

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет «Автоматики і комп'ютеризованих технологій»
(повна назва)

Кафедра «Кафедра комп'ютерно-інтегрованих технологій, автоматизації та
робототехніки»
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)
«Розроблення віртуального макету для дослідження гнучкої виробничої
дільниці»
(тема)

Виконав:
здобувач 4 року навчання,
групи АКТСІ-21-1
Богдан БЕРЕСТ
(власне ім'я, прізвище)

Спеціальність 151 «Автоматизація та
комп'ютерно інтегровані технології»
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма «Системна інженерія»
(повна назва освітньої програми)

Керівник ст. викл. Дмитро ГУРІН
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри _____

(підпис)

Ігор НЕВЛЮДОВ
(власне ім'я, прізвище)

2025 р.

Я, Берест Богдан Романович, як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

" 9 " червня 2025 р.



Богдан БЕРЕСТ

Харківський національний університет радіоелектроніки

Факультет _____ «Автоматики і комп'ютеризованих технологій»
Кафедра «Комп'ютерно-інтегрованих технологій, автоматизації та
робототехніки»
Рівень вищої освіти _____ перший (бакалаврський)
Спеціальність 151 «Автоматизація та комп'ютерно інтегровані технології»
(код і повна назва)
Тип програми _____ освітньо-професійна
Освітня програма _____ «Системна інженерія»
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Бересту Богдану Романовичу
(прізвище, ім'я, по батькові)

1. Тема роботи _____ «Розроблення віртуального макету для дослідження гнучкої виробничої ділянки»
затверджена наказом університету від _____ 19 _____ травня _____ 20 25 р. № 391ст
2. Термін подання здобувачем роботи до екзаменаційної комісії _____ 24 _____ червня _____ 20 25 р.
3. Вихідні дані до роботи _____ Дані про існуючі гнучкі виробничі ділянки; дані про системи віртуального моделювання; дані про Visual studio та OpenGL; дані про необхідні модулі управління
4. Перелік питань, що потрібно опрацювати в роботі Аналіз існуючих гнучких виробничих ділянок їх класифікація; Вибір та аналіз вибраних засобів програмування; Вибір необхідних технічних компонентів системи; Створення алгоритму роботи проекту; розроблення віртуального макету; Розрахунок стійкості системи управління гравіровально-фрезерного верстата MSF6090.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Графічний матеріал у вигляді презентації формату pptx кількість сторінок 12

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання кваліфікаційної роботи	28.04.2025	виконано
2	Вступ	30.04-05.05.2025	виконано
3	Аналіз існуючих гнучких виробничих дільниць їх класифікація та методи	06.05-10.05.2025	виконано
4	Розроблення алгоритму роботи програми	21.05-24.05.2025	виконано
5	Розроблення та дослідження віртуального макету гнучкої виробничої дільниці	24.05-03.06.2025	виконано
6	Розрахунок стійкості системи управління гравіровально-фрезерного верстата MSF6090	03.06-05.06.2025	виконано
7	Охорона праці	05.06-06.06.2025	виконано
8	Висновки	06.06-07.06.2025	виконано
9	Оформлення пояснювальної записки	08.06.2025	виконано
10	Подання роботи на нормоконтроль	09.06.2025	виконано
11	Подання на перевірку Інтернет-сервісом Strike Plagiarism	11.06.2025	виконано
12	Подання роботи на рецензію	11.06.2025	виконано
13	Подання роботи на підпис зав. кафедри	13.06.2025	виконано
14	Подання кваліфікаційної роботи в ЕК		

Дата видачі завдання 28 квітня 2025 р.

Здобувач _____
(підпис)

Богдан БЕРЕСТ
(власне ім'я, прізвище)

Керівник роботи _____
(підпис)

ст. викл. Дмитро ГУРІН
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 178 с., 1 табл., 41 рис., 6 дод., 28 джерел.

РОЗРОБКА, АВТОМАТИЗАЦІЯ, ГНУЧКА ВИРОБНИЧА ДІЛЬНИЦЯ,
ОБРОБКА ДЕТАЛЕЙ, ПІДГОТОВКА ОПЕРАТОРІВ.

Об'єктом розробки є процес подачі, транспортування та обробки деталей за допомогою гнучкої виробничої дільниці.

Предметом розробки є гнучка виробнича дільниця для подачі транспортування та обробки деталей.

Мета роботи полягає в підвищенні ефективності підготовки операторів гнучких виробничих дільниць шляхом розробки віртуального макета для моделювання процесів обробки та транспортування деталей.

У процесі виконання роботи було проведено аналіз наявних гнучких виробничих дільниць та автоматизації виробничих процесів. Дослідження охоплює класифікацію гнучких виробничих дільниць, методи та цілі їх впровадження у виробництво. На основі отриманих результатів створено експериментальний макет гнучкої виробничої дільниці.

Підсумком роботи став віртуальний макет гнучкої виробничої дільниці, призначений для навчання за допомогою моделювання процесів обробки та транспортування деталей.

Результати кваліфікаційної роботи відповідають переліку Цілей сталого розвитку:

- 4 Якісна освіта;
- 9 Інновації та інфраструктура.

THE ABSTRACT

Explanatory note: 178 p., 1 tables, 41 Fig., 6 add., 28 sources.

DEVELOPMENT, AUTOMATION, FLEXIBLE MANUFACTURING SYSTEM, PARTS PROCESSING, OPERATOR TRAINING.

The object of the development is the process of feeding, transporting and processing parts using a flexible manufacturing system.

The subject of the development is a flexible manufacturing system for feeding, transporting and processing parts.

The purpose of the work is to increase the efficiency of training operators of flexible production sites by developing a virtual model for modeling the processes of processing and transporting parts.

In the process of performing the work, an analysis of existing flexible manufacturing system and automation of production processes was conducted. The study covers the classification of flexible manufacturing system, methods and goals of their implementation in production. Based on the results obtained, an experimental model of a flexible manufacturing system was created.

The result of the work was a virtual model of a flexible manufacturing system, designed for training through simulation of processing and transportation of parts.

The results of the qualification work correspond to the list of Sustainable Development Goals:

- 4 Quality Education;
- 9 Innovation and Infrastructure.

