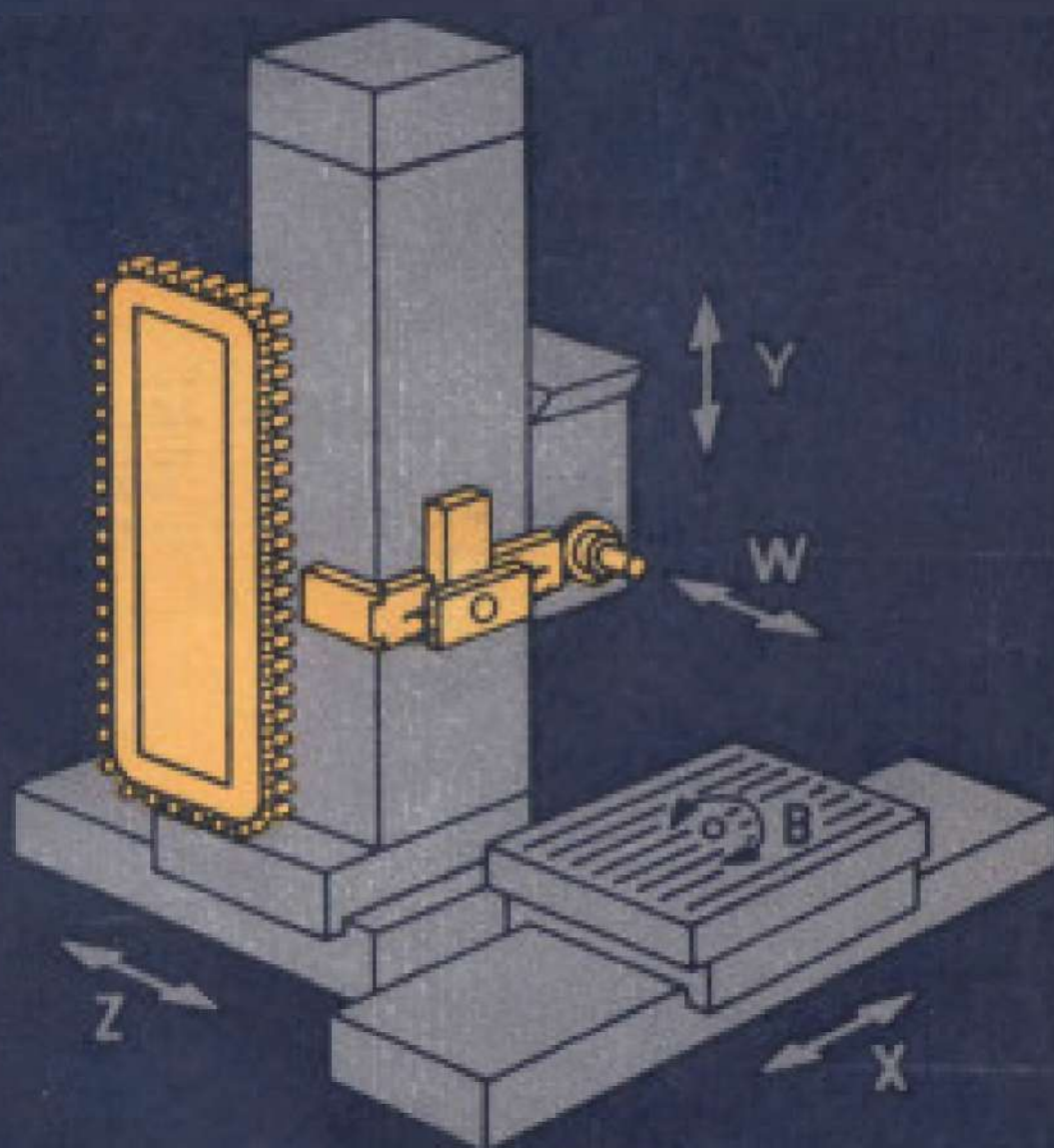
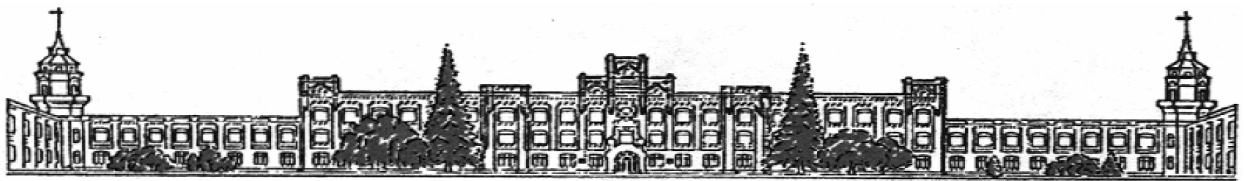


І.О. Коваленко, М.В. Пушкар

ПРОГРАМНЕ КЕРУВАННЯ ЕЛЕКТРОМЕХАНІЧНИМИ СИСТЕМАМИ

I





НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Коваленко І.О. , Пушкар М.В.

ПРОГРАМНЕ КЕРУВАННЯ ЕЛЕКТРОМЕХАНІЧНИМИ СИСТЕМАМИ -1

Курс лекцій

*для студентів напряму підготовки "Електромеханіка"
спеціальності "Електромеханічні системи автоматизації та
електропривод"
денної форми навчання*

Київ
НТУУ "КПІ"
2013

Програмне керування електромеханічними системами - 1: Курс лекцій для студентів напряму підготовки 6.050702 "Електромеханіка" спеціальності "Електромеханічні системи автоматизації та електропривод" денної форми навчання /Уклад. І.О.Коваленко., М.В.Пушкар – К.: НТУУ "КПІ", 2013. –114 с.

Гриф надано Вченою радою ФЕА НТУУ "КПІ"
(Протокол № 7 від «25» лютого 2013 р.)

На в ч а л ь н е в и д а н н я

**ПРОГРАМНЕ КЕРУВАННЯ ЕЛЕКТРОМЕХАНІЧНИМИ
СИСТЕМАМИ -1**

Курс лекцій з кредитного модуля для студентів
напряму підготовки 6.050702 "Електромеханіка"
спеціальності "Електромеханічні системи автоматизації та електропривод"
денної форми навчання

Укладачі: Коваленко Ігор Омелянович, канд. техн. наук, доц.

Пушкар Микола Васильович асистент.

Відповідальний редактор М.Я. Островерхов, канд. техн. наук, доц.

Рецензент М.Г. Анпілогов, канд. техн. наук, доц.

© Київ, НТУУ КПІ, 2013

ВСТУП

Кредитний модуль «Програмне керування електромеханічними системами-1» призначений для вивчення студентами сучасних систем числового програмного керування.

Метою лекційного курсу є отримання студентами необхідних навиків проектування апаратної частини, розробки алгоритмів та складання керуючих програм для сучасних систем числового програмного керування з подальшим застосуванням отриманих навиків при виконанні курсових та дипломних проектів.

У лекційному курсі розглядаються методи побудови та структура сучасних систем числового програмного керування.

Особливу увагу приділено застосуванню мікроконтролерів у системах програмного керування.

Для поглибленого вивчення матеріалу подано список рекомендованої літератури.

СИСТЕМИ ЧИСЛОВОГО ПРОГРАМНОГО КЕРУВАННЯ

Числове програмне керування (ЧПК) означає комп'ютеризовану систему керування, яка зчитує інструкції спеціалізованої мови програмування (наприклад, G-код) і в реальному часі керує приводами метало-, дерево- і пластмасооброблюючих верстатів і верстатним оснащенням.

Верстати, обладнані числовим програмним керуванням, називаються верстатами з ЧПК. Крім металорізальних (наприклад, фрезерних або токарських), існує устаткування для різання листових заготовок, для обробки тиском та ін. Зовнішній вид верстата з ЧПК наведено на рис. 1. На рис. 2,3,4 показані типові елементи системи ЧПК.



Рис. 1. Токарний верстат з системою ЧПК



Рис. 2. Корзина з контролерами й платами обв'язки системи ЧПК
Siemens Sinumerik (висунуто плату енкодерів)



Рис. 3. Панель керування системи ЧПК Siemens Sinumerik



Рис. 4. Шафи систем керування промислових роботів FANUC
R200i

Інтерпретатор системи ЧПК робить переклад програми із вхідної мови в команди керування головним приводом, приводами подач,

контролерами керування вузлів верстата (включити/виключити охолодження, наприклад). Для визначення необхідної траєкторії руху робочого органа в цілому (інструмента/заготовки) відповідно до керуючої програми (КП) використовується інтерполятор, який розраховує положення проміжних точок траєкторії на основі заданих початкових та кінцевих координат.

Абревіатура ЧПК відповідає двом англomовним абревіатурам - NC і CNC, які відображають еволюцію розвитку систем керування промисловим обладнанням.

Системи типу NC (англ. Numerical control), що з'явилися першими, передбачали використання жорстко заданих схем керування обробкою - наприклад, завдання програми за допомогою штекерів або перемикачів, зберігання програм на зовнішніх носіях. Яких-небудь пристроїв оперативного зберігання даних, керуючих процесорів у ранніх системах не передбачалося.

Більш сучасні системи ЧПК, названі CNC (англ. Computer numerical control), засновані на системі керування, побудованій на мікроконтролері, програмувальному логічному контролері або керуючому комп'ютері на базі мікропроцесора.

Програма для устаткування з ЧПК може бути завантажена із зовнішніх носіїв (наприклад, магнітної стрічки, перфорованої паперової стрічки (перфострічки), дискети або флеш-накопичувачів) у власну пам'ять (або тимчасово, до вимикання живлення - в оперативну пам'ять, або постійно - у ПЗП, карту пам'яті або інший накопичувач: жорсткий диск або твердотільний накопичувач). Крім цього, сучасне устаткування підключається до централізованих систем керування за допомогою заводських (цехових) мереж зв'язку.

У системі керування, крім самої програми, присутні дані інших форматів і призначення. Як мінімум, це машинні дані й дані користувача, специфічно прив'язані до конкретної системи керування або до певної серії (лінійці) однотипних моделей систем керування.

Найпоширеніша мова програмування ЧПУ для металорізального устаткування, описана документом ISO 6983 Міжнародного комітету по стандартах, і називається «G-код». В окремих випадках - наприклад, системи керування гравірувальними верстатами - мова керування може принципово відрізнятися від стандарту. Для простих завдань, наприклад, розкрою плоских заготівель, система ЧПК в якості вхідної інформації може використовувати текстовий файл у форматі обміну даними - наприклад DXF або HPGL.

Кілька верстатів з системою ЧПУ можуть бути об'єднані в гнучку автоматизовану виробничу систему (ГПС), яка у свою чергу може бути доповнена гнучкою автоматизованою дільницею (ГАД) і увійти до складу автоматичної лінії (виробництва масштабу дільниці або цеху).

В якості CNC-контролерів у наш час зазвичай використовується керуюче обладнання на основі DSP-процесорів або програмувальних логічних контролерів. Важливою характеристикою CNC-контролера є кількість осей (каналів), які він здатний синхронізувати (управляти) - для цього потрібна висока продуктивність і відповідне ПЗ. Для передачі даних між виконавчим механізмом і системою керування верстатом звичайно використовується промислова інформаційна мережа (наприклад, CAN, Profibus, Industrial Ethernet). Як виконавчі механізми застосовуються сервоприводи, крокові двигуни.

Програмне забезпечення

Розробка керуючих програм у цей час виконується з використанням спеціальних модулів для систем автоматизованого проектування (САПР) або окремих систем автоматизованого програмування (САМ), які за допомогою електронної моделі генерують програму обробки.

Історія

В СРСР перші верстати зі ЧПК для комплексної автоматизації дрібносерійного й серійного виробництва були спроектовані в ЭНИМС і були визнані гідними "Гранд-При" на виставці в Брюсселі в 1958 році.

У наступні роки для токарських верстатів найбільше поширення одержали системи ЧПК вітчизняного виробництва 2Р22 і Електроніка НЦ-31.

Ретроспектива розвитку систем ЧПК в СРСР та Україні

Основні етапи розвитку систем ЧПК наведені в таблиці.

№ п/п	Вузли та структура СЧПК
1	<u>Елементна база</u>
1.1	Напівпровідникові дискретні (елементи на платі)
1.2	Малі інтегральні схеми (серії 155 ТТЛ) (МІС)
1.3	Середні інтегральні схеми (серії 176) (СІС)
1.4	Великі інтегральні схеми (серії 155) (ВІС)
1.5	Надзвичайно великі інтегральні схеми
2	<u>Спосіб завдання програми</u>
2.1	Магнітна стрічка
2.2	Перфострічка, 5 доріжок (код БЦК – 5)
2.3	Перфострічка, 8 доріжок (код ISO – 7)
2.4	Магнітний диск
2.5	Флеш-пам'ять

3	<u>Привод подачі</u>
3.1	Електрогідравлічний з кроковим двигуном $f_n=1\text{кГц}$
3.2	Електрогідравлічний з кроковим двигуном $f_n=8,16\text{кГц}$
3.3	Високомоментний двигун, понижена індуктивність якірного кола
3.4	Безколекторний двигун: АД с векторним керуванням, синхронний двигун
4	<u>Структура СЧПК</u>
4.1	З постійною структурою і схемною (апаратною) реалізацією алгоритму
4.2	З постійною структурою і додатковими функціями (індикація, різьбо-нарізання, корекція)
4.3	З постійною структурою і пам'яттю на всю програму
4.4	З гнучкою структурою на базі мікро-ЕОМ

f_n – частота приймальності, тобто частота яку двигун може відпрацьовувати без втрати кроку.

Контрольні питання

1. Які носії програм застосовуються в системах ЧПК?
2. Що таке «G-код?»
3. Які інформаційні мережі застосовуються в ЧПК?
4. Які двигуни застосовуються у виконавчих механізмах систем ЧПК?
5. Які функції виконує інтерполятор?

КОНСТРУКТИВНІ ОСОБЛИВОСТІ ВЕРСТАТІВ З ЧПК

Верстати, оснащені ЧПК прийнято розділяти на 3 класи:

1) Верстати токарної групи, які обробляють деталі типу тіл обертання, циліндрів – це вали, заготовки для шестерень, шайб і т. д.;

2) Верстати свердлильно-фрезерної групи. Сюди входять процеси свердління на площині, фрезерування на площині, об'ємне фрезерування. Типове застосування – свердління друкованих плат, фрезерування корпусних деталей складної форми (лопатки газових турбін, гвинти суден і т.д.);

3) верстати типу «обробний центр», які об'єднують в собі функції перших двох груп.

Пристрої автоматичної зміни інструменту

У ході технологічного процесу необхідно виконувати операції різними інструментами, для цього у верстатах з ЧПК використовуються пристрої автоматичної зміни інструменту.

Залежно від класу і призначення верстата використовуються такі пристрої:

1. Магазин інструментів поворотного типу (з револьверною головкою) (рис. 3.1). Такий метод застосовується в малих і середніх верстатах, оскільки у них є обмеження за кількістю інструментів.

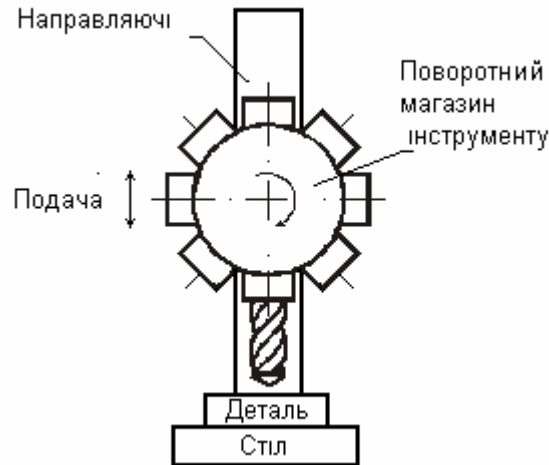


Рис. 2.1 – Інструментальний магазин поворотного типу

2. Пристрій на основі інструментального транспортера, використовується у більш складних верстатах. Це ланцюгова конструкція, що має розташовані по всій своїй довжині лотки, в яких зберігаються інструменти. Застосовується у великих верстатах,

Алгоритм зміни інструменту

1. Видається команда "№ інструменту". За цією командою починається обертання транспортера і інструмент з заданим номером подається в позицію зміни.
2. Видається команда на взяття старого і нового інструменту захоплювачами механічної руки.
3. Проводиться підйом механічної руки перпендикулярно кресленням і поворот на 180° .
4. Видається команда на опускання руки.
5. Розмикання захватів.
6. У результаті відбувся обмін інструментами.

Схема алгоритму зміни інструменту приведена на рис. 3.2.

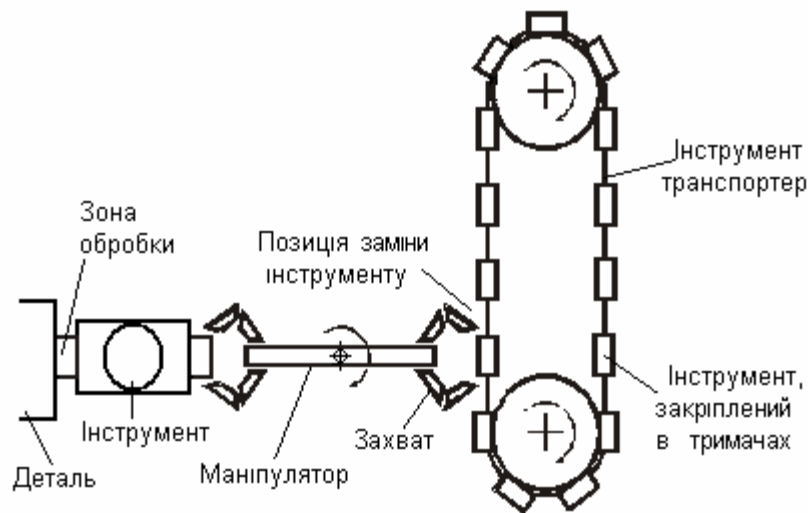


Рис. 2.2 – Ілюстрація алгоритму зміни інструменту

Розглянемо загальну конструкцію вертикально-фрезерного верстата з ЧПУ, що є найбільш універсальним і затребуваним для будь-якого типу виробництва

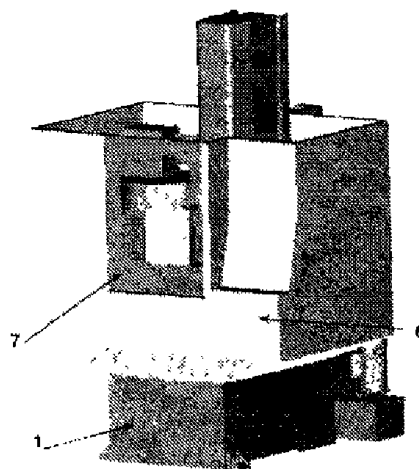


Рис. 2.3 - Корпус верстата

Станина (1) призначена для кріплення всіх вузлів і механізмів верстата.

Робочий стіл (2) може переміщатися в поздовжньому (вліво/вправо) і поперечному (вперед/назад) напрямках по *напрямяючих* (3).

На робочому столі закріплюють заготовку й різні технологічні пристосування. Для цього на столі є спеціальні Т-образні пази. *Шпиндель* (4) призначений для затискання різального інструменту й надання йому обертового

руху. Шпиндель закріплений на *колоні* (5), що може переміщатися у вертикальному напрямку (вверх/вниз). Від точності обертання шпинделя, його

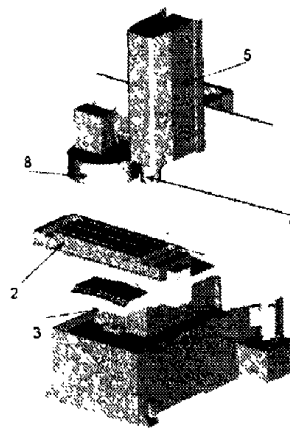


Рис. 2.4 - Конструктивні елементи верстата

жорсткості й вібростійкості значною мірою залежать точність і якість обробки. Таким чином, розглянутий верстат є 3-х осьовим.

Захисні кожухи (6) необхідні для забезпечення безпеки. Вони захищають оператора верстата від стружки, що утворюється під час обробки, і мастильно-охолоджуючої рідини (МОР), яка подається в зону обробки під тиском. *Дверцята* (7) забезпечують доступ у робочу зону верстата. У *магазині інструментів* (8) барабанного типу розміщується набір різальних інструментів. При цьому взяття необхідного інструмента й

фіксація його в шпинделі забезпечується пристроєм автоматичної зміни інструмента.

Функціональні складові (підсистеми) ЧПК

Умовно систему ЧПК можна розділити на три підсистеми:

- підсистему керування;
- підсистему приводів;
- підсистему зворотного зв'язку.

Підсистема керування

Центральною частиною всієї системи ЧПК є підсистема керування. З однієї сторони вона читає керуючу програму й віддає команди різним агрегатам верстата на виконання тих або інших операцій. З іншої сторони взаємодіє з людиною, дозволяючи оператору верстата контролювати процес обробки.

Ядром підсистеми керування є контролер (процесор), що звичайно розташований у корпусі стійки ЧПК. Сама стійка має набір кнопок і екран (усе разом називається інтерфейсом користувача) для вводу та виводу необхідної інформації.

Системи керування можуть бути закритими або відкритими (ПК – сумісними). Закриті системи керування мають власні алгоритми й цикли роботи, власну логіку. Виробники таких систем, як правило, не поширюють

інформацію про їхню архітектуру. Найчастіше користувач не може самотійно оновити програмне забезпечення й редагувати налаштування такої системи. У систем закритого типу є важлива перевага - вони, як правило, мають високу надійність, тому що всі компоненти системи пройшли тестування на сумісність.



3

Рис. 2.5 - Пристрій (стійка) ЧПК Fanuc 6М

Останнім часом стало з'являтися усе більше відкритих, так званих ПК - сумісних систем керування. Їхня апаратна начинка практично така ж, як і у добре відомого персонального комп'ютера. Перевага такого методу - у доступності й дешевизні електронних компонентів, більшість із яких можна

придбати у звичайному комп'ютерному магазині. Однак є й певні недоліки. Вважається, що такі системи поки що мають меншу надійність, порівняно з закритими системами.

Підсистема приводів

Підсистема приводів містить у собі різні **двигуни й гвинтові передачі** для остаточного виконання команд підсистеми керування - для реалізації переміщення виконавчих органів верстата.

Високоточні ходові гвинти

Важливими компонентами підсистеми приводів є високоточні ходові гвинти.

Вдосконалений ходовий гвинт верстата зі ЧПУ дозволяє виконувати переміщення виконавчого органа з мінімальним тертям і практично без люфтів. Усунення люфту дуже важливо з двох причин.

По-перше, це необхідно для забезпечення надточного позиціонування.

По-друге, тільки при дотриманні цієї умови можливо якісне фрезерування.

Двигуни

Другою складовою підсистеми є двигун (а точніше - кілька двигунів). Обертання вала двигуна призводить до повороту високоточного ходового гвинта й лінійного переміщення робочого стола або колони. У приводах верстатів з ЧПК застосовуються крокові електродвигуни та серводвигуни.

Підсистема зворотного зв'язку

Підсистема зворотного зв'язку головним чином покликана забезпечувати підсистему керування інформацією про реальну позицію виконавчого органу верстата й про швидкості двигунів.

