

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

Факультет прикладних інформаційних технологій та електроінженерії

(повна назва факультету)

Кафедра автоматизації технологічних процесів і виробництв

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему:

РОЗРОБКА АВТОМАТИЗОВАНОЇ
СИСТЕМИ ОПТИМАЛЬНОГО УПРАВЛІННЯ
ПРОМИСЛОВИМИ МОБІЛЬНИМИ РОБОТАМИ

Виконав студент

VI курсу, групи КТМ-61

спеціальності

151 Автоматизація та комп'ютерно-

інтегровані технології

(шифр і назва спеціальності)

(підпис)

Білоус П.В.

(прізвище та ініціали)

Керівник

(підпис)

Медвідь В.Р.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Козбур І.Р.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Савків В.Б.

(прізвище та ініціали)

Рецензент

(підпис)

Тотосько О.В.

(прізвище та ініціали)

Тернопіль
2023

ЗМІСТ

ВСТУП	7
1 АНАЛІТИЧНА ЧАСТИНА	8
1.1 Виклик	8
1.2 Нова парадигма.....	12
1.3 Описання завдання	12
2 НАУКОВО-ДОСЛІДНИЦЬКА ЧАСТИНА	14
2.1 Автоматизовані виробничі системи та системи автономної автоматизації	14
2.2 Тенденції в автоматизації.....	15
2.3 Вступ та прогрес в сходинковій логіці.....	21
2.3.1 Логіка сходів	23
2.3.2 Спроби автоматичної генерації коду.....	28
2.4 Розподілені системи управління	30
2.4.1 Виробництво на основі агентів.....	31
2.4.2 Холонічне виробництво	32
2.4.3 Результати та висновки	33
2.5 Виробництво напівпровідників.....	35
2.6 Підсумки та висновки	35
3 ТЕХНОЛОГІЧНА ЧАСТИНА	37
3.1 Вступ.....	37
3.2 Жадібна автономна система автоматизації.....	37
3.3 Огляд симулятора мережевої автономної системи автоматизації	37
3.3.1 Моделювання систем транспортування матеріалів	38

3.3.2	Вирішення проблеми стохастичної природи реального виробництва	39
3.3.3	Кодування симуляції	40
3.4	Структури зберігання даних NAASS.....	40
3.5	Графічний інтерфейс користувача AASS	41
3.5.1	Конфігурація компонентів.....	42
3.5.2	Компонування компонентів.....	42
3.5.3	Підтвердження взаємодії та макет	42
3.5.4	Визначення часу проходження та пропускнуої здатності.....	43
3.5.5	Конфігурація симуляції.....	44
3.5.6	Виведення сирих даних.....	45
3.6	Компоненти NAASS.....	45
3.6.1	Вихідний вузол.....	45
3.6.2	Вузол стоку	46
3.6.3	Вузол розгалужувача доріжок перемикача	46
3.6.3.1	Функція роботи розгалужувача доріжок перемикача	47
3.6.3.2	Функція фінішної обробки розгалужувача доріжок.....	47
3.6.4	Вузол з'єднання стрілочних перемикачів	49
3.6.5	Вузол робочої станції	49
3.7	Результати та статистика AASS	50
3.8	Валідація та висновок NAASS	51
4	КОНСТРУКТОРСЬКА ЧАСТИНА	54
4.1	Кількісна та якісна оцінка	54
4.1.1	Пробна кількість	54

4.1.2	Кількість оброблених візків.....	55
4.1.3	Системні варіації та важливість варіативності.....	56
4.2	Тести макета та SOP.....	57
4.2.1	Тестові макети.....	58
4.2.2	Результати та обговорення	60
4.3	Масштабованість системи	63
4.3.1	Вступ	63
4.3.2	Тест єдиної системи.....	61
4.3.3	Результати єдиної системи.....	64
4.3.4	Випробування неоднорідних систем	66
4.3.5	Результати та обговорення неоднорідної системи	67
4.4	Надійність системи.....	68
4.4.1	Вступ	68
4.4.2	Результати та обговорення	69
4.5	Висновки	71
5	СПЕЦІАЛЬНА ЧАСТИНА.....	72
5.1	Внески	72
5.2	Локалізовані системи прийняття рішень	72
5.3	Симулятор мережевої автономної системи автоматизації	73
5.3.1	Існуючі компоненти.....	73
5.3.2	Додаткові компоненти.....	74
5.3.3	Подорож у двох напрямках.....	74
5.3.4	Реалізація обмеження черги	75
5.3.5	Мова реалізації.....	75

5.4	Продуктивність АСА, масштабування системи та надійність системи	75
5.5	Подальші дослідження	76
5.5.1	Використані алгоритми оптимізації	76
5.5.2	Використані показники	77
5.5.3	Виведення псевдокоду.....	77
5.5.4	Відношення масштабування.....	77
5.5.5	Подальша перевірка.....	78
5.6	Підсумок	78
6	ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВЧАЙНИХ СИТУАЦІЯХ ...	79
6.1	Заходи, які зменшують небезпеку виникнення вибухів та пожеж.....	79
6.2	Фізичні основи електробезпеки.....	81
6.3	Підвищення стійкості роботи об'єктів електроенергетики в воєнний час	82
	ВИСНОВКИ	86
	ПЕРЕЛІК ПОСИЛАНЬ.....	87
	ДОДАТКИ.....	93

ВСТУП

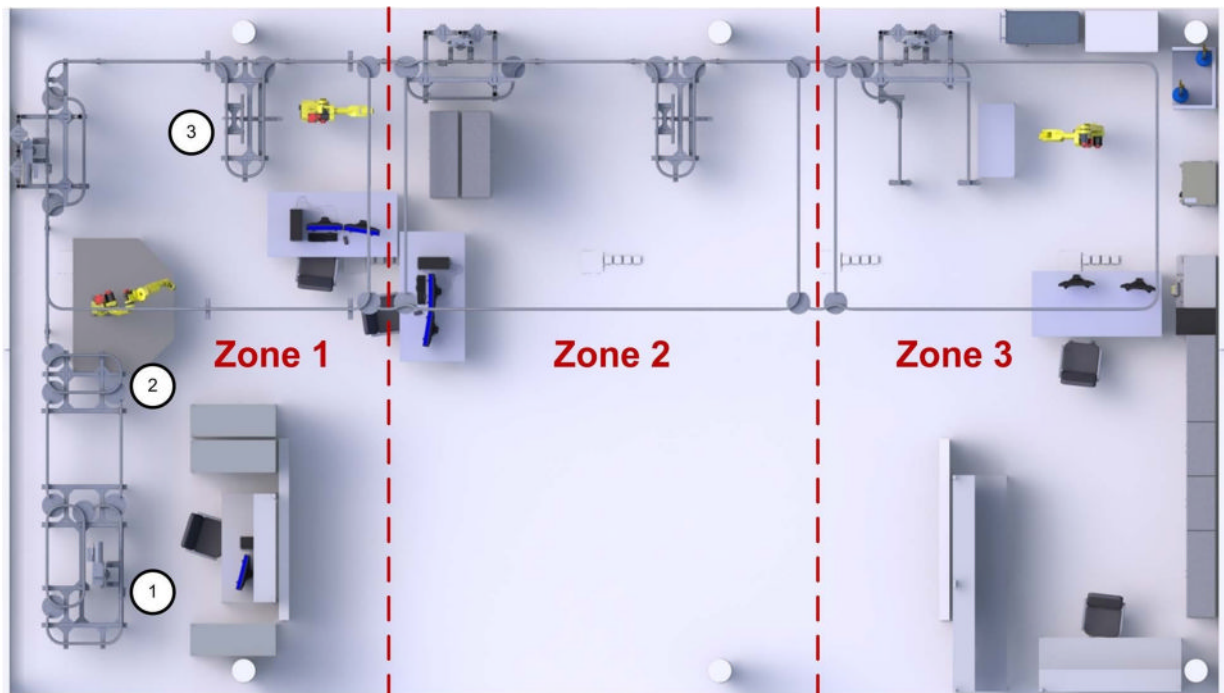
Автоматизація виробничих процесів дає багато переваг, деякі з яких включають підвищення якості, надійності та гнучкості, а також зменшення кількості нещасних випадків, виробничих витрат, часу циклу виготовлення деталей та відходів.

Ці переваги пов'язані з витратами на придбання, встановлення та програмування системи автоматизації. З цих трьох етапів найбільш трудомістким і дорогим, як правило, є програмування системи автоматизації. За оцінками Дугенске [1], інтеграція системи може коштувати до чотирьох разів більше, ніж ціна її придбання. Лукас [2] стверджує, що "написання логіки, яку має виконувати система", є найдорожчою і найтривалішою частиною. Джемс [3] підрахував, що протягом усього терміну служби виробничого підприємства більше третини всіх понесених витрат можна віднести на встановлення та налагодження. Також можна зробити висновок, що модернізація або зміна конфігурації існуючої системи автоматизації не менш затратна з точки зору часу на програмування. В епоху, коли гнучке виробництво користується все більшим попитом, можливість швидкого і доступного перепрограмування і реконфігурації системи автоматизації набуває все більшого значення.

1 АНАЛІТИЧНИЙ РОЗДІЛ

1.1 Виклик

Для того, щоб дослідити проблеми, пов'язані з програмуванням, перепрограмуванням та переналаштуванням системи промислової автоматизації, макет Лабораторії робототехніки та автоматизації W325 (рис. 1.2) у Коричневому корпусі Гірничої школи Колорадо був розглянутий як макет невеликого виробничого підприємства. Лабораторії було передано значну кількість компонентів Montrac, з яких було сконструйовано систему транспортування матеріалів. W325 можна умовно розділити на три "зони", кожна з яких містить від 2 до 3 боксів для роботів. Передбачається, що в різних зонах відбуватимуться різні дослідницькі проекти. Бачення лабораторії полягає в тому, щоб роботи в кожній зоні були пов'язані з власною системою транспортування матеріалів (СТМ) як на рівні підлоги, так і зі сховищем на рівні стелі, що дозволить проводити невеликі експерименти і сприятиме дослідницькій роботі. За бажанням, будь-яка комбінація зон може бути об'єднана таким чином, що вся лабораторія може бути використана для більш масштабних симуляцій та експериментів, наприклад, для багатоетапного виробничого процесу. Наразі побудовано лише невелику кількість незалежних установок Montrac (ті, що пронумеровані та названі на рис. 1.1).



1 позначає місце розташування BARD, 2 позначає цикл характеристики контейнера, а 3 - цикл перевірки

Рисунок 1.1: Вид зверху вниз пропонованої СТМ у W325.

Демонстраційна установка з базової автоматизації та робототехніки (BARD), як показано на рис. 1.3, наразі складається з двох візків та низки рухомих і нерухомих компонентів доріжки. BARD та інші діючі установки Montrac контролюються програмованими логічними контролерами Allen-Bradley (ПЛК), які запрограмовані в логіці сходів (LL). Проста демонстраційна програма на BARD наразі вимагає 23 рядки коду. Екстраполюючи, припускаючи, що кожна зона містить трьох роботів, наземну доріжку, а також стелю для зберігання і обхідну доріжку, виходячи з довжини коду BARD, можна спрогнозувати, що для однієї зони легко перевищить 200-300 рядків LL-коду. Сюди входили б протоколи безпеки, правила взаємодії між наземним і стельовим шлейфом і сенсорним екраном користувацького інтерфейсу.

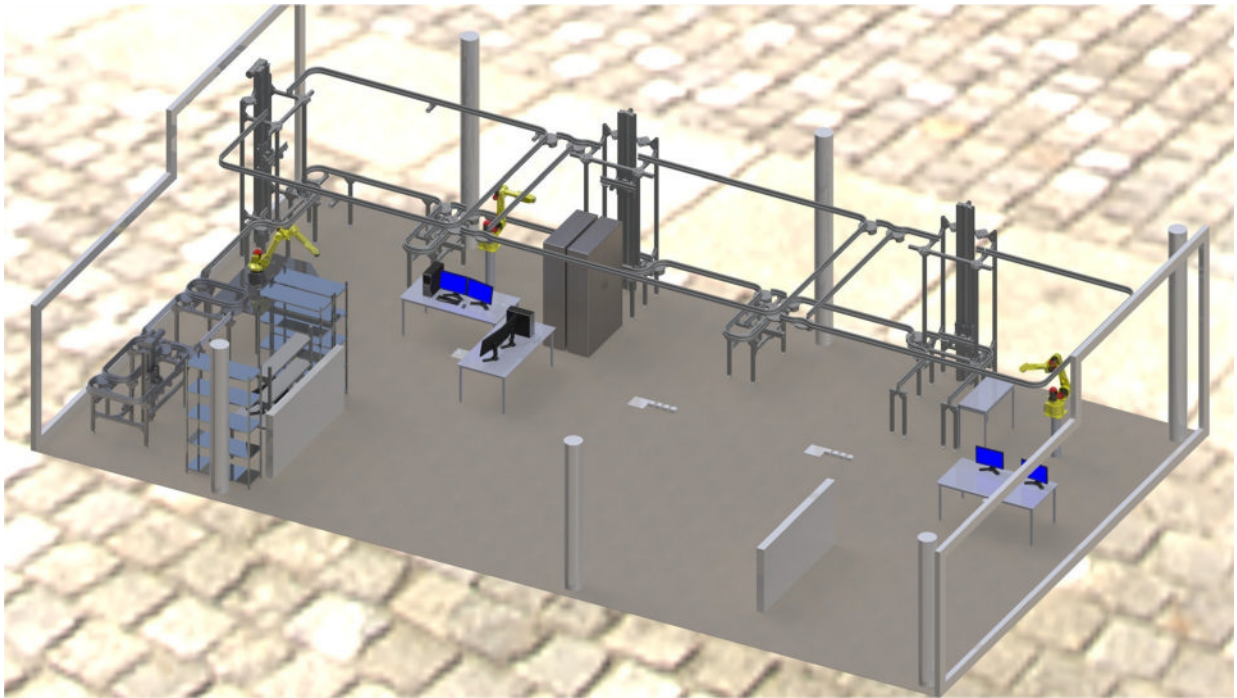


Рисунок 1.2: Ізометричний вигляд запропонованої СТМ у W325.

На перший погляд, програмування або модифікація 300 рядків LL-коду не здається складним завданням, якщо не взяти до уваги низку особливостей LL-програм. По-перше, зі збільшенням розміру та кількості рядків LL-програми читабельність, масштабованість та ремонтпридатність такого коду зменшується в геометричній прогресії. Саме через це в індустрії стандартною практикою LL-програмістів є переписування коду з нуля, а не спроби прочитати, виправити чи модифікувати існуючий код. По-друге, хоча кожен рядок LL-коду може викликати по суті необмежену кількість входів, він може контролювати або змінювати лише один вихід. Можна використати аналогію з ланками або кільцями в кольчужі, де костюм в цілому - це програма LL, а кожна ланка - це рядок коду LL. Для того, щоб змінити розмір однієї ланки (рядка коду LL) і зберегти функціональність костюма, кожна з навколишніх ланок також може потребувати модифікації. Модифікації швидко ускладнюються і можуть перетворитися на вічний цикл.

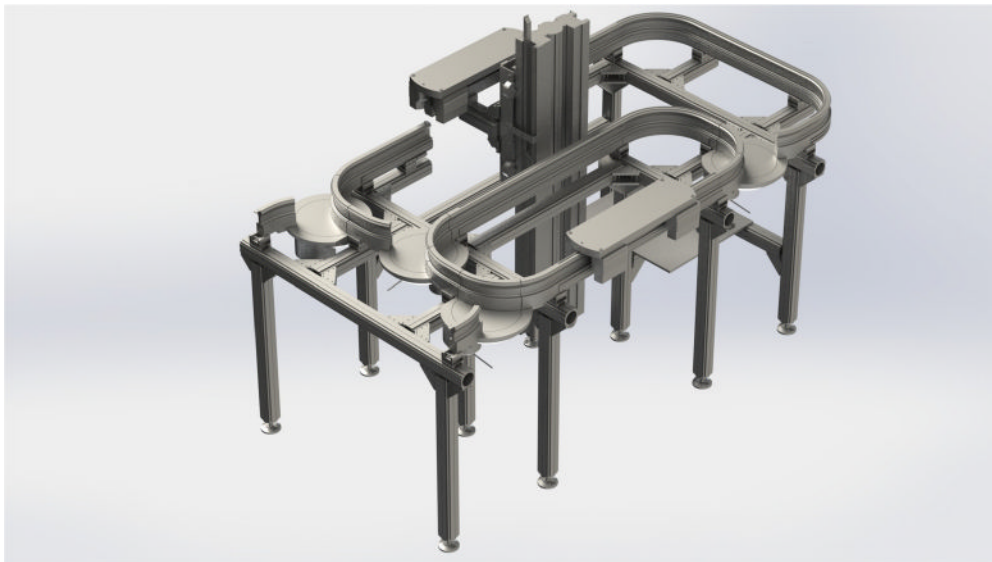


Рисунок 1.3: Візуалізація BARD.

Розглянемо невелику систему автоматизації, яка потребує близько 3000 рядків LL- коду, наприклад, проект, досліджений Лукасом та ін. [2], вимагав до 100 000 рядків. Програміст з досвідом роботи понад 30 років може в середньому створювати один рядок LL-коду кожні 5 хвилин. Згідно з нещодавнім опитуванням, проведеним Американським товариством цивільних інженерів [4], базовий дохід середньостатистичного інженера становить близько 96 000 доларів США. Якщо припустити 1928 робочих годин на рік, ми отримуємо середню зарплату в розмірі 50 доларів за годину. Для кодування 3000 рядків знадобилося б 250 годин, що коштувало б 12 500 доларів США. Для проекту, який вимагав 100 000 рядків коду, лише програмування зайняло б 3 місяці. Слід зазначити, що оціночні 250 годин охоплюють лише написання програми, будь-яке налагодження, перевірка та валідація програми, а також встановлення необхідного обладнання потребують додаткових ресурсів. Запланована СТМ для W325 з її приблизно 900 рядками коду займе щонайменше 75 годин для програмування. Зважаючи на те, що метою проектування є полегшення швидкої реконфігурації зазначеної СТМ для численних дослідницьких проектів, ситуація швидко стає нездійсненною з точки зору планування для дослідницьких цілей.

Починаючи з 1994 року, були зроблені різні спроби автоматизувати програмування виробничих систем (в основному ПЛК). У зв'язку з тим, що більшість ПЛК в промисловості працюють на LL-кодi, спроби

автоматизованого створення програм в основному були зосереджені в цьому напрямку (LL). Всі досліджені методи вимагають або діаграм станів [5], або компонентних моделей [6], або високорівневих системних моделей [7], або мереж Петрі [8-10]. Всі ці методи вимагають від користувача різного рівня спеціалізованої підготовки. Що ще важливіше, всі ці методи нічого не роблять для забезпечення можливості реконфігурації або перепрограмування системи в доступний і своєчасний спосіб.

1.2 Нова парадигма

Зрозуміло, що для створення системи, яка швидко реконфігурується або динамічно самооптимізується, як того вимагають сучасні потреби, жорстка ієрархічна структура ПЛК може виявитися не найефективнішим варіантом. Замість цього розглянемо концепцію рою автономних систем автоматизації (АСА). Кожен рухомий компонент в контурі Montrac (стрілочна доріжка, ліфти, перевантажувальні ворота, робочі станції і т.д.) визначається як повна і самодостатня АСА. Кожна АСА має власну функцію прийняття рішень і, базуючись на обмежених вхідних даних, приймає локально оптимальні рішення в межах своєї сфери впливу, створюючи загальну надійну систему, що динамічно оптимізується, з низькою індивідуальною складністю. Оскільки кожен компонент АСА здатен працювати незалежно від інших, під час роботи системи можна буде додавати або видаляти сегменти системи і переналаштовувати її в режимі реального часу, фактично надаючи системі можливість "plug-and-play".

1.3 Описання завдання

Виклик полягає в тому, щоб:

- Створити алгоритми для використання різними класами ААС, які можна застосувати до сучасного промислового обладнання. До таких класів можна віднести:
 - робочі станції
 - стрілочні перемикачі

- ліфти
- перехрестя
- лінійні стрілочні переводи;
- Розробити метод моделювання макетів СТМ, до яких можна застосувати алгоритми.
- Розробити графічний інтерфейс користувача, який дозволить мінімально підготовленому користувачеві ввести загальний план поверху і визначити, як кожна машина взаємопов'язана з іншими;
- Визначити протоколи зв'язку, за допомогою яких кожна АСА взаємодіє з іншими;
- Визначити метод, за допомогою якого система в цілому та компоненти ААС окремо визначають, яке рішення є оптимальним;
- Створити додаткову функціональність для графічного інтерфейсу, дозволивши відображати статистику системи в режимі реального часу.

2 НАУКОВО-ДОСЛІДНИЦЬКА ЧАСТИНА

2.1 Автоматизовані виробничі системи та системи автономної автоматизації

Для цілей цього дослідження автоматизовані виробничі системи (АВС) можна розділити на дві підсистеми: робочі станції (РС) та систему транспортування матеріалів (СТМ). Робочі місця визначаються як будь-яке місце, де виконується робота над виробами, що виготовляються та/або обробляються, і розглядаються як системи з чорним ящиком. Вони можуть включати перевірку, модифікацію та складання. Під СТМ розуміються всі компоненти, які сприяють переміщенню виробів між робочими місцями. Парадигма, яка буде застосована до СТМ в цій статті, є апаратно-незалежною, оскільки введені концепції можуть бути застосовані, в межах розумного, до будь-якої форми СТМ, такої як конвеєрна стрічкова система, моно- або мультирейкова система, автоматизовані моторизовані транспортні засоби або багатороторні транспортні засоби. СТМ може бути розбита на ряд складових частин. Однак, для підтвердження та ілюстрації, в якості прикладу буде використана монорейкова транспортна система.

Деякі з компонентів, які можуть бути об'єднані для формування СТМ, показані на рис. 2.1. З цих компонентів СТМ пряма доріжка (рис. 2.1(б)) та крива доріжка (рис. 2.1(в)) є статичними і, отже, вважаються латентними компонентами СТМ. Решта компонентів, окрім візка (рис. 2.1(а)), здатні впливати на зміну стану СТМ і, отже, кожен з них розглядається як окремий тип АСА. На основі порядку потоку, що зберігається у візку, даних від відповідних РС (час процесу, довжина черги, зайнято/доступно) та інформації від інших АСА, кожна АСА здатна приймати рішення про те, як спрямовувати потік візків/заготовок по всій системі.

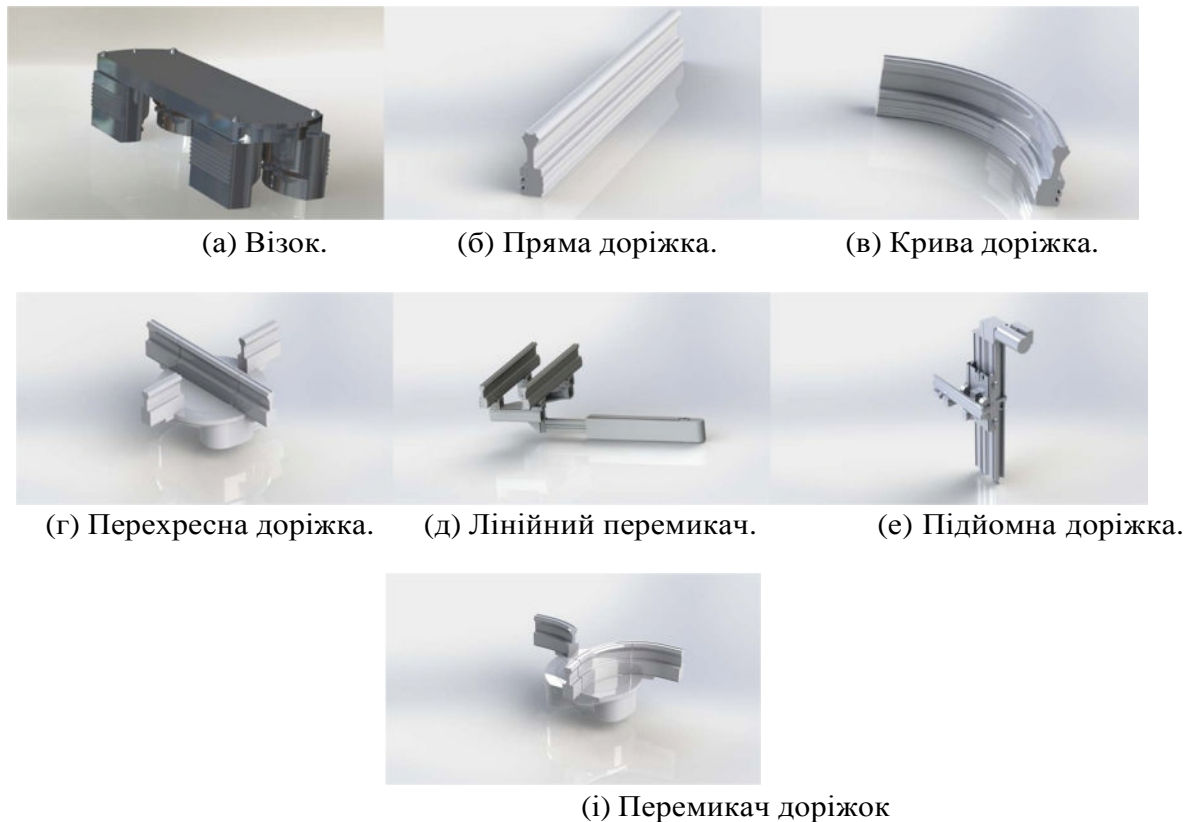


Рисунок 2.1: Компоненти Montrac.

2.2 Тенденції в автоматизації

Переваги автоматизації заводів для виробництва та виробничих цілей добре задокументовані. Згідно з дослідженням, проведеним компанією Meredith [11] у 1987 році, автоматизація заводу може призвести до скорочення кількості верстатів на 87%, трудовитрат - на 20-90%, загального часу процесу - на 75-97%, часу виконання замовлення - на 25%, прибутковості системи - втричі, а операційного часу - на стільки ж. Інші переваги автоматизації включають значне поліпшення якості, більш точні дати поставок, більш надійний процес тощо.