ЗМІСТ

Перелік умовних скорочень	9
Вступ.....	10
1 Аналіз існуючих гнучких виробничих дільниць та віртуального моделювання	12
1.1 Сутність та значення гнучких виробничих дільниць.....	12
1.2 Основні складові гнучкої виробничої дільниці	13
1.3 Класифікація гнучких виробничих дільниць	14
1.4 Принципи функціонування гнучкої виробничої дільниці.....	15
1.5 Приклади реалізації гнучких виробничих дільниць у сучасній промисловості.....	20
1.6 Гнучка виробнича дільниця в контексті Індустрії 4.0	23
1.7 Вибір середовища розробки.....	25
1.8 OpenGL – графічна бібліотек для 3D-візуалізації	26
2 Проєктування віртуального макета гнучкої виробничої дільниці.....	29
2.1 Визначення вимог до функціональної структури макета	29
2.2 Вибір технічних компонентів	30
2.3 Алгоритми функціонування системи та взаємодії її компонентів ...	35
3 Розроблення віртуального макета	38
3.1 Розроблення функцій роботи програми	38
3.2 Дослідження роботи програми	51
4 Розрахунок стійкості системи управління гравіровально-фрезерного верстата MSF6090	58
4.1 Електрична частина, рівняння напруги	58
4.2 Механічна частина, рівняння руху ротора	59
4.3 Передавальні функції основних елементів сервоприводу.....	60
4.3.1 Передавальна функція контуру струму	60
4.3.2 Передавальна функція контуру швидкості	60

4.3.3 Передавальна функція контуру положення	61
4.3.4 Передавальна функція механічного навантаження.....	61
4.4 Момент інерції кульково-гвинтової пари.....	62
4.5 Приведений момент інерції лінійного навантаження	62
4.6 Розрахунок заданих параметрів для верстата MSF6090	63
4.7 П Визначення стійкості за критерієм Гурвіца.....	67
4.8 Охорона праці.....	70
Висновки	72
Перелік джерел посилань	74
Додаток А Апробація результатів роботи (теза).....	78
Додаток Б Апробація результатів роботи (стаття)	82
Додаток В Код головної форми «MyForm.h» Visual Studio	87
Додаток Г Код допоміжної форми «MachineQuantityDialog.h» Visual Studio	149
Додаток Д Код допоміжної форми «Settingsform.h» Visual Studio	155
Додаток Е Демонстраційний матеріал	177

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ГВД – гнучка виробнича діляниця;

ГВС – гнучка виробнича система;

ДБН – державні будівельні норми;

ДСТУ – державні стандарти України;

ЗІЗ – засоби індивідуального захисту;

КГП – кульково-гвинтова пара;

НС – надзвичайна ситуація;

ТАУ – теорія автоматично управління;

ЧПК – числовим програмним керуванням;

API – application programming interface.

ВСТУП

На сьогоднішній день під час швидкого розвитку промисловості, коли ринок постійно змінюється, а вимоги до якості, точності та швидкості виробництва збільшуються, особливої уваги набуває здатність підприємств швидко адаптуватися до нових задач. Гнучкі виробничі дільниці (ГВД) стають важливим елементом ефективного формування виробничих процесів, оскільки забезпечують швидке реагування на зміни попиту, можливість переналаштовувати обладнання в залежності від завдання та оптимальне використання ресурсів. Впровадження гнучкої виробничої дільниці сприяє підвищенню продуктивності, зменшенню витрат часу та ресурсів, а також зміцненню конкурентоспроможності підприємства.

Мета роботи – підвищення ефективності підготовки операторів гнучких виробничих дільниць шляхом розробки віртуального макета для моделювання процесів обробки та транспортування деталей.

Об'єкт розробки – процес подачі, транспортування та обробки деталей за допомогою гнучкої виробничої дільниці.

Предмет розробки – віртуальний макет гнучкої виробничої дільниці для подачі, транспортування та обробки деталей.

Для досягнення поставленої мети потрібно вирішити такі завдання:

- провести аналіз існуючих гнучких виробничих дільниць;
- визначити вимоги до функціональності та структури віртуального макета;
- підібрати відповідні технічні компоненти, які будуть моделюватися в віртуальному середовищі;
- розробити алгоритми функціонування системи та взаємодії між її компонентами;
- реалізувати програмну модель з використанням відповідного програмного забезпечення для візуалізації та керування;

– протестувати віртуальний макет на відповідність поставленим вимогам та оцінити його ефективність у контексті навчання операторів;

– оформити кваліфікаційну роботу згідно ДСТУ 3008:2015 [1], а також з методичними вказівками з підготовки й оформлення кваліфікаційної роботи здобувачами першого (бакалаврського) рівня вищої освіти спеціальності 151 «Системна інженерія» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» [2].

1 АНАЛІЗ ІСНУЮЧИХ ГНУЧКИХ ВИРОБНИЧИХ ДІЛЬНИЦЬ ТА ВІРТУАЛЬНОГО МОДЕЛЮВАННЯ

1.1. Сутність та значення гнучких виробничих дільниць

У сучасних умовах виробництва, що характеризуються динамічними умовами зміни ринку та попиту гнучкі виробничі дільниці набувають неабиякої популярності.

Гнучка виробнича дільниця – це сукупність технологічно взаємопов’язаного устаткування, керованого централізованою або децентралізованою системою управління, яке забезпечує виконання широкого спектру завдань та виробничих операцій з можливістю швидкої адаптації до нових завдань та попиту ринку без значних витрат часу та ресурсів на перелагодження [3].

Основою ГВД є її гнучкість, здатність системи змінювати свою конфігурацію, функціональність або режим роботи відповідно до виробничих потреб. Приклад гнучкої виробничої лінії можна побачити на рисунку 1.1.



Рисунок 1.1 – Горизонтальна обробна гнучка автоматизована лінія DZX–001

1.2 Основні складові гнучкої виробничої дільниці ГВД

Гнучка виробнича дільниця включає в себе різноманітні компоненти, що забезпечують її функціонал. До них належить механічне та технологічне обладнання, таке як роботи-маніпулятори, верстати ЧПК та системи транспортування. Важливою складовою частиною є системи автоматизованого управління, представлені програмованими логічними контролерами (PLC), SCADA-системами та вбудованими мікроконтролерами. Для ефективного переміщення матеріалів використовуються засоби транспортування та позиціонування, включаючи транспортні лінії, конвеєри, автоматичні візки та маніпулятори (рис. 1.2). Моніторинг та забезпечення безперебійної роботи забезпечують системами діагностики та контролю з їх датчиками та засобами моніторингу обладнання [5]. Крім того, невід'ємною частиною є програмне забезпечення, призначене для планування, керування та оптимізації виробництва, а також інтерфейси взаємодії з операторами та системи візуалізації технічного процесу.

