

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр	
(назва освітнього ступеня)	
на тему:	Розроблення та дослідження розподіленої автоматизованої системи керування CNC верстатами

Виконав: студент VI курсу, групи КТм-61
174 Автоматизація, комп'ютерно-інтегровані технології та
робототехніка
спеціальності (шифр і назва спеціальності)

	(підпис)	Сорба Є.І. (прізвище та ініціали)
Керівник	(підпис)	Золотий Р.З. (прізвище та ініціали)
Нормоконтроль	(підпис)	Чихіра І.В. (прізвище та ініціали)
Завідувач кафедри	(підпис)	Голотенко О.С. (прізвище та ініціали)
Рецензент	(підпис)	Капаціла Ю.Б. (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет Факультет прикладних інформаційних технологій та електроінженерії
(повна назва факультету)
Кафедра комп'ютерно-інтегрованих технологій
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Голотенко О.С.
(підпис) (прізвище та ініціали)

« » 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
(шифр і назва спеціальності)

Студенту Сорбі Євгеу Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення та дослідження розподіленої
автоматизованої системи керування CNC верстатами

Керівник роботи Золотий Роман Захарійович, канд. техн. наук, доцент кафедри КТ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «12» листопада 2025 року № 4/7-962

2. Термін подання студентом завершеної роботи 22 грудня 2025р.

3. Вихідні дані до роботи Технічні вимоги щодо процесу виробництва ЧПУ верстататми

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ

Аналітична частина

Технологічна частина

Конструкторська частина

Науково-дослідна частина

Спеціальна частина

Охорона праці та безпека в надзвичайних ситуаціях

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Титульний слайд

Актуальність роботи

Завдання роботи

Основна частина

Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та	доц. Тотосько О.В.		
безпека в надзв. ситуаціях			

7. Дата видачі завдання 12 листопада 2025 року

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент

(підпис)

Сорба Є.І.

(прізвище та ініціали)

Керівник роботи

(підпис)

Золотий Р.З.

(прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота магістра складається з пояснювальної записки та графічної частини (ілюстративний матеріал – слайди).

Об'єм графічної частини дипломної роботи становить _____.

Об'єм пояснювальної записки складає ____ друкованих сторінок формату А4.

В роботі використано ____ літературних джерел.

Метою роботи була розробка та впровадження розподіленої системи керування для верстатів з ЧПК. Були реалізовані різні структури.

Головним вузлом цієї системи керування є ПК та відповідне програмне забезпечення керування. Система керування в реальному часі не може керуватися безпосередньо ПК. Переривання викликаються підлеглими вузлами. Необхідний запит або надання інформації зворотного зв'язку. Тому обсяг даних, які необхідно передати по комунікаційній мережі, збільшиться. Цю ситуацію було вирішено, застосовуючи швидші комунікаційні інтерфейси.

Розроблену розподілену систему керування, що включає апаратне забезпечення, прошивку та програмне забезпечення, було реалізовано на керованому (3-осьовому) обробному центрі з ЧПК з розімкнутим циклом. Було досліджено продуктивність обробки за різних умов. Загальна продуктивність системи керування знаходилась в оптимальних діапазонах.

Розроблений графічний інтерфейс користувача розширив межі розподілених контролерів. Спочатку було перевірено здатність до 2D-інтерполяції. Потім розподілені контролери були змушені продемонструвати свої навички 3D-інтерполяції.

Ключові слова: КОНТРОЛЕР, ВЕРСТАТ, ОБРОБКА, КЕРУВАННЯ, ОПТИМІЗАЦІЯ.

ЗМІСТ

ВСТУП	5
1. АНАЛІТИЧНА ЧАСТИНА	6
1.1. Огляд існуючих архітектур	6
1.2. Розподілені системи керування.....	10
1.3. Мережеві системи керування та затримки часу.....	12
1.4. Синхронізація	16
1.5. Дослідницькі можливості.....	18
2 ТЕХНОЛОГІЧНА ЧАСТИНА.....	20
2.1. Системи зв'язку і розробка протоколів	20
2.2 Фізичні комунікаційні інтерфейси	24
3 КОСТРУКТОРСЬКА ЧАСТИНА.....	43
3.1 Архітектура апаратного забезпечення системи керування	43
3.2. Архітектура апаратного забезпечення.....	44
4 НАУКОВО-ДОСЛІДНА ЧАСТИНА.....	54
4.1 Дослідження розробленої системи.....	54
5 СПЕЦІАЛЬНА ЧАСТИНА	62
5.1 Графічний інтерфейс користувача та його особливості.....	62
6 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ... 68	
6.1. Безпека електрообладнання у виробничих процесах.....	68
6.2. Аналіз ризиків на робочому місці оператора	70
6.3. Охорона праці при експлуатації системи	73

ВИСНОВКИ.....	76
----------------------	-----------

ПЕРЕЛІК ПОСИЛАНЬ.....	77
------------------------------	-----------

ВСТУП

Верстати з числовим програмним керуванням (ЧПК), які автоматично керуються за допомогою закодованих (цифрових) команд, здатні обробляти компоненти з високою якістю та кількістю. Виробнича промисловість сильно залежить від цих верстатів. У сучасній технології ЧПК було адаптовано багато різних архітектур керування. Централізована система керування досить популярна в промисловості завдяки простоті її реалізації. Якщо кількість керованих осей на верстаті з ЧПК (>3), збільшується і обчислювальне навантаження на центральні процесори. Отже, потрібні потужніші процесори. Альтернативною архітектурою, яка зазвичай не використовується в технології ЧПК, є децентралізована (розподілена) система керування. У цій топології завдання обробляються розподіленими контролерами, які з'єднані між собою мережею зв'язку. За потреби новий контролер можна легко додати до мережі без розширення фізичної конфігурації. Незважаючи на привабливі особливості, ця архітектура не була повністю прийнята індустрією ЧПК. Синхронізація між осями в скоординованому русі виявилася досить складною.

У цій роботі було запропоновано альтернативну архітектуру розподіленого контролера для верстатів з ЧПК. Вона була реалізована на 3-вісному фрезерному верстаті з ЧПК.

1. АНАЛІТИЧНА ЧАСТИНА

1.1. Огляд існуючих архітектур

Верстати з числовим програмним керуванням (ЧПК), які автоматично керуються за допомогою закодованих (цифрових) команд, здатні обробляти компоненти з високою якістю та кількістю. Ранні верстати програмувалися за допомогою програми, що зберігалася на (перфострічці), тоді як їхні робочі параметри не можна було легко змінити через зручний інтерфейс користувача. Пізніше програми обробки деталей передавалися на верстат через протокол послідовного зв'язку з мейнфрейму. Ці верстати з апаратно-керованими системами керування з часом еволюціонували завдяки досягненням у цифровій електроніці. Впровадження мікрокомп'ютерів можна вважати проривом для верстатів з ЧПУ, що перетворило їх на верстати з числовим програмним керуванням (ЧПК), де до системи можна інтегрувати кілька бортових комп'ютерів. Як наслідок, виробнича промисловість, яка значною мірою залежить від цієї технології для високоякісної обробки, зазнала радикально змін.

Окрім покращеного керування (положенням інструменту), розвиток комп'ютерних технологій також запропонував багато універсальних утиліт, таких як редагування програм, керування файлами/даними та вбудована діагностика. Крім того, в блоки ЧПК інтегровані засоби автоматизованого проектування, що дозволяють виконувати складне моделювання обробки.

Сприяючи розвитку мови програмування обробки деталей з ЧПУ, ці інструменти з часом перетворилися на невід'ємну частину виробничої промисловості.

У сучасних технологіях ЧПК було адаптовано багато різних архітектур керування. Серед цих архітектур централізована система керування, яка

проілюстрована на рис. 1.1, є досить популярною в промисловості завдяки простоті реалізації.

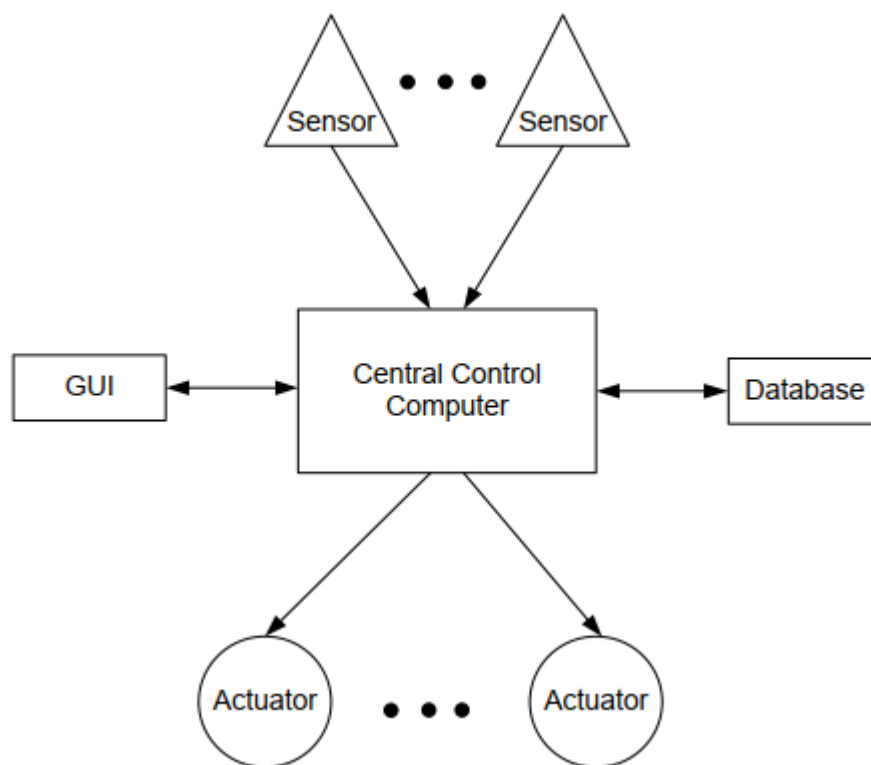


Рисунок 1.1 - Архітектура централізованого контролера

У таких системах центральний процесор (зазвичай цифровий сигнальний процес) не лише генерує відповідні команди позиціонування для кожної осі, але й виконує відповідні обчислення керування в режимі реального часу. Очевидно, що зі збільшенням кількості керованих осей на верстаті з ЧПК (>3) обчислювальне навантаження на центральний процесор значно зростає. Отже, для обробки потрібні потужніші процесори. Незважаючи на те, що швидкі процесори (з кількома ядрами) стають широко доступними на ринку (за розумними цінами), такий підхід не є економічно обґрунтованим через те, що масштабні зусилля з розробки (тобто проектування друкованих плат та прошивки) зазвичай пов'язані з проектуванням всієї системи, оскільки потреби (та очікування щодо продуктивності) виробничої промисловості постійно зростають.

Альтернативна архітектура, що використовується в технології ЧПК, - це децентралізована (розподілена), як показано на рис. 1.2. У цій топології завдання, що обробляються системою, розподілені між різними вузлами мережі. Кожен вузол виконує певну операцію, використовуючи виділений процесор з відносно скромними ресурсами. Вузли з'єднані один з одним високошвидкісною послідовною шиною, такою як RS-485, CAN, Profibus, Ethernet тощо. Зазвичай координація, а також зв'язок цих вузлів здійснюються головним процесором.

За потреби, новий вузол можна легко додати до мережі без розширення фізичної конфігурації. Незважаючи на свої привабливі характеристики, ця архітектура не була повністю прийнята індустрією ЧПК через те, що синхронізація між осями в скоординованому русі виявилася досить складною. Крім того, для застосувань електронних зубчастих передач, де одна або кілька осей повинні точно слідувати руху головної осі, система шини не може ефективно передавати величезну кількість (надлишкових) даних, якими необхідно обмінюватися між «електронно» пов'язаними осями.

Важливо зазначити, що безщіткові двигуни постійного струму та їх приводи є основою застосувань керування рухом. Сучасні системи приводів двигунів мають здатність точно контролювати положення свого серводвигуна за умови дотримання відповідних вимог.

Давачі підключені до системи приводу. Якщо для кожної системи приводу двигуна буде розгорнуто спеціальний контролер, який керуватиме не лише передачею даних, але й синхронізацією осей (за допомогою відповідної абстракції), проектування децентралізованої системи керування, яка відповідає очікуванням галузі, може стати економічно доцільним.

Основною тенденцією промислових систем керування є інтеграція обчислювальних технологій, зв'язку та керування на різних рівнях машин [1, 5, 9, 12, 13]. Традиційна комунікаційна архітектура, яка впроваджується в

промисловості протягом десятиліть, використовує централізований контролер, який має можливість інтегрувати датчики та виконавчі механізми.

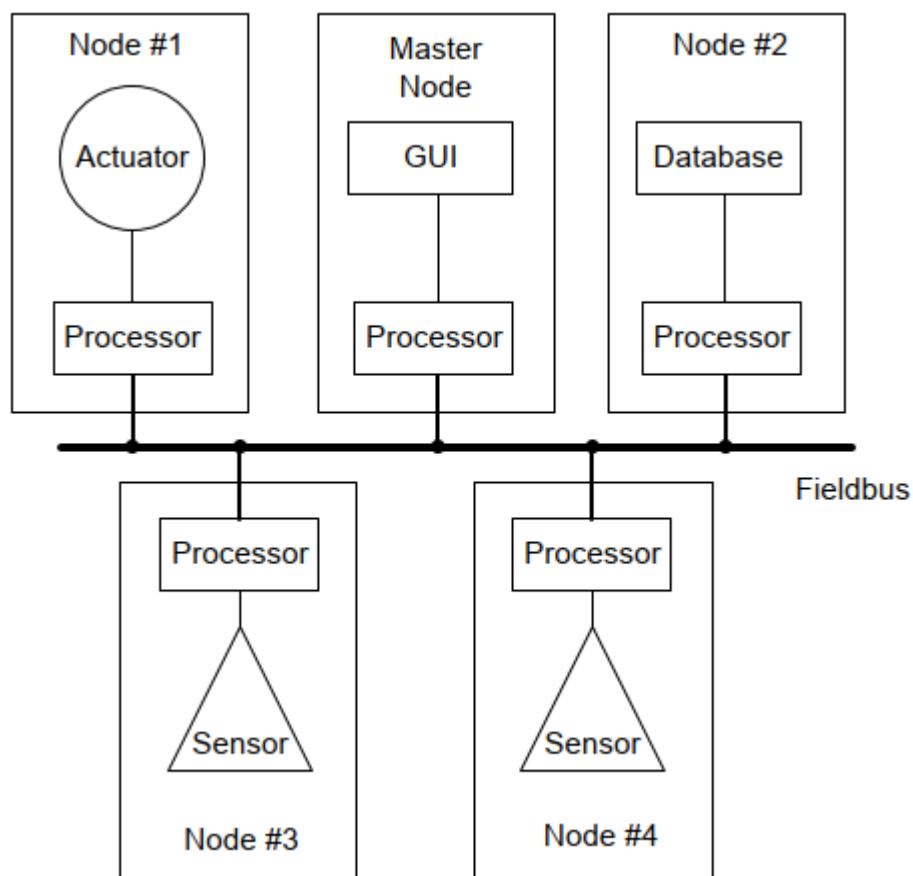


Рисунок 1.2 - Розподілена архітектура керування

На жаль, розширення фізичних конфігурацій та очікування розширеної функціональності часто призводить до меж таких систем [3, 17].

Сьогодні промисловість вимагає модульності, децентралізації управління, інтегрованої діагностики, швидкого та легкого обслуговування, а також низької вартості, щоб змусити свої виробничі потужності розширюватися [12,13,17]. Впровадження мережевих комунікаційних систем може зменшити складні апаратні архітектури, підвищити гнучкість та масштабованість, а також підвищити ідею реконфігурації та функціональності [6, 10]. Зміна архітектури комунікації призводить до часових затримок, які безпосередньо пов'язані з часом, необхідним для

формування сигналу та обробки інформації [2]. Такі характеристики часових затримок можуть бути постійними, обмеженими або випадковими [16]. В основному, це залежить від фізичних атрибутів комунікаційного інтерфейсу, кількості вузлів, підключених до мережі, та розробленого комунікаційного протоколу [14]. Однак аналіз мережевих затримок є різноманітною та прогресивною галуззю досліджень.

Більшість досліджень мережевих систем керування зосереджені на двох напрямках: комунікаційних технологіях та проектуванні контролерів. Простіше кажучи, протокол передачі повідомлень гарантує якість мережі, тоді як контролер відповідає за якість роботи.

1.2. Розподілені системи керування

Розподілені системи керування (РСК) – це, по суті, просторово мережеві вбудовані вузли, які з'єднані між собою за допомогою дротової та/або бездротової комунікаційної інфраструктури та протоколів, що взаємодіють з навколишнім середовищем та один з одним. Прикладами розподілених систем керування є промислова автоматизація, автоматизація будівель, офісна та домашня автоматизація, інтелектуальні транспортні системи та передові системи літаків і космічних апаратів [14]. Спільною рисою цих застосувань є велика кількість пристроїв, з'єднаних між собою на відносно великій площі для виконання певного завдання. Очевидно, що обчислювальне навантаження на централізований блок керування є великим, якщо всі обчислення мають оброблятися цим блоком. Крім того, для підключення всіх блоків до централізованого процесора необхідно прокласти велику кількість проводів.

Розподілені системи керування пропонують альтернативне рішення. У літературі ці системи також називають мережевими системами керування (NCS), де різні модулі можуть взаємодіяти один з одним. Розробка

відповідних архітектур керування та пов'язаних з ними алгоритмів проектування контролерів для таких систем має вирішальне значення для успіху, оскільки датчики та виконавчі механізми часто розподілені по великій площі [19].

Розподілені системи керування зазвичай використовуються у складних інженерних системах [11, 18]. Комунікаційні мережі та сенсорні мережі вважаються важливими елементами основної інфраструктури.

Г. Вайет та ін. розробили розподілений цифровий контролер для промислового робота-маніпулятора PUMA 560 з шістьма ступенями свободи [17]. У їхньому дослідженні розроблена централізована система керування складається з контролера для кожного з'єднання, які об'єднані в мережу через шину CAN. ПК забезпечує інтерфейс користувача для запропонованої системи керування. Вони використовують цифровий сигнальний процесор (DSP) для кожного контролера, щоб виконувати бажані операції спільного керування роботом-маніпулятором. Фактично, шина CAN усуває складну проводку та розширює функціональність спільних контролерів. Зауважте, що в сучасних системах керування продуктивність відстеження визначається коефіцієнтами посилення (стану) контролера зворотного зв'язку, а також частотою дискретизації. Зазвичай, для найкращої продуктивності відстеження команд бажані вищі коефіцієнти посилення в поєднанні з високою частотою дискретизації. Отже, розподілені контролери можна використовувати на дуже високих частотах дискретизації (20 кГц і вище). Крім того, діагностика та налагодження системи керування стають простішими. Що стосується їхнього дослідження, запропонований контролер, який є простим, економічно ефективним та легшим у реалізації, перевершує продуктивність оригінальної системи.

Аналогічно, НА Thompson та ін. демонструють розподілену систему керування авіаційним двигуном. Система визначена як критично важлива для безпеки. Отже, вони використовують дволінійну архітектуру для збільшення

обчислювальної потужності для операцій керування та сенсорних систем. Вони також отримують такі переваги, як: можливість модернізації, підвищена надійність та покращена діагностика.

Головна мета прогресивних дослідницьких та дослідно-конструкторських робіт об'єднана під назвою інтелектуальні виробничі системи. Розподілена система керування взаємодіє через мережу між своїми розподіленими процесорами. Розподілені контролери належним чином скоординовані для досягнення бажаного(их) завдання(й) з максимальною швидкістю та точністю. Таким чином, вони взаємодіють не лише з виробничим середовищем, але й із системою керування та її елементами (давачами).

1.3. Мережеві системи керування та затримки часу

Рішенням сучасних проблем керування є розподіл функцій обробки між мережевими контролерами [4, 17]. Такі контролери (які називаються вузлами) розподіленої/мережевої системи керування мають спільний канал зв'язку, який зазвичай має шинну топологію. Отримані архітектури потребують менше проводки, ніж традиційні. Тому вартість обладнання різко знижується, незважаючи на зростання гнучкості.

Системи керування можна розділити на модульні підсистеми, які безпосередньо підключаються до основної системи. Така модульність призводить до покращення діагностики та обслуговування [15]. Як зазначалося раніше, розподілені системи стали привабливою альтернативою традиційним централізованим рішенням. Однак ці системи також створюють низку важливих проблем, які необхідно вирішувати належним чином. В іншому випадку продуктивність системи керування різко знизиться.

Можна виділити дві основні проблеми: адресацію та синхронізацію. У мережі спільної шини кожен пакет даних має бути доповнений

ідентифікаторами, заголовками або закодованими операндами, що вказують на пункт призначення та/або джерело переданого повідомлення. Крім того, розподіленим пристроям, можливо, доведеться чекати певний час, перш ніж вони надішлють повідомлення. Фактично, основним обмеженням тут є як фізичний інтерфейс зв'язку, так і використовуваний протокол зв'язку. Залежно від протоколу зв'язку, кількість вузлів, підключених до мережі керування, також погіршують продуктивність зв'язку. Тобто, часові затримки за своєю суттю присутні в мережі можуть безпосередньо впливати на загальну продуктивність системи. Під час проектування контролера необхідно змодельовати елементи системи зв'язку (як протокол, так і фізичний інтерфейс зв'язку) та додати їх до загальної моделі системи [6].

Протягом кількох десятиліть вивчається моделювання та управління мережевими системами. Загалом, під час передачі інформації виникають затримки. Типовими прикладами систем із затримкою часу можна назвати великомасштабні системи (такі як телекомунікаційні, виробничі, транспортні та енергетичні системи).

Для аналізу цих систем використовуються підходи в частотній області (класичний) та часовій області (сучасний). Класичний підхід використовує аналітичні або графічні методи для визначення власних значень характеристичної системи. Стандартні графічні методи, такі як геометричне місце коренів, графік Боде та графік Найквіста, застосовні для знаходження відповідних передавальних функцій. Для дискретного часу з часовими затримками стійкість системи також можна визначити за допомогою аналізу геометричного місця коренів в z -області.