Проблеми та витрати, пов'язані з автоматизацією заводу або будівництвом автоматизованого заводу з нуля, також добре задокументовані. За словами Meredith, автоматизація фабрики займе щонайменше 12-18 місяців, якщо припустити, що впровадженням займаються досвідчені фахівці. У 2000 році Дугенске [1] стверджував, що проектування та впровадження нової заводської інформаційної системи може зайняти до двох років, що в деяких випадках набагато довше, ніж життєвий цикл технології виробництва

продукції. Він також підрахував, що "вартість інтеграції типової заводської інформаційної системи в чотири рази перевищує вартість придбання самої системи". Лукас [2] стверджує, що з усіх витрат на автоматизацію заводу або проектування і будівництво нового автоматизованого заводу найдорожчим і найбільш трудомістким є написання керуючого коду. У 2005 році Джеймс [3] виявив, що протягом усього терміну експлуатації виробничого підприємства третина загальних витрат припадає на встановлення та налагодження обладнання. Джеймс також зазначив, що інша значна частина операційних витрат пов'язана з простоями на технічному обслуговуванні, і що якщо завод необхідно адаптувати до випуску нової продукції шляхом зміни технологічного процесу та/або заміни застарілого або неконкурентоспроможного обладнання, то час простою та витрати на встановлення різко зростають.

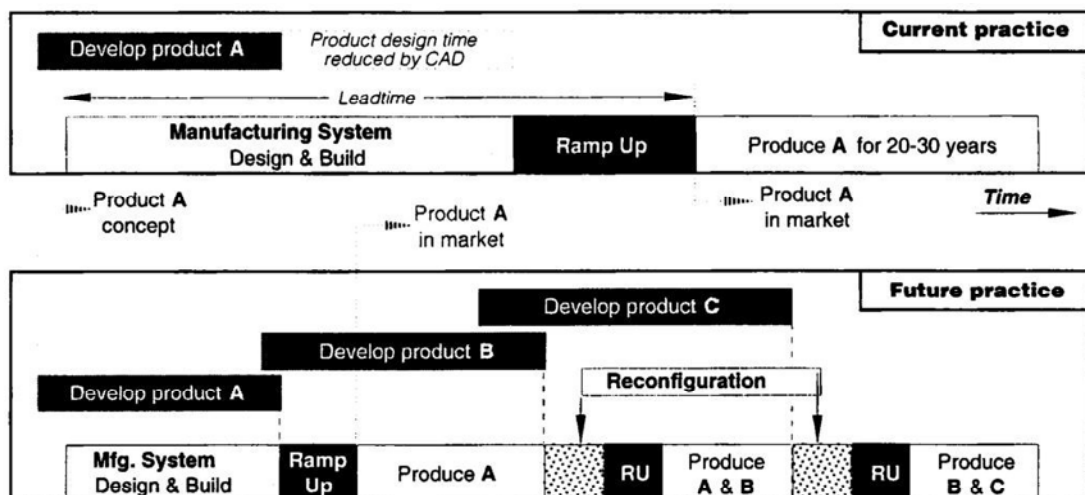


Figure 1: While product development time was reduced dramatically by CAD, nothing equivalent was done with the manufacturing system (top). Increase in frequency of new products introduction requires shortening the manufacturing system design time, and enabling its adaptation to production of new products through rapid reconfiguration.

Рисунок 2.2: Графік, що показує зміну термінів виготовлення продукції [12].

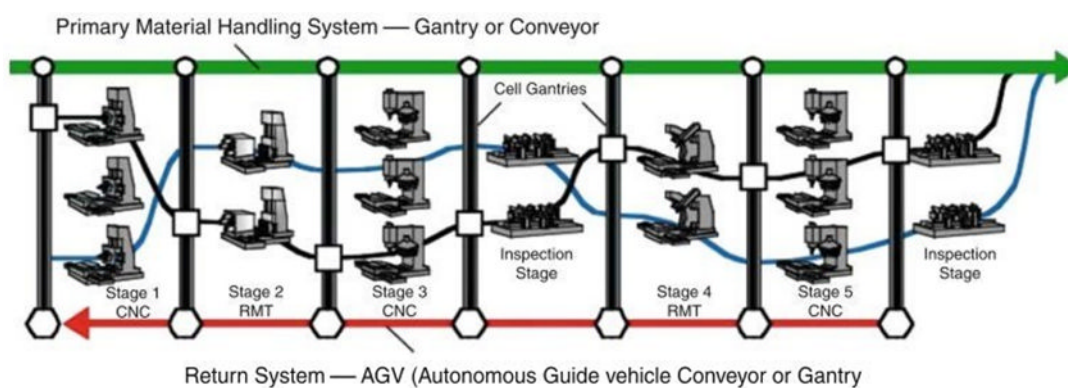
Корен [12] задокументував, що більшість переробних галузей промисловості використовують для виробництва своєї продукції спеціальні виробничі лінії (СВЛ). СВЛ складаються з недорогої стаціонарної автоматизації і використовуються для виробництва основної продукції компанії. Як правило, кожна СВЛ призначена для швидкого та масового виробництва одного компонента. При роботі на повну або близьку до повної потужності собівартість однієї деталі є низькою. Цей підхід найкраще відповідає типу ринкового попиту, показаному у верхній частині рис. 2.2. Однак, як зазначає

Корен [12], вже існує зростаюча потреба у виробничій системі, яка була б більш реконфігурованою та гнучкою через:

- збільшення частоти появи нових продуктів,
- зміни в деталях для існуючих продуктів,
- значні коливання попиту на продукцію та її асортименту,
- зміни в державному регулюванні,
- зміни в технологічних процесах.

Це свідчить про необхідність того, щоб часова шкала продуктів була більш схожою на нижню половину рис. 1.2. компанії повинні були мати можливість швидко та економічно ефективно реагувати на зміни. Було запроваджено новий тип виробничих ліній, який отримав назву "гнучкі виробничі системи" (ГВС). ГВС були розроблені для виробництва різноманітної продукції, що дозволяло змінювати обсяги та асортимент. ГВС зазвичай склалися з дорогих універсальних верстатів з числовим програмним управлінням (ЧПУ) та інших програмованих засобів автоматизації. Слід зазначити, що пропускна здатність ГВС набагато нижча, ніж СВЛ. Також, як видається, існує низький рівень сприйняття та задоволеності роботою ГВС. Це незадоволення виникло через високу вартість, а також часто надлишковий і невикористаний капітал, що було пов'язано з тим, що верстати з ЧПК були розроблені для загальних цілей. Корен [12] запропонував новий клас систем, які він назвав реконфігурованими виробничими системами (РВС) [12, 13], які повинні поєднувати високу пропускну здатність СВЛ з гнучкістю ГВС. Це може бути досягнуто шляхом проектування системи та машин з регульованою структурою для забезпечення масштабованості системи. Приклад показано на рис. 2.3. Структура повинна регулюватися на рівні системи шляхом додавання або заміни машин, а також на рівні машини шляхом зміни апаратного забезпечення машини і програмне забезпечення для управління. Вартість РВС також можна знизити, якщо розробляти її на основі сімейства деталей, а не однієї окремої деталі. З того часу термін РВС став стандартизованим і був включений до Енциклопедії виробничого інжинірингу CIRP 2014 року [14]. Згідно з Гебхардтом [14], ідеальна РВС повинна володіти наступними шістьма характеристиками: модульність, інтегрованість, індивідуальна гнучкість, масштабованість, конвертованість та діагностуємість.

У 2001 році Сілвера [15] опублікував статтю, присвячену концепції масової кастомізації, в якій стверджував, що зростає бажання компаній мати можливість надавати кастомізовані продукти або послуги за допомогою гнучких сервісів у потенційно великих обсягах за відносно низьких витрат для задоволення конкретних потреб окремих клієнтів. У 2006 році Шен [16] повторив висновки Корен і Сілвера, заявивши, що вимоги клієнтів, які швидко змінюються, змушують змінювати стилі виробництва. Традиційні централізовані та послідовні механізми планування, складання графіків і контролю виробничих процесів були недостатньо гнучкими, щоб відповідати на змішані та малотиражні умови виробництва.. Традиційні підходи обмежували можливості розширення та реконфігурації, а ієрархічна організаційна структура, що використовувалася, також була вразливою до



Reconfigurable Manufacturing System, Fig. 1 Schematic diagram of a typical RMS каскадних збоїв, спричинених одномоментними відмовами.

Рисунок 2.3: Приклад реконфігурованої виробничої системи [14].



(а) Фактори, що визначають динамічну адаптацію виробничих структур.

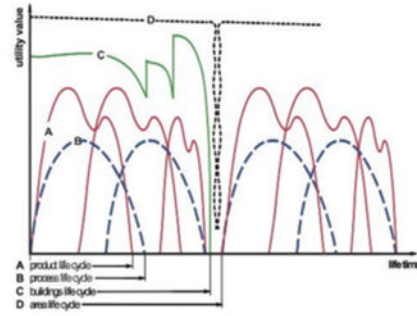


Figure 1.2: Diverging life cycles of the constituent elements of a factory (Wirth).

(б) Розбіжність життєвих циклів складових елементів заводу.

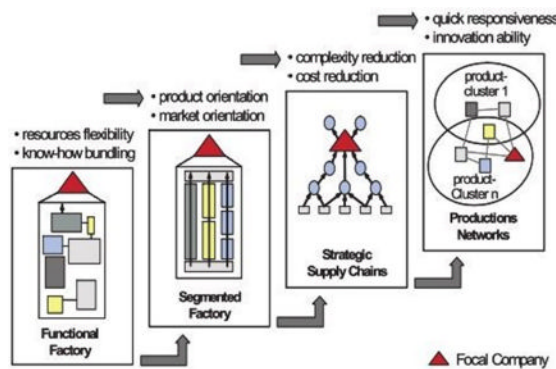


Figure 1.3: Evolution of factories (H-P Wiendahl/ Hernandez).

(в) Еволюція фабрик.

Рисунок 2.4: Рисунок за Віндалем [17].

Віндаль [17] повторив ці ж висновки у 2007 році в комплексній роботі, в якій запропонував класифікацію перемінного виробництва, що включає ГВС та РВС. Він класифікував внутрішні та зовнішні фактори, що зумовлюють динамічну адаптацію промислового виробництва, на рис. 2.4(а). Віндаль стверджує, що "з'являється все більше доказів того, що на зміну епосі масового виробництва приходить епоха ринкових ніш", і що "ключем до створення продуктів, здатних задовольнити потреби диверсифікованої клієнтської бази, є короткий цикл розробки, що дає змогу виробляти недорогі, високоякісні товари в достатній кількості для задоволення попиту". Віндаль також відзначив зміни в життєвих циклах складових елементів фабрики на рис. 2.4(б) що узгоджується зі спостереженнями, зробленими раніше Корен на рис. 2.2. Щоб пристосуватися до зростаючих потреб у все більш індивідуалізованих продуктах, потрібно швидко впроваджувати нові продукти та/або модифікувати

чи оновлювати існуючі продукти. Віндаль задокументував те, що він назвав "еволюцією фабрик" на рис. 2.4(в).

Таблиця 2.1: Типи виробничої гнучкості[17].

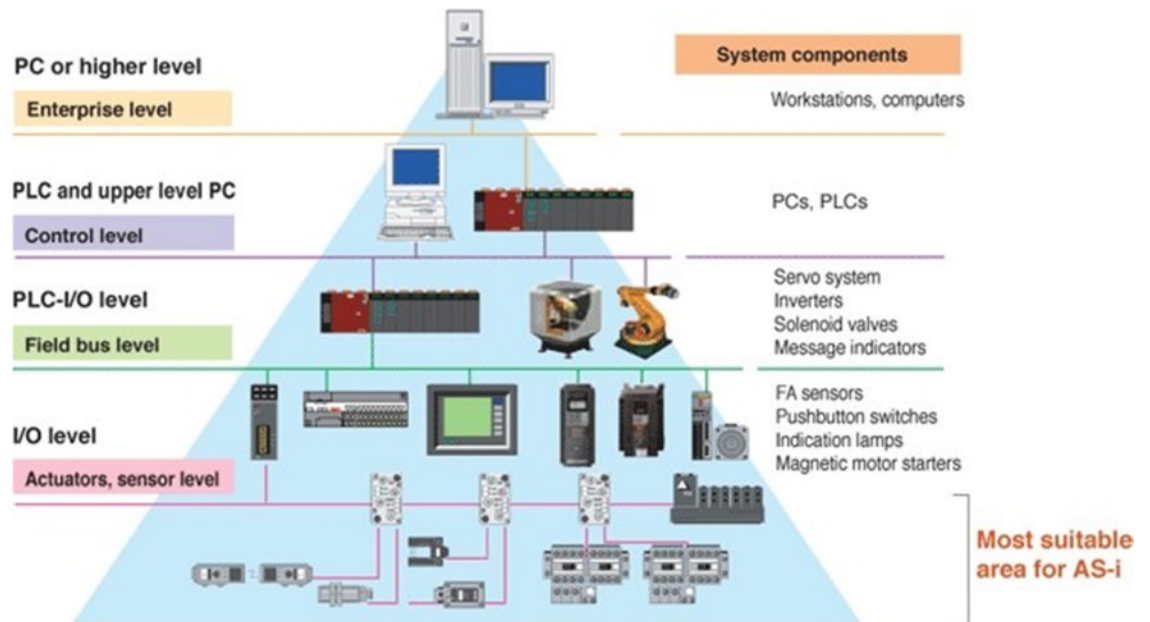
Типи гнучкості	Опис
Машина.	Різноманітні операції, що виконуються без зміни налаштувань.
Поводження з матеріалами	Кількість використаних доріжок на загальну кількість можливих доріжок між усіма машинами.
Операція	Кількість різних планів обробки, доступних для виготовлення деталей виробництва.
Процес	Набір типів деталей, які можуть бути виготовлені без значних змін у налаштуванні, тобто гнучкість комбінації деталей.
Продукт	Легкість (час і вартість) впровадження продуктів в існуючий асортимент продукції.
Маршрутизація	Кількість можливих маршрутів для всіх типів деталей або кількість типів деталей.
Обсяг	Можливість вигідно змінювати обсяги виробництва в межах виробничих потужностей.
Розширення	Простота (зусилля та витрати) збільшення потужності та/або спроможності, за необхідності, через фізичні зміни в системі.
Програма управління	Здатність системи працювати практично безперервно (напр., під час другої та третьої зміни) завдяки наявності інтелектуальних машин та програмного забезпечення для управління системою, через фізичні зміни в системі.
Виробництво	Кількість усіх типів деталей, які можна виготовити без додавання великого капітального обладнання.

Ця еволюція почалася з функціональної фабрики, яка мала дуже гнучкі ресурси і була пристосована до змін в асортименті та обсягах продукції, але мала довгі терміни постачання і великі запаси. Потім з'явилися сегментовані фабрики, які забезпечували буфери для напівфабрикатів та складальні зони і були організовані у вигляді фракталів або сегментів. Зовсім недавно з'явилися виробничі мережі. Далі Віндаль класифікує змінність і 10 типів гнучкості виробництва, які перераховані в Таблиці 2.1. З перелічених у Таблиці 2.1 типів ця теза зосередиться на гнучкості обробки матеріалів та гнучкості

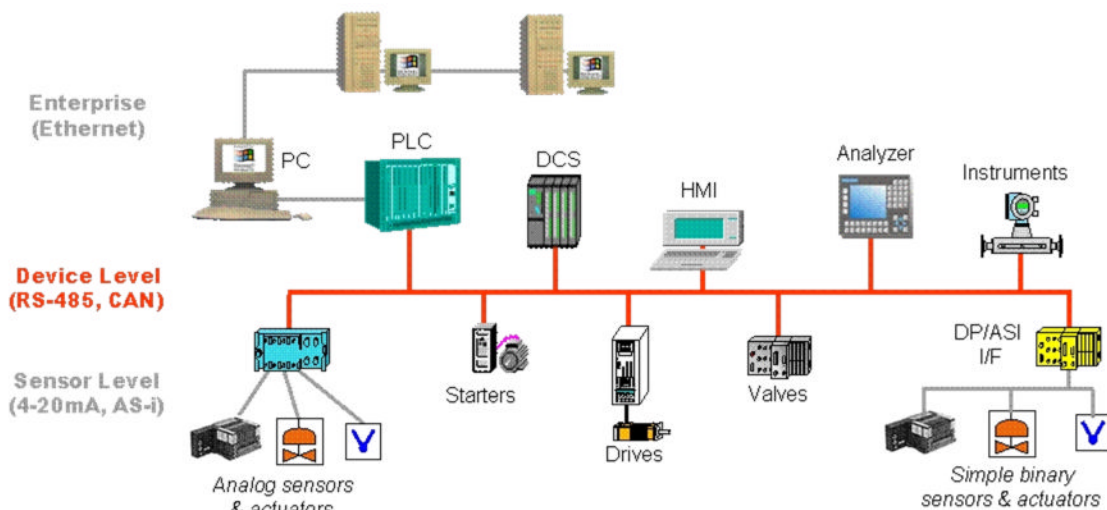
маршрутизації. Як зазначалося вище, існує зростаючий попит на швидке переналаштовувана та перепрограмована виробнича система або макет.

2.3 Вступ та прогрес у сходянкій логіці

Системи автоматизації та транспортування матеріалів зазвичай працюють на промислових стандартних напругах (наприклад, 24, 48 або 120 В постійного струму) і взаємодіють переважно за допомогою двійкових сигналів. Наприклад, датчик наближення може надсилати сигнал 24 В, щоб показати, що він виявив присутність заготовки, і 0 В в іншому випадку. Для керування лінійним переводом може знадобитися серія сигналів 24 В. Один сигнал - для розблокування доріжки з поточних точок з'єднання, інший - для сигналу приводу про переміщення в нову позицію, а третій - для фіксації доріжки в новому положенні. СТМ промислового масштабу може мати сотні, якщо не тисячі, входів і виходів (ІО), які зазвичай контролюються ПЛК. ПЛК прийшли на зміну дротовим електромеханічним реле та перемикачам наприкінці 1960-х років[18, 19]. ПЛК, як правило, складається з декількох компонентів, які включають процесор, модуль живлення, модулі вводу/виводу та додаткові комунікаційні модулі. ПЛК зазвичай працюють в ієрархічній системі (рис. 2.10), часто з одним процесорним блоком (з декількома резервними), що діє як головний контролер, пов'язаний з різними підлеглими процесорами. Кожен підпроцесор підключений до банків модулів вводу-виводу або модулів, які дозволяють їм обмінюватися даними або керувати стороннім обладнанням. Ці комунікації відповідають різним промисловим стандартам, включаючи PROFIBUS, PROFINET, Industrial Ethernet та Device Net.



(а) Приклад 1. Зображення надано з сайту www.fujielectric.co.jp. [22]



(б) Приклад 2. Зображення надано з сайту www.cnx.org. [23].

Рисунок 2.10: Приклади ієрархії ПЛК.

У 1993 році Міжнародна електротехнічна комісія (ІЕС) опублікувала міжнародний стандарт ІЕС 61131 для ПЛК. Стандарт охоплює вимоги до обладнання, мови програмування, протоколи зв'язку, функціональну безпеку тощо. ІЕС 61131-3 містить перелік промислових стандартів на мови програмування для ПЛК. Існує п'ять міжнародно визнаних мов програмування для ПЛК, а саме:

- переліки інструкцій;
- структурований текст;
- сходинокві діаграми або сходиноква логіка;

- функціональні блок-схеми;
- послідовна діаграма функцій.

З вищезгаданих мов програмування, згідно з Лукасом [2], Манесісом [5] та Ханом [20], найпопулярнішим методом створення логіки управління для промислових виробничих ліній є сходинокві діаграми. Отже, в решті цієї тези ми зосередимось на LL-кодi.

2.3.1 Логіка сходів

Логіка сходів дуже схожа на електричні схеми сходів і може розглядатися як система, заснована на правилах. Приклад гаражних воріт, керованих ПЛК, показано на рис. 2.5, а просте рішення показано на рис. 2.6. Задача з гаражними воротами є поширеною домашньою задачею при вивченні програмування на LL.

Розглядаючи рис. 2.5, можна побачити, що система має 5 входів і 5 виходів, які перераховані в Таблиці 2.2. Перше завдання полягає у написанні коду, який досягає наступних результатів:

- Кнопки відчинення та зачинення призначені для керування рухом дверей. Рух не повинен продовжуватися, коли будь-яка з кнопок відпускається, тому в цій вправі не потрібен і не використовується вимикач "Стоп". Однак, всі інші доступні входи і виходи використовуються в цій вправі.
- Натискання на кнопку відчинення має призвести до того, що дверцята піднімуться вгору (відчиняться), якщо вони ще не повністю відчинені. Процес відчинення повинен тривати доти, доки вимикач утримується в натиснутому стані. Якщо вимикач відпустити, або якщо спрацює кінцевий вимикач LS1, рух воріт повинен негайно припинитися.
- При натисканні на кнопку закриття дверцята повинні опуститися (зачинитися), якщо вони ще не повністю зачинені. Процес зачинення повинен тривати доти, доки вимикач утримується в натиснутому положенні. Якщо вимикач відпустити, або якщо кінцевий вимикач LS2 замкнеться, рух воріт повинен зупинитися негайно.

- Якщо дверцята вже повністю відчинені, натискання перемикача відчинення не повинно вмикати двигун.
- Якщо дверцята вже повністю зачинені, натискання на кнопку закриття не повинно вмикати двигун.
- За жодних обставин не можна подавати напругу на обидві обмотки двигуна одночасно.
- Індикатор відчинення повинен світитися, якщо дверцята знаходяться у повністю відчиненому положенні.
- Індикатор закриття повинен світитися, якщо дверцята знаходяться в повністю зачиненому положенні.

Перш ніж продовжити, слід зазначити кілька особливостей LL-коду. По-перше, код виконується зверху вниз і зліва направо, рядок за рядком, і зазвичай виконується кілька разів на секунду. По-друге, на одне джерело вхідних даних можна посилатися кілька разів в одній програмі LL, але кожен вихід (наприклад, світло відчинених дверей на рис. 2.5, перераховане в таблиці 2.2) може бути викликаний лише один раз у всій програмі. По-третє, кожен рядок коду LL може містити лише одну команду виводу. Нарешті, вихід будь-якого окремого рядка коду буде подано тільки тоді, коли всі умови виконання в цьому рядку є істинними.

Розглядаючи код, який задовольняє вищезазначеному переліку вимог, як показано на рис. 2.6, його структура схожа на електричні схеми сходів, основна структура коду LL - це структура, схожа на сходи між двома вертикальними рейками. Кожен рядок LL-коду можна розділити на дві секції: вхідну і вихідну (символ у крайньому правому куті кожного рядка коду). Два символи у вхідній секції рядка 2 коду, показаного на рис. 2.6, - це символи "істина, якщо відкрито" та "істина, якщо закрито", відповідно. Коли ПЛК намагається виконати цей рядок коду, він перевіряє, чи повертає кінцевий вимикач зачинених дверей (який повертає 0 В, якщо вони відкриті) 0 Вольт. Якщо це так, то програма перевірить, чи кінцевий вимикач відчинених дверей (який повертає 24 В постійного струму, якщо його натиснути) повертає 24 В або сигнал "істина". Якщо обидва сигнали істинні, програма увімкне світло при відчинених дверях.

Слід зазначити, що повний опис усіх доступних символів у кодуванні LL виходить за рамки цієї тези.

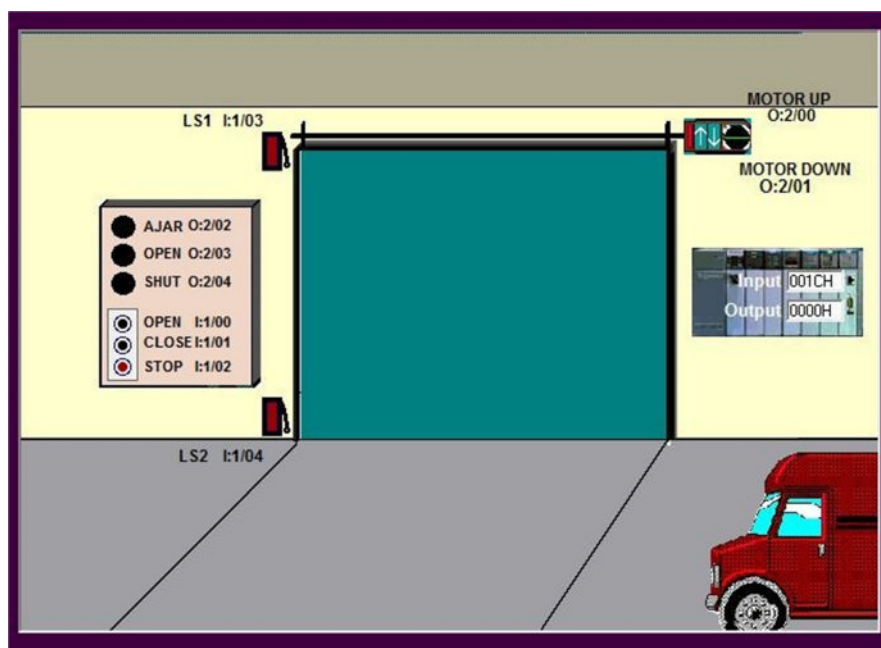


Рисунок 2.5: Налаштування гаражних воріт [21].

Враховуючи кількість вимог програми, отриманий код є відносно простим для розуміння і містить загалом 17 відношень, як показано на рис. 2.7. Якщо список вимог програми розширити до:

- Рух дверей повинен негайно зупинятися при першому натисканні на стоп-кран і залишатися зупиненим, якщо стоп-кран відпустити.
- Натискання на кнопку відчинення має призвести до відчинення дверей, якщо вони ще не повністю відчинені. Операція відчинення повинна продовжуватися до завершення, навіть якщо вимикач відпустити.
- Натискання на кнопку зачинення має призвести до зачинення дверей, якщо вони ще не повністю зачинені. Операція зачинення повинна продовжуватися до завершення, навіть якщо вимикач відпустити.
- Якщо дверцята вже повністю відчинені, натискання перемикача відкривання не повинно вмикати двигун.
- Якщо дверцята вже повністю зачинені, натискання на кнопку закриття не повинно вмикати двигун.
- За жодних обставин не можна подавати напругу на обидві обмотки двигуна одночасно.

- Індикатор привідчених дверей повинен світитися, якщо двері не перебувають у повністю зачиненому або повністю відчиненому положенні.
- Індикатор відчинення повинен світитися, якщо дверцята знаходяться у повністю відчиненому положенні.
- Індикатор закриття повинен світитися, якщо дверцята знаходяться в повністю зачиненому положенні.