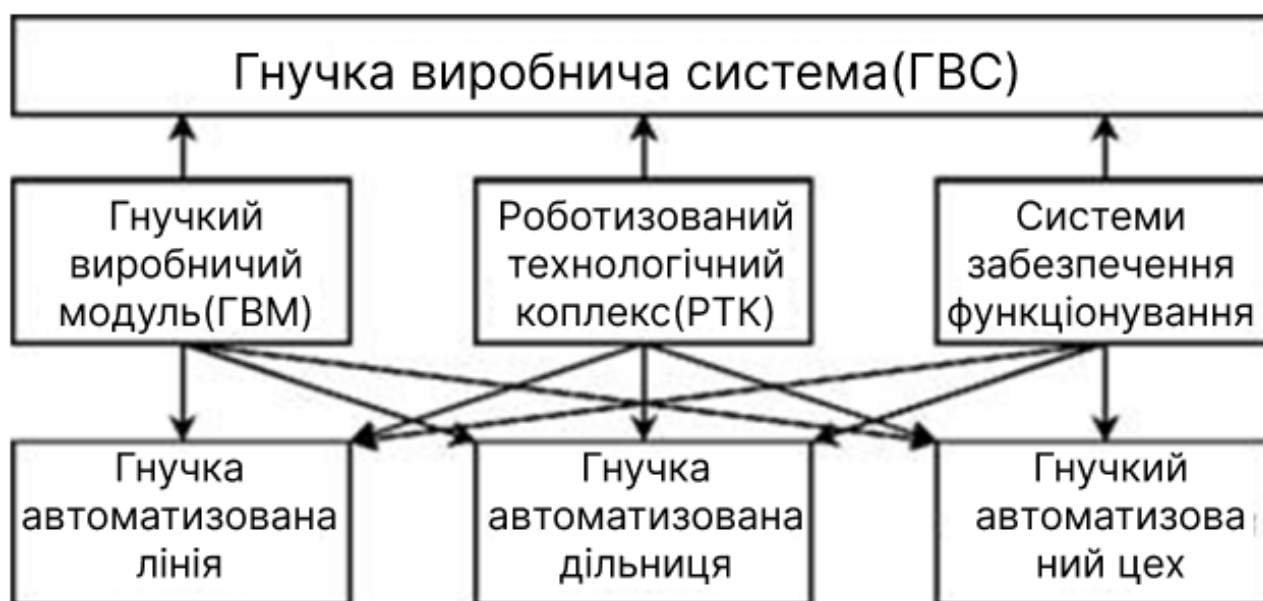


Рисунок 1.2 – Складові гнучкої виробничої системи (ГВС)

1.3 Класифікація гнучких виробничих дільниць

Гнучкі виробничі дільниці класифікують за різними критеріями, що враховують особливості їхньої побудови, рівень автоматизації, характер продукції та управління. За ступенем автоматизації виділяють напівавтоматизовані та повністю автоматизовані ГВД. У напівавтоматизованих системах частина операцій виконується автоматично, обробка на верстатах ЧПК, однак важливу роль відіграє людський фактор, оскільки оператор здійснює прямий контакт з об'єктами на виробництві і налагодження обладнання та контроль якості виконуються вручну, хоча транспортування та зберігання можуть бути частково автоматизовані [6]. Такі дільниці вигідніші за впровадженням, гнучкіші та придатні до навчальних цілей, але мають низьку продуктивність, вищу ймовірність браку та залежать від кваліфікації оператора. Натомість повністю автоматизовані системи – це високотехнологічні комплекси, де всі виробничі процеси здійснюються без участі оператора. Це досягається завдяки повній автоматизації обробки, транспортування та контролю якості, інтегруванню ERP/MES-системами, а також використанню різних видів роботів та систем комп'ютерного зору. Такі дільниці вирізняються високою продуктивністю та точністю роботи, проте їхнє впровадження є дорогим, обслуговування є складним і займає багато часу.

За типом продукції ГВД поділяються на дільниці одиничного, дрібносерійного та серійного виробництва.

Одиничне виробництво передбачає виготовлення унікальних або нестандартних виробів в малій кількості, наприклад дослідних зразків тощо. Для нього характерне часте переналаштування обладнання, висока гнучкість і універсальність, а також використання універсальних верстатів та роботів.

Дрібносерійне виробництво орієнтоване на виготовлення відносно невеликих партій однакових виробів або деталей. Тут спостерігається помірна частота переналагоджень системи, оптимальне поєднання гнучкості та продуктивності, а також можливість швидкого переходу між різними серіями.

Серійне виробництво стосується масового виробництва однотипних виробів, таких як деталі двигуна тощо. Його відмінності, це мала потреба в переналаштуванні, висока стандартизація операцій і перевага автоматизованих рішень з мінімальним людським втручанням.

Щодо типу управління, ГВД діляться на централізовані, децентралізовані та гібридні системи.

У централізованих системах усі функції управління зосереджені в одному головному контролері або центральному комп'ютері, що забезпечує чітку ієрархію управління та високу злагодженість дій, але робить систему вразливою до збоїв.

Децентралізовані системи мають високу автономність, де кожен модуль або вузол здатен самостійно виконувати певні функції, взаємодіючи на основі обміну даними. Це забезпечує гнучкість, відмовостійкість та масштабованість, але ускладнює регулювання та синхронізацію.

Гібридні системи поєднують переваги обох підходів: частина функцій виконується централізовано, тоді як виконання операцій здійснюється на місцях. Це дозволяє гнучко розмежовувати обов'язки, надавати модулям локальну автономність та інтегруватися з корпоративними ІТ-системами, забезпечуючи оптимальний баланс ефективності та адаптивності, хоча вимагає складної розробки та синхронізації даних.

Вибір конкретного типу ГВД залежить від особливостей виробництва, бюджету, цілей підприємства та рівня технологічної зрілості, тому для досягнення максимальної ефективності слід ретельно аналізувати вимоги до продукції, необхідний рівень автоматизації та доцільну структуру управління.

1.4 Принципи функціонування гнучкої виробничої ділянки

Гнучка виробнича ділянка є складною техніко-організаційною системою, яка базується на низці ключових принципів. Ці принципи забезпечують її

ефективну роботу в умовах динамічних змін попиту, варіативність продукції та потреби в оперативному переналаштуванні виробництва (рис. 1.3).

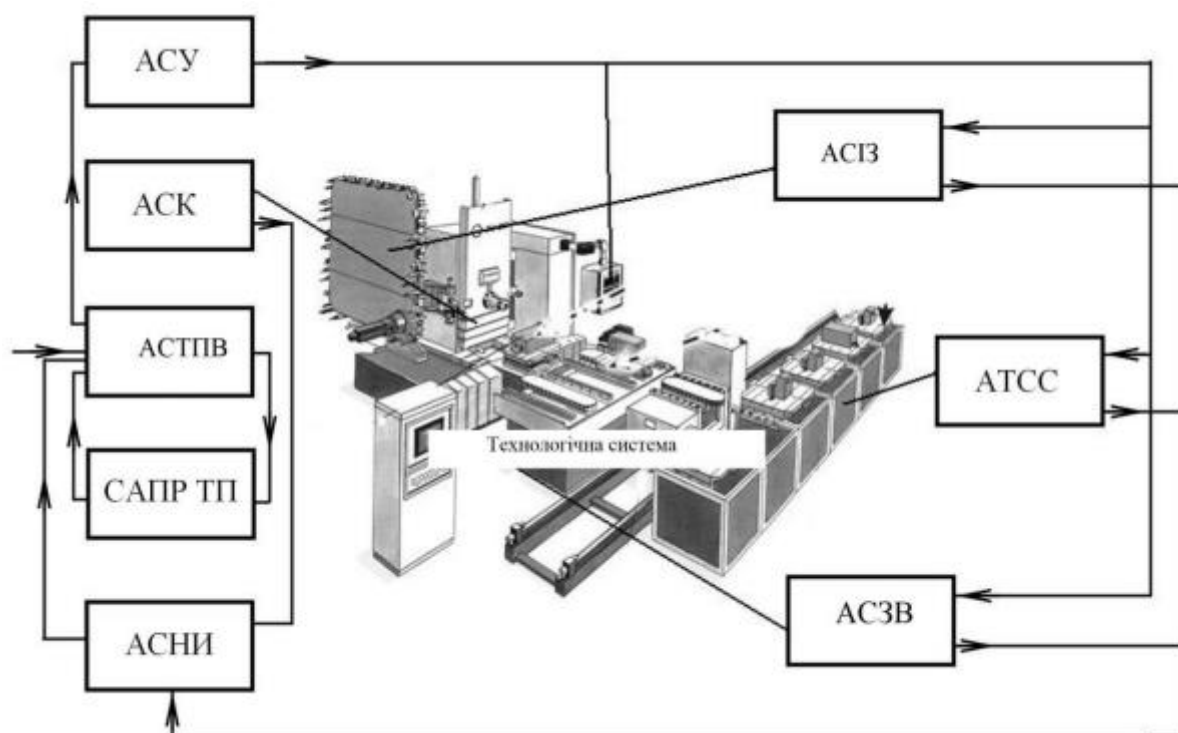


Рисунок 1.3 – Структура ГВС

До основних принципів функціонування НВД належать модульність, автономність, інтегрованість, адаптивність та взаємозамінність.