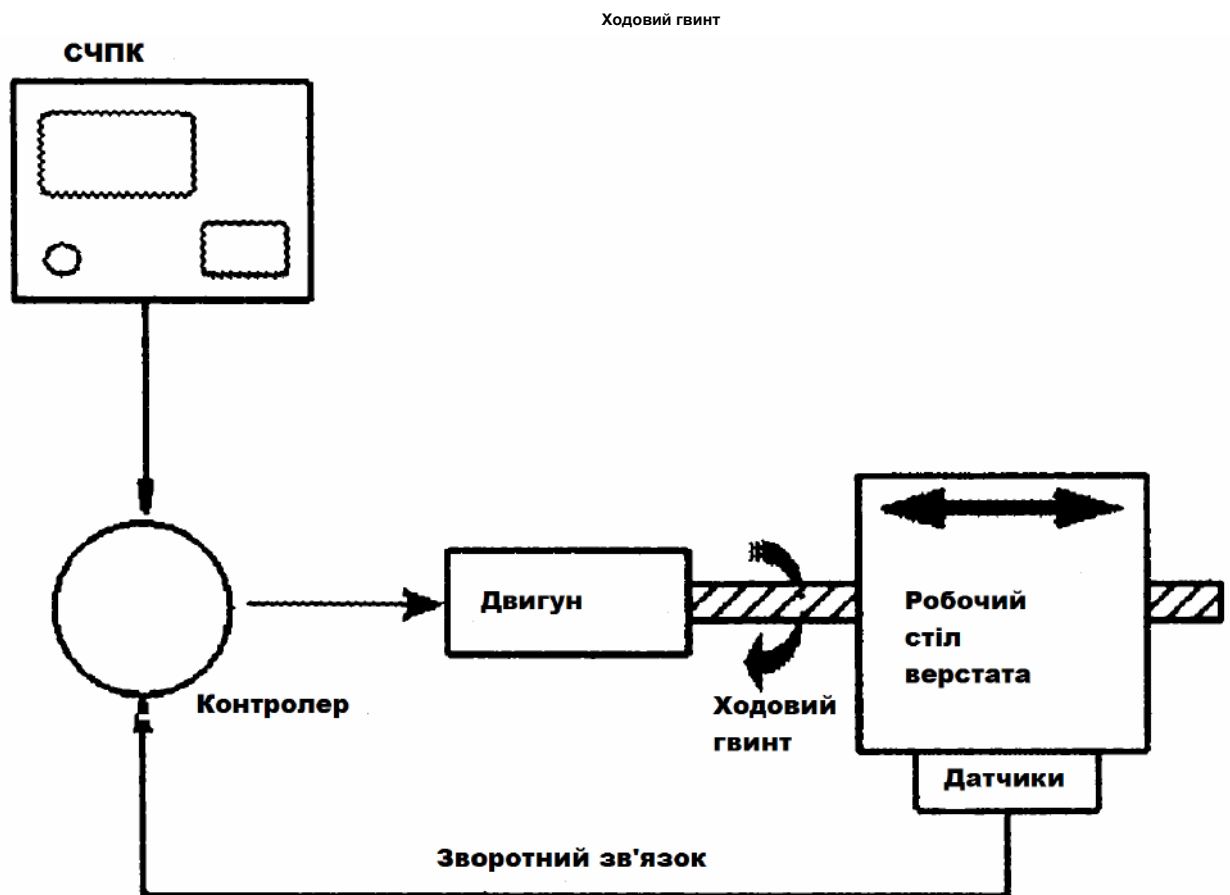


Рис. 2.6 - Схема зворотного зв'язку на верстаті з ЧПК

Датчики для визначення положення

Як правило, у верстатах з ЧПК для визначення положення й стану виконавчих органів застосовуються два типи датчиків: лінійні датчики положення й обертові датчики положення.

Обертовий датчик положення кріпиться на валу двигуна й дозволяє визначати його кутове положення. Цей датчик складається з джерела світла, оптичного датчика (приймача) і диска з малими радіальними прорізами (растрами). Растровий диск укріплений на валу, джерело світла й оптичний датчик розміщено по різні сторони диска (рис. 2.7).

Коли диск обертається, то промені проходять крізь його прорізи й падають на оптичний датчик. Оптичний датчик працює як перемикач, генеруючи серію імпульсів при попаданні на нього променів світла. Це дає можливість визначити відносне або абсолютне положення й напрямок обертання двигуна. Отримана інформація відправляється в підсистему керування.

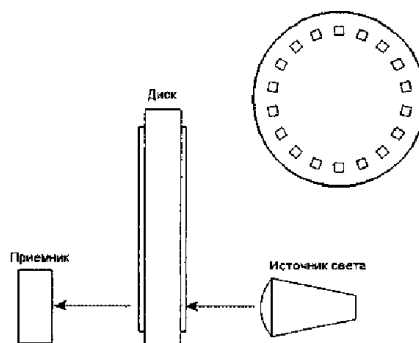


Рис. 2.7 - Обертовий датчик положення.

Всі обертові датчики мають один істотний недолік. Оскільки вони встановлюються безпосередньо на валу двигуна, то не можуть прямо виміряти лінійне положення виконавчого органа верстата. Вони дають

розраховане положення, засноване на даних про крок ходового гвинта, і у високоточних верстатах для визначення лінійного положення не застосовуються. Їх можна використати в конструкції шпинделя для визначення числа обертів при обертанні й для знаходження його кутового положення.

Лінійні датчики положення застосовуються практично у всіх сучасних верстатах з ЧПК для точного визначення абсолютної або відносної позиції виконавчих органів. Датчики містять два взаємозалежних вузли, а саме - растрову шкалу і головку для зчитування даних (рис. 2-7). *Растрова шкала* (1), розташована уздовж напрямляючих, являє собою лінійку з маленькими прямокутними прорізами (растрами). Зчитувальна головка, що переміщується разом з виконавчим органом верстата, складається з освітлювачів (2) , фотоприймачів (3) і *індикаторної пластини* (4). Причому освітлювачі й індикаторна пластина перебувають з однієї сторони від растрової шкали, а фотоприймачі з іншої.

На індикаторній пластині нанесено дві растрові ділянки зі зміщеним кроком для формування двох сигналів. Коли зчитувальна головка переміщується уздовж растрової шкали, те світлові сигнали від освітлювачів проходять через індикаторну пластину, потім через шкалу й реєструються фотоприймачами. Отримані сигнали дають можливість визначити величину й напрямок переміщення. На растровій шкалі може перебувати додаткова доріжка референтних міток для завдання власного початку відліку.

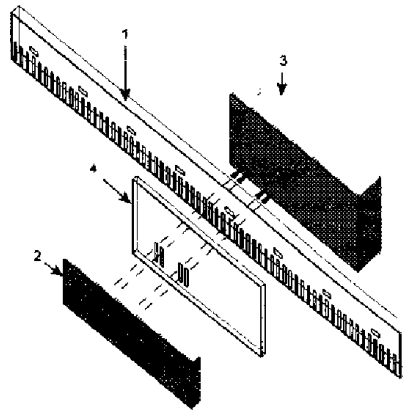


Рис. 2.8 - Лінійний датчик положення

Системі ЧПК також необхідна інформація про швидкість, прискорення й з виконавчого органа верстата. Розрахунок величини прискорення й з необхідна для точного позиціонування. Справа в тому, що коли робочий стіл переміщується в необхідну позицію, він заздалегідь сповільнює швидкість переміщення, щоб "не промахнутися" повз задану координату зупинки.

Датчики стану виконавчих органів

Крім перерахованих вище датчиків, звичайно ж, застосовуються й інші. **Температурні датчики** (термопари) застосовують для визначення температури виконавчих органів, розрахунку температурного лінійного розширення компонентів верстата й для контролю над температурою масла й повітря. **Інфрачервоні датчики** використовуються у верстатних системах автоматичного виміру.

Контрольні питання

1. На які класи розділяються верстати з ЧПК?
2. Які датчики застосовуються в системах ЧПК?
3. Поясніть конструкцію обертового датчика положення.
4. Поясніть поняття закритих та відкритих систем керування.
5. Які підсистеми входять до складу системи ЧПК?

КЛАСИФІКАЦІЯ СИСТЕМ ПРОГРАМНОГО КЕРУВАННЯ

Система програмного керування – це комплекс апаратних і програмних засобів, що забезпечують реалізацію заданої послідовності операцій відповідно до програми.

Системи програмного керування відносно молоді і почали розвиватися на початку 50-го року минулого століття. Тим не менш вони проникли практично в усі сфери промисловості, починаючи від побутових приладів і до космічних систем.

Функціонально можна класифікувати всього кілька класів цих систем, як показано на рис. 3.1.

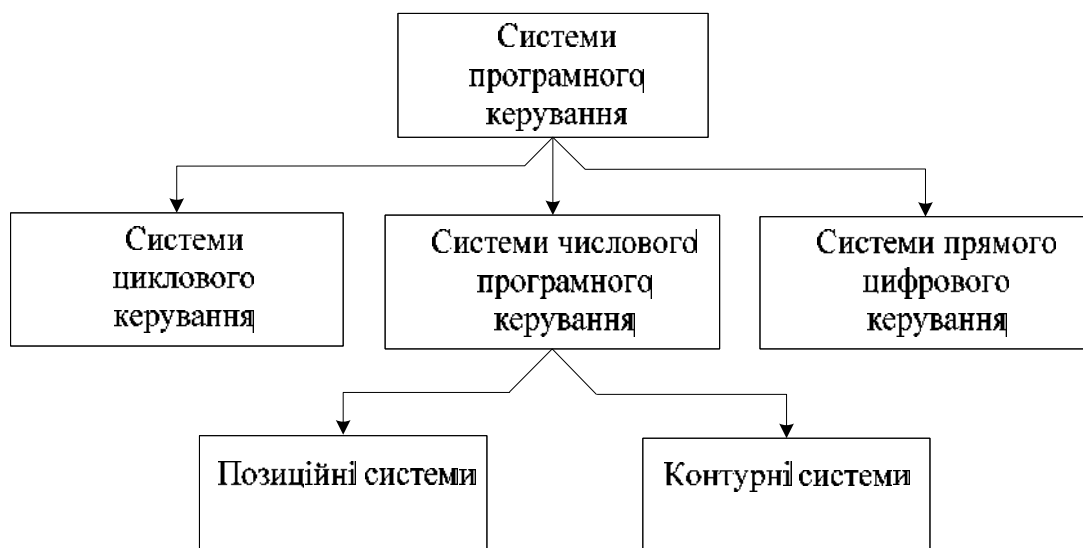


Рис. 3.1 – Класифікація систем програмного керування

Циклові СПК – це відносно прості системи, призначені для управління процесами з явно вираженими технологічними циклами. (прості роботи типу: взяти, перемістити і т.п.)

Системи ЧПК застосовуються в промисловості для керування верстатами та механізмами, які поділяються на два види: позиційні і контурні.

1. Позиційні використовуються для керування процесами в яких подача робочого органу проводиться без обробки деталі, тобто різець (інструмент) не стикається з деталлю. До цієї групи можна віднести верстати свердлильно-розточувальної групи (рис. 3.2). Завдання приводу подачі полягає в швидкому переміщенні інструменту від однієї точки до іншої за мінімальний час, і позиціонуванні в даній точці з максимальною точністю.

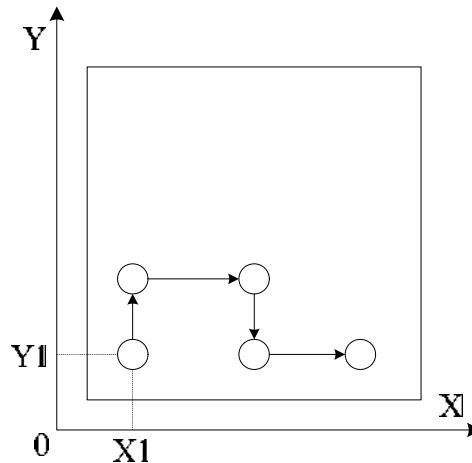


Рис. 3.2 – Схема подач в позиційних системах

Особливості приводів подач в позиційних системах:
- Привід трьохконтурний, регулювання положення.

- Оскільки переміщення між точками відбувається без контакту з деталлю, то можна реалізувати алгоритм максимальної швидкодії (рис. 3.3).

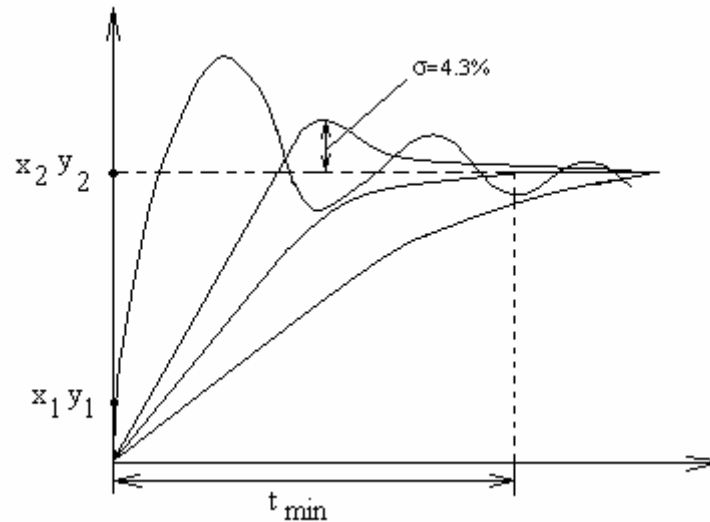


Рис. 3.3 – Алгоритм максимальної швидкодії

Залежно від налаштувань регулятора процес може бути аперіодичний, коливальний або процес оптимальний за швидкодією.

2. Контурні системи. Процес подачі йде під час обробки деталі, як показано на рис. 3.4 (токарна, фрезерна обробка).

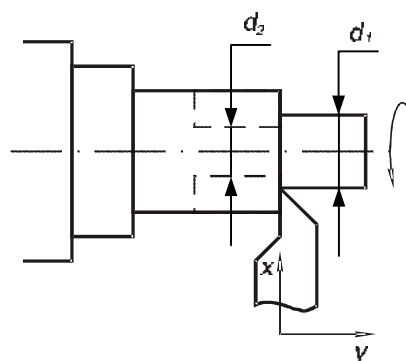


Рис. 3.4 – Схема подачі в контурній системі

Особливості приводів подач контурних систем:

- Подача виконується по трьох контурній схемі регулювання положення;
- Регулювання положення не може бути налаштоване на оптимум за швидкодією через особливості технологічного процесу.

Розглянемо практичний випадок, коли необхідно оптимальним за швидкодією способом зменшити діаметр деталі від d_1 до d_2 (рис. 3.5).

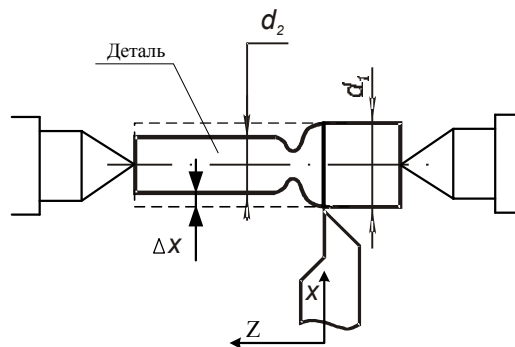


Рис. 3.5 – Вплив перерегулювання на процес обробки

При цьому перерегулювання неприпустимо, тому що буде великий вибір металу (брак). Це називається «заріз металу». У цьому місці будуть значні місцеві напруги в металі і надійність роботи такої деталі є під питанням. Такі системи називаються контурними. У них приводи подачі настраюються на оптимальний аперіодичний режим, тобто на максимальну швидкодію без перерегулювання. У цьому випадку це буде оптимальний за швидкодією аперіодичний процес.

Системи ПЦК. Системи прямого цифрового керування. Це великий клас СПК, в яких управління здійснюється безпосередньою передачею керуючих сигналів від комп'ютера на виконавчий пристрій управління

(регульовані електроприводи, клапани, сигнальні лампи і т.д.). Приклад: система наведення супутникової антени.

Міжнародні позначення систем програмного управління

NC – Numerical Control — система ЧПК;

HNC – Hand NC — система ЧПК з вводом програми з пульта станка;

SNC – Stored NC — система ЧПК з пам'яттю на всю програму;

CNC – Computer NC — система ЧПК на базі комп'ютера;

PCNC – Personal Computer NC - система ЧПК на базі персонального комп'ютера;

DNC – Direct NC — система прямого цифрового керування.

Контрольні питання

1. Поясніть призначення циклових систем керування.
2. На які види розділяються системи числового програмного керування?
3. Поясніть принципи настройки регулятора положення приводу подачі в позиційній системі.
4. Що таке оптимальний за швидкодією аперіодичний процес?
5. Наведіть пояснення міжнародних позначень систем ЧПК.

ЗАГАЛЬНА СТРУКТУРА КОНТУРНОЇ СИСТЕМИ ЧПК

Система ЧПК – це програмно-апаратний комплекс, призначений для управління технологічними процесами за програмою складеної в спеціалізованому алфавітно-цифровому коді.

У загальному випадку функціональну схему контурного пристрою ЧПК можна представити наступним чином (рис. 4.1):

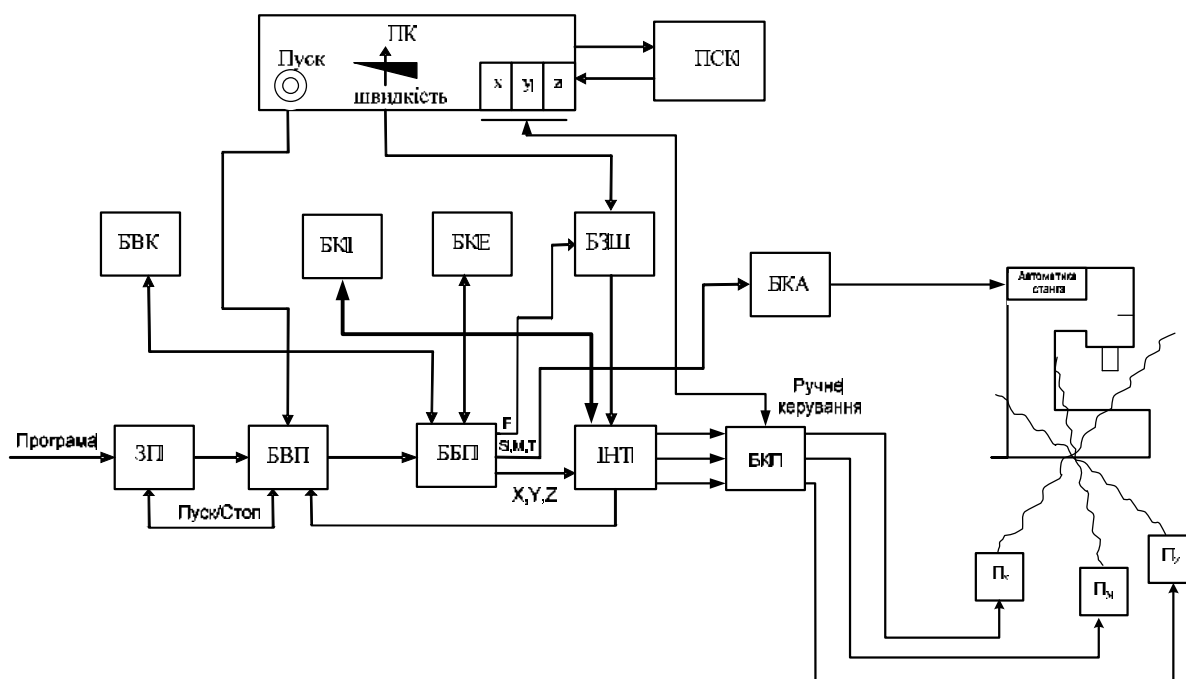


Рис. 4.1 – Загальна структура ЧПК

ПК – пульт керування;

ПСК – панель станкового керування;

ЗП – зчитувач програми;

БВП – блок вводу програми;

ББП – блок буферної пам'яті;

ВІНТ – інтерполятор;

БКП – блок керування приводами подач;

БВК – блок вводу корекції;

БЦІ – блок цифрової індикації;

БКЕ – блок корекції еквідистанти;

БЗШ – блок завдання швидкості подач;

БКА – блок керування автоматикою верстата;

Зчитувач програми ЗП призначений для введення програми з носія в блок буферної пам'яті БВП. Цей процес йде під управлінням блоку введення програми БВП. В якості носія може бути використаний будь-який фізичний носій, що дозволяє записувати масиви байтів. Кодування інформації здійснюється в коді ISO - 7 у якому кожен символ програми (літера, цифра, спеціальний символ) кодується одним байтом. Таким чином, будь-яка програма на носії представлена послідовністю байтів.

Принцип роботи контурної системи ЧПК

Головним принципом роботи ЧПУ є положення про те, що будь-який скільки-завгодно складний контур може бути з прийнятною точністю представлений у вигляді комбінації відрізків прямих і дуг кіл. Однорідна ділянка траєкторії, представлена відрізком прямої або дуги називається кадром. У цьому кадрі знаходяться команди, що визначають технологію процесу на даній ділянці. Програма складається з певної кількості кадрів.

Роботу системи розглянемо за наведеною функціональною схемою. Загальне управління системою здійснює блок введення програм БВП. При натисканні кнопки "Пуск" БВП видає сигнал на ЗП, який здійснює

зчитування одного кадру програми. Цей кадр надходить у вигляді потоку кадрів в БВП, де виконуються такі операції:

1) двоступінчастий контроль введеної інформації;

1-а ступінь – контроль парності кожного байту;

2-а ступінь – контроль структури адреси.

Інформація кожної команди перевіряється на правильність її структури відповідно до закладених в системі логічними правилами наступного порядку: 1-м символом команди повинна бути буква, потім три цифри, пробіл, буква.

2) перетворення двійково-десятькового коду переміщень в двійковий код, придатний для завантаження в робочі регістри інтерполятора;

3) зворотне перетворення вихідного двійкового коду інтерполятор в двійково-десятьковий код для видачі на систему індикації.

Відзначимо, що в двійковий код перетворюється тільки геометрична інформація, решта числова інформація використовується системою ЧПК у своєму початковому, тобто двійково-десятьковому коді.

Введений кадр розміщується в БВП. З БВП інформація розподіляється на різні пристрої. Геометрична інформація в двійковому вигляді завантажується в робочі регістри інтерполятор (адреси X, Y, Z). Інформація під адресою F з БВП завантажується в регістр БЗШ, який на своєму виході формує робочу частоту $f_{роб}$. Ця частота визначає швидкість інтерполяції, а отже, частоту видачі інформації на приводи подачі і таким чином швидкість подачі інструменту.

Інформація під адресами S, M, T називається технологічною.

Інформація за цими адресами зберігається у блоці буферної пам'яті, звідки передається в блок керування автоматикою верстату БКА.

Блок буферної пам'яті

Призначений для проміжного зберігання одного кадру програми, введеного з носія перед початком його відпрацювання (рис. 4.2).

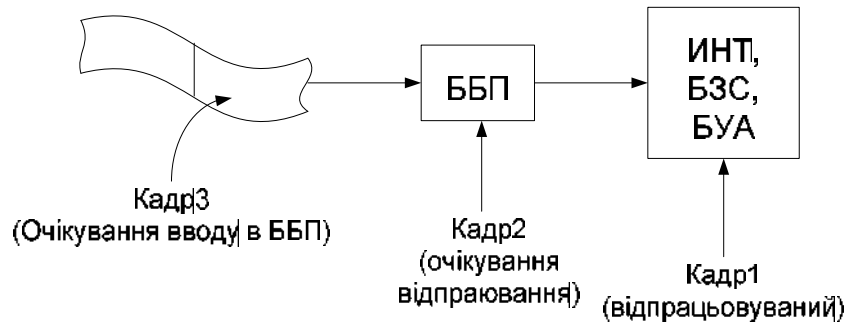


Рис. 4.2 – Схема введення кадрів програми

ББП – блок буферної пам'яті. Простий ББП повинен складатися з 7-ми комірок. Мінімально необхідний набір адрес для роботи системи ЧПК: X, Y, Z – команди переміщення по координатах (геометрична інформація). F – швидкість подачі; S – швидкість головного приводу (шпинделя); T – номер інструменту; M – допоміжні команди (керування автоматикою верстата). Функціональна схема блоку буферної пам'яті показана на рис. 4.3.