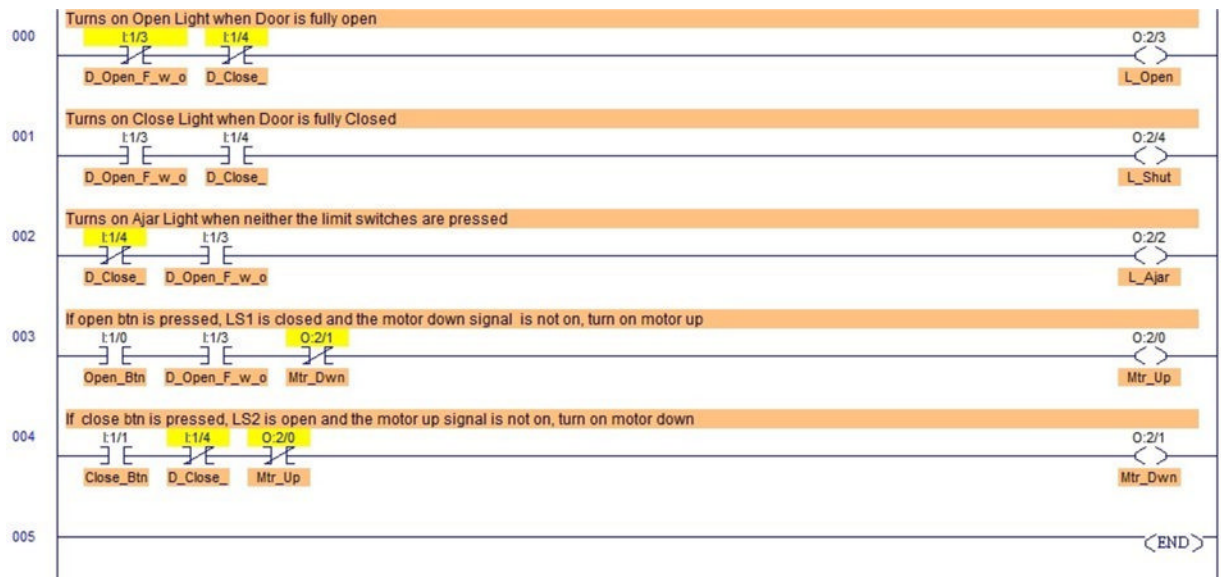


Рисунок 2.6: Рішення проблеми з гаражними воротами 1.

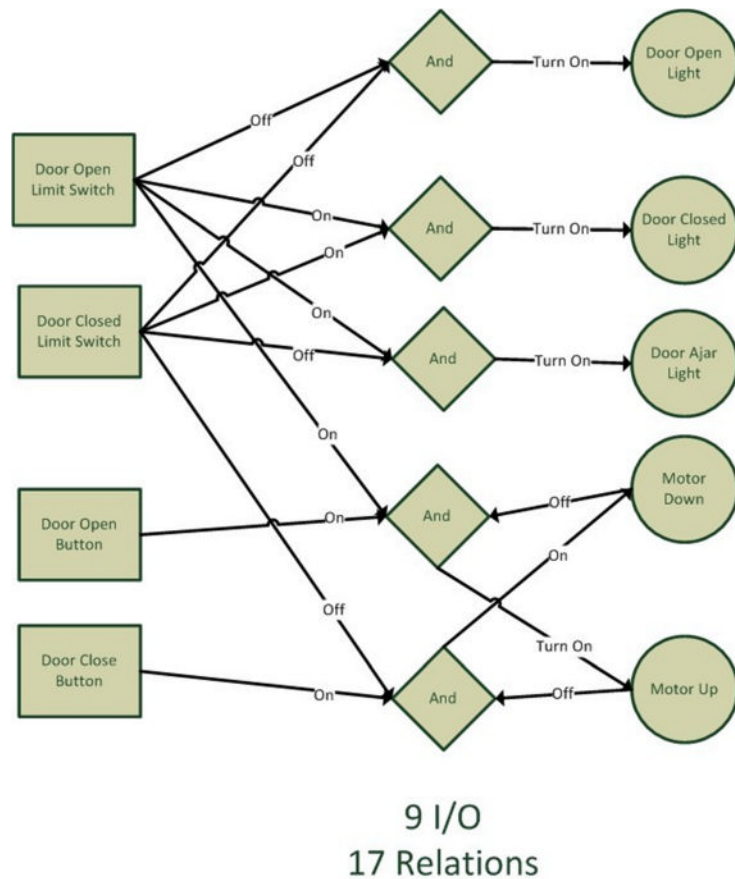


Рисунок 2.7: Діаграма зв'язків рішення 1.

Код може змінитися на щось подібне до рис. 2.8, який містить 33 повторення, як показано на рис. 2.9. При невеликій зміні специфікацій складність результуючого коду різко зростає, і, окрім рядків, які керують світловим індикатором стану, решту коду потрібно було значно модифікувати. Це узгоджується з і спостереженнями Лукаса і Дугенске, згаданими в розділі 1. Беручи до уваги цю інформацію, а також висновок, зроблений у 2.2, стає зрозуміло, що ручне перепрограмування або модифікація коду LL не задовольнить потреби у швидкій реконфігурації.

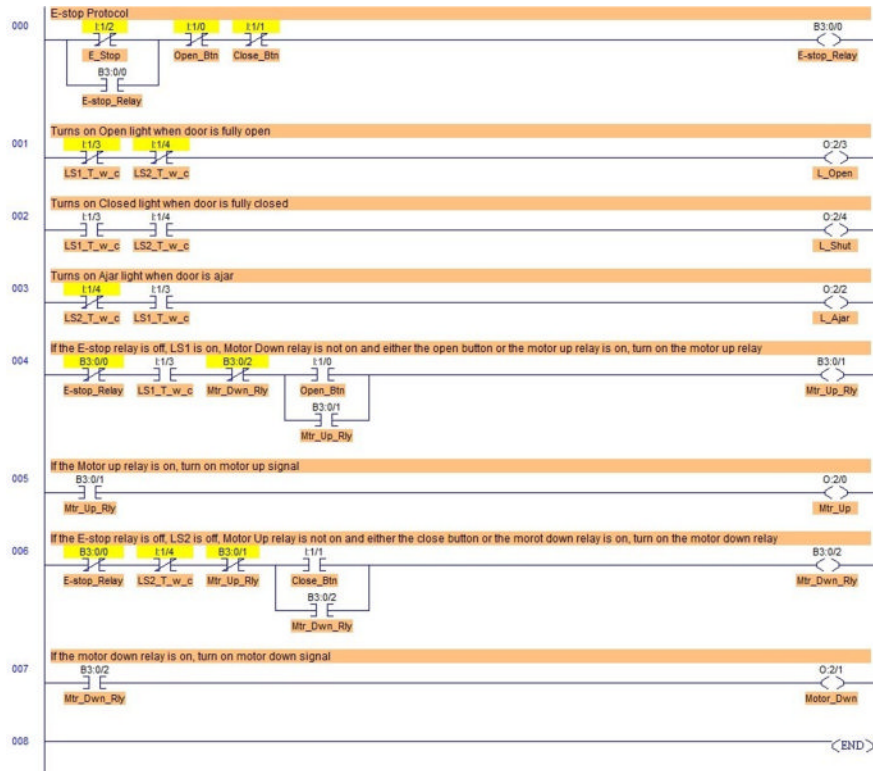


Рисунок 2.8: Рішення проблеми з гаражними воротами 2.

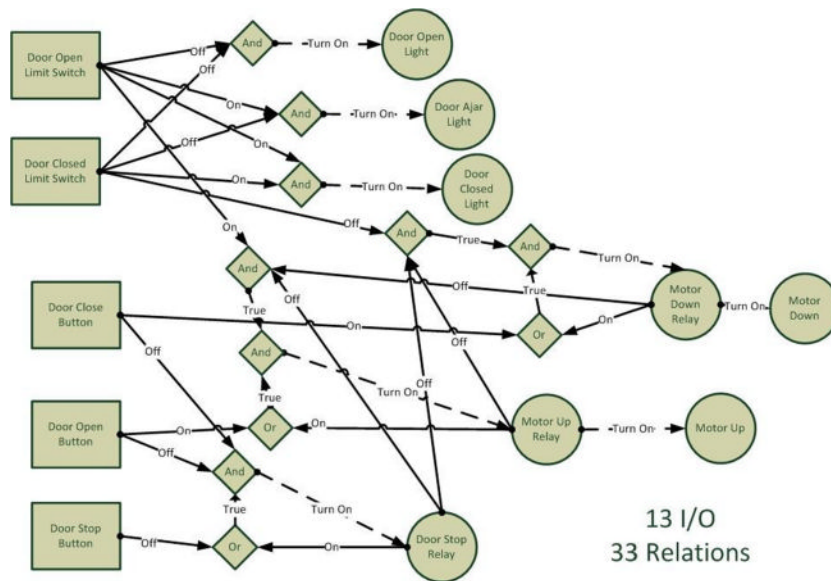


Рисунок 2.9: Діаграма зв'язків для рішення 2.

2.3.2 Спроби автоматичної генерації коду

У відповідь на необхідність скорочення часу програмування були зроблені різні спроби автоматизувати частково або повністю процес кодування. У 1994 році Джафарі [24] запропонував метод перетворення специфікації

Integrated Computer- aided Manufacturing Definition 0 (IDEF0) в проміжну програму контролера на основі мережі Петрі, а потім подальшого перетворення її в програму LL, яка може бути запущена на ПЛК. Цей метод вимагав повністю визначеної моделі IDEF0, щоб бути практичним. У 1996 році Узам [8] запропонував метод перетворення мережі Петрі з логікою передачі маркерів у програму на основі сходової логіки. У 1998 році Узам [25] представив новий тип мереж Петрі під назвою автоматизовані мережі Петрі (APN) як інструмент проектування для створення високорівневих і низькорівневих систем управління дискретними подіями (DECS), які потім будуть перетворені в TPL, а потім в сходову логіку. Слід зазначити, що коли TPL використовується для генерації LL-коду, він перевершує основні методи налагодження, які використовуються в промисловості [2].

Таблиця 2.2: Перелік входів/виходів для усунення несправностей гаражних воріт.

Вхідні дані	Адреса	Тип
Датчик відчинення дверей	I:1/03	Кінцевий вимикач
Датчик зачинення дверей	I:1/04	Кінцевий вимикач
Кнопка відчинення дверей	I:1/00	Кнопка
Кнопка закриття дверей	I:1/01	Кнопка
Кнопка "Стоп"	I:1/02	Кнопка
Результати	Адреса	Тип
Заводьте двигун.	O:2/00	Двигун
Двигун вимкнено.	O:2/01	Двигун
Освітлення дверного отвору	O:2/02	Світло
Світло відчинених дверей	O:2/03	Світло
Світло зачинених дверей	O:2/04	Світло

У 2000 році Фрей [26] представив ще один тип мереж Петрі під назвою Signal Interpreted Petri Nets (SIPN), які потім можуть бути перетворені у

відповідні сегменти коду LL. У 2003 році Юніс [27] описав метод формалізації існуючого LL-коду для полегшення модифікації та верифікації, але зіткнувся з проблемами отримання достатньої інформації про проект з різних інструментів програмування ПЛК від різних постачальників. У 2012 році Жанг [28] запропонував метод перекладу між списками інструкцій та сходовими діаграмами; однак для цього потрібно мати попередньо створену програму зі списками інструкцій. Незважаючи на вищезгадані спроби скоротити час, необхідний для створення програми LL або автоматизувати процес, жодна з вищезазначених методологій не була реалізована в промислових масштабах. Література показує, що за останнє десятиліття не було досягнуто значного прогресу в області автоматизованої генерації LL-коду.

Лукас [2] стверджує, що в промисловості зростає потреба в "машинах, якими можуть безпомилково керувати користувачі без спеціальної підготовки; машинах зі складною, правильною діагностикою проблем; машинах, які є повністю безпечними; машинах, здатних відстежувати деталі; і машинах, які можуть підключатися до систем збору даних на рівні підприємства". Він також заявив, що станом на 2003 рік машини, які зазвичай використовувалися, вимагали певного навчання і до 50% часу простою, а для розширеної діагностики був потрібен додатковий час, і ще більше часу для впровадження розширеного відстеження деталей.

2.4 Розподілені системи управління

У відповідь на зростаючу потребу у виробничих системах, здатних швидко реагувати на динамічні зміни і задовольняти потреби клієнтів на все більш динамічному і турбулентному світовому ринку, почалися дослідження низки концепцій та їх застосування в автоматизованій виробничій галузі. До них належали виробничі системи на основі агентів, холонічні виробничі системи та фрактальні виробничі системи. Традиційним методом програмування логіки виробничої системи є послідовний, коли планування процесу завершується до планування виробництва, як показано на рис. 2.11.

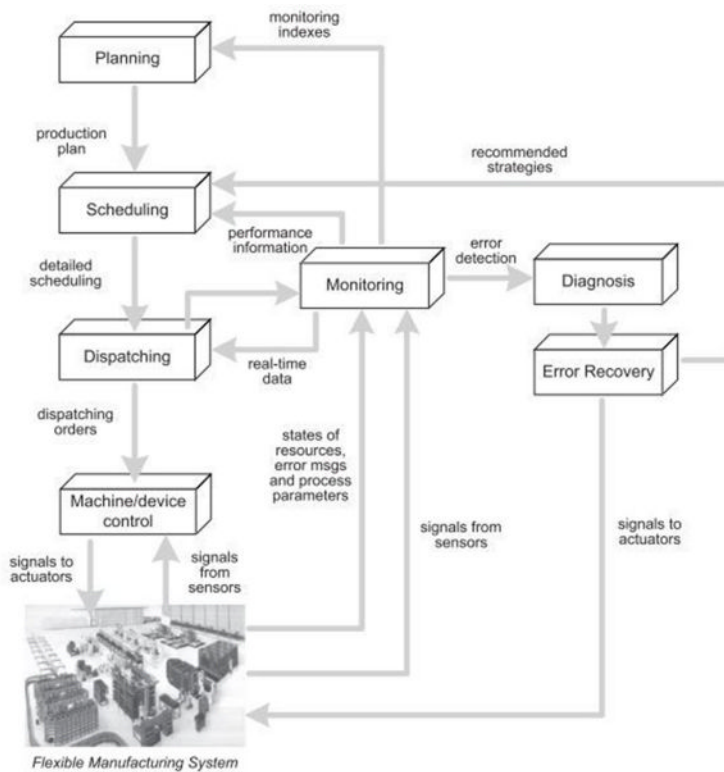


Fig. 1. Traditional approach to manufacturing control systems (adapted from MESA International, 1995 and Colombo, 1998).

Рисунок 2.11: Традиційні підходи до систем управління виробництвом[29].

2.4.1 Виробництво на основі агентів

Натхнення для агентного або мультиагентного виробництва походить зі сфери розподіленого штучного інтелекту. Існує низка визначень того, якою має бути концепція агента [31-34]. Однак, на думку Лейтао [29], підходящим визначенням буде "автономний компонент, який представляє фізичні або логічні об'єкти в системі, здатний діяти для досягнення своїх цілей і здатний взаємодіяти з іншими агентами, коли він не володіє знаннями і навичками для досягнення своїх цілей поодиночки". Лейтао стверджує, що найважливішими властивостями агента є автономність, інтелект, адаптація та співпраця, що схоже на список, який Лім [30] описав у Таблиці 2.3.

Таблиця 2.3: Властивості автономних агентів за Лімом [30].

Власність	Опис
Автономія	Агенти інкапсують деякі стани свого оточення і приймають рішення про те, що робити на основі цих станів
Реактивність	Агенти здатні сприймати навколишнє середовище і реагувати на зміни, що в ньому відбуваються

Проактивність	Агенти можуть демонструвати цілеспрямовану поведінку, беручи на себе ініціативу
Соціальні здібності	Агенти взаємодіють з іншими агентами за допомогою агентської комунікаційної мови і мають можливість брати участь у соціальній діяльності для досягнення колективних цілей

Коротко кажучи, мультиагентна система - це "сукупність агентів, що представляють об'єкти системи, здатні взаємодіяти для досягнення своїх індивідуальних цілей, коли їм не вистачає знань та/або навичок для досягнення індивідуальних цілей. Агенти організуються в гетерархічну структуру, що характеризується високим рівнем автономії та кооперації, будучи клієнт-серверна структура з фіксованими відносинами більше не застосовується [35]". Гетерархічна організаційна структура забезпечує високу стійкість до збурень і легкість розширення. Приклад такої структури можна побачити на рис. 2.12.

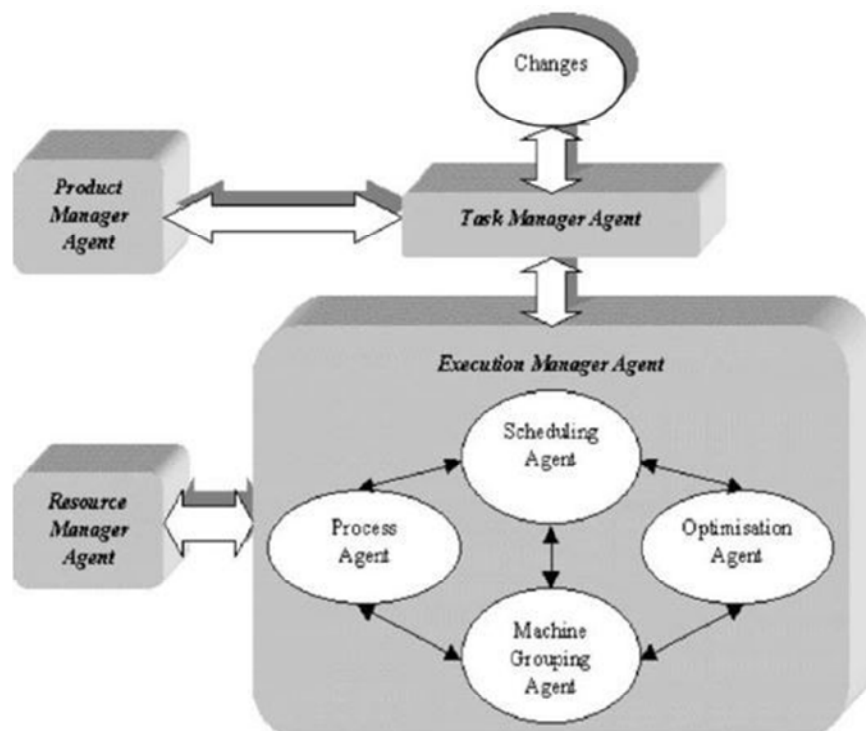


Fig. 1. The structure of the framework.

Рисунок 2.12: Структура мультиагентного фреймворку[30].

2.4.2 Холонічне виробництво

Ідея про те, що виробничі системи і надалі потребуватимуть ієрархічної структури, незважаючи на збільшення автономії, наданої окремим суб'єктам

(АСА), належить кільком авторам, а саме Варнеке [36], Окіно [37] та Брюсселю [38]. Ідея полягала в тому, щоб гарантувати вирішення конфліктів між суб'єктами і підтримувати загальну узгодженість системи. Консорціум з холістичних виробничих систем (HSM) розробив перелік визначень, щоб допомогти зрозуміти і спрямувати переклад холістичних концепцій у виробниче середовище [39]:

- **Холон:** Автономний і кооперативний будівельний блок виробничої системи для перетворення, транспортування, зберігання та/або перевірки інформації та фізичних об'єктів. Холон складається з частини, що обробляє інформацію, і часто з частини, що обробляє фізичні об'єкти. Холон може бути частиною іншого холону.
- **Автономія:** Здатність організації створювати та контролювати виконання власних планів та/або стратегій.
- **Співпраця:** Процес, за допомогою якого група суб'єктів розробляє взаємоприйнятні плани та виконує ці плани.
- **Холархія:** Система холонів, які можуть співпрацювати для досягнення певної мети або завдання. Холархія визначає основні правила співпраці холонів і тим саамим обмежує їхню автономію.
- **Холонічна виробнича система:** холархія, яка інтегрує весь спектр виробничої діяльності від бронювання замовлення до проектування, виробництва та маркетингу для реалізації гнучкого виробничого підприємства.

2.4.3 Результати та висновки

Незважаючи на те, що багато авторів демонструють численні переваги розподілених систем над традиційними системами, прийняття цих систем було однаково низьким. У 2002 році Шен [40] стверджував, що хоча агентні підходи дозволяють одночасно оптимізувати планування технологічних процесів і складання графіків виробництва, вони також значно підвищують складність задачі. Шен також зазначив, що бракує критеріїв, по-перше, для порівняння різних систем на основі агентів, а по-друге, для порівняння цих систем з

традиційними підходами. У 2006 році Монсторі [41] повторив деякі висновки, зроблені Шеном [16], стверджуючи, що хоча агентний підхід дозволяє проектувати та впроваджувати складні системи, він не зменшує зусиль, необхідних для вирішення проблеми. Він також стверджує, що безпека масштабування і якість програмного забезпечення були серйозними вузькими місцями в розвитку агентних підходів. Основними бар'єрами на шляху промислового прийняття і впровадження агентних систем були "ризик послідовної глобальної роботи, поява неминучих конфліктів між зацікавленими суб'єктами і додатковий тягар комунікації", а також труднощі "в поетапній інтеграції з існуючими застарілими системами" [16].

Щодо холонних систем, у 2006 році Бабічану [42] стверджував, що існує потреба в перевірених методологіях проектування, які були б здатні забезпечити узгодженість і надійність холонних систем, а також наочну демонстрацію покращеної продуктивності для традиційних систем управління. Щодо впровадження холонних систем управління, він також зазначив, що існує потреба в адаптації для використання з наявними та існуючими обчислювальними системами, а також у стандартизації обміну даними, внутрішніх алгоритмів і архітектури холонних систем управління.

У 2009 році Трентесо [43] перерахував п'ять основних проблем, які "заважають концепції розподіленого управління бути достатньо зрілою, щоб бути прийнятою промисловістю, навіть незважаючи на те, що очікувані переваги від підвищеної гнучкості - це те, чого вони хочуть для своїх систем". Багато з побоювань Трентесо поділяв і Лейтао [29]. Першою з них було гарантування майже оптимальних або задовільних показників у сферах економічності, ефективності (або прийняттого компромісу між ними), безпеки, відмовостійкості та уникнення тупикових ситуацій. Якщо припустити, що продуктивність не викликає сумнівів, іншим бар'єром є розробка розподіленої системи управління з урахуванням специфікацій кінцевого користувача, тобто методологія проектування. Крім того, існує проблема сумісності та розгортання розподіленої системи управління з існуючими організаційними та інформаційними системами. Наступним є питання масштабованості. І Лейтао, і Трентесо відзначають, що більшість дослідницьких розробок зупиняються на

стадії прототипу, без великомасштабних впроваджень в реальному світі, що призводить до відсутності доказів, які б доводили реальну гнучкість розподіленої системи управління. Нарешті, існує проблема людського фактору, довіри менеджерів та повернення інвестицій. У порівнянні з централізованими або ієрархічними системами управління, які добре зрозумілі, складність взаємодій, які відбуваються в розподіленій системі управління, для багатьох є складною для розуміння, підтримки та управління. Це особливо вірно, коли можливу поведінку системи неможливо легко екстраполювати на неї, а час, необхідний для повернення інвестицій, є невизначеним.

У 2011 році Фанден [44] опублікував огляд інтеграції планування процесів і складання розкладів, який знову повторив багато питань, порушених Лейтао, Трентесо, Бабічану, Шеном і Моносторі, включаючи складність системи, сумісність з існуючими системами і відсутність тестування на великомасштабних реальних реалізаціях.

Отже, незважаючи на понад два десятиліття досліджень у галузі розподілених систем управління (агентних, мультиагентних, холонічних тощо), їхнє застосування або впровадження у виробничій галузі в цілому було незначним. Отже, хоча вищезазначені питання і проблеми вивчаються і досліджуються, існує потреба в системі або парадигмі, яка допоможе в кінцевому підсумку перейти і з'єднати ієрархічний і гетерархічний світи.

2.5 Виробництво напівпровідників

Виробництво напівпровідників є дуже складним і заплутаним процесом, що включає " кілька типів робочих центрів, великі та мінливі різновиди продукції, що залежать від певної послідовності часу налаштування, повторного входу в технологічний процес і т.д "[45, 46]. Деякі з проблем, з якими стикається цей сектор, включають відсутність інтеграції між виробничими стратегіями та операційною діяльністю, підтримання послідовного виробничого потоку та врахування стохастичності у виробничому процесі [46-48]. Були спроби застосувати парадигми управління на основі агентів з обмеженим успіхом [45], однак найпоширеніші підходи базуються на

емпіричних правилах або виборі з бібліотеки емпіричних або евристичних правил [46, 49]. В результаті перегляду літератури з'явилося бажання створити масштабовану парадигму, яка б пропонувала можливості динамічної оптимізації, а також була стійкою до стохастичної поведінки системи.

2.6 Підсумки та висновки

Сходинкова логіка (LL) - це еквівалент мови асемблера для ПЛК. Вона повільна, ієрархічна та обмежена і не може задовольнити вимоги, які все більше висуваються до неї, головним чином через свою ієрархічну структуру. Багато альтернативних архітектур були б більш прийнятними. Однак, як свідчать літературні джерела, виробнича галузь в цілому повільно сприймає зміни і приймає нові або альтернативні методи. Зважаючи на вартість повної модернізації та значну вартість простою, малоімовірно, що багато нових систем отримають широке розповсюдження. Потрібна парадигма, яка забезпечить та/або імітує переваги неієрархічної структури, водночас залишаючись сумісною з ієрархічною структурою програмування, а також придатною для використання з сучасним обладнанням. Це та ніша, для якої розроблена парадигма АСА.

3 ТЕХНОЛОГІЧНА ЧАСТИНА

3.1 Вступ

У цьому розділі буде представлено теорію, що лежить в основі нової запропонованої парадигми, а також імітаційну платформу, на якій можна застосувати нову парадигму і порівняти її з іншими парадигмами.