На жаль, класичний підхід можна використовувати переважно для простих випадків. Для систем з багаторазовими затримками необхідно використовувати функціональний випадок. Тобто, використовуючи функціональний підхід, можна вивести диференціальні рівняння із затримкою та охарактеризувати системи. Фактично, багаторазові затримки та

змінні в часі затримки також можна моделювати за допомогою цього підходу, а їхню стійкість можна аналізувати за допомогою другого методу Ляпунова. Зауважте, що більшість застосувань із часовими затримками вважаються недетермінованими. Тому на відомій структурі систем можна реалізувати робустні контролери, щоб подолати невизначеність затримки.

Затримка обчислень мікропроцесора розглядається як ще одне джерело затримки. Виконання інструкцій у прошивці керування потребує часу. Час виконання має прямий вплив на продуктивність керування.

Мережева система керування використовує в мережі низку датчиків та виконавчих механізмів. Надсилання команд та отримання даних потребує часу для реалізації відповідних процедур. Час передачі може бути майже постійним або змінюватися випадковим чином. Також ще одним питанням для таких систем є те, чи керуються вузли подіями, чи тактовими вузлами. Вузли, керовані подіями, починають свою діяльність, коли відбувається подія, тоді як вузли, керовані тактовими вузлами, починають свою діяльність у задані часові рамки. Як показано на рис. 1.3, у мережевій системі керування існують по суті три види часових затримок:

1. Затримка зв'язку між датчиком і контролером,
2. τ_k^{sc} обчислювальна затримка в контролері,
3. τ_k^c затримка зв'язку між контролером та виконавчим механізмом.

Затримка часу, τ_k , для системи керування включає вибірку даних датчиків, розрахунок керуючого сигналу та надсилання керуючого сигналу до вузла виконавчого механізму:

$$\tau_k = \tau_k^{sc} + \tau_k^c + \tau_k^{ca} \quad (1.1)$$

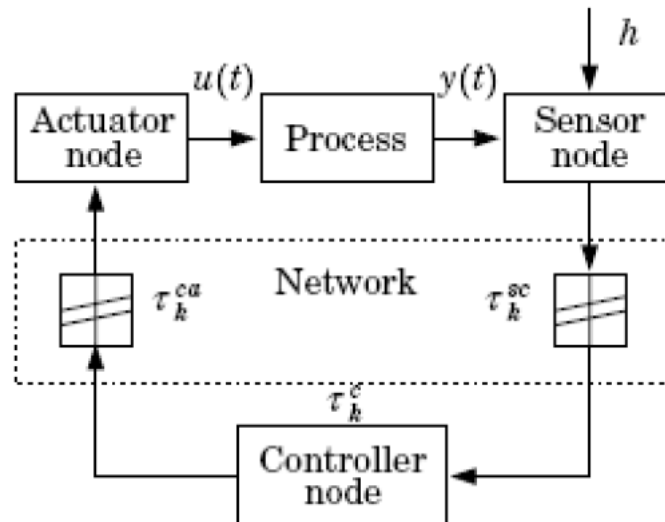


Рисунок 1.3 - Розподілена система керування з індукованими затримками [16]т

Часові затримки мають різні характеристики залежно від апаратного та програмного забезпечення, які необхідно моделювати. Затримки в мережі змінюються через мережеве навантаження, операції на основі мережевого протоколу (наприклад, виклик головного або підлеглого вузлів) та фізичні збої в мережі. Характеристики часової затримки мереж керування, встановлених у промислових системах, можна класифікувати як стохастичні, обмежені та постійні. Найпростіша модель затримки в мережі припускає, що затримки є постійними для всіх передач у комунікаційній мережі. Це може бути гарною моделлю, навіть якщо мережа має різні затримки. У цьому аналізі часто практикується призначення затримки найгіршого випадку або затримки середнього значення [16]. Призначення такого постійного елемента затримки важливе для стабільності та продуктивності системи.

Зауважте, що замкнуту систему можна перетворити на незмінну в часі систему шляхом введення буферів, як показано на рис. 1.4 [16]. Усі вузли вважаються тактованими та синхронізованими. Якщо буфери довші за максимальну затримку часу в системі керування, рівняння системи перетворюються на вираз:

$$x_{k+1} = Ax_k + Bu_{k-\Delta_1} \quad (1.2)$$

$$y_k = Cx_k \quad (1.3)$$

де Δ_1 – довжина буфера у вузлі виконавчого механізму, виражена у вибірках. Якщо буфер у вузлі контролера вважається таким, що має довжину Δ_2 зразка, вихід процесу, доступний для контролера в даний момент часу k стає

$$w_k = y_k - \Delta_2 \quad (1.4)$$

У такому випадку задачу проектування керування можна переформулювати як стандартну задачу вибірових даних. Набір інформації доступний для розрахунку є:

$$W_k = \{w_k, w_{k-1}, \dots\} \quad (1.5)$$

Для цієї задачі керування можна розробити оптимальний контролер LQG [16]:

$$u_k = \xi W_k \quad (1.6)$$

Цей метод обробляє затримки, які довші за період вибірки. Але оцінені затримки керування цим методом стають більшими, ніж необхідно. Звідси випливає, що це підходящий метод для систем, керованих подіями. Для систем, керованих часом, продуктивність контролера різко знижується.

1.4. Синхронізація

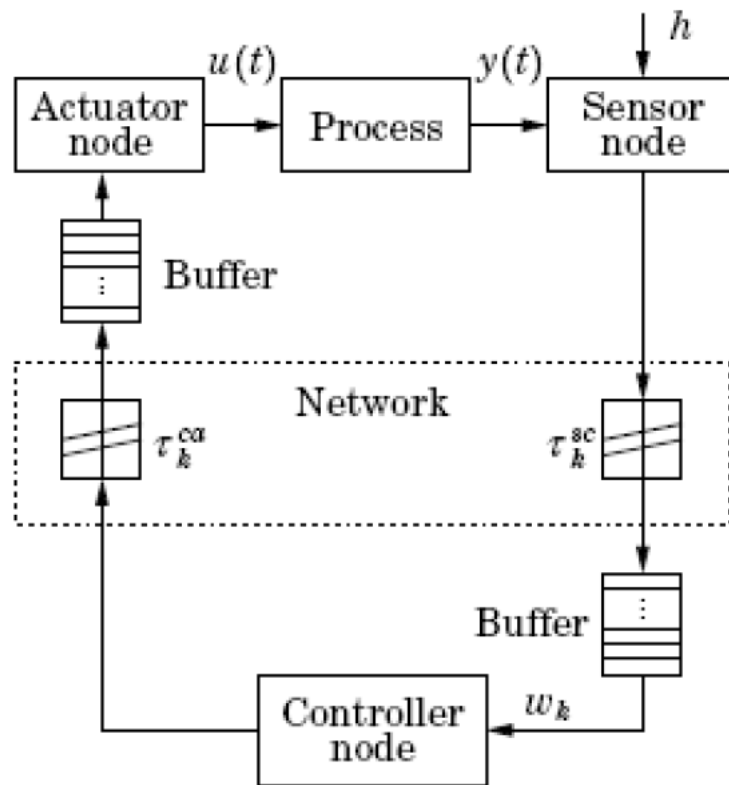


Рисунок 1.4 - Розподілена система керування з буферами [16].

Більшість розподілених систем, що зустрічаються на практиці, є асинхронними. Традиційні детерміновані, відмовостійкі алгоритми синхронізації годинників припускають обмежені затримки зв'язку [7]. Таким чином, їх не можна безпосередньо використовувати для синхронізації годинників в асинхронних системах.

Синхронізація розподілених вузлів є окремою галуззю досліджень. Єдина мета синхронізації годинника полягає у встановленні узгоджених значень годинників вузлів системи. У будь-якій розподіленій системі існує вимога координації та зв'язку, що особливо важливо в програмах керування в реальному часі [8, 16]. Глобальна координація та синхронізація повинні бути реалізовані відносно спільного часу. Однак, єдиним годинником, який зазвичай доступний на кожному вузлі, є локальний годинник використовуваних мікрокомп'ютерів. Цей годинник коливається на

номінальній частоті та генерує імпульсні потоки, які використовуються як прирости лічильника годинника. Синхронізація локального годинника розподілених контролерів підтримується за допомогою часових приростів лічильників годинника контролерів, тому вони діють разом як централізований контролер. Зауважте, що синхронізація може бути реалізована програмно або апаратно.

Надсилання сигналів синхронізації через мережу зв'язку можна розглядати як програмну синхронізацію. З іншого боку, апаратна синхронізація також є відповідним методом синхронізації вузлів системи. Цей метод, який вимагає додаткового підключення лише для розподілу глобального тактового сигналу, можна розглядати як дуже суворий метод синхронізації. Теоретично, це єдиний спосіб синхронізації вузлів системи.

1.5. Дослідницькі можливості

Розподілені системи керування – це просторово взаємопов'язані системи, які мають датчики, виконавчі механізми та контролери, з'єднані між собою комунікаційними мережами. Впровадження таких систем може задовольнити потреби сучасних промислових та комерційних систем. Зміна архітектури зв'язку з централізованої на мережеву/розподілену виявилася складною, оскільки вона вводить різні форми невизначеності часової затримки в динаміку замкнутої системи. Ці часові затримки безпосередньо пов'язані з обчислювальним часом, необхідним для фізичного кодування сигналів та обробки зв'язку.

Розподілені пристрої в мережі виконують керування спеціалізованими операціями, такими як керування виконавчим механізмом(ами) або кондиціонування даних датчиків. Таким чином, вони взаємодіють не лише з виробничим середовищем, але й із системою керування (та її відповідними елементами, такими як датчики). Окрім затримок зв'язку, існують

обчислювальні затримки, які необхідно ідентифікувати та враховувати в моделі системи керування. Представлені розподілені архітектури керування сприяють створенню комунікаційних мереж, що складаються з дорогого обладнання. Крім того, вони використовують дорогі та потужні вбудовані контролери.

Головною метою цього дослідження є вивчення (відносно) недосліджених аспектів вищезгаданих систем та розробка децентралізованої архітектури керування, яка підійде для «не дуже вимогливих» застосувань.

2 ТЕХНОЛОГІЧНА ЧАСТИНА

2.1. Системи зв'язку і розробка протоколів

Як було представлено в попередньому розділі, комунікація є однією з головних проблем у розподілених системах керування. Правильний вибір та реалізація фізичного інтерфейсу зв'язку зі зручним протоколом зв'язку є надзвичайно важливими. У цьому розділі представлені послідовні інтерфейси зв'язку разом з відповідними протоколами. Потім представлено розроблені протоколи зв'язку та надано їх оцінку.

Передача даних відбувається між термінальним обладнанням передачі даних (data terminal equipment, DTE) та кінцевим обладнанням каналу передачі даних (circuit terminating equipment, DCE) через різні комунікаційні інтерфейси, специфікації багаторівневих комунікаційних протоколів яких визначені відповідними стандартами. У передачі даних існують дві різні схеми електричного інтерфейсу: односторонні та диференціальні. Визначення цих систем наведено нижче.

Одностороння передача даних.

Основною характеристикою односторонніх схем передачі даних є те, що передача даних здійснюється по одній лінії, а логічний стан сигналу інтерпретується відносно землі. На рис. 2.1 показано таку схему.

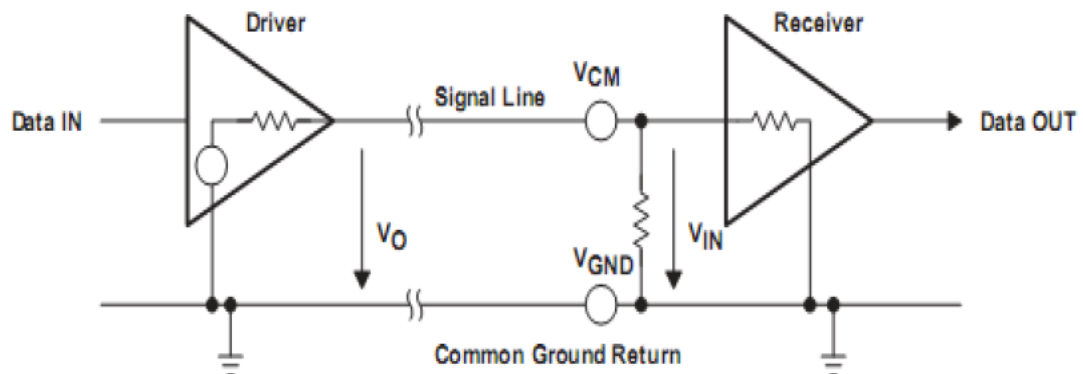


Рисунок 2.1 - Одностороння передача даних.

Одностороння схема передачі вимагає лише однієї лінії, тому така конфігурація дуже проста, легка у впровадженні та економічно ефективна. Великі відстані або шумне середовище іноді змушують додавати до системи краще екранування кабелю та додаткові лінії заземлення. Відомо, що ця конфігурація має погану завадостійкість. Основною проблемою тут є дрти заземлення. Вони є частиною схеми, і перехідні напруги можуть призвести до погіршення сигналу. Це призводить до неправильної інтерпретації сигналу або помилкового спрацьовування приймача. Також на вищих частотах можуть спостерігатися перехресні перешкоди.

Диференціальна передача даних

У цій конфігурації для диференціальної передачі даних необхідні як сигнальні лінії приймача, так і передавача. По одній лінії передається власне сигнал, тоді як друга лінія використовується для передачі інвертованої форми цього сигналу. Приймач виявляє різницю напруги між двома передавальними лініями та перемикає вихід залежно від величини цієї різниці потенціалів. На рис. 2.2 показано типову схему диференціальної передачі даних.

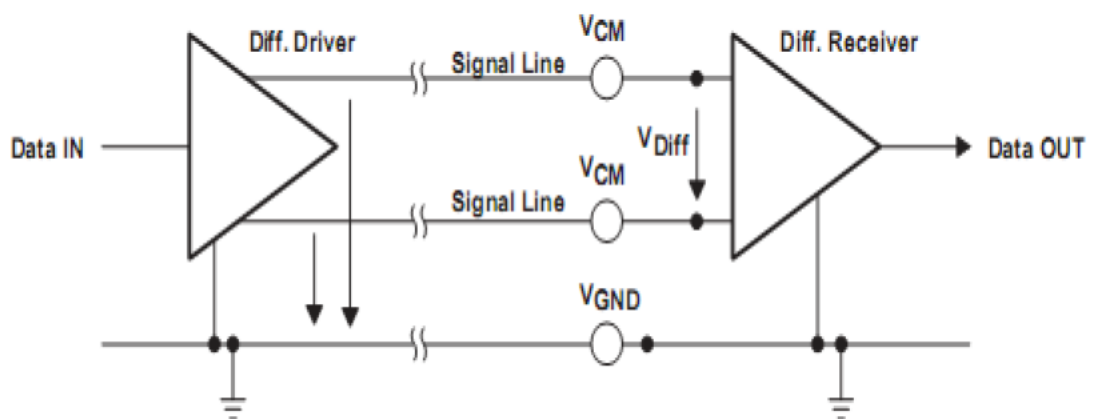


Рисунок 2.2 - Диференціальна передача даних.

Як показано на рис. 2.3, у цих інтерфейсах використовується виключно вита пара, оскільки такі конфігурації кабелів пропонують переваги щодо

перешкодостійкості порівняно з плоскими або стрічковими кабелями. Якщо використовується вита пара, обидва дроти однаково зазнають впливу джерел шуму. Шум створює синфазний сигнал, який називається один одним, коли приймач приймає різницевий сигнал. Крім того, правильне завершення лінії дозволяє уникнути будь-яких небажаних коливань і дозволяє передавати дані на вищі швидкості. За різних напрузі інтерфейсу швидкість передачі сигналу може сягати 10 Гбіт/с.

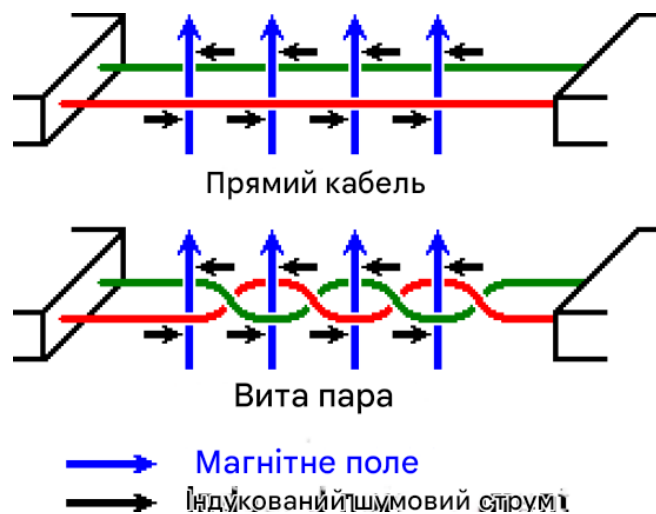


Рисунок 2.3 - Конфігурації кабелів.

Хоча схема збалансованого інтерфейсу складається з генератора з диференціальними виходами, збалансовані лінії передачі збільшують вартість таких інтерфейсів. Однак розробка цієї складної схеми інтерфейсу за допомогою CMOS-процесу виготовлення долає цю труднощі. Отже, мікросхеми приймачів-передавачів даних широко доступні на ринку за доступними цінами.

Топології передачі даних.

У телекомунікаційних технологіях існують топології передачі даних, які можна використовувати під час спроби з'єднати такі інтерфейси один з одним. Далі наведено детальний опис цих топологій.

Точка-точка (симплекс)

Як показано на рис. 2.4, топологію передачі даних "точка-точка" можна реалізувати з одним передавачем та одним приймачем на лінію. У цій топології (яку зазвичай називають односпрямованою передачею даних) дані можуть передаватися лише в одному напрямку.



Рисунок 2.4 -Точкове з'єднання.

Багатоточковий (розподілений, симплексний).

Як показано на рис. 2.5, багатоточкова топологія передачі даних отримується, коли використовується один передавач і більше одного приймача. Оскільки ця топологія успадковує характеристики передачі даних "точка-точка", дані можуть передаватися в одному напрямку.

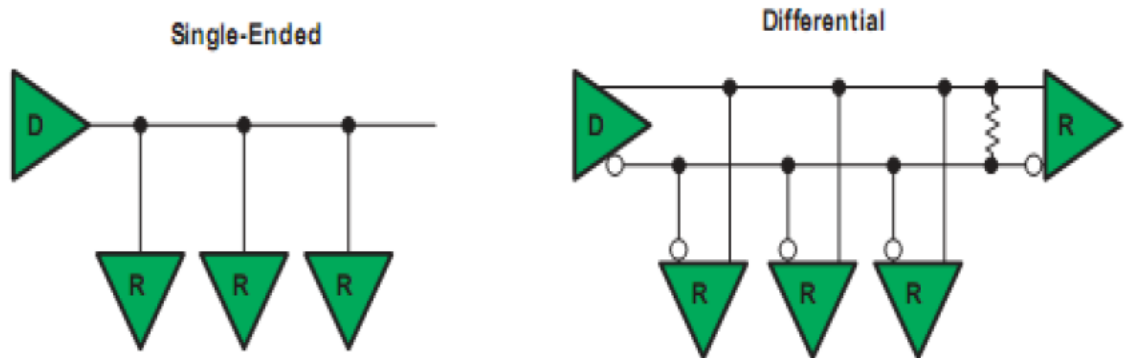


Рисунок 2.5 - Багатоточкове з'єднання.

Багатоточковий (мультиплексний)

Як показано на рис. 2.6, ця розширена топологія забезпечує двонаправлену передачу даних. Тут на кожній лінії існує пара передавач-приймач. Ця пара називається приймачем.

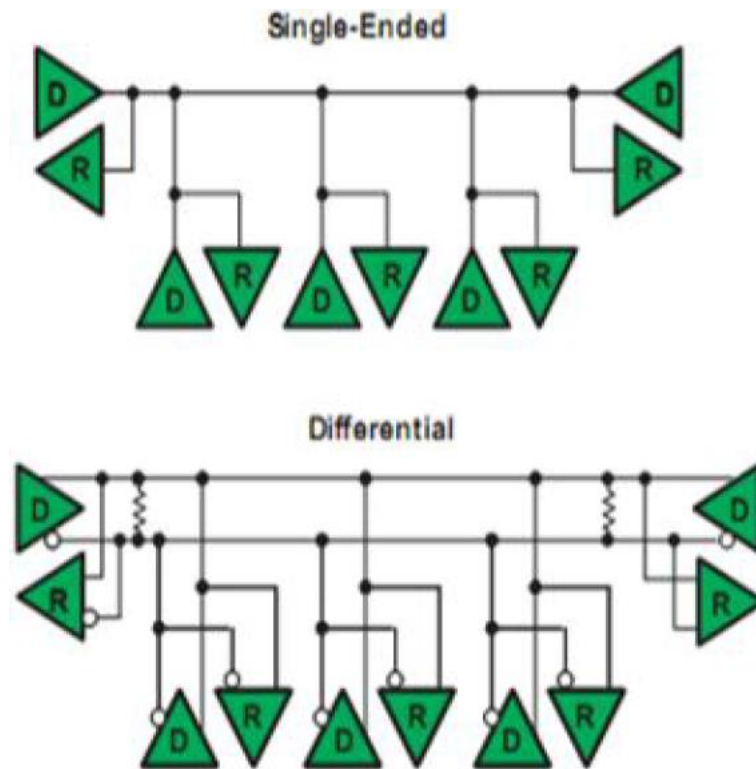


Рисунок 2.6 - Багатоточкове з'єднання.

2.2 Фізичні комунікаційні інтерфейси

У цьому розділі будуть оцінені різні стандарти зв'язку, що використовуються у вищезгаданих топологіях.