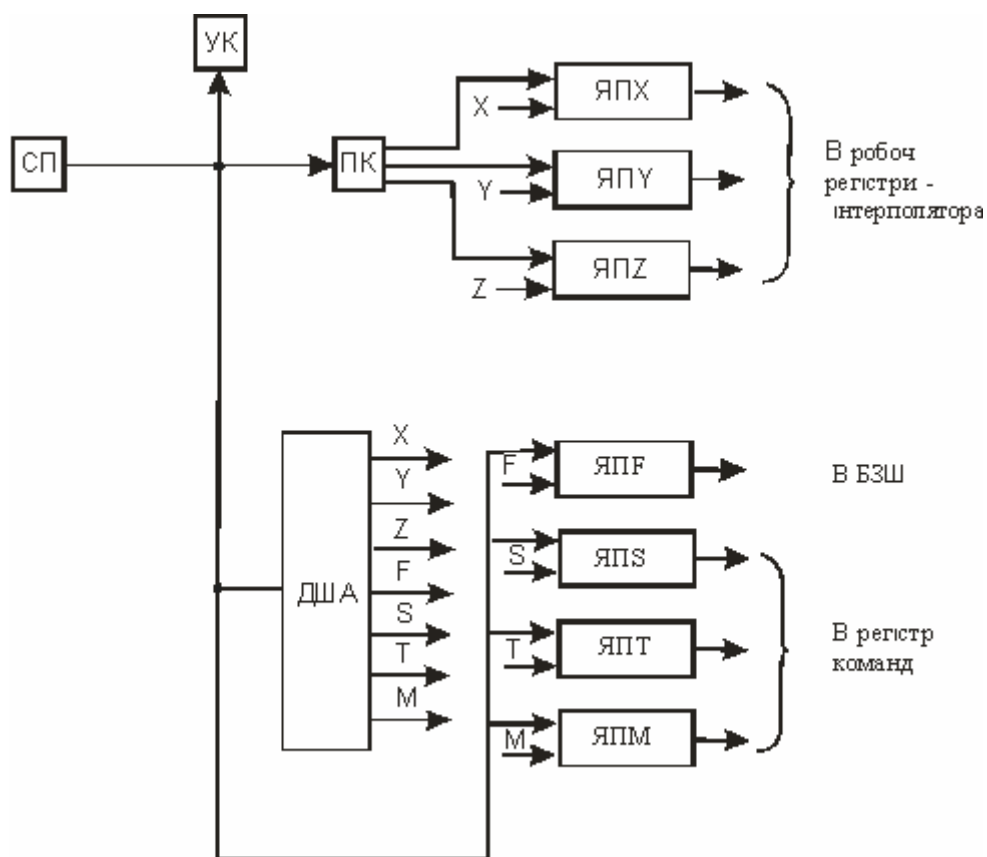


Рис. 4.3 – Функціональна схема ББП

Ця схема містить комірки пам'яті, дешифратор адреси ДША, блок контролю інформації, що вводиться, перетворювач кодів ПК. Інформація, що зчитується з носія програми, поступає на пристрій контролю парності та структури адреси. Геометрична інформація, записана в перших чотирьох розрядах у двійково-десятковому коді, поступає на перетворювач кодів, де перетворюється у двійковий код і зберігається і комірках пам'яті відповідно своїм адресам. З цих комірок пам'яті інформація про переміщення передається в робочі реєстри інтерполятора.

Записані у командах адреси поступають на дешифратор адрес, побудований за принципом «1 із N». Це означає, що при надходженні на вхід дешифратора коду адреси лише один вихід дешифратора стає активним і відкриває для запису комірку пам'яті з адресою, код якої подано на вхід дешифратора.

Окрім адрес дешифратор також розпізнає ознаку кінця кадру КК і посилає в зчитувач програми сигнал зупинки зчитування.

Контрольні питання

1. Як здійснюється кодування інформації в коді ISO-7?
2. Поясніть принцип роботи блоку буферної пам'яті.
3. Як працює дешифратор адреси?
4. Наведіть мінімально необхідний набір адрес для роботи системи ЧПК.
5. Поясніть принцип роботи блоку введення програм.

СТРУКТУРА ПРОГРАМИ І ПРИНЦИП РОБОТИ СИСТЕМИ

Структура програми в кодї ISO наступна: програма складається з великої кількості логічно самостійних ділянок, які називаються кадрами. Кожен кадр має символ початку і символ закінчення. Поле кадру заповнюється командами, що мають явно виражену технологічну спрямованість. Серед них є команди переміщення, завдання швидкості подачі, швидкості головного приводу та ін.

Розглянемо приклад.

Обробка букви W, вписаної в прямокутник 34x27 мм, рис. 5.1, на умовному вертикально-фрезерному верстаті зі ЧПК, фрезою діаметром 4 мм, у заготовці з органічного скла:

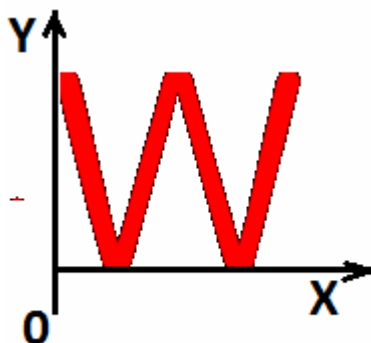


Рис. 5.1 – Фрезерування букви W

Червоним кольором виділено результат обробки.

Кадр	Зміст	Коментар
%		Початок програми
N1	G90 G40 G17	Система координат абсолютна, компенсація на радіус інструмента виключена, площина інтерполяції X-0-Y

N2	S500 M3	Задати швидкість обертання шпинделя й включити шпиндель
N3	G0 X2.54 Y26.15	Перехід у крапку початку обробки на холостому ходу
N4	Z1.0	Підхід до заготовки по Z, не доходячи 1 мм, на холостому ходу
N4	G1 Z-1.0 F100	Врізання в заготівлю на подачі 100 мм/хв
N5	X5.19 Y 2.0	Перший штрих букви W
N6	X7.76	Продовження руху
N7	X16.93 Y26.15	Другий штрих букви W
N8	X18.06	Продовження руху
N9	X25.4 Y2.0	Третій штрих букви W
N10	X25.96	Продовження руху
N11	X32.17 Y 26.15	Четвертий штрих букви W
N12	G0 Z12	Відведення інструменту від заготовки на холостому ходу
N13	M5	Стоп шпиндель
N14	M30	Кінець програми

Кодування технологічних команд і логічної інформації

Для запису програми на носій у різних країнах застосовують різні коди.

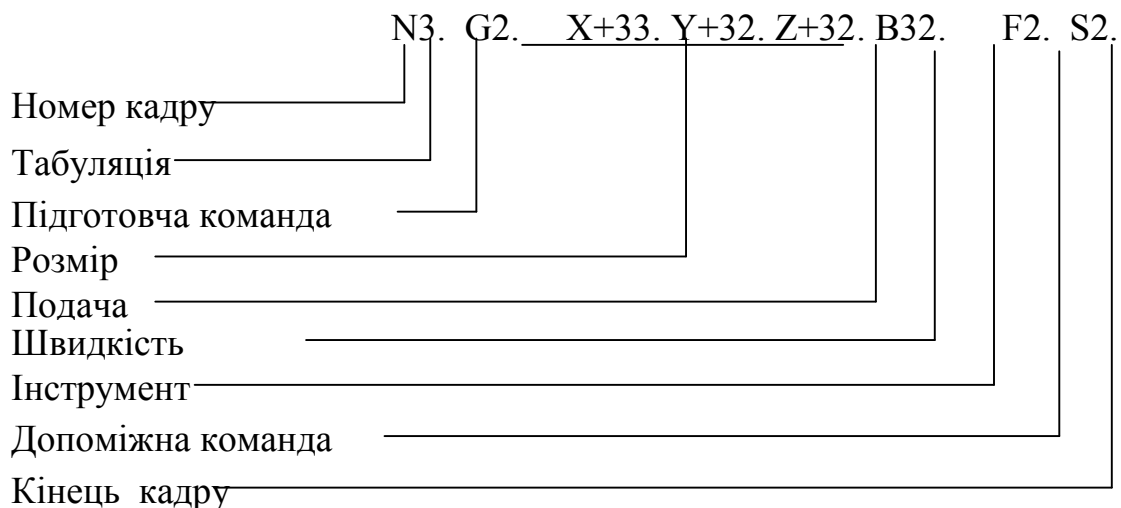
Код EIA - восьмирозрядний, що дозволяє закодувати 63 символи. Цей код використає тільки 6 доріжок з 8. Цифри кодуються двійково-десятковим кодом з вагами 8-4-2-1 для доріжок 4-3-2-1. Аналогічно

кодуються букви (застосовуються тільки малі літери латинського алфавіту) від а до і, від j до r і від s до z. Ознака групи букв записується додатково на 6 і 7 доріжках. Восьма доріжка використовується для позначення тільки лише кінця кадру, а 5-я - контрольна, що забезпечує завжди непарну кількість перфорацій у рядку.

Код ASCII має можливість закодувати 127 символів, стільки ж, скільки й у коді ISO, і використовується в основному в обчислювальній техніці. Основною відмінністю цього коду від коду ISO є те, що він у значній мірі зберігає апаратну частину системи, що працює в коді EIA, при необхідності переходу з коду EIA на код ASCII.

Код ISO, застосовуваний у СНД, є семизначним кодом для 127 символів на восьмидоріжковій перфострічці шириною 25,4 мм. Сім доріжок використовуються для кодування інформації у двійково-десятковій системі, а восьма - для контролю зчитування інформації й доповнює кількість перфорацій у рядку до парного числа. Код ISO забезпечує 16 кодових позначень від 0 до 15.

Кожний символ коду (адреса, цифри, знак або ознака) розташовуються у вигляді комбінації пробивок (перфорацій) на одному рядку стрічки.



У перших чотирьох доріжках від 0 до 9 ознака цифри записується додатковими пробивками на 5-й і 6-й доріжках. Букви від А до О закодовані аналогічно цифрам від 1 до 15, а ознакою букви є пробивка на 7-й доріжці. Друга група букв від Р до Z закодована аналогічно числам від 16 до 26, а ознакою букви є також додаткове пробивання на 7-й доріжці.

Крім адрес код ISO має ряд службових символів, наприклад: (+), (-), (:), (LE), (%) та ін.

Для позначення чіткого визначення вимог обумовлених системою ЧПК, і її можливостей розроблені рекомендації, що передбачають повний формат кадру. Кадр складається зі слів "номер кадру", слів основної інформації й символу "кінець кадру", що позначає кінець кожного кадру й передує новому кадру.

Вся інформації в кадрі повинна записуватися в строго певному порядку й не повторюється усередині кадру. Але в кожному кадрі може бути не вся вищевказана інформація, а тільки частина її. Кадр повинен будуватися в наступній послідовності:

1. Номер кадру із трьох цифр. Усього можливо 999 номерів в одній програмі.

2. Підготовчі команди групи G (режим роботи) із двох цифр. Проектом рекомендацій ISO ці команди закріплені певними кодами. Наприклад :

G00...G09 - команди загального порядку: позиціонування, лінійна або кругова інтерполяція, прискорення, уповільнення, витримка.

G10...G39 - особливості безперервної обробки: вибір осей, площин, видів інтерполяції.

G49...G59 - корекція розмірів інструмента, без відліку.

G60...G79 - вид роботи: точно, швидко.

G80...G99 - стандартні автоматичні цикли.

3. Розміри по координатах X,Y і Z. Розмір задається шістьма позиціями, що визначають кількість десяткових розрядів при завданні координати. Крім того, після адреси записується знак переміщення, тобто (+) або (-).

4. Слова "подача" F, і "швидкість шпинделя" S визначають завдання режимів різання кодовим числом. ISO передбачає два способи кодування: арифметична прогресія (3, 4 або 5 цифр) або геометрична прогресія (дві цифри). Другий спосіб передбачає більший ступінь уніфікації, більш перспективний і застосовується у вітчизняних системах ЧПК. При цьому способі подача й швидкість виражаються кодом із двох цифр, що представляють $\log f(s)$, а всі подачі або швидкості вибираються з ряду, представленого цілими ступенями числа "1,12". При цьому 00 відповідає "зупинці", 99 - "швидкому ходу", цифри ж від 1,12 до 75000 ($1,12^{99}$) охоплюють весь діапазон використовуваних на практиці величин подач і швидкостей. Лінійні подачі, що не залежать від швидкості шпинделі, задаються в мм/хв, що залежать - у мм/об; швидкості поворотного стола або шпинделя - у об/хв.

5. Слово "інструмент" T виражає в кодованому виді номер інструмента й, при необхідності, номер пов'язаної з ним корекції. Кількість цифр у слові "інструмент" установлюється в кожному конкретному випадку й вказується в позначенні формату. Звичайно для позначення номера інструмента застосовують 2 цифри.

6. Допоміжні команди групи M визначають маніпуляції, виконувані верстатом. Їхнє виконання підтверджується верстатом, що є необхідною умовою для продовження автоматичної роботи.

Розглянемо найбільше часто застосовувані команди цієї групи:

M00 - стоп по програмі. По цій команді після відпрацювання кадру відбувається зупинка автоматичної роботи. Використається при необхідності виконання окремих операцій оператором: контроль, переустановка деталей і т.д. Команда на продовження роботи дається оператором натисканням кнопки "Пуск".

M02 - кінець програми. По цій команді після відпрацювання кадру відбувається відключення шпинделя, охолодження, подачі й система ЧПК приводиться у вихідний стан.

M03 - шпиндель за годинниковою стрілкою.

M04 - шпиндель проти годинникової стрілки.

M05 - стоп шпинделя.

M06 - зміна інструменту.

M38 - M40 - вибір діапазону швидкостей шпинделя.

Слід зазначити, що в одному кадрі перфострічки може бути кілька функцій G і M, наприклад, G02, G17, G45, G61, G83, але тільки з різних груп. Наприклад, не може бути в одному кадрі функцій G01 і G02, тобто лінійна й кругова інтерполяція одночасно. Те ж саме відноситься й до команд M. Звичайно є можливість програмувати в одному кадрі до 4-х і більше команд G і M. Помилкове програмування цих команд із однієї групи повинне виявлятися системою ЧПК з виводом відповідного сигналу.

Режим роботи із циклів використовується при послідовному повторенні певних операцій, наприклад, при свердлінні в деталі однакових отворів. У звичайному програмуванні необхідно кілька разів повторити ті самі кадри, такі як: швидке підведення, робоча подача, відвід із включенням, реверсування й вимикання шпинделя. При роботі з постійними циклами перед обробкою першого отвору дається кадр із циклом G90 (початок роботи із циклів), після чого в наступний кадр

включається, крім інформації про величину переміщення, подачі й т.д., також номер циклу. Після закінчення першого циклу може бути записана програма для роботи із другого циклу й т.д.

Контрольні питання

1. Поясніть структуру програми у коді ISO-7.
2. Які функції виконують команди групи M?
3. Які способи кодування подач і швидкостей шпинделя визначені у коді ISO-7?
4. Поясніть структуру кадру у коді ISO-7.
5. Поясніть особливості кодів EIA та ASCII/

КОНТРОЛЬ ВХІДНОЇ ІНФОРМАЦІЇ

Перший етап контролю - контроль парності байтів.

Власне код ISO-7 використовує 7 бітів, 8-й біт призначений для контролю парності.

Ідея полягає в тому, що байт без помилок повинен завжди мати парну кількість одиниць. Такий байт називається парним.

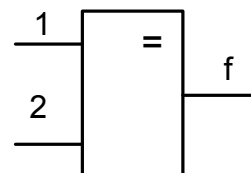
Власне в коді ISO-7 в молодших 7-ми бітах бувають парні і непарні комбінації, то для звернення всього байта в парний, на етапі програмування в 8-й біт заноситься 1 або 0, таким чином, з носія програми байт завжди зчитується парним, але при передачі через канал зв'язку один з бітів може вводиться неправильно, тобто інверсно.

Для усунення цього на приймальній стороні проводиться контроль парності прийнятого байта, цей контроль може виконуватися апаратно або програмно.

- *Апаратний метод* полягає у використанні апаратно реалізованої функції "Виключаюче АБО" у вигляді відповідного логічного елемента. Логічна функція "Виключаюче АБО" еквівалентна додаванню двох аргументів з відкиданням переносу в старший розряд.

Таблиця 6.1 – Таблиця істинності

X1	X2	X3
0	0	0
0	1	1
1	0	1
1	1	0



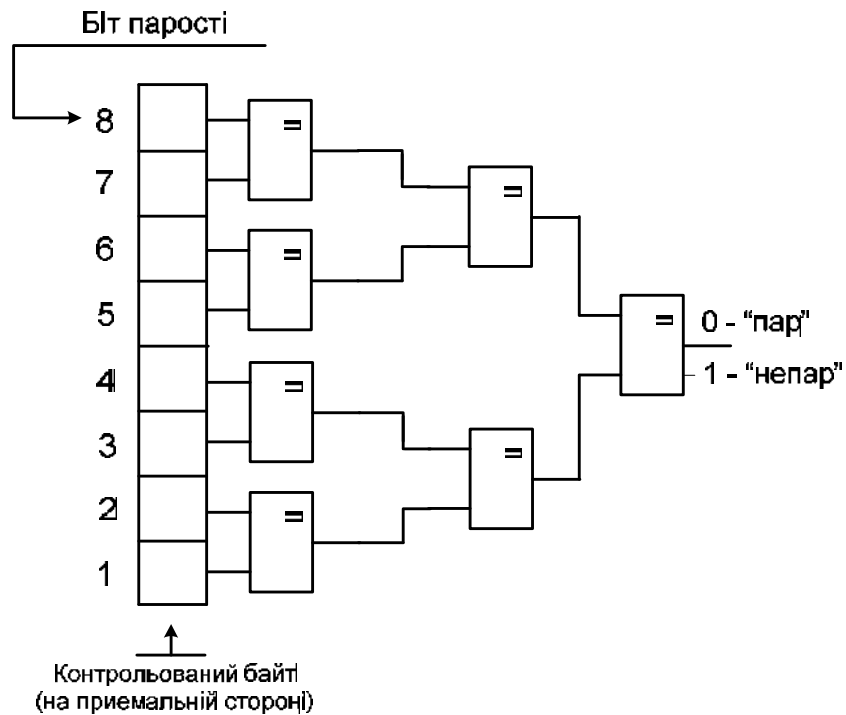


Рис. 6.1 – Апаратний метод

- *Програмний метод контролю* передбачає послідовний аналіз кожного біта байта за таким алгоритмом: якщо в аналізованому біті «0» то попередньо скинутий прапор парності не інвертується, а якщо «1», то біт парності інвертується. У результаті після аналізу восьми бітів для непарного байта прапор дорівнює «1 » (рис. 6.2).

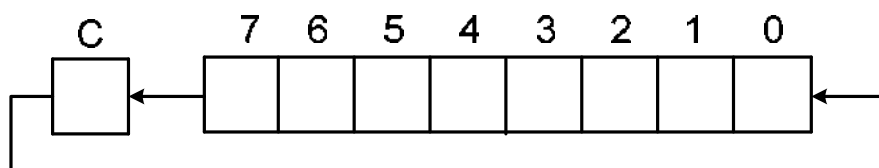


Рис. 6.2 – Програмний метод контролю

Після завантаження в регістр «C» чергового біта команди умовного переходу перевіряють вміст «C», якщо там «0» то інвертування прапора не виконується, а якщо «1» - то виконується.

Приклад:

```
.include 8535def.inc  
.def tmp=r16  
.def count=r17  
.def musk=r18
```

```
rjmp RESET  
.org 10
```

```
RESET:  clr tmp  
        clc  
        out DDRA, tmp  
        ser tmp  
        out DDRB, tmp  
        sbi DDRC, 0           ;тільки порт C на ввід  
        cbi PortC, 0         ;скидання прапору помилок  
        ldi count, 9         ;кількість циклів  
        ldi musk, 1          ;маска  
        in r20, pinA         ;ввід байта  
  
m1:     rot r20  
        brcc m2  
        in r19, PortC        ;ввід прапору  
        eor r19, musk        ;інверсія
```

```

out PortC, r19          ;вивід прапора

m2:    dec count
        brne m1
        sbis PortC, 0    ;якщо байт парний
        out Port, r20    ;вивід байта

m3:    rjmp m3

```

Другий етап контролю – контроль структури адреси

На другому етапі проводиться контроль правильності структури адреси в коді ISO-7. Адреси мають фіксовану структуру, наприклад, команди геометричних переміщень X, Y, Z мають таку структуру:

- 1-й символ <БУКВА>;
- 2-й символ «+», «-», тобто <Не буква> і <не цифра>, а <спеціальний символ>. Далі йдуть 4, 5 або 6 цифр десяткових цифр у двійково-десятковому коді (залежно від вибраного формату).

Код ISO-7 спеціально орієнтований на даний вид контролю для чого в кожному байті передбачені ознаки закодованого символу, а саме: літери, цифри або спеціального символу.

Таким чином, програмі контролю досить взяти ці ознаки і порівняти з маскою, в якій закладена правильна структура.

Зазначені два етапи є стандартними для систем ЧПК загального призначення фактично така система може вловлювати лише поодинокі помилки в байтах якщо тільки один біт є невірним.

Якщо в байті буде два невірних біта, то він залишиться парним і буде оцінений системою контролю як байт без помилок.

У складних і відповідальних системах ЧПК також використовуються перешкодостійкі коди, які виявляють подвійні помилки бітів і виправляють їх. В основному більш складні перешкодостійкі коди застосовуються на розгалужених робототехнічних комплексах з підвищеними вимогами до надійності роботи. На практиці частіше застосовується контроль непарності, коли правильним вважається непарний байт. Це пов'язано з тим, що «0» вважається парним числом, і під час вступу такого сигналу іноді складно визначити його зміст – чи є цей сигнал ознакою правильного байти або це наслідок відмови обладнання.

Перетворення кодів

Перетворення двійково-десятькового коду програми в двійковий застосовується тільки для геометричної інформації. Це пов'язано з тим, що інформація в програмі подається до двійково-десятьковому коді, а процесор інтерполятора працює в двійковому коді.

З іншого боку, на виході інтерполятор інформація представлена у двійковому коді, а для передачі в підсистему індикації її необхідно перетворити в двійково-десятьковий формат. Отже, необхідне зворотне перетворення.

Контрольні питання

1. Наведіть таблицю істинності для функції «Виключаюче АБО».
2. Поясніть, як працює апаратна схема контролю парності.
3. Яким чином забезпечується парність байтів у коді ISO-7?
4. Поясніть алгоритм програмного контролю парності байта.
5. Поясніть необхідність перетворення кодів у системі ЧПК.

ПЕРЕТВОРЮВАЧІ КОДІВ

У програмі, складеній у коді ISO-7 числова інформація про геометричні переміщення представлена у двійково-десятковому кодi, а процесор інтерполятора працює у двійковому кодi. Отже, необхідне перетворення двійково-десятьового коду у двійковий. З іншої сторони, інформація про переміщення на виході інтерполятора представлена у двійковому кодi, а для відображення цієї інформації на цифровому індикаторі вона має бути представлена у двійково-десятьовому кодi. Тобто, необхідне перетворення двійкового коду у двійково-десятьовий.

В залежності від пред'явлених вимог перетворювач кодів, який виконує вказані вище перетворення, може бути реалізований апаратно або програмно. Роботу такого перетворювача розглянемо за допомогою функціональної схеми на рис. 7.1.