3.2 Жадібна автономна система автоматизації

Як зазначалося в Розділі 2, існує потреба в методі емуляції ефектів і поведінки неієрархічної системи, який залишався б достатньо простим, щоб його можна було реалізувати на сучасному обладнанні. Парадигма ЖАСА має на меті забезпечити перспективний підхід, залишаючись при цьому сумісною з попередніми версіями, шляхом створення для кожного компонента прийняття рішень компактної програми, яка може працювати незалежно від усіх інших процесів. Кожна з цих програм потребуватиме лише доступу до обмеженої кількості вхідних даних і контролюватиме невелику кількість вихідних даних. Спочатку кожна автономна система автоматизації (АСА) використовує однокроковий жадібний алгоритм, який намагається перемістити візок по шляху найменшої затримки (найкоротша черга).

3.3 Огляд симулятора мережевої автономної системи автоматизації

Для того, щоб мати платформу, на якій можна було б застосовувати і тестувати нову парадигму ЖАСА, була створена програма моделювання, яка називається Симулятор мережевої автономної системи автоматизації (NAASS). NAASS виконує ряд функцій, які включають:

- Можливість створення віртуального середовища для моделювання СТМ будь-якої конфігурації, складу та розміру, усуваючи різні ресурси, необхідні для їх фізичного створення або побудови.

- Уможливити порівняння та бенчмаркінг різних парадигм оптимізації розкладу (Schedule Optimization Paradigms - SOP) у застосуванні до СТМ.
- Уможливити характеристику поведінки та продуктивності для SOP при застосуванні різних макетів СТМ та параметрів налаштувань.
- Дозволити програмне введення збурень в систему СТМ для спостереження за стійкістю різних SOP.
- Уможливити моделювання стохастичних процесів в СТМ та спостерігати їх вплив на продуктивність різних SOP.

Коротше кажучи, використовуючи NAASS, користувач (проектувальник/тестувальник СТМ) повинен мати можливість ввести в програму будь-яку бажану схему, незалежно від кількості компонентів. Користувач також повинен мати можливість вказати деталі та робочі параметри зазначених компонентів (час переміщення між компонентами, час роботи робочих станцій тощо), а також робочі параметри системи в цілому (частота надходження матеріалу в систему, загальна кількість одиниць, що підлягають обробці, парадигма планування, що використовується системою тощо). Після виконання імітаційного експерименту користувач повинен мати можливість провести статистичний аналіз результатів і зробити висновки про структуру СТМ в цілому або про ефективність використовуваної парадигми планування.

3.3.1 Моделювання систем транспортування матеріалів

У будь-якій СТМ, незалежно від методу, який використовується для транспортування предметів з одного місця в інше (монорейка, конвеєр, роботи, що підбирають і розміщують, колісні транспортні засоби, літальні апарати і т.д.), при розгляді СТМ з точки зору системи, всі вони поділяють ряд загальних компонентів, які перераховані в Таблиці 3.1. Саме ці компоненти доступні користувачеві NAASS для побудови макета. NAASS була написана для моделювання "мономатеріальної" системи, в якій на кожен транспортну одиницю (візок, піддон, лоток і т.д.), що входить в систему, припадає одна транспортна одиниця, що виходить з системи. Також передбачається, що кожна

одиниця не потребує ресурсів від іншої одиниці для створення готового продукту. Симуляція була написана за допомогою припущення, що для кожної схеми СТМ матеріал, який переміщується від одного компонента до іншого, повинен слідувати заздалегідь визначеними шляхами, які мають фіксований час переміщення та пропускну спроможність. Нарешті, симуляція виконується з дискретними часовими кроками і може моделювати лише дискретизовану виробничу систему.

Таблиця 3.1: Перелік компонентів СТМ.

Опис	Назва компонента NAASS
Як матеріал потрапляє до СТМ	Вихідний вузол
Де матеріал виходить з СТМ	Вузол стоку
Де виконуються роботи із зазначеним матеріалом	Робоча станція
Там, де матеріал може розходитися або зливатися з кількома шляхами	З'єднувач/розгалужувач комутаційних доріжок
Шлях від одного компонента до іншого	Дуга

3.3.2 Вирішення проблеми стохастичної природи реального виробництва

При побудові системи планування, програми або парадигми легко припустити, що всі виробничі процеси є детермінованими; однак будь-хто, хто мав досвід роботи з виробничими процесами, підтвердить, що виробничі процеси, як правило, є стохастичними в природі. Для більш точного моделювання реальних систем було б корисно дозволити моделювання зазначених стохастичних ефектів. Ця природа представлена в симуляції гауссовим розподілом ймовірностей, як показано на рис. 3.1. Де x - це середнє значення, y - 3σ , а $x-z$ контролює мінімальне значення, яке може бути прийняте. Якщо, наприклад, $x = 5$, $y = 3$, $z = 2$. 84% всього часу обробки припадатиме на проміжок між 3 і 8 часовими кроками.

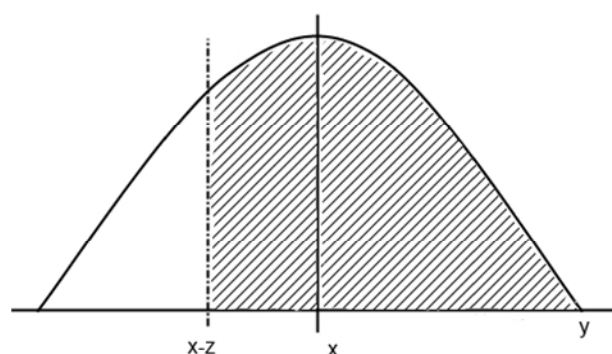


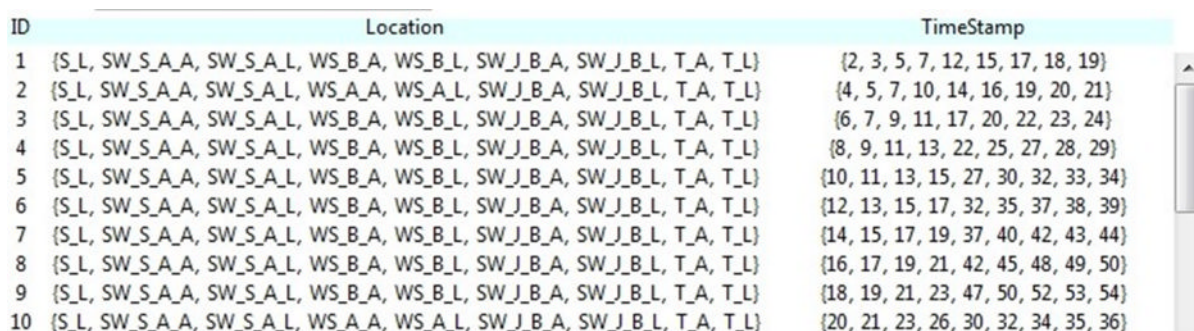
Рисунок 3.1: Гауссівський розподіл NAASS.

3.3.3 Кодування симуляції

Симуляція була написана мовою математичного програмування Mathematica (версія 9.0.0.0). Слід зазначити, що ця симуляція не написана для оптимального обчислення на одному комп'ютері. Скоріше, вона побудована таким чином, що якщо взяти код для кожної АСА і перекласти його на відповідну мову мікроконтролера, то його можна буде негайно виконати.

3.4 Структури зберігання даних NAASS

Симуляція містить одну основну матрицю даних - матрицю записів про візки (рис. 3.2). Ця матриця записує ідентифікатор кожного візка, послідовний список місць, через які він пройшов, а також час, коли візок прибув і покинув кожне з цих місць. У цій симуляції ця матриця замінює частину роботи, яку виконував би кожен окремих компонент, діє як бортова пам'ять кожного візка, а також дозволяє досягти безперервності всієї симуляції. Коли візок досягає вузла, ім'я вузла і часова мітка заноситься в матрицю. Той самий процес відбувається, коли візок залишає вузол. Розглянемо перший рядок на рис. 3.2 як приклад.



ID	Location	TimeStamp
1	{S_L, SW_S_A_A, SW_S_A_L, WS_B_A, WS_B_L, SW_J_B_A, SW_J_B_L, T_A, T_L}	{2, 3, 5, 7, 12, 15, 17, 18, 19}
2	{S_L, SW_S_A_A, SW_S_A_L, WS_A_A, WS_A_L, SW_J_B_A, SW_J_B_L, T_A, T_L}	{4, 5, 7, 10, 14, 16, 19, 20, 21}
3	{S_L, SW_S_A_A, SW_S_A_L, WS_B_A, WS_B_L, SW_J_B_A, SW_J_B_L, T_A, T_L}	{6, 7, 9, 11, 17, 20, 22, 23, 24}
4	{S_L, SW_S_A_A, SW_S_A_L, WS_B_A, WS_B_L, SW_J_B_A, SW_J_B_L, T_A, T_L}	{8, 9, 11, 13, 22, 25, 27, 28, 29}
5	{S_L, SW_S_A_A, SW_S_A_L, WS_B_A, WS_B_L, SW_J_B_A, SW_J_B_L, T_A, T_L}	{10, 11, 13, 15, 27, 30, 32, 33, 34}
6	{S_L, SW_S_A_A, SW_S_A_L, WS_B_A, WS_B_L, SW_J_B_A, SW_J_B_L, T_A, T_L}	{12, 13, 15, 17, 32, 35, 37, 38, 39}
7	{S_L, SW_S_A_A, SW_S_A_L, WS_B_A, WS_B_L, SW_J_B_A, SW_J_B_L, T_A, T_L}	{14, 15, 17, 19, 37, 40, 42, 43, 44}
8	{S_L, SW_S_A_A, SW_S_A_L, WS_B_A, WS_B_L, SW_J_B_A, SW_J_B_L, T_A, T_L}	{16, 17, 19, 21, 42, 45, 48, 49, 50}
9	{S_L, SW_S_A_A, SW_S_A_L, WS_B_A, WS_B_L, SW_J_B_A, SW_J_B_L, T_A, T_L}	{18, 19, 21, 23, 47, 50, 52, 53, 54}
10	{S_L, SW_S_A_A, SW_S_A_L, WS_A_A, WS_A_L, SW_J_B_A, SW_J_B_L, T_A, T_L}	{20, 21, 23, 26, 30, 32, 34, 35, 36}

Рисунок 3.2: Знімок екрана запису візка.

Перша колонка - це унікальний індивідуальний ідентифікаційний номер візка, який у цьому випадку дорівнює 1. Наступні дві колонки пов'язані між собою: колонка "Location" містить список усіх місць, які відвідав візок (прибув

та/або покинув), а колонка "TimeStamp" містить кроки системного часу, коли відбулася подія. Остання буква в кожному записі стовпчика "Location" означає або прибуття до місця (A), або відправлення з місця (L). Передостанній символ у кожному записі в колонці "Location" означає назву компонента в межах відповідного сімейства компонентів. Решта символів у кожному записі в колонці "Location" позначають назву та підтип компонента. Наприклад, "S" позначає вузол-джерело, а оскільки в усій системі існує лише один тип і, власне, один вузол-джерело, "S L" означає, що візок залишив вузол-джерело. Аналогічно, "SW S A L", "SW" означає, що компонент є стрілочною доріжкою, наступна "S" означає, що це розгалужувальна стрілочна доріжка, наступний символ "A" означає назву стрілочної доріжки і останній символ вказує на те, що візок залишив цю стрілочну лінію у відповідний момент часу. Нарешті, якщо ми застосуємо це до першого рядка на рис. 3.2, то візок 1 виїхав з джерела в момент часу $(ts) = 2$, прибув на роздільну доріжку A в момент часу $ts = 3$, залишив її в момент часу $ts = 5$, прибув на робочу станцію B в момент часу $ts = 7$, залишив її в момент часу $ts = 12$, прибув на з'єднувальну доріжку B в момент часу $ts = 15$, залишив її в момент часу $ts = 17$, прибув на вихідний вузол в момент часу $ts = 18$ і завершив обробку в момент часу $ts = 19$.

3.5 Графічний інтерфейс користувача AASS

У цьому розділі буде показано приклад кодування макета та експорту результатів. Візьмемо схему BARD (рис. 1.3), яку можна спростити до вигляду системи, показаного на рис. 3.3. Видно, що ця конкретна схема буде містити одне джерело, один стік, один розгалужувач стрілочних переводів, один з'єднувач стрілочних переводів і дві робочі станції, тобто загалом 6 компонентів.

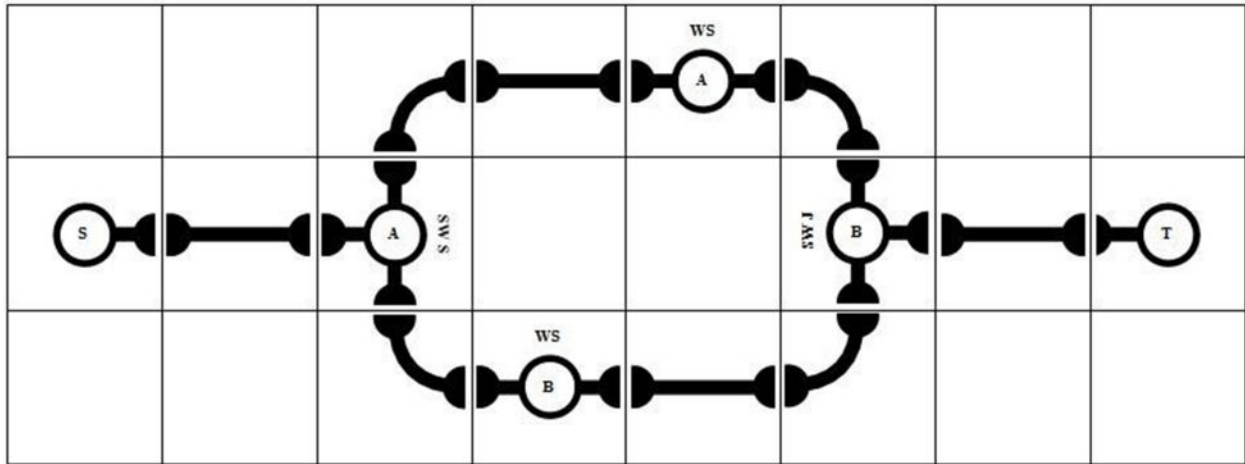


Рисунок 3.3: Схема системи BARD.

3.5.1 Конфігурація компонентів

На рис. 3.4(а) показано перший екран системи NAASS. Цей екран дозволяє користувачеві визначити загальну кількість компонентів, а також їхні індивідуальні номери.

3.5.2 Компонування компонентів

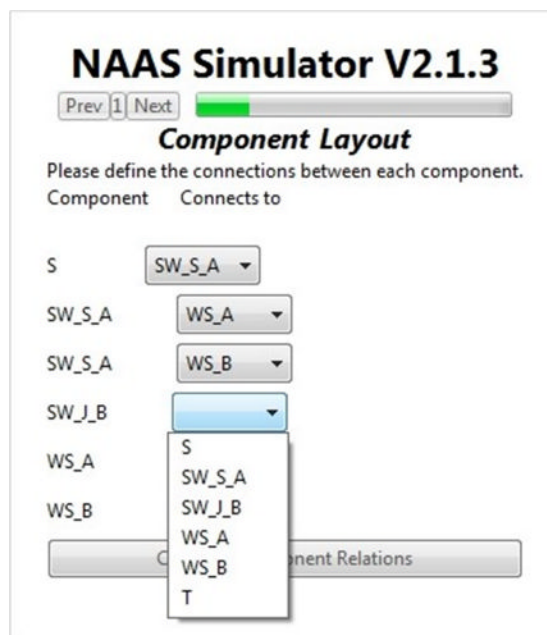
Наступне вікно, показане на рис. 3.4(б), дозволяє користувачеві встановити, який компонент з'єднаний з іншим компонентом за допомогою випадаючих меню. Слід зазначити, що програма автоматично присвоїть ім'я кожному компоненту, а також визначить кількість з'єднань, які повинен мати кожен тип компонентів, і налаштує опції відповідно до цього.

3.5.3 Підтвердження взаємодії та макет

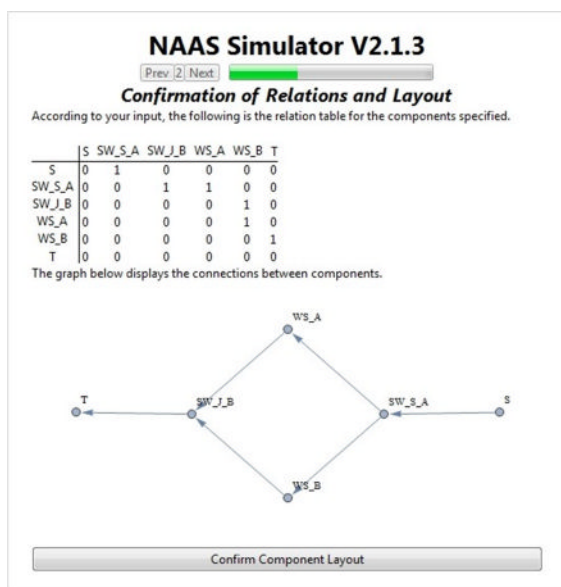
Екран, показаний на рис. 3.4 (в), дозволяє користувачеві візуалізувати введений макет, а також напрямок, в якому матеріал буде проходити через макет. На будь-якому етапі користувач може вільно і легко повернутися назад і внести будь-які зміни в будь-який з попередніх екранів за бажанням.



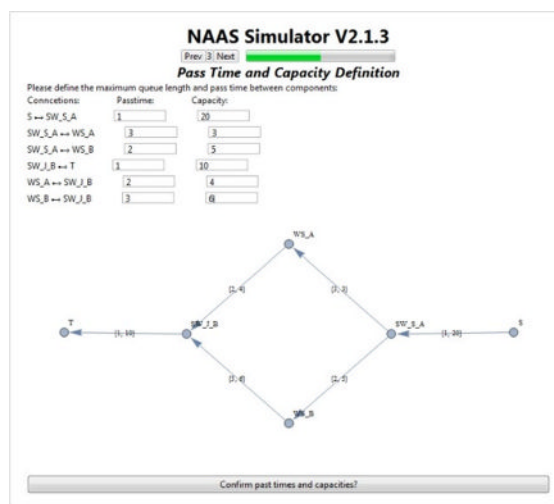
(а) Екран конфігурації компонентів NAASS.



(б) Екран компоновання компонентів.



(в) Екран підтвердження зв'язків та макетів NAASS.



(г) Екран визначення часу пропуску та пропускної спроможності NAASS.

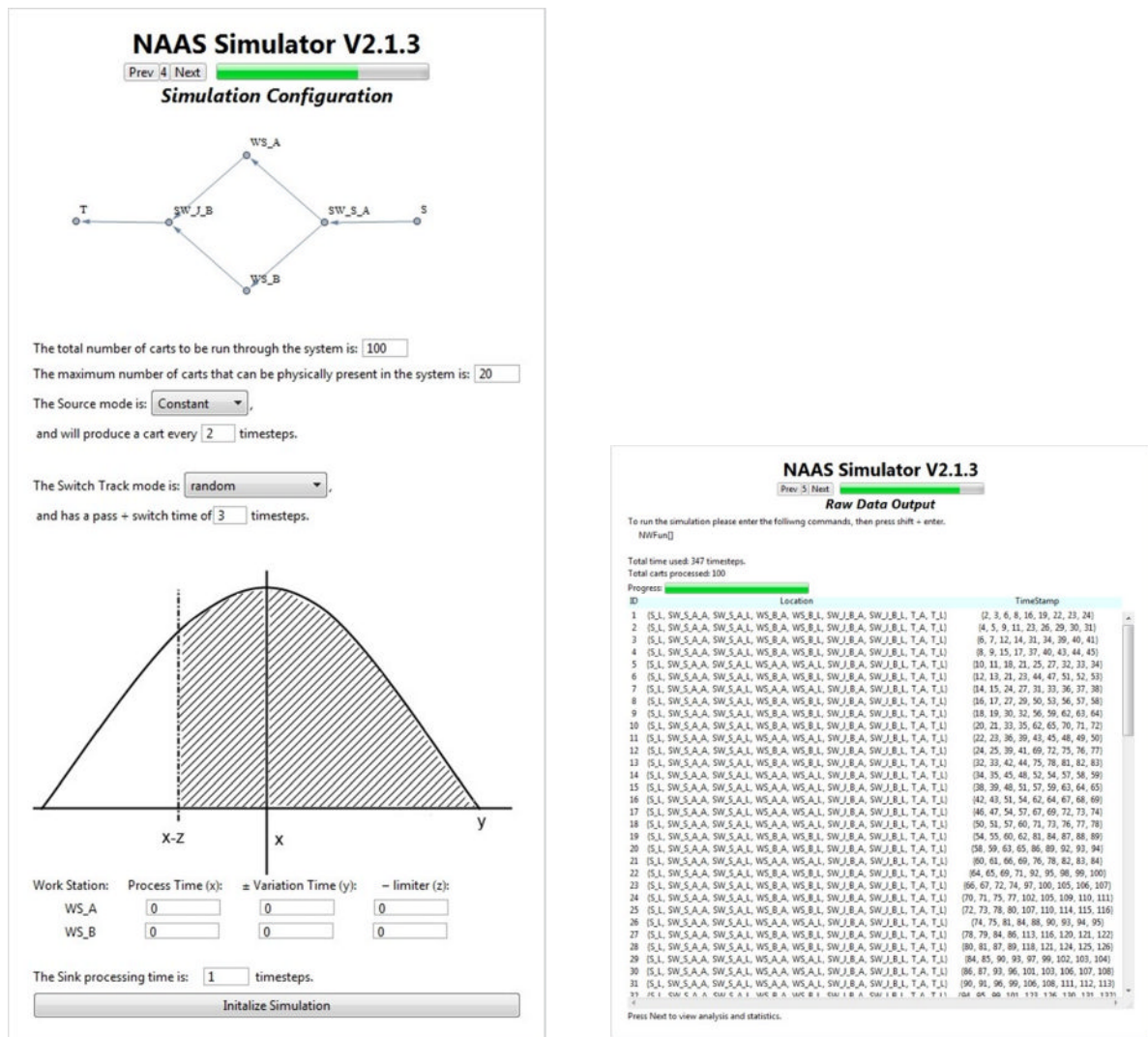
Рисунок 3.4: Знімки екрану NAASS, частина I.

3.5.4 Визначення часу проходження та пропускної здатності

Екран, показаний на рис. 3.4(г), дозволяє користувачеві визначити час, який знадобиться матеріалу для переміщення від одного компонента до іншого,

а також максимальну кількість одиниць, які можуть одночасно перебувати на одному шляху.

3.5.5 Конфігурація симуляції



(а) Екран конфігурації моделювання NAASS. (б) Екран виведення вихідних даних NAASS.

Рисунок 3.5: Знімки екрану NAASS, частина 2.

Екран, показаний на рис 3.5(а), дозволяє користувачеві встановити робочі параметри симуляції, включаючи загальну кількість одиниць, що проходять через систему, максимальну кількість одиниць, які можуть перебувати в системі одночасно, режим генерації вузла-джерела, парадигму планування, що використовується для комутаційних шляхів, час обробки на робочих станціях і в вузлі-споживачі. У підрозділі 3.6 буде надано більш детальне пояснення щодо роботи кожного компонента.

3.5.6 Виведення сирих даних

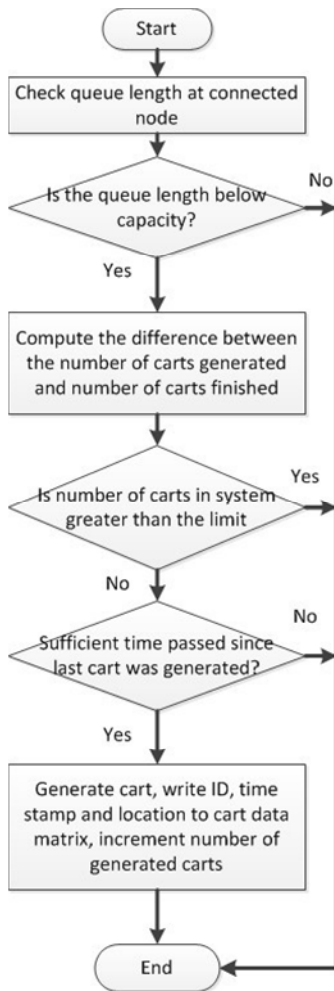
Екран виведення вихідних даних (рис. 3.5(б)) дозволяє користувачеві переглянути хід симуляції та визначити, коли симуляція завершиться. Після завершення користувач може експортувати результати моделювання у будь-який бажаний формат і місце.

3.6 Компоненти NAASS

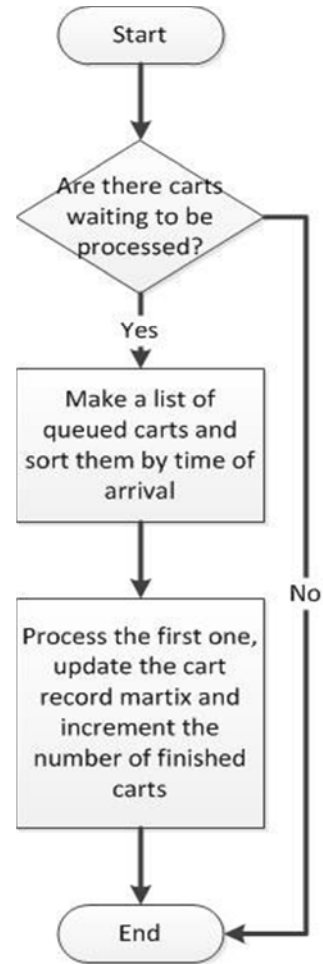
У наступних розділах буде надано детальне пояснення функцій компонентів, перелічених у Таблиці 3.1, а також кілька додаткових сегментів коду.

3.6.1 Вихідний вузол

На рис. 3.6(a) показано блок-схему вузла-джерела, де матеріал вводиться в систему заздалегідь визначеним способом. Слід зазначити, що час між генераціями візків розраховується відповідно до встановленого користувачем режиму, яких є два. Перший режим є детермінованим, де вузол-джерело буде генерувати візок через кожні x часових кроків. Другий режим вводить стохастичний елемент у процедуру генерації, тому користувач може вказати, що візок повинен генеруватися кожні $x + y$ кроків. У цьому випадку найкоротший проміжок часу між генеруванням візків становитиме x тактів, а найдовший (якщо дозволять перевірки) - $x + y$ тактів. Така варіація досягається за допомогою рівномірного розподілу ймовірностей, де x і y - цілі значення часових кроків.



(а) Діаграма потоків вихідного вузла..



(б) Діаграма потоку вузла стоку.

Рисунок 3.6: Блок-схеми вихідного вузла та вузла стоку.