ОВНС – 232

EIA-232 (Рекомендований стандарт 232) – це найпростіший протокол послідовного зв'язку, адаптований Асоціацією електронної промисловості. Остання версія RS-232 C була випущена наприкінці 1960-х років. Сигнал EIA-232 має два логічні рівні: високий логічний рівень (+5 В до +15 В) та низький логічний рівень (-5 В до -15 В). Високий логічний рівень визначається негативними рівнями напруги, тоді як низький рівень визначається позитивними рівнями напруги. Згідно зі стандартами EIA-232, максимальна довжина кабелю повинна мати максимальну ємність 2500 пФ. Щоб обмежити будь-які відбиття, що виникають під час наростання та спаду

сигналу, максимальний нахил (градієнт) сигналу обмежений 30 В/мкс. Зауважте, що цей допустимий градієнт також залежить від швидкості передачі даних. Високочастотні сигнали значно погіршуються, якщо передаються на великі відстані. У таблиці 2.1 ілюструється кореляція швидкості передачі даних між довжиною кабелю.

Таблиця 2.1 - Швидкість передачі даних EIA-232 у залежності від довжини кабелю

Швидкість передачі даних	Максимальна довжина кабелю [м]
19200	15
9600	150
4800	30
2400	900

Стандарт EIA-232 – це, по суті, асинхронний метод послідовного зв'язку, де потік бітів даних не надсилається протягом суворих часових рамок. На рис. 2.7 показано потік послідовних даних, що передаються через EIA-232. Зверніть увагу, що вихідний потік даних має бути доповнений стартовими, стоп-бітами та бітами виявлення помилок, щоб гарантувати правильну передачу даних. Кожен символ даних починається з біта уваги, який також відомий як стартовий біт. Біти даних йдуть безпосередньо за цим стартовим бітом. Значення біта 1 (TRUE) викликає негативні рівні напруги, тоді як значення біта 0 (FALSE) представлене як пробіл. Для цілей виявлення помилок можна включити (парний або непарний) біт парності. Нарешті, переданий потік даних завершується стоп-бітом.

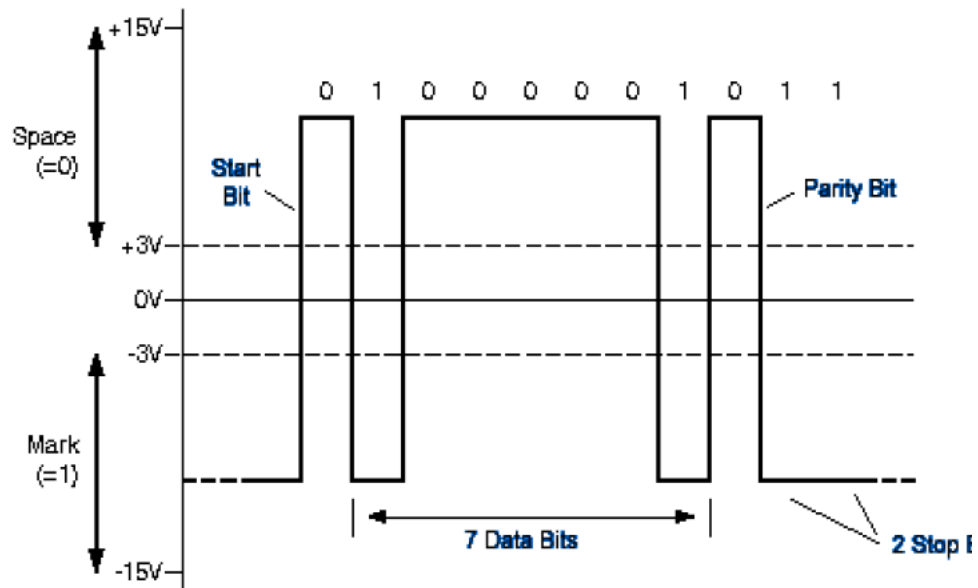


Рисунок 2.7 - Потік EIA-232.

ОВНС – 422

На відміну від EIA-232, EIA-422 (раніше відомий як RS 422) дозволяє багатоточкове підключення драйвера до (максимум) десяти приймачів. Швидкість передачі даних може бути збільшена до 35 Мбіт/с з (максимальною) довжиною кабелю 1200 м. Як зазначалося раніше, вищі швидкості передачі даних зазвичай вимагають зменшення довжини кабелю.

ОВНС – 485

EIA-485 (RS 485) – це найпоширеніший комунікаційний інтерфейс для багатьох промислових систем збору даних та зв'язку. Стандарт TIA/EIA-485 визначає напівдуплексний та диференціальний зв'язок. Теоретично можливі 32 вузли. Згідно з цим стандартом, драйвер повинен забезпечувати мінімальну вихідну напругу 1,5 В. Загальні рівні напруги можуть коливатися від -7 В до 12 В. Зверніть увагу, що цей метод зв'язку має високу стійкість до шуму.

CAN

Контрольована мережа (CAN) була впроваджена компанією Bosch у 1980 році для автомобільних застосувань. Вона мала замінити складну

проводку сигнальних кабелів та зменшити її до двопровідної шини. Фактично, CAN застосовується для приладів, які потребують великої кількості невеликих повідомлень за короткий період. На відміну від інших протоколів зв'язку на основі адрес, CAN є системою, що базується на повідомленнях, і особливо корисна, коли потрібна узгодженість даних у всій системі.

У CAN кожен вузол на шині зчитує ідентифікатор, а потім вирішує, чи читати решту повідомлення. Коли вузол хоче передати повідомлення, він повинен порівняти свій відносний пріоритет з пріоритетом мережевого повідомлення. Якщо він менший або дорівнює важливості, він повинен чекати, поки шина звільниться. ISO 11898 (стандарт CAN) описує диференціальну та напівдуплексну передачу даних. У цьому стандарті максимальна довжина кабелю становить 40 м, і на шині дозволено максимум 30 вузлів. Зверніть увагу, що швидкість передачі сигналу може сягати 1 Мбіт/с. У CAN також використовується один екранований або неекранований кабель витой пари з опором 120 Ом.

Протоколи зв'язку

Протокол зв'язку вважається набором стандартних правил для представлення даних, сигналізації, автентифікації та виявлення помилок, необхідних для передачі інформації через канал зв'язку. Протоколи зв'язку для цифрового комп'ютерного мережевого зв'язку мають функції, призначені для забезпечення надійного обміну даними через недосконалий канал зв'язку. Протокол зв'язку в основному дотримується певних правил, щоб система працювала належним чином. Далі розроблено популярні протоколи зв'язку.

TCP/IP

Протокол керування передачею/Інтернет-протокол (TCP/IP) також відомий як Інтернет-протокол, хоча його розробка почалася в 1960-х роках. TCP/IP спочатку розглядався для операційної системи UNIX. Він був

розроблений для забезпечення надійності мережі та автоматичного відновлення після збою будь-якого пристрою в мережі.

Більше того, це дозволяє будувати дуже великі мережі, які потребують мінімального централізованого управління.

TCP працює на транспортний шар, середній рівень у семерівневій еталонній моделі OSI (взаємозв'язок відкритих систем). Цей рівень відповідає за підтримку надійного наскрізного зв'язку в мережі. IP, навпаки, є мережа шар протокол, який є рівнем одразу під транспортним рівнем.

З іншого боку, IP-протокол має справу лише з пакетами (тобто найфундаментальніша одиниця передачі даних TCP/IP), TCP дозволяє двом хостам встановлювати з'єднання та обмінюватися потоками даних. TCP гарантує доставку даних, а також гарантує, що пакети будуть доставлені в тому ж порядку, в якому вони були надіслані.

MODBUS

Modbus — це протокол послідовного зв'язку типу «головний-підлеглий», запропонований компанією Modicon. Він був вперше випущений у 1979 році. Більшість пристроїв Modbus використовують фізичний інтерфейс зв'язку EIA-485. Але формат повідомлень Modbus не залежить від типу використовуваного фізичного інтерфейсу. Кожне повідомлення Modbus має чотири основні елементи, які представлені в таблиці 2.2.

Таблиця 2.2 - Основні елементи повідомлень Modbus

Поле	Опис
Адреса пристрою	Адреса одержувача
Код функції	Код, що визначає тип повідомлення
Дані	Блок даних
Перевірка на помилки	Перевірка на наявність помилок зв'язку

Оскільки Modbus є протоколом головний-підлеглий, польовий пристрій (підлеглий вузол) не може передавати будь-які дані, якщо їх не запитує головний вузол. Рис. 3.8 зображує загальний кадр повідомлення Modbus.

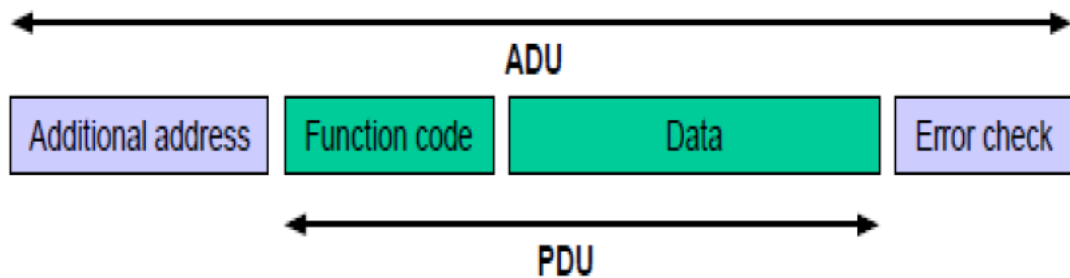


Рисунок 2.8 - Загальний модуль Modbus.

MACRO.

Макрос – це стандарт зв'язку для розподіленого керування машинами. Він розшифровується як Motion and Control Ring Optical (оптичне кільце керування рухом та керуванням) і в першу чергу розроблений для кількох головних та ведених вузлів. Зв'язок завжди ініціюється головним вузлом, надсилаючи пакет даних з відповідною адресою (рис. 2.9). Наступний вузол отримує потік даних та перевіряє адреса перевіряє, чи збігаються дані адреси з його власною локальною адресою. Потім вузол бере дані з пакета та пропускає його через кільце. Якщо локальна адреса відрізняється, вузол просто передає дані без зчитування. Зрештою, дані повертаються головному вузлу.

SERCOS

SERCOS (або послідовна система зв'язку в реальному часі) – це цифрова шина керування рухом. Це відкритий інтерфейс між контролером та інтелектуальним цифровим пристроєм, призначений для високошвидкісної послідовної передачі стандартизованих даних із замкнутим циклом (у реальному часі) через (завадостійке) волоконно-оптичне кільце (SERCOS I та II) або промисловий кабель Ethernet (SERCOS III). Загальний інтерфейс SERCOS показано на рис. 2.10.

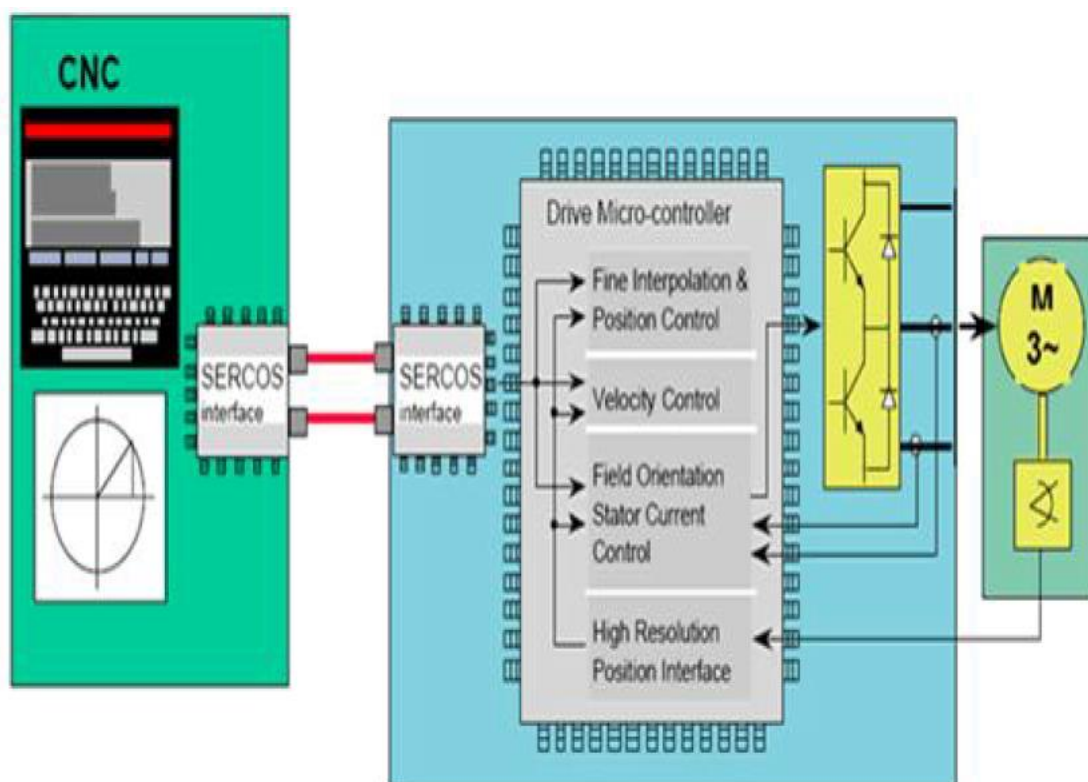


Рисунок 2.9 - Кільцева мережа MACRO.

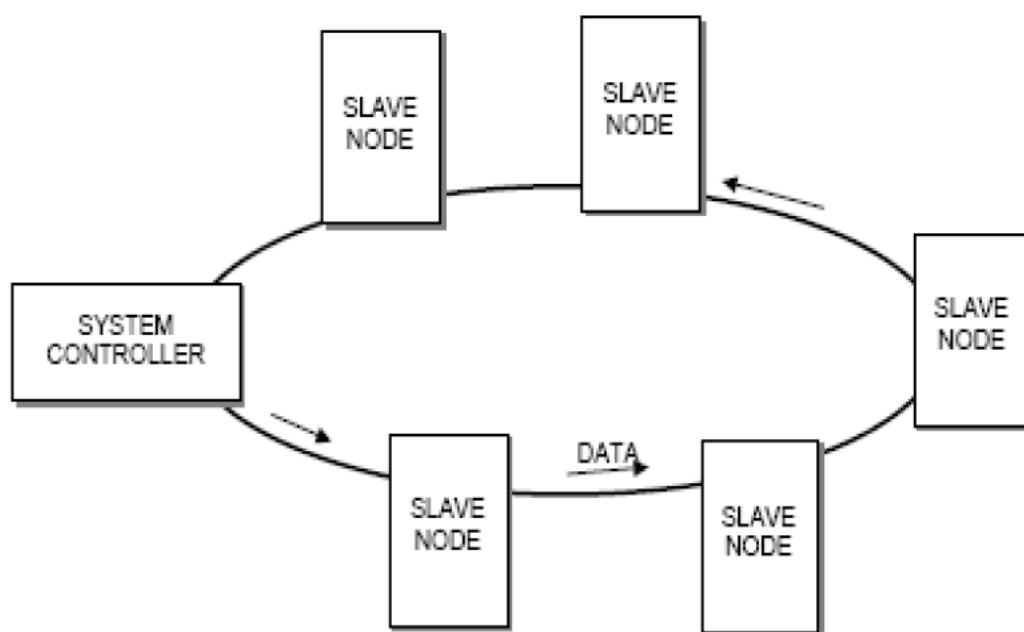


Рисунок 2.10 - Інтерфейс SERCOS.

Розроблені протоколи зв'язку.

Використовуючи комунікаційні інтерфейси EIA-232, повнодуплексний EIA-422 та EIA-485, можна розробити кілька комунікаційних протоколів. Спочатку в мережі керування будуть застосовані двонаправлені підходи. Потім будуть використовуватися однонаправлені протоколи.

Двонаправлений протокол.

Комунікаційний рівень програмного забезпечення керує мережею зв'язку. Він перетворює інтерпольовані команди для кожної осі на двобайтові дані та запускає потік зв'язку. Зауважте, що цей потік можна розглядати як повідомлений, так і адресний, оскільки транзакція запускається передачею даних, тоді як кожен контролер (вузол) у мережі отримує дані у відповідній позиції (адресі) переданого масиву.

Масив даних складається з 13 байтів, як показано на рис. 2.11. Він може обробляти дуже обмежену кількість розподілених пристроїв, з'єднаних між собою через комунікаційний інтерфейс EIA-232.

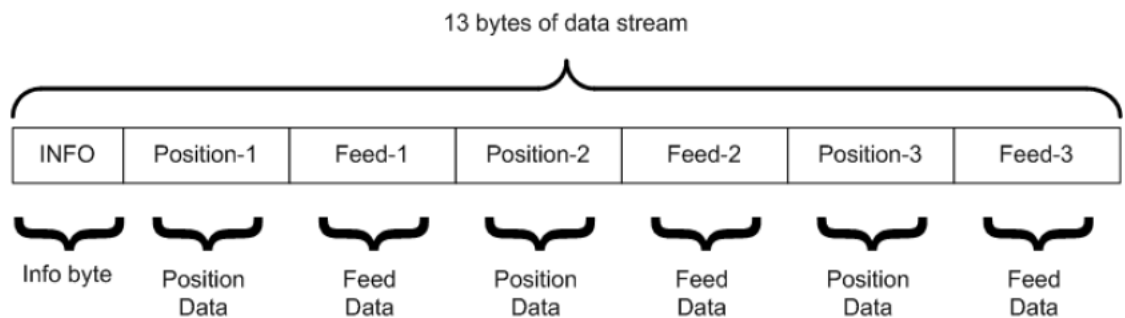


Рисунок 2.11 - Масив даних двонаправленого протоколу

Коли потік отримано, всі контролери отримують 13 байтів. Відповідно до інформаційного байта (першого байта потоку), контролер визначає режим роботи. Відповідні параметри команди (позиції) адресуються в масиві байтів. Зауважте, що весь шлях інструменту верстата визначається програмою обробки деталей з ЧПУ. Траєкторія інструменту обробляється та поділяється

на невеликі відрізки ліній, які визначаються позиційними приростами, припускаючи, що швидкість вздовж певного відрізка («швидкість подачі») є постійною. Фізична конфігурація протоколу проілюстрована на рис. 2.12.

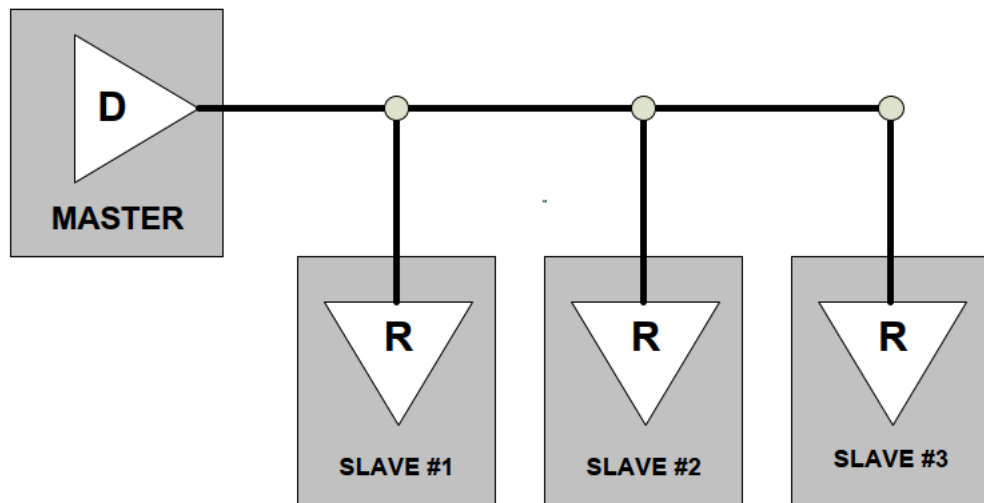


Рисунок 2.12 - Фізична конфігурація двонаправленого протоколу

Як зазначалося раніше, цей зв'язок є двонаправленим. Отже, ведені вузли не можуть відповідати на жодні повідомлення. Це серйозний недолік програм керування рухом, оскільки інтерфейс користувача потребує інформації про стан ведених вузлів (контролерів осей). Крім того, збільшення кількості пристроїв також призведе до збільшення навантаження мережі. Іншими словами, якість системи (QoS) знижуватиметься зі збільшенням кількості ведених вузлів. Експериментальні результати та обмеження. Протокол зв'язку показано на рис. 2.13. Зверніть увагу, що бажана смуга пропускання системи керування положенням у рамках цього дослідження становить 100 Гц. У цьому випадку алгоритм керування має виконуватися за 10 мс. Згідно з цією оцінкою, збільшення навантаження мережі різко погіршує якість роботи (QoP).