Принцип модульності передбачає побудову ГВД з окремих функціональних блоків, які можуть легко замінюватися або конфігуруватися відповідно до вимог виробничого процесу. Такий підхід забезпечує високу гнучкість у зміні структури виробництва, а також спрощує технічне обслуговування та модернізацію обладнання. У разі потреби модуль можна вилучити з системи або змінити іншим без критичного впливу на всю ділянку, що значно підвищує живучість і масштабованість ГВД [7].

Автономність полягає в тому, що кожен модуль або виробничий елемент ГВД може виконувати задані функції самостійно, без постійного втручання з боку центральної системи управління. Наприклад, робот-маніпулятор може самостійно визначати наявність заготовки, захоплювати її, транспортувати та

повертатися до початкового положення, виконуючи при цьому вбудовані функції дій. Це значно підвищує ефективність і надійність виробничих процесів, знижуючи залежність від централізованого управління та зменшуючи ризики зупинки всієї системи через збій в одному компоненті [8].

Принцип інтегрованості означає цілісність усіх підсистем ГВД в єдиному інформаційному, технічному та логістичному просторі. Застосування інтегрованих рішень дозволяє узгоджено працювати різним підсистемам – обробці, транспортуванню, управлінню, контролю якості, зберіганню тощо [9]. Наприклад, використання єдиної цифрової платформи або промислової мережі забезпечує безперервний обмін даними між елементами системи, що дозволяє відслідковувати стан обладнання, управляти виробничими процесами в режимі реального часу та автоматично корегувати дії у разі змін у виробничому середовищі. Такий підхід також полегшує інтеграцію ГВД у загальну цифрову інфраструктуру підприємства.

Адаптивність – це здатність ГВД оперативно реагувати на зміну виробничих умов, наприклад, зміну порядку робіт, кількості продукції або обсягу замовлення. Завдяки використанню автоматизованих систем управління та інтелектуальних алгоритмі планування, ГВД може автоматично переналаштовуватись відповідно до нових вимог без значного зниження продуктивності [10]. Така властивість особливо важлива в умовах індивідуального або змінного виробництва, де традиційні лінії не можуть забезпечити належного рівня ефективності. Адаптивність також дозволяє підвищити рівень обслуговування клієнтів за рахунок термінів виготовлення і гнучкого реагування на запити ринку.

Принцип взаємозамінності передбачає використання стандартизованих компонентів і технологічного оснащення, які можуть бути легко замінені в разі необхідності. Це спрощує технічне обслуговування. Скорочує час простоїв у разі виходу з ладу окремих елементів, а також полегшує модернізацію ГВД. Завдяки цьому підприємство має можливість підтримувати актуальність технічного забезпечення без повного демонтажу або масштабних реконструкцій.

Впровадження ГВД є важливим кроком на шляху модернізації промислових підприємств. Завдяки інтеграції сучасних автоматизованих рішень, використанню роботизованих систем та цифрових технологій, ГВД дозволяють значно покращити ефективність виробничих процесів.

Автоматизація технологічних процесів, включаючи завантаження/розвантаження деталей, обробку, транспортування та контроль якості, дозволяє значно скоротити час циклу виготовлення продукції. Роботизовані системи працюють без перерв, що забезпечує цілодобовий режим роботи без втрат продуктивності через втому або людський фактор.

Однією з ключових переваг ГВД є можливість швидкого та ефективного переналаштування на новий тип продукції. Завдяки використанню універсальних верстатів з числовим програмним керування (ЧПК), автоматичних систем налаштування та гнучкого програмного забезпечення, час простою обладнання при переході на нове завдання значно зменшується.

ГВД дозволяють виготовляти широкий спектр виробів без необхідності повної реконфігурації виробничої лінії. Це особливо важливо для підприємств, які працюють з індивідуальними або малосерійними замовленнями, де часто змінюється конструкція виробу, матеріали або послідовність технологічних операцій.

Хоча впровадження ГВД пов'язане з початковими капіталовкладеннями, в подальшому це дає змогу оптимізувати витрати на технічне обслуговування обладнання завдяки використанню модульних рішень, самодіагностики, попереджувального ремонту на основі даних датчиків та аналітики. Крім того, автоматизація зменшує потребу у великій кількості операційного персоналу.

Системи автоматичного контролю якості та високоточне обладнання гарантують стабільну якість та зменшення браку. Також зменшується вплив людського фактору, що часто є причиною помилок.

ГВД можуть бути повністю інтегровані з сучасними цифровими платформами, що базуються на технологіях Інтернету речей (IoT), цифрових двійників, машинного навчання та штучного інтелекту. Це відкриває нові

можливості для моніторингу, прогнозування технічного стану обладнання, оптимізації логістики, планування виробництва в реальному часі та персоналізованого виготовлення продукції.

ГВД легко масштабуються шляхом додавання нових модулів або функціональних блоків. Це дозволяє підприємствам поступово нарощувати виробничі потужності без повної перебудови.

Незважаючи на численні переваги, впровадження ГВД супроводжується низкою викликів та обмежень, які необхідно враховувати під час проєктування і реалізації таких систем, особливо на підприємствах малого та середнього бізнесу.

Одним з головних бар'єрів є значні капіталовкладення, необхідні для придбання обладнання, програмного забезпечення, побудови інфраструктури автоматизації та навчання персоналу. Особливо суттєвими ці витрати є на етапі початкового впровадження, коли ще не досягнуто ефекту масштабування та економії на обсязі.

Експлуатація та обслуговування ГВД вимагає персоналу з глибокими знаннями в галузі автоматизації, програмування, електроніки та обробки даних [11]. Недостатній рівень кваліфікації може призвести до неправильного налаштування обладнання, зниження ефективності роботи або навіть до пошкодження техніки.

Зростання рівня автоматизації й варіативності виробництва створює додаткові труднощі в управлінні виробничими процесами. Планування, контроль, облік і аналіз вимагають використання складних інформаційних систем.

Для забезпечення стабільної роботи ГВД необхідно створити надійну локальну мережу, організувати системи безперебійного живлення, захисту даних та резервного копіювання. Відсутність відповідної інфраструктури може стати причиною частих збоїв або втрати даних.

Через високі початкові інвестиції повна окупність проєкту може розтягнутися на кілька років, що є суттєвим ризиком для підприємств з

обмеженим бюджетом. Особливо це актуально для малого бізнесу, де не має можливості швидко отримати віддачу від вкладень.