3.6.2 Вузол стоку

Вузол стоку є кінцевим вузлом у будь-якій схемі СТМ в NAASS і має лише один параметр, який може бути змінений користувачем, а саме час обробки. Цей вузол дозволяє остаточно підтвердити, що візок завершив обробку. Вузол стоку приймає візки індивідуально відповідно до порядку прибуття, оновлює матрицю записів візків, де це доречно, а також збільшує кількість оброблених візків. Логічна послідовність роботи вузла стоку показана на рис. 3.6(б).

3.6.3 Вузол розгалужувача доріжок перемикача

Вузол розгалужувача стрілочної доріжки можна розглядати як такий, що складається з двох частин: робочої функції та функції фінішної обробки, які

працюють послідовно. Передбачається, що час, необхідний для проходження візка через вузол стрілочного перевалу, є постійним і визначається користувачем.

3.6.3.1 Функція перемикання розгалужувача доріжок

На рис. 3.7 показано послідовність виконання функції роботи для вузла розгалужувача стрілочної доріжки. Основною функцією цієї секції є визначення того, чи слід дозволити візку в'їхати у вузол (вільний шлях до наступного вузла і відсутність візків, що обробляються), а також власне обробка візка. Слід ще раз зазначити, що час проходження перемикаючої доріжки визначається користувачем, як показано на рис. 3.5 (а).

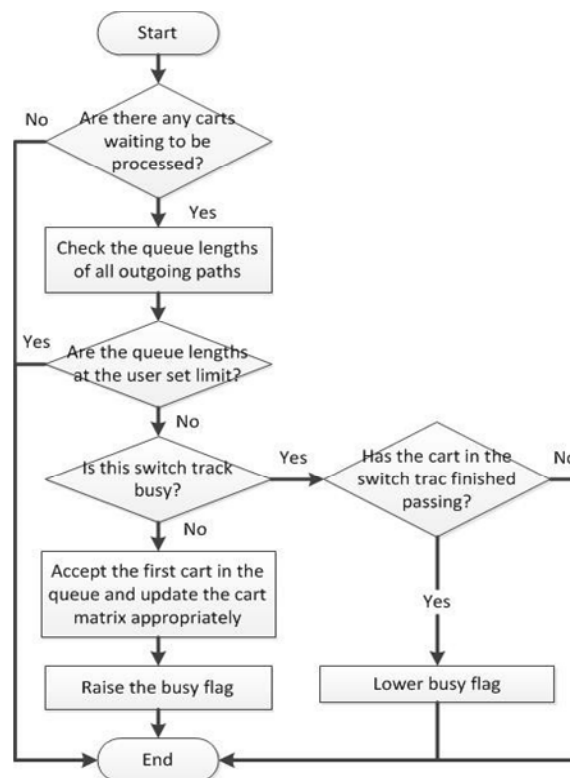


Рисунок 3.7: Функціональна схема роботи стрілочного розгалужувача.

3.6.3.2 Функція фінішної обробки розгалужувача доріжок

На рис. 3.8 показано процес і порядок виконання функції завершення для вузла розгалужувача стрілочної доріжки. Мета функції завершення - визначити, куди спрямувати візок, який був оброблений робочою функцією. Як

видно з блок-схеми, є чотири можливі схеми планування, які можуть бути застосовані за бажанням користувача. Ці чотири схеми такі: Випадкова, Альтернативна, на основі черги та черга + час проходження відповідно.

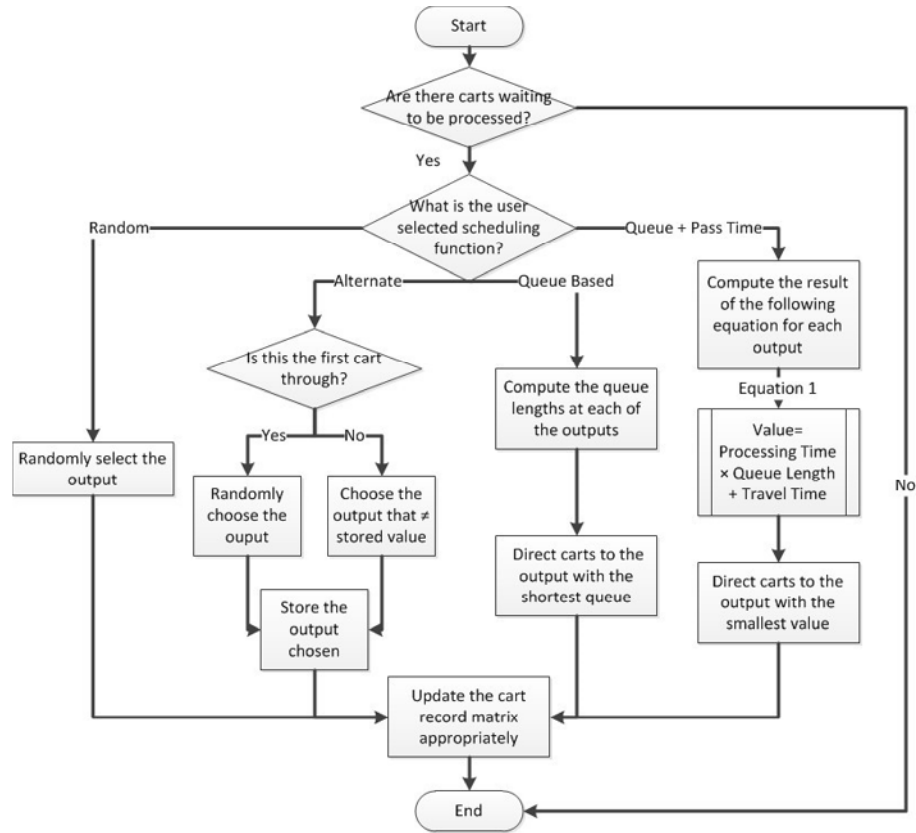


Рисунок 3.8: Блок-схема функції фінішної обробки стрілочного розгалужувача.

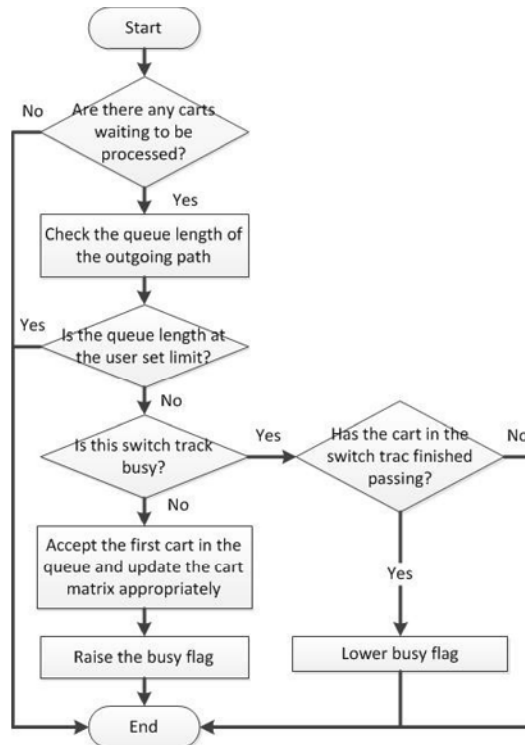


Рисунок 3.9: Блок-схема вузла з'єднання стрілочного розгалужувача.

3.6.4 Вузол з'єднання стрілочних перемикачів

На рис. 3.9 показано процес і порядок виконання вузла з'єднання стрілочних перемикачів. Мета цієї функції - об'єднати візки, що рухаються двома окремими доріжками, в одну вихідну доріжку. Вузол з'єднання обробляє вагонетки послідовно, а час проходження вузла з'єднання встановлюється користувачем.

3.6.5 Вузол робочої станції

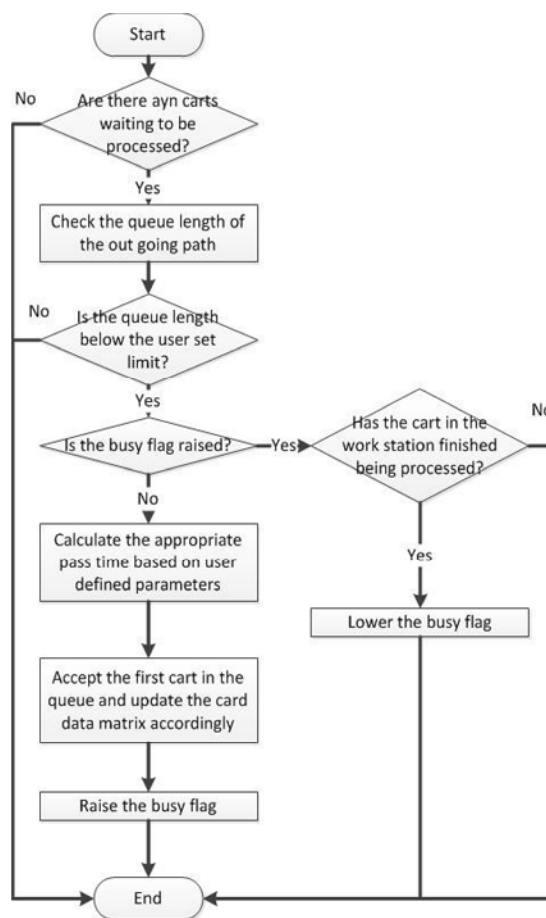


Рисунок 3.10: Блок-схема вузлів робочої станції.

На рис. 3.10 показано процес і порядок виконання вузла робочої станції. Мета цього вузла - діяти як чорний ящик, що імітує роботу, яка виконується на візку в системі, наприклад, у виробничому процесі. Час обробки для кожної робочої станції може бути встановлений користувачем, як показано на рис. 3.5(a). Як показано на рисунку і згадувалося в попередніх розділах, для кожної

робочої станції можна встановити три параметри, які називаються x , y і z відповідно. Якщо всі параметри, крім x , дорівнюють нулю, то час обробки на робочій станції буде постійним і дорівнюватиме значенню x . Якщо тільки z дорівнює нулю, то час обробки на робочій станції буде розраховуватися відповідно до гаусівського розподілу ймовірностей, де середнє значення дорівнює x , а y - значенням трьох стандартних відхилень. Якщо жоден з параметрів не дорівнює нулю, час обробки буде розраховуватися відповідно до гаусівського розподілу ймовірностей, де x є середнім значенням, а y - значенням трьох стандартних відхилень, але не допускається жодне значення менше, ніж $x - z$. Це дозволяє контролювати дисперсію часу обробки, щоб реалістично імітувати різні реальні ситуації.

3.7 Результати та статистика AASS

Таблиця 3.2: Зразок зведеної таблиці записів кошика.

Оброблено візків:	200	Н/Д
Виконані часові кроки:	516	ts
Режим джерела:	Постійний	Н/Д
Джерело gen.:	4	ts
Джерело +:	2	ts
Перемикання режиму:	Альтернативний	Н/Д
Перемикання перепустки:	2	ts
Теоретичний максимум візків:	232	Н/Д
Режим робочої станції:	Постійний	Н/Д
Перемикання стоку:	1	ts
S	SW_S_A	{1,200}
SW_S_A	WS_A	{3,200}
SW_S_A	WS_B	{2,200}
SW_J_B	T	{1,200}
WS_A	SW_J_B	{2,200}
WS_B	SW_J_B	{3,200}
4	0	0
5	0	0

Після завершення кожного імітаційного експерименту можна викликати функцію, яка створить підсумок вхідних параметрів для цього експерименту і відформатує його в таблицю, подібну до тієї, що наведена в Таблиці 3.2. Перші 10 рядків є відносно зрозумілими. Слід зазначити, що в рядку 9 режим роботи

робочих станцій буде вважатися постійним лише в тому випадку, якщо всі робочі станції мають постійний час обробки. Починаючи з перших десяти рядків і далі, доки стовпчик 1 не містить числа, перераховано таблицю зв'язків. Це список усіх зв'язків між усіма компонентами в схемі та їх параметрами. Оскільки модель NAASS припускає, що візки можуть рухатися лише в одному напрямку, то рядок 11 можна перекласти: Візки будуть рухатися від вихідного вузла ("S") до стрілочного перевалу A ("SW_S_A"), на цьому шляху потрібно зробити один крок, і одночасно в черзі можуть перебувати 200 візків. Решта рядків після таблиці зв'язків містять параметри робочих станцій (у форматі x, y, z). Передостанній рядок вказує на те, що перша робоча станція (WS_A) має середній і постійний час проходження чотирьох часових кроків. Нарешті, перед експортом вся матриця записів про візки об'єднується в підсумкову таблицю, яку потім можна експортувати в потрібному користувачеві форматі.

3.8 Валідація та висновок NAASS

Ми валідуємо NAASS, порівнюючи результати моделювання з результатами фізичної системи за ідентичних параметрів. Було використано фізичну установку BARD (рис. 1.3), і на момент валідації BARD можна було керувати за допомогою компактного RIO від National Instruments (NI-cRIO 9025), а також раніше згаданого ПЛК від Allen & Bradley (Compact Logix L35E). Для того, щоб спробувати реалізувати на промисловому обладнанні, було використано компактний RIO. Для валідації даних через BARD було пропущено 100 візків, щоб виміряти час проходження між кожним з компонентів BARD, як показано на рис. 3.3. Вимірний час був усереднений і дещо змінений для того, щоб його можна було ввести в систему NAASS. Параметри можна побачити в Додатку А в Таблиці А.1. Потім алгоритми, що керують кожним з компонентів, як пояснювалося раніше в цьому розділі, були запрограмовані в BARD за допомогою LabVIEW 2012. Спочатку бенчмаркінг будь-якої СТМ та ефективність будь-якої SOP буде оцінюватися за кількістю візків, які можуть бути оброблені за одиницю часу.

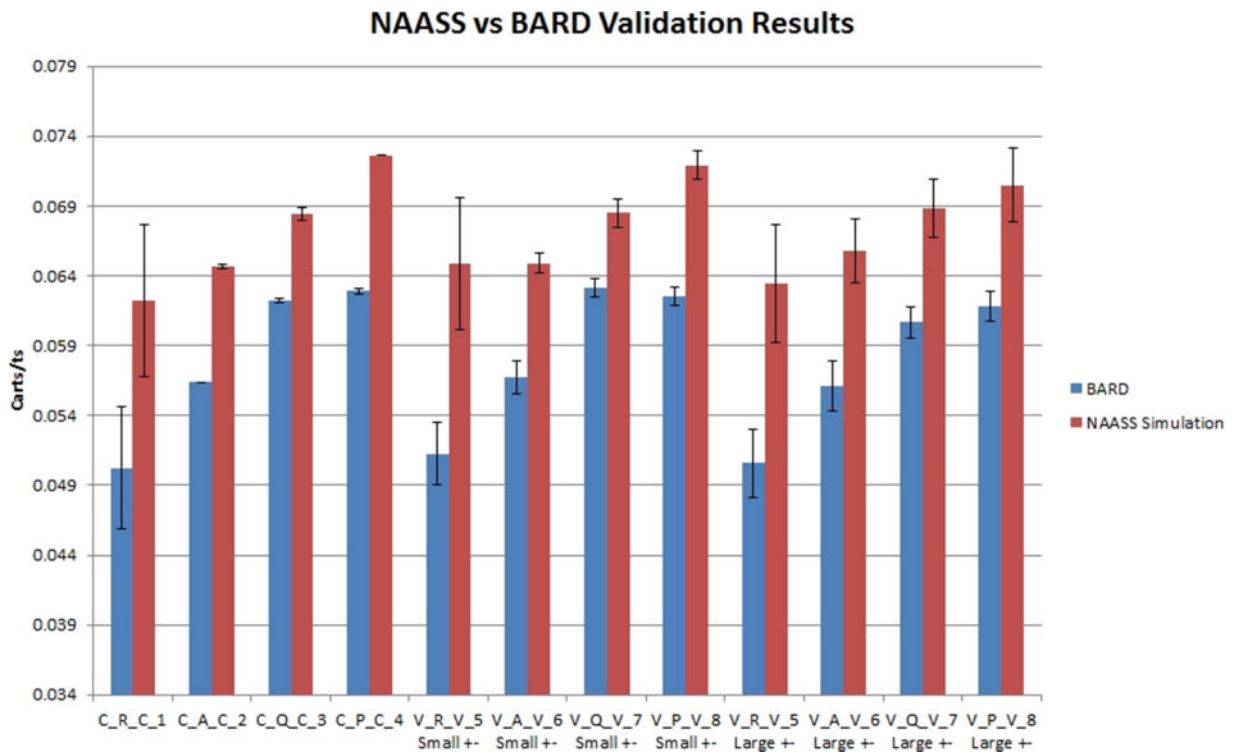


Рисунок 3.11: Результати валідації NAASS та BARD.

Дані з BARD було зібрано з трьома різними наборами вхідних параметрів, названих детермінованим (D), малою варіацією (SV) та великою варіацією (LV), відповідно. Для детермінованого набору вхідних параметрів, як впливає з назви, всі умови були детермінованими, вузол-джерело генерував візки детермінованим чином, а час обробки для кожної робочої станції був постійним, про що свідчить назва випробування. C_R_C_1 вказує на те, що режим джерела є постійним (перша буква "C"), друга буква позначає використовуваний SOP ("R" - випадковий, "A" - альтернативний, "Q" - на основі черги і "P" - черга + час проходження), третя буква вказує на режим робочих станцій ("C" - постійний, "V" - змінний). Для вхідних параметрів SV, як впливає з назви, допускається невелика кількість варіацій у часі обробки на робочих станціях. Аналогічно, для вхідних параметрів LV спостерігається більший рівень стохастичності. Фактичні вхідні параметри, що використовувалися, можна побачити в Додатку А в Таблиці А.2. Для кожного з трьох наборів вхідних параметрів було застосовано 4 SOP. Через обмеженість ресурсів було проведено 100 випробувань для кожного з 12 сценаріїв на BARD, а результати були усереднені. 100 випробувань було проведено для кожного з дванадцяти сценаріїв в NAASS, і результати були усереднені. Порівняння

результатів можна побачити на рис. 3.11. Смуги похибок на рис. 3.11 представляють одне стандартне відхилення. Порівняння результатів показало, що загальна кореляція становить 0,807114 у всіх 12 випадках. Слід зазначити, що через програмні обмеження, коли варіація була запрограмована в BARD, використовувався рівномірний розподіл ймовірностей, а не гаусівський розподіл, який використовується в NAASS. Повний розподіл кореляцій можна побачити в Таблиці 3.3. Виходячи з наведених вище результатів, NAASS вважається здатною моделювати фізичну систему (BARD). На основі цієї валідації ми вважаємо, що ми досягли валідної методології моделювання.

Таблиця 3.3: Кореляційна таблиця для результатів BARD та NAASS.

Діапазон	Кореляція (R^2)
Загальний	0.807
Детермінований	0.855
Стохастичний SV	0.668
Стохастичний LV	0.965

4 КОНСТРУКТОРСЬКА ЧАСТИНА

4.1 Кількісна та якісна оцінка

У цьому розділі детально описано симуляції, які були виконані на NAASS з використанням різних SOP і макетів, а також висновки, зроблені на основі аналізу результатів. Починаючи зі статистичного аналізу впливу номера візка та кількості повторень моделювання на якість та достовірність результатів.

4.1.1 Пробна кількість

На етапі перевірки концепції, а згодом і розробки повного коду NAASS, до кожного досліджуваного макету було застосовано набір з восьми прикладів. Ці 8 тестів, наведені в Таблиці 4.1, були розроблені для повного тестування впливу макета на продуктивність кожної SOP в детермінованих і стохастичних умовах. Спочатку кожен з восьми випадків був виконаний двадцять разів, а результати були усереднені. Для простих випадків, таких як макет, показаний на рис. 3.3, виконання одного випадку може зайняти від однієї до трьох хвилин. Для найпростіших макетів ($8 \times 20 = 160$) може знадобитися до восьми годин. Зі збільшенням складності схем і зміною кількості візків, що проходять через кожен з них, час виконання ставав значним фактором. Таким чином, дослідження збіжності визначило мінімальну кількість випробувань, необхідну для кожного випадку, щоб досягти статистичної стабільності, мінімізуючи при цьому загальний час виконання. Використовуючи результати 20 пробних випадків як еталон, було виявлено, що коли всі інші параметри, окрім кількості досліджень, залишалися незмінними, то загальна кореляція на рівні 97% була досягнута лише для 10 досліджень. Таким чином, загальний час виконання скоротився при збереженні якості та достовірності зібраних даних.

Повний список Кореляційні порівняння можна побачити в Таблиці 4.2, а повні результати для 10 і 20 пробних випадків можна знайти в Додатку А в Таблиці А.3.

Таблиця 4.1: Імітаційні групи NAASS.

Категорія	Група	Режим джерела	Режим SW	Режим WS
Детермінований	1	Постійний	Випадковий	Постійний
	2	Постійний	Альтернативний	Постійний
	3	Постійний	Довжина черги	Постійний
	4	Постійний	Довжина черги + час проходження	Постійний
Стохастичний	5	Змінний	Випадковий	Змінний
	6	Змінний	Альтернативний	Змінний
	7	Змінний	Довжина черги	Змінний
	8	Змінний	Довжина черги + час проходження	Змінний

Таблиця 4.2: Кореляційна таблиця для тесту AASS на кількість спроб.

Діапазон	Кореляція(R^2)
Загалом	0.973
Детермінований	0.996
Стохастичний	0.943

4.1.2 Кількість оброблених візків

Після того, як було досліджено кількість випробувань у кожній справі, наступним кроком було вивчення впливу кількості візків, що проходять через систему за одне випробування, на продуктивність системи та стабільність і якість результатів. Кількість візків на одне дослідження також має значний вплив на споживання ресурсів. На прикладі "Unit Layout", показаному на рис. 3.3, симуляція 100 візків вимагала менше 1 хвилини, 300 візків - від однієї до двох хвилин, а для 500 візків в середньому було потрібно близько п'яти хвилин. Для встановлення еталонного показника було використано макет установки з 500 візками, проведено вісім випробувань з десятьма випробуваннями в кожному, а результати для кожного з випробувань були усереднені. Всі умови були постійними, а кількість використовуваних візків

варіювалася. Порівнюючи результати з еталоном, коли імітаційні випробування проводилися з 300 візками, загальний кореляція становила 97,9%, тоді як у випробуваннях на 100 візках значення кореляції становило 99,5%. Повний перелік кореляційних порівнянь можна побачити в Таблиці 4.3, а повні результати дослідження можна побачити в Додатку А в Таблиці А.4. Не було достатніх доказів того, що моделювання маршрутів з меншою кількістю візків, ніж 100, дасть точні та стабільні результати. Таким чином, 100 візків було визначено як мінімально необхідну кількість візків для забезпечення впевненості в точності та стабільності результатів. Для того, щоб ще більше підвищити впевненість у цих результатах, було вирішено, що кожне імітаційне випробування буде проводитися з 200 візками. Таке збільшення підвищило б довіру до результатів, не збільшивши при цьому час виконання імітаційного експерименту.

Таблиця 4.3: Кореляційна таблиця для тесту NAASS на визначення номера візка.

300 візків проти 500 візків	
Діапазон	Кореляція(R^2)
Загальний	0.979
Детермінований	0.966
Стохастичний	0.971
100 візків проти 500 візків	
Діапазон	Кореляція(R^2)
Загальний	0.995
Детермінований	0.991
Стохастичний	0.991

4.1.3 Системні варіації та важливість варіативності

Спочатку на етапі перевірки концепції для кожного з восьми вищезгаданих випадків було застосовано три додаткові типи систем. Ці типи систем мали на меті дослідити вплив рівня стохастичності системи на продуктивність системи та SOP. На прикладі прикладу підрозділу,

зображеного на рис. 3.3, детальну інформацію про три типи систем можна побачити в Таблиці 4.4. Система типу 1 спроектована таким чином, що час переміщення між робочими станціями набагато більший, ніж час обробки та варіації на цих робочих станціях. Це зменшує вплив стохастичності робочих станцій на систему. Система типу 2 спроектована таким чином, що час у дорозі між компонентами є порівнянним з часом обробки та варіаціями на робочих станціях, по суті, присвоюючи майже однакову вагу для другий. Система типу 3 спроектована таким чином, що час обробки та варіації робочих станцій значно перевищує час переміщення між компонентами. Було вирішено, що основна увага в дослідженні буде зосереджена на виконанні SOP в одному типі системи, і було обрано систему типу 2.

Таблиця 4.4: Параметри типу системи.

Назва компонента / шлях до нього	Власність	Тип системи			Одиниця
		1	2	3	
Джерело (S)	Частота та варіація генерації	2 + 2	2 + 2	2 + 2	ts
Перемикач доріжок (SW_)	Час проходження	3	2	2	ts
Робоча станція A (WS_)	Час проходження та варіація	2 ± 1	4 ± 3	9 ± 5	ts
Робоче місце B	Час проходження та варіація	3 ± 2	5 ± 4	12 ± 5	ts
Стік (T)	Час процесу	1	1	1	ts
S > SW_S_A	Час проходження	3	1	1	ts
SW_S_A > WS_A	Час проходження	7	3	3	ts
SW_S_A > WS_B	Час проходження	4	2	2	ts
SW_J_B > T	Час проходження	3	1	1	ts
WS_A > SW_J_B	Час проходження	4	2	2	ts
WS_B > SW_J_B	Час проходження	7	3	3	ts

4.2 Тести макета та SOP

У цьому розділі обговорюються чотири тестові схеми, які були запрограмовані в NAASS (одиночна, послідовна, паралельна та послідовно-паралельна), а також чотири вищезгадані SOP (випадкова, альтернативна, на

основі черги та черга + час проходження), які застосовувалися до кожної з них.

4.2.1 Тестові макети

Загалом, для тестування в NAASS було запрограмовано чотири схеми. Це блочна схема (U), послідовна схема (S), паралельна схема (P) і послідовно-паралельна схема (SP). Блочна схема вважається найпростішою схемою, яка містить стрілочну доріжку, здатну приймати рішення щодо планування. Блочна схема зображує робочу комірку, що містить дві робочі станції, які виконують одну і ту ж задачу. Графічне зображення блочної схеми показано на рис. 4.1.

