТЕМАХ

Список скорочень

АРМ	– автоматизоване робоче місце
АС	– автоматизований склад
АСКПВ	– автоматизована система конструкторської підготовки виробництва
АСНД	– автоматизована система наукових досліджень
АСТПВ	– автоматизована система технологічної підготовки виробництва
АСУВ	– автоматизована система управління виробництвом
АСУТП	– автоматизована система управління технологічним процесом
АТЗ	– автоматизований транспортний засіб
АТСС	– автоматизована транспортно-складська система
АУ	– алгоритм управління
АЦП	– аналогово-цифровий перетворювач
ВП	– виробничий процес
ГАД	– гнучка автоматизована діляниця
ГАЛ	– гнучка автоматизована лінія
ГВК	– гнучка виробнича комірка
ГВМ	– гнучкий виробничий модуль
ГВС	– гнучка виробнича система
ГКІС	– гнучка комп'ютерно-інтегрована система
ГКС	– гнучка комп'ютеризована система
ГКС ПОУ	– гнучкі комп'ютеризовані системи підготовки, організації і управління
ДЛП	– дискретне лінійне програмування
ЕОМ	– електронна обчислювальна машина
ІВС	– інтегрована виробнича система
ІКСП	– ієрархічна кольорова сітка Петрі
КД	– конструкторська документація
КІС	– комп'ютерно-інтегрована система
КСП	– кольорова сітка Петрі
КСУІВ	– комп'ютеризована система управління інтегрованим виробництвом
ЛП	– лінійне програмування
ЛЦП	– лінійне цілочисельне програмування
МПС	– мікропроцесорна система
ОС ЧПУ	– операційна система ЧПУ
ОТАСУ	– організаційно-технічна автоматизована система управління

ОУ	– об'єкт управління
ПЗ	– програмне забезпечення
ПМЗ	– програмно-математичне забезпечення
ПОІ	– понадоперативний інтервал
ППП	– пакети прикладних програм
ПС	– перероджуюча сітка
ПЧПУ	– пристрій числового програмного управління
РО	– робочий орган
САП	– система автоматизації програмування
САПР	– система автоматизованого проектування
СОУ	– система оперативного управління
СП	– сітка Петрі
СС	– селективна сітка
СУ	– система управління
СУБД	– система управління базою даних
ТУ	– технологічне устаткування
УП	– управляюча програма
ЦАП	– цифро-аналоговий перетворювач
ЦДА	– цифровий диференціальний аналізатор
ЦФ	– цільова функція
ЧПУ	– числове програмне управління

*“Гарний початок – не дріб’язок,
хоч і починається з дріб’язку”
(Сократ)*

Вступ

Як показує зарубіжний досвід і досвід передових галузей нашої країни, гнучкість автоматизованих технологічних процесів в умовах зростання питомої ваги дрібносерійного багатомініклатурного виробництва може бути забезпечена організацією останнього за модульним і ієрархічним принципами, основу якого складають гнучкі комп’ютерно-інтегровані системи (ГКІС).

Перехід від гнучких виробничих систем до комп’ютерно-інтегрованих систем (КІС), разом з тим, ґрунтується на нових тенденціях як у технічному вдосконаленні самих компонентів таких систем, так і на підвищенні ефективності їх зкоординованої роботи і організації надійного гнучкого управління. Практика показує, що цим питанням приділяється недостатня увага, а це призводить до збільшення термінів освоєння нової продукції і недостатньої ефективності використання коштовного технологічного устаткування через низьку якість і надійність управлінських рішень. Отже, однією з важливих компонентів підготовки студентів даних спеціальностей є викладання дисциплін, пов’язаних із задачами управління ГКІС.

Метою підготовки даної книги було надання студентам знань в галузі сучасних уявлень щодо методології алгоритмізації та верифікації задач управління комп’ютеризованим інтегрованим виробництвом (КІВ). Разом з тим, вивчення матеріалу має на меті надати студентам навички у використанні ефективних способів побудови програмно-математичного забезпечення організаційно-економічного, організаційно-технічного управління ГКІС та числового програмного управління (ЧПУ) його окремими обробними ресурсами.

Проаналізовано еволюцію автоматизованого виробництва і наведено причини, які призвели до створення сучасних інтегрованих систем з використанням комп’ютерних технологій відображення і перетворення інформації, управління на всіх рівнях таких систем.

Розглянуто теоретичну та інструментальну бази, за допомогою яких створюються інженерні методи розв’язання проблеми організації ефективного алгоритмічного та надійного програмного забезпечення ГКІС. Викладається досвід створення та подаються принципи побудови систем організаційно-економічного, організаційно-технічного та виконавчого рівнів управління.

В книзі знайшли відображення сучасні тенденції до інтеграції управління елементами КІВ, розглянуто ієрархію управління виробничими процесами з визначення задач на кожному рівні, методи та засоби диспетчеризації матеріальних потоків, а також сучасні методи їх верифікації. Як математична форма подання матеріально-

інформаційних процесів в КІВ та їх верифікації розглядаються дискретно-подійні моделі на базі апарату сіток Петрі, а як методологія розв'язання – імітаційні методи досліджень, методи математичного програмування.

Також подаються основи організації програмного управління виконавчим рівнем ГКІС, наводиться технологія програмування формоутворення об'єктів обробки та руху робочого органу технологічного устаткування. Розкривається суттєвість інтерполяційного процесу і його вплив на якість процесу формоутворення, наводяться структура та способи реалізації задач програмного управління в системах ЧПУ, визначаються основи систем автоматизованої підготовки управляючих програм до устаткування з ЧПУ.

*“Тому що Ви не можете робити
усе, що хочете, то бажайте
тільки того, що можете
зробити” (Теренцій Публій)*

Розділ 1

ОРГАНІЗАЦІЙНО-ТЕХНОЛОГІЧНІ ОСНОВИ ПОБУДОВИ ГКІС В ІНТЕГРОВАНОМУ ВИРОБНИЦТВІ

1.1. Напрямки та шляхи розвитку автоматизації виробничих систем

Діалектика розвитку основних методів організації виробництва показує існування стадій та їх складових етапів автоматизації, зміна яких обумовлена об'єктивно існуючими протиріччями між вимогами ефективності та мобільності виробництва, зокрема, між підвищенням продуктивності та скороченням терміну освоєння нової продукції. Загальна тенденція розвитку стадій передбачає повторний рух у напрямку від універсальних одиниць устаткування до їх спеціалізації та групування у комплекси. Таким чином, кожна стадія загалом може включати чотири основних етапи:

- універсальне устаткування як одиниця обладнання;
- спеціалізоване устаткування як одиниця обладнання;
- універсальні системи устаткування, об'єднані у групи, лінії, дільниці;
- спеціалізовані системи устаткування.

Так, на початку розвитку автоматизації виробництва головною задачею було збільшення обсягу випуску досить обмеженого кола товарів зниження питомих витрат на виробництво одиниці товару. Основний спосіб розв'язку цієї задачі полягав у поглибленні поділу праці, коли робітник з відносно вузькою спеціалізацією виконував багаторазово повторювані найпростіші операції, не потребуючи високої кваліфікації. Відбувалися стандартизація товарів, збільшення числа окремих деталей і операцій техпроцесу їх виготовлення, уніфікація виробництва. Однак у подальшому почалися зміни в структурі попиту, коли первинні потреби були задоволені масовим виробництвом основних видів товарів, і почалася диференціація попиту за номенклатурою товарів та їх якістю. Одночасно в середині самого виробництва з'явилися фактори, що сприяли переходу до нової моделі виробництва.

Таким чином, виникло соціально-економічне замовлення суспільства на створення комплексно-автоматизованих виробництв на базі гнучких комп'ютерно-інтегрованих систем (ГКІС), яке можна реалізувати як сучасним рівнем науково-технічних знань, так і рівнем розвитку технічних засобів.

Отже, діалектика розвитку автоматизації виробничих процесів

на основі ГКІС може бути відтворена у вигляді спіралі, яка наведена на рис. 1.1.

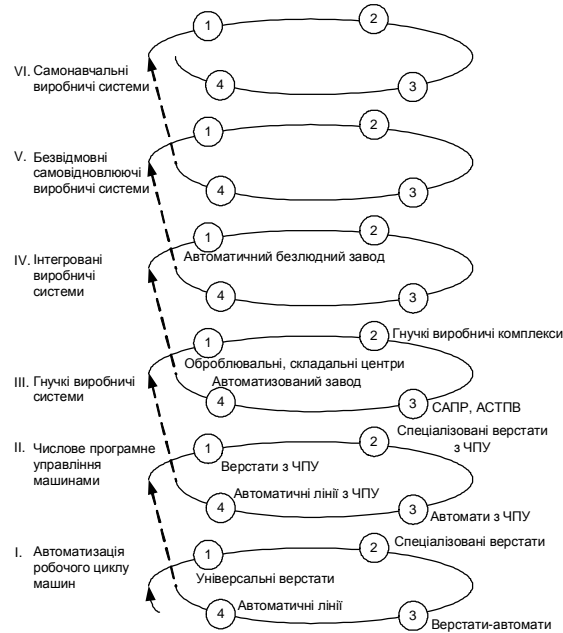


Рис. 1.1. Схема діалектичного розвитку гнучкого виробництва

На I стадії розвитку автоматизації – “Автоматизація робочого циклу машин” – засоби виробництва пройшли шлях від універсальних верстатів, універсальних, спеціальних і спеціалізованих автоматів і напівавтоматів до агрегатних верстатів, автоматичних ліній з агрегатних верстатів, універсальних автоматів і ліній на їх основі. Людина передала машинам функції обробки предметів праці, залишивши собі функції обслуговування та управління машинами і виробництвом в цілому. Принцип управління, що застосовується до технологічного устаткування цієї стадії, заснований на електромеханічному управлінні.

Граничною формою організації виробництва цієї стадії розвитку автоматизації в машинобудуванні стали комплексні автоматичні лінії та автоматичні заводи (зокрема, автоматичні заводи по виробництву підшипників, виготовленню автомобільних поршнів тощо) з високим рівнем продуктивності, але малою мобільністю. Цей рівень розвитку виробництва забезпечив значне підвищення продуктивності праці (у 5–10 разів) і зниження собівартості (на 30–50 %), однак застосовується тільки у масовому виробництві виробів, конструкція яких тривалий час залишається стабільною.

На створення та налагодження жорстких автоматичних ліній витрачається до 5 років, термін автоматизації складає 8 років і більше. Тому конструкція виробів, що випускаються, на таких виробництвах тривалий час повинна залишатися незмінною. Консерватизм твердої автоматизації не задовольняє вимогам науково-технічного прогресу,

прискорення змінюваності виробів у машинобудуванні. Таким чином, підвищення продуктивності жорстких автоматичних виробництв було досягнуто за рахунок повної втрати їх мобільності. Рішення цього протиріччя – підвищення мобільності при одиничному і серійному характері виробництва – повернуло машинобудування знову до універсальних верстатів, але оснащених системами числового програмного управління (ЧПУ).

В результаті II стадії розвитку автоматизації практично повторила I, але на новому принципі переналагодження – зміні керуючої програми в системі ЧПУ, що дозволило значно підвищити гнучкість устаткування та створюваних на його засадах автоматичних ліній.

Розвиток засобів виробництва на II стадії – *“Числове програмне управління устаткуванням”* – еволюціонувало від верстатів із ЧПУ, автоматів із ЧПУ, спеціальних верстатів із ЧПУ до багатоопераційних верстатів – обробних центрів з індивідуальними системами ЧПУ і побудови на їх основі ділянок із закінченим циклом виготовлення окремих виробів. Людина передає машині функції управління обробкою предметів праці, залишивши собі функції обслуговування машин, підготовки та організації виробництва. Принцип управління, використаний на цієї стадії, – це програмне управління устаткуванням від ЕОМ. Використання устаткування з ЧПУ дійсно дозволило одержати значний ефект в одиничному і серійному виробництвах, однак індивідуальне для кожного верстата управління від своєї стійки ЧПУ виявилось громіздким і дорогим. Крім того, залишалися невирішеними задачі автоматизації управління матеріальними потоками на таких дільницях, а також завантаження самих верстатів. Формою організації виробництва стали автоматичні лінії устаткування з ЧПУ.

Розвиток робототехніки і подальше удосконалення електроніки на основі мікропроцесорних засобів розкрили нові можливості підвищення рівня автоматизації виробничих процесів у машинобудуванні.

Створенням промислових роботів і універсальних технологічних машин і верстатів із ЧПУ, безпосередньо керованих від ЕОМ у режимі поділу часу, почалася III стадія розвитку автоматизації в машинобудуванні та інших галузях промисловості – *“Гнучкі виробничі системи”*. З’явилась можливість зв’язати технологічне устаткування єдиним управлінням і автоматичним транспортом, тобто створити системи машин, об’єднаних спільністю цільової функції – виробництвом визначеного виду виробів. На даній стадії людина займається організацією та підготовкою виробництва, поклавши на машини функції обробки і управління групами устаткування. Застосований принцип управління стадії – управління від декількох ЕОМ групами машин з формою організації виробництва у вигляді

гнучких автоматизованих дільниць, ліній, цехів. На цьому рівні з'явилися передумови і почалося з'єднання в єдину комп'ютерно-інтегровану систему (КІС) усіх виробничих функцій: конструювання, технологічної підготовки виробництва і виготовлення виробів, що і стало основою створення гнучких виробничих систем (ГВС). Виходячи із загальних тенденцій розвиток засобів виробництва на цій стадії визначається такими формами реалізації етапів автоматизації:

- багатофункціональне універсальне устаткування з ЧПУ;
- автоматизований транспорт;
- системи автоматизованого проектування і підготовки виробництва;
- автоматизований завод.

Незважаючи на такий діалектичний хід подій, поява ГВС породжує серйозні проблеми, одна з яких (найбільш істотних) полягає в наступному. У зв'язку з тим, що ГВС мають продуктивність у декілька разів більшу, ніж звичайна людино-машинна виробнича система, виникли тверді вимоги до скорочення термінів технічної підготовки виробництва, що стала одним з основних стримуючих факторів збільшення швидкості виробничого процесу в цілому. Наслідком появи ГВС виявилася проблема створення інтегрованих виробничих систем, у яких об'єктом автоматизації є весь виробничий процес, починаючи від наукових досліджень, конструювання виробів і закінчуючи їх виготовленням, випробовуванням і складуванням.

Наступна, IV стадія – *“Інтегровані виробничі системи”*, почнеться зі створення автоматизованого виробництва, повністю інтегрованого на базі ЕОМ п'ятого покоління, а закінчиться створенням гнучкого автоматичного заводу з безлюдною технологією. На цій стадії передбачається проведення повної комплексної автоматизації виробництва від проектування до виготовлення продукції та створення на базі гнучких комп'ютеризованих систем (ГКС) інтегрованих виробничих систем з принципом управління – штучний інтелект.

Подальший розвиток виробництва піде по шляху вирішення проблеми надійності, самостійної діагностики робочих машин і значного підвищення рівня інтелектуальності інформаційно-управляючих та гнучких комп'ютеризованих систем. Розвиток застосування штучного інтелекту забезпечить успішне розв'язання цієї задачі на V – *“Безвідмовні самовідновлювані виробничі системи”* і VI – *“Самонавчальні виробничі системи”* стадіях розвитку автоматизації гнучкого виробництва.

1.2. Узагальнена структура ГКІС в інтегрованому виробництві

Інтегрована виробнича система (ІВС) – це багаторівнева ієрархічна система, що реалізує комплексно-автоматизоване групове

багатономенклатурне виробництво, яке перебудовується оперативно у визначеному параметричному діапазоні продукції.

Комплексно-автоматизоване виробництво припускає наскрізну автоматизацію виробничого циклу випуску продукції – від завдання на проектування до готової продукції (“проектування – планування – виготовлення”), тобто автоматизації наступних його стадій:

- проектування предметів виробництва;
- підготовка засобів і технологічних процесів виробництва;
- організація планування виробництва;
- реалізація планів виробництва.

Групова технологія виробництва припускає можливість організації багатономенклатурного виробництва, коли устаткування повинне автоматично перебудовуватися на випуск нового виробу і використовувати при цьому весь необхідний комплекс інструментів.

Отже, автоматизація і інтеграція всіх стадій виробничого циклу призводить до створення ГКС на кожній із наведених стадій. Внаслідок цього утворюється ІВС, що реалізує функцію гнучкого комп'ютеризованого виробництва, робота всіх функціональних елементів якого синхронізується *багаторівневою комп'ютеризованою системою управління*.

Структурно ІВС поділяється на (рис. 1.2):

1) гнучкі комп'ютеризовані системи підготовки, організації та управління (ГКС ПОУ);

2) гнучку виробничу систему (ГВС).

У ГКС ПОУ входять:

а) автоматизовані системи наукового дослідження (АСНД);

б) система автоматизованого проектування (САПР) і автоматизована система технологічної підготовки виробництва (АСТПВ);

в) автоматизована система управління виробництвом (АСУВ).

АСНД в послідовності складових виробничого циклу є першою і важливою стадією при розробці нових конструкцій, вузлів, виробів. Метою роботи цієї системи є аналіз можливості втілення ідеї нового виробу та вироблення концепції його створення. Залучення і інтегрування АСНД до наскрізного виробничого циклу дозволить сформувати технічне завдання (ТЗ) та ескізний проект (ЕП) нового виробу і підвищити ефективність та якість створюваних конструкцій.

САПР як наступна стадія наскрізного виробничого циклу забезпечує автоматизацію процесу конструювання нового виробу з формуванням конструкторської документації (КД) технічного і робочого проекту виробу. Під процесом конструювання розуміється проведення різноманітних розрахунків об'єктів і систем, які проектуються, дослідження динамічних, геометричних та інших властивостей конструкцій, моделювання роботи виробу в умовах зміни зовнішнього середовища, а також виготовлення креслень, специфікацій

у складі технічної документації. Внаслідок застосування такої системи прискорюється процес переходу виробництва на нові види продукції.

АСТПВ охоплює широке коло задач, пов'язаних з підпрацюванням конструкції нового виробу на технологічність і розробки технологічних процесів виготовлення і контролю складання виробу. Застосування системи призведе до визначення необхідних технологічних операцій і відповідного устаткування, на якому вони будуть здійснюватися, з метою пошуку раціональних технологічних маршрутів (ТМ) цих операцій. Крім цього, проводиться нормування витрат ресурсів і режимів роботи устаткування, що впливає на загальну ефективність виробництва.

Нарешті, АСУВ завершує організаційно-підготовчі стадії виробничого циклу перед здійснення безпосередньо самого виробництва. Складність і взаємна обумовленість задач управління виробництвом призвела до необхідності створення та використання на практиці автоматизованих систем управління виробництвом, як багаторівневої автоматизованої системи управління організаційно-економічного класу, мета якої координувати і синхронізувати роботу всіх функціональних елементів ГКС в інтегрованому виробництві для реалізації найбільш ефективного управління технологічними режимами виробничих ланок, диспетчеризації матеріальних потоків і планування процесу виробництва.

Слід зазначити, що використаний термін “автоматизована система управління виробництвом” є більш узагальненим, який підкреслює функціональну ознаку класа організаційно-економічних систем, а для позначення рівня автоматизації управління за структурною ознакою може бути застосовано термін “автоматизована система управління підприємством”. Таким чином, за ознакою об'єкта управління АСУВ слід поділити на системи управління структурним підрозділом, підприємством, об'єднанням, галузью, територіальним комплексом та інше.

Гнучка виробнича система – це сукупність у різних поєднаннях устаткування з числовим програмним управлінням (роботизованих технологічних комплексів, виробничих модулів, окремих одиниць технологічного устаткування) і систем забезпечення їх функціонування в автоматичному режимі протягом заданого інтервалу часу, що має властивість автоматизованого переналагодження при виробництві виробів довільної номенклатури у встановлених межах значень їх характеристик.

ГВС включає наступні функціональні підсистеми:

а) технологічну підсистему – складається з комплексів основного технологічного і допоміжного устаткування;

б) підсистему управління – складається з організаційно-технічних АСУ й автоматизованої системи управління технологічними процесами (АСУТП);

в) сервісну підсистему – виконує діагностику і ремонт основного технологічного устаткування;

г) підсистему підготовки виробництва – здійснює підготовку технологічних процесів, управляючих програм до технологічного устаткування, інструментів та пристроїв.

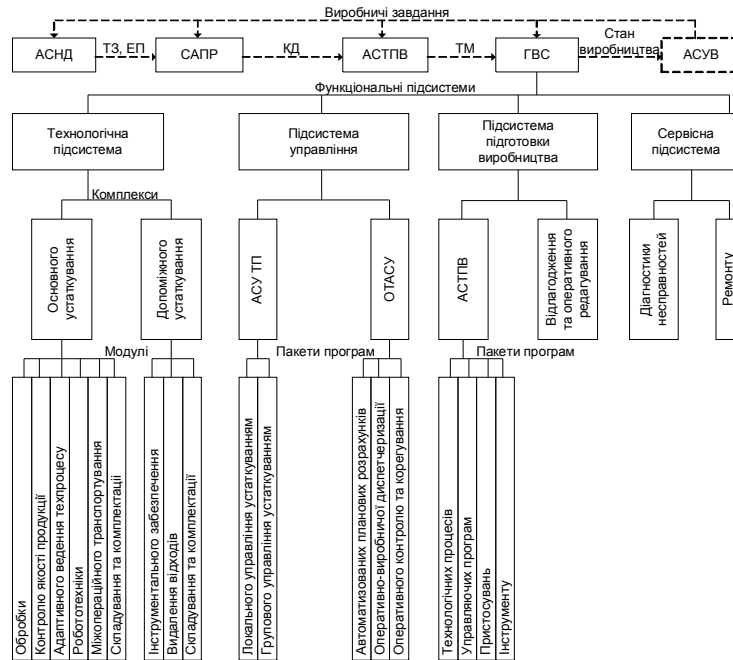


Рис. 1.2. Узагальнена структура гнучких комп'ютеризованих систем в інтегрованому виробництві

Таким чином, під ГВС розуміють виробничу одиницю (лінія, дільниця, цех, завод), що функціонує на основі безлюдної технології, робота усіх виробничих компонентів якої координується як єдине ціле з багаторівневою системою управління, яка забезпечує швидку зміну програм функціонування компонентів ГВС при переході на випуск нової продукції.

Класифікація ГВС здійснюється за наступними ознаками:

- організаційна ознака будови;
- функціональна ознака використання;
- тип конфігурації транспортної системи;
- рівень автоматизації.

За першою ознакою визначається розвиненість структури ГВС:

а) гнучкий виробничий модуль (ГВМ);
б) гнучка автоматизована лінія і гнучка автоматизована дільниця;

в) гнучкий автоматизований цех (декілька ліній чи дільниць, об'єднаних в єдине ціле).

За другою ознакою визначається сфера використання ГВС, зокрема:

а) за видом виробництва – операційні ГВС (заготівельні, механо-

оброблювальні, складальні тощо);

б) за видом роботи – регулювальні, слюсарні ГВС тощо;

в) за видом деталей – предметні ГВС (корпусні, типу тіл обертання тощо);

г) за видом складальних з'єднань – вузлові ГВС.

Третя ознака визначає тип конфігурації АТСС та компоновання ГВС (рис. 1.3):

а) за просторовим розташуванням устаткування – на лінійні, багаторядні та кругові;

б) за зоною обслуговування – на ГВС із поздовжньою орієнтацією устаткування та ГВС із поперечною орієнтацією;

в) за організацією транспортних потоків – на прямоточні (рух тільки в одному напрямку) і поворотні (двосторонній характер руху);

г) за організацією матеріальних потоків – на ГВС із роздільною подачею матеріалів, комплектуючих та інструментів, ГВС зі спільною подачею і ГВС із комбінованою подачею.

Рівень автоматизації в ГВС може бути різним. Але головною задачею передпроектної стадії є визначення оптимального рівня автоматизації обраного устаткування, що повинен забезпечити нормативну ефективність капітальних вкладень чи припустимі капітальні вкладення при досяжному рівні росту продуктивності.

Тому, у гнучкому виробництві допускається кілька рівнів механізації та автоматизації, але вищим рівнем є створення системи оперативного управління (СОУ) на базі обчислювальної техніки, яка дозволяє автоматизувати процеси планування завантаження технологічного устаткування, регулювати якнайкраще запаси матеріалів, підтримувати ритмічний хід виробництва.



Рис. 1.3. Показники компоновання ГВС

1.3. Організація управління в інтегрованому виробництві

1.3.1. Класифікація систем управління

Будь-який цілеспрямований процес, що протікає у ІВС,

представляє собою сукупність операцій, що, в залежності від виду виконуваних функцій, умовно можна розбити на дві групи: виробничі операції та операції управління.

Виробничі операції реалізують виробничу функцію ІВС, яка полягає у виконанні технологічною підсистемою в автоматизованому чи автоматичному режимі технологічного процесу по випуску визначеної продукції.

Під *технологічним процесом* розуміється виконувана сукупність дій, що забезпечує спрямоване рішення визначених задач, орієнтованих на перетворення за допомогою елементів технологічної підсистеми (засобів виробництва) матеріальних потоків, що протікають (предметів і продуктів виробництва).

Для досягнення мети технологічного процесу виробничі операції повинні організовуватися і направлятися діями іншого роду – операціями управління. *Операції управління* реалізують інформаційну функцію ІВС, яка полягає в реалізації в автоматизованому чи автоматичному режимі збору, обробки і видачі інформації для виконання управління технологічним процесом.

Управління – це процес, що розвивається в реальному часі і забезпечує нормальний хід виконання технологічного процесу. Тому система управління (СУ), в рамках якої функціонує процес управління, складає спеціальний підклас динамічних систем.

У результаті такого поділу функцій система управління інтегрованим виробництвом може бути представлена класичною схемою організації управління за подійним принципом, що застосовується в динамічних системах зі зворотним зв'язком по стану (рис. 1.4).

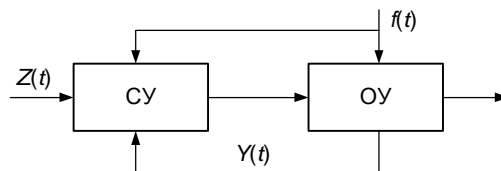


Рис. 1.4. Загальна схема системи управління ІВС

Процес управління в системах зі зворотним зв'язком організовується відповідно до програми управління, яка задана для всіх моментів часу або відома тільки для поточних моментів часу. Цей процес полягає у визначенні відхилень оцінок поточного стану від стану, передбаченого на даний момент часу відповідно до програми формування закону управління в системі, з метою утримання її на заданій траєкторії руху за умови виконання обмежень на залучені ресурси та обов'язкового забезпечення прийнятних показників ефективності управління.

У ІВС поняття *об'єкта управління* (ОУ) використовується в широкому розумінні. У нього включаються як технологічні процеси

виробництва, так і процеси, з якими зв'язане забезпечення обумовлених вимогами експлуатації нормальних режимів функціонування виробничого устаткування і руху матеріальних потоків. Як об'єкти управління розглядаються окремі верстати, агрегати, ділянки, цехи, економічні та конструкторські підрозділи. Взаємодія елементів СУ та ОУ визначається інформаційними зв'язками. Границя між СУ та ОУ проходить там, де інформаційні процеси перетворюються у зміни ходи матеріальних процесів.

Системи управління елементами ІВС дуже різноманітні, як і зв'язані з ними відповідні об'єкти управління. Тому розглянемо класифікацію ознак побудови застосовуваних систем управління. Виділяють наступні ознаки:

1. Ступінь автоматизації інформаційної функції.
2. Ступінь складності системи управління;
3. Ступінь визначеності системи управління;
4. Принцип управління;
5. Число рівнів управління.

В залежності від *ступеня автоматизації* розрізняють ручне, автоматичне і автоматизоване управління. При ручному керуванні усі функції процесу управління виконуються людиною. У автоматичних системах процеси управління реалізуються без особистої участі людини. Автоматизованими називають системи, у яких частина функцій виконується людиною (в основному при ухваленні рішення), а інша частина – машинами (при зборі і первинній обробці інформації).

За *ступенем складності* системи управління поділяють на прості та складні.

Поняття “складна система” виникло як відображення факту існування в реальному світі таких об'єктів, повний опис яких важкий або зовсім неможливий.

Складні системи мають наступні відмінні риси:

- число параметрів, якими описується система, дуже велика, а багато параметрів не піддаються кількісному опису і виміру;
- мета управління не піддається формальному опису без істотних спрощень і є функцією часу;
- важко дати строгий формальний опис складної системи управління і, як правило, основною задачею при моделюванні таких систем є пошук розумного спрощення їхнього опису.

За *ступенем визначеності* системи управління розбивають на детерміновані та випадкові.

Детермінованою називають систему, у якій по її попередньому стану і деякій додатковій інформації можна чітко сформулювати наступний стан.

У імовірнісній системі на основі попереднього стану і додаткової інформації можна визначити лише множину наступних станів та ймовірність кожного з них.

Розбиття систем на прості та складні, детерміновані та імовірнісні деякою мірою умовне. В міру розвитку засобів моделювання і дослідження конкретної системи може перейти з одного класу в інший. Наприклад, простою імовірнісною системою можна назвати систему статичного контролю якості продукції підприємства по одному чи декількох параметрах, що передбачає вибірккову перевірку заданих параметрів з визначеною періодичністю.

Однією з найважливіших характеристик системи управління є її структура. Вона характеризує внутрішню побудову системи та описує стійкі зв'язки між елементами.

За *принципом управління* структури систем поділяються на централізовані та децентралізовані. У централізованих системах усі процеси управління зосереджені в єдиному керуючому органі, який обробляє всю інформацію, що надходить від всіх елементів об'єкта управління. У децентралізованих системах для кожного елемента об'єкта управління створюється власний керуючий орган, що реалізує локальну мету функціонування елемента, а глобальна (єдина) мета системи забезпечується за рахунок обліку й узгодження інформації, що надходить від інших елементів об'єкта управління (рис. 1.5).

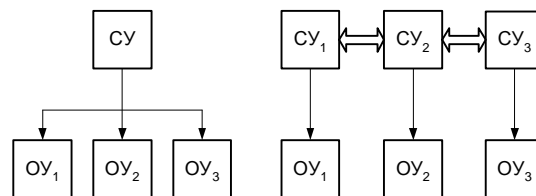


Рис. 1.5. Централізована та децентралізована системи управління

За *кількістю рівнів підпорядкованості* розрізняють однорівневу та багаторівневу (ієрархічну) системи управління. В ієрархічних системах функції управління розподілені між декількома супідрядними органами з одночасним дотриманням принципу централізації. У таких системах керуюча інформація надходить з верхніх рівнів на нижні, а інформація про стан – у зворотному напрямку (рис. 1.6).

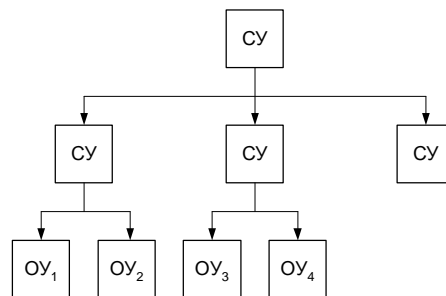


Рис. 1.6. Ієрархічна система управління

Гнучкі комп'ютеризовані системи управління в інтегрованому виробництві, як правило, є багаторівневими.

1.3.2. Ієрархія та задачі рівнів управління ІВС

Структура системи управління ІВС, як правило, будується з використанням ієрархічного та функціонального принципів виділення підсистем. Кожен рівень характеризується особливостями цілей і задач управління.

За *ієрархічним* принципом виділяють наступні рівні та підсистеми в ІВС.

Виконавчий (перший) рівень складається із множини систем управління окремими виробничими ланками (агрегатами, верстатами, устаткуванням). Метою управління на цьому рівні зазвичай є вибір і підтримка заданих технологічних режимів виконання технологічних операцій. Управління зводиться в основному до контролю параметрів виконання технологічної операції і до впливу безпосередньо на виконавчі механізми ланок.

Тактичний (другий) рівень ієрархії утворюють системи диспетчерського управління виробничими дільницями і лініями. Основною метою управління цього рівня є оперативна диспетчеризація руху матеріальних потоків через вибір і підтримку режиму спільного функціонування виробничих ланок, на яких виконуються взаємозалежні технологічні операції, що складають технологічний процес. На цьому рівні виробляється адаптація календарного плану виконання технологічних операцій до зміни виробничої ситуації.

Стратегічний (третій) рівень ієрархії утворюють системи організаційного управління виробничими підрозділами. Основною метою рівня є вироблення погодженого плану функціонування структурних підрозділів на оперативний плановий період. На цьому рівні виробляється деталізація виробничих завдань по випуску продукції виробничими цехами.

Сукупність систем управління другого і третього рівнів складають ОТАСУ виробничими підрозділами.

Адміністративний (четвертий) рівень ієрархії утворюють системи, що забезпечують управління підприємством у цілому. Ціль управління полягає в підготовці й організації спільного функціонування виробничих та інших підрозділів підприємства по випуску готової продукції у визначеній кількості при встановлених техніко-економічних показниках. Для реалізації поставленої мети в процесі управління необхідно реалізувати функції економічного характеру.

Специфіка організаційно-економічних і організаційно-технічних систем відбивається в змісті основних задач управління (табл. 1.1), що забезпечують реалізацію глобальної мети підприємства. Глобальною метою є виконання планового завдання, що регламентує обсяг, номенклатуру, терміни та умови використання виробничих ресурсів. У процесі планування на основі планового завдання визначаються техніко-економічні показники підприємства в цілому і виробляється їх деталізація по всіх підрозділах, цехах, дільницях, лініях.

Таблиця 1.1

Ієрархія і задачі рівнів управління ІВС

Рівень	Тип системи управління	Мета і виконувана функція управління	Задачі
Адміністративний	АСУВ – система організаційно-виробничого планування	Техніко-економічне планування; Організаційно-економічне управління	1. Формування виробничої програми підприємства 2. Статистичний облік показників роботи підприємства 3. Управління збутом продукції 4. Контроль якості готової продукції 5. Управління матеріально-технічним постачанням
Стратегічний	ОТАСУ – система оперативного планування	Оперативне планування; Організаційно-технічне управління	1. Формування календарних планів і змінно-добових завдань 2. Оперативний облік загальних показників роботи ІВС 3. Корегування календарних планів і завдань 4. Ведення оперативних баз даних
Тактичний	ОТАСУ – система оперативно-диспетчерського управління	Оперативна диспетчеризація; Технологічне управління	1. Диспетчеризація матеріальних потоків 2. Аналіз та оперативний контроль 3. Облік первинної інформації про стан устаткування 4. Диспетчеризація інформаційних потоків 5. Оперативне регулювання
Виконавчий	АСУТП СУ ГВМ СУ АТЗ СУ АНС	Логіко-програмне управління устаткуванням; Локальне управління	1. Управління технологічними операціями 2. Управління транспортними операціями 3. Управління допоміжними операціями 4. Управління

			промисловими роботами
--	--	--	-----------------------

В залежності від того, на які періоди складаються плани, планування поділяють на перспективне і поточне.

Перспективне планування припускає розробку планів функціонування підприємства на тривалий період (більше 1 року) і є прогнозним плануванням макроекономічних показників діяльності.

Поточне планування поділяють на річне (техніко-економічне) і оперативне. Техніко-економічне планування деталізує показники перспективного плану на даний період і коректує їх відповідно до вимог планового завдання підприємства. В оперативному плануванні конкретизуються показники техніко-економічного плану на короткі інтервали планового періоду (місяць, декада, доба) і визначаються дії, необхідні для їх виконання. Оперативні плани розробляються із врахуванням результатів виконання плану на попередніх інтервалах.

В організаційно-технічних та технологічних системах спільно розглядаються три задачі: оперативне планування, контроль (визначення відхилення поточного стану виробництва від планового) і регулювання (диспетчеризація відхилень). Сукупність цих задач лежить в основі оперативного управління. Систему, що реалізує дану сукупність задач, часто називають системою оперативного управління (СОУ).

Крім ієрархічної (вертикальної) декомпозиції ІВС, виконується її розбиття за *функціональним* принципом (горизонтальна декомпозиція) на підсистеми, зокрема:

- виробництва готової продукції (основне виробництво);
- виробництва допоміжного устаткування, інвентарю, інструмента та оснащення (ВДО);
- технічної (конструкторської та технологічної) підготовки виробництва (ТПВ);
- матеріально-технічного забезпечення (МТЗ);
- організаційно-трудова підготовки (ОТП);
- фінансово-бухгалтерської діяльності (ФБД);
- реалізації готової продукції (РГП).

Функціональні підсистеми призначені для реалізації наступних функцій:

- ВДО - ремонтні, енергетичного постачання, транспортні та інструментальні;
- ТПВ - конструювання, розробка технології виробництва, нормування витрат і ресурсів;
- МТЗ - обліку матеріалів, розрахунку потреби у матеріалах та інших ресурсах;
- ОТП - набір і навчання кадрів, нормування та організація праці, розміщення кадрів, оплата праці;
- ФБД - аналіз, облік фінансового стану підприємства, нарахування зарплати, контроль використання матеріальних цінностей;
- РГП - складування, організація відвантаження, здійснення

оперативних контактів із замовниками.

1.3.3. Організація багаторівневої ієрархічної системи управління в ІВС

Розглядаючи ІВС з позиції організації управління, важливо відзначити, що в кожній з її підсистем діють закони управління зі зворотним зв'язком, функціонування яких спрямоване на досягнення бажаних динамічних і економічних показників виробництва як на оперативних інтервалах часу, так і в цілому. У результаті інтеграція ГВС з АСНД, САПР, АСТПВ, АСУВ призводить до необхідності дії механізмів прямого директивного управління виробництвом на всіх його рівнях і у всіх ланках. Відомо, що реальний шлях організації управління виробництвом зв'язаний не тільки з розвитком і удосконаленням принципів, систем і засобів управління зі зворотним зв'язком у реальному часі на базі різних СУ, що діють у рамках ГВС, але і з розвитком прямого програмного управління в умовах єдиної комп'ютеризованої системи управління інтегрованим виробництвом (КСУІВ). Отже, взаємодія автоматизованих підсистем КСУІВ та їх функціонування пов'язане з процесами передачі інформації.

Інформаційні потоки вихідні від верхніх рівнів КСУІВ спрямовані до нижніх рівнів, містять директиви управління, виконання яких елементами і підсистемами призводить до бажаних цілеспрямованих змін станів устаткування і матеріальних потоків. Інформаційні потоки, що рухаються знизу вгору, переносять сигнали зворотного зв'язку про хід виробництва, про оцінки характеристик функціонування елементів і підсистем, про стан матеріальних потоків і устаткування, обробка яких породжує нові директиви управління.

Можлива схема організації багаторівневої ієрархічної системи управління в ІВС наведена на рис. 1.7, на якому розкрита структура процесів інформаційного обміну між елементами, підсистемами і рівнями ІВС.

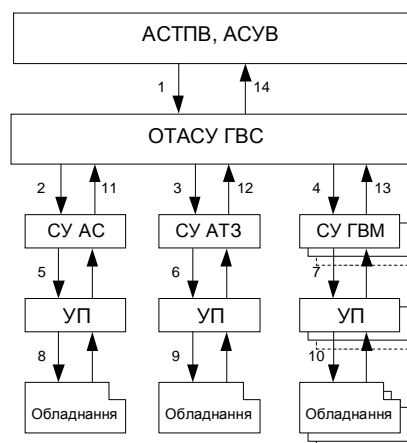


Рис. 1.7. Схема багаторівневої системи управління в ІВС САПР за замовленням на розробку конструкторської

документації (КД) на продукцію, що надходить від АСНД, пройшовши експертну оцінку, здійснює проведення конструкторських робіт і видачу АСТПВ інформації про спроектований виріб, оснащення та інструмент. САПР не тільки проектує КД на виріб, але і забезпечує його технологічність із врахуванням можливості обробки на існуючому устаткуванні. Тому для САПР зворотний зв'язок – це інформація про обліковий склад наявного устаткування та завдання від АСТПВ на проектування інструмента й оснащення, що не є типовим, а директива управління – це КД на виріб, інструмент і оснащення.

АСТПВ за отриманою від САПР інформацією забезпечує розробку і передачу АСУВ і ОТАСУ маршрутних та операційних технологічних процесів виготовлення продукції в ГВС, а також розробляє керуючі програми для устаткування з числовим програмним управлінням і необхідну нормативну документацію.

АСУВ здійснює розробку виробничої програми ГВС на планові інтервали часу, забезпечуючи їх необхідним матеріально-технічним постачанням, контролює та погоджує з виробництвом збут готової продукції.

ОТАСУ одержує від АСУВ директиви управління 1 у вигляді виробничої програми по номенклатурі та обсягу на плановий період, технології виробництва продукції, зведень про надані ресурси, даних про терміни випуску, вимог до якості готової продукції. Зворотний зв'язок 11 від ОТАСУ до АСУВ утворюється на основі даних про виконання календарних планів і змінно-добових завдань, зведень про незавершене виробництво, про стан елементів і підсистем ГВС (збої, простої, ремонт), даних про завантаження складів (дефіцит матеріалів, інструмента, оснащення, тари), заявок на відвантаження готової продукції і видалення виробничих відходів, зведень про якість продукції, довідок про неможливість виконання виробничих завдань.

До складу ГВС входять наступні підлеглі підсистеми управління: автоматизованим складом (АС), автоматизованими транспортними засобами (АТЗ), гнучкими виробничими модулями (ГВМ).

Основними параметрами АС є місткість і фактичне заповнення. В АС наявні матеріальні потоки, що надходять у ГВС ззовні: комплектуючі, матеріали, інструмент, оснащення, тара, а на виході діють потоки, що залишають ГВС: товарна продукція, інструмент, оснащення, тара. Крім того, на вході і виході АС наявні матеріальні потоки, що не є ні вхідними, ні вихідними потоками ГВС: продукти обробки, інструмент та оснащення, що повторно використовуються в операціях технологічних процесів. По матеріальному потоку АС взаємодіє з АТЗ.

Характеристиками АТЗ є ступінь завантаження транспортних засобів системи, оцінка корисного часу роботи і непродуктивних витрат часу, середнє значення довжини черг заявок на обслуговування.

Заявки надходять в АТЗ від ГВМ.

Технологічна підсистема ГВС поєднує сукупність ГВМ, що виконують технологічні операції обробки, складання продукції. Характеристиками ГВМ є функціональний стан устаткування, інструмента, ступінь завантаження устаткування. ГВМ взаємодіють по матеріальному потоку з АТЗ, формуючи заявки на доставку матеріалів, комплектуючих, інструмента, а також заявки на розвантаження від готової продукції при верстатних нагромаджувачів.

Погодженість дій підлеглих ОТАСУ підсистем управління забезпечується інформаційними потоками директив управління 2, 3, 4 і сигналами зворотного зв'язку 12, 13, 14.

Директива управління 2 – це відомості про терміни та обсяги видачі сировини, матеріалів, комплектуючих, інструмента, оснащення, запити про стан складського устаткування і збережених одиниць матеріального потоку, а також керуючі завданням на переміщення вантажів.

Директива управління 3 – це дані про терміни завантаження і кількість одиниць матеріального потоку, запити про стан устаткування і перевезених вантажів, завдання на транспортування вантажів.

Директива управління 4 – це змінно-добове завдання по обсягу і номенклатурі продукції, технологічна інформація (керуючої програми, технологічні карти), запити про хід технологічного процесу і стан устаткування, дані про терміни відвантаження готової продукції та надання ресурсів.

Зворотний зв'язок 12 – це інформація про наявність предметів виробництва на складі, довідки про стан устаткування і про виконання керуючого завдання, заявки на постачання одиниць матеріального потоку і термінове відвантаження продукції.

Зворотний зв'язок 13 – це інформація про терміни та обсяги виконаних перевезень одиниць матеріального потоку, довідки про стан устаткування, про завантаження, простої та ремонт.

Зворотний зв'язок 14 – це інформація про виконання виробничого завдання, довідки про стан устаткування й інструмента, заявки на передачу технологічної інформації, на відвантаження продукції та відходів.

Інші інформаційні зв'язки деталізують передані директиви управління і представляють собою параметризовані програми (5–7) для мікропроцесорних управляючих пристроїв (УП) і декодовані керуючі команди (8–10) для виконавчих механізмів одиниць устаткування.

У результаті циркулюючі в ІВС інформаційні потоки використовуються для створення замкнутих контурів управління, підсистем збору і відображення інформації, а також підсистем, що включають автоматизовані робочі місця диспетчерів, технологів і економістів – організаторів виробництва.

1.3.4. Функціональна структура системи оперативного управління ГВС

Отже, основним елементом виробничих відносин у структурі ГВС стає ГВС, що зв'язує підсистеми проектування, підготовки та організації виробництва безпосередньо з технологічними засобами переробки матеріалів у готову продукцію. Керуючою структурою в ГВС виступає ОТАСУ, основною метою якої є організація оперативного управління матеріальними потоками, тобто досягнення погодженості дій при проходженні одиниць матеріального потоку через обробні ресурси. Тому ОТАСУ часто називають системою оперативного управління матеріальними потоками (СОУ МП).

Розглянемо функціональну структуру і принципи роботи СОУ МП в ГВС (рис. 1.8). СОУ реалізує принцип управління із зворотним зв'язком і складається з наступних основних блоків-модулів:

- оперативно-календарного планування (P1);
- оперативного обліку (P2);
- оперативного контролю (P3);
- оперативної диспетчеризації (P4);
- локального управління устаткуванням (P5);
- оперативного корегування (P6);
- статистичного обліку (P7).

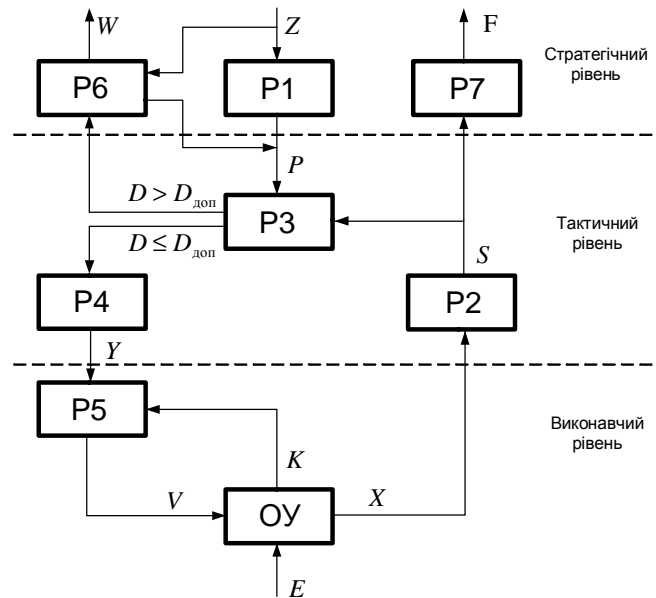


Рис. 1.8. Функціональна структура системи оперативного управління в ГВС

На вхід СОУ надходить цільова вказівка Z , що включає планове завдання у вигляді виробничої програми, технологію обробки, фонди матеріальних ресурсів, що використовуються, і техніко-економічні показники для оцінки ефективності роботи ГВС. Блок P1 на підставі мети Z складає поопераційні календарні плани і формує змінно-добове

завдання P (календарний план) завантаження устаткування.

Блок Р2 здійснює обробку та облік первинної інформації про стан об'єкта управління ОУ (устаткування і матеріальних потоків), що надходить на вхід блока у вигляді сигналів X про зміну стану компонентів ОУ (наприклад, у вигляді сигналів про завершення виконання керуючого завдання Y , передбачених планом P). При цьому блок Р2 робить розпізнавання поточного стану компоненти і формує вектор поточного стану S усього комплексу компонентів (устаткування) ОУ.

Далі в блоці Р3 проводиться аналіз подій, що відбулися в ОУ, і оцінюється рівень відповідності дійсного ходу виробництва запланованому. Для цього визначається ступінь неузгодженості $D = S - P$ (відхилення від планових термінів завершення технологічних операцій). Якщо величина неузгодженості не перевищує допустимої величини $D_{\text{доп}}$ (локальні резерви часу виконання технологічних операцій), то інформація про поточний стан ОУ передається блоку Р4 для вироблення керуючого завдання Y , спрямованої на усунення неузгодженості та виконання чергового завдання з плану P . З цією метою на етапі планування, виходячи з прийнятої стратегії оперативного регулювання, у змінно-добове завдання вноситься оперативна надлишковість по використовуваному устаткуванню і виконуваних технологічних операціях. Відпрацюванням керуючого завдання Y займається блок Р5, який для цього використовує власні набори сигналів K про стан виконання завдання і керуючі команди V (мікрокоманди на виконання елементарних операцій).

Якщо величина неузгодженості перевищує допустиму величину, то управління ГВС передається блоку Р6, який робить перерахунок календарного плану на частину планового періоду, що залишилася, із врахуванням виробничих резервів наданих ресурсів (устаткування, технологічних маршрутів і процесів). За результатами перерахунку приймається рішення про подальший варіант функціонування ГВС:

1) здійснювати роботу ГВС виходячи з перерозрахованого плану у випадку дотримання директивних термінів виконання виробничої програми;

2) інформувати підсистему управління верхнього рівня про неможливість виконання виробничої програми в потрібні терміни без зміни планового завдання і виробничих ресурсів (системні повідомлення W).

Облік і статистична обробка стану S в ОУ для використання на верхньому рівні управління (АСУВ) виробляється у блоці Р7, що переробляє облікові дані про ОУ за зміну і формує узагальнену інформацію F у вигляді даних про число виготовлених деталей, обсяг незавершеного виробництва, завантаження та простої устаткування тощо.

1.4. Організація програмно-математичного забезпечення управління інтегрованим виробництвом

1.4.1. Принципи побудови програмно-математичного забезпечення

Реалізація функцій управління ІВС із використанням ЕОМ здійснюється шляхом розробки і застосування програмно-математичного забезпечення (ПМЗ) комп'ютеризованих систем управління інтегрованим виробництвом (КСУІВ).

Математичне забезпечення включає сукупність математичних методів, моделей і алгоритмів обробки інформації для організації управління, що використовуються при створенні КСУІВ.

Програмне забезпечення – це частина інформаційного забезпечення КСУІВ, що забезпечує прикладну розробку і реалізацію математичного забезпечення на ЕОМ.

Оскільки ІВС будується у вигляді інтегрованого взаємозалежного комплексу функціональних підсистем із складу АСУВ і ГВС, в основу побудови ПМЗ КСУІВ покладені наступні принципи: інтеграції, гнучкості, ієрархічності, типізації.

Під *інтеграцією* розуміється об'єднання різних функцій управління елементів ГВС у єдину систему управління, що координується загальною метою виробництва з одночасним збереженням їх автономності роботи.

Гнучкість базується на програмно-технічному перенастроюванні елементів ГВС на нові умови функціонування як природного, так і випадкового характеру.

Ієрархічність передбачає такий розподіл функцій управління, при якому взаємодія між рівнями переважає над взаємодією в середині рівнів, а вищестоящі рівні вирішують більш великі (глобальні за ступенем важливості та ступенем впливу отриманого результату на функціонування системи) задачі і мають більш тривалий період і меншу частоту прийняття рішень.

Очевидно, що АСУВ та інші функціональні підсистеми верхнього рівня забезпечують роботу ІВС у режимі поділу часу, а елементи ГВС – роботу технологічної підсистеми ІВС в умовах реального масштабу часу, оскільки час ухвалення рішення повинен бути меншим за час виробничого процесу.

Таким чином, швидкість реакції системи на зміни виробничих умов стає основною характеристикою КСУІВ. У той же час розробка ПМЗ КСУІВ є найбільш трудомісткою, а отже, і дорогою частиною у створенні ІВС. В результаті збільшується потреба у створенні стандартних пакетів програм управління виробництвом і зростає ступінь значущості принципу типізації рішень застосовуваних у ПМЗ.

Під *типізацією* ПМЗ розуміється уніфікація задач, структур даних і використовуваних методів і засобів при реалізації заданих функцій управління виробництвом. Відомо, що адаптація типового ПМЗ під конкретне застосування складає 10–15 % вартості розробки

оригінального забезпечення.

Переваги використання типового забезпечення полягають в наступному:

- реалізація випробуваних стандартних рішень;
- високий ступінь придатності до використання в різних умовах функціонування виробництва;
- невисокі вимоги до обслуговуючого персоналу;
- відсутність витрат на розробку і супровід.

Переваги використання оригінального забезпечення наступні:

- точне відображення особливостей застосування;
- відсутність непотрібних функцій;
- простота в адаптуванні та модифікації в рамках використовуваної прикладної області.

Таким чином, при виборі ПМЗ необхідно враховувати можливості технічних засобів, рівень підготовки персоналу, фінансові можливості, а також необхідний рівень автоматизації функцій управління виробництвом і показники основних характеристик КСУІВ, що проектується.

Основними характеристиками КСУІВ є швидкість реакції, ступінь автоматизації, ступінь гнучкості та рівень інтеграції виробництва.

Ступінь автоматизації – це показник, який визначає відношення обсягів робіт, виконуваних без участі і за участю людини.

Ступінь гнучкості – це показник мобільності, тобто обсягу витрат, з якими можна перейти на випуск нової продукції.

Рівень інтеграції – це показник кількості різних виробничих функцій, що поєднуються в єдину систему.

1.4.2. Структура і функції підсистем ПМЗ КСУІВ

Вияткова складність повного набору розв’язуваних у рамках управління ІВС задач і виконуваних функцій роблять необхідним їх групування на підсистеми, підтримувані відповідними модулями і пакетами ПМЗ. Існуючі в даний час реалізації структур ПМЗ управління виробництвом мають відмінності в тому, що розробники дотримуються власних точок зору щодо принципу декомпозиції ПМЗ управління на модулі. Однак, значна частина таких модулів не залежить від конкретної реалізації виробничої системи, на базі введених раніше принципів такі модулі, представляють собою основу типового рішення структур ПМЗ КСУІВ (рис. 1.9).

Характерною рисою сучасного стану ПМЗ КСУІВ є набори модульних програм, призначених для рішення визначених класів задач на різних типах ЕОМ і розрахованих на конкретні вимоги великої кількості споживачів. Тому ПМЗ поділяється на системне і прикладне.

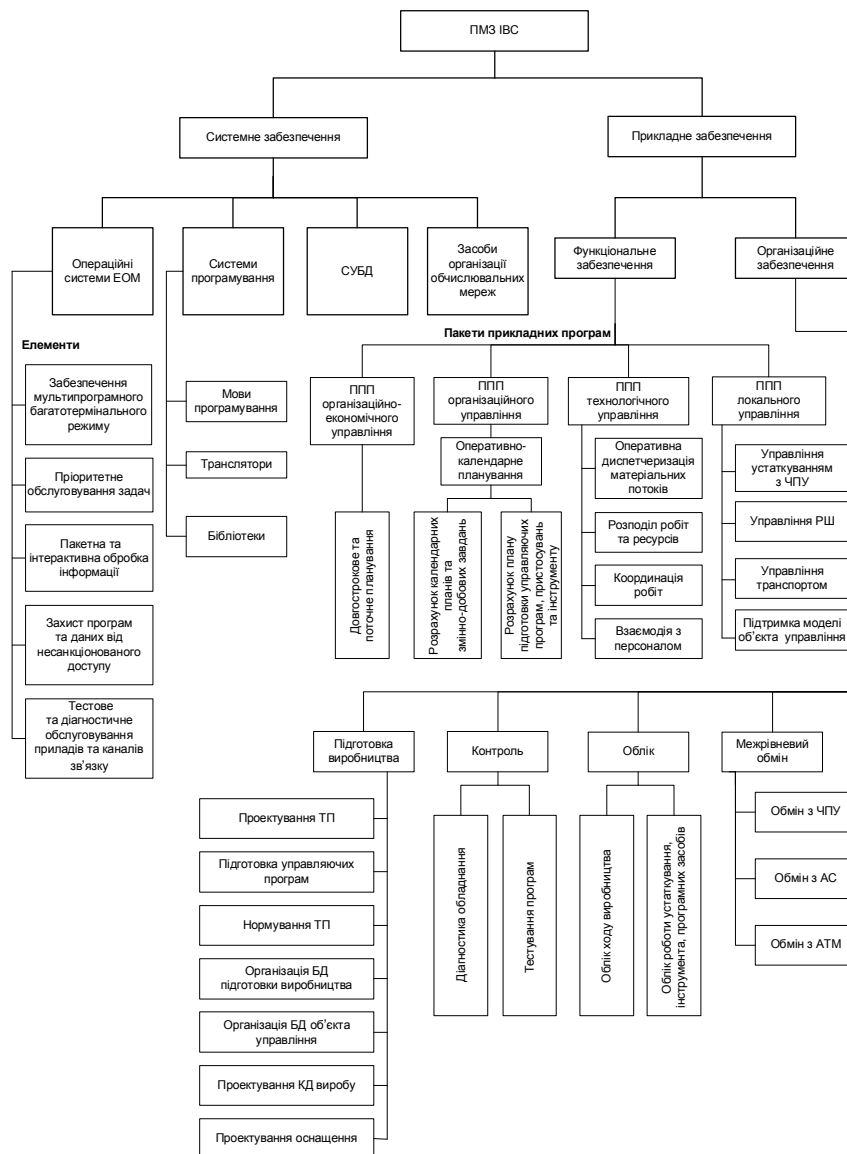


Рис. 1.9. Програмно-математичне забезпечення КСУІВ

Системне забезпечення призначене для управління обчислювальними ресурсами ЕОМ у ІВС і для автоматизації розробки прикладних задач управління. Воно дозволяє об'єднати функціональні підсистеми ІВС у єдиний комплекс, організовувати їх взаємодію, будувати системи доступу до даних, підтримувати режим роботи з багатьма користувачами.

Системне забезпечення включає операційну систему ЕОМ, системи програмування прикладних задач, системи управління базами даних, засоби організації обчислювальних мереж.

Прикладні задачі реалізуються в середовищі операційної системи (ОС). ОС дозволяє виділити обмеження, що накладаються конкретними системними засобами, і визначити функціональні

можливості організації управління.

До системного забезпечення включають також додаткові засоби налаштування математичного та інформаційного забезпечення (баз даних) системи управління на конкретний технологічний об'єкт, а також додаткові засоби з експлуатаційних (тестових і діагностичних) програм, що перевіряють справність устаткування та елементів ЕОМ.

Системне забезпечення тісно зв'язане зі структурою й апаратурою ЕОМ і спрямоване на реалізацію закладених у ній можливостей.

Прикладне забезпечення призначене для реалізації виробничої та інформаційної функцій ІВС на базі рішення прикладних задач управління. Воно функціонує в середовищі засобів системного забезпечення і поділяється за призначенням на функціональне та організаційне.

Основу функціонального прикладного забезпечення складають пакети прикладних програм (ППП) різних рівнів управління в ІВС: PPP економічного управління (довгострокове і поточне планування виробництва), PPP організаційного управління (оперативно-календарне планування для розрахунку календарних планів і змінно-добових завдань випуску деталей, плану підготовки керуючих програм, пристосувань та інструмента), PPP технологічного управління (оперативно-диспетчерське управління, зв'язане з поточним розподілом робіт і ресурсів, координацією їх взаємодії), PPP локального управління (групового управління устаткуванням, складом, транспортом).

Організаційне прикладне забезпечення складається з модулів підсистем:

- міжрівневого обміну з устаткуванням, складом і транспортом;
- оперативно-статистичного обліку ходу виробництва і роботи устаткування, інструмента, оснащення, персоналу;
- тестування програм і діагностики устаткування;
- підготовки виробництва (проектування техпроцесів, підготовки керуючих програм, нормування техпроцесів, проектування оснащення).

1.4.3. Організація налагодження та впровадження програмного забезпечення в ІВС

Досвід показує, що в процесі налагодження алгоритмів і програм виявляється значне число помилок (приблизно 5 % від загального числа команд у програмі), а витрати на їх виявлення і виправлення порівняні з витратами на проектування програмного забезпечення.

Помилки класифікують наступним чином:

- 1) системні – обумовлені неправильним розумінням вимог задачі та умов її реалізації;

2) алгоритмічні – зв'язані з некоректним формулюванням і реалізацією алгоритмів;

3) програмні помилки кодування алгоритмів;

4) технологічні – ті, що виникають в процесах підготовки документації на програму.

Процес налагодження та впровадження програм поділяється на наступні основні етапи:

1) програмне налагодження;

2) системне налагодження;

3) дослідна експлуатація.

Програмне налагодження призначене для індивідуальної перевірки окремих програм на тестових модельних даних, у процесі якої виявляються та усуваються в основному різні алгоритмічні, програмні та технологічні помилки.

Основними функціями програмного налагодження є:

– перевірка відповідності одержуваних результатів програмним специфікаціям;

– перевірка правильності виконання операцій;

– перевірка логіки програми;

– перевірка правильності кодування.

Системне налагодження призначене для перевірки правильності роботи комплексу програм з використанням реальних інформаційних масивів неповного обсягу. У процесі системного налагодження перевіряється еквівалентність логічної схеми комплексу програм її функціональному призначенню. При цьому усувається більшість складних алгоритмічних і системних помилок.

Дослідна експлуатація призначена для перевірки функціонування системи з використанням реальних повнорозмірних масивів даних і в реальному масштабі часу. Передбачається, що на цьому етапі комплекс програм не містить великих системних помилок, а випадки неправильного функціонування виникають через неправильну обробку непередбачених раніше комбінацій вихідних даних.

Контрольні запитання

1. Обґрунтуйте основні фактори і етапи розвитку автоматизації виробничих систем.

2. Розкрийте структуру ГКС в інтегрованому виробництві.

3. Назвіть характерні риси класифікації систем управління для застосування в інтегрованому виробництві.

4. Визначте ієрархію та задачі рівнів управління ІВС.

5. Розкрийте зміст організації багаторівневої ієрархічної системи управління в ІВС.

6. Поясніть особливості організації оперативного управління ГВС.

7. Обґрунтуйте принципи побудови та розкрийте зміст структури і функцій програмно-математичного забезпечення КСУІВ.

8. Розкрийте мету організації процесу відлагодження програмно-математичного забезпечення КСУІВ.

Коли хтось запропонував Аристипу із Кірені задачу і сказав «Розплутай!», Аристип викликнув: «Навіщо, дурень, хочеш розплутати вузол, що, навіть заплутаний, доставляє нам стільки турбот?»

Розділ 2

АЛГОРИТМІЗАЦІЯ ЗАДАЧ УПРАВЛІННЯ ІНТЕГРОВаним ВИРОБНИЦТВОМ

Алгоритмізація – це етап розв’язання задачі, що полягає у знаходженні за її формулюванням методу рішення.

Розділ інформатики, який вивчає методи, прийоми побудови алгоритмів та їх властивостей називається *алгоритмікою*.

Таким чином, під алгоритмізацією управління інтегрованим виробництвом будемо розуміти процес проектування моделей, методів і алгоритмів задач управління ГВС та їх верифікації.

2.1. Автоматизоване планування в інтегрованому виробництві

Як було вказано у розділі 1, два верхніх рівні виробничої системи управління відносяться до організаційно-економічних та організаційно-технічних систем управління.

Так, на першому рівні здійснюється організація спільного функціонування виробничих підрозділів (цехів) для випуску готової продукції в заданій кількості при визначених техніко-економічних показниках, а на другому визначаються показники функціонування окремих підрозділів через встановлення заданої кількості випуску продукції у визначені терміни роботи.

Головна відмінність організаційно-економічних систем від систем технологічного управління 3 і 4 рівнів полягає в характері об’єкта управління. Якщо в системах технологічного управління об’єктом є технологічні одиниці та комплекси устаткування різного призначення, то в організаційно-економічних системах – це колективи людей, які взаємодіють, обслуговують, організують роботу технологічних комплексів. Тому одними з визначальних задач, що забезпечують скоординоване функціонування всього виробництва, є задачі планування різних рівнів управління.

В організаційно-економічних системах в процесі планування на підставі глобальної мети визначають цілі управління всіма підрозділами таким чином, щоб забезпечити досягнення глобальної мети.

Глобальною метою є виконання планового завдання, що регламентує обсяг, номенклатуру, терміни та умови використання

виробничих ресурсів. У процесі планування на основі планового завдання визначаються техніко-економічні показники підприємства в цілому і виробляється їх деталізація по всіх підрозділах, цехах, дільницях, лініях.

В залежності від того, на які періоди складаються плани, загальнозаводське планування (планування на рівні цілого підприємства) поділяють на перспективне і поточне (рис.2.1).



Рис. 2.1. Структура загальнозаводського планування

Перспективне планування визначає терміни і загальні обсяги виробництва в цілому і припускає розробку планів функціонування підприємства на тривалий період (більше 1 року).

Основною формою планування роботи підприємства як економічної одиниці є перспективний план, завдання якого з урахуванням розвитку підприємства, зміни в ресурсах і потребах галузі в його продукції деталізуються й уточнюються на наступному ієрархічному рівні управління – річному плані підприємства.

Перспективний і річний плани розробляються на основі контрольних прогнозних цифр за окремими фінансово-економічними показниками розвитку виробництва, такими як собівартість продукції, рентабельність і прибуток виробництва, ступень амортизації виробничих засобів та віддачі основних виробничих фондів, термін окупності капітальних вкладень та інше. Завдання річного плану підприємства конкретизуються в процесі оперативного планування – при розробці календарних планів.

Поточне планування визначає склад робіт, їх розподіл по підрозділах. Воно поділяється на техніко-економічне (річне, календарне з робівкою по інтервалах періоду) і оперативне.

Техніко-економічне планування деталізує показники перспективного плану на даний період і коригує їх відповідно до вимог планового завдання підприємства.

Найважливішою задачею техніко-економічного планування є встановлення основних показників по випуску і реалізації продукції, праці і заробітній платі, собівартості продукції, а також контроль за

їхнім виконанням, тобто розрахунок техпромфінплану. За допомогою техпромфінплану поєднуються матеріальні, трудові і фінансові плани, плани розвитку потужності і т.і., а також забезпечується їхнє балансове ув'язування з виробничою програмою. Вихідні дані для розрахунку техпромфінплану беруться з планів випуску продукції в грошовому і натуральному вираженні, зниження трудомісткості виробів, собівартості, постачань і реалізації продукції, сортності, планів підготовки кадрів і впровадження нової техніки, з лімітів по праці і капіталовкладенням.

Основна частина техпромфінплану це головні показники виробничо-господарської діяльності підприємства, а також виробництва і реалізації продукції. Директивною його частиною є річна (загальнозаводська) виробнича програма підприємства. Вона містить перелік продукції, що підлягає випуску і реалізації в плановому періоді. План виробництва дає кількісну характеристику усієї виробничо-господарської діяльності підприємства і є основою для розробки наступних частин техпромфінплану.

Річна виробнича програма складається в натуральному, тобто кількісному, і грошовому вираженнях з розбивкою завдання по випуску і реалізації продукції по кварталах. Критеріями оптимальності при розробці виробничої програми є: найбільш повне і своєчасне задоволення споживачів; максимізація продуктивності праці; мінімізація витрат дорогих і дефіцитних матеріалів, капітальних витрат, собівартості продукції і т.і.

У техпромфінплані важливими для всіх наступних розрахунків є планові техніко-економічні нормативи і норми. У техпромфінплані застосовуються нормативи і норми витрати матеріальних ресурсів на випуск основної продукції, на ремонтно-експлуатаційні нестатки, на виготовлення інструмента й оснащення, а також засобів механізації й автоматизації, на створення запасів сировини і матеріалів, на незавершене виробництво. На основі використання техніко-економічних нормативів і норм виконують планові розрахунки, що полягають в оптимізації планів виробництва і реалізації продукції.

На основі техпромфінплану підприємства розробляються кількісні і якісні показники роботи окремих виробництв, дільниць, цехів на місяць, квартал, рік в процесі оперативно-календарного планування. У цих планових документах містяться показники виробничої програми ланок основного і допоміжного виробництва, росту продуктивності праці, зниження трудомісткості і ліміти по праці.

Календарне планування це визначення завдання на виробництво і випуск продукції на протязі відрізків часу, що відповідають тривалості встановленого планово-облікового періоду. Таким чином, при календарному плануванні у відповідності до річного завдання виробничої програми забезпечується розробка календарних (квартальних, місячних, добових) план-графіків цехам, дільницям,

устаткуванню.

Оперативне планування це диференціювання (розподіл) виробничого завдання за місцем виконання його складових з призначенням термінів їх початку і завершення. При оперативному плануванні конкретизуються показники техніко-економічного плану на короткі інтервали планового періоду (місяць, декада, доба) і визначаються дії, необхідні для їх виконання у вигляді виробничого завдання. Оперативні плани-завдання розробляються з урахуванням результатів виконання плану на попередніх інтервалах. Оперативне планування входить до складу оперативного управління, яке передбачає облік термінів виконання запланованих видів робіт, прийняття рішень щодо недопущення відхилень від запланованого ходу виробництва.

Оперативне планування тісно пов'язане з календарним і часто за сферою (масштабом, місцем) виконання поділяється на: міжцехове і всередині-цехове. При цьому для міжцехового планування характерно розв'язання саме задачі календарного планування, а всередині-цехове – встановлення виробничих завдань дільницям, устаткуванню на заданий плановий період. Тому, функцію оперативно-календарного планування визначають як деталізацію за часом і місцем виконання плану виробництва, отриманого після техніко-економічного планування.

Таким чином, незалежно від типу виробництва, планування здійснюється на різних рівнях управління, тобто на загальнозаводському, міжцеховому та всередині-цеховому.

Загальнозаводське планування – це встановлення загальних планів виробництва за обсягами на визначений період діяльності.

Міжцехове планування – це встановлення скоординованих складових загальних планів виробництва – календарних планів підрозділів.

Всередині-цехове планування – це встановлення планових завдань на визначений період дільницям, змінам з доведенням розкладів роботи до робочих місць, устаткування.

Таким чином, задачі автоматизованого планування можна класифікувати за наступними ознаками належності до таких групи:

1) за об'єктом або рівнем управління – загальнозаводське планування, міжцехове планування (планування роботи великих виробничих підрозділів), всередині-цехове планування (планування завдань робочим місцям);

2) за періодом планування – перспективне планування (розробка планів розвитку і функціонування виробництва на період більше 1 року), техніко-економічне планування (розробка загальнозаводських та цехових планових завдань на визначений період до 1 року), календарне планування (розробка календарних план-графіків на планові періоди – квартал, місяць, доба), оперативне планування (розробка виробничих завдань підрозділам, робочим місцям на період від 1 місяця до зміни);

3) за видом отриманих результатів – з просторовим впорядкуванням за устаткуванням, підрозділами обсягів і номенклатури, з часовим впорядкуванням комплексу планових робіт (номенклатури випуску).

В цілому в комплексі задач планування інтегрованого виробництва можна виділити такі класи типових задач планування, як:

- планування обсягу і номенклатури випуску продукції (охоплює задачі перспективного і техніко-економічного планування загальнозаводського рівня з просторовим впорядкуванням результатів);

- об'ємно-календарного планування (охоплює задачі перспективного і техніко-економічного планування загальнозаводського і міжцехового рівня з часовим впорядкуванням результатів);

- оперативно-календарного планування (охоплює задачі оперативного планування всередині-цехового рівня з часовим і просторовим впорядкуванням результатів).

2.2. Типові задачі планування в інтегрованому виробництві та методи їх розв'язання

2.2.1. Дослідження задачі планування обсягу і номенклатури виготовлення продукції

Формулювання задачі. При розв'язанні задач організаційно-економічного управління виробництвом, одним з основних факторів, що регламентують діяльність промислового підприємства, є збільшений розрахунок виробничих потужностей і завантаження устаткування, результатом якого є виробнича програма обсягу виготовлення продукції.

В умовах масового і серійного виробництва ефективним методом такого розрахунку, який забезпечує максимальне завантаження устаткування, є лінійне програмування, орієнтоване на побудову лінійних оптимізаційних моделей виробництва. Моделі, які використовуються у цих розрахунках, є статичними, адже вони призначені для аналізу керуючих рішень, які поширюються на єдиний, попередньо визначений відрізок часу.

Сфера застосування моделей та методів такого роду у даний час охоплює широкий спектр функцій організаційного управління [10].

Розглянемо приклади типових задач виробничого планування обсягу і номенклатури випуску продукції на двох рівнях управління підприємством: адміністративного управління (задача управління збутом продукції) і управління підрозділами (задача оптимального розподілу ресурсів).

Задача управління збутом продукції. Як приклад задачі управління збутом продукції розглянемо спрощене планування зовнішньоторговельних операцій.

Виробник має перед собою мету: виробити вітчизняний товар,

який користується попитом у Європі, вивезти та продати його за валюту, на зароблені кошти придбати товар у Європі, який користується попитом на Україні, перевезти та продати його оптом чи в роздріб, одержавши найбільший прибуток. При цьому він повинен виходити із наступних умов:

- 1) витрати виробника на виготовлення вітчизняних товарів встановленої номенклатури, враховуючи ліцензію на вивіз – $C_i, i = \overline{1, n}$;
- 2) ціни на вітчизняні товари при продажу оптом за валюту у Європі – $CP_i, i = \overline{1, n}$;
- 3) максимально доступні розміри партій вітчизняних товарів – $P_i, i = \overline{1, n}$;
- 4) фрахтові ставки при перевезенні вітчизняних товарів – $F_i, i = \overline{1, n}$;
- 5) ціни на товари закордонного виробництва встановленої номенклатури при купівлі оптом за валюту у Європі – $CE_j, j = \overline{1, m}$;
- 6) фрахтові ставки при перевезенні іноземних товарів – $FE_j, j = \overline{1, m}$;
- 7) ціни на європейські товари при продажу за гривні в роздріб – $S_j, j = \overline{1, m}$;
- 8) коефіцієнт перерахунку оптової ціни в роздрібну при продажу – K ;
- 9) ставка податків на продаж – NP ;
- 10) валютний курс гривні – CCE ;
- 11) коефіцієнти загальних втрат при роботі з вітчизняними товарами – $D_i, i = \overline{1, n}$;
- 12) коефіцієнти загальних втрат при роботі з іноземними товарами при продажу їх оптом M_j^1 та в роздріб $M_j^2, j = \overline{1, m}$;
- 13) максимальні розміри партій іноземних товарів, які можна реалізувати у рамках зазначених втрат – $B_j, j = \overline{1, m}$;
- 14) сума грошових коштів, які має виробник – Ω_1 ;
- 15) сума одержаного кредиту – Ω_2 ;
- 16) процентна ставка з кредиту – KR ;
- 17) обмеження на вивезення валюти з Європи – T ;
- 18) обмеження на вивезення валюти з України – TU .

Необхідно визначити обсяг виробництва вітчизняних товарів $X_i, i = \overline{1, n}$ та обсяг продажу іноземних товарів оптом $Y_j^1, j = \overline{1, m}$ та в роздріб $Y_j^2, j = \overline{1, m}$, завдяки яким буде одержаний найбільший прибуток. Для цього перерахуємо всі ціни в гривнях і визначимо

цільову функцію (ЦФ) як прибуток від всього комплексу зовнішньоторговельних операцій, а обмеження задачі подамо як умови виконання всіх операцій з виробництва, перевезення, компенсації втрат, продажу, придбання, тощо.

В результаті математичний запис задачі має наступний вигляд:

$$\max \left\{ \begin{aligned} & \sum_{j=1}^m (S_j(1-NP)K - S_j \mathbf{g} M_j^1 \mathbf{g} K - FE_j - CE_j) Y_j^1 + \\ & \sum_{j=1}^m (S_j(1-NP) - S_j \mathbf{g} M_j^2 - FE_j - CE_j) Y_j^2 + \\ & \sum_{i=1}^n (CP_i - C_i - F_i - CP_i \mathbf{g} D_i) X_i + \end{aligned} \right\}$$

за умов

$$\sum_{i=1}^n (C_i + F_i + CP_i \mathbf{g} D_i) X_i \leq \Omega_1 + \Omega_2 \quad - \quad \text{обмеження коштів на}$$

виробництво, перевезення і компенсації втрат;

$$\sum_{i=1}^n CP_i \mathbf{g} X_i - \sum_{j=1}^m CE_j \mathbf{g} Y_j^1 - \sum_{j=1}^m CE_j \mathbf{g} Y_j^2 \geq 0 \quad - \quad \text{обмеження коштів на}$$

придбання іноземних товарів

$$\sum_{i=1}^n CP_i \mathbf{g} X_i - \sum_{j=1}^m CE_j \mathbf{g} Y_j^1 - \sum_{j=1}^m CE_j \mathbf{g} Y_j^2 \leq T \quad - \quad \text{обмеження на вивезення}$$

валюти з Європи;

$$\sum_{i=1}^n C_{i1} \mathbf{g} X_i \geq \Omega_1 + \Omega_2 - TU \quad - \quad \text{обмеження на вивезення валюти з}$$

України;

$$X_i \leq P_i, i = \overline{1, n} \quad - \quad \text{обмеження на максимальний обсяг партій}$$

вітчизняних товарів;

$$Y_j^1 + Y_j^2 \leq B_j, j = \overline{1, m} \quad - \quad \text{обмеження на максимальний обсяг партій}$$

іноземних товарів.

Задача розподілу ресурсів. На підприємстві, яке виготовляє неоднорідну продукцію, для формування виробничої програми необхідно визначити, якими повинні бути рівні виробництва кожного продукту за встановлений час планового періоду. Ці рівні обмежені наступними технологічними умовами:

1) виробництво продукції утворюється на базі n різних технологічних процесів;

2) технологічні процеси орієнтовані на одержання продукції з наявних в системі m матеріалів;

3) витрати ресурсів системи, які пов'язані з кожним із технологічних процесів, визначаються трудовитратами

$D_i, i = \overline{1, n}$ (вимірними в людино-тижнях), а також кількістю матеріалів $A_{ij}, i = \overline{1, n}, j = \overline{1, m}$, (в одиницях ваги), які витрачаються за встановлений час планового періоду;

4) загальні трудовитрати виробництва повинні не перевищувати B^1 , а матеріальні запаси - $B_j^2, j = \overline{1, m}$.

У рамках цих обмежень критерієм оцінки ефективності організаційного управління виробництвом є отримання максимального прибутку (або мінімальних витрат) $C_i, i = \overline{1, n}$. Необхідно визначити обсяг виробництва $X_i, i = \overline{1, n}$ для кожного технологічного процесу.

Використовуючи виробничо-економічні показники і всі встановлені обмеження формується наступна лінійна модель виробництва:

$$\max \left\{ \sum_{i=1}^n C_i \cdot X_i \right\}$$

за умов

$$\sum_{i=1}^n D_i \cdot X_i \leq B^1;$$

$$\sum_{i=1}^n A_{ij} \cdot X_i \leq B_j^2, j = \overline{1, m}.$$

Іншим варіантом задачі розподілу ресурсів є розрахунок завантаження устаткування, який виконується за такими початковими даними:

- 1) B_i – пропускна здатність i - ої групи устаткування ($i = \overline{1, m}$);
- 2) A_{ij} – трудомісткість j - го виробу ($j = \overline{1, n}$), виготовленого на обладнанні i - ої групи;
- 3) S_j – собівартість виробів;
- 4) C_j – вартість виробів;
- 5) X_j – шукана програма випуску;
- 6) обмеження з пропускної здатності устаткування:

$$\sum_{j=1}^n A_{ij} \cdot X_j \leq B_i, i = \overline{1, m}.$$

В залежності від економічних вимог використовуються різні критерії оцінки шуканої програми випуску при задоволенні обмеження з пропускної здатності:

- 1) найменший сумарний розмір простоїв:

$$\min \left\{ \sum_{i=1}^m T_i \right\},$$

$$\text{де } T_i = B_i - \sum_{j=1}^n A_{ij} \cdot X_j;$$

2) максимізація випуску у вартісному вираженні:

$$\max \left\{ \sum_{j=1}^n C_j \cdot X_j \right\};$$

3) максимізація прибутку:

$$\max \left\{ \sum_{j=1}^n (C_j - S_j) \cdot X_j \right\}.$$

Метод розв'язання задачі. Розглянуті задачі відносяться до єдиного математичного класу задач – лінійного програмування (ЛП).

Симплекс-метод є найбільш поширеним та універсальним обчислювальним методом, який використовується для розв'язання будь-яких задач ЛП за допомогою ЕОМ. Ідея методу полягає у послідовному проходженні по базисних рішеннях опорних планів задачі доти, доки не буде одержане оптимальне рішення (іншими словами, у послідовному покращенні планів задачі за визначеним критерієм).

Розглянемо процес підготовки початкових даних і алгоритм розв'язання задачі ЛП табличним симплекс-методом.

Математична модель задачі ЛП має одну із наступних лінійних форм:

$$\max \left\{ \sum_{j=1}^n C_j \cdot X_j \right\} \quad (2.1)$$

за умов:

$$\sum_{j=1}^n A_{ij} \cdot X_j \leq B_i, \quad i = \overline{1, m}; \quad (2.2)$$

$$X_j \geq 0, \quad j = \overline{1, n} \quad (2.3)$$

або

$$\min \left\{ \sum_{j=1}^n C_j \cdot X_j \right\} \quad (2.4)$$

за умов:

$$\sum_{j=1}^n A_{ij} \cdot X_j \geq B_i, \quad i = \overline{1, m} \quad (2.5)$$

$$X_j \geq 0, \quad j = \overline{1, n}. \quad (2.6)$$

Перед розв'язанням задачі необхідно попередньо виконати такі процедури:

1) привести математичну модель до канонічного вигляду, тобто обмеження (2.2), (2.5) перетворити на рівняння за допомогою додаткових змінних;

2) визначити початковий прийнятний (невід'ємний) базисний розв'язок задачі (у випадку обмежень (2.5) ввести штучні змінні);

3) ввести в початкову симплекс-таблицю такі параметри, які відповідають початковому базисному розв'язку (рис.2.2):

– вагові коефіцієнти при змінних X_j в ЦФ (рядок C_j);

– змінні X_j , які входять до наявного базису;

– значення базисних змінних $X_i = B_i = A_0$ (стовпець A_0);

– елементи A_{ij} , $i = \overline{1, m}$, $j = \overline{1, n}$, матриці умов задачі $A[m \times n]$ (стовпці A_1, A_2, \dots, A_n);

– оцінки D_j , $j = \overline{1, n}$, що відповідають стовпцям A_1, A_2, \dots, A_n і визначаються за формулою:

$$D_j = \sum_{i=1}^m C_i \cdot A_{ij} - C_j, \quad j = \overline{1, n}. \quad (2.7)$$

				C_j		
X_i	A_0	A_1				A_n
				A_{ij}		
				D_j		

Рис. 2.2. Структура початкової симплекс-таблиці

Далі, використовуючи алгоритм симплекс-методу, знаходять оптимальне рішення плану – базисний розв'язок X_i .

Алгоритм симплекс-методу (для (2.1)) має вигляд [5]:

1. Заповнити початкову симплекс-таблицю.

2. Якщо $D_j \geq 0$ для всіх $j = \overline{1, n}$, то отриманий план оптимальний.

3. Якщо є $D_j < 0$ і в стовпці A_j всі елементи $A_{ij} \leq 0$, то ЦФ необмежена зверху умовами задачі і знайти оптимальний план неможливо.

4. Якщо у стовпцях A_j , що відповідають від'ємним оцінкам D_j , існує хоча б один $A_{ij} > 0$, то можливий перехід до нового, кращого плану.

5. Стовпець A_k , який необхідно ввести до базису для покращення плану, визначається за найменшим $D_j < 0$. Стовпець, в якому міститься ця оцінка, називається напрямним.

6. Стовпець A_r , який необхідно вивести з базису, визначається за відношенням $r: \min \{A_{i0} / A_{ik}, i = \overline{1, m}\} = A_{r0} / A_{rk}, (A_{ik} \neq 0)$. Рядок r називається напрямним. Елемент A_{rk} , який стоїть на перехресті

напрямого рядка і стовпця, називається напрямним елементом.

7. Заповнюється нова симплекс-таблиця, яка відповідатиме новому базисному розв'язку:

$$A_{ij}(l+1) = A_{ij}(l) - A_{rj}(l) \cdot A_{ik}(l) / A_{rk}(l) \quad \text{при } i \neq r; \quad (2.8)$$

$$A_{ij}(l+1) = A_{rj}(l) / A_{rk}(l) \quad \text{при } i = r; i = \overline{1, m+1}; j = \overline{0, n},$$

де l – номер ітерації.

Значення D_j визначається одним з двох способів:

а) як кожний елемент таблиці за виразом (2.8);

б) за формулою (2.7).

Процес обчислень завершується, якщо знайдене оптимальне рішення або коли функція буде необмежена.

Особливості застосування методу:

1. Якщо за початковий базис беруть базис із вільних змінних, для яких $C_i = 0$, то оцінки для всіх небазисних змінних дорівнюють

$$D_j = A_{0j} = -C_j, \text{ а відповідні значення } A_{ij} = \sum_{i=1}^m C_i \cdot X_i = 0.$$

2. При розв'язанні задачі з математичною моделлю у формі на мінімум в базис вводиться стовпець з найбільшою додатною оцінкою.

3. Відсутність стовпців з $D_j < 0$ (при розв'язанні задачі з математичною моделлю у формі на максимум) або з $D_j > 0$ (при розв'язанні задачі з математичною моделлю у формі на мінімум) є ознакою оптимальності відповідного базисного рішення.

4. Якщо є хоча б одна від'ємна (задача максимуму) або додатна (задача мінімуму) оцінка для небазисного вектора, а його стовпець містить лише від'ємні елементи, то в області прийнятних рішень ЦФ необмежена.

Аналіз задачі на чутливість. Визначення оптимальних значень виробничої програми не завжди є достатньою умовою для формування виробником правильного керуючого рішення при організації виробництва. В більшості випадків формулюється задача дослідження впливу на значення одержуваного прибутку (або витрат) збільшення кожного з використаних ресурсів, вдосконалення того чи іншого технологічного процесу, зміни вартості використовуваної сировини, що впливає на прибутковість виробничо-технологічних процесів. Це означає, що необхідно знати, в якому інтервалі можна змінювати вхідні параметри моделі без істотного відхилення від знайденого оптимуму і без порушення структури базису, що формує оптимальне рішення. Дослідження, що проводяться в рамках такої задачі, називаються аналізом моделі на чутливість.

Нехай в задачі ЛП $B = [B_i]$ – обмеження значень ресурсів, а $dB = [dB_i]$ – прийнятні варіації цих ресурсів. Дослідимо їх вплив на зміну оптимального значення ЦФ $L(X_{opt}(B)) = L(B)$. Якщо прийняти, що $B = B + dB$, то нове оптимальне значення ЦФ буде рівним $L(B + dB)$.

Обчисливши приріст $dL(B + dB) = L(B + dB) - L(B)$, знайдемо границю співвідношення:

$$\lim_{dB_i} \frac{dL(B + dB)}{dB_i} = \frac{dL(B)}{dB_i} - \text{частинна похідна ЦФ по змінній } B_i.$$

За теоремою подвійності одержимо:

$$C \cdot X_{onm} = L(X_{onm}) = L_{подв}(Y_{onm}) = B \cdot Y_{onm},$$

де Y_{onm} – оптимальний розв’язок подвійної задачі ЛП;

$L_{подв}$ – ЦФ подвійної задачі.

Звідси $dL(B)/dB_i = Y_{i\ onm}$, що визначає максимальне значення приросту ЦФ прямої задачі при зміні i -го ресурсу B_i на 1 одиницю, а саме: чим більше $Y_{i\ onm}$, тим більша чутливість ЦФ прямої задачі до зміни вільних членів B_i в оптимальному розв’язку X_{onm} . Якщо $Y_{i\ onm} = 0$, то це означає, що відповідний ресурс B_i не є суттєвим для моделі, тобто зміна його значення на 1 одиницю не призведе до зміни оптимального значення ЦФ і отриманого розв’язку X_{onm} . Це обумовлює перетворення

i -го обмеження прямої задачі у строгу нерівність $\sum_{j=1}^n A_{ij} \cdot X_{j\ onm} < B_i$,

оскільки ресурс B_i знаходиться у надлишку на величину:

$$\Delta_i = B_i - \sum_{j=1}^n A_{ij} \cdot X_{j\ onm}.$$

Таким чином, при проведенні аналізу моделі на чутливість можливе встановлення таких результатів:

1) якщо оптимальне рішення задачі подвійності $Y_{i\ onm} = 0$, то відповідний ресурс прямої задачі B_i використовується частково і його

значення може бути зменшене до величини $\sum_{j=1}^n A_{ij} \cdot X_{j\ onm}$;

2) якщо оптимальне рішення задачі подвійності $Y_{i\ onm} \neq 0$, то при зміні відповідного ресурсу B_i на 1 одиницю ЦФ прямої задачі одержить приріст на величину $Y_{i\ onm}$.

Слід зазначити, що знаходження значення $Y_{i\ onm}$ можливе через значення D_j оцінок симплекс-таблиці оптимального розв’язку прямої задачі, тобто $Y_{i\ onm} = D_{n+i}$.

2.2.2. Дослідження задачі календарного планування виробничої системи

Організація оперативного управління підприємством неможлива без деталізації виробничої програми випуску продукції за часовими інтервалами в межах встановленого планового періоду. Реалізація цієї функції здійснюється задачею календарного планування, результатом розв’язку якої є часове упорядкування комплексу запланованих робіт програми. Часове упорядкування виражається у визначені строків початку та завершення виконання робіт, тобто календарний план

визначає, скільки продукції необхідно виготовити у кожному інтервалі встановленого періоду. У ГВС оперативний плановий період, як правило, не перевищує місячного терміну, а строками запуску–випуску є наступні часові інтервали: декади, тижні або дні.