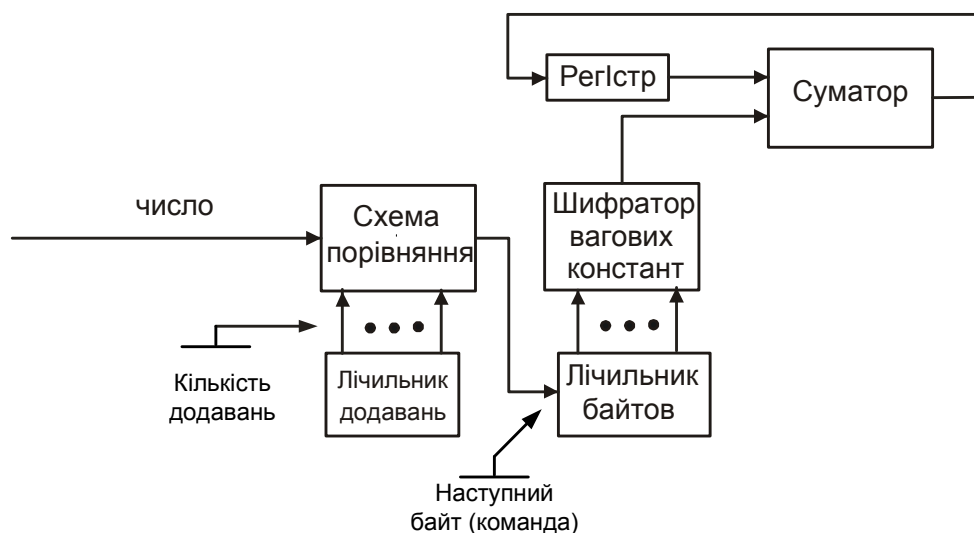


Рис. 7.1 – Функціональна схема перетворювача двійково-десятьового коду у двійковий

Завдання цієї схеми: по черговому введенню від чотирьох до шести двійково- кодованих десяткових чисел з програми, перетворення їх в еквівалентне двійкове число і розміщення його в регістрі результату Rг.

Принцип роботи схеми

Інформація в програмі представляється у вигляді послідовності байтів в кодї ISO в кожному байті тільки чотири молодших біта представляють числову інформацію.

Наприклад: X - 014256 - команда в кодї ISO на носії буде виглядати наступним чином (рис. 7.2):

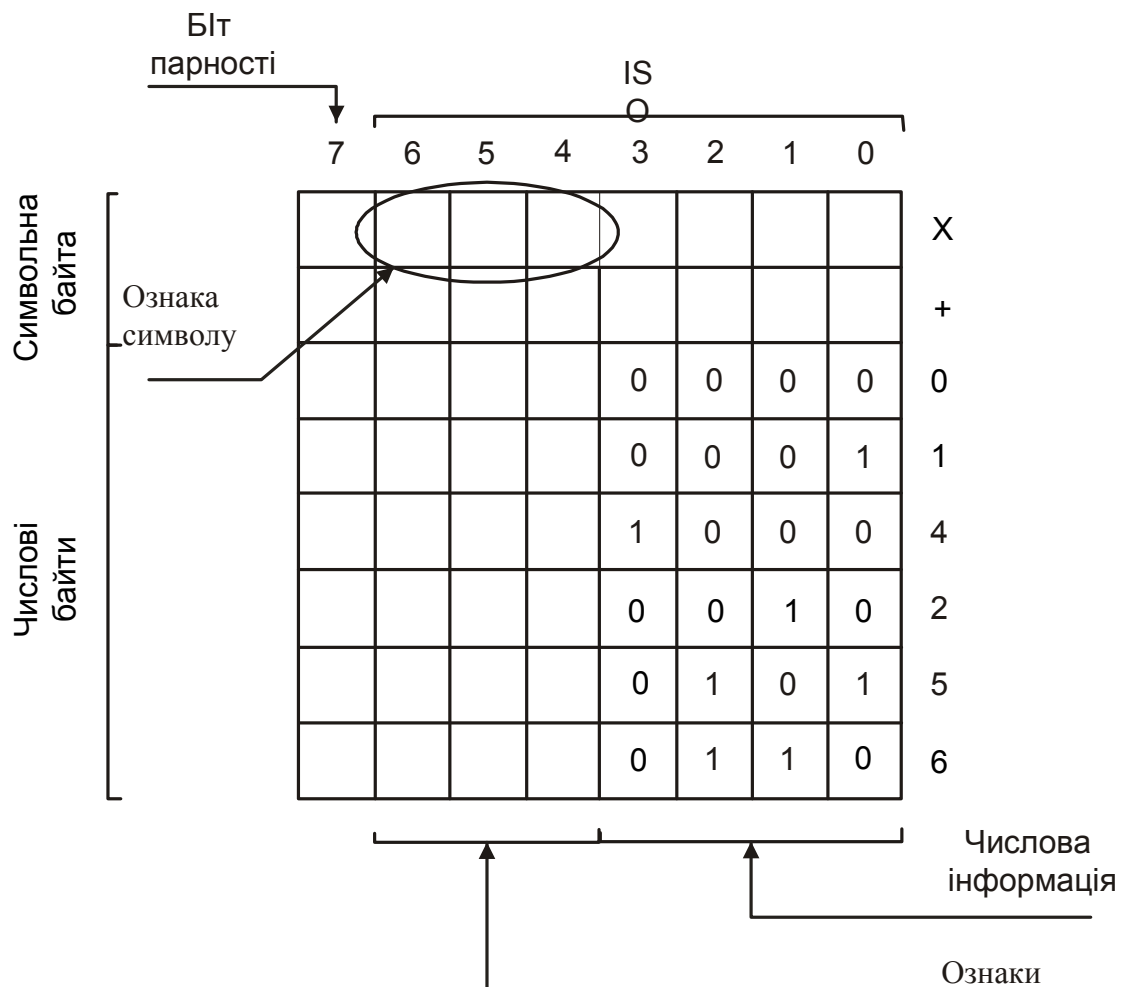


Рис. 7.2 – Зображення команди X- 014256 в кодї ISO

Для визначення ваги розряду необхідно його вагову константу помножити на цифру в даному розряді, результат множення можна назвати приватною сумою, остаточний результат виходить складанням часткових сум отриманих для кожного розряду.

У наведеній функціональній схемі процес множення замінюється необхідною кількістю додавань. Всього додавань від 1 до 9. Процесом додавань керує частина схеми, яка включає схему порівняння і лічильник додавань.

Спочатку друге число розряду, після чого починається процес додавання вагової константи до вмісту регістра, при кожному додаванні в лічильник складань додається «1». Процес триває поки вміст лічильника складання не виявиться рівним другому числу розряду.

Таким чином отримується часткова сума даного розряду і вона зберігається в регістрі P_r . Потім вводиться числова тетрада наступного розряду. Лічильник байтів збільшується на «1» и передає в шифратор номер наступного байту, на основі чого останній генерує нову вагову константу. Цей процес, після перетворення всіх чисел байтів от 4 до 6, після чого в регістрі P_r отримуємо еквівалентне двійкове число, представляє задане переміщення в дискретах, в даному випадку по осі X.

Потім число завантажується в робочий регістр інтерполятор. Дана схема дозволяє виконати зворотне перетворення двійкового числа в двійково-десятькове. Ця необхідність виникає при відображенні вихідного коду інтерполятора на панель індикації. У цьому випадку вихідне двійкове число завантажується в регістр P_r , суматор переводиться в режим віднімання і процес виконується так: з вмісту регістра P_r починає відніматися вагова константа старшого розряду. Кількість віднімань фіксується лічильником віднімань (колишній лічильник додавань) процес

віднімань продовжується до тих пір поки результат не стане негативним, після чого до цього результату один раз додається вагова константа і відповідно лічильник віднімання зменшується на одиницю. Таким чином вміст лічильника віднімання буде представляти десяткову цифру старшого розряду. Аналогічно обчислюються константи молодших розрядів.

Приклад: Нехай задано двійкове число на виході інтерполятора V_{ix} , його десятковий еквівалент 486321



Рис. 7.3 – Приклад обчислення

Константа: 100000

10000

Залишок: 86321

-10000

Розглянемо перетворення двійково-десятькового числа у двійковий код.

000042_{10} – десятковий код

Це число в програмі має вид 0000.0000.0000.0000.0100.0010₂₋₁₀ .

Таблиця 7.1

Число в байті	Лічильник додавань	Лічильник байтів	Вагова константа	Вміст регістра	Схема порівнянь	Примітка
0100	0000	0101	1010	000000 1010	≠	Початок обробки 5-го байта
0100	0001	0101	1010	001010 1010	≠	1-е додавання
0100	0010	0101	1010	010100 1010	≠	2-е додавання
0100	0011	0101	1010	011110 1010	≠	3-е додавання
0100	0100	0101	1010	101000	=	4-е додавання
0010	0000	0110	0001	101000 0001	≠	Початок обробки 6-го байта
0010	0001	0110	0001	101001 0001	≠	1-е додавання
0010	0010	0110	0001	<u>101010</u> 42 ₁₀	=	2-е додавання
						Кінець перетворення

Односпрямований перетворювач двійково-десятькового коду в двійковий

У тих випадках, коли не потрібно виконувати зворотне перетворення, можна використовувати більш простий перетворювач, виконаний за такою функціональною схемою (рис. 7.4).

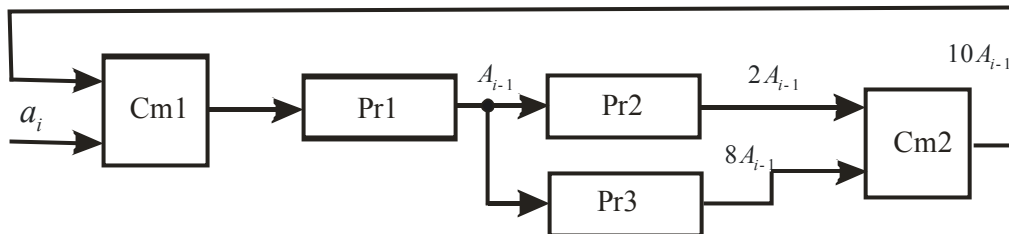


Рис. 7.4 – Односпрямований перетворювач двійково-десятькового коду у двійковий

Схема складається з 2-х суматорів: одного повнорозрядного регістру Рг1 і регістрів зсуву Рг2, Рг3, що виконують відповідно операції множення на 2 і на 8. Ця схема також демонструє приклад, коли правильно вибраний алгоритм дозволяє ефективно вирішити задачу.

Принцип роботи перетворювача

Для прикладу візьмемо 5-ти розрядне десяткове число, яке в загальному випадку може бути представлено як:

$$A = 10000a_1 + 1000a_2 + 100a_3 + 10a_4 + 1a_5$$

$a_1 \dots a_5$ – цифри десяткових розрядів

Безпосередня реалізація цього виразу потребує великих обчислювальних витрат з використанням команд множення. Цього можна уникнути, якщо представити вираз у вигляді:

$$A = \{[(10a_1 + a_2)10 + a_3]10 + a_4\}10 + a_5$$

З останнього виразу видно, що алгоритм зведений до повторення однорідних операцій і полягає в наступному:

1) Цифра старшого розряду множиться на 10 і додається до цифри наступного розряду. Результат називають проміжною сумою.

2) Кожна проміжна сума (називається проміжною сумою A) множиться на 10 і додається з цифрою наступного розряду.

Процес перетворення:

- в початковому стані в регістрі результату $РГ1 = 0$. Тому, нульова часткова сума $A_0 = 0$. Кожна наступна часткова сума A_i визначається формулою:

$$A_i = 10A_{i-1} + a_i$$

- із надходженням цифри старшого розряду визначається 1-а часткова сума:

$$A_1 = 10A_0 + a_1 = 0 + a_1 = a_1$$

- потім друга, третя і т. д.

$$A_2 = 10A_1 + a_2 = 10a_1 + a_2$$

$$A_3 = 10A_2 + a_3 = (10a_1 + a_2) \cdot 10 + a_3$$

$$A_4 = 10A_3 + a_4 = [(10a_1 + a_2) \cdot 10 + a_3] \cdot 10 + a_4$$

$$A_5 = 10A_4 + a_5 = \{[(10a_1 + a_2) \cdot 10 + a_3] \cdot 10 + a_4\} \cdot 10 + a_5$$

Процес перетворення займає кількість циклів, яке дорівнює кількості десяткових розрядів.

Фактично даний алгоритм використовує операцію множення на 10, замість множення на 10 тис., 1 тис., 100, 10 по вихідній формулі. Це призводить до економії апаратних і програмних засобів. Множення на 10 тут виконується апаратним способом за допомогою регістрів $РГ2$, $РГ3$ і суматора $См2$. Регістр $РГ2$ забезпечує множення на 2 шляхом зсуву вліво на 1 розряд. Регістр $РГ3$ виконує множення на 8 шляхом зсуву вліво на 3 розряди. У результаті на виході суматора $См2$ отримуємо множення на 10. Схеми порівняння 2-10 коду і \sum_i

$$10_{10}=1010_2$$

$$42_{10}=101010_2$$

Приклад

$$A = 42_{10} = 0000.0000.0000.0100.0010_{210}$$

Таблиця 7.2

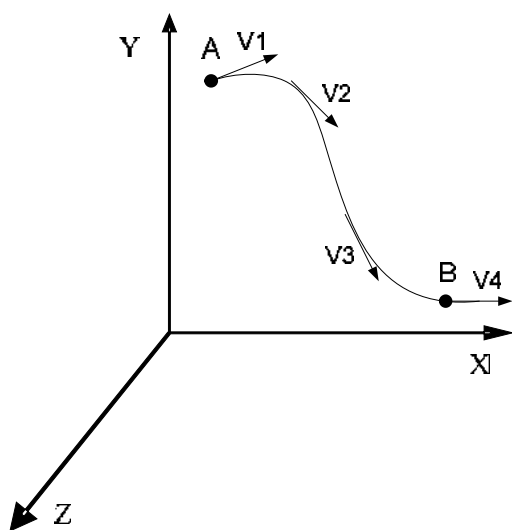
Розряд	Цифра розряду, a_i	A_{i-1}	$2A_{i-1}$	$8A_{i-1}$	$10A_{i-1}$	A_i	Примітка
-	-	0	0	0	0	0	Початковий стан
5,4,3	0,0,0	0	0	0	0	0	1,2,3 сума
2	0100	000000	000000	000000	000000	0100	4 сума
1	0010	000100	001000	100000	101000	101010	5 сума

Контрольні питання

1. З якою метою в системі ЧПК проводиться перетворення кодів?
2. Поясніть принцип роботи перетворювача кодів в режимі прямого перетворення.
3. Поясніть принцип роботи перетворювача кодів в режимі зворотного перетворення.
4. Поясніть принцип роботи односпрямованого перетворювача кодів.
5. Як виконується множення на 10 у схемі односпрямованого перетворювача кодів.

БЛОК ЗАВДАННЯ ШВИДКОСТІ ПОДАЧ

При складанні програми обробки, технолог закладає в неї необхідну швидкість подачі. Ця швидкість визначається в програмі кодом під адресою F. Задача відпрацювання заданого вектора швидкості вирішується в два етапи:



1-й етап вирішується в БЗШ, де на основі коду швидкості виробляється проміжний параметр $f_{роб}$, пропорційний коду завдання швидкості. По суті ця величина є скалярним значенням швидкості або її модулем. Робоча частота потім використовується як сигнал синхронізації для інтерполятора, визначаючи швидкість інтерполяції.

Рис. 8.1 – Відпрацювання вектора швидкості

Наприклад, з підвищенням $f_{роб}$ - швидше відбувається інтерполяція, відповідно вище частота видачі коду на виході інтерполятор і більше швидкість подачі.

2-й етап. Розкладання вектора заданої швидкості на координати (проекції) виконує інтерполятор так, щоб дійсна векторна швидкість була б рівною заданій в програмі (рис. 8.1). Відновлення заданого вектора швидкості шляхом знаходження проекцій цього вектора на координати та видачі відповідних завдань на приводи подач.

Блок завдання швидкості подачі (БЗШ) призначений для приймання коду швидкості із програми й формування на підставі цього коду робочої частоти $f_{роб}$ для синхронізації інтерполятора. Імпульси $f_{роб}$, що надходять на вхід інтерполятора, визначають швидкість інтерполяції, частоту вихідних імпульсів інтерполятора, а отже й швидкість подачі. У сучасних УЧПУ $f_{роб}$ звичайно перебуває в межах $1 \div 5$ кГц.

Визначимо швидкість подачі для крайніх значень робочої частоти.

Нехай: $f_{роб} = 1$ кГц, $\Delta = 0,01$ мм.

$V = f_{роб} \cdot \Delta = 1000 \cdot 0,01 = 10$ мм/з = 600 мм/хв = 0,6 м/хв.

При $f_{роб} = 10$ кГц,

$V_{max} = 5 \cdot 1,2 = 6$ м/хв.

Програмно-апаратна реалізація БЗШ

Існують три підходи до побудови БЗШ:

1. Побудова БЗШ на основі керованого дільника частоти (рис 8.2)

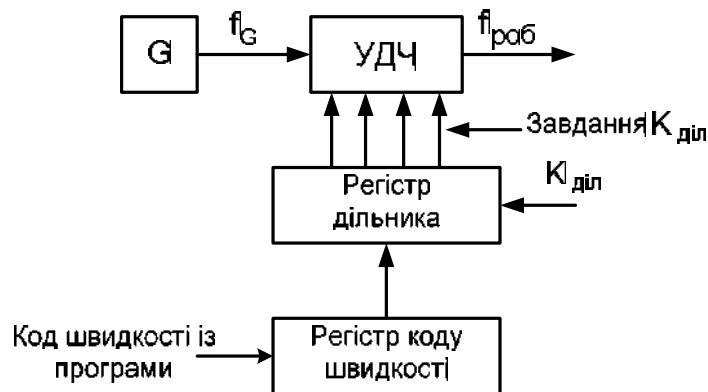


Рис. 8.2 – Побудова БЗШ на основі керованого дільника частоти

Недоліком такого підходу є те, що немає зручного зв'язку між кодом швидкості із програми й швидкістю подачі. Цей зв'язок задається у вигляді досить великої таблиці.

2. Модифікована перша схема, що дозволяє обходитися без таблиці за рахунок апаратного ускладнення схеми.

3. Реалізація на мікроконтролері за допомогою таймера. Для цього використовується 8- або 16-бітний таймер, настроєний на частоту переповнення в діапазоні від 1 до 5 кГц. У цьому випадку програма інтерполяції інтегрується в підпрограму-обробник переривань за переповненням таймера.

Розглянемо реалізацію БЗШ із застосуванням елементів середнього ступеня інтеграції, таких як суматори та регістри. Функціональну схему такого БЗШ наведено на рис. 8.3.

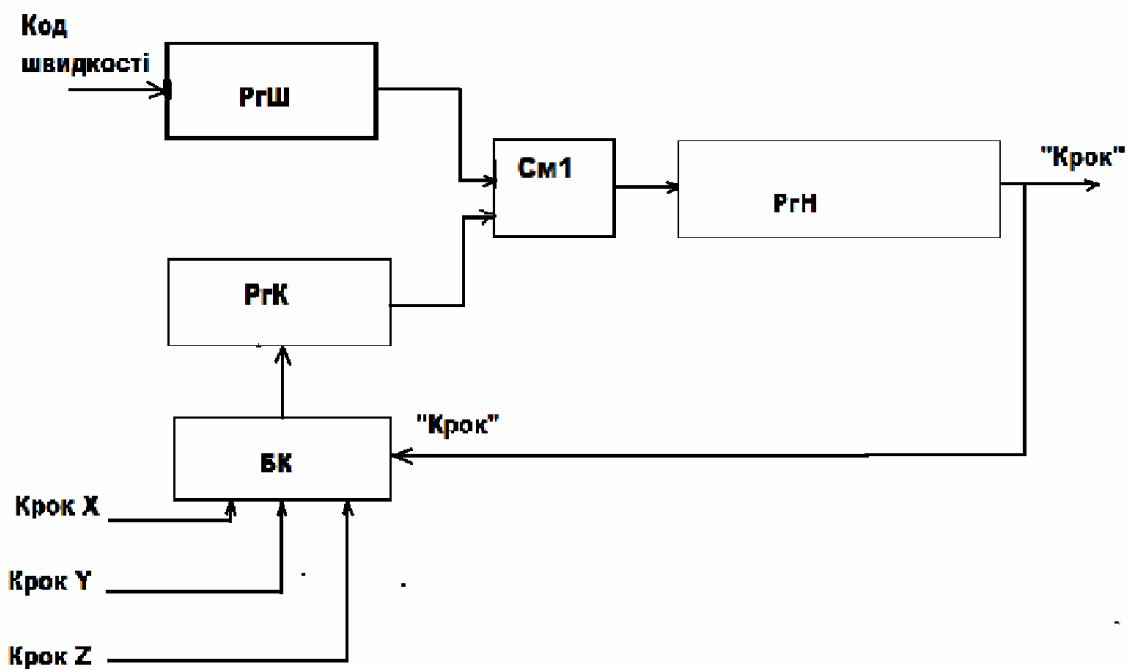


Рис. 8.3 - Функціональна схема блоку завдання швидкості

У цій схемі застосовано три регістри, а саме – регістр-накопичувач РгН, регістр швидкості РгШ та регістр констант РгК, а також суматор См1.

Принцип роботи схеми полягає в тому, що вміст регістра швидкості РгШ, в якому зберігається код швидкості, безперервно за допомогою суматора См1 складається з константою, що зберігається у регістрі

констант РгК, і додається до вмісту регістра-накопичувача РгН. В момент переповнення цього регістра формується сигнал «Крок».

Припустимо, що робота БЗШ відбувається з тактовою частотою 10 кГц. Тоді для отримання імпульсів «Крок» з частотою 1 кГц вміст регістра швидкості РгШ повинно бути в 10 разів менше ємності регістра-накопичувача. Прийmemo, що регістр-накопичувач РгН має ємність 14-розрядів (тобто, максимальне число, яке може в ньому розміститися) становить 16.383.

Для отримання імпульсів «Крок» з частотою 1 кГц необхідно, щоб переповнення регістра-накопичувача відбувалося кожного десятого додавання. Очевидно, що для цього в регістрі швидкості РгШ має бути число 1.638.

Важливою перевагою такої схеми є те, що в програмі можна задавати не номер швидкості, а саму швидкість у загальноприйнятих одиницях – міліметрах/хвилину. Однак, для цього необхідно, щоб ємність регістра-накопичувача була дещо меншою, ніж 16.383.

Дійсно, частоті імпульсів «Крок» 1 кГц при дискретності 0,01 мм відповідає швидкість подачі 600 мм/хв. Якщо такий код має міститися в регістрі швидкості РгШ, то ємність регістра накопичувача повинна бути 6.000. Для отримання такої ємності кожного імпульсу Крок у регістр-накопичувач РгШ додається константа, яка визначається, як різниця між дійсною і необхідною ємністю регістра-накопичувача:

$$16.383 - 6.000 = 10.383.$$

Ця константа формується блоком констант БК і з появою імпульсу «Крок» вводиться в регістр констант РгК, після чого за допомогою суматора См2 додається до вмісту регістра-накопичувача РгН.

Вказане значення константи справедливе лише у випадку кроків по одній із координат. Якщо відпрацьовуються одночасні кроки по двох координатах, то ємність регістра накопичувача повинна бути в 1,41 рази більша, а у випадку одночасних кроків по 3-х координатах – в 1,73 рази більша. Це досягається відповідною зміною значення константи у блоці констант БК.

При роботі в межах робочих швидкостей подач частоти вихідних імпульсів БЗШ лежать в діапазоні до 2 кГц. При таких частотах сучасні приводи подач не потребують плавних розгонів та гальмувань, що суттєво спрощує реалізацію БЗШ.

В режимі швидких переміщень частота вихідних імпульсів БЗШ значно вища і може досягати 8 кГц. В цьому режимі необхідно застосовувати плавний розгін та гальмування, тобто поступове зростання частоти до 8 кГц на початку ділянки переміщення і поступове зниження частоти в кінці ділянки.

Для цього застосовується спеціальний генератор, частота якого поступово збільшується від 400 – 500 Гц до 8 кГц, а потім, коли до кінця ділянки залишається задана відстань, частота поступово знижується до вихідного значення.

Контрольні питання

1. Як виконується відпрацювання заданого вектора швидкості?
2. Поясніть функції блоку завдання швидкості.
3. Наведіть типові значення подач у верстатах з ЧПК.
4. Поясніть принцип роботи БЗШ на ділянках частоти.
5. Поясніть принцип роботи БЗШ на регістрах та суматорах.