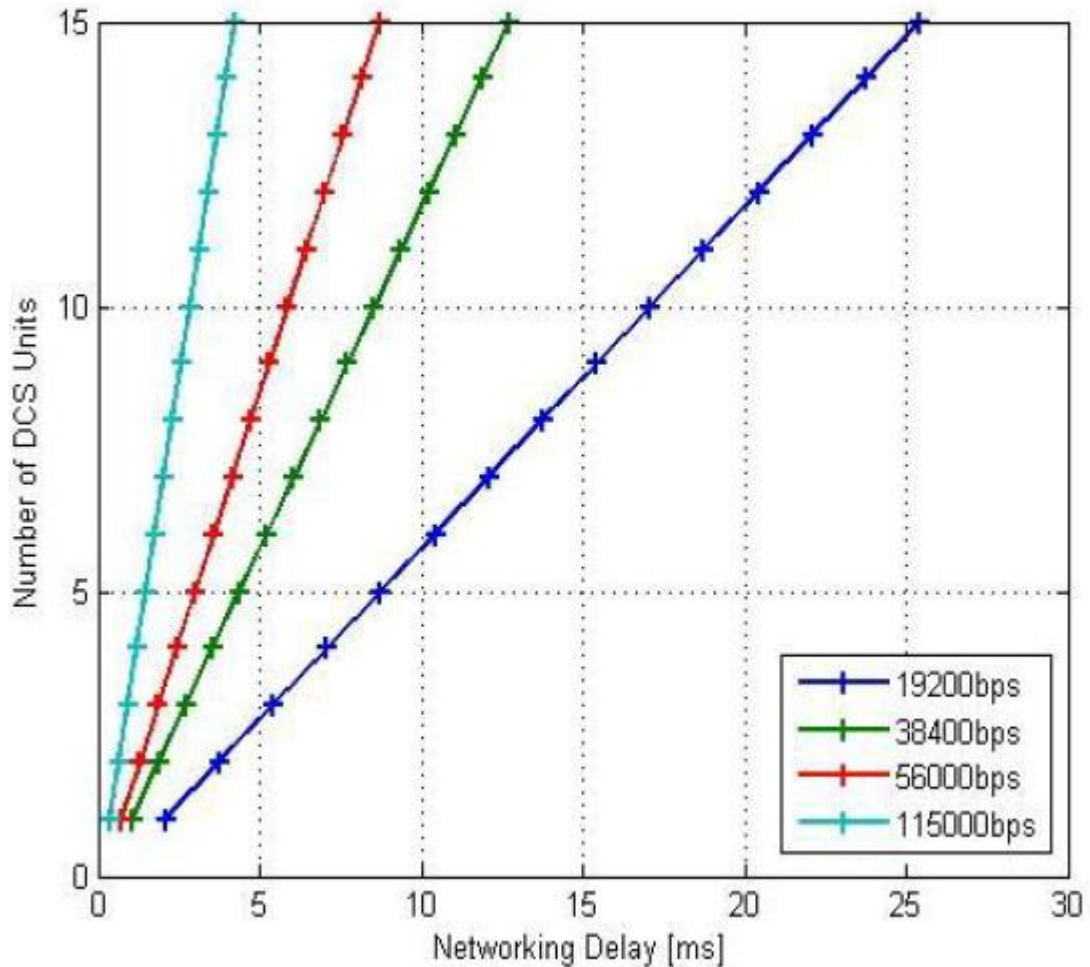


Рисунок 2.13 - Оцінка двонаправленого протоколу

Головною перевагою цього протоколу є його надзвичайна простота реалізації. Для виконання достатньо лише одного потоку даних. Визначення адрес фіксовані, тому немає потреби в додаткових індикаторах для звернення до таких контролерів. З іншого боку, з іншого боку, збільшення кількості розподілених контролерів також збільшить кількість байтів у потоці даних. Отже, якість продуктивності (QoP) погіршуватиметься зі збільшенням кількості контролерів. Крім того, це негнучко, оскільки кожен контролер має різну прошивку для вилучення відповідної частини з переданих даних.

Однонаправлений протокол.

Однонаправлений протокол зв'язку забезпечує зв'язок між головним та підлеглим пристроями, де їхній фізичний інтерфейс зв'язку описано на рис. 2.14.

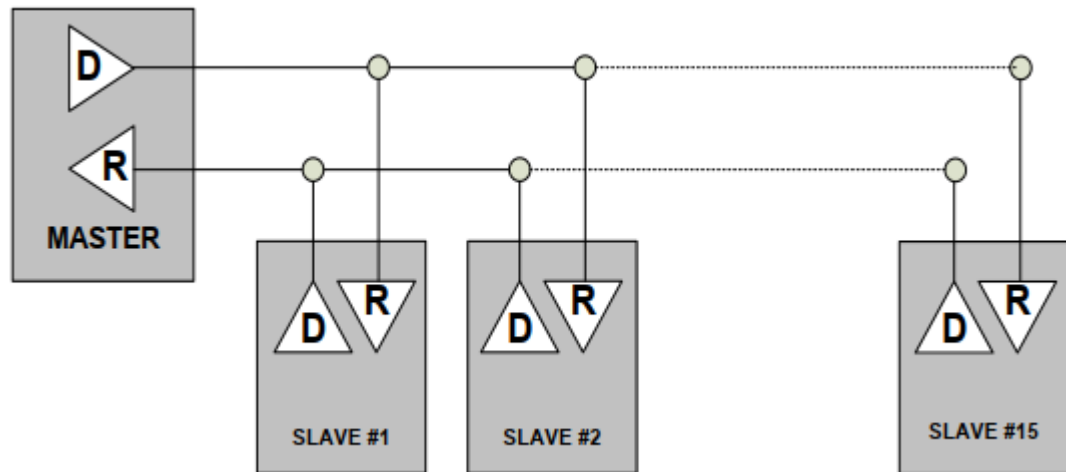


Рисунок 2.14 - Фізична конфігурація для односпрямованого протоколу

Використовуючи повнодуплексний зв'язок EIA-485, розподілені пристрої можуть відповідати на деякі запити, що передаються головним вузлом. За визначенням, цей протокол обмежений 15 розподіленими контролерами. Кожен блок забезпечує ідентичну прошивку з адресою блоку, яку вибирає користувач. Це суворо визначений (адресний) протокол зв'язку, який використовує власний синтаксис.

Протокол використовує лише два байти даних. Використання буфера, який періодично потрібно заповнювати командами опорного положення, знижує продуктивність системи. Відповідно, було протестовано різні швидкості зв'язку, щоб визначити необхідний час для заповнення буфера. Результати наведено на рис. 2.15. Поки система виконує попередньо визначені команди в буфері, кількість вузлів у системі не впливає на продуктивність. Хост передає сигнал лише тоді, коли очікує запит на синхронізацію. Це становить лише 3 байти даних. Припускаючи, що період дискретизації контролера становить 100 Гц, ефект програмної синхронізації

описано на рис. 2.16. Зверніть увагу, що це надійний протокол, оскільки підлеглі вузли можуть взаємодіяти лише з головним вузлом. Тому протокол можна класифікувати як протокол "головний-підлеглий".

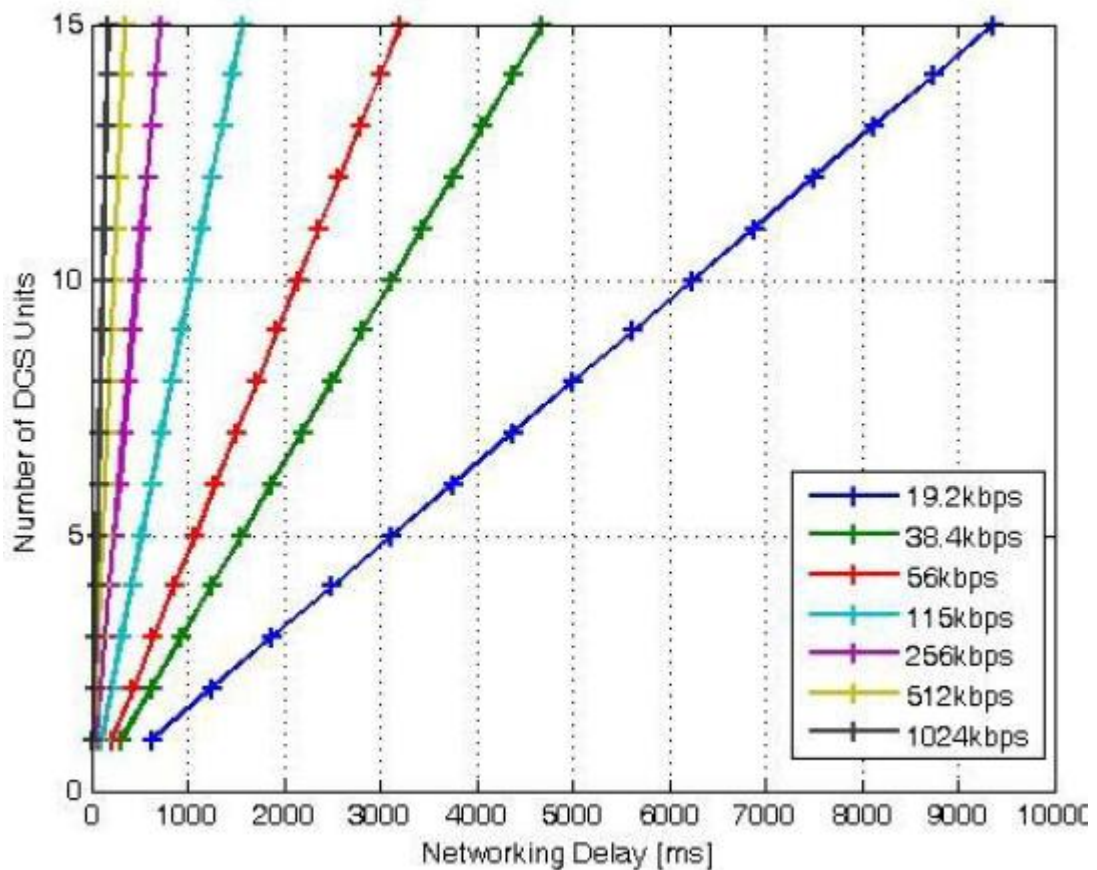


Рисунок 2.15 - Оцінка однонаправленого протоколу

Що стосується переваг, то це простий (але гнучкий) протокол зв'язку, який використовує зв'язок головний-підлеглий. Підлеглі пристрої можуть відповідати на запити, передані головним вузлом. Збільшення кількості розподілених вузлів не впливає ні на якість обслуговування (QoS), ні на якість обслуговування (QoP). Крім того, кожен контролер (підлеглий вузол) має ідентичну прошивку.

З іншого боку, у цьому протоколі може використовуватися обмежена кількість пристроїв (до п'ятнадцяти). Підлеглі вузли не можуть передавати

жодних даних, якщо цього не запитує головний вузол. Крім того, підлеглі пристрої не можуть спілкуватися один з одним.

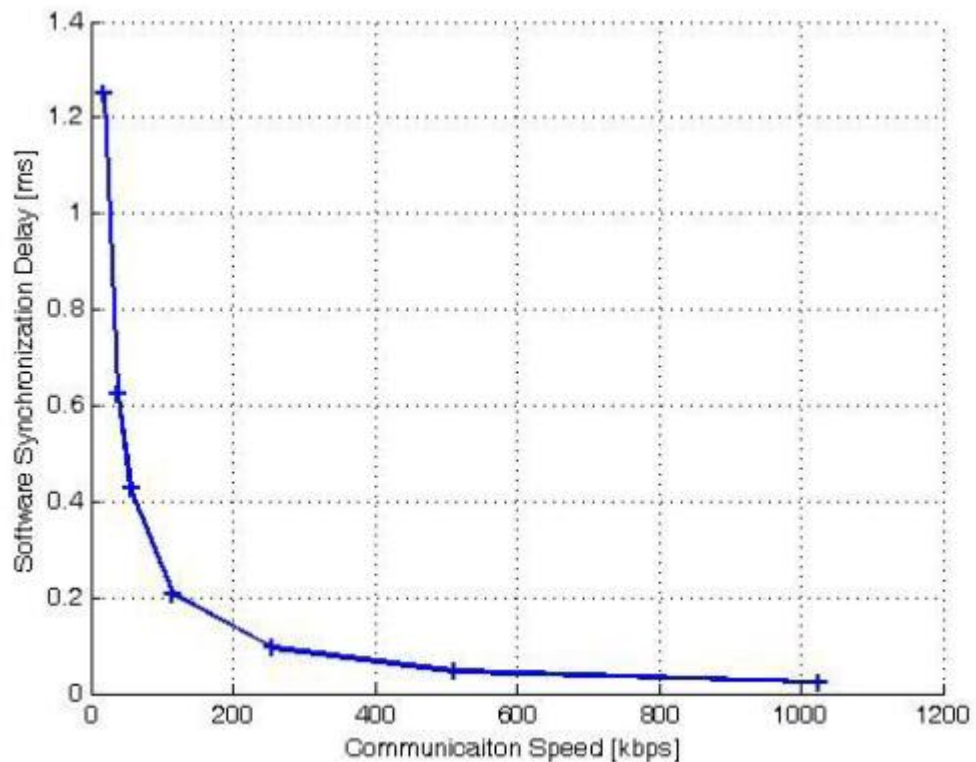


Рисунок 2.16 - Оцінка затримки синхронізації програмного забезпечення

Однонаправлений протокол зі зв'язком між відомими пристроями

У цій роботі розроблений контролер має реконфігурований комунікаційний інтерфейс. За допомогою перемичок на контролерах, комунікаційний інтерфейс EIA-485 можна налаштувати як напівдуплексний або повний дуплексний. Встановлення комунікаційного інтерфейсу в напівдуплексний режим дозволяє здійснювати зв'язок між веденими пристроями. Це вдосконалення реалізує ідею зв'язку між кількома ведучими пристроями. Діаграма такої схеми показана на рис. 2.17. Як зазначалося раніше, цей протокол має свої проблеми, такі як колізія даних під час

передачі, оскільки лінія передачі даних є однаковою як для прийому, так і для передачі.

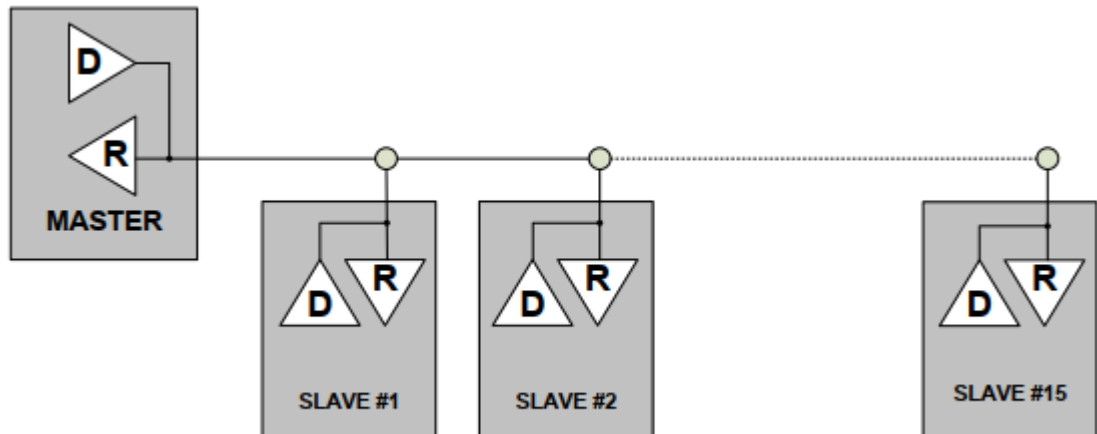


Рисунок 2.17 - Фізична конфігурація односпрямованого протоколу зі зв'язком між підлеглими

Схема розподіленого контролера.

Під час роботи над дисертацією було розглянуто різні реалізації контролерів. Для визначення найкращого проекту розподіленого контролера було розроблено та реалізовано три моделі контролерів (інтерфейси зв'язку, протоколи). У цьому розділі підсумовано зусилля з розробки.

Розроблений протокол зв'язку відхиляє транзакції повідомлень, ініційовані веденими вузлами. Крім того, цей метод може бути застосовним, коли використовується лише обмежена кількість пристроїв у розподіленій мережі керування. Немає потреби в додатковій синхронізації між веденими вузлами. Команди опорного положення, які генеруються головним ПК, також можуть служити сигналом синхронізації програмного забезпечення. З метою синхронізації на ПК повинна бути реалізована операційна система реального часу. Псевдокод, що реалізує цей код, наведено нижче:

Таблиця 2.3 - Псевдокод для методу А

```

Void main () {
    // initialization and get unit ID
    ID = InitializeController();
    While(TRUE) {
        If(serial_is_data_in()){
            // read transferred message and encode it
            Data = Read_serial_data(ID);
            // execute transferred message
            Execute(Data);
        }
    }
}

```

Можна помітити, що в методі А дані про положення спочатку створюються у вигляді невеликих відрізків ліній. Отже, менші відрізки збільшують точність поступового руху, одночасно збільшуючи навантаження на мережу. Зауважте, що EIA-232 підтримує швидкість передачі даних до 115 кбіт/с при використанні 15 вузлів. Як показано на рис. 3.18, збільшення частоти дискретизації системи керування значно скорочує час виконання, що залишився для алгоритму контролера. Отже, ця ситуація розширює межі можливостей, і зрештою система не може відтворити жодної реакції.

Метод Б

Метод Б розроблено як альтернативну архітектуру контролера для розподіленого керування положенням. Він має комунікаційний рівень, інтерполяційний рівень та рівень керування положенням. На відміну від інших методів, він включає кілька мікрокомп'ютерів, які призначені для обробки кожного рівня. Шина даних з'єднує всі мікрокомп'ютери один з одним через комунікаційний інтерфейс SPI. Програмне забезпечення ПК розміщує розподілену комунікаційну мережу. Комунікаційна мережа базується на фізичному інтерфейсі EIA-232.

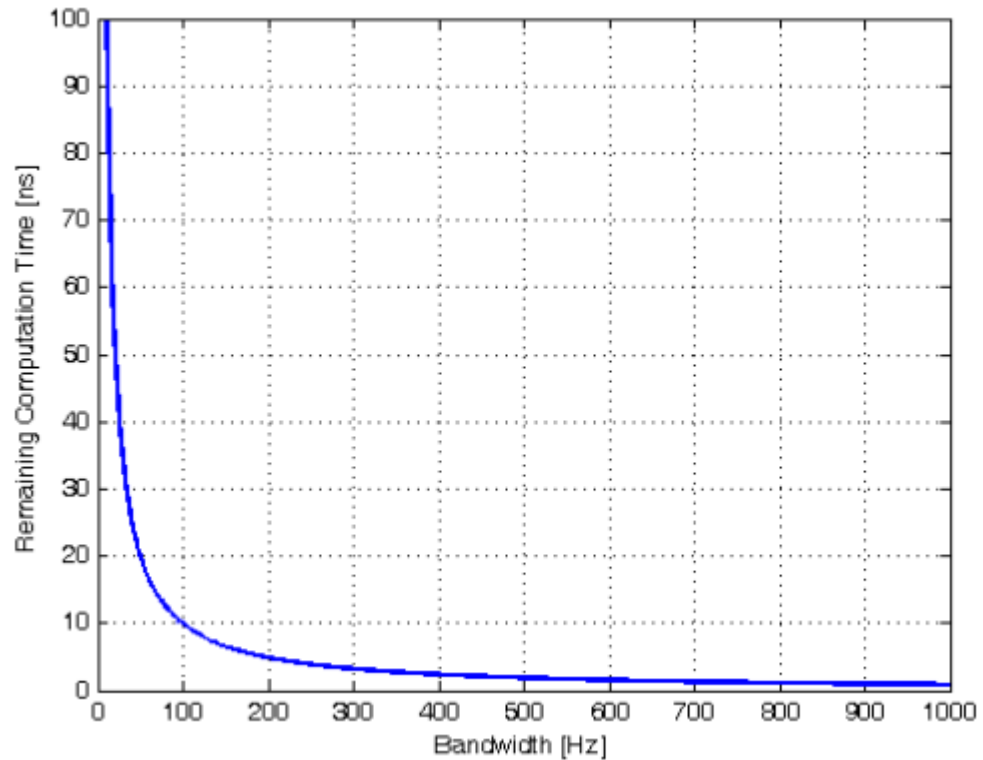


Рисунок 2.18 - Залишок обчислювального часу

Головною метою методу Б, псевдокод якого наведено в таблиці 2.5, є зменшення потреб у зв'язку та оптимізація між часом обчислення та продуктивністю зв'язку. Зверніть увагу, що всі переміщення, які виконуються за допомогою виконавчі механізми можна моделювати як малі відрізки ліній та дуг. На відміну від методу А, який використовує лише малі відрізки ліній, метод В включає рівняння різниці з постійними коефіцієнтами (CCDE) заданих рухів. Рухи верстатів, які зазвичай описуються рівняннями (2.1) та (2.2) (для кругових шаблонів) та рівняннями (2.3) та (2.4) (для лінійних шаблонів), включають постійні параметри (A, B, ..., G), які потім передаються ПК як командні сигнали.

$$x(k) = A \cdot x(k-1) - B \cdot y(k-1) \quad (2.1)$$

$$y(k) = C \cdot x(k-1) + D \cdot y(k-1) \quad (2.2)$$

$$x(k) = E.x(k-1) + F \quad (2.3)$$

$$y(k) = G.y(k-1) + H \quad (2.4)$$

На завершення, програмне забезпечення ПК (головний вузол) виконує інтерпретацію команд позиціонування (тобто обчислює коефіцієнти CCDE), тоді як контролери (підлеглі вузли) безпосередньо обробляють інтерполяцію. Зверніть увагу, що підлеглі вузли можуть прослуховувати повідомлення головного вузла, але вони також можуть відповідати на його запити.

Таблиця 2.4 - Псевдокод для методу Б

```
#INT_SERIAL
Void Inquiry() {
    If(host_inquiries_position()) {
        Send_serial_data(Position0);
    }
}
Void main () {
    // initialization and get unit ID
    ID = InitializeController();
    While(TRUE) {
        If(serial_is_data_in()){
            // read transferred message and encode it
            Data = Read_serial_data(ID);
            // use previous position data and interpolate
            Position1 = Interpolate(Position0,Data);
            // execute transferred message
            Execute(Position1);
            // now data history is replaced
            Position0 = Position1;
        }
    }
}
```

Метод В

Метод В успадковує найкращі риси попередніх методів. Тобто, як і метод А, він використовує невеликі сегменти ліній. Як мережа

використовується звичайна польова шина EIA-485, яка контролюється головним ПК. Зазвичай ПК не мають послідовного комунікаційного інтерфейсу EIA-485. Тому можна інтегрувати карту послідовного зв'язку на базі PCI.

У цій техніці хост спочатку заповнює буфери контролерів і вибирає контролер як головний. Контролери координуються за допомогою програмної синхронізації. Коли отримується сигнал запуску, кожен контролер починає очищати свій буфер. Кожні дані, які займають певне місце в буфері, мають часовий індекс.

Цей часовий індекс контролюється як головним, так і підлеглим вузлами. Отже, головному вузлу не потрібна генерація додаткових запитів. Псевдокод для методу В наведено в таблиці 2.5.