В даний час на підприємствах звичайно прийняті наступні методи оперативно-календарного планування: по-детальне, добо-комплектне, машино-комплектне [4]. Такий розподіл визначається відповідними календарно-плановими нормативами, що повинні бути єдиними для всього підприємства. Однак різні цехи і дільниці можуть істотно відрізнятися календарно-плановими нормативами.

Наприклад, для крупномеханічних цехів (дільниць) характерна велика тривалість виробничого циклу, багатоопераційність, мала річна номенклатура. Для дрібномеханічних цехів (дільниць) характерна велика спеціалізація робочих місць, мала тривалість виробничого циклу і велика річна номенклатура. Тому для дрібномеханічних цехів (дільниць) в основному формуються об'ємно-календарні плани, а для крупномеханічних цехів (дільниць) – по-операційні календарні плани, однак можливо і змішане планування.

Об'ємно-календарний план – це завдання цеху (дільниці) на виробництво і випуск продукції протягом відрізків часу, що відповідають тривалості планово-облікового періоду, що обґрунтований розрахунками на рівномірність завантаження груп взаємозамінного устаткування. *По-операційний календарний план* – це гранично диференційоване виробниче завдання цеху (дільниці), у якому зазначені конкретні терміни початку і закінчення окремих операцій технологічного процесу по різних партіях деталей на відповідному устаткуванні з урахуванням дотримання крайніх термінів їхнього випуску. При змішаному плануванні планово-економічний відділ підприємства визначає номенклатуру й обсяг виробництва, а по-операційне планування здійснюється цеховими службами. Як правило, на таких підприємствах діють машино-комплектні системи.

Система оперативно-календарного планування забезпечує розробку відповідно до річного плану місячних, змінно-добових календарних планів-графіків для окремих цехів, дільниць, видачу детальних завдань дільницям, робочим центрам і формування переліку робіт і програм для всього технологічного устаткування по випуску того чи іншого продукту виробництва. Тому весь перелік робіт з оперативно-календарного планування при використанні будь-якої системи планово-облікових одиниць складаються з двох взаємозалежних етапів: календарного планування (попередніх розрахунків планів-графіків, складання календарних графіків) і поточного оперативного планування.

Розв'язання задач календарного планування характерне для міжцехового планування. Перелік типових задач календарного планування наведено в табл. 2.1.

Таблиця 2.1

Задачі календарного планування на рівні міжцехового планування

Найменування задачі	Призначення задачі
Розрахунок календарно-планових нормативів по цехах і дільницям виробництва	Визначення нормативів руху деталей, складальних одиниць; розмір партії; тривалість циклу виготовлення; випередження запуску і випуску; періодичність запуску
Розрахунок нормативних заділів, нормативних матеріалів по типорозмірах і комплектуючих виробів у незавершеному виробництві	Визначення технічно обґрунтованих нормативами матеріалів по типорозмірах і комплектуючих виробів для створення нормативних заділів у незавершеному виробництві
Розрахунок перспективного календарного графіка запуску партій деталей у першому цеху по маршруті	Проведення в службах матеріально-технічного постачання заходів щодо завчасного забезпечення виробництва матеріалами і комплектуючими виробами; проведення в службах технологічної підготовки виробництва заходів щодо забезпечення виробництва технологічним оснащенням
Розрахунок подетальної виробничої програми запуску і випуску на квартал по цехах і виробничих дільницях	Складання квартальної виробничої програми, використовуваної цехами і службами виробництва для своєчасного і цілеспрямованого проведення робіт з технічної підготовки виробництва
Розрахунок завантаження устаткування по квартальній оперативній програмі для цехів і дільниць	Виявлення «вузьких» місць і розробка заходів щодо їхньої ліквідації у виробництві
Розрахунок календарного плану-графіка запуску і випуску партій деталей, складальних одиниць по цехах і дільницях	Встановлення термінів запуску і випуску партій деталей, складальних одиниць

Отже, математичною формою представлення задач даного класу є лінійна або лінійно-дискретна оптимізаційна модель, а методологією розв'язання – лінійне або лінійне цілочисельне програмування (ЛЦП).

Формулювання задачі. Номенклатурний список продукції, що

виробляється за плановий період, складається з n найменувань ($j = \overline{1, n}$). Плановий період включає T часових інтервалів ($k = \overline{1, T}$). У виробництві використовується m видів ресурсів ($i = \overline{1, m}$). Ресурсами можуть бути групи устаткування, матеріальні ресурси, групи спеціалістів тощо.

Ресурси, які надходять у виробництво, вважаються заданими і характеризуються:

1) технологічними умовами – нормативними витратами на виготовлення продукції $A = [A_{ij}]$, де A_{ij} – обсяг i -го ресурсу, необхідного для виготовлення деталей j -го найменування (нормативна трудомісткість виготовлення j -ої деталі на i -му обладнанні; нормативні витрати i -го виду матеріалів на виготовлення j -ої деталі);

2) організаційними умовами – нормативними запасами ресурсів у k -му інтервалі $B_k = [B_{ki}]$, де B_{ki} – обсяг i -го ресурсу в k -му інтервалі (нормативний фонд часу роботи i -го устаткування у k -му інтервалі; надходження i -го виду матеріалів у k -му інтервалі).

Перед підрозділами формулюється задача виконання виробничої програми за обсягом випуску продукції $P = (P_j | j = \overline{1, n})$ за плановий період так, щоб своєчасно постачати деталі відповідно до зовнішніх потреб $F_k = (F_{kj} | j = \overline{1, n})$ в разі міжцехового планування (наприклад, потреб складального виробництва або умов постачання продукції) або оптимізувати витрати чи прибуток $C = (C_j | j = \overline{1, n})$ від реалізації продукції в разі загальнозаводського планування, де P_j – плановий обсяг випуску деталей j -го найменування; F_{kj} – потреби деталей j -го найменування у k -му інтервалі, C_j – витрати (прибуток) від реалізації деталей j -го найменування.

Формально подамо наведену задачу за допомогою наступної лінійної оптимізаційної моделі.

Стан виробництва у k -му інтервалі будемо задавати вектором $X_k = (X_{kj} | j = \overline{1, n})$, компонента якого X_{kj} – випуск деталей j -го найменування у k -му інтервалі. Тоді обсяг використаного i -го ресурсу у k -му інтервалі не повинен перевищувати встановленого значення норми, тобто

$$\sum_{j=1}^n A_{ij} \cdot X_{kj} < B_{ki}, \quad i = \overline{1, m}, \quad k = \overline{1, T}. \quad (2.9)$$

Якщо ресурси, які використовуються у виробництві, мають здатність накопичення та переносу у наступні часові інтервали, то обмеження на використання ресурсів, що надаються, будуть мати вигляд:

$$\sum_{j=1}^n \sum_{l=1}^k A_{ij} \cdot X_{lj} \leq \sum_{l=1}^k B_{li}, \quad i = \overline{1, m}, \quad k = \overline{1, T}, \quad (2.10)$$

тобто обсяг i -го ресурсу, використаного за k інтервалів, не повинен перевищувати обсягу, який надійшов до k -го інтервалу.

Таким чином, обмеження (2.9), яке не враховує використання ресурсів у попередні інтервали, доцільно використовувати у випадку часового обліку ресурсів (фонд часу устаткування), а обмеження (2.10) – при кількісному обліку ресурсів (витрати матеріалів).

Але якщо не встановлені організаційні умови витрат ресурсів за час планового періоду, то обмеження на їх використання повинно мати вигляд:

$$\sum_{j=1}^n \sum_{k=1}^T A_{ij} \cdot X_{kj} \leq \sum_{k=1}^T B_{ki}, \quad i = \overline{1, m}$$

або

$$\sum_{j=1}^n \left(A_{ij} \cdot \sum_{k=1}^T X_{kj} \right) \leq B_i, \quad i = \overline{1, m},$$

де

$$B_i = \sum_{k=1}^T B_{ki}.$$

Кінцевий випуск продукції за умовами задачі повинен дорівнювати або задовольняти плану виробництва, тобто

$$\sum_{k=1}^T X_{kj} \geq P_j, \quad j = \overline{1, n}.$$

Якщо врахувати, що виготовлена продукція безпосередньо даною виробничою одиницею не використовується, то

$X_j = (X_{kj} | k = \overline{1, T})$ – невід’ємна послідовність, тобто $X_{kj} \geq 0, k = \overline{1, T}, j = \overline{1, n}$.

Крім того, на значення X_{kj} можуть бути застосовані обмеження на розмір отриманої продукції, тобто встановлена дискретність значень обсягу.

Календарний план випуску продукції $(X_{kj} | k = \overline{1, T})$, який задовольняє наведеним обмеженням, називають прийнятним. В реальних виробничих умовах існує кінцева, але достатньо велика множина прийнятних планів випуску, серед яких необхідно вибрати найкращий з точки зору максимального задоволення потреби $(F_k | k = \overline{1, T})$ або прибутку від реалізації $(C_j | j = \overline{1, n})$. Тому як критерій задачі необхідно розглядати цільову функцію мінімізації сумарного по інтервалах та деталях відхилення випуску від потреби:

$$\min \sum_{k=1}^T \sum_{j=1}^n (F_{kj} - X_{kj}),$$

або максимізації загального прибутку:

$$\max \sum_{k=1}^T \sum_{j=1}^n C_j \cdot X_{kj}.$$

Якщо перевиробництво неприпустиме (випуск не повинен перевищувати потреби), то у виразі критерію значення відхилення випуску від потреби необхідно врахувати за модулем $(|F_{kj} - X_{kj}|)$.

Метод розв'язання задачі. Метод “гілок та границь” належить до групи комбінованих методів дискретного програмування і є одним з найбільш поширених методів, які використовуються при розв'язанні задач ЛЦП.

Реалізація цього методу полягає у послідовному розгалуженні початкової множини рішень на дерево підмножин з визначенням рішень в усіх підмножинах, доки не буде знайдено шукане, яке задовольняє умові цілочисельності (дискретності).

Математичне формулювання задачі ЛЦП, до якої застосовують метод має такий вигляд:

$$F(X) = \max \left\{ \sum_{j=1}^n C_j \cdot X_j \right\} \quad (2.11)$$

за умов:

$$\sum_{j=1}^n (A_{ij} \cdot X_j) \leq B_i, \quad i = \overline{1, m}; \quad (2.12)$$

$$X_j \geq 0, \quad j = \overline{1, n}; \quad (2.13)$$

$$X_j - \text{цілі числа.} \quad (2.14)$$

Процес знаходження оптимального рішення починають із розв'язання неперервної задачі ЛП. Якщо одержаний при цьому оптимальний план X_o не задовольняє умові (2.14), то значення ЦФ $F(X_o)$ дає верхню оцінку Ω для шуканого рішення.

Далі використовують багатоітераційну процедуру розгалуження (розбиття множини прийнятних рішень), перерахунку оцінки та перевірку умови цілочисельності.

Схема алгоритму ітераційної процедури [5]:

Крок 1 – *розгалуження*. Вибрати змінну X_{jo} , значення якої не є цілочисельним. Покласти $L = [X_{jo}]$, де $[]$ – процедура виділення цілої частини. Сформулювати дві задачі: у першу додати обмеження $X_j \geq L+1$, а у другу – $X_j \leq L$. Відібрати одну з них як поточну, а другу ввести у список задач G для подальшого створення множини рішень.

Крок 2. Розв'язати поточну ЛП-задачу як неперервну та знайти її оптимальний план X_o .

Крок 3. Розрахунок оцінки $\Omega = F(X_o)$.

Крок 4 – *перевірка оптимальності*. Якщо X_o – цілочисельне та $\Omega = \max\{\Omega(G)\}$, то X_o – оптимальне рішення. В іншому випадку відбираємо з G задачу з нецілочисельним рішенням, у якої оцінка є $\max\{\Omega(G)\}$, та переходимо до початку наступної ітерації.

Перед розв’язанням задачі необхідно попередньо виконати такі процедури:

1) привести математичну модель до канонічного вигляду у формі задачі на максимум, тобто обмеження (2.12) перетворити на рівняння за допомогою додаткових (структурних) змінних, а форму ЦФ подати як (2.11);

2) визначити початковий прийнятний (невід’ємний) базисний розв’язок задачі і у випадку необхідності ввести штучні змінні.

Особливості застосування методу:

1. Метод можна застосовувати як для повністю, так і для частково цілочисельних задач.

2. Нові обмеження виду $X_j \geq [X_{jo}] + 1$ чи $X_j \leq [X_{jo}]$, які вводяться на кожній ітерації, виступають у вигляді відтинів.

3. При введенні нового обмеження немає необхідності знову розв’язувати всю задачу (2.11)–(2.14), а можна використовувати результати попередньої ітерації, безпосередньо вводючи в таблицю оптимального рішення нове обмеження.

4. При розв’язанні ЛП-задачі на мінімум використовують нижню границю $\Omega(G) = \min\{F(X)\}$. В цьому випадку ознака оптимальності формулюється протилежним чином.

2.2.3. Дослідження задачі оперативного планування

Організація оперативно-диспетчерського управління виробництвом ґрунтується на деталізації по виконавцях раніше розрахованого календарного плану випуску продукції в межах заданого планового інтервалу. Реалізація цієї функції здійснюється задачею оперативного планування, результатом якої є просторове та часове впорядкування комплексу запланованих робіт. Просторове упорядкування виражається у визначенні кожному виконавцю поопераційного плану робіт, а часове – встановлення черговості надходження або терміну виконання робіт.

У ГВС оперативний плановий інтервал, як правило, не перевищує зміни (добу), а виконавцем є технологічне устаткування.

Математичною формою подання задач даного класу є дискретна оптимізаційна модель, а методологією розв’язання – дискретне лінійне програмування (ДЛП) та імітаційні методи дослідження.

Формулювання задачі. Об’єктами процесу планування є роботи – технологічні операції, партії деталей, для яких необхідно встановити порядок проходження через устаткування при фіксованих технологічних маршрутах обробки.

Формально задача подається наступним чином.

Номенклатурний перелік продукції, яка виробляється на m групах $(k = \overline{1, m})$ устаткування, складається з n найменувань $(j = \overline{1, n})$.

Виготовлення партії деталей кожного найменування заздалегідь визначене послідовністю проходження деталей через групи устаткування, яку називають *технологічним маршрутом* $G_j = (L_{ij} | i = \overline{1, M_j})$, де L_{ij} – технологічна операція, яка виконується i -ою за порядком виготовлення j -ої деталі; M_j – кількість операцій, які виконуються над j -ою деталлю.

У маршруті технологічні операції $L_{ij} = (Q_{ij}, T_{ij})$ мають такі характеристики:

$Q_{ij} = k$ – номер групи устаткування, налагодженого на виконання операції L_{ij} ;

T_{ij} – нормативна тривалість виконання операції L_{ij} .

Необхідно скласти розклад $P = (T_{ij}^{\Pi} | i = \overline{1, M_j}, j = \overline{1, n})$, який визначає моменти початку виконання операцій L_{ij} (моменти запуску партії деталей на одиницях устаткування) і задовольняє системі обмежень:

– умові виконання технологічної послідовності:

$$T_{ij}^{\Pi} \geq T_{i-1,j}^K;$$

– умові виконання технологічних маршрутів;

– умові виконання операцій без перерв:

$$T_{ij}^K = T_{ij}^{\Pi} + T_{ij};$$

– умові виконання в кожний момент часу тільки однієї операції на одиниці устаткування:

$$(T_{i_1 j_1}^{\Pi} \leq T_{i_2 j_2}^{\Pi}) \Rightarrow (T_{i_1 j_1}^K \leq T_{i_2 j_2}^K),$$

де T_{ij}^K – момент закінчення виконання операції L_{ij} .

Пошук найкращого розкладу виконується за критерієм ефективності, вибір якого індивідуально залежить від економічних, організаційних та технічних особливостей роботи конкретного виробничого підрозділу в умовах досягнення найбільших показників господарської діяльності.

Відомо, що від структури планованого розподілу деталей по обладнанню (переналагодження у просторі) та розміру партій деталей (переналагодження у часі) залежать значні виробничо-господарські показники. Тому процес побудови критерію ефективності складається з визначення показника оцінювання та вибору форми виразу оцінки в залежності від економічних умов організації виробництва. Найчастіше показником оцінювання в критерії визначають час виробничого циклу, фондівіддачу устаткування, обсяг незавершеного виробництва, а

формою оцінювання – сумарне, максимальне чи мінімальне значення показника або його середнього значення за плановий час.

Найбільш поширеними типами критеріїв є:

1) *оптимізація виробничого циклу* – часу випуску заданого обсягу продукції як сумарної тривалості обробки всіх деталей:

$$\begin{aligned} & \min \left(\max_{i,j} \{T_{ij}^K\} \right); \\ & \min \left(\max_k \{T_k^P + T_k^{PP}\} \right); \\ & \min \left(\max_j \left\{ \sum_i (T_{ij}^{oc} + T_{ij}) \right\} \right), \end{aligned}$$

де T_k^P – сумарний час виконання операцій на k -ій одиниці устаткування;

T_k^{PP} – сумарний час простоїв k -ої одиниці устаткування;

T_{ij}^{oc} – очікування j -ої деталі перед обробкою на i -ій операції;

2) *оптимізація використання устаткування* (фондовіддачі):
– максимізація завантаження устаткування, а саме:

мінімального – $\max \left(\min_k \{K_k^3\} \right);$

загального – $\max \left(\sum_k K_k^3 \right),$

де $K_k^3 = T_k^P / (T_k^P + T_k^{PP})$ – коефіцієнт завантаження k -ої одиниці устаткування;

– мінімізація часу простою устаткування, а саме:

максимального сумарного – $\min \left(\max_k \{T_k^{PP}\} \right);$

максимального міжопераційного – $\min \left(\max_{i,j} \{T_{ij}^{PP}\} \right);$

загального – $\min \left(\sum_k T_k^{PP} \right),$

де T_{ij}^{PP} – простій k -ої одиниці устаткування ($k = Q_{ij}$) перед виконанням операції L_{ij} ;

$T_k^{PP} = \sum_{(i,j|Q_{ij}=k)} T_{ij}^{PP}$ – сумарний простій k -ої одиниці устаткування;

– мінімізація середнього міжопераційного простою устаткування, а саме:

максимального – $\min \left(\max_k \{T_k^{PP} / N_k\} \right);$

$$\text{загального} - \min \left(\sum_k T_k^{PP} / N_k \right),$$

де N_k – кількість операцій, що виконується на k -ій одиниці устаткування, або кількість одиниць простою у випадку, якщо устаткування виконує однакову кількість операцій;

3) *оптимізація незавершеного виробництва*:

– мінімізація очікування деталей перед обробкою, а саме:

$$\text{максимального міжопераційного} - \min \left(\max_{i,j} \{T_{ij}^{oc}\} \right);$$

$$\text{максимального подетального} - \min \left(\max_j \left\{ \sum_i T_{ij}^{oc} \right\} \right);$$

$$\text{загального} - \min \left(\sum_{i,j} T_{ij}^{oc} \right);$$

– мінімізація середнього очікування деталей перед обробкою, а саме:

максимального –

$$\min \left(\max_j \left\{ \sum_i T_{ij}^{oc} / M_j \right\} \right);$$

$$\min \left(\max_j \left\{ \sum_i T_{ij}^{oc} / N_j \right\} \right);$$

загального –

$$\min \left(\sum_{i,j} T_{ij}^{oc} / M_j \right)$$

$$\min \left(\sum_{i,j} T_{ij}^{oc} / N_j \right),$$

де N_j – кількість одиниць часу очікування j -ої деталі перед обробкою (може застосовуватися у випадку, якщо M_j однакове для всіх деталей).

Кожен з наведених типів критеріїв орієнтований на задоволення тільки власного показника ефективності виробництва. Проте, перший є дещо більш загальним та багатофункціональним по відношенню до другого та третього, оскільки може їх оптимізувати за різних технологічних умов організації виробництва. Так, якщо всі деталі обробляються за однаковим технологічним маршрутом, то використання першого критерію означає також оптимізацію використання устаткування, тобто мінімізацію простою.

Часто в реальних виробничих умовах необхідно вирішувати задачу оперативного планування, враховуючи одночасно декілька критеріїв. Одним із засобів розв'язання багатокритеріальної задачі є створення компромісного критерію. Його створення передбачає

наступне:

- для кожного локального критерію ($E_i | i = 1, r$) розв'язується задача оптимізації і обчислюється її екстремальне значення E_i^* ;
- задаються вагові коефіцієнти пріоритету I_i та визначаються рівняння відхилень $V_i = E_i - E_i^*$ кожного критерію від свого оптимального значення у кожному іншому випадку розв'язання задачі;
- будується вираз компромісного критерію з використанням адитивної функції $\min \left(\sum_i \lambda_i \cdot V_i \right)$ або функції рівномірного відхилення

$$\min \left(\sum_i \lambda_i \cdot V_i / E_i^* \right);$$

- розв'язується задача із застосуванням компромісного критерію або серед раніше знайдених рішень обирається те, яке оптимізує компромісний критерій.

Важливими показниками якості сформованого розкладу, з точки зору його реалізації при оперативно-диспетчерському управлінні, виступають прості устаткування, час очікування деталей перед обробкою та локальні резерви часу.

Простий – це інтервал часу між завершенням виконання попередньої операції та початком наступної за розкладом роботи устаткування. Прості бувають доопераційні (час до початку виконання першої операції на обладнанні) та міжопераційні.

Очікування – це інтервал часу між завершенням виконання обробки деталі на попередній за технологічним маршрутом операції та початком виконання наступної операції.

Локальний резерв часу – це інтервал, на який можна збільшити тривалість операції, не змінюючи момент початку наступної за розкладом робіт операції.

Локальний резерв операції L_{ij} розраховується як мінімальне значення між простим устаткуванням $k = Q_{ij} = Q_{gh}$ після виконання операції L_{ij} та часом очікування j -ої деталі перед обробкою на операції $L_{i+1,j}$:

$$T_{ij}^{PP} = \min \{ T_k^{PP}, T_{i+1,j}^{Oч} \}; T_k^{PP} = T_{gh}^{\Pi} - T_{ij}^K; T_{i+1,j}^{Oч} = T_{i+1,j}^{\Pi} - T_{ij}^K,$$

де $T_{i+1,j}^{Oч}$ – час очікування j -ої деталі перед обробкою на операції $L_{i+1,j}$;

T_k^{PP} – час простою k -го устаткування після виконання операції L_{ij} ;

L_{gh} – наступна після L_{ij} операція, що виконується на цьому обладнанні.

Резерв створюється як за рахунок неможливості повного завантаження устаткування навіть за умови оптимального розв'язання

задачі оптимального планування, так і за рахунок цілеспрямованого введення його у розклад роботи.

Основне призначення локального резерву полягає у використанні його в оперативно-диспетчерському управлінні з метою компенсування зовнішніх впливів на час виконання операцій.

Також резерви можуть бути використані для включення у розклад роботи додаткових робіт, що не порушують основну структуру розкладу, але підвищують завантаження устаткування.

Методи розв'язання задач. Усі методи розв'язання задач цього типу умовно розбивають на наступні основні класи:

- аналітичні методи, які ґрунтуються на апараті дискретної оптимізації;

- імітаційні методи, які ґрунтуються на імітації роботи об'єкта планування та використання повного або часткового перебору варіантів запуску деталей в обробку;

- комбіновані аналітико-імітаційні методи.

Існуючі аналітичні методи звичайно прямо або непрямо пов'язані з перебором варіантів, але їх трудомісткість експоненційно залежить від розмірності задачі. Відомі методи відсіювання варіантів типу “тілок та границь” дозволяють зменшити, іноді суттєво, коефіцієнт пропорційності в залежності між трудомісткістю алгоритму та розмірністю задачі і тим самим розширити область практичного застосування алгоритмів, що перебираються. Однак експоненційний характер цих алгоритмів для задач у загальному вигляді залишається незмінним. Проте існує клас окремих задач оперативного планування, що мають аналітичні алгоритми вирішення неекспоненційного характеру складності. Ці задачі мають обмеження у застосуванні, пов'язані з вихідними умовами задачі, наприклад, однаковий час або маршрут обробки, обмежену кількість устаткування – однієї до трьох одиниць технологічного устаткування, тощо, тобто ефективні аналітичні методи існують тільки для простих випадків формулювань задач оперативного планування.

Типовим представником таких задач, в яких відображаються найбільш поширені умови виробництва, є *задача Джонсона* “про два верстати”, що має оптимальний алгоритм розв'язання.

Виробнича ділянка складається з двох одиниць устаткування, яке обробляє вироби n типів за однаковим технологічним маршрутом, тобто деталі повинні послідовно пройти через усе устаткування. Необхідно визначити черговість запуску-випуску виробів за критерієм мінімізації виробничого циклу (загального часу) виготовлення усіх виробів.

Очевидно, що для отримання оптимального рішення потрібно мінімізувати сумарний час простою другого устаткування в очікуванні завершення обробки деталей на першому. Принципом оптимальності черговості запуску за визначеним критерієм ефективності є $\min(T_{1k}, T_{2l})$

$< \min(T_{1l}, T_{2k})$, якщо деталь k йде раніше деталі l . За допомогою цієї нерівності формується наступний алгоритм оптимізації:

- всі деталі поділяються на дві групи. До першої належать деталі, в яких $T_{1k} < T_{2k}$, а до другої – деталі, в яких $T_{1k} \geq T_{2k}$;

- в першій групі деталі впорядковуються за зростанням часу T_{1k} , а в другій – за зменшенням часу T_{2k} ;

- загальна черговість запуску деталей визначається як послідовність обробки деталей з першої групи, а потім з другої.

Аналогічний алгоритм може застосовуватися і у випадку трьох верстатів. В цьому випадку принцип оптимальності набуває вигляду:

$$\min(T_{1k} + T_{2k}, T_{3l} + T_{2l}) < \min(T_{1l} + T_{2l}, T_{3k} + T_{2k}),$$

якщо деталь k йде раніше на обробку деталі l , тобто, в першій групі ($T_{1k} < T_{3k}$) деталі впорядковуються за зростанням часу ($T_{1k} + T_{2k}$), а в другій ($T_{1k} \geq T_{3k}$) – за зменшенням часу ($T_{3k} + T_{2k}$).

У випадку часткового проходження деталей через одиниці устаткування, таку ситуацію також можна звести до вихідної, прийнявши нульовий час роботи устаткування. Проте, якщо кількість устаткування перевищує три одиниці, то застосований принцип оптимальності та наведений алгоритм не може бути використаний, оскільки не дає оптимального розв'язку задачі.

Імітаційні методи. Для складних задач (різні технологічні маршрути обробки, кількість устаткування перевищує три тощо) з точки зору практичних цілей отримання результату за короткий термін найчастіше використовують евристичні алгоритми складання розкладів на базі вирішальних правил в режимі імітації роботи виробничої системи. У цьому режимі виконуються паралельна (одночасна) побудова діаграм Ганта для всіх одиниць технологічного устаткування, які беруть участь у процесі планування.

Діаграма Ганта – це часовий графік виконання операцій технологічних маршрутів обробки деталей на визначених одиницях устаткування. Графічно подається у вигляді діаграми, де проти кожної одиниці устаткування у визначені моменти часу призначені операції обробки.

Алгоритм побудови розкладів за даним методом наступний:

0. Нехай у деякий момент часу $T = \min(T_{ij}^K)$ верстат $l = Q_{ij}$ закінчив обробку операції i поточної деталі j . Цей момент визначається як мінімальний серед усіх встановлених на поточний момент термінів завершення виконання операцій в графіку робіт.

1. Оброблена деталь j заноситься у портфель робіт наступного за технологічним маршрутом верстата $l' = Q_{i+1j}$.

Портфель робіт це підготовлені (у стані очікування) до виконання на верстаті операції обробки деталей.

Якщо є декілька варіантів технологічного маршруту, деталь одночасно заноситься у відповідну кількість портфельів. Якщо виконана

операція була останньою за технологічним маршрутом ($i = M_j$), то деталь виключається з розгляду. Цей пункт повторно виконується для всіх верстатів, які на момент T завершили виконання операцій.

2. Якщо портфель робіт верстата l порожній, то він буде переведений у стан простою. В іншому випадку за допомогою вирішального правила переваги з портфелю вибирається одна деталь j' та записується як поточна в розклад робіт даного верстата із зазначенням часу завершення операції $T_{ij'}^K = T_{ij'}^H + T_{ij'}$.

Обрана таким чином деталь виключається з усіх портфелів, де вона знаходилася. Цей пункт повторно виконується для всіх верстатів, які на момент T завершили виконання операцій.

3. Якщо у випадку виконання п. 1 з'явилась можливість завантажити верстат, який знаходиться у стані простою, то відповідна деталь записується в розклад робіт цього верстата, для якого формується нове значення часу завершення операції за правилом п. 2.

4. Обирається наступний верстат з мінімальним поточним значенням T та виконується перехід до п. 0. Планування ведеться до повного виконання усіх операцій над деталями або до тих пір, поки не буде побудований розклад на потрібний інтервал планування (змін/добу).

За вирішальні правила в алгоритмах імітаційного моделювання найчастіше використовуються наступні правила переваги:

1) правило найкоротшої операції – з поточного портфелю робіт, які підготовлені до обробки на поточному верстаті, вибирається деталь з мінімальним часом обробки; мета правила – якнайшвидше завантажити роботою наступні за технологічним маршрутом верстати;

2) правило максимальної залишкової трудомісткості – з поточного портфелю робіт, які підготовлені до обробки, вибирається деталь з максимальною сумою часу обробки на усіх ще невиконаних операціях; мета правила – закінчити обробку всіх деталей приблизно одночасно;

3) правило вирівнювання завантаження верстатів – з портфелю робіт вибирається деталь, яка потім надходить на верстат, який має у даний час мінімальний за трудомісткістю портфель підготовлених робіт; мета правила – рівномірно завантажити верстати (правило можна використовувати при багатоваріантних маршрутах);

4) правило мінімальної залишкової трудомісткості – альтернатива правилу 2;

5) правило найдовшої операції – альтернатива правилу 1;

6) правило призначення у порядку надходження (FIFO) – з поточного портфелю робіт вибирається деталь, яка надійшла в чергу на обробку до верстата першою;

7) правило LIFO – альтернатива правилу 6.

Із наведених правил видно, що вони мають евристичний

характер, тобто за їх допомогою неможливо встановити та оцінити наближення до оптимальності отриманого рішення, але можна виробити “добре” рішення в залежності від їх призначення по застосуванню чи від критерію функціонування виробничої системи, для якої розробляється розклад роботи.

Приклад розрахунку розкладу роботи за правилом найкоротшої операції наведено на рис. 2.3.

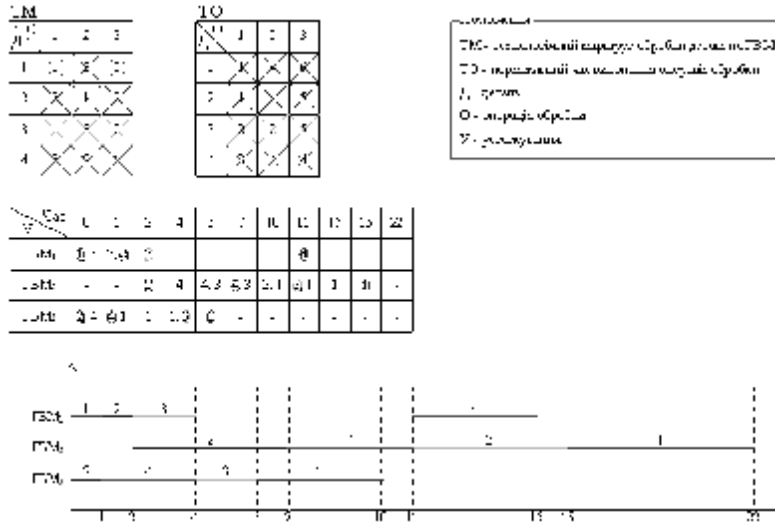


Рис. 2.3. Розрахунок розкладу за правилом найкоротшої операції

Аналітико-імітаційні методи будуються за схемою, що передбачає використання одночасно як аналітичних, так і імітаційних процедур. Всі ці методи відрізняються один від одного підходом до черговості застосування аналітичних та імітаційних процедур.

Один з таких підходів передбачає двоетапне розв’язання задачі. На першому етапі планується обробка невеликої кількості збільшених партій деталей на двох або трьох групах технологічного устаткування, тобто проводиться зменшення розмірності задачі для того, щоб використати точні аналітичні методи, наприклад, алгоритм задачі Джонсона. На другому етапі будуються уточнені календарні плани для кожної зі створених груп устаткування з використанням імітаційних методів. Групування устаткування виконується за критерієм мінімізації кількості деталей, що проходять обробку більше, ніж у одній групі. Групування деталей виконується за критерієм мінімізації кількості устаткування, яке обробляє більше, ніж одну групу деталей.

Таким чином, декомпозиція задачі планування дозволить суттєво зменшити розмірність, складність та прискорить розв’язання задачі. Проте існує залежність якості рішення від точності оцінок часу обробки збільшених партій деталей, що застосовуються на першому етапі. Тому при використанні такого підходу в аналітико-імітаційних методах необхідно розробляти декілька варіантів виконання першого етапу з метою проведення якісного аналізу отриманих результатів

задачі.

2.3. Оперативне управління ГВС

Для реалізації розроблених виробничих програм і календарних графіків необхідна система оперативного контролю і регулювання ходу виробничого процесу. Це пов'язано з тим, що на хід виробничого процесу впливає ряд випадкових факторів, які виключити повністю практично неможливо. До них відносяться поломки устаткування, перебої в постачанні, брак, ступінь забезпеченості трудовими ресурсами, іноді виникаюча необхідність випуску незапланованої продукції.

Оперативний контроль та регулювання (диспетчеризація) ходу виробництва здійснюється як у масштабі всього підприємства, так і на рівні окремих виробничих підрозділів. Поряд з функцією, зв'язаною з оперативним усуненням відхилень ходу виробничого процесу від нормального, передбаченого календарним планом, диспетчеризація повинна носити і попереджувальний характер. Це означає, що диспетчеризація повинна включати завчасне виявлення та усунення намічених відхилень від розроблених календарних графіків, тобто коригування поточних планів.

2.3.1. Призначення задачі та методи організації оперативного контролю за допомогою ЕОМ

Диспетчерський (оперативний) контроль здійснюється на основі оперативної інформації про фактичний хід виробничого процесу.

Функції та задачі диспетчеризації залежать від виду виробництва. У *масовому виробництві* диспетчерському контролю підлягають у першу чергу задані ритми роботи технологічних ліній, а також рівні внутрішньолінійних і міжлінійних заділів. В умовах *серійного виробництва* основними параметрами при диспетчеризації служать терміни запуску і випуску партії деталей і складальних одиниць, рівень запасів і ступінь забезпеченості складання. В *одиночному і дрібносерійному* виробництві контролюються терміни виконання етапів робіт, а також оперативна підготовка до виконання завдань. Диспетчерський контроль здійснюється відповідно до циклових чи мережевих планів-графіків виконання замовлень.

Оперативний контроль здійснюється на всіх трьох рівнях системи управління інтегрованим виробництвом.

На *третьому і четвертому ієрархічному рівнях* контролюється працездатність устаткування, правильність виконання програм системи числового керування, точність відпрацьовування керуючих впливів виконавчими механізмами. Важливе значення має діагностика стану устаткування та інструмента, за допомогою якої вдається запобігти аваріям і відмовленням. Для діагностування стану устаткування використовується інформація, одержувана від датчиків зусиль і

моментів, встановлених на найбільш навантажених кінематичних елементах устаткування, а також від датчиків, що вимірюють значення струму в електроприводах.

Керуюча ЕОМ періодично “зчитує” покази цих датчиків, порівнює їх із заздалегідь заданими критичними значеннями й у випадку перевищення цих значень або видає повідомлення оператору, або аварійно припиняє роботу устаткування.

Для діагностики стану інструмента, дещо рідше, використовується інформація про зміну струму в приводі робочого органу верстата. Графік зміни струму привода в часі має досить складну форму, обумовлений великим числом різноманітних факторів. Тому задача діагностики стану різального інструмента часто формулюється як задача на розпізнавання змін, викликаних зносом чи поломкою інструмента.

У комплекс технічних засобів оперативного контролю входять також і вимірювальні (контрольні) роботизовані комплекси. На основі інформації, одержуваної від цих комплексів, формується множина робіт, яку необхідно виконати для виправлення браку чи виготовлення деталей замість бракованих.

У зв’язку зі зростанням складності, інтенсивності, енергонасиченості та рівня автоматизації сучасних технологій велика увага приділяється як оперативності інформування оператора про стан виробничої системи, так і формі його подання, яка повинна бути легкою до сприйняття, сприятливою до швидкого аналізу ситуації і своєчасного прийняття рішень. З цією метою широко використовуються кольорові графічні дисплеї, на яких представляється мнемосхема контрольованої ділянки. Інформація про стан виробничої системи надходить на екран дисплея у вигляді мнемонічних символів, колір яких може асоціюватися зі ступенем важливості даного повідомлення для оператора. Звичайно на мнемосхемі надходить сигнальна інформація і декілька найбільш важливих кількісних показників виробничого процесу. Якщо оператора цікавить більш докладна інформація, він може викликати її на екран цього ж чи іншого дисплея, передавши відповідний запит ЕОМ.

На *другому* ієрархічному рівні виробляється диспетчерський контроль ходу виконання оперативного плану. Він характерний для будь-якого типу дискретного виробництва, у тому числі і для ГВС. У масовому виробництві диспетчерському контролю підлягають у першу чергу ритми роботи технологічних ліній, а також рівні заділів деталей. В умовах серійного виробництва основними контрольованими параметрами є терміни запуску і випуску партій деталей, складальних одиниць, готових виробів і рівень запасів деталей, матеріалів, сировини. В одиничному і дрібносерійному виробництві об’єктами диспетчерського контролю стають етапи чи робіт, навіть окремі технологічні операції, підготовка до виконання робіт, а також

забезпеченість ресурсами усіх видів.

Диспетчерський контроль у ГВС має свою специфіку. Основна особливість – складність одержання інформації про стан устаткування, хід виконання технологічних операцій і окремих етапів робіт, викликана “безлюдністю” таких виробництв.

В умовах високого рівня автоматизації, коли на ділянці присутній тільки диспетчер і небагато операторів - наладчиків (що особливо характерно для другої і третьої змін роботи), інформація про стан виробництва може бути отримана тільки від відповідних датчиків. Однак їх кількість (як і надійність) обмежена, що призводить до визначеного “дефіциту” інформації про процес.

Не набагато краще положення при прямому числовому керуванні. Зв’язок між центральною керуючою ЕОМ і пристроями числового програмного керування дозволяє передавати в центральну ЕОМ інформацію про стан верстата і виконувати у даний момент операції. Однак і у випадку прямого числового керування проблема інформаційного дефіциту не зважається повністю.

Існує ряд методів, що дозволяють організувати диспетчерський контроль із застосуванням комп’ютерної техніки і методів математичного моделювання. Один з них побудований на використанні синхронних імітаційних моделей технологічного процесу. Основна ідея цього методу – створити імітаційну модель технологічної підсистеми, процес функціонування якої синхронізований із процесом у реальному об’єкті, засоби аналізу неузгодженостей реального і модельованого процесів, а також апарат, що дозволяє фіксувати на моделі і повідомляти диспетчеру про важливі, з погляду контролю, події, що не ідентифікуються інформаційними засобами на реальному об’єкті.

Структура системи диспетчерського контролю такого типу представлена на рис. 2.4. Її центральним елементом є імітаційна модель синхронного типу.

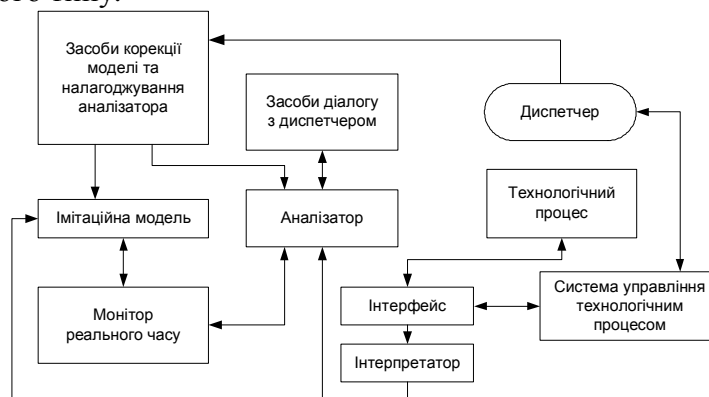


Рис. 2.4. Структура системи оперативного контролю із синхронною моделлю

Розглянемо деякі принципи її побудови.

1. Концептуальною основою моделі є схема станів, відповідно

до якої процес представляється послідовною зміною станів $s \in S$ у часі, де S – множина станів, у яких може знаходитися модельований об'єкт. Перехід зі стану s_i у стан s_j ($s_i, s_j \in S$) називають подією $e_k \in E$, де E – множина подій, можливих у модельованій системі.

2. Вибір множини станів для побудови моделі здійснюється із врахуванням їх значущості для диспетчерського контролю. Якщо стан у модельованому об'єкті має заздалегідь відому тимчасову тривалість, то в моделі цьому стану приписується така ж тривалість.

3. Імітаційна модель описує тільки технологічний процес (технологічну підсистему). Входами моделі є керуючі впливи, що надходять від системи керування технологічним процесом одночасно як в об'єкт керування, так і в його модель.

4. Функціонування імітаційної моделі здійснюється під керуванням монітора реального часу і сигналів системи керування технологічним процесом, причому темпи реального процесу і процесу в моделі повинні збігатися.

5. Будь-якому стану $s \in S$ в імітаційній моделі можуть бути присвоєні мітки.

6. У процесі функціонування імітаційної моделі генеруються події $e_k \in E$. Якщо попередній чи наступний стани деякої події e_k мають мітку, то ця ж мітка присвоюється події e_k . Мітка події складається з двох частин: унікальний код події і код типу події. Модель такого типу може бути побудована з використанням апарату сіток Петрі.

Події, що відбуваються в моделі та об'єкті, поділяються на три типи: синхронізуючі, контрольовані і такі, що ідентифікуються. Інформація про синхронізуючі події використовується для усунення тимчасових неузгодженостей між процесами в моделі та в об'єкті через випадкові відхилення реальної тривалості процесу від заданої. Як синхронізуючі можуть використовуватися такі події, як відправлення на обробку партії деталей, прибуття деталей на склад після обробки. Інформація про ці події передається в інтерпретатор системою керування автоматизованим складом, а потім обробляється аналізатором. Результатом обробки інформації є коригування поточного стану моделі, а також перерахування часу в календарі подій і станів, що ведеться в моніторі реального часу.

До контрольованих відносяться події, настання яких потрібно фіксувати й аналізувати в ході диспетчеризації технологічного процесу. Інформація про ці події надходить в аналізатор з реального процесу і моделі у вигляді двох пар: (e, t_p) і (e, t_m) , де e – код події; t_p і t_m – відповідно час настання реальної події і час настання цієї ж події в моделі. В аналізаторі міститься інформація про допустиму величину неузгодженості t_p і t_m . Якщо неузгодженість перевищує допустиме значення, то формується і передається повідомлення диспетчеру. Інформація про контрольовані події також використовується для

синхронізації процесів у моделі та об'єкті.

До тих, що ідентифікуються, відносяться події, які не можуть бути виявлені інформаційною системою ГВС, однак вони важливі для контролю та регулювання технологічного процесу. Мітки станів, зв'язані з цими подіями, і алгоритми їх ідентифікації задаються при настроюванні моделі й аналізатора. Після ідентифікації події інформація про нього передається диспетчеру, як правило, у вигляді запиту, що вимагає деякої відповіді, наприклад підтвердження факту здійснення даної події. В залежності від відповіді аналізатор коригує стан моделі і календар.

Розглянемо варіант складу програмних і апаратних засобів реалізації системи контролю, що використовує описані принципи (див. рис. 2.4).

Інформація від датчиків і системи керування ГВС через стандартний інтерфейс надходить в інтерпретатор, функціями якого є аналіз сигналів і їх переклад на мову імітаційної моделі. Наприклад, якщо імітаційна модель представляє собою сітку Петрі, то інтерпретатор на основі аналізу вхідних сигналів керує відповідними переходами сітки.

Сигнали інтерпретатора, що несуть інформацію про події, сприймаються також аналізатором, у якому відбувається розпізнавання типу події та занесення його у відповідний список подій.

Монітор реального часу представляє собою “службу часу”, що складається із системи таймерів, програми керування таймерами, а також програмних чи апаратних засобів перерахування часу відповідно до синхронізуючої інформації. Систему таймерів можна розглядати як календар станів і подій.

Аналізатор – це комплекс, який включає програми логічної обробки інформації, що надходить з реального об'єкта і моделі, програми ведення списків подій і програми, що керує функціонуванням аналізатора.

Засоби діалогу з диспетчером здійснюють переклад повідомлень з мови внутрішнього представлення (мови імітаційної моделі) на мову, яка сприймається диспетчером (наприклад, символи на мнемосхемі процесу чи фрази на професійному діалекті диспетчера). Ці ж програмні засоби служать для зворотного перекладу.

Повідомлення, що використовуються в діалозі, поділяються на три групи: перша – інформація диспетчеру, що не вимагає від нього відповіді; друга – запити в диспетчера визначеної інформації; третя – запити диспетчером інформації про стан системи.

Засоби коригування моделі і настроювання аналізатора представляють собою сукупність редакторів діалогового типу, що дозволяють вносити структурні та параметричні зміни в модель, що відповідають змінам, які відбулися в об'єкті, чи розставляти змінювати мітки станів, а також коригувати алгоритми логічної обробки

інформації.

Диспетчерська служба здійснює щодобовий контроль за ходом виконання програм заготівельних цехів за допомогою рапортів про виконання плану у вартісних показниках та про виконання по-детальних завдань. Відповідно до інформації, що міститься в добовому рапорті про виконання планів задачі, складаються зведення відстаючих деталей. Ступінь відставання деталей відповідає відхиленню від норми заділу по даній деталі. Відставання визначається звичайно в днях чи у штуках. Рішення задачі диспетчеризації дозволяє забезпечити комплектність виконання завдань заготівельним виробництвом, що, в свою чергу, є одним з головних факторів ритмічної роботи підприємства в цілому.

2.3.2. Призначення задачі та стратегії оперативного корегування у ГВС

Оперативне корегування полягає в розробці нових планів на весь оперативний період чи його частину, у зміні фрагментів плану за умови спряженості з частинами, що залишилися без зміни, у зсуві часів запуску деталей в обробку в межах допустимих інтервалів. Необхідність корекції виникає звичайно через відмовлення устаткування, дефіцит заготовок, інструмента, матеріалів, відсутність керуючих програм, виготовлення браку.

Можливості корегування оперативного плану повинні бути передбачені загальною стратегією оперативного керування та методикою календарного планування.

Однієї з розповсюджених стратегій оперативного управління є виділення понадоперативних інтервалів планування. Відповідно до неї місячний оперативний плановий період поділяється на короткі понадоперативні інтервали (ПОІ), наприклад, декадні, тижневі чи добові.

Нехай для визначеності як ПОІ обраний тиждень. Задача календарного планування зважується для перших двох тижнів. Після закінчення першого тижня складається календарний план для третього тижня, після закінчення другого – для четвертого.

Якщо система оперативного планування більш високого рівня організована так, що план наступного місяця розробляється не пізніше третього тижня поточного місяця, то описана стратегія дозволяє здійснювати безупинне оперативне планування з “пересувним” обрієм.

Корегування в такій системі здійснюються в такий спосіб. Нехай закінчився деякий k -й ПОІ і планується $(k + 2)$ -й. На основі даних контролю виконання плану k -го ПОІ визначається множина P_1 невиконаних робіт. Потім визначається множина робіт P_2 , які необхідно виконати для виправлення браку чи виготовлення деталей замість забракованих. На основі додаткової інформації диспетчером формується множина додаткових робіт P_3 , поява яких може бути пов’язана, наприклад, зі зміною пріоритетів замовлень на готову продукцію.

Множина робіт P^{k+2} на $(k + 2)$ -му ПОІ утвориться об'єднанням робіт:

$$P^{k+2} = P_1 \cup P_2 \cup P_3 \cup P_4,$$

де P_4 – роботи, заплановані заздалегідь на $(k + 2)$ -й ПОІ.

Очевидно, для виконання множини робіт P_1 , P_2 і P_3 необхідно мати визначений резерв продуктивності, що повинен бути передбачений на етапі розбиття місячної програми на ПОІ.

Розглянемо один зі способів забезпечення резерву. Усі роботи поділяються на обов'язкові і фонові. До обов'язкових робіт $P_{об}^{k+2}$ відносяться роботи, критичні терміни виконання яких знаходяться в запланованому ПОІ, а до фонових $P_{ф}^{k+2}$ – роботи, критичні терміни виконання яких знаходяться за межами запланованого періоду.

Множина робіт P^{k+2} аналізується з погляду забезпеченості ресурсами і виробничими потужностями. Якщо існує можливість виконання всієї множини робіт P^{k+2} , то вона використовується як вихідна для рішення задачі календарного планування. Якщо такої можливості немає, то P^{k+2} представляється як об'єднання:

$$P^{k+2} = P_{об}^{k+2} \cup P_{ф}^{k+2}.$$

Календарний план складається для робіт $P_{об}^{k+2}$. Виробничі потужності, що залишилися вільними, завантажуються роботами з множини $P_{ф}^{k+2}$.

Гнучкість системи оперативного коригування багато в чому визначається тривалістю ПОІ. Очевидно, чим коротші ПОІ, тим більшу швидкість реакції на збурення можна забезпечити. Однак тривалість ПОІ істотно залежить від гнучкості та оперативності інших планувальних служб підприємства, зокрема, заготівельного та інструментального виробництв, матеріально-технічного постачання, служби підготовки керуючих програм тощо. При достатній гнучкості цих служб можна вибрати добовий ПОІ і скласти календарний план, наприклад, на 3 доби. Після закінчення k -ої доби задача календарного планування із врахуванням коригувань вирішується для $(k + 3)$ -ї доби.

Описана стратегія оперативного коригування може бути розширена за рахунок зміни календарного плану в середині одного ПОІ. Можливість такого коригування повинна бути передбачена на етапі календарного планування за рахунок введення локальних резервів часу по всіх чи найбільш важливих одиницях устаткування.

Локальний резерв – це інтервал часу, на який можна збільшити тривалість операції, не зрушуючи момент початку наступної операції.

Локальні резерви часу утворюються як за рахунок неможливості повного завантаження устаткування навіть при оптимальному рішенні задачі календарного планування, так і за рахунок цілеспрямованого введення їх у розклад. Варіюючи локальними резервами, диспетчер може компенсувати тимчасові відмовлення устаткування, включити в план

додаткові замовлення, а також врахувати ряд інших вимог, не порушуючи основну структуру календарного плану.

Для того щоб локальні резерви часу не призводили до зниження завантаження устаткування, диспетчеру дається множина фонових робіт, якими можуть бути заповнені локальні резерви часу в тому випадку, якщо не потрібно їх використовувати для компенсації збурень.

Маневруючи локальними резервами часу, вдається компенсувати відносно слабкі збурення у виробничій системі, такі як нетривала зупинка одного чи декількох верстатів, короткочасні порушення постачань заготовок, сировини, матеріалів тощо. Збурення, пов'язані зі значними змінами пріоритетів замовлень на виробі, істотними порушеннями термінів постачань комплектуючих виробів, матеріалів, сировини, тривалими відмовами устаткування, як правило, не вдається компенсувати за рахунок локальних резервів часу. При виникненні таких ситуацій потрібно швидко розрахувати новий оперативний план для частини ПОІ, що залишилася. Новий план повинен бути розрахований за час, що не перевищує інтервал, протягом якого виробничий підрозділ може працювати відповідно до колишнього плану. На практиці цього часу буває недостатньо для пошуку оптимального плану, тому необхідно використовувати наближені методи, тобто поступатися його якістю з метою прискорення розрахунку.

Час, за який потрібно розрахувати новий план, в залежності від виробничих умов може змінюватися в широких межах і в найбільш критичних ситуаціях складати кілька десятків хвилин. Отже, у складі математичного забезпечення системи оперативного планування ГВС необхідно мати ряд алгоритмів, що розрізняються “ступенем оперативності”, тобто очікуваним максимальним часом пошуку нового плану.

Існує також перспективна диспетчеризація. До її задач входить контроль за оперативною підготовкою виробництва на ділянках, за забезпеченістю ділянок матеріалами, технологічним оснащенням і технічною документацією. У процесі перспективної диспетчеризації можуть коригуватися календарні плани-графіки на найближчий період часу, наприклад, на тиждень. Це коригування виробляється із врахуванням наступного.

Здійснюється аналіз відхилень з погляду причин, що викликали їх. У результаті аналізу з плану-графіка викреслюються операції, які неможливо виконати до заданого терміну, і ті, котрі виконані достроково. При цьому враховуються ресурси по устаткуванню, а також трудові ресурси із врахуванням тривалості ремонту і часу зайнятості устаткування. Якщо виниклі відхилення не викликають значних збоїв у ході виробничих процесів, то вони ліквідуються за рахунок ресурсів, що звільнилися в результаті зняття ряду чи робіт за рахунок перерозподілу

ресурсів. У випадку більш серйозних збоїв здійснюється перебудова плану-графіка. З цією метою для всіх партій деталей, запущених у виробництво, визначається допустимий час завершення обробки – скільки операцій по обробці кожної партії уже виконано і скільки часу дана партія знаходиться у стані затримки.

Із врахуванням тривалості ремонту устаткування обчислюються моменти вивільнення верстатів, виконується обчислення “вікон”, що утворилися, в які вбудовуються деталі у відповідності з термінами готовності. Якщо виявляється, що деталі не можуть бути вбудовані у “вікна” плану-графіка, то складається новий графік за прийнятим на даному заводі алгоритмом календарного планування.

2.3.3. Організація оперативно-диспетчерського управління ГВС

Оперативна диспетчеризація пов'язана з визначенням фактичних моментів запуску деталей у виробництво та регулюванням їх надходження на технологічне устаткування. При цьому враховуються витрати часу на підготовчо-заклучні операції, переналагодження, операції обслуговування та транспортування. Оперативна диспетчеризація здійснюється в реальному масштабі часу на підставі результатів оперативного контролю і встановлених оперативним плануванням планових графіків запуску-випуску деталей. Тому синхронна модель роботи всіх одиниць устаткування (технологічних, транспортних) для використання на стадії оперативного контролю і оптимальна стратегія обслуговування запуску-випуску деталей у виробництво при регулюванні ходу виробничого процесу повинні бути результатами спільного розгляду задач оперативного планування й диспетчеризації.

Мове йде про те, що при оперативному плануванні необхідно досить детально проробити графік роботи всього комплексу обладнання або визначити план роботи тільки технологічного устаткування та встановити стратегію диспетчеризації операцій обслуговування й транспортування, а диспетчеризацію здійснити в режимі застосування синхронної моделі функціонування або в режимі прямої диспетчеризації транспортних операцій відповідно до стратегії обслуговування основного технологічного устаткування.

Отже, реалізація задач оперативного планування та оперативної диспетчеризації тісно взаємопов'язані, оскільки вибір алгоритму організації диспетчерського управління суттєво впливає на тривалість проходження партій деталей через виробничу систему, а саме на заплановані терміни запуску-випуску деталей на технологічному устаткуванні. Крім того, чим точніша процедура призначення планових термінів, тим простіше реалізуються алгоритми диспетчеризації, оскільки вимоги до обсягу додаткової оперативної інформації при прийнятті рішень істотно знижуються. Однак зростають обчислювальні витрати на розробку точного розкладу.

Безпосередньо ходом виробничого процесу керує система оперативного-диспетчерського управління, що відповідно до обраного способу (або при оперативному плануванні детально проробляється план роботи всього комплексу устаткування або визначається тільки план роботи технологічного устаткування та встановлюється стратегія диспетчеризації) реалізує спланований розклад роботи технологічного устаткування. Критерієм роботи такої системи є дотримання строків запуску-випуску деталей на технологічне устаткування відповідно до розробленого системою оперативного планування розкладу роботи за рахунок своєчасного транспортного обслуговування заявок, що надходять від технологічного устаткування.

Способи організації оперативного-диспетчерського управління. Система оперативного-диспетчерського управління відповідно до її задач функціонує у двох режимах: 1) формування диспетчерського управління за синхронною моделлю в разі, якщо не встановлено значних відхилень поточного стану виробничого процесу від запланованого; 2) формування директивного управління за стратегію прямої диспетчеризації в разі, якщо система перейшла в стан корегування календарного завдання і план-графіку роботи внаслідок встановлення значних відхилень, які не можливо компенсувати технологічними резервами поточного розкладу роботи усього устаткування.

Згідно з 1-им режимом порядок функціонування системи, тобто виникнення деяких подій, визначається і зберігається таким же, як і запланований у синхронній моделі. Синхронна модель формується на результатах розв'язку задач оперативного планування і диспетчеризації в повному обсязі, тобто із врахуванням усіх виробничих ресурсів і всього комплексу виробничих операцій (налагодження, транспортування, обробки тощо), необхідних для забезпечення виконання технологічних процесів у системі.

В результаті алгоритм диспетчеризації (регулювання) за синхронною моделлю використовує правило вибору заявки на обслуговування (формування наступної події) за найранішим часом закінчення виконання чергової роботи, що визначається зі сформованого раніше розкладу. Правило вибору формалізується у вигляді вирішувальної процедури, що визначається для всієї множини подій – станів запланованих робіт синхронної моделі.

Таким чином, розклад роботи в синхронній моделі представляється сукупністю тимчасових рядів кожної одиниці виробничого устаткування (технологічного, транспортного, обслуговуючого тощо), які відображають моменти завершення відповідних виробничих операцій системи, а результатом вирішальної процедури буде вибір найближчої за часом чергової події в моделі.

Даний підхід із синхронною моделлю відрізняється простотою задання і визначення управляючих впливів за невеликих відхилень від

модельних моментів реалізації подій у системі. У випадку значних відхилень – це гарантії того, що строге дотримання модельного порядку виникнення подій у системі призведе до своєчасного виконання планового завдання. В результаті потрібна перебудова синхронної моделі на базі перерахунку нового розкладу роботи на період, що залишився.

Згідно з 2-им режимом порядок функціонування системи визначається оптимальною стратегією розв'язання конфліктних ситуацій у наданні транспортних засобів для виконання обслуговуючих операцій за рахунок прямої диспетчеризації. Пряма диспетчеризація представляє собою розрахунок і складання розкладу роботи у задані моменти часу – події, коли система переходить до нового стану при завершенні виконання технологічної або обслуговуючої операції. Передбачається формування імітаційної моделі, що враховує усі виробничі засоби системи, в якій визначено стратегію вибору претендента обслуговування.

Коли в системі виникає велика кількість ситуацій, що призводить до значних відхилень від запланованого порядку або у яких необхідно реалізувати подійно-орієнтований принцип керування, то для розрахунку управління за поточною подією застосовують стратегії диспетчеризації транспортних операцій.

Цей спосіб припускає вибір претендента обслуговування за розрахованим раніше розкладом роботи технологічного устаткування (на етапі розв'язку задачі оперативного планування), а організацію його транспортування – за встановленою стратегією у вигляді правил переваги заявок на транспортне обслуговування. Порядок проходження заявок через технологічне устаткування відомий з розрахованого раніше розкладу робіт. Необхідно організувати транспортне обслуговування встановленого порядку заявок.

Алгоритм диспетчеризації будується на базі послідовного розв'язання конфліктів заявок на транспортне обслуговування вільними засобами таких типів:

- обслуговування заявок спільним засобом;
- надання засобів загальній заявці.

Для розв'язання конфліктів першого типу встановлюються пріоритети на першочерговий порядок обслуговування заявок у системі. Пріоритети формуються на базі правил переваги, що оцінюють всі готові до реалізації технологічні операції з розкладу роботи технологічного устаткування. Порядок переваги може бути наступним:

- заявки на обслуговування деталей, які повинні надійти на склад з технологічного устаткування;
- заявки на обслуговування деталей, які повинні надійти з одного технологічного устаткування на інше (у випадку неможливості виконання такого обслуговування воно може бути замінене на обслуговування через склад з тимчасовим зберіганням деталей і

подальшим їх відправленням на необхідне устаткування).

- заявки на обслуговування деталей, які повинні надійти на зі складу на технологічне устаткування.

Для розв'язання другого типу конфліктів встановлюються пріоритети на надання вільного транспортного засобу обраній заявці. Стратегія надання може формалізувати один з наступних принципів адресування заявок вільному устаткуванню:

- заявка направляється на устаткування з мінімальним поточним завантаженням;

- заявка направляється на найближче за часом доставки устаткування;

- заявка направляється випадковим чином чи за жорстко заданим пріоритетом.

Даний підхід найчастіше використовують у випадках, коли виникають великі відхилення від планових показників, що локальними резервами часу неможливо компенсувати або його застосовують у тих випадках, коли оперативне корегування розкладу роботи зв'язане з великими витратами або неможливе за часом.

Результатом алгоритму є формування комплексного завдання транспортній підсистемі ГВС на виконання обслуговуючої операції і технологічній підсистемі на виконання підготовчої операції для забезпечення запланованої технологічної операції.

У свою чергу другий підхід має деякі переваги перед першим, а саме:

- по-перше, при формуванні управління достатнє знання оптимальної стратегії, а не розкладу роботи всього устаткування, у зв'язку з цим потрібно менше ресурсів пам'яті ЕОМ для організації процесу управління;

- по-друге, досить висока гнучкість організації управління.

Формулювання задачі оперативної диспетчеризації та її розв'язання. Безпосередньо ходом виробничого процесу керує система оперативно-диспетчерського управління, яка відповідно до обраного алгоритму реалізує спланований розклад роботи устаткування. Порядок проходження деталей (заявок на обробку) через технологічне устаткування відомий з розрахованого попередньо розкладу роботи. Метою задачі диспетчеризації є необхідність організувати транспортне обслуговування встановленого порядку робіт із запуску-випуску деталей на технологічному устаткуванні. Критерієм роботи такої системи є дотримання строків запуску-випуску деталей на технологічне устаткування відповідно до розробленого системою оперативного планування розкладу роботи за рахунок своєчасного транспортного обслуговування заявок, що надходять від технологічного устаткування.

Кожний розрахунковий момент у процесі диспетчеризації визначається наявністю заявок на транспортне обслуговування деталей

та вільними засобами транспортування. Кожна заявка характеризується місцем доставки деталі, директивним терміном завершення доставки (обумовленим графіком обробки деталі) та переліком можливих засобів її транспортування. Місце доставки визначає тип заявки обслуговування, а наявність декількох заявок або декількох вільних засобів обслуговування породжує конфлікт вибору заявки та призначення їй засобу обслуговування.

Отже, диспетчеризація повинна передбачати розв'язання наступних типів конфліктів:

- обслуговування заявок спільним засобом;
- надання засобів спільній заявці.

Стратегія обслуговування будується на базі послідовного розв'язання конфлікту з вибору заявки та визначення засобу її подальшої реалізації.

Вибір заявки базується на встановленні її типу, серед яких є:

- заявки на обслуговування деталей, які повинні надійти з технологічного устаткування на склад;
- заявки на обслуговування деталей, які повинні надійти з одного технологічного устаткування на інше (у випадку неможливості виконання такого обслуговування воно може бути замінене на обслуговування через склад з тимчасовим зберіганням деталей і подальшим їх відправленням на необхідне устаткування);
- заявки на обслуговування деталей, які повинні надійти зі складу на технологічне устаткування.

Вибір заявки-претендента на обслуговування здійснюється за правилом переваги одного типу над іншим. Наприклад, спочатку розглядаються заявки першого, потім другого і надалі третього типів. Переваги типів заявок визначаються пріоритетом на першочерговий порядок виконання операцій у виробничій системі.

Організація вибору претендента в середині кожного типу заявок здійснюється на базі пріоритетних правил переваги заявок:

- найкоротшої транспортної операції;
- найбільш критичний директивний термін запуску на технологічне устаткування;
- мінімальної залишкової трудомісткості технологічних операцій;
- мінімальної залишкової трудомісткості транспортних операцій;
- правила FIFO, LIFO та ін.

Призначення засобу обслуговування обраної заявки здійснюється на базі пріоритетів на надання устаткування. Стратегія надання може формалізувати один з наступних принципів адресування заявок вільному устаткуванню:

- заявка направляється на устаткування з мінімальним поточним завантаженням;

– заявка направляється на найближче за часом доставки устаткування;

– заявка направляється випадковим чином або за жорстко заданим пріоритетом.

Таким чином, загальний *алгоритм диспетчерського управління* на базі стратегії транспортного обслуговування складається з наступних етапів:

1. Виконати опитування ГВМ на завершення виконання технологічних операцій (відповідно до розкладу роботи у вигляді діаграми Ганта) і сформувати черги заявок на завантаження транспортних засобів.

2. Визначити порядок обслуговування заявок з черг вільних у поточний момент транспортних засобів відповідно до встановленого пріоритету заявок.

3. Для кожної обраної заявки призначити вільний транспортний засіб відповідно до обраної стратегії адресування заявок.

4. Завантажити транспортне устаткування і скоригувати всі черги завантаження, виключивши обрану заявку.

5. Повторювати алгоритм з п. 1, поки є заявки на транспортне обслуговування.

Результатом алгоритму є діаграма Ганта (рис. 2.5), в якій відтворені всі одиниці технологічного і транспортного устаткування, скоординовані всі операції обробки і обслуговування. Таку діаграму найчастіше називають технологічним (розширеним) розкладом роботи ГВС.

Надалі технологічний розклад роботи ГВС перетворюється у синхронну управляючу модель, яка застосовується при оперативному контролі і диспетчеризації.

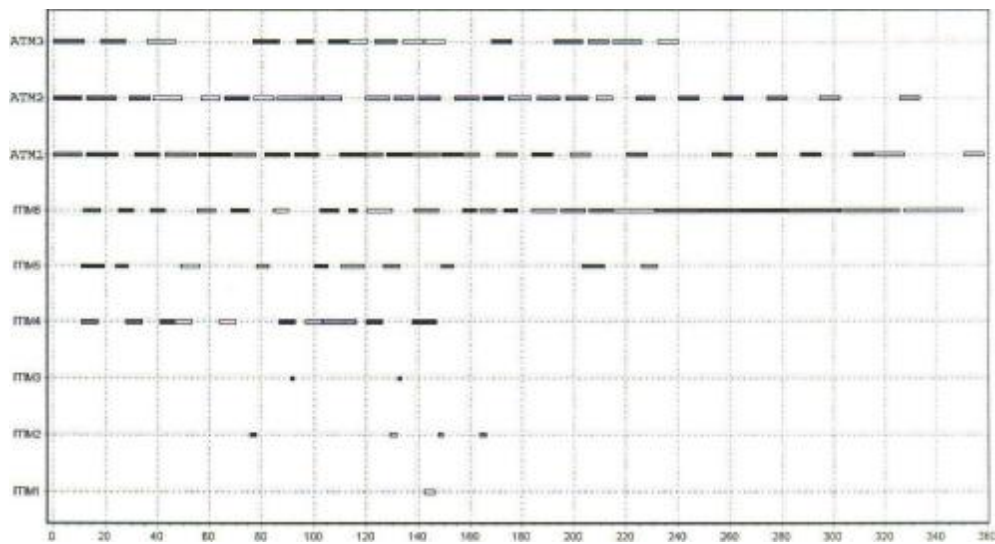


Рис. 2.5. Розрахунок розширеного розкладу за правилом найкоротшої технологічної операції і за стратегією вибору найближчого за часом доставки устаткування

Контрольні запитання

1. Наведіть основні ознаки класифікації задач планування і методів їх розв'язання в інтегрованому виробництві.
2. Визначте місце та можливості задачі планування обсягу і номенклатури випуску продукції.
3. Розкрийте зміст задачі календарного планування.
4. Розкрийте зміст та напрямки застосування задачі оперативного планування.
5. Обґрунтуйте необхідність організації оперативного контролю за допомогою ЕОМ.
6. Визначте напрямки застосування стратегій оперативного коректування планів.
7. Поясніть можливості та відмінності способів організації оперативно-диспетчерського управління.

«Якщо не висловлені протилежні думки, то не з чого вибрати найкраще» (Геродот)

Розділ 3

МОДЕЛЮВАННЯ ТА ВЕРИФІКАЦІЯ

МАТЕРІАЛЬНО-ІНФОРМАЦІЙНИХ ПРОЦЕСІВ ГВС

3.1. Специфікація предметної області інтегрованого виробництва

Формування специфікації предметної області інтегрованого виробництва проводиться з метою визначення параметрів опису матеріальних та інформаційних процесів ГВС.

Матеріальні процеси визначають порядок впливу на предмети виробництва, що функціонують у виробничих підрозділах (обробка деталей, транспорт тощо). Предмети – це деталі, інструменти, тара.

Інформаційні процеси – управляючі процеси, що забезпечують нормальне функціонування виробничих процесів (забезпечують реалізацію розкладу роботи основного устаткування, транспортної системи). Це, фактично, – послідовність команд, що видаються на устаткування для реалізації технологічних чи транспортних операцій.

Основною організаційною формою ГВС на виробництві є гнучка автоматизована дільниця (ГАД) або лінія (ГАЛ). Будь-яка ГАД (ГАЛ) має у своєму складі такі модулі:

- гнучкі виробничі модулі (ГВМ) основного технологічного устаткування (верстати, складальні машини, тощо);
- автоматизований склад (виробів, напівфабрикатів, комплектуючих, інструментів);
- автоматизовану транспортну систему.

Для забезпечення подавання виробів на ГВМ та організації їх взаємодії з автоматизованим складом в ГАД (ГАЛ) використовується автоматизована транспортна система в складі транспортних модулів або роботів-штабелерів, які мають транспортні маршрути пересування. ГВМ обов'язково мають в своєму складі вхідний та вихідний накопичувачі, в які деталі надходять для подальшої обробки згідно технології, а після завершення обробки тимчасово зберігаються для подальшого транспортування на інше технологічне устаткування. Це дозволяє спочатку привезти наступну деталь для обробки, а потім вивезти ту, що чекає на подальшу обробку. Проте, на функціонування накопичувачів та роботу ГВМ накладаються деякі обмеження, пов'язані із забезпеченням вимог на порядок ініціювання технологічної операції на устаткуванні. Це:

- транспортний модуль не може привезти наступну деталь на обробку, поки завантажений вхідний накопичувач;
- обробка деталі з вхідного накопичувача не може початися на

ГВМ, поки зайнятий вихідний накопичувач;

- після завершення обробки деталей одночасно надходить до вихідного накопичувача.

Для створення ефективно діючої СОУ ГАД (ГАЛ) необхідно в першу чергу визначити технологічні маршрути обробки деталей на ГВМ для заданого номенклатурного переліку виробів, розрахувати порядок запуску деталей у виробництво для визначення часу виробничого циклу і розкладу роботи устаткування, а потім розробити графік проведення транспортних операцій для своєчасного виконання технологічних операцій на ГВМ згідно розкладу їх роботи. Одночасно в ході складання розкладу та графіку обґрунтовується вибір достатньої кількості та призначення (спеціалізації) ГВМ та транспортних модулів, який забезпечить виконання розробленого виробничого циклу обробки деталей. Отримана інформація буде включати дані про початок і завершення різних операцій на ГВМ, транспортних модулях, напрям та їх мету пересування.

З метою аналізу та верифікації можливості реалізації розробленого розкладу та графіку роботи всього устаткування ГАД (ГАЛ) виконується моделювання матеріально-інформаційних процесів, яке перевіряє умови ініціювання операцій, виконання всього запланованих переліку дій у системі, повернення устаткування в початковий стан, обмеження ресурсів та інше. Крім, того виконується пошук вузьких місць, конфліктів при роботі устаткування. Дослідження завершується виробленням рекомендацій щодо усунення помилок та неточностей, які були припущені при проектуванні структури та алгоритмів функціонування ГАД (ГАЛ).

Отже, до *особливостей функціонування матеріально-інформаційних процесів*, які повинні враховуватися при складанні моделі слід віднести:

1. Асинхронність роботи устаткування (в окремі моменти відбувається синхронізація для взаємодії).
2. Паралельне виконання процесів обробки, транспортування, обслуговування та управління.
3. Циклічність та конвеєрність виконання операцій.
4. Багатоваріантність використання устаткування та обробки деталей.

Внаслідок необхідності врахування перерахованих особливостей до організації управління висувають наступні вимоги:

1. Якість прийнятих рішень (слід враховувати особливості та обставини функціонування технологічної підсистеми).
2. Надійність управління устаткуванням (слід гарантувати безвідмовне функціонування технологічних процесів).
3. Гнучкість управління (властивість системи управління перебудовувати порядок роботи при зміні в складі об'єкта управління).

Завдання (цілі) задачі моделювання і верифікації:

1. Управляючий процес не повинен призводити до тупикових ситуацій при виконанні виробничого процесу.

2. Управляючий процес не повинен призводити до блокування роботи устаткування.

3. Управляючий процес повинен забезпечувати повторне виконання технологічних, транспортних операцій після їх завершення.

Вихідними даними для розв'язання цих завдань та створення автоматизованої системи управління визначаються наступні параметри виробництва:

1. Виробнича програма ГВС – представляється у вигляді номенклатурних планів, встановлених на заданий період роботи.

2. Технологічні маршрути виготовлення заданої номенклатури виробів – послідовність виконання технологічних операцій на технологічному обладнанні.

3. Працємісткість (час виконання) основних технологічних операцій (обробки, складання, контролю тощо).

4. Транспортні маршрути перевезення одиниць матеріальних потоків (виробів, інструмента тощо), які визначаються структурною схемою компонування ГВС.

5. Час виконання транспортних та обслуговуючих операцій – час руху за участками транспортних маршрутів та перевантаження об'єктів транспортування.

6. Місткість застосованих у ГВС оперативних накопичувачів.

7. Критерій оцінки ефективності роботи ГВС на заданий період.

Таким чином, на концептуальному рівні може бути сформована така специфікація предметної області ГВС, подана у вигляді:

$$SP = \{D, M, O, G, TM, S_D, S_M\},$$

де $D = \{D_6, D_K, D_T, D_i\}$ – предмети виробництва;

D_6 – базові деталі;

D_K – комплектуючі;

D_T – тара;

D_i – інструменти;

$M = \{M_{осн}, M_{тр}, M_{зб}\}$ – засоби виробництва;

$M_{осн}$ – основне технологічне устаткування;

$M_{тр}$ – транспортне устаткування;

$M_{зб}$ – засоби збереження;

$O = \{O_{осн}, O_{тр}, O_{зб}, O_n\}$ – виробничі операції;

$O_{осн}$ – операції на основному технологічному устаткуванні:

$$O_{осн} = (\tau, (D_6 \bullet D_K \bullet D_i) \bullet M_{осн}) \Rightarrow M_{осн} \bullet (D_6 \bullet D_T \bullet D_i)$$

$O_{тр}$ – операції на транспортному устаткуванні:

$$O_{тр} = (\tau, (D \bullet M_{тр})) \Rightarrow M_{тр} \bullet D$$

$O_{зб}$ – операції на засобах збереження:

$$O_{зб} = (\tau, (D \bullet M_{зб}), (M_{зб} \bullet D));$$

O_n – операції налагодження виробничих операцій;

$$O_n = (\tau, (D_M \bullet M_{\text{осн}}), (M_{\text{осн}} \bullet D_M));$$

G – граф взаємодії виробничих операцій, що відображає такі їх зв'язки:

- слідування (визначає будь-який порядок реалізації, але не одночасний);
- паралелізм (одночасне виконання операцій);
- конкуренція (використання операціями загального ресурсу);
- альтернатива (можливість багатомаршрутного виконання операцій);

TM – визначає спеціалізацію устаткування при виконанні операцій;

S_D – початковий стан предметів виробництва (готовність предметів виробництва);

S_M – стан засобів виробництва (готовність засобів виробництва до виконання операцій).

Для моделювання та верифікації матеріально-інформаційних процесів ГВС найбільше застосування знайшов математичний апарат сіток Петрі.

3.2. Базовий апарат сіткових моделей дискретних виробничих процесів

3.2.1. Основні визначення та правила роботи сіток Петрі

Ефективним засобом моделювання дискретних процесів є сітки Петрі. Їх основні властивості полягають у можливості відображення паралелізму, асинхронності, ієрархічності об'єктів, що моделюються, більш простішими засобами. Тому використання сіток Петрі для дослідження ієрархічних дискретних систем, зокрема ГВС, є найкращим.

Сітка Петрі – причинно-наслідкова модель представлення подій, що виникають у процесі роботи дискретної системи.

Процеси, які моделюються, подаються як множина подій та умов, між якими встановлені причинно-наслідкові зв'язки.

Події – це дії, послідовність настання яких керується станами системи.

Стан системи визначається сукупністю умов, серед яких виділяють:

- 1) доумови – умови, пов'язані з фактом настання події;
- 2) постумови – умови, пов'язані з фактом здійснення події.

Умови можуть виконуватися або не виконуватися. Тільки виконана умова є підставою для настання події, результатом здійснення якої є виконання іншої умови. В результаті послідовність настання подій породжує зміну станів системи, доумови подій відтворюють причинні зв'язки, а постумови – наслідкові зв'язки у системі.

Таким чином, сітка Петрі представляється сукупністю

пов'язаних подій та умов, що виникають у системі, яка моделюється, а правила функціонування сітки є способом відображення концепції причинно-наслідкових зв'язків між умовами і подіями у системі.

Для задання сіток Петрі найчастіше використовують такі способи, як теоретико-множинне визначення, графічне представлення та матричне подання.

1. Теоретико-множинне визначення сіток Петрі.

Формально сітка Петрі N може бути задана у вигляді наступної п'ятірки елементів:

$$N = (P, T, F, H, \mu_0),$$

де $P = \{p_1, p_2, \dots, p_n\}$ – кінцева непорожня множина позицій (умов);

$T = \{t_1, t_2, \dots, t_m\}$ – кінцева непорожня множина переходів (подій);

$$P \cap T = \emptyset;$$

$F: P \times T \rightarrow \{0, 1, 2, \dots\}$ – функція, що, визначивши доумови здійснення подій, призначає кожному переходу вхідну множину позицій $\bullet t_j = \{p_i \mid F(p_i, t_j) \neq 0\}$;

$H: T \times P \rightarrow \{0, 1, 2, \dots\}$ – функція, що, визначивши постумови, призначає кожному переходу вихідну множину позицій $t_j \bullet = \{p_i \mid H(t_j, p_i) \neq 0\}$;

$\mu_0: P \rightarrow \{0, 1, 2, \dots\}$ – початкове маркування (стан), яке через кількість $\mu_0(p_i)$ маркерів у позиції p_i сітки визначає виконання умов.

Необхідно зауважити, що перші чотири елементи визначають структуру системи, а останній елемент – динаміку поведінки системи, її початковий стан. Динаміка сітки пов'язана з рухом маркерів по позиціях у результаті спрацювань переходів, внаслідок чого маркуються нові позиції $\mu(p_i)$.

2. Графічне представлення сіток Петрі

Графічно сітка Петрі – це дводольний орієнтований мультиграф, де:

– дводольність означає наявність двох типів вершин (позицій та переходів);

– орієнтованість означає, що всі дуги мають певний напрямок;

– мультиграф – дуги можуть мати кратність (вона позначається значенням над дугою або кількістю дуг).

Графічне представлення пов'язане з теоретико-множинним визначенням наступним чином (рис. 3.1):

1) позиції відображаються кругами;

2) переходи відображаються рисками;

3) функції F і H – орієнтованими дугами, кількість або кратність яких визначається значенням функцій;

4) маркування сітки відображається кількістю маркерів у

позиціях.

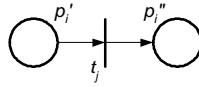


Рис. 3.1. Елементи сітки Петрі

3. Матричне подання.

Матричне подання – це аналітичний спосіб представлення сітки Петрі. У цьому випадку функції інцидентів та початкове маркування зображаються у вигляді матриць розміром $[n \times m]$:

$$F = [F_{ij} | i = \overline{1, n}; j = \overline{1, m}]; \quad H = [H_{ij} | i = \overline{1, n}; j = \overline{1, m}]$$

та вектором-стовпцем розміром $[n \times 1]$:

$$\mu_0 = [\mu_{0i} | i = \overline{1, n}],$$

де $F_{ij} = F(p_i, t_j)$; $H_{ij} = H(t_j, p_i)$; $\mu_0 = \mu_0(p_i)$.

4. Правила роботи сітки Петрі.

Робота (функціонування) сіток Петрі визначається як послідовність спрацьовування переходів, внаслідок яких відбувається зміна маркувань позицій.

Перехід може спрацьовувати, якщо він збуджений.

Перехід t_j вважається збудженим, якщо виконується наступна умова:

$$\forall p_i \in {}^{\bullet}t_j : \mu(p_i) \geq F(p_i, t_j),$$

тобто виконуються доумови здійснення модельованої цим переходом події – у кожній вхідній позиції переходу кількість маркерів не менша кратності дуги, що їх з'єднує (рис. 3.2).

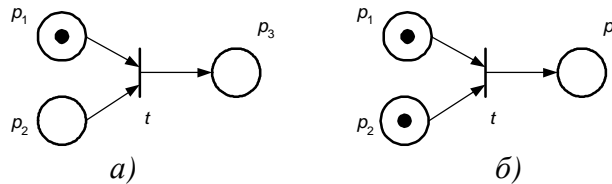


Рис. 3.2. Умови збудження переходу:

а – умова не виконана; б – умова виконана

Тоді правило спрацьовування збудженого переходу t_j має вигляд:

$$\forall p_i \in {}^{\bullet}t_j \cup t_j^{\bullet} : \mu'(p_i) = \mu(p_i) + H(t_j, p_i) - F(p_i, t_j),$$

тобто при спрацьовуванні збудженого переходу маркування μ замінюється маркуванням μ' за таким правилом: з вхідних позицій переходу забирається певна кількість маркерів, яка визначається функцією $F(p_i, t_j)$, а вихідні позиції переходу отримують іншу кількість маркерів, яка визначається вже функцією $H(t_j, p_i)$ (рис. 3.3).

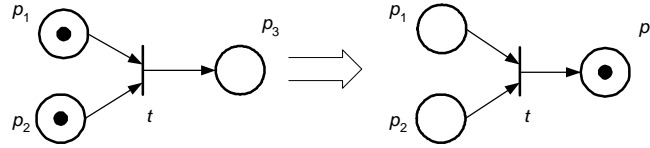


Рис. 3.3. Правило спрацьовування збудженого переходу

Необхідно відмітити, що у випадку використання матричного способу подання сіток Петрі умова збудження переходу t_j має вигляд:

$$\mu \geq F \times U,$$

а правило його спрацьовування:

$$\mu' = \mu + (H - F) \times U,$$

де $U = [U_j | j = \overline{1, m}]$ – вектор-стовпець розміром $[m \times 1]$, у якого всі елементи дорівнюють 0, крім $U_j = 1$.

У будь-якому стані сітки Петрі може існувати декілька одночасно збуджених переходів. Але послідовність їх спрацьовування не встановлена і може бути будь-якою, але без одночасного спрацьовування переходів. Тому в сітках Петрі визначають декілька прийнятних послідовностей спрацьовувань переходів, що породжують послідовності виникаючих маркувань. Це відображає паралелізм та недетермінізм сіток Петрі.

Таким чином, з функціонуванням сітки Петрі пов'язують дві послідовності:

- 1) послідовність спрацьовуючих переходів;
- 2) послідовність виникаючих (досяжних) маркувань.

Ці послідовності є взаємозв'язаними.

Два маркування μ і μ' вважаються безпосередньо досяжними, якщо у функціонуванні сітки існує перехід t_j , спрацьовування якого переводить сітку з μ у μ' (рис. 3.4).

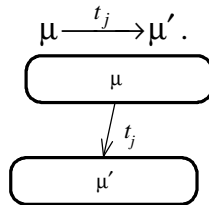


Рис. 3.4. Безпосередньо досяжне маркування

Два маркування μ і μ' вважаються досяжними, якщо у функціонуванні сітки існує послідовність переходів $G = (t_{j1}, t_{j2}, \dots, t_{jk})$, яка переводить сітку з μ у μ' :

$$\mu \rightarrow \mu',$$

тобто виникає послідовність безпосередньо досяжних маркувань (рис. 3.5):

$$\mu \xrightarrow{t_{j1}} \mu_1 \xrightarrow{t_{j2}} \dots \xrightarrow{t_{jk-1}} \mu_k \xrightarrow{t_{jk}} \mu'$$

Таким чином, формально функціонування сітки подається:

– мовою сітки Петрі $L(N)$ – множиною послідовностей

спрацьовуючих переходів;

– множиною досяжності $R(N)$ – множиною маркувань, досяжних з початкового маркування.

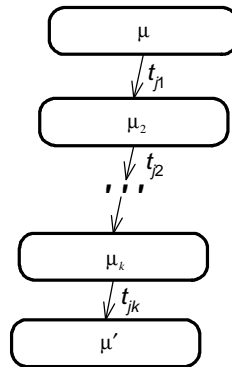


Рис. 3.5 Послідовність досяжних маркувань

Ці послідовності об'єднуються у рамках єдиної моделі представлення роботи сітки – графа досяжності – орієнтованого графа, вершинами якого є маркування з множини $R(N)$, а дугами – спрацьовуючі переходи з $L(N)$ (рис. 3.6, б). Граф з відсутніми циклами є деревом досяжності (рис. 3.6, а).

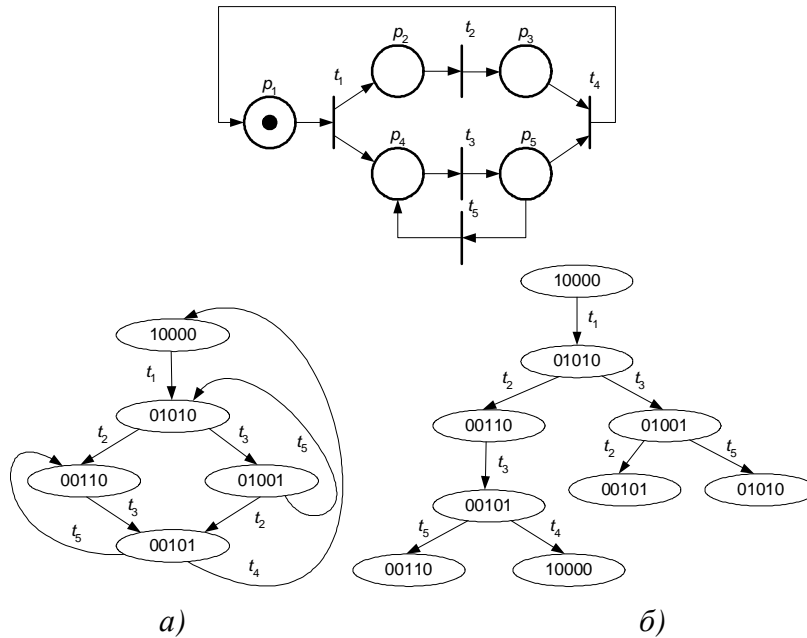


Рис. 3.6. Дерево та граф досяжності сітки Петрі:
а – граф досяжності; б – дерево досяжності

3.2.2. Властивості сіток Петрі

За допомогою сіток Петрі можна моделювати широкий клас об'єктів і систем, якщо представити належним чином взаємодію різних процесів, що можуть у них виникати. Проте, здійснення тільки моделювання не є кінцевою метою у проведенні дослідження об'єктів.

Необхідне також проведення аналізу їх поведінки.

Отже, розглянемо основні властивості сіток Петрі, які дозволяють провести якісний аналіз поведінки об'єктів, що моделюються сітками.

1. Обмеженість сітки.

Позицію p_i називають k -обмеженою, якщо кількість маркерів у цій позиції не перевищує деяке число k для всіх досяжних маркувань з множини $R(N)$, тобто

$$\forall \mu \in R(N) \exists k : \mu(p_i) \leq k.$$

У свою чергу, сітка Петрі є k -обмеженою, якщо всі позиції сітки є k -обмеженими, тобто якщо у сітки позиції p_i обмежені числами k_1, k_2, \dots , то в цілому сітка Петрі буде k -обмеженою, причому $k = \max\{k_1, k_2, \dots\}$.

У випадку, коли $k=1$, отримуємо окремий випадок обмеженості – безпечна позиція та безпечна сітка.

Обмеженість свідчить про кінцевий стан окремих елементів системи, яка моделюється сіткою Петрі, а безпечність визначає факт виконання умов при роботі об'єкта моделювання.

2. Збереженість сітки.

Сітка Петрі називається збереженою по відношенню до вагового вектора $\lambda = [\lambda_i | \lambda_i \geq 0, i = \overline{1, n}]$, якщо для кожного $\mu \in R(N)$ виконується умова:

$$\sum_{i=1}^n \lambda_i \mu(p_i) = \sum_{i=1}^n \lambda_i \mu_0(p_i) = const.$$

Якщо $\lambda_i=1$ для всіх $i = \overline{1, n}$, то сітка називається точно збереженою.

Необхідною умовою збереженості є обмеженість сітки, а достатньою – наявність вектора λ . Сітка є збереженою, якщо існує розв'язок системи $RM \cdot S = const$, де розв'язком є вектор S , а RM – матриця досяжних маркувань з $R(N)$.

Збереженість сітки свідчить про неможливість знищення або виникнення додаткових ресурсів у системі, що моделюється.

3. Живучість сітки.

Живучість – це властивість, що пов'язана з відсутністю тупикових ситуацій та зациклювань процесу функціонування сітки.

Об'єкти, моделі яких мають властивість живучості, можуть переходити з будь-якого досяжного стану в інший, у тому числі і в початковий.