БЛОК КЕРУВАННЯ ПРИВОДАМИ ПОДАЧ

Блок керування приводами подач (БКП) містить системи управління приводами на кожен керовану координату. Завдання приво-ду полягає в перетворенні керуючих сигналів з виходу інтерполятора до виду, придатного для управління приводами подач. З виходу інтерполятора сигнали надходять у вигляді певного коду. Зазвичай можливі два види такого коду:

1. *Унітарний код*. Кожен імпульс відповідає одній дискреті переміщення робочого органу (рис. 9.1). Таким чином, імпульси надходять нерівномірно, кожен імпульс відповідає одній дискреті переміщення. У даному прикладі за 11,5 мс надійшло 12 імпульсів унітарного коду, що призвело до переміщення $\Delta x = 12 \cdot 0,010 = 0,12$ мм (рис. 5.1).

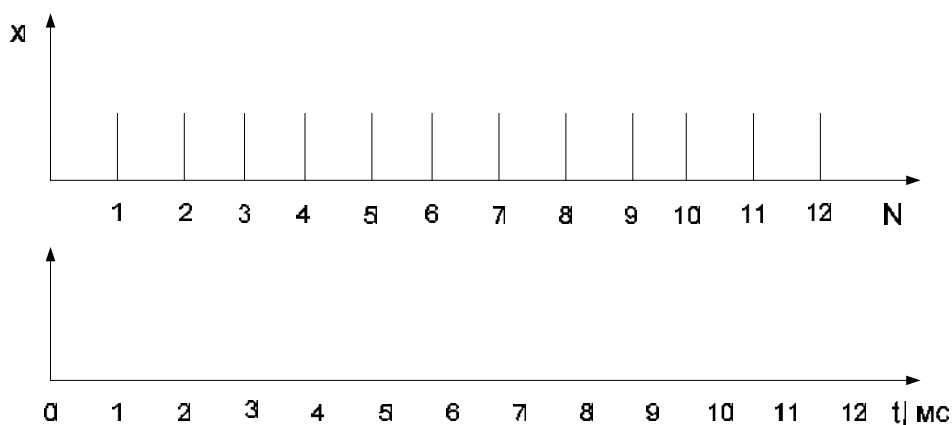
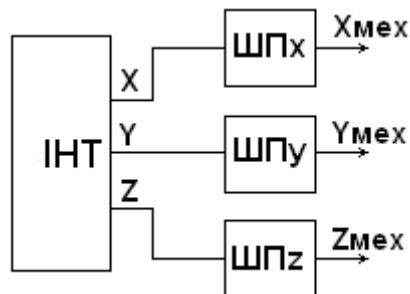


Рис. 9.1 – Діаграма унітарного коду

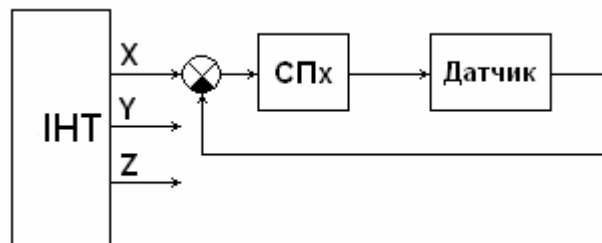
В малих системах ЧПК часто застосовується вихідний унітарний код. Це пов'язано з простим подальшим його використанням для керування приводом. Наприклад, якщо привод подачі кроковий, то взагалі не

потрібне додаткове перетворення. Кожен імпульс забезпечує поворот крокового двигуна на один крок (рис. 9.2 а)).

У разі застосування слідкуючого приводу (середні, великі потужності) схема будується таким чином, що імпульс унітарного коду видає завдання на регулятор положення слідкуючого приводу, який починає відпрацьовувати завдання, поки з датчика положення не надійде сигнал зворотного зв'язку про виконання завдання, після чого слідкуючий привід зупиняється (рис. 9.2 б)).



а) – Керування кроковим двигуном



б) Керування слідкуючим риводом

Рис. 9.2

2. *Цифровий код.* В цьому випадку з інтерполятора надходить цифровий код, який також може бути різного вигляду (паралельний, послідовний), можливе використання звичайного (позиційного) коду, а також різних варіантів завадостійких кодів і т.д.

Такий підхід застосовується, коли використовується спеціалізована цифрова система управління приводами подач.

В приводах використовують керовані від цифрових перетворювачів синхронні або асинхронні двигуни. Безколекторні синхронні (вентильні) двигуни для верстатів з ЧПК виготовляють із постійним магнітом на основі рідкоземельних елементів і оснащують датчиками зворотного зв'язку й гальмами. Асинхронні двигуни застосовують рідше, ніж синхронні.

Привод руху подач характеризується мінімально можливими зазорами, малим часом розгону й гальмування, невеликими силами тертя, зменшеним нагріванням елементів привода, широким діапазоном регулювання швидкості (типове значення $D = 10000$). Забезпечення цих характеристик можливе завдяки застосуванню кулькових та гідростатичних гвинтових передач, направляючих кочення й гідростатичних направляючих, безлюфтових редукторів з короткими кінематичними ланцюгами і т.д.

Принципи роботи решти блоків

Решта блоків носять допоміжний характер.

Блок введення корекцій (БВК) дозволяє з пульту керування вручну ввести корекцію в програму. В основному це стосується коригування геометричних переміщень на величину зносу інструменту.

У технологічну карту обробки занесені дані зносу інструмента. Наприклад, через 2 години роботи різець коротшає на 0,02 мм. Для того, щоб виготовлена деталь мала задані розміри необхідно відповідно відкоригувати переміщення центра фрези на цю величину кожні 2 години.

Блок цифрової індикації (БЦІ) призначений для виведення на пульт керування значення шляху, що залишився до кінцевої точки у зворотному відліку.

Блок корекції еквідистанти (БКЕ)

Математичне поняття: еквідистанта – це крива, проведена на однаковій відстані щодо іншої кривої (рис. 9.3).

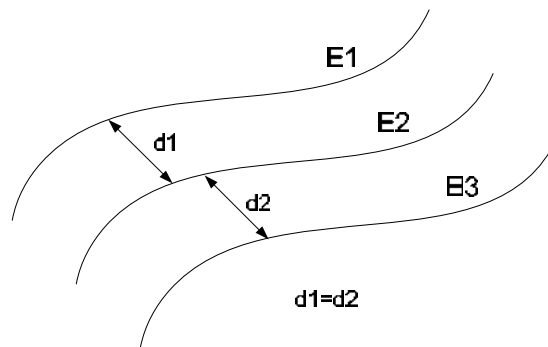


Рис. 9.3 – Три еквідистанти

В системах ЧПК еквідистантою називається траєкторія центру фрези при фрезерній обробці. В процесі обробки фреза зношується, її радіус зменшується, тому для отримання заданих розмірів деталі вихідну еквідистанту необхідно коригувати.

Принципи корекції еквідістанти

Як відомо, траєкторія обробки складається з прямих і кругових ділянок (рис. 9.4).

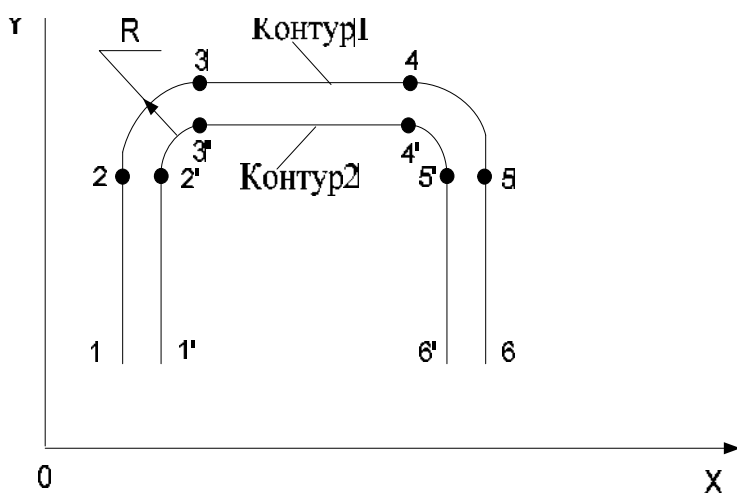


Рис. 9.4 – Траєкторія обробки

Контур1 виконаний на вихідній еквідістанті з інструментом нормальної якості.

Контур 2 отриманий після зносу інструменту.

Розміри деталі змінилися.

Аналіз даного рисунку дозволяє визначити такі принципи корекції

еквідістанти:

- 1) довжина прямих ділянок вихідної і скоригованої еквідістант однакова;
- 2) радіус кругових ділянок повинен бути відкоректований на величину зміни радіусу фрези.

Блок керування автоматикою верстата БКА

Функціонально блок представляє собою булевський процесор, який приймає інформацію під адресами S, M, T і забезпечує керування:

- головним приводом (приводом шпинделя);
- пристроєм зміни інструменту;
- дискретною автоматикою верстата (клапани, соленоїди, вентиляція та ін.).

Контрольні питання

1. Які функції виконує блок завдання швидкості?
2. Поясніть поняття унітарного коду.
3. Як організоване керування кроковим приводом подачі?

4. Що таке еквідистанта?
5. Поясніть принципи корекції еквідистанти.

ІНТЕРПОЛЯТОРИ

Інтерполятор – це центральний пристрій системи ЧПУ. Його призначення полягає в тому, щоб на підставі координат початкової й кінцевої точки траєкторії розраховувати в реальному часі координати проміжних точок, необхідні для керування приводами подач.

Система ЧПК являється дискретною і її замикання здійснюється через фіксовані інтервали часу, що дорівнюють періоду квантування. Частоти квантування зазвичай перебувають у межах від 0,5 до 10 кГц, тобто період квантування становить $\Delta t = 0,1 \div 2,0$ мс. За вказаний проміжок часу процесор інтерполятора повинен виконати значний обсяг обчислень із трьох- або чотирьох- байтними числами.

У минулому процесори інтерполяторів будувалися винятково апаратно. Зараз активно впроваджуються й програмні рішення (рис. 10.1).

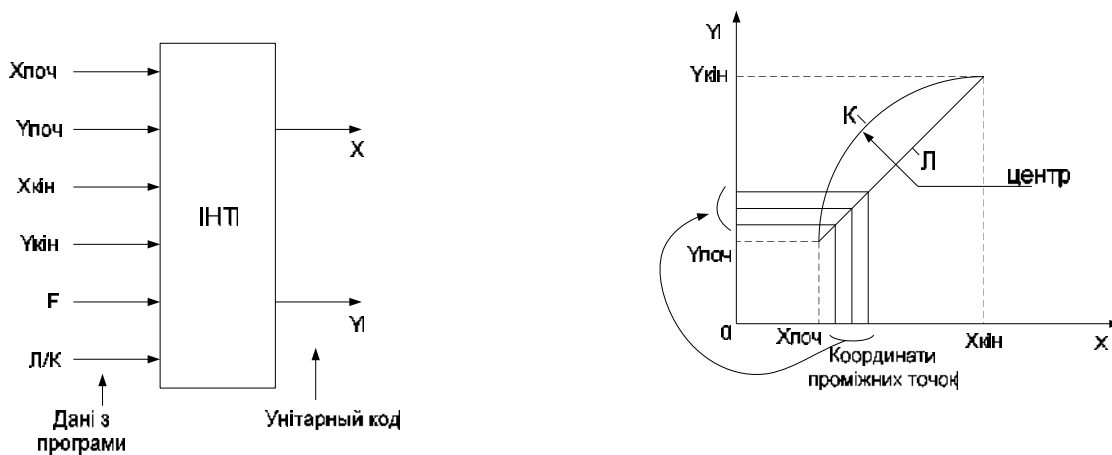


Рис. 10.1 Блок інтерполятора та типові траєкторії

F – швидкість переміщення по траєкторії;

$Л/ К$ – режим роботи (лінійний або кругова інтерполяція).

Виходом інтерполятора є завдання на привод подачі по кожній координаті. Як правило, вихідні сигнали представляються в унітарному коді в імпульсному

вигляді, де кожний імпульс відповідає переміщенню інструмента на одну дискрету.

Побудова ЧПК з інтерполятором побудована на допущенні, що будь-який, як завгодно складний контур руху може бути представлений у вигляді комбінації відрізків прямих і дуг кіл.

На практиці знаходять застосування інтерполятори 2-х видів:

- 1) лінійні;
- 2) лінійно-кругові.

Перші простіші за конструкцією, але другі більш точні й універсальні.

Приклад лінійного інтерполятора

Найбільш простою реалізацією методу інтерполяції є лінійний інтерполятор, що працює за принципом відбору імпульсів зі спеціального десяти-розрядного лічильника імпульсів. Інтерполятор реалізований на ділках частоти. Його схема має вигляд (рис. 10.2):

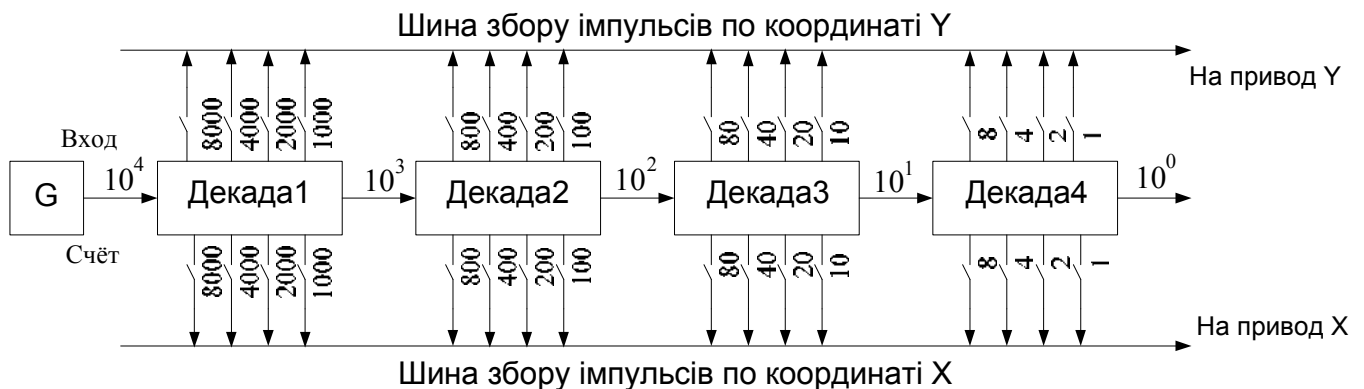


Рис. 10.2 – Схема лінійного інтерполятора

Для правильної роботи схеми необхідно відповідним чином замкнути ключі на лініях відбору імпульсів.

Для розуміння принципу керування ключами розглянемо програму відпрацювання лінійного відрізка траєкторії у коді ISO-7:

X+7625

Y+4375

Виконання цих команд пристроєм ЧПК призведе до руху робочого органу по траєкторії, представлений на рис. 10.3.

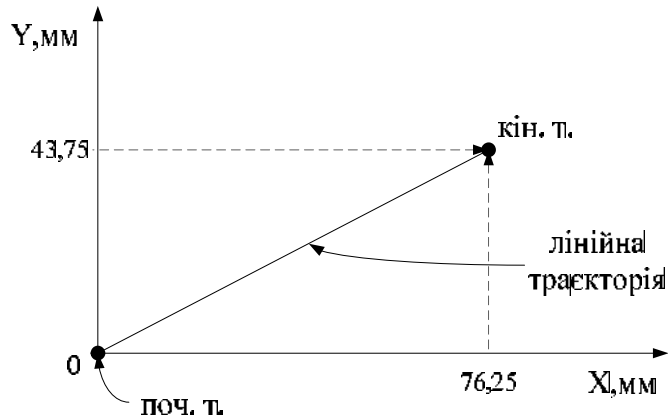


Рис. 10.3 – Траєкторія руху робочого органу

Тепер визначимо правило, за яким треба замкнути вимикачі на схемі інтерполятора, щоб отримати правильне відпрацювання заданої траєкторії. Для цього розглянемо схему розміщення команд на носії програми. Як видно із цієї схеми, числова інформація кожної команди займає

					X
					н
		0	1	1	7
		0	1	1	6
		0	0	1	2
		0	1	0	5
					Y
					н
		0	1	0	4
		0	0	1	3
		0	1	1	7
		0	1	0	5

Виявляється, що для відпрацювання заданих команд необхідно в кожній декаді замкнути ті ключі, яким відповідають одиничні біти у відповідному розряді. Наприклад, у команді по X у старшому розряді 7. Це означає, що на вихід дана декада повинна видати 7 тис. імпульсів. Це відбудеться якщо замкнути ключі $4000+2000+1000$.

Швидкість інтерполяції (швидкість подачі) визначається частотою вхідних імпульсів лічильників, тобто, чим вище частота, тим більша швидкість подачі.

Розглянута схема лінійного інтерполятора являє собою приклад спеціалізованого процесора, виконаного нестандартним чином.

Контрольні питання

1. Поясніть алгоритм відбору імпульсів у схемі лінійного інтерполятора.
2. Яким чином можна змінювати швидкість подачі?
3. Які дані з програми вводяться в інтерполятор?
4. Яку функцію виконує блок «Декада» в схемі інтерполятора?
5. За якою адресою задається швидкість переміщення по траєкторії?

ЛІНІЙНО-КРУГОВІ ІНТЕРПОЛЯТОРИ

У сучасному машинобудуванні більшість траєкторій переміщень інструменту містять прямолінійні й криволінійні ділянки. Хоча останні можуть бути апроксимовані відрізками прямих, очевидно, що апроксимація криволінійних ділянок дугами кіл може бути виконана більш точно.

Фізична реалізація системи керування в цьому випадку приводить до використання лінійно-кругових інтерполяторів (ЛКІ).

Методи побудови ЛКІ

Інтерполятор це спеціалізований процесор, що виконує математичні операції з двійковими числами великої розрядності (3 - 4 байти) у режимі реального часу.

Розвиток ЛКІ відбувалося в період коли керуюча обчислювальна техніка, тільки починала розвиватися, тому дуже важливим був пошук ефективних алгоритмів реалізації ЛКІ, що ґрунтувалися на їхній математичній моделі.

Теорія сучасних ЛКІ створювалася під час дефіциту обчислювального ресурсу (процесора), коли безпосередній розв'язок завдань інтерполяції був утрудненим, тому були розроблені алгоритми, що забезпечують розв'язок завдань інтерполяції з мінімальним використанням обчислювальних ресурсів. Таких алгоритмів побудови ЛКІ була розроблено декілька. З них розглянемо два, які часто застосовуються на практиці, а саме – метод оціночної функції та метод цифрових диференціальних аналізаторів.

1. Метод оціночної функції

Інтерполятор працюючий за методом ОФ можна представити у вигляді блоку, на який із програми надходить наступна інформація – координати початкових і кінцевих точок (X_n, Y_n, X_k, Y_k), значення швидкості подачі F і вид траєкторії, що відпрацьовується, - лінійна або кругова (л/к) (рис. 9.1). Вихідні сигнали інтерполятора являють собою дискретні сигнали в

послідовному або паралельному коді, які управляють приводами подач по координатах X і Y .

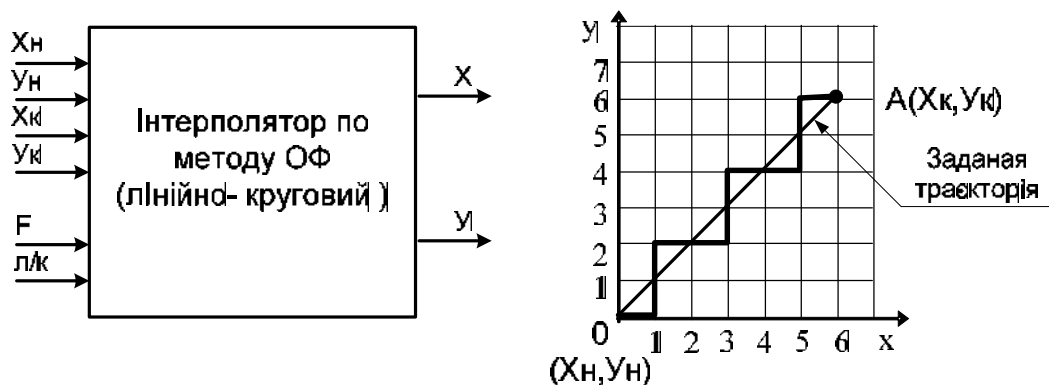


Рис. 11.1 – Представлення інтерполятора, працюючого за методом ОФ
Лінійна інтерполяція за методом ОФ

Розглянемо побудову лінійної траєкторії на площині XY . На рисунку 9.2 показана задана лінійна траєкторія OA та її дискретне наближення, яке формується ортогональними приводами подач.

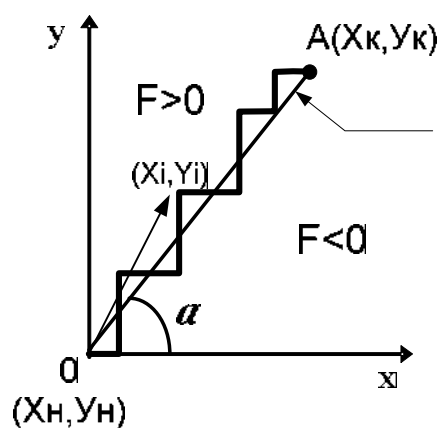


Рис. 11.2 – Задана лінійна траєкторія та її дискретне наближення

Для керування процесом інтерполяції необхідно визначити міру, за допомогою якої можна оцінити відхилення дійсної траєкторії від заданої. У якості такої міри використовується кут нахилу поточного радіус-вектора до осі абсцис, а точніше тангенс цього кута.

Тоді відхилення контуру, що відпрацьовується, від заданого можна визначити в такий спосіб:

$$\operatorname{tg} \alpha_{ij} - \operatorname{tg} \alpha = \frac{y_j}{x_i} - \frac{y_k}{x_k} = \frac{1}{x_i \cdot x_k} (y_j \cdot x_k - x_i \cdot y_k) \quad (1)$$

Знак виразу визначається знаком чисельника, оскільки знаменник завжди позитивний. Тому для оцінки дійсного контуру від заданого досить використовувати тільки чисельник, який і називається *оціночною функцією*

$$F = y_i \cdot x_k + x_i y_k. \quad (2)$$

Властивості оціночної функції

З виразу (2) випливає, що задана траєкторія ділить площину інтерполяції на дві області:

$F > 0$ – над траєкторією;

$F < 0$ – під траєкторією.

Очевидно, що на самій заданій траєкторії $F = 0$.

Алгоритм руху за методом ОФ

Перший крок завжди виконується по осі X , потім рух виконується почерговими одиничними кроками по кожній з координат залежно від знака оціночної функції, а саме:

якщо $F \geq 0$, то виконується крок по координаті X ;

якщо $F < 0$, то виконується крок по координаті Y .

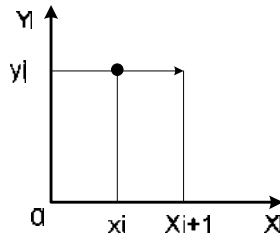
Спрощене обчислення оціночної функції

Безпосередня реалізація виразу (2) потребує операцій множення багато-байтних чисел, що на час розробки цих інтерполяторів було важко реалізувати апаратним способом. Тому були розроблені методи більш простого обчислення ОФ.

Метод спрощеного обчислення ОФ базується на знанні величини ОФ у попередній точці й напрямку кроку.

Розглянемо можливі випадки:

1. Обчислення оціночної функції при кроці по осі X

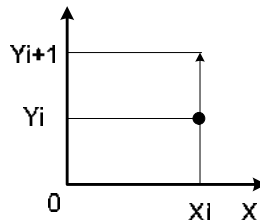


Визначимо значення оцінної функції в новій точці

$$F_{i+1,j} = y_j x_k - x_{i+1} y_k = y_j x_k - (x_i + 1) y_k = y_j x_k - x_i y_k - y_k = F_{ij} - y_k$$

Отже, при кроці по осі X нове значення оціночної функції знаходиться шляхом віднімання від старого значення y_k .