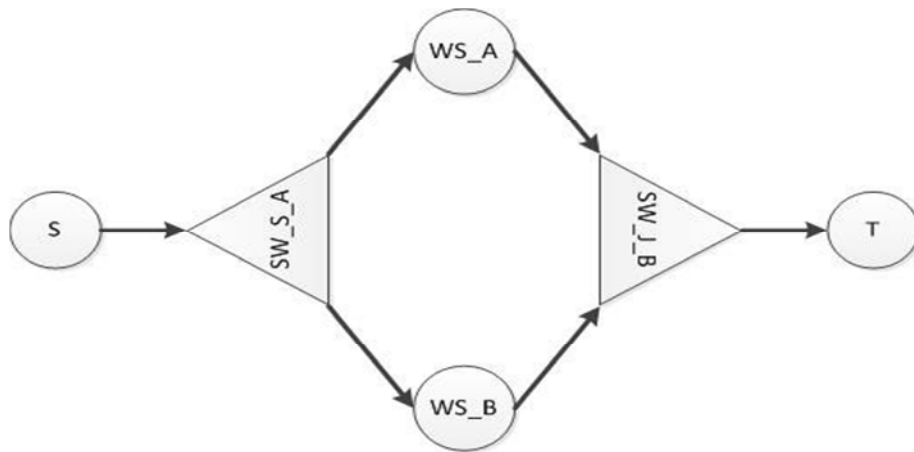


Рисунок 4.1: Розташування підрозділів NAASS.

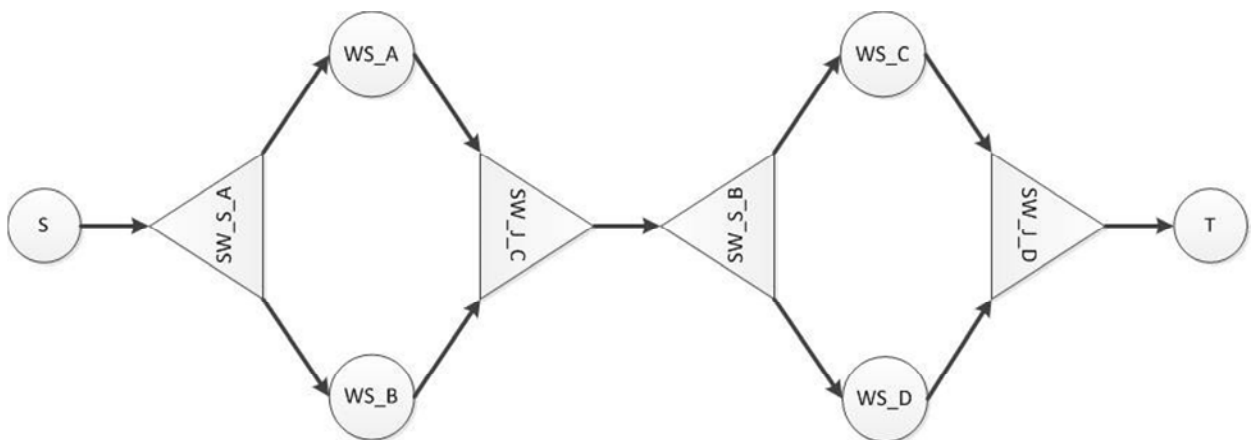


Рисунок 4.2: Серійний макет NAASS.

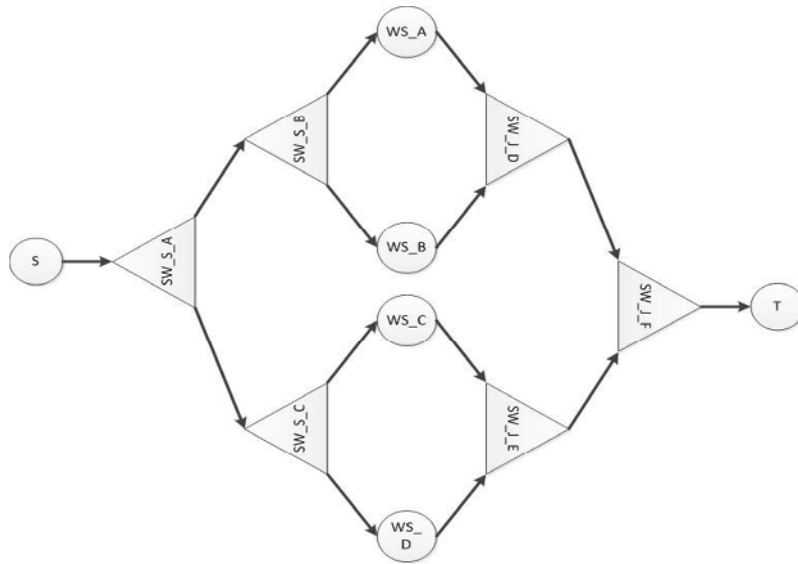


Рисунок 4.3: Паралельне розташування NAASS.

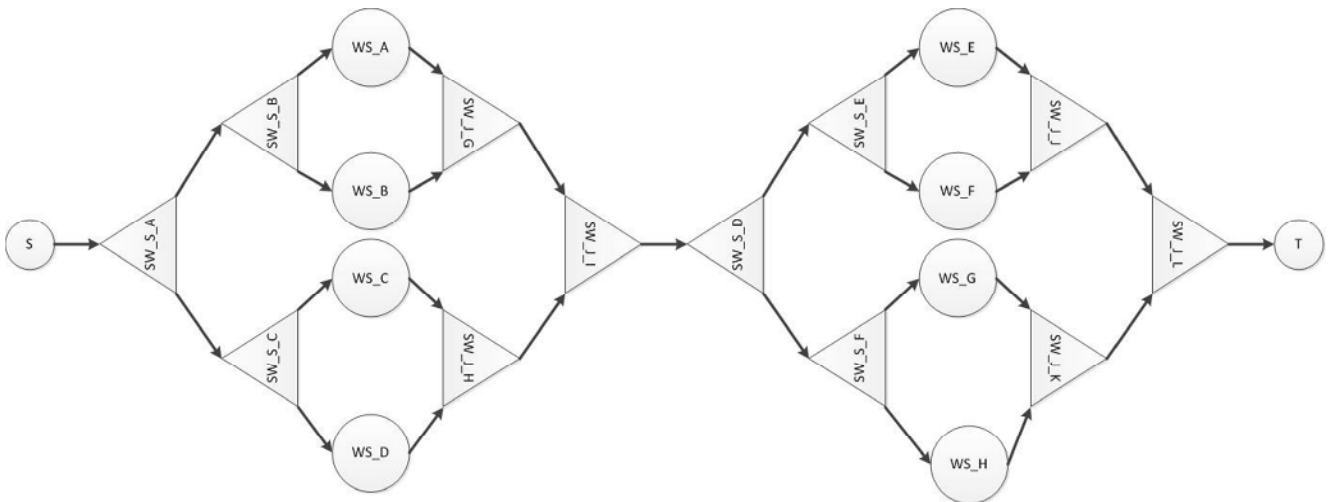


Рисунок 4.4: Послідовно-паралельна схема NAASS.

Паралельна схема імітує чотири робочі станції, які виконують однакові завдання, але розділені на дві групи, імітуючи складніший і масштабніший виробничий процес, ніж одинична схема. Графічне зображення паралельної схеми можна побачити на рис. 4.3.

Останній макет є комбінацією паралельного і послідовного макетів і називається послідовно-паралельним макетом. Ця схема складається з двох паралельних схем у послідовній конфігурації і є найскладнішою схемою з чотирьох. Кожен візок, який потрапляє в цю схему, може бути направлений на будь-яку з чотирьох робочих станцій для кожного з двох процесів, які повинні

бути виконані на ньому. Графічне зображення цієї схеми можна побачити на рис. 4.4.

Теоретично, будь-який виробничий процес може бути побудований з комбінації вищезгаданих макетів. Крім того, якби існував метод, за допомогою якого велику і складну схему системи можна було б представити схемою меншої складності, це дозволило б NAASS імітувати поведінку будь-якої складної виробничої системи, не вимагаючи великої кількості ресурсів. Якби існував метод, за допомогою якого поведінку і продуктивність одиначної схеми можна було б імітувати за допомогою робочої станції, оснащеної ідентичною кількістю візків з однаковою частотою, тоді логічно можна було б імітувати паралельну схему за допомогою однієї одиначної схеми. Аналогічно, паралельно-послідовну схему можна звести до послідовної. Таким чином, будь-яка реальна виробнича система може бути зібрана з чотирьох вищеописаних схем, а її поведінка може бути легко і точно змодельована за допомогою NAASS.

4.2.2 Результати та обговорення

Як зазначалося в попередніх розділах, для кожної з чотирьох схем було створено 8 імітаційних груп (табл. 4.1), по 10 випробувань у кожній групі та 200 візків на одне випробування. Для кожної схеми було проведено порівняння ефективності кожного з SOP відповідно до попередньо наведених критеріїв визначений еталонний показник кількості візків на кожному часовому кроці.

Точні вхідні параметри, використані для тестування кожної з схем, можна знайти в Додатку А, Таблиця А.5, Таблиця А.6, Таблиця А.7 і Таблиця А.8. Результати для макетів блоків можна побачити на рис. 4.5. На рис. 4.5 (D) позначає детерміновані випадки, а (S) - стохастичні випадки, як зазначено в таблиці 4.1. Смуги похибок на рисунку представляють стандартне відхилення

усереднених результатів. З результатів розміщення агрегатів чітко видно, що як у детермінованих, так і в стохастичних випадках два SOP ЖАСА працювали краще, ніж планування без вбудованого інтелекту. Передбачається, що через брак доказів динамічно оптимізуючі програми управління для систем автоматизації та програмування ПЛК не є поширеними в промисловості.

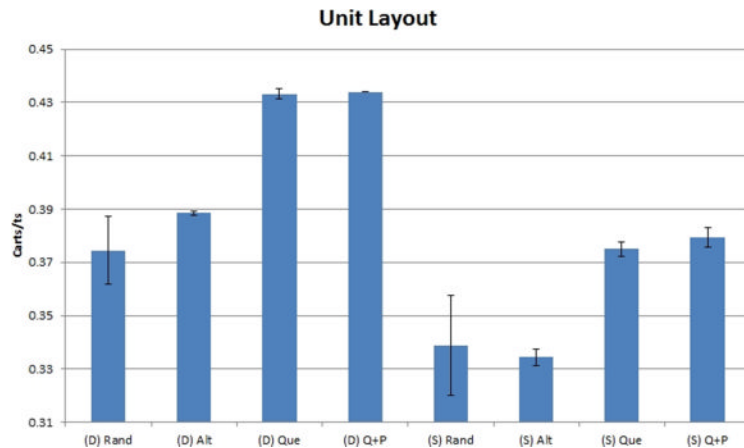


Рисунок 4.5: Результати розміщення підрозділів NAASS.

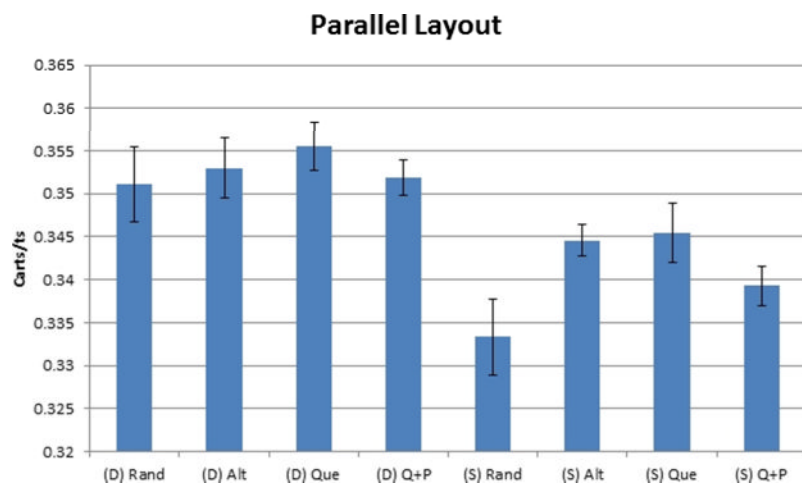


Рисунок 4.6: Результати паралельного розміщення NAASS.

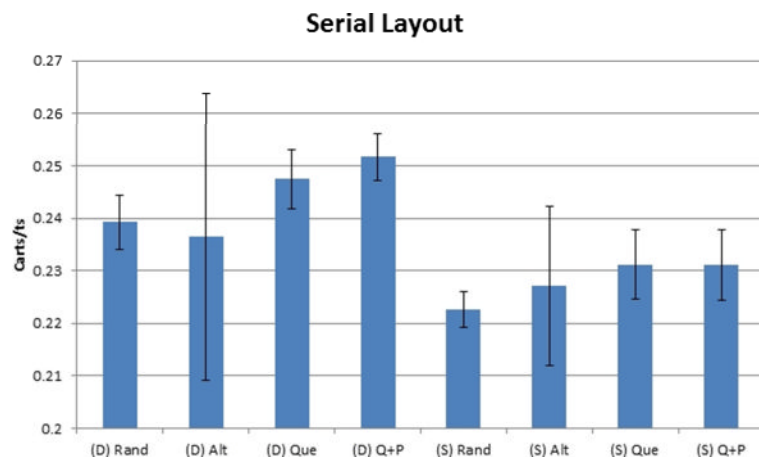


Рисунок 4.7: Результати послідовного макета NAASS.

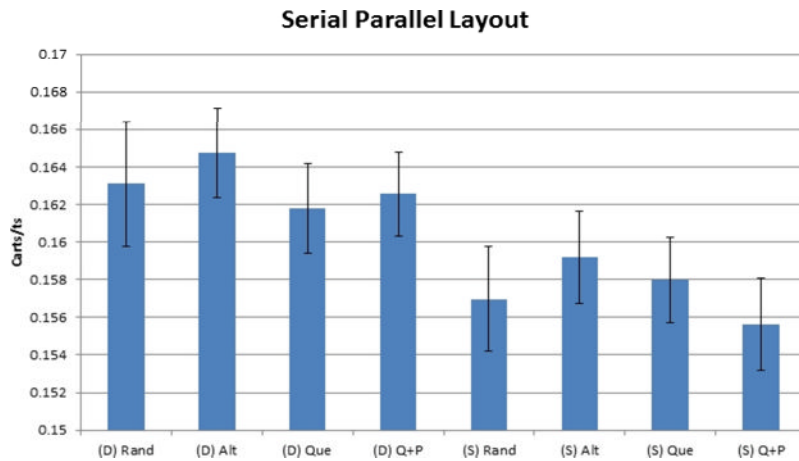


Рисунок 4.8: Результати послідовно-паралельного розміщення NAASS.

На рис. 4.6 показано результати для паралельної схеми. Незважаючи на збільшення складності системи, два SOP з ЖАСА продовжують перевершувати SOP з випадковим та почерговим плануванням, хоча масштаб збільшення продуктивності відрізняється від того, що спостерігається при одиничному плануванні.

На рис. 4.7 показано результати для послідовної схеми, збільшення розміру та складності системи не вплинуло на продуктивність SOP ЖАСА порівняно з продуктивністю SOP з випадковим та альтернативним плануванням.

На рис. 4.8 показано результати послідовно-паралельного макета. Ця схема є найбільшою і найскладнішою з чотирьох схем. Як видно з графіка, і в детермінованому, і в стохастичному сценаріях альтернативний SOP перевершує обидва SOP ЖАСА. Обидва SOP ЖАСА базуються на алгоритмі жадібної оптимізації, який заглядає вперед лише на один крок або компонент. Добре задокументовано, що жадібні алгоритми можуть мати тенденцію вести систему до локального оптимуму, а не до глобального. Подальші результати покажуть, однак, що незалежно від макета, SOP ЖАСА постійно перевершують інші SOP.

4.3 Масштабованість системи

У цьому розділі досліджується можливість взяти більш складну схему, таку як паралельна схема (рис. 4.3) і змоделювати її за допомогою набагато простішої, наприклад, блочної схеми (рис. 4.1). Результати дослідження представлені та обговорені та зроблено відповідні висновки.

4.3.1 Вступ

Мета цього дослідження - з'ясувати, чи можна відтворити надзвичайно складні системи за допомогою простіших систем, тим самим зменшивши обчислювальні ресурси, необхідні для моделювання та запуску симуляцій для дослідження продуктивності системи за різних умов. З огляду на це, мета полягає в тому, щоб з'ясувати, чи можливо відтворити продуктивність паралельної схеми за допомогою блокової схеми, а також відтворити продуктивність послідовної паралельної схеми за допомогою послідовної схеми. Для того, щоб блочна схема імітувала продуктивність і поведінку паралельної схеми, кожна робоча станція в цій схемі сама по собі буде представляти блочну схему, і аналогічний процес застосовується для послідовної схеми, щоб імітувати послідовну паралельну схему.

Для того, щоб представити схему з однією робочою станцією, спочатку було виконано симуляцію схеми з однією робочою станцією та зібрано результати. Час, необхідний для проходження кожного візка через систему, був записаний, усереднений і розрахований стандартне відхилення. Для вузла

$$x = \frac{MeanCycleTime}{2} \quad (4.1)$$

$$y = 3\sigma \quad (4.2)$$

$$y = \frac{std \times 3}{2} \quad (4.3)$$

$$z = \frac{MinCycleTime}{-} \quad (4.4)$$

робочої станції є три можливі параметри, які можна варіювати, а саме: x , y та z , як показано на рис. 3.1. Оскільки схема вузла містить дві робочі станції, які працюють паралельно і можуть обробляти візки одночасно, було вирішено, що середній час циклу буде розраховуватися шляхом ділення середнього часу циклу на два, щоб врахувати одночасну обробку двох візків двома робочими станціями, як показано в рівнянні 4.1, і використовуватися в якості вхідного параметра x .

Формула для вхідної змінної y наведена у рівнянні 4.2, де середнє стандартне відхилення модифіковано відповідно до рівняння 4.3. Вхідний параметр z було визначено шляхом обчислення мінімально можливий час циклу для візка і поділивши його на два, як показано в рівнянні 4.4. Під час цих розрахунків фізично в системі одночасно могло перебувати лише десять візків; отже, як тільки починається моделювання, потрібен певний час, щоб система досягла стаціонарного стану. Експериментально було визначено, що це приблизно 100 часових кроків; отже, всі візки, оброблені до цього, були проігноровані при обчисленні значень для x , y і z .

4.3.2 Тест єдиної системи

Спочатку, щоб перевірити гіпотезу про можливість масштабування системи, було охарактеризовано макет блоку, в якому обидві робочі станції мали однакові параметри. Схема була виконана з використанням параметрів, наведених у таблиці А.5, з тією різницею, що параметри робочої станції А були застосовані до робочої станції В, а частота та варіація генерації джерела були змінені до 4+2. Результати визначення характеристик рівномірного розташування блоків наведено в Додатку А, Таблиця А.9. Застосовуючи процес, описаний у попередньому розділі, можна створити набір вхідних параметрів для будь-якої робочої станції для імітації рівномірного розташування блоків, які наведено в таблиці А.10.

4.3.3 Результати єдиної системи

Таблиця 4.5: Кореляційна таблиця для результатів паралельної та паралельної компоновки імітатора блоків.

Діапазон	Кореляція (R^2)
Загалом	0.759
Детермінований (D)	0.459
Стохастичний (S)	0.800
S&D Випадкові та альтернативні SOP	0.703
S&D черга & черга + SOP часу проходження	0.784

Параметри, наведені в Таблиці А.9, були застосовані до одиничного та послідовного макетів, а отримані результати порівнювалися з результатами паралельного та послідовно-паралельного макетів відповідно.

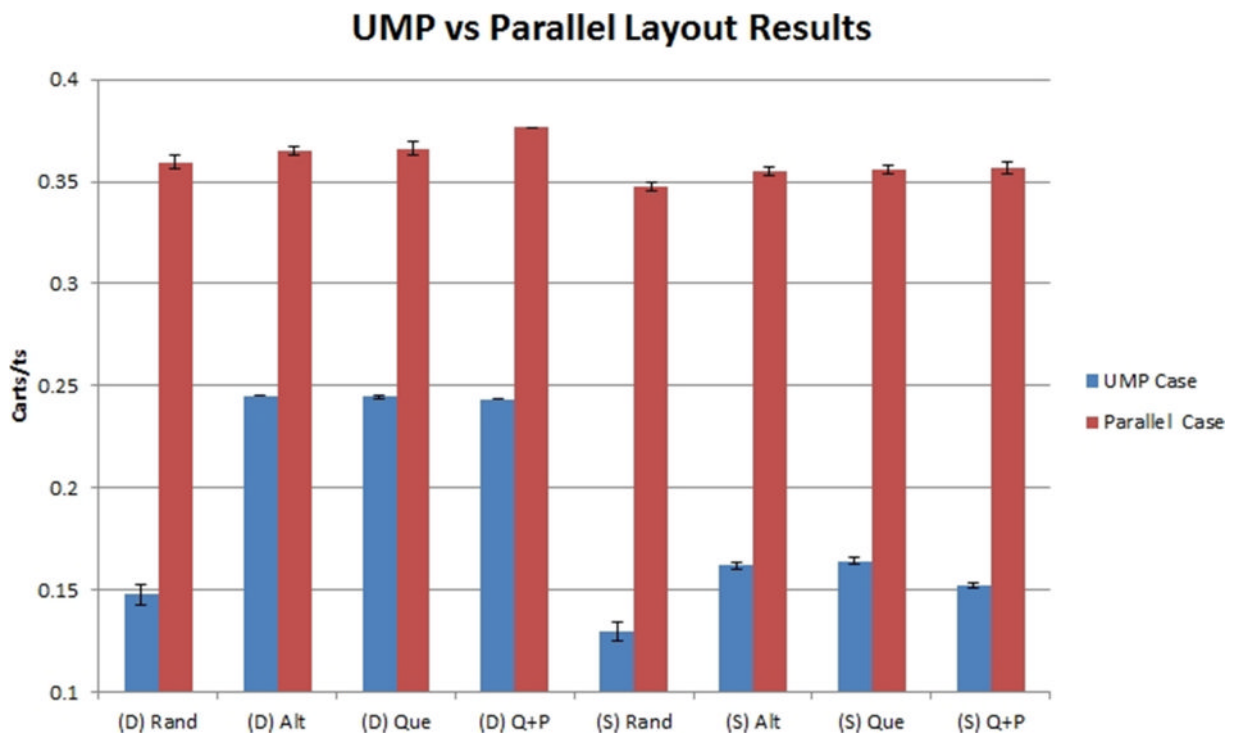


Рисунок 4.9: Результати імітації паралельної та паралельної компоновки блоків.

Таблиця 4.6: Кореляційна таблиця для результатів SMSP та послідовно-паралельної компоновки.

Діапазон	Кореляція (R^2)
Загалом	0.443
Детермінований (D)	0.487
Стохастичний (S)	0.1005
S&D Випадкові та альтернативні SOP	0.383
S&D черга & черга + SOP часу проходження	0.733

Чисельні результати для імітації паралельного розташування блоків (unit mimic parallel - UMP) та паралельного розташування наведені в Додатку А, Таблиці А.11 та Таблиці А.12. При порівнянні між усіма групами моделювання, результати компонування UMP мали загальну кореляцію 0,759, причому кореляція виявилася найсильнішою за стохастичних умов, як показано в Таблиці 4.5. Цікаво відзначити, що тенденції як у детермінованому, так і в стохастичному режимах демонструють схожі тенденції між результатами паралельного та UMP, хоча різниця у величині цих змін є різною. Наприклад, на рис. 4.9 результати для паралельного режиму показують, що альтернативний SOP перевершує випадковий SOP (0,365 візків/ts з 0,360), результати UMP також показують, що альтернативний SOP перевершує випадковий SOP (0,245 візків/ts з 0,150); однак, величина збільшення продуктивності не однакова. Було зроблено припущення, що ця різниця може бути пов'язана з проблемами масштабування.

Результати для послідовно-імітованої послідовно-паралельної (serial mimic serial-parallel - SMSP) та послідовно-паралельної компоновки також порівнювалися і наведені в Додатку А, Таблиці А.13 і Таблиці А.14. Як показано в Таблиці 4.6, загальна кореляція для SMSP і послідовно-паралельної компоновки становить 0,443. Оскільки математична залежність масштабування не була повністю визначена, початкові апроксимаційні рівняння можуть містити неточності. Ми припускаємо, що ці неточності пов'язані зі складністю компонування. Це пояснює нижчий рівень кореляції SMSP порівняно з тестами UMP.

4.3.4 Випробування неоднорідних систем

Для подальшого дослідження потенційної масштабованості різних схем було проведено ще один тест. Модифікована версія UMP у порівнянні з паралельною компоновкою була проведена зі зміненими вхідними параметрами, як зазначено в Таблиці 4.7. Компонування блоків знову було охарактеризовано за допомогою методів, описаних у розділі 4.3.1, а результати були використані для визначення параметрів, використаних у компонуванні UMP.

Таблиця 4.7: Вхідні параметри робочої станції для тестування неоднорідної системи.

Робоча станція(WS)	x (ts)	y (ts)	z (ts)
A	4	3	1
B	5	4	1

Таблиця 4.8: Кореляційна таблиця для результатів UMP та паралельної розкладки.

Діапазон	Кореляція (R^2)
Загалом	0.453
Детермінований (D)	0.338
Стохастичний (S)	0.169
S&D Випадкові та альтернативні SOP	0.732
S&D черга & черга + SOP часу проходження	0.682

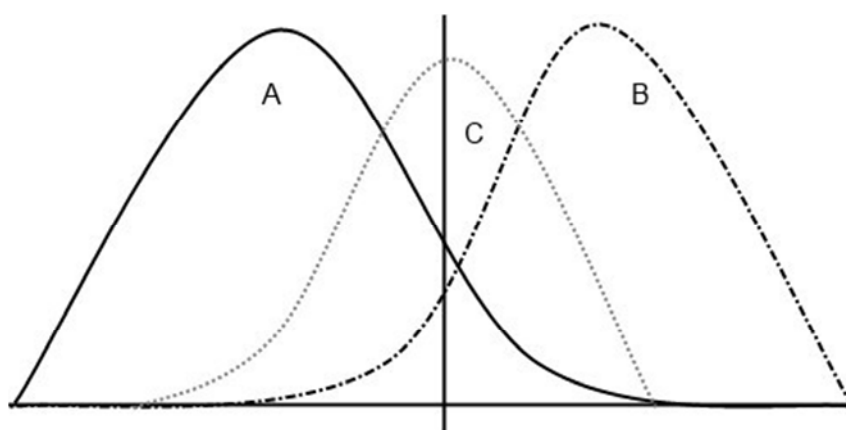


Рисунок 4.10: Апроксимація розподілу ймовірностей.