Гнучкі виробничі дільниці мають значний потенціал для підвищення ефективності, якості та конкурентоспроможності сучасного виробництва. Водночас, для успішного впровадження таких рішень необхідно комплексно підходити до оцінки ресурсів підприємства, технічної готовності персоналу та економічної доцільності інвестицій. Баланс між потенційними вигодами та можливими труднощами повинен визначати доцільність використання ГВД у кожному конкретному випадку.

1.5 Приклади реалізації гнучких виробничих дільниць у сучасній промисловості

ГВД дедалі активніше застосовуються на підприємствах різних галузей промисловості, зокрема в автомобілебудуванні, електроніці, авіаційній промисловості, а також у сфері медичного обладнання. Їхня популярність пояснюється високою ефективністю, універсальністю та здатністю адаптуватися до змін у попиті чи специфікаціях продукції без значних витрат часу та ресурсів.

Одними з піонерів впровадження ГВД є компанії Toyota, BMW, Ford, Tesla ті інші світові автовиробники. На заводах Toyota реалізована концепція гнучкої виробничої системи, яка дозволяє на одній виробничій лінії виготовляти різні моделі автомобілів з мінімальним переналаштуванням. Завдяки інтеграції роботів-маніпуляторів, автономних візків та цифрового управління, компанія досягає високої продуктивності при збереженні гнучкості виробництва.

Tesla активно застосовує принципи гнучкого виробництва на своїх «гігафабриках», де автоматизовані лінії адаптуються до різних типів акумуляторних блоків і компонентів для електромобілів. Така адаптивність дозволяє компанії оперативно змінювати виробничі пріоритети в залежності від ринкової ситуації (рис. 1.5).



Рисунок 1.5 – Гнучка виробнича дільниця для збирання автомобілів

Компанії Siemens та Bosch впровадили ГВД на своїх виробничих лініях, зокрема в Німеччині. На заводі Siemens в місті Амберг реалізовано повністю автоматизовану дільницю, фабрика виготовляє програмовані логічні контролери. Ця дільниця є прикладом цифрового виробництва.

У Bosch діє концепція «інтелектуальної фабрики», де модульні виробничі станції з'єднані в єдину систему з використанням хмарних технологій, IoT-сенсорики та AI-аналізу даних. Виробництво може миттєво адаптуватися до нових замовлень, змінюючи параметри без зупинки лінії.

У компанії Airbus використовуються адаптивні виробничі системи, які поєднують людину та роботів у гнучких дільницях для збирання фюзеляжів, крил та інших великих елементів (рис. 1.6). У виробництві широко застосовуються автоматизовані платформи з точним позиціонування, які керуються централізованою системою в реальному часі.



Рисунок 1.6 – Завод Airbus в США

Медичне виробництво, де особливо важливі точність, гігієна та персоналізація, також активно впроваджує ГВД. Наприклад компанії GE Healthcare і Medtronic використовують модульні виробничі системи для виготовлення різноманітних медичних пристроїв, де кожен модуль налаштовується під конкретний продукт або серію.

У дрібносерійних або індивідуальних виробництвах все більше поширення отримують модульні мікрофабрики, які поєднують універсальні обробні центри, роботів, транспортні системи та інтелектуальне програмне забезпечення. Прикладом є платформи типу FlexFactory від компанії Festo, які дозволяють швидко змінювати конфігурацію виробництва залежно від запитів клієнтів.

Такі рішення дають змогу малим та середнім підприємствам оперативно запускати виробництво без необхідності будівництва великих заводів і значних капіталовкладень, забезпечуючи при цьому високу точність і повторюваність.

1.6 Гнучка виробнича дільниця в контексті Індустрії 4.0

ГВД є ключовим елементом реалізації концепції Індустрії 4.0, яка передбачає створення повністю цифровізованого, автоматизованого та інтегрованого середовища. Індустрія 4.0 ставить за мету побудову розумного виробництва, де фізичні пристрої, цифрові технології та алгоритми штучного інтелекту об'єднані в єдину адаптивну систему.

У межах індустрії 4.0 ГВД функціонують як кіберфізичні системи. Кожен модуль дільниці має сенсори, які передають інформацію в хмарні сервіси або локальні аналітичні системи для подальшої обробки.

Сучасні ГВД мають використовувати алгоритми машинного навчання для оптимізації виробничих графіків, передбачення поломок устаткування, автоматичного переналаштування обладнання та покращення логістики. Це сприяє підвищенню ефективності та зниженню витрат.

Інтернет речей (IoT) відіграє важливу роль у забезпеченні зв'язку між модулями ГВД. Дані від усіх сенсорів та виконавчих пристроїв збираються в єдину інформаційну систему, що дозволяє проводити моніторинг у реальному часі, дистанційне управління, а також застосовувати моделі цифрових двійників.

Індустрія 4.0 орієнтується на персоналізоване масове виробництво, де кожен виріб може адаптуватися під індивідуальні вимоги замовника. ГВД у цьому контексті дозволяє швидко адаптувати виробництво під нові технічні умови без значних витрат часу та ресурсів.

Один з принципів Індустрії 4.0 полягає в тому, що виробничі системи мають здатність до самоорганізації – тобто окремі компоненти ГВД можуть узгоджувати свою роботу без втручання центрального керування. Наприклад, модулі обробки самі вибирають оптимальний маршрут проходження деталі в залежності від завантаження інших модулів.

Інтеграція гнучких виробничих дільниць (ГВД) в навчальний процес та дослідницьку діяльність набуває все більшої популярності завдяки можливості використовувати віртуальні моделі для симуляції, тренувань та досліджень.

Віртуалізація виробничих систем надає ряд переваг, які значно полегшують процес навчання студентів, а також покращують ефективність досліджень у цій галузі. Використання таких технологій, як цифрові тренажери, віртуальні лабораторії та імітаційні середовища, дозволяє значно покращити якість навчання, а також знизити витрати на практичні заняття.

Сучасні навчальні заклади, що готують фахівців для роботи з гнучкими виробничими лініями, активно використовують цифрові тренажери та віртуальні лабораторії. Це програмні засоби, які дозволяють імітувати роботу реальних виробничих систем, зокрема роботизованих ліній, автоматизованих складських систем та іншу складові ГВД. Такі тренажери дають можливість студентам навчатися принципам роботи з ГВД без реального ризику пошкодження обладнання, опанувати методи програмування роботів і систем управління використовуючи віртуальні моделі, студенти можуть безпосередньо взаємодіяти з програмними інтерфейсами, відпрацьовувати програмування та налаштування роботів і автоматизованих систем. Студенти можуть відпрацьовувати різні сценарії виробництва в умовах змінного навантаження, дефіциту ресурсів або зміни характеристик продукції, що дозволяє оптимізувати роботу виробничої дільниці в різних ситуаціях.

Ці тренажери можуть мати різні рівні складності і дозволяють здійснювати навчання на різних етапах, від загального ознайомлення до детального налаштування складних систем.