Сітку вважають живою, якщо виконуються дві умови:

$$1) \forall t_j \in T \exists \mu_l, \mu_k \in R(N) : \mu_l \xrightarrow{t_j} \mu_k,$$

тобто будь-який перехід повинен спрацьовувати при моделюванні роботи сітки;

$$2) \forall \mu_l, \mu_k \in R(N): \mu_l \longrightarrow \mu_k,$$

тобто в сітці існує взаємодосяжність маркувань, у тому числі і початкового маркування, що визначає відсутність тупиків та зациклювань у роботі сітки.

3.3. Модифікації сіткових моделей дискретних виробничих процесів

3.3.1. Призначення класів сіткових моделей дискретних виробничих процесів

Сітки Петрі, як математичний засіб опису досліджуваних дискретних об'єктів, відображають логічну послідовність подій, що виникають у процесі їх функціонування, до яких відносять і виробничі системи. Ця послідовність подій має характеристики паралельних процесів, кількісний аналіз яких потребує визначення у моделі часових характеристик об'єкта моделювання, властивостей дискретних об'єктів, що динамічно змінюються, фіксованого (пріоритетного) порядку виконання запланованих дій тощо.

Крім класичного апарату сітки Петрі, для опису роботи виробничих багатомітенклатурних систем використовуються такі її модифікації:

1. Кольорові сітки.

В цих сітках маркери є засобами представлення динамічних об'єктів, які характеризуються набором атрибутів. Кожному атрибуту ставиться у відповідність колір маркера, а правила збудження та спрацювання переходів враховують комбінації кольорів, що з'являються у позиціях сітки. Правило збудження переходу у кольоровій сітці доповнюється умовою, що передбачає вибір маркерів тільки визначених кольорів з вхідних позицій цього переходу. Спрацювання переходу доповнюється умовою розміщення у вихідних позиціях маркерів відповідних кольорів.

При моделюванні роботи дискретних процесів виробничої системи кольорові маркери можуть відображати різні елементи матеріальних потоків, зокрема типи оброблюваних деталей, інструмент або елементи інформаційних потоків, наприклад, функціональний стан устаткування.

2. Часові сітки.

Ці сітки використовуються для реалістичного відображення процесів та подій, які вимагають для свого звершення певного часу.

В часових сітках кожній позиції призначається час затримки маркера, який враховується при виконанні процедури перевірки умови збудження переходів, тобто якщо маркер надійшов до позиції, то протягом встановленого часу цей маркер не враховується у правилі збудження будь-якого переходу сітки, пов'язаного вхідною дугою з цією позицією.

Часові сітки частіше використовуються у випадках, коли необхідно виконати певні розрахунки часу роботи модельованих об'єктів або виконати кількісний аналіз роботи системи.

Сітки Петрі виду (P, T, F, H, Z, μ_0) називаються часовими сітками Петрі, де $Z: P \rightarrow R^+$ – час затримки маркерів у позиціях, R^+ – множина додатних дійсних чисел.

Функціонування часових сіток підпорядковується наступним правилам:

1) маркери в позиціях можуть перебувати в двох станах: доступному і недоступному;

2) переходи $t_j \in T$ вважаються ініційованими, якщо для них виконані умови збудження $\forall p_i \in \bullet t_j: \mu(p_i) \geq F(p_i, t_j)$ і у кожній позиції є відповідне цій умові число доступних маркерів;

3) переходи спрацьовують миттєво у той самий момент, коли вони виявляються ініційованими. При спрацьовуванні з кожної позиції $p_i \in \bullet t_j$ видаляються $F(p_i, t_j)$ доступних маркерів, а в позиції $p_r \in t_j^\bullet$ додаються $H(t_j, p_r)$ маркерів;

4) кожен маркер, що зробив перехід з p_i у p_r , буде недоступним у позиції p_r протягом часу Z_r починаючи з моменту його появи в p_r , тобто Z_r – час блокування маркера в позиції p_r , по закінченні якого маркер знову стає доступним.

3. Предикатні (пріоритетні) сітки.

Ця модифікація сіток Петрі використовується для моделювання детермінованої поведінки виробничих систем. В цих сітках вибір переходу, котрий повинен спрацьовувати, задається предикатом (пріоритетом), який додатково визначає умови збудження кожного переходу, що перебуває у конфлікті за маркери сітки відносно інших збуджених переходів. Таким чином, предикат визначає пріоритетність надання маркерів збудженим переходам та черговість їх спрацьовування.

4. Інгібіторні сітки Петрі.

При описі алгоритмів управління ГВС сітками Петрі необхідно мати засіб, за допомогою якого забороняється повторне ініціювання операторів під час їх виконання. Для цього вводяться інгібіторні (забороняючі) дуги, що виконують перевірку сітки на нульове маркування. Сітки, що мають інгібіторні дуги, називаються інгібіторними.

Інгібіторна сітка представляє собою сітку Петрі, доповнену спеціальною функцією інцидентності $F_t: P \times T \rightarrow \{0,1\}$, яка вводить інгібіторні дуги для тих пар (p, t) , у яких $F_t(p, t) > 0$.

В результаті умова збудження переходу t в інгібіторній сітці має наступний вигляд:

$$\forall p \in \bullet t \bigcup \{p \mid F_t(p, t) > 0\} : \mu(p) \geq F(p, t) \& \mu(p) F_t(p, t) = 0.$$

Інгібіторні дуги зв'язують тільки позиції з переходами (на рисунках їх відображають не стрілками, а маленькими колами) (рис. 3.7). Спрацьовування переходу в інгібіторній сітці виконується відповідно до правил простої сітки Петрі, тобто позиції, пов'язані інгібіторними дугами з спрацьованим переходом не змінюють свого маркування.

В загальному випадку інгібіторні дуги можуть мати кратність $F_i : P \times T \rightarrow \{0, 1, 2, \dots\}$, що призводить в умові збудження переходу до перевірки $\mu(p) < F_i(p, t)$ на неперевищення кількості маркерів значення кратності дуги.

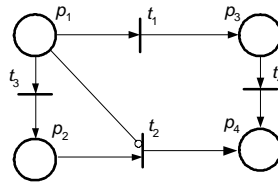


Рис. 3.7. Інгібіторна сітка

5. Ієрархічні сітки Петрі.

Ці класи сіток розглядаються як моделі ієрархії процесів або алгоритмів управління у дискретно-подійних системах [19].

Так, алгоритми подаються з n ієрархічних рівнів, де деякі або всі оператори i -го рівня самі є алгоритмами, що складаються з операторів, віднесених до $(i - 1)$ -го рівня, $2 \leq i \leq n$. Алгоритми кожного рівня описують своїми кольоровими сітками Петрі. При цьому підалгоритми i -го рівня подаються простими структурними елементами сітки Петрі (позицією або переходом) в одній із сіток i -го рівня, а підлеглий цьому алгоритму підалгоритм (оператор) представляється сіткою $(i - 1)$ -го рівня.

Така особливість ієрархічної сітки як моделі ієрархії обумовлена прийнятим способом інтерпретації алгоритму оперативного управління, коли операторам (діям) співставлені позиції сітки Петрі. Це відрізняє її від інших ієрархічних моделей, що базується на представленні дій переходами. Використання для опису підалгоритмів кожного рівня ієрархії кольорових сіток Петрі відрізняє ці сітки від простих ієрархічних моделей.

3.3.2 Кольорові сітки Петрі

Кольорові сітки Петрі є однією з найбільш важливих модифікацій традиційних сіток Петрі. За їх допомогою вдається досить компактно моделювати складні системи з великим числом різних динамічних об'єктів.

До таких систем, зокрема, відносяться багатоміністерні багатостанційні гнучкі виробництва.

Кольорова сітка Петрі (КСП) формально визначається як набір виду:

$N = (P, T, \Omega, F, H, \lambda, \psi, \mu_0)$,
 де $P = \{p\}$ – непорожня кінцева множина позицій;
 $T = \{t\}$ – непорожня кінцева множина переходів;
 $\Omega = \{\omega\}$ – непорожня кінцева множина кольорів маркерів;
 $F: P \times T \rightarrow \{0, 1, 2, \dots\}$ і $H: T \times P \rightarrow \{0, 1, 2, \dots\}$ – функції
 інцидентності множин позицій і переходів відповідно;
 $\lambda: (P \times \Omega) \times T \rightarrow (0, 1)$ – функція розподілу кольорів маркерів по
 вхідних позиціях переходів сітки;
 $\psi: T \times (P \times \Omega) \rightarrow (P \times \Omega)$ – функція розподілу кольорів маркерів
 по вихідних позиціях переходів сітки;
 $\mu_0: (P \times \Omega) \rightarrow \{0, 1, 2, \dots\}$ – початкове маркування сітки.

Функція λ до визначає умови збудження, а ψ – правило спрацьовування переходів, внаслідок чого здійснюється розподіл кольорів маркерів по позиціях сітки при її функціонуванні.

Умова збудження переходу $t \in T$ при деякому маркуванні μ має вид:

$$\begin{aligned}
 &\forall p_{i_k} \in \bullet t \exists \omega_{j_k} \in \Omega: \mu(p_{i_k}, \omega_{j_k}) \geq F(p_{i_k}, t) \& \\
 &\lambda_t((p_{i_1}, \omega_{j_1}), \dots, (p_{i_n}, \omega_{j_n})) = 1, k = \overline{1, n}, n = |\bullet t|
 \end{aligned} \tag{3.1}$$

Для кожного $t \in T$, що задовольняє умові (3.1), у загальному випадку функція λ_t визначає множину допустимих вхідних розподілів кольорів маркерів, при яких ця умова виконується

$$\bullet C_t = \left\{ \left[(p_{i_1}, \omega_{j_1}), \dots, (p_{i_n}, \omega_{j_n}) \right] \mid p_{i_k} \in \bullet t, k = \overline{1, n}, n = |\bullet t| \right\}.$$

Функція ψ_t для кожного допустимого вхідного розподілу з $\bullet C_t$ ставить у відповідність єдиний допустимий вихідний розподіл кольорів маркерів з множини

$$C_t^\bullet = \left\{ \left[(p_{i_1}, \omega_{j_1}), \dots, (p_{i_m}, \omega_{j_m}) \right] \mid p_{i_l} \in t^\bullet, \omega_{j_l} \in \Omega, l = \overline{1, m}, m = |t^\bullet| \right\},$$

тобто визначає результат спрацьовування переходу і правило формування нового маркування сітки

$$\psi_t((p_{i_1}, \omega_{j_1}), \dots, (p_{i_n}, \omega_{j_n})) = [(p_{i_1}, \omega_{j_1}), \dots, (p_{i_m}, \omega_{j_m})].$$

Тоді умова спрацьовування переходу t , збудженого при вхідному розподілі кольорів маркерів $\bullet C_t$, в наслідок чого буде утворено нове маркування, має вид:

$$\begin{aligned}
 &\forall p_{i_k} \in \bullet t \mathbf{U} t^\bullet \exists \omega_{j_k} \in \Omega: \\
 &\mu'(p_{i_k}, \omega_{j_k}) = \mu(p_{i_k}, \omega_{j_k}) - \lambda_t((p_{i_k}, \omega_{j_k})) F(p_{i_k}, t) + \varphi_t((p_{i_k}, \omega_{j_k})) H(t, p_{i_k}), \\
 &\varphi_t((p_{i_k}, \omega_{j_k})) = \begin{cases} 1, \psi_t(\bullet C_t) = [(p_{i_k}, \omega_{j_k})] \\ 0, \psi_t(\bullet C_t) \neq [(p_{i_k}, \omega_{j_k})] \end{cases}, k = \overline{1, n}, n = |\bullet t \mathbf{U} t^\bullet|
 \end{aligned}$$

На рис. 3.8 представлений приклад графічного зображення КСП

для двох кольорів: ω_1 і ω_2 (позначені відповідно «*» і «•»).

Функції λ і ψ для даної сітки представлені нижче:

для t_1 :	для t_2 :	для t_3 :
$\lambda_{t1}((p_1, \omega_1)) = 1$;	$\lambda_{t2}((p_2, \omega_1)) = 0$;	$\lambda_{t3}((p_3, \omega_1), (p_1, \omega_1)) = 0$;
$\lambda_{t1}((p_1, \omega_2)) = 0$;	$\lambda_{t2}((p_2, \omega_2)) = 1$;	$\lambda_{t3}((p_3, \omega_2), (p_1, \omega_2)) = 1$;
		$\lambda_{t3}((p_3, \omega_1), (p_1, \omega_2)) = 1$;
		$\lambda_{t3}((p_3, \omega_2), (p_1, \omega_1)) = 0$;
$\psi_{t1}(p_1, \omega_1) = (p_2, \omega_2)$;	$\psi_{t2}(p_2, \omega_2) = (p_3, \omega_1)$;	$\psi_{t3}((p_3, \omega_2), (p_1, \omega_2)) = (p_2, \omega_1)$;
		$\psi_{t3}((p_3, \omega_1), (p_1, \omega_2)) = (p_2, \omega_1)$.

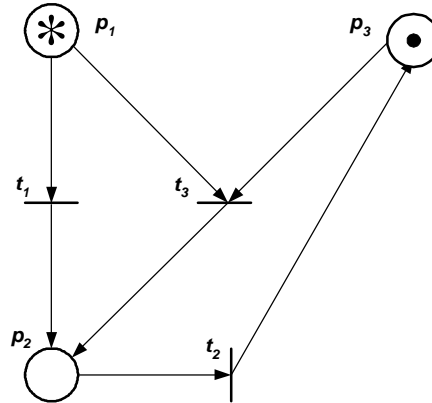


Рис. 3.8. Графічне зображення КСП

Виділимо два окремих види КСП: перероджуючі (ПС) і селективні (СС) сітки.

У *перероджуючих сітках* (ПС) спрацьовування будь-якого переходу t дозволене за наявності у всіх вхідних позиціях $p \in \bullet t$ маркерів одного кольору $\omega_j \in \Omega$. Після спрацьовування даного переходу в усіх його вихідних позиціях містяться маркери іншого кольору $\omega_g \in \Omega$.

Функція λ у ПС задається наступним чином:

$$\lambda_t(\bullet c_{t\omega_j}) = \begin{cases} 1, & \bullet c_{t\omega_j} \in \bullet C_t \\ 0 & \text{в іншому випадку} \end{cases}$$

де $\bullet c_{t\omega_j}$ – допустимий вхідний розподіл виду:

$$\bullet c_{t\omega_j} = \left[(p_{i_1}, \omega_j), \dots, (p_{i_n}, \omega_j) \right], p_{i_k} \in \bullet t, k = \overline{1, n}, n = |\bullet t|, \omega_j \in \Omega$$

Функція $\psi_t(\bullet c_{t\omega_j})$ у ПС набуває значень з області допустимих вихідних розподілів:

$$\bullet c_{t\omega_g} = \left[(p_{i_1}, \omega_g), \dots, (p_{i_m}, \omega_g) \right], \bullet c_{t\omega_g} \in C_t, p_{i_l} \in t^\bullet, l = \overline{1, m}, m = |t^\bullet|, \omega_g \in \Omega$$

Якщо $\psi_t(\bullet c_{t\omega_j}) = \bullet c_{t\omega_g}$, то перехід t при спрацьовуванні перероджує маркери кольору ω_j у вхідних позиціях в маркери кольору ω_g у вихідних позиціях.

Умова збудження переходу $t \in T$ при вхідному розподілі $\bullet c_{t\omega}$ маркування μ для ПС записується у виді:

$$\forall p \in \bullet t \exists \omega \in \Omega : \mu(p, \omega) \geq F(p, t) \& \lambda_t(\bullet c_{t\omega}) = 1 \quad (3.2)$$

Нове маркування μ' , одержане в результаті спрацювання переходу t при вхідному розподілі $\bullet c_{t\omega}$, визначається за правилом:

$$\forall p \in \bullet t \cup t \bullet \exists \omega, \omega' \in \Omega : \mu'(p, \omega) = \mu(p, \omega) - \delta_t(\omega) F(p, t) + \varphi_t(\omega') H(t, p),$$

$$\delta_t(\omega) = \begin{cases} 1, \lambda_t(\bullet c_{t\omega}) = 1 \\ 0 - \text{в іншому випадку} \end{cases}, \varphi_t(\omega') = \begin{cases} 1, \psi_t(\bullet c_{t\omega'}) = c_{t\omega'} \\ 0 - \text{в іншому випадку} \end{cases}$$

Приклад ПС з $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$ показаний на рис. 3.9. Початкове маркування μ_0 і функція ψ описуються матрицями (табл. 3.1, 3.2).

Таблиця 3.1

μ_0		Кольори			
		ω_1	ω_2	ω_3	ω_4
Позиції	p_1	1	0	0	0
	p_2	0	0	1	0
	p_3	0	0	0	0

Таблиця 3.2

ψ		Кольори			
		ω_1	ω_2	ω_3	ω_4
Переходи	t_1	ω_2	ω_3	ω_1	ω_2
	t_2	ω_4	ω_1	ω_2	ω_3
	t_3	ω_4	ω_4	ω_3	ω_2
	t_4	ω_1	ω_1	ω_4	ω_1

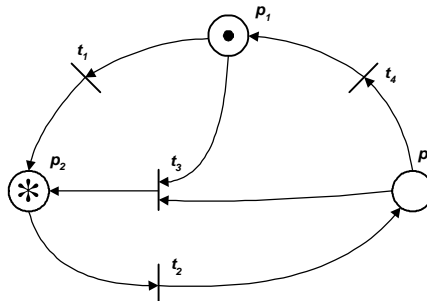


Рис. 3.9. Графічне зображення ПС

Селективною сіткою (СС) називається така сітка, у якій переходи пропускають тільки маркери певних кольорів, не змінюючи їх.

Функція λ_t для СС задається так само, як і для ПС, а функція $\psi_t(\bullet c_{t\omega_i}) = c_{t\omega_i}$.

Умови збудження переходу $t \in T$ СС і ПС при маркуванні μ аналогічні (3.2). Нове маркування μ' при спрацюванні переходу $t \in T$ по розподілу $\bullet c_{t\omega}$ визначається за правилом:

$$\forall p \in \bullet t \cup t \bullet \exists \omega \in \Omega : \mu'(p, \omega) = \mu(p, \omega) + \delta_t(\omega) (-F(p, t) + H(t, p))$$

Приклад СС для двох кольорів ω_1 і ω_2 (позначені «•» і «*») зображений на рис. 3.10. Функція λ представлена матрицею (табл. 3.3).

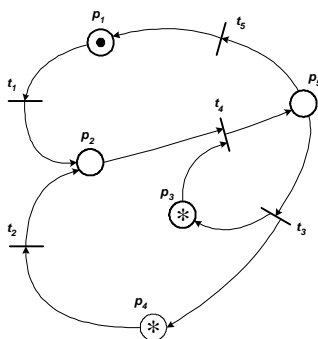


Рис. 3.10. Графічне зображення СС

Отже, у СС множину переходів T можна розбити на підмножини T_i , де $\forall t \in T_i$ пропускає маркер кольору ω_i . При цьому $T = \bigcup_{i=1}^r T_i, r = |\Omega|$.

Таблиця 3.3

λ		Кольори	
		ω_1	ω_2
Переходи	t_1	1	0
	t_2	0	1
	t_3	1	0
	t_4	0	1
	t_5	0	1

Підсіткою СС, що визначена на множині переходів $T_i \subseteq T$, називається така сітка Петрі $N_i = (P_i, T_i, F_i, H_i, \mu_{i0})$, у якій:

$$P_i = \bigcup_{t \in T_i} (\bullet t \cup t \bullet);$$

$$\forall p \in P_i \forall t \in T_i : F_i(p, t) = F(p, t);$$

$$\forall t \in T_i \forall p \in P_i : H_i(t, p) = H(t, p);$$

$$\forall p \in P_i : \mu_{i0}(p) = \mu_0(p, \omega_i).$$

Кожна підсітка N_i , є одноколірною сіткою з маркерами кольору ω_i .

Отже, аналіз СС можна звести до аналізу сукупності одноколірних сіток.

Властивості КСП. Розглянемо властивості обмеженості, безпеки, зберігаємості та живучості КСП.

Обмеженість та безпека. Будемо розрізняти безпеку та обмеженість КСП у цілому і по кожному визначеному кольору $\omega \in \Omega$. Назвемо позицію $p \in P$ *k-обмеженою* по кольору $\omega \in \Omega$, якщо існує $k \in \{0, 1, \dots\}$ таке, що на множині всіх досяжних маркувань $\mu(p, \omega) \leq k$.

Сітка Петрі є *k-обмеженою* по кольору ω , якщо всі її позиції *k-обмежені* по кольору.

Позиція $p \in P$ називається *k-обмеженою*, якщо існує

$k \in \{0, 1, \dots\}$ таке, що на множині всіх досяжних маркувань $\forall \omega \in \Omega$, $\mu(p, \omega) \leq k$.

Сітка Петрі називається *k-обмеженою*, якщо всі її позиції *k-обмежені*. Окремим випадком *k-обмеженості* сітки в цілому й обмеженості її по кольору $\omega \in \Omega$ є безпека і безпека по кольору ω (випадок $k = 1$).

Очевидно, що якщо сітка з різнокольоровими маркерами k_i обмежена по кожному кольору ω_i , то вона *k-обмежена*, де $k = \max_{i=1, |\Omega|} (k_i)$.

Вірно і зворотнє: якщо сітка з різнокольоровими маркерами *k-обмежена*, то вона *k-обмежена* по будь-якому кольору $\omega_i \in \Omega$.

Аналіз обмеженості сітки Петрі з різнокольоровими маркерами аналогічний аналізу обмеженості звичайної сітки.

Збереженість. У КСП виділимо збереженість сітки в цілому і по окремих кольорах.

КСП є збереженою по кольору ω_j , якщо існує вектор $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$, $\lambda_i > 0$ такий, що для всіх маркувань $\mu(p, \omega_j)$, досяжних з початкового $\mu_0(p, \omega_j)$, справедлива рівність:

$$\sum_{i=1}^n \lambda_i \mu(p_i, \omega_j) = \sum_{i=1}^n \lambda_i \mu_0(p_i, \omega_j).$$

Сітка Петрі з різнокольоровими маркерами називається збереженою, якщо існує вектор $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$, $\lambda_i > 0$, такий, що для кожного маркування $\mu(p, \omega)$, $\forall p \in P$, $\forall \omega \in \Omega$, досяжного з початкової сітки, справедлива рівність:

$$\sum_{i=1}^n \lambda_i \sum_{j=1}^{|\Omega|} \mu(p_i, \omega_j) = \sum_{i=1}^n \lambda_i \sum_{j=1}^{|\Omega|} \mu_0(p_i, \omega_j).$$

Якщо КСП є збереженою по будь-якому кольору $\omega \in \Omega$, то вона є збереженою в цілому.

Живучість. КСП N назовемо живою, якщо:

- 1) $\forall t \in T \exists \mu_i, \mu_j$ досяжні з початкового маркування μ_0 такі, що $\mu_i \xrightarrow{t} \mu_j$ (тобто маркування μ_j досягне з μ_i);
- 2) $\forall \mu_i, \mu_j$ досяжні з початкового маркування $\mu_i \xrightarrow{t} \mu_j$;
- 3) $\forall \omega \in \Omega \exists \mu$ досягне з початкової КСП таке, що $\exists p \in P, \mu(p, \omega) \neq 0$.

Селективна сітка N є живою по кольору ω_i , якщо сітка N_i , яка є фрагментом селективної сітки N і визначена на множині переходів T_i , жива.

Графи досяжності сіток N_i є підграфами графа досяжності СС N при $\mu_{0i}(p) = \mu_0(p, \omega_i)$. При цьому маркування μ_i, μ_j , що належать графу досяжності СС, будуть належати графу досяжності G_i сітки N_i тоді і тільки тоді, коли:

$$\mu_i(p, \omega_l) = \mu_j(p, \omega_l); \quad \mu_i(p, \omega_i) \neq \mu_j(p, \omega_i), \quad l = \overline{1, r}; \quad t \neq i.$$

Отже, селективна сітка N жива тоді і тільки тоді, коли кожна N_i , $i = \overline{1, r}$, жива.

Перероджуюча сітка, наведена на рис. 3.9, не є живою. Селективна сітка, показана на рис. 3.10, також не є живою, оскільки сітки N_1, N_2 неживі.

Відзначимо, що КСП із кінцевим числом кольорів маркерів еквівалентна звичайній сітці Петрі з точки зору можливостей відображення властивостей об'єкта моделювання. Однак використання КСП дозволяє значно скоротити кількість позицій і переходів, необхідних для опису об'єкта, що значно спрощує і прискорює процес моделювання.

Практика моделювання показує, що незважаючи на теоретичну можливість, часто не вдається через велику розмірність та складність побудувати звичайну сітку Петрі для таких об'єктів, як дільниці ГВС із значною кількістю технологічного і транспортного устаткування. Використання ж КСП істотно спрощує цю задачу моделювання.

Оцінимо збільшення обсягу даних, необхідного для опису об'єкта сіткою при переході від КСП до еквівалентної їй звичайній сітці Петрі.

Нехай для КСП $|P| = n$; $|T| = m$; $|\Omega| = r$. Тоді F – матриця розмірності $n \times m$, H – матриця розмірності $m \times n$, m – матриця розмірності $n \times r$.

Обсяг даних (число елементів) для опису КСП визначається виразом $2mn + nr$.

Щоб у сітці Петрі зберегти таку ж кількість станів, як і в еквівалентній їй КСП, необхідно кожній позиції КСП поставити у відповідність r позицій звичайній сітці. Очевидно, при цьому число переходів також збільшиться не менше, ніж у r разів. Отже, обсяг даних для її опису можна оцінити виразом $2nmr^2 + nr$.

Таким чином, оцінкою зміни обсягу даних може служити відношення:

$$\gamma = \frac{2nmr^2 + nr}{2nm + nr} = \frac{2mr^2 + r}{2m + r} = \frac{2r^2 + r/m}{2 + r/m}.$$

Очевидно, що при $r = 1$ значення $g = 1$. Для будь-якого m величина g оцінюється знизу виразом $2r^2/(2 + r)$. Це означає, що при перетворенні КСП в еквівалентну їй звичайну сітку Петрі обсяг даних зростає не менше, ніж у $2r^2/(2 + r)$ разів.

Відмітимо, що при $m \rightarrow \infty$ величина $g \rightarrow r^2$, тобто при досить великій розмірності для опису КСП потрібно в r^2 разів менше пам'яті, ніж для опису еквівалентної їй звичайній сітці Петрі.

3.3.3. Ієрархічні кольорові сітки Петрі

Перш, ніж визначити ієрархічну кольорову сітку Петрі введемо наступні поняття.

Визначення 3.1. Перехід t_l у КСП $N = (P, T, \Omega, F, H, \lambda, \psi, \mu_0)$ називається джерелом, якщо

$$\forall p \in P: F(p, t_l) = 0,$$

тобто у перехід t_l не входить ні однієї дуги.

Визначення 3.2. Перехід t_s у КСП N називається стоком, якщо

$$\forall p \in P: H(t_s, p) = 0,$$

тобто з переходу t_s не виходить ні одна дуга.

Визначення 3.3. КСП N , що зображується зв'язним графом щонайменше з одним джерелом і з одним стоком, називається блоком.

Визначення 3.4. Зв'язувальною сіткою N_S щодо блока N називають таку КСП, що утворюється з блока N додаванням до нього позиції p_0 , яка називається нульовою позицією, і дуг, що ведуть з p_0 в усі джерела і з усіх стоків в p_0 . Таким чином, зв'язуюча сітка N_S немає жодного джерела t_l і жодного стоку t_s .

Приклад блока з джерелом t_4 , t_6 і стоками t_5 , t_7 і відповідна цьому блоку зв'язуюча сітка зображені на рис. 3.11.

Визначення 3.5. Нульовим маркуванням зв'язувальної сітки N_S блока N називають таке маркування $\mu_0(p_0, \omega)$, де в нульовій позиції p_0 є по одному маркеру всіх кольорів, а всі інші позиції порожні; відповідне їй маркування блока називають "порожнім", тобто

$$\forall p \in N, \forall \omega \in \Omega: \mu(p, \omega) = 0;$$

$$\forall p \in N_S, \forall \omega \in \Omega \exists p_0 \in N_S: \mu(p_0, \omega) = 1.$$

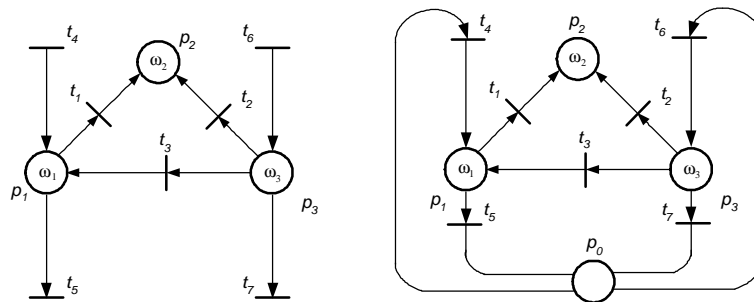


Рис. 3.11. Приклади зображення сіток:

а – сітка у вигляді блоку; б – зв'язувальна сітка

Визначення 3.6. Блок N називається коректно сформованим, якщо:

- його зв'язувальна сітка N_S при початковому нульовому маркуванні $\mu_0(p_0, \omega)$ безпечна в цілому (по всіх кольорах), тобто

$$\forall p \in N_S, \forall \omega \in \Omega: \mu(p, \omega) \leq 1;$$

- будь-яка послідовність спрацьованих переходів, яка реалізована в блоці з порожнього початкового маркування $\mu_0(p_0, \omega)$, закінчується порожнім маркуванням тільки за умови, якщо кількість спрацьовань переходів t_l і t_s однакова;

- як початкові маркування $\mu_{поч}(p, \omega)$ використовуються лише такі, для яких мають місце співвідношення:

$$\forall \mu_{\text{поч}}(p, \omega) \in R(N) : \mu_0(p_0, \omega) \xrightarrow{t_l} \mu_{\text{поч}}(p, \omega);$$

$$\forall p_0 \in \bullet t_l \exists \lambda_{t_l}(p_0, \omega) : \lambda_{t_l}(p_0, \omega) = 1.$$

Звичайно передбачається, що ніякі два переходи в сітці Петрі не можуть спрацювати одночасно.

На відміну від цього, для деяких переходів не тільки допускається одночасне спрацювання, але, навпаки, забороняється неодночасне спрацювання. Такі переходи утворюють перехід-зв'язку.

Визначення 3.7. Переходом-зв'язкою називають підмножину переходів $\% = \{t_j^1, \dots, t_j^n\}$, при якій з будь-якої їх вхідної позиції дуга веде тільки в один з переходів, що знаходяться у зв'язці.

Спрацювання переходу-зв'язки $\%$ може відбутися тільки тоді, коли порушені усі вхідні в нього переходи.

У сітці Петрі переходи-зв'язки завжди можна замінити простими переходами. Приклад сітки Петрі з переходами-зв'язками $\%_1 = \{t_1^1, t_1^2\}$, $\%_2 = \{t_2^1, t_2^2\}$ та їх заміною простими переходами показаний на рис. 3.12.

За допомогою переходів-зв'язок легко відображати зв'язки між алгоритмами, підалгоритмами та операторами.

Використовуючи наведені поняття, дамо визначення ієрархічної кольорової сітки Петрі (ІКСП).

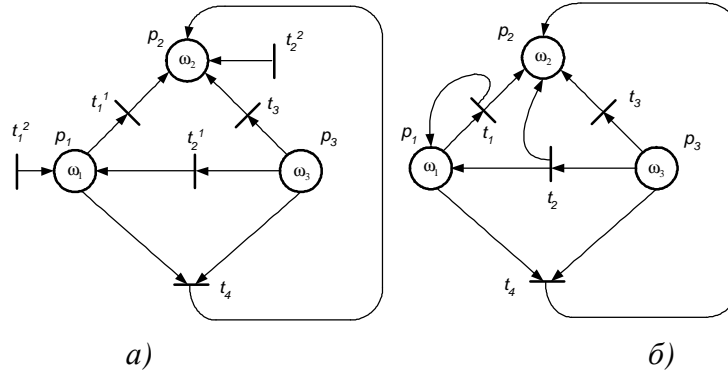


Рис. 3.12. Сітка Петрі

з переходами-зв'язками (а) і простими переходами (б)

Визначення 3.8. Ієрархічною кольоровою сіткою Петрі називається набір $N^* = (N, Q)$, який складається з кінцевої множини коректно сформованих блоків $N = \{N_i \mid N_i = (P_i, T_i, \Omega_i, F_i, H_i, \lambda_i, \psi_i, \mu_{0i}), i = 0, 1, 2, \dots\}$, що частково впорядковано у відповідності з відношення $Q : N \rightarrow N$.

Відношення Q між блоками графічно представляється деревом, кореню якого відповідає старша сітка N_0 , що може і не бути коректно сформованим блоком (рис. 3.13, а).

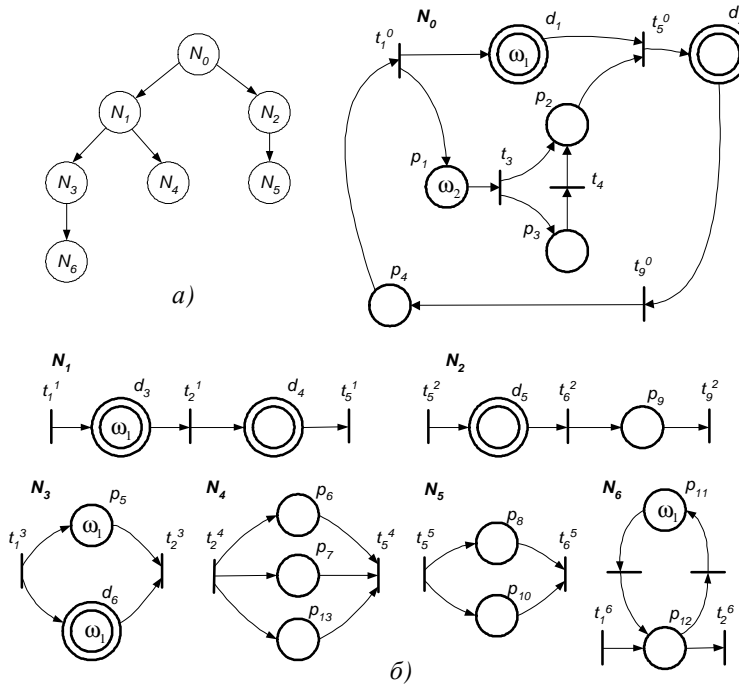


Рис. 3.13. Ієрархічна кольорова сітка Петрі:
а – дерево відношення; б – зображення сітки

Для будь-яких двох сіток N_u і N_v , що відповідають початковій і кінцевій вершинам дуги дерева відношення сітки N^* , присутня особлива позиція $d_v \subseteq N_u$, що називається дублером блока N_v .

Позиція-дублер блока N_v в ІКСП зображається подвійним кругом із символом d_v . Блок N_v дублюється позицією-дублером d_v , що належить тільки сітці N_u . Таким чином, індекси дублера і дублюючого ним блока завжди збігаються. Між входними (вихідними) переходами дублера і джерелами (стоками) дублюючого ним блока має місце взаємнооднозначна відповідність. При цьому кожен входний (вихідний) перехід дублера і співставлене йому джерело (стік) у дублюючому блоці входять в один перехід-зв'язку.

У старшій ІКСП, показаній на рис. 3.13, б, містяться дублери d_1 і d_2 блоків N_1 і N_2 , у блоці N_1 – дублери d_3 і d_4 блоків N_3 і N_4 , у блоці N_2 – дублер d_5 блока N_5 , у блоці N_3 – дублер d_6 блока N_6 .

У ІКСП існують наступні переходи-зв'язки:

$$\mathcal{P}_1^0 = \{t_1^0, t_1^1, t_1^2, t_1^3, t_1^6\}; \quad \mathcal{P}_2^0 = \{t_2^1, t_2^3, t_2^4, t_2^6\};$$

$$\mathcal{P}_5^0 = \{t_5^0, t_5^1, t_5^2, t_5^4, t_5^5\}; \quad \mathcal{P}_6^0 = \{t_6^3, t_6^5\};$$

причому входному переходу t_1^0 дублера d_1 в старшій сітці N_0 співставлене джерело t_1^1 в блоці N_1 , а вихідному переходу t_5^0 – стік t_5^1 , входному переходу t_1^1 дублера d_3 в блоці N_1 співставлене джерело t_1^3 в блоці N_3 , а вихідному переходу t_2^1 – стік t_2^3 і т.д.

Визначення 3.9. Початкові маркування блоків N_i і старшої сітки

N_0 в ІКСП $N = \{N_i\}$ погоджені наступним чином:

- якщо всі позиції дублюючого блока порожні, то в дублері маркер відсутній, тобто

$$\forall p \in N_i \forall \omega \in \Omega_i : \mu_i(p, \omega) = 0 \Rightarrow \mu(d_i, \omega) = 0;$$

- якщо не всі позиції дублюючого блоку порожні, то в дублері знаходиться маркер того ж кольору, що й у дублюючому блоці і, крім цього,

$$\forall p_j \in \cdot t_s^i \subseteq N_i \forall \omega \in \Omega_i : \lambda_{t_s}(d_i, \omega) = \lambda_{t_s}((p_{j_1}, \omega), \dots, (p_{j_n}, \omega)), n = |\cdot t_s^i|.$$

Для алгоритмів управління ГВС позиції-дублери ІКСП означають наступне: кожному дублеру відповідає підалгоритм, що є алгоритмом, опис якого задається дублюючим блоком. Якщо в дублюючому блоці міститься свій дублер, то йому також відповідає підалгоритм, але вже більш низького рівня складності, і так доти, поки якому-небудь дублеру не буде відповідати оператор. Співвідношення алгоритмів за складністю визначається деревом відношень: рівень складності підалгоритму тим вищий, чим ближче до кореня дерева знаходиться сітка, яка містить дублер, що відповідає цьому підалгоритму.

Як підалгоритми алгоритму управління ГВС можна виділити алгоритми управління АТСС, роботом-штабелером, автономним транспортним модулем тощо.

Таким чином, ієрархічна сітка Петрі дозволяє у явному вигляді відбивати ієрархію операторів в алгоритмах управління ГВС і використовувати її при описі алгоритмів.

З метою модифікації застосуємо до ІКСП дві операції: заміщення дублера блоком і заміщення вузла дублером.

Операція заміщення дублера блоком полягає в наступному: сітка з дублером, що заміщається, поєднується з дублюючим блоком шляхом заміни відповідних переходів-зв'язок еквівалентними їм простими переходами; з отриманої сітки віддаляється позиція-дублер разом з інцидентними з нею дугами.

Приклад виконання цієї операції наведений на рис. 3.14. Ієрархічна сітка (рис. 3.14, а) після об'єднання сітки, що містить дублер d_1 , з дублюючим блоком N_1 перетворюється в сітку, зображену на рис. 3.14, б. Після видалення позиції d_1 та інцидентних їй дуг отримуємо сітку, показану на рис. 3.14, в.

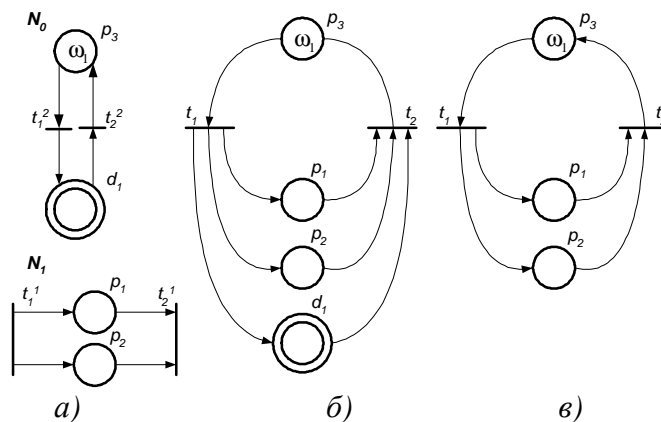


Рис. 3.14. Ієрархічна сітка (а) і приклад операцій заміщення дублера (б) і вузла (в)

В результаті послідовного заміщення дублерів (у довільному порядку) з вихідної ІКСП виходить проста сітка. Зміст операції заміщення дублера полягає в тому, що, маючи множину операторів у підалгоритмах у вигляді окремих блоків і застосовуючи суперпозицію, можна побудувати алгоритм управління ГВС у вигляді простої сітки Петрі.

Операція заміщення локалізованого вузла є оберненою до операції заміщення дублера і дозволяє з простої сітки Петрі отримати ієрархічну.

3.4. Засоби верифікації та аналізу властивостей сіткових моделей

Сітки Петрі, як математичний засіб опису досліджуваних дискретних об'єктів, відображають логічну послідовність подій, що виникають у процесі їх функціонування, до яких відносять і виробничі системи.

Ця послідовність подій має характеристики паралельних процесів, аналіз яких потребує визначення наступних умов:

- 1) повторна ініціалізація операцій в процесі допустима тільки після завершення попереднього її виконання у системі;
- 2) використання в дискретній системі ресурсів має кількісне обмеження по застосуванню;
- 3) у взаємодії частин паралельних процесів не повинно з'являтися блокувань розвитку подій, тупикових станів або зациклювань у виконанні послідовних дій.

Перевірка цих умов практично означає встановлення властивостей динаміки роботи сітки Петрі, а саме обмеженості, безпечності та життєвості.

Розглянемо сітку Петрі як математичну систему для якісного аналізу динаміки роботи дискретних об'єктів.

Так структурні елементи сітки повинні інтерпретуватися

наступним чином: позиції – це умови виконання або ознаки здійснення дій (операцій) на об'єкті; переходи – це події, що здійснюються на об'єкті у випадку виконання необхідних умов; маркування – це поточний стан об'єкта.

Тому статичні властивості об'єкта визначає графова частина сітки Петрі, а динамічні – початкове маркування і правила збудження-спрацьовування переходів (моделювання роботи).

При описі процесів сітками Петрі будь-яке маркування сітки асоціюється з визначеним станом моделюемого об'єкта. Спрацьовування будь-якого переходу t_j сітки N пов'язане зі зміною його стану (виконанням елементарної дії у процесі).

Перевірка властивостей сіток може виконуватись шляхом:

- 1) дослідження структури дерева досяжності;
- 2) дослідження матричного подання сітки Петрі;
- 3) редукційного аналізу структури і динаміки роботи сітки

Петрі.

3.4.1. Дослідження дерева досяжності

Початковому маркуванню відповідає коренева вершина дерева, а дуги, відмічені переходами t_j , з'єднують вершини-маркування, що є безпосередньо досяжними при спрацьовуванні t_j .

Дерево досяжності в загальному випадку може бути нескінченним з вершинами таких типів:

- внутрішнє маркування, яке є досяжним з початкового маркування та не є тупиковим (рис. 3.15, а);
- тупикове маркування (рис. 3.15, б), з якого не може спрацьовувати ні один перехід;
- дублююче маркування μ_d (рис. 3.15, в), яке відповідає вже введеному раніше у дерево маркуванню $\mu = \mu_d$. Якщо на шляху з початкового маркування до μ_d зустрічається маркування $\mu = \mu_d$, то μ_d стає маркуванням-циклом або циклічним маркуванням μ_c (рис. 3.15, г);
- накопичуюче маркування μ_n , відповідно якому на шляху від початкового маркування існує інше маркування μ таке, що $\mu \leq \mu_n$ (рис. 3.15, д).

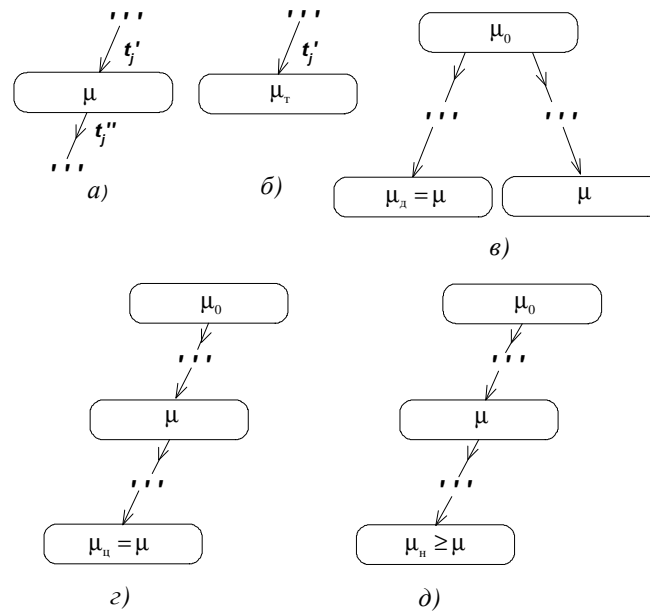


Рис. 3.15 Типи маркувань: а – внутрішнє; б – тупикове; в – дублююче; г – циклічне; д – накопичуюче

Нескінченність дерева можлива тільки у випадку існування накопичуючих маркувань, які породжують циклічне повторення однакових послідовностей спрацьовуючих переходів.

Для того щоб побудувати скінченне дерево досяжності і подати процес нескінченного накопичування маркерів у позиціях сітки, вводиться позначення у вигляді символу w , який має наступні властивості:

$$w + a = w; \quad w - a = w; \quad a < w.$$

Тоді алгоритм побудови скінченного дерева досяжності базується на наступних положеннях:

1. Послідовно обробляються вершини-маркування, які утворюються внаслідок спрацьовування переходів із початкового маркування сітки. Результатом обробки кінцевих (необроблених) вершин є їх перетворення в один з наступних типів: тупикову чи дублюючу.

2. Якщо поточне маркування μ не кваліфікується як один з двох наведених типів вершин, то μ стає внутрішньою. Для кожної внутрішньої вершини напідставі виконання умови збудження переходів і правила їх спрацьовування формується підмножина безпосередньо досяжних маркувань, котрі у дереві стають кінцевими вершинами.

Нові кінцеві маркування μ' визначаються за результатами спрацьовування збуджених у μ переходів за наступними правилами:

– якщо $\mu(p_i) = w$, то $\mu'(p_i) = w$;

– якщо на шляху з початкового маркування до μ' існує таке маркування μ'' , що $\mu'' < \mu'$ і $\mu''(p_i) < \mu'(p_i)$, то $\mu'(p_i) = w$;

– у протилежному випадку $\mu'(p_i)$ зберігає своє значення, отримане внаслідок спрацювання переходу.

3. Коли всі вершини будуть оброблені, алгоритм завершується.

Побудова скінченного дерева досяжності дозволить практично використати його для дослідження властивостей сітки Петрі.

Визначення властивостей сітки Петрі на обмеженість та безпечність за допомогою дерева досяжності передбачає дослідження кожної вершини графа на поточну кількість маркерів в кожній позиції, а живучість встановлюється за умови виконання наступних вимог:

- в позначеннях дуг дерева використані усі переходи сітки;
- немає вершин дерева, що позначають як тупикові або накопичуючі маркування;
- шляхи, що ведуть до вершин – циклічних маркувань, мають хоча б одне розгалуження, яке спрямоване до початкового маркування;
- існує хоча б один шлях повернення у вершину початкового маркування.

Проте визначити властивість зберігаємості сітки, використавши дерево досяжності не можливо.

Приклад визначення та аналізу властивостей сітки Петрі.

Подано за графічним зображенням сітки на рис. 3.16 її теоретико-множинне та матричне визначення, побудуємо дерево досяжності (рис. 3.17) і визначимо властивості сітки.

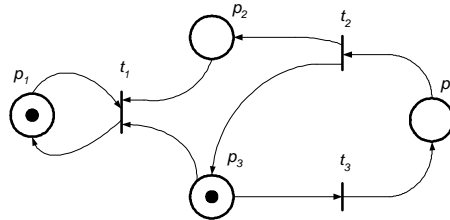


Рис. 3.16. Сітка Петрі з накопиченням маркерів

Теоретико-множинне визначення сітки:

$$P = \{p_1, p_2, p_3, p_4\};$$

$$T = \{t_1, t_2, t_3\};$$

$$F(p_i, t_1) = 1 \text{ для } i = 1, 2, 3;$$

$$F(p_4, t_1) = 0, F(p_4, t_2) = 1, F(p_i, t_2) = 0 \text{ для } i = 1, 2, 3;$$

$$F(p_3, t_3) = 1, F(p_i, t_3) = 0 \text{ для } i = 1, 2, 4;$$

$$H(t_1, p_1) = 1, H(t_1, p_i) = 0 \text{ для } i = 2, 3, 4;$$

$$H(t_2, p_i) = 1 \text{ для } i = 2, 3; H(t_2, p_i) = 0 \text{ для } i = 1, 4;$$

$$H(t_3, p_4) = 1, H(t_3, p_i) = 0 \text{ для } i = 1, 2, 3;$$

$$\mu_0(p_1) = \mu_0(p_3) = 1; \mu_0(p_2) = \mu_0(p_4) = 0.$$

Множини вхідних позицій переходів:

$$\bullet t_1 = \{p_1, p_2, p_3\}; \bullet t_2 = \{p_4\}; \bullet t_3 = \{p_3\}.$$

Множини вихідних позицій переходів:

$$t_1^\bullet = \{p_1\}; t_2^\bullet = \{p_2, p_3\}; t_3^\bullet = \{p_4\}.$$

Матричне подання:

$$F = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mu = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Позиції p_1, p_3, p_4 – безпечні, а позиція p_2 – необмежена. Тому сітка в цілому необмежена і незберігаюча.

Переходи спрацьовують усі, але є тупикове маркування μ_3 , циклічне повторення маркувань $\mu_2 \rightarrow \mu_4 \rightarrow \mu_5$ та початкове маркування недосяжне з будь-якого іншого у дереві. Тому сітка нежива.

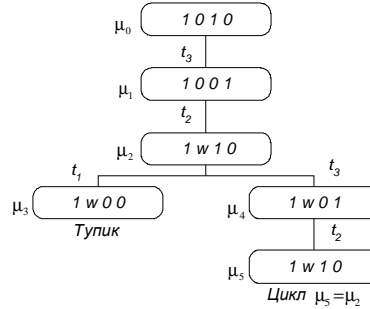


Рис. 3.17. Кінцеве дерево досяжності

3.4.2. Дослідження матричного подання сітки Петрі

Побудова і аналіз дерева досяжності потребує встановлення усіх досяжних у сітці маркувань, що є важкою задачею повного перебору станів. Тому існує інший спосіб, який представляє собою структурний аналіз сітки, що базується на матриці інцидентій та початкового маркування.

Виходячи з умови збудження та правила спрацьовування переходів, динаміку сітки в просторі станів (маркувань) можна описати рекурентним рівнянням – *рівнянням стану сітки*:

$$M_k = M_{k-1} + A \bullet U_k; k = 1, 2, \dots, \quad (3.3)$$

де M_k – стан, у який перейде сітка зі стану μ_{k-1} у результаті k -го впливу U_k (спрацьовування переходу);

$U_k = [U_{jk}]$ – керуючий вектор, компонента якого $U_{jk} = 1$, якщо в k -й момент асинхронного часу виникає спрацьовування переходу t_j , або $U_{jk} = 0$, якщо спрацьовування не відбувається;

$A = [A_{ij}]$ – матриця інцидентій позицій та переходів, елементи якої визначаються наступним чином: $A_{ij} = l$, якщо перехід t_j має l вихідних дуг до позиції p_i , $A_{ij} = -l$, якщо перехід t_j має l вхідних дуг з позиції p_i , $A_{ij} = 0$, якщо перехід t_j немає зв'язку з позицією p_i .

Матрицю A можна розрахувати на підставі операції над матрицями $F = [F_{ij}]$ і $H = [H_{ij}]$, які задають кількість дуг, що виходять відповідно з позицій і переходів:

$$A = H - F,$$

тобто елементи F_{ij} задають кількість маркерів, які потрібно забрати з

позиції p_i при спрацьовуванні переходу t_j , а елементи H_{ij} визначають кількість маркерів, які направляються у позицію p_i при спрацьовуванні переходу t_j .

Очевидно, що в будь-якому стані компоненти вектора маркування не можуть бути від'ємними. Вони можуть набувати лише нульових або додатних цілочисельних значень. Ця умова у матричному записі має вигляд: $M_k \geq 0$ для всіх k . Враховуючи останнє, з рівняння станів одержуємо $M_{k-1} + A \bullet U_k \geq 0$, а послідовність маркувань (3.3) можна замінити єдиним виразом через початкове маркування M_0 і вектор підрахунку спрацьовувань переходів сітки:

$$S = \sum_k U_k; S = [s_j | j = \overline{1, m}]. \quad (3.4)$$

Елемент s_j вказує, яку кількість разів спрацьовує перехід t_j в послідовності маркувань, яка йде від M_0 до M_k . Враховуючи рівняння (3.4), з (3.3) одержимо:

$$M_k = M_0 + A \bullet S, \quad (3.5)$$

або якщо додати вектор зміни маркування

$$dM = M_k - M_0,$$

з рівняння (3.5) одержимо рівняння:

$$A \bullet S = dM. \quad (3.6)$$

Всі можливі рішення даного рівняння можна отримати за допомогою одного з методів розв'язку задач цілочисельного програмування.

Розглянемо використання методів лінійної алгебри у розв'язанні рекурентних рівнянь зміни станів моделі сітки. Вони дозволяють на основі математичного дослідження структури біграфа сітки і початкового маркування M_0 оцінити такі якісні характеристики сітки, як обмеженість, живучість.

Цілочисельний вектор $X = [x_i | i = \overline{1, n}]$, який є розв'язком лінійної системи

$$A^T \bullet X = 0 \quad (3.7)$$

називається *p-інваріантом*.

Розглянемо рівняння (3.3), обидві частини якого помножимо на транспонований вектор X^T :

$$X^T \bullet M = X^T \bullet M_0 + X^T \bullet A \bullet S. \quad (3.8)$$

Враховуючи (3.7) і те, що $A^T \bullet X = X^T \bullet A$, з виразу (3.8) одержимо:

$$X^T \bullet M = X^T \bullet M_0, \quad (3.9)$$

тобто будь-який *p-інваріант* характеризує всі досяжні маркування сітки з точки зору збереження деяких властивостей процесів, що моделюються сіткою.

Якщо позначити

$$X^T \bullet M = K_0,$$

де K_0 – константа,

то інваріантність досяжних маркувань сітки представимо у вигляді співвідношення:

$$X^T \cdot M = K_0 = \text{const.} \quad (3.10)$$

Вектор X тому називають p -інваріантом, що він визначає властивості структури сітки у розподілі маркерів за позиціями p_i незалежно від будь-якого досяжного маркування.

Враховуючи, що система (3.7) може мати нескінченну кількість рішень, фундаментальною системою розв'язків системи лінійних однорідних рівнянь називають таку сукупність розв'язків, за допомогою якої виражаються всі інші розв'язки.

Якщо ранг матриці A дорівнює числу невідомих ($r = n$), то система (3.7) має тільки нульовий розв'язок. Якщо $r < n$, то система (3.7), крім нульового, має нескінченну множину інших розв'язків, причому фундаментальна (базисна) система складається з $(n - r)$ векторів p -інваріант.

Ранг матриці $A = [a_{ij}]$ розміром $n \times m$ дорівнює найвищому порядку відмінного від нуля визначника, одержаного викреслюванням $(n - r)$ стовпців і $(m - r)$ рядків з матриці A . Таким чином, всі інваріанти X для маркувань сітки можна отримати з $(n - r)$ базисних рішень, тобто

$X = \sum_{k=1}^{n-r} \alpha_k B_k$, де α_k - коефіцієнт лінійної комбінації векторів базисних рішень B_k . Об'єднавши записані у вигляді векторів-рядків розв'язки фундаментальної системи, одержимо матрицю інваріантів чи базисних розв'язків $B = [B_k | k = \overline{1, n-r}]$.

Тоді подібно до (3.9) для будь-якого досяжного маркування будемо мати:

$$B \cdot M = B \cdot M_0 = K_0. \quad (3.11)$$

Якщо всі компоненти p -інваріанта невід'ємні, його називають p -ланцюгом.

Повний p -ланцюг – це p -інваріант, всі компоненти якого додатні, тобто повний p -ланцюг включає всі позиції сітки.

Сітка Петрі інваріантна, якщо для неї існує повний p -ланцюг. Повний p -ланцюг потрібно шукати серед усіх рішень фундаментальної системи B або їх лінійної комбінації.

Інваріантна сітка Петрі є обмеженою, але обмежена сітка може не бути інваріантною, тобто не мати повного ланцюга. Це впливає з того, що якщо X – повний p -ланцюг і $X^T \cdot M = K_0$, то зважена сума маркерів за

всіма позиціями $\sum_{i=1}^n x_i \mu_i = K_0$ є обмеженою. А оскільки x_i – додатні і вся

сума обмежена, то і маркування μ_i всіх позицій сітки обмежені. Необхідно зауважити, що якщо повний p -ланцюг є одиничним вектором, то сітка є безпечною.

Розглянемо наступну характеристику сітки – живучість, визначення якої базується на обчисленні t -інваріантів.

Цілочислений вектор $Y = [y_j | j = \overline{1, m}]$ називається t -інваріантом, якщо він є розв'язком лінійної однорідної системи:

$$A \bullet Y = 0. \quad (3.12)$$

Якщо значення t -інваріанта підставити в рівняння (3.5) замість вектора підрахунку спрацьовувань переходів S , то виявиться, що

$$M = M_0 + A \bullet Y = M_0.$$

Звідси випливає, що якщо $Y \neq 0$, то сітка стійка, тобто після деяких спрацьовувань переходів вона повертається в початковий стан M_0 . Стійкість сітки пов'язана з її циклічною повторюваністю, починаючи зі стану M_0 . Необхідно зазначити, що серед розв'язків системи (3.12) можуть бути і такі вектори Y , компоненти яких від'ємні.

Повний t -ланцюг – це t -інваріант, всі компоненти якого додатні. Повний t -ланцюг включає усі переходи сітки. Якщо сітка має повний t -ланцюг, то вона *стійка*, що є тільки необхідною умовою живучості при будь-якому початковому маркуванні, оскільки встановлено, що в послідовності маркувань від μ_0 до $\mu = m$ спрацьовують всі переходи, але не визначено чи існують тупикові маркування. Пошук повного t -ланцюга здійснюється подібно до пошуку повного p -ланцюга.

Тобто, якщо ранг матриці A дорівнює числу невідомих ($r = m$), то система (3.12) має тільки нульовий розв'язок. Якщо $r < m$, то система (3.12), крім нульового, має нескінченну множину інших розв'язків, причому фундаментальна (базисна) система складається з $(m - r)$ векторів t -інваріант.

Таким чином, якщо сітка жива і обмежена, то вона стійка і інваріантна. Тому для остаточного вирішення задачі аналізу на живучість потрібно перевірити сітку на досяжність тупикових станів.

Тупик – це досяжний з початкового маркування стан, у якому не збуджений жоден з переходів сітки.

Запишемо умову збудження переходу t_j у наступному вигляді:

$$\sum_{i=1}^n F_{ij} \mu_i \geq \sum_{i=1}^n F_{ij}, j = \overline{1, m}$$

або

$$F \bullet M \geq F \bullet E,$$

де E – одиничний вектор.

Оскільки множина досягаємих маркувань $R(N)$ повинна задовольняти умові (3.11), то відсутність збуджуваних переходів для $\mu \in R(N)$ необхідно визначити з розв'язку системи

$$\begin{cases} BgM = BgM_0 \\ FgM = FgE - E \end{cases} \quad (3.13)$$

Якщо ця система має розв'язок M , то деяке її маркування є

тупиковим, в іншому випадку сітка немає тупиків і є живою.

3.4.3. Редукційний аналіз структури і динаміки роботи сітки Петрі

Важкість застосування матричного дослідження полягає в тому, що розв'язання рівнянь є необхідним для визначення живучості та потребує знаходження рангу матриці, що є складною задачею при великій розмірності матриці сітки Петрі. Крім того, розв'язкам можуть відповідати недопустимі послідовності запуску переходів або взагалі декілька послідовностей.

Тому набули поширення редукційні методи дослідження, що дозволяють застосувати до сітки Петрі структурно-функціональні перетворення, що базуються на декомпозиції з метою зменшення розмірності сітки або множини досяжності та спрощення процедури аналізу властивостей динаміки роботи.

Редукція – це процедура гомоморфічного перетворення первинної сітки Петрі в її редукований образ, в якому зберігаються досліджувані властивості та принципи функціонування оригінальної сітки. Як наслідок, аналіз спрощеної сітки дасть відповідь про властивості первинної.

Всі методи проведення редукції в сітках Петрі базуються на різних способах *виділення підсіток* та їх подальшого *заміщення* іншими структурними поданнями.

Виділення підсіток, які надалі будуть називатися сітковими блоками, базується на таких способах поділу первинної сітки на частини:

- 1) за динамічними (функціональними) ознаками роботи;
- 2) за структурними ознаками побудови.

Редукція за *динамічними ознаками* полягає у виділенні таких частин-блоків у первинній сітці, в якій функціонування замінників цих блоків не змінює роботу їх суміжних елементів і первинної сітки вцілому (рис. 3.18). Тоді аналіз властивостей буде полягати у визначенні їх в сіткових блоках та в редукованому образі первинної сітки.

Якщо розмір первинної сітки є надто великим, то застосування цього способу виявляється проблематичним у зв'язку з необхідністю визначення властивостей роботи за допомогою одного із наведених раніше методів аналізу у створених сіткових блоках та умов взаємодії блоків з іншими елементами первинної сітки.

Таким чином, даний спосіб редукції не передбачає формалізованих процедур виділення підсіток і створення сіткових блоків (рис. 3.19).

Редукція за *структурною побудовою* полягає у виділенні у первинній сітці таких частин, які мають заздалегідь визначену графову структуру з вхідними і вихідними елементами (позиціями або переходами) однакового типу, в якій гарантована наявність

досліджуваних властивостей і визначені умови коректної взаємодії із зовнішніми елементами первинної сітки. Зокрема, такими графовими структурами побудови сіткових блоків можуть бути визначені (рис. 3.20):

- 1) автоматні блоки – структури з послідовним спрямуванням позицій і переходів від входу до виходу;
- 2) умовні блоки – структури із альтернативним спрямуванням позицій і переходів від входу до виходу;
- 3) паралельні блоки – структури, внутрішні елементи яких мають спільні входи і спільні виходи;
- 4) циклічні блоки – автоматні структури, в яких додатково визначені зворотні послідовності, що зв'язують вихід із входом.

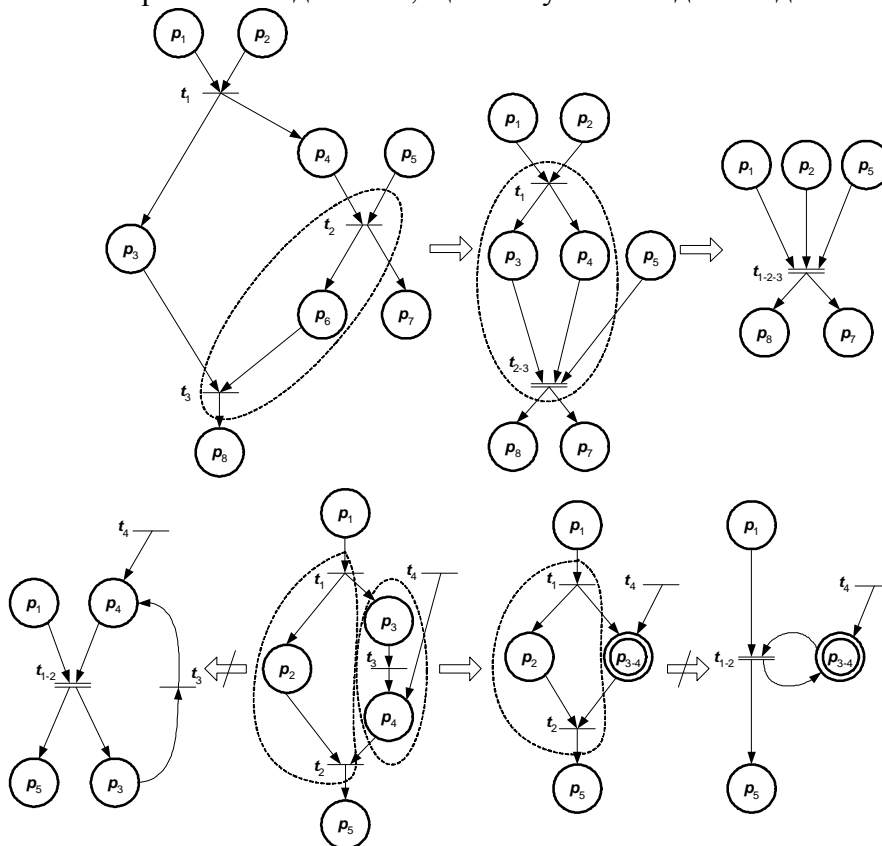


Рис. 3.18. Механізм редукції за динамічними ознаками

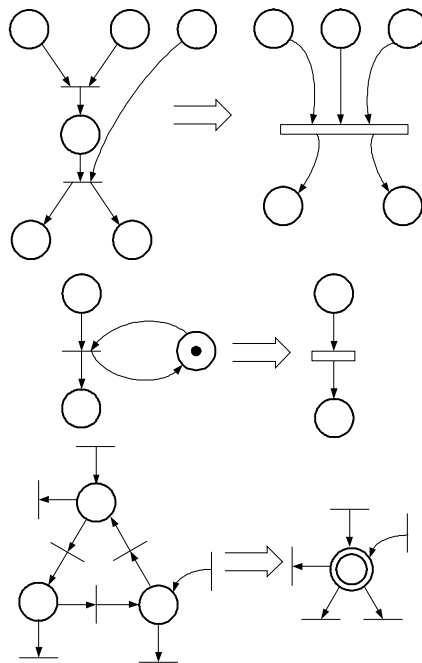


Рис. 3.19. Приклади динамічної редуції

Характерною ознакою коректної побудови блоків для дотримання властивостей роботи є взаємодія блока із зовнішніми елементами первинної сітки тільки через входи та виходи, тобто відсутні зв'язки внутрішніх елементів із зовнішніми елементами.

Заміщення виділених блоків на спрощене їх подання виконується за умови, якщо це не призведе до зміни функціонування суміжних елементів первинної сітки, тобто не зміниться порядок спрацьовування переходів і маркування позицій. Найчастіше виділені сіткові блоки подаються макропозиціями (рис. 3.21), макропереходами (рис. 3.22) у випадку, якщо входи і виходи блоків є одного типу, або об'єднувальними дугами в іншому випадку (рис. 3.23).

Проте, вхідні або вихідні елементи блока завжди є одного типу: позиції чи переходи.

Макропозиції використовуються для заміщення структур блоків з граничними елементами – позиціями, що мають зв'язок із зовнішніми елементами – переходами.

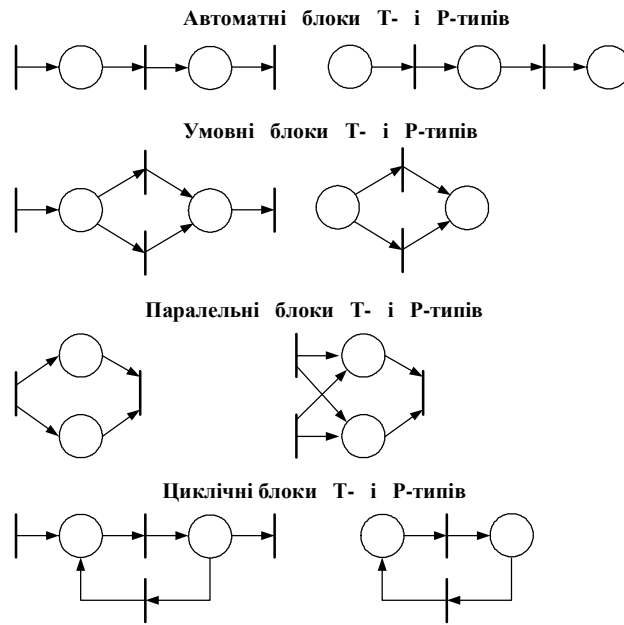


Рис. 3.20. Графові структури побудови сіткових блоків

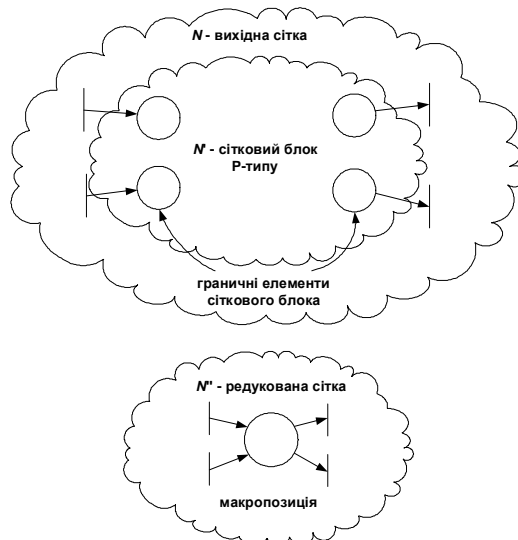


Рис. 3.21. Заміщення сіткового блока макропозицією

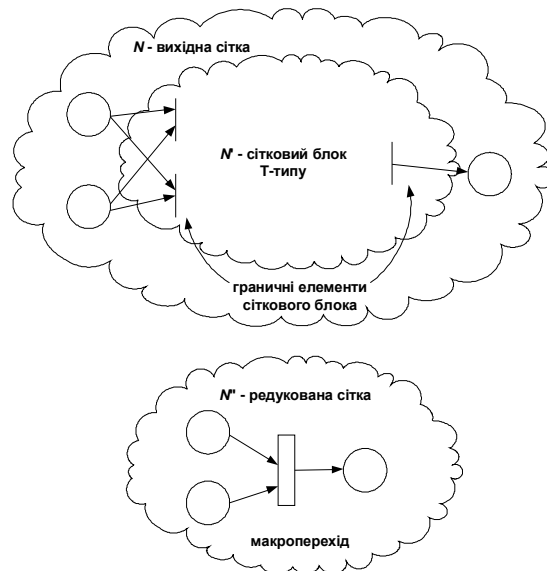


Рис. 3.22. Заміщення сіткового блока макропереходом

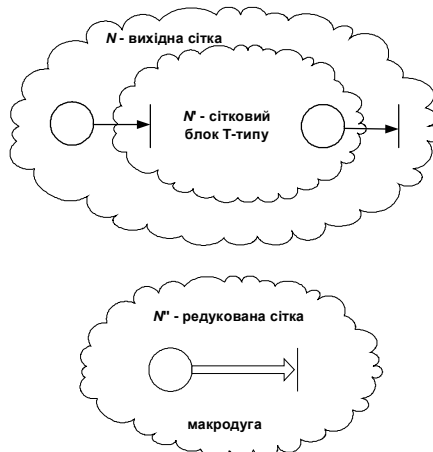


Рис. 3.23. Заміщення сіткового блока об'єднувальними дугами

Макропереходи використовуються для заміщення структур блоків з граничними елементами – переходами, що мають зв'язок із зовнішніми елементами – позиціями.

Об'єднання дуг використовується для заміщення структур блоків з граничними елементами різних типів і подається як зв'язка “позиція – перехід” або “перехід – позиція”, де позиція та перехід відображають відповідні граничні елементи блока.

Таким чином, редукційні способи аналізу можна вважати комбінованими методами дослідження властивостей сіток, що дозволяють, зменшивши розмір сітки (див. рис. 3.24), ефективно застосовувати методи аналізу дерева досяжності чи матричного подання.

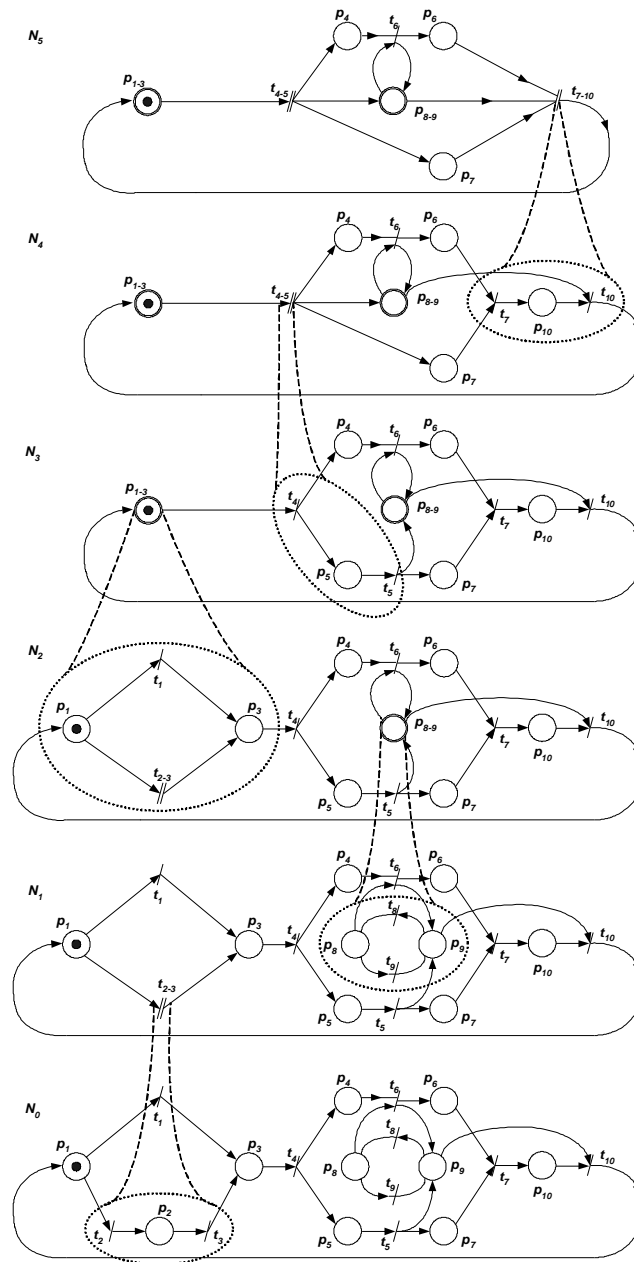


Рис. 3.24. Застосування методів редукції при дослідженні сітки Петрі

3.4.4. Методи аналізу модифікацій сіток Петрі

Для кольорових сіток існують аналогічні методи аналізу властивостей роботи, що розглядалися у попередніх пунктах.

Наведемо тільки відмінності у застосуванні, що існують для методу аналізу матричного подання при знаходженні інваріант кольорової сітки [11].

Динаміку руху маркерів у кольоровій сітці записують

рівнянням:

$$M_k = M_{k+1} + D \bullet U_k,$$

де $D = \left[D_{ij} \mid D_{ij} = (d_{ijl} \mid l = \overline{1, q}), i = \overline{1, n}, j = \overline{1, m} \right]$ – тривимірна матриця,

елементи якої D_{ij} подаються у вигляді векторів-рядків розмірністю q . Якщо позиція p_i має вихідні дуги кольору c_l до переходу t_j , то $d_{ijl} = -1$, якщо позиція p_i має входні дуги кольору c_l з переходу t_j , то $d_{ijl} = 1$, якщо немає дуг цього кольору, то $d_{ijl} = 0$, тобто матриця D формується на базі функцій F і H із врахуванням функцій λ і ψ .

Властивості сітки встановлюють наступним чином. Для визначення обмеженості розв'язують систему:

$$D^T \bullet V = 0,$$

де $V = [V_i]$ – вектор-стовпець p -інваріанти, компоненти якого подаються як $V_i = \left[v_{il} \mid l = \overline{1, q} \right]$.

Фундаментальна система B має $(n - rq)$ розв'язків, якщо r – ранг матриці D , n – кількість позицій, q – кількість кольорів маркерів.

Для визначення живучості розв'язують систему:

$$D \bullet W = 0,$$

де $W = [W_j]$ – вектор-стовпець t -інваріанти, компоненти якого подаються як $W_j = \left[w_{jl} \mid l = \overline{1, q} \right]$.

Умова існування тупиків в кольоровій сітці знаходиться з аналізу системи:

$$B \bullet M = B \bullet M_0;$$

$$O \bullet M \leq O \bullet E' - E,$$

де $O = \left[O_{ij} \mid O_{ij} = (o_{ijl} \mid l = \overline{1, q}), i = \overline{1, n}, j = \overline{1, m} \right]$; $o_{ijl} = 1$, якщо існує

дуга кольору c_l з p_i в t_j ;

$E' = [E'_j]$ – вектор, компоненти якого подані наборами з q одиниць.

3.5. Методика сіткового моделювання процесів функціонування ГВС

Основу формалізованого подання функціонування виробничої системи при розв'язанні задач оперативного управління складає принцип відображення в моделі матеріально-транспортних зв'язків виробничих процесів при організації обслуговування роботи основного технологічного устаткування. Цей принцип передбачає, що процедура побудови моделі задовольняє вимогам модульності та структурної подібності.

Модульність визначає поділ модельованого об'єкта на елементи (модулі), для кожного з яких будується власна модель.

Модуль представляє собою достатньо автономний з технологічної

точки зору об'єкт, тобто з відносно невеликою кількістю зв'язків цього об'єкта з іншими.

Структурна подібність полягає у тому, що кожному значущому структурному елементу об'єкта ставиться у відповідність набір елементів моделі, інформаційний стан яких однозначно визначає функціональний стан об'єкта. Моделі модулів агрегуються для отримання узагальненої моделі системи.

Такий підхід до подання дозволить сформувати модель, інваріантну до технологій виробничих процесів, одиниць матеріальних потоків, станів виробничого устаткування, які будуть виступати як параметри моделі.

Зміст методики розкриємо на прикладі опису роботи роботизованої системи з двох верстатів, одного транспортного робота, вхідних та вихідних накопичувачів деталей і готової продукції. Кожен верстат має вхідний накопичувач, де знаходиться партія оброблюваних деталей, та вихідний накопичувач, у який буде складовано результати обробки. Транспортний робот виконує операції завантаження робочої позиції верстата деталлю з вхідного накопичувача та розвантаження готової продукції у вихідний накопичувач. На верстаті одночасно може оброблятися тільки одна деталь, тобто на робочій позиції має бути не більше однієї деталі.

Складання моделі функціонування системи здійснюється поетапно.

1. Визначення схем руху предметів виробництва по обробних ресурсах і виділення типових операцій і процесів.

1) визначення складу операцій процесів функціонування окремих елементів системи:

t_1, t_2 – операції завантаження робочої позиції верстата 1, 2 з вхідних накопичувачів;

t_3, t_4 – операції обробки деталей на верстатах;

t_5, t_6 – операції розвантаження готової продукції з верстатів у вихідні накопичувачі.

2) визначення умов виконання операцій:

p_1, p_2 – стан завантаження вхідних накопичувачів верстатів 1, 2 деталями для обробки;

p_3, p_4 – завершення завантаження робочої позиції верстатів і запит обробки;

p_5, p_6 – завершення обробки і очікування розвантаження робочої позиції верстатів;

p_7, p_8 – стан завантаження вихідних накопичувачів готової продукції;

p_9, p_{10} – стан готовності верстатів до обробки “Верстат вільний”;

p_{11} – стан готовності транспортного робота до операцій завантаження-розвантаження.

3) складання графа (діаграми) взаємозв'язків операцій процесів

функціонування кожного елемента системи:

верстат 1 – послідовно виконує операції $t_1 \rightarrow t_3 \rightarrow t_5$ і повертається у стан готовності;

верстат 2 – послідовно виконує операції $t_2 \rightarrow t_4 \rightarrow t_6$ і повертається у стан готовності;

робот – виконує одну з операцій t_1, t_2, t_5, t_6 і повертається у стан готовності.

2. Структуризація повного опису функціонування системи.

Виконується композиція графів процесів у єдину модель-сітку функціонування системи. Композиція проводиться через об'єднання спільних операцій та умов їх виконання. У сітці операції подаються переходами, умови – позиціями, а в цілому модель є сіткою Петрі з невизначеним початковим маркуванням і без кольорових маркерів. Тому ця модель буде відображати множину варіантів транспортного обслуговування верстатів роботом і може застосовуватися для визначення організації оперативного управління рухом деталей. Якщо алгоритм транспортного обслуговування встановлений, то необхідно його формалізувати у параметрах моделі.

3. Формування опису матеріальних потоків і станів виробничих ресурсів.

Проводиться формалізація кольоровими маркерами станів деталей і виробничих елементів системи: верстатів і робота. Визначені такі стани:

c_1, c_2, c_3, c_4 – готовність робота до виконання відповідних операцій t_1, t_2, t_5, t_6 ;

c_5 – готовність верстатів до виконання операцій завантаження t_1 або t_2 ;

c_6 – оброблювана деталь.

4. Параметризація моделі згідно з визначеними технологіями виготовлення і допустимими схемами руху матеріальних потоків.

Параметризація моделі означає формування в кольоровій сітці для кожного переходу функцій допустимих розподілів λ, ψ . Виходячи з того, що встановлений алгоритм транспортного обслуговування верстатів передбачає циклічне виконання послідовності операцій t_1, t_2, t_5, t_6 , значення функцій “зафарбовування дуг” мають бути такими:

$$\lambda_1(p_1, c_6) = \lambda_1(p_{11}, c_1) = \lambda_1(p_9, c_5) = 1; \quad \psi_1(p_3, c_6) = \psi_1(p_{11}, c_2) = 1;$$

$$\lambda_2(p_2, c_6) = \lambda_2(p_{11}, c_1) = \lambda_2(p_{10}, c_5) = 1; \quad \psi_2(p_4, c_6) = \psi_2(p_{11}, c_3) = 1;$$

$$\lambda_3(p_3, c_6) = \lambda_4(p_4, c_6) = 1; \quad \psi_3(p_5, c_6) = \psi_4(p_6, c_6) = 1;$$

$$\lambda_5(p_5, c_6) = \lambda_5(p_{11}, c_3) = 1; \quad \psi_5(p_{11}, c_4) = \psi_5(p_9, c_5) = 1;$$

$$\lambda_6(p_6, c_6) = \lambda_6(p_{11}, c_4) = 1; \quad \psi_6(p_8, c_6) = \psi_6(p_{11}, c_1) = \psi_6(p_{10}, c_5) = 1.$$

Всі інші значення функцій дорівнюють 0.

5. Визначення початкового стану системи.

Початковий стан системи подається як початкове маркування позицій кольоровими маркерами і визначає завдання на обробку

деталей та поточний функціональний стан устаткування.

Нехай вхідні накопичувачі мають по одній деталі, верстати готові до завантаження і робот очікує виконання першої операції. Тому початкове маркування сітки має вигляд:

$$\mu_0(p_1, c_6) = 1; \mu_0(p_9, c_5) = \mu_0(p_{10}, c_5) = 1; \mu_0(p_{11}, c_1) = 1.$$

В цілому модель роботизованої системи показана на рис. 3.25. Побудована сітка є обмеженою та умовно живою, тобто структура сітки без позицій p_1 , p_2 , p_7 , p_8 є живою, оскільки виробниче устаткування системи після обробки деталей повертається у початковий стан.

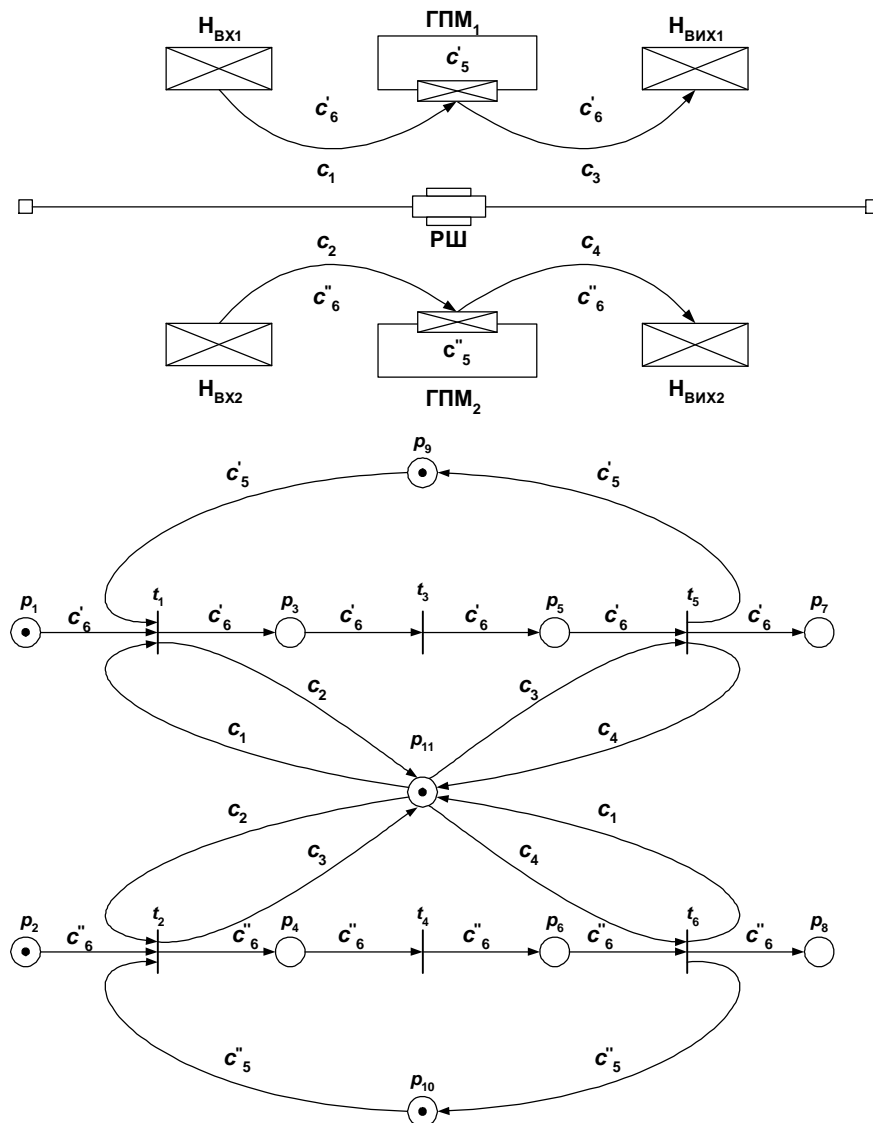


Рис. 3.25. Агрегатна модель двоверстатної роботизованої системи

Контрольні запитання

1. Розкрийте складові специфікації предметної області інтегрованого виробництва для визначення матеріальних та інформаційних процесів ГВС.
2. Сформулюйте можливості та переваги математичного апарату сіткових моделей дискретних виробничих процесів ГВС.
3. Наведіть відмінності та умови використання підкласів сіткових моделей дискретних виробничих процесів.
4. Розкрийте мету верифікації сіткових моделей та обґрунтуйте ефективність аналізу їх властивостей за деревом досяжності.
5. Розкрийте можливості та ефективність матричного засоба аналізу властивостей сіткових моделей.
6. Визначте базові принципи та особливості редуційного засоба аналізу властивостей сіткових моделей.
7. Наведіть та дайте пояснення до етапів методики сіткового моделювання процесів функціонування ГВС.

*«Щира думка веде нас до
правильних дій нітрохи не гірше,
ніж розум» (Платон Аристокл)*

Розділ 4

ПРОЕКТУВАННЯ ПРОЦЕСІВ УПРАВЛІННЯ ГВС

4.1. Технологія проектування алгоритмів управління виробничими процесами ГВС

Алгоритм управління (АУ) – це послідовність цілеспрямованих дій, пов'язаних із забезпеченням виконання операцій виробничого процесу (ВП) у ГВС. АУ може бути поданий як набір операторів процесу управління, що відповідають операціям ВП, який визначає порядок їх виконання. В результаті модель ВП може слугувати вихідними даними для проектування АУ ГВС.

Отже, розробка системи управління ГВС передбачає послідовне здійснення наступних етапів проектування:

- формалізація матеріально-інформаційних процесів ГВС як об'єкту управління (ОУ);
- аналіз властивостей функціонування ОУ за його моделлю;
- синтез моделей алгоритмів управління;
- проектування структури системи управління.

Про перший та другий етап проектування частково йшла мова у розділі 3, коли процес функціонування виробничої системи представлявся у формі сіткової моделі, в якій відтворюється ієрархія підпроцесів функціонування складових елементів ГВС.

Розглянемо сіткову інтерпретацію ієрархії підпроцесів функціонування складових елементів ГВС. Багатономенклатурне виробництво відноситься до класу складних систем з множиною різних динамічних об'єктів – матеріальних потоків, устаткування. Найчастіше ці об'єкти мають додаткові атрибути, які дозволяють розрізняти їх і використовувати ці розбіжності для управління функціонуванням системи.

В залежності від необхідного за специфікацією предметної області ГВС рівня детальності подання, ієрархія в складних виробничих процесах задається структурою, яка складається з множини більш простих підпроцесів, функції яких розкриваються операціями над динамічними об'єктами ГВС. Інакше кажучи, ВП складається з n ієрархічних рівнів, де деякі чи всі операції підпроцесу i -го рівня ($2 \leq i \leq n$) самі є процесами, які складаються з операцій, віднесених до $(i - 1)$ -го рівня. Старший рівень відповідає тільки одному підпроцесу.

Як модель ВП, яка ефективно відображає встановлену ієрархію підпроцесів і адекватно представляє операції перетворення матеріальних потоків в ГВС, застосовується ІКСП з наступною

інтерпретацією елементів: позиціям поставлені у відповідність операції моделюємого процесу і надані їм ресурси технологічної підсистеми (устаткування); переходам – події зміни операцій; маркерам – одиниці матеріальних потоків і стани ресурсів. З огляду на обрану форму інтерпретації, позиції-дублери набувають наступного сенсу: кожній такій позиції відповідає макрооперація, яка сама є підпроцесом більш низького рівня складності представлення, опис якого задається заміщаючим сітковим блоком.

Будемо вважати, що ІКСП N^* адекватна виробничому процесу, якщо між її елементами і атрибутами специфікації виробничого процесу SP встановлена взаємооднозначна відповідність.