4.3.5 Результати та обговорення неоднорідної системи

Результати для нерівномірного UMP та паралельного розташування наведені в Таблиці 4.8. Знову ж таки, загальна кореляція не дуже висока, проте, схоже, існують тенденції, подібні до тих, що згадувалися в розділі 4.3.3.

Розглядаючи результати тестування рівномірної та нерівномірної системи, можна стверджувати, що існує математична залежність, яка пов'язує результати немасштабованого моделювання СТМ (паралельна та послідовно-паралельна схеми) та їх масштабовані аналоги (UMP, UMSP). Однак, метод, за

допомогою якого розподіли ймовірностей двох окремих робочих станцій об'єднуються та апроксимуються (рис. 4.10) третім розподілом, може потребувати модифікації. Поточний метод усереднення двох розподілів не дає точних результатів, про що свідчать низькі значення кореляції. На взаємозв'язок між масштабованим і немасштабованим макетами явно впливають складність і неоднорідність, що призводить до нелінійного зв'язку. Більш повне і глибоке дослідження точного математичного взаємозв'язку між масштабованими і немасштабованими системами буде залишено для подальшої роботи.

4.4 Надійність системи

Як зазначалося раніше, мета NAASS полягала не лише в тому, щоб уможливити моделювання різних макетів СТМ, але й у тому, щоб дозволити вводити збурення в макет СТМ для того, щоб спостерігати за стійкістю різних SOP. У цьому розділі було проведено тест, в якому зміни в параметрах робочої станції були введені в середині моделювання і зібрані результати.

4.4.1 Вступ

Таблиця 4.9: Таблиця вхідних параметрів робочої станції для тестів на стійкість.

Сценарій	Робочі станції	x (ts)	y (ts)	z (ts)
1	Непарні (A, C, E..)	4	3	1
	Парні (B, D, F..)	5	4	1
2	Непарні	8	3	1
	Парні	5	4	1

Для експерименту для кожного макета було запущено три сценарії. Сценарій 1 використовує параметри, перелічені в розділі 4.3.2, доповнені параметрами, переліченими в таблиці 4.9 для робочих станцій у сценарії 1. Другий сценарій був подібний до першого, за винятком того, що використовувалися параметри робочих станцій зі сценарію 2 Таблиці 4.9. Третій і останній сценарій починався з ідентичних параметрів першого тесту,

але параметри робочої станції були програмно змінені на параметри другого тесту після того, як було виконано 75 візків. Сценарій 1 вважався "хорошим", і в ньому робоча станція А (і кожна інша робоча станція) мала дещо швидший час обробки, ніж інші робочі станції. Сценарій 2 вважався "поганим" випадком, в якому потенційна несправність або знос інструменту призвели до того, що робоча станція А (і кожна інша робоча станція) з самого початку мала повільніший час обробки, ніж інші робочі станції. Сценарій 3 передбачав, що несправності на робочих станціях виникали в середині симуляції у вигляді ступінчастої функції (випадок перемикання).

4.4.2 Результати та обговорення

Результати тестів на стійкість показані на рис. 4.11, рис. 4.12, рис 4.13 та рис. 4.14. Як і очікувалося, у більшості випадків "випадки перемикання", в яких параметри робочої станції змінювалися під час моделювання, показали нижчу продуктивність, ніж "хороші" випадки, але кращу, ніж "погані" випадки. У всіх чотирьох макетах, як для стохастичних, так і для детермінованих умов, одна або декілька SOP ЖАСА послідовно перевершували інші SOP, демонструючи надійну оптимізаційну здатність. Цікаво відзначити, що в усіх випадках, окрім паралельного планування, Q+P SOP постійно перевершував SOP на основі черги. Однак, можливо, це питання потребує подальшого вивчення, перш ніж робити загальні висновки.

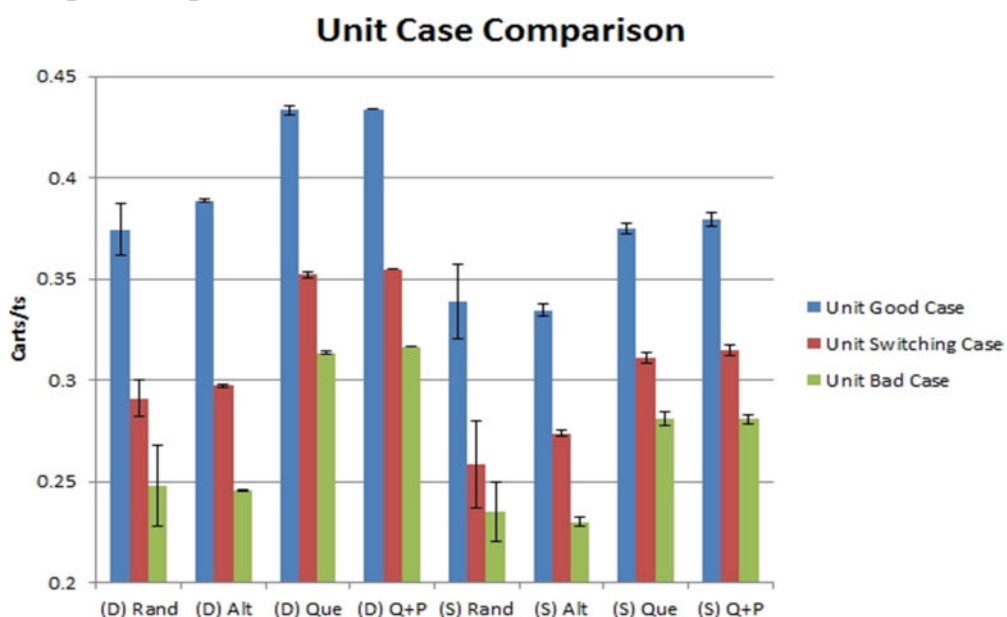


Рисунок 4.11: Результати тесту на стійкість компонування блоків.

Serial Comparison

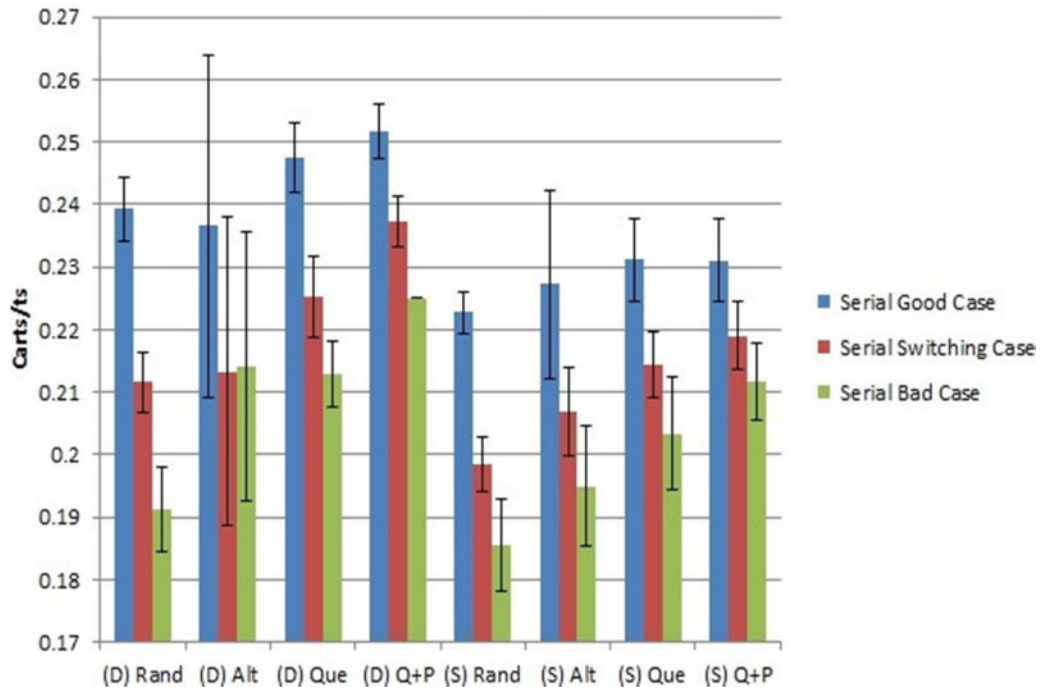


Рисунок 4.12: Результати тесту на стійкість послідовного макета.

Parallel Case Comparison

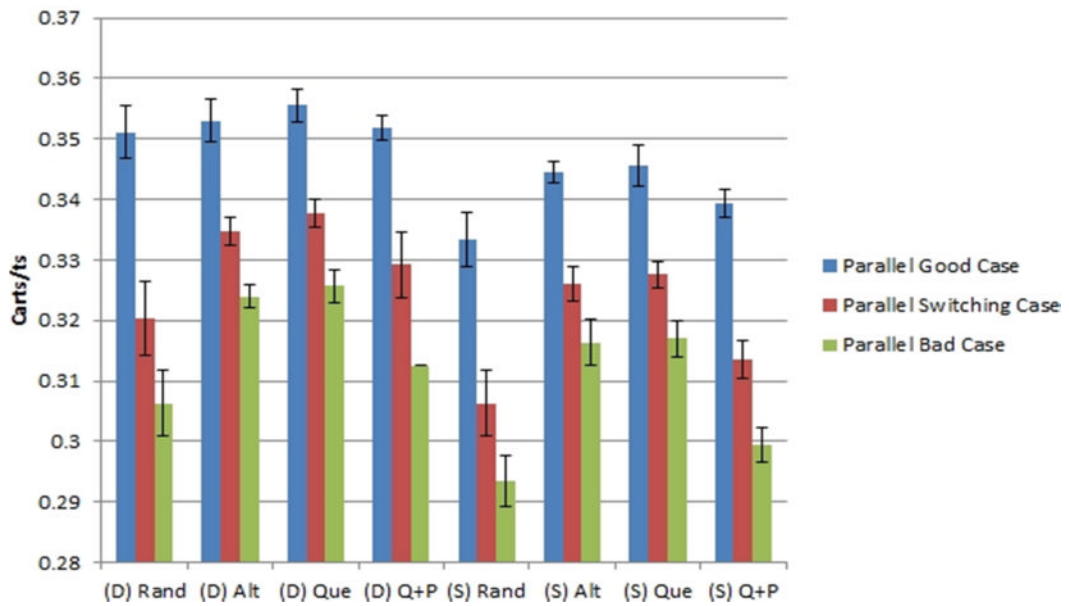


Рисунок 4.13: Результати тесту на стійкість до паралельної компоновки.

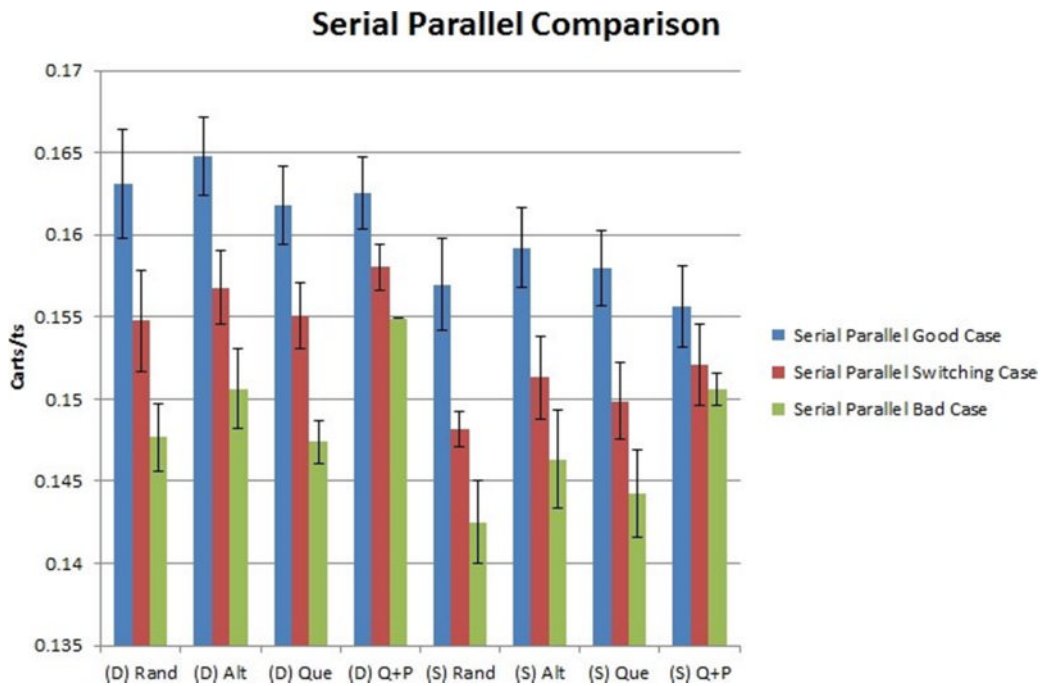


Рисунок 4.14: Результати тесту на стійкість послідовно-паралельної КОМПОНОВКИ.

4.5 Висновки

Підсумовуючи цей розділ, було визначено відповідну кількість візків для прогону через систему, а також кількість повторень для кожного випробування, щоб забезпечити статистичну впевненість у точності та стабільності результатів, отриманих при ефективному використанні ресурсів. Було визначено чотири макети для NAASS, які охоплюють широкий діапазон складності та розміру системи для цілей тестування SOP. Було проведено дослідження можливості масштабування макетів, використовуючи прості макети для моделювання великих і складних макетів, але результати не були переконливими і натякали на складну нелінійну залежність між масштабованими і немасштабованими результатами. Нарешті, надійність чотирьох SOP було перевірено шляхом внесення змін до параметрів робочих станцій, які вимагали б зміни в плануванні для підтримки максимальної продуктивності. Результати вищезазначених тестів показали, що два SOP ЖОСА постійно перевершували два інших SOP.

5 СПЕЦІАЛЬНА ЧАСТИНА

5.1 Внески

Досягнення та внесок у сферу досліджень є наступними:

- Створено локалізовану систему прийняття рішень на основі жадібної оптимізації, яка може бути запрограмована на стандартне обладнання.
- Створено програму, яка може імітувати макети СТМ, та перевірили програму на обладнанні.
- Проведено дослідження більш складних питань, таких як масштабування, надійність та продуктивність системи.
- Визначено привабливі напрямки для майбутніх досліджень на основі отриманих результатів.

Кожен з вищезгаданих пунктів буде розглянуто більш детально в наступних розділах.

5.2 Локалізовані системи прийняття рішень

У розділі 1.2 АСА визначається як самодостатня система, яка намагається приймати рішення на основі наданої їй інформації за допомогою оптимізаційних алгоритмів. У Розділі 3 представлено та визначено дві версії жадібних алгоритмів АСА (ЖАСА). Це: ЖАСА на основі черги та ЖАСА на основі черги + часу проходження, відповідно.

Завдання, представлене в Розділі 2, полягає в тому, щоб створити алгоритми і парадигму, які були б орієнтовані на майбутнє і імітували ефективність виробничих парадигм, запропонованих за останні два десятиліття, обмежуючи при цьому складність таким чином, щоб їх можна було застосувати до сучасного промислового обладнання. Дослідження не виявили жодних можливостей динамічної оптимізації або подібну практику, що наразі застосовується в промисловості. Приклад двох алгоритмів ЖАСА, закодованих у LabVIEW (використаних для валідації імітаційної платформи) і реалізованих на National Instruments Compact RIO, наведено в Додатку В, розділ

В.1. Ці ж два алгоритми були також запрограмовані в промисловому стандартному логічному коді, реалізовані на ПЛК Allen & Bradley і використані для управління BARD і показали таку ж поведінку, як і код в LabVIEW. Обидві програми на сходовому алгоритмі можна знайти в Додатку Б. Код LL для SOP на основі черги показано на рисунках Б.1, Б.2, Б.3 та Б. 4. Код LL для SOP на основі черги + часу проходження показано на рис. В.5, рис. В.6 та рис. В.7.

5.3 Симулятор мережевої автономної системи автоматизації

Мета та призначення NAASS детально описані в Розділі 3.3. Вперше NAASS було перевірено шляхом порівняння результатів фізичних випробувань, проведених на BARD, з віртуальною моделлю тієї ж системи в NAASS. NAASS здатна моделювати СТМ будь-якої конфігурації, складу і розміру (за умови, що відповідні компоненти були закодовані). NAASS полегшила порівняння та бенчмаркінг ряду парадигм оптимізації розкладу (SOP), оскільки вони були застосовані до численних макетів. NAASS дозволила реєструвати, аналізувати і спостерігати продуктивність і поведінку різних SOP, коли застосовувалися різні вхідні параметри і вводилися різноманітні макети. NAASS дозволила програмно вводити системні збурення в різних макетах, щоб полегшити тестування стійкості SOP. Нарешті, NAASS також дозволила включити стохастичне моделювання різних процесів в СТМ для спостереження за впливом на продуктивність SOP. Система NAASS була перевірена на фізичному обладнанні і є повністю функціональною, але має багато можливостей для вдосконалення і розширення з точки зору програмування, обсягу і можливостей. У наступних розділах перераховані деякі з запропонованих напрямків майбутньої роботи.

5.3.1 Існуючі компоненти

Наразі єдиними компонентами, запрограмованими в NAASS, з яких користувач може побудувати схему, є: вихідні вузли, приймально-відправні вузли, робочі станції, стрілочні переводи та з'єднувачі стрілочних переводів. З них вузли джерела та приймача не потребують жодних змін або розширення,

оскільки вони є початковою та кінцевою точками будь-якого плану в NAASS. Робочі станції можуть бути розширені, щоб дозволити додаткові розподіли ймовірностей для моделювання стохастичності, окрім існуючого гауссівського розподілу ймовірностей.

Якщо припустити, що завданням вузла з'єднання стрілочних переводів є лише об'єднання двох шляхів в один, то вузол з'єднання не потребуватиме подальших змін. Однак є можливість для розширення та модифікації вузла розгалужувача стрілочних переводів. По-перше, поточний вузол-розгалужувач стрілочної доріжки передбачає, що два шляхи, по яких він може направляти візки, закінчуються однаковими процесами. Для інших сценаріїв, в яких можуть використовуватися розгалужувачі стрілочних переводів, результатом можуть бути неідентичні процеси або декілька (>2) виходів. По-друге, чотири раніше згадані SOP жорстко закодовані в NAASS. Отже, існує потреба в модифікаціях, які дозволять кодувати нові SOP та/або завантажувати їх зі створених користувачем файлів.

5.3.2 Додаткові компоненти

Беручи до уваги існуючі компоненти в NAASS, існує ряд компонентів, які можуть бути додані, щоб дозволити NAASS більш реалістично і повно моделювати мережі транспортування матеріалів у промисловості. Запропоновані компоненти, які слід додати, включають в себе наступні:

- Лінійні перемикачі (рис. 2.1(д))
- Перехресна доріжка (рис. 2.1(г))
- Підйомна доріжка (рис. 2.1(е))

Для кожного з перерахованих вище компонентів потрібно буде створити окремі функції, а також методи застосування різних SOP до кожного з них.

5.3.3 Подорож у двох напрямках

Оскільки NAASS була змодельована на існуючому обладнанні, яке знаходиться в лабораторії, вона передбачає, що всі візки можуть рухатися тільки в одному напрямку. Було б корисно розширити NAASS, щоб вона могла

моделювати двосторонній рух по одній і тій же траєкторії, і включити всі додаткові вимоги, які з цього випливають. Це дозволило б розширити типи систем, які можна моделювати за допомогою NAASS.

5.3.4 Реалізація обмеження черги

Наразі в NAASS максимальна кількість візків, які можуть одночасно рухатися по шляху між будь-якими двома компонентами, задається користувачем, як показано на рис. 3.4(г). Це може вплинути на точність моделі. Наприклад, якщо одна доріжка від перемикача досягає пропускної здатності, незалежно від кількості візків на іншій вихідній доріжці, перемикач не пропустить жодного візка, доки обидві можливі доріжки не стануть пропускними. Це результат того, як було реалізовано роздільник доріжок, і можна було б розглянути інші методи реалізації.

5.3.5 Мова реалізації

Наразі NAASS реалізовано в системі Mathematica. Хоча Mathematica є потужною мовою математичного моделювання і має багато переваг на етапах створення прототипів і тестування програмного забезпечення або симуляції, вона може бути не найефективнішою мовою для виконання великомасштабних, складних симуляцій. Для подальшої роботи та впровадження NAASS пропонується більш придатна мова, наприклад, C.

5.4 Продуктивність АСА, масштабування системи та надійність системи

На початку цього дослідження було поставлено завдання оцінити гіпотезу про те, що "мережеві та локальні аспекти прийняття рішень (раніше визначеної) автономної системи автоматизації (АСА) можуть бути застосовані до промислової автоматизації для створення надійних, масштабованих, динамічно самооптимізуючих систем". Як згадувалося вище, платформа для моделювання (NAASS) була розроблена для того, щоб порівняти і оцінити

продуктивність двох вищезгаданих алгоритмів один з одним, а також з двома додатковими SOP.

Результати, наведені в Розділі 4, показують, що в усіх чотирьох СТМ, до яких застосовувалися чотири SOP, незалежно від розміру або складності системи, два алгоритми ЖАСА постійно перевершували SOP без вбудованого інтелекту. Ця продуктивність зберігалася в детермінованих і стохастичних системах. Таким чином, самооптимізаційна частина проектного завдання була підтверджена.

У Розділі 4 також наведено коротке дослідження можливості моделювання складної системи за допомогою простішої, але репрезентативної системи. Результати вказують на можливість існування взаємозв'язку між масштабованими і немасштабованими результатами, але він, швидше за все, буде нелінійним і математично складним.

Після тестів масштабування були проведені подальші тести для визначення стійкості SOP ЖАСА до системних збурень. У всіх чотирьох макетах, як в стохастичних, так і в детермінованих умовах, два ЖАСА SOP в цілому перевершили два SOP без вбудованого інтелекту, продемонструвавши адаптивність і здатність до динамічної оптимізації.

Отже, на основі результатів, представлених у Розділі 4, проектне завдання було дійсно підтверджено. Незалежно від складності макета або розміру системи, алгоритми ЖАСА постійно перевершували алгоритми без вбудованого інтелекту і демонстрували докази надійності та динамічної оптимізації.

5.5 Подальші дослідження

Окрім запропонованого розширення та модифікації кодексу NAASS, існують додаткові сфери, в яких пропонуються подальші дослідження.

5.5.1 Використані алгоритми оптимізації

З двох існуючих алгоритмів ЖАСА один є простим однокроковим алгоритмом жадібної оптимізації (на основі черги), а інший - простим

однокроковим алгоритмом евристичної оптимізації. Вони обидва класифікуються як "однокрокові алгоритми", які враховують лише вхідні дані та на компоненти, що безпосередньо з ними пов'язані, а не на те, який шлях чи вибір дозволить поточному візку дістатися до вузла стоку за найкоротший час. Алгоритми жадібної оптимізації можуть зупинитися на локальних оптимумах і не вести систему в цілому до глобального оптимуму. Дослідження багатокрокових жадібних алгоритмів може дати кращі SOP. Також не жадібні алгоритми оптимізації можуть перевершити жадібні алгоритми, якщо складність алгоритмів може бути реалізована на сучасному апаратному забезпеченні.

5.5.2 Використані метрики

Наразі єдиним показником, який використовується для порівняння та бенчмаркінгу різних SOP, є кількість візків, виконаних за один часовий крок у симуляції. Хоча цієї єдиної метрики достатньо для початкового тестування, валідації та досліджень, було б корисно розробити і визначити додаткові метрики, за якими можна оцінити ефективність і продуктивність SOP при застосуванні до макету з різними вхідними параметрами.

5.5.3 Виведення псевдокоду

Після тестування різних SOP і визначення того, який підходить для практичного застосування в реальному світі, було б корисно мати можливість виводити блок-схему для кожного з SOP або, принаймні, якусь форму "псевдокоду", що дозволило б програмісту застосувати алгоритм до обраного ним або нею обладнання для керування. В ідеалі, це була б опція в NAASS. Методи автоматизації генерації логічного коду на основі діаграм станів [5], компонентних моделей [6], моделей систем високого рівня [7] або мереж Петрі [8-10] вже існують; отже, якби псевдокод можна було виводити в одному з вищезгаданих форматів, весь процес від симуляції до LL-коду можна було б автоматизувати.

5.5.4 Відношення масштабування

Як зазначалося в розділі 4.3, результати експериментів з масштабуванням не були остаточними, але продемонстрували складний нелінійний зв'язок між масштабованими та немасштабованими результатами. Отже, необхідні подальші дослідження, щоб визначити та охарактеризувати точний взаємозв'язок, що дозволить зменшити ресурси, необхідні для моделювання складних систем.

5.5.5 Подальша перевірка

У розділі 4.1.3 було перераховано три типи систем. Через обмеженість ресурсів під час тестування продуктивності системи та характеристик SOP було використано лише систему типу 2. Решта два типи систем також повинні бути досліджені на всіх чотирьох макетах, щоб дослідити вплив співвідношення часу проходження між компонентами та часу обробки компонентів на можливі співвідношення масштабування та надійність SOP.

5.6 Підсумок

Досягнення і прогрес, досягнуті в цьому дослідженні, показують, що надійність і продуктивність виробничої системи можна покращити, додавши окремим компонентам можливість прийняття локальних рішень. Продуктивність системи в цілому може бути змодельована і налаштована для досягнення бажаної продуктивності. Блок-схеми доопрацьованих SOP можуть бути використані для створення відповідного коду, а отже, скоротити час програмування промислової виробничої системи і забезпечити більш швидку реконфігурацію.

6 Охорона праці та безпека в надзвичайних ситуаціях

6.1 Заходи, які зменшують небезпеку виникнення вибухів та пожеж

Запобігання вибухів та пожеж – це комплекс організаційних заходів і технічних засобів, спрямованих на виключення можливості виникнення вибухів та пожеж.

Організаційні і технічні заходи щодо запобігання пожежі реалізуються ще на стадії проектування окремих об'єктів підприємств. При цьому заздалегідь вивчаються особливості технологічних процесів і об'єктів, можливі причини і джерела виникнення вибухів та пожеж. Запобігання пожежі великою мірою сприяє правильне планування, розміщення основних об'єктів з урахуванням рельєфу місцевості, дотримання протипожежних розривів між будівлями відповідно до вимог генерального плану.

Попередження вибухів та пожеж на підприємствах досягається:

- запобіганням утворенню горючого середовища;
- запобіганням виникненню в горючому середовищі або появи в ньому джерел запалювання.