Для проведення досліджень і тестування різних концепцій і варіантів виробничих ліній активно використовуються імітаційні середовища, такі як Unity, RoboDK, Factory I/O та інші платформи, які дозволяють моделювати гнучкі виробничі дільниці та їх компоненти [15].

Віртуальні моделі ГВД є корисними не лише для студентів, але й для навчання кваліфікованих операторів та інженерів, які працюють безпосередньо з автоматизованими виробничими системами.

Одним із прикладів успішного впровадження віртуальних моделей ГВД є платформа Unity, яка використовується для моделювання та симуляції

автоматизованих виробничих процесів. В цій програмі можна створювати та налаштовувати роботизовані лінії та тестувати їх ефективність.

RoboDK – ще одна популярна платформа, що дозволяє модулювати роботи роботів та автоматизованих систем. За її допомогою можна практикуватися в програмуванні роботів для різних виробничих завдань [15].

З розвитком технологій віртуалізації та штучного інтелекту, а також з удосконаленням моделей імітаційних середовищ, можливості для навчання та дослідження з використанням ГВД будуть тільки зростати. В результаті впроваджень таких технологій у навчання та дослідження сприятиме формуванню нових поколінь висококваліфікаційних спеціалістів, готових до роботи в умовах швидкозмінюваних виробничих технологій.

1.7 Вибір середовища розробки Visual Studio

Для реалізації віртуального макета гнучкої виробничої дільниці у кваліфікаційній роботі було обрано інтегроване середовище розробки Visual Studio у поєднанні з графічною бібліотекою OpenGL. Такий вибір є результатом аналізу ряду факторів, серед яких: функціональні можливості, сумісність із мовою програмування C++, наявність інструментів для 3D-візуалізації, підтримка розширень, зручність налагодження та тестування, а також постійна підтримка продукту.

Visual Studio – це потужне середовище розробки, що робить створення програмних проєктів будь-якої складності комфортним та зрозумілим. Visual Studio підтримує мову програмування C++, одну із базових мов для роботи з OpenGL, має розвинені функції налагодження, профілювання та управління проєктом [16].

Серед ключових переваг Visual Studio є підтримка різних мов програмування та компіляторів, включно зі стандартним MSVC та багато інших. Інструмент також містить вбудовані засоби налагодження, що допомагають ефективно знаходити помилки в компонентах коду. Крім того Visual Studio

підтримує інтеграцію з системами контролю версій, що дозволяє налаштовувати середовище за допомогою шаблонів проєкту, менеджерів пакетів та сторонніх бібліотек, а також пропонує гнучкі можливості розширення проєкту завдяки інтеграції з GUI-фреймворками, різними рушіями чи системами штучного інтелекту.

Для виконання роботи Visual Studio було обрано також через можливість зручно поєднувати графічну частину з логікою поведінки об'єктів, створювати багатофайлову структуру проєкту, автоматично збирати проєкт із залежностями, а також здійснювати модульне тестування.

1.8 OpenGL – графічна бібліотек для 3D-візуалізації

OpenGL – це універсальна бібліотека яка дозволяє створювати як двовимірні так і тривимірні об'єкти. Його основна перевага в високій продуктивності, досягнутій завдяки прямій взаємодії з відеокартою. Це робить OpenGL ідеальним інструментом для завдань пов'язаних із візуалізацією виробничого середовища, оскільки він дозволяє будувати деталізовані 3D-сцени з точним розміщенням об'єктів у просторі, створювати реалістичні візуальні ефекти завдяки підтримці камери та освітлення, деталізувати поверхні віртуального обладнання за рахунок роботи з текстурами та матеріалами, ефективно керувати трансформаціями об'єктів та гнучко налаштовувати вигляд сцени завдяки підтримці шейдерів.

У контексті виробничих процесів, OpenGL функціонує як графічний пристрій, що візуалізує окремі компоненти гнучкої виробничої дільниці, такі як транспортні лінії, деталі, різного типу верстати та інше обладнання [17].

Поєднання Visual Studio та OpenGL є цілком виправданим завдяки його функціональності, гнучкості та відкритості. Ця зв'язка дозволяє створити власну структуру віртуального макета, не прив'язуючись до конкретної платформи. В цьому поєднанні можна отримати повний контроль над логікою руху об'єктів, маючи можливість програмувати власні анімації, алгоритми переміщення та

взаємодії. Крім того можна легко реалізувати інтерактивну взаємодію з користувачем через клавіатуру, мишу чи інші пристрої введення. Важливою перевагою є можливість розширювати проєкт у майбутньому.

Такий підхід також сприяє глибокому розумінню принципів комп'ютерної графіки на низькому рівні, що є особливо цінним для дослідників і розробників у сфері автоматизації та вбудованих систем.

Існують інші рішення для реалізації подібних проєктів такі як, ігрові рушії Unity або Unreal Engine. Проте вони є вискорівневими і не завжди дозволяють гнучко керувати внутрішніми процесами, особливо в частині обробки нестандартних алгоритмів управління або специфічного промислового сценарію. Крім того, такі рушії вимагають глибшої графічної підготовки, часто мають обмеження ліцензування та велику залежність від внутрішньої архітектури рушія.

У той час як Visual Studio з OpenGL забезпечує повну незалежність, відкритість коду та можливість поступового ускладнення проєкту, починаючи з базової геометрії й закінчуючи складними анімаціями та логікою.

Рішення використати Visual Studio з OpenGL для реалізації віртуального макету гнучкої виробничої ділянки (ГВД) ґрунтується на ряді технічних та педагогічних переваг, що обумовлюють доцільність такого підходу як для навчання так і для дослідницької діяльності.

Дуже важливо те, що у Visual Studio можлива інтеграція сторонніх бібліотек і фреймворків, таких як GLUT, GLFW, GLEW або FreeGLUT для керування вінками, вводу з клавіатури та миші, підключення анімацій тощо. Це дозволяє створювати повноцінні симуляції, які є гнучкими з точки зору архітектури та функціоналу.

Також, самостійне програмування в OpenGL формує навички роботи з матрицями трансформацій, 3D-координатами, перспективою, освітленням, а також взаємодією з апаратними ресурсами.

Visual Studio та OpenGL – це не просто засіб для візуалізації, це середовище розвитку, яке дозволяє реалізувати будь-яку ідею від найпростішої до

найскладнішої без прив'язки до готових шаблонів або закритих фреймворків. Саме тому цей підхід забезпечує оптимальну навчальну цінність, технічну гнучкість і повну контрольованість процесу розробки.

2 ПРОЄКТУВАННЯ ВІРТУАЛЬНОГО МАКЕТА ГНУЧКОЇ ВИРОБНИЧОЇ ДІЛЬНИЦІ

2.1 Визначення вимог до функціональної структури макета

Проєктування віртуального макета гнучкої виробничої дільниці (ГВД) передбачає ретельне визначення вимог до його функціональної структури, які повинні забезпечити адекватну імітацію реальних виробничих процесів транспортування та обробки деталей. Визначення таких вимог ґрунтується на аналізі сучасних автоматизованих виробничих систем та специфіки їх функціонування.