Визначення 4.1. Сітковою моделлю виробничого процесу будемо називати таку ІКСП $N_{\Pi} = (\{N_i\}, Q, Z)$, для якої виконуються наступні умови:

$$N_i = (P_i, T_i, \Omega_i, F_i, H_i, \lambda_i, \psi_i, \mu_{i0}),$$

$$P_i = P_{ci} \cup P_{di} \cup P_{ri}, P_{ci} \cap P_{di} = \emptyset, (P_{ci} \cup P_{di}) \cap P_{ri} = \emptyset, |(P_{ci} \cup P_{di})| \geq |P_{ri}|;$$

$$\Omega_i = \Omega_{ci} \cup \Omega_{ri}; \Omega_{ci} \cap \Omega_{ri} = \emptyset;$$

де $Q: \{N_i\} \rightarrow \{N_i\}$ – ієрархія складових виробничого процесу (підпроцесів);

$$P_{ci} = \{p_c \mid p_c = f_1(o_i), o_i \in O\} \text{ – множина операцій } i\text{-го підпроцесу};$$

$P_{di} = \{p_d \mid p_d = f_2(N_j), Q(N_j) = N_i\}$ – множина складових підпроцесів i -го підпроцесу;

$P_{ri} = \{p_r \mid p_r = f_3(m), m \in M\}$ – множина ресурсів операцій i -го підпроцесу;

$T_i = \{t \mid t = f_4(g), g(o_i, o_j) \in G\}$ – множина подій у зміні станів i -го підпроцесу;

$\Omega_{ci} = \{\omega_c \mid \omega_c = f_5(d), d \in D\}$ – множина елементів матеріального потоку;

$\Omega_{ri} = \{\omega_r \mid \omega_r = f_6(s(m)), s(m) \in S_M\}$ – множина функціональних станів ресурсів;

$$\lambda_i = \{\lambda_{ii} \mid t \in T_i, \lambda_{ii} = f_7(TM)\}; \psi_i = \{\psi_{ii} \mid t \in T_i, \psi_{ii} = f_8(TM)\};$$

$$\mu_{i0}(P_i \setminus P_{ri}) = \left\{ \mu_{i0}(p, \omega_c) \mid p \in P_i \setminus P_{ri}, \omega_c \in \Omega_{ci}, \mu_{i0}(p, \omega_c) = f_9(s(d)), s(d) \in S_D \right\};$$

$$\mu_{i0}(P_{ri}) = \left\{ \mu_{i0}(p_r, \omega_r) \mid p_r \in P_{ri}, \omega_r \in \Omega_{ri}, \mu_{i0}(p_r, \omega_r) = e(m), m \in M \right\};$$

$Z = \{z(p, \omega) | p \in P_{ci}, \omega \in \Omega_{ci}, z(p, \omega) = \tau(o_i), o_i \in O\}$ – параметри

часу виконання операцій виробничого процесу;

$e(m)$ – ємність ресурсу m ;

$t(o_i)$ – час виконання операції.

Тоді виробничий процес функціонування ГВС можна представити як сукупність взаємозалежних процесів обслуговування ГВМ, кожний з яких включає різні підпроцеси підготовки ГВМ до обслуговування заявок-виробів (підготовка до налагодження, налагодження, доставка об'єктів складання), самообслуговування заявок, що надходять, на виконання виробничих операцій і доставку результатів на АС.

Таким чином, процедура формування сіткового подання виробничого процесу полягає у виконанні наступних дій:

- формування описів матеріальних потоків і станів виробничих засобів;
- визначення схем руху засобів виробництва за обробними ресурсами і виділення типових операцій і підпроцесів;
- декомпозиція досліджуваного процесу;
- структуризація повного опису з сіткових подань підпроцесів;
- параметризація моделі відповідно до існуючих технологій виготовлення і припустимих схем руху матеріальних потоків.

4.2. Синтез моделей алгоритмів управління виробничими процесами ГВС

Третій етап проектування систем управління ГВС полягає в одержанні обумовленої виконуваним виробничим процесом структури процесу управління, а потім у формуванні коректного алгоритму управління. Оскільки в отриманій моделі N_{Π} можна реалізувати майже будь-яку послідовність рухів матеріального потоку, визначену в рамках сформованої специфікації предметної області ГВС, то структура процесу управління вважається цілком заданою та адекватно відтвореною у ІКСП, якщо з кожною виробничою операцією співставити оператор управління, який ініціює її виконання. Тоді задача синтезу алгоритму управління полягає у заданні для множини переходів T моделі виробничого процесу N_{Π} відображень $h: P \times \Omega \rightarrow T$, що адекватно визначають порядок реалізації операцій у конфліктних ситуаціях $(P \times \Omega)$, та введенні додаткових елементів, що моделюють інформаційний обмін управляючої підсистеми і об'єкта управління.

Процес формування управління рухом матеріальних потоків по обробляючих ресурсах у ГВС полягає у встановленні в конфліктних ситуаціях невизначеності обслуговуваної заявки-предмета виробництва і засобу її обслуговування-ресурсу виробництва.

Згідно з таким поданням, конфліктні ситуації, що утворюються в ході виконання виробничого процесу (під час оперативного-

диспетчерського управління), задаються наступними типовими ситуаціями:

- обслуговування заявок спільним засобом (ресурсом);
- надання засобів (ресурсів) спільній заявці;
- виконання незалежних однієї від одної операцій.

У сітковій моделі виробничого процесу виділені ситуації мають своє формалізоване подання у вигляді наступних типових конфліктів на переходах T .

1. Конфлікт пов'язаних з доступом до спільного ресурсу p_r заявок ω_{ci}, ω_{cj} :

а) K_1 - конфлікт заявок від операції за виконання наступної спільної операції (рис. 4.1, а), де

$$K_1 = \bigcup_{t \in T} K_1(t); \quad K_1(t) = \{ \bullet c_t \mid \bullet c_t \in \bullet C_t : |\bullet C_t| > 1 \}; \quad T = \bigcup_{i=0}^p T_i;$$

б) K_2 - конфлікт заявок від різних операцій за виконання спільної операції p_c (рис. 4.1, б), де

$$K_2 = \bigcup_{p_c \in P_c} K_2(p_c); \quad K_2(p_c) = \{ t \mid t \in \bullet p_c, |\bullet p_c| > 1 \}; \quad P_c = \bigcup_{i=0}^p P_{ci};$$

в) K_3 - конфлікт заявок за спільний ресурс p_r (рис. 4.1, в), де

$$K_3 = \bigcup_{p_r \in P_r} K_3(p_r); \quad P_r = \bigcup_{i=0}^p P_{ri};$$

$$K_3(p_r) = \left\{ t \mid \left(\forall t_i, t_j \in p_r : (t_i \mathbf{I} t_j) = \emptyset \ \& \ (\bullet t_i \mathbf{I} \bullet t_j) \setminus \{ p_r \} = \emptyset \right) \right. \\ \left. \left(\exists \bullet c_{t_i} \in \bullet C_{t_i}, \bullet c_{t_j} \in \bullet C_{t_j} : \bullet c_{t_i} = \bullet c_{t_j} \right) \right\}.$$

2. K_4 - конфлікт різноманітних ресурсів p_{ri}, p_{rj} за заявку ω_{ci} від спільної операції p_c (рис. 4.2, а), де

$$K_4 = \bigcup_{p_c \in P_c} K_4(p_c);$$

$$K_4(p_c) = \{ t \mid \forall t_i, t_j \in p_c \exists \bullet c_{t_i} \in \bullet C_{t_i}, \bullet c_{t_j} \in \bullet C_{t_j} : \bullet c_{t_i} = \bullet c_{t_j} \}.$$

3. K_5 - конфлікт незалежно виконуваних заявок ω_{ci}, ω_{cj} (рис. 4.2, б), де

$$K_5 = \bigcup_{p_c \in P_c} K_5(p_c); \quad K_5(p_c) = \{ t \mid \forall t \in (T \setminus (K_2 \cup K_3 \cup K_4)) : t \in p_c \bullet \} \cup$$

$$\cup \{ K_2(p_c), K_3(p_r), K_4(p_c) \mid p_r \mathbf{I} \bullet p_c \neq \emptyset \ \& \ \bullet p_r \mathbf{I} p_c \neq \emptyset \}.$$

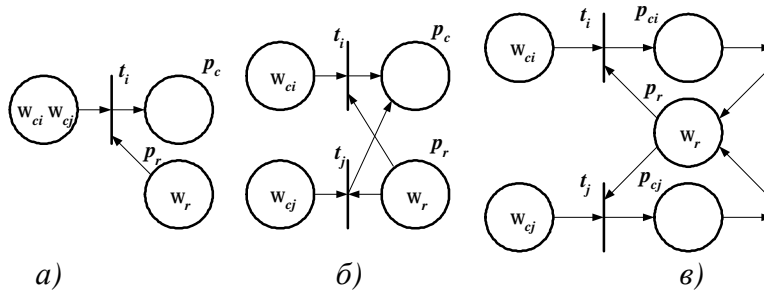


Рис. 4.1. Сіткове подання конфліктів заявок:

а, б – види конфліктів заявок за виконання спільної операції;
в – конфлікт заявок за загальний ресурс

Розгляд як конфлікту незалежно виконуваних заявок пов'язано з орієнтацією сіткової моделі на опис паралельно здійснюваних підпроцесів в умовах послідовного прийняття керуючих рішень.

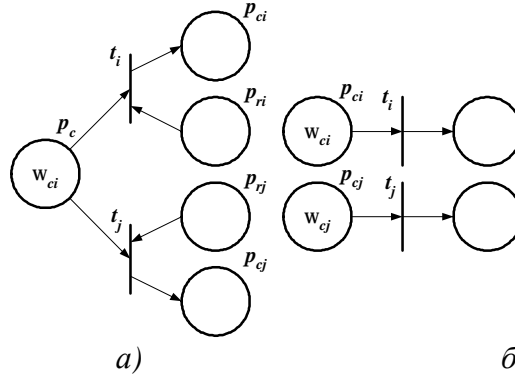


Рис. 4.2. Сіткове подання конфлікту:

а – устаткування за обслуговування заявки;
б – незалежно виконуваних операцій

Для вирішення конфліктів першого виду встановлюються пріоритети заявок, що надходять у кожному конкретному випадку, на обслуговування $PN_1: D \rightarrow \{1, 2, 3, \dots\}$ на базі правил переваги, які застосовувалися у розрахунку розкладу роботи технологічного устаткування при оперативному плануванні.

Тоді процедура вибору R_l , $l = \overline{1, 3}$ активізованої операції, яка входить у конфлікт K_1 **першого виду**, сформується на базі наступних співвідношень:

$$\begin{aligned} & \forall \cdot c_t^i, \cdot c_t^j \in K_1(t) \exists d_i, d_j \in D: [PN_1(d_i) > PN_1(d_j) \Rightarrow \\ & \Rightarrow \eta_t^1(\omega_{ci}) > \eta_t^1(\omega_{cj}) \Rightarrow R_1(\cdot c_t^i, \cdot c_t^j) = 1], \cdot c_t^i = [(p_c, \omega_{ci})]; \\ & \forall t_i, t_j \in K_l(p_{h_l}) \exists d_i, d_j \in D: [PN_1(d_i) > PN_1(d_j) \Rightarrow \\ & \Rightarrow \eta_{t_i}^l(\omega_{ci}) > \eta_{t_j}^l(\omega_{cj}) \Rightarrow R_l(t_i, t_j) = 1], l = \overline{2, 3}, p_{h_2} \in P_c, p_{h_3} \in P_r. \end{aligned}$$

Для вирішення конфлікту **другого виду** K_4 встановлюються

пріоритети на порядок надання устаткування $PN_2 : M \rightarrow \{1, 2, 3, \dots\}$ на базі стратегій транспортного обслуговування, визначених при оперативно-диспетчерському управлінні:

а) заявка направляється на найближчий за часом доставки вільний ГВМ;

б) заявка направляється на вільний ГВМ із мінімальним поточним завантаженням;

в) вибір вільного ГВМ здійснюється випадковим чином.

Процедура вибору R_4 активізованої операції, що входить у конфлікт K_4 другого виду, сформується на базі наступного співвідношення:

$$\begin{aligned} & \forall t_i, t_j \in K_4(p_c) \\ & (\exists m_i, m_j \in M : t_i \in p_{ri}, t_j \in p_{rj}, p_{ri} = f_3(m_i), p_{rj} = f_3(m_j)) : \\ & [PN_2(m_i) > PN_2(m_j) \Rightarrow \eta_{p_c}^4(t_i) > \eta_{p_c}^4(t_j) \Rightarrow R_4(t_i, t_j) = 1]. \end{aligned}$$

Конфлікт K_5 **третього виду** вирішується призначенням пріоритетів $PN_3 : O \rightarrow \{1, 2, 3, \dots\}$, виходячи з прийнятого першочергового порядку виконання незалежно активізованих операцій і підпроцесів у ВП. Процедура вибору R_5 активізованої операції, що входить у конфлікт K_5 третього виду, сформується на базі наступних співвідношень:

$$\begin{aligned} & \forall t_i \in K_g(p_h) \subset K_5 \forall t_j \in K_k(p_l) \subset K_5 \\ & (\exists o_i, o_j \in O : t_i \in p_{ci}, t_j \in p_{cj}, p_{ci} = f_1(o_i), p_{cj} = f_1(o_j), h \neq l) : \\ & [PN_3(o_i) > PN_3(o_j) \Rightarrow \eta^5(t_i) > \eta^5(t_j) \Rightarrow R_5(t_i, t_j) = 1]; g, k \in \{2, 3, 4\}. \end{aligned}$$

Враховуючи, що будь-якій перехід сіткової моделі може одночасно належати декільком різним типовим конфліктам, відображення η на підмножині переходів представляє собою сукупність предикатів PR умов спрацьовування, співвідношення яких формується на базі ієрархічної системи вирішення типових конфліктів (“вибір заявки обслуговування”–“надання ресурсу заявці”–“вибір порядку виконання незалежних заявок”). Іншими словами, на кожному рівні ієрархії типових конфліктів формується значення локального предикату умови спрацьовування переходу із врахуванням результатів вирішення на попередньому рівні. Тоді переходу $\overset{\circ}{t}_j$ сіткової моделі можна поставити у відповідність узагальнений предикат умови спрацьовування наступного виду:

$$B(\overset{\circ}{t}_j) = \bigwedge_{t_j^i \in \overset{\circ}{t}_j, c_{t_j^i} \in K_1(t_j^i)} q\left(\overset{\circ}{c}_{t_j^i} \bigwedge_{l=1,5} \left(A_l(\overset{\circ}{c}_{t_j^i}) \vee \overline{q_l}(t_j^i)\right)\right),$$

де $\overset{\circ}{t}_j = \{t_j^i \mid t_j^i \in N_i\}$ – перехід-зв'язка;

$\overline{q_l}(t_j^i)$ – логічна умова того, що перехід t_j^i входить до конфлікту l -го виду;

$$A_l(\cdot c_{t_j^i}) = \bigwedge_{t_h^i \in K_l \setminus \{t_j^i\}} \left(\overline{q\left(\cdot c_{t_h^i}\right)} \bigwedge_{k=1, l-1} \left(A_k(\cdot c_{t_h^i}) \vee \overline{q_k(t_h^i)} \right) \vee R_l(t_j^i, t_h^i) \right) \quad -$$

локальний предикат умови спрацьовування переходу t_j^i за розподілом $\cdot c_{t_j^i}$ в ієрархічній системі вирішення конфлікту l -го виду, $l = \overline{2, 5}$;

$q(\cdot c_{t_j^i})$ – логічна умова того, що перехід t_j^i ініційований за розподілом $\cdot c_{t_j^i}$;

$$A_1(\cdot c_{t_j^i}) = \bigwedge_{\cdot c_{t_j^i}' \in K_1(t_j^i) \setminus \{\cdot c_{t_j^i}\}} \left(\overline{q}(\cdot c_{t_j^i}') \vee R_1(\cdot c_{t_j^i}, \cdot c_{t_j^i}') \right).$$

Визначення 4.2. Сітковою моделлю управляючого процесу називається предикатна ІКСП наступного виду:

$$N_y = (N_{\Pi}, PR),$$

де $PR = \left\{ pr_j \mid pr_j = B(t_j^{\circ}), j = \overline{1, |T|} \right\}$ – множина предикатів умов спрацьовування переходів.

При формалізації і побудові сіткових моделей АУ на основі управляючого процесу (УП) необхідно забезпечити асинхронне узгодження взаємодій між операторами управління так, щоб моменти ініціації виробничих операцій визначалися тільки готовністю необхідних даних про стан ОУ. У цьому випадку управляюча підсистема ГВС формує управляючі впливи y_i у вигляді команд активізації виконання відповідних виробничих операцій в ОУ, після чого ОУ працює автономно, а управляюча підсистема очікує сигналу x_i від ОУ про закінчення виконання цих операцій. Тоді кожен оператор-дія, який виконує процедуру по управлінню деякою виробничою операцією визначеної тривалості, у моделі ВП заміняємо сітковим фрагментом, зображеним на рис. 4.3, де w_i , u_i – зовнішні позиції, що відображають відповідно видачу управляючого впливу і прийом сигналу про стан устаткування. Таким чином, для опису АУ на базі моделі УП N_y введемо спеціальні множини позицій, що представляють деякий вхідний і вихідний алфавіти, які характеризують асинхронні взаємодії з ОУ.

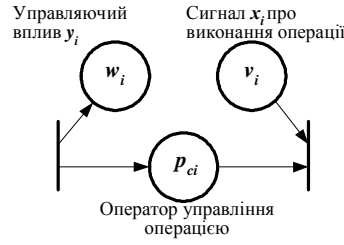


Рис. 4.3. Сіткове подання процедури управління виробничою операцією

Визначення 4.3. Сітковою моделлю алгоритму управління називається навантажена предикатна ІКСП, яка задається набором наступного виду:

$$N_A = (N_y, X, Y, V, W, f, \phi),$$

де $X = \{x_l \mid x_l = (b_1, b_2, b_3, b_4), l = \overline{1, L}\}$ – множина сигналів про стан устаткування;

b_i – логічні змінні, що відповідно вказують ознаку стану ОУ: несправність, зайнятість, налагодженість і робота;

$Y = \{y_k \mid y_k = (d, par_1, par_2), d \in D, k = \overline{1, K}\}$ – множина управляючих впливів; par_1, par_2 – управляючі параметри: адреса відправлення, номер програми обробки;

$$V = \left\{ v_j \mid \forall p_{cj} \in P_c \left(\exists \omega_l \in \bigcup_{i=0}^p \Omega_i : \mu(p_{cj}, \omega_l) \neq 0 \right) \exists v_j : v_j = p_{cj} \right\};$$

$$W = \left\{ w_j \mid \forall p_{cj} \in P_c \left(\exists \omega_k \in \bigcup_{i=0}^p \Omega_i : \mu(p_{cj}, \omega_k) \neq 0 \right) \exists w_j : w_j = p_{cj} \right\} \quad -$$

множини зовнішніх позицій для подання асинхронної взаємодії з ОУ;

$$f : V \times \bigcup_{i=0}^p \Omega_i \rightarrow X; \quad \phi : W \times \bigcup_{i=0}^p \Omega_i \rightarrow Y \quad - \text{ функції навантаження}$$

на зовнішні позиції.

Згідно з принципом подійності управління, момент часу управління відповідає моменту надходження сигналу про готовність устаткування x_i (розміщення в позиції u_i мітки-сигналу). Тоді задача управління за сітковою моделлю полягає у виборі в цей момент однієї з множини готових до виконання операцій (вибрати за істинним значенням предиката pr_j відповідний перехід спрацьовування) і вказівки шляху її реалізації у заданні управляючого впливу u_i (помістити в позицію w_i мітку управління). Множина предикатів сіткової моделі N_A формалізує стратегію управління ГВС.

Таким чином, *метод синтезу* сіткових моделей коректних алгоритмів управління на основі отриманого і перевіреного опису ВП можна представити у вигляді послідовності наступних етапів:

- сформувати структуру УП і визначити склад необхідних операторів управління операціями ВП;
- усунути всі конфлікти через наявність спільних ресурсів та альтернативний характер ВП за допомогою предикатів умов спрацьовування переходів;
- вибрати всі оператори-дії, для яких сформувати набори управляючих впливів і сигналів про стан відповідного устаткування;
- сформувати для кожної оператора-дії сіткове подання асинхронної взаємодії з виробничим устаткуванням.

У кожному конкретному випадку правила призначення пріоритетів при формуванні множини предикатів обираються за результатами імітаційних експериментів над сітковою моделлю ВП і порівнянням їх за обраним критерієм. Використання ж правил переваги допускає ефективну організацію управління по ситуаціях при прямій диспетчеризації функціонування ГВС.

4.3. Організація робіт по створенню систем оперативного управління ГВС

Задачі та особливості побудови програмних засобів управління. При створенні ГВС однією з головних є задача оперативного управління функціонуванням виробничого устаткування і синхронізації матеріальних потоків. При цьому розробка програмного забезпечення, що реалізує алгоритми управління, синтезовані на основі розглянутих модифікацій сіток Петрі, буде найбільш трудомістким етапом проектування ГВС.

Розроблювані сьогодні ГВС є унікальними системами, пристосованими для розв'язку конкретних задач, обумовлених специфікою автоматизованого об'єкта. Витрати на проектування і подальший розвиток можна знизити типізацією рішень в області створення ГВС, у тому числі і при розробці програмних середовищ оперативного управління матеріальними потоками ГВС.

Одним зі шляхів типізації рішення в цій області є використання принципу модульності, відповідно до якого ГВС компонуються з окремих типових модулів і підсистем.

Система оперативного управління ГВС представляє собою баторівневу ієрархічну систему зі зворотними зв'язками, організовану у вигляді розподіленої мережі комп'ютерних та мікропроцесорних пристроїв управління. Схема організації СОУ ГВС була наведена на рис. 1.8. Нижній рівень вирішує задачі локального управління послідовністю технологічних операцій на виробничому устаткуванні. Управляючі сигнали, сформовані цим рівнем управління, визначають моменти включення та виключення компонентів устаткування для виконання необхідних виробничих операцій, забезпечують програмну роботу устаткування. Верхній рівень управління забезпечує взаємодію всіх одиниць устаткування при проходженні через ГВС запланованих

на плановий період виробів-замовлень. На цьому рівні збирається і відображається поточна інформація, що визначає стан ділянки, координується робота всіх засобів управління нижнього рівня, формується інформація і повідомлення (запити до операторів ГВС), здійснюється взаємодія з вищестоячим рівнем управління.

При експлуатації ГВС, використовуючи програмні засоби управління, користувач може вирішувати задачі автоматизації оперативного управління, диспетчеризації матеріальних потоків, оперативного регулювання і коректування планового завдання, автоматизації обліку стану виробництва, автоматизації передачі інформації в ієрархічних системах управління.

Головною відмінною рисою і принципом побудови розроблених програмних засобів є управління процесами функціонування ГВС на основі інтерпретації сіткових управляючих моделей. На етапі проектування алгоритмів чи управління їх гнучкою перебудовою інтерпретація управляючих моделей виконується з метою аналізу властивостей алгоритмів і розв'язку різних оптимізаційних задач. На етапі реалізації алгоритмів управління спроектовані сіткові моделі використовуються для управління процесами функціонування ГВС. При цьому управляюча модель інтерпретується на основі тих же принципів, що і на етапі проектування.

На рис. 4.4 зображена схема організації управління ГВС на основі використання управляючих моделей, що відображає послідовність взаємозалежних фаз управління.

На вхід СОУ надходить інформація про конкретну конфігурацію об'єкта управління ОУ, про умови його функціонування, специфікації управляємих процесів, а також планове завдання та вимоги оптимізації. У блоці $B1$ будується ієрархічна структура управляючої моделі ГВС і формується область значень оптимізованих параметрів. Виходом блоку $B1$ служить сіткова модель N_{Π} процесу функціонування ГВС і набір правил призначення пріоритетів (PN_i) . Блок $B2$ реалізує фазу оперативного планування й оптимізує параметри сіткової моделі процесу функціонування з метою формування N_{γ} – управляючої моделі ГВС. Вихідними даними є сіткова модель N_{Π} і кодовані параметрами одержуваної управляючої моделі правила призначення пріоритетів, що формалізують прийняті в СОУ стратегії управління.

Фаза диспетчеризації реалізується блоками $B3$ і $B4$. Блок $B3$ диспетчеризації запитів і команд відповідно до вибраної у фазі планування стратегії управління за поточним значенням вектора-сигналу V зворотного зв'язку з ОУ формує нові значення маркування M_{k+1} управляючої моделі і вектора вхідного алфавіту V взаємодії з ОУ. Блок $B4$ здійснює інтерпретацію поточного стану управляючої моделі і перетворює відповідні елементи N_{γ} на параметри управління – управляючі сигнали Y_k .

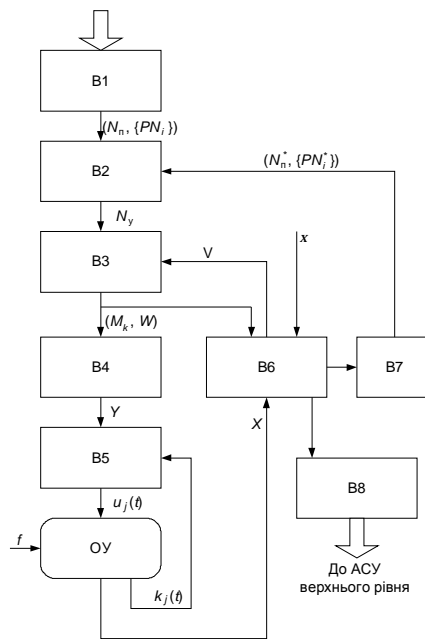


Рис. 4.4. Схема організації управління ГВС на основі використання управляючих моделей: B1 – генерація структури та настроювання параметрів управляючої моделі; B2 – оптимізація параметрів управляючої моделі; B3 – трансляція управляючої моделі (диспетчеризація запитів та команд); B4 – інтерпретація стану управляючої моделі M_k ; B5 – локальне управління обладнанням; B6 – діагностика стану та контроль показників функціонування ОУ; B7 – обробка аварійних ситуацій та відхилень (коректування структури та параметрів управляючої моделі);

B8 – статистичний облік; ОУ – об'єкт управління (обладнання ГВС)

Кожна операція для СОУ характеризується тільки тривалістю, ресурсом, що використовується для її виконання, і предметом праці – партією деталей. У зв'язку з цим управляючими сигналами є команди запуску операцій із вказівкою часу початку і місця їх виконання для всіх технологічних процесів ГВС у плановому періоді. Ці команди обробляє пристрій локального управління відповідними ресурсами (блок B5). Пристрій локального управління, приймаючи з верхнього рівня команди запуску необхідної операції, реалізує управління устаткуванням, використовуючи свій набір сигналів про стан ОУ $\{K_i(t)\}$ і набір управляючих команд $\{u_i(t)\}$ механізмів устаткування. По закінченні виконання необхідної операції ОУ формує сигнали X_k про зміну стану своїх компонентів, що надходять до блоку діагностики станів і показників функціонування ГВС (блок B6).

Блок B6 реалізує функції фази контролю. В цьому блоці обробляється первинна інформація про стан ОУ і встановлюється відповідність дійсного ходу функціонування виробництва X_k запланованому M_k . Якщо відхилення вважаються допустимими, то

управління передається блоку *B3*, на вхід якого подається вектор вхідного алфавіту сигналів *V* зворотного зв'язку по управлінню. Якщо блок *B6* фіксує аварійні ситуації (вихід з ладу устаткування) чи недопустимі відхилення функціонування ОУ (порушення у виконанні планового завдання), то управління передається на блок *B7*.

Блок *B7* виконує функції регулювання і призначений для обробки аварійних ситуацій і усунення відхилень виробництва додаванням у систему ресурсів чи зміною планового завдання. У блоці коректуються структури і параметри управляючої моделі і формується нове планове завдання із врахуванням введених оператором ГВС додаткових ресурсів. У результаті на вхід блока *B2* подається нова відкоректована модель N_p^* процесу функціонування ОУ. При цьому до закінчення процесу формування нової сіткової управляючої моделі управління ГВС організовується за старою моделлю в режимі прямої диспетчеризації по одній із встановлених стратегій управління.

У СОУ реалізується також статистичний облік станів виробництва для використання і передачі інформації АСУ верхнього рівня (блок *B8*). На блок *B8* може вплинути і оператор ГВС через повідомлення x про зміни в інформаційному стані СОУ.

Таким чином, управління ГВС складається з фаз генерації, планування, диспетчеризації, контролю, обліку і регулювання, що виконуються послідовно, а сама СОУ реалізує принцип управління зі зворотним зв'язком по стану, що враховує вплив на ОУ всіх збурюючих впливів (f, x).

Організація управління на базі управляючих моделей дозволяє говорити про інваріантність розробленого принципу до конкретного змісту виконуваних технологічних операцій і конфігурації ОУ. Цей принцип може бути типізований не тільки для дискретних складальних, але і для механообробних виробництв. У цьому полягає принципова відмінність і перевага розробленого пакету СОУ.

Для автоматизації фази оперативного планування розклад функціонування устаткування одержують способом динамічної трансляції сіткових управляючих моделей. Однак управління ведеться не відповідно до заздалегідь встановленого у фазі планування розкладом послідовності дій, а за оптимальними стратегіями управління, що формують цю послідовність дій у ході обчислень. Розклад використовується у фазі регулювання, де корегується динаміка управляючої підсистеми до реального ходу керованих процесів ГВС. При цьому враховуються реальні умови, що накладаються на ОУ: стан устаткування, виробів, виробничих показників функціонування.

Така організація управління дозволяє виявляти збої в системотехнічних діях управляючої підсистеми на основі аналізу фактичних станів процесів і співставлення їх із допустимими, які визначаються інтерпретацією управляючої моделі.

Отже, усі фази управління ГВС можна об'єднати єдиною інформаційною базою та системою програмного забезпечення. Сіткові управляючі моделі забезпечують гнучкість управляючої підсистеми, що реалізується адаптацією цієї єдиної бази даних до конкретних технологічних процесів і умов виробництва. Перехід до випуску нового виробу при такій організації управління супроводжується підготовкою відповідної управляючої моделі та стандартної структури даних.

На основі таких принципів організації управління бази даних і програмні засоби моделювання поєднуються в єдину систему, що підтримує аналіз, синтез і оптимізацію алгоритмів управління та інтерпретуючу управляючі моделі. Представляючи єдиний комплекс, така система оперативного управління забезпечує підвищення якості та скорочення термінів проектування і переналагодження системотехнічних дій управляючої підсистеми ГВС.

Склад і функції програмного комплексу типової СОУ ГВС. Комплекс програм типової СОУ ГВС призначений для синхронізації функціонування виробничого устаткування гнучких виробничих систем, досягнення максимально можливого коефіцієнта завантаження устаткування і забезпечення ефективного розподілу матеріальних і інформаційних потоків на стадії експлуатації ГВС. Програмне забезпечення реалізоване із врахуванням наступних вимог до принципів організації комплексу програм: єдність бази даних, розширюваність, модульність, сумісність знизу вгору, інваріантність.

Функції, реалізовані системою оперативного управління, визначаються складом устаткування, задачами управління цим устаткуванням і організацією управління ГВС.

Кожна функція, реалізація якої покладена на систему оперативного управління, задає перетворення:

$$f_i: X_i \rightarrow Y_i,$$

де f_i – найменування функції;

X_i – множина вихідних даних – аргументів функції;

Y_i – множина результатів застосування функції.

Таким чином, функція задає правило одержання результату на основі вихідних даних, причому вся множина функцій системи $F_i = \{f_i | i = 1, n\}$ визначає єдину базу даних, що характеризується

$$\text{множиною } D = \bigcup_{i=1}^n (X_i \cup Y_i).$$

З кожною функцією f_i у СОУ ГВС зв'язується програмний модуль, що представляє цю функцію у формі деякої задачі управління, реалізованої на ЕОМ.

У СОУ ГВС реалізуються наступні функції:

– створення і відновлення в інтерактивному режимі інформаційного стану системи (при повному запуску і після аварійного виходу із системи);

- розрахунок змінно-добового завдання і чергового запуску партій деталей у виробництво;
- оперативне корегування моделі інформаційного стану системи і планових завдань (за вказівками диспетчера ГВС);
- управління взаємодією устаткування (диспетчеризація) відповідно до отриманої управляючої моделі;
- контроль відхилень у процесі функціонування ГВС;
- формування і видача управляючих впливів виробничому устаткуванню;
- облік і формування довідкових даних про стан об'єкта управління і виконання планового завдання;
- тестування елементів комплексу програм СОУ ГВС;
- завершення роботи системи.

Ефективність застосування тієї чи іншої системи, як правило, обернено пропорційна складності її концептуального представлення, тому при проектуванні і розробці програмного забезпечення СОУ ГВС однією з основних задач було створення простих програмних засобів, що не вимагають від диспетчерів ГВС великого обсягу знань з програмування та обчислювальної техніки. Виходячи з вимог гнучкості програмного забезпечення СОУ ГВС побудований за блоково-модульним принципом відповідно до виконуваних функціональних завдань. Це дозволяє модифікувати модулі і розширювати склад комплексу при введенні нових функцій без зміни інших програм. Комплекс програм СОУ ГВС має трирівневу структуру, схема якої зображена на рис. 4.6.

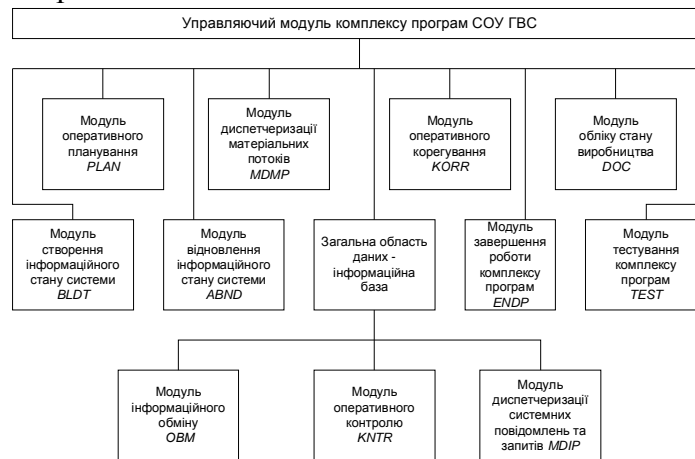


Рис. 4.6. Структура комплексу програм СОУ ГВС

На верхньому рівні комплексу знаходиться управляючий модуль, що здійснює діалог між диспетчером ГВС і програмним забезпеченням СОУ, а також зв'язок програмних модулів у системі. Управляючий модуль генерує режими роботи комплексу програм, управляє запуском і проходженням програмних модулів у системі.

Модулі середнього рівня реалізують прикладні цілі

функціонування СОУ ГВС і забезпечують:

- створення інформаційного стану системи, оперативного управління;
- оперативне планування;
- диспетчеризацію матеріальних потоків;
- оперативне корегування стану системи і ходу виробництва;
- облік стану виробництва;
- тестування комплексу програм за контрольними прикладами;
- відновлення інформаційного стану при аварійному виході із системи;
- завершення роботи комплексу програм.

Використання загальної області інформаційної бази забезпечує одночасний доступ до даних про стан ГВС декількох програмних модулів всіх рівнів комплексу і миттєву “реакцію” на зміни даних у цій базі.

Нижній рівень комплексу програм складають модулі, що перебувають у функціональному підпорядкуванні у модулів середнього рівня і запуск яких виконується одночасно із запуском визначених модулів середнього рівня. Модуль інформаційного обміну *OBM*, використовуючи загальну базу даних, передає інформацію між комплексом СОУ ГВС і управляючими пристроями виробничого устаткування. Модуль функціонує незалежно, однак його запуск здійснюється одночасно з запуском модуля диспетчеризації *MDMP* чи оперативного корегування *KORR*.

Модуль оперативного контролю *KNTR* функціонально підлеглий модулю диспетчеризації *MDMP*, оскільки переробляє поточну інформацію, що надходить з модуля інформаційного обміну *OBM*, про стан виконання запускених модулем диспетчеризації виробничих операцій і встановлює необхідність коректування ходу функціонування виробництва. Модуль запускається одночасно з модулем диспетчеризації матеріальних потоків *MDMP*, а в ході свого функціонування інформує модуль диспетчеризації про необхідні надалі дії по управлінню ГВС.

Форма роботи СОУ ГВС – діалогова з диспетчером ГВС. Вихідною інформацією для роботи системи є змінно-добові завдання у вигляді комплектації (завантаження) автоматизованого складу. Склад і функції програмних модулів СОУ ГВС наведені в табл. 4.1.

Таблиця 4.1

Характеристика модулів комплексу програм

Назва модуля	Функції модуля
<i>MAIN</i>	Ініціалізація комплексу програм і управління проходженням задач у системі
<i>BLDT</i>	Створення інформаційного стану системи управління ГВС

<i>PLAN</i>	Оперативне планування через побудову розкладу функціонування виробничого устаткування і управляючої моделі ГВС
<i>MDMP</i>	Диспетчеризація матеріальних потоків через обробку інформації, отриманої від виробничого устаткування, і формування управляючих впливів подальшого порядку роботи ГВС
<i>KNTR</i>	Оперативний контроль, діагностика й аналіз станів виробничого устаткування і ходу виробництва. Ухвалення рішення про порядок подальшої діяльності комплексу у фазі диспетчеризації. Викликається одночасно з модулем <i>MDMP</i> і інформує його про результати роботи
<i>KORR</i>	Оперативне корегування планового завдання та інформаційного стану системи. Викликається в міру необхідності за ініціативи модуля <i>KNTR</i>
<i>MDIP</i>	Диспетчеризація запитів і системних повідомлень для інформування диспетчера про виникаючі події та реалізацію диспетчерських повідомлень у системі. Викликається одночасно з модулем <i>MDMP</i> і управляє проходженням і реалізацією запитів і системних повідомлень
<i>DOC</i>	Облік стану виробництва і формування довідкової інформації здійснення діалогу з диспетчером і видача необхідної інформації на запит. Викликається в міру необхідності диспетчером
<i>ABND</i>	Відновлення інформаційного стану системи, перевірка файлів даних після чи аварії збою в системі
<i>OBM</i>	Міжрівневий обмін, зв'язок рівнів управління і передача інформації між комплексом програм і пристроями локального управління виробничого устаткування. Викликається одночасно з модулем <i>MDMP</i>
<i>TEST</i>	Тестування комплексу програм за контрольними прикладами
<i>ENDP</i>	Завершення роботи комплексу програм

4.4. Алгоритмічне забезпечення типової системи оперативного управління ГВС

Опис програмних модулів і структур даних комплексу програм СОУ ГВС. Розглянемо роботу модулів комплексу програм.

Модуль *BLDT* призначений для створення інформаційного стану СОУ ГВС. Він вводить вихідні дані в інформаційну базу.

Модуль реалізує два етапи своєї роботи:

- 1) введення і контроль вихідної інформації (у діалоговому режимі);
- 2) формування структур робочих масивів і запис їх у базу даних

(в автономному режимі).

На *першому етапі* роботи через інтерфейсну частину модуля вводяться поточні параметри ГВС, моделі стану устаткування, моделі стану складу і стану транспортних засобів і фіксується стан виробництва за минулу зміну. Моделі стану елементів ГВС відображають останню зміну стану в попередній період функціонування. В міру необхідності можна змінити стан модулів на початок наступного періоду роботи елементів ГВС.

На *другому етапі* роботи модуля здійснюється програмна переробка введеної інформації і формуються масиви даних про носії (касети) з деталями, виготовленими в поточну зміну, комплекти інструменту і комплектуючих деталей (при складанні), маршрутні карти технологічного процесу виготовлення виробів, змінно-добові завдання, стани транспортних засобів, стан складу і стан устаткування. У кінці етапу сформовані масиви записуються у відповідні блоки бази даних. Модуль *BLDT* через єдину інформаційну базу готує в системі необхідну інформацію для функціонування модулів *PLAN* і *MDMP*.

Модуль *PLAN* реалізує функцію оперативного планування у призначений для складання розкладів роботи модулів основного технологічного устаткування і побудови управляючої моделі ГВС виходячи з номенклатури та обсягу виробництва. Планування здійснюється на основі імітаційного моделювання функціонування ГВС сітковою моделлю процесу. Використання аналітико-імітаційного методу планування дозволяє враховувати компонування ГВС, технологічні маршрути партії деталей, їх трудомісткості, структуру транспортної сітки і витрати часу на переналагодження устаткування, що задаються структурою та параметрами сіткової моделі процесів, сформованої в модулі *BLDT*. Партії деталей з наявної множини претендентів на завантаження основного технологічного устаткування вибираються виходячи з набору правил переваги і стратегій транспортного обслуговування. При цьому з різних варіантів розкладів, побудованих у результаті імітаційного моделювання за кожним правилом, вибирається варіант, що задовольняє критерію мінімуму загальної тривалості виробничого циклу. Даний критерій забезпечує безупинне і рівномірне завантаження основного технологічного устаткування, мінімізацію загального терміну виготовлення і комплектування деталей, розміру і вартості незавершеного виробництва, а також можливих відхилень від встановленого плану за обсягом і номенклатурою.

Модуль працює циклічно в три етапи. *Перший етап* – підготовка вихідних даних для проведення імітаційного експерименту. При цьому за інформацією про процес функціонування й за вибраними для даного експерименту правилами переваги формується сіткова модель процесу управління ГВС, що враховує тимчасові параметри виконуваних операцій.

На *другому етапі* роботи модуля проводиться, власне,

планування і побудова розкладу роботи одиниць виробничого устаткування за допомогою трансляції сіткової управляючої моделі (УМ). При цьому моменти запуску/випуску деталей на деякій одиниці виробничого устаткування визначаються за модельним часом моменту $G_k^*(t_j)$ – k -го спрацьовування переходу сіткової моделі процесу управління ГВС, а розклад представляється як множина часових рядів $G_k^* = \{(G_k^*(t_j) / k \in \{1, 2, 3, \dots\}, j = 1, m\}$ моментів спрацьовувань переходів сітки. Конфлікти на переходах вирішуються за допомогою вибраного правила переваги на першому етапі.

На *третьому етапі* роботи модуля відбирається варіант плану з мінімальним часом виробничого циклу з усіх складених розкладів, що відповідають визначеним правилам переваги зі встановленого набору. Кожний знову побудований розклад порівнюється з тим, що вважався найкращим. При цьому час виробничого циклу побудованого розкладу визначається як максимум модельного часу моментів останніх спрацьовувань усіх переходів сіткової моделі процесу управління ГВС. За найкращим варіантом розкладу, з огляду на оптимальне правило переваги і зв'язок входів і виходів об'єкта управління з елементами сіткової моделі, будується сіткова управляюча модель ГВС. Модуль *PLAN* завершує свою роботу записом результатів оперативного планування (розклад роботи і сіткова управляюча модель) у визначені блоки інформаційної бази ГВС.

Модуль MDMP диспетчеризації матеріальних потоків призначений для обробки інформації про хід виробничого процесу, вибору чергових претендентів (носіїв з деталями) на обслуговування, формування і передачі управляючих впливів до нижнього рівня управління.

Модуль працює в два етапи:

- 1) опитування гнучких виробничих модулів, транспортних засобів, робота-штабелера автоматизованого складу і формування черги претендентів на завантаження виробничих засобів;
- 2) вибір і посилення чергового носія на обслуговування.

На *першому етапі* роботи модуля визначається поточний стан виробничого устаткування за станом M маркування сіткової управляючої моделі і вектора V сигналів зворотного зв'язку з об'єктом управління. Далі серед усіх справних і незайнятих одиниць устаткування визначається та, яка на поточний момент часу має найвищий пріоритет відповідно до вибраного в модулі *PLAN* правила переваги. У сітковій моделі ці дії пов'язані з пошуком збуджених переходів T_M і вибором того з них, котрий у даній ситуації може спрацювати.

На *другому етапі* роботи модуля формується новий стан управляючої моделі, що є наслідком спрацьовування вибраного на першому етапі збудженого переходу. У новому стані маркування визначається вектор управляючих впливів Y і видається завдання роботу-штабелеру автоматизованого складу чи автономному транспортному

засобу на транспортування носія, або гнучкому виробничому модулю на виконання необхідної виробничої операції. Потім робота модуля *MDMP* циклічно повторюється.

Алгоритм функціонування модуля *MDMP* зображений на рис. 4.7.

Модуль *KNTR* оперативного контролю призначений для диспетчерської діагностики ходу виконання розкладу роботи, обробки інформації диспетчера про хід виробничого процесу і прийняття рішення про порядок подальшої роботи. Контрольованими параметрами служать терміни запуску/випуску носіїв деталей відповідно до розкладу, побудованого в модулі *PLAN*, а також параметри поточного стану одиниць виробничого устаткування ГВС. Вихідними даними модуля є розклад роботи виробничого устаткування G^* , управляюча модель і локальні резерви часу виконання операцій виробничого процесу $\Delta_{\text{доп}, k}$. Відхиленнями в роботі ГВС від передбаченого розкладу вважаються збій (аварія) у роботі виробничої системи, відхилення Δ_k^j від планового моменту часу виконання виробничої операції. У тих випадках, коли зафіксоване відхилення, модуль *KNTR* діагностує ситуацію і приймає рішення про її ліквідацію наявними засобами в складі модуля корегування *KORR*. Контроль і діагностика ситуацій ведеться безупинно протягом усього періоду функціонування модуля *MDMP* (періоду управління в реальному часі).

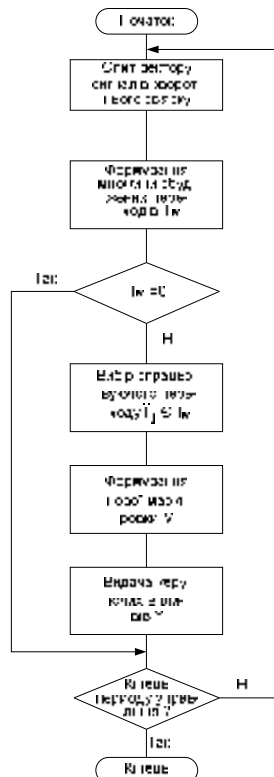


Рис. 4.7. Алгоритм диспетчеризації за управляючою моделлю

Алгоритм функціонування модуля *KNTR* показаний на рис. 4.8.

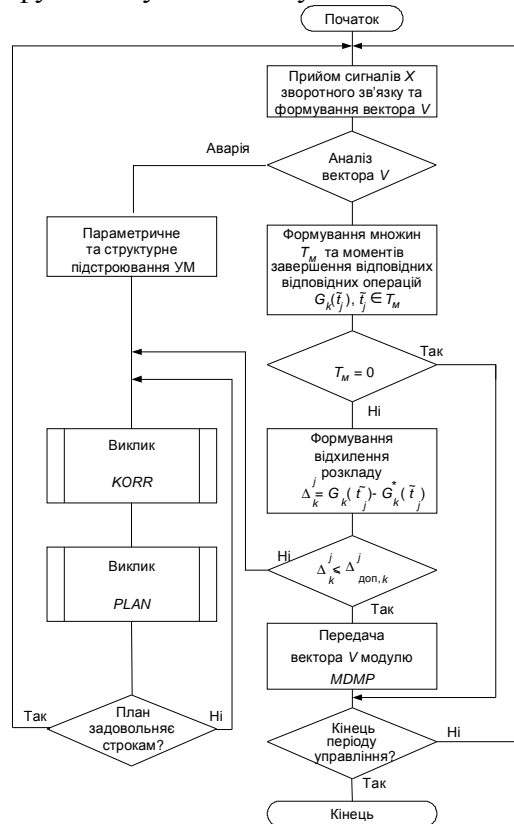


Рис. 4.8. Алгоритм оперативного контролю

Модуль KORR оперативного корегування призначений для відпрацьовування аварійних ситуацій і відхилень ходу виробництва додаванням у систему ресурсів чи зміною планового завдання. Модуль коригує інформацію про стан виробничих потужностей і планове завдання у діалоговому режимі з диспетчером ГВС.

За своїми функціональними можливостями модуль *KORR* подібний модулю *BLDT*, оскільки в модулі оперативного корегування модифікуються параметри і структура управляючої моделі на підставі введених диспетчером даних. Однак модуль *KORR* може викликатися за необхідності диспетчером і виконувати додаткову функцію включення фонових завдань виробничої підсистеми. Використовуючи локальні резерви, диспетчер ГВС може включити додаткові замовлення на обслуговування фонових робіт, виконання транспортних операцій і операцій завантаження складу через дану функцію модуля *KORR*. Тому модуль оперативного корегування має дві точки входу: з модулів *MAIN* і *KNTR*. Робота модуля завершується передачею управління модулю *MAIN* чи модулю *PLAN*.

Алгоритм функціонування модуля *KORR* зображений на рис. 4.9.

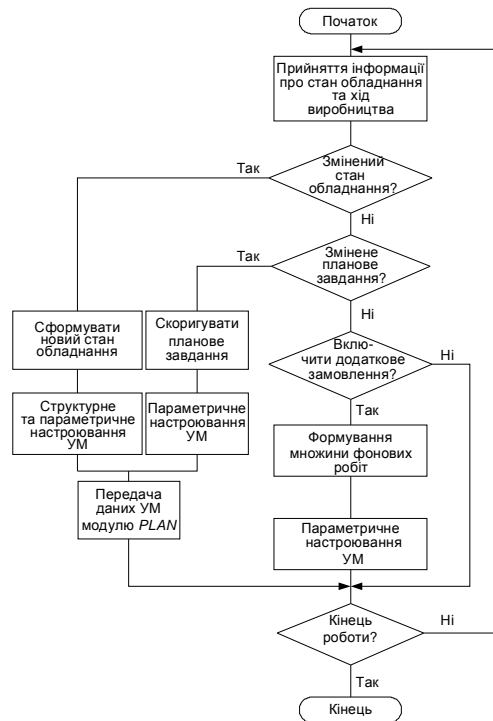


Рис. 4.9. Алгоритм функціонування модуля оперативного корегування

Модуль MDIP призначений для обробки запитів диспетчера ГВС і формування диспетчеру системних повідомлень. При роботі з комплексом програм у період управління в реальному часі диспетчеру ГВС подаються різні повідомлення, у відповідь на які він повинен виконати визначені дії. Усі повідомлення формуються в результаті інтерпретації стану управляючої моделі ГВС мовою, зрозумілою диспетчеру.

Повідомлення, що видаються в ході диспетчеризації, поділяються на запитальні та синхронізуючі.

Синхронізуючі повідомлення інформують оператора про стан виробничого устаткування, предметів виробництва і про важливі з погляду контролю події, що не ідентифікуються інформаційними засобами на об'єкті управління.

За допомогою *запитальних повідомлень* модуль MDIP запитує необхідну інформацію для роботи модулів MDMP, KNTR, KORR.

Модуль запускається в роботу одночасно з модулями MDMP, KNTR і взаємодіє з ними за допомогою посилення повідомлень у ході диспетчеризації матеріальних потоків ГВС.

Алгоритм функціонування модуля MDMP показаний на рис 4.10.

Модуль DOC призначений для обліку стану виробництва і формування довідкової інформації диспетчеру ГВС. Він представляє собою діалоговий комплекс програм видачі необхідних інформаційних

повідомлень.

Облік стану виробництва ведеться за показниками, серед яких можна виділити наступні: змінно-добове завдання, виконання змінно-добового завдання, простої устаткування через транспортування, переналагодження та аварії, сумарні простої, модель стану складу, модель стану виробничого устаткування, стан оперативних нагромаджувачів, незавершене виробництво на поточну добу, оперативний розклад запуску/випуску деталей, інформація про носії з деталями, що виготовляються в поточну зміну.

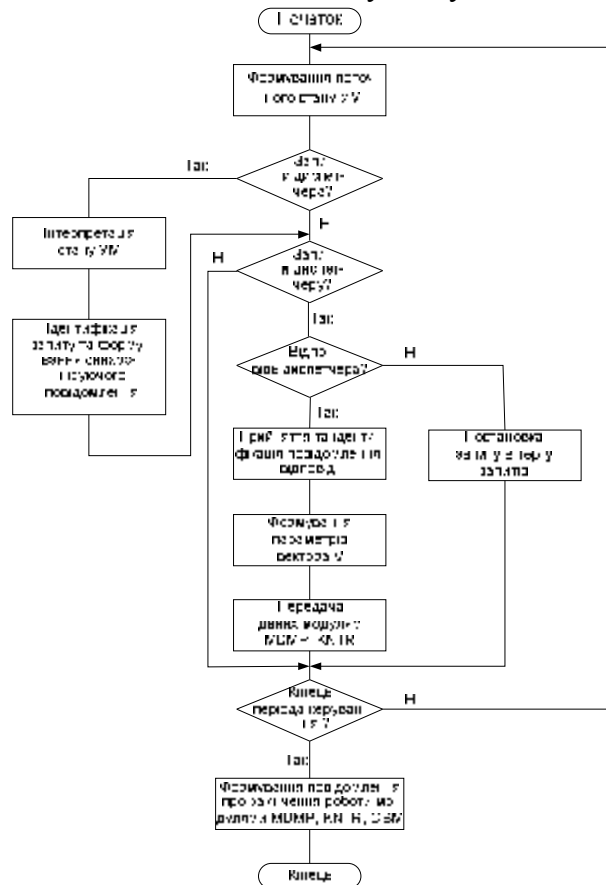


Рис. 4.10. Алгоритм функціонування модуля диспетчеризації запитів і системних повідомлень

Модуль запускається в міру необхідності диспетчером ГВС незалежно від задіяних в конкретний момент модулів комплексу.

Модуль ABND призначений для відтворення інформаційного стану системи при аварійному виході модулів з комплексу програм чи збоях устаткування, у тому числі й ЕОМ.

Модуль виконує наступні функції: встановлення наявності та можливості доступу до даних файлів системи; визначення можливої точки входу для продовження роботи системи.

Робота модуля *ABND* ініціюється вказівкою режиму роботи

комплексу програм у модулі *MAIN* і здійснюється в автономному режимі з видачею діагностичних повідомлень і рекомендацій диспетчеру ГВС по продовженню роботи з комплексом програм.

У випадку аварійного виходу із системи необхідно запустити модуль *ABND* і привести устаткування ГВС у вихідний стан.

Модуль *TEST* призначений для перевірки працездатності модулів комплексу СОУ ГВС.

Тестування проводиться відповідно до встановлених в системі контрольних прикладів, що задаються визначеними програмами (протоколами роботи) системи, вибраними диспетчером ГВС за своїм розсудом. Потім робота здійснюється, як і в нормальному режимі функціонування СОУ, але над тестовою інформаційною базою даних.

Модуль *OBM* призначений для обміну інформацією комплексу програм з нижчим рівнем управління виробничими модулями і є модулем міжрівневого обміну даними каналами зв'язку.

Модуль *OBM* виконує наступні функції: передачу повідомлень каналами зв'язку; чекання підтвердження прийому переданого повідомлення; прийом символів, що підтверджують зміну стану ГВС; прийом управляючих повідомлень.

Даний модуль слугує для передачі даних (команд управління) транспортним автономним засобам і роботам-штабелерам автоматизованого складу, а також управляючих програм для пристроїв ЧПУ гнучких виробничих модулів.

Модулі обміну, що функціонують на нижчому рівні в складі програмних модулів управління *GPM*, *AS*, *ATM*, виконують наступні функції: прийом повідомлень (управляючих команд); передача інформаційних повідомлень. Ці модулі, здійснюючи зв'язок з верхнім рівнем і приймаючи командні вказівки, реалізують власне управління одиницями виробничого устаткування ГВС.

Управління організується на основі сіткових моделей алгоритмів управління гнучкими виробничими модулями, роботами-штабелерами, автоматизованими складами і електророботами, отриманими у результаті декомпозиції управляючої моделі ГВС на алгоритми нижчих рівнів управління.

Повідомлення, передані від ЕОМ верхнього рівня до пристроїв ЧПУ, поділяються на інформаційні та управляючі. Від пристрою ЧПУ до верхнього рівня передаються тільки інформаційні повідомлення. Для ЕОМ верхнього рівня управління встановлені наступні види повідомлень процедури обміну: інформаційні (*D* – дані, *U* – вказівки); управляючі (*S* – ініціалізація обміну, *E* – кінець повідомлення). Для пристрою ЧПУ нижнього рівня управління встановлені тільки інформаційні повідомлення наступних видів: *N* – заперечення прийому; *C* – підтвердження прийому; *F* – кінець виконання операції на устаткуванні; *A* – аварія на устаткуванні. Комплекс модулів міжрівневого обміну на основі наведених повідомлень реалізує єдину мову обміну, тобто деякий

формат ідентифікації тексту повідомлень, що надходять по каналах зв'язку між ЕОМ верхнього рівня і пристроями ЧПУ нижнього рівня ГВС.

Інтерфейс комплексу і подання даних у пакеті програм СОУ ГВС. В основі програмного забезпечення СОУ ГВС лежить управляючий модуль *MAIN*, що ініціює початок функціонування комплексу програм, управляє порядком проходження програмних модулів і виконує діалог між диспетчером і програмним забезпеченням СОУ ГВС (рис. 4.11).

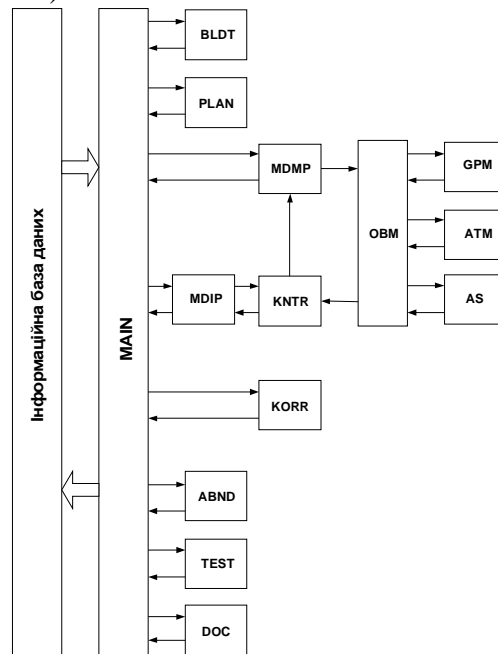


Рис. 4.11. Схема взаємодії програмних модулів комплексу програм СОУ ГВС

Диспетчер ГВС у діалоговій формі вводить у вихідні дані для оперативного планування і дані про початковий стан устаткування, тобто створює інформаційний стан СОУ, використовуючи для цього модуль *BLDT*. Потім диспетчер передає управління модулю оперативного планування *PLAN*, що складає розклад і підготовляє дані до процесу диспетчеризації. Після завершення планування запускається модуль диспетчеризації матеріальних потоків ГВС, що через модуль міжрівневого обміну *OBM* власне і здійснює управління транспортом (*ATM*), складом (*AS*), технологічним устаткуванням (*GPM*). Коли на об'єкті управління відбувається яка-небудь подія, аналізується ситуація, що створилася (*KNTR*), і в залежності від результатів аналізу управління повертається до *MDMP* чи передається модулю *KORR*. Якщо в результаті аналізу виявлене порушення нормального функціонування виробництва, то в діалоговому режимі корегується оперативна інформація і формуються вихідні дані для оперативного редагування розкладу функціонування ГВС (за допомогою модуля *PLAN*) на період роботи, що залишився.

Диспетчер управляє ходом функціонування системи в період диспетчеризації через модуль *MDIP* і може одержувати довідкову інформацію про стан виробництва в будь-який момент часу через модуль *DOC*.

Оскільки комплекс програм побудований за модульним принципом, то процес функціонування програмного забезпечення представляється у вигляді сукупності підпроцесів взаємодії диспетчера ГВС із кожним модулем окремо.

Реалізація функцій, покладених на СОУ ГВС, вимагає передачі, прийому, збереження та перетворення великого обсягу інформації. Виходячи з модульного принципу побудови комплексу програм з організацією управління на основі управляючої моделі ГВС, усі вхідні та вихідні дані програмних модулів організовані у вигляді блоків єдиної інформаційної бази даних ЕОМ верхнього рівня. Дані блоки поділяються на чотири групи: основні, службові, допоміжні, інформаційні.

До групи *основних* відносяться блоки, що відповідають за стан одиниць виробничого устаткування ГВС, матеріальних та інформаційних потоків, тобто описують управляючу модель ГВС.

До групи *службових* відносяться блоки, що містять маршрутні карти технологічного процесу виготовлення всіх допустимих для даної ГВС типів деталей, блоки управляючих програм технологічного устаткування. Наявність службових блоків у системі обов'язкова для початку функціонування комплексу програм.

До групи *допоміжних* відносяться блоки, що створюються і використовуються під час роботи комплексу програм для збереження і передачі проміжних результатів і знищуються по закінченні роботи.

До групи *інформаційних* відносяться блоки, що відповідають за виробничу діяльність ГВС і призначені для інформування диспетчера ГВС.

Отже, програмне забезпечення СОУ ГВС, спроектоване із врахуванням ряду вимог, виконання яких дозволило уніфікувати програмні засоби управління, забезпечує їх незалежність від змісту предметної області виробничої системи. Досягнута уніфікація програмного забезпечення дозволяє тиражувати і супроводжувати його у виробничих системах без принципових змін в організації програмних засобів управління шляхом реорганізації єдиної інформаційної бази даних.

Контрольні запитання

1. Розкрийте складові сіткової моделі виробничого процесу ГВС.
2. Поясніть зміст формалізованого подання і процедури вирішення конфліктних ситуацій, що утворюються в ході виконання виробничого процесу під час оперативно-диспетчерського управління

ГВС.

3. Наведіть відмінності сіткової моделі управляючого процесу від сіткової моделі алгоритму управління.

4. Сформулюйте етапи методу синтеза сіткової моделі алгоритмів управління ГВС.

5. Розкрийте задачі і особливості побудови програмних засобів системи оперативного управління ГВС.

6. Визначте базові модулі системи оперативного управління ГВС

7. Поясніть порядок застосування програмних модулів і структур даних системи оперативного управління ГВС.

*«Щира думка веде нас до
правильних дій нітрохи не гірше,
ніж розум» (Платон Аристокл)*

Розділ 4

ПРОЕКТУВАННЯ ПРОЦЕСІВ УПРАВЛІННЯ ГВС

4.1. Технологія проектування алгоритмів управління виробничими процесами ГВС

Алгоритм управління (АУ) – це послідовність цілеспрямованих дій, пов'язаних із забезпеченням виконання операцій виробничого процесу (ВП) у ГВС. АУ може бути поданий як набір операторів процесу управління, що відповідають операціям ВП, який визначає порядок їх виконання. В результаті модель ВП може слугувати вихідними даними для проектування АУ ГВС.

Отже, розробка системи управління ГВС передбачає послідовне здійснення наступних етапів проектування:

- формалізація матеріально-інформаційних процесів ГВС як об'єкту управління (ОУ);
- аналіз властивостей функціонування ОУ за його моделлю;
- синтез моделей алгоритмів управління;
- проектування структури системи управління.

Про перший та другий етап проектування частково йшла мова у розділі 3, коли процес функціонування виробничої системи представлявся у формі сіткової моделі, в якій відтворюється ієрархія підпроцесів функціонування складових елементів ГВС.

Розглянемо сіткову інтерпретацію ієрархії підпроцесів функціонування складових елементів ГВС. Багатономенклатурне виробництво відноситься до класу складних систем з множиною різних динамічних об'єктів – матеріальних потоків, устаткування. Найчастіше ці об'єкти мають додаткові атрибути, які дозволяють розрізняти їх і використовувати ці розбіжності для управління функціонуванням системи.

В залежності від необхідного за специфікацією предметної області ГВС рівня детальності подання, ієрархія в складних виробничих процесах задається структурою, яка складається з множини більш простих підпроцесів, функції яких розкриваються операціями над динамічними об'єктами ГВС. Інакше кажучи, ВП складається з n ієрархічних рівнів, де деякі чи всі операції підпроцесу i -го рівня ($2 \leq i \leq n$) самі є процесами, які складаються з операцій, віднесених до $(i - 1)$ -го рівня. Старший рівень відповідає тільки одному підпроцесу.

Як модель ВП, яка ефективно відображає встановлену ієрархію підпроцесів і адекватно представляє операції перетворення матеріальних потоків в ГВС, застосовується ІКСП з наступною

інтерпретацією елементів: позиціям поставлені у відповідність операції моделюємого процесу і надані їм ресурси технологічної підсистеми (устаткування); переходам – події зміни операцій; маркерам – одиниці матеріальних потоків і стани ресурсів. З огляду на обрану форму інтерпретації, позиції-дублери набувають наступного сенсу: кожній такій позиції відповідає макрооперація, яка сама є підпроцесом більш низького рівня складності представлення, опис якого задається заміщаючим сітковим блоком.

Будемо вважати, що ІКСП N^* адекватна виробничому процесу, якщо між її елементами і атрибутами специфікації виробничого процесу SP встановлена взаємооднозначна відповідність.

Визначення 4.1. Сітковою моделлю виробничого процесу будемо називати таку ІКСП $N_{\Pi} = (\{N_i\}, Q, Z)$, для якої виконуються наступні умови:

$$N_i = (P_i, T_i, \Omega_i, F_i, H_i, \lambda_i, \psi_i, \mu_{i0}),$$

$$P_i = P_{ci} \cup P_{di} \cup P_{ri}, P_{ci} \cap P_{di} = \emptyset, (P_{ci} \cup P_{di}) \cap P_{ri} = \emptyset, |(P_{ci} \cup P_{di})| \geq |P_{ri}|;$$

$$\Omega_i = \Omega_{ci} \cup \Omega_{ri}; \Omega_{ci} \cap \Omega_{ri} = \emptyset;$$

де $Q: \{N_i\} \rightarrow \{N_i\}$ – ієрархія складових виробничого процесу (підпроцесів);

$$P_{ci} = \{p_c \mid p_c = f_1(o_i), o_i \in O\} \text{ – множина операцій } i\text{-го підпроцесу};$$

$P_{di} = \{p_d \mid p_d = f_2(N_j), Q(N_j) = N_i\}$ – множина складових підпроцесів i -го підпроцесу;

$P_{ri} = \{p_r \mid p_r = f_3(m), m \in M\}$ – множина ресурсів операцій i -го підпроцесу;

$T_i = \{t \mid t = f_4(g), g(o_i, o_j) \in G\}$ – множина подій у зміні станів i -го підпроцесу;

$\Omega_{ci} = \{\omega_c \mid \omega_c = f_5(d), d \in D\}$ – множина елементів матеріального потоку;

$\Omega_{ri} = \{\omega_r \mid \omega_r = f_6(s(m)), s(m) \in S_M\}$ – множина функціональних станів ресурсів;

$$\lambda_i = \{\lambda_{ii} \mid t \in T_i, \lambda_{ii} = f_7(TM)\}; \psi_i = \{\psi_{ii} \mid t \in T_i, \psi_{ii} = f_8(TM)\};$$

$$\mu_{i0}(P_i \setminus P_{ri}) = \left\{ \mu_{i0}(p, \omega_c) \mid p \in P_i \setminus P_{ri}, \omega_c \in \Omega_{ci}, \mu_{i0}(p, \omega_c) = f_9(s(d)), s(d) \in S_D \right\};$$

$$\mu_{i0}(P_{ri}) = \{\mu_{i0}(p_r, \omega_r) \mid p_r \in P_{ri}, \omega_r \in \Omega_{ri}, \mu_{i0}(p_r, \omega_r) = e(m), m \in M\};$$

$Z = \{z(p, \omega) | p \in P_{ci}, \omega \in \Omega_{ci}, z(p, \omega) = \tau(o_i), o_i \in O\}$ – параметри

часу виконання операцій виробничого процесу;

$e(m)$ – ємність ресурсу m ;

$t(o_i)$ – час виконання операції.

Тоді виробничий процес функціонування ГВС можна представити як сукупність взаємозалежних процесів обслуговування ГВМ, кожний з яких включає різні підпроцеси підготовки ГВМ до обслуговування заявок-виробів (підготовка до налагодження, налагодження, доставка об'єктів складання), самообслуговування заявок, що надходять, на виконання виробничих операцій і доставку результатів на АС.

Таким чином, процедура формування сіткового подання виробничого процесу полягає у виконанні наступних дій:

- формування описів матеріальних потоків і станів виробничих засобів;
- визначення схем руху засобів виробництва за обробними ресурсами і виділення типових операцій і підпроцесів;
- декомпозиція досліджуваного процесу;
- структуризація повного опису з сіткових подань підпроцесів;
- параметризація моделі відповідно до існуючих технологій виготовлення і припустимих схем руху матеріальних потоків.

4.2. Синтез моделей алгоритмів управління виробничими процесами ГВС

Третій етап проектування систем управління ГВС полягає в одержанні обумовленої виконуваним виробничим процесом структури процесу управління, а потім у формуванні коректного алгоритму управління. Оскільки в отриманій моделі N_{Π} можна реалізувати майже будь-яку послідовність рухів матеріального потоку, визначену в рамках сформованої специфікації предметної області ГВС, то структура процесу управління вважається цілком заданою та адекватно відтвореною у ІКСП, якщо з кожною виробничою операцією співставити оператор управління, який ініціює її виконання. Тоді задача синтезу алгоритму управління полягає у заданні для множини переходів T моделі виробничого процесу N_{Π} відображень $h: P \times \Omega \rightarrow T$, що адекватно визначають порядок реалізації операцій у конфліктних ситуаціях $(P \times \Omega)$, та введенні додаткових елементів, що моделюють інформаційний обмін управляючої підсистеми і об'єкта управління.

Процес формування управління рухом матеріальних потоків по обробляючих ресурсах у ГВС полягає у встановленні в конфліктних ситуаціях невизначеності обслуговуємої заявки-предмета виробництва і засобу її обслуговування-ресурсу виробництва.

Згідно з таким поданням, конфліктні ситуації, що утворюються в ході виконання виробничого процесу (під час оперативного-

диспетчерського управління), задаються наступними типовими ситуаціями:

- обслуговування заявок спільним засобом (ресурсом);
- надання засобів (ресурсів) спільній заявці;
- виконання незалежних однієї від одної операцій.

У сітковій моделі виробничого процесу виділені ситуації мають своє формалізоване подання у вигляді наступних типових конфліктів на переходах T .

1. Конфлікт пов'язаних з доступом до спільного ресурсу p_r заявок ω_{ci}, ω_{cj} :

а) K_1 - конфлікт заявок від операції за виконання наступної спільної операції (рис. 4.1, а), де

$$K_1 = \bigcup_{t \in T} K_1(t); \quad K_1(t) = \{ \bullet c_t \mid \bullet c_t \in \bullet C_t : |\bullet C_t| > 1 \}; \quad T = \bigcup_{i=0}^p T_i;$$

б) K_2 - конфлікт заявок від різних операцій за виконання спільної операції p_c (рис. 4.1, б), де

$$K_2 = \bigcup_{p_c \in P_c} K_2(p_c); \quad K_2(p_c) = \{ t \mid t \in \bullet p_c, |\bullet p_c| > 1 \}; \quad P_c = \bigcup_{i=0}^p P_{ci};$$

в) K_3 - конфлікт заявок за спільний ресурс p_r (рис. 4.1, в), де

$$K_3 = \bigcup_{p_r \in P_r} K_3(p_r); \quad P_r = \bigcup_{i=0}^p P_{ri};$$

$$K_3(p_r) = \left\{ t \mid \left(\forall t_i, t_j \in p_r : (t_i \mathbf{I} t_j) = \emptyset \ \& \ (\bullet t_i \mathbf{I} \bullet t_j) \setminus \{ p_r \} = \emptyset \right) \right. \\ \left. \left(\exists \bullet c_{t_i} \in \bullet C_{t_i}, \bullet c_{t_j} \in \bullet C_{t_j} : \bullet c_{t_i} = \bullet c_{t_j} \right) \right\}.$$

2. K_4 - конфлікт різноманітних ресурсів p_{ri}, p_{rj} за заявку ω_{ci} від спільної операції p_c (рис. 4.2, а), де

$$K_4 = \bigcup_{p_c \in P_c} K_4(p_c);$$

$$K_4(p_c) = \{ t \mid \forall t_i, t_j \in p_c \exists \bullet c_{t_i} \in \bullet C_{t_i}, \bullet c_{t_j} \in \bullet C_{t_j} : \bullet c_{t_i} = \bullet c_{t_j} \}.$$

3. K_5 - конфлікт незалежно виконуваних заявок ω_{ci}, ω_{cj} (рис. 4.2, б), де

$$K_5 = \bigcup_{p_c \in P_c} K_5(p_c); \quad K_5(p_c) = \{ t \mid \forall t \in (T \setminus (K_2 \cup K_3 \cup K_4)) : t \in p_c \bullet \} \cup$$

$$\cup \{ K_2(p_c), K_3(p_r), K_4(p_c) \mid p_r \mathbf{I} \bullet p_c \neq \emptyset \ \& \ \bullet p_r \mathbf{I} p_c \neq \emptyset \}.$$

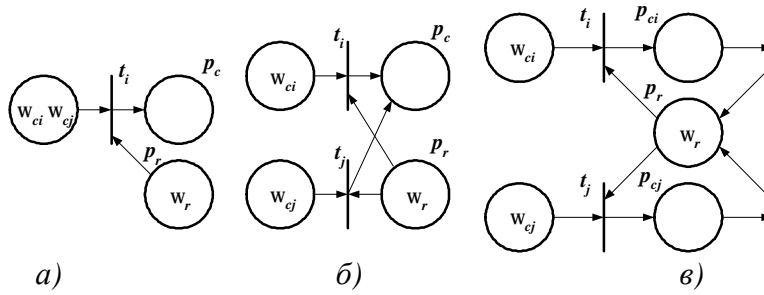


Рис. 4.1. Сіткове подання конфліктів заявок:

а, б – види конфліктів заявок за виконання спільної операції;
в – конфлікт заявок за загальний ресурс

Розгляд як конфлікту незалежно виконуваних заявок пов'язано з орієнтацією сіткової моделі на опис паралельно здійснюваних підпроцесів в умовах послідовного прийняття керуючих рішень.

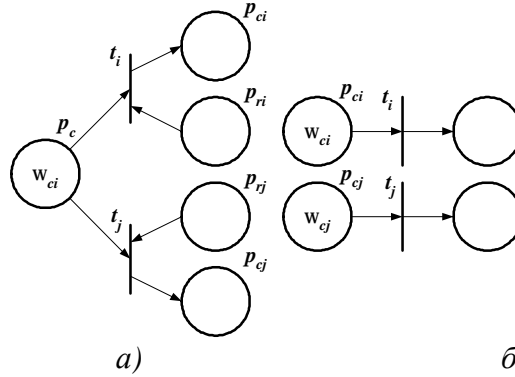


Рис. 4.2. Сіткове подання конфлікту:

а – устаткування за обслуговування заявки;
б – незалежно виконуваних операцій

Для вирішення конфліктів першого виду встановлюються пріоритети заявок, що надходять у кожному конкретному випадку, на обслуговування $PN_1: D \rightarrow \{1, 2, 3, \dots\}$ на базі правил переваги, які застосовувалися у розрахунку розкладу роботи технологічного устаткування при оперативному плануванні.

Тоді процедура вибору R_l , $l = \overline{1, 3}$ активізованої операції, яка входить у конфлікт K_1 **першого виду**, сформується на базі наступних співвідношень:

$$\begin{aligned} & \forall \cdot c_t^i, \cdot c_t^j \in K_1(t) \exists d_i, d_j \in D: [PN_1(d_i) > PN_1(d_j) \Rightarrow \\ & \Rightarrow \eta_t^1(\omega_{ci}) > \eta_t^1(\omega_{cj}) \Rightarrow R_1(\cdot c_t^i, \cdot c_t^j) = 1], \cdot c_t^i = [(p_c, \omega_{ci})]; \\ & \forall t_i, t_j \in K_l(p_{h_l}) \exists d_i, d_j \in D: [PN_1(d_i) > PN_1(d_j) \Rightarrow \\ & \Rightarrow \eta_{t_i}^l(\omega_{ci}) > \eta_{t_j}^l(\omega_{cj}) \Rightarrow R_l(t_i, t_j) = 1], l = \overline{2, 3}, p_{h_2} \in P_c, p_{h_3} \in P_r. \end{aligned}$$

Для вирішення конфлікту **другого виду** K_4 встановлюються

пріоритети на порядок надання устаткування $PN_2 : M \rightarrow \{1, 2, 3, \dots\}$ на базі стратегій транспортного обслуговування, визначених при оперативно-диспетчерському управлінні:

а) заявка направляється на найближчий за часом доставки вільний ГВМ;

б) заявка направляється на вільний ГВМ із мінімальним поточним завантаженням;

в) вибір вільного ГВМ здійснюється випадковим чином.

Процедура вибору R_4 активізованої операції, що входить у конфлікт K_4 другого виду, сформується на базі наступного співвідношення:

$$\begin{aligned} & \forall t_i, t_j \in K_4(p_c) \\ & \left(\exists m_i, m_j \in M : t_i \in p_{ri}, t_j \in p_{rj}, p_{ri} = f_3(m_i), p_{rj} = f_3(m_j) \right): \\ & \left[PN_2(m_i) > PN_2(m_j) \Rightarrow \eta_{p_c}^4(t_i) > \eta_{p_c}^4(t_j) \Rightarrow R_4(t_i, t_j) = 1 \right]. \end{aligned}$$

Конфлікт K_5 **третього виду** вирішується призначенням пріоритетів $PN_3 : O \rightarrow \{1, 2, 3, \dots\}$, виходячи з прийнятого першочергового порядку виконання незалежно активізованих операцій і підпроцесів у ВП. Процедура вибору R_5 активізованої операції, що входить у конфлікт K_5 третього виду, сформується на базі наступних співвідношень:

$$\begin{aligned} & \forall t_i \in K_g(p_h) \subset K_5 \forall t_j \in K_k(p_l) \subset K_5 \\ & \left(\exists o_i, o_j \in O : t_i \in p_{ci}, t_j \in p_{cj}, p_{ci} = f_1(o_i), p_{cj} = f_1(o_j), h \neq l \right): \\ & \left[PN_3(o_i) > PN_3(o_j) \Rightarrow \eta^5(t_i) > \eta^5(t_j) \Rightarrow R_5(t_i, t_j) = 1 \right]; g, k \in \{2, 3, 4\}. \end{aligned}$$

Враховуючи, що будь-якій перехід сіткової моделі може одночасно належати декільком різним типовим конфліктам, відображення η на підмножині переходів представляє собою сукупність предикатів PR умов спрацьовування, співвідношення яких формується на базі ієрархічної системи вирішення типових конфліктів (“вибір заявки обслуговування”–“надання ресурсу заявці”–“вибір порядку виконання незалежних заявок”). Іншими словами, на кожному рівні ієрархії типових конфліктів формується значення локального предикату умови спрацьовування переходу із врахуванням результатів вирішення на попередньому рівні. Тоді переходу $\overset{\circ}{t}_j$ сіткової моделі можна поставити у відповідність узагальнений предикат умови спрацьовування наступного виду:

$$B(\overset{\circ}{t}_j) = \bigwedge_{t_j^i \in \overset{\circ}{t}_j, c_{t_j^i} \in K_1(t_j^i)} q\left(\overset{\circ}{c}_{t_j^i} \bigwedge_{l=1,5} \left(A_l(\overset{\circ}{c}_{t_j^i}) \vee \overline{q_l}(t_j^i) \right) \right),$$

де $\overset{\circ}{t}_j = \{t_j^i \mid t_j^i \in N_i\}$ – перехід-зв'язка;

$\overline{q_l}(t_j^i)$ – логічна умова того, що перехід t_j^i входить до конфлікту l -го виду;

$$A_l(\cdot c_{t_j^i}) = \bigwedge_{t_h^i \in K_l \setminus \{t_j^i\}} \left(\overline{q_l(\cdot c_{t_h^i})} \bigwedge_{k=1, l-1} \left(A_k(\cdot c_{t_h^i}) \vee \overline{q_k(t_h^i)} \right) \vee R_l(t_j^i, t_h^i) \right) \quad -$$

локальний предикат умови спрацьовування переходу t_j^i за розподілом $\cdot c_{t_j^i}$ в ієрархічній системі вирішення конфлікту l -го виду, $l = \overline{2, 5}$;

$q(\cdot c_{t_j^i})$ – логічна умова того, що перехід t_j^i ініційований за розподілом $\cdot c_{t_j^i}$;

$$A_1(\cdot c_{t_j^i}) = \bigwedge_{\cdot c_{t_j^i}' \in K_1(t_j^i) \setminus \{\cdot c_{t_j^i}\}} \left(\overline{q(\cdot c_{t_j^i}')} \vee R_1(\cdot c_{t_j^i}, \cdot c_{t_j^i}') \right).$$

Визначення 4.2. Сітковою моделлю управляючого процесу називається предикатна ІКСП наступного виду:

$$N_y = (N_{\Pi}, PR),$$

де $PR = \{pr_j \mid pr_j = B(t_j^{\circ}), j = \overline{1, |T|}\}$ – множина предикатів умов спрацьовування переходів.

При формалізації і побудові сіткових моделей АУ на основі управляючого процесу (УП) необхідно забезпечити асинхронне узгодження взаємодій між операторами управління так, щоб моменти ініціації виробничих операцій визначалися тільки готовністю необхідних даних про стан ОУ. У цьому випадку управляюча підсистема ГВС формує управляючі впливи u_i у вигляді команд активізації виконання відповідних виробничих операцій в ОУ, після чого ОУ працює автономно, а управляюча підсистема очікує сигналу x_i від ОУ про закінчення виконання цих операцій. Тоді кожен оператор-дія, який виконує процедуру по управлінню деякою виробничою операцією визначеної тривалості, у моделі ВП заміняємо сітковим фрагментом, зображеним на рис. 4.3, де w_i , u_i – зовнішні позиції, що відображають відповідно видачу управляючого впливу і прийом сигналу про стан устаткування. Таким чином, для опису АУ на базі моделі УП N_y введемо спеціальні множини позицій, що представляють деякий вхідний і вихідний алфавіти, які характеризують асинхронні взаємодії з ОУ.

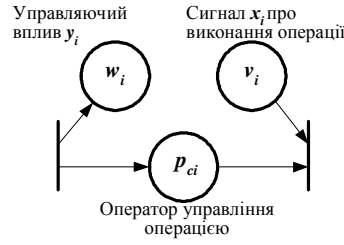


Рис. 4.3. Сіткове подання процедури управління виробничою операцією

Визначення 4.3. Сітковою моделлю алгоритму управління називається навантажена предикатна ІКСП, яка задається набором наступного виду:

$$N_A = (N_y, X, Y, V, W, f, \phi),$$

де $X = \{x_l \mid x_l = (b_1, b_2, b_3, b_4), l = \overline{1, L}\}$ – множина сигналів про стан устаткування;

b_i – логічні змінні, що відповідно вказують ознаку стану ОУ: несправність, зайнятість, налагодженість і робота;

$Y = \{y_k \mid y_k = (d, par_1, par_2), d \in D, k = \overline{1, K}\}$ – множина управляючих впливів; par_1, par_2 – управляючі параметри: адреса відправлення, номер програми обробки;

$$V = \left\{ v_j \mid \forall p_{cj} \in P_c \left(\exists \omega_l \in \bigcup_{i=0}^p \Omega_i : \mu(p_{cj}, \omega_l) \neq 0 \right) \exists v_j : v_j = p_{cj} \right\};$$

$$W = \left\{ w_j \mid \forall p_{cj} \in P_c \left(\exists \omega_k \in \bigcup_{i=0}^p \Omega_i : \mu(p_{cj}, \omega_k) \neq 0 \right) \exists w_j : w_j = p_{cj} \right\} -$$

множини зовнішніх позицій для подання асинхронної взаємодії з ОУ;

$$f : V \times \bigcup_{i=0}^p \Omega_i \rightarrow X; \quad \phi : W \times \bigcup_{i=0}^p \Omega_i \rightarrow Y - \text{функції навантаження}$$

на зовнішні позиції.

Згідно з принципом подійності управління, момент часу управління відповідає моменту надходження сигналу про готовність устаткування x_i (розміщення в позиції u_i мітки-сигналу). Тоді задача управління за сітковою моделлю полягає у виборі в цей момент однієї з множини готових до виконання операцій (вибрати за істинним значенням предиката pr_j відповідний перехід спрацьовування) і вказівки шляху її реалізації у заданні управляючого впливу u_i (помістити в позицію w_i мітку управління). Множина предикатів сіткової моделі N_A формалізує стратегію управління ГВС.

Таким чином, *метод синтезу* сіткових моделей коректних алгоритмів управління на основі отриманого і перевіреного опису ВП можна представити у вигляді послідовності наступних етапів:

- сформувати структуру УП і визначити склад необхідних операторів управління операціями ВП;
- усунути всі конфлікти через наявність спільних ресурсів та альтернативний характер ВП за допомогою предикатів умов спрацьовування переходів;
- вибрати всі оператори-дії, для яких сформувати набори управляючих впливів і сигналів про стан відповідного устаткування;
- сформувати для кожної оператора-дії сіткове подання асинхронної взаємодії з виробничим устаткуванням.

У кожному конкретному випадку правила призначення пріоритетів при формуванні множини предикатів обираються за результатами імітаційних експериментів над сітковою моделлю ВП і порівнянням їх за обраним критерієм. Використання ж правил переваги допускає ефективну організацію управління по ситуаціях при прямій диспетчеризації функціонування ГВС.

4.3. Організація робіт по створенню систем оперативного управління ГВС

Задачі та особливості побудови програмних засобів управління. При створенні ГВС однією з головних є задача оперативного управління функціонуванням виробничого устаткування і синхронізації матеріальних потоків. При цьому розробка програмного забезпечення, що реалізує алгоритми управління, синтезовані на основі розглянутих модифікацій сіток Петрі, буде найбільш трудомістким етапом проектування ГВС.

Розроблювані сьогодні ГВС є унікальними системами, пристосованими для розв'язку конкретних задач, обумовлених специфікою автоматизованого об'єкта. Витрати на проектування і подальший розвиток можна знизити типізацією рішень в області створення ГВС, у тому числі і при розробці програмних середовищ оперативного управління матеріальними потоками ГВС.

Одним зі шляхів типізації рішення в цій області є використання принципу модульності, відповідно до якого ГВС компонуються з окремих типових модулів і підсистем.

Система оперативного управління ГВС представляє собою баторівневу ієрархічну систему зі зворотними зв'язками, організовану у вигляді розподіленої мережі комп'ютерних та мікропроцесорних пристроїв управління. Схема організації СОУ ГВС була наведена на рис. 1.8. Нижній рівень вирішує задачі локального управління послідовністю технологічних операцій на виробничому устаткуванні. Управляючі сигнали, сформовані цим рівнем управління, визначають моменти включення та виключення компонентів устаткування для виконання необхідних виробничих операцій, забезпечують програмну роботу устаткування. Верхній рівень управління забезпечує взаємодію всіх одиниць устаткування при проходженні через ГВС запланованих

на плановий період виробів-замовлень. На цьому рівні збирається і відображається поточна інформація, що визначає стан ділянки, координується робота всіх засобів управління нижнього рівня, формується інформація і повідомлення (запити до операторів ГВС), здійснюється взаємодія з вищестоячим рівнем управління.

При експлуатації ГВС, використовуючи програмні засоби управління, користувач може вирішувати задачі автоматизації оперативного управління, диспетчеризації матеріальних потоків, оперативного регулювання і коректування планового завдання, автоматизації обліку стану виробництва, автоматизації передачі інформації в ієрархічних системах управління.

Головною відмінною рисою і принципом побудови розроблених програмних засобів є управління процесами функціонування ГВС на основі інтерпретації сіткових управляючих моделей. На етапі проектування алгоритмів чи управління їх гнучкою перебудовою інтерпретація управляючих моделей виконується з метою аналізу властивостей алгоритмів і розв'язку різних оптимізаційних задач. На етапі реалізації алгоритмів управління спроектовані сіткові моделі використовуються для управління процесами функціонування ГВС. При цьому управляюча модель інтерпретується на основі тих же принципів, що і на етапі проектування.

На рис. 4.4 зображена схема організації управління ГВС на основі використання управляючих моделей, що відображає послідовність взаємозалежних фаз управління.

На вхід СОУ надходить інформація про конкретну конфігурацію об'єкта управління ОУ, про умови його функціонування, специфікації управляємих процесів, а також планове завдання та вимоги оптимізації. У блоці $B1$ будується ієрархічна структура управляючої моделі ГВС і формується область значень оптимізованих параметрів. Виходом блоку $B1$ служить сіткова модель N_{Π} процесу функціонування ГВС і набір правил призначення пріоритетів (PN_i) . Блок $B2$ реалізує фазу оперативного планування й оптимізує параметри сіткової моделі процесу функціонування з метою формування N_y – управляючої моделі ГВС. Вихідними даними є сіткова модель N_{Π} і кодовані параметрами одержуваної управляючої моделі правила призначення пріоритетів, що формалізують прийняті в СОУ стратегії управління.

Фаза диспетчеризації реалізується блоками $B3$ і $B4$. Блок $B3$ диспетчеризації запитів і команд відповідно до вибраної у фазі планування стратегії управління за поточним значенням вектора-сигналу V зворотного зв'язку з ОУ формує нові значення маркування M_{k+1} управляючої моделі і вектора вхідного алфавіту V взаємодії з ОУ. Блок $B4$ здійснює інтерпретацію поточного стану управляючої моделі і перетворює відповідні елементи N_y на параметри управління – управляючі сигнали Y_k .