2. Обчислення оціночної функції при кроці по осі Y



$$F_{i,j+1} = y_{j+1} x_i - x_i y_k = (y_j + 1) x_k - x_i y_k = y_j x_k - x_i y_k + x_k = F_{ij} + x_k$$

Таким чином, нове значення оціночної функції при кроці по осі Y знаходиться шляхом додавання до старого величини x_k . Очевидно, що для апаратної реалізації обчислення буде потрібен суматор, що працює в режимі віднімання чисел відповідної розрядності (3 - 4 байти).

Контрольні питання

1. Поясніть, які функції в системі ЧПК виконує інтерполятор.
2. Як визначається оціночна функція при лінійній інтерполяції?
3. На які області ділить площину задана траєкторія?
4. Як визначається нове значення оціночної функції при кроці по осі X?
5. Як визначається нове значення оціночної функції при кроці по осі Y?

ПОБУДОВА ЛІНІЙНОГО ІНТЕРПОЛЯТОРА ЗА МЕТОДОМ ОЦІНОЧНОЇ ФУНКЦІЇ

Розглянемо побудову лінійного інтерполятора за методом оціночної функції.

Вихідні умови (рис. 12.1):

1. Лінійна інтерполяція по методу ОФ.
2. Координати кінцевої точки $X_K = 4, Y_K = 7$.
3. Координати початкової точки $X_K = 0, Y_K = 0$.

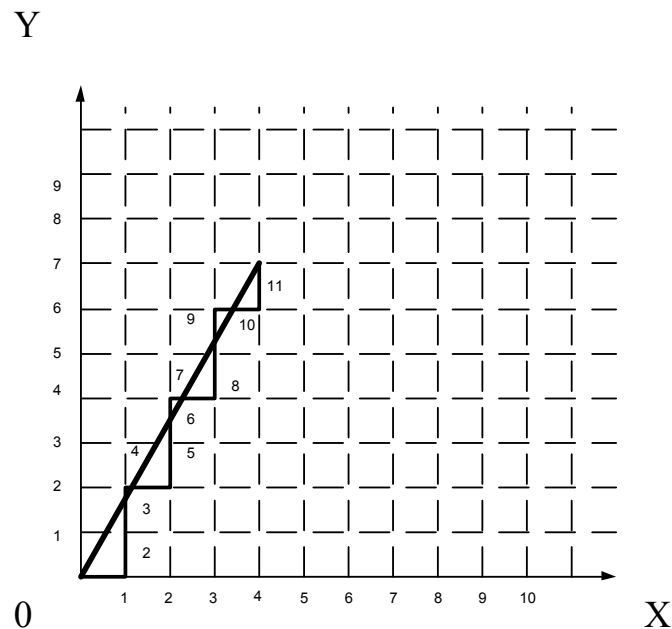


Рис. 12.1 – Задана та відпрацьована траєкторії

1. Обчислення оціночної функції

$$F_{i+1,j} = y_j x_k - x_{i+1} y_k = y_j x_k - (x_i + 1) y_k = y_j x_k - x_i y_k - y_k = F_{ij} - y_k$$

$$F_{i,j+1} = y_{j+1} x_i - x_i y_k = (y_j + 1) x_k - x_i y_k = y_j x_k - x_i y_k + x_k = F_{ij} + x_k$$

На основі отриманих виразів можна побудувати функціональну схему лінійного інтерполятора за методом ОФ.

4. Функціональна схема лінійного інтерполятора за методом ОФ представлена на рис. 12.2.

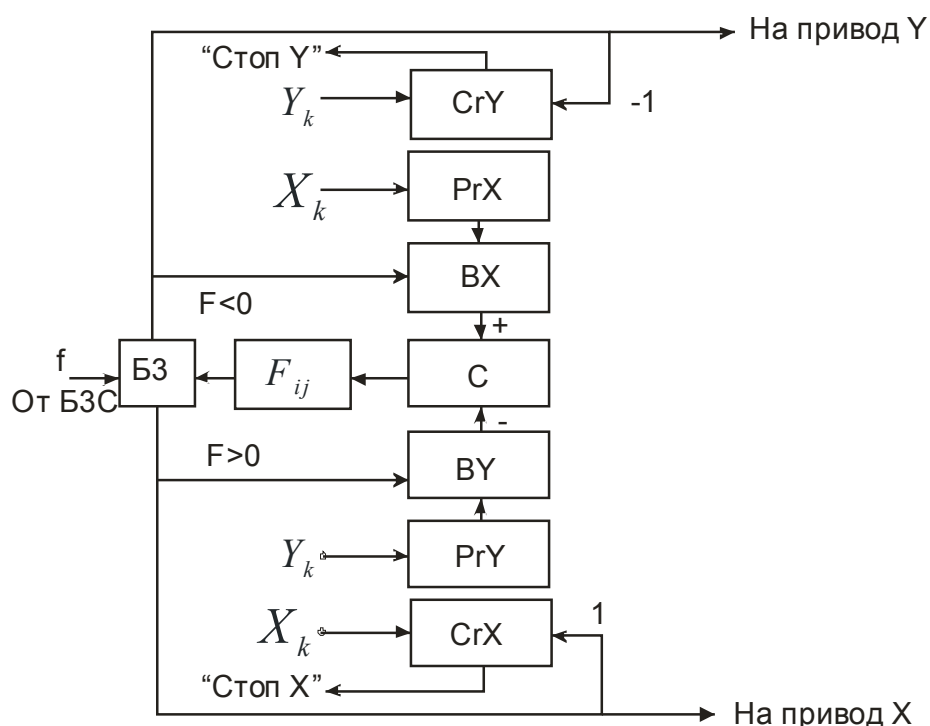


Рис. 12.2 – Функціональна схема лінійного інтерполятора по методу ОФ

Перед початком інтерполяції регістр оціночної функції містить нуль, а в регістри координат і відстані занесені координати кінцевої точки. Процес інтерполяції починається з надходження першого імпульсу робочої частоти. Оскільки у вихідному стані оціночна функція дорівнює нулю, то перший імпульс завжди відправляється на координату X. Одночасно цей же імпульс відкриває вентиль ВУ і виконує віднімання від старого значення оціночної функції величини Y_k , тобто з 0 обчислюється додатна величина. Нове значення оціночної функції буде від'ємне, отже, другий імпульс від БЗШ буде спрямований на привод Y.

Таким чином, незалежно від нахилу траєкторії перший імпульс завжди направляється на координату X, а другий – на координату Y.

При подальшому переміщенні знак ОФ уже залежить від нахилу траєкторії й, отже, 3-й імпульс може бути спрямований на координату X або Y.

Лічильники відстані ЛХ і ЛУ не приймають прямої участі в процесі інтерполяції й призначені для звіту відстані до кінцевої точки траєкторії. Обнулення цих лічильників визначає закінчення процесу інтерполяції.

Перед початком інтерполяції в ці регістри завантажуються відстань до кінцевої точки. При кожній видачі імпульсу на координату відбувається декремент регістру відстані даної координати.

У момент обнулення регістрів формуються команди Стоп Х, Стоп У, які зупиняють процес інтерполяції по відповідній координаті.

5. Для більшої наочності представимо роботу інтерполятора у вигляді таблиці 12.1:

Таблиця 12.1

Кількість імпульсів $t_{роб}$	Pr X	Pr Y	F_{ij}	Cr X	Cr Y	N_x	N_y
0	4	7	0	4	7	-	-
1	4	7	-7	3	7	1	-
2	4	7	-3	3	6	-	1
3	4	7	+1	3	5	-	1
4	4	7	-6	2	5	1	-
5	4	7	-2	2	4	-	1
6	4	7	+2	2	3	-	1
7	4	7	-5	1	3	1	-
8	4	7	-1	1	2	-	1
9	4	7	3	1	1	-	1
10	4	7	-4	0 Стоп Х	1	1	-
11	4	7	0		0 Стоп У	-	-

6. Часові діаграми й оцінка рівномірності проходження імпульсів

На основі отриманої таблиці побудуємо тимчасову діаграму проходження імпульсів (рис 12.3):

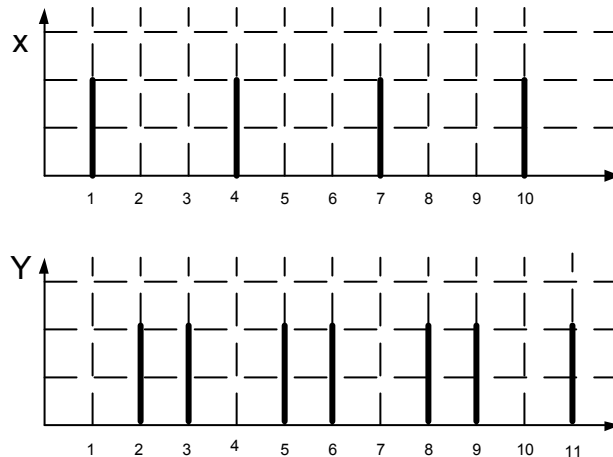


Рис. 12.3 – Часова діаграма надходження імпульсів

Ступінь нерівномірності видачі імпульсів визначає динамічну похибку інтерполяції. Чим більш рівномірно видаються імпульси, тим менше ця помилка. Аналіз діаграми показує, що по координаті X імпульси видаються рівномірно, а по координаті Y частота видачі імпульсів непостійна й змінюється в 2 рази.

Існують спеціальні методи для підвищення рівномірності видачі імпульсів на координатні приводи подач.

Висновки:

1. У методі ОФ 1-й імпульс видається на координату X , а другий на координату Y незалежно від характеру траєкторії.
2. Оціночна функція по закінченню інтерполяції дорівнює 0.
3. Оцінка рівномірності видачі імпульсів проводиться за допомогою часових діаграм.

Реалізація лінійного інтерполятора по методу ОФ на мікроконтролері

Програма, що реалізує метод ОФ на мікроконтролері має вигляд:

; Програма лінійної інтерполяції за методом оціночної функції

; Linear interpolation OF method

.include "m8535def.inc"

.def tmp=r16

.def reg_OF=r17

.def reg_X=r18

.def reg_Y=r19

.def reg_distance_X=r20

.def reg_distance_Y=r21

.def counter_cycle=r22

.def reg_position_X=r23

.def reg_position_Y=r24

.equ Xk=4

.equ Yk=7

.org 0

rjmp Reset

.org 9

rjmp t0_ovf

Reset: in tmp, low(RAMEND)
out SPL, tmp
in tmp, high(RAMEND)
out SPH,tmp
clr reg_OF
clr reg_position_X
clr reg_position_Y
clr counter_cycle
ldi reg_X, Xk
ldi reg_Y, Yk
ldi reg_distance_X, Xk
ldi reg_distance_Y, Yk
ser tmp
out DDRB,tmp
out DDRD,tmp
cli
ldi tmp,1
out TIMSK, tmp
ldi tmp,1
out TCCR0, tmp
sei

main: rjmp main

t0_ovf: in tmp, SREG

```

        push tmp
        tst reg_OF
        brmi step_Y
step_X:
        tst reg_distance_X
        breq end
        sub reg_OF, reg_Y
        inc reg_position_X
        out PortB, reg_position_X
        dec reg_distance_X
        inc counter_cycle
        rjmp end
step_Y:
        tst reg_distance_Y
        breq end
        add reg_OF, reg_X
        inc reg_position_Y
        out PortD, reg_position_Y
        dec reg_distance_Y
        inc counter_cycle
end:
        pop tmp
        out SREG, tmp
        reti

```

Контрольні питання

1. Поясніть, які значення заносяться в регістри інтерполятора перед початком інтерполяції.
2. На які координати видаються перші два імпульси?
3. Як оцінюється нерівномірність видачі імпульсів?
4. Вкажіть в програмі команду запуску таймера на рахунок.
5. Наведіть команди, які встановлюють вказівник стеку.

КРУГОВА ІНТЕРПОЛЯЦІЯ ЗА МЕТОДОМ ОЦІНОЧНОЇ ФУНКЦІЇ

У ЧПК криволінійні відрізки траєкторії апроксимуються дугами кіл. Для ведення інструмента по дузі кола необхідно в режимі реального часу із частотою квантування 2-10 кГц видавати на привод подачі координати поточних точок.

Ці завдання реалізує круговий інтерполятор, причому частота квантування задається у вигляді робочої частоти від БЗШ.

Принцип кругової інтерполяції за методом ОФ

Заданий відрізок дуги характеризується чотирма параметрами:

1. Координати початкової точки X_n, Y_n .
2. Координати кінцевої точки X_k, Y_k .
3. Радіус дуги.
4. Напрямок інтерполяції.

Розглянемо випадок кругової інтерполяції в 1-м квадранті. Центр дуги кола завжди розташований на початку координат (рис. 13.1).

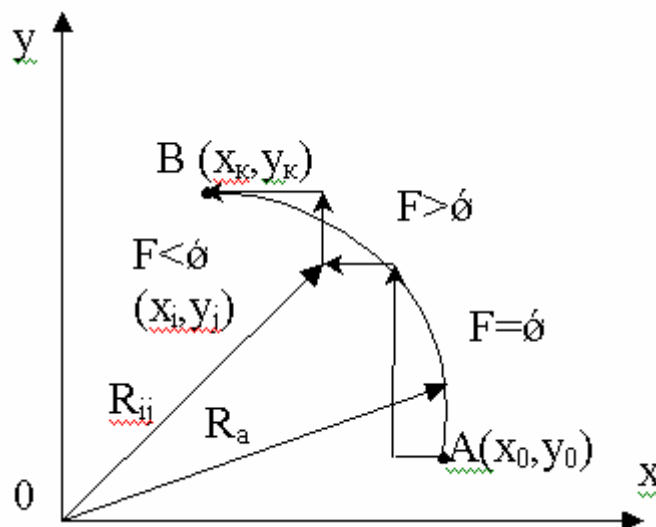


Рис. 13.1 – Траєкторії: задана та відпрацьована

Ступінь відхилення поточної траєкторії від заданої можна оцінити по різниці

$$R_{ij} - R = \Delta. \quad (1)$$

Ця формула відображує лише ідею методу ОФ.

Для практичних цілей ОФ визначають як різницю квадратів радіусів

$$F_{ij} = R_{ij}^2 - R^2 = x_i^2 + y_j^2 - x_H^2 - y_H^2 \quad (2)$$

Оскільки початкова й кінцева точки лежать на заданій траєкторії, то в цих точках оціночна функція рівна 0.

Аналізуючи наведений малюнок, можна зробити висновок, що для руху вздовж заданої траєкторії перший крок треба робити по координаті X у негативному напрямку. Зробивши цей крок, ми попадаємо в область $F < 0$, отже, другий і наступні кроки необхідно робити по координаті Y у позитивному напрямку до тих пір, поки не перейдемо в область $F \geq 0$.

Для визначення руху вздовж заданого відрізка необхідно знати тільки знак ОФ, але для визначення знака необхідно обчислити й саму ОФ.

Обчислення ОФ за виразом (2) малоефективне, тому розроблений швидкий алгоритм обчислення значення ОФ у новій точці на підставі значення ОФ у попередній точці й напрямку кроку. Цей алгоритм придатний тільки для методів з по черговим кроком по координатах. Таким є метод ОФ.

Алгоритм спрощеного обчислення ОФ

При інтерполяції дуги кола розташованої в 1-му квадранті проти годинникової стрілки при кроці по осі X ця координата зменшується на 1. Крок по координаті Y збільшує цю координату на 1.

Значення ОФ у попередній точці (старе значення):

$$F_{ij} = x_i^2 + y_j^2 - R_a^2 - \text{старе значення ОФ.}$$

Значення ОФ у новій точці при кроці по осі X

$$F_{i+1,j} = x_{i+1}^2 + y_j^2 - R^2 = (x_i - 1)^2 + y_j^2 - R^2 = x_i^2 - 2x_i + 1 + y_j^2 - R^2 = F_{ij} - 2x_i + 1 \quad (3)$$

Значення ОФ у новій точці при кроці по Y

$$F_{i,j+1} = x_i^2 + y_{j+1}^2 - R^2 = x_i^2 + (y_j + 1)^2 - R^2 = x_i^2 + y_j^2 + 2y_j + 1 - R^2 = F_{ij} + 2y_j + 1 \quad (4)$$

Вирази (3) і (4) містять додавання, віднімання й множення на 2, що значно простіше вихідних виразів. Можна припустити, що саме цей простий алгоритм дозволив практично впровадити метод ОФ у реальні СПК на початку розвитку процесорної техніки.

Функціональна схема кругового інтерполятора по методу ОФ

Функціональна схема кругового інтерполятора по методу ОФ показана на рисунку 13.2. В цілому вона подібна до схеми лінійного інтерполятора, але відрізняється алгоритмом обчислення оціночної функції.

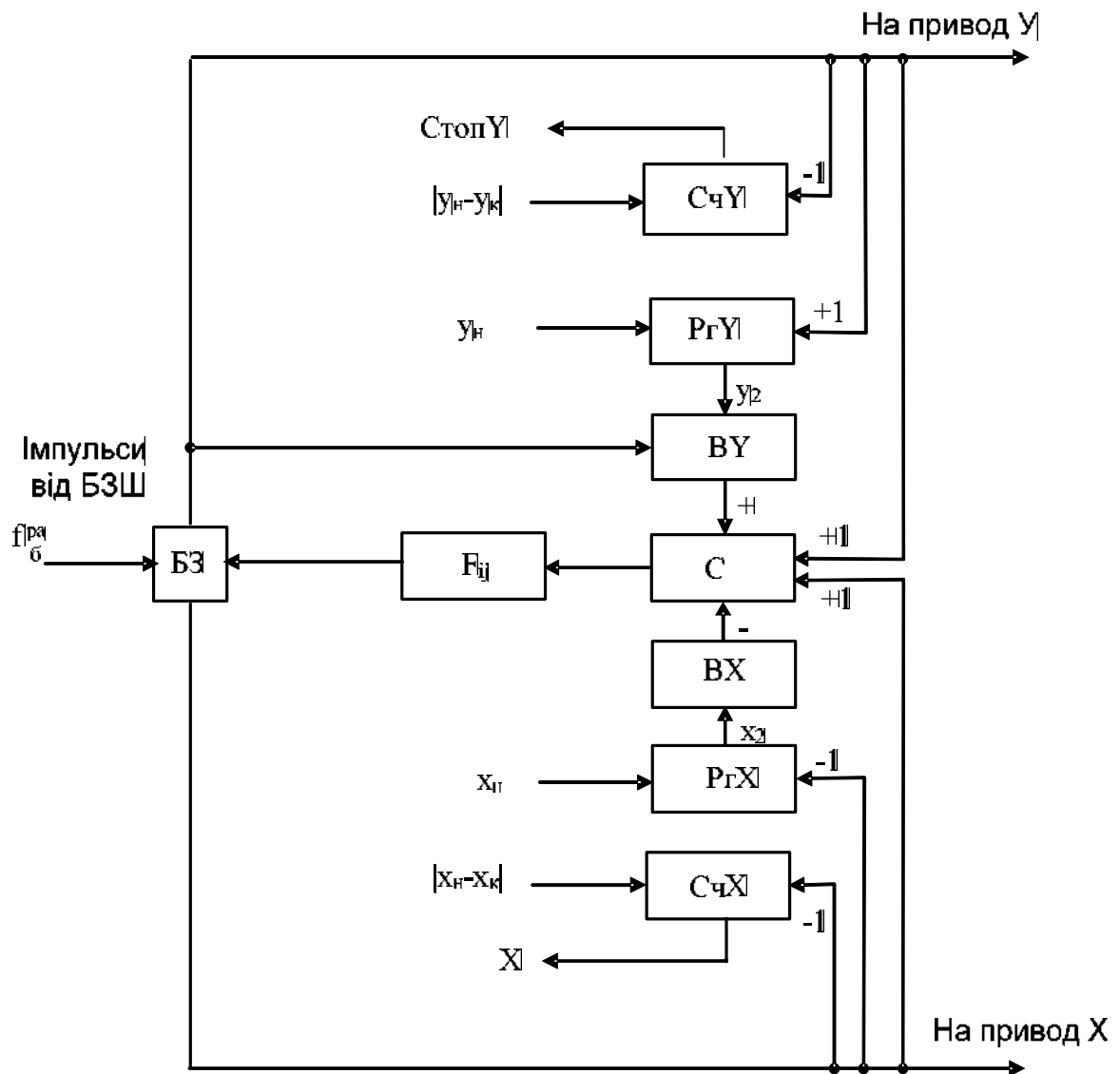
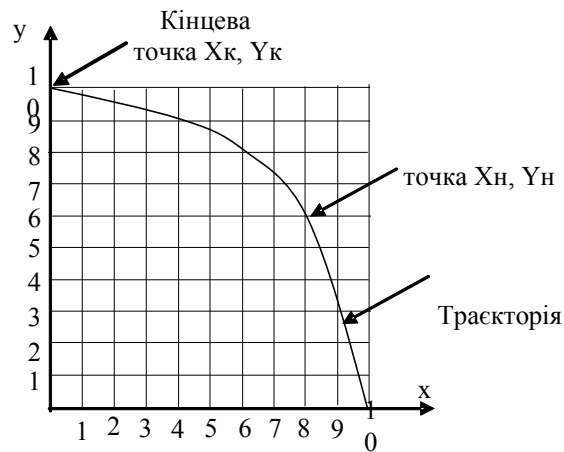


Рис. 13.2 – Функціональна схема кругового інтерполятора за методом ОФ

У вихідному стані регістр ОФ інтерполятора містить нуль. У регістри-лічильники відстані заносяться модулі відстаней до кінцевої точки по кожній з координат. В регістри координат PrX та PrY занесені координати початкової точки. Оскільки у вихідній точці $F=0$, то перший імпульс від БЗШ через блок знаку надходить на координату X . Одночасно відкривається вентиль BX і від значення 0 віднімається значення $(2x_j + 1)$. Таким чином, оціночна функція F стає негативною й другий крок виконується по координаті Y , (аналогічно тому, як це відбувається при лінійній інтерполяції). Надалі кроки чергуються залежно від нахилу дуги окружності.



Контрольні питання

1. Поясніть відмінності кругового інтерполятора від лінійного?
2. На які координати видаються перші два імпульси?
3. Які значення заносяться в регістри інтерполятора перед початком інтерполяції?
4. Як розраховуються нові значення оціночної функції?
5. Що є умовою закінчення процесу інтерполяції?

ПРОГРАМУВАННЯ КРУГОВОГО ІНТЕРПОЛЯТОРА ЗА МЕТОДОМ ОЦІНОЧНОЇ ФУНКЦІЇ

Розглянемо приклад програмної реалізації кругового інтерполятора за методом ОФ.

Вихідні дані:

Координати початкової та кінцевої точок:

$X_{\text{п}} = 8, Y_{\text{п}} = 6.$

$X_{\text{к}} = 0, Y_{\text{к}} = 10.$

Напрямок інтерполяції – проти годинникової стрілки.

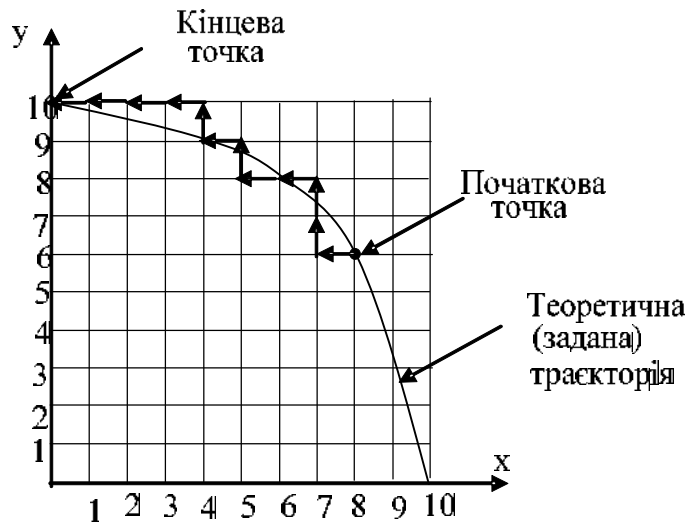


Рис. 14.1 – Задана лінійна траєкторія та її дискретне наближення

Нові значення оцінної функції F_{ij} при круговій інтерполяції:

1) При кроці по осі X нове значення ОФ $F_{i+1,j} = F_{ij} - 2x_i + 1, \quad (1)$

2) При кроці по осі Y нове значення ОФ $F_{i,j+1} = F_{ij} - 2y_i + 1, \quad (2)$

3) Оскільки в початковій точці $F_{ij} = 0$, то перший крок робимо по осі X у

від'ємному напрямку.