Однією з базових характеристик майбутньої системи має бути модульність, це означає, що макет повинен складатися з незалежних, але взаємодіючих між собою функціональних блоків, таких як підсистема транспортування, модулі обробки, система керування, а також інтерфейс взаємодії з користувачем. Така архітектура дозволить спростити процес розробки, налаштування, оновлення та тестування окремих компонентів, не порушуючи при цьому цілісності системи.

Важливою вимогою до функціональної структури є забезпечення інтерактивності. Макет має підтримувати динамічну взаємодію з користувачем у режимі реального часу. Це включає можливість спостереження за поведінкою елементів системи, керування процесами та отримання зворотного зв'язку від логіки керування. Забезпечення інтерактивності є критично важливим у контексті використання макета як навчального інструменту для підготовки операторів ГВД.

Не менш важливою є реалізація повноцінної візуалізації виробничих процесів. Макет має відображати всі елементи системи у тривимірному середовищі з використанням відповідних графічних засобів. Особливу увагу слід приділити візуалізації роботи транспортування деталей, реакції системи на дії користувача та внутрішні події. Реалістична графіка сприятиме кращому

сприйняттю навчального матеріалу, що значною мірою підвищить ефективність підготовки операторів.

Структура віртуального макета повинна бути масштабованою, це означає, що її архітектура повинна допускати можливість подальшого розширення, наприклад, шляхом додавання нових виробничих модулів. Така гнучкість забезпечуватиметься відповідною програмною реалізацією, що ґрунтується на принципах об'єктно-орієнтованого підходу, розділення логіки системи та її графічного відображення.

Ключовим елементом є також інтеграція логіки керування, віртуальний макет повинен не тільки відображати виробничі процеси, але й моделювати процеси прийняття рішень, що відбуваються в реальній ГВД. Логіка керування повинна реагувати на змін станів об'єктів, ініціювати відповідні дії з боку системи, координувати роботу її компонентів та забезпечувати відлагоджену взаємодію між ними. Реалізація таких алгоритмів дозволить ефективно імітувати типові сценарії роботи та протестувати їх поведінку у змінних умовах.

Таким чином, формування вимог до функціональної структури віртуального макета є критично важливим елементом для процесу розробки. Дуже важливо врахувати модульність, інтерактивність, візуалізацію, масштабованість та логіку керування, що створюють основу для формування ефективного інструменту навчання, що дозволяє моделювати реальні виробничі процеси в безпечному та контрольованому середовищі.

2.2 Вибір технічних компонентів

Розроблення віртуального макета ГВД потребує ретельного добору технічних компонентів, які будуть імітуватися у віртуальному середовищі з метою максимально точного відтворення логіки та динаміки функціонування реальних виробничих систем. Вибір цих компонентів здійснюється на основі аналізу сучасних підходів до автоматизації виробничих систем, з урахуванням їх

поширеності у промисловості, функціонального призначення та можливостей моделювання.

Ключовим елементом віртуальної моделі гнучкої виробничої ділянки є однобалочний мостовий кран, який виконує функції захоплення, переміщення ті точного позиціонування деталі у межах робочої зони. Цей тип підйомного обладнання широко застосовується у сучасних промислових підприємствах для механізації та автоматизації процесів обробки, складання, транспортування і зберігання вантажів. У реальному виробництві ці крани застосовуються в механічній обробці, на складах, у рентних цехах, на ділянках складання та багатьох інших технічних зонах.

Однобалочний мостовий кран виготовляється відповідно до стандартів GB3811-2008 та JB/T1306-2008 [18] та використовується в комплекті з електричними телями типів CD1, MD1, WH164 або електричними ланцюговими телями, що виступають у ролі основного підйомного механізму. Живлення такого обладнання забезпечується трифазним змінним струмом з частотою 50 Гц або 60 Гц і напругою в межах 220 – 660 В. Це забезпечує сумісність з різними інфраструктурами підприємств (рис. 2.1).



Рисунок 2.1 – Однобалочний мостовий кран

Робочий механізм крана функціонує в режимі незалежного переміщення, використовуючи конічний роторний двигун з плавним пуском для забезпечення контрольованого руху та надійного гальмування. Передача обертового

моменту здійснюється через зубчасту передачу з одним відкриванням та двома замиканнями, що забезпечує ефективну і точну роботу механізму. Залежно від умов експлуатації можливе використання різних режимів керування: наземного, дистанційного або керування з кабіни, що підвищує адаптивність системи до конкретних виробничих потреб.

Конструктивно кран складається з однієї мостової балки, кінцевих візків, підкранових балок та рейок, по яких пересувається підйомник з вантажем. Така конфігурація забезпечує високу гнучкість у розміщенні та експлуатації, мінімізуючи займану площу у цеху завдяки монтажу в верхній частині виробничого простору. Це дозволяє максимально використовувати робочий об'єм приміщення без перешкод для інших технологічних процесів.

У контексті віртуального моделювання однобалочний кран виступає не лише як механічна конструкція, але й як складна багатоланкова система з кількома ступенями свободи, що імітує поведінку реального промислового обладнання. Завдяки цьому стає можливим створення навчальних і тестових сценаріїв для опрацювання різноманітних технологічних операцій від точного позиціонування деталей до моделювання аварійних ситуацій або зміни режимів експлуатації. Гнучкість конфігурації дозволяє адаптувати систему під конкретні умови користувача, змінюючи параметри крану у відповідності до особливостей виробництва, об'єктів обробки та режимів навантаження.

Таким чином, однобалочний мостовий кран у складі віртуального макета виступає не тільки як функціональний модуль транспортування, але й як центральний як центральний елемент, що забезпечує достовірне і динамічне моделювання виробничих процесів, сприяючи підвищенню якості підготовки операторів та інженерів у сфері автоматизованих систем

Ще одним важливим функціональним компонентом у складі віртуального макета ГВД є модулі обробки деталей, які симулюють виконання ключових технологічних операцій таких як – фрезування, сверління, шліфування та інші. У контексті віртуального макету ці елементи реалізовані у вигляді анімованих

обробних верстатів, що включають в себе логіку запуску, зміну станів під час процесу обробки та візуальну індикацію завершення технологічного циклу.

Для моделювання процесів обробки деталей у системі враховано параметри властиві фрезерному верстату ЧПК MSF6090 [19], який у реальному виробництві є прикладом компактного, універсального та високопродуктивного обладнання. Дана модель оснащена серводвигунами замість традиційних крокових моторів, що забезпечує значне підвищення точності позиціонування та плавності руху інструмента (рис. 2.2).



Рисунок 2.2 – Фрезерний верстат з ЧПК MSF6090

Фрезерний верстат типу MSF6090 з числовим програмним керуванням орієнтований на виготовлення дрібних деталей, а також активно застосовується у рекламній сфері, при створенні 3D-форм, штампів, а також обробки пластиків,

дерева та металів. У віртуальному макеті він відтворює як анімацію обробки так і технічну логіку, включаючи моделювання параметрів різання, затримки, переходів між режимами, що дозволяє користувачеві спостерігати та аналізувати повний цикл обробки.

Інтеграція такого функціонального модуля у віртуальну модель не лише підвищить реалізм симуляції, але й створює умови для тренування операторів ЧПК обладнання, відпрацювання алгоритмів взаємодії із системами автоматизованого керування, а також для оцінки ефективності виробничих сценаріїв у безпечному, контрольованому середовищі.