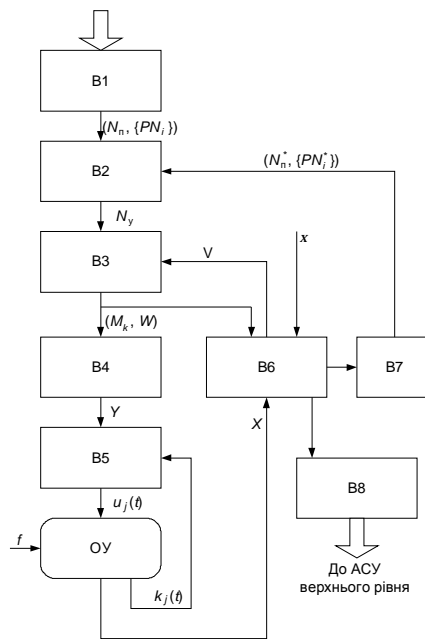


Рис. 4.4. Схема організації управління ГВС на основі використання управляючих моделей: B1 – генерація структури та настроювання параметрів управляючої моделі; B2 – оптимізація параметрів управляючої моделі; B3 – трансляція управляючої моделі (диспетчеризація запитів та команд); B4 – інтерпретація стану управляючої моделі M_k ; B5 – локальне управління обладнанням; B6 – діагностика стану та контроль показників функціонування ОУ; B7 – обробка аварійних ситуацій та відхилень (коректування структури та параметрів управляючої моделі);

B8 – статистичний облік; ОУ – об'єкт управління (обладнання ГВС)

Кожна операція для СОУ характеризується тільки тривалістю, ресурсом, що використовується для її виконання, і предметом праці – партією деталей. У зв'язку з цим управляючими сигналами є команди запуску операцій із вказівкою часу початку і місця їх виконання для всіх технологічних процесів ГВС у плановому періоді. Ці команди обробляє пристрій локального управління відповідними ресурсами (блок B5). Пристрій локального управління, приймаючи з верхнього рівня команди запуску необхідної операції, реалізує управління устаткуванням, використовуючи свій набір сигналів про стан ОУ $\{K_i(t)\}$ і набір управляючих команд $\{u_i(t)\}$ механізмів устаткування. По закінченні виконання необхідної операції ОУ формує сигнали X_k про зміну стану своїх компонентів, що надходять до блоку діагностики станів і показників функціонування ГВС (блок B6).

Блок B6 реалізує функції фази контролю. В цьому блоці обробляється первинна інформація про стан ОУ і встановлюється відповідність дійсного ходу функціонування виробництва X_k запланованому M_k . Якщо відхилення вважаються допустимими, то

управління передається блоку *B3*, на вхід якого подається вектор вхідного алфавіту сигналів *V* зворотного зв'язку по управлінню. Якщо блок *B6* фіксує аварійні ситуації (вихід з ладу устаткування) чи недопустимі відхилення функціонування ОУ (порушення у виконанні планового завдання), то управління передається на блок *B7*.

Блок *B7* виконує функції регулювання і призначений для обробки аварійних ситуацій і усунення відхилень виробництва додаванням у систему ресурсів чи зміною планового завдання. У блоці коректуються структури і параметри управляючої моделі і формується нове планове завдання із врахуванням введених оператором ГВС додаткових ресурсів. У результаті на вхід блока *B2* подається нова відкоректована модель N_p^* процесу функціонування ОУ. При цьому до закінчення процесу формування нової сіткової управляючої моделі управління ГВС організовується за старою моделлю в режимі прямої диспетчеризації по одній із встановлених стратегій управління.

У СОУ реалізується також статистичний облік станів виробництва для використання і передачі інформації АСУ верхнього рівня (блок *B8*). На блок *B8* може вплинути і оператор ГВС через повідомлення x про зміни в інформаційному стані СОУ.

Таким чином, управління ГВС складається з фаз генерації, планування, диспетчеризації, контролю, обліку і регулювання, що виконуються послідовно, а сама СОУ реалізує принцип управління зі зворотним зв'язком по стану, що враховує вплив на ОУ всіх збурюючих впливів (f, x).

Організація управління на базі управляючих моделей дозволяє говорити про інваріантність розробленого принципу до конкретного змісту виконуваних технологічних операцій і конфігурації ОУ. Цей принцип може бути типізований не тільки для дискретних складальних, але і для механообробних виробництв. У цьому полягає принципова відмінність і перевага розробленого пакету СОУ.

Для автоматизації фази оперативного планування розклад функціонування устаткування одержують способом динамічної трансляції сіткових управляючих моделей. Однак управління ведеться не відповідно до заздалегідь встановленого у фазі планування розкладом послідовності дій, а за оптимальними стратегіями управління, що формують цю послідовність дій у ході обчислень. Розклад використовується у фазі регулювання, де корегується динаміка управляючої підсистеми до реального ходу керованих процесів ГВС. При цьому враховуються реальні умови, що накладаються на ОУ: стан устаткування, виробів, виробничих показників функціонування.

Така організація управління дозволяє виявляти збої в системотехнічних діях управляючої підсистеми на основі аналізу фактичних станів процесів і співставлення їх із допустимими, які визначаються інтерпретацією управляючої моделі.

Отже, усі фази управління ГВС можна об'єднати єдиною інформаційною базою та системою програмного забезпечення. Сіткові управляючі моделі забезпечують гнучкість управляючої підсистеми, що реалізується адаптацією цієї єдиної бази даних до конкретних технологічних процесів і умов виробництва. Перехід до випуску нового виробу при такій організації управління супроводжується підготовкою відповідної управляючої моделі та стандартної структури даних.

На основі таких принципів організації управління бази даних і програмні засоби моделювання поєднуються в єдину систему, що підтримує аналіз, синтез і оптимізацію алгоритмів управління та інтерпретуючу управляючі моделі. Представляючи єдиний комплекс, така система оперативного управління забезпечує підвищення якості та скорочення термінів проектування і переналагодження системотехнічних дій управляючої підсистеми ГВС.

Склад і функції програмного комплексу типової СОУ ГВС. Комплекс програм типової СОУ ГВС призначений для синхронізації функціонування виробничого устаткування гнучких виробничих систем, досягнення максимально можливого коефіцієнта завантаження устаткування і забезпечення ефективного розподілу матеріальних і інформаційних потоків на стадії експлуатації ГВС. Програмне забезпечення реалізоване із врахуванням наступних вимог до принципів організації комплексу програм: єдність бази даних, розширюваність, модульність, сумісність знизу вверху, інваріантність.

Функції, реалізовані системою оперативного управління, визначаються складом устаткування, задачами управління цим устаткуванням і організацією управління ГВС.

Кожна функція, реалізація якої покладена на систему оперативного управління, задає перетворення:

$$f_i: X_i \rightarrow Y_i,$$

де f_i – найменування функції;

X_i – множина вихідних даних – аргументів функції;

Y_i – множина результатів застосування функції.

Таким чином, функція задає правило одержання результату на основі вихідних даних, причому вся множина функцій системи $F_i = \{f_i | i = 1, n\}$ визначає єдину базу даних, що характеризується

$$\text{множиною } D = \bigcup_{i=1}^n (X_i \cup Y_i).$$

З кожною функцією f_i у СОУ ГВС зв'язується програмний модуль, що представляє цю функцію у формі деякої задачі управління, реалізованої на ЕОМ.

У СОУ ГВС реалізуються наступні функції:

– створення і відновлення в інтерактивному режимі інформаційного стану системи (при повному запуску і після аварійного виходу із системи);

- розрахунок змінно-добового завдання і чергового запуску партій деталей у виробництво;
- оперативне корегування моделі інформаційного стану системи і планових завдань (за вказівками диспетчера ГВС);
- управління взаємодією устаткування (диспетчеризація) відповідно до отриманої управляючої моделі;
- контроль відхилень у процесі функціонування ГВС;
- формування і видача управляючих впливів виробничому устаткуванню;
- облік і формування довідкових даних про стан об'єкта управління і виконання планового завдання;
- тестування елементів комплексу програм СОУ ГВС;
- завершення роботи системи.

Ефективність застосування тієї чи іншої системи, як правило, обернено пропорційна складності її концептуального представлення, тому при проектуванні і розробці програмного забезпечення СОУ ГВС однією з основних задач було створення простих програмних засобів, що не вимагають від диспетчерів ГВС великого обсягу знань з програмування та обчислювальної техніки. Виходячи з вимог гнучкості програмного забезпечення СОУ ГВС побудований за блоково-модульним принципом відповідно до виконуваних функціональних завдань. Це дозволяє модифікувати модулі і розширювати склад комплексу при введенні нових функцій без зміни інших програм. Комплекс програм СОУ ГВС має трирівневу структуру, схема якої зображена на рис. 4.6.

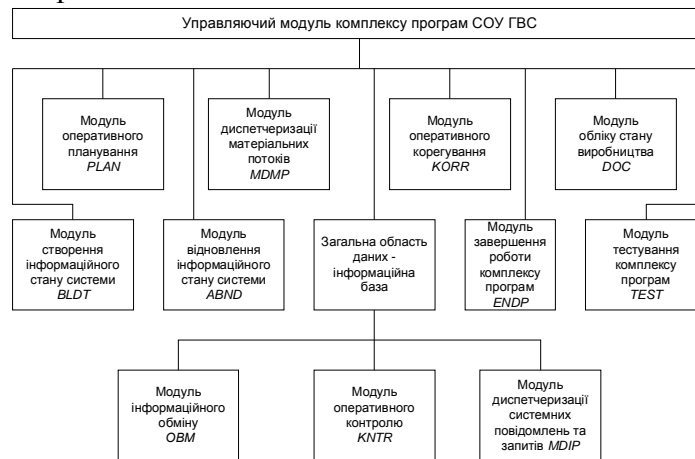


Рис. 4.6. Структура комплексу програм СОУ ГВС

На верхньому рівні комплексу знаходиться управляючий модуль, що здійснює діалог між диспетчером ГВС і програмним забезпеченням СОУ, а також зв'язок програмних модулів у системі. Управляючий модуль генерує режими роботи комплексу програм, управляє запуском і проходженням програмних модулів у системі.

Модулі середнього рівня реалізують прикладні цілі

функціонування СОУ ГВС і забезпечують:

- створення інформаційного стану системи, оперативного управління;
- оперативне планування;
- диспетчеризацію матеріальних потоків;
- оперативне корегування стану системи і ходу виробництва;
- облік стану виробництва;
- тестування комплексу програм за контрольними прикладами;
- відновлення інформаційного стану при аварійному виході із системи;
- завершення роботи комплексу програм.

Використання загальної області інформаційної бази забезпечує одночасний доступ до даних про стан ГВС декількох програмних модулів всіх рівнів комплексу і миттєву “реакцію” на зміни даних у цій базі.

Нижній рівень комплексу програм складають модулі, що перебувають у функціональному підпорядкуванні у модулів середнього рівня і запуск яких виконується одночасно із запуском визначених модулів середнього рівня. Модуль інформаційного обміну *OBM*, використовуючи загальну базу даних, передає інформацію між комплексом СОУ ГВС і управляючими пристроями виробничого устаткування. Модуль функціонує незалежно, однак його запуск здійснюється одночасно з запуском модуля диспетчеризації *MDMP* чи оперативного корегування *KORR*.

Модуль оперативного контролю *KNTR* функціонально підлеглий модулю диспетчеризації *MDMP*, оскільки переробляє поточну інформацію, що надходить з модуля інформаційного обміну *OBM*, про стан виконання запускених модулем диспетчеризації виробничих операцій і встановлює необхідність коректування ходу функціонування виробництва. Модуль запускається одночасно з модулем диспетчеризації матеріальних потоків *MDMP*, а в ході свого функціонування інформує модуль диспетчеризації про необхідні надалі дії по управлінню ГВС.

Форма роботи СОУ ГВС – діалогова з диспетчером ГВС. Вихідною інформацією для роботи системи є змінно-добові завдання у вигляді комплектації (завантаження) автоматизованого складу. Склад і функції програмних модулів СОУ ГВС наведені в табл. 4.1.

Таблиця 4.1

Характеристика модулів комплексу програм

Назва модуля	Функції модуля
<i>MAIN</i>	Ініціалізація комплексу програм і управління проходженням задач у системі
<i>BLDT</i>	Створення інформаційного стану системи управління ГВС

<i>PLAN</i>	Оперативне планування через побудову розкладу функціонування виробничого устаткування і управляючої моделі ГВС
<i>MDMP</i>	Диспетчеризація матеріальних потоків через обробку інформації, отриманої від виробничого устаткування, і формування управляючих впливів подальшого порядку роботи ГВС
<i>KNTR</i>	Оперативний контроль, діагностика й аналіз станів виробничого устаткування і ходу виробництва. Ухвалення рішення про порядок подальшої діяльності комплексу у фазі диспетчеризації. Викликається одночасно з модулем <i>MDMP</i> і інформує його про результати роботи
<i>KORR</i>	Оперативне корегування планового завдання та інформаційного стану системи. Викликається в міру необхідності за ініціативи модуля <i>KNTR</i>
<i>MDIP</i>	Диспетчеризація запитів і системних повідомлень для інформування диспетчера про виникаючі події та реалізацію диспетчерських повідомлень у системі. Викликається одночасно з модулем <i>MDMP</i> і управляє проходженням і реалізацією запитів і системних повідомлень
<i>DOC</i>	Облік стану виробництва і формування довідкової інформації здійснення діалогу з диспетчером і видача необхідної інформації на запит. Викликається в міру необхідності диспетчером
<i>ABND</i>	Відновлення інформаційного стану системи, перевірка файлів даних після чи аварії збою в системі
<i>OBM</i>	Міжрівневий обмін, зв'язок рівнів управління і передача інформації між комплексом програм і пристроями локального управління виробничого устаткування. Викликається одночасно з модулем <i>MDMP</i>
<i>TEST</i>	Тестування комплексу програм за контрольними прикладами
<i>ENDP</i>	Завершення роботи комплексу програм

4.4. Алгоритмічне забезпечення типової системи оперативного управління ГВС

Опис програмних модулів і структур даних комплексу програм СОУ ГВС. Розглянемо роботу модулів комплексу програм.

Модуль *BLDT* призначений для створення інформаційного стану СОУ ГВС. Він вводить вихідні дані в інформаційну базу.

Модуль реалізує два етапи своєї роботи:

- 1) введення і контроль вихідної інформації (у діалоговому режимі);
- 2) формування структур робочих масивів і запис їх у базу даних

(в автономному режимі).

На *першому етапі* роботи через інтерфейсну частину модуля вводяться поточні параметри ГВС, моделі стану устаткування, моделі стану складу і стану транспортних засобів і фіксується стан виробництва за минулу зміну. Моделі стану елементів ГВС відображають останню зміну стану в попередній період функціонування. В міру необхідності можна змінити стан модулів на початок наступного періоду роботи елементів ГВС.

На *другому етапі* роботи модуля здійснюється програмна переробка введеної інформації і формуються масиви даних про носії (касет) з деталями, виготовленими в поточну зміну, комплекти інструменту і комплектуючих деталей (при складанні), маршрутні карти технологічного процесу виготовлення виробів, змінно-добові завдання, стани транспортних засобів, стан складу і стан устаткування. У кінці етапу сформовані масиви записуються у відповідні блоки бази даних. Модуль *BLDT* через єдину інформаційну базу готує в системі необхідну інформацію для функціонування модулів *PLAN* і *MDMP*.

Модуль *PLAN* реалізує функцію оперативного планування у призначений для складання розкладів роботи модулів основного технологічного устаткування і побудови управляючої моделі ГВС виходячи з номенклатури та обсягу виробництва. Планування здійснюється на основі імітаційного моделювання функціонування ГВС сітковою моделлю процесу. Використання аналітико-імітаційного методу планування дозволяє враховувати компонування ГВС, технологічні маршрути партії деталей, їх трудомісткості, структуру транспортної сітки і витрати часу на переналагодження устаткування, що задаються структурою та параметрами сіткової моделі процесів, сформованої в модулі *BLDT*. Партії деталей з наявної множини претендентів на завантаження основного технологічного устаткування вибираються виходячи з набору правил переваги і стратегій транспортного обслуговування. При цьому з різних варіантів розкладів, побудованих у результаті імітаційного моделювання за кожним правилом, вибирається варіант, що задовольняє критерію мінімуму загальної тривалості виробничого циклу. Даний критерій забезпечує безупинне і рівномірне завантаження основного технологічного устаткування, мінімізацію загального терміну виготовлення і комплектування деталей, розміру і вартості незавершеного виробництва, а також можливих відхилень від встановленого плану за обсягом і номенклатурою.

Модуль працює циклічно в три етапи. *Перший етап* – підготовка вихідних даних для проведення імітаційного експерименту. При цьому за інформацією про процес функціонування й за вибраними для даного експерименту правилами переваги формується сіткова модель процесу управління ГВС, що враховує тимчасові параметри виконуваних операцій.

На *другому етапі* роботи модуля проводиться, власне,

планування і побудова розкладу роботи одиниць виробничого устаткування за допомогою трансляції сіткової управляючої моделі (УМ). При цьому моменти запуску/випуску деталей на деякій одиниці виробничого устаткування визначаються за модельним часом моменту $G_k^*(t_j)$ – k -го спрацьовування переходу сіткової моделі процесу управління ГВС, а розклад представляється як множина часових рядів $G_k^* = \{(G_k^*(t_j) / k \in \{1, 2, 3, \dots\}, j = 1, m\}$ моментів спрацьовувань переходів сітки. Конфлікти на переходах вирішуються за допомогою вибраного правила переваги на першому етапі.

На *третьому етапі* роботи модуля відбирається варіант плану з мінімальним часом виробничого циклу з усіх складених розкладів, що відповідають визначеним правилам переваги зі встановленого набору. Кожний знову побудований розклад порівнюється з тим, що вважався найкращим. При цьому час виробничого циклу побудованого розкладу визначається як максимум модельного часу моментів останніх спрацьовувань усіх переходів сіткової моделі процесу управління ГВС. За найкращим варіантом розкладу, з огляду на оптимальне правило переваги і зв'язок входів і виходів об'єкта управління з елементами сіткової моделі, будується сіткова управляюча модель ГВС. Модуль *PLAN* завершує свою роботу записом результатів оперативного планування (розклад роботи і сіткова управляюча модель) у визначені блоки інформаційної бази ГВС.

Модуль MDMP диспетчеризації матеріальних потоків призначений для обробки інформації про хід виробничого процесу, вибору чергових претендентів (носіїв з деталями) на обслуговування, формування і передачі управляючих впливів до нижнього рівня управління.

Модуль працює в два етапи:

- 1) опитування гнучких виробничих модулів, транспортних засобів, робота-штабелера автоматизованого складу і формування черги претендентів на завантаження виробничих засобів;
- 2) вибір і посилення чергового носія на обслуговування.

На *першому етапі* роботи модуля визначається поточний стан виробничого устаткування за станом M маркування сіткової управляючої моделі і вектора V сигналів зворотного зв'язку з об'єктом управління. Далі серед усіх справних і незайнятих одиниць устаткування визначається та, яка на поточний момент часу має найвищий пріоритет відповідно до вибраного в модулі *PLAN* правила переваги. У сітковій моделі ці дії пов'язані з пошуком збуджених переходів T_M і вибором того з них, котрий у даній ситуації може спрацювати.

На *другому етапі* роботи модуля формується новий стан управляючої моделі, що є наслідком спрацьовування вибраного на першому етапі збудженого переходу. У новому стані маркування визначається вектор управляючих впливів Y і видається завдання роботу-штабелеру автоматизованого складу чи автономному транспортному

засобу на транспортування носія, або гнучкому виробничому модулю на виконання необхідної виробничої операції. Потім робота модуля *MDMP* циклічно повторюється.

Алгоритм функціонування модуля *MDMP* зображений на рис. 4.7.

Модуль *KNTR* оперативного контролю призначений для диспетчерської діагностики ходу виконання розкладу роботи, обробки інформації диспетчера про хід виробничого процесу і прийняття рішення про порядок подальшої роботи. Контрольованими параметрами служать терміни запуску/випуску носіїв деталей відповідно до розкладу, побудованого в модулі *PLAN*, а також параметри поточного стану одиниць виробничого устаткування ГВС. Вихідними даними модуля є розклад роботи виробничого устаткування G^* , управляюча модель і локальні резерви часу виконання операцій виробничого процесу $\Delta_{\text{доп}, k}$. Відхиленнями в роботі ГВС від передбаченого розкладу вважаються збій (аварія) у роботі виробничої системи, відхилення Δ_k^j від планового моменту часу виконання виробничої операції. У тих випадках, коли зафіксоване відхилення, модуль *KNTR* діагностує ситуацію і приймає рішення про її ліквідацію наявними засобами в складі модуля корегування *KORR*. Контроль і діагностика ситуацій ведеться безупинно протягом усього періоду функціонування модуля *MDMP* (періоду управління в реальному часі).

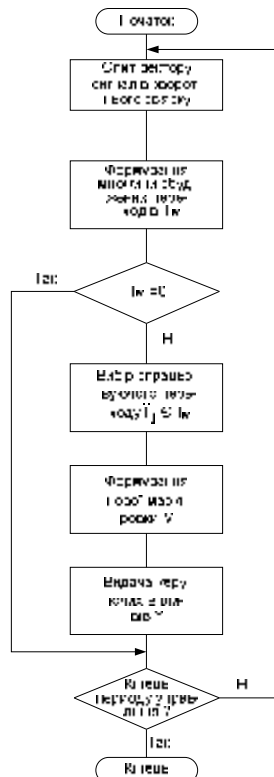


Рис. 4.7. Алгоритм диспетчеризації за управляючою моделлю

Алгоритм функціонування модуля *KNTR* показаний на рис. 4.8.

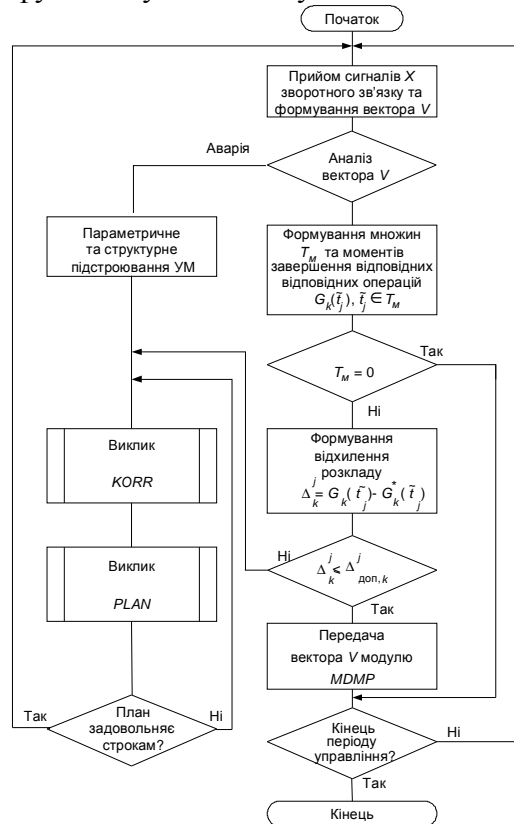


Рис. 4.8. Алгоритм оперативного контролю

Модуль KORR оперативного корегування призначений для відпрацювання аварійних ситуацій і відхилень ходу виробництва додаванням у систему ресурсів чи зміною планового завдання. Модуль коригує інформацію про стан виробничих потужностей і планове завдання у діалоговому режимі з диспетчером ГВС.

За своїми функціональними можливостями модуль *KORR* подібний модулю *BLDT*, оскільки в модулі оперативного корегування модифікуються параметри і структура управляючої моделі на підставі введених диспетчером даних. Однак модуль *KORR* може викликатися за необхідності диспетчером і виконувати додаткову функцію включення фонових завдань виробничої підсистеми. Використовуючи локальні резерви, диспетчер ГВС може включити додаткові замовлення на обслуговування фонових робіт, виконання транспортних операцій і операцій завантаження складу через дану функцію модуля *KORR*. Тому модуль оперативного корегування має дві точки входу: з модулів *MAIN* і *KNTR*. Робота модуля завершується передачею управління модулю *MAIN* чи модулю *PLAN*.

Алгоритм функціонування модуля *KORR* зображений на рис. 4.9.

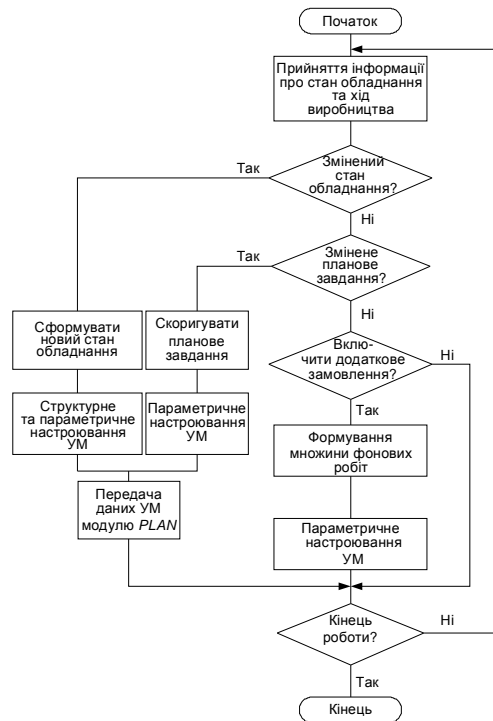


Рис. 4.9. Алгоритм функціонування модуля оперативного корегування

Модуль MDIP призначений для обробки запитів диспетчера ГВС і формування диспетчеру системних повідомлень. При роботі з комплексом програм у період управління в реальному часі диспетчеру ГВС подаються різні повідомлення, у відповідь на які він повинен виконати визначені дії. Усі повідомлення формуються в результаті інтерпретації стану управляючої моделі ГВС мовою, зрозумілою диспетчеру.

Повідомлення, що видаються в ході диспетчеризації, поділяються на запитальні та синхронізуючі.

Синхронізуючі повідомлення інформують оператора про стан виробничого устаткування, предметів виробництва і про важливі з погляду контролю події, що не ідентифікуються інформаційними засобами на об'єкті управління.

За допомогою *запитальних повідомлень* модуль MDIP запитує необхідну інформацію для роботи модулів MDMP, KNTR, KORR.

Модуль запускається в роботу одночасно з модулями MDMP, KNTR і взаємодіє з ними за допомогою посилення повідомлень у ході диспетчеризації матеріальних потоків ГВС.

Алгоритм функціонування модуля MDMP показаний на рис 4.10.

Модуль DOC призначений для обліку стану виробництва і формування довідкової інформації диспетчеру ГВС. Він представляє собою діалоговий комплекс програм видачі необхідних інформаційних

повідомлень.

Облік стану виробництва ведеться за показниками, серед яких можна виділити наступні: змінно-добове завдання, виконання змінно-добового завдання, простої устаткування через транспортування, переналагодження та аварії, сумарні простої, модель стану складу, модель стану виробничого устаткування, стан оперативних нагромаджувачів, незавершене виробництво на поточну добу, оперативний розклад запуску/випуску деталей, інформація про носії з деталями, що виготовляються в поточну зміну.

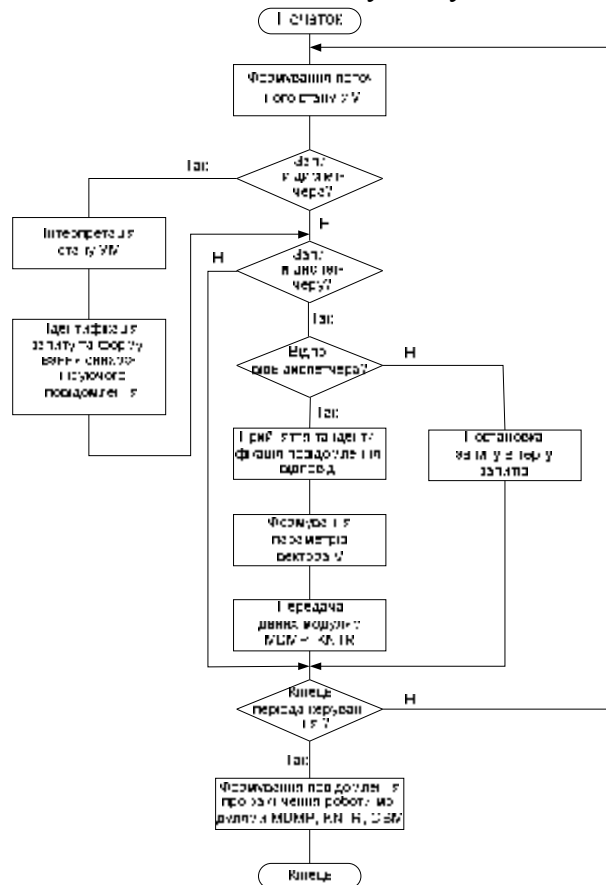


Рис. 4.10. Алгоритм функціонування модуля диспетчизації запитів і системних повідомлень

Модуль запускається в міру необхідності диспетчером ГВС незалежно від задіяних в конкретний момент модулів комплексу.

Модуль ABND призначений для відтворення інформаційного стану системи при аварійному виході модулів з комплексу програм чи збоях устаткування, у тому числі й ЕОМ.

Модуль виконує наступні функції: встановлення наявності та можливості доступу до даних файлів системи; визначення можливої точки входу для продовження роботи системи.

Робота модуля **ABND** ініціюється вказівкою режиму роботи

комплексу програм у модулі *MAIN* і здійснюється в автономному режимі з видачею діагностичних повідомлень і рекомендацій диспетчеру ГВС по продовженню роботи з комплексом програм.

У випадку аварійного виходу із системи необхідно запустити модуль *ABND* і привести устаткування ГВС у вихідний стан.

Модуль *TEST* призначений для перевірки працездатності модулів комплексу СОУ ГВС.

Тестування проводиться відповідно до встановлених в системі контрольних прикладів, що задаються визначеними програмами (протоколами роботи) системи, вибраними диспетчером ГВС за своїм розсудом. Потім робота здійснюється, як і в нормальному режимі функціонування СОУ, але над тестовою інформаційною базою даних.

Модуль *OBM* призначений для обміну інформацією комплексу програм з нижчим рівнем управління виробничими модулями і є модулем міжрівневого обміну даними каналами зв'язку.

Модуль *OBM* виконує наступні функції: передачу повідомлень каналами зв'язку; чекання підтвердження прийому переданого повідомлення; прийом символів, що підтверджують зміну стану ГВС; прийом управляючих повідомлень.

Даний модуль слугує для передачі даних (команд управління) транспортним автономним засобам і роботам-штабелерам автоматизованого складу, а також управляючих програм для пристроїв ЧПУ гнучких виробничих модулів.

Модулі обміну, що функціонують на нижчому рівні в складі програмних модулів управління *GPM*, *AS*, *ATM*, виконують наступні функції: прийом повідомлень (управляючих команд); передача інформаційних повідомлень. Ці модулі, здійснюючи зв'язок з верхнім рівнем і приймаючи командні вказівки, реалізують власне управління одиницями виробничого устаткування ГВС.

Управління організується на основі сіткових моделей алгоритмів управління гнучкими виробничими модулями, роботами-штабелерами, автоматизованими складами і електророботами, отриманими у результаті декомпозиції управляючої моделі ГВС на алгоритми нижчих рівнів управління.

Повідомлення, передані від ЕОМ верхнього рівня до пристроїв ЧПУ, поділяються на інформаційні та управляючі. Від пристрою ЧПУ до верхнього рівня передаються тільки інформаційні повідомлення. Для ЕОМ верхнього рівня управління встановлені наступні види повідомлень процедури обміну: інформаційні (*D* – дані, *U* – вказівки); управляючі (*S* – ініціалізація обміну, *E* – кінець повідомлення). Для пристрою ЧПУ нижнього рівня управління встановлені тільки інформаційні повідомлення наступних видів: *N* – заперечення прийому; *C* – підтвердження прийому; *F* – кінець виконання операції на устаткуванні; *A* – аварія на устаткуванні. Комплекс модулів міжрівневого обміну на основі наведених повідомлень реалізує єдину мову обміну, тобто деякий

формат ідентифікації тексту повідомлень, що надходять по каналах зв'язку між ЕОМ верхнього рівня і пристроями ЧПУ нижнього рівня ГВС.

Інтерфейс комплексу і подання даних у пакеті програм СОУ ГВС. В основі програмного забезпечення СОУ ГВС лежить управляючий модуль *MAIN*, що ініціює початок функціонування комплексу програм, управляє порядком проходження програмних модулів і виконує діалог між диспетчером і програмним забезпеченням СОУ ГВС (рис. 4.11).

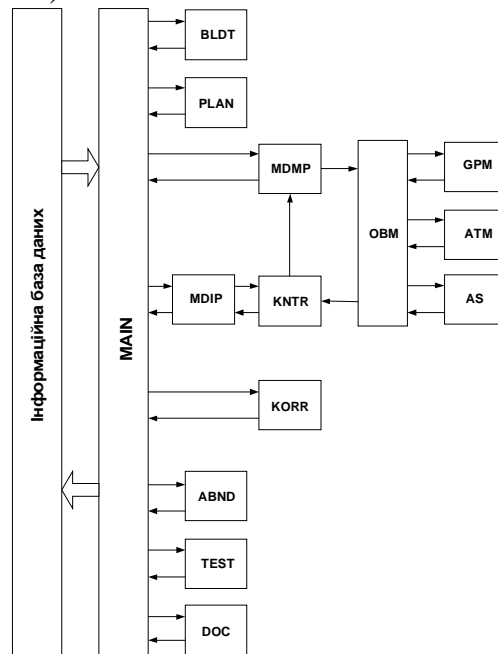


Рис. 4.11. Схема взаємодії програмних модулів комплексу програм СОУ ГВС

Диспетчер ГВС у діалоговій формі вводить у вихідні дані для оперативного планування і дані про початковий стан устаткування, тобто створює інформаційний стан СОУ, використовуючи для цього модуль *BLDT*. Потім диспетчер передає управління модулю оперативного планування *PLAN*, що складає розклад і підготовляє дані до процесу диспетчеризації. Після завершення планування запускається модуль диспетчеризації матеріальних потоків ГВС, що через модуль міжрівневого обміну *OBM* власне і здійснює управління транспортом (*ATM*), складом (*AS*), технологічним устаткуванням (*GPM*). Коли на об'єкті управління відбувається яка-небудь подія, аналізується ситуація, що створилася (*KNTR*), і в залежності від результатів аналізу управління повертається до *MDMP* чи передається модулю *KORR*. Якщо в результаті аналізу виявлене порушення нормального функціонування виробництва, то в діалоговому режимі корегується оперативна інформація і формуються вихідні дані для оперативного редагування розкладу функціонування ГВС (за допомогою модуля *PLAN*) на період роботи, що залишився.

Диспетчер управляє ходом функціонування системи в період диспетчеризації через модуль *MDIP* і може одержувати довідкову інформацію про стан виробництва в будь-який момент часу через модуль *DOC*.

Оскільки комплекс програм побудований за модульним принципом, то процес функціонування програмного забезпечення представляється у вигляді сукупності підпроцесів взаємодії диспетчера ГВС із кожним модулем окремо.

Реалізація функцій, покладених на СОУ ГВС, вимагає передачі, прийому, збереження та перетворення великого обсягу інформації. Виходячи з модульного принципу побудови комплексу програм з організацією управління на основі управляючої моделі ГВС, усі вхідні та вихідні дані програмних модулів організовані у вигляді блоків єдиної інформаційної бази даних ЕОМ верхнього рівня. Дані блоки поділяються на чотири групи: основні, службові, допоміжні, інформаційні.

До групи *основних* відносяться блоки, що відповідають за стан одиниць виробничого устаткування ГВС, матеріальних та інформаційних потоків, тобто описують управляючу модель ГВС.

До групи *службових* відносяться блоки, що містять маршрутні карти технологічного процесу виготовлення всіх допустимих для даної ГВС типів деталей, блоки управляючих програм технологічного устаткування. Наявність службових блоків у системі обов'язкова для початку функціонування комплексу програм.

До групи *допоміжних* відносяться блоки, що створюються і використовуються під час роботи комплексу програм для збереження і передачі проміжних результатів і зникають по закінченні роботи.

До групи *інформаційних* відносяться блоки, що відповідають за виробничу діяльність ГВС і призначені для інформування диспетчера ГВС.

Отже, програмне забезпечення СОУ ГВС, спроектоване із врахуванням ряду вимог, виконання яких дозволило уніфікувати програмні засоби управління, забезпечує їх незалежність від змісту предметної області виробничої системи. Досягнута уніфікація програмного забезпечення дозволяє тиражувати і супроводжувати його у виробничих системах без принципових змін в організації програмних засобів управління шляхом реорганізації єдиної інформаційної бази даних.

Контрольні запитання

1. Розкрийте складові сіткової моделі виробничого процесу ГВС.
2. Поясніть зміст формалізованого подання і процедури вирішення конфліктних ситуацій, що утворюються в ході виконання виробничого процесу під час оперативно-диспетчерського управління

ГВС.

3. Наведіть відмінності сіткової моделі управляючого процесу від сіткової моделі алгоритму управління.

4. Сформулюйте етапи методу синтеза сіткової моделі алгоритмів управління ГВС.

5. Розкрийте задачі і особливості побудови програмних засобів системи оперативного управління ГВС.

6. Визначте базові модулі системи оперативного управління ГВС

7. Поясніть порядок застосування програмних модулів і структур даних системи оперативного управління ГВС.

«Ніколи не починай складних ходів, якщо того ж можна досягти набагато більш простими способами» (Ремарк Еріх Марія)

Розділ 6

АЛГОРИТМІЗАЦІЯ ЗАДАЧ ПРОГРАМНОГО УПРАВЛІННЯ В СИСТЕМАХ ВИКОНАВЧОГО РІВНЯ ГВС

6.1. Структура реалізації геометричної задачі ЧПУ в технологічному устаткуванні ГВС

Геометрична задача ЧПУ, як перша й основна функція програмного управління устаткуванням, включає кілька фаз реалізації процесу ЧПУ і може бути визначена наступним чином: відобразити геометричну інформацію креслення в сукупність таких формоутворюючих рухів робочого органу (РО) технологічного обладнання верстата, що матеріалізують креслення в кінцевому виробі.

Перша фаза цього інформаційного перетворення в процесі ЧПУ полягає у підготовці (програмуванні) та введенні управляючої програми ЧПУ, а також у виконанні підготовчих розрахунків у машинному масштабі часу.

У сучасних пристроях ЧПУ обов'язково передбачена можливість ручного введення та редагування управляючих програм з панелі оператора і збереження архіву управляючих програм у пам'яті.

Існує тенденція використання в рамках першої фази процесу ЧПУ елементів автоматизованої розробки управляючих програм із залученням елементів діалогу і техніки меню.

Другу фазу в процесі ЧПУ складають обчислення, виконувані в реальному часі. Сюди відносяться: інтерполяція (складність якої обумовлена формою траєкторії, що інтерполюється, і розмірністю інтерпольованого простору); розрахунки розгонів і гальмувань; стикування кадрів управляючої програми; аналіз робочого простору; корекція подачі; аналіз і логічна обробка інформації про стан циклових механізмів верстата тощо.

Обчислення супроводжуються підготовкою даних для візуалізації всієї оперативної інформації.

Обчислення другої фази, як правило, вимагають складних і точних розрахунків, що повторюються з високою частотою.

Третю фазу в процесі ЧПУ складає управління приводами, що здійснюється в реальному часі. Найбільшу частку займають тут розрахунки, пов'язані із замкненням контурів слідкуючих приводів подачі та корекцією систематичних похибок гвинтових пар верстата. Трудомісткість розрахунків прямопропорційна числу керованих координат, загальне число яких досягає іноді десяти, дванадцяти. Інша

частина роботи, виконуваної в межах третьої фази, відноситься до керування численними приводами циклових механізмів верстата і до прийому сигналів, що визначають стани (положення) зазначених приводів.

Нижче докладно викладається суть кожної фази реалізації геометричної задачі ЧПУ.

6.2. Технологія програмування формоутворення деталей та руху робочого органу технологічного устаткування ГВС

6.2.1. Координатні системи верстатів з ЧПУ

Виготовлення продукції на верстатах з ЧПУ багато в чому ґрунтується на перевірених практикою надійних типових технологічних рішеннях, що застосовуються на універсальному устаткуванні, яке обслуговується робітником-верстатником. Ці рішення закладаються в основу технологічних управляючих програм (УП) для верстатів з ЧПУ при ручній і автоматизованій підготовці УП. Але з'являються і нові технічні рішення, обумовлені необхідністю автоматичного підведення інструмента до контуру обробки, відведення інструмента з робочої зони при зміні заготовок, зміні інструмента, базування заготовки та іншими задачами, які раніше виконував робітник-верстатник, використовуючи отримані при навчанні навички.

Основою для складання технологічної УП є інформація про заготовку і деталь, яка традиційно представляється графічно на папері або дисплеї системи автоматизованого проектування у вигляді креслення. Всі основні елементи деталей можна поділити за способом їх обробки (фрезерна, токарна і свердлильна), серед яких можна виділити наступні елементи поверхні, що потребують багатопрохідної обробки: поверхні, що виходять проходом фрези вздовж контуру; тіла обертання ступінчастої та криволінійної форм; отвори, отримані переміщенням інструментів, що обертаються навколо своєї осі.

Для обробки кожного елемента деталі існують типові схеми обробки. Основними інструментами при обробці є фрези, різці та свердла або їм подібні інструменти.

Кромка різального інструмента не завжди має просторову форму. Тому для правильного переміщення інструмента вздовж контуру обробки необхідно правильно задати траєкторію інструмента, а точніше – траєкторію розрахункової точки інструмента, яка в більшості випадків не співпадає з контуром деталі. Положення точки інструмента, що обирається як розрахункове, залежить від форми інструмента (рис. 6.1) і траєкторії контуру обробки.

Однією із задач програміста або системи автоматизованої підготовки програм, що розробляють УП для пристрою ЧПУ, є задача проектування траєкторії розрахункової точки інструмента відносно контура деталі, представленної на кресленні. Але при виготовленні деталі остання встановлюється нерухомо на станину або рухомо в РО

верстата, а інструмент закріплюється в державку і разом з державкою встановлюється в інший робочий орган верстата. Задача розробника УП – так розрахувати траєкторії робочих органів верстата, щоб розрахункова точка інструмента рухалась по заздалегідь визначеній траєкторії.

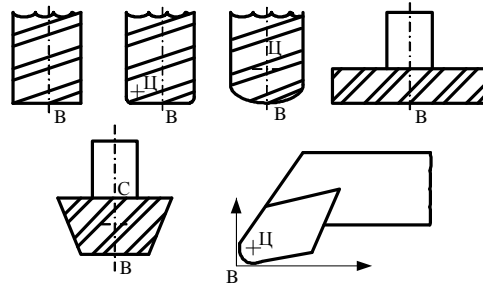


Рис. 6.1. Інструмент і розрахункові точки:
B, C, Ц – розрахункові точки інструмента

Для спрощення розрахунків геометричних переміщень використовують системи координат деталі, інструмента і верстата.

Система координат деталі служить для розрахунку координат опорних точок траєкторії розрахункової точки інструмента.

Опорними називаються точки початку, кінця, перетину або дотику геометричних елементів, з яких утворені контури деталі та траєкторії інструментів, найчастіше це – прямі лінії та дуги кіл.

За систему координат деталі приймається права прямокутна система. Іноді застосовують циліндричну або сферичну системи координат. При виборі початку системи координат деталі та орієнтації її осей необхідно звернути увагу на забезпечення простоти розрахунків опорних точок траєкторій.

Система координат інструмента призначена для задання положення його різальної частини відносно державки. Інструмент описується в робочому положенні в зборі з державкою.

При описі всієї різноманітності інструментів для верстатів з ЧПУ зручно використовувати єдину систему координат інструмента $X_I Z_I$, осі якої паралельні осям стандартної системи координат верстата і направлені в ту ж сторону, а початок системи координат знаходяться в базовій точці I інструментального блока (рис. 6.2), що вибирається із врахуванням особливості його встановлення на верстаті.

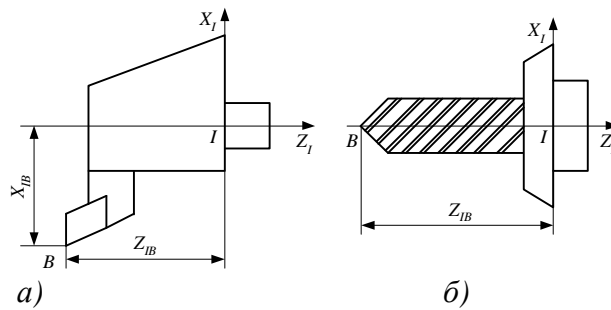


Рис. 6.2. Система координат інструмента:

а – державка з різцем; б – державка зі свердлом

Різальна частина інструмента характеризується положенням його вершини і різальних кромek. Вершина інструмента задається радіусом заокруглення і координатами X_{IB} і Z_{IB} її настроювальної точки B , положення якої відносно системи координат інструментального блока забезпечується настроюванням інструментального блока поза верстатом на спеціальному пристосуванні.

Настроювальна точка B різця зазвичай використовується як розрахункова при обчисленні траєкторії інструмента, елементи якої паралельні координатним осям. Розрахунковою точкою криволінійної траєкторії різця зазвичай служить центр заокруглення C при вершині інструмента (рис. 6.1).

Система координат верстата. Осі координат і напрямів руху є головною розрахунковою системою, в якій визначаються граничні переміщення, початкові та поточні переміщення РО.

Положення РО верстата характеризують їх базові точки.

Базовими точками є:

- 1) для шпиндельного вузла – точка перетину торця шпинделя з віссю його обертання;
- 2) для супорта токарно-револьверного верстата – центр повороту різцетримача в площині, паралельній до напрямних супорта, що проходять через вісь обертання шпинделя;
- 3) для хрестового стола – точка перетину його діагоналей або спеціальна настроювальна точка, що визначається конструкцією пристосування;
- 4) для поворотного стола – центр повороту на дзеркалі стола.

Систему координат верстата, обрану у відповідності з рекомендаціями комітету ISO, називають стандартною. Вона представляє собою праву прямокутну декартову систему координат, зв'язану із заготовкою. Осі цієї системи паралельні прямолінійним напрямним верстата. Позначення осей координат і напрямку руху встановлюється так, щоб програмування операцій обробки не залежало від того, чи переміщується інструмент або заготовка.

Орієнтація осей стандартної системи координат верстата зв'язується з напрямом руху при свердлінні на свердильних,

розточувальних, фрезерних і токарних верстатах. Напрямок виходу свердла із заготовки прийнятий як додатний для осі Z . Вісь X перпендикулярна до осі Z і паралельна площині встановлення заготовки. Якщо такому визначенню відповідають дві осі, то за вісь X приймають ту, за якою можливе найбільше переміщення вузла верстата. При встановлених осях Z і X однозначно визначається напрям осі Y .

Початок стандартної системи координат верстата вибирається довільно, але звичайно поєднується з базовою точкою вузла, в якому закріплена заготовка і зафіксованого в такому положенні, при якому всі переміщення РО верстата могли б описуватися додатними координатами.

Рухи РО верстата задаються в УП координатами базових точок в системі координатних осей, визначених в стандартній системі координат. Додатному напрямку руху РО верстата відповідає напрямок відведення інструмента від заготовки.

Для визначення спільності підготовки програм Рекомендації комітету ISO регламентують позначення та напрями осей координат робочих органів верстата.

Координатні осі робочих органів верстата, що несуть інструмент, позначаються буквами X , Y , Z без штрихів. Якщо на верстаті є декілька РО для переміщення інструментів, то вказані позначення відносяться до первинних осей. Для позначення вторинних осей, паралельних до первинних, використовуються літери U , V , W , а для третинних – P , Q і R . Координатні осі РО верстата, що переміщує заготовку, направлені протилежно до відповідних осей РО, що переміщують інструмент, і позначаються тими ж літерами, але зі штрихами.

Поворот РО з інструментом навколо X , Y , Z позначається літерами A , B , C , а навколо осей X' , Y' , Z' – A' , B' і C' відповідно. Початок відліку кутів повороту і додатні напрями обертання, аналогічно до початку системи координат, визначаються довільно, як показано на рис. 6.3. Для позначення кутових рухів навколо спеціальних осей використовуються літери D і E .

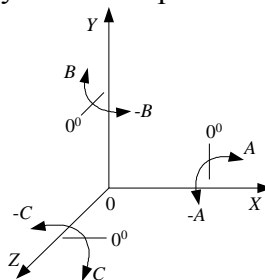


Рис. 6.3. Осі та напрямки обертання у стандартній системі координат

В пристрої ЧПУ напрям координатних осей РО верстата відображається так, що якщо для реалізації запрограмованого руху

інструмента відносно заготовки необхідно перемістити РО з інструментом, то цей рух виконується із заданим в УП знаком, а якщо необхідно перемістити РО із заготовкою, то знак напрямку руху зміниться на протилежний.

При програмуванні та налагодженні верстата для роботи за УП використовують деякі *характерні точки*, які називаються *нульовою*, *вихідною* та *фіксованою* точками верстата.

Нульова точка верстата (нуль верстата) – точка, прийнята за початок системи координат верстата. Вона визначається відносно конструктивних елементів верстата. Відносно неї задаються в УП абсолютні розміри переміщень РО верстата, якщо початок відліку переміщень не зміщений за допомогою засобів “плаваючого нуля”.

“Плаваючий нуль” – це властивість пристрою ЧПУ поміщати (зрушувати) початок відліку переміщень РО в будь-яке положення відносно нульової точки верстата.

Система координат зі зрушеним початком відліку називається робочою.

Вихідна точка верстата – точка, яка визначається відносно нульової точки верстата і використовується для початку роботи за УП.

З вихідними точками верстата поєднують базові точки робочих органів перед початком роботи верстата за УП.

Вихідні точки вибирають з умов мінімізації допоміжних ходів, запезпечення безпеки обслуговуючого персоналу при зміні інструмента і зручності закріплення інструмента на верстаті.

Фіксована точка верстата – точка, яка визначається відносно нульової точки верстата і використовується для визначення положення РО верстата.

Базові точки РО поєднуються з фіксованими точками верстата за допомогою вмонтованих в кути верстата спеціальних датчиків положення. В цю точку РО приводиться в два етапи: спочатку РО переводиться в зону дії датчика по команді з пульта пристрою ЧПУ або по заданому в УП переміщенню, а потім – з високою точністю безпосередньо у фіксовану точку верстата, що дозволяє виключити раніше накопичену похибку положення РО.

Сучасні верстати з пристроєм ЧПУ точно відпрацьовують задані в УП переміщення. Тому фіксовані точки верстата використовують тільки після переналагодження верстата перед виготовленням першої деталі партії, а для наступних деталей партії знаходження робочих систем координат обирають, за можливістю, ближче до заготовки.

Зв'язок систем координат відбувається через базові точки робочих органів верстата, що несуть заготовку та інструмент. При закріпленні заготовки на верстаті технологічна база для обробки деталі в даному положенні встановлення поєднується з відповідною опорною поверхнею пристосування. Для того щоб зв'язати між собою системи

координат деталі та верстата, необхідно задати в цих системах базову точку B (рис. 6.4), що визначає положення з'єднаних поверхонь.

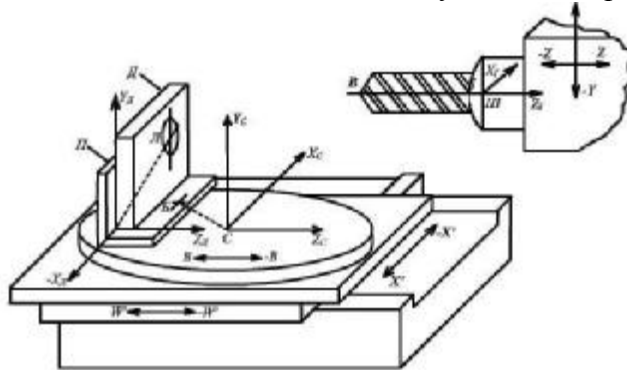


Рис. 6.4. Зв'язок систем координат деталі, верстата та інструмента

Системи координат зв'язані через базові точки РО верстата і кріпильного пристосування. Початок системи координат верстата прийнято в базовій точці C хрестового поворотного стола в крайньому лівому і дальньому положенні, якщо дивитись на нього з боку шпиндельного вузла. Настроювальна точка інструмента B , що задана в системі координат інструмента $X_I Z_I$, переводиться в систему координат верстата через базову точку шпиндельного вузла III , з якою суміщений початок I системи координат інструмента. Поточна точка траєкторії інструмента A переводиться програмістом-технологом або системою автоматизованої підготовки УП з системи координат деталі $D X_D Y_D Z_D$ в систему координат верстата через базову точку B кріпильного пристосування II .

Такий зв'язок систем координат деталі, верстата та інструмента дозволяє витримати задану точність при перевстановленнях заготовки і врахувати діапазон переміщень РО верстата при розрахунку траєкторій інструмента в процесі підготовки УП.

6.2.2. Подання формоутворення деталей у пристрої ЧПУ

Формоутворючі рухи часто відтворюються слідкуючими приводами подачі верстата. Слідкуючі приводи подачі розташовані так, щоб вони відповідали координатній системі верстата, в якій здійснюється у процесі обробки керований відносний рух інструмента та заготовки.

Визначимо склад та обсяг інформації, яку необхідно переробити пристроєм ЧПУ, на прикладі контурної обробки плоскої деталі. На вертикально-фрезерному верстаті з ЧПУ у декартових координатах обробляють плоске фасонне вікно кінцевою фрезою (рис. 6.5, а). Сукупними рухами по осях X та Y приводів подачі стола та салазок верстата необхідно забезпечити складний відносний рух заготовки та інструмента (рис. 6.5, б). При цьому після вибору напрямку обходу вікна

потрібно буде запрограмувати не відносний рух по контуру, заданому кресленням деталі (траєкторія 1, 2, ..., 14, 1), а рух по еквідистанті заданого контуру (траєкторія 1', 2', ..., 14', 1').

Еквідистанта – це траєкторія руху кінцевого елемента робочого органу (кінцевої фрези, токарного різця), яка віддалена від заданого контуру деталі на радіус його кінцевого елемента (фрези або різальної кромки різця).

Для опису еквідистанти траєкторію руху поділяють на елементарні ділянки, серед яких здебільшого виділяють прямі лінії та кола. Точки, які розташовані на стику елементарних ділянок, називають *опорними*.

Поділимо траєкторію 1', 2', ..., 14', 1' на елементарні ділянки, які можна описати одноманітними рівняннями прямої лінії або кола. У даному прикладі таких ділянок буде десять: 1'2'–2'3'4'–4'5'–5'6'7'–7'8'–8'9'–9'10'11'–11'12'–12'13'–13'14' (на рис. 6.5 точки 9'–11' не позначені), де точки 1', 2', 4', 5', 7'–9', 11'–13' – опорні.

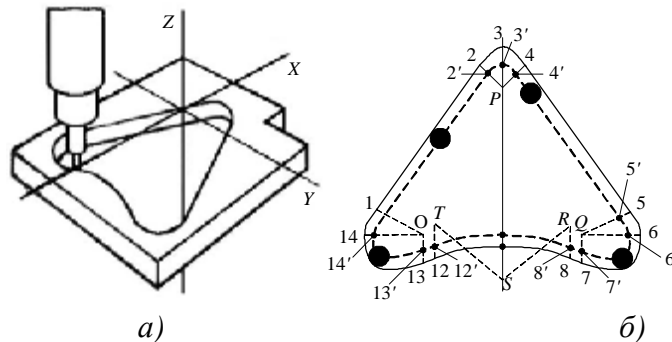


Рис. 6.5. Контурна обробка плоскої деталі:

а – приклад обробки фасонного вікна кінцевою фрезою;

б – складний відносний рух заготовки та інструмента

Отже, загальна стратегія управління формоутворенням деталі може бути подана наступним чином. Комбінацією встановлюючих переміщень у всіх трьох координатах (стола, салазок, шпиндельної бабки) сумістити вісь фрези з початковою опорною точкою 1' у площині обробки (підготовча стадія). Послідовними робочими переміщенням по елементарних ділянках 1'2', 2'3'4', 4'5' і т.д. обробити поверхню контуру (стадія обробки). Комбінацією встановлюючих переміщень у всіх трьох координатах відвести фрезу з початкової опорної точки 1' у вихідне положення (завершальна стадія). Таким чином, всі переміщення інструмента при формоутворенні деталі є встановлюючими або робочими.

Загалом, у верстатах з ЧПУ *встановлююче переміщення* виконується на максимальній швидкості без контакту інструмента з деталлю, до якого висувають тільки вимоги точності позиціонування у кінцевому положенні. *Робоче переміщення* пов'язане з обробкою деталі і контактом інструмента з поверхнею заготовки. Тому до нього

висувають вимоги щодо точності додержання траєкторії руху та заданої швидкості обробки.

Необхідно зазначити, що стадія обробки в залежності від складності оброблювального контуру може складатися як з робочих, так і з встановлюючих переміщень.

Отже, алгоритм управляючої програми формоутворення деталі буде складатися із фрагментів (кадрів), кожен з яких містить опис чергової елементарної ділянки. Опис цих елементарних ділянок здійснюється переважно за рахунок подання геометричної інформації про траєкторію руху.

Кадр відпрацювання руху по прямій лінії буде містити у кодованому вигляді наступні дані: позначення кадру у вигляді номера, ознаку руху по прямій, значення переміщень по осях X та Y при русі від початкової опорної точки до кінцевої опорної точки елементарної ділянки, швидкість подачі інструмента на елементарній ділянці.

Кадр відпрацювання руху по колу буде містити у кодованому вигляді такі дані: позначення кадру, ознаку руху по колу за або проти годинникової стрілки, значення переміщень по осях X та Y при русі від початкової опорної точки до кінцевої опорної точки елементарної ділянки, місцеві (локальні) координати початкової опорної точки елементарної ділянки відносно центра кола, що створює визначеність у заданні кола приростами переміщень по осях X та Y , швидкість подачі інструмента на елементарній ділянці.

Покажемо, як приклад, опис геометрії початкових елементарних ділянок 1'2' і 2'3'4' з рис. 6.5. Для цієї мети задамо конкретні чисельні значення параметрів відрізка загальної траєкторії (рис. 6.6).

Кодований опис елементарних ділянок у вигляді кадрів управляючої програми буде мати вигляд:

N001 G01 X375. Y160. F1200.

N002 G02 X0. Y183.96 I39.24 J-91.98.

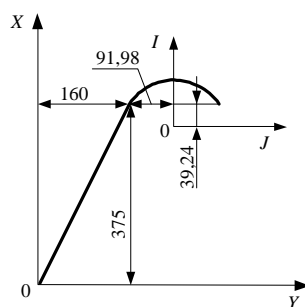


Рис. 6.6. Геометричне подання елементарних ділянок

Кадр відкривається символом-адресою N , який позначає номер кадру, а завершується символом “Перехід на наступний рядок” (не відображається при кодуванні), який виконує функцію роздільника “Кінець кадру”.

Основний зміст наведених у прикладі кадрів складає

послідовність слів. Слова $G01$ та $G02$ називають підготовчими функціями. Вони вказують відповідно на рух по прямій лінії у першому кадрі та по дузі кола за напрямом годинникової стрілки у другому кадрі. Слова з адресами X , Y – розмірні переміщення, які представляють собою відносні кадрові прирости по відповідних координатних осях, що задані з десятковою комою (у коді ISO-7bit замість десяткової коми ставлять крапку). Слова з адресами I , J вказують положення початкової точки дуги відносно її центра у локальній системі координат. Для всіх слів із геометричною інформацією знак плюс скасований і вважається присутнім. Слово із адресою F визначає швидкість подачі у мм/хв, а його відсутність у другому кадрі свідчить про те, що значення подачі не змінюється.

Розглянемо послідовність роботи пристрою ЧПУ при відпрацюванні послідовності елементарних ділянок при формоутворенні деталі на прикладі прямокутної позиційної обробки на токарному верстаті (рис. 6.7). Прямокутною її називають тому, що вона послідовно використовує окремі слідкуючі приводи подачі у декартовій системі координат. Нехай обробляють шків шістьма послідовно змінними інструментами, закріпленими у револьверній головці.

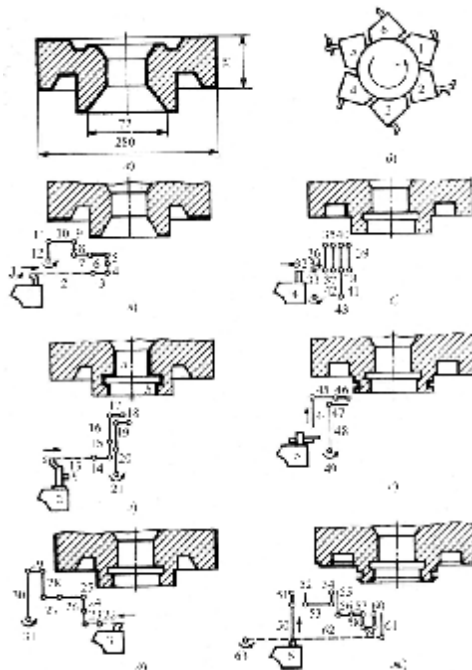


Рис. 6.7. Геометричне подання багатоінструментальної прямокутної обробки

Цикл автоматичного управління за програмою починається саме з початкової опорної точки першої ділянки. Відпрацювання першого кадру почнеться з моменту його активізації, коли він стане *робочим*.

Відпрацювання полягає у розрахунках проміжних відносних

положень інструмента і заготовки від початкової опорної точки до кінцевої опорної точки поточної ділянки, а також в управлінні рухом у відповідності до розрахунків. В результаті руху за програмою першого кадру вісь фрези суміститься з кінцевою опорною точкою, яка відкриває другу елементарну ділянку. Відпрацювання другого кадру управляючої програми завершиться суміщенням осі фрези у наступній кінцевій опорній точці.

Послідовна активізація кадрів управляючої програми, послідовна обробка інформації, яка у них знаходиться, що супроводжується видачею оперативних команд на виконавчі приводи верстата, призводять до послідовного обходу запрограмованого контуру від однієї опорної точки до наступної. Для запобігання призупинення подачі на стикові двох елементарних ділянок, тобто у момент активізації наступного кадру, цей наступний кадр повинен бути заздалегідь підготовлений до оперативних розрахунків і відпрацювання. У такому підготовленому вигляді кадр називають *буферним*.

Отже, управляюча програма буде мати у своєму складі шість блоків, які містять групу кадрів, що передбачають використання у рамках групи якогось одного інструмента. Блок відкривається кадром, у якому закодовано пересування потрібного інструмента у початкове перед наступною операцією положення. Далі йдуть кадри, у кожному з яких закодовано переміщення між сусідніми опорними точками.

Опорні точки розставляють, по-перше, у тих місцях, де відбувається зміна керованих координат, і, по-друге, там, де примусово змінюється швидкість подачі. Відпрацювання блоку завершується виведенням револьверної головки у початкове положення, у якому відбувається зміна інструмента.

Для того щоб окремі блоки стали автоматично “зшитими”, потрібне точне визначення положень робочих кромek усіх інструментів і автоматична корекція переміщень із врахуванням вказаних станів.

Специфіка прямокутної обробки не змінює у пристрої ЧПУ у порівнянні з контурною обробкою формального змісту процесу переробки управляючої інформації: введення, розрахунки, пов’язані з визначення проміжних положень інструмента між опорними точками, видача оперативних управляючих команд до слідкуючих приводів подачі верстата. Тому, алгоритм роботи пристрою ЧПУ в автоматичному циклі в цілому може бути поданий на рис. 6.8.

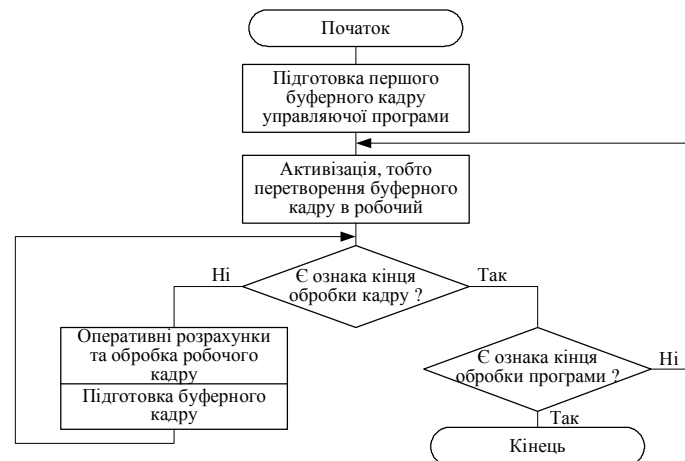


Рис. 6.8. Алгоритм роботи пристрою ЧПУ

Основним змістом переробки інформації у автоматичному циклі є підготовка буферного кадру та відпрацювання робочого кадру. Підготовка буферного кадру полягає у перетворенні інформації з того вигляду, у якому вона представлена на фізичному програмоносії та у пам'яті пристрою ЧПУ, у той вигляд, який найбільше потрібен для виконання розрахунків, пов'язаних з відпрацюванням робочого кадру. Таким чином, будь-який кадр управляючої програми врешті-решт стає буферним, а потім і робочим.

6.2.3. Розрахунок траєкторії руху робочого органу технологічного устаткування

З моменту активізації управляючої програми в автоматичному циклі її інформація включається у процес обчислення. Важливу роль в цьому процесі відіграє підготовка буферного кадру, необхідна на стільки, що вихідні тексти управляючої програми можуть не містити у явному вигляді всієї інформації, необхідної для траєкторних розрахунків.

Підготовка буферного кадру включає розрахунки еквідистант, перетворення координатних систем, визначення даних, що використовуються алгоритмами інтерполяції на другій фазі реалізації геометричної задачі пристроїв ЧПУ, тобто у випадку програмування руху робочого органу як у ручному режимі (програмування еквідистанти контуру обробки), так і в безеквідистантному режимі (програмування безпосередньо контуру деталі) з'являються наступні задачі обчислювального характеру: визначення координат опорних точок еквідистантного контуру, синтез додаткових кадрів руху робочого органу, необхідних для поєднання суміжних ділянок контуру, перетворення координат основного контуру деталі до координат еквідистантного контуру руху робочого органу.

Розрахунок опорних точок еквідистантного контуру. Обробка контуру проводиться інструментом фіксованого (незмінного)

розміру. Очевидно, координати траєкторії центра інструмента, наприклад, фрези, радіуса r_ϕ (рис. 6.9) відрізняються від координат контуру деталі, поданих в кадрі управляючої програми.

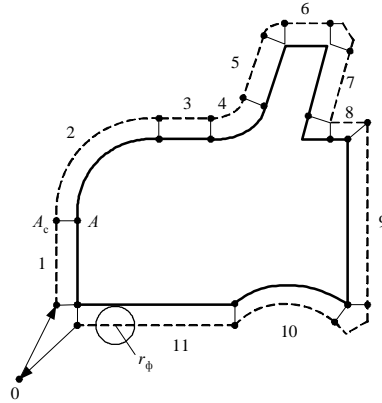


Рис. 6.9. Поєднання ділянок контуру еквідистанти

Розрахунок зміщення опорних точок еквідистантного контуру відносно контуру деталі (наприклад, точки A_c відносно A) повинен автоматично виконуватись пристроєм ЧПУ або верстатником-технологом при ручному програмуванні еквідистанти. Ця задача є відносно простою тільки при *гладкому (плавному) поєднанні* елементів контуру (з кутом з'єднання елементів $\beta = 180^\circ$), наприклад, при поєднанні елементів 2, 3, 4. У цьому випадку довжина прямолінійних ділянок не змінюється і розрахунок еквідистантних зміщень проводиться тільки для кадрів, у яких запрограмована дуга кола. Цей розрахунок виконується з використанням значень координат центра кола I, J і приростів координат X, Y між початковою та кінцевою точками дуги (рис. 6.10), які задаються у кадрі програми.

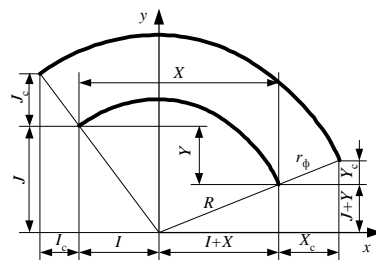


Рис. 6.10. Гладке поєднання елементів контуру

Неважко побачити, що проекції зміщення I_c, J_c початкової точки на вісь X та вісь Y визначаються виразами:

$$I_c = \frac{I}{R} r_\phi; \quad J_c = \frac{J}{R} r_\phi,$$

де $R = \sqrt{I^2 + J^2}$ – радіус кола;

r_ϕ – радіус інструмента.

Відповідно проекції X_c, Y_c зміщення кінцевої точки:

$$X_c = \frac{X + I}{R} r_\phi; \quad Y_c = \frac{Y + J}{R} r_\phi.$$

Розрахунок зміщень значно ускладнюється за наявності *зламів контуру*, що програмується (при кутах з'єднання елементів контуру $\beta > 180^\circ$ або $\beta < 180^\circ$). Найчастіше у цьому випадку при куті $0^\circ < \beta < 90^\circ$ використовується *метод зв'язуючої дуги* (поєднання елементів 5 і 6, 10 і 11), суть якого полягає в тому, що у програму вставляється додатковий кадр з круговим рухом, який відповідає обходу дуги з радіусом, рівним чи більшим r_ϕ (рис. 6.11, а). Наприклад, при поєднанні дугою елементів ламаної з координатами кінцевих точок X_1, X_2 і Y_1, Y_2 вирази для розрахунку координат центра дуги I_c, J_c та координат її кінця X_c, Y_c у приростах визначаються формулами:

$$I_c = \pm r_\phi \frac{Y_1}{L_1}; \quad X_c = r_\phi \left(\pm \frac{Y_1}{L_1} \pm \frac{Y_2}{L_2} \right);$$

$$J_c = \pm r_\phi \frac{X_1}{L_1}; \quad Y_c = r_\phi \left(\pm \frac{X_1}{L_1} \pm \frac{X_2}{L_2} \right);$$

$$L_1 = \sqrt{X_1^2 + Y_1^2}; \quad L_2 = \sqrt{X_2^2 + Y_2^2}.$$

Метод зв'язуючої дуги не завжди допустимий за технологією обробки, оскільки гострі зовнішні кути контуру деталі при обході по дузі виявляються сточеними (поєднання елементів 6 і 7, 9 і 10), а у внутрішніх кутах залишається дуже багато невибраного матеріалу (7 і 8). У цьому випадку доцільно застосовувати *метод поєднання по лініях продовження еквідистант* (при поєднанні прямих) або по *лініях продовження дотичних до еквідистант* (при поєднанні прямих та дуг кола).

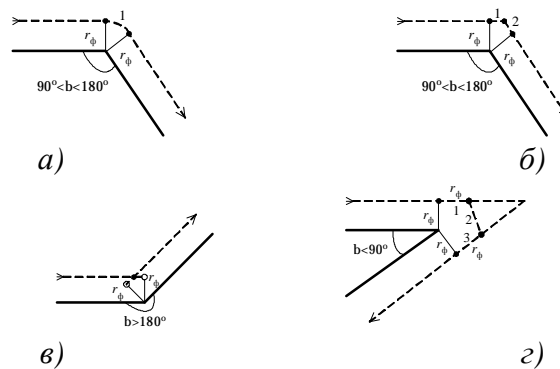


Рис. 6.11. Способи з'єднання елементів контуру при зламі

а – метод зв'язуючої дуги; б – метод продовження еквідистанти; в – метод перетину еквідистант; г – метод скорочених ліній продовження

У методі продовження еквідистант (рис. 6.11, б) необхідно розрахувати опорну точку, яка є перетином елементів продовження еквідистантного контуру через розв'язок відповідних систем рівнянь

першого (при поєднанні прямих) або другого порядку (при поєднанні прямих та дуг кола). Враховуючи обмежені можливості пристрою ЧПУ при розрахунку арифметичних виразів, розв'язання рівнянь є непростю задачею.

Розглянемо її рішення на прикладі обходу точки поєднання двох прямих інструментом з радіусом r_ϕ (рис. 6.12).

Аналітичні вирази суттєво спрощуються, якщо основний контур повернути на кут α так, щоб ділянка з координатами X_1, Y_1 співпала з віссю абсцис. Розглядаючи трикутники BDE і ABC , отримуємо для значень еквідистантних зміщень розвернутого контуру (рис. 6.12, б):

$$X_{co} = \frac{AC}{\cos \beta} \pm DE \operatorname{ctg} \beta; \quad Y_{co} = r_\phi;$$

$$\cos \beta = \frac{H}{L_2}; \quad \operatorname{ctg} \beta = \frac{\sqrt{L_2^2 - H^2}}{H}; \quad L_2 = \sqrt{X_2^2 + Y_2^2},$$

де $AC = DE = r_\phi$;

L_2 – довжина відрізка другої прямої;

H – відстань від кінцевої точки другої прямої до першої прямої:

$$H = \frac{X_1 Y_2 - Y_1 X_2}{L}.$$

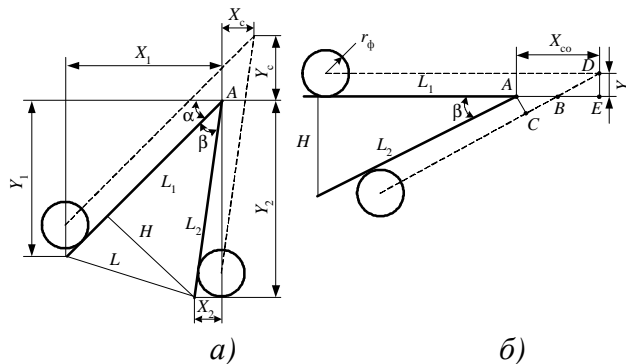


Рис. 6.12. Розрахунок з'єднання двох прямих:
а – вихідний контур; б – розвернутий контур

Із врахуванням наведеного вище отримаємо:

$$X_{co} = r_\phi \frac{L_2 \pm \sqrt{L_2^2 - H^2}}{H}; \quad Y_{co} = r_\phi.$$

Для визначення знака у формулі для X_{co} необхідно порівняти величини:

$$L = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$$

та

$$L_0 = \sqrt{L_1^2 + L_2^2}.$$

Якщо $L < L_0$, то кут β – гострий і у формулі використовується знак “+”, у протилежному випадку – знак “-”. Суттєвою є та обставина,

що для обчислення X_{co} , Y_{co} за формулою у повороті координатної системи немає необхідності, оскільки у формулу входять величини L_1 , L_2 , H , які є інваріантними до переміщень контуру. Поворот координат необхідний тільки на останньому етапі розрахунку при визначенні значень зміщень X_c , Y_c у вихідній системі координат і проводиться відповідно до виразу:

$$\begin{bmatrix} X_c \\ Y_c \end{bmatrix} = \begin{bmatrix} \frac{X_1}{L_1} & -\frac{Y_1}{L_1} \\ \frac{Y_1}{L_1} & \frac{X_1}{L_1} \end{bmatrix} \begin{bmatrix} X_{co} \\ Y_{co} \end{bmatrix},$$

де $\frac{X_1}{L_1} = \cos \alpha$, $\frac{Y_1}{L_1} = \sin \alpha$.

З'єднання прямих і дуг кіл з використанням інваріант контуру дає ще більш помітне спрощення алгоритмів і скорочення часу обчислень.

Необхідно зазначити, що якщо зовнішній кут контуру деталі $\beta > 180^\circ$ стає внутрішнім (рис. 6.11, в), то точка з'єднання знаходиться на перетині еквідистант суміжних ділянок контуру (поєднання елементів 7 і 8).

Проте, якщо зовнішній кут контуру деталі β – гострий ($0^\circ < \beta < 90^\circ$), то застосування методу з'єднання по лініях продовження стає неефективним з точки зору витрат часу на перехід від однієї ділянки контуру до суміжної. В такому випадку часто витрачений час найчастіше може бути зменшений за рахунок *скорочення довжини ліній продовження* до величини радіуса фрези r_ϕ і побудови траєкторії з трьох прямих (рис. 6.11, г). Крім того, такий спосіб дозволить застосувати на другій прямій максимальну швидкість приводів подачі і призначити встановлююче переміщення у зв'язку з тим, що не відбувається контакту інструмента з деталлю.

Перетворення координатних систем. Перехід від координат x_0 , y_0 основного контуру до координат x_e , y_e еквідистантного контуру для прямих і кіл (рис. 6.13) здійснюється за формулами:

$$x_e = x_0 \pm r_\phi \frac{k}{\sqrt{1+k^2}}; \quad y_e = y_0 \pm r_\phi \frac{k}{\sqrt{1+k^2}}$$

або

$$x_e = x_0 \pm r_\phi \sin \alpha; \quad y_e = y_0 \pm r_\phi \cos \alpha,$$

де $k = \tan \alpha$;

r_ϕ – радіус фрези, для якого підраховується еквідистанта;

α – кут нахилу дотичної в точці x_0 , y_0 . Знак “+” або “–” застосовується в залежності від розташування інструмента при обході контуру деталі відповідно ліворуч або праворуч, якщо дивитися в напрямку руху інструмента.

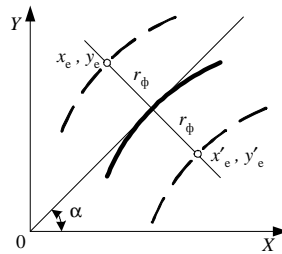


Рис. 6.13. Перетворення координат контуру деталі

Даний розрахунок виконується при призначенні в кадрі управляючої програми підготовчих функцій “Корекція на радіус фрези” G41, G42.

Також у пристроях ЧПУ може виконуватися перетворення координат у випадку врахування довжини робочого інструмента. Для цього в кадрі управляючої програми використовуються підготовчі функції “Корекція довжини інструмента” G43, G44, що є вказівкою додати значення довжини до координати або відняти від координати, заданої в кадрі.

Скасування всіх корекцій контуру на розміри інструмента виконується підготовчою функцією G40.

Крім перетворення координат, у пристроях ЧПУ може виконуватися перетворення швидкостей подачі. Розрахунок швидкості подачі здійснюється з метою забезпечення при управлінні сталості контурної швидкості і полягає у розкладі вектора контурної швидкості на складові в декартовій системі координат.

Для лінійних контурів справедливі наступні співвідношення:

$$V_x = \frac{V \Delta x}{\sqrt{\Delta x^2 + \Delta y^2}}; \quad V_y = \frac{V \Delta y}{\sqrt{\Delta x^2 + \Delta y^2}},$$

де V , V_x , V_y – відповідно контурна швидкість подачі та її складові;

Δx , Δy – кадрові переміщення по осях X , Y .

Для кругових контурів справедливі співвідношення:

$$V_x = \frac{V x_i}{\sqrt{\Delta x_{\text{ц}}^2 + \Delta y_{\text{ц}}^2}}; \quad V_y = \frac{V y_i}{\sqrt{\Delta x_{\text{ц}}^2 + \Delta y_{\text{ц}}^2}},$$

де x_i , y_i – координати поточної точки дуги кола відносно його центра;

$x_{\text{ц}}$, $y_{\text{ц}}$ – координати центра дуги відносно її початкової точки.

Визначення складових вектора швидкості подачі не є самоціллю і проводиться лише в тих випадках, коли цього вимагають алгоритми інтерполяції й алгоритми управління приводами подачі.

6.3. Інтерполяція формоутворення деталей

6.3.1. Види інтерполяції в пристроях ЧПУ

Інформація робочого кадру при програмуванні формоутворення деталі носить узагальнений характер. Вона узагальнено описує геометричну задачу (тобто ту траєкторію, яку необхідно відтворити),

але нічого не говорить про засоби рішення (тобто про те, якими повинні бути команди, що оперативно видаються на слідкуючі приводи подачі).

Обчислювальну процедуру пристрою ЧПУ, що забезпечує перехід від загального опису заданого переміщення до оперативних команд у функції часу для виконавчих приводів, називають *інтерполяцією*, а блок, який реалізує цю процедуру – *інтерполятором*.

Інтерполяцію здійснюють над цілими числами, кожна одиниця яких відповідає найменшому переміщенню чи куту повороту робочого органу верстата, який може бути проконтрольований у процесі управління. Таку відповідність трактують як *дискретність переміщення*.

Оскільки контроль переміщення на верстаті вздовж кожної координатної осі виконує датчик зворотного зв'язку по положенню слідкуючого приводу подачі, то і дискретність переміщення визначається ціною розбиття шкали цього датчика.

Нехай, наприклад, дискретність переміщення деякої системи ЧПУ складає 1 мкм. Це означає, що ціна слідкуючого приводу подачі дорівнює 1 мкм. Тоді задане у деякому кадрі переміщення, підготовлене для інтерполяції, повинно бути подане цілим числом мікрометрів. У загальному випадку задане переміщення на рівні робочого кадру подається цілим числом *дискрет*. Таким чином, під дискретою можна розуміти і керуючу команду, яка надходить на вхід слідкуючого приводу подачі. Слідкуючий привод подачі відпрацьовує кожну дискрету відповідно до її ціни, тобто у вигляді деякого елементарного переміщення.

При дискретності у 1 мкм переміщення X375. Y160. еквівалентне переміщенню, що інтерполюється, по осі X на 375000 дискрет і по Y – на 160000 дискрет.

Процедура інтерполяції обслуговує той кадр управляючої програми, який на даний момент є робочим. Інтерполяційні обчислювальні цикли відтворюються з високою частотою для того, щоб одержати інформацію, необхідну для управління слідкуючими приводами подачі з метою руху по заданій траєкторії. Для цього потрібні великі витрати обчислювальної потужності пристрою ЧПУ. У зв'язку з цим існує декілька видів інтерполяції, що застосовуються у пристроях ЧПУ в залежності від *виду елементарних ділянок руху*: лінійна, кругова інтерполяція, а також від *принципу видачі керуючих дискрет* на координатні приводи подачі: рівномірна інтерполяція, інтерполяція з ведучою і веденою координатами, інтерполяція на постійній несучій частоті.

На рис. 6.14 наведені циклограми початкового і завершуючого етапів відпрацювання кадру при різних видах *лінійної інтерполяції*.

Виходячи з заданої у кадрі швидкості подачі V , нескладно розрахувати час повного відпрацювання кадру при лінійній

інтерполяції:

$$\tau = \frac{\sqrt{X^2 + Y^2}}{V},$$

де X, Y – кадрові прирости координат по відповідних осях;
 V – контурна швидкість подачі.

Лінійна *рівномірна інтерполяція* могла б складатись із рівномірної видачі за цей час керуючих дискрет у координатні приводи подачі X і Y . Періоди видачі дискрет T_x, T_y для кожної координати розраховуються окремо, тому й обчислювальний цикл (період) інтерполяції T є змінним за час відпрацювання кадру:

$$T_x = \frac{\tau}{X}; \quad T_y = \frac{\tau}{Y}.$$

Тому, з метою економії витрат обчислювальної потужності процес інтерполяції вигідніше будувати за *принципом управління за ведучою та веденою координатами*.

Координату, по якій виконується найбільше переміщення, називають *ведучою*, а всі інші – *веденими*. Управляючі дискрети формують так, щоб вони поступали у привід подачі ведучої координати насправді рівномірно. Для ведених координат видача управляючих дискрет на приводи подачі дозволена лише у моменти часу, які визначені дискретами ведучої координати (рис. 6.14, б). Це призводить, як правило, до деякого запізнення у часі видачі дискрет відомої координати, тобто теоретично буде присутнє деяке відхилення від заданої прямої лінії, яке не перевищує ціни однієї дискрети. Перевага ж цього принципу у тому, що обчислювальний цикл інтерполяції $T = \min\{T_x, T_y\}$ завершується одночасною видачею керуючих команд для приводів подачі усіх координат.

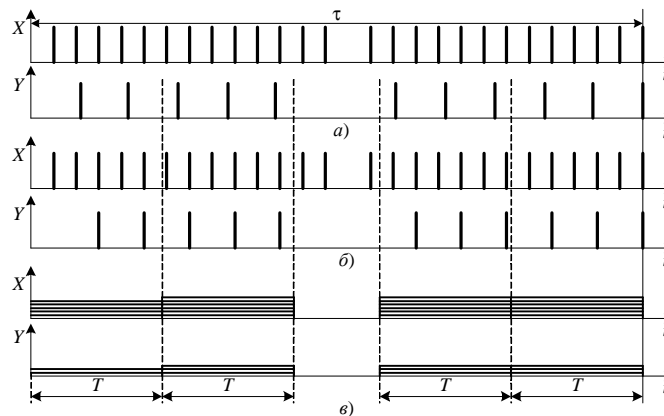


Рис. 6.14. Види лінійної інтерполяції:

а – рівномірна видача управляючих дискрет у приводи подачі координат X, Y; б – інтерполяція з використанням принципу управління за ведучою та веденою координатами; в – інтерполяція на постійній несучій частоті

Практично інтерполяцію організовують наступним чином. В результаті чергового обчислювального циклу, який виконується із максимально високою швидкістю у машинному масштабі часу, визначають, у які приводи подачі повинні бути видані дискрети на поточному етапі управління. Результат зберігають у буфері, який опитують з частотою, що відповідає швидкості подачі для ведучої координати. Таким чином, розрахунки машинного масштабу прив'язують до реального часу, розвиток подій у якому визначається виходячи з технологічних міркувань. Проте, значення періоду інтерполяції залишається фіксованим тільки протягом опрацювання поточного кадру. При переході до наступного кадру цю величину необхідно перерахувати, і вона може бути змінена. З метою покращення роботи інтерполатора пристрою ЧПУ краще було б забезпечити постійних період інтерполяції протягом виконання всієї управляючої програми.

Тому можлива інша принципіальна схема інтерполяції – *на постійній несучій частоті* (рис. 6.14, в). У кожному періоді постійної частоти і для кожної координати окремо розраховують число дискрет, які у цьому періоді повинні бути відпрацьовані на підставі швидкості подачі приводу даної координати:

$$\tilde{\Delta x} = \frac{X \cdot T}{\tau}; \quad \tilde{\Delta y} = \frac{Y \cdot T}{\tau}.$$

Результати розрахунків, скоріш за все, виявляться дробовими, хоча при управлінні дробова частина від дискрети фізичного сенсу немає. Просте перетворення до найближчого цілого призвело б до накопичення помилок, тобто до відхилення від заданої траєкторії. Тому пачки дискрет, підготовлені до видачі у приводи подачі у поточному періоді постійної частоти, перетворюють до цілих чисел, а дробові залишки накопичують та враховують у наступних періодах. У зв'язку з цим картина видачі пачок у приводи від періоду до періоду нерівномірна, проте період інтерполяції T залишається незмінним для всіх управляючих програм ЧПУ.

Розрахунки, пов'язані з *круговою інтерполяцією*, суттєво складніші. Цілком зрозуміло, що при круговій інтерполяції частоти видачі управляючих дискрет повинні бути суттєво нерівномірними (рис. 6.15).

Якісна картина появи управляючих дискрет на входах слідкуючих приводів подачі по координатах X і Y при незалежній їх видачі та відпрацюванні показана на рис. 6.15, а.

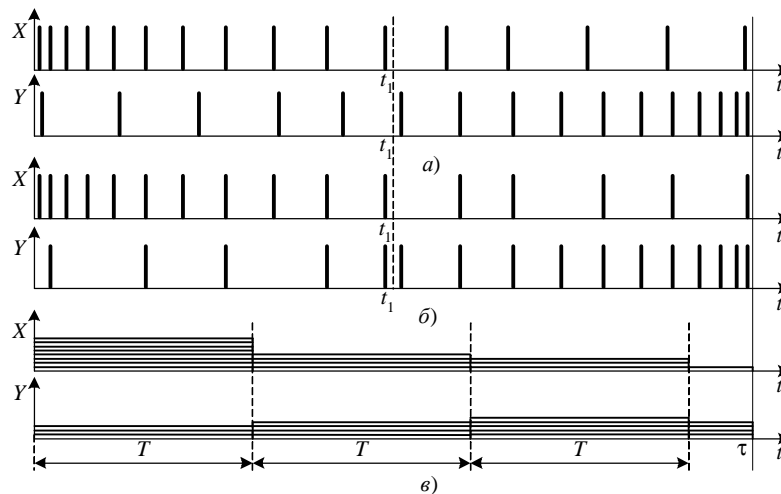


Рис. 6.15. Види кругової інтерполяції:

а – незалежна поява управляючих дискрет на входах слідкуючих приводів подачі по координатах X, Y; б – інтерполяція з використанням принципу управління за ведучою та веденою координатами; в – інтерполяція на постійній несучій частоті

При круговій інтерполяції з використанням принципу управління за ведучою та веденими координатами необхідно відмітити таку особливість її реалізації: ведуча та ведена координати, починаючи з деякого моменту часу, міняються ролями (рис. 6.15, б). До моменту t_1 ведучою є координата X, і саме на дискрети цієї координати налагоджені інтерполяційні цикли, а по проходженню цього моменту ведучою стає координата Y, і вже вона починає визначати моменти часу видачі управляючих дискрет. Цей момент часу t_1 визначається тим фактом, що приріст ведучої координати при відпрацюванні дискрет дорівнює приросту веденої координати. Взагалі при обході кола ведуча і ведена координати по чергово змінюються. Якщо дискретність переміщення по координатах однакова для верстата, то площина кола поділяється лініями, проведеними під кутом 45° до координатних осей, на сектори, де діють як ведучі різні координати (рис. 6.16).

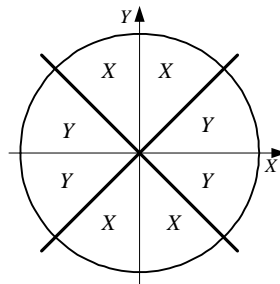


Рис. 6.16. Сектори розподілу ведучої координати при круговій інтерполяції

Кругова інтерполяція на постійній частоті продемонстрована на циклограмах, зображених на рис. 6.15, в. Об'єми пачок управляючих

дискрет, що визначаються інтерполяційними розрахунками у сусідніх періодах, можуть значно відрізнятись.