Результати інтерполяції представимо у вигляді таблиці (таблиця 14.1).

Таблиця 14.1

Кількість імпульсів $f_{роб}$	$P_T X$	$P_T Y$	F_{ij}	Ліч X	Ліч Y	N_x	N_y
-	8	6	0	8	4	-	-
1	7	6	-15	7	4	1	-
2	7	7	-2	7	3	-	1
3	7	8	13	7	2	-	1
4	6	8	0	6	2	1	-
5	5	8	-11	5	2	1	-
6	5	9	6	5	1	-	1
7	4	9	-3	4	1	1	-
8	4	10	16	4	0	-	1
9	3	10	9	3	0	1	-
10	2	10	4	2	0	1	-
11	1	10	1	1	0	1	-
12	0	10	0	0	0	1	-

Програмна реалізація кругового інтерполятора за методом ОФ на мікроконтролері

Структурно програма складається з 2-х частин:

Перша частина реалізує БЗШ на таймері 0. Таймер настраюється так, щоб його переповнення відбувалися із частотою від 2 до 10 кГц, що відповідає значенням частоти квантування реального інтерполятора.

Основна програма інтерполяції виконана у вигляді підпрограми, що викликається по переповненню таймера 0. Алгоритм основної програми наведено на рис. 14.1.



Рис. 14.1 – Алгоритм роботи основної програми

Далі розглянемо власне програму інтерполяції як основний елемент вищенаведеної структури.

У програмі інтерполяції введено два накопичувальні регістри по координатах X і Y, у яких підсумуються імпульси по цих координатах. У такий спосіб зміст накопичувального регістру являє собою завдання для позиційного сервоприводу подачі по даній координаті.

; Circular interpolation OF method

```
.include "m8535def.inc"
```

```
.def tmp=r16
```

```
.def reg_OF=r17
```

```
.def reg_X=r18
```

```
.def reg_Y=r19
```

```
.def reg_distance_X=r20
```

```
.def reg_distance_Y=r21
```

```

.def step_counter=r22
.def reg_position_X=r23
.def reg_position_Y=r24
.equ Xn=8
.equ Yn=6
.equ Xk=0
.equ Yk=10
.org 0
        rjmp Reset

.org 9
        rjmp t0_ovf
Reset:   ldi tmp, low(RAMEND)
        out SPL, tmp
        ldi tmp, high(RAMEND)
        out SPH,tmp

        clr reg_OF
        clr reg_position_X
        clr reg_position_Y
        clr step_counter
        ldi reg_X, Xk
        ldi reg_Y,Yk
        ldi reg_distance_X, abs(Xk-Xn)
        ldi reg_distance_Y, abs(Yk-Yn)
        ser tmp
        out DDRB,tmp
        out DDRD,tmp
        cli
        ldi tmp,1

```

```

        out TIMSK, tmp
        ldi tmp,1
        out TCCR0, tmp
        sei
main:    rjmp main

t0_ovf:

        in tmp, SREG
        push tmp
        tst reg_OF
        brmi step_Y

step_X:

        tst reg_distance_X
        breq end
        sub reg_OF, reg_X
        sub reg_OF, reg_X
        inc reg_OF
        dec reg_position_X
        out PortB, reg_position_X
        dec reg_distance_X
        inc step_counter
        rjmp end

step_Y:

        tst reg_distance_Y
        breq end
        add reg_OF, reg_Y
        add reg_OF, reg_Y
        inc reg_OF
        inc reg_position_Y

```

```
        out PortD, reg_position_Y
        dec reg_distance_Y
        inc step_counter
end:     pop tmp
        out SREG, tmp
        reti
```

Контрольні питання

1. Як визначаються нові значення оціночної функції?
2. Поясніть функцію регістрів-лічильників.
3. Вкажіть команди, які заносять початкові значення в регістри інтерполятора.
4. Вкажіть команди, які визначають закінчення інтерполяції.
5. Які вектори переривань задіяні в програмі?

ЛІНІЙНА ІНТЕРПОЛЯЦІЯ ЗА МЕТОДОМ ЦИФРОВИХ ДИФЕРЕНЦІАЛЬНИХ АНАЛІЗАТОРІВ

У методі цифрових диференціальних аналізаторів (ЦДА) траєкторія задається у вигляді диференціальних рівнянь, а не алгебраїчних, як у методі ОФ.

Для відновлення траєкторії необхідно розв'язати систему диференціальних рівнянь. В англomовній технічній літературі пристрій для розв'язування диференціальних рівнянь називається диференціальним аналізатором. Звідси й назва методу.

Для реалізації процесу інтерполяції за методу ЦДА рівняння траєкторії представляються в параметричному виді

$$x = \varphi_1(U) \quad (1)$$

$$y = \varphi_2(U) \quad (2)$$

$$z = \varphi_3(U) \quad (3)$$

У цих рівняннях параметр U визначається, як деяка функція часу:

$$U = f(t).$$

Обчислення поточних координат по рівняннях (1), (2), (3) проводиться по методу чисельного інтегрування. Розглянемо варіант інтерполяції по 2-м координатам (робота на площині в координатах X-Y).

Поточні координати можна визначити в такий спосіб:

$$x(U) \approx x^*(U) = \sum_{i=1}^N \varphi_{1i}' \Delta U \quad (3)$$

$$y(U) \approx y^*(U) = \sum_{j=1}^N \varphi_{1j}' \Delta U \quad (4)$$

$$\text{де} \quad \varphi_1'(U) = \frac{d\varphi_1'(U)}{dt}$$
$$\varphi_2'(U) = \frac{d\varphi_2'(U)}{dt}$$

Обчислення проводяться при нульових початкових умовах, тому початкові значення в рівняннях (3), (4) відсутні. Символи без “*” позначають безперервні координати, а із символом “*” – їх дискретне наближення. У реальному пристрої обчислюються $x^*(U)$, $y^*(U)$ в реальному часі з дискретністю ΔU . Період дискретизації ΔU за часом є жорстко фіксованим параметром, задається таймером і є постійною величиною.

$$U = \frac{t}{T} \quad \text{при } V = \text{const} \quad \Delta U = \frac{\Delta t}{T}$$

За таких умов рівняння (3), (4) приймають вигляд:

$$x^*(U) = \Delta U \sum_{i=1}^N \varphi_{1i} \quad (5)$$

$$y^*(U) = \Delta U \sum_{j=1}^N \varphi_{1j} \quad (6)$$

Отримані рівняння (5), (6) представляють основну ідею методу ЦДА, що полягає в тому, що при дотриманні вищевказаних умов поточне значення координат можна визначити шляхом підсумовування похідних.

Ядро інтерполятора по методу ЦДА являє собою цифровий інтегратор, виконаний найчастіше у вигляді суматора-накопичувача, де накопичується сума похідних, що представляє поточну координату.

У загальному випадку функціональна схема інтерполяторів по методу ЦДА для однієї координати має вигляд (рис. 15.1):

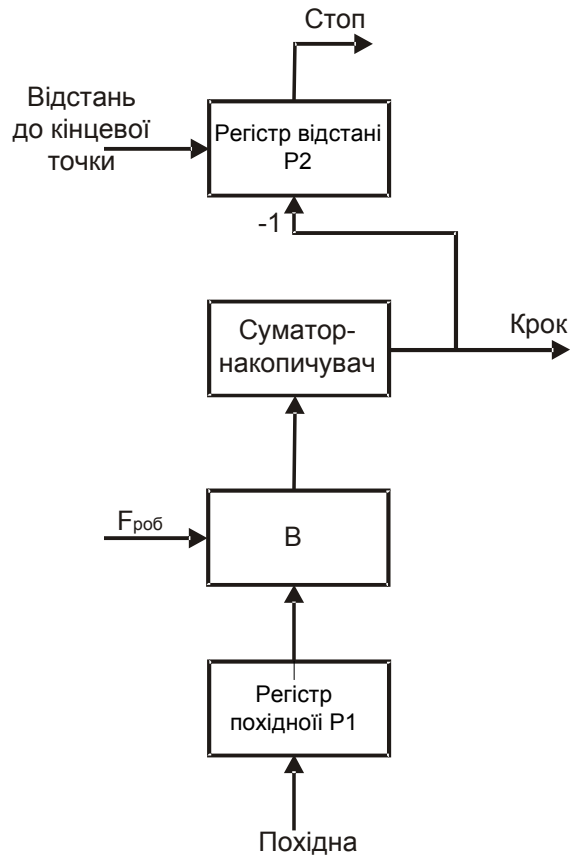
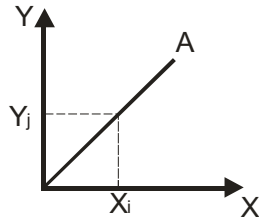


Рис. 15.1 – Функціональна схема інтерполяторів по методу ЦДА
для однієї координати

Робота схеми: перед початком роботи суматор-накопичувач містить нуль, в реєстр похідної записане значення по даній константі в початковій точці траєкторії. В реєстр відстані P2 занесений модуль відстані до кінцевої точки по даній координаті. Початок інтерполяції здійснюється шляхом подачі імпульсів $f_{роб}$ на вентиль В. Кожний такий імпульс відкриває вентиль В і вміст реєстру похідної додається до вмісту суматора-накопичувача. Після якоїсь кількості підсумовувань, останній переповнюється. У момент переповнення на привод подачі даної координати подається імпульс “Крок”.

Лінійна інтерполяція за методом ЦДА

У цьому випадку метод інтерполяції використовується для обчислення проміжних координат відрізка прямої, проведеного з початку координат у кінцеву точку.



Для випадку лінійної інтерполяції параметр U визначається як:

$$U = \frac{x}{x_k} = \frac{y}{y_k} \quad (7)$$

Звідки параметричні рівняння руху:

$$x = x_k \cdot U \quad (8)$$

$$y = y_k \cdot U \quad (9)$$

Найбільш простим і практично важливим випадком є випадок руху з постійною швидкістю U .

$$\begin{aligned} \varphi_1'(U) &= \frac{d(x_k U)}{dU} = x_k \\ \varphi_2'(U) &= \frac{d(y_k U)}{dU} = y_k \end{aligned}$$

Вирази показують, що при лінійній інтерполяції по методу ЦДА похідні для кожної константи постійні й рівні відповідно x_k й y_k .

Отже, 2-х координатний лінійний інтерполятор по методу ЦДА може бути побудований автономно для кожної координати за умови синхронної роботи від одного джерела робочої частоти (рис. 15.2).

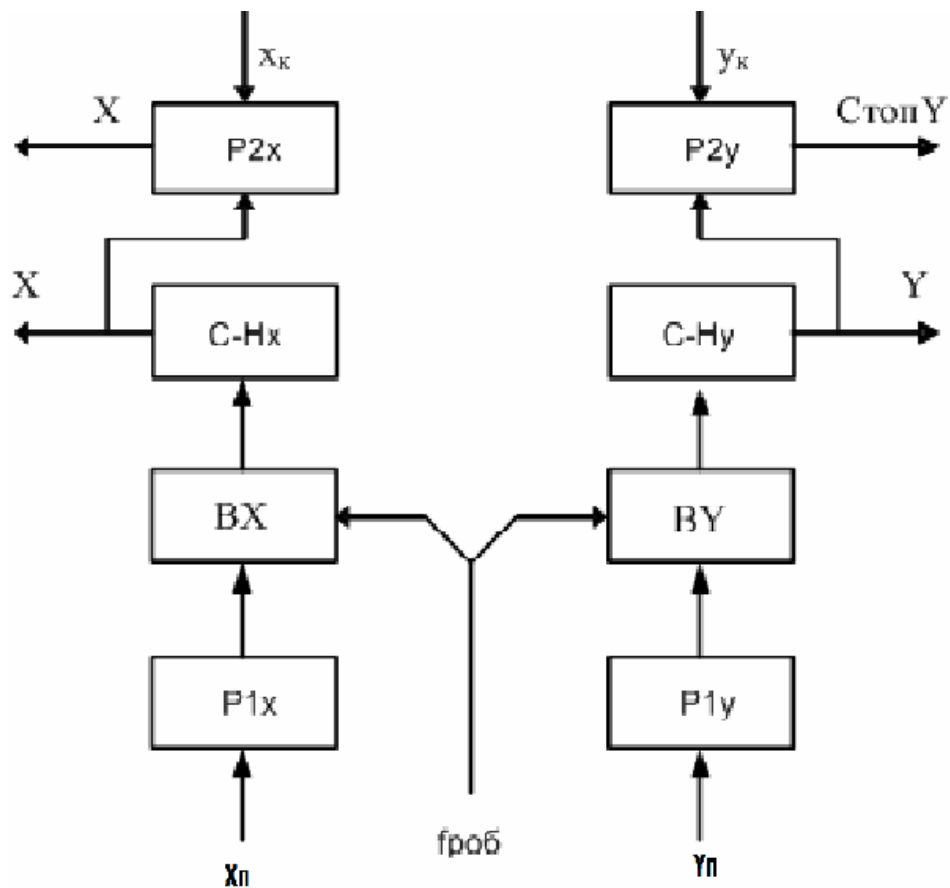


Рис. 15.2 – Функціональна схема інтерполятора за методом ЦДА для двох координат

Робота інтерполятора

Два цифрові інтегратори по кожній координаті працюють синхронно від одного джерела робочої частоти. У зв'язку з тим, що не кожний імпульс робочої частоти проходить на вихід можливі випадки, коли моменти переповнення 2-х суматорів-накопичувачів збігаються. В цьому випадку відбувається видача імпульсів на обидві координати. Отже, елементарний крок буде зроблений на обох координатах, тобто під кутом 45° . Звідси можна зробити висновок, що метод ЦДА забезпечує більш точні характеристики для ділянок траєкторії з кутами нахилу, близькими до 45° (рис. 15.3).

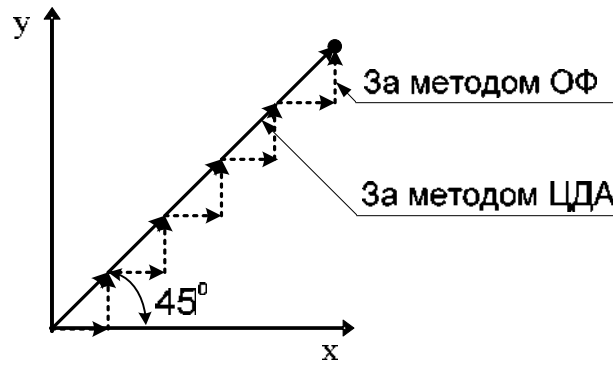


Рис. 15.3 – Траєкторії інтерполяції за методами ОФ та ЦДА

Контрольні питання

1. Поясніть відмінності методу ЦДА від методу ОФ.
2. Чому дорівнюють похідні по кожній координаті при лінійній інтерполяції за методом ЦДА?
3. Поясніть роботу цифрового інтегратора.
4. Як працює регістр-накопичувач?
5. За якої умови відбувається видача імпульсу «Крок»?

ПРОГРАМНА РЕАЛІЗАЦІЯ ЛІНІЙНОГО ІНТЕРПОЛЯТОРА ЗА МЕТОДОМ ЦДА

Розглянемо процес лінійної інтерполяції відрізка прямої, проведеного з початку координат у точку з координатами (4,7), як показано на рисунку 16.1. Ємність регістра-накопичувача приймемо рівною 10.



$$x_n = 0, y_n = 0$$

$$x_k = 4, y_k = 7$$

$$R_0 = 10$$

Рис. 16.1 – Приклад лінійної траєкторії

Результати інтерполяції представимо у вигляді таблиці (Таблиця 16.1)

Таблиця 16.1

Кількість імпульсів $t_{роб}$	$P1_x$	$C-N_x$	N_x	$P2_x$	$P1_y$	$C-N_y$	N_y	$P2_y$
0	4	0	—	4	7	0	—	7
1	4	4	—	4	7	7	—	7
2	4	8	—	4	7	4	1	6
3	4	2	1	3	7	1	1	5
4	4	6	—	3	7	8	—	5
5	4	0	1	2	7	5	1	4
6	4	4	—	2	7	2	1	3

7	4	8	–	2	7	9	–	3
8	4	2	1	1	7	6	1	2
9	4	6	–	1	7	3	1	1
10	4	0	1	0 Стоп Х	7	0	1	0 Стоп У

За результатами таблиці можуть бути побудовані часові діаграми (рис. 16.2), які характеризують важливий параметр – рівномірність видачі інтерполятором вихідних імпульсів.

По координаті Х період вихідних імпульсів змінюється в 1,5 рази, а по координаті У в 2 рази. Вважається, що чим більш рівномірно надходять імпульси від інтерполятора, тим точніше буде відпрацювання заданої траєкторії.

Існують спеціальні методи поліпшення рівномірності проходження вихідних імпульсів, що дозволяють поліпшити точність відпрацювання траєкторії.

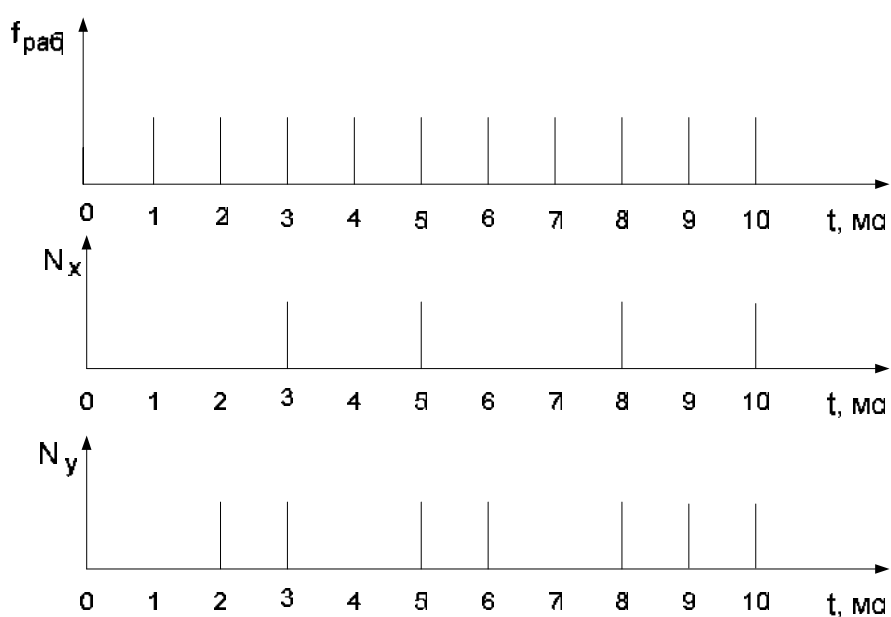


Рис. 16.2 – Часові діаграми рівномірності проходження вихідних імпульсів

На основі раніше викладеного алгоритму лінійної інтерполяції за методом ЦДА складемо програму на мові асемблер.

; Linear interpolation by CDA method

.include "m8535def.inc"

.def tmp=r16

.def reg_summator_X=r17

.def reg_summator_Y=r25

.def reg_der_X=r18

.def reg_der_Y=r19

.def reg_distance_X=r20

.def reg_distance_Y=r21

.def step_counter=r22

.def reg_position_X=r23

.def reg_position_Y=r24

.equ Xn=0

.equ Yn=0

.equ Xk=4

.equ Yk=7

.equ R=10

.org 0

 rjmp Reset

.org 9

 rjmp t0_ovf

Reset: ldi tmp, low(RAMEND)

 out SPL, tmp

 ldi tmp, high(RAMEND)

out SPH,tmp

clr reg_position_X

clr reg_position_Y

clr step_counter

clr reg_summator_X

clr reg_summator_y

ldi reg_distance_X, abs(Xk-Xn)

ldi reg_distance_Y, abs(Yk-Yn)

ldi reg_der_X, Xk

ldi reg_der_Y, Yk

ser tmp

out DDRB,tmp

out DDRD,tmp

cli

ldi tmp,1

out TIMSK, tmp

ldi tmp,1

out TCCR0, tmp

sei

main: rjmp main

t0_ovf:

in tmp, SREG

push tmp

step_X:

```

tst reg_distance_X
breq step_Y
add reg_summator_X, reg_der_X
cpi reg_summator_X, R

brmi step_Y
subi reg_summator_X, R      ; залишок
inc reg_position_X
out PortB, reg_position_X
dec reg_distance_X
inc step_counter
rjmp step_Y

```

step_Y:

```

tst reg_distance_Y
breq end
add reg_summator_Y, reg_der_Y
cpi reg_summator_Y, R
brmi end
subi reg_summator_Y, R      ; залишок
inc reg_position_Y
out PortD, reg_position_Y
dec reg_distance_Y
inc step_counter

```

end:

```

pop tmp
out SREG, tmp
reti

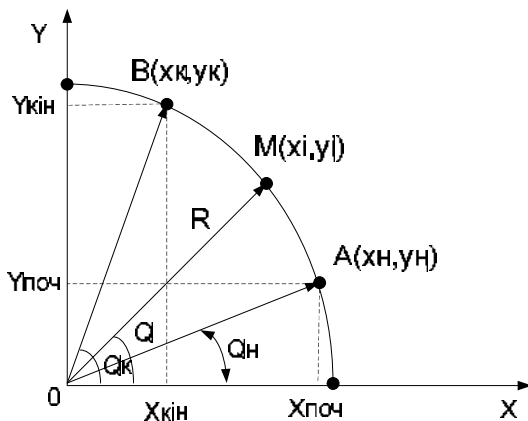
```

Контрольні питання

1. За яких умов можливі одночасні кроки по двох координатах?
2. Як будуються часові діаграми вихідних імпульсів?
3. Поясніть програмну реалізацію регістра-накопичувача.
4. Вкажіть команду, яка дозволяє переривання за переповненням таймера.
5. Наведіть команди, які модифікують регістри-лічильники.

КРУГОВА ІНТЕРПОЛЯЦІЯ ЗА МЕТОДОМ ЦИФРОВИХ ДИФЕРЕНЦІАЛЬНИХ АНАЛІЗАТОРІВ

Розглянемо процес кругової інтерполяції за методом ЦДА в 1-м квадранті проти годинникової стрілки (рис. 17.1).



А, В – початкова й кінцева точки відрізка дуги;

М – поточна точка;

Координати початкової точки $(x_п, y_п)$, кінцевої точки $(x_к, y_к)$; Кут нахилу відрізка до осі X – Q;

Рис. 17.1- Кругова інтерполяція

за методом ЦДА

Параметри рівняння для дуги із центром на початку координат мають вигляд:

$$x(U) = \sum_{i=1}^N \phi'_{1i} \Delta U \quad (1)$$

$$y(U) \approx y^*(U) = \sum_{j=1}^N \phi'_{1j} \Delta U \quad (2)$$

Параметри рівняння для дуги із центром на початку координат мають вигляд:

$$x = \varphi_1(\theta) = R \cos(\theta)$$

$$y = \varphi_2(\theta) = R \sin(\theta)$$

$$\theta_n \leq \theta \leq \theta_k$$

Значення похідних:

$$x' = -R \sin(\theta)$$

$$y' = R \cos(\theta)$$

$$x' = -y$$

$$y' = x$$

Висновок: при круговій інтерполяції по методу ЦДА похідна по даній координаті дорівнює поточному значенню іншої координати. В цьому полягає відмінність від випадку лінійної інтерполяції, коли похідні були незмінними. Очевидно, що при апаратній або програмній реалізації кругового інтерполятора необхідно передбачити засоби для модифікації вмісту регістрів похідних після кожного виконаного кроку.