Запобігання утворення горючого середовища повинно досягатися:

- максимально можливим застосуванням негорючих і важко горючих речовин і матеріалів;
- обмеженням маси і об'єму горючих речовин, матеріалів та найбільш безпечним способом їх розміщення;
- ізолюванням горючого середовища;
- підтримуванням концентрації горючих газів, пари, суспензій і окислювача в суміші за межею їх спалаху;
- достатньої концентрації флегматизатора в повітрі захищованого об'єкту;
- підтримуванням його температури і тиску, за якими розповсюдження полум'я неможливе;
- максимальною механізацією і автоматизацією технологічних процесів, пов'язаних з вживанням горючих речовин;

- встановленням пожежонебезпечного обладнання, по можливості, в ізольованих приміщеннях чи на відкритих площадках;
- застосуванням для горючих речовин герметичного обладнання і тари;
- застосуванням пристроїв захисту виробничого обладнання з горючими речовинами від ушкоджень і аварій, встановленням відключаючих, відсікаючих та інших пристроїв;
- застосуванням ізольованих відсіків, камер, кабін. Попередження утворення в горючому середовищі джерел запалювання повинно досягатися такими основними заходами:
 - застосуванням машин, механізмів, обладнання, пристроїв, під час експлуатації яких не утворюються джерела запалювання;
 - застосуванням електрообладнання, що відповідає класу пожежовибухонебезпеки приміщення або зовнішньої установки, групі і категорії вибухонебезпечної суміші;
 - застосуванням в конструкції швидкодіючих засобів захисного відключення можливих джерел запалювання;
 - застосуванням технологічного процесу і обладнання, що відповідає вимогам електростатичної іскробезпеки;
 - пристроєм блискавкозахисту будівель, споруд і обладнання. Будівлі та споруди складів паливно-мастильних матеріалів захищають від прямих ударів блискавки, електростатичної та електромагнітної індукції та заносу потенціалів;
 - підтримкою тиску в горючому середовищі нижчого за максимально припустимий за горючістю;
 - зменшенням визначального розміру горючої суміші середовища нижче максимально припустимого за горючістю.;
 - регламентацією виконання, застосування і режиму експлуатації машин, механізмів та іншого обладнання, матеріалів і виробів, що можуть бути джерелом запалювання горючого середовища;
 - застосуванням енергоустаткування, що відповідає класу пожежовибухонебезпеки приміщення або зовнішньої установки, групі і категорії вибухонебезпечної суміші;
 - застосуванням технологічного процесу і обладнання, що відповідає вимогам електростатичної іскробезпеки.;

– регламентацією максимально допустимої температури нагрівання поверхонь обладнання і матеріалів, що можуть увійти в контакт з горючим середовищем. Режимми роботи насосів, перекачувальних паливно-мастильних матеріалів не повинні спричиняти підвищене нагрівання їх поверхонь;

– регламентацією максимально допустимої енергії іскрового розряду в горючому середовищі. Знижувати енергію іскрового розряду можна, зменшуючи напруження між частинами обладнання, при якому відбувається іскровий розряд в горючому середовищі;

– регламентацією максимально допустимої температури нагрівання горючих речовин, матеріалів і конструкцій;

– застосуванням інструмента, що не іскрить під час роботи з легкозаймистими речовинами. Слід застосовувати інструмент і пристосування, що не висікають іскри під час ударів і падіння;

– ліквідацією умов для хімічного самозагоряння речовин і матеріалів. До samozапалювальних речовин в технологічних процесах належать пірофорні речовини, що розігріваються при окисленні киснем повітря до 600 °С;

– усуненням контакту з повітрям пірофорних речовин.

6.2 Фізичні основи електробезпеки

Величина струму, що проходить через тіло людини при її попаданні під напругу, в найбільшій мірі визначає тяжкість ураження. Для розробки технічних і організаційно-технічних заходів і засобів профілактики електротравм важливо знати, від яких конструктивних особливостей електроустановок, їх робочих параметрів і стану залежить можлива величина струму через людину при потраплянні під напругу. Крім того, важливо, щоб весь електротехнічний персонал, усі працівники, робота яких пов'язана з експлуатацією електроустановок, чітко розуміли, чим обумовлена, що є причиною тієї чи іншої вимоги з електробезпеки. Таке знання, розуміння вимог чинних нормативів з електробезпеки сприятиме дотриманню їх працівниками, і якраз розуміння цих вимог відрізняє працівників п'ятої групи з електробезпеки від четвертої, і є обов'язковою складовою їх професійної підготовки з питань безпеки.

У реальній електричній мережі (повітряній чи кабельній) опір ізоляції проводів відносно землі розподіляється по всій довжині мережі — опорні, підвісні, натяжні ізолятори, ізоляція кабелю. Чим більша протяжність мережі, тим більше ізоляторів, які працюють паралельно, і менший загальний опір ізоляції проводів відносно землі. Необхідний опір ізоляції регламентується чинними нормативами. На практиці ізоляція струмопроводів виконується з реальних діелектриків, питомий опір яких не дорівнює нескінченності. Внаслідок старіння ізоляції, її частого зволоження, забруднення, нагріву, дії агресивного середовища тощо, питомий опір ізоляції знижується. Тому кожна ділянка довжини проводу має опір ізоляції певного значення або провідність, яка відрізняється від нуля, а при роботі реальної мережі мають місце постійні втрати струму (виток струму) через ізоляцію і землю. Таким чином, незважаючи на наявність ізоляції, токопроводи електромережі електрично зв'язані між собою і землею провідниками (ізоляцію) з великим опором.

Відповідно до зазначеного вище, кожна ділянка довжини проводу електромережі, що знаходиться під напругою, крім опору ізоляції має певну ємність відносно землі. Тому при дотиці людини до неізольованої струмовідної частини (проводу тощо) функціонуючої електромережі струм через людину обумовлюється величиною напруги дотику і ємністю зазначеної вище системи. Ємнісна складова струму через людину при потраплянні під напругу в розгалужених мережах може досягати небезпечних для людини значень. Тому навіть при відключенні мережі від джерела живлення для ремонтнопрофілактичних робіт тощо, необхідно заземлити кожен провід переносним заземленням і тільки після цього та перевірки відсутності напруги допускати персонал до роботи.

6.3 Підвищення стійкості роботи об'єктів електроенергетики в воєнний час

Збереження енергосистеми під час війни

Це питання має три відповіді. Перша – суто військова. Україні потрібно більше засобів протиповітряної та протиракетної оборони, які б захищали енергетичні об'єкти.

Велика кількість обладнання для високовольтних мереж, наприклад, трансформаторів не лежить просто так на складах, а потребує тривалого виготовлення.

Тому після ракетних ударів енергетики спочатку повинні оглянути об'єкт, визначити, що саме пошкоджено і потребує заміни, а вже потім починати працювати над усуненням пошкоджень.

Тому оперативно відновлювати енергосистему після пошкоджень – це складне завдання.

Другий фронт – це робота Уряду над залученням технічної та грантової допомоги для відновлення енергосистеми.

Уряд загалом, і Міністерство енергетики зокрема, наполегливо працюють над залученням коштів і ресурсів міжнародних партнерів для того, щоб в енергетиків було все необхідне для усунення результатів обстрілів.

Наприклад, нещодавно повідомлялося про надання 550 млн євро на відновлення енергетичної інфраструктури з боку Європейського інвестиційного банку.

Окремі закупівлі здійснюються з боку USAID за кошти міжнародних донорів в рамках програм, що реалізують Міненерго.

Третій аспект – це поведінка споживачів. Мова йде про економію і грамотне споживання.

Енергосистема переживає найбільші навантаження протягом, т.зв. "пікових годин" доби, тобто з 6:00 до 9:00 і з 17:00 до 23:00. В цей час українці можуть допомогти енергетикам за рахунок таких простих кроків:

не вмикати одночасно всі електроприлади та зменшити їх потужність;

не використовувати вуличне освітлення;

змінити побутові звички і максимальну кількість справ, наприклад, прання та нагрівання води бойлером робити у нічні "непікові" години;

за можливістю, замінити використання електроприладів для опалення або приготування їжі на менш енергомісткі пристрої.

Реалізація цих простих, на перший погляд заходів, у критичні моменти вже дозволяє Україні економити від 7 до 20% електроенергії.

За рахунок цього, енергосистема може працювати не на повну потужність, електростанції – економити вугілля, що знадобиться для

проходження зимового опалювального періоду, а енергетики у випадку аварії – отримують можливість вести роботи з мінімальними відключеннями споживачів.

У довгостроковій перспективі побутовим споживачам слід очікувати зростання цін на енергоресурси.

Чинний тариф — до 2,64 грн за 1 кВт-год - є субсидованим за рахунок промислових споживачів. За теперішніх обставин, довго така ситуація тривати не буде.

Промисловість просто не витримає додаткового зростання вартості електроенергії, що становить значну частину собівартості продукції. Тому на тлі очікування зростання цін інвестиції в енергоефективність та альтернативні джерела енергії будуть розумним рішенням.

Найпростіші рішення, такі як розетки з таймером, коштують кілька сотень гривень та можуть бути куплені будь-яким господарством, а для реалізації складніших – таких як встановлення сонячних панелей або термомодернізація будинку – доцільно розглянути через можливість залучення грантів чи кредитів у банках.

Вимоги до будівництва комунально-енергетичних систем

Електропостачання є основою усякого виробництва. Порушення нормальної подачі електроенергії на об'єкт чи окремі ділянки виробництва може призвести до повного припинення роботи об'єкта.

Для забезпечення надійного електропостачання в умовах війни при його проектуванні та будівництві повинні бути враховані наступні основні вимоги, що впливають із задачі ЦЗ.

Вимоги до систем електропостачання:

1. Електропостачання повинне здійснюватися від енергосистем, до складу яких входять електростанції, що працюють на різних видах палива. Великі електростанції варто розміщати одну від одної і від великих міст на відстані не менше двох радіусів зон можливих руйнувань.

2. Районні понижуючі станції, диспетчерські пункти енергосистем і лінії електропередач варто розміщати за межами зон можливих сильних руйнувань.

3. Постачання електроенергією великих міст і об'єктів, що не припиняють роботу у воєнний час, необхідно передбачати від двох незалежних джерел. При електропостачанні об'єкта від одного джерела повинне бути не менш двох уведень з різних напрямків.

4. Трансформаторні підстанції необхідно надійно захищати, їхня стійкість повинна бути не нижчою ніж стійкість самого об'єкта.

5. Електроенергію до ділянок виробництва варто подавати по незалежних електрокабелях, прокладених у землі на глибині 0,8 – 1,2 м.

6. Крім того, необхідно створювати автономні резервні джерела електропостачання. Для цього можна використовувати пересувні електростанції на залізничних платформах і суднах, малопотужні електростанції, не включені в енергосистеми, тощо.

7. При проектуванні систем електропостачання варто зберігати в якості резервних дрібні стаціонарні електростанції об'єктів.

8. У містах, розташованих на берегах морів і рік, необхідно створювати берегові пристрої для прийому електроенергії від судових енергоустановок.

9. Система електропостачання повинна мати захист від впливу електромагнітного імпульсу ядерного вибуху.

ВИСНОВОК

Було розроблено методологію побудови гнучких виробничих систем, що реконфігуруються, надаючи можливість окремим компонентам приймати локальні рішення під впливом алгоритмів оптимізації. Для оцінки цієї методології було створено імітаційну платформу - симулятор автономної системи автоматизації (NAASS), яка була перевірена на фізичному об'єкті - демонстраційному стенді базової автоматизації та робототехніки (BARD). NAASS був використаний для оцінки продуктивності, надійності та масштабованості різних систем транспортування матеріалів при різних парадигмах оптимізації розкладу (SOP). Результати показали, що SOP з вбудованим інтелектом були більш надійними і перевершували ті, які не мали можливості локального прийняття рішень як в стохастичних, так і в детерміністичних умовах. Дослідження можливості масштабування системи дало змогу отримати докази того, що зв'язок між масштабованими і немасштабованими системами є складним і нелінійним. Були також визначені привабливі напрямки для майбутніх досліджень, які могли б принести корисні для промисловості результати.

ПЕРЕЛІК ПОСИЛАНЬ

1. [1] A Dugenske and A Fraser. The national electronics manufacturing initiative (nemi) plug and play factory project. *International Journal of Computer Integrated Manufacturing*, 13(3):225–244, 2000. URL <http://www.tandfonline.com/doi/abs/10.1080/095119200129876>.
2. [2] D.M. Tilbury M.R. Lucas. A study of current logic design practices in the automotive manufacturing industry. *International Journal of Human-Computer Studies*, 59:725–753, 2003.
3. [3] F. Jammes and H. Smit. Service-oriented paradigms in industrial automation. *IEEE Transactions on Industrial Informatics*, 1(1):62–70, February 2005. ISSN 1551-3203. doi: 10.1109/TII.2005.844419. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1411764>.
4. [4] K. Wilcox. Engineering salaries rise, October 2013. URL <http://www.asce.org/cemagazine/articlens.aspx?id=25769810172#.UlXvzUGG3Rl>.
5. [5] K. Akantziotis S. Manesis. Automated synthesis of ladder automation circuits based on state-diagrams. *Advances In Engineering Software*, 36:225–233, 2005.
6. [6] D. Orive E. Estevez, M. Marcos. Automatic generation of PLC automation projects from component-based models. *International Journal of Advanced Manufacturing Technology*, 35:527–540, 2007.
7. [7] Mohsen a. Jafari and Thomas O. Boucher. A rule-based system for generating a ladder logic control program from a high-level systems model. *Journal of Intelligent Manufacturing*, 5(2):103–120, April 1994. ISSN 0956-5515. doi: 10.1007/BF00572403. URL <http://link.springer.com/10.1007/BF00572403>.
8. [8] N. Ajloni M. Uzam, A. H. Jones. Conversion of petri net controllers for

- manufacturing systems into ladder logic diagrams. Technical report, Intelligent Machinery Division, Research Institute for Design, Manufacturing and Marketing, University of Salford, 1996.
9. [9] E. Twiss M. Zhou, editor. Design of Industrial Automated Systems Via Relay Ladder Logic Programming and Petri Nets, volume 28 of IEEE Transactions on Systems, Man, and Cybernetics- Part C: Applications and Reviews, 1998. IEEE. 79
 - 10.[10] D. C. McFarlane J. L. Chirn. Petri nets based design of ladder logic diagrams. Technical report, Institute for Manufacturing, University of Cambridge, 2000.
 - 11.[11] Jack R. Meredith. Implementing the automated factory. Journal of Manufacturing Systems, 6(1):1–13, January 1987. ISSN 02786125. doi: 10.1016/0278-6125(87)90045-8. URL <http://linkinghub.elsevier.com/retrieve/pii/0278612587900458>.
 - 12.[12] Y Koren, U Heisel, and F Jovane. Reconfigurable manufacturing systems. CIRP Annals - Manufacturing Technology, 48(2):527–540, 1999. URL <http://www.sciencedirect.com/science/article/pii/S0007850607632326>.
 - 13.[13] Yoram Koren and Moshe Shpitalni. Design of reconfigurable manufacturing systems. Journal of Manufacturing Systems, 29(4):130–141, October 2010. ISSN 02786125. doi: 10.1016/j.jmsy.2011.01.001. URL <http://linkinghub.elsevier.com/retrieve/pii/S0278612511000021>.
 - 14.[14] Andreas Gebhardt and Jan-steffen Hoetter. CIRP Encyclopedia of Production Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-642-20616-0. doi: 10.1007/978-3-642-20617-7. URL <http://link.springer.com/10.1007/978-3-642-20617-7>.
 - 15.[15] Giovani Da Silveira, Denis Borenstein, and FS Fogliatto. Mass customization: Literature review and research directions. International Journal

- of Production Economics, 72(49):1–13, 2001. URL <http://www.sciencedirect.com/science/article/pii/S0925527300000797>.
- 16.[16] Weiming Shen, Lihui Wang, and Qi Hao. Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, 36(4):563–577, 2006. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1643848.
- 17.[17] H.-P. Wiendahl, H.a. ElMaraghy, P. Nyhuis, M.F. Z'ah, H.-H. Wiendahl, N. Duffie, and M. Brieke. Changeable manufacturing - classification, design and operation. CIRP Annals - Manufacturing Technology, 56(2):783–809, January 2007. ISSN 00078506. doi: 10.1016/j.cirp.2007.10.003. URL <http://linkinghub.elsevier.com/retrieve/pii/S0007850607001552>.
- 18.[18] JR Hackworth and FD Hackworth. Programmable Logic Controllers: Programming Methods and Applications. 2004. URL <http://research.philadelphia.edu.jo/1744/1/ProgrammableLogicControllersProgrammingMethods.pdf>. 80
- 19.[19] G. Frey and L. Litz. Formal methods in plc programming. SMC 2000 Conference Proceedings. 2000 IEEE International Conference on Systems, Man and Cybernetics. 'Cybernetics Evolving to Systems, Humans, Organizations, and their Complex Interactions' (Cat. No.00CH37166), 4:2431–2436, 2000. doi: 10.1109/ICSMC.2000.884356. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=884356>.
- 20.[20] KH Han. Object-oriented modeling, simulation and automatic generation of plc ladder logic. cdn.intechopen.com, (January):1–17, 2010. URL http://cdn.intechopen.com/pdfs/6900/InTech-Object_oriented_modeling_simulation_and_automatic_generation_of_plc_ladder_logic.pdf.
- 21.[21] Bill Simpson. URL <http://www.thelearningpit.com/lp/doc/dl/dl-rl.html>.

- 22.[22] Ltd Fuji Electric FA Components & Systems Co. Fuji as-interface, 2006.
URL http://www.fujielectric.co.jp/fcs/eng/as-interface/as_i/control.html.
- 23.[23] Thomas Kugelstadt. Industrial interface transceivers, January 2013. URL
<http://cnx.org/content/m44375/latest/>.
- 24.[24] T. O. Boucher M. A. Jafari. A rule -based system for generating a ladder logic control program from a high-level systems models. *Journal of Intelligent Manufacturing*, 5: 103–120, 1994.
- 25.[25] M Uzam and AH Jones. Discrete event control system design using automation petri nets and their ladder diagram implementation. *The International Journal of Advanced Manufacturin* pages 716–728, 1998.
- 26.[26] Georg Frey. Automatic implementation of petri net based control algorithms on plc. *Proceedings of the American Control Conference*, 2000., pages 2819–2823, 2000. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=878725.
- 27.[27] MB Younis and G Frey. Formalization of existing plc programs: A survey. *Proceedings of CESA*, 2003. URL http://www.aut.uni-saarland.de/uploads/media/GF_MBY_CESA_jul_2003.pdf.
- 28.[28] Zhi Sheng Zhang, Cheng Li, Yi Fan Zhou, and Wen Dong Liu. A transformation method between plc ladder diagrams and instruction lists using recursive and stack algorithms. *Advanced Materials Research*, 490-495:1908–1912, March 2012. ISSN 1662-8985. doi: 10.4028/www.scientific.net/AMR.490-495.1908. URL <http://www.scientific.net/AMR.490-495.1908>. 81
- 29.[29] Paulo Leitao. Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, 22(7):979–991, October 2009. ISSN 09521976. doi: 10.1016/j.engappai.2008.09.005. URL <http://linkinghub.elsevier.com/retrieve/pii/S0952197608001437>.

- 30.[30] M.K Lim and Z Zhang. A multi-agent based manufacturing control strategy for responsive manufacturing. *Journal of Materials Processing Technology*, 139(1-3):379– 384, August 2003. ISSN 09240136. doi: 10.1016/S0924-0136(03)00535-1. URL <http://linkinghub.elsevier.com/retrieve/pii/S0924013603005351>.
- 31.[31] Norving P. Canny J. F. Malik J. M. Edwatds D. D Russel, S. J. *Artificial Intelligence: A Modern Approach*. 1995.
- 32.[32] Jennings Nicholas. R Wooldridge, Michael. *Intelligent agents: Theory and practice*. *The Knowledge Engineering Review*, 10:115–152, 1995.
- 33.[33] Jacques Ferber. *Milti-Agent Systems: An Introduction to Distributed Artificial Intelligence*. 1999.
- 34.[34] M Wooldridge. *An Introduction to Multi-Agent Systems*. Wiley, New York, 2002.
- 35.[35] Boyd N. Whorms H. Diltis, D. The evolution of control architectures for automated manufacturing systems. *Journal of Manufacturing Systems*, 10:63–79, 1991.
- 36.[36] Warneke. *The Fractical Company*. Springer, 1993.
- 37.[37] Norio Okino. Bionic manufacturing systems. In J. Peklenik, editor, *Conference on Flexible Manufacturing Systems, Past, Present-Future*, pages 73–95, 1993.
- 38.[38] Hendrik Van Brussel, Jo Wyns, Paul Valckenaers, Luc Bongaerts, and Patrick Peeters. Reference architecture for holonic manufacturing systems: Prosa. *Computers in Industry*, 37(3):255–274, 1998.
- 39.[39] V Botti and AG Boggino. *ANEMONA: A multi-agent methodology for Holonic Manufacturing Systems*. Springer-Verlag London Limited, 2008. ISBN 9781848003095. doi: 10.1007/978-1-84800-310-1.

- 40.[40] Weiming Shen. Distributed manufacturing scheduling using intelligent agents. *Intelligent Systems*, IEEE, 2002. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=988492. 82
- 41.[41] L. Monostori, J. Vancza, and S.R.T. Kumara. Agent-based systems for manufacturing. *CIRP Annals - Manufacturing Technology*, 55(2):697–720, January 2006. ISSN 00078506. doi: 10.1016/j.cirp.2006.10.004. URL <http://linkinghub.elsevier.com/retrieve/pii/S1660277306000053>.
- 42.[42] RF Babiceanu and FF Chen. Development and applications of holonic manufacturing systems: a survey. *Journal of Intelligent Manufacturing*, 17:111–131, 2006. URL <http://link.springer.com/article/10.1007/s10845-005-5516-y>.
- 43.[43] Damien Trentesaux. Distributed control of production systems. *Engineering Applications of Artificial Intelligence*, 22(7):971–978, October 2009. ISSN 09521976. doi: 10.1016/j.engappai.2009.05.001. URL <http://linkinghub.elsevier.com/retrieve/pii/S0952197609000797>.
- 44.[44] Rakesh Kumar Phanden, Ajai Jain, and Rajiv Verma. Integration of process planning and scheduling: a state-of-the-art review. *International Journal of Computer Integrated Manufacturing*, 24(6):517–534, June 2011. ISSN 0951-192X. doi: 10.1080/0951192X.2011.562543.
- 45.[45] Amit Kumar Gupta and Appa Iyer Sivakumar. Job shop scheduling techniques in semiconductor manufacturing. *The International Journal of Advanced Manufacturing Technology*, 27(11-12):1163–1169, November 2004. ISSN 0268-3768. doi: 10.1007/s00170-004-2296-z.
- 46.[46] CF Chien, CY Hsu, and CW Hsiao. Manufacturing intelligence to forecast and reduce semiconductor cycle time. *Journal of Intelligent Manufacturing*, pages 2281–2294, 2012.
- 47.[47] Chen-Fu Chien, Stephane Dauzere-Peres, Hans Ehm, John W. Fowler, Zhibin Jiang, Shekar Krishnaswamy, Lars Monch, and Reha Uzsoy. Modeling

- and analysis of semiconductor manufacturing in a shrinking world: Challenges and successes. In 2008 Winter Simulation Conference, pages 2093–2099. IEEE, December 2008. ISBN 978-1- 4244-2707-9.
- 48.[48] Haldun Aytug, Mark a. Lawley, Kenneth McKay, Shantha Mohan, and Reha Uzsoy. Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research*, 161(1):86–110, February 2005. ISSN 03772217.
- 49.[49] Huai Zhang, Zhibin Jiang, and Chengtao Guo. Simulation-based optimization of dispatching rules for semiconductor wafer fabrication system scheduling by the response surface methodology. *The International Journal of Advanced Manufacturing Technology*, 41(1-2):110–121, April 2008. ISSN 0268-3768/
- 50.Проектування мікропроцесорних систем керування: навчальний посібник/ І.Р. Козбур, П.О. Марущак, В.Р. Медвідь, В.Б. Савків, В.П. Пісьціо.– Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2022.–324с.
- 51.Я.І. Проць, В.Б. Савків,О.К. Шкодзінський, О.Л. Ляшук. Автоматизація виробничих процесів. Навчальний посібник для технічних спеціальностей вищих навчальних закладів. – Тернопіль: ТНТУ ім. І.Пулюя, 2011. – 344с.
- 52.Основи наукових досліджень і теорія експерименту : Навчальний посібник / укл. Ю. Б. Капаціла, П. О. Марущак, В. Б. Савків, О. П. Шовкун. Тернопіль : ФОП Паляниця В.А., 2023. 186 с.».
<http://elartu.tntu.edu.ua/handle/lib/40843>.
- 53.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>.
- 54.Методичний посібник для здобувачів освітнього ступеня «магістр» всіх

- спеціальностей денної та заочної (дистанційної) форм навчання «Безпека в надзвичайних ситуаціях» / В.С. Стручок –Тернопіль: ФОП Паляниця В. А., –156 с. <https://elartu.tntu.edu.ua/handle/lib/39196>.
55. Навчальний посібник «Техноекологія та цивільна безпека. Частина «Цивільна безпека»» / автор-укладач В.С. Стручок – Тернопіль: ФОП Паляниця В. А., – 156 с. <http://elartu.tntu.edu.ua/handle/lib/39424/>
56. Платформа .NET та мова програмування C# 8.0: навчальний посібник / Коноваленко І.В., Марущак П.О. – Тернопіль: ФОП Паляниця В. А., 2020 – 320 с. /Рекомендовано до друку Вченою радою Тернопільського національного технічного університету імені Івана Пулюя. Протокол № 10 від 20 жовтня 2020 року
57. Капаціла Ю.Б., Михайлишин Р.І., Савків В.Б., Методичні вказівки до виконання кваліфікаційної роботи бакалавра спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології». Тернопіль.: Видавництво ТНТУ. 2021. 40 с.
58. Савків В.Б., Капаціла Ю.Б., Михайлишин Р.І. Методичні вказівки до виконання кваліфікаційної роботи бакалавра спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології». Тернопіль.: Видавництво ТНТУ. 2021. 50 с. <https://elartu.tntu.edu.ua/handle/lib/35172>