Як і при лінійній інтерполяції, інтерполяційні розрахунки машинного масштабу часу відокремлені від процесу видачі управляючих дискрет на приводи подачі. Видача робиться за результатами опитування буфера з частотою, що визначається швидкістю контурної подачі. Отже, інтерполяційний процес є сукупністю обчислювальних циклів, які неперервно повторюються до повної обробки усієї інформації кадру і завершуються визначенням комбінацій, видаваних на приводи подачі управляючих дискрет або пачок управляючих дискрет.

Одними з головних переваг верстатів з ЧПУ є *підвищення та стабільність якості обробки* (точність розмірів і форми оброблюваної деталі, шорсткість оброблених поверхонь), що забезпечують ідентичність деталей всієї партії. Представляє інтерес вплив пристроїв ЧПУ на точність обробки в результаті таких факторів, як вид інтерполяції, дискретність, величина і форма траєкторії контуру, що відпрацьовується, тощо.

При лінійній інтерполяції контуру похибка переміщення інструмента відносно заготовки залежить від кута нахилу лінійного контуру (рис. 6.17). Найбільша похибка в одну дискрету буде при найменшому куті нахилу контуру 1:10. При круговій інтерполяції контуру, утвореного з дуг кола, відхилення по радіусу може досягати двох дискрет. Відхилення реального контуру внаслідок згладжуючої дії кінцевої фрези буде менше однієї дискрети. Таким чином, величина дискрети і є найменшою величиною відхилення (похибки) за відсутності складових похибки від інших факторів (теплові деформації, знос інструмента і механічних частин верстата тощо).

Величина дискрети в сучасних верстатах з ЧПУ складає 10, 5, 2 і 1 мкм.

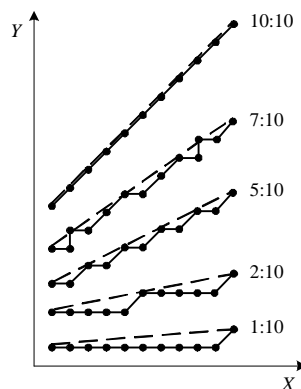


Рис. 6.17. Похибки інтерполяції

Для технологічно складних задач у процес інтерполяції може бути включено до п'яти керованих координат. П'ятикоординатне формоутворення зустрічається при обробці так званих "скульптурних"

поверхонь, коли інструмент утримується вздовж нормалі або під кутом до оброблюваної поверхні і повинна бути врахована корекція на змінний радіус інструментів, що застосовувалися. Для реалізації п'ятикоординатного формоутворення до трьох основних приводів подачі у декартовій координатній системі повинні бути добавлені приводи подачі столів, які можуть обертатись, інструментальні головки, що качаються або обертаються, тощо.

Алгоритми багатокординатної інтерполяції відрізняються високою складністю обчислення, хоча у саму структуру геометричної задачі ЧПУ нічого нового не вносять.

Характер підготовки даних для інтерполяції істотно залежить від вибору алгоритмів інтерполяції. Алгоритм інтерполяції виконується циклічно з високою частотою. При цьому вимагаються великі витрати обчислювальної потужності. У зв'язку з цим розробці алгоритмів інтерполяції традиційно приділяли велику увагу, і на сьогоднішній день існує декілька методів інтерполяції та десятки їх версій та варіантів. Далі розглянемо основні групи методів інтерполяції, що використовуються в пристроях ЧПУ: оцінної функції, цифрових диференціальних аналізаторах, прогнозу та корекції.

6.3.2. Методи оцінної функції

Відповідно до методу оцінної функції виконується моделювання алгебраїчного рівняння відтвореної кривої, тобто в основу рішення покладений розв'язок алгебраїчних рівнянь. Суть методу полягає в тому, що за результатом кроку вздовж будь-якої керованої координати обчислюється допоміжна (оцінна) функція F , знак якої визначає напрям наступного кроку, причому переміщення, що виникає в результаті цього кроку, наближає траєкторію, що відпрацьовується, до заданої кривої. Всі обчислення ведуться в цілих числах – одиницях дискрет, тобто величина кроку дорівнює дискреті.

Нехай інтерполяції підлягає деяка довільна крива, що описується рівнянням $0 = y - f(x)$, де x – координата по осі абсцис (аргумент функції); y – координата по осі ординат (значення функції), або рівнянням $0 = x - f'(y)$ при протилежному напрямку осей координат (рис. 6.18). Тоді оцінна функція має вигляд: $F = y - f(x)$ або $F = x - f'(y)$.

Для оцінної функції F самого загального виду справедливе твердження: $F = 0$, якщо точка, що відтворює необхідну траєкторію, знаходиться на заданій кривій; $F > 0$, якщо точка, що відтворює необхідну траєкторію, знаходиться в області вище кривої; $F < 0$, якщо згадувана точка знаходиться в області нижче кривої.

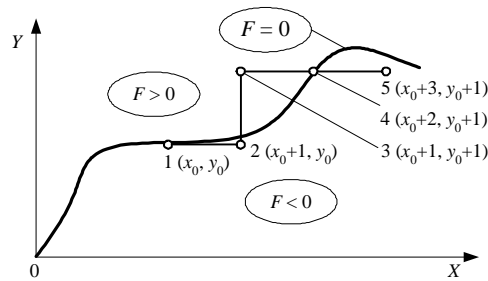


Рис. 6.18. Реалізація інтерполяційних рухів методом оцінної функції

Таким чином, знак оцінної функції F може послужити індикатором того, у якому напрямку варто зробити черговий крок заданої траєкторії. Якщо поточна точка траєкторії інтерполяції знаходиться в області $F > 0$ або $F = 0$, то наступний крок з переміщення на дискрету виконується по осі абсцис. Якщо поточна точка траєкторії інтерполяції знаходиться в області $F < 0$, то наступний крок по переміщенню на дискрету виконується по осі ординат. Кроки виконуються безперервно (з частотою, що визначається блоком задання швидкості) доти, поки траєкторія інтерполяції не досягне кінцевої точки переміщення.

У точці 1 (x_0, y_0) оцінна функція $F = 0$, робимо крок по осі X . Черговою точкою, що відтворює необхідну траєкторію, стане точка 2 $(x_0 + 1, y_0)$. Ця точка знаходиться в області $F < 0$. Отже, новий крок повинен бути зроблений по осі Y і т.д. Із наведених міркувань випливає, що для управління потрібно лише знати спосіб визначення знака F . Такий спосіб у загальному вигляді отримати не можна, однак він без великих зусиль встановлюється для конкретної траєкторії, що інтерполюється.

Лінійна інтерполяція методом оцінної функції. При лінійній інтерполяції рівняння траєкторії буде мати вигляд: $0 = y_i X - x_i Y$, де x_i, y_i – координати поточної точки прямої лінії; X, Y – кадрові прирости по осях X, Y (або координати x_k, y_k кінцевої точки відносно початкової точки відрізка лінії).

Оцінна функція для прямої лінії з I квадранта площини (правої декартової системи координат) має вигляд:

$$F_i = y_i X - x_i Y.$$

Цей вираз дозволяє обчислювати оцінну функцію і визначати її знак.

Нехай зроблений крок в одну дискрету по осі X . Тоді:

$$F_{i+1} = y_i X - (x_i + 1)Y = (y_i X - x_i Y) - Y = F_i - Y.$$

Нехай зроблений крок в одну дискрету по осі Y . У цьому випадку:

$$F_{i+1} = (y_i + 1)X - x_i Y = (y_i X - x_i Y) + X = F_i + X.$$

Таким чином, обчислення нового значення оцінної функції опираються на попереднє значення, що зберігається. Оскільки

початкове значення оцінної функції дорівнює 0, всі наступні її значення визначаються тільки за допомогою величин кадрових приростів, вибраних з програми. При цьому знак оцінної функції, отриманий в результаті чергового кроку, визначає напрям наступного кроку.

Так, пряма OA , що інтерполюється (рис. 6.19), ділить площину XY на дві області: $F > 0$ і $F < 0$. Область $F > 0$ знаходиться над прямою, а область $F < 0$ – нижче від неї. Відрізок прямої OA , що інтерполюється, є областю, де $F = 0$.

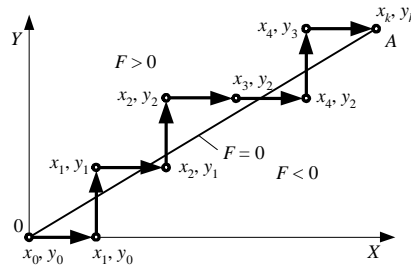


Рис. 6.19. Лінійна інтерполяція методом оцінної функції

Якщо проміжна точка траєкторії інтерполяції знаходиться в області $F < 0$, то наступний крок переміщення на одну дискрету виконується по осі X . Якщо ж проміжна точка траєкторії знаходиться в області $F > 0$, то наступний крок виконується по осі Y .

Оскільки робота відбувається у відносній системі координат, початок відрізка, що інтерполюється, завжди знаходиться у початку координат. При цьому початкова точка траєкторії інтерполяції знаходиться на початку відрізка, що інтерполюється (в області $F = 0$), і має координати $x_0 = 0, y_0 = 0$.

Якщо відрізок лінії розташований в інших квадрантах координатної площини, то для застосування наведених формул і алгоритму інтерполяції необхідно попередньо виконати перехід до I квадранта. У випадку, якщо кут нахилу лінії до осі абсцис складає більше 45° , то доцільно поміняти місцями координати ($x' = y, y' = x$) і врахувати це у формулах.

Кругова інтерполяція методом оцінної функції. При круговій інтерполяції рівняння відтвореної траєкторії буде мати вигляд:

$$0 = (y_i^2 - y_0^2) + (x_i^2 - x_0^2),$$

а оцінна функція:

$$F_i = (y_i^2 - y_0^2) + (x_i^2 - x_0^2),$$

де x_i, y_i – координати поточної точки кола;

x_0, y_0 – координати початкової точки дуги відносно її центра.

У випадку руху в I квадранті проти годинникової стрілки отримаємо наступні вирази для оцінної функції:

– при кроці по осі X координата $x_{i+1} = x_i - 1$, а оцінна функція:

$$F_{i+1} = (y_i^2 - y_0^2) + [(x_i - 1)^2 - x_0^2] = F_i - 2x_i + 1;$$

– при кроці по осі Y координата $y_{i+1} = y_i + 1$, а оцінна функція:

$$F_{i+1} = [(y_i + 1)^2 - y_0^2] + (x_i^2 - x_0^2) = F_i + 2y_i + 1.$$

Оскільки початкове значення оцінної функції дорівнює нулю, всі наступні значення цієї функції визначаються поточними координатами траєкторії інтерполяції.

Як впливає з наведених формул і алгоритму, на кожному кроці потрібно обчислити тільки одну змінну F . Всього можливі вісім варіантів обходу чотирьох квадрантів кола. Правила вибору напрямку кроку для цих випадків показані на рис. 6.20.

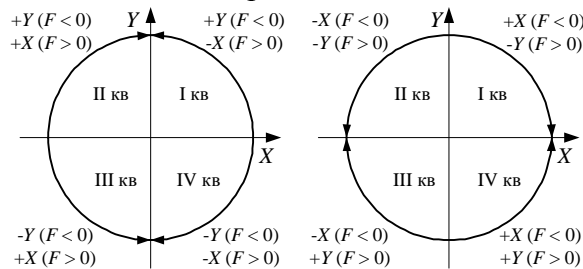


Рис. 6.20. Правило вибору кроку при інтерполяції з оцінною функцією в залежності від напрямку руху та квадранта

Кругова інтерполяція методом оцінної функції полягає в наступному (рис. 6.21). Коло, на якому розташований відрізок дуги, що інтерполюється, ділить площину XY на дві області: $F > 0$, що лежить поза колом, і область $F < 0$, що лежить в середині кола; саме коло є областю, де $F = 0$. Відрізок дуги, що інтерполюється, має початкову точку з координатами x_0, y_0 і кінцеву точку з координатами x_k, y_k (початок відносних координат знаходиться в центрі кола).

Якщо проміжна точка траєкторії інтерполяції (наприклад, точка з координатами (x_1, y_3)) знаходиться в області $F > 0$, то наступний крок робиться по осі X . Якщо ж проміжна точка траєкторії інтерполяції (наприклад, точка з координатами (x_2, y_3)) знаходиться в області $F < 0$, то наступний крок робиться по осі Y .

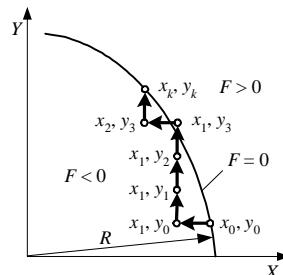


Рис. 6.21. Кругова інтерполяція методом оцінної функції

Отже, за кожний поточний відрізок часу лінійно-круговий інтерполятор повинен відпрацьовувати один з чотирьох можливих режимів:

Лінійна інтерполяція

- | | |
|---|---|
| <p>Режим 1 ($F > 0$)</p> <ol style="list-style-type: none"> 1. Крок по осі X 2. $F_{i+1} = F_i - Y$ 3. $x_{i+1} = x_i + 1$ 4. Перевірка $x_{i+1} = X$ | <p>Режим 2 ($F < 0$)</p> <ol style="list-style-type: none"> 1. Крок по осі Y 2. $F_{i+1} = F_i + X$ 3. $y_{i+1} = y_i + 1$ 4. Перевірка $y_{i+1} = Y$ |
|---|---|

Кругова інтерполяція

- | | |
|--|--|
| <p>Режим 3 ($F > 0$)</p> <ol style="list-style-type: none"> 1. Крок по осі X 2. $F_{i+1} = F_i - 2x_i + 1$ 3. $x_{i+1} = x_i - 1$ 4. Перевірка $x_{i+1} = X$ | <p>Режим 4 ($F < 0$)</p> <ol style="list-style-type: none"> 1. Крок по осі Y 2. $F_{i+1} = F_i + 2y_i + 1$ 3. $y_{i+1} = y_i + 1$ 4. Перевірка $y_{i+1} = Y$ |
|--|--|

Алгоритм інтерполяції методом оцінної функції досить просто реалізується програмно або апаратно. Всі операції режиму виконуються одночасно протягом одного періоду роботи. У випадку апаратного виконання інтерполяції тривалість одного періоду роботи визначається часом обертання кодів у регістрах зсуву, на яких побудований інтерполятор. За цей час повинен бути встановлений потрібний режим роботи, коди чисел, що містяться в регістрі (наприклад, код величини поточної координати x_i), повинні бути виведені з нього шляхом зсуву з одночасним виконанням арифметичних операцій над цими кодами ($x_{i+1} = x_i \pm 1$), а коди нового значення цих величин (x_{i+1}) – введені в регістр. Введення та виведення кодів відбувається одночасно – молодший розряд нового коду йде відразу ж за старшим (знаковим) розрядом попереднього коду.

Проте, підвищені вимоги до шорсткості утворюваної поверхні вимагають заміни покоординатного руху при інтерполяції. З цієї метою був розроблений наступний варіант методу оцінної функції із застосуванням принципу управління за ведучою і веденою координатами.

6.3.3. Методи оцінної функції з прогнозуючим кроком

Трикоординатні лінійно-кругові інтерполятори здійснюють лінійну інтерполяцію одночасно за трьома координатами (X , Y , Z) і кругову інтерполяцію в одній із трьох площин обробки (XY , XZ чи YZ), що обираються за програмою. Напрямок чергового кроку інтерполяції в цих інтерполяторах відбувається за методом оцінної функції з прогнозуючим кроком, причому кроки здійснюються або по одній осі, або по двох одночасно (крок по діагоналі).

Лінійна інтерполяція методом оцінної функції з прогнозуючим кроком. При лінійній інтерполяції вибирається ведуча (з найбільшою величиною переміщення) координата, відносно якої ведеться інтерполяція за двома іншими координатами. При цьому крок по ведучій координаті робиться завжди, а крок по веденій (меншій) – в залежності від знака оцінної функції. Розрізняють кроки прогнозуючі і

кроки на привід подачі (робочий рух).

Припустимо, що в І квадранті ведучою є координата x , а веденою координата y . При інтерполяції відрізка OA прогнозуючі кроки здійснюються звичайним порядком (рис. 6.22): якщо $F > 0$, виконується крок по осі X ; якщо $F < 0$ – по осях X, Y одночасно (тобто крок по діагоналі). Кроки на привід відстають від прогнозуючих на один такт, тобто знак оцінної функції, отриманої в результаті прогнозуючого кроку, визначає напрямок поточного кроку на привід.

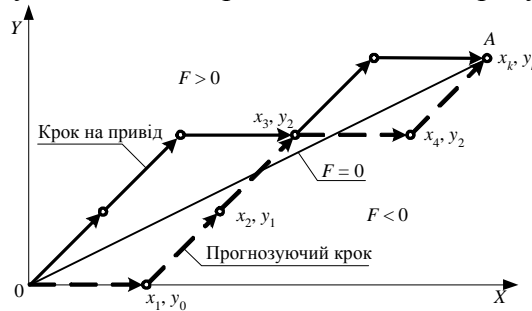


Рис. 6.22. Лінійна інтерполяція методом оцінної функції з прогнозуючим кроком

Оскільки $F_0 = 0$, перший прогнозуючий крок виконується по осі X і перша проміжна точка траєкторії прогнозуючих кроків (x_1, y_0) знаходиться в області $F < 0$. Внаслідок цього перший крок на привід виконується по діагоналі так само, як і другий прогнозуючий крок. Друга проміжна точка траєкторії (x_2, y_1) знаходиться в області $F < 0$, і другий крок на привід (так само, як і третій прогнозуючий крок) виконується по діагоналі. Третій крок на привід і четвертий прогнозуючий крок виконуються по осі X і т.д.

При прогнозуючому кроці по осі X оцінна функція має вигляд:

$$F_{i+1} = F_i - Y,$$

де Y – кадровий приріст відповідної координати, $Y = y_k$.

При прогнозуючому кроці по осі X і Y (по діагоналі) оцінна функція буде:

$$F_{i+1} = F_i - Y + X,$$

де X – кадровий приріст відповідної координати, $X = x_k$.

Отже, загалом алгоритм лінійної інтерполяції з прогнозуючим кроком наступний:

- якщо $F > 0$ або $F = 0$, то наступний прогнозуючий крок виконується по ведучій координаті;
- якщо $F < 0$, то наступний прогнозуючий крок виконується одночасно по ведучій та веденій координатам;
- крок на привід відстає на один такт від прогнозуючого і відповідає наступному прогнозуючому кроку.

Кругова інтерполяція методом оцінної функції з прогнозуючим кроком. Кругова інтерполяція дещо відрізняється від лінійної тим, що необхідно враховувати той факт, що при проходженні лінії

переключення ведуча і ведена координати міняються місцями. Тому при круговій інтерполяції застосовується два окремих алгоритми проведення розрахунку.

При круговій інтерполяції проти годинникової стрілки в першому за напрямком руху напівквадраті першого квадранта (рис. 6.23), для якого максимальним координатним переміщенням є зсув по осі Y , кроки виконуються або по осі Y , або по діагоналі. Якщо у проміжній точці траєкторії прогнозуючих кроків $F > 0$, то виконується прогнозуючий крок по діагоналі, якщо $F < 0$ – по осі ведучої координати Y . Кроки на привід виконуються після виділення нового значення оцінної функції і збігаються за напрямком з наступним прогнозуючим кроком.

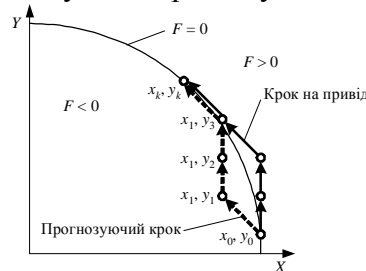


Рис. 6.23. Кругова інтерполяція методом оцінної функції з прогнозуючим кроком

При прогнозуючому кроці по осі Y координата дорівнює $y_{i+1} = y_i + 1$, а оцінна функція:

$$F_{i+1} = F_i + 2y_i + 1.$$

При прогнозуючому кроці по діагоналі координати змінюються $y_{i+1} = y_i + 1$, $x_{i+1} = x_i - 1$, а оцінна функція дорівнює:

$$F_{i+1} = F_i + 2y_i + 2x_i + 2.$$

При круговій інтерполяції проти годинникової стрілки в другому за напрямком руху напівквадраті першого квадранта кроки виконуються: при $F > 0$ – по осі ведучої координати X , а при $F < 0$ – по діагоналі.

При прогнозуючому кроці по осі X координата $x_{i+1} = x_i - 1$, оцінна функція:

$$F_{i+1} = F_i - 2x_i + 1.$$

Кроки по діагоналі в кожному окремому випадку вибираються з метою кращого (плавного) наближення до заданої траєкторії, що інтерполюється. Одночасний рух по обох осях призводить до зниження похибки апроксимації кола та прямої.

Враховуючи, що кругова інтерполяція може виконуватися у двох напрямках руху за і проти годинникової стрілки існує 8 варіантів застосування алгоритмів проведення розрахунків інтерполяції (рис. 6.24).

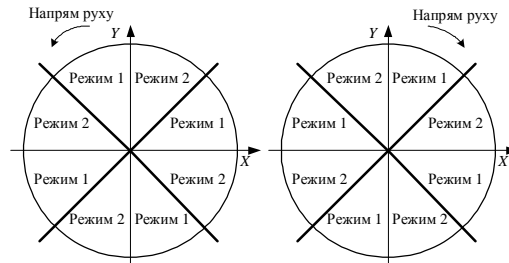


Рис. 6.24. Правило застосування алгоритму кругової інтерполяції методом оцінної функції з прогнозуючим кроком

Узагальнений алгоритм кругової інтерполяції складається з наступних режимів:

Режим 1 (точки знаходяться до лінії переключення координат):

– якщо $F > 0$ або $F = 0$, то прогнозуючий крок виконується по ведучій та веденій координатах;

– якщо $F < 0$, то прогнозуючий крок виконується по ведучій координаті.

Режим 2 (точки знаходяться після лінії переключення координат):

– якщо $F > 0$ або $F = 0$, то прогнозуючий крок виконується по ведучій координаті;

– якщо $F < 0$, то прогнозуючий крок виконується по ведучій та веденій координатах;

– крок на привід відстає на один такт від прогнозуючого і відповідає наступному прогнозуючому кроку.

Аналіз показує, що використання фіксованого циклу із розрахунку одного кроку інтерполяції в межах кожного циклу висуває надмірно високі вимоги щодо швидкодії обчислювача або накладає обмеження на швидкісні можливості верстата, змушуючи зменшувати частоту фіксованих циклів. Тому метод оцінної функції в чистому вигляді можна використовувати лише при апаратній реалізації інтерполятора.

6.3.4. Методи оцінної функції на постійній несучій частоті

Інша можливість застосування ідеї методу оцінної функції полягає у використанні модифікованого алгоритму, що працює на постійній несучій частоті. При цьому інтерполятор в кожному періоді видає у слідкуючі приводи цифрові паралельні коди.

Постійну несучу частоту вибирають з таким розрахунком, щоб виконавчі слідкуючі приводи верстата виконували роль природного фільтра, що згладжує дискретний характер управління.

Наприклад, нехай постійна несуча частота складає 100 Гц. У цьому випадку для приводів верстата як командні будуть сформовані не окремі поодинокі імпульси, а пачки імпульсів або міжаткові прирости, обсяги яких залежать від співвідношення контурної

швидкості та несучої частоти, а також від заданої траєкторії. Керуючі сигнали, що містять інформацію про міжтактові прирости, будуть надходити до виконавчих приводів у паралельному коді з частотою 100 Гц. Однак, динамічні властивості виконавчих приводів такі, що дискретні керуючі сигнали на вході, квантовані за часом і рівнем, будуть згладжені і монотонно відтворені на виході.

Загальний алгоритм інтерполяції методом оцінної функції на постійній несучій частоті побудований за блоковим принципом. При цьому набір блоків, що залучаються у процесі інтерполяції, визначається орієнтацією площини інтерполяції (при лінійній одночасно за трьома координатами і при круговій в одній із трьох площин обробки) і видом інтерполяції.

Лінійна інтерполяція методом оцінної функції на постійній частоті. Розглянемо процес інтерполяції прямолінійної ділянки, заданої кадровими приростами координат X та Y , а також часом t відпрацьовування кадру. Цей час може бути обчислений у блоці позациклових розрахунків за заданою в кадрі швидкістю подачі. Для системи, що працює на постійній несучій частоті з періодом T , міжтактові прирости Δx_i , Δy_i , у деякому i -му циклі інтерполяції зв'язані наступними залежностями:

$$\begin{aligned}\Delta x_i \tau - X \mathcal{G} T &= 0; & x_i &= x_{i-1} + \Delta x_i; \\ \Delta y_i \tau - Y \mathcal{G} T &= 0; & y_i &= y_{i-1} + \Delta y_i.\end{aligned}$$

Підсумовування міжтактових приростів від початку кадру до i -го циклу інтерполяції дозволяє перейти до наступних формул:

$$\sum_{k=1}^i \Delta x_k - i \frac{XT}{\tau} = 0; \quad \sum_{k=1}^i \Delta y_k - i \frac{YT}{\tau} = 0,$$

де $\sum_{k=1}^i \Delta x_k$, $\sum_{k=1}^i \Delta y_k$ – сумарні переміщення відтворюючої траєкторію точки за координатами X і Y від початку кадру до i -го циклу періоду постійної несучої частоти.

Якщо врахувати, що інтерполяція ведеться по цілих числах, то взагалі праві частини цих рівнянь не є нульовими і їх необхідно позначити відповідно через N_i та M_i . В результаті отримаємо систему оцінних функцій:

$$\begin{aligned}N_i &= \sum_{k=1}^i \Delta x_k - i \frac{XT}{\tau} = \sum_{k=1}^i \Delta x_k - i \tilde{\Delta} x; \\ M_i &= \sum_{k=1}^i \Delta y_k - i \frac{YT}{\tau} = \sum_{k=1}^i \Delta y_k - i \tilde{\Delta} y,\end{aligned}$$

де $\tilde{\Delta} x = \frac{XT}{\tau}$, $\tilde{\Delta} y = \frac{YT}{\tau}$ – середні міжтактові прирости відповідних координат, які повинні відпрацьовуватися у кожному циклі інтерполяції, тобто оцінна функція показує сумарне відхилення у

прирості координати за певну кількість циклів (різницю між відпрацьованим і заданим).

Беручи до уваги наведені вирази, стратегію управління для чергового i -го циклу можна побудувати наступним чином:

- якщо $N_i > 0$, то $\Delta x_i = \text{int} \left(\tilde{\Delta x} \right)$, інакше $\Delta x_i = \text{int} \left(\tilde{\Delta x} \right) + \text{sign} (X)$;
- якщо $M_i > 0$, то $\Delta y_i = \text{int} \left(\tilde{\Delta y} \right)$, інакше $\Delta y_i = \text{int} \left(\tilde{\Delta y} \right) + \text{sign} (Y)$,

де функція int здійснює виділення цілої частини числа, а функція sign дорівнює одиниці зі знаком, що збігається зі знаком аргументу функції.

З наведених формул випливає, що чергові міжтактові прирости Δx_i і Δy_i є цілими числами дискрет. Отже, і середні міжтактові прирости також повинні вимірюватись у дискретах. Звичайно, кадрові прирости X , Y задаються в мм, контурна швидкість V – у мм/хв, період постійної частоти T – у сек, а дискретність d – у мм, що необхідно враховувати у формулах визначення середніх міжтактових приростів:

$$\tilde{\Delta x} = \frac{XT}{60d\tau}; \quad \tilde{\Delta y} = \frac{YT}{60d\tau}; \quad \tau = \frac{\sqrt{X^2 + Y^2}}{V}.$$

В результаті алгоритм лінійної інтерполяції буде полягати у паралельному розрахунку тактових приростів всіх координат для поточного циклу інтерполяції:

– прийняти для поточного i -го циклу попередні значення тактових приростів координат $\Delta x_i = \text{int} \left(\tilde{\Delta x} \right)$, $\Delta y_i = \text{int} \left(\tilde{\Delta y} \right)$;

– розрахувати значення оцінних функцій M_i та N_i ;

– якщо значення оцінної функції від'ємне, то скоригувати попереднє значення тактового приросту координати, додавши одну дискрету, інакше остаточною значенням приросту стане попереднє значення.

Необхідно врахувати, що кількість циклів інтерполяції у кадрі визначається як:

$$i_{\max} = \left\lceil \frac{\tau}{T} \right\rceil.$$

Тривалість останнього циклу може бути менша, ніж значення періоду T . Тому тактові прирости координат для останнього циклу визначаються як:

$$\Delta x_{i_{\max}} = X - \sum_{i=1}^{i_{\max}-1} \Delta x_i; \quad \Delta y_{i_{\max}} = Y - \sum_{i=1}^{i_{\max}-1} \Delta y_i.$$

Кругова інтерполяція методом оцінної функції на постійній частоті. Розглянемо процес інтерполяції дуги кола при русі проти годинникової стрілки в першому квадранті. Для кругової інтерполяції в площині (X, Y) відносно i -го періоду справедливими наступні вирази, що

випливають з чисто геометричних співвідношень:

$$\Delta x_i = VT \sin \varphi = \frac{VT}{R} y_i; \quad x_i = x_{i-1} - \Delta x_i;$$

$$\Delta y_i = VT \cos \varphi = \frac{VT}{R} x_i; \quad y_i = y_{i-1} - \Delta y_i,$$

де V – контурна швидкість подачі;

x_i, y_i – поточні координати відтворюючої точки дуги кола в i -му циклі інтерполяції (відносно центра кола);

φ – поточний кут відтворюючої точки дуги, що відліковується проти годинникової стрілки від позитивного напрямку осі X ;

$$R = \sqrt{x_0^2 + y_0^2} \text{ – радіус дуги кола;}$$

x_0, y_0 – координати початкової опорної точки дуги кола.

Представлені формули є аналітичною формою запису умови руху по запрограмованій дузі кола із заданою швидкістю контурної подачі.

Підсумовування міжтактових приростів аж до i -го циклу призводить до наступних рівнянь:

$$\sum_{k=1}^i \Delta x_k - \frac{VT}{R} \sum_{k=1}^i y_k = 0; \quad \sum_{k=1}^i \Delta y_k - \frac{VT}{R} \sum_{k=1}^i x_k = 0,$$

де $\sum_{k=1}^i \Delta x_k, \sum_{k=1}^i \Delta y_k$ – накопичені до i -го циклу суми поточних координат відтворюючої точки дуги кола.

Переходячи до оцінних функцій для кожної координати, що визначають сумарне відхилення від заданої траєкторії руху, отримаємо наступну систему рівнянь:

$$N_i = \sum_{k=1}^i \Delta x_k - \tilde{\Delta\varphi} \sum_{k=1}^i \Delta y_k; \quad M_i = \sum_{k=1}^i \Delta y_k - \tilde{\Delta\varphi} \sum_{k=1}^i \Delta x_k,$$

де параметр $\tilde{\Delta\varphi} = \frac{VT}{R}$ представляє собою середній та постійний в межах всього кадру приріст кута повороту відтворюючої точки за період T несучої частоти вздовж дуги запрограмованого кола.

Наведені вирази оцінних функцій показують складність проведення кругової інтерполяції порівняно з лінійною, що визначається у взаємозалежності проведення розрахунків координат не тільки між собою, а й від поточних їх значень.

Тому вид оцінних функцій дозволяє сформулювати наступну стратегію управління для чергового i -го циклу: встановити $\Delta x_i = 0$ і розрахувати значення оцінної функції (вона повинна бути від'ємною), після чого послідовно збільшуємо Δx_i , доти, поки $N_i > 0$, що відповідає остаточному значенню Δx_i , яке передаємо у вихідний регістр. Процедура визначення Δy_i за своїм змістом аналогічна.

Виходячи з того, що розрахунки для кожної координати є

взаємозалежними, їх виконують паралельно, використовуючи результати однієї координати для іншої і навпаки.

Ступінь врахування значень інших координат дозволяє створити декілька варіантів алгоритмів з різною похибкою у розрахунках.

Один із таких варіантів алгоритмів розрахунків з найбільшим ступенем врахування значень координат може бути таким:

- 1) нехай для i -го циклу інтерполяції $\Delta x_i = 0$ і $\Delta y_i = 0$;
- 2) розрахувати $x_i = x_{i-1} - \Delta x_i$, $y_i = y_{i-1} - \Delta y_i$, а потім N_i , M_i ;
- 3) якщо $N_i \geq 0$ та $M_i \geq 0$, то Δx_i та Δy_i є кінцевими значеннями координатних приростів і цикл інтерполяції завершено;
- 4) якщо $N_i < 0$, то $\Delta x_i = \Delta x_i + 1$, якщо $M_i < 0$, то $\Delta y_i = \Delta y_i + 1$; виконати перехід до п. 2.

Отже, схему інтерполяції методом оцінної функції на постійній несучій частоті можна подати як процес вироблення компенсацій на кожному кроці циклів інтерполяції з метою недопущення відхилень від заданої траєкторії руху (знак оцінної функції на кожному кроці повинен бути додатним). Внаслідок цього інтерполяційна траєкторія буде пролягати переважно зверху над заданою траєкторією для модифікованих методів оцінної функції. Крім того, оцінні функції N_i , M_i оцінюють накопичену похибку інтерполяції у порівнянні з функцією F_i , яка оцінює похибку інтерполяції тільки поточного кроку, тобто метод оцінної функції на постійній несучій частоті повинен більш точно відтворювати задану траєкторію руху, ніж інші методи оцінної функції.

6.3.5. Методи цифрових диференціальних аналізаторів

При інтерполяції за методом цифрових диференціальних аналізаторів (ЦДА) моделюється диференціальне рівняння відтвореної кривої, рішенням якого є задана відтворена крива.

Метою створення цього методу є підвищення точності відпрацювання заданих складних кривих (від 2 порядку і вище). Ідея полягає у зниженні порядку кривої заданого руху за рахунок диференціювання і наступної інтерполяції з подальшим інтегрування різницевого рівнянь, що задані координатами кінцевих точок. Наслідком цього є включення в загальну структурну схему розрахунків координат пристрою – диференціатора (рис. 6.25). Враховуючи те, що метод ЦДА призначено для підвищення точності відтворення складних кривих, його робота реалізована на постійній несучій частоті.

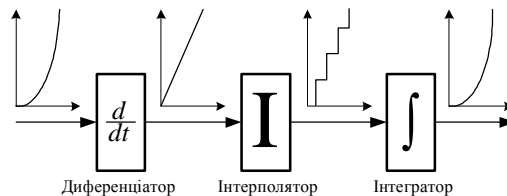


Рис. 6.25. Загальна структурна схема реалізації методу ЦДА
Загальний метод побудови диференціальних рівнянь, розв'язком

яких є задані криві, описаний В.С. Кулебакіним і названий методом К (D)-перетворення. Для найпростіших же траєкторій (прямої та кола) диференціальні рівняння можуть бути знайдені прямим способом.

Лінійна інтерполяція методом ЦДА. Відтворимо схему реалізації методу ЦДА для алгебраїчного рівняння прямої лінії:

$$y_i = x_i \frac{Y}{X},$$

де X, Y – кадрові прирости координат.

Продиференціюємо рівняння по x , перейдемо до системи параметричних диференціальних рівнянь координат відносно параметра $\omega = \frac{1}{\tau}$, проінтегруємо обидві частини рівнянь і отримаємо наступні вирази для координат точок траєкторії руху:

$$y = \frac{1}{T} \int_0^t \tilde{\Delta y} dt; \quad x = \frac{1}{T} \int_0^t \tilde{\Delta x} dt,$$

де T – період постійної частоти;

$$\tilde{\Delta x} = \frac{XT}{\tau}, \quad \tilde{\Delta y} = \frac{YT}{\tau} - \text{середні міжтактові прирости по}$$

відповідних координатах, що мають цілі та дробові частини, підраховані з високою точністю.

При обчисленні інтеграла скористаємося наближенням Ейлера і перейдемо до дискретного часу. Отримаємо наступні вирази для координат точок кожного циклу інтерполяції на постійній несучій частоті:

$$y_i = \sum_1^i \tilde{\Delta y}; \quad x_i = \sum_1^i \tilde{\Delta x}.$$

Таким чином, обчислення поточних координат x_i, y_i відтворюючої точки здійснюється додаванням середніх міжтактових приростів, а реалізацію методу ЦДА для лінійної інтерполяції можна провести за допомогою двох програмно чи апаратно побудованих регістрів-накопичувачів, що і є цифровим диференціальним аналізатором.

Наведені вище вирази продемонстрували методичну сутність побудови цифрового диференціального аналізатора.

В основу реального алгоритму лінійної інтерполяції за методом ЦДА покладені наступні залежності:

$$\overline{\Delta x_i} = \tilde{\Delta x} + \Delta_{i-1}^x; \quad \Delta x_i = \text{int}(\overline{\Delta x_i}); \quad \Delta_i^x = \overline{\Delta x_i} - \Delta x_i; \quad x_i = x_{i-1} + \Delta x_i;$$

$$\overline{\Delta y_i} = \tilde{\Delta y} + \Delta_{i-1}^y; \quad \Delta y_i = \text{int}(\overline{\Delta y_i}); \quad \Delta_i^y = \overline{\Delta y_i} - \Delta y_i; \quad y_i = y_{i-1} + \Delta y_i,$$

де $\overline{\Delta x_i}, \overline{\Delta y_i}$ – повні величини приростів координат, що містять дробову і цілу частини дискрет;

$\Delta x_i, \Delta y_i$ – цілі частини приростів координат, що призначені для видачі у i -му циклі інтерполяції на постійній несучій частоті;

Δ_i^x, Δ_i^y – накопичувачі дробових залишків координат;

int – функція виділення цілої частини, що полягає у відкиданні дробового залишку;

x_i, y_i – накопичувач координат.

Отже, алгоритм лінійної інтерполяції методом ЦДА полягає у проведенні паралельних розрахунків координат за наступними кроками:

– визначення повних приростів координат $\overline{\Delta x_i}, \overline{\Delta y_i}$ із врахуванням дробових залишків від попередніх циклів інтерполяції;

– визначення приростів координат $\overline{\Delta x_i}, \overline{\Delta y_i}$ та їх дробових частин Δ_i^x, Δ_i^y , які будуть використані в наступному інтерполяційному циклі;

– видача керуючих сигналів і визначення значень координат x_i, y_i .

В рамках алгоритму здійснюється перевірка кінця відпрацьовування кадру, тобто для останнього циклу інтерполяції координатні прирости розраховуються як:

$$\Delta x_{i_{\max}} = X - \sum_{i=1}^{i_{\max}-1} \Delta x_i; \quad \Delta y_{i_{\max}} = Y - \sum_{i=1}^{i_{\max}-1} \Delta y_i.$$

Таким чином, схему інтерполяції методом ЦДА можна подати як процес накопичення відхилень на кожному кроці циклів інтерполяції та вироблення корегуючого впливу (компенсації), коли це накопичення досягне значення однієї дискрети. Внаслідок цього інтерполяційна траєкторія буде пролягати переважно нижче заданої траєкторії.

Кругова інтерполяція методом ЦДА. Розглянемо за такою ж схемою побудову методу ЦДА для кругової інтерполяції траєкторії руху проти годинникової стрілки у першому квадранті.

Рівняння кола з радіусом R записується наступним алгебраїчним виразом:

$$x_i^2 + y_i^2 = R^2.$$

Продиференціюємо рівняння по x , перейдемо до системи параметричних диференціальних рівнянь координат відносно параметра

$\omega = \frac{V}{R} = \frac{\tilde{\Delta\varphi}}{T}$, проінтегруємо обидві частини рівнянь і отримаємо наступні вирази для координат точок траєкторії руху:

$$y = \frac{1}{T} \tilde{\Delta\varphi} \int_0^t x dt; \quad x = -\frac{1}{T} \tilde{\Delta\varphi} \int_0^t y dt,$$

де $\tilde{\Delta\phi} = \frac{VT}{R}$ – середній прирост кута повороту відтворюючої точки за період T постійної частоти інтерполятора.

При обчисленні інтеграла скористаємося наближенням Ейлера і перейдемо до дискретного часу. Отримаємо наступні вирази для координат точок кожного циклу інтерполяції на постійній несучій частоті:

$$y_i = \tilde{\Delta\phi} \sum_{k=1}^i x_k; \quad x_i = -\tilde{\Delta\phi} \sum_{k=1}^i y_k.$$

Від’ємний знак у виразі координати X показує на те, що абсолютне значення координати зменшується, а прирости координат отримують значення:

$$\Delta y_i = y_i - y_{i-1} = \tilde{\Delta\phi} x_i; \quad \Delta x_i = x_{i-1} - x_i = \tilde{\Delta\phi} y_i.$$

Наведені рівняння є описом цифрового диференціального аналізатора. Операція підсумовування виконується в кожному новому циклі періоду постійної несучої частоти. При цьому в регістр-накопичувач подається значення поточної координати відтворюючої точки.

Структурна схема зв’язків у цифровому диференціальному аналізаторі при круговій інтерполяції наведена на рис. 6.26. Звернемо увагу на те, що в регістрі-накопичувачі координати Y здійснюється підсумовування поточних координат x_i , а в регістрі-накопичувачі координати X – y_i .

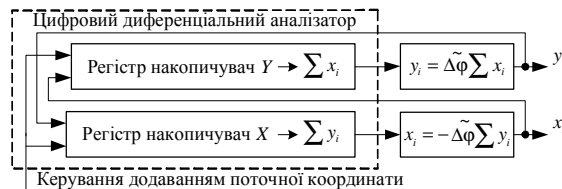


Рис. 6.26. Структурна схема зв’язків у цифровому диференціальному аналізаторі при круговій інтерполяції

Метод цифрових диференціальних аналізаторів трактують звичайно більш широко, відносячи до цього методу такі алгоритмічні структури з апаратно чи програмно реалізованими регістрами-накопичувачами, у рамках яких (мова йде про структури) відбувається підсумовування дуже точно підрахованих координат чи їх приростів, виділення для керування приводами цілих частин, накопичення дробових залишків тощо. Іншими словами, цифровий диференціальний аналізатор як пристрій, що моделює диференціальне рівняння відтвореної траєкторії, може в явному вигляді не виявлятися. Проте власне алгоритмічну структуру отримують на підставі суто геометричного підходу без залучення диференціальних форм.

Розглянемо один з варіантів кругової інтерполяції, що належить до методу ЦДА (рис. 6.27). Інтерполяція здійснюється на постійній

несучій частоті з періодом T . З кожним періодом постійної несучої частоти відбувається приріст $\tilde{\Delta\phi} = \frac{VT}{R}$ кута повороту відтворюючої точки, де використані уже введені раніше значення. Цей кут повороту може бути заданий його косинусом, а той, у свою чергу, підрахований на етапі підготовки інтерполяції за допомогою розкладу в ряд:

$$\cos \tilde{\Delta\phi} = 1 - \frac{\tilde{\Delta\phi}^2}{2!} + \frac{\tilde{\Delta\phi}^4}{4!} - \dots$$

Можна скористатися також табличними значеннями косинусів, збереженими в пам'яті пристрою ЧПУ.

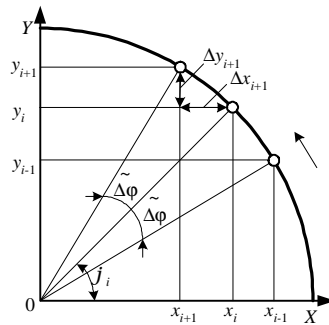


Рис. 6.27. Кругова інтерполяція методом ЦДА

Поставимо задачу визначення міжтактових приростів Δx_{i+1} , Δy_{i+1} геометричним шляхом:

$$x_{i+1} = R \cos(\phi_i + \tilde{\Delta\phi}) = 2x_i \cos \tilde{\Delta\phi} - x_{i-1};$$

$$\Delta x_{i+1} = x_i - x_{i+1} = -k_{\text{инт}} x_i + x_{i-1};$$

$$y_{i+1} = R \sin(\phi_i + \tilde{\Delta\phi}) = 2y_i \cos \tilde{\Delta\phi} - y_{i-1};$$

$$\Delta y_{i+1} = y_i - y_{i+1} = k_{\text{инт}} y_i - y_{i-1},$$

де $k_{\text{инт}} = 2 \cos \tilde{\Delta\phi} - 1$ – коефіцієнт кругової інтерполяції, який розраховується на початку кадру.

Наведені формули залучаються до рекурсивних процедур розрахунку чергових приростів на підставі попередніх значень поточних координат, що зберігаються і коригуються у регістрах-накопичувачах. Якщо врахувати, що кожне розраховане значення координат виконане з деякою похибкою, то з аналізу виразу для обчислення координат видно, що ця похибка входить два рази. Отже, необхідно запобігти використанню у розрахунках координат інших значень, крім попередніх.

Одним з таких варіантів модифікації методу ЦДА при круговій інтерполяції є метод прогнозу та корекції.

6.3.6. Метод прогнозу і корекції

Кожен цикл інтерполяції за методом прогнозу і корекції складається з двох етапів: на першому етапі виробляється груба оцінка (прогноз) координат наступної точки (вузла) інтерполяції, а на другому ці координати коригуються з метою підвищення точності та виключення можливості нагромадження похибок.

Таким чином, задача побудови алгоритму інтерполяції зводиться до вибору методу прогнозу та методу корекції координат наступного вузла. Загальна ідея алгоритму показана на рис. 6.28.

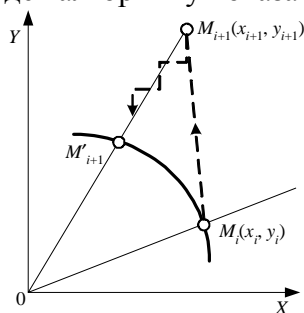


Рис. 6.28. Інтерполяція за методом прогнозу та корекції

На першому етапі за допомогою спрощеної модифікації методу ЦДА роблять прогнозувальний крок вздовж дотичної до траєкторії у поточній точці M_i :

$$\Delta x_{i+1} = R(\cos \varphi_i - \cos \varphi_{i+1}) = 2R \sin\left(\frac{\varphi_{i+1} + \varphi_i}{2}\right) \sin\left(\frac{\varphi_{i+1} - \varphi_i}{2}\right) \approx \tilde{\Delta \varphi} y_i;$$

$$\Delta y_{i+1} = R(\sin \varphi_{i+1} - \sin \varphi_i) \approx \tilde{\Delta \varphi} x_i;$$

$$x_{i+1} = x_i - \Delta x_{i+1} = x_i - \tilde{\Delta \varphi} y_i; \quad y_{i+1} = y_i + \Delta y_{i+1} = y_i + \tilde{\Delta \varphi} x_i.$$

Якщо точка $M_i(x_i, y_i)$ належить дузі кола радіуса R , то точка $M_{i+1}(x_{i+1}, y_{i+1})$ знаходиться поза колом радіуса R , оскільки перший наближений етап інтерполяції здійснюється вздовж дотичної до точки M_i .

Тому на другому етапі здійснюється лінійна інтерполяція поодинокими кроками в напрямку до центра O кола вздовж прямої $M_{i+1}O$. Інтерполяція закінчується при зміні знака оцінної функції F_K , початкове значення якої в точці M_{i+1} складає $F_K = x_{i+1}^2 + y_{i+1}^2 - R^2$.

Лінійна інтерполяція вздовж прямої $M_{i+1}O$ проводиться з використанням оцінної функції $F_L = y_{i+1}x - x_{i+1}y$, де x, y – координати поточної точки траєкторії лінійної інтерполяції. У процесі лінійної інтерполяції на кожному кроці по осі X (при $F_L < 0$) виконуються такі операції:

$$F_K = F_K - 2x + 1; \quad F_L = F_L + y_{i+1}; \quad x = x - 1.$$

Аналогічно на кожному кроці по осі Y (при $F_L > 0$ або $F_L = 0$) виконуються операції:

$$F_k = F_k - 2y + 1; \quad F_l = F_l - x_{i+1}; \quad y = y - 1.$$

Зміна знака функції F_k говорить про те, що точка інтерполяційної траєкторії ввійшла в коло радіуса R з відхиленням, що не перевищує однієї дискрети.

Різноманітність алгоритмів інтерполяції пояснюється необхідністю компромісу між прагненням до високої точності обчислень з одного боку, і бажанням вкластися в мінімальний за часом інтерполяційний цикл з іншого боку. Завершивши розгляд проблеми інтерполяції, перейдемо до чергової фази розв'язку геометричної задачі, що полягає в управлінні слідкуючими приводами подачі.

6.4. Управління формоутворенням деталей у слідкуючому приводі технологічного обладнання

Заключний етап (третя фаза) реалізації геометричної задачі полягає у відпрацюванні керуючих дискрет слідкуючими приводами подачі в формоутворюючій системі координат верстата. На рис. 6.29 представлена схема слідкуючого приводу, що є узагальненням достатньо довільної структури слідкуючих приводів, які застосовуються у пристроях ЧПУ.

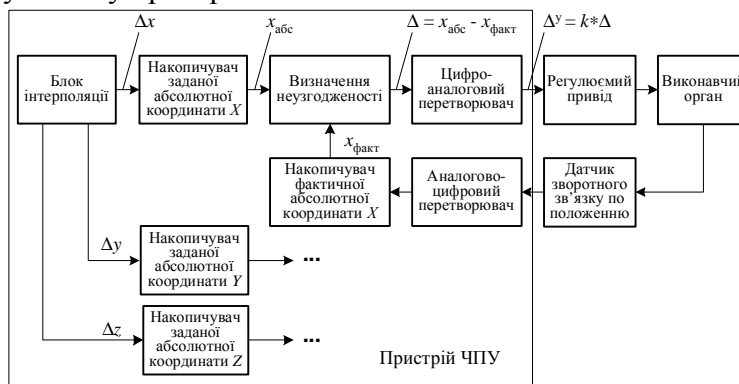


Рис. 6.29. Відпрацювання керуючих дискрет слідкуючими приводами подачі

Абсолютна задана координата $x_{\text{абс}}$ утворюється у накопичувачі абсолютної координати із врахуванням чергового завдання на відпрацювання Δx та встановленої дискретності слідкуючого приводу технологічного обладнання. Датчик зворотного зв'язку по положенню передає пряму (аналогову) інформацію про пересування виконавчого органу до аналогово-цифрового перетворювача (АЦП) для приймання і обробки сигналу в пристрої ЧПУ. Абсолютна фактична координата виконавчого органу $x_{\text{факт}}$ утворюється на підставі цих даних з АЦП та із врахуванням дискретності слідкуючого приводу.

Управління слідкуючим приводом здійснюється на підставі визначення різниці Δ значень абсолютних координат у блоці неузгодженості, перетворення цього кодового значення в аналоговий сигнал Δy , який і буде відпрацьований приводом подачі виконавчого

органу до тих пір, поки неузгодженість Δ не стане нульовою. В результаті слідкуючий привід буде приведений у рух і після пересування виконавчого органу на величину Δx датчик зворотного зв'язку по положенню зафіксує це у відповідному сигналі, і слідкуючий привід зупиниться.

Однак, крім виконання основної функції по управлінню формоутворенням деталей, слідкуючий привід подачі виконує ще й такі функції контролю:

- корекція систематичних похибок верстата;
- компенсація сталої похибки слідкуючого приводу;
- усунення перехідних динамічних похибок у каналі зворотного зв'язку.

Маючи на увазі те, що реалізація цих функцій здійснюється в системі ЧПУ програмно, умовна структурна схема блоку управління виконавчими слідкуючими приводами подачі зображена на рис. 6.30. Блок складається з таких компонентів: блока корекції систематичних похибок позиціонування; блока масиву констант та корекцій систематичних похибок; блока послідовної корекції, порівняльного пристрою.

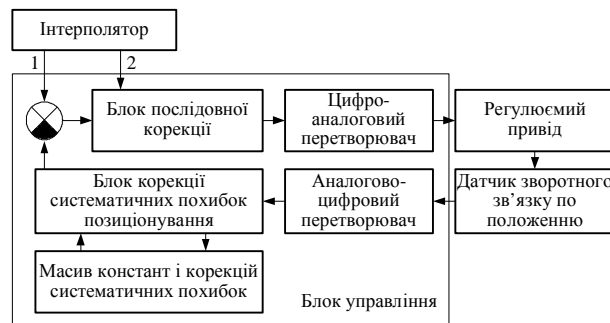


Рис. 6.30. Структурна схема блоку управління виконавчими слідкуючими приводами подачі

Інтерполятор працює на постійній несучій частоті й у кожному періоді виробляє керуючий сигнал Δx (для координати X), що надходить у вигляді $x_{\text{абс}}$ на вхід порівняльного блоку для порівняння його із сигналом зворотного зв'язку у вигляді скоригованого $x_{\text{факт}}$. Неузгодженість через блок послідовної корекції та аналого-цифровий перетворювач керує регульованим приводом, а сигнал зворотного зв'язку після проходження через аналого-цифровий перетворювач і блок корекції систематичних похибок повертається до порівняльного блоку.

Корекція систематичних похибок верстата. Систематична похибка виникає внаслідок механічних зазорів і допусків у виконавчих механізмах приводів подачі верстата (зазори в кулькових парах “гвинт – гайка”, опорних парах). Ідея корекції полягає в тому, що повне переміщення вздовж координатної осі розбите на інтервали Δ і для кожної k -ої точки на межі сусідніх інтервалів з визначеним значенням

абсолютної координати $x_{\text{кор}}$ заздалегідь встановлена необхідна корекція δ_k . Усі корекції зберігаються в масиві констант і корекцій.

Необхідність корекції виникає тоді, коли задана фактична абсолютна координата $x_{\text{абс}}$ виконавчого органу збігається з абсолютною координатою чергової точки корекції. Це відбувається, коли пройдений від попередньої точки корекції шлях Δs перевищує Δ .

Алгоритм корекції систематичних похибок показаний на рис. 6.31. За необхідності врахування впливу на похибку, що обумовлений змінним напрямком руху, в алгоритм додають такі процедури: визначення моменту зміни напрямку руху виконавчого органу, визначення знака зміни напрямку (з додатного напрямку на від'ємний чи навпаки), компенсації зазору.

Компенсація сталої похибки слідкуючого приводу. Стала швидкісна похибка слідкуючого приводу не залежить від запрограмованого шляху. Вона тим менша, чим більший коефіцієнт підсилення по швидкості. При постійній швидкості подачі стала швидкісна похибка постійна.

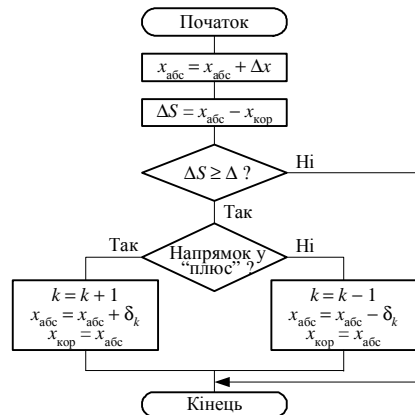


Рис. 6.31. Алгоритм корекції систематичної похибки

У зв'язку з цим компенсація сталої похибки можлива шляхом введення в контур управління через блок послідовної корекції додаткового сигналу Δv , пропорційного заданій швидкості $\frac{\Delta x}{T}$ (для координати X), де T – період постійної несучої частоти:

$$\Delta v = k_1 \frac{\Delta x}{T} = k_2 \Delta x,$$

де k_1 і k_2 – коефіцієнти пропорційності.

З формули видно, що компенсуючий сигнал формується шляхом множення результату інтерполяційних розрахунків на постійний коефіцієнт, який може бути заданий з панелі оператора, у тексті управляючої програми, у масиві констант або отриманий шляхом розрахунків.

Усунення перехідних динамічних похибок. Помітним джерелом динамічних перехідних похибок є стрибкоподібні зміни неузгодженості

приводу. Причина такої зміни пов'язана з роботою блока корекції систематичних похибок позиціонування, що миттєво змінює значення сигналу, який надходить з перетворювача зворотного зв'язку по положенню, при проходженні виконавчого органу через точку корекції. Різка зміна значення сигналу зворотного зв'язку викликає відповідний стрибок неузгодженості слідкуючого приводу. Це збуджує динамічний перехідний процес, що також впливає на неузгодженість, тобто динамічну похибку слідкуючого приводу.

Усунення подібних динамічних похибок здійснюють за допомогою спеціального алгоритму плавної ліквідації неузгодженостей, що стрибкоподібно змінюються. Цей алгоритм доцільно розмістити в блоці послідовної корекції. В основі алгоритму лежить наступна стратегія: найбільш сприятливим, з динамічної точки зору, є управління, при якому значення двох послідовно виданих на регульований привід неузгодженостей розрізняються на величину, що не перевищує 5% значення найбільшої неузгодженості.

Алгоритм усунення для координати X наведений на рис. 6.32.

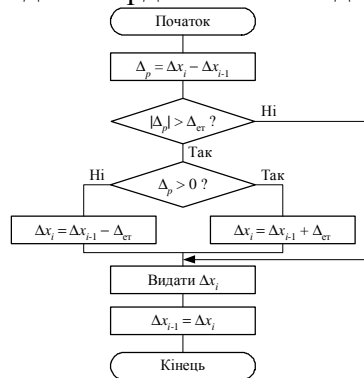


Рис. 6.32. Алгоритм усунення динамічних перехідних похибок

Зміст алгоритму складають такі процедури: визначення значення і знака різниці Δ_p неузгодженостей у сусідніх періодах: i -му і $(i-1)$ -му; порівняння отриманої різниці з еталоном $\Delta_{ст}$; корекція неузгодженості в i -му періоді, якщо різниця перевищує етalon, видача неузгодженості в аналого-цифровий перетворювач.

6.5. Реалізація логічної задачі в системах електроавтоматики пристроїв ЧПУ

6.5.1. Структура логічної задачі в системах електроавтоматики пристроїв ЧПУ

На сучасних верстатах, крім основних операцій обробки, підлягають автоматизації численні допоміжні операції, які умовно називаються операціями технологічного забезпечення. До цих допоміжних операцій відносяться: керування автоматичною зміною інструмента, керування переключенням у приводах подачі, пов'язане з

обмеженнями робочої зони; керування переключеннями у приводі головного руху; керування затискними пристроями, охолодженням, змащуванням, пересуванням огороження тощо. Всі ці функції виконуються системою циклової електроавтоматики, яка забезпечує підготовку до роботи, роботу верстата у заданих режимах, індикацію стану електрообладнання верстата і самої системи автоматики в усіх режимах, вихід з аварійних ситуацій, збереження інформації при відключенні живлення, захист електрообладнання тощо.

Таким чином, під *системою циклової електроавтоматики* розуміють систему автоматичного керування механізмами та групами механізмів, поведінка яких визначається множиною дискретних операцій з відношеннями слідування та паралелізму. Причому окремі операції ініціюються електричними керуючими сигналами, а умови їх зміни формуються під впливом інформаційних сигналів, які поступають з боку об'єкта управління.

Системи циклової електроавтоматики створюються для розв'язку *логічної задачі ЧПУ*.

Декомпозиція логічної задачі ЧПУ на складові наведена у табл. 6.1.

Таблиця 6.1

Декомпозиція реалізації логічної задачі ЧПУ

Рівень	Структурний компонент	Елементи реалізації
1	Логічна задача ЧПУ	Система циклової електроавтоматики верстата
2	Підзадачі ЧПУ	Управляючі приводи дискретних механізмів або груп механізмів верстата
3	Цикли ЧПУ	Окремі вузли дискретних механізмів верстата
4	Операції	Виконавчі двигуни вузлів

В результаті декомпозиції за рівнями реалізації функцій логічна задача розпадається на більше число незалежних, але пов'язаних між собою через взаємні блокування *підзадач*, де окрема підзадача описує циклічний процес деякого дискретного механізму (об'єкта) на верстаті.

Під *процесом циклової електроавтоматики* розуміють відповідним чином подану програму управління в реальному часі деяким дискретним механізмом або групою механізмів верстата.

Процес складається з окремих ситуацій, кожна з яких реалізує функцію управління в деякому стані керованого об'єкта. В конкретний момент часу в кожному процесі діючою (активною) є лише одна ситуація. Кожна ситуація складається з декількох фаз. В першій здійснюється прийом і підготовка даних, необхідних для виконання логічних дій і розгалужень. Далі виконується фактичний аналіз даних з

наступним прийняттям рішення. Потім виконуються дії у відповідності з прийнятим рішенням, але у цій фазі можливі варіанти.

Якщо зміна ситуації не передбачається, то ці дії будуть полягати у підтримці необхідних характеристик вихідних сигналів. Якщо прийняте рішення про перехід до нової ситуації, то будуть викликані дії, пов'язані із завершенням ситуації, які за суттю можуть бути віднесені до наступної ситуації, оскільки є частиною процесу визначення її початкового значення. За цим варіантом фаза завершується фактичним призначенням нової ситуації по даному процесу. Управління ситуаціями циклічно (синхронно) передається від однієї активної ситуації одного процесу до іншої активної ситуації іншого процесу. На цю процедуру можуть накладатися асинхронні процеси, які ініціюються за перериванням.

Взаємодія ставить перед процесом завдання по виконанню деякої нової функції. Завдання можуть надходити із зовнішнього середовища, від зовнішніх по відношенню до системи електроавтоматики процесів управління або від інших процесів системи електроавтоматики, наприклад: завдання початкового запуску всіх процесів електроавтоматики; завдання переведення в автоматичний режим роботи електроавтоматики; завдання на аварійну зупинку роботи електроавтоматики; завдання на відпрацювання функцій, пов'язаних із початком або завершенням управляючої програми ЧПУ; завдання на відпрацювання циклів-функцій управляючої програми ЧПУ тощо. Таким чином, всі складні циклічні процеси, що виконуються на верстаті з ЧПУ, можна подати у вигляді циклів автоматики по відпрацюванню типових операцій технологічного забезпечення.

Циклом автоматики верстата з ЧПУ називають фіксовану (незмінну) послідовність дій, яку викликають в управляючій програмі ЧПУ за ідентифікатором її типу. Цикл автоматики складається з операцій, причому під *операцією* можна розуміти будь-яку незалежну дію дискретного механізму, яка виконується одним двигуном, відкривається самостійним керуючим сигналом, а при закритті підтверджується або не підтверджується інформаційним (сповіщаючим) сигналом.

В управляючій програмі ЧПУ застосовують такі ідентифікатори (інформаційні слова кадру) для операцій технологічного забезпечення: “Функція подачі”, “Швидкість головного руху”, “Функція інструмента”, “Допоміжна функція”.

Інформаційне слово “Функція подачі” визначає результуючу швидкість подачі, яку записують під адресою F після усіх слів з розмірними переміщеннями. Якщо необхідно визначити складові цієї швидкості за координатами, то адреса F записується безпосередньо за розмірним переміщенням цієї координати. Розмірність швидкості подачі кодується в залежності від встановленої в кадрі підготовчої

функції: $G94$ – в мм/хв, $G95$ – мм/оберт.

Інформаційне слово “Швидкість головного руху”

починається з адреси S і визначає лінійну швидкість точки контакту інструмента із заготовкою або частоту обертання шпинделя (стола). Варіант подання розмірності швидкості головного руху кодується в залежності від встановленої в кадрі підготовчої функції: $G96$ – в м/хв, $G97$ – об/хв.

Значення швидкості визначається комбінацією цифр, що слідує за адресою інформаційного слова.

Для кодування швидкостей подачі та головного руху застосовують такі методи: прямого позначення, геометричної та арифметичної прогресій, символічний.

Метод прямого позначення є найбільш наочним і тому рекомендованим: швидкість подачі 20 мм/хв записується чотиризначним числом $F0020$, слово $S0800$ позначає виклик циклу, який встановлює частоту обертання 800 хв^{-1} .

При кодуванні *методом геометричної прогресії* швидкість подачі та головного руху умовно позначають двозначним кодом в діапазоні від 00 до 99, а дійсні значення утворюють геометричну прогресію: 0; 1.12; 1.25; 1.40; ...; 80 000.

Кодування *методом арифметичної прогресії* передбачає подання швидкостей подачі та головного руху три-, чотири- або п'ятизначними кодовими числами. Перша цифра кодового числа представляє собою десятковий множник, значення якого на 3 більше кількості цифр ліворуч від коми у значенні швидкості, що кодується. Якщо число, що кодується, менше за 1, то перша цифра кодового числа утворюється вирахуванням з числа 3 кількості нулів, що стоять праворуч від коми у значенні швидкості. Наступні цифри кодового числа позначають значення швидкості, округлене з точністю до двох, трьох або чотирьох знаків. Наприклад, значення швидкості 150,3 кодується тризначним кодом 615, чотиризначним – 6150 та п'ятизначним – 61503.

Символічний метод кодування передбачає позначення швидкостей подачі та головного руху умовним числом з двох або більше розрядів. Сутність цього методу подається у вигляді таблиці відповідності в інструкції з програмування для конкретного верстата з ЧПУ.

Інформаційне слово “Функція інструмента”

використовується для призначення інструмента і його коректора (для корекції траєкторії руху із врахуванням радіуса і довжини інструмента). Функція інструмента задається кодовим числом під адресою T з однією або двома групами цифр. У першому випадку слово вказує тільки номер викликаного інструмента або його позиції, а номер коректора для цього інструмента визначається іншим словом з адресою D . У другому випадку друга група цифр визначає номер коректора

довжини, положення або діаметра інструмента. Наприклад, у слові T1218: T – адреса; 12 – номер інструмента; 18 – номер коректора. Якщо програмується номер інструмента без вказання номера коректора, то друга група цифр містить нулі, наприклад, T1200. Якщо програмується номер коректора для заданого в одному з попередніх кадрів інструмента, то нулі містить перша група цифр, наприклад, T0018.

Інформаційне слово “Допоміжна функція” визначає різноманітні команди цикловим механізмам електроавтоматики верстата, а також самому пристрою ЧПУ.

Допоміжні функції задаються словами з адресою M і умовним двозначним кодом в діапазоні від 00 до 99. Деякі загальноприйняті допоміжні функції наведені у розділі 7.

Інші допоміжні функції визначають при створенні конкретного верстата і конкретного пристрою ЧПУ. В одному кадрі може бути записано в порядку зростання кодових номерів декілька команд різним виконавчим механізмам верстата.

Дискретні механізми верстата можна поділити на наступні функціональні групи: привід головного руху і шпіндель; дискретна група у приводах подачі; механізм автоматичної зміни інструмента; механізм автоматичної зміни заготовки; затискні пристосування верстата; механізм прибирання стружки; система охолодження; люнети; задня бабка і піноль токарного верстата та інші. Окремі механізми можуть виконувати не один, а декілька взаємовиключаючих циклів, які називаються ортогональними. Як приклади ортогональних груп циклів можна навести групу циклів зміни діапазонів частоти обертання шпинделя (M38, M39), групу циклів вибору напрямку обертання і зупинення шпинделя (M03...M05). Цикли пошуку інструмента (T00...T99) і цикли включення частоти обертання шпинделя (S00...S99) також представляють подібні ортогональні групи.

Таким чином, загальна структура допоміжного технологічного забезпечення побудована за ієрархічною схемою: функціональні групи дискретних механізмів, ортогональні групи циклів автоматики для кожного механізму, операції кожного окремого циклу автоматики.

6.5.2. Організація роботи дискретних механізмів автоматичної заміни інструмента у верстатах з ЧПУ

Розглянемо дискретний механізм автоматичної заміни інструмента (рис. 6.33). Автоматична заміна здійснюється у рамках двох незалежних циклів: циклу пошуку потрібного інструмента у магазині, який викликається інформаційним словом, що складається з адреси T та коду розшукуваного інструмента, і циклу заміни інструмента, який викликається інформаційним словом M06. Цикл пошуку нового інструмента зазвичай суміщений з обробкою попереднім інструментом.

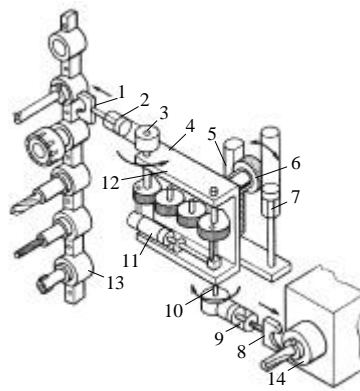


Рис. 6.33. Дискретний механізм автоматичної заміни інструмента

Цикл $M06$ організований наступним чином. У вихідному положенні захвати 1 та 8 автооператора 12 знаходяться на невеликій відстані відповідно від магазину 13 та шпинделя 14. За командою $M06$ на зміну інструментів обидва захвати одночасно подаються за допомогою гідроциліндрів 2 та 9 до інструментальних оправок та захоплюють їх (у кожному захваті є важіль-заскок). Після цього важелі з гідроциліндрами захватів синхронно обертаються гідроциліндром 11 навколо вертикальних осей 3 та 10, витягуючи оправки з інструментами з магазину та шпинделя. Для зміни інструмента корпус 4 автооператора обертається на 180° за допомогою зубчастого колеса 6 та рейок, зв'язаних з гідроциліндрами 5 та 7. Після цього важелі захватів, обертаючись навколо своєї осі, вводять інструменти у магазин та шпиндель.

Інструментальні магазини оснащують реверсивними приводами і пошук інструмента в цьому випадку доцільно вести за найкоротшим шляхом. Ці магазини в загальному випадку можуть бути зображені у вигляді кола з розбиттям на задану кількість позицій відповідно до кількості застосованих інструментів. Вибір напрямку обертання здійснюється із результатів аналізу номера позиції інструмента N_z , заданого в інформаційному слові з адресою T , і номера робочої N_ϕ позиції інструментального магазину.

Розробці системи циклової електроавтоматики передую формальний опис дискретних механізмів в трьох режимах: автоматичному, налагоджувальному (ручного управління), нерегулярних ситуацій. В описах повинна бути передбачена можливість переходу з режиму у режим шляхом виділення станів механізму, що допускає такий перехід.

За командою з адресою T на заміну інструмента ініціюється автоматичний цикл, що починається з аналізу коду N_ϕ робочої позиції револьверної головки. Якщо цей код збігається з кодом позиції N_z , у якій знаходиться викликаємий інструмент, заданий в управляючій програмі ЧПУ, то підтверджується виконання команди. У противному

випадку за допомогою електромагніту *ЕМ1* проводиться розтискання револьверної головки, що підтверджується кінцевим вимикачем *В1*. Далі повинна бути викликана обчислювальна процедура, що визначає напрямок обертання головки до заданої позиції, після чого включаються відповідно електромагніти *ЕМ2* (поворот проти годинникової стрілки) або *ЕМ3* (поворот за годинниковою стрілкою). При будь-якому проходженні позиції можливої фіксації спрацьовує кінцевий вимикач *В2* і здійснюється перевірка збігу кодів $N_3 = N_{\phi}$. Якщо збігу не відбудеться протягом заздалегідь встановленої тривалості часу, то це буде свідчити про відмовлення, що буде оброблене у режимі нерегулярної ситуації. Збіг кодів означає досягнення заданої позиції, у якій відключаються електромагніти *ЕМ2* або *ЕМ3* в залежності від напрямку обертання, що завершується. Потім виключається електромагніт *ЕМ1*, а головка затискається. Підтвердженням нормального затискання буде сигнал від кінцевого вимикача *В3*.

Логічна задача ЧПУ саме складається з подібних процесів. Як ілюстрацію цього виділимо у складі розглянутої задачі автоматичної заміни інструмента три паралельних процеси: автоматичне відтворення циклу зміни інструмента, затискання-розтискання револьверної головки, поворот револьверної головки.

Таким чином, існує можливість незалежної розробки численних окремих циклічних процесів із встановленням згодом необхідного зв'язку між ними.

6.5.3. Модель розв'язання логічної задачі в системах електроавтоматики пристрою з ЧПУ

Узагальненою моделлю розв'язання логічної задачі ЧПУ може слугувати функціональний автомат, показаний на рис. 6.34. Під функціональним автоматом розуміють автономний пристрій (комплекс пристроїв) із чітко вираженими цільовими функціями роботи, у якому можна виділити виконавчу частину (операційний автомат) і керуючу частину (управляючий автомат). Операційний та управляючий автомати обмінюються інформаційними сповіщуючими (типу зворотного зв'язку) та керуючими сигналами між собою та із зовнішнім середовищем.

При розв'язку логічної задачі ЧПУ як операційний автомат виступає сукупність виконавчих циклових механізмів об'єкта (верстата), а функції управляючого автомата виконує система циклової електроавтоматики. Багато двійкових чи цифрових інформаційних сигналів x (від кінцевих та шляхових вимикачів, пристроїв відображення параметрів), які надходять з боку об'єкта, визначають його поточний стан. Багато керуючих сигналів y (типу "увімкнути-вимкнути") підтримують стан об'єкта незмінним чи переводять об'єкт у новий стан. Система циклової електроавтоматики має і зовнішній

інтерфейс, отримуючи ззовні від інших керуючих пристроїв команди управління (коди викликаємих циклів), інформуючи інші пристрої управління про свої стани (про виконання отриманого завдання).

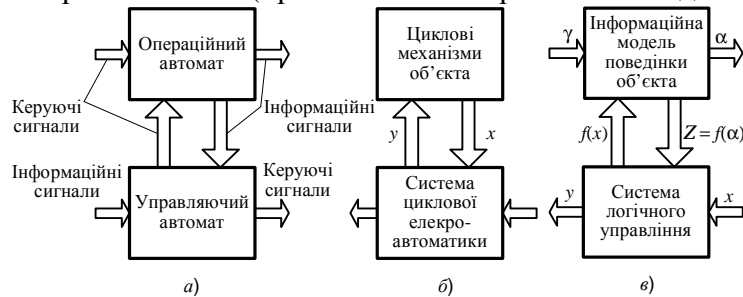


Рис. 6.34. Узагальнена модель розв'язання логічної задачі ЧПУ:
а – структурна схема функціонального автомата; б – модель розв'язання логічної задачі ЧПУ; в – функціональний автомат системи циклової електроавтоматики

Внутрішня структура системи циклової електроавтоматики також побудована за типом функціонального автомата. Роль операційного автомата відіграє інформаційна модель поведінки об'єкта, тобто модель відтворених циклів автоматики, а роль управляючого автомата – система логічного управління. Як керуючі сигнали виступають логічні функції x , що визначають поточний стан α відтворених циклів автоматики в інформаційній моделі і переходи між станами цієї моделі, а як інформаційні сигнали – назви викликаних операцій або процедур $\{Z\}$, які є функціями сталих станів a циклів автоматики $\{Z\} = f(\alpha)$. В подальшому назви $\{Z\}$ транслюються у зовнішні керуючі сигнали y . Деякі стани α інформаційної моделі (початкові, заключні стани циклів) можуть бути використані у зовнішніх пристроях управління. Зовнішні керуючі сигнали γ надходять з метою початкової ініціалізації циклів.

Загальна конфігурація системи управління цикловою автоматикою показана на рис. 6.35. У цю конфігурацію входять: технологічний пульт верстата, система циклової електроавтоматики, об'єкт (представлений компонентами системи управління на верстаті), пристрій ЧПУ.

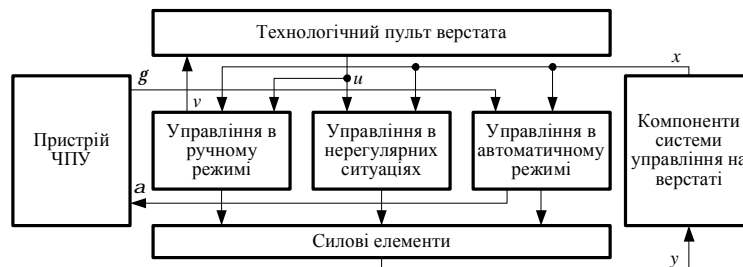


Рис. 6.35. Загальна конфігурація системи управління цикловою автоматикою

У автоматичному режимі система управління електроавтоматикою отримує від пристрою ЧПУ команди γ на запуск

циклів, а сигнали α сповіщають пристрій ЧПУ про завершення циклів на верстаті.

У ручному режимі оператор за допомогою органів управління технологічного пульта або панелі управління пристроєм ЧПУ викликає окремі переходи циклів або повні циклі командами u .

Реакція системи циклової електроавтоматики на нерегулярні ситуації (зміна режиму, натискання кнопки аварійної зупинки, тайм-аут, скидання шляхових обмежувальних вимикачів тощо) проявляється у зупинці верстата на поточній операції при переводі у заданий стан.

У всіх режимах система циклової електроавтоматики діє на об'єкт командами u . В результаті керуючої дії виконуються допоміжні операції технологічного забезпечення, а в міру завершення окремих етапів цих операцій змінюються значення сповіщуючих сигналів x .

Обробка операцій на верстаті візуалізується за допомогою засобів індикації по сигналах v , які є підмножиною множини u .

Тепер розглянемо фази розв'язання логічної задачі ЧПУ у автоматичному режимі. Початкова фаза полягає у введенні управляючої програми у пам'ять пристрою ЧПУ та подальшої її активізації. Логічна задача подана в управляючій програмі ідентифікаторами (кодами) циклів, що викликаються. Сам же опис циклів, як це витікає з попередніх міркувань, вже заздалегідь, на рівні з'єднання системи управління з верстатом, введений у пам'ять та постійно там зберігається. Друга фаза починається з безпосереднього виклику циклів, що проводиться з робочого кадру управляючої програми. З цього моменту починається обчислювальний процес, який складається з аналізу сповіщуючих сигналів, виконання переходів по циклу з одночасним викликом необхідних операцій та процедур, видачі керуючих сигналів на верстат, тобто третя фаза складається з реалізації окремих операцій, процедур та іншого безпосереднього управління.

Ступінь складності логічної задачі оцінюють посередньо за числом входів-виходів системи циклової електроавтоматики. Тому існує декілька варіантів реалізації як самої системи циклової електроавтоматики, так і її взаємодії з пристроєм ЧПУ:

- систему будують у вигляді спеціалізованого логічного автомата і розміщують автономно від пристрою ЧПУ. Реалізація системи проводиться на контактних реле або за допомогою програмованих контролерів;

- система вбудована у пристрій ЧПУ. Інтерфейс з пристроєм ЧПУ забезпечується за допомогою програмно-апаратних засобів. Реалізація системи проводиться як мікропрограмний автомат;

- система є одним з процесорів мультипроцесорного пристрою ЧПУ;

- функції системи покладають на пристрій ЧПУ із взаємодією на програмному рівні.

6.6. Реалізація технологічної задачі в системах управління ефективністю обробки пристроїв ЧПУ

Технологічна задача виникає в системах ЧПУ тоді, коли робочий процес стає об'єктом управління з метою оптимізації його параметрів (визначення подачі та швидкості різання) для досягнення заданої на виконавчому рівні управління мети (збільшення продуктивності, зниження витрат, підвищення якості обробки).

Успішність розв'язання технологічної задачі пов'язана з визначенням таких параметрів, як якість обробки деталей і ефективність процесу їх виготовлення. В результаті технологічна задача полягає в досягненні необхідної якості деталей з найменшими витратами при ефективній обробці.

До основних показників якості деталей відносять: точність обробки, стан поверхневого шару оброблюваної деталі.

Під точністю деталі розуміють ступінь її наближення до геометрично правильного прототипу, включаючи врахування макроегеометрії, а також хвилястість та шорсткість.

Основою реалізації технологічної задачі в технологічному обладнанні з ЧПУ є контрольно-вимірювальна система ГВМ.

6.6.1. Контрольно-вимірювальна система ГВМ

Об'єкти, етапи і функції контролю. Контрольно-вимірювальну систему застосовують для контролю деталі, інструменту, основного робочого процесу і окремих підсистем ГВМ.

Основне призначення контрольно-вимірювальної системи – забезпечення безперебійності робочого процесу, запобігання аваріям і пошкодженням, контроль за якістю, поточна діагностика і накопичення статистичної інформації.

Для досягнення вказаних цілей контрольно-вимірювальні функції виконуються перед обробкою, під час обробки, між окремими операціями робочого процесу і після завершення обробки, тобто можна структурувати виконувані функції контролю в ГВМ за об'єктами контролю та за різними етапами робочого процесу.

Отже, об'єкти контролю – це деталі, інструменти, робочий процес, а етапи контролю – це фази перед, під час і після завершення виконання процесу обробки.

Розглянемо суть функцій контролю.

При контролі деталей перед обробкою повинні бути ідентифіковані палета і заготовка, проконтрольоване просторове розташування заготовки, введені відповідні корекції та встановлений зв'язок систем координат верстата і деталі, виміряні допуск і твердість для визначення числа проходів; **під час обробки і між операціями** повинні бути забезпечені контроль якості (перевірені розміри, допуски, відхилення форми) і розрахунок необхідних корекцій; **після завершення обробки** повинен бути виконаний остаточний контроль якості та

враховані всі дані, що використовуються при накопиченні статистичної інформації і встановленні трендів.

При контролі інструменту перед обробкою повинні бути проконтрольовані правильність вибору інструменту, ступінь зношення і стан робочої кромки, визначені необхідні корекції на довжину і діаметр інструменту; **під час обробки** повинні здійснюватися контроль непередбаченого зношування інструменту і спостереження за ресурсом стійкості, контроль контакту з деталлю, контроль поломки інструменту (з оцінкою браку, як виправного, так і невиправного), контроль режимів, недопустимих вібрацій, зливної стружки; між операціями і після завершення обробки необхідно встановлювати неочікувані зношування і поломку інструменту, фіксувати режими, як проходить зливна стружка.

При контролі основного робочого процесу перед обробкою повинен бути забезпечений контроль наявності заготовки, інструменту, управляючої програми, моделі робочого процесу; **під час обробки і між операціями** необхідно здійснювати збір і накопичування інформації про протікання робочого процесу (діагностичні дані, відмови, корекції, зупинки); **після завершення обробки** повинні бути проаналізовані причини появи будь-яких негативних чинників, враховані всі необхідні дані, що беруть участь в накопиченні статистичної інформації та визначенні трендів. Для окремих підсистем ГВМ постійно або періодично повинні бути забезпечені збір, накопичення і аналіз діагностичної інформації, формування моделей функціонування, локалізація аварій та збоїв, прогнозування відмов, своєчасне залучення резервів.

Щодо перерахованих функцій контрольно-вимірювальної системи, то можна сказати наступне. По-перше, наведений перелік ні в одній відомій ГВМ у всьому обсязі не реалізований. По-друге, скільки-небудь помітна тенденція до централізації контрольно-вимірювальних функцій в рамках єдиної підсистеми не виявляється. Навпаки, значна частина вказаних функцій виконується засобами системи ЧПУ і засобами системи управління електроавтоматикою ГВМ. Інша частина контрольно-вимірювальних функцій виконується спеціальними засобами: системою ідентифікації та розпізнавання образів; моніторною системою контролю інструменту; системою адаптивного управління процесом різання; контрольно-вимірювальним автоматом.

Таким чином, контрольно-вимірювальні функції в ГВМ є децентралізованими.

Переважна частина контрольно-вимірювальних функцій в існуючих ГВМ пов'язана з двома класами функцій відповідно до показників якості: обробкою геометричної і розмірної інформації та обробкою інформації про силові параметри процесу різання.

Отже, при реалізації технологічної задачі виділяють дві основні фази: управління точністю обробки і управління ефективністю

обробки. Для підтримки виконання цих функцій в системі ЧПУ задіяні спеціальні засоби і процедури їх здійснення.

Організація контрольно-вимірювальних циклів контролю. Контрольно-вимірювальні функції виконуються спеціальними вимірювальними головками спільно із слідкуючими приводами подачі і пристроєм ЧПУ. Прикладом такої головки може бути вимірювальна головка, показана на рис. 6.36.

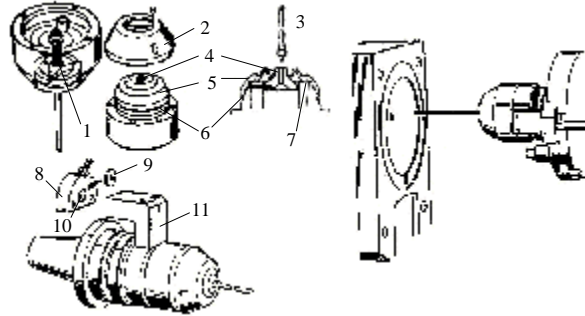


Рис. 6.36. Конструкція вимірювальної головки з індуктивним зв'язком:

1 – механічний тригер; 2 – верхня захисна кришка; 3 – щуп; 4, 6 – пружинні кільця; 5, 7 – зовнішня та внутрішня діафрагми; 8 – модуль прийому вимірювального сигналу, встановлений на верстаті; 9 – юстирувальна шайба; 10 – котушки індуктивності; 11 – модуль передачі вимірювального сигналу

Головка не має яких-небудь шкал, а вимірювальний сигнал формується в певному (нульовому) положенні щупа при миттєвому переключенні “механічного тригера”. Інформація про переключення передається в різних модифікаціях головки одним з трьох способів: дистанційно за допомогою індуктивного зв'язку; дистанційно за допомогою оптичного зв'язку в інфрачервоному діапазоні частот (в цьому випадку головка має вбудовану батарею живлення); за допомогою дротяного зв'язку.

Вимірювальні головки перших двох модифікацій зберігаються в інструментальному магазині. У вимірювальному циклі контролю деталей головки автоматично перевантажуються у шпиндель точно так, як і це робиться з інструментом (кінцевий хвостовик головки виконаний відповідно до того ж стандарту, що і хвостовик інструментального налагодження).

Третя модифікація призначена для встановлення на станині, столі або коробці верстата. Її використовують для вимірювання інструменту, закріпленого у шпинделі або різцетримачі.

Слідкуючий привід під управлінням пристрою ЧПУ організує наступні відносні рухи вимірювальної головки у напрямку контрольованої поверхні (рис. 6.37): I – вимірювальна головка в робочій позиції; II – підведення до контуру з прискореною подачею і зупинка руху по сигналу нульового переходу; III – відведення головки

зі зменшеною подачею до звільнення щупа, зупинка подачі; IV – повторне підведення з малою подачею і остаточна зупинка по сигналу нульового переходу. В результаті вимірювальна головка генерує сигнал нульового переходу, а пристрій ЧПУ зупиняє слідкуючий привід і знімає дані зі шкали перетворювача зворотного зв'язку приводу подачі. Зняті дані після обробки зберігаються в пам'яті пристрою ЧПУ як корекції, зсуви або дані, що використовуються в умовних переходах. Вимірювальні цикли можуть бути стандартними або розробляються як підпрограми.

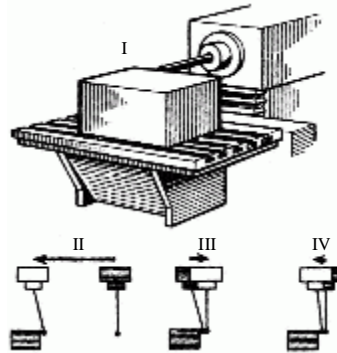


Рис. 6.37. Реалізація вимірювальною головою контрольно-вимірювальних функцій

До типових задач, що вирішуються за допомогою вимірювальних головок, відносяться: ідентифікація заготовок; виявлення зсуву положення заготовок або пристосувань; корекція положення або перетворення координатної системи; вимірювання заздалегідь оброблених або готових деталей для внесення необхідної корекції в управляючу програму ЧПУ.

Найпростіший спосіб ідентифікації полягає у перевірці щупом чітких ознак розпізнавання на палеті або заготовці. Можна також виконувати обстеження певних характерних поверхонь за наперед розробленими підпрограмами. Для контролю фіксації палети і розпізнавання її положення необхідні дотики щупа в декількох певних точках (рис. 6.38). Коректування положення виконується автоматично пристроєм ЧПУ, причому воно зводиться до паралельних перенесень і кутових поворотів координатної системи.

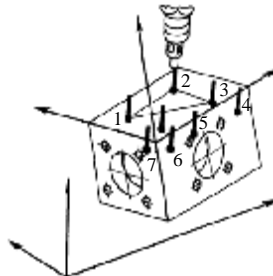


Рис. 6.38. Розпізнавання положення точок у просторі

Для контролю заздалегідь оброблених або готових деталей

розробляють різноманітні вимірювальні цикли, у складі яких елементи управління рухами і розрахункові процедури дуже складні (табл. 6.2).

За допомогою тих же вимірювальних головок (третя модифікація) виконується контроль інструменту. При цьому вирішують задачі трьох класів. Задача першого класу полягає у встановленні початкового положення різальної кромки інструменту в конкретній інструментальній наладці і передачі відповідних даних в таблицю корекцій, що зберігається в пам'яті пристрою ЧПУ. При цьому відпадає необхідність в тонкому попередньому настроюванні інструменту на спеціальних високоточних приладах поза верстатом (рис. 6.39). Задачі другого класу пов'язані з епізодичною оцінкою інструменту і внесенням змін в таблицю корекцій. Задачі третього класу полягають у перевірці цілісності інструменту за допомогою датчиків, встановлених на столі верстата.

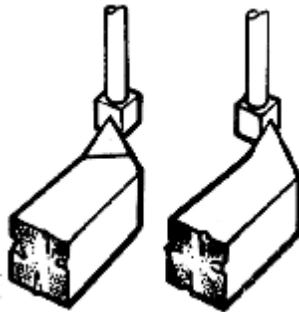


Рис. 6.39. Контроль різальної кромки інструмента

Датчики контролю цілісності інструмента зазвичай безконтактні: інструмент наближають у циклі контролю до датчика без дотику. Робота датчика може базуватися на різних фізичних ефектах, що забезпечують достатню чутливість. Наприклад, поломку зубів фрези виявляють так: фрезу повільно обертають перед датчиком, кількість сигнальних “сплесків” за один оберт повинна дорівнювати числу зубців, а рівень сигналів повинен бути однаковим.

Поряд з універсальними контрольно-вимірювальними засобами, що забезпечують одержання геометричної та розмірної інформації, деяке поширення знаходять пристрої спеціального характеру. До числа подібних належить, наприклад, вимірювальний пристрій, що використовують для автоматичного внесення корекції у програму токарної обробки через систему ЧПУ.