Таблиця 2.5 - Псевдокод для методу В

```

Void main () {
    // initialization and get unit ID
    ID = InitializeController();
    k = 0;    // reset time index
    While(TRUE) {
        If(serial_is_data_in()){
            // read transferred message and encode it
            Data = Read_serial_data(ID);
            [Select, Master] = Implement_Protocol(Data);
            Switch(Select){
                Case 1: Fill_Buffer(Select);
                Case 2: Jog(Select);
                Case 3: Go_Home_Position();
                Case 4: Execute(Select);
                    //increase time index
                    ++k;
                    If(MASTER) Send_serial_data(OK());
            }
        }
    }
}

```

Головним внеском методу В є те, що кількість одиниць може бути обмежена лише пропускною здатністю комунікаційної мережі. Зауважте, що повторне заповнення буфера порушує безперервний процес обробки. Таким чином, використання циклічного буфера може покращити продуктивність системи. Відповідні переваги представлені на рис. 2.15 та 2.16.

3 КОСТРУКТОРСЬКА ЧАСТИНА

3.1 Архітектура апаратного забезпечення системи керування

Централізовані системи керування зазвичай використовують один (або декілька) потужних мікрокомп'ютерів, що виконують усі необхідні операції, такі як генерація команд, ведення обліку, багатоосьове (в режимі реального часу) керування рухом тощо. Функціональне використання цієї архітектури суворо визначене, а відповідне обладнання не допускає жодних помилок синхронізації.

Головною метою цього дослідження є розробка (практичного) розподіленого контролера, який використовує переваги високошвидкісної послідовної комунікаційної мережі. Під час цього дослідження створюється комунікаційна мережа, через яку розподіляються ідентичні блоки контролерів.

У роботі розроблено кілька архітектур розподілених контролерів. Розподілений контролер, що використовує інтерфейс EIA-232, представлено в Додатку А. Оскільки EIA-232, який має низьку завадостійкість, має низьку швидкість передачі даних, отримана топологія контролера не підходить для вимогливих застосувань. Тому розроблено новий контролер, який використовує інтерфейс EIA-485 (налаштовується як напівдуплексний/повний дуплекс). Атрибути цього контролера наведено нижче.

3.2. Архітектура апаратного забезпечення

Перш за все, розроблене обладнання (системи керування) має працювати спільно з головним ПК або промисловим ПК (IPC). ПК з високорівневим програмним забезпеченням керує мережею зв'язку та виконує відповідні функції інтерполяції. Зауважте, що в сучасних технологіях модулі САМ (автоматизоване виробництво) можуть легко генерувати програми обробки деталей з ЧПУ. Замість перетворення даних CAD у програми обробки деталей з ЧПУ, програми САМ можуть безпосередньо генерувати відповідні команди позиціонування для розподілених контролерів. Отже, єдиною метою розподіленого контролера є реалізація передових алгоритмів керування рухом синхронізованим чином.

Розроблений контролер оснащений двома процесорами: PIC16F88 та dsPIC30F4011 від Microchip. dsPIC30F4011, що містить RISC-процесор, має широкий спектр периферійних інтерфейсів. Цей 16-бітний цифровий сигнальний контролер спеціально розроблений для керування двигунами та промислового застосування. Деякі з важливих особливостей цього чіпа наведено нижче:

- Швидкість роботи 30 MIPS,
- 2048 байт SRAM,
- каналів 10-бітного (1 Msps) аналого-цифрового (АЦП) перетворювача
- модулі UART з буферами FIFO,
- 1 CAN-модуль (сумісний з CAN 2.0B),
- 3-провідні модулі послідовного периферійного інтерфейсу (SPI) (підтримують чотири режими кадрів),
- 2-провідна шина I²C (підтримує режим кількох головних/підлеглих пристроїв та 7/10-бітну адресацію),
- Вихідні канали ШІМ,

Інтерфейс квадратурного енкодера з 16-бітним лічильником позицій вгору/вниз, 16-бітними функціями порівняння виходу.

PIC16F88 — це потужний 8-бітний мікроконтролер на базі CMOS-флеш-пам'яті. Він має наступне:

Внутрішній генератор 8 МГц,
256 байт пам'яті даних EEPROM,
захоплення/порівняння/ШИМ, адресний USART,
синхронний послідовний порт, який можна налаштувати як 3-провідний послідовний периферійний інтерфейс (SPI) або 2-провідну шину міжінтегральних схем (I²C),

7 каналів 10-бітного аналого-цифрового (АЦП) перетворювача

2 компаратори, що роблять його ідеальним для застосування з аналоговими/інтегральними рівнями в автомобільній, промисловій, побутовій та споживчій техніках.

Функціональна блок-схема розробленого обладнання наведена на рис. 3.1. У цій системі dsPIC30F4011 відіграє важливу роль у розробленій розподіленій системі керування. По-перше, він функціонує як комунікаційний інтерфейс. Його 2 КБ SRAM використовуються як буфер команд опорного сигналу. Як було показано в попередньому розділі, за допомогою заданого протоколу команди опорного сигналу, надіслані головним ПК, зберігаються в SRAM мікросхеми для подальшого виконання.

Важливо зазначити, що за допомогою інтерфейсу квадратурного енкодера dsPIC30F4011 можна виконувати замкнутий цикл керування електродвигунами (серводвигунами змінного та постійного струму). Розроблений контролер має інтерфейс для щіткових двигунів постійного струму, безщіткових двигунів постійного струму та крокових двигунів. Використовуючи аналого-цифрові (АЦП), можна реалізувати інтерфейс аналогових датчиків.

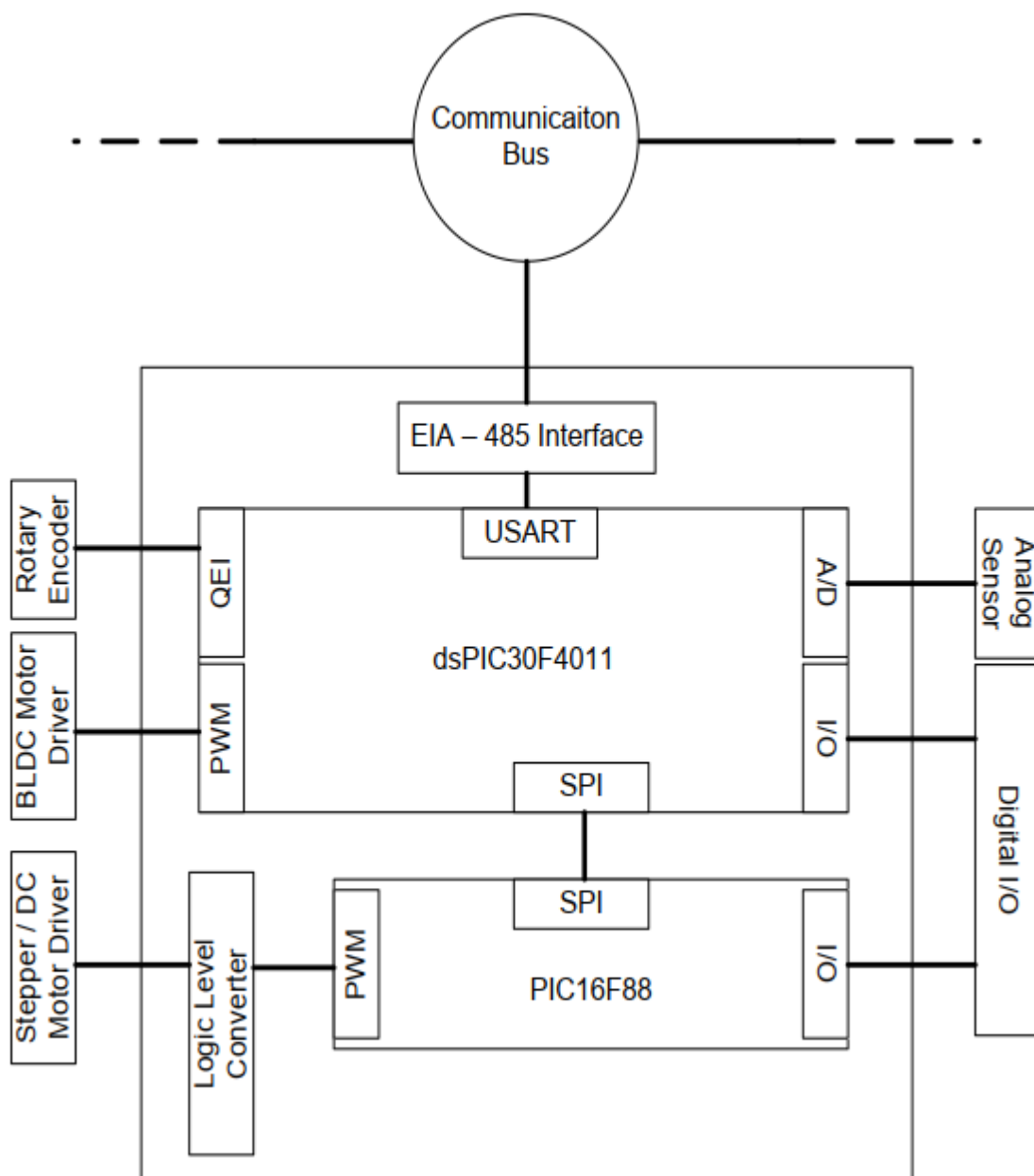


Рисунок 3.1 - Блок-схема апаратного забезпечення

Один з UART-ів dsPIC30F4011 підключений до приймачів EIA-485. Два приймачі MAX485 EIA-485 використовуються для взаємодії з контролерами та повнодуплексною мережею зв'язку. Це вдосконалення робить контролер частиною багатоточкової мережі зв'язку. Друкована плата розробленого контролера показана на рис. 3.2, а рис. 3.3, 3.4 та 3.5 ілюструють його принципові схеми.

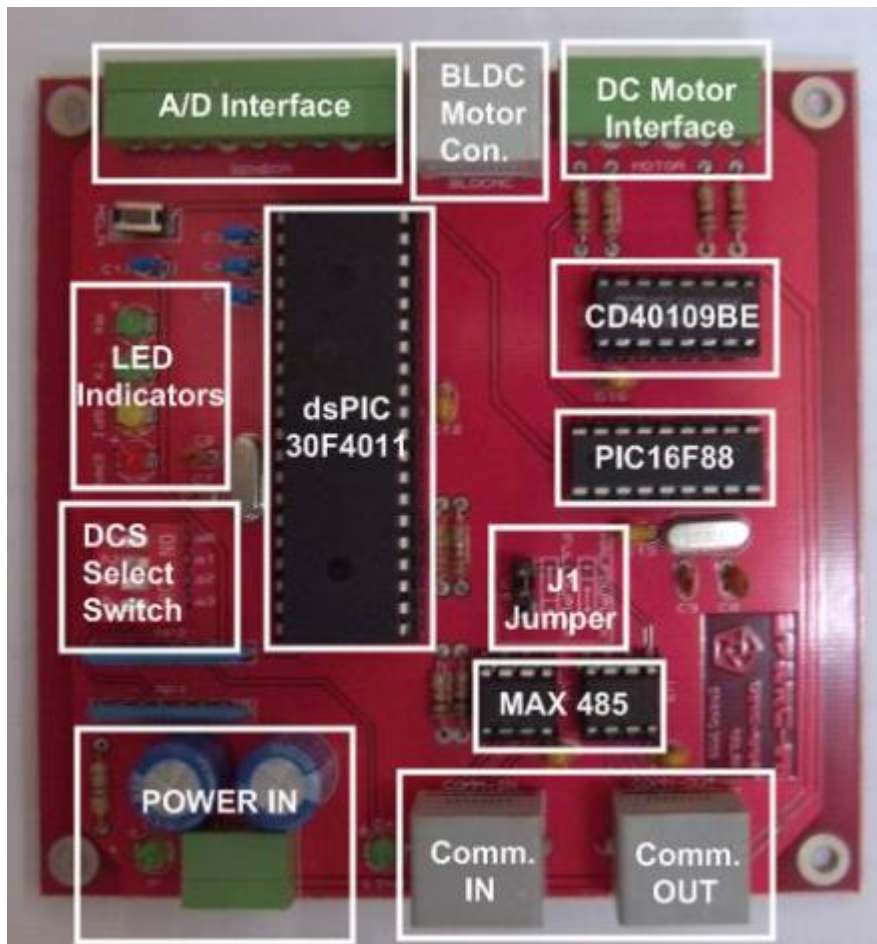


Рисунок 3.2 - Друкована плата контролера

Контролер може працювати з напівдуплексними або повнодуплексними мережами зв'язку. За допомогою перемички J1 на друкованій платі можна перемикається між режимами зв'язку. dsPIC30F4011 та PIC16F88 використовують інтерфейс зв'язку SPI для взаємодії, де dsPIC30F4011 керує зв'язком SPI.

Як показано на рис. 3.6, продуктивність контролера знижується зі збільшенням частоти. Фактично, під час тестування межі CD4019BE розширюються, і очікується, що він оброблятиме імпульсні сигнали на частоті 2 МГц - сигнал, який ніколи не буде згенерований самим контролером. Як видно на рис. 3.7 та 3.8, на CD40109 подається командний TTL-сигнал (який представлений як світла лінія на екрані осцилографа), і очікується, що він зсуне цей сигнал до рівня CMOS (10 В).

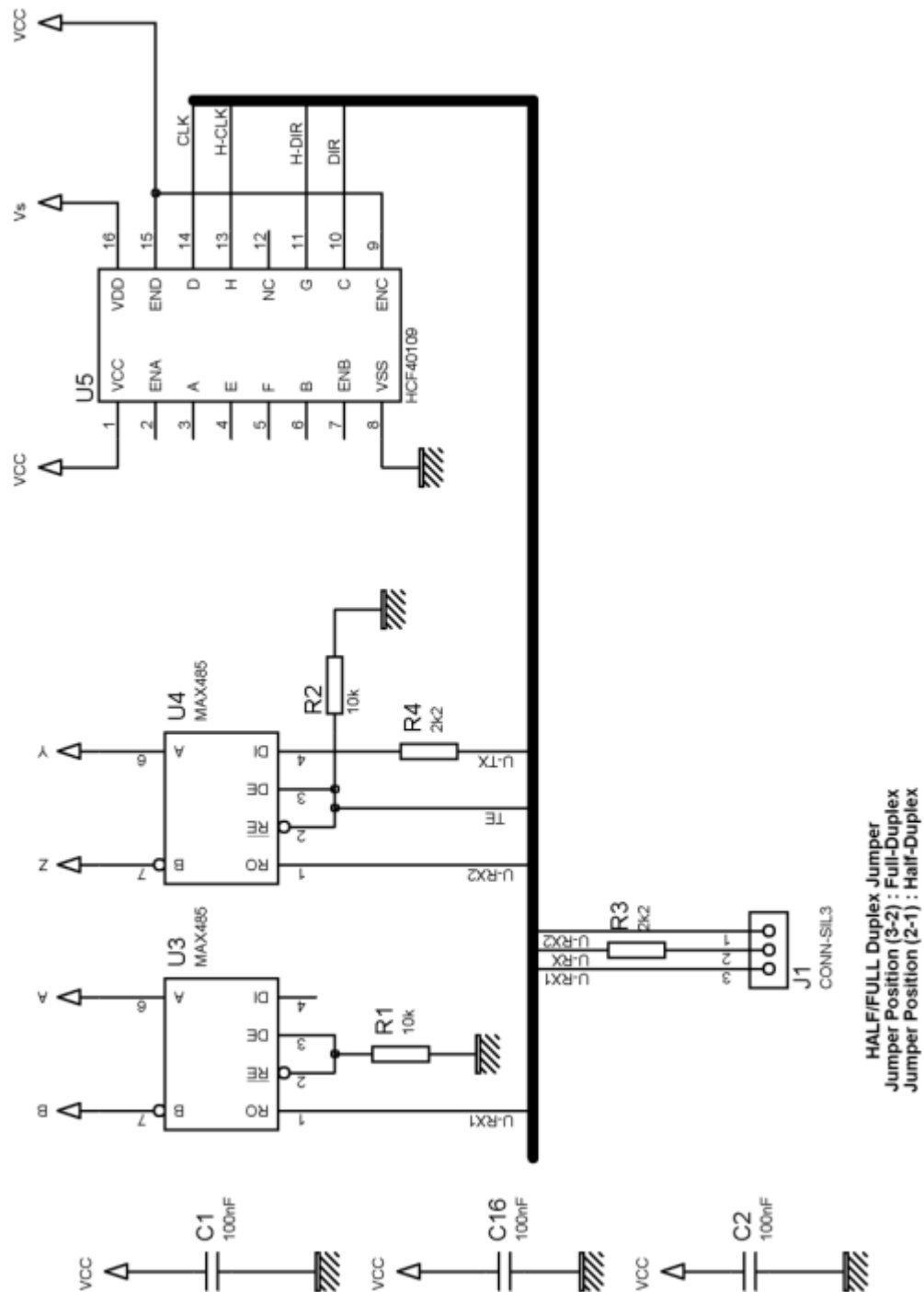


Рисунок 3.3 - Інтерфейс зв'язку

Зсунутий сигнал представлений як темна лінія на рисунках. Як чітко видно на рис. 4.7, 2 МГц тактового сигналу значно виходить за межі робочої зони CD40109BE.

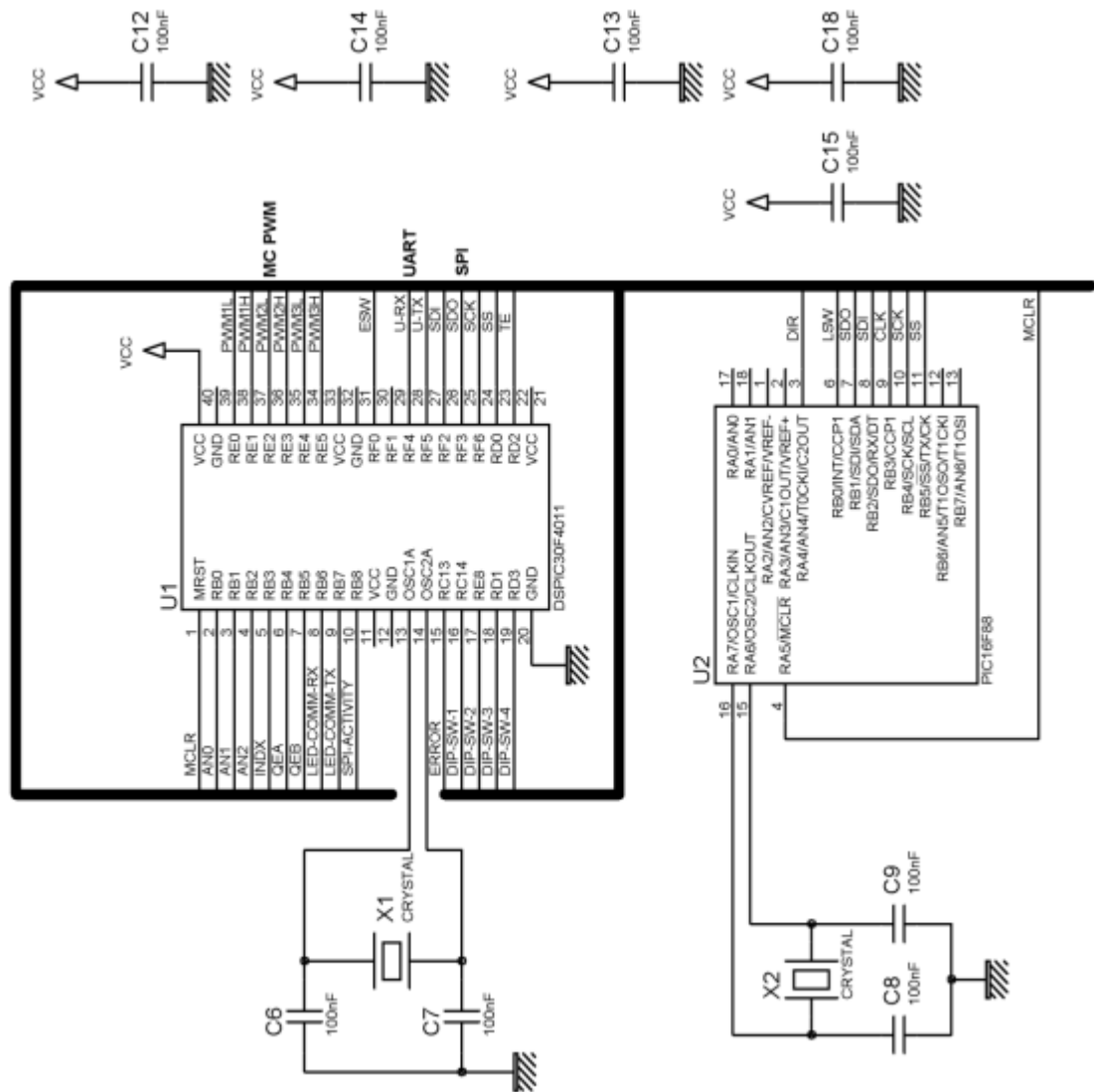


Рисунок 3.4 - Головний контролер

Щоб врахувати динамічну характеристику мікросхеми, тактова частота знижена до 100 кГц. Як показано на рис. 3.8, CD40109BE може належним чином зсунути ці сигнали до рівня 10 В. Таким чином, 100 кГц також вважається граничним значенням командного сигналу для цієї апаратної конфігурації.