Отже, функціональну схему кругового інтерполятора по методу ЦДА можна представити у вигляді показаному на рисунку 15.2.

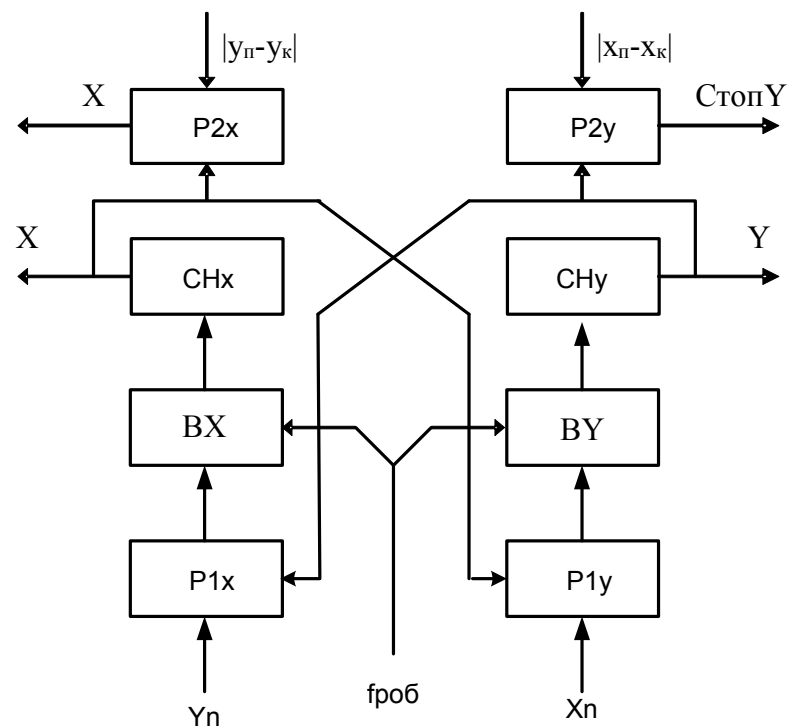


Рис. 17.2 – Функціональна схема кругового інтерполятора за методом ЦДА

Робота інтерполятора

При відновленні дуги кола, розташованої в першому квадранті із центром на початку координат, при русі проти годинникової стрілки в регістр похідної завантажуються значення похідної, чисельно рівне початковому значенню іншої координати. У процесі обчислення імпульси з виходів регістрів накопичувачів модифікують вміст регістрів похідних. В регістр похідної по координаті X додається одиниця, а від регістру похідної по координаті Y

одиниця віднімається. Цією простою операцією проводиться обчислення нових значень похідних.

В усі регістри записуються модулі чисел. Залежно від квадранта й напрямку обходу формуються додаткові знакові сигнали, що використовуються для правильної організації обчислень і керування приводами подач.

Розглянемо процес кругової інтерполяції за методом ЦДА для випадку формування траєкторії, що зображена на рисунку 17.3 з такими вихідними даними:

Координати початкової точки $x_{\text{п}} = 8, y_{\text{п}} = 5$;

Координати кінцевої точки $x_{\text{к}} = 0, y_{\text{к}} = 10$;

Ємність регістра-накопичувача $R_0 = 10$.

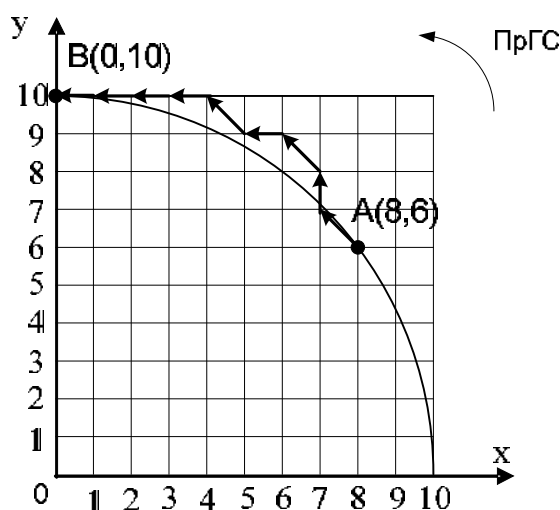


Рис. 17.3 – Траєкторії руху: задана і відпрацьована за методом ЦДА

Результати інтерполяції занесемо в таблицю 17.1.

Таблиця 17.1

Кількість імпульсів $t_{\text{роб}}$	$P1_x$	$C-N_x$	N_x	$P2_x$	$P1_y$	$C-N_y$	N_y	$P2_y$
0	6	0	—	8	8	0	—	4
1	6	6	—	8	8	8	—	4

2	6→7	2	1	7	8→7	6	1	3
3	7→8	9	–	7	7	3	1	2
4	8→9	7	1	6	7→6	0	1	1
5	9	6	1	5	6→5	6	–	1
6	9→10	5	1	4	5→4	2	1	0
								СтопУ
7	10	5	1	3				
8	10	5	1	2				
9	10	5	1	1				
10	10	5	1	0				
				СтопХ				

Контрольні питання

1. Як визначаються похідні при круговій інтерполяції за методом ЦДА?
2. Поясніть відмінності побудови кругового інтерполятора від лінійного в методі ЦДА.
3. Вкажіть, в яких напрямках змінюються координати при круговій інтерполяції за методом ЦДА.
4. Як провадиться обчислення нових значень похідних?
5. Яким чином закінчується процес інтерполяції?

ТЕНДЕНЦІЇ РОЗВИТКУ СУЧАСНИХ СИСТЕМ ПРОГРАМНОГО КЕРУВАННЯ

Сучасні моделі світових виробників систем ЧПК мають двокомп'ютерну архітектуру, управляють 16 і більше осями на різних каналах керування, реалізують інтерполяцію з нанометричною точністю та високошвидкісну обробку.

Ціна таких систем у європейських виробників - від 15 000€ при мінімальній базовій конфігурації й може перевищувати 60 000€ залежно від комплектації. Системи такого класу в СНГ не виробляються в силу їхньої складності, хоча потреба в них постійно росте. Системи ЧПК класу Hi-End попадають під технології подвійного призначення, тому ряд країн-виробників, як Японія, відмовляють у продажах до певних країн, а щоб придбати, наприклад, системи ЧПК Sinumeric 840D/Di фірми Siemens, потрібно отримати дозвіл уряду ФРГ.

На даному етапі вартість розробки програмного забезпечення становить більше 80% від усього процесу створення системи ЧПК. Основна мета виробників систем ЧПК - зберегти вкладені інвестиції, керуючись нею, виробники систем ЧПК воліють розвивати раніше створені системи шляхом розширення їх функціональності та (або) зміни ядра системи керування. Цим, власне, пояснюється той факт, що останнім часом на ринку не з'являються революційно нові технічні рішення, а системи керування планомірно еволюціонують.

1. Відмінні можливості сучасних систем ЧПК

Аналіз сучасних систем ЧПК дозволяє виділити наступні стратегічні напрямки й технології, де ведеться дуже активна розробка:

- відкритість – надання верстатобудівнику та кінцевому користувачеві можливості реалізовувати власні технології та вбудовувати прикладне програмне забезпечення в систему ЧПК. Це надає можливість застосовувати

системи ЧПК не лише для металообробних верстатів, а також в інших галузях, наприклад, для камер вакуумного зварювання тугоплавких матеріалів;

- багатоканальність – можливість паралельно виконувати кілька керуючих програм на одній системі ЧПК. Найчастіше ця функція використовується для багатошпіндельної обробки, або для сумісного керування верстатом та завантажником.
- високошвидкісна обробка – для системи ЧПК це означає короткий такт інтерполяції та високі швидкості обробки кадру керуючої програми;
- наноінтерполяція – інтерполяція с дискретністю обчислень, що обчислюється в нанометрах; практично всі сучасні системи ЧПК мають таку функціональність, але FANUC в якості маркетингового ходу представляє її як зарезервовану марку;
- Look-ahead – алгоритм попереднього перегляду кадрів, що дозволяє обробляти контур з максимально можливою подачею і завчасно знижувати швидкість в критичних ситуаціях для запобігання колізій.
- сплайнова інтерполяція – реалізація алгоритмів кубічного сплайну, Акіма-сплайну, NURBS-сплайну, поліномів та інших інтерполяцій в комбінації з алгоритмом попереднього перегляду кадрів (look-ahead);
- компресія кадрів – згладжування контуру обробки в керуючих програмах шляхом конвертування лінійних кадрів в NURBS-контури в процесі обробки керуючих програм;
- віддалене управління – передбачає побудову віддалених терміналів і реалізацію функцій віддаленої діагностики и настройки системи ЧПК;
- мережеві функції – дозволяють інтегрувати систему ЧПК в процес управління виробництвом та устаткуванням. Дозволяють проводити облік виробленої продукції, контролювати час простою, відслідковувати планові ремонти та техобслуговування устаткування;

- моделювання процесу обробки – 3D-моделлювання процесу різання для попереднього перегляду результатів виконання керуючих програм, виявлення колізій і скорочення часу обробки;
- діагностичні функції – логічний аналізатор (Logic Analyzer) вхідних/вихідних сигналів для тестування електроавтоматики, осцилограф для настройки приводів подач, аналізатор частотних діаграм, тест кола (Circle Test) та ін.;
- мова програмування високого рівня та інструментарій розробки і налагодження керуючих програм на мові високого рівня. Цю функціональність застосовують при розробці стандартних циклів та групових технологій.
- додатки цехового програмування – CAD/CAM системи цехового рівня, які дозволяють створювати керуючі програми безпосередньо на самій системі ЧПК, під час обробки іншої деталі. Зазвичай, ці додатки орієнтовані на 2,5D обробку;
- застосування технології штучного інтелекту для адаптивного керування та прогнозування зносу інструменту.

На сьогодні склалася стійка тенденція до уніфікації устаткування шляхом реалізації все більшого числа стандартів і протоколів зв'язку для інтеграції в єдині системи, як OPC, TCP/IP, SOAP та ін. Але, з іншого боку, фірми прагнуть виставити на ринок комплексні технологічні рішення «під ключ», а не розрізнені компоненти.

2. Сучасні розробки по системах ЧПК

Світовий лідер в області систем ЧПК - фірма FANUC представила тридцяті серію своїх систем ЧПК (Series 30i/31i/32i-MODEL A), яка має такі характеристики: кількість каналів - 10, кількість осей – 32, кількість шпинделів - 8. Максимальна кількість осей, що інтерполюються, - 24.

Інтерес представляють закладені функції штучного інтелекту:

- Система керування с функціями штучного інтелекту. Застосування новітньої технології цифрового керування сервоприводом шпинделя HRV та

функції віртуального попереднього прогону циклу обробки дозволяє зменшити похибку запізнення сервоприводу та забезпечує максимально точне керування траєкторією ріжучого інструменту.

- Modeler Control II («Розумна» система контурного и наноконтурного керування). Керування розгоном та гальмуванням відбувається шляхом попереднього зчитування 40 блоків даних в режимі контурного контурного и 180 блоків в режимі наноконтурного керування, що забезпечує високошвидкісну прецизійну обробку.

- Система компенсації ударних та вібраційних збурень. Автоматичне керування подачею при обробці кутів з метою компенсації ударних збурень.

- «Розумна» прецизійна система контурного и наноконтурного керування. При керуванні розгоном та гальмуванням з використанням RISC-карти попередньо зчитується до 600 блоків програмних даних. Цим самим забезпечується високошвидкісна прецизійна обробка за рахунок високостабільної швидкості подачі навіть для програм обробки, що складаються із надмілких лінійних сегментів. Для отримання деталей з високим ступенем чистоти обробки, за якої фінішна обробка практично не потрібна, програму контурної обробки можна доповнити функцією наноінтерполяції.

- «Розумна» система компенсації температурної деформації по осі Z. Поправка на температурну деформацію шпинделя и оси Z здійснюється шляхом контролю цього параметру та оцінки величини зміни по осі Z (рівень точності поправки залежить від конкретних робочих умов).

- Система автоматичного обліку зношування інструментів (контролю стану інструментів) з застосуванням елементів штучного інтелекту. Система регулювання життєвого циклу інструментів слідує за тривалістю та частотою експлуатації та автоматично провадить заміну їх в разі перевищення експлуатаційних параметрів. Вбудована система обліку зношування інструментів визначає фактичне навантаження на свердло через навантаження

сервоприводу шпинделя, що дозволяє оптимізувати термін служби інструментів.

Система ЧПК Fanuc має найширші асортименти прикладних функцій і визначає вектор розвитку для багатьох виробників.

Лідер в області промислової автоматизації фірма Siemens представила нові модифікації своєї флагманської системи SINUMERIK 840D/Di sl (Solution Line), у якій додані функціональності:

- утиліта для пуско-налагодочних робіт для початкового етапу роботи;
- Motion Control Information System – для оптимальної інтеграції верстатів в систему електронної обробки даних – призначена для координації планування, розміщення, виконання, скорочення підготовчо-заклучного часу, скорочення простоїв, спрощення аналізу неполадок;
- сплайн-інтерполяція, компресія кадрів;
- расширення відкритої архітектури;
- ShopMill, ShopTurn – системи підтримки цехового програмування;
- симуляція процесу обробки на верстаті.

У системах ЧПК 840D/Di sl кількість керованих осей досягає 64. Архітектура системи й політика відкритості породили на ринку безліч незалежних постачальників прикладних додатків для систем ЧПК Siemens, продукція яких систематизується в каталогах.

За рубіж, у тому числі й у Росію, фірма Heidenhain поставляє тільки експортний варіант систем ЧПК з обмеженим набором функціональних можливостей. Системи ЧПК мають орієнтацію на підтримку технологічного процесу своїм потужним набором верстатних циклів, що робить їх особливо привабливими для кваліфікованих операторів і технологів-програмістів.

Представлена Hi-End модель ЧПК серії iTNC-530, орієнтована на обробку поверхонь довільної форми, дозволяє управляти 13 осями при часі обробки кадру до 0,5 мс. Надається опція використання системи ЧПК із двома

процесорами й операційною системою Windows XP. Система має наступні можливості:

- Expanded look-ahead – функція попереднього перегляду з буфером до 1024 кадрів - своєчасно розпізнає зміни напрямку руху інструменту для наступної корекції розгону та гальмування;
- AFC (Advanced Feed Control) – адаптивне керування подачею - регулює контурну швидкість переміщення по траєкторії в залежності від ступеня завантаження приводу шпинделя;
- DriveDiag, TNCopt - утиліти для діагностики складових елементів приводів та ініціалізації цифрових контурів регулювання;
- PLCdesignNT, PhytonOEM – інструментарій розробки та функція високорівневої об'єктно-орієнтованої мови керуючих програм електроавтоматики;
- CicleDecign – інструментарій розробки верстатних циклів;
- TeleService – інструментарій для віддаленої діагностики, контролю та управління системою ЧПК;
- TNCremoPlus – утиліта передачі даних між ЧПК та ПК, яка реалізує також і режим LiveScreen – передачу вмісту екрану;
- RemoToolSDK – комплект засобів розробки, який містить COM-компоненти та ActiveX керуючі елементи для доступу до ядра системи ЧПК Heidenhain;
- VirtualTNC – компонент управління віртуальними верстатами, які застосовуються для моделювання роботи верстатів;
- інструментарій вимірювань та аналізу сигналів, в який входять: 6-канальний осцилограф, логічний аналізатор, тест кола;
- DCM (Dynamic Collision Monitoring) – 3D-функція моніторингу зіткнень та виявлення колізій при відпрацюванні керуючої програми на верстаті.

Німецька фірма Bosch Rexroth представила свою систему MTX advanced, у якій число керованих осей досягає 64, вона має 12 каналів керування й до 8 осей з

інтерполяцією у каналі. Система створена на базі систем ЧПК Typ3 osa (Bosch) і MTC (Indramat), має двокомп'ютерну архітектуру й сучасний користувацький інтерфейс. MTC використовує інтерфейс SERCOSIII для керування приводами й електроавтоматикою.

Слід зазначити просунутий стиль екрану й інформативність користувацького інтерфейсу, що обумовлено використанням технології .NET і новітніх розробок останніх років, на відміну від інших європейських систем, що сформувалися в 90-х роках.

Система оснащена наступними функціональними можливостями:

- NC-Analyzer – інструментарій розробки й налагодження керуючих програм на мові високого рівня CPL (Custom Program Language) ,
- Logic Analyzer и Drive Oscilloscope – інструментарій для діагностики входів/виходів автоматики та цифрових приводів;
- Cycle time analyzer – інтелектуальний інструмент аналізу часових циклів в системі керування, виконує запис потоку команд ЧПК, подій з ПЛС, сигналів від приводів та від периферії верстата. Результати представляються у вигляді графічної часової діаграми;
- IndraWorks view 3D – тривимірна візуалізація процесу обробки керуючої програми та виявлення колізій.
- IndraWorks machine simulator – симулятор периферії верстата, підключеної по шині PROFIBUS.
- Nanometer resolution – обчислення та інтерполяція нанометричною точністю;
- CPL (Custom Programming Language) – Паскаль-подібна мова програмування високого рівня;
- Remote diagnostics I-Remote – інструментарій для віддаленого доступу, діагностики та підтримки систем керування;
- SCP (Scaleble Communication Platform) – масштабована комунікаційна платформа, яка здійснює доступ до відкритої архітектури системи.

Двокомп'ютерна модель Mitsubishi M750 (фірми Mitsubishi Electric), створена на базі комп'ютера з RISC-процесором (для завдань реального часу) і інтерфейсного комп'ютера з операційною системою Windows XP, зв'язаних по Ethernet. Система ЧПК Mitsubishi M750 підтримує технологію нанокерування й забезпечує високошвидкісну обробку. Залежно від конкретної модифікації вона реалізує до 4 каналів керування й використовує до 16 керованих осей, вісім з яких одночасно підтримують інтерполяцію. Сервоприводи управляються через високошвидкісну оптичну мережу. Система забезпечує обробку до 151 000 кадрів за хвилину при чистових технологічних операціях. Платформа і Platform, заснована на гігабітному Ethernet, дозволяє об'єднувати в єдину систему керування окремі контролери, виступаючи в якості інтегруючої внутрішньої шини.

Система реалізує функції:

- Complete Nano Control - дискретність виконуваних інтерполяційних розрахунків складає 1 нм;
- SSS-control (Super Smooth Surface) – керування якістю й забезпечення сталих показників обробки, незалежно від геометрії форми та швидкості різання. Застосовуються алгоритми: оптимізації розгону та гальмування, компенсації люфту та нерівномірності ходового гвинта, сплайнової інтерполяції (кубічний сплайн та NURBS), автоматичне розпізнавання кутів;
- OMR-control (Optimum Machine Response) – на основі моделі функція виконує: корекцію похибок, пов'язаних з деформацією координати; синхронізацію руху сервоприводу та шпинделя шляхом врахування мережевої затримки команд; мінімізацію впливу вібрацій;
- 5 Axis Machining Control – функція запобігання зіткнень в процесі обробки - враховує модель верстата, відносне положення деталі та траєкторію руху інструменту. В разі виявлення небезпеки зіткнення привод зупиняється, а в тривимірній моделі підсвічується місце колізії;
- Machining time calculation - розрахунок часу циклу без запуску верстата;

- MS Configurator – програмний продукт для настройки приводів ЧПК, який дозволяє налаштовувати сервоприводи в автоматичному режимі. Він будує діаграму bode, виконує налаштування підсистем регулювання швидкості та прискорення, налаштування фільтру резонансних частот, будує тест кола та осцилограму сигналів сервоприводів;
- NC Configurator – інструментарій настройки машинних параметрів системи ЧПК, параметрів інструментів та глобальних змінних;
- NC-Designer – пакет для створення екранів інтерфейсу користувача;
- NC Monitor – віддалений термінал на PC, що дозволяє керувати системою ЧПК через Ethernet;
- NC-Explorer – для підключення системи ЧПК до мережі Ethernet та передачі технологічних програм;
- · NAVI MILL, NAVI LATHE – інструментарій створення фрезерних та токарних керуючих програм на програмній PC-станції;

Контрольні питання

1. Наведіть характеристики сучасних систем ЧПК.
2. Що означає термін «наноінтерполяція»?
3. Поясніть політику відкритості при розробці систем ЧПК?
4. Як працює функція запобігання зіткнень?
5. Поясніть функцію багатоканальності системи ЧПК.

Список літератури

1. Гост р 50369—92. Электроприводы. Термины и определения. — м.: изд-во стандартов, 1993. — 16 с.
2. Завгороднев п. И. Работа оператора на станках с программным управлением : учеб. Пособие для техн. Училищ / п. И. Завгороднев. — м. : высш. Шк., 1981. — 136 с.
3. Программное управление станками и промышленными роботами : учебник / [в.л.косовский, ю.г.козырев, а.н.ковшов и др.]. — м. : высш. Шк., 1989. — 272 с.
4. Сергиевский л. В. Пособие наладчика станков с чпу / л. В. Сергиевский, в.в.русланов. — м.: машиностроение, 1991. — 176 с.
5. Соломенцев ю. Н. Управление гибкими производственными системами / ю. Н. Соломенцев, в.л.сосонкин. — м.: машиностроение, 1988. — 552 с.
6. Сосонкин в. Л. Микропроцессорные системы числового программного управления станками / в.л.сосонкин. — м. : машиностроение, 1985. - 288 с.
7. Черпаков б. И. Металлорежущие станки: учебник для нач. Проф. Образования / б. И. Черпаков, т.а.альперович. — м.: издательский центр «академия», 2004. — 368 с.
8. Числовое программное управление станками / [в.л.сосонкин, о. П. Михайлев, ю. А. Павлов и др.]; под ред. В. Л. Сосонкина. — м.: машиностроение, 1981. — 398 с.
9. Чудаков а. Л.. Проектирование систем управления станками и станочными комплексами : гл. 1.8 // машиностроение : энциклопедия. Т. 1у-7. Металлорежущие станки и деревообрабатывающее оборудование. — м.: машиностроение, 1999. — с. 268—292.

ЗМІСТ

	Вступ.....	3
<i>Лекція 1.</i>	Системи числового програмного керування...	4
<i>Лекція 2.</i>	Конструктивні особливості верстатів з ЧПК..	10
<i>Лекція 3.</i>	Класифікація систем програмного керування	21
<i>Лекція 4.</i>	Загальна структура систем ЧПК.....	26
<i>Лекція 5.</i>	Структура програми і принцип роботи системи.....	32
<i>Лекція 6.</i>	Контроль вхідної інформації.....	39
<i>Лекція 7.</i>	Перетворювачі кодів.....	44
<i>Лекція 8.</i>	Блок завдання швидкості.....	52
<i>Лекція 9.</i>	Блок керування приводами подач.....	57
<i>Лекція 10.</i>	Інтерполятори.....	63
<i>Лекція 11.</i>	Лінійно-кругові інтерполятори.....	67
<i>Лекція 12.</i>	Побудова лінійного інтерполятора за методом оціночної функції.....	71
<i>Лекція 13.</i>	Кругова інтерполяція за методом оціночної функції.....	77
<i>Лекція 14.</i>	Робота кругового інтерполятора за методом ОФ.....	82
<i>Лекція 15.</i>	Лінійна інтерполяція за методом цифрових диференціальних аналізаторів.....	88
<i>Лекція 16.</i>	Програмна реалізація лінійного інтерполятора за методом ЦДА.....	94
<i>Лекція 17.</i>	Кругова інтерполяція за методом цифрових диференціальних аналізаторів.....	100
<i>Лекція 18.</i>	Тенденції розвитку сучасних систем програмного керування.....	104
	Список літератури.....	113