Вище розглянуті деякі контрольно-вимірювальні функції, пов'язані з обробкою і використанням геометричної та розмірної інформації.

Питання контролю та вимірювань, що належать до сфери обслуговування інструмента і спостереження за процесом різання, заслуговують самостійного аналізу.

Таблиця 6.2
Типові вимірювальні цикли проміжного або остаточного контролю деталей

Вимірювальне завдання	Спосіб виконання завдання
Контроль лінійного розміру	
Визначення центра кола, його діаметр	
Визначення точки перетину, лінії перетину	
Визначення площини, визначення відхилень від площини	
Відхилення від круглості, паралельності, перпендикулярності	

Системи обслуговування інструмента. До основних форм обслуговування інструмента (крім його ідентифікації та корекції за результатами прямого вимірювання, про які говорилося вище) належать: контроль граничної стійкості за часом; контроль стану (тобто поточної точності) на основі непрямих оцінок,

експериментальних даних розрахункових моделей; контроль поломок. Спостереження за процесом різання створює можливість для адаптивного керування – за обмеженнями або із залученням деякого критерію оптимізації.

Інструмент у ГВМ є найбільш слабкою ланкою, тому той чи інший різновид його контролю застосовується обов'язково. У найпростішому випадку співставляється фактичний час роботи інструмента з його нормативною стійкістю. При цьому можливі такі варіанти контролю: за часом різання, який у цьому випадку варто якось визначати, за часом обертання шпинделя разом з даним інструментом, розрахункова стійкість якого задається числом оброблених деталей. Якщо нормативний ресурс стійкості вичерпаний, інструмент примусово замінюється на резервний без врахування його фактичного стану. Аналогічним способом здійснюється контроль стійкості для багатоінструментальних головок і насадок.

Недоліком цього способу є помилки при призначенні нормативу, що значно коливається в залежності від якості виготовлення інструмента, стабільності властивостей оброблюваного матеріалу, сталості припуску на обробку тощо.

Найбільш розповсюдженим способом контролю поточного стану інструмента є вимірювання якого-небудь силового фактора процесу різання. Такий спосіб вимагає попереднього навчання системи керування контролем (ці системи часто називають моніторами).

У процесі навчання при роботі новим інструментом за налагодженою управляючою програмою ЧПУ в пам'ять монітора надходять дані про фактичні умови різання. Потім у виробничому циклі ці дані служать орієнтиром для оцінки фактичного стану інструмента. На вибір технолога встановлюють одне з двох можливих обмежень: неперевищення максимального значення навантаження сил різання, невихід за верхні та нижні межі допустимих відхилень від номінального (експериментального) значення навантаження.

Поломку інструмента виявляють різними способами. Так, якщо в пам'яті монітора зберігається значення часу врізання, то фактичне збільшення цього часу проти збереженого в пам'яті (що визначається за моментом сплеску навантаження різання порівняно з навантаженням холостого ходу) може означати поломку. Точно так само про поломку можна судити в тих випадках, коли фактичний час різання (визначений за допомогою датчика контакту з деталлю) менший за еталонний, встановлений при навчанні. Однак на точність зазначених двох способів впливають флуктуації подачі та коливання припуску.

Інший спосіб фіксації поломки полягає в тому, що на підставі даних навчання встановлюють дві верхні межі відхилення навантаження різання від номінального значення, одна з яких відповідає максимально допустимому зношуванню, а друга – руйнуванню інструмента. Перспективним напрямком у діагностиці

поломок є виявлення у навантажувальних характеристиках різання топологічно єдиних закономірностей, що відповідають моменту руйнування і навіть попередньому до нього моменту. У тих випадках, коли такі закономірності існують, діагностування поломок зводиться до задачі розпізнавання образу, тобто до задачі виділення об'єкта в тріаді об'єкт – псевдооб'єкт – шум. Найпростішим прикладом може бути осцилограма обертального моменту сили різання при токарній обробці, що оцінюється по струму двигуна $I_{дв}$ (рис. 6.40).

Відомо, що при поломці різця виникає невеликий спад сили струму, пов'язаний з холостим ходом протягом декількох обертів шпинделя, а потім сила струму збільшується на ΔI_1 внаслідок різання тупою кромкою. Якщо сигнал з датчика обертаючого моменту відфільтрувати від шумів, то збільшення сили струму ΔI_1 , пов'язане з поломкою (об'єкт), легко помітне на тлі збільшення сили струму ΔI_2 у зв'язку із зростанням глибини різання (псевдооб'єкт).

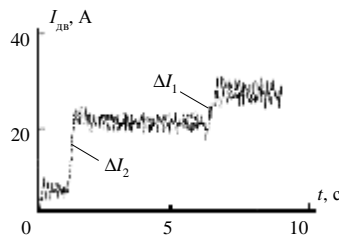


Рис. 6.40. Розпізнавання поломки інструменту

Великий інтерес викликають віброакустичні методи контролю інструмента, що базуються на використанні високочастотних акселерометрів у частотному діапазоні, верхня межа якого перевищує 100 кГц. За потужністю акустичної емісії можна робити висновки про зношування інструмента. Спеціальні методи обробки сигналу датчика акустичної емісії дозволяють встановити поломку інструмента.

6.6.2. Формування заданої точності обробки деталей

Розглянемо механізм формування заданої точності обробки у зв'язку з особливостями початкового встановлення, статичного і динамічного настроювань деталі.

При розміщенні оброблюваної деталі в робочому просторі верстата (тобто при включенні деталі в кінематичні та розмірні ланцюги системи верстата) необхідно забезпечити *початкове встановлення*, тобто керування точністю початкового положення деталі відносно баз верстата чи пристосування. Для цього заготовку орієнтують відповідним чином на столі або у пристосуванні верстата. Комплект технологічних баз, що визначають положення деталі у процесі її обробки, утворює координатну систему деталі. Поверхні стола, пристосування або інших компонентів, за допомогою яких деталь координують у робочому просторі, складають комплект баз верстата, що формують його координатну систему.

Шість теоретичних опорних точок, необхідних для визначення положення деталі, обумовлюють накладені на деталь зв'язки, завдяки яким виключається можливість переміщення деталей у напрямку координатних осей і повороту відносно цих осей. Якщо відомі координати опорних точок контакту в координатній системі верстата, то похибку встановлення деталі можна визначити розрахунковим шляхом. Але похибка встановлення деталі, як правило, не може бути визначена розрахунковим шляхом, тому після встановлення обстежують поверхні деталі за допомогою вимірювальних головок у рамках спеціальних автоматичних вимірювальних циклів. За результатами вимірювань похибки встановлення деталі вводяться у пам'ять пристрою ЧПУ, і таким чином координатні системи верстата і деталі виявляються узгодженими, а управляюча програма ЧПУ в системі координат деталі – придатної для свого відтворення в координатній системі верстата.

Статичне настроювання деталі в робочому просторі верстата – це процес первинного встановлення точності відносного руху і положення виконавчих поверхонь інструмента, устаткування та пристосування для отримання необхідної точності оброблюваної деталі. Іншими словами, статичне настроювання полягає в узгодженні на рівні управління вже трьох координатних систем: верстата, деталі, інструмента. Параметри узгодження зберігають звичайно у вигляді коректур інструмента в пам'яті пристрою ЧПУ. Під коректурами розуміють координати виконавчих поверхонь інструмента в системі координат верстата.

В процесі обробки деталей встановлена первісна точність відносного руху і положення елементів технологічної системи знижується внаслідок дії різного роду похибок, що носять систематичний або випадковий характер. Прикладом систематичної похибки може бути змінна в координатах робочого простору верстата похибка кулькових пар “гвинт – гайка”, а прикладом випадкової похибки – розмірне зношування багаторазово використаного в різних операціях інструмента.

Розмірне піднастроювання (як компонент статичного настроювання) – це керування відновленням (при обробці) точності відносного руху і положення інструмента, устаткування та пристрою для продовження робочого процесу із заданою якістю. Розмірне піднастроювання з метою компенсації систематичних похибок здійснюють шляхом періодичного звертання до таблиць корекцій відповідних похибок, збереженим у пам'яті пристрою (наприклад, до таблиць систематичних похибок кулькових пар гвинт-гайка). Випадкові похибки можна компенсувати шляхом періодичного відновлення відповідних таблиць корекцій, збережених у пам'яті пристрою ЧПУ на основі епізодичних вимірювальних циклів (наприклад, циклів обстеження виконавчих поверхонь інструмента).

Розмірне перенастроювання – це процес встановлення необхідної точності відносного руху і положення виконавчих поверхонь інструмента, устаткування і пристосування для отримання заданої якості при переході до обробки деталей іншого типорозміру або при переході до наступної поверхні оброблюваної деталі.

Для здійснення статичного настроювання на верстаті з ЧПУ використовують в основному три методи: метод, пов'язаний зі встановленням координат інструмента в системі координат деталі (метод пробних проходів); метод, пов'язаний із встановленням координат інструмента в системі координат верстата (абсолютний метод); метод, пов'язаний із встановленням координат інструмента в проміжній системі координат, положення якої відносно координатної системи верстата відоме (відносний метод). У верстатах із ЧПУ отримали поширення два останніх методи, які реалізовані за допомогою вимірювальних головок.

Динамічне настроювання – це етап управління точністю обробки безпосередньо в умовах різання, коли спотворенню точності сприяють деформаційні, теплові та динамічні процеси.

В основі зазначених процесів лежать різні фізичні ефекти (пружні та контактні деформації, температурні деформації, тертя, зношування, вимушені коливання, автоколивання). Однак вплив будь-яких факторів на точність обробки виявляється врешті-решт через розмірні зв'язки верстатної системи. Під дією цих факторів відбуваються зміни розмірів і відносних поворотів поверхонь, що приймають участь в утворенні розмірних зв'язків. В результаті виникають відхилення від заданої при статичному настроюванні точності відносного положення і руху інструмента, баз верстата й оброблюваної заготовки. Ці відхилення носять змінний характер і змінюються випадково або за певним законом у функції часу і у функції координат.

Отже, розміри деталей є функціями початкового встановлення, а також статичного і динамічного настроювань. Виходячи з цього, досягнення підвищеної точності можливе шляхом автоматичного управління в межах початкового встановлення, статичного та динамічного настроювання, одночасного залучення декількох видів управління. Очевидно, що управління якогось одного виду може усунути як власні похибки, так і похибки управління інших видів.

Таким чином, якість обробки, а точніше точність обробки, стає керованим показником технологічного процесу, а досягнення якості – компонентом реалізації технологічної задачі ЧПУ.

Зі сказаного випливає спосіб розв'язку тієї частини технологічної задачі, що належить до управління якістю (рис. 6.41). Управляюча програма повинна мати у своєму складі кадри з описами вимірювальних циклів. Вимірювальні цикли формують масиви корекцій різноманітного призначення, що і забезпечує в кінцевому

результаті статичне настроювання.

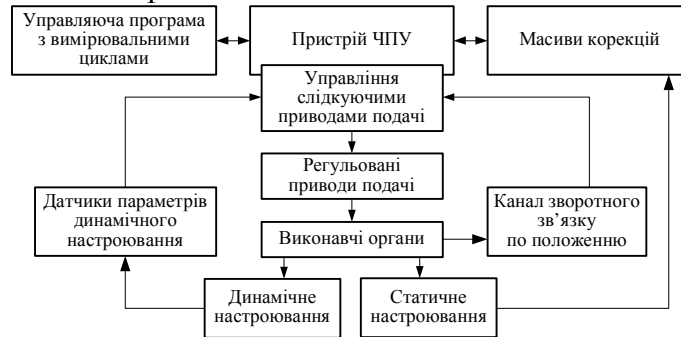


Рис. 6.41. Схема управління якістю обробки в системі ЧПУ

У процесі різання вимірюють параметри динамічного настроювання за допомогою датчиків силових параметрів різання (сили різання, обертального моменту на шпинделі), датчиків температури, вібрацій, віброакустичного спектра, датчиків деформацій і зсувів тощо. Подібна інформація дозволяє при відповідній її обробці управляти динамічним настроюванням.

Оскільки управління точністю здійснюється через приводи подачі, технологічна задача на якомусь етапі свого виконання поєднується з геометричною задачею ЧПУ.

6.6.3. Управління ефективністю обробки та виготовлення деталей

Формулювання задачі оптимізації. Перейдемо до характеристики другої частини технологічної задачі, пов'язаної з управлінням ефективністю обробки.

Найбільшого розповсюдження для оцінки ефективності технологічних процесів в цілому і окремих операцій технологічного процесу отримала собівартість, яка належить до числа найважливіших показників і оцінок роботи.

Систематичне зниження собівартості відбиває інтенсивність технічного прогресу, безперервне підвищення продуктивності праці в промисловості, поліпшення використання устаткування, удосконалення технології та організації виробництва тощо.

За критерієм мінімуму собівартості розрізняють два види оптимізації: варіантну та параметричну.

Варіантна оптимізація включає вибір оптимального варіанта технологічного процесу обробки деталей, оскільки забезпечити потрібну якість деталі при заданій продуктивності можна різними способами обробки, які відрізняються устаткуванням, оснащенням, інструментом і заготовками.

Параметрична оптимізація – це вибір оптимальних технологічних параметрів для конкретного варіанта технологічного процесу обробки деталі.

Варіантна оптимізація дозволяє виявити оптимальний варіант технологічного процесу тільки у тому випадку, коли виконана параметрична оптимізація, тобто попередньо визначені оптимальні технологічні параметри або передбачена їх автоматична оптимізація під час обробки. Тому надалі має сенс розглядати тільки параметричну оптимізацію.

Проблему оптимізації найчастіше зводять до визначення оптимальних режимів різання (швидкості різання, подачі). Однак подібні методи можуть бути залучені до оптимізації перехідних процесів, допусків, послідовності переходів. Зазвичай під економічним режимом різання розуміють такий період стійкості та відповідні йому параметри режиму різання (при заданій вартості одиниць машинного часу, допоміжного часу, накладних витрат), при яких вартість операції буде найменшою. Запис критеріїв оптимальності базується на відомих залежностях між швидкістю різання, подачею, глибиною і періодом стійкості.

У загальному вигляді принцип оптимальності можна сформулювати так: для конкретного варіанта технологічного процесу необхідно визначити такі значення шуканих технологічних параметрів, які забезпечили б найбільшу ефективність процесу при дотриманні обмежень по якості обробки, продуктивності, витратах оборотних коштів, технічних, технологічних і організаційно-технічних можливостях верстата.

Шуканими параметрами для конкретного варіанта технологічного процесу є режими різання (швидкість, подача), міжперехідні припуски і допуски, стійкість різального інструмента (період стійкості та кількість деталей, оброблених за цей період), допуски на розмірне настроювання верстатної системи.

Оптимальні значення шуканих параметрів визначають із врахуванням критерію, який мінімізується або максимізується при виконанні умов комплексу обмежень.

Крім шуканих параметрів, технологічний процес характеризується сукупністю фазових параметрів. Ці залежні параметри (сила різання, потужність, зношування інструмента) є функціями шуканих незалежних параметрів.

Обмеження складають систему нерівностей, кожне з яких встановлює межі варіювання деякого фазового параметра.

Система адаптивного управління технологічним процесом. Системи управління верстатами, що забезпечують пошук і підтримку технологічних параметрів робочого процесу, що задовольняють критерію оптимізації та діючим обмеженням, отримали назву *адаптивних*. Ця назва зберігається, звичайно, й у тих випадках, коли конкретна технічна реалізація не повністю відповідає класу систем адаптивного управління в термінах теорії автоматичного управління. Це пов'язано з існуючими у верстатобудуванні традиціями.

Узагальнена структурна схема адаптивного управління зображена на рис. 6.42. Блок (модуль) адаптивного управління може бути частиною пристрою ЧПУ, а може бути й автономним пристроєм. Зі схеми зрозуміло, що пристрій ЧПУ управляє технологічним процесом лише побічно через приводи подачі, головного руху, автоматички.

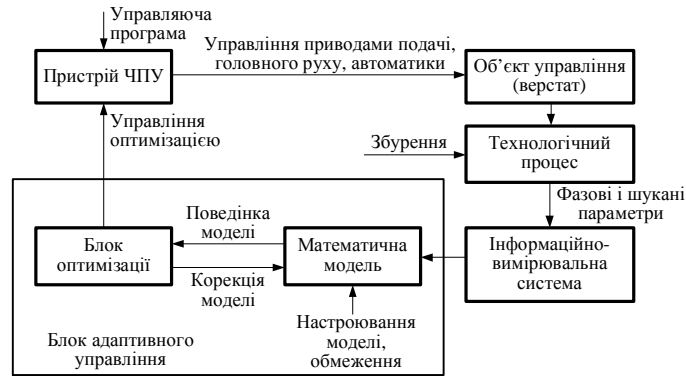


Рис. 6.42. Узагальнена структурна схема адаптивного управління

Таким чином, тільки адаптивне управління здатне знизити негативний вплив діючих на технологічний процес збурювань, а також негативні наслідки недосконалої управляючої програми ЧПУ.

До числа параметрів об'єкта управління – технологічного процесу – відносять збурення, фазові та шукані параметри.

Збурюючим зовнішнім впливом є глибина різання, причому зміна глибини різання представляє собою стаціонарний випадковий процес в залежності від функції узагальненої координати положення інструмента.

До *фазових параметрів* відносяться: сила різання, жорсткість верстатної системи в напрямку нормалі до оброблюваної поверхні, пружне переміщення верстатної системи в напрямку нормалі до оброблюваної поверхні, потужність приводу головного руху, значення розмірного зношувального різального інструмента, миттєве значення поля розсіювання розміру оброблюваної деталі, відносні коливання деталі та інструмента.

До *шуканих параметрів* звичайно відносяться подача і швидкість різання, причому управління може здійснюватися: тільки одним з цих параметрів, обома параметрами незалежно, обома параметрами спільно.

В структурі системи адаптивного управління виділений блок оптимізації, який виконує обчислювальні функції та реалізований за допомогою засобів мікропроцесорної техніки. На рівні налаштування в блок надходять обмеження на процес управління у ході оптимізації цього процесу: обмеження на фазові параметри (на розмірне зношування за період стійкості, на витрати інструмента за одиницю

часу, на продуктивність, на силу або потужність різання, на обертальний момент при різанні тощо), обмеження на шукані параметри, обумовлені технічними можливостями відповідних приводів (подачі, головного руху), обмеження на показники виконуваних операцій. Поточні значення тих чи інших показників операції безперервно обчислюються тим же блоком оптимізації. При цьому переслідується суто інформаційна мета (накопичення статистичної інформації) або мета порівняння з наведеними вище обмеженнями.

Показниками операції можуть бути: *показники точності* (поле розсіювання, миттєве значення поля розсіювання, вихід за межі, встановлені допуском, висота мікронерівностей, рівень вібрацій), *організаційно-виробничі показники* (число оброблених деталей, час обробки, продуктивність операції, штучно-калькуляційний час), *економічні показники* (технологічна собівартість одиниці продукції), *функціональні показники* (фактична стійкість, число деталей, оброблених за період стійкості, площа обробленої поверхні, розмірне зношування за період стійкості, фактичні витрати інструмента за одиницю часу).

Вимірювані параметри залежать від шуканих технологічних. Зазвичай вимірюють силу різання, потужність, зношування інструмента тощо. До параметрів настроювання належать коефіцієнти рівнянь, що визначають математичну модель. Обмеження складають систему нерівностей, кожне з яких установлює межі варіювання вимірюваних і шуканих параметрів. Математична модель зв'язує критерій оптимальності із шуканими технологічними і вимірюваними параметрами. Блок оптимізації, діючи відповідно до деякого постійного алгоритму чи алгоритму, який вибирається, знаходить такі значення шуканих технологічних параметрів, що забезпечують критерію оптимальності максимум чи мінімум. При цьому враховуються необхідні обмеження.

Варто відмітити також, що інформаційно-вимірювальна система сама є безупинно працюючим експериментатором, а обчислювальний блок оптимізації здатен сам виконувати розрахунки, необхідні при побудові моделі методами планування експерименту. У зв'язку з цим блок оптимізації може автоматично вносити необхідні корекції в математичну модель, і, в принципі, синтезувати таку модель.

Математична модель оптимізації режимів обробки. Отримати математичну модель оптимізації режимів обробки суто теоретичним шляхом вдається далеко не завжди. З цієї причини при розробці моделі ефективними є методи теорії планування експерименту.

Розглянемо підхід до побудови системи адаптивного управління на прикладі фрезерної обробки. Як шукані технологічні параметри візьмемо подачу S і частоту обертання n шпинделя. Це означає, що оптимальний режим будемо шукати на площині (S, n) , яка називається виробничою характеристикою верстата.

Визначимо оптимальний режим як таку точку на площині

виробничої характеристики, що відповідає мінімуму наведених витрат q_{min} .

Аналітичний вираз для мінімуму приведених витрат знаходять у вигляді аналітико-емпіричної функції:

$$q_{min} = (B, h, n, S),$$

де B і h – відповідно ширина і глибина фрезерування, причому ці параметри є за своїм значенням збурюваннями;

n і S – регульовані параметри, оптимальне співвідношення яких і потрібно відшукати.

У процесі оптимізації режиму фрезерування доводиться враховувати обмеження. Так, найменша частота обертання n_{min} фрези не може бути меншою деякого певного значення, оскільки в зоні дуже низьких частот погіршуються умови стружкоутворення. В інших випадках n_{min} може визначатися нижньою межею діапазону регулювання головного приводу. Обмеження на n_{max} встановлюють з тих міркувань, що при високих частотах спостерігаються швидке зношування і руйнування інструмента через надмірне нагрівання різальних кромek. Крім того, n_{max} може бути верхньою межею діапазону регулювання головного приводу. Отже, частота обертання повинна задовольняти обмеженню $n_{max} \geq n \geq n_{min}$.

Найбільша подача на один зуб фрези $S_{z\ max}$ визначається вимогами до шорсткості поверхні, а нерідко і міцністю інструмента. Найменша подача на зуб залежить від технологічних вимог. Таким чином, подача на один зуб фрези визначається обмеженням $S_{z\ max} \geq S_z \geq S_{z\ min}$.

Обмеження по “швидкості подачі” складаються з регульовальних можливостей приводу і мають вигляд: $S_{max} \geq S \geq S_{min}$.

Допустимий обертальний момент $M_{об}$ на шпинделі (фрези) визначається міцністю інструмента або механізмів верстата і деформацією верстатної системи $M < M_{об}$.

Максимальна потужність N_{max} на шпинделі обумовлена можливостями приводу, а отже, $N < N_{max}$.

У підсумку всі обмеження можуть бути поділені на конструктивно-технологічну групу (рис. 6.43, а), що виділяють допустиму область на площині виробничої характеристики, а також на технологічні групи обмежень за силовими параметрами різання (рис. 6.43, б), тобто за потужністю і обертальним моментом на шпинделі.

Обмеження конструктивно-технологічної групи створюють на площині виробничої характеристики так звану технологічну зону (заштрихована), а обмеження технологічної групи утворюють у межах цієї зони чотири області, з яких допустима тільки третя. Цю третю область варто розглядати разом з функцією q_{min} , що представлена сімейством кривих (рис. 6.43, в). Очевидно, що оптимальні умови

обробки відповідають якійсь точці на лівій межі цієї області, а саме: тій, для якої миттєве значення приведених витрат буде найменшим.

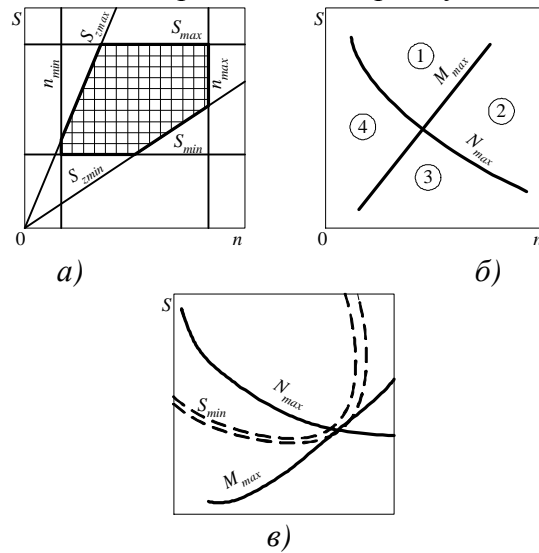
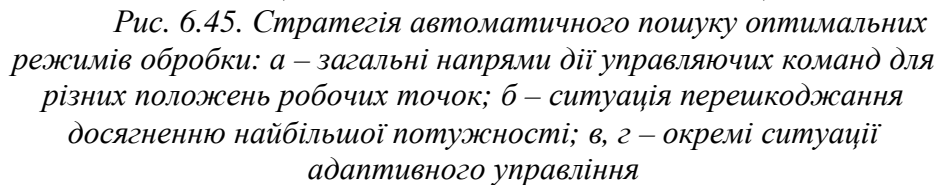
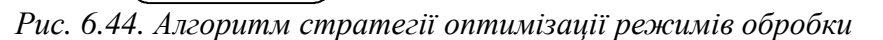


Рис. 6.43. Обмеження на оптимальний вибір режимів обробки:
а – конструктивно-технологічні обмеження; б – технологічні
обмеження по силових параметрах; в – сімейство кривих приведених
витрат

При пошуку оптимуму необхідно впевнитися, на якій із кривих, що складають сімейство, знаходиться поточна робоча точка. Це робиться за допомогою інформаційно-вимірювальної системи і математичної моделі блока адаптивного управління. Далі в роботу вступає блок оптимізації, що діє на підставі стратегії.

Стратегія оптимізації представлена у вигляді алгоритму, зображеному на рис. 6.44. Насамперед устанавлюється положення робочої точки відносно області конструктивних і технологічних обмежень. Якщо точка знаходиться поза технологічною областю, подаються команди на введення її в середину області. Якщо точка знаходиться в межах технологічної області, то за допомогою датчиків потужності та обертового моменту реєструється положення робочої точки відносно чотирьох областей 1–4. Далі подаються команди на досягнення оптимальної точки. Процедура повторюється з деякою частотою сканування.



Також продемонстровані приклади роботи блоком адаптивного управління у різних ситуаціях, що виникають при обробці деталі, а саме: ситуація, коли технологічне обмеження по S_{max} не дозволяє досягти максимальної потужності (рис. 6.45, б), та інші ситуації відтворення технологічних обмежень по n_{max} та n_{min} (рис. 6.45, в, г).

На закінчення можна відзначити, що структурна схема системи адаптивного управління верстатом з ЧПУ вимальовується в міру вибору критерію ефективності процесу обробки (економічного, точнісного або критерію, що характеризує ступінь досконалості механізмів верстата), а також обмежень, в межах яких використання критерію має сенс, конкретних методів розробки математичної моделі (аналітичних, експериментальних, аналітико-експериментальних), алгоритмів оптимізації роботи верстата відповідно до прийнятого критерію і накладеними на нього обмеженнями, технічних засобів адаптивного управління та їх розміщення в системі ЧПУ.

6.7. Реалізація термінальної задачі в інтерактивних системах взаємодії пристроїв ЧПУ

6.7.1. Структура реалізації термінальної задачі в пристрої ЧПУ

Пристрій ЧПУ початково був створений як проблемно-орієнтована обчислювальна машина реального часу. Але потім, в міру зростання активної ролі оператора в управлінні, все більше збільшувалась питома вага інтерактивних діалогових процедур та пов'язаних з ними процесів машинного масштабу часу. Поступово сформувалось достатньо автономне коло задач, розв'язання яких не вимагало спеціальної, і навіть спеціалізованої, апаратури ЧПУ, але повністю могло бути виконане універсальними обчислювальними засобами на основі взаємодії оператора з терміналом (пасивним або активним). Так сформувалась термінальна задача ЧПУ.

До термінальної задачі ЧПУ віднесені всі прояви взаємодії пристрою ЧПУ із навколишнім середовищем: по-перше, це – діалог з оператором, а по-друге, – діалог з іншими системами управління. Технічними засобами підтримання діалогу є, перш за все, пасивний термінал (панель оператора) або активний термінал (персональний комп'ютер) і, крім цього, інтерфейс із управляючими пристроями зовнішнього середовища, тобто можна говорити, що реалізація термінальної задачі потребує власного виділеного обчислювача (ЕОМ), зовнішні та внутрішні зв'язки якого з пристроєм ЧПУ показані на рис. 6.46.

До ряду функцій, що відтворюються в рамках термінальної задачі в наведеній структурі, відносяться: інтерактивне введення завдань ЧПУ, автоматизоване проектування завдань ЧПУ, автоматизоване проектування процесів ЧПУ, підтримка мережевих функцій пристрою ЧПУ, підтримка бібліотек завдань і процесів ЧПУ,

підтримка дисплейних функцій інших задач ЧПУ.

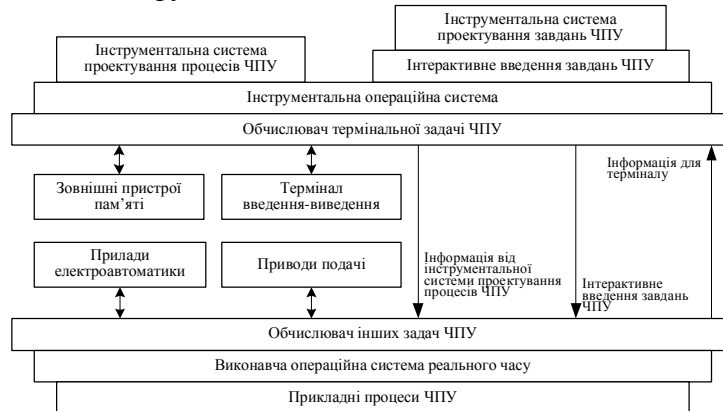


Рис. 6.46. Узагальнена структура реалізації термінальної задачі ЧПУ

До завдань ЧПУ відносяться введені в діалозі вказівки оператора і директиви управляючої програми ЧПУ, що ініціюють або настраюють ті чи інші прикладні резидентні процеси ЧПУ. Таким чином, до функцій інтерактивного введення завдань ЧПУ відносяться засоби підтримки діалогу з оператором і пошуку управляючої програми ЧПУ в бібліотеці.

Оскільки управляюча програма ЧПУ містить директиви мовою керування завданнями, система автоматизованого проектування управляючих програм є по суті інструментальною системою розробки завдань (точніше частини завдань, оскільки іншу частину визначає оператор). Подібна система побудована на основі діалогу і має графічний супровід.

Процеси ЧПУ, як уже відзначалося, є резидентними, і створення нових процесів користувачу недоступне. Однак на стадії стикування універсального пристрою ЧПУ з верстатом доводиться проектувати і вводити в пристрій систему електроавтоматики верстата. При цьому використовують інструментальні засоби автоматизованого проектування процесів електроавтоматики. Та ж система займається підтримкою передачі процесів електроавтоматики обчислювачеві інших задач ЧПУ.

Підтримка мережних функцій відноситься до проблеми включення пристрою ЧПУ в локальну обчислювально-керуючу мережу інтегрованої системи управління гнучким виробництвом. Підтримка мережних функцій виливається в програмно-апаратну підтримку протоколів інформаційного обміну з віддаленими абонентами.

У бібліотеку завдань входять управляючі програми, підпрограми і стандартні цикли. У бібліотеці процесів містяться процеси електроавтоматики, параметри верстата, таблиці корекцій. Підтримка бібліотеки полягає в організації доступу до неї для внесення нових файлів, знищення старих, переміщення файлів з бібліотечної

пам'яті в оперативну пам'ять обчислювача термінальної задачі ЧПУ.

Підтримка дисплейних функцій інших задач ЧПУ (під іншими задачами розуміються всі, крім термінальної) полягає у прийомі інформації для терміналу з боку обчислювача інших задач ЧПУ і передачі відповідних повідомлень на дисплей терміналу.

Програмно-математичне забезпечення термінальної задачі ЧПУ утворює віртуальну обчислювальну машину, вертикальна ієрархічна структура якої складається з процесора, комунікаційного середовища, операційної системи, драйверів зовнішніх пристроїв. На верхньому прикладному рівні групи прикладних процесів, зв'язаних єдиною функціональністю, інтегровані в пакети, що виступають як процесори. Звертання до процесора можливі через інтерфейсне комунікаційне середовище, що підтримує протоколи міжпроцесорного інформаційного обміну.

Горизонтальна структура віртуальної обчислювальної машини для рішення термінальної та всіх інших задач ЧПУ складається з процесорів діалогу з оператором, технологом, конструктором, процесора інших задач ЧПУ, процесора терміналу, процесора зовнішньої пам'яті, мережевого процесора, що об'єднані віртуальною шиною. Окремі віртуальні процесори обмінюються даними і керуючими директивами в рамках протоколу, що забезпечується інтерфейсним комунікаційним середовищем. Крім того, комунікаційне середовище надає процедури вбудовування та вилучення процесорів у систему. Таким чином, комунікаційне середовище утворює загальну віртуальну шину.

Уявлення про віртуальні процесори і загальну віртуальну шину дозволяють трактувати програмно-математичне забезпечення термінальної задачі ЧПУ як емулятор системи паралельно працюючих процесорів, зв'язаних загальною шиною.

Уточнимо функції окремих процесорів.

Процесор діалогу з оператором вирішує задачу інтерактивного введення завдань ЧПУ. Процесор діалогу з технологом вирішує задачу інтерактивного автоматизованого проектування завдань за допомогою інструментальної системи розробки управляючих програм ЧПУ. Процесор діалогу з конструктором вирішує задачу інтерактивного автоматизованого проектування процесів ЧПУ за допомогою інструментальної системи розробки електроавтоматики.

Наявність мережевого процесора і процесора інших задач ЧПУ означає, що до загальної віртуальної шини можна підключати не тільки програмні блоки, що існують в обчислювачі термінальної задачі ЧПУ, але і програмно-апаратні пристрої поза обчислювачем. Процесор терміналу підтримує дисплейні функції інших задач ЧПУ. Процесор зовнішньої пам'яті підтримує бібліотеки завдань ЧПУ і процесів ЧПУ.

Зі сказаного випливає, що в програмно-математичному забезпеченні термінальної задачі ЧПУ значну частку складають

інтерактивні (діалогові) процедури, розроблені на основі заздалегідь складеного сценарію. У зв'язку з цим при проектуванні програмно-математичного забезпечення необхідні інструментальні засоби реалізації діалогу за сценарієм розробника.

6.7.2. Режими і засоби діалогової роботи оператора з пристроєм ЧПУ

Термінальна задача стала настільки чіткою, що термінали пристрою ЧПУ все частіше стають окремим конструктивом в модульному виконанні пристрою ЧПУ, який включає власне систему управління та пов'язані з нею оптоволоконними каналами пультовий модуль (пасивний термінал) і модуль введення-виведення сигналів електроавтоматики верстата (керуючих та інформаційних).

Пульти оператора може бути виконаний у вигляді універсальної консолі (пасивний термінал), яка підключена до системи управління за допомогою послідовного каналу на основі кабеля. Крім того, до складу модульного виконання пристрою ЧПУ частіше став входити і універсальний персональний комп'ютер (активний термінал) з комплектом периферії, яка відповідає замовленню користувача.

Інструментом спілкування оператора з панеллю оператора, до якого б терміналу (пасивного, активного) вона б не належала, є дисплей та клавіатура. Клавіатура призначена для вибору режиму роботи, введення алфавітно-цифрових даних, керування курсором з метою редагування інформації, зміни сторінок інформації, що виводиться на дисплей, зміни системи відліку, вибору дискретності переміщень, введення команд виконання типу “ввімкнути”, “вимкнути”, “пуск”, “стоп”, задання елементів геометрії, задання укрупнених команд типу технологічних циклів тощо.

Приклад організації панелі оператора показаний на рис. 6.47.

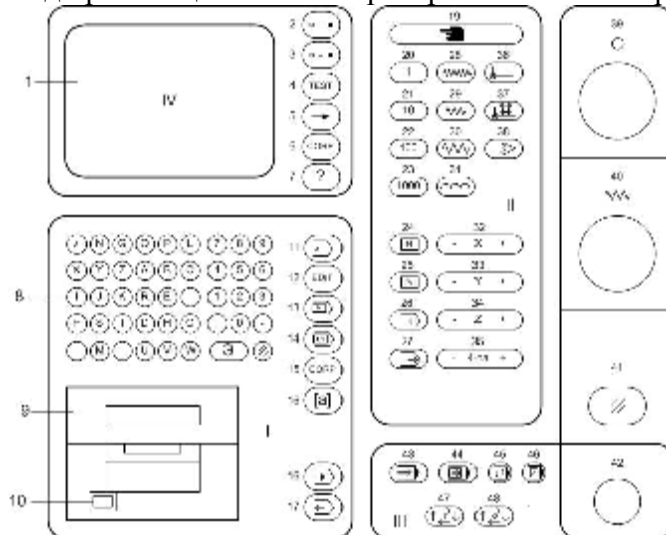


Рис. 6.47. Організація панелі оператора

Елементами цієї панелі оператора є: 1 – екран дисплея; 2–7 – клавіатура виклику визначеної інформації; 8 – алфавітно-цифрова клавіатура; 9–10 – касетний магнітофон; 11–19 – режимні клавіші; 20–23 – вибір переміщення на встановлене число дискрет; 24 – просування до кінця програми; 25 – повернення на початок програми; 26 – просування вперед; 27 – просування назад; 28–31 – ручна подача різної швидкості; 32–35 – вибір напрямку переміщення по відповідних координатах; 36 – рух до початку відліку; 37 – переміщення до реперної точки вимірювальної системи; 38 – скидання в нуль показів координатних переміщень; 39–40 – потенціометри ручного регулювання частоти обертання шпинделя і подачі; 41 – встановлення системи управління в початковий стан; 42 – аварійна зупинка; 43 – автоматичний режим; 44 – покадрове відпрацювання програми; 45 – зупинка з підтвердженням; 46 – пропуск відмічених кадрів; 47 – пуск циклу; 48 – зупинка подачі. Всі наведені елементи панелі поділені на функціональні зони: I – введення в пам'ять, виведення з пам'яті, маніпуляції із введеною інформацією; II – управління переміщеннями в режимі налагодження; III – управління автоматичним режимом; IV – візуалізація та засоби управління режимом візуалізації.

Екран дисплея показує наступну інформацію: стан системи управління (вказання режиму та підрежимних розгалужень, так званих “мод”, повідомлення про помилки та збої); стан об'єкта управління, тобто верстата (положення робочих органів у різних системах координат, подача, частота обертання шпинделя, імена виконуваних циклів автоматики, інформація про управляючу програму), опис всіх “минулих” дій оператора; опис стандартних циклів обробки, включаючи їх графічну інтерпретацію; надання статистично-графічної моделі управляючої програми (накладені ескізи заготовки і виробу із вказівками всіх робочих та допоміжних проходів), надання динаміко-графічної моделі процесу обробки (зображення поточного стану і його розвитку у реальному часі).

Інформація, яка з'являється на екрані дисплея, відображає ту частину рішення термінальної задачі, яка у даний момент цікавить оператора. Якщо панель оператора є компонентом пасивного терміналу (тобто консолі оператора), то саме рішення термінальної задачі здійснюється у пристрої ЧПУ. Якщо ж панель оператора належить активному терміналу (тобто персональному комп'ютеру), то рішення термінальної задачі виконується у самому терміналі.

Обсяг інформації, яка надається оператору, тобто обсяг дисплейних функцій, багато у чому визначає функціональні можливості пристрою ЧПУ.

Розглянемо основні діалогові режими пристрою ЧПУ при реалізації термінальної задачі. Ці режими налагоджують систему управління на певний рід роботи. В пристрої ЧПУ передбачається шість таких режимів.

У режимі “Верстат” здійснюють підготовчі операції типу штовхаючих (невимірюваних) переміщень виконавчих органів верстата, вимірювані переміщення на встановлене число дискрет, а також проводять обнулення буферів досягнутих позицій.

У режимі “Ручне введення” виконують окремі команди мови управляючих програм або окремі незалежні кадри програми без збереження команд і кадрів у пам’яті пристрою ЧПУ.

У режимі “Пам’ять програм” вводять у пам’ять із клавіатури і будь-яких носіїв управляючі програми ЧПУ, редагують програми і виводять їх на зовнішні носії.

У режимі “Корекція” працюють з таблицями, що зберігаються в пам’яті пристрою ЧПУ, вводять і редагують таблиці корекцій інструмента тощо.

У режимі “Відпрацювання” відтворюється автоматичний цикл обробки деталі відповідно до активізованої управляючої програми ЧПУ.

У режимі “Діагностика” встановлюють працездатність системи ЧПУ в цілому і будь-яких окремих її підсистем.

Пристрій ЧПУ може працювати і з декількома терміналами одночасно. Звичайно у цьому випадку лише один термінал використовують для ведення діалогу (будь-який за вибором), а інші служать тільки для виведення поточної інформації.

Виходячи з наведених режимів, існує чотири варіанти діалогу оператора з пристроєм ЧПУ: керування процесом і об’єктом, системна робота, автоматизоване проектування керуючих програм, редагування керуючих програм.

Діалог керування процесом і об’єктом протікає безпосередньо під час обробки деталі (ручної або автоматичної). Мова іде про запрошення з боку системи управління вибрати режим (наприклад, з числа запропонованих на екрані), вибрати команду на пересування у ручному режимі (наприклад, шляхом вказання альтернативних адрес).

Під діалоговою системною роботою розуміють маніпулювання різноманітними наборами констант, параметрів настроювання, коректорів, архівну роботу із зовнішніми носіями, вхід у інформаційний обмін з ЕОМ верхнього рангу.

Деякого пояснення потребують вказані тут (і дещо раніше у зв’язку з дисплейними функціями) параметри настроювання. Справа в тому, що пристрій ЧПУ здатний у деякому ступені адаптуватися до об’єкта (верстата) та користувача. Для забезпечення адаптації виділяють спеціальні області пам’яті, які закріплені за константами системи управління і верстатними параметрами (далі і те, й інше будемо називати параметрами настроювання), які служать для введення корекцій чи компенсацій, уточнення чи обмеження (функцій ЧПУ, мови управляючих програм), контролю, візуалізації, діагностики і оптимізації зв’язку пристрою ЧПУ з верстатом.

Параметри настройки поєднуються у наступні однорідні групи:

- ознаки і вказівки, які уточнюють синтаксис керуючої програми, семантику адрес інформаційних слів мови управляючих програм, реакцію системи управління на ті або інші команди;
- ознаки, які визначають початковий стан пам'яті (наприклад, коди *G*-функцій, які діють початково);
- ознаки, що уточнюють структуру окремих стандартних циклів;
- ознаки, що блокують окремі функції клавіатури панелі оператора, уточнюючі функції клавіатури і реакцію системи управління на клавішні команди;
- ознаки, які керують обсягом і складом інформації, що виводиться на дисплей;
- константи, покази, допуски, поточні та граничні значення, які використовують при розв'язку задач ЧПУ;
- протоколи інформаційного обміну з периферією, зовнішніми пристроями, об'єктом;
- значення коректуючих і компенсуючих сигналів і зміщення, вказівки на знак корекції;
- параметри настроювання слідкуючих приводів подачі;
- параметри погоджування з приводом головного руху, параметри настроювання приводу;
- ознаки, що визначають ціну дискрети, діапазон вимірювання, розмірність, масштаб, границі робочої зони, наявність базових точок та зміщень;
- ознаки настроювання конфігурації системи ЧПУ;
- ознаки, які задають чи відміняють автоматичне виконання тих або інших функцій.

Повне число параметрів у сучасних пристроях ЧПУ може досягати декількох сотень. Вони суттєво різняться за структурою (однобітні ознаки, багаторозрядні константи, динамічно змінювані змінні), за призначенням (блокування, умовний та безумовний переходи, фіксація деякого чисельного значення, позначення границь, меж тощо), за характером введення (однократний при стикуванні з технологічним устаткуванням і підготовці до першого пуску у користувача, періодичний в міру виробничої необхідності, безперервний по автоматично діючих каналах), за досяжністю (потребують випробовування та вимірювань; досяжні представникам інженерних служб, наладнику, оператору) тощо.

Діалог автоматизованого проектування управляючих програм припускає використання мови більш високого рівня, ніж код ISO-7bit.

Процедура введення даних, необхідних для побудови управляючої програми, базується на одному з двох варіантів систем автоматизованого проектування: циклова система, інструментальна система.

Циклова система автоматизованого проектування використовує

наступну ідею: управляючу програму обробки деталі можна скласти зі стандартних технологічних рішень (циклів), які розроблені заздалегідь у параметричній формі і потребують лише того, щоб параметрам надали конкретні числові значення. Зі стандартних системних циклів оператор виділяє той, який першим необхідно застосувати для обробки, потім задає усі необхідні геометричні та технологічні дані, які відносяться до цього циклу. Якщо для обробки деталі недостатньо одного циклу, то наведений процес повторюється для наступного стандартного циклу, який додається до управляючої програми.

Оператор в цикловій системі починає роботу з вибору одного зі способів обробки отвору, повний набір яких надає система управління. У відповідності до зробленого за допомогою віртуальної клавіші вибору на екрані з'являється ескіз обробки з узагальненими параметрами замість розмірів і технологічних даних. Система управління послідовно запитує оператора конкретні чисельні значення параметрів. Після їх введення система управління уточнює у оператора загальну схему розташування отворів і формує програмний код виконання операції.

Для *інструментальної системи* автоматизованого проектування управляючої програми характерна наступна схема дій оператора: визначення геометрії деталі та заготовки; визначення геометрії робочої частини кожного використовуваного інструмента та його положення у координатній системі деталі; визначення усіх необхідних проходів; формування ISO-коду управляючої програми.

Для деталей складних форм найбільш трудомісткою частиною є геометричний опис даних креслення. Використовують звичайну систему проєкцій, а контрольні ізометричні зображення синтезуються автоматично.

Оператор в такій системі починає програмування з вибору квадранта, у якому буде показана геометрія деталі. Далі він визначає форму і розміри заготовки та її положення у системі координат верстата, після чого приступає до послідовного введення геометричних компонентів кінцевого профілю деталі. Потім потрібно зв'язати взаємне розташування деталі та інструментальної револьверної головки, вибрати послідовність технологічних операцій та використовувані інструменти, призначити режими різання. У результаті система управління надасть на екрані повну статистично-графічну модель процесу обробки, окремі фрагменти якої оператор може при бажанні побачити у збільшеному масштабі. В результаті система управління запропонує еквівалентний текст управляючої програми у коді ISO-7bit.

Динамічно-графічне моделювання процесу обробки служить з метою верифікації розробленої управляючої програми. Траєкторія руху інструмента викреслюється на екрані дисплея у темпі, який регулює оператор. Особливо наочним є просторове динамічно-графічне

моделювання.

Діалогове редагування управляючих програм (четвертий із наведених вище варіантів діалогу) охоплює саме редагування та ручне введення тексту управляючої програми. Типовими директивами процесу редагування є “Вставити”, “Вирізати”, “Замінити”, які можуть бути віднесені до окремих команд і цілих кадрів.

Завершуючи аналіз термінальної задачі ЧПУ, відмітимо два додаткових її компоненти: діалог “термінал – пристрій ЧПУ” та діалог “пристрій ЧПУ – ЕОМ”. Обидва діалоги розвиваються автоматично у відповідних каналах зв’язку, забезпечуються відповідними протоколами, підтримуються програмно-апаратними засобами системи керування.

У загальному обсязі програмно-математичного забезпечення сучасного пристрою ЧПУ обсяг забезпечення термінальної задачі займає основне місце. У цьому зв’язку при його розробці необхідно приділяти особливу увагу питанням загальної організації та архітектури. Оскільки багато підсистем термінальної задачі інтерактивні, винятково важливого значення набувають інструментальні засоби проектування діалогу. Із врахуванням сказаного виникає можливість регулярного конструювання модуля термінальної задачі, яке дозволяє вирішити проблеми надзвичайно високого рівня трудомісткості. Мається на увазі, що подоланню проблеми служать й інші відомі прийоми: використання потужних стандартних мультипроцесних операційних систем, використання стандартних графічних пакетів та існуючих напрацювань в області автоматизованого проектування управляючих програм, програмування термінальної задачі на одній із сучасних мов високого рівня.

Контрольні запитання

1. Сформулюйте фази реалізації геометричної задачі ЧПУ в технологічному устаткуванні ГВС
2. Визначьте склад геометричної інформації необхідної для подання формоутворення деталей у пристрої ЧПУ.
3. Подайте відмінності застосування методів поєднання ділянок контуру еквідистанти при програмуванні формоутворення деталей.
4. Наведіть різновиди інтерполяції формоутворення деталей в системах ЧПУ
5. Охарактеризуйте можливості інтерполяції за методом оціночної функції
7. Охарактеризуйте можливості інтерполяції за методом оціночної функції з прогнозуючим кроком
8. Охарактеризуйте можливості інтерполяції за методом оціночної функції на постійній частоті
9. Охарактеризуйте можливості інтерполяції за методом цифрових диференціальних аналізаторів

10. Охарактеризуйте можливості інтерполяції за методом прогнозу та корекції

11. Розкрийте організацію управління слідкуючим приводом подачі пристроїв з ЧПУ при відтворенні формоутворюючих рухів робочих органів технологічного устаткування

12. Поясніть структуру реалізації та фази розв'язання логічної задачі в системах електроавтоматики пристроїв з ЧПУ

13. Визначте необхідність реалізації технологічної задачі програмного управління в системах формування заданої точності обробки пристроїв з ЧПУ

14. Обґрунтуйте необхідність реалізації технологічної задачі програмного управління в системах управління ефективністю обробки пристроїв з ЧПУ

15. Розкрийте можливості реалізації термінальної задачі програмного управління в інтерактивних системах пристроїв з ЧПУ

«Видатним досягненням сучасної науки стала програма, яка на будь-яке запитання дає будь-яку відповідь» (з першоквітневого випуску "Наркому")

Розділ 7

ПРОЕКТУВАННЯ ПРОГРАМНОГО УПРАВЛІННЯ В СИСТЕМАХ ВИКОНАВЧОГО РІВНЯ ГВС

7.1. Методи підготовки управляючих програм ЧПУ

Створення управляючих програм (УП) є одним з головних виробничих етапів при використанні устаткування з ЧПУ. Без врахування трудомісткості цього етапу не можна визначити область застосування та ефективність верстатів із ЧПУ.

Підготовка УП для обробки деталі на верстаті з ЧПУ в загальному випадку означає підготовку і нанесення на програмоносій необхідних команд, що можуть бути автоматично прочитані та виконані системою управління і самим верстатом.

Наладник верстатів із ЧПУ повинен добре знати процес підготовки УП, легко читати програму і вміти при необхідності коригувати програму. Він повинен вміти визначити причини браку при виготовленні деталі або поломки інструмента, що можуть виникнути в результаті неправильно складеної УП, або через збої в системі ЧПУ, або через відмовлення механічних вузлів.

При програмуванні вирішальне значення має збір і упорядкування інформації перед тим, як вона буде нанесена на програмоносій. Спочатку з креслення оброблюваної деталі отримують геометричну інформацію, тобто розміри елементів деталі (наприклад, координати отвору, радіус дуги кола фрезеруемого контуру тощо). Потім, користуючись довідниками та інструкціями, формують технологічну інформацію, індивідуальну для кожного технологічного переходу (наприклад, вид інструмента, частоту обертання інструмента, подачу тощо). На основі геометричної та технологічної інформації з кожного переходу і складають УП.

При підготовці УП докладно вивчають технологічні операції, визначають послідовність переходів і найбільш раціональний режим їх виконання, що забезпечує отримання деталі заданих розмірів за мінімальний машинний час.

Існують наступні методи підготовки УП (рис. 7.1): *ручне програмування*, при якому збір і упорядкування інформації та нанесення її на програмоносій здійснює технолог-програміст; *машинне програмування*, при якому такі роботи, як кодування інформації, визначення переміщень інструмента, вибір режимів різання та оптимальної послідовності виконання переходів тощо, виконує ЕОМ;

машинне програмування біля верстата, оснащеного мікропроцесорним пристроєм ЧПУ (в найбільш вдосконаленому випадку – це задання УП з клавіатури пристрою ЧПУ в режимі діалогу, у процесі якого багато питань призначення режимів різання вирішуються мікро-ЕОМ пристрою ЧПУ і використовуються типові технологічні цикли, що зберігаються в пам'яті пристрою ЧПУ).

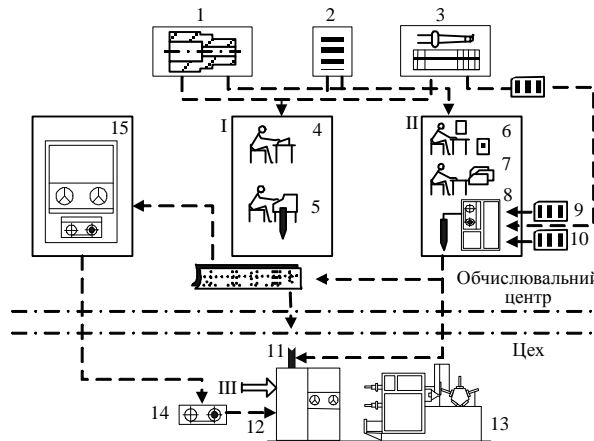


Рис. 7.1. Методи підготовки УП: I – ручне програмування; II – машинне програмування за допомогою ЕОМ; III – машинне програмування біля верстата, оснащеного мікропроцесорним пристроєм ЧПУ (запис УП на програмоносій у вигляді касети чи електронної пам'яті); 1 – вихідні дані по геометричних параметрах деталі; 2 – технологія обробки; 3 – інструмент і пристосування; 4 – розрахунок програми на клавішних апаратах; 5 – перфорування стрічки з УП; 6 – запис програми розрахунку на ЕОМ; 7 – підготовка перфокарти (перфострічки) з УП; 8 – ЕОМ; 9 – програма процесора; 10 – програма постпроцесора для конкретного верстата; 11 – УП; 12 – пристрій ЧПУ із заданням програми на перфострічці; 13 – верстат; 14 – магнітна мініка세트 (чи блок електронної пам'яті) з УП; 15 – пристрій запису УП на магнітну мінікасету (чи блок електронної пам'яті)

Методи кодування УП, вид програмоносія та щільність запису на ньому, способи зчитування інформації з УП є основними показниками систем ЧПУ і залежать, в основному, від елементної бази пристрою ЧПУ.

Для першого покоління пристроїв ЧПУ характерні наступні показники: програмоносій на магнітній стрічці; запис УП у фазомодульованому вигляді чи унітарному коді (послідовності імпульсів); мала щільність запису і, як наслідок, значна довжина УП. Однак пристрій ЧПУ завдяки цьому спрощується і забезпечується його надійність. Таке рішення відповідало рівню розвитку техніки в той період.

Майже одночасно як програмоносій стали застосовувати

перфострічку із заданням УП у кодованому вигляді. Спочатку використовували п'ятидоріжкову перфострічку (код БЦК-5), потім – восьмидоріжкову перфострічку із записом УП за Міжнародними стандартами кодування інформації ISO.

Стандартизація кодування в пристроях ЧПУ другого покоління дозволила уніфікувати підготовку УП і застосовувати однакове введення програми в різних пристроях ЧПУ. Стандартизація кодування УП зберігається при розширенні технологічних можливостей верстатів і, відповідно, розширенні функцій пристрою ЧПУ.

У період створення пристроїв ЧПУ другого покоління розвиваються методи автоматичного програмування на універсальних ЕОМ. При цьому, крім траєкторії руху інструмента відносно заготовки, програмуються параметри режимів обробки.

Створення управляючих мікро-ЕОМ з електронною пам'яттю обумовило подальше удосконалення пристроїв ЧПУ. Системи ЧПУ третього і четвертого поколінь забезпечують не тільки редагування УП, але і задання нової УП безпосередньо за кресленням деталі (в оперативних системах управління інтегрованих із системами проектування та програмування).

В міру введення мов високого рівня, типових циклів і самонастроювання режимів різання програмування поширюється на все більш складні види обробки. Підготовка УП та їх редагування здійснюються безпосередньо на верстаті. Це дозволяє підвищити оперативність виробництва, скоротити час налагодження та перевірки УП, виключити етап підготовки перфострічки, підвищити надійність роботи пристрою ЧПУ.

Наступним етапом, пов'язаним зі створенням гнучких виробничих систем, є автоматичний розрахунок УП за допомогою ЕОМ верхнього рівня.

УП записують на перфострічці у вигляді послідовності фраз. Якщо верстат має магнітну чи електронну касету, то УП спочатку записують на перфострічці, а потім переписують на програмоносій пристрою ЧПУ. Кожна фраза містить інформацію (геометричну та технологічну), необхідну для обробки визначеної ділянки деталі або для виконання спеціальних функцій (початок програми, прискорене підведення інструмента тощо). Послідовністю фраз у програмі визначається послідовність обробки деталі на верстаті.

Кожна фраза складається з деякого числа рядків, у яких записують кодові позначення символів програми. Декілька послідовно записаних рядків, що визначають одне поняття, утворюють слово. Декілька слів, що визначають групу дій за деякий проміжок часу, складають кадр.

Фразами задають тільки ту інформацію, що змінюється відносно попередньої фрази. Застосовують адресну систему запису, при якій використовують буквені адреси, що передують числовій інформації.

Застосування буквених адрес дозволяє робити запис слів у фразах у будь-якій послідовності і використовувати єдиний спосіб кодування для верстатів різного технологічного призначення та різного рівня автоматизації. При цьому за визначеними командами управління закріплені відповідні символи коду адреси.

Кодом називається умовна позначка цифри, числа і букви. Кодування необхідне для складання УП, нанесення її на програмоносій та правильного читання УП системою управління верстатом. Вимоги, що висуваються до коду: максимальна стислість запису; легкість його читання людиною після нетривалого тренування; мінімальна імовірність помилки при його автоматичному зчитуванні та записі.

Тому була розроблена комбінована двійково-десятькова система числення, що у даний час знайшла найширше практичне застосування.

Припустимо, що потрібно перемістити стіл верстата по осі X на 273,41 мм при дискретності 0,01 мм. Отже, у програмі нам потрібно вказати, що по осі X необхідно переміститися на 27341 імпульс. Спочатку кожен цифру числа 27341 записують в двійковому коді: $2 = 0010$; $7 = 0111$; $3 = 0011$; $4 = 0100$; $1 = 0001$. Потім набивають ці цифри послідовними рядками на перфострічку та отримують двійково-десятьковий код заданого числа, оскільки номер n кожного з п'яти рядків (що відповідає десятковому розряду числа) визначає коефіцієнт $K = 10^{n-1}$, на який потрібно помножити записану в рядку (в двійковому коді) цифру, щоб отримати потрібне число. Якщо перед написанням (у двійково-десятьковому коді) числа вказати (у відповідному коді), що далі йде X , то система управління верстатом інтерпретує наступну вказівку: перемістити виконавчий орган верстата (наприклад, стіл) по осі X на 27341 імпульс.

7.2. Основи програмування управляючих програм верстатів з ЧПУ

7.2.1. Кодування інформації в управляючій програмі

Структура управляючої програми. Технологічний процес обробки на верстаті, деталізований до елементарних переміщень і технологічних команд, служить вихідною інформацією для кодування і запису управляючої програми верстата з ЧПУ.

Сукупність команд на мові програмування, що відповідає заданому алгоритму функціонування верстата по обробці деталі, складає *управляючу програму*.

УП на програмоносії містить інформацію про різальний інструмент і швидкості його переміщення відносно оброблюваної заготовки, частоту обертання шпинделя, зміну інструмента, введення корекції, подачу охолодження, інші команди виконавчим механізмам верстата, записаних у послідовності технологічного процесу обробки на верстаті з ЧПУ.

Міжнародним стандартом ISO 1056-75, 1113-73, 2539-74

регламентовані єдині для всіх видів верстатів правила кодування інформації УП на носії даних (перфострічку, магнітну стрічку тощо). УП записується на програмоносій у вигляді послідовності кадрів, що представляють собою закінчені за змістом фрази мовою кодування геометричної, технологічної та допоміжної інформації. Отже, складовою частиною УП, що вводиться і відпрацьовується як єдине ціле є *кадр*.

Окремі послідовності кадрів для обробки ділянок заготовки об'єднуються в *глави* УП, кожна з яких починається з головного кадру.

Головний кадр містить інформацію про умови обробки, з нього можна починати чи відновлювати роботу верстата за програмою. Інші кадри глави УП несуть тільки змінену відносно попередніх кадрів частину інформації і називаються *додатковими*. В УП головні кадри характеризують початковий стан наступної за ним ділянки програми, а додаткові містять всі дані, необхідні для поновлення процесу обробки заготовки після перерви.

Структурно кадр складається зі *слів*, розташованих у визначеному порядку, а слова – із символів. Першим символом слова є буква, що позначає *адресу*, а інші символи утворюють число зі знаком або цілочисельний код без знаку. Адреса слова встановлює призначення наступних за ним даних. В цілому кадр містить послідовність слів, що утворюють інформацію для однієї робочої операції, а слово – дані про параметр процесу обробки або інші дані про виконання управління операцією.

Для верстатів з ЧПУ основним кодом управляючих програм є семирозрядний буквено-цифровий код ISO-7bit. Цей код був призначений для запису інформації на восьмидорожжовій перфострічці. Код ISO-7bit дозволяє кодувати 128 символів. Першим чотирьом доріжкам перфострічки приписані ваги двійково-десятькового коду 1-2-4-8 (двійкова комбінація в рядку може мати різну десяткову вагу). Таким чином, у будь-якому рядку можуть бути закодовані числа від 0 до 15. Ці числа є власне числами, буквами, знаками в залежності від коду ознаки на п'ятій–сьомій доріжках.

Перешкодозахищеність у коді ISO-7bit забезпечена порядковим контролем на парність. Для цього пробивається отвір на восьмій доріжці, якщо на сімох попередніх доріжках комбінація отворів непарна. На кожному рядку перфострічки повинно знаходитися і зчитуватися тільки чітке число отворів.

Для кодування управляючої інформації використаний адресний спосіб, відповідно до якого інформаційне слово (функція) складається з буквеної адреси і числа зі знаком або без нього. Звичайно в словах можна опускати незначущі нулі і знак “плюс”, що мається на увазі по замовчуванню. Для кодування адреси слова у кадрі використовують буквені, графічні та цифрові символи, призначення яких наведені в табл. 7.1 та 7.2. Кожна УП повинна починатися символом “Початок

програми”, після якого може ставитись номер програми і обов’язково символ “Кінець кадру”. Перед символом “Початок програми” може бути записана будь-яка інформація, що немає символу “Початок програми”. Ця інформація сприймається як коментар, у якому наведені номер креслення, найменування деталі, модель верстата, прізвище технолога-програміста, дата розробки, різні ідентифікатори програми тощо.

Таблиця 7.1

Символи адрес інформаційних слів у кадрі

Символ	Призначення
<i>A, B, C</i>	Кутові переміщення навколо осей <i>X, Y, Z</i>
<i>D</i>	Функція корекції інструмента (друга функція інструмента)
<i>E</i>	Кутове переміщення навколо спеціальної осі або друга функція подачі
<i>F</i>	Функція подачі
<i>G</i>	Підготовча функція
<i>H</i>	Не визначене
<i>I, J, K</i>	Параметри інтерполяції або крок різі вздовж осей <i>X, Y, Z</i>
<i>L</i>	Не визначене
<i>M</i>	Допоміжна функція
<i>N</i>	Номер кадру
<i>P, Q</i>	Третя функція переміщення вздовж осей <i>X, Y</i> або параметри корекції інструмента
<i>R</i>	Переміщення на швидкому ході вздовж осі <i>Z</i> або параметр корекції інструмента
<i>S</i>	Швидкість головного руху
<i>T</i>	Функція інструмента
<i>U, V, W</i>	Другі функції переміщення вздовж осей <i>X, Y, Z</i>
<i>X, Y, Z</i>	Переміщення вздовж осей <i>X, Y, Z</i>

Таблиця 7.2

Символи у інформаційних словах кадру

Символ	Назва	Значення
ГТ	Табуляція	Символ, який управляє переміщенням у наступну знакову позицію того ж рядка
ПС	Переведення рядка	Символ, який відображається переведенням на наступний рядок, але діє як кінець кадру УП
%	Відсоток	Початок УП. Застосовується для зупинки носія даних при зворотному перемотуванні
(Кругла ліва дужка	Інформація, що йде за цим знаком, не опрацьовується на верстаті. Призначена

		для позначення початку коментаря
)	Кругла права дужка	Інформація, що йде за цим знаком, опрацьовується на верстаті. Призначена для позначення кінця коментаря
+	Плюс	Математичний знак визначення розмірних значень
-	Мінус	Математичний знак визначення розмірних значень
.	Крапка	Десятковий знак у розмірному значенні
:	Двокрапка	Головний кадр УП
/	Похила риска	Пропуск кадру, який позначає, що наступна за ним інформація до символу “Кінець кадру” може опрацьовуватися на верстаті тільки в залежності від положення перемикача на пульті пристрою ЧПУ. Перед символами “Номер кадру” і “Головний кадр” символ діє на цілий кадр УП

При розміщенні на носії даних декількох УП перед символом “Початок програми” першої УП допускається записувати ще один символ “Початок програми”. Після символу “Початок програми” вся інформація сприймається як управляюча, якщо не застосовані символи початку і кінця коментаря. Коментарі відображаються у круглих дужках і розміщуються після номера програми між кадрами. Текст програми складається з кадрів, серед яких можуть бути: головний (позначений символом “:”), додаткові (починається з символу “Номер кадру”), пропущені за наявності підтвердження, з панелі оператора (номер кадру додатково позначається символом “/”), а також звернення до підпрограм (за допомогою слова *L* з номером підпрограми).

За необхідності працювати в режимі “Пропуск кадру” (наприклад, для здійснення налагоджувальних переходів при налагодженні верстата і виключення цих переходів після закінчення налагодження) перед символами “Номер кадру” і “Головний кадр” треба записувати символ “Пропуск кадру”.

Програма повинна закінчуватися словом “Кінець програми” – *M02*. Перед фізичним кінцем перфострічки рекомендується розміщати слово “Кінець інформації” – *M30*. Інформація, поміщена після *M30*, не сприймається і не обробляється пристроєм ЧПУ. Перед символом “Початок програми” і після слів “Кінець програми” і “Кінець інформації” на перфострічці рекомендується залишати ділянки із символом “Пусто”.

Отже, структуру програми можна умовно подати послідовністю записів, наведених у табл. 7.3.

Формат управляючої програми. Переходячи до формату

управляючої програми ЧПУ, відмітимо, що її основним інформаційним блоком є кадр. Структуру кадру для верстата з ЧПУ визначає формат УП.

Формат УП – це умовний запис кадру з максимальним обсягом інформації, що визначає набір застосовуваних слів, порядок їх розташування та обсяг інформації кожного слова.

Існує зручний формалізм, який дозволяє описати формат кадру управляючої програми, прийнятий для конкретного пристрою ЧПУ. Цей формалізм, по-перше, вводить перелік використовуваних символів, а по-друге, вказує, як повинна бути побудована числова частина кожного слова.

Таблиця 7.3

Структура управляючої програми верстата з ЧПУ

Запис у програмі	Значення запису
коментар % номер програми (коментар) :..... N..... M02	Початок першої УП і коментар до неї Коментар по тексту програми Головний кадр УП Додатковий кадр УП з кінцем програми
коментар % номер програми /:..... /N..... M02 M30	Початок наступної УП і коментар до неї Головний кадр УП, який виключено Додатковий кадр УП, який виключено Кінець програми та кінець інформації

Правила побудови формального запису УП такі:

- слова позначають символами їх адрес і описом їх числової частини; позначення впливають у прийнятій для кадру послідовності;
- якщо за адресою безрозмірного слова можна опустити нулі, що знаходяться перед першою значущою цифрою, то числову частину безрозмірного слова описують нулем і цифрою, що вказує на найбільше число десяткових розрядів числової частини;
- якщо за адресою безрозмірного слова не можна опускати незначущі нулі, то числову частину безрозмірного слова описують цифрою, що вказує на фіксований формат розрядів числової частини;
- один варіант опису числової частини кожного слова “Розмірних переміщень” полягає в тому, що за адресою слова записують дві цифри, перша з яких показує число розрядів перед десятковою крапкою, що відокремлює цілу частину від дробової, а друга – число розрядів після крапки;
- інший варіант опису числової частини кожного слова “Розмірних переміщень” полягає в тому, що за адресою кожного слова

записують три цифри, перша чи остання з яких нуль – в залежності від того, чи можна опустити нулі перед першою чи останньою значущою цифрою, а інші дві цифри мають той же зміст, що й у попередньому варіанті;

– якщо абсолютні розміри завжди додатні, то між адресою та наступним за ним числом не ставлять ніякого знака;

– якщо розміри можуть бути додатними і від’ємними, то між адресою та наступним за ним числом ставлять знак “плюс”; при цьому покладається, що знак плюс може бути опущений і буде матися на увазі за замовчуванням;

– якщо розміри можуть бути додатними і від’ємними, причому знак “плюс” опускати не можна, то між адресою та наступним за ним числом ставлять знаки “плюс” і “мінус”.

Для пояснення правил розглянемо такий формальний запис, що розкриває структуру і формат деякої управляючої програми:

% : / N3 G2 X + 053 Y + 053 Z + 053 F031 S04 T04 M2.

З наведеного прикладу витікає, що пристрій ЧПУ сприймає десяткову крапку, символ початку програми, головного кадру і пропуску кадру, а також має таке кодування інформації:

N3 – тризначний номер кадру з ведучими нулями у слові;

G2 – двозначна підготовча функція з ведучими нулями у слові;

X+053 – восьмирозрядне переміщення по осі X зі знаком, п’ять цифр до десяткової крапки і три після, ведучі нулі дозволено опускати;

F031 – швидкість подачі з трьома цифрами до десяткової крапки, одна після і ведучі нулі дозволено опускати;

T04 – чотиризначна функція інструмента, ведучі нулі дозволено опускати;

M2 – двозначна допоміжна функція.

У посібнику до конкретного пристрою ЧПУ, прикладеного до верстата, завжди описуються формат кадру та УП. Умовний запис формату УП показує, як необхідно формувати його при конкретному програмуванні для даного верстата.

У посібнику до верстата наводяться також наступні дані: перелік і призначення всіх реалізованих підготовчих і допоміжних функцій; таблиці кодів швидкостей подачі і головного руху; таблиці кодових номерів позицій інструмента; перелік номерів коректорів із вказанням їх призначення та особливостей застосування; межі розмірних переміщень по всіх осях координат; перелік усіх сприйманих і реалізованих символів кодового набору; перелік і кодові номери всіх підпрограм, що зберігаються в пам’яті пристрою ЧПУ.

Зміст кадру УП також регламентований. Кожен кадр повинен містити: слово “Номер кадру”, інформаційні слова, символ “Кінець кадру”. В залежності від виконання верстата зміст кадру може розрізнятися, але послідовність запису символів повинна відповідати стандарту на формат кадру. При використанні символів табуляції вони

проставляються перед кожним інформаційним словом (за винятком слова “Номер кадру”) у кадрі УП.

Інформаційні слова в кадрі рекомендується записувати в такій послідовності: “Підготовча функція” (G); “Розмірні переміщення” ($X, Y, Z, U, V, W, P, Q, R, A, B, C$); “Параметр інтерполяції або крок різі” (I, J, K); “Функція подачі” (якщо відноситься тільки до визначеної осі, то повинно впливати безпосередньо за словом “Розмірне переміщення” по цій осі, якщо відноситься до двох і більше осей – за останнім словом “Розмірне переміщення”, до якого воно відноситься); “Функція головного руху” (S); “Функція інструмента” (T); “Допоміжна функція” (M). Слова “Розмірні переміщення” і “Параметр інтерполяції чи крок різі” не повинні повторюватися в одному кадрі УП.

Слово “Номер кадру” служить для позначення елементарної ділянки УП і є допоміжною інформацією, задається адресою N і цілим десятковим числом. У головному кадрі управляючої програми замість імені використовують символ “:”. Послідовна нумерація кадрів раціональна, але допускає будь-які переходи номерів, і обумовлюється тільки їх неповторюваність у межах однієї УП. Наприклад, якщо при редагуванні після кадру з номером 107 необхідно вставити кілька нових кадрів, їх можна нумерувати 10701, 10702, 10703 і т.д.

Підготовчі функції визначають режим роботи присторою ЧПУ. Слово “Підготовча функція” містить адресу G і умовний дворозрядний десятковий код, тобто передбачено 100 підготовчих функцій $G00$ – $G99$.

В пристроях ЧПУ нового покоління десятковий код функції – тризначний. Підготовчі функції розбиті на групи. У кадрі можна задавати тільки одну функцію з кожної групи. Підготовчі функції записуються в кадрі одна за одною у порядку зростання їх кодових номерів.

Значення підготовчих функцій зазначені в табл. 7.4. Ці значення відповідають рекомендаціям ISO, однак у конкретних пристроях ЧПУ можуть бути відхилення, що обумовлюється методикою програмування.

Таблиця 7.4

Значення підготовчих функцій

Функція	Назва	Значення	Група
$G00$	Позиціонування	Скасування координатного зв'язку, прискорене переміщення в задану точку	1
$G01$	Лінійна інтерполяція	Встановлення постійного відношення між швидкостями по осі координат	1
$G02$	Кругова інтерполяція	Рух виконавчого органу за годинниковою стрілкою, якщо дивитися з боку додатного	1

		напрямку осі, перпендикулярної до оброблюваної поверхні	
G03	Кругова інтерполяція	Рух проти часової стрілки	1
G04	Пауза	Витримка часу, задана в програмі або в самому пристрої ЧПУ поза програмою	1
G08	Розгін	Автоматичне збільшення швидкості руху до запрограмованого в кадрі	–
G09	Гальмування	Автоматичне зменшення швидкості руху до нуля чи до запрограмованого в наступному кадрі	–
G16*	Просторова інтерполяція	Об'єднання варіантів плоскої інтерполяції в одному кадрі	–
G17– G19	Площина обробки	Вибір площини обробки відповідно XY, XZ, YZ для задання площини інтерполяції, площини корекції на розміри інструмента	2
G25*	Циклічне повторення	Повторення групи кадрів чи програми	–
G26*, G27*	Умовний перехід	Перевірка виконання умов відповідно “більше”, “менше”	–
G28*, G29*	Безумовний перехід	Виконання переходу відповідно на кадр і на програму	–
G30*	Семафор	Встановлення семафора та його значення з метою синхронізації різних програм у пристроях ЧПУ ГВМ	–
G31*	Переривання	Переривання виконання програми на підставі перевірки значення семафора	–
G33, G34, G35	Різь	Вибір режиму нарізання різі відповідно з постійним, збільшуючим, зменшуючим кроком	1
G40	Скасування корекції	Скасування корекції контуру на розміри інструмента	3
G41, G42	Корекція на радіус фрези ліва і права	Встановлення корекції на діаметр (радіус) інструмента при контурному управлінні, коли інструмент знаходиться відповідно ліворуч або праворуч від оброблюваної поверхні, якщо дивитися в напрямку руху інструмента	3

G43, G44	Корекція довжини інструмента додатна і від'ємна	Вказівка, що значення корекції на довжину інструмента, встановлену на пульті, необхідно відповідно скласти або відняти від координати, заданої в кадрі	3
G45– G52	Корекція в площині двох координат при прямолінійному формоутворенні	Вказівка, що значення корекції, встановлену на пульті, необхідно врахувати для двох координат відповідно із знаками: +/+; +/-; -/-; -/+; 0/+; 0/-; -/0 (0 – корекція відсутня)	3
G53	Скасування зв'язку координатних систем	Скасування встановленого раніше зв'язку координатних систем верстата і деталі	4
G54– G59	Встановлення зв'язку координатних систем	Лінійний зсув нульової точки деталі відносно нуля верстата відповідно по осях X, Y, Z, XY, XZ, YZ. Здійснюється корекція довжини або положення інструмента на величину, встановлену з пульта	4
G60– G62	Точне позиціонування на стику кадрів	Обов'язкове відпрацьовування приводами кадрового завдання відповідно з малим, середнім і великим допуском на точність позиціонування до переходу до наступного кадру	5
G80	Скасування постійного циклу	Скасування заданого раніше циклу, що представляє собою послідовність переходів, які забезпечують виконання технологічно завершеної операції	6
G81– G89	Постійні цикли	Виклик постійного циклу, що описує технологічно типову послідовність переходів, параметри яких заздалегідь задані поза циклом	6
G90	Абсолютний розмір	Відлік переміщень в абсолютній системі координат з початком у нульовій точці системи	7
G91	Розмір у приростах	Відлік переміщень у відносній системі координат з початком у попередній запрограмованій точці	7
G92	Встановлення абсолютних	Встановлення абсолютних нагромаджувачів поточних	-

	нагромаджувачів	координат нерухомого виконавчого органу на основі визначення його фактичного положення	
G93	Швидкість подачі у функції, оберненої часу	Вказівка, що число, що впливає за адресою P , дорівнює оберненому значенню часу в хвилинах, необхідному для відпрацьовування кадру	8
G94, G95	Розмірність подачі	Вказівка, що швидкість подачі дана відповідно в мм/хв і мм/об	8
G96	Постійна швидкість різання	Вказівка, що число, що впливає за адресою S , дорівнює швидкості різання в м/хв. Відбувається автоматична підтримка постійної швидкості за рахунок регулювання частоти обертання шпинделя	9
G97	Розмірність частоти обертання	Вказівка, що число, що впливає за адресою S , дорівнює частоті обертання шпинделя в об/хв	9
* – вказівка на неузаконену підготовчу функцію. Невизначені підготовчі функції призначені для індивідуального використання в конкретних пристроях ЧПУ. Якщо деякі функції не використовуються, то вони стають невизначеними			

Звичайно підготовчі функції поєднують у групи взаємовиключаючих, тобто ортогональних, функцій. У групі ортогональних підготовчих функцій діючою є одна до її скасування чи заміни іншою підготовчою функцією з тієї ж групи. Дія підготовчої функції, як правило, виходить за рамки одного кадру. Одночасно діяти можуть по одній підготовчій функції з кожної групи ортогональних функцій.

Слово “Розмірні переміщення” призначене для задання геометричної інформації. Розмірні переміщення визначають або координати опорних точок траєкторії інструмента – абсолютні розміри, або розміри в приростах – відносні розміри. У слові “Розмірні переміщення” можуть бути використані такі адреси: $X, Y, Z, U, V, W, P, Q, R, I, J, K, A, B, C, D, E$.

Кількість слів “Розмірні переміщення” у кадрі визначається кількістю встановлюваних значень координат. Траєкторія інструмента може містити в собі ділянки переміщень на швидкому ході ($G00, G60, G61, G62$), ділянки лінійної ($G01$), кругової інтерполяції ($G02, G03$), які інструмент проходить на робочій подачі. З огляду на те, що початкова точка кожної з ділянок, крім першої, є одночасно кінцевою точкою попередньої ділянки, у кадрах УП задається інформація тільки про кінцеві точки. Спосіб задання значень розмірних переміщень

вибирається підготовчими функціями G90 (абсолютна система) і G91 (відносна система).

Числові частини слів у складі “Розмірних переміщень” можуть бути побудовані одним із способів в залежності від внутрішньої організації обробки інформації в пристрої ЧПУ:

1) розміри записуються цілими числами зі знаками із врахуванням дискретності задання розмірів для конкретних пристроїв ЧПУ. Знак “+” можна не вказувати, наприклад, X288400.

2) лінійні переміщення задають в міліметрах та їх десяткових частках, кутові розміри – у радіанах. Якщо фактичне положення десяткової крапки зафіксоване і визначене в характеристиці формату конкретного пристрою ЧПУ, то в записі значення розмірного переміщення десяткова крапка не передбачена, інакше положення десяткової крапки вказується явно. Наприклад, X0288400 при форматі X+43, X288.4 в явному вигляді.

Слово “*Функція подачі*” може визначати як результуючу (контурну) швидкість подачі, так і складові цієї швидкості, що розкладені по координатних осях. Результуючу швидкість подачі записують під адресою *F* після всіх слів “Розмірних переміщень”. Складову швидкості подачі, що відноситься до визначеної координатної осі, записують під адресою *F* відразу ж після слова, що задає розмірне переміщення по цій осі. Одиниця вимірювання швидкості подачі кодується підготовчими функціями G94 (мм/хв) або G95 (мм/об).

Слово “*Швидкість головного руху*” записується з адресою *S* і числовим значенням, яке визначає лінійну швидкість (мм/хв) точки контакту інструмента з деталлю або частоту обертання шпинделя (об/хв). Одиниця вимірювання швидкості головного руху кодується підготовчими функціями G96 (мм/хв) або G97 (об/хв).

Для кодування значень швидкостей подачі і руху головного приводу застосовуються методи прямого позначення, геометричної, арифметичної прогресії та символічний. Формат значень встановлюється методом кодування швидкості, призначення яких наведено в описі логічної задачі ЧПУ.

В пристроях ЧПУ більш рекомендований до застосування метод прямого позначення.

При кодуванні швидкості подачі потрібно враховувати, що при заданні подач більших 300 мм/хв, необхідно віддавати перевагу режиму встановлення швидкості з гальмуванням або розгоном до фіксованої швидкості. Тим самим виключаються миттєві перепади швидкості. Для цього застосовують підготовчі функції G08 і G09. Крім того, в пристроях ЧПУ нового покоління такий режим встановлення швидкості програмується в слові “*Функція подачі*” з використанням методу арифметичної прогресії, додавши спереду до кода значення швидкості один розряд, який задає режим зміни швидкості: 0-

нормальний режим, 4 – гальмування або 1 – розгін, 2 – гальмування, 7 – швидкий хід.

Слово “*Функція інструмента*” використовується для вказівки інструмента і коректора. У цьому слові з адресою *T* записується кодове число з однією чи двома групами цифр. У першому випадку слово задає тільки номер інструмента або його позицію, а коректор для цього інструмента визначається іншим словом з адресою *D*. В другому випадку, друга група цифр визначає номер коректора довжини або діаметра інструмента. Наприклад, у слові *T1218*: *T* – адреса, 12 – номер інструмента, 18 – номер коректора. Якщо програмується номер інструмента без вказівки коректора, то друга група цифр містить нулі (*T1200*), а якщо програмується коректор для заданого в одному з попередніх кадрів інструмента, то нулі містить перша група цифр (*T0018*). Застосування у кадрі цього слова вказує пристрою ЧПУ на виконання підготовчих дій, пов’язаних з пошуком і вибором нового інструмента, що буде використаний для обробки у наступних кадрах програми. Процедура заміни інструмента на робочій позиції верстата виконується іншою функцією – допоміжною *M06*, яка викликає стандартний цикл заміни інструмента.

Слово “*Допоміжна функція*” містить адресу *M* і двозначний числовий код функції. Допоміжні функції визначають команду електроавтоматики або пристрою ЧПУ і призначені для організації дискретного управління: об’єктом (*M00–M02*), головним приводом (*M03–M05*, *M13–M14*, *M19*, *M38–M39*), приводом подачі (*M36–M37*, *M40–M45*), системою охолодження (*M07–M09*, *M50–M51*), інструментом (*M06*, *M55–M56*), деталлю (*M10–M11*, *M60–M62*, *M71–M72*). Перелік основних допоміжних функцій та їх значення наведені в табл. 7.5.

Більшість допоміжних функцій виконується до початку переміщень, запрограмованих у тому ж кадрі, і діє до скасування або заміни їх командами аналогічного призначення. В одному кадрі в порядку зростання кодових номерів може бути записано кілька команд різним виконавчим органам верстата з ЧПУ.

Таблиця 7.5

Загальноприйняті значення допоміжних функцій

Функція	Назва	Значення
<i>M00</i>	Запрограмована зупинка	Зупинення без втрати інформації по закінченні відпрацювання кадру, після чого відбувається зупинення шпинделя, системи охолодження, подачі
<i>M01</i>	Зупинка з підтвердженням	Аналогічно <i>M00</i> , але виконується тільки при попередньому підтвердженні з пульта
<i>M02</i>	Кінець програми	Завершення відпрацювання програми деталі та зупинення шпинделя, подачі,

		включення системи охолодження після виконання всіх команд у кадрі
M03 (M04)	Обертання шпинделя за годинниковою стрілкою (проти годинникової стрілки)	Включення шпинделя у напрямку, при якому гвинт з правою різью вгвинчується у заготовку (вигвинчується із заготовки)
M05	Зупинка шпинделя	Зупинення шпинделя, вимкнення системи охолодження
M06	Заміна інструмента	Виклик циклу заміни інструмента
M07, M08, M50, M51	Ввімкнення системи охолодження	Ввімкнення відповідно системи охолодження 1, 2, 3, 4
M09	Вимкнення системи охолодження	Відміна команд M07, M08, M50, M51
M10 (M11)	Затискання (розтискання)	Команда на затискання (розтискання) оброблюваної деталі, приладу або шпинделя
M13 (M14)	Обертання шпинделя	Включення обертання шпинделя за годинниковою стрілкою (проти годинникової стрілки), ввімкнення системи охолодження
M15 (M16)	Перехід “у плюс” (“у мінус”)	Вибір напрямку швидкого ходу або подачі
M19	Зупинка шпинделя	Зупинення шпинделя у заданій кутовій позиції
M20	Кінець стрічки	Завершення відпрацювання програми деталі та зупинення шпинделя, вимкнення системи охолодження після виконання всіх команд у кадрі. Встановлення у вихідне положення
M31 (M32)	Відміна блокування (ввімкнення блокування)	Команда на тимчасову відміну блокування (ввімкнення блокування)
M36, M37	Діапазон подачі	Задання діапазону подачі шляхом перемикання кінематичного зв'язку
M38, M39	Діапазон частоти	Задання діапазону частоти обертання шпинделя шляхом перемикання

	обертання шпинделя	кінематичного зв'язку
M40... M45	Перемикання передач	Перемикання у приводі подачі, якщо є засоби перемикання
M55, M56	Зміщення інструмента	Лінійні зміщення інструмента у положення 1, 2, що визначаються жорсткими механічними та іншими видами упорів або датчиків
M60	Зміна заготовки	Ввімкнення циклу, який забезпечує заміну заготовки у робочій позиції
M61, M62	Зміщення заготовки	Лінійне зміщення заготовки у положення 1, 2, які визначаються електричними, механічними або іншими видами датчиків, упорів
M71, M72	Кутове зміщення заготовки	Кутове зміщення заготовки у положення 1, 2, які визначаються електричними, механічними або іншими видами датчиків, упорів

7.2.2. Опис типових операцій обробки у пристрої ЧПУ

В пристрої ЧПУ операції обробки здійснюються на підставі застосування підготовчих функцій в УП.

Підготовчі функції пов'язані з режимами роботи, формою руху, циклами обробки тощо. Перелік цих функцій та їх рекомендоване кодування наведені в табл. 7.4. Об'єднання функцій у групи по ознаках визначає можливість взаємного скасування їх дії. Перелік означає, що функція діє тільки в тому кадрі, у якому вона зазначена. Якщо в одному кадрі необхідні кілька підготовчих функцій, то їх записують у порядку зростання номерів їхніх ознак. Крім того, деякі функції визначають тільки тип виконуваної операції, параметри якої задаються іншими інформаційними словами, тобто повний опис дії у кадрі складається з комплексу взаємопов'язаних і взаємодоповнюючих за змістом інформаційних слів.

Розглянемо опис типових операцій обробки у пристрої ЧПУ.

Відлік координат. Для визначення поточної системи координат, в якій будуть програмуватися координати переміщень інструмента, застосовують ортогональні підготовчі функції G53–G59. Якщо в кадрі задана функція G53, то переміщення задається в системі координат верстата, інакше – в одній з робочих систем координат, що визначаються функціями G54–G59. При цьому початок робочих систем координат зсувається по кожній координаті лінійно відносно системи координат верстата у відповідності із попередньо заданими константами з пульта системи.

Узгодження координат. Для узгодження системи координат

деталі (у якій задані координати опорних точок траєкторії інструмента) із системою координат верстата поряд із зсувом нуля відліку переміщень робочих органів (встановлюються вручну з пульта системи) застосовується програмований зсув нуля, який кодується підготовчою функцією G92 у кадрі, що відпрацьовується без переміщення робочих органів верстата. За допомогою цієї функції початок робочої системи координат можна перемістити в іншу точку в межах робочого простору верстата. Значення координат, що задані в поточному кадрі з функцією G92, представляють собою координати кінцевої точки попереднього кадру у новій системі координат.

Програмований зсув нуля використовується також для врахування в УП змін розміру інструмента при його заміні або зношуванні. Наприклад, після повороту різетримача (рис. 7.2, а) для заміни свердла різцем функція G92 (у кадрі обліку зсуву нуля) записується в наступному вигляді (i – номер кадру):

$N\{i\} \dots G92 X\{x_0 - x_1\} Z\{z_0 - z_1\} \dots$

Величини X і Z наведені (у фігурних дужках) в приростах, а значення i визначає номер кадру. Три крапки показують, що в кадр можуть входити також інші функції.

Лінійне переміщення (рис. 7.2, б) записується у вигляді:

$N\{i\} \dots G91 X\{x_1 - x_0\} Y\{y_1 - y_0\} Z\{z_1 - z_0\} \dots$

Функція G91 показує, що величини задані в приростах. Якщо ж вони задаються в абсолютних значеннях, то замість функції G91 варто вказати функцію G90:

$N\{i\} \dots G90 X\{x_1\} Y\{y_1\} Z\{z_1\} \dots$

Коли в пристрої ЧПУ реалізована можливість працювати тільки з одним із наведених способів задання розмірів, то ці підготовчі функції не вказуються.