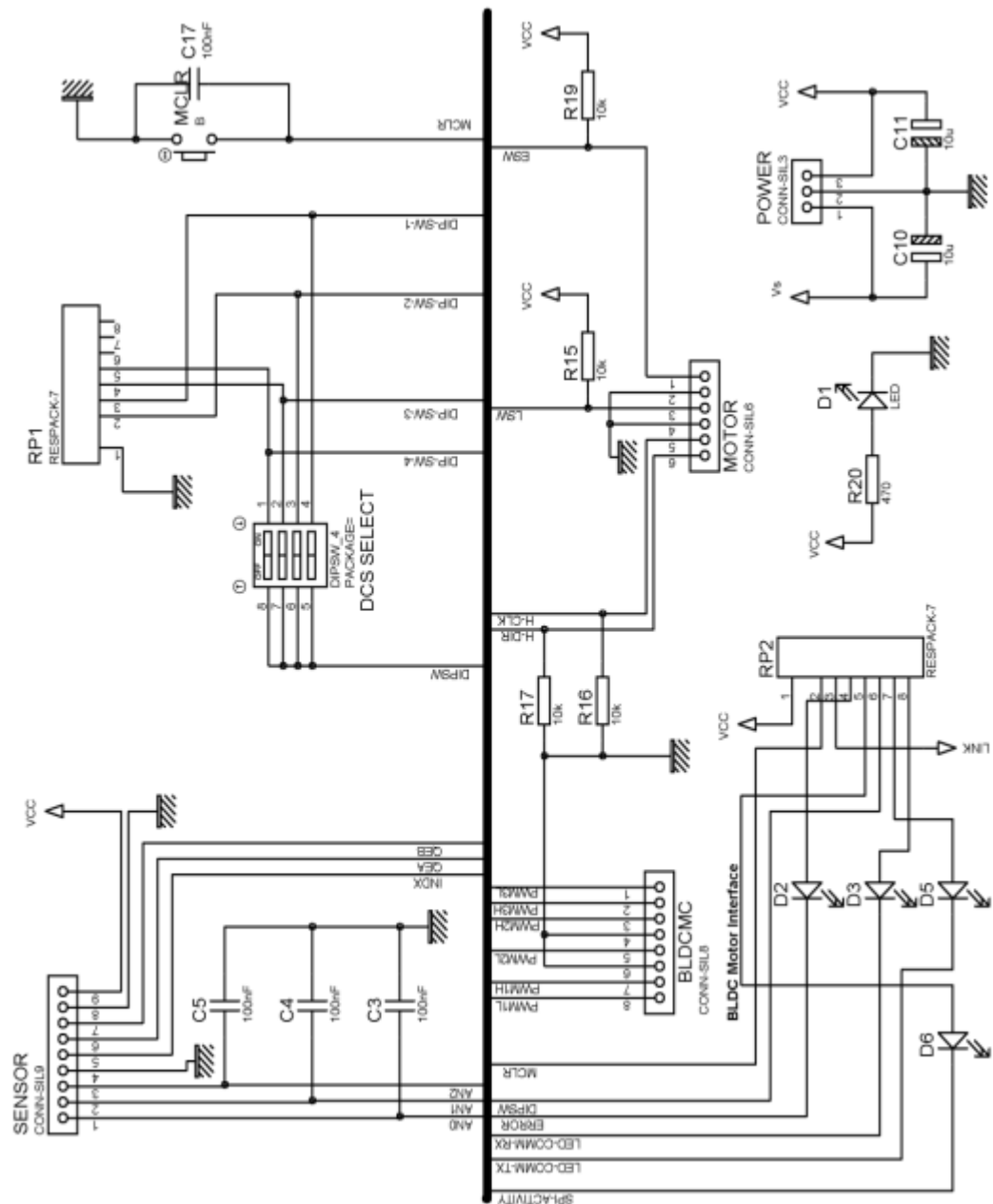


Рисунок 3.5 - Периферійні з'єднання

Зверніть увагу, що PIC16F88 призначений для генерації командних сигналів для драйверів крокових двигунів. Цей 8-бітний мікроконтролер значно зменшує обчислювальне навантаження на dsPIC30F4011. Тут CD40109BE (буфер) з'єднаний з PIC16F88. Для правильного з'єднання між контролером та драйвером двигуна, які працюють на різних логічних сигналах (CMOS, TTL). Містить схеми перемикання рівнів напруги від

низького до високого з можливістю роботи з трьома окремими вихідними станами. Перетворення рівнів сигналу широко використовується в інтерфейсних схемах.

На етапі проектування було проведено кілька експериментів з CD40109BE (квадратною КМОП-інтегральною схемою з перемикачем рівнів напруги від низького до високого). Тривалість процесу перетворення безпосередньо впливає на продуктивність системи керування. Для визначення продуктивності CD40109BE застосовуються різні частоти тактових сигналів та спостерігається реакція CD40109BE. Результати підсумовано на рис. 3.6.

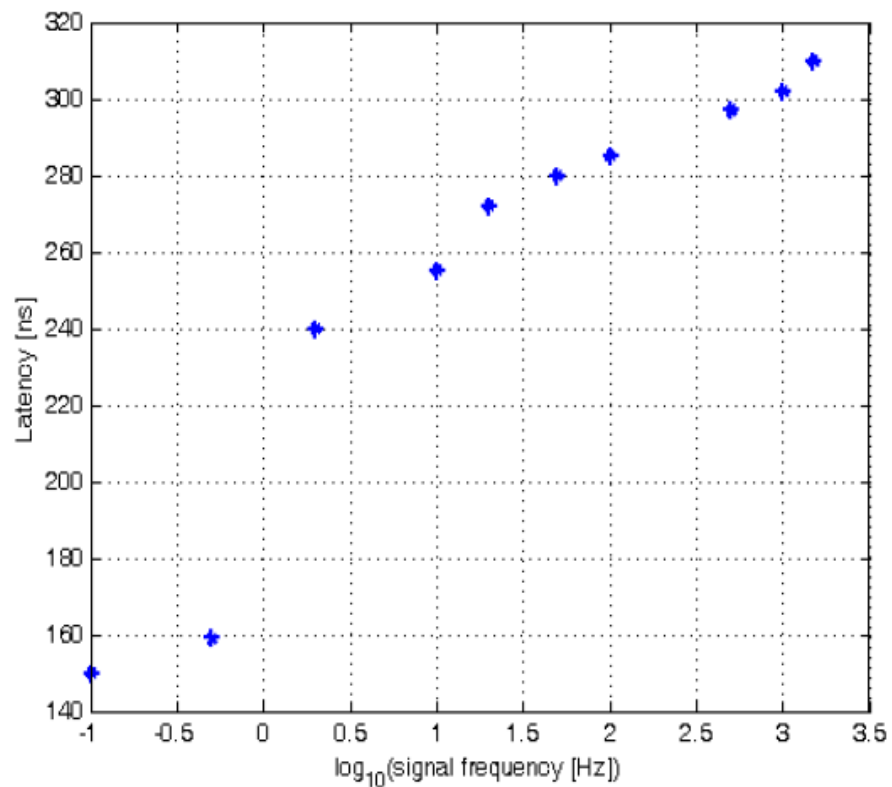


Рисунок 3.6 - Частота сигналу в порівнянні з затримкою

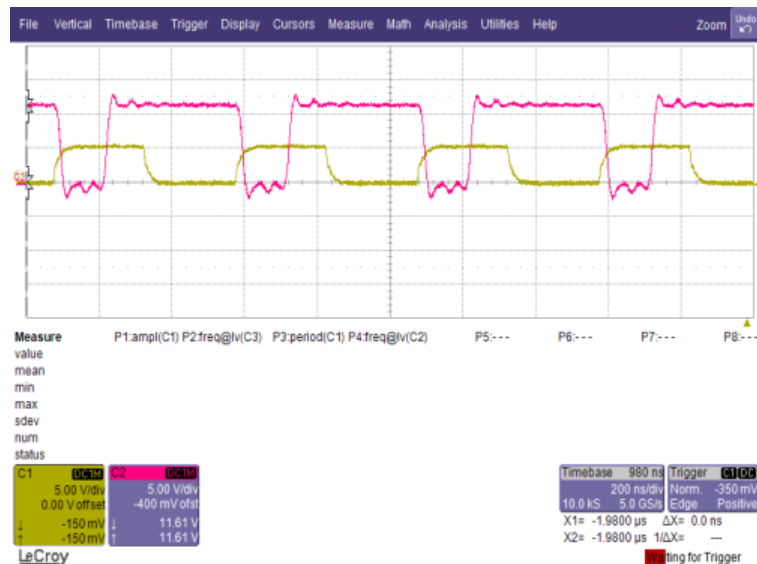


Рисунок 3.7 - Командний сигнал (2 МГц)

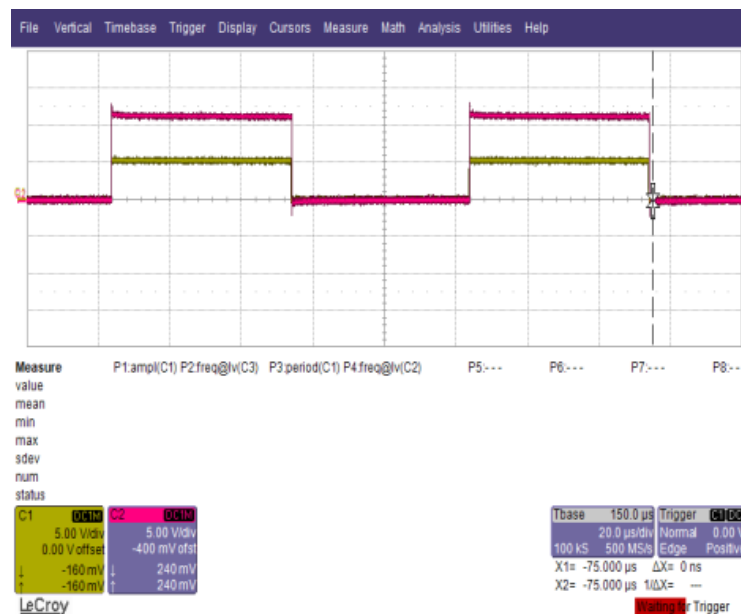


Рисунок 3.8 - Командний сигнал (100 кГц)

Прошивка

Прошивки для мікроконтролерів dsPIC30F4011 та PIC16F88 розроблені на базі середовища розробки CCS C IDE, яке представлено в розділі ДОДАТОК В. Також функціональні блок-схеми наведено на рис. 3.9. Як обговорювалося вище, програма починається з отримання одного байта,

елемент якого суворо визначений у протоколі зв'язку. Відповідно до цієї пари байтів, пристрій налаштовується на виконання запитуваних операцій.

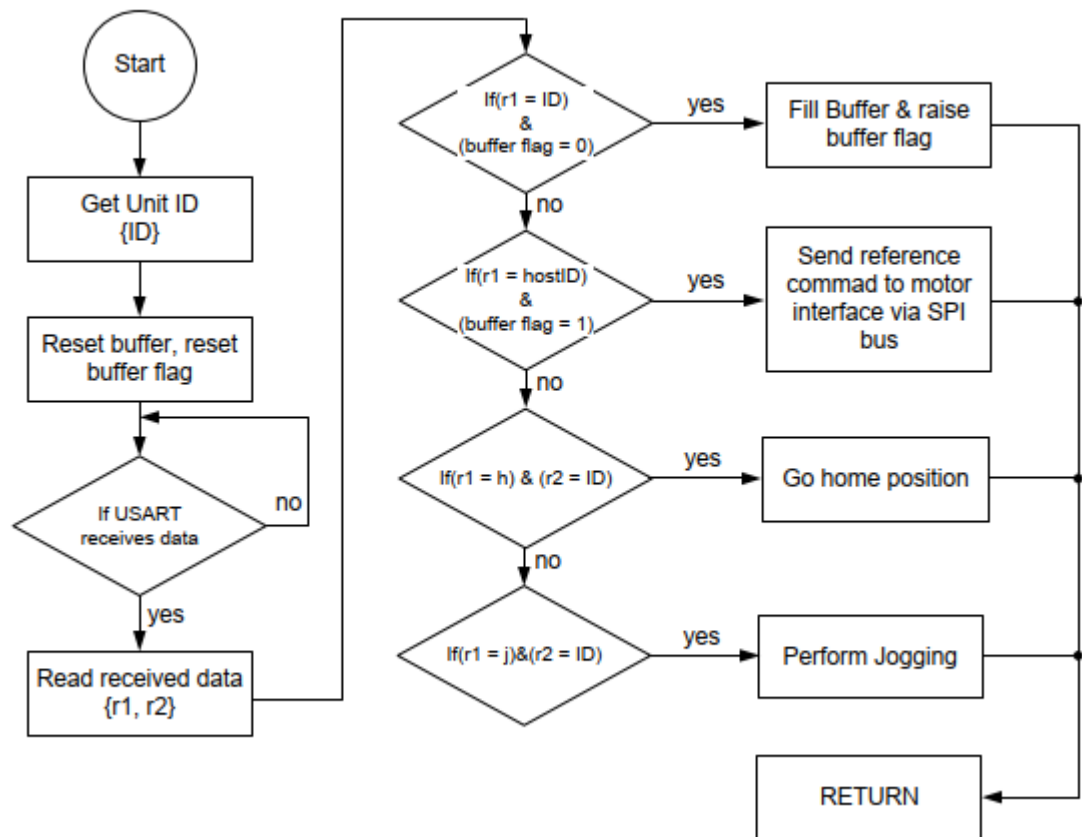


Рисунок 3.9 - Загальний опис прошивки

4 НАУКОВО-ДОСЛІДНА ЧАСТИНА

4.1 Дослідження розробленої системи.

Розподілена система керування, розроблена в рамках цієї роботи, реалізована на настільному фрезерному верстаті з ЧПК. Цей верстат є одним зі старих фрезерних верстатів DENFORD (модель: STARMILL-ATC). Як показано на рис. 4.1, STARMILL-ATC – це тривісний фрезерний верстат, що використовує систему автоматичної зміни інструменту. Оскільки деякі з його компонентів (включаючи плати керування та пристрій автоматичної зміни інструменту) вийшли з ладу, оригінальний електронний блок керування цього верстата довелося повністю замінити.

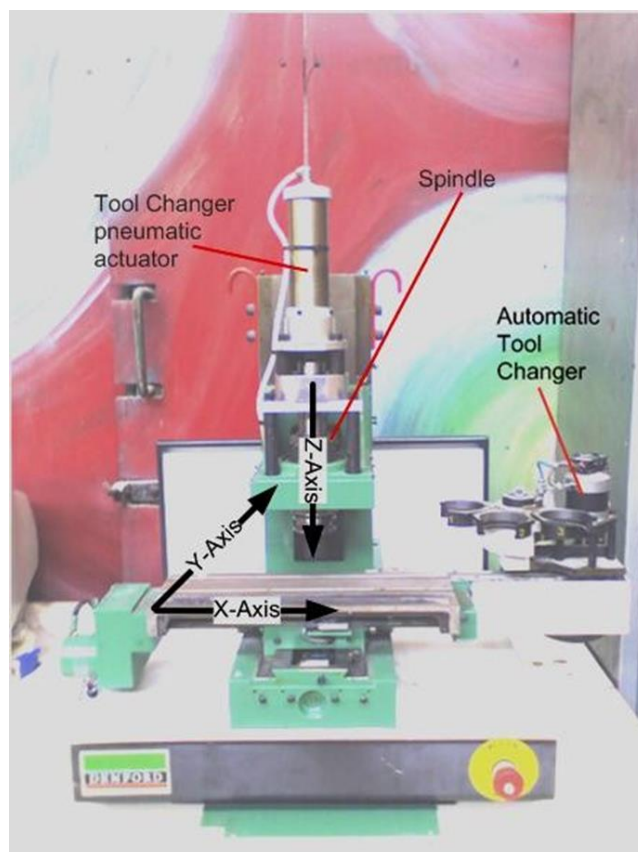


Рисунок 4.1 - Центр управління повітряним рухом STARMILL компанії Denford

Система керування STARMILL-ATC

Як згадувалося раніше, STARMILL-ATC використовує крокові двигуни для керування основними осями. Принципи роботи крокових двигунів суттєво відрізняються від двигунів постійного струму, які можуть обертатися безперервно, поки до їх клем прикладена постійна різниця потенціалів. З іншого боку, крокові двигуни ділять повний оберт на кілька невеликих кроків. Як і у випадку з реактивними двигунами з перемиканням, коли схема керування живленням (тобто драйвер крокового двигуна) живить обмотки двигуна у правильному порядку, ротор обертається на задану величину. STARMILL використовує драйвери крокових двигунів Parker-Hannifin (модель: SD2). Цей драйвер двигуна, який ефективно використовує мікросхеми драйверів крокових двигунів L298/L297, може видавати безперервний (сумарний) струм 2 А.

Розроблений розподілений контролер застосовано до верстата. Графічний інтерфейс користувача, який визначено в наступному розділі, тісно пов'язаний з цією системою керування для безперебійного керування верстатом. На відміну від оригінальної системи керування, головний ПК та контролери, розроблені (рис. 4.2) у цій роботі, з'єднані один з одним через польову шину EIA-485, як показано на рис. 4.3.

Важливо зазначити, що стандартний ПК не має жодного комунікаційного інтерфейсу EIA-485. Тому на головному ПК встановлюється комунікаційна карта на базі PCI, яка підтримує два пристрої EIA-485 (дво- або чотирипровідні).

Дослідження ефективності

Потребує вивчення продуктивності розробленого контролера. Оскільки STARMILL-ATC не оснащений жодними датчиками положення, можна оцінити лише продуктивність машини в розімкнутому циклі.



Рисунок 4.2 - Розроблена система управління

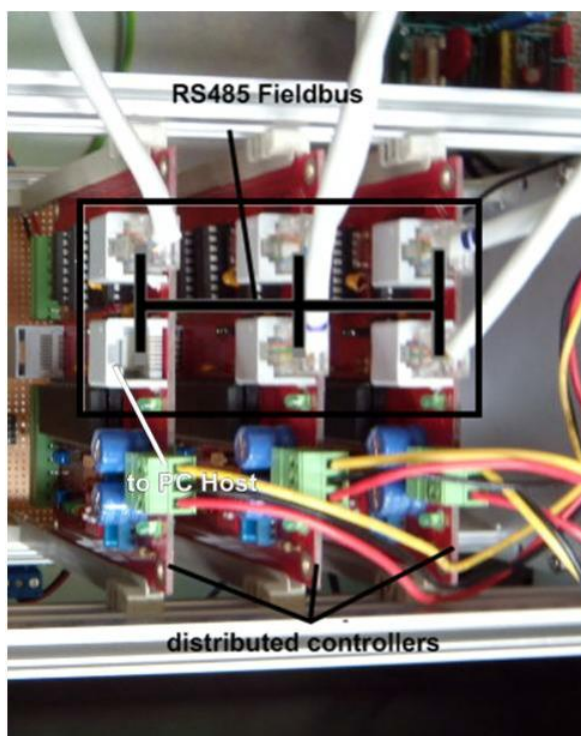


Рисунок 4.3 - Розвинена мережа управління

Зверніть увагу, що верстат з ЧПК з BLU 0,01 [мм] не відповідає вимогам промислової обробки.додатків. Однак, Впровадження має на меті висвітлити деякі ключові атрибути розробленої системи керування. Отже, можна опосередковано визначити масштабне застосування цієї нової системи.

Спочатку спостерігається здатність до швидкого позиціонування (Експеримент I). Зауважте, що в режимі швидкого позиціонування не очікується обробка поверхонь, але потрібне швидке та точне позиціонування ріжучого інструменту. Залежно від швидкостей прискорення та уповільнення осей, що рухаються, можуть виникати високі сили інерції. Як наслідок, крокові двигуни можуть пропускати кілька кроків. Тому належне прискорення та уповільнення осьового руху є життєво важливим для підтримки належної роботи верстата в цьому режимі.

У цьому експерименті для перевірки похибки швидкого позиціонування столу використовується компаратор (індикатор годинникового типу). Довільна траєкторія також була визначена. Після виконання програми очікується, що стіл повернеться у початкове положення. У всіх проведених експериментах не спостерігалось суттєвого відхилення на індикаторі годинникового типу, коли стіл повертався у початкове положення.

По-друге, досліджується продуктивність скульптурного різання заготовки з алюмінію (6061-T6). Існує кілька параметрів, які впливають на сили, що діють як на різець, так і на заготовку. Відомо, що сили різання (або збурення) негативно впливають на продуктивність системи керування. Під час цих експериментів (Експеримент II) підтримується постійна глибина різання (0,5 мм) вздовж осі X для заданої швидкості подачі. Потім реєструються відхилення калібрувального лічильника для різних діапазонів переміщення. Результати представлені на рис. 4.4.

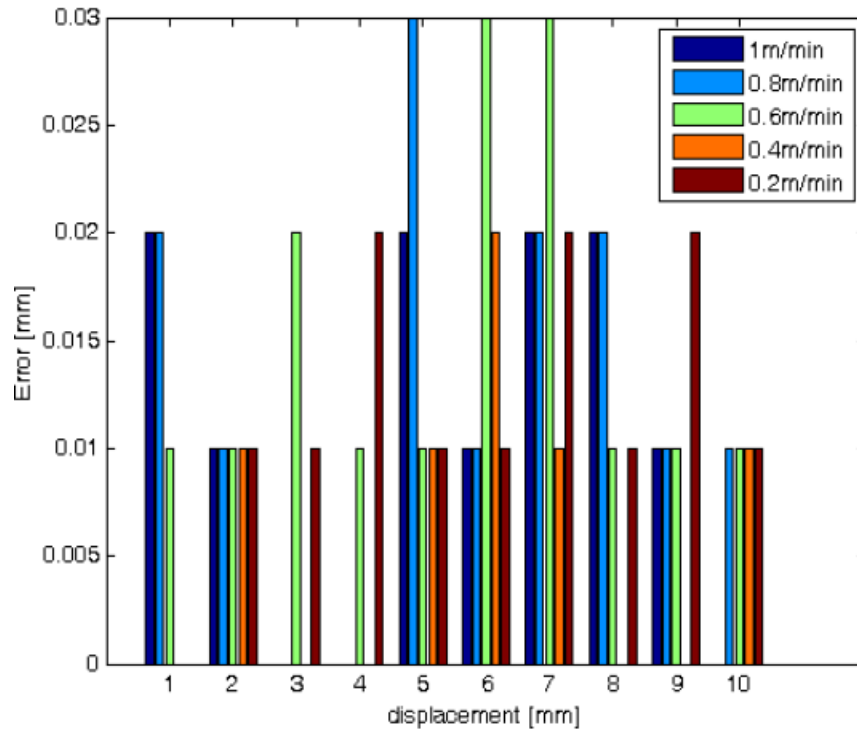


Рисунок 4.4 - Помилки позиціонування, що спостерігалися в експерименті II.