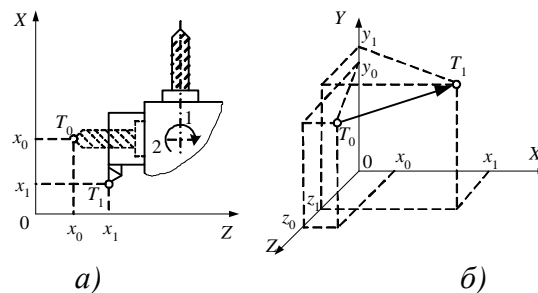


Рис. 7.2. Програмований зсув нуля (а)
і лінійне переміщення інструмента (б)

Більшість пристроїв ЧПУ третього покоління дозволяє виконати програмування обома методами. Однак рекомендується користуватися заданням у приростах (функція G91), оскільки в цьому випадку зберігається пряме співвідношення між переміщенням інструмента, зазначеним у кадрі, і його фактичним переміщенням. Необхідно пам'ятати, що незалежно від того, чи переміщається шпиндель з інструментом відносно заготовки або заготовка відносно інструмента,

при програмуванні передбачається, що переміщається інструмент.

Траєкторія руху інструмента (еквідистанта) включає ділянки прискорених переміщень, а також ділянки лінійної або кругової інтерполяції, які інструмент проходить зі швидкістю робочої подачі. Спосіб руху вздовж заданої траєкторії визначається підготовчою функцією, а спосіб задання геометричних параметрів ділянки траєкторії залежить від способу руху, прийнятої координатної системи, прийнятого набору параметрів ділянки, площини обробки.

Позиціонування на швидкому ході кодується функцією G00. На ділянці прискореного переміщення у кінцевому результаті потрібно лише потрапити в задану точку T_1 , а проміжна траєкторія несуттєва (рис. 7.3, а). Тому у випадку руху з точки T_0 у точку T_1 запис у кадрі має вигляд:

– в абсолютній системі:

$N\{i\} \text{ G00 } \dots \text{ G90 } X\{x_1\} Y\{y_1\} Z\{z_1\} \dots$

– у відносній системі:

$N\{i\} \text{ G00 } \dots \text{ G91 } X\{x_1 - x_0\} Y\{y_1 - y_0\} Z\{z_1 - z_0\} \dots$

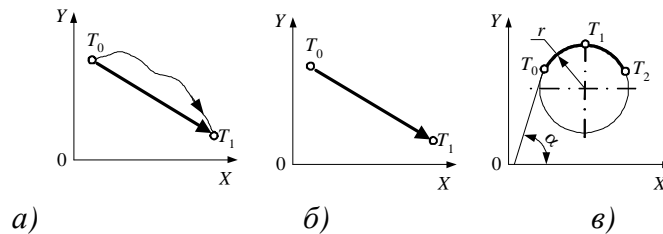


Рис. 7.3. Способи руху вздовж заданої траєкторії

При такому записі рух виконується одночасно по всіх координатах з максимальною швидкістю. Тому у випадку руху по трьох координатах результуюча швидкість більша за максимальну швидкість по координаті в $\sqrt{3}$ раз.

У тих випадках, коли позиціонування доцільно виконувати з почерговим рухом по координатах, замість зазначеного кадру в абсолютній системі записують три:

$N\{i\} \dots \text{ G00 } \dots \text{ G90 } X\{x_1\} \dots$

$N\{i + 1\} \dots Y\{y_1\} \dots$

$N\{i + 2\} \dots Z\{z_1\} \dots$

Таке переміщення використовується при виконанні встановлюючих переміщень інструмента, відведеннях, підведеннях інструмента, переходах від однієї суміжної ділянки обробки до іншої, заміні інструмента тощо, коли немає контакту інструмента з деталлю і не потрібна висока точність руху на ділянці прискореного переміщення.

Переміщення на робочій подачі задається ділянками інтерполяції: при русі по прямій – лінійна інтерполяція, при русі по дузі кола – кругова інтерполяція. Швидкість робочої подачі задається словом “Функція подачі”. У це слово входить адреса F і величина

подачі, що діє на всіх ділянках інтерполяції до її заміни.

Для ділянки переміщення з точки T_0 у T_1 з лінійною інтерполяцією запис у кадрі має вигляд:

– в абсолютній системі:

$N\{i\} G01 \dots G90 X\{x_1\} Y\{y_1\} Z\{z_1\} \dots$

– у відносній системі:

$N\{i\} G01 \dots G91 X\{x_1 - x_0\} Y\{y_1 - y_0\} Z\{z_1 - z_0\} \dots$

На ділянці лінійної інтерполяції (рис. 7.3, б) рух інструмента здійснюється з максимальним наближенням до заданої прямої T_0T_1 .

На ділянці кругової інтерполяції (рис. 7.3, в) траєкторія руху інструмента визначається геометричними параметрами дуги T_0T_1 . В усіх випадках початковим положенням інструмента є точка T_0 , у яку інструмент потрапив за результатами відпрацювання попереднього кадру. Таким чином, координати точки T_0 вважаються відомими.

Виходячи з цієї обставини, у слові “Розмірні переміщення” задають такі геометричні параметри ділянки траєкторії, при яких його просторове положення стало б визначеним.

Для цього вказується напрямок руху за годинниковою стрілкою або проти неї функціями $G02$, $G03$, площина обробки – функціями $G17$, $G18$, $G19$, проекції радіус-вектора дуги – розмірними переміщеннями I , J , K , та координати кінцевої точки:

– в абсолютній системі:

$N\{i\} G02 G17 \dots G90 X\{x_1\} Y\{y_1\} I\{x_c - x_0\} J\{y_c - y_0\} \dots$

– у відносній системі:

$N\{i\} G02 G17 \dots G91 X\{x_1 - x_0\} Y\{y_1 - y_0\} I\{x_c - x_0\} J\{y_c - y_0\} \dots$

Кругова інтерполяція і відповідний рух може вестись тільки у площині двох координат XY ($G17$), XZ ($G18$), YZ ($G19$). Всі значення координат кінцевої точки і проекції радіус-вектора необхідно задавати явно, в тому числі і нульові значення. Проекції радіус-вектора задають завжди в приростах із врахуванням знаків проекції.

В пристроях ЧПУ класу CNC рух по колу може програмуватися заданням радіуса R кола і кута нахилу a початкової точки:

– в абсолютній системі:

$N\{i\} G02 G17 \dots G90 X\{x_1\} Y\{y_1\} I\{a\} J\{R\} \dots$

– у відносній системі:

$N\{i\} G02 G17 \dots G91 X\{x_1 - x_0\} Y\{y_1 - y_0\} I\{a\} J\{R\} \dots$

Проте, спосіб задання дуги кола кінцевою точкою та центром у відносній системі найбільше розповсюджений в пристроях ЧПУ. При цьому більшість ЧПУ передбачає задання в одному кадрі інформації про частину дуги, що розташована в межах тільки одного квадранта кола, тобто рух по дузі у двох і більше квадрантах необхідно подавати декількома кадрами.

Нарізання різі з постійним кроком (рис. 7.4) кодується підготовчою функцією $G33$. Траєкторія руху при виконанні цієї

операції буде комплексною і повинна бути записана у чотирьох кадрах:

– прискорене підведення (T_0T_1):

$N\{i\} G00 G90 X\{x_1\} \dots$

– нарізання різи (T_1T_2) із кроком, що задається параметром K :

$N\{i+1\} G33 \dots Z\{z_2\} K\{c\} \dots$

– прискорене відведення (T_2T_3):

$N\{i+2\} G00 \dots X\{x_3\} \dots$

– відведення по осі Z (T_3T_0):

$N\{i+3\} \dots Z\{z_0\} \dots$

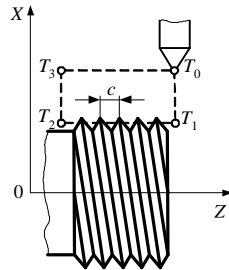


Рис. 7.4. Нарізання різи з постійним кроком

Гальмування наприкінці відпрацьовування кадру використовується досить часто для зменшення динамічної помилки приводу, наприклад, при реверсуванні.

На рис. 7.5 показана ділянка фрезерування канавки, у якій при підході до точки T_1 потрібно знизити швидкість. У цьому випадку рух по траєкторії $T_0-T_1-T_2$ записується двома кадрами із вказанням підготовчої функції $G09$:

$N\{i\} G01 G09 G90 X\{x_1\} F \dots$

$N\{i+1\} \dots Y\{y_2\} \dots$

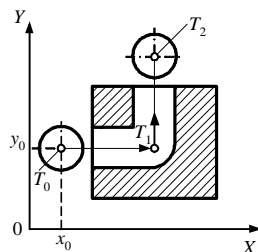


Рис. 7.5. Частина траєкторії з гальмуванням

У багатьох сучасних пристроях з ЧПУ передбачене також безеквідистантне програмування, тобто задання інформації не по траєкторії руху центра інструмента (еквідистантне), а по параметрах оброблюваної поверхні, зазначеної в кресленні деталі. При цьому розрахунки, пов'язані з обчисленням еквідистантного контуру, виконується автоматично пристроєм ЧПУ. Оператору необхідно тільки ввести величини радіуса і довжини інструмента з пульта ЧПУ.

Корекція на інструмент ліва і права. Корекція на радіус інструмента при обробці криволінійного контуру кодується функціями $G41$, $G42$ в залежності від розташування інструмента (ліворуч або

праворуч від контуру, якщо дивитися у напрямку руху інструменту). Наприклад, траєкторія $T_0-T_1-T_2-T_3-T_4-T_5-T_6-T_0$ (рис. 7.6, а), що розрахована для обробки зовнішнього контуру фрезою радіусом R , коригується за допомогою функції G41 і коректора № 1 (T0001 або D01), до якого з пульта системи заноситься зміна радіуса ΔR (в абсолютній системі):

```
N{i} G00 G17 G90 X{x1} Y{y1}
N{i+1} G01 G41 X{x2} ... T0001
N{i+2} X{x3} ...
N{i+3} G02 X{x4} Y{y4} I0 J{y4-y3} ...
N{i+4} G01 Y{y5} ...
N{i+5} G40 Y{y6} ...
N{i+6} G00 X{x0} Y{y0} ...
```

Ця ж траєкторія для обробки внутрішнього контуру (рис. 7.6, б) коригується функцією G42, а УП відрізняється від попередньої кадром:

```
N{i+1} G01 G42 X{x2} ... T0001.
```

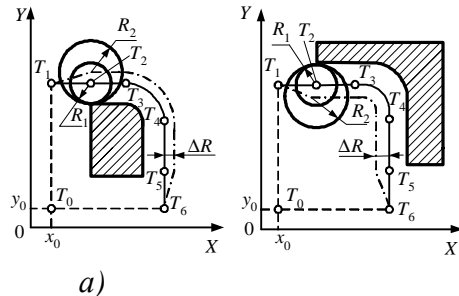


Рис. 7.6. Корекція на інструмент ліва і права

Корекція на інструмент додатна і від'ємна. Підготовчі функції G43, G44 призначені для корекції довжини або радіуса інструмента в площині обробки при прямокутному формоутворенні. Функції обираються відповідно для додавання або вирахування із заданої у кадрі координати величини, що занесена в коректор з пульта інструмента. Таким чином, корекція додатна або від'ємна подається в кадрі функціями G43, G44 і словом T (або D), які записуються в суворій послідовності перед кожною коригованою координатою. Відміна корекції здійснюється функцією G40 або заданням D00.

Функція G40 відмінює усі види корекції по всіх координатах, а слово D00 відмінює корекцію тільки по тій координаті, перед якою вона була задана і не анулює задання G43, G44. Корекція вважається від'ємною (G44), якщо координата центра інструмента менша, ніж координата контуру і навпаки.

Кадри задання корекції інструмента на радіус в абсолютній системі має вигляд (рис. 7.7, а):

```
N{i} G90 G00 G44 D01 X50 G44 D01 Y40
N{i+1} G01 G43 Y400 F1000
N{i+2} G43 X150
```


$N\{i + 3\} \text{ G44 Y400}$

$N\{i + 4\} \text{ G44 X50}$

$N\{i + 5\} \text{ G00 D00 X0 D00 Y0}$

Для корекції інструмента на довжину (рис. 7.7, б) в коректор заноситься абсолютна величина різниці між розрахунковою та дійсною довжиною інструмента $|z_0 - z_1|$ або $|z_0 - z_2|$, і в УП записується один з кадрів в залежності від того, коротший чи довший запрограмованого встановлений на верстаті інструмент: $N\{i\} \dots \text{G44Z}\{z_0\} \dots \text{T0201}$ або $N\{i\} \dots \text{G43Z}\{z_0\} \dots \text{T0201}$.

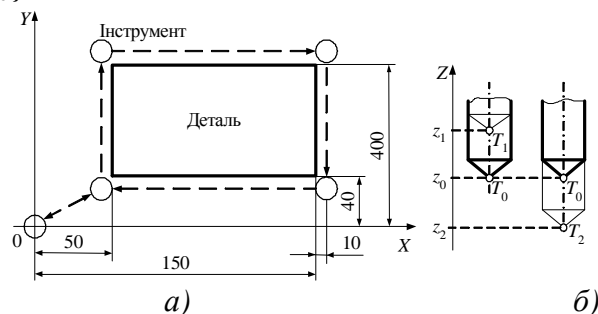


Рис. 7.7. Корекція на інструмент додатна і від'ємна із врахуванням радіуса і довжини

Підготовча функція використовується також при записі інших підпрограм. Найбільше це відноситься до запису постійних циклів.

Рекомендоване кодування постійних циклів свердильно-розточувальної обробки використовує функції G81–G89.

При розробці УП наведені елементи програми потрібно розглядати в їх взаємозв'язку з попередніми кадрами.

7.2.3. Програмування технологічного процесу обробки деталі

Етапи розробки управляючих програм. Розробка управляючих програм у коді ISO-7bit, тобто на стандартній входній мові пристроїв ЧПУ, вимагає клопіткого добору технологічних рішень, трудомістких геометричних розрахунків, ретельного документування окремих етапів. При цьому необхідна велика довідкова і допоміжна інформація, що повинна бути методично впорядкована.

До етапів розробки управляючої програми і використовуваної при цьому інформації відносять: розробку маршрутної технології (маршрутна карта, креслення деталі та заготовки), складання плану операції (операційна карта, операційне креслення), розробку попережидної технології (карта налагодження), розрахунок траєкторії інструмента (технологічна карта, ескіз траєкторії), кодування і запис управляючої програми, контроль і налагодження управляючої програми (графік траєкторії).

Управляюча програма призначена для керування верстатом з ЧПУ при виконанні лише однієї з операцій маршрутного технологічного процесу, причому таке виконання може

супроводжувати зміна встановлень деталі у пристосуванні, зміна базової позиції деталі в робочому просторі верстата.

Головними технологічними елементами, з яких формується операція, є переходи.

Основні *технологічні переходи* представляють собою закінчений процес утворення кожної нової поверхні або поєднання поверхонь деталі при обробці одним інструментом.

Допоміжні переходи пов'язані зі зміною встановлення, позиції інструмента по завершенні деякого основного технологічного переходу.

При обробці з ЧПУ будемо вважати закінченою частиною технологічного переходу робочі та допоміжні (холості) проходи, тобто однократні переміщення інструмента вздовж контуру без зміни робочих режимів.

Таким чином, найважливішим компонентом структури проектного технологічного процесу обробки деталі є операція, елементами якої у свою чергу служать встановлення, позиції, переходи і проходи.

Методика ручного програмування. При ручному програмуванні по кресленню деталі складають таблицю з УП, в яку вносять у тимчасовій послідовності всі робочі рухи верстата, необхідні для обробки деталі. Кожен відрізок УП містить досить багато додаткової інформації (у вигляді команд, характерних для відповідного верстата), що враховує властивості оброблюваного матеріалу, розміри і положення інструмента, швидкість різання тощо. Потім отриману УП переносять на перфострічку.

Процес ручного програмування розглянемо на прикладі підготовки УП для чорнової обробки чавунної литої заготовки на токарно-револьверному верстаті з двома револьверними головками, розташованими на одному супорті (рис. 7.8). На шестигранній головці (з вертикальною віссю обертання) закріплюють стрижневі та розточувальні інструменти, а на круглій (з горизонтальною віссю обертання) – різці для зовнішньої обробки.

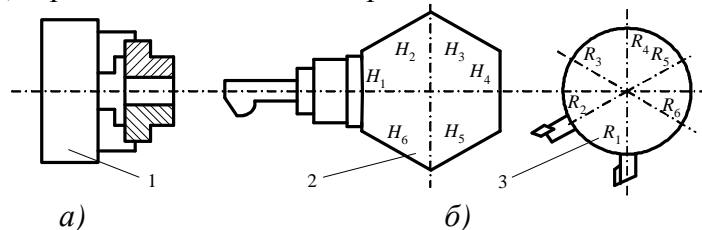


Рис. 7.8. Розташування інструментів в круглій 3 і шестигранній 2 головках та закріплення у патроні 1 заготовки

Спочатку розробляють технологічні переходи, вибирають інструменти та їх розташування на револьверних головках, визначають метод кріплення заготовки і режим різання для кожного переходу.

Позначивши буквами *A, B, C, D, E* та *F* (рис. 7.9, *a*) елементи заготовки, що підлягають обробці, і використовуючи дані про інструмент (розміри і різальні властивості якого наводяться в довідковій документації), технолог визначає інструмент для кожного переходу обробки і заповнює технологічну карту (табл. 7.6).

З табл. 7.6 видно: торці *A* і *E* обробляються інструментом, закріпленим у позиції R_1 круглої головки; елементи *B, C* і *D* обробляють інструментом, закріпленим у позиції R_2 круглої головки; отвір *F* розточують різцем, закріпленим у позиції H_1 шестигранної головки. Взаємне розташування інструментів у револьверних головках і заготовки також визначене. Після того, як вирішені технологічні питання, переходять до геометричних розрахунків.

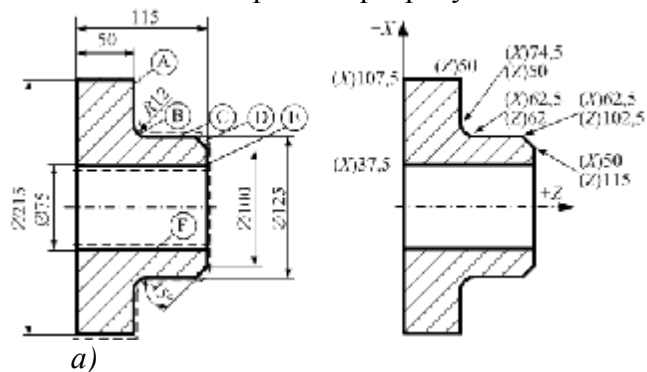


Рис. 7.9. Креслення заготовки (*a*) та її розміри в абсолютних координатах (*б*); штриховою лінією показаний контур заготовки до обробки

Якщо верстат працює в абсолютній системі координат, то креслярські розміри перераховують в абсолютні розміри в системі координат, у якій звичайно одна вісь збігається з лівим торцем деталі, а друга – з віссю деталі (рис. 7.9, *б*). Наприклад, абсолютний розмір (*X*) 37,5 відповідає радіусу отвору $\varnothing 75$ мм, абсолютний розмір (*Z*) 115 – довжині деталі 115 мм і т.д.

Таблиця 7.6

Технологічна карта обробки

Позиція револьверних головок		Технологічні дані				
круглої	шести- гранної	діаметр обробки , мм	швидкіс ть різання, мм/хв	частота обертан ня, об/хв	подача, мм/об	подач а, мм/хв
Торець <i>A</i>						
R_1		215	120	180	0,45	82
Торець <i>E</i>						
		125	120	300	0,4	120

Зняття фаски D і обточування циліндра C						
R_2		125	120	300	0,25	75
Обточування скруглення B						
		149	120	250	0,25	62
Розточування отвору F						
	H_1	75	90	390	0,25	100

Потім визначають траєкторію руху кожного інструмента.

Розглянемо, наприклад, траєкторію підрізного різця, встановленого в позиції R_1 (рис. 7.10, *a*). У вихідній точці центр радіуса заокруглення вершини різця має координати (X) 125 і (Z) 250 мм, а радіус заокруглення при вершині різця дорівнює 1,5 мм. Ці координати визначаються, як правило, за умови, що центр супорта знаходиться в “нульовій” точці, а також із врахуванням радіуса круглої револьверної головки і вильоту різця.

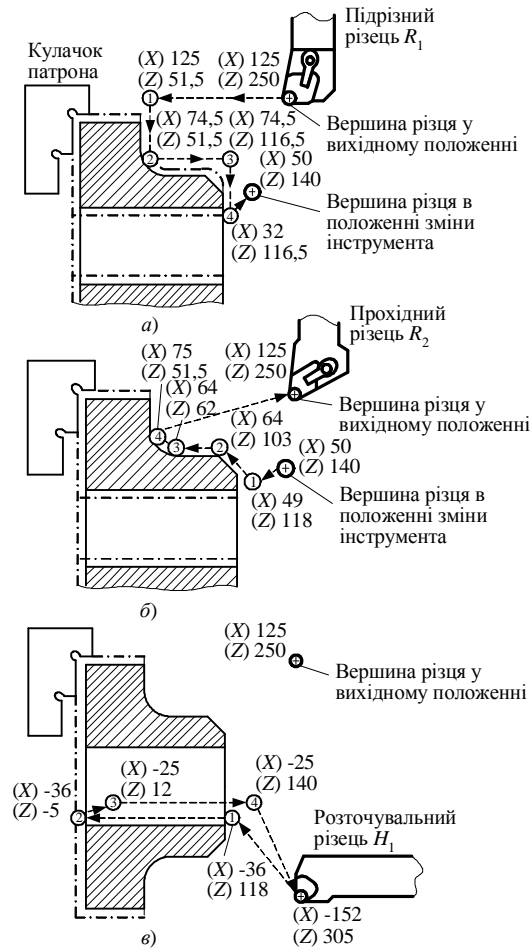


Рис. 7.10. Траєкторія руху змінних інструментів

З вихідної точки різць рухається зі швидкістю швидких переміщень по горизонталі до точки 1 з координатами (X) 125 і (Z) 51,5; координата (Z) враховує радіус заокруглення вершини різця 1,5

мм так, що відстань між передньою кромкою різця і точкою 1 складає 50 мм. З точки 1 різець опускається (зі швидкістю робочої подачі) до точки 2 з координатами (X) 74,5 і (Z) 51,5, здійснюючи підрізання торця. З точки 2 різець переміщається (зі швидкістю швидких переміщень) у точку 3 (початок підрізання правого крайнього торця) з координатами (X) 75 і (Z) 116,5. З точки 3 різець опускається (зі швидкістю робочої подачі) у точку 4 з координатами (X) 32 і (Z) 116,5, а потім переміщається в точку з координатами (X) 50 і (Z) 140, у якій револьверна головка повертається. У результаті цього повороту підводиться прохідний різець, встановлений у позиції R_2 , і починається виконання наступного технологічного переходу.

На рис. 7.10, б показана траєкторія прохідного різця, а на рис. 7.10, в – траєкторія руху розточувального різця, встановленого в позиції H_1 шестигранної револьверної головки.

Для визначення траєкторії прохідного різця, закріпленого в позиції R_2 круглої револьверної головки, при обробці фаски виконують наступні обчислення:

1. Розраховують координату (Z_2) точки 2 (рис. 7.10, б): $(Z_2) = 102,5 + a = 103,13$, де 102,5 – координата (Z) вершини фаски на деталі (рис. 7.9, б); $a = 0,63$ при $\alpha = 45^\circ$.

2. Розраховують координату (X_2) точки 2 (рис. 7.10, б): $(X_2) = 62,5 + 1,5 = 64$, де 62,5 – координата (X) вершини фаски (рис. 7.9, б); 1,5 – радіус вершини різця.

3. Розраховують координату (Z_1) точки 1 (рис. 7.10, б): $(Z_1) = (Z_2) - 12,5 + 2 \cdot 1,5 = 103,13 + 12,5 + 3 = 118,63$, де $12,5 = (Z)115 - (Z)102,5$ (рис. 7.9, б); 1,5 – радіус вершини різця.

4. Для визначення катета прямокутного трикутника з $\alpha = 45^\circ$ визначають різницю $(Z_1) - (Z_2) = 118,63 - 103,13 = 15,5$.

5. Розраховують координату (X_1) точки 1 (рис. 7.10, б): $(X_1) = (X_2) - 15,5 = 64 - 15,5 = 48,5$.

Після визначення координат траєкторії переміщення інструментів здійснюють кодування інформації УП за допомогою спеціальних бланків (табл. 7.7).

Розглянемо цей процес, вважаючи, що пристрій ЧПУ верстата має лінійну і кругову інтерполяцію. Для того щоб приступити до кодування, необхідно ознайомитися з наступними кодами команд, що наводяться в інструкції до верстата: N – номер фрази; S – код частоти обертання шпинделя, об/хв; M – допоміжні команди; G – підготовчі функції; $\pm X$, $\pm Z$ – коди переміщення; Y і K – коди положення дуг кіл; F – код подачі.

Номери кадрів управляючої програми позначаються трирозрядним числом, і тому перший кадр має номер 001. Далі йде код $S01^*$, що означає, що частота обертання шпинделя 180 об/хв, як і передбачено в табл. 7.6. (Тут і далі коди, позначені знаком “*”,

показують, що ці команди діють до появи нової команди, у даному випадку $S02$). Допоміжні команди: $M27$ – поворот круглої револьверної головки вверх; $M21$ – поворот головки до позиції R_1 . Підготовчий код $G90$ – позиціонування в нульовій точці.

Кадри $N001$ – $N008$ відносяться до управління роботою інструмента, закріпленого в позиції R_1 , кадри $N009$ – $N013$ – до інструмента, закріпленого в позиції R_2 , кадри $N014$ – $N022$ – до інструмента, закріпленого в позиції H_1 .

Таблиця 7.7

Номер кадру	Код швидкості шпинделя	Допоміжні команди		Підготовчий код	Коди переміщень		Код положення дуги		Код подачі
							$X_{напр}$	$Z_{напр}$	
N	S	M	M	G	X	Z	Y	K	F
Вибір частоти обертання шпинделя, поворот круглої револьверної головки вверх до позиції R_1 , вибір абсолютних координат									
$N001$	$S01^*$	$M27^*$	$M21$	$G90^*$					
Включення шпинделя вперед, прискорене підведення (7500 мм/хв) у вихідне положення									
$N002$		$M03^*$		$G73^*$	$X125,0$	$Z250,0$			$F90^*$
Охолодження, включення, прискорене підведення до точки I									
$N003$		$M08$				$Z051,5$			
Переміщення з швидкістю 120 мм/хв до точки 2									
$N004$				$G01^*$	$X074,5$				$F54$
Прискорене переміщення (2500 мм/хв) до точки 3									
$N005$						$Z116,5$			$F99$
Змінити частоту обертання шпинделя, перемістити з подачею 82 мм/хв в точку 4									
$N006$	$S02^*$				$X032,0$				$F50$
Прискорене переміщення (2500 мм/хв) в точку повороту револьверної головки									
$N007$					$X050,0$	$Z140,0$			$F99$
Поворот круглої револьверної головки в положення $R = 2$									
$N008$		$M28^*$	$M22$						

В табл. 7.7 наведена інформація тільки на перші вісім кадрів (всього чорнова обробка займає 22 кадри). Після заповнення цієї таблиці здійснюють запис УП відповідно до правил, викладених вище.

Наведений приклад є спрощеним поданням процесу програмування обробки деталей. Звісно, що для багатьох випадків

процес програмування більш громіздкий. Наприклад, при фрезерній обробці значно складніші розрахунки дуг кіл, заданих різним чином, прямих, дотичних до дуг тощо.

У зв'язку зі зростаючою складністю верстатів і розвитком обчислювальної техніки створюється велике число систем автоматизації підготовки УП на універсальних ЕОМ. Отримують розвиток також системи програмування, орієнтовані на використання мікро-ЕОМ.

Для ряду верстатів (токарних, шліфувальних) при обробці деталей простої конфігурації, використовується програмування біля верстата з мікропроцесорним пристроєм ЧПУ. У цьому випадку термін “ручне програмування” також передбачає використання елементів автоматизації підготовки УП.

В залежності від прийнятого методу підготовки УП змінюється і супровідна документація, що у загальному випадку включає: операційне креслення деталі; карту налагодження верстата та інструмента; операційно-розрахункову карту (при ручній підготовці УП); УП на програмоносію та її друк; графік траєкторії інструментів (у випадку креслення його при контролі УП); акт перевірки УП.

Операційне креслення містить вказівки про взаємне розташування базових поверхонь деталі, крипильного пристосування та інструмента, а також опис технологічних переходів.

Карта налагодження верстата заповнюється технологом-програмістом у процесі ручної підготовки УП чи видається при підготовці УП на ЕОМ. У карті, використовуваний при настроюванні інструмента поза верстатом на спеціальних пристосуваннях, записані координати вершин інструментів.

Операційна розрахунково-технологічна карта (табл. 7.7) призначена для ручної підготовки УП, передує написанню УП у заданому форматі чи безпосередній перфорації стрічки.

Наведений приклад ручного програмування відносився до конкретної деталі токарної обробки. Однак можна виділити ряд елементів, що, будучи загальними для різних УП, полегшують запис і розуміння підготовленої програми. До таких елементів потрібно віднести типові операції обробки, а також розширення можливостей кодування через спрощення програмування типових завдань при виконанні певних циклів обробки.

7.2.4. Розширення функцій програмування обробки деталей на верстатах з ЧПУ

Прагнення зменшити тривалість УП, скоротити час її розробки і число помилок призводить до необхідності розширення мови програмування та обчислювальних можливостей пристрою ЧПУ. Цьому сприяє застосування формальних параметрів, організацій переходів, підпрограм, стандартних циклів, що є своєрідним

оперативним архівом фрагментів програм обробки.

Формальні параметри (ФП) і дії над ними. Для розширення можливостей застосування технологічних програм і циклів, а також для введення умовних переходів використовуються формальні параметри.

Формальні параметри допустимо застосовувати з будь-якими адресами кадрів технологічної програми, за винятком адрес N , G , M , L , D . Найчастіше як адреса слова формальних параметрів використовується R у форматі $R2$.

Формальний параметр записується слідом за числом-кодом, зазначеним безпосередньо після адреси того слова, до якого застосований параметр. У цьому випадку результуюче чисельне значення адреси дорівнює алгебраїчній сумі числа і поточного значення формального параметра.

Наприклад, при поточному значенні параметра $R12 = -100000$ чисельне значення адреси $Z-50000-R12$ буде дорівнювати $+50000$. Знак “+” перед формальним параметром може бути опущений.

Якщо в якому-небудь кадрі УП використовується формальний параметр, то в розрахунок береться те числове значення ФП, яке він має в момент використання.

Можливі наступні способи формування значень ФП: присвоєння значень, дії над параметрами.

Присвоєння поточного значення параметра відбувається в інформаційному слові R після визначення номера параметра, наприклад, $N12R12+734000$ (параметру $R12$ присвоюється значення 734000).

Знак між номером параметра і значенням параметра обов’язковий.

Дії над ФП вводяться за допомогою оператора $@pp$, де значення двох цифр pp визначає конкретний вид арифметичної операції над цілими числами: $pp = 04$ – додавання; $pp = 05$ – вирахування; $pp = 06$ – множення; $pp = 07$ – розподіл (з відсіканням дробової частини результату).

Формат запису дії на ФП має вигляд: $R_1 @ pp R_2$, де R_1 і R_2 – лівий і правий ФП, що беруть участь в операції. Результат операції присвоюється лівому параметру.

Декілька дій над ФП допустимо поєднувати в зв’язку, яка має структуру:

$$R_1 @ pp_1 R_2 @ pp_2 R_3 \dots @ pp_{n-1} R_n.$$

Оператори у зв’язці виконуються в порядку їх запису. Результат операції присвоюється параметру, що примикає до оператора зліва. Наприклад, обчислення значень формальних параметрів: $R01 = 15000$; $R02 = 700$; $R03 = -20$; $R04 = -10$; $R05 = -2$. Після обчислення $R01 @ 04 R02 @ 05 R03 @ 06 R04 @ 07 R05$ одержимо нові значення параметрів: $R01 = R01 + R02 = 2200$; $R02 = R02 - R03 = 720$; $R03 = R03 * R04 = 200$; $R04 = R04 : R05 = 5$; $R05 = -2$.

Способи зміни послідовності виконання кадрів УП. За відсутності спеціальних вказівок кадри в УП виконуються послідовно. Послідовність виконання кадрів можна змінити багаторазовим проходом виділеної ділянки програми, переходом (безумовним чи умовним) з поточного кадру на спеціально зазначений кадр, зверненням до підпрограми.

Повторення ділянки програми (цикл). Границі повторюваної ділянки відзначаються: початок ділянки – кадр, що починається символом *H*; кінець ділянки – кадр, що закінчується допоміжною функцією *M20*.

Число повторень встановлюється в інформаційному слові *H* у вигляді цілого числа без знака від 1 до 99.

Приклад повторювання ділянки до п'яти разів:

N501 G01 G91 X200000 Y-50000 F500 R01+200000 R02-100000 R03-1

N502 H2 X+R01 @06 R03

N503 Y+02 @06 R03 M20

За допомогою подвійного повторення кадрів 502 і 503 формується прямокутний контур *Y100 X200* мм.

Переходи в технологічній програмі. Інструкція переходу до кадру записується у вигляді:

E ± nnnn @dd R_m,

де $\pm nnnn$ – адреса переходу – номер кадру, до якого переходять вниз програми при знаку “+”, у верх – при знаку “-”;

@dd – вид переходу;

R_m – параметр із номером *m*, по якому оцінюється умова переходу;

@00 – безумовний перехід;

@01 – умовний перехід, якщо $R_m = 0$;

@02 – умовний перехід, якщо $R_m > 0$;

@03 – умовний перехід, якщо $R_m \geq 0$.

Приклад свердління деталі на глибину 52 мм кроками по 5 мм із підняттям свердла після кожного кроку у вихідне положення – 7 мм над поверхнею деталі:

R50 – загальна глибина свердління 52 мм

R51 – хід свердла 5 мм

R52=0 – крок заглиблення свердла 7 мм

N601 G91 G00 Z-R52

N602 G01 Z-R51

N603 R52 @04 R51

N604 G00 Z+R52

N605 R50 @05 R51

N606 E-601 @02 R50

N607 M17.

Існує й інший варіант інструкції переходу в технологічній

програмі з умовою перевірки двох параметрів:

$@dd \pm nnnn R_m R_n$,

де використовуються такі види переходу: @01 – умовний перехід, якщо $R_m = R_n$; @02 – умовний перехід, якщо $R_m > R_n$; @03 – умовний перехід, якщо $R_m \geq R_n$ і т.д.

Технологічні підпрограми. Повторювані процеси обробки деталей можуть бути зведені у підпрограми. Підпрограми вводяться в пам'ять пристрою ЧПУ до початку роботи. Підпрограма починається символом L .

Кодування підпрограм, здійснюване в основному аналогічно кодуванню основних УП, має деякі особливості.

Підпрограма представляє собою систему команд, викликану до дії основною УП, що управляє визначеною закінченою послідовністю дій робочих органів верстата. Звертання до підпрограми може здійснюватися і вручну з пульта пристрою ЧПУ. У підпрограмах замість конкретних чисел адресної частини слів (фактичних параметрів) під адресами 0...9 записують формальні параметри. При цьому формальні параметри можуть впливати за будь-якими адресами, крім N , G і M . Разом з формальним параметром під адресою слова може бути записаний фактичний параметр. Тоді запис сприймається як їх алгебраїчна сума. Номери кадрів підпрограм задають три- чи п'ятизначними числами під тими ж адресами, що і кадри основної УП. Перші дві цифри п'ятизначного числа позначають номер підпрограми.

Виклик підпрограми задають словом з адресою L і чотиризначним десятковим числом (у форматі $L4$ – “Звертання до підпрограми”). Перші два розряди цього числа визначають номер підпрограми, а наступні два – кількість її прогонів. Звернення до підпрограми записують у будь-якому місці кадру основної УП між номером кадру і символом “Кінець кадру” або значеннями формальних параметрів. Значення параметрів можуть бути сформовані до звернення до підпрограми. У кадрі основної УП разом зі зверненням до підпрограми може міститися й інша управляюча інформація. Якщо звернення задане в кадрі разом з іншою інформацією, то спочатку обробляється вся інформація кадру, а потім виконується підпрограма.

Опис підпрограми починається кадром у форматі $L2$. В інформаційних словах кадру, крім G , D , L , N , слідом за адресою записується числова інформація, а за нею – формальний параметр зі знаком “+” чи “-”, тобто інформація із заданої адреси буде дорівнювати алгебраїчній сумі числа і числової інформації формального параметра.

Останній кадр підпрограми повинен містити слово $M17$ – кінець підпрограми. При цьому відбувається повернення до кадру основної програми, що іде за кадром зі зверненням до підпрограми.

Приклад застосування основної програми і підпрограми:

- основна програма

% 1

N050 L0201 R1+60000 R2+35000 R9+25000 F1000

- підпрограма

L02

N021 G01 G91 X-R2 G43 D01 Y-R9

N022 G02 G43 D01 X-R9 D00 YR9 I0 JR9

N023 X0 Y0 IR1 J0

N024 D00 XR9 G44 D01 YR9 IR9 J0

N025 D00 G40 XR2 Y-R9

N026 M01 M17.

У відповідність адресам підпрограми поставлені параметри, числові значення яких визначені раніше, ніж здійснено виклик підпрограми, яка повинна виконатися один раз.

Постійні цикли. Постійні цикли визначаються користувачем пристрою ЧПУ і вводяться в зону пам'яті для збереження технологічних підпрограм і постійних циклів.

Вони призначені для запису частоповторюваних ділянок програм, характерних для даного типу верстата.

Правила запису і процедури роботи з постійними циклами такі ж, як і в підпрограмах.

Звернення до постійних циклів і технологічних підпрограм: L00–L80 – адресація до технологічних підпрограм; L81–L99 – адресація до постійних циклів. До постійних циклів L81–L89 можна звертатися по командах G81–G89, що діють до появи нової команди з цієї групи, або по команді скасування G80.

У кадрах, що містять постійні цикли, обов'язково повинні бути задані координати переміщення. Якщо рух не потрібен, то необхідно задати нульове значення будь-якої координати.

Постійні цикли L83, L90, L96 використовуються в пристроях, програмне забезпечення яких містить дії над формальними параметрами.

7.3. Засоби автоматизації підготовки управляючих програм ЧПУ

7.3.1. Підвищення мовного рівня управляючих програм

Могутнім засобом збільшення функціональності пристроїв ЧПУ є підвищення мовного рівня управляючих програм у порівнянні зі звичайним стандартом ISO-7bit. Таке підвищення здійснюється на основі макровизначень (стандартних циклів, готових форм), звичайних підпрограм і підпрограм макромови користувача, спрощених описів контуру, безеквідистантного програмування, автоматизованої діалогової підготовки управляючих програм, у тому числі й у процесі відпрацьовування попередньої управляючої програми, тобто в рамках так званого “рівнобіжного програмування пристрою ЧПУ”.

Стандартний цикл встановлює строго визначену послідовність рухів, параметри яких дані поза циклом у тексті управляючої програми. Фіксація деяких параметрів може бути виконана розробником системи чи управління верстатобудівником при стикуванні системи управління з об'єктом. Іноді цикли трактують як звичайні підпрограми, але постійно збережені в пам'яті пристрою ЧПУ, на відміну від підпрограм користувачів, що вільно вводяться та вільно знищуються.

Деякі типові стандартні цикли показані на рис. 7.11. Цикл “схема свердління” (рис. 7.11, а) задають набором параметрів R , наприклад, таким: $R22$, $R23$ – координати центра схеми свердління; $R24$ – радіус кола, на якому розташовані отвори; $R25$ – початковий кут розташування першого отвору; $R26$ – кутовий крок отворів; $R27$ – число отворів; $R28$ – номер іншого стандартного циклу обробки отвору. Усі ці параметри повинні бути визначені в конкретних значеннях у тому кадрі управляючої програми, що передує безпосередньому виклику циклу по його імені.

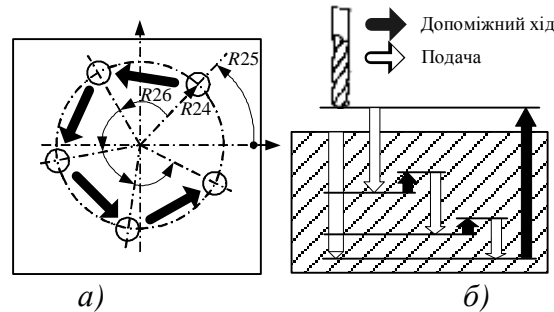


Рис. 7.11. Стандартні цикли і схеми їх обробки

Цикл “схема свердління” забезпечує послідовне позиціонування, при якому вісь шпинделя збігається з віссю чергового отвору, перехід до стандартного циклу обробки отвору після кожного чергового позиціонування, повернення до переходів схеми свердління по завершенні циклу обробки отвору. Таким чином, атрибути циклу “схема свердління” разом з атрибутами деякого циклу обробки отвору дозволяють надзвичайно лаконічно описати закінчений складний технологічний процес обробки групи отворів.

Згаданий цикл обробки отвору представляє собою могутню групу стандартних свердлильних циклів, що викликаються звичайно підготовчою G -функцією (наприклад, $G81$, ..., $G89$). Викликаний стандартний цикл діє аж до скасування спеціальною командою також на основі G -функції (наприклад, $G80$). Іноді свердлильним циклом виділяють й іншу адресу (наприклад, L) або допускають виклик циклу під адресою G чи під додатково виділеною адресою. Різниця в цьому випадку в тому, що виклик під додатковою адресою буде однократним і не буде вимагати скасування.

Приклад найпростішого стандартного свердлильного циклу наведений на рис. 7.11, б. В рамках циклу, виходячи із заданої

величини однократного заглиблення та загальної глибини свердління, необхідне число послідовних заглиблень визначається автоматично.

Ось як, наприклад, може виглядати перелік стандартних свердлильних циклів пристрою ЧПУ: високошвидкісний цикл свердління з багаторазовим відведенням інструмента; цикл нарізання правої різі мітчиком; цикл чистового розточування з виведенням інструмента при орієнтованій зупинці шпинделя; скасування оголошеного циклу, що виконується після кожного чергового переміщення (G80); цикл простого свердління (G81); цикл свердління з витримкою часу наприкінці робочого ходу (G82); цикл глибокого свердління (G83); цикл нарізання лівої різі мітчиком (G84); цикл розточування (G85); цикл розточування із зупинкою шпинделя наприкінці робочого ходу (G86); цикл розточування із зупинкою шпинделя для ручних переміщень і наступним запуском для автоматичного завершення (G87); цикл розточування із зупинкою шпинделя після витримки часу наприкінці робочого ходу ручних переміщень і наступним запуском для автоматичного завершення (G88); цикл розточування із витримкою часу наприкінці робочого ходу (G89). Додатковими вказівками для перерахованих свердлильних циклів можуть бути: G98 – завершення стандартного циклу у вихідній точці інструмента; G99 – завершення стандартного циклу в точці початку робочого ходу інструмента.

Цикли нарізання різей є неодмінною належністю пристрою ЧПУ, орієнтованого на токарну обробку. У рамках цих циклів можливість автоматичного багатопрохідного різенарізання поширюється на циліндричну зовнішню та внутрішню різь на конічній поверхні. За циклами закріплюють групу параметрів, задаючи які, визначають цикли конкретних розмірів.

Форма задання чисельного значення параметра зазвичай стандарта: номер параметра – чисельне значення параметра. Наприклад, у деякому пристрої ЧПУ за циклом різенарізання закріплена така група параметрів: R20 – крок різі; R21, R22 – координати початкової точки; R23 – число чистових робочих ходів; R24 – глибина різі (“+” – для внутрішньої різі, “-” – для зовнішньої); R25 – глибина останнього ходу; R26 – ділянка врізання; R27 – ділянка вильту; R28 – число чорнових ходів; R29 – кут нахилу профілю; R31, R32 – координати кінцевої точки.

З числа токарних циклів виділяється, як найбільш “могутній”, цикл багатопрохідний з автоматичним розподілом припуску обробки довільного контуру.

Формальний опис циклу за допомогою стандартних правил може бути складений як:

$N\{i\} \dots$ <список параметрів>...

$N\{i + 1\} L$ <виклик циклу>,

де <список параметрів> – R20 (номер підпрограми, що описує

оброблюваний профіль), R21 (вихідна точка по координаті X), R22 (вихідна точка по координаті Z), R24 (припуск на чистову обробку по осі X), R25 (припуск на чистову обробку по осі Z), R26 (глибина чорнової обробки), R27 (корекція радіуса інструмента), R29 (технологічні вказівки);

<виклик циклу> – L95.

Технологічні вказівки представлені дворозрядним кодом, перший розряд якого визначає умови отримання контуру, а другий розряд свідчить про розташування зони обробки і напрямку руху інструмента в середині цієї зони.

Готові форми, на відміну від стандартних циклів, повністю задані в конкретних розмірах і є стандартизованими компонентами контуру деталі. Вони входять до математичного забезпечення пристрою спеціалізованого ЧПУ верстата. Задання готової форми полягає лише у її прив'язці до контуру деталі, а виклик здійснюється шляхом звернення до підпрограми за допомогою G-функції або спеціально виділеної клавіші з відповідним символічним зображенням.

Підпрограми є зручним засобом скорочення обсягу основної управляючої програми. Особливо різкого скорочення вдається досягти у випадку багаторазового виклику однієї й тієї ж підпрограми, багаторазового повторення підпрограми, багаторазового вкладення підпрограм.

Приклад структурної організації програмування з використанням підпрограм показаний на рис. 7.12.

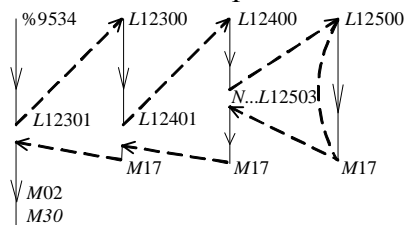


Рис. 7.12. Структурна організація програмування з використанням підпрограм

Виклик підпрограми тут здійснюють командою L (трирозрядний номер підпрограми) з дворозрядним числом, що вказує на число повторень підпрограми.

Звичайно підпрограми розробляють таким чином, що місце числових значень займають пронумеровані параметри. Наприклад, у пристрої ЧПУ передбачено сто параметрів R00–R99 і якщо вони використані в підпрограмі, то їх значення повинні бути задані в тексті основної програми.

У сучасних пристроях ЧПУ допустимі різні операції над параметрами, в тому числі параметри можуть бути використані для організації переходів. Розробка ефективних програм неможлива без широкого набору операцій над змінними підпрограми.

Розвиток подібного набору призвів до виділення і відокремлення макромови користувача.

Макромови користувача має у своєму розпорядженні наступні можливості: маніпуляції з параметрами, організацію умовних і безумовних переходів; доступ до вхідних цифрових регістрів системи управління з будь-якою наступною обробкою прийнятих сигналів; доступ до вихідних регістрів систем управління, можливість формування будь-яких повідомлень користувача на екрані дисплея; автоматичний доступ до будь-яких таблиць корекцій з можливістю внесення будь-яких виправлень; вільне використання таймерів.

У числі багатьох додатків мовні макрозасоби можуть бути використані для вільного нарощування діагностичного фонду, організації координуючих зв'язків “по горизонталі” між окремими управляючими підсистемами ГВМ, організації прийому і передачі файлів на рівні зв'язку “по вертикалі”, програмування спеціальних складних технологічних циклів та циклів електроавтоматики тощо.

Розглянемо одну з версій макромови користувача.

З формальної точки зору мова представлена макрокомандами, включеними в основну управляючу програму ЧПУ. Кожній макрокоманді відповідає підпрограма, яка зберігається в пам'яті пристрою ЧПУ як звичайна підпрограма. Однак, відмінними рисами подібної підпрограми є: використання змінних і виконання над ними широкого класу операцій; використання умовних, безумовних переходів і циклів; задання значень вихідних змінних у вихідній макрокоманді.

Є декілька способів звернення до підпрограм за допомогою макрокоманд, що включаються в текст звичайної управляючої програми в кодї ISO.

Простий однократний виклик здійснюється за допомогою послідовності адрес:

G65 P<номер підпрограми> L<число повторень> <список значень аргументів>.

Модальний виклик підпрограми припускає багаторазове звертання до неї – після чергового руху у вихідному кадрі управляючої програми. Форма модального виклику така ж, як і простого, але за допомогою функції *G66*, а скасування модального виклику виконується функцією *G67*.

Мультиплексорний виклик застосовують тоді, коли в тілі підпрограми передбачене звертання до іншої підпрограми, що виконується багаторазово всякий раз по завершенні руху в першій підпрограмі.

Нарешті, існує можливість виклику підпрограми за допомогою *G-функції*, код якої не використаний за основним призначенням. У цьому випадку заздалегідь шляхом налаштування пристрою ЧПУ повинна бути зазначена відповідність коду *G-функції* номеру

підпрограми. Форма виклику підпрограми стає найбільш простою:

G <зарезервований код> <список значень аргументів>.

Звернення до підпрограм у макромові здійснюються за допомогою адреси P .

Структура окремої підпрограми підкоряється формулі:

P <номер підпрограми> <послідовність команд> $M99$.

Загальне число підпрограм може досягати 9999.

Список значень аргументів у тілі макрокоманди виглядає наступним чином:

A <значення> B <значення> ... Z <значення>.

Як самі аргументи можуть бути використані будь-які адреси (крім G , L , M , N , O , P), причому алфавітний порядок необов'язковий. Значення можуть бути від'ємними і з десятковою крапкою, однак формат значень жорстко визначений для кожної адреси так само, як і в основній управляючій програмі.

Черговим засобом підвищення мовного рівня є *спрощення опису контуру*. Таке спрощення дозволяє зосередити в одному кадрі опис двох-трьох і навіть більшого числа суміжних елементарних ділянок, розташованих звичайно в декількох кадрах управляючої програми. Найбільш важливим тут є те, що вдається уникнути розрахунків тих розмірів, що відносяться до точок спряження, що не проставлені на кресленні (тобто задані побічно). При цьому виникає можливість вводити опис контуру безпосередньо з креслення через клавіатуру панелі управління. Як правило, подібна процедура повністю доступна оператору верстата з ЧПУ. На рис. 7.13 показані деякі способи спрощеного задання фрагментів контуру.

Точки перетину прямолінійних ділянок контуру можуть бути задані координатами (в абсолютних або відносних розмірах) або кутом. Точки спряження прямої з колом, двох кіл можуть бути визначені самою системою управління. Розрахунки, що відносяться до ділянок спряження (фаски, заокруглення), виконуються автоматично. Таким чином, пристрій ЧПУ замість одного блоку спрощеного задання фрагмента контуру повинен згенерувати кілька кадрів програми мовою *ISO* перш ніж почнеться відтворення управляючої інформації.

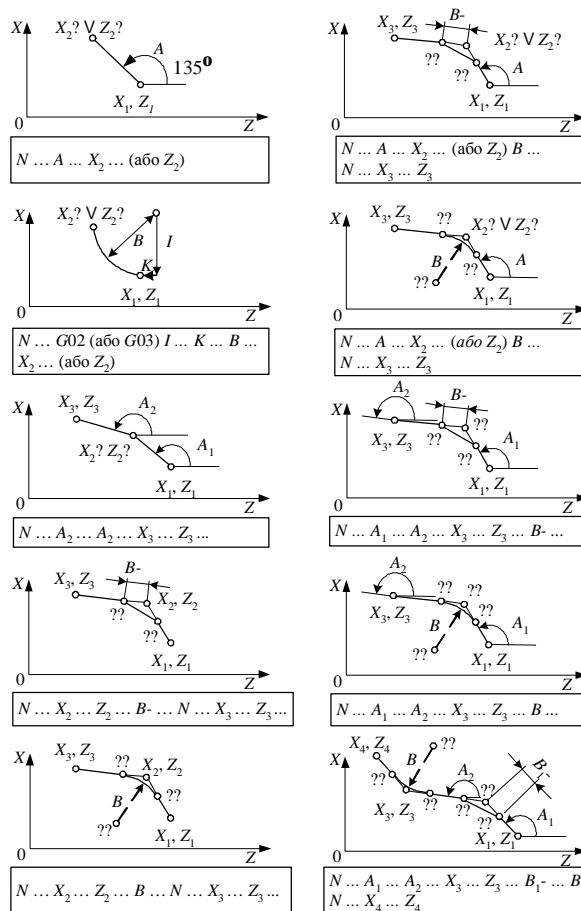


Рис. 7.13. Способи спрощеного задання фрагментів контуру

Додатковим засобом підвищення мовного рівня управляючих програм може бути так зване *безеквідистантне програмування*, що дозволяє описувати лише необхідний контур, а параметри інструмента вводити як коригувальні. Подібна проблема не така проста, як це може показатися на перший погляд.

На рис. 7.14 показано контур плоскої деталі і траєкторію центра фрези, причому контур і траєкторія нееквівалентні. При безеквідистантному програмуванні в деякі кадри вносяться серйозні корективи, а крім того, доводиться вставляти додаткові кадри на основі аналізу двох суміжних ділянок.

При безеквідистантному програмуванні необхідно передбачити наступні етапи: підхід до поверхні обробки, прохід робочої поверхні, перехід від одного типу руху до іншого по лініях продовження, перехід по зламу, перехід від робочої поверхні до вихідної точки (виведення інструмента). При цьому на початкових етапах формування траєкторії руху необхідно вказувати корекції на інструмент, активізувавши функції G41 або G42. Вектор компенсації радіуса фрези для початкової та всіх інших точок будується автоматично. Скасування компенсації здійснюється функцією G40, причому всі траєкторні

розрахунки, зв'язані з цим скасуванням, також виконуються автоматично.

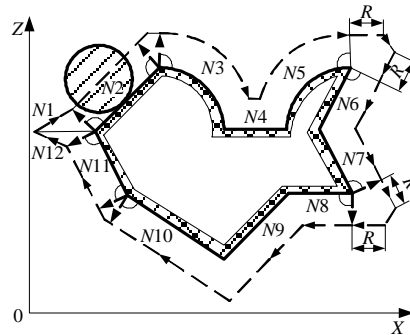


Рис. 7.14. Контур деталі та еквідистанта інструмента

Найбільш високим мовним рівнем є пристрої ЧПУ, орієнтовані на *діалогову автоматизовану підготовку управляючих програм* безпосередньо на робочому місці.

Як правило, діалог організований через графічний дисплей. Процедура підготовки управляючої програми використовує звичайно так звану техніку меню, коли оператору пропонується на вибір декілька варіантів. Зробивши визначений вибір, оператор отримує черговий набір запитів. В міру того, як інформація накопичується в системі управління, на екрані графічного дисплея виникають контури заготовки і деталі, розробляються технологічні переходи, проектується траєкторії інструмента, а нерідко визначаються і режими різання. Після того, як інформація стає вичерпною, вона транслюється в код ISO-7bit і може бути скоригована чи відредагована.

Діалогова автоматизована підготовка управляючих програм вимагає потужного математичного забезпечення, при розробці якого може бути використаний досвід створення систем автоматизованої підготовки управляючих програм на великих машинах.

7.3.2. Системи автоматизації програмування пристроїв з ЧПУ

Структура і призначення систем автоматизації програмування. Часткове чи повне рішення задач, пов'язаних з реалізацією етапів розробки управляючої програми, є метою *системи автоматизації програмування* (САП) управляючих програм.

Для встановлення структури САП введемо уявлення про фазові простори (тобто простори можливих станів) технологічного процесу і верстата. Найбільш істотною характеристикою фазового простору будемо вважати його розмірність, тобто число параметрів, необхідних для опису процесу, що протікає у фазовому просторі.

Визначимо фазовий простір технологічного процесу через такі його параметри: координати, що визначають розташування робочих органів у процесі їхнього переміщення відносно оброблюваної деталі, швидкість руху по зазначених координатах. Додатковими параметрами

фазового простору верстата є технологічні команди, що, як правило, є дискретними і призначені для виконання всякого роду допоміжних команд (керування охолодженням, зміною інструмента тощо). Різні верстати можуть мати різні фазові простори, хоча й однакові фазові простори технологічних процесів.

Звідси виникають дві можливості автоматизованої розробки управляючих програм ЧПУ: у фазовому просторі верстата й у фазовому просторі технологічного процесу. Один шлях веде до створення спеціалізованих САП, розрахованих на верстати, що належать одній структурній гамі, інший – до створення універсальних САП, що передбачають двоетапну розробку управляючих програм. Перший етап орієнтує всі розрахунки на фазовий простір технологічних процесів. На другому етапі здійснюється відображення управляючої програми з фазового простору технологічного процесу у фазовий простір верстата.

У цьому зв'язку в САП виділять на два блоки: *процесор*, що виконує розрахунок управляючих програм у фазовому просторі технологічного процесу, і *постпроцесор*, що перетворює управляючі програми у фазовий простір конкретного верстата. Кожен структурно самостійний верстат із ЧПУ повинен мати свій постпроцесор. Розділяючи процесор і постпроцесор, одержуємо можливість будувати об'єктно незалежну частину САП, що однак залишається залежною від уявлень користувача САП. Ці уявлення відбиваються у вхідній мові процесора.

Прагнення до стандартизації вхідної мови при одночасному врахуванні специфічних уявлень користувачів призводить до виділення у складі САП ще одного блока – *препроцесора*, що є транслятором з мови користувача на вхідну мову процесора. Деякий пакет препроцесорів може відбити будь-які користувацькі запити (у тому числі й у відношенні діалогу).

Типова структурна схема САП зображається у вигляді, показаному на рис. 7.15. Таким чином, структура САП, що входить до складу універсальних і більшості спеціалізованих систем автоматизації підготовки управляючих програм, побудована за принципом процесор-постпроцесор. Це означає, що в САП є основний обчислювальний блок – процесор, призначений для рішення загальних геометричних і технологічних задач підготовки управляючих програм, і ряд погоджувальних блоків – постпроцесорів, що узгоджують рішення, які видаються процесором, з можливостями й особливостями конкретних верстатів із ЧПУ.

Найбільш розвинуті САП оснащені додатковими блоками – препроцесорами, у яких за типовою технологією проектується план операції для обробки окремих елементів або класів деталей, що дозволяє об'єднати властивості спеціалізованих і універсальних САП.



Рис. 7.15. Типова структурна схема САП

До типової структурної схеми САП, крім препроцесорів, процесора і постпроцесорів, входить *блок сервісу*, що здійснює переробку постійної інформації про устаткування, інструменти і матеріали, підготовленої в анкетних формах для введення до ЕОМ. Блок сервісу систематизує і записує цю інформацію в постійну пам'ять ЕОМ у вигляді таблиць параметрів, звертання до яких здійснюється за назвами устаткування, інструмента і матеріалів, які вказуються у вихідній інформації.

У *препроцесорах* технологічні операції поділяються на встановлення та позиції, вибираються схеми кріплення заготовки, визначається послідовність переходів і комплектується інструментальне налагодження. Вихідна інформація для препроцесорів задається переважно в табличній або упорядкованій словниковій формі на спеціалізованій вхідній мові. Вихідна інформація препроцесора передається в процесор на універсальній вхідній мові, у термінах якої підготовлюється вихідна інформація для процесора при програмуванні на середньому рівні автоматизації.

Основними задачами процесорів, що розробляються в орієнтації на окремі технологічні групи устаткування з ЧПУ, є визначення проходів по обраних у препроцесорі або заданих у вихідній інформації узагальнених схемах обробки, окремих переходів, розрахунок режимів різання та обчислення траєкторії переміщення інструментів. Результати роботи процесора – послідовність і умови руху інструментів відповідно деталі – передаються в постпроцесор проміжною мовою. У процесорі формуються вхідні до складу супровідної документації карти налагодження верстата й інструмента. Сприйнята ЕОМ вихідна інформація та повідомлення про виявлені в ній помилки виводяться на контрольний бланк програміста.

До функцій постпроцесорів, розроблювальних для конкретних комбінацій верстат – пристрій ЧПУ, входять перетворення системи координат деталі, у якій розраховані процесором або задані у вихідній інформації рухи інструментів, у систему координат верстата, у якій задаються записувані в управляючій програмі рухи робочих органів

верстата, формування елементарних переміщень із врахуванням динаміки верстата, переведення швидкостей головного руху в частоти обертання шпинделя і подач у хвилинні подачі, перерахування величин переміщень у кількість дискрет, організація зміни інструментів і розподіл коректорів, кодування і запис управляючої програми на програмоносій.

Супровідна документація, що видається постпроцесором, містить роздруківку управляючої програми, дані про тривалість роботи верстата за програмою, тривалість роботи окремих інструментів та інші зведення, необхідні для нормування та організації роботи верстатів із ЧПУ.

Призначення САП визначається конструктивно-технологічними ознаками оброблюваних деталей і технологічною групою верстатів.

В залежності від характеру обробки САП призначені для:

1) плоскої обробки контурів і площин, рівнобіжних координатним площинам, на верстатах фрезерної групи з двокоординатним керуванням прямокутного чи криволінійного формоутворення і позиціонуванням по третій координаті, а також обробки контурів на електроерозійних верстатах і газорізальних машинах;

2) об'ємної обробки контурів і поверхонь на верстатах фрезерної групи з три- і більше координатним керуванням;

3) обробки тіл обертання зі східчастим і криволінійним профілем на верстатах токарної групи;

4) обробки отворів на свердлильних верстатах з позиційним управлінням;

5) комплексної обробки корпусних деталей на свердлильно-розточувальних верстатах і в обробних центрах.

Рівень автоматизації САП визначається співвідношенням обсягів ручного й автоматизованого рішень задач процесу підготовки управляючої програми.

Умовно розрізняють три рівні автоматизації: низький, середній, високий.

Для *низького рівня* характерне рішення на ЕОМ лише геометричних задач по визначенню координат опорних точок траєкторії інструмента на основі заданих у вихідній інформації узагальнених геометричних характеристик ділянок траєкторії.

При *середньому рівні* автоматизації на ЕОМ додатково зважаються технологічні задачі вибору послідовності переходів на основі заданих узагальнених технологічних схем обробки окремих ділянок заготовки.

Високий рівень автоматизації означає рішення на ЕОМ задач по оптимізації технологічного процесу, проектування інструментального налагодження та послідовності переходів по опису деталі та умов її обробки.

В залежності від рівня автоматизації САП вихідна інформація, що представлена вхідною мовою, буде більшою чи меншою за обсягом. Склад вихідної інформації встановлює технолог-програміст, виходячи з можливостей конкретної САП. Для високорівневої САП описують вихідну заготовку, кінцеву деталь і деякі умови обробки. На середньому рівні додатково задають переходи і встановлюють узагальнені схеми обробки. На низькому рівні вказують усі подробиці переходів та їх послідовність у ході обробки. Виразні можливості вхідної мови САП повинні надати технологу-програмісту всі необхідні засоби для здійснення вищезгаданих описів.

Таким чином, вхідна мова САП носить чітко помітне проблемне забарвлення.

Класифікація систем автоматизації програмування. Системи автоматизованої підготовки керуючих програм можуть бути реалізовані на різній обчислювальній основі. Класифікація САП і використовуваних обчислювальних засобів наведена на рис. 7.16.

Проте, САП класифікуються також за призначенням (проблемною орієнтацією), характером структури, формою представлення вихідної інформації та режимом її обробки, рівнем автоматизації та використовуваної ЕОМ.

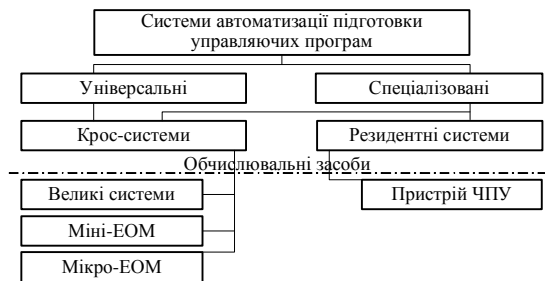


Рис. 7.16. Класифікація САП та їх обчислювальних засобів

За характером структури САП можна поділити на спеціалізовані, універсальні та комплексні.

Спеціалізовані САП розробляються для окремих класів деталей, оброблюваних на унікальних верстатах або автоматизованих ділянках.

Універсальні САП призначені для широкого кола деталей, оброблюваних на верстатах із ЧПУ визначеної технологічної групи.

Комплексні САП поєднують ряд спеціалізованих і універсальних САП верстатів із ЧПУ, що базуються на єдиній вхідній мові і загальних блоках САП при розв'язку ідентичних задач.

Універсальні САП побудовані на базі інструментальних крос-засобів, тобто створені та встановлені на зовнішніх відносно пристрою ЧПУ обчислювальних машинах.

Найбільш перспективне використання доступних ЕОМ класу міні та мікро. Подібні машини, оснащені необхідною периферією та відповідним математичним забезпеченням, називають автоматизованими робочими місцями (АРМ) технолога.

У типову конфігурацію АРМ технолога входить: персональний комп'ютер, пристрої нагромадження на магнітному диску (вінчестер), алфавітно-цифровий дисплей, графічний дисплей, клавіатура, вказуючий пристрій (миша), принтер, пристрій міжмашинного зв'язку та порти підключення.

Інший популярний у виробництві варіант полягає у використанні САП на персональному комп'ютері, що може бути зв'язаний з пристроєм ЧПУ стандартним каналом зв'язку. У цьому випадку розроблена управляюча програма передається в пам'ять пристрою ЧПУ для остаточного налагодження безпосередньо на верстаті.

Спеціалізовані САП можуть бути як крос-системами, так і резидентними системами. В останньому випадку вони вбудовані безпосередньо в пристрої ЧПУ, а автоматизована підготовка управляючих програм виконується в рамках термінальної задачі пристрою ЧПУ. Процедури розробки і налагодження управляючих програм виявляються спільними.

Форми представлення вихідної інформації в САП. Форми представлення вихідної інформації в САП дуже різноманітні. До основних форм відносяться табличний, вільний словниковий та упорядкований словниковий записи. Кожна з цих форм має свої переваги і недоліки. При їх оцінці необхідно враховувати трудомісткість підготовки вихідної інформації, характер і призначення САП, а також складність розробки транслятора – блока обчислювальних програм для перетворення записаної вхідною мовою САП вихідної інформації в канонічний вид, зручний для переробки інформації в ЕОМ.

Табличний запис даних дуже лаконічний і при використанні спеціальних бланків досить наочний. Ця форма запису прийнята в основному в спеціалізованих САП, вузька орієнтація яких дозволяє використовувати при заданні вихідної інформації характерні риси конфігурації конкретних класів деталей або окремих їх елементів.

Словниковий запис даних вільним текстом служить для задання груп параметрів обробки і вказівок ЕОМ у вигляді довільної послідовності фраз, структура яких підкоряється синтаксису прийнятої проблемно-орієнтованої вхідної мови. Ця форма запису вихідної інформації дозволяє оперувати широким набором понять і, тим самим, задовольняє вимогам універсальної САП. Недоліками запису вільним текстом є велике число правил, які необхідно засвоїти для запису різних за структурою фраз, громіздкість запису і складність транслятора для переробки тексту вихідної інформації.

Для скорочення обсягу записів застосовуються аббревіатури і скорочені позначення, що часто перетворюють текст вихідної інформації у важко-сприйнятливую послідовність літер, знаків і чисел, що ускладнює його контроль.

Скорочення обсягу запису в ряді випадків набагато ефективніше досягається в результаті використання спеціальних бланків, графі яких відображають структуру фраз даної вхідної мови. Застосування таких бланків дозволяє виключити в тексті назви найчастіше використовуваних параметрів і упорядкувати запис.

Упорядкований словниковий запис коротший і наочніший вільного тексту. Застосування його дозволить зменшити імовірність пропуску необхідних даних у процесі підготовки вихідної інформації.

Автоматизовану підготовку управляючих програм можна проводити в різних режимах, в залежності від організації роботи ЕОМ, складу зовнішніх пристроїв і можливостей конкретної САП. Організація роботи ЕОМ за допомогою операційної системи зводиться до двох основних режимів: групового і діалогового.

Груповий режим передбачає послідовне виконання на ЕОМ групи завдань в міру того, як вивільняються її ресурси, *діалоговий* – безупинний ефективний зв'язок з ЕОМ одного або декількох користувачів, кожен запит яких негайно викликає її відповідну дію.

Етапи процесу підготовки управляючої програми на ЕОМ у режимі групової обробки вихідних даних показані на рис. 7.17. На перших трьох етапах цього процесу уточнюються умови обробки, позначаються геометричні елементи на кресленні деталі (I), що програмується, записується в бланки вхідною мовою САП вихідна інформація (II) і переноситься на перфострічку або перфокарти з наступним контролем відповідності перфорації записаному в бланках тексту (III). З підготовленої для декількох деталей вихідної інформації комплектується завдання ЕОМ (IV). У процесі розрахунку управляючої програми на ЕОМ (V) працює діагностична програма САП, що при виявленні помилок видає на друк повідомлення про характер і місце їх розташування в тексті вихідної інформації. Якщо помилок немає, то розрахунок поточної управляючої програми проходить до кінця, після чого вона видається на перфострічку із супровідною документацією. Потім за інформацією з комплекту завдання ЕОМ починається розрахунок наступної управляючої програми.

За наявності підключеного до ЕОМ графопобудовника супровідна документація може супроводжуватися графіком траєкторії переміщення інструмента в системі координат деталі, більш зручним для контролю траєкторії (VI), ніж графік, що відтворює на автономному графопобудовнику закодовані в управляючій програмі рухи робочих органів верстата.

Для виправлення помилок, виявлених на етапах розрахунку управляючої програми, і контролю траєкторії інструмента технолог-програміст коректує вихідну інформацію (VII), додаючи до неї вказівки про вставлення, видалення чи заміну помилкових ділянок тексту, і організує повторний розрахунок управляючої програми на ЕОМ. Наступний контроль управляючої програми на верстаті (VIII) також

може привести до необхідності коректування вихідної інформації та повторного розрахунку програми на ЕОМ, якщо коректування програми на верстаті з ЧПУ неможливе або менш ефективне, ніж за допомогою САП. Кінцевим етапом підготовки управляючої програми є комплектація документації (ІХ).

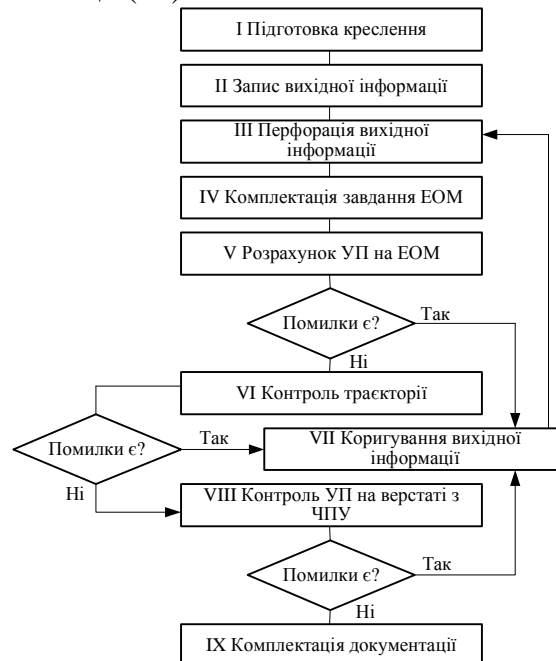


Рис. 7.17. Етапи процесу підготовки управляючої програми

Підготовка управляючої програми на ЕОМ у діалоговому режимі найбільш зручна при використанні дисплеїв – пристроїв введення-виведення текстової та графічної інформації на електронно-променевих трубках. Для того щоб запрограмувати обробку деталі, технолог-програміст, використовуючи функціональну клавіатуру графічного дисплея та його світлове перо, будує на екрані контур деталі, що програмується, оперативно виправляючи всі допущені помилки. Далі, задаючи технологічні інструкції, він одержує на екрані графік розрахованої на ЕОМ траєкторії інструмента, що дає можливість візуально проконтролювати розрахунок і за необхідності внести в раніше задані інструкції відповідні корективи.

Можливі й інші варіанти діалогового режиму підготовки управляючої програми. Наприклад, технолог-програміст по запитам ЕОМ може набирати на клавіатурі дисплея ЕОМ текст вихідної інформації мовою САП і паралельно з введенням інформації візуально контролювати висвітлені на екрані або накреслені графопобудовником контур деталі та траєкторію інструмента, виправляючи помилкові ділянки тексту.

Контрольні запитання

1. Наведіть призначення і відмінності методів підготовки управляючих програм ЧПУ.
2. Визначьте структуру і головні інформаційні блоки кадрів управляючої програми ЧПУ.
3. Наведіть приклад формального опису структури кадрів управляючої програми ЧПУ.
4. Подайте формальний опис типових операції обробки та їх призначення при застосуванні у пристроях ЧПУ.
5. Сформулюйте етапи розробки управляючої програми при програмуванні технологічного процесу обробки деталей.
6. Розкрийте можливості розширених функцій програмування обробки деталей на верстатах з ЧПУ.
7. Наведіть засоби автоматизації підготовки управляючих програм ЧПУ.
8. Визначте призначення і структуру систем автоматизації програмування пристроїв з ЧПУ.

*«Майбутнє наповнено
невизначеності, але ця
обманливість майбутнього є
найбільшим благом» (Фукидід)*

Післямова

Питання, розглянуті в даній книзі, зв'язані з проблемою забезпечення підвищення ефективності і гнучкості комп'ютеризованих систем керування інтегрованим виробництвом. Передумовами для гнучкості систем є об'єктно-орієнтованість засобів представлення, а основою проектування – моделювання й аналіз (верифікація).

Хоча КІС орієнтована на перетворення інформації, а технологічний процес організує перетворення матеріальних предметів виробництва, системні моделі того й іншого елементів виробництва на рівні причинно-наслідкових зв'язків запропоновано будувати на основі єдиних принципів, єдиного математичного забезпечення. У цьому випадку забезпечується ефективне відображення взаємодіючих інформаційно-керуючих і матеріальних процесів інтегрованого виробництва в цілому.

Тому що, задачі проектування процесів управління не можна вирішувати за допомогою математичних моделей тільки одного класу, вони поєднуються єдиним підходом для досягнення глобальної мети організації інтегрованого виробництва, стрижнем якого служать сіткові моделі і їхні розширення, модифікації. Це дозволяє відображати і досліджувати процеси управління, забезпечити єдиний підхід у використанні різних типів аналітичних, чисельних і імітаційних моделей у плануванні, моделюванні і управлінні, а також формалізувати процес аналізу і синтезу КІС у цілому.

Автори не претендують на одиничність запропонованого підходу при рішенні розглянутих задач. Він має різні шляхи для свого розвитку й удосконалювання.

Список використаної та рекомендованої літератури

1. Автоматизированная подготовка программ для станков с ЧПУ: Справочник / Под общ. ред. Р.Э. Сафраган. – К.: Техника, 1986. – 191 с.
2. Блехерман М.Х. Оперативно-производственное планирование гибких производственных систем. – М.: Высш. шк., 1989. – 95 с.
3. Васильев В.В., Кузьмук В.В. Сети Петри, параллельные алгоритмы и модели мультипроцессорных систем. – К.: Наукова думка, 1990. – 216 с.
4. Довбня Н.М., Кондратьев А.Н., Юревич Е.И. Роботизированные технологические комплексы в ГПС. – Л.: Машиностроение, 1990. – 303 с.
5. Зайченко Ю.П. Исследование операций. – К.: Вища школа. Головное изд-во, 1988. – 552с.
6. Коровин Б.Г., Прокофьев Г.И., Рассудов Л.Н. Системы программного управления промышленными установками и робототехническими комплексами. – Л.: Энергоатомиздат, 1990. – 352 с.
7. Косовский В.Л., Козырев Ю.Г. Программное управление станками и промышленными роботами. – М.: Высш. шк., 1989. – 287 с.
8. Котов В.Е. Сети Петри. – М.: Наука, 1984. – 245 с.
9. Лазарев В.Г., Пийль Е.И. Синтез управляющих автоматов. – М.: Энергоатомиздат, 1989. – 328 с.
10. Основы автоматизации управления производством / Под ред. И.М. Макарова. – М.: Высш. шк., 1983. – 504 с.
11. Питерсон Дж. Теория сетей Петри и моделирование систем. – М.: Мир, 1984. – 325 с.
12. Робототехника и ГАП: В 9-ти кн. Кн. 3. Управление робототехническими системами и гибкими автоматизированными производствами / Под ред. И.М. Макарова. – М.: Высш. шк., 1986. – 159 с.
13. Робототехника и ГАП: В 9-ти кн. Кн. 5. Моделирование робототехнических систем и гибких автоматизированных производств / Под ред. И.М. Макарова. – М.: Высш. шк., 1986. – 159 с.
14. Слепцов А.И., Юрасов А.А. Автоматизация проектирования управляющих систем гибких автоматизированных производств. – К.: Техника, 1986. – 110 с.
15. Соломенцев Ю.М., Сосонкин В.Л. Управление гибкими производственными системами. – М.: Машиностроение, 1988. – 352 с.
16. Сосонкин В.Л. Программное управление технологическим оборудованием. – М.: Высш. шк., 1991. – 512 с.
17. Управление ГПС: модели и алгоритмы / Под общ. ред.

- С.В. Емельянова. – М.: Машиностроение, 1987. – 368 с.
18. Управление дискретными процессами в ГПС / Под ред. Л.С. Ямпольского. – К.: Техника, 1992. – 251 с.
19. Ямпольский Л.С., Банашак З. Автоматизация проектирования и управления в гибком производстве. – К.: Техника, Варшава: Научно-техническое издательство, 1989. – 268 с.

Предметний покажчик

А	184
Автоматизація виробничого циклу (9)	– – логічна (172, 226)
Автоматизована система наукового дослідження (9)	– – термінальна (172, 252)
– – технологічної підготовки виробництва (9, 19)	– – технологічна (172, 235)
– – управління виробництвом (9, 16, 19)	Засоби верифікації властивостей сітки Петрі (93)
– – – організаційно-технічна (10, 16, 19)	І
– – – підприємством (10)	Ієрархія рівнів управління (16)
– – – технологічними процесами (10, 16)	Інваріант сітки Петрі (97)
Алгоритм диспетчеризації (66)	Інтерполяція формоутворення деталі (200)
– управління (112)	– лінійна (201)
Алгоритмізація (30)	– кругова (201)
Аналіз задачі на чутливість (40)	К
– редуційний сітки Петрі (101)	Координата ведена (202, 210)
В	– ведуча (202, 210)
Виробництво	– керована (149)
– комплексно-автоматизоване (9)	– – одночасно (149)
Властивість сітки Петрі (78)	Л
– – – кольорової (86)	Лінія гнучка автоматизована (11)
Г	М
Граф досяжності (78)	Маршрут технологічний (10, 48)
Д	Метод “гілок та границь” (46)
Діаграма Ганта (53)	– задачі Джонсона (52)
Дільниця гнучка автоматизована (11)	– імітаційного моделювання (53)
Дерево досяжності (78, 94)	– інтерполяції оцінної функції (206)
Документація конструкторська (9, 19)	– – – на постійній несучій частоті (213)
Е	– – – з прогнозуючим кроком (210)
Еквідистанта (191)	– – прогнозу і корекції (222)
Ескізний проект (9)	– – ЦДА (217)
З	– підготовки УП (262)
Завдання технічне (9)	– “симплекс-метод” (38)
Задача ЧПУ геометрична (172,	Модуль гнучкий виробничий (11, 140)
	– оперативно-календарного планування (22)
	– оперативної диспетчеризації

- (22, 64
- оперативного контролю (22, 56
- – корегування (22, 61
- – обліку (22
- статистичного обліку (22
- О**
- Об'єкт управління (13
- Операція виробнича (13
- управління (13
- Оптимізація виробничого циклу (49
- використання устаткування (49
- незавершеного виробництва (50
- П**
- План об'ємно-календарний (42
- по-операційний (42
- техпромфінансовий (32
- Планування всередині-цехове (31
- загальнозаводське (31
- календарне (31, 41
- міжцехове (31
- обсягу і номенклатури (34
- об'ємно-календарне (34
- оперативне (31, 47
- оперативно-календарне (34
- перспективне (31
- поточне (31
- техніко-економічне (31
- Правила вирішальні (54
- Пристрій ЧПУ (157
- Програма управляюча (157
- Програмування лінійне (38
- дискретне лінійне (47
- – цілочисельне (43
- формоутворення деталі (185, 190, 265
- Простий устаткування (51
- Процес виробничий (112
- інформаційний (71
- матеріальний (71
- технологічний (13
- управління (13
- Р**
- Резерв часу локальний (51
- С**
- Сітка Петрі (74
- – ієрархічна (82, 88
- – – кольорова (90
- – інгібіторна (81
- – кольорова (80, 82
- – предикатна (81
- – часова (80
- перероджуюча (84
- селективна (85
- Сіткова модель алгоритму управління (119
- – виробничого процесу (113
- – управляючого процесу (118
- Система автоматизації програмування пристроїв ЧПУ (301
- автоматизованого проектування (9
- адаптивного управління технологічним процесом (246
- виробнича безвідмовна самовідновлювана (8
- – самонавчальна (8
- гнучка виробнича (7,10
- – комп'ютеризована (8
- – комп'ютерно-інтегрована (5
- комп'ютерно-інтегрована (8
- контрольно-вимірювальна (235
- інтегрована виробнича (8, 9
- оперативного управління (22
- циклової електроавтоматики (227
- ЧПУ (157
- – апаратна (165
- – контурна (163
- – мікропроцесорна (165
- – позиційна (163
- – циклова (164
- Специфікація предметної області (71

Т

Технологія групова (9)

У

Управління

– оперативне (56

– оперативно-диспетчерське (65

– числове програмне (7, 156

Ф

Функції системи ЧПУ (170

Ц

Цех гнучкий автоматизований (11

Ч

Час очікування (51

– простою (51

Зміст

Список скорочень	1
Вступ	3
Розділ 1. ОРГАНІЗАЦІЙНО-ТЕХНОЛОГІЧНІ ОСНОВИ ПОБУДОВИ ГКІС В ІНТЕГРОВАНОМУ ВИРОБНИЦТВІ	5
1.1. Напрямки та шляхи розвитку автоматизації виробничих систем	5
1.2. Узагальнена структура ГКІС в інтегрованому виробництві	8
1.3. Організація управління в інтегрованому виробництві	12
1.3.1. Класифікація систем управління	12
1.3.2. Ієрархія та задачі рівнів управління ІВС	16
1.3.3. Організація багаторівневої ієрархічної системи управління в ІВС	19
1.3.4. Функціональна структура системи оперативного управління ГВС	22
1.4. Організація програмно-математичного забезпечення управління інтегрованим виробництвом	24
1.4.1. Принципи побудови програмно-математичного забезпечення	24
1.4.2. Структура і функції підсистем ПМЗ КСУІВ	25
1.4.3. Організація налагодження та впровадження програмного забезпечення в ІВС	27
Контрольні запитання	28
Розділ 2. АЛГОРИТМІЗАЦІЯ ЗАДАЧ УПРАВЛІННЯ ІНТЕГРОВАНИМ ВИРОБНИЦТВОМ	30
2.1. Автоматизоване планування в інтегрованому виробництві	30
2.2. Типові задачі планування в інтегрованому виробництві та методи їх розв'язання	34
2.2.1. Дослідження задачі планування обсягу і номенклатури виготовлення продукції	34
2.2.2. Дослідження задачі календарного планування виробничої системи	41
2.2.3. Дослідження задачі оперативного планування	47
2.3. Оперативне управління ГВС	56
2.3.1. Призначення задачі та методи організації оперативного контролю за допомогою ЕОМ	56
2.3.2. Призначення задачі та стратегії оперативного корегування у ГВС	61
2.3.3. Організація оперативно-диспетчерського управління ГВС	64
Контрольні запитання	70
Розділ 3. МОДЕЛЮВАННЯ ТА ВЕРИФІКАЦІЯ МАТЕРІАЛЬНО-ІНФОРМАЦІЙНИХ ПРОЦЕСІВ ГВС	71
3.1. Специфікація предметної області інтегрованого виробництва	71
3.2. Базовий апарат сіткових моделей дискретних виробничих	

процесів	74
3.2.1. Основні визначення та правила роботи сіток Петрі	74
3.2.2. Властивості сіток Петрі	78
3.3. Модифікації сіткових моделей дискретних виробничих процесів	80
3.3.1. Призначення класів сіткових моделей дискретних виробничих процесів	80
3.3.2. Кольорові сітки Петрі	82
3.3.3. Ієрархічні кольорові сітки Петрі	88
3.4. Засоби верифікації та аналізу властивостей сіткових моделей	93
3.4.1. Дослідження дерева досяжності	94
3.4.2. Дослідження матричного подання сітки Петрі	97
3.4.3. Редукційний аналіз структури і динаміки роботи сітки Петрі	101
3.4.4. Методи аналізу модифікацій сіток Петрі	106
3.5. Методика сіткового моделювання процесів функціонування ГВС	107
Контрольні запитання	111
Розділ 4. ПРОЕКТУВАННЯ ПРОЦЕСІВ УПРАВЛІННЯ ГВС	112
4.1. Технологія проектування алгоритмів управління виробничими процесами ГВС	112
4.2. Синтез моделей алгоритмів управління виробничими процесами ГВС	114
4.3. Організація робіт по створенню систем оперативного управління ГВС	120
4.4. Алгоритмічне забезпечення типової системи оперативного управління ГВС	127
Контрольні запитання	136
Розділ 5. ОРГАНІЗАЦІЯ ПРОГРАМНОГО УПРАВЛІННЯ ВИКОНАВЧИМ РІВНЕМ ГВС	140
5.1. Класифікація та склад виконавчого рівня управління ГВС	140
5.1.1. Склад устаткування виконавчого рівня ГВС як відображення його функцій	140
5.1.2. Параметри класифікації ГВМ	142
5.1.3. Основні технічні характеристики верстатів з ЧПУ	145
5.1.4. Типові структурно-компонувальні схеми ГВМ	151
5.2. Організація систем з числовим програмним управлінням	156
5.2.1. Базові поняття про числове програмне управління	156
5.2.2. Класифікація та функціональні можливості систем числового програмного управління	161
5.2.3. Режими роботи та функції систем числового програмного управління	167
5.3. Задачі систем числового програмного управління	172

5.3.1. Класифікація та характеристика задач програмного управління	172
5.3.2. Конфігурація системи числового програмного управління	176
5.3.3. Склад програмно-математичного забезпечення пристрою ЧПУ	178
Контрольні запитання	183
Розділ 6. АЛГОРИТМІЗАЦІЯ ЗАДАЧ ПРОГРАМНОГО УПРАВЛІННЯ В СИСТЕМАХ ВИКОНАВЧОГО РІВНЯ ГВС	184
6.1. Структура реалізації геометричної задачі ЧПУ в технологічному устаткуванні ГВС	184
6.2. Технологія програмування формоутворення деталей та руху робочого органу технологічного устаткування ГВС	185
6.2.1. Координатні системи верстатів з ЧПУ	185
6.2.2. Подання формоутворення деталей у пристрої ЧПУ	190
6.2.3. Розрахунок траєкторії руху робочого органу технологічного устаткування	195
6.3. Інтерполяція формоутворення деталей	200
6.3.1. Види інтерполяції в пристроях ЧПУ	200
6.3.2. Методи оцінної функції	206
6.3.3. Методи оцінної функції з прогнозуючим кроком	210
6.3.4. Методи оцінної функції на постійній несучій частоті	213
6.3.5. Методи цифрових диференціальних аналізаторів	217
6.3.6. Метод прогнозу і корекції	222
6.4. Управління формоутворенням деталей у слідкуючому приводі технологічного обладнання	223
6.5. Реалізація логічної задачі в системах електроавтоматики пристроїв ЧПУ	226
6.5.1. Структура логічної задачі в системах електроавтоматики пристроїв ЧПУ	226
6.5.2. Організація роботи дискретних механізмів автоматичної заміни інструмента у верстатах з ЧПУ	230
6.5.3. Модель розв'язання логічної задачі в системах електроавтоматики пристрою з ЧПУ	232
6.6. Реалізація технологічної задачі в системах управління ефективністю обробки пристроїв ЧПУ	235
6.6.1. Контрольно-вимірювальна система ГВМ	235
6.6.2. Формування заданої точності обробки деталей	242
6.6.3. Управління ефективністю обробки та виготовлення деталей	245
6.7. Реалізація термінальної задачі в інтерактивних системах взаємодії пристроїв ЧПУ	252
6.7.1. Структура реалізації термінальної задачі в пристрої ЧПУ	252
6.7.2. Режими і засоби діалогової роботи оператора з пристроєм ЧПУ	255

Контрольні запитання	260
Розділ 7. ПРОЕКТУВАННЯ ПРОГРАММНОГО УПРАВЛІННЯ В СИСТЕМАХ ВИКОНАВЧОГО РІВНЯ ГВС	262
7.1. Методи підготовки управляючих програм ЧПУ	262
7.2. Основи програмування управляючих програм верстатів з ЧПУ	265
7.2.1. Кодування інформації в управляючій програмі	265
7.2.2. Опис типових операцій обробки у пристрої ЧПУ	278
7.2.3. Програмування технологічного процесу обробки деталей	284
7.2.4. Розширення функцій програмування обробки деталей на верстатах з ЧПУ	290
7.3. Засоби автоматизації підготовки управляючих програм ЧПУ	294
7.3.1. Підвищення мовного рівня управляючих програм	294
7.3.2. Системи автоматизації програмування пристроїв з ЧПУ	301
Контрольні запитання	309
Післямова	310
Список використаної та рекомендованої літератури	311
Предметний покажчик	313