Зверніть увагу, що сили різання, що розвиваються на верстаті з ЧПК, викликають відхилення від бажаної траєкторії. Для компенсації результуючої похибки позиціонування в технології ЧПК використовуються виключно вимірювальні пристрої для подачі інформації про положення до (замкнутої) системи керування. Однак для системи керування з розімкнутим циклом, що використовується в цьому дослідженні, будь-яке відхилення від бажаної траєкторії накопичується з часом і залишається некомпенсованим. Як наслідок, спостерігаються похибки, показані на рис. 4.4 (1...3 BLU). Важливо зазначити, що вібрація компараторного стенда разом з похибками затискання (заготовки) також сприяють цим похибкам.

В якості третього експерименту досліджується продуктивність обробки системи (у площині XY). Алюмінієвий блок (6061-T6) має бути виіплений. Зауважте, що цей пластичний матеріал має низьку міцність і може бути легко

оброблений на малопотужному верстаті. На рис. 4.5 показано отриману заготовку.

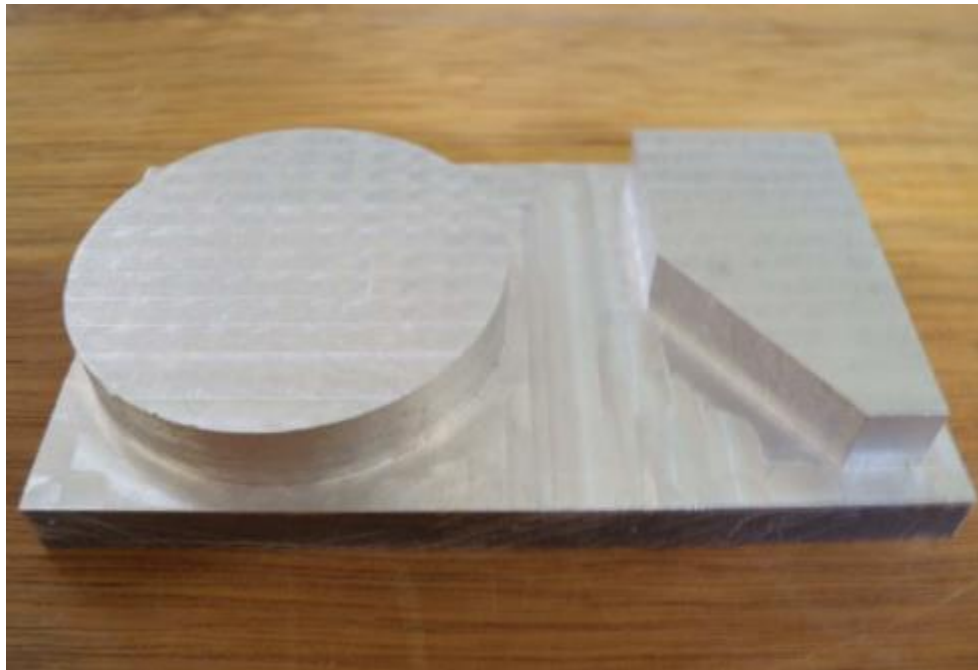


Рисунок 4.5 - Скульптурна заготовка, експеримент III

У цьому експерименті використовується твердосплавний інструмент. Використовуючи доступну потужність шпинделя (160 Вт), можна розрахувати максимальну швидкість подачі при повному (або частковому) занурюванні для вибраної глибини різання (a_e) та ширини різання (a_p).

Використовуючи компаратор з роздільною здатністю 1 мікрон, можна дослідити відхилення від бажаної геометрії.

Відхилення від опорної точки представлено на рис. 4.6. Найбільша відносна похибка становить 0,126 мм, тоді як стандартне відхилення вимірювань розраховане як 0,0520 [мм].

Абсолютні вимірювання на поверхні можна отримати, скинувши компаратор за допомогою концевих калібрів Джонсона. Після встановлення опорної висоти вимірювання можна виконати на сітці. Використовуючи зібрані дані, отримують графіки поверхні для круглих та прямокутних перерізів заготовки, як показано на рис. 4.7 та 4.8.

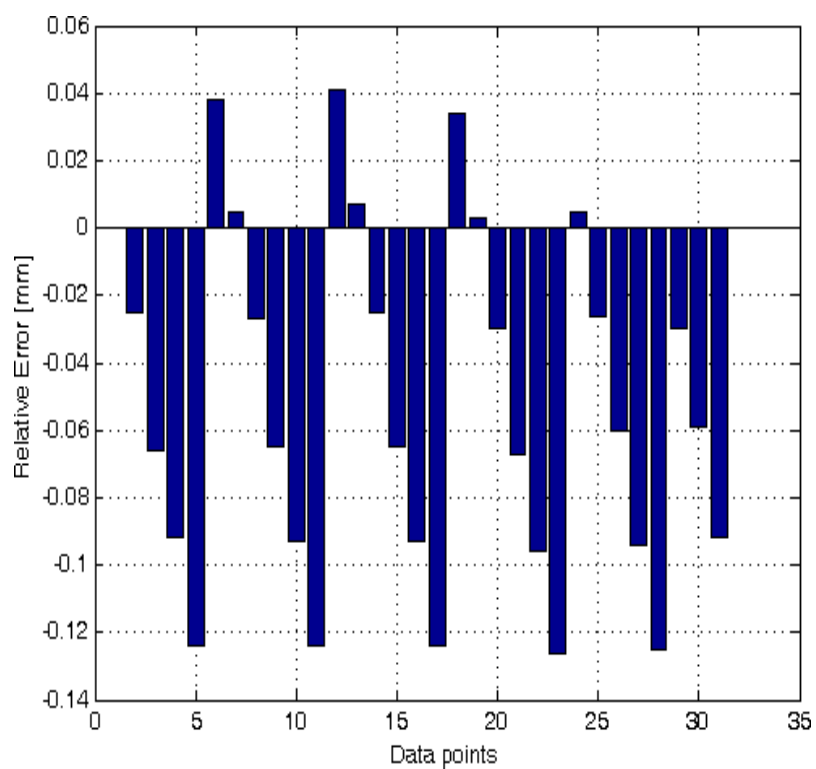


Рисунок 4.6 - Відносна похибка від опорної точки

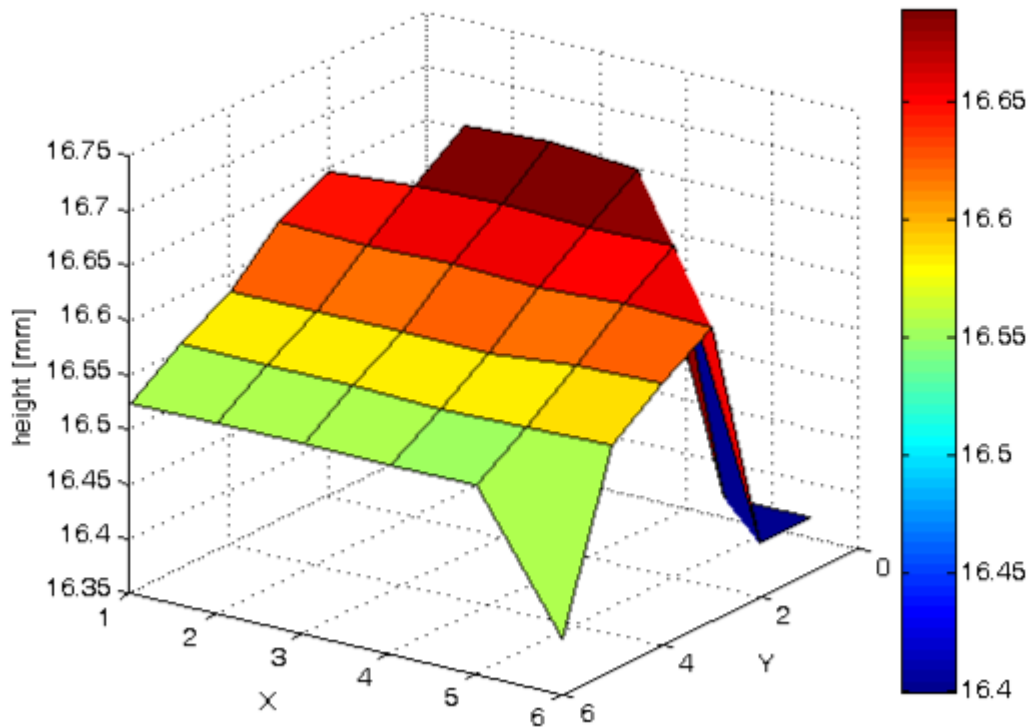


Рисунок 4.7 - Діаграма поверхні, круглий переріз

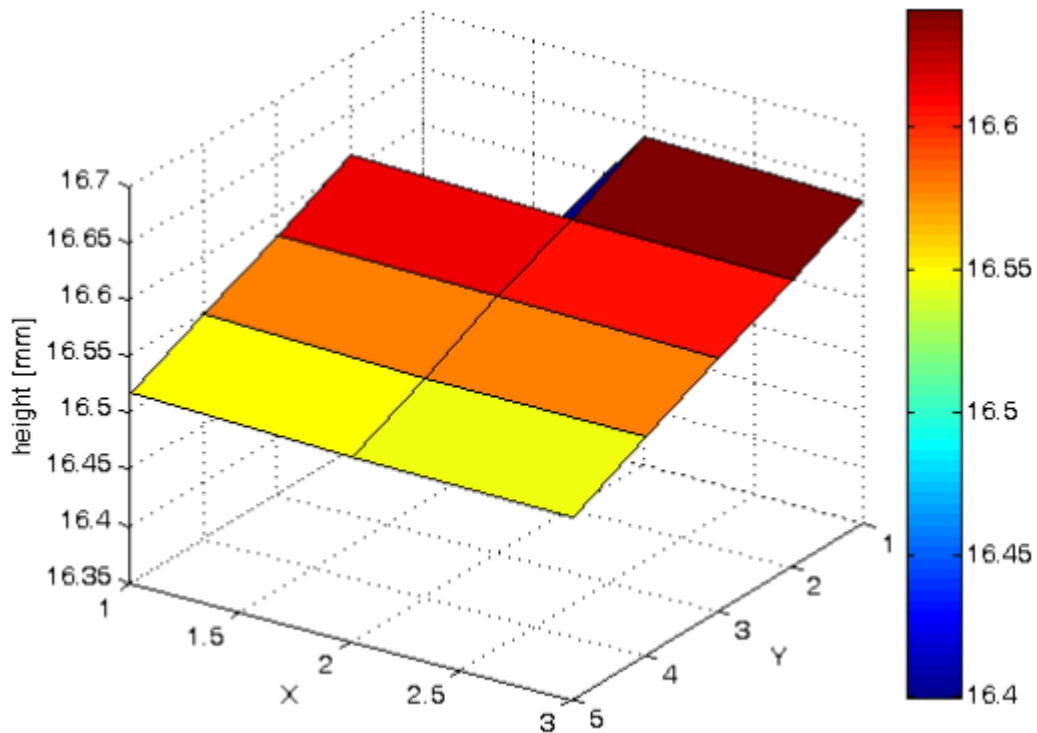


Рисунок 4.9 - Поверхнева ділянка, прямокутний переріз

Згідно з наведеними графіками, середня висота кругової траєкторії становить 16,5965 [мм], тоді як середня висота прямокутної ділянки становить 15,5719 [мм]. Як інтерпретація, спостерігається деяка хвилястість вздовж напрямку X через те, що під час обробки двигун осі Z знаходиться у стані спокою. Тому різниця у висоті для кожного проходу може бути не пов'язана з двигуном осі Z. Це, ймовірно, пов'язано з геометричними похибками, пов'язаними із затискним пристроєм (лещатами).

За допомогою цифрового компаса (з роздільною здатністю 10 мікрон) вимірюються певні розміри деталі. Спостерігалися деякі відхилення від номінальних розмірів. Максимальна похибка виникла в 3-й точці даних, яка становить 0,12 [мм], середнє значення становить 0,035 [мм]. Зверніть увагу, що на початку команди позиціонування для цього кругового шляху можуть містити похибки через перетворення чисел з плаваючою комою в цілі.

5 СПЕЦІАЛЬНА ЧАСТИНА

5.1 Графічний інтерфейс користувача та його особливості

Графічний інтерфейс користувача (GUI) дозволяє користувачам повноцінно керувати та працювати з верстатом з ЧПК. На відміну від текстових інтерфейсів, GUI пропонують графічні значки та візуальні індикатори для повного представлення інформації та дій, доступних користувачеві. Усі елементи керування програмою здійснюються шляхом безпосереднього маніпулювання графічними елементами.

Проектування візуальних елементів, їх поєднання та представлення є найважливішими моментами дизайну графічного інтерфейсу користувача (GUI). Популярні контролери ЧПК мають свої власні особливі графічні інтерфейси користувача (GUI). Один з популярних виробників контролерів верстатів з ЧПК, Fanuc, використовує різноманітний графічний інтерфейс користувача для своїх продуктів, який представлений на рис. 5.1.

Коротко кажучи, графічний інтерфейс користувача (GUI) інкапсулює візуальні елементи та об'єкти контролера. Гарний дизайн інтерфейсу користувача безпосередньо залежить від користувачів, а не від компонентів системи. Більшість контролерів верстатів з ЧПК використовують візуалізацію траєкторії інструменту під час роботи, текстовий редактор, числове положення ріжучого інструменту та моделювання програми обробки деталей з ЧПУ.

За графічним інтерфейсом користувача (GUI) існує багато класів керування, призначених для керування та експлуатації машини, зв'язку, інтерпретації програм обробки деталей, взаємодії з САМ-модулями програм САПР тощо.

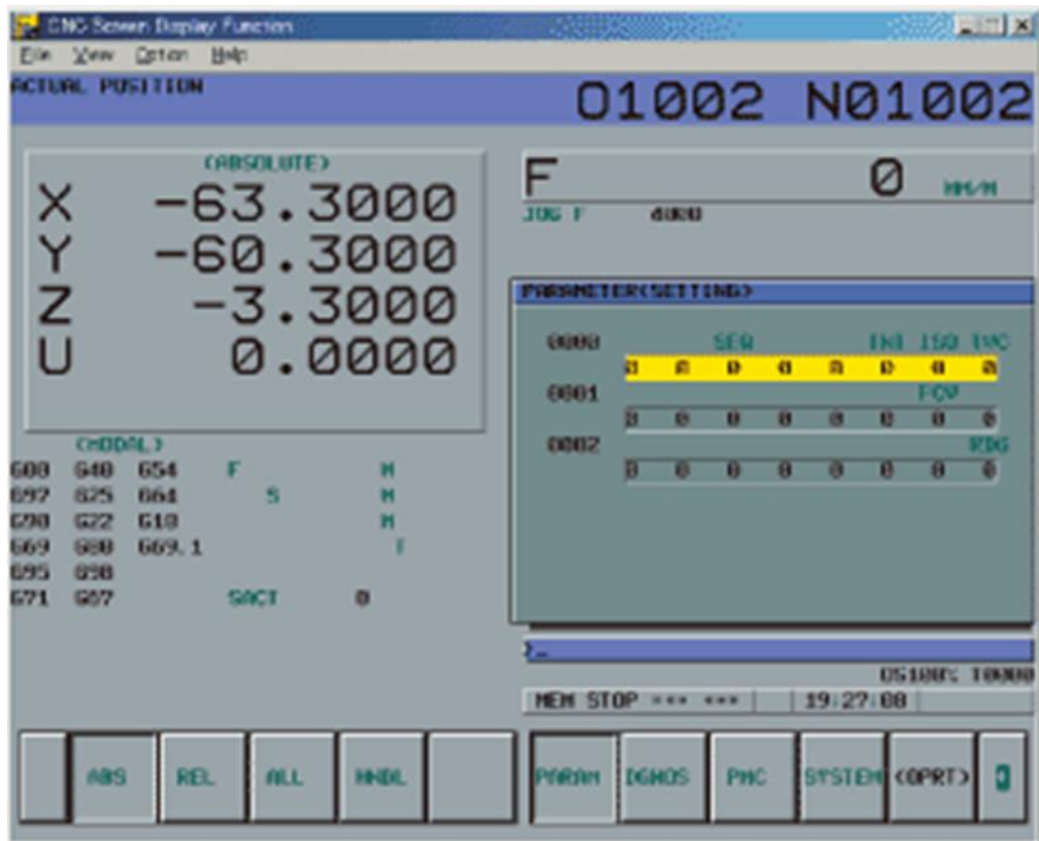


Рисунок 5.1 - Графічний інтерфейс користувача блоку керування ЧПК

По суті, розробка просунутого графічного інтерфейсу користувача виходить за рамки цієї дисертації. Розроблений простий графічний інтерфейс користувача являє собою розробку простого процесу графічного інтерфейсу користувача як програмного забезпечення для керування машиною та включає використання машинної dll.

Графічний інтерфейс користувача можна ідентифікувати як:

- ГРАФІЧНЕ ВІКНО для 2D-зображення інструменту на площинах XY, YZ та ZX
- ВІКНО ДІЙСНОГО ПОЛОЖЕННЯ для числового відображення швидкості подачі, швидкості шпинделя та фактичного положення різального інструменту
- ВІКНО СТАНУ для представлення операндів програми обробки деталей з ЧПУ

- ВІКНО РЕДАКТОРА для завантаження, збереження та редагування програм обробки деталей ЧПУ
- РЕЖИМ ПОПЕРЕДНЬОГО ПЕРЕГЛЯДУ для моделювання програми обробки деталей з ЧПУ, щоб запобігти неправильному виконанню програми обробки деталей з ЧПУ.

Програма FMILL розроблена для належної роботи розроблених розподілених контролерів. На жаль, її не можна адаптувати до жодної САМ-програми, але за допомогою редактора програм обробки деталей з ЧПУ можна скласти та виконати програму обробки деталей з ЧПУ. Графічний інтерфейс представлено на рис. 5.2.

Це простий інтерфейс верстата з ЧПК, який може задовольнити основні вимоги оператора. До програми також додано додаткові індикатори. Вікно фактичного положення показує відповідне абсолютне положення відповідно до попередньо встановленого початкового положення. Сувої, розташовані у вікні фактичного положення, показують положення ріжучого інструменту в системах координат верстата, і користувач може інтерпретувати неминучість спрацьовування кінцевих вимикачів. Вікна індикаторів швидкості шпинделя та швидкості подачі розташовані для відображення поточної швидкості шпинделя та швидкості подачі.

У вікні стану відображаються поточна робоча площа, режим компенсації інструменту та режим руху. Коли потрібне моделювання траєкторії інструменту, натискання кнопки попереднього перегляду відобразить траєкторію траєкторії інструменту у графічному вікні. Виконання можна запустити натисканням кнопки «Виконати». Блок-схема графічного інтерфейсу показана на рис. 5.3.

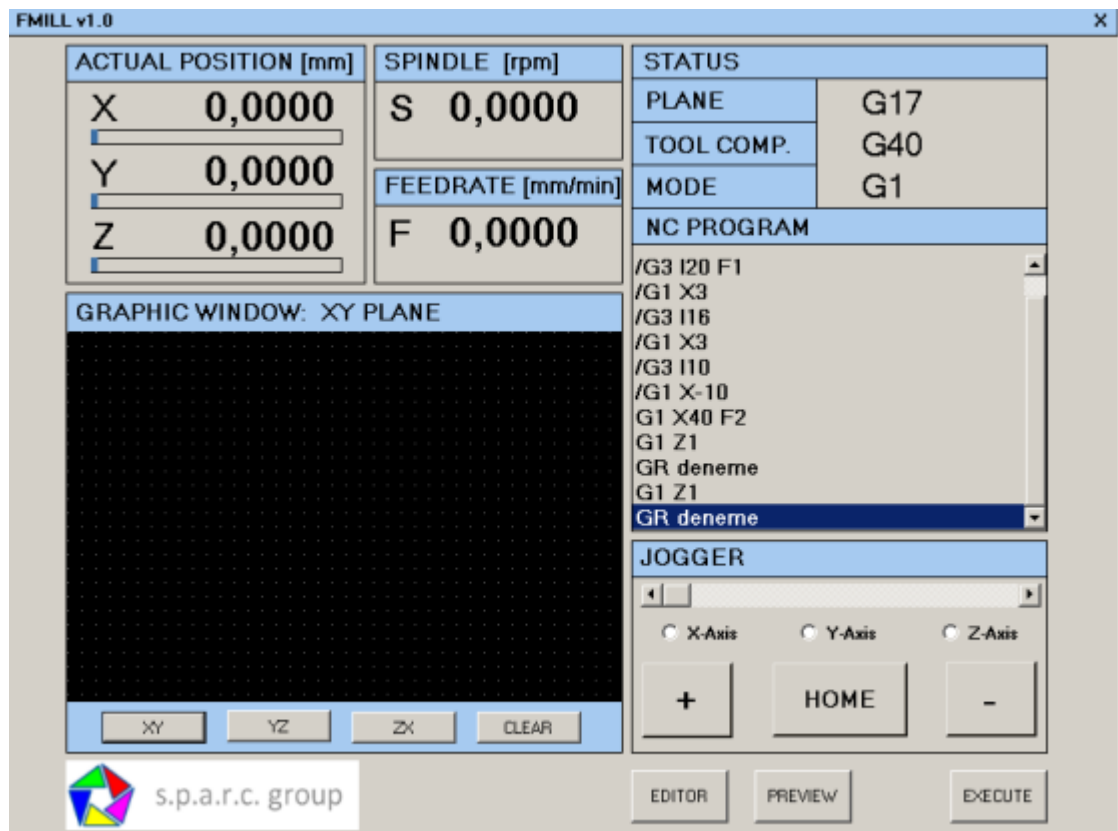


Рисунок 5.2 - Графічний інтерфейс користувача FMILL

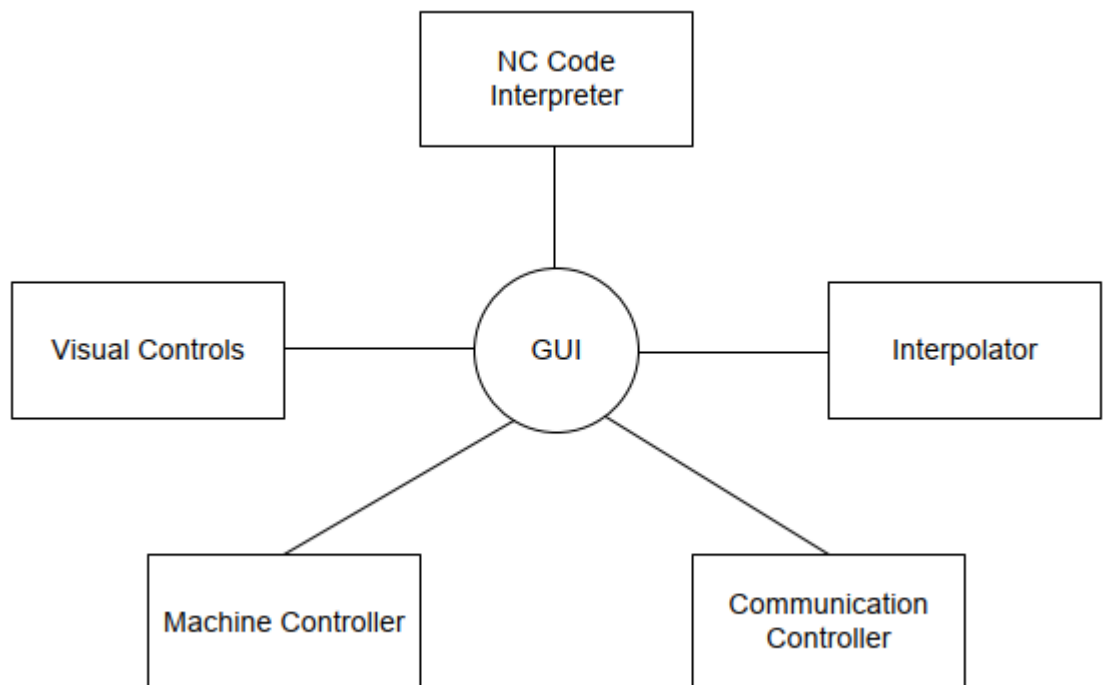


Рисунок 5.3 - Блок-схема графічного інтерфейсу користувача

Машинний .dll

DLL розшифровується як динамічно-компонована бібліотека. Основна ідея полягає в завантаженні підпрограм у прикладну програму під час її виконання, а не в компонуванні під час компіляції з розміщенням їх в окремих місцях на жорсткому диску. Таким чином, компілятор виконує лише мінімальний обсяг роботи.

Для програмного використання розподілених контролерів розроблено машинну DLL-бібліотеку. Вона містить деякі директиви та обробляє протокол зв'язку.

Конструктор:

SPARCMILL_DCS можна включити будь-який файл з назвою послідовного порту та ініціалізованою швидкістю зв'язку.

Підготовка даних:

Підготовка даних повертає масив байтів, що містить закодовані команди опорного положення. Вхідними аргументами є дельта-зміщення у міліметрах з типом даних double та швидкість подачі, бажана швидкість заданого зміщення у попередньому аргументі у метрах за хвилину з типом даних double.

Додати до черги:

додати в чергу використовується для побудови масиву байтів для кожної осі відповідно до заданих приростних переміщень. Вхідними аргументами є приростні переміщення в міліметрах для кожної осі, dx, dy та dz. Також аргументом є тангенціальна бажана швидкість у метрах за хвилину. Усі аргументи для цих функцій мають подвійний розмір.

Надіслати дані:

Надіслати дані підлягає надсиланню 1500 байтів побудованого масиву даних. Його аргументами є ідентифікатор розподіленого блоку керування призначенням за допомогою байтів, ідентифікатор розподіленого блоку

керування головним блоком за допомогою байтів та масив байтів, який містить закодовані дані команд опорного положення.

Виконати:

Ця функція використовує глобальні змінні, створені функцією `addtoqueue`. Також вона керує протоколом зв'язку, а також синхронізацією програмного забезпечення.

Біг:

Біг підтюпцем Функція використовує ідентифікатор одиниці та швидкість подачі за допомогою байтів як аргументи. По суті, ця функція викликає операцію поштовхового переміщення в прошивці контролера, яка описана в попередніх розділах.

Стоп:

Ця функція зупиняє поштовх. Вона викликає переривання зупинки поштовху в прошивці контролера. Її слід використовувати після виконання функції поштовху. Вона нічого не повертає.

Цей клас керування не включає додаткові класи керування машинами. Отже, той, хто намагається розробити програмне забезпечення для розподілених контролерів, повинен розробити клас інтерполяції.

6 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

6.1. Безпека електрообладнання у виробничих процесах

Забезпечення безпеки електрообладнання у виробничих процесах є одним із ключових аспектів охорони праці. В умовах сучасного виробництва, де активно використовуються автоматизовані системи, включаючи інтелектуальні модулі контролю якості, забезпечення електробезпеки стає критично важливим. Ефективність електробезпеки залежить від правильного впровадження технічних рішень, дотримання стандартів та регулярного навчання персоналу.

Безпека електрообладнання базується на використанні таких технічних рішень, як надійна ізоляція проводів, заземлення корпусів обладнання та автоматичні вимикачі. Ізоляція запобігає випадковому контакту з провідниками струму, а заземлення забезпечує захист від струмових витоків. Автоматичні вимикачі дозволяють швидко реагувати на короткі замикання або перевантаження, мінімізуючи ризик виникнення аварій. Значна увага приділяється моніторингу стану електрообладнання. Регулярна перевірка дозволяє вчасно виявляти потенційні проблеми, такі як пошкодження ізоляції чи знос елементів системи. Застосування автоматизованих засобів моніторингу, наприклад, сенсорів стану електропроводки, допомагає підвищити рівень безпеки та зменшити витрати на аварійний ремонт.

Виробничі процеси створюють додаткові ризики для електрообладнання, серед яких основними є:

- Ураження електричним струмом, що виникає через несправність ізоляції чи необережний контакт з оголеними провідниками.
- Підвищений рівень вологості або забруднення, які можуть спричинити коротке замикання.
- Перегрів обладнання, що може викликати займання.

Для попередження цих ризиків необхідно:

- Використовувати електрообладнання з класом захисту IP67 або вище, яке є стійким до вологи та пилу.
- Забезпечити встановлення захисних кожухів навколо рухомих і струмопровідних частин.
- Впроваджувати температурний контроль обладнання з автоматичним відключенням при перегріві.

Система електробезпеки регулюється рядом міжнародних і національних стандартів. До них відносяться:

- **IEC 60364** – стандарт міжнародної електротехнічної комісії для проектування електроустановок.
- **ISO 45001** – стандарт системи управління охороною праці, який охоплює електробезпеку як її частину.

Ці нормативи передбачають використання сучасних технологій захисту та надання пріоритету автоматизованим системам моніторингу та аварійного реагування.

На сучасних підприємствах впровадження заходів електробезпеки базується на інтеграції технічних, організаційних та навчальних ініціатив. Наприклад, працівники проходять регулярне навчання з безпечного використання електрообладнання. Використовуються діелектричні рукавички, підкладки та килимки для роботи в небезпечних зонах. Планово проводяться перевірки стану ізоляції та працездатності автоматичних вимикачів.

Таблиця 6.1.1 – Вплив впровадження стандартів електробезпеки на показники виробництва

Показник	До впровадження стандартів	Після впровадження стандартів
Кількість аварій (за рік)	15	3
Час простою обладнання (годин)	120	30
Кількість травмувань	7	1

Проведені заходи демонструють ефективність інтеграції стандартів та сучасних технологій моніторингу. У результаті значно знизились ризики аварійних ситуацій, зменшилися витрати на ремонт і час простою обладнання.

Забезпечення безпеки електрообладнання є фундаментальною умовою для стабільної роботи виробничих процесів. Інтеграція таких підходів дозволяє знизити аварійність, підвищити продуктивність та забезпечити відповідність сучасним стандартам охорони праці.

6.2. Аналіз ризиків на робочому місці оператора

Робоче місце оператора відіграє ключову роль у забезпеченні безперебійного функціонування виробничих процесів. Проте умови праці на такому місці можуть бути пов'язані з ризиками, що впливають як на ефективність праці, так і на здоров'я працівників. Для зниження цих ризиків необхідно проводити комплексний аналіз і впроваджувати відповідні заходи захисту.

Одним із ключових ризиків є ергономічні проблеми. Робоче місце оператора повинно забезпечувати комфорт під час тривалого виконання

завдань. Неправильне розташування обладнання або недостатня підтримка спини можуть призвести до фізичного дискомфорту, який знижує концентрацію і продуктивність. Для усунення цих ризиків необхідно забезпечити оптимальну висоту робочих столів, розміщення моніторів на рівні очей та використання стільців із регульованими спинками й підлокітниками. Якість освітлення також відіграє важливу роль, адже недостатнє освітлення може спричинити перевтому очей і головний біль.

Електробезпека є другим важливим аспектом, який потребує особливої уваги. Використання електрообладнання вимагає ретельного контролю його стану та регулярного обслуговування. Наприклад, всі струмопровідні частини повинні бути надійно ізольовані, а обладнання — підключене до захисного заземлення. Застосування автоматичних вимикачів дозволяє швидко реагувати на перевантаження чи коротке замикання, запобігаючи виникненню аварійних ситуацій. Для мінімізації ризиків від електрообладнання необхідно забезпечити наявність резервних джерел живлення, які підтримуватимуть роботу ключових систем навіть у разі збоїв у мережі.

Психологічні ризики також впливають на продуктивність оператора. Висока відповідальність, монотонність роботи та необхідність постійної уваги можуть призводити до стресу та емоційного виснаження. Для зниження цього виду ризиків слід впроваджувати регулярні перерви, створювати комфортні умови для роботи та відпочинку, а також використовувати автоматизовані системи, які зменшують кількість рутинних завдань, виконуваних оператором.

Інформаційні ризики пов'язані з перевантаженням оператора великою кількістю даних, які надходять від систем моніторингу. Якщо інтерфейс програмного забезпечення недостатньо зрозумілий, це може ускладнити прийняття рішень та збільшити ймовірність помилок. Для вирішення цієї

проблеми слід забезпечити чітку візуалізацію даних, використовуючи кольорові індикатори для відображення критичних станів обладнання.

Таблиця 6.2 – Вплив заходів з покращення робочого місця оператора

Показник	До впровадження заходів	Після впровадження заходів
Кількість скарг на фізичний дискомфорт	Висока	Низька
Час реакції оператора на аварії	15 секунд	10 секунд
Рівень стресу операторів	Середній	Низький
Кількість помилок через перевантаження інформацією	Висока	Низька

Після впровадження системних заходів для мінімізації ризиків спостерігається значне покращення умов праці операторів. Зниження кількості скарг на фізичний дискомфорт свідчить про покращення ергономіки робочих місць. Зменшення часу реакції на аварії є результатом впровадження зрозумілого інтерфейсу програмного забезпечення та використання автоматизованих систем. Зниження рівня стресу операторів свідчить про ефективність створення зон відпочинку та зменшення навантаження.

Аналіз ризиків на робочому місці оператора демонструє, що більшість із них можна успішно зменшити за допомогою комплексного підходу. Застосування сучасних ергономічних рішень, забезпечення електробезпеки, впровадження автоматизації та організація комфортних умов праці суттєво знижують ризики для операторів. Результати свідчать про те, що інвестиції в

покращення умов праці не лише забезпечують здоров'я та безпеку працівників, але й підвищують загальну ефективність виробничого процесу.

6.3. Охорона праці при експлуатації системи

Охорона праці при експлуатації інтелектуальної системи контролю якості є критично важливим аспектом для забезпечення безпеки працівників, зниження ризиків травматизму та підтримання стабільної роботи обладнання. Інтеграція таких систем у виробничий процес вимагає врахування низки факторів, що впливають на взаємодію персоналу з апаратними та програмними компонентами, а також з іншими елементами виробничої лінії.

Одним із основних завдань є аналіз потенційних ризиків, які можуть виникнути під час роботи системи. Це включає ідентифікацію зон підвищеної небезпеки, наприклад, ділянки встановлення камер та датчиків, місця взаємодії оператора із системою або зони, де вилучається бракована продукція. Під час проектування системи важливо забезпечити мінімальну необхідність фізичного контакту оператора з рухомими частинами обладнання.

Важливим елементом безпеки є фізичний захист обладнання та робочої зони. Камери, сенсори, механічні маніпулятори та інші апаратні компоненти системи повинні бути встановлені у захищених корпусах або мати додаткові бар'єри, які запобігають випадковому контакту працівників із небезпечними частинами. Крім того, важливо передбачити наявність знаків і маркувань, що вказують на небезпечні зони та способи їх уникнення.

Значна увага повинна бути приділена електробезпеці, адже система функціонує на основі високопродуктивних обчислювальних пристроїв і сенсорів, підключених до електромережі. Усі електричні з'єднання повинні відповідати стандартам безпеки, мати належну ізоляцію та бути захищеними

від вологи, пилу чи можливих механічних пошкоджень. Необхідно передбачити використання автоматичних вимикачів для запобігання короткому замиканню чи перевантаженню мережі.

Особливу роль відіграє ергономіка робочих місць операторів, які взаємодіють із системою. Монітори, на яких відображаються результати роботи системи, повинні бути встановлені на оптимальній висоті та відстані, щоб знизити навантаження на зір і шийний відділ хребта. Робочі місця слід обладнати ергономічними кріслами та столами, а також забезпечити належне освітлення. Такі заходи сприяють зниженню втоми персоналу та підвищенню їхньої продуктивності.

Навчання персоналу є одним із найважливіших аспектів охорони праці. Працівники повинні бути ознайомлені з принципами роботи системи, правилами безпечної експлуатації обладнання та алгоритмами дій у разі виникнення аварійних ситуацій. Особливо важливо навчити операторів правильно реагувати на повідомлення системи, наприклад, на сигнали про виявлення критичних дефектів чи несправностей у роботі обладнання.

Для забезпечення безпеки необхідно впровадити системи аварійного відключення та сигналізації. Усі компоненти системи мають бути обладнані кнопками аварійної зупинки, що дозволяють миттєво вимкнути обладнання у разі виникнення небезпечної ситуації. Світлові та звукові індикатори повинні попереджати персонал про виявлені дефекти, технічні несправності чи перевантаження системи.

Важливим компонентом є систематичний моніторинг стану системи та її регулярне технічне обслуговування. Використання програмного забезпечення для моніторингу дозволяє автоматично фіксувати показники роботи обладнання, відстежувати потенційні несправності та своєчасно планувати ремонт чи заміну компонентів. Це знижує ризики аварійних зупинок та несподіваних поломок.

Інтеграція цих заходів у повсякденну експлуатацію системи сприяє створенню безпечного виробничого середовища, мінімізує ризики для персоналу та забезпечує стабільну й ефективну роботу інтелектуальної системи. Таким чином, охорона праці при експлуатації інтелектуальної системи є багатогранним процесом, який охоплює як технічні, так і організаційні заходи, спрямовані на безпеку працівників та підтримку високих стандартів виробничих процесів.

ВИСНОВКИ

Метою роботи була розробка та впровадження розподіленої системи керування для верстатів з ЧПК. Були реалізовані різні структури.

Головним вузлом цієї системи керування є ПК та відповідне програмне забезпечення керування. Система керування в реальному часі не може керуватися безпосередньо ПК. Переривання викликаються підлеглими вузлами. Необхідний запит або надання інформації зворотного зв'язку. Тому обсяг даних, які необхідно передати по комунікаційній мережі, збільшиться. Цю ситуацію було вирішено, застосовуючи швидші комунікаційні інтерфейси.

Розроблену розподілену систему керування, що включає апаратне забезпечення, прошивку та програмне забезпечення, було реалізовано на керованому (3-осьовому) обробному центрі з ЧПК з розімкнутим циклом. Було досліджено продуктивність обробки за різних умов. Загальна продуктивність системи керування знаходилась в оптимальних діапазонах.

Розроблений графічний інтерфейс користувача розширив межі розподілених контролерів. Спочатку було перевірено здатність до 2D-інтерполяції. Потім розподілені контролери були змушені продемонструвати свої навички 3D-інтерполяції.

ПЕРЕЛІК ПОСИЛАНЬ

1. Taira T., Yamasaki N. Functionally Distributed Control Architecture. Nippon Kikai Gakkai Robotikusu, Mekatoronikusu Koenkai Koen Ronbunshu. 2003.
2. Yook J. K., Tilbury D. M., Soparkary N. R. A Design Methodology for Distributed Control Systems to Optimize Performance in the Presence of Time Delays. IEEE, American Control Conference. 2001.
3. Luntz J., Messner W. A Distributed Control System for Flexible Materials Handling. IEEE Control System Magazine. 1997. Vol. 17, no. 1.
4. Caspi P., Mazuet C., Paligot N. P. About the Design of Distributed Control Systems: The Quasi-Synchronous Approach. Lecture Notes in Computer Science. Springer Berlin/Heidelberg, 2001.
5. Robert W., Fletcher M., Norrie D. H. An Agent-Based Approach to Reconfiguration of Real-Time Distributed Control Systems. IEEE Transactions on Robotics and Automation. 2002. Vol. 10, no. 1.
6. Chan H., Ozguner U. Closed-loop Control of Systems over a Communication Network with Queues. Proceeding of the American Control Conference. USA, 1994. Vol. 2. P. 811–815.
7. Koninckx R., Brussel H. Closed-loop, Fieldbus-based Clock Synchronisation for Decentralised Control Systems. Reconfigurable Manufacturing Systems and Transformable Factories. Springer Berlin/Heidelberg, 2006. P. 213–235.
8. Ferrarini L., Veber C., Schwab C., Tangermann M., Prayati A. Control Functions Development for Distributed Automation Systems Using Torero Approach. IFAC. 2005.
9. Peterson L., Austin D., Chiristensen H. DCA: A Distributed Control Architecture for Robotics. IEEE/RSJ. 2001. Vol. 11, no. 2.

10. Angelov C., Sierszecki K. Component-Based Design of Software for Distributed Embedded Systems. International Conference on Software and Systems Engineering ICSSEA. 2003. Vol. 3, no. 1.
11. Cauffriez L., Ciccotelli J., Conrard B., Bayart M. Design of Intelligent Distributed Control Systems: A Dependability Point of View. Reliability Engineering and System Safety. 2004. P. 19–32.
12. Aarsten A., Brugali D., Menga G. Designing Concurrent and Distributed Control System. Communications of the ACM. 1996.
13. Williams T. The Development of Reliability in Industrial Control Systems. IEEE MICRO. 1984. No. 0272-1732/84/1200.
14. Yasuda G. Distributed Autonomous Control of Modular Robot Systems Using Parallel Programming. Journal of Materials Processing Technology. 2003. P. 357–364.
15. D'Andrea R., Dullerud G. E. Distributed Control Design for Spatially Interconnected Systems. IEEE Transactions on Automatic Control. 2003. Vol. 48, no. 9.
16. Nilsson J. Real-Time Control Systems with Delays : PhD. Thesis. Lund Institute of Technology, Department of Automatic Control, 1998.
17. Wyeth G., Kennedy J., Lillywhite J. Distributed Digital Control of a Robot Arm. Proceedings of the Australian Conference on Robotics and Automation: ACRA. 2000. P. 217–222.
18. Zongying S., Wenli X., Xu W., Peigang J. Distributed Hierarchical Control System of Humanoid Robot THBIP-1. Proceedings of the 4th World Congress on Intelligent Control and Automation. 2002. Vol. 2. P. 1265–1269.
19. Schafer T., Chevalier M. Distributed Motor Control Using the 80C196KB. AP-428 Application Note. Intel Corporation, 1993.
20. Brennan R. W., Norrie D. H. Metrics for Evaluating Distributed Manufacturing Control Systems. Computers in Industry. 2003. P. 225–235.

21. А.Г. Микитишин, М.М. Митник, П.Д. Стухляк, В.В. Пасічник Комп'ютерні мережі. Книга 1. [навчальний посібник] (Лист МОНУ №1/11-8052 від 28.05.12р.) - Львів, "Магнолія 2006", 2013. – 256 с.
22. А.Г. Микитишин, М.М. Митник, П.Д. Стухляк, В.В. Пасічник Комп'ютерні мережі. Книга 2. [навчальний посібник] (Лист МОНУ №1/11-11650 від 16.07.12р.) - Львів, "Магнолія 2006", 2014. – 312 с.
23. Микитишин А.Г., Митник, П.Д. Стухляк. Комплексна безпека інформаційних мережевих систем: навчальний посібник – Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2016. – 256 с.
24. Микитишин А.Г., Митник М.М., Стухляк П.Д. Телекомунікаційні системи та мережі : навчальний посібник для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2017 – 384 с.
25. Введення в комп'ютерну графіку та дизайн: Навчальний посібник для студентів спеціальності 174 "Автоматизація, комп'ютерно-інтегровані технології та робототехніка"/Укладачі: О.В. Тотосько, П.Д. Стухляк, А.Г. Микитишин, В.В. Левицький, Р.З. Золотий - Тернопіль: ФОП Паляниця В.А., 2023 - 304с. <http://elartu.tntu.edu.ua/handle/lib/41166>.
26. Пилипець М. І. Правила заповнення основних форм технологічних документів : навч.-метод. посіб. / Уклад. Пилипець М. І., Ткаченко І. Г., Левкович М. Г., Васильків В. В., Радик Д. Л. Тернопіль : ТДТУ, 2009. 108 с. <https://elartu.tntu.edu.ua/handle/lib/42995>.