Ще одним важливим модулем у складі віртуального макета ГВД є токарна обробка, яку у системі представляє універсальний токарний верстат CORMAK TORNADO 360 [20]. Цей верстат є зразком високоякісного металообробного обладнання, що поєднує гнучкість, точність, надійність і зручність експлуатації. У реальних виробничих умовах його застосовують для обробки внутрішніх і зовнішніх поверхонь, свердління, нарізання метричної та дюймової різьби, а також виконання ряду інших токарних операцій (рис. 2.3).



Рисунок 2.3 – Універсальний токарний верстат CORMAK TORNADO 360

Інтеграція моделі CORMAK TORNADO 360 віртуального макета ГВД забезпечує повноцінне моделювання токарного циклу як частини загального

технологічного процесу, дозволяючи сформувати у користувачів розуміння послідовності дій, логіки взаємодії з елементами системи та оптимізації виробничих рішень у контексті автоматизованого середовища.

Усі перелічені компоненти координуються через єдину систему керування, яка моделюється за допомогою віртуального контролера. Він відповідає за виконання алгоритмів взаємодії між компонентами, обробку сигналів від оператора, а також забезпечення зворотного зв'язку. Реалізація логіки керування є важливою частиною для досягнення цілісного функціонування системи та її подальшої інтеграції у навчальний процес.

Обрані технічні компоненти дозволяють створити віртуальне середовище, яке відповідає структурі і логіці реальної ГВД, забезпечуючи при цьому навчальну цінність, гнучкість налаштування та можливість подальшої модернізації системи.

2.3 Алгоритми функціонування системи та взаємодії її компонентів

Алгоритмічне забезпечення віртуального макета гнучкої виробничої дільниці (ГВД) є фундаментом, що визначає логіку функціонування системи. В умовах моделювання, де реальні фізичні процеси замінені на програмні симуляції, важливим завданням стає розроблення чітко структурованих алгоритмів, які відображають логіку виробничих операцій, порядок дій виконавчих механізмів та взаємодію між компонентами.

Важливим елементом алгоритму є цикл управління, що ініціює виконання всіх операцій відповідно до змін у станах системи. В основі цього циклу лежить реакція на команди, які надходять від оператора. Завдяки цьому систему може функціонувати в подієво-орієнтованому режимі, що дозволяє наблизити її логіку до реального виробничого середовища, де рішення приймаються динамічно на основі поточної ситуації.

Однобалочний мостовий кран функціонує за допомогою рухових алгоритмів, які визначають траєкторії його переміщення, правила захоплення та

відпускання об'єктів. Точність переміщення є залежністю від команд оператора та подій в самій системі.

Модулі обробки також мають власні алгоритми, вони функціонують за принципом виклику операції після надходження деталі до верстату та отримання сигналу від оператора. Алгоритм включає контроль часу виконання операцій, індикацію завершення для переходу до наступного етапу.

Усі вищеперераховані алгоритми об'єднані в єдину структуру, реалізовану у вигляді логіки віртуального контролера. Він виконує моніторинг станів компонентів системи, приймає рішення на основі команд оператора, ініціює дії виконавчих механізмів і забезпечує узгоджену взаємодію між модулями. У межах реалізації в середовищі Visual Studio з використанням OpenGL контролер функціонує як ядро, що постійно оновлює графічну сцену відповідно до змін у логічному стані системи (рис. 2.4).

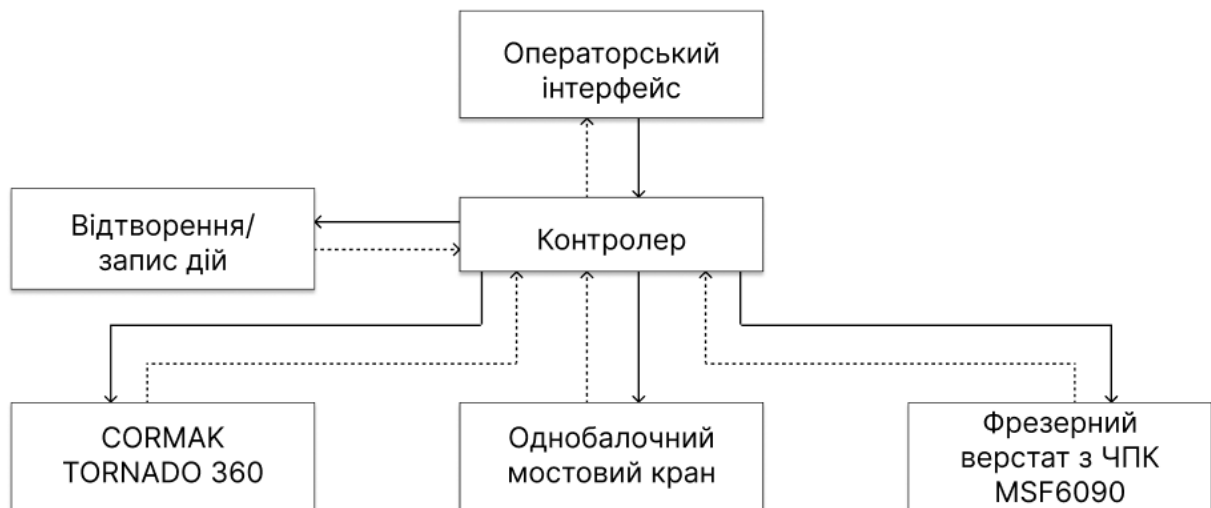


Рисунок 2.4 – Структурна схема алгоритму роботи програми

Отже проектування алгоритмів дозволило закласти функціональну основу для створення цифрового двійника виробничої дільниці, придатного для освітнього використання та тестування концепцій автоматизації. Подальший розвиток цієї системи може бути спрямований на її інтеграцію з зовнішніми

цифровими платформами, підтримку сценаріїв з розширеною логікою прийняття рішень для автоматичного коригування виробничих процесів.

3 РОЗРОБЛЕННЯ ВІРТУАЛЬНОГО МАКЕТА

3.1 Написання коду програми

Розроблення віртуального макета гнучкої виробничої дільниці (ГВД) вимагає створення інтерактивного програмного продукту, який поєднує графічну візуалізацію, логіку функціонування та засоби керування. У цьому розділі розглядається практична реалізація програмної моделі від початку до кінця.

Для реалізації проєкту були необхідні бібліотеки та заголовочні файли, які дозволяють створювати вікна, працювати з 2D та 3D графікою, а також взаємодіяти з користувачем через інтерфейс (рис. 3.1).

```
#pragma comment(lib, "user32.lib")
#pragma comment(lib, "gdi32.lib")
#pragma comment(lib, "glu32.lib")
#pragma comment(lib, "opengl32.lib")

#include <windows.h>
#include <GL/gl.h>
#include <GL/glu.h>
#include <cmath>
#include "MachineQuantityDialog.h"
```

Рисунок 3.1 – Заголовочні файли та бібліотеки необхідні для реалізації проєкту

Головними бібліотеками для реалізації проєкту слугують «windows.h», «GL/gl.h», «GL/glu.h» та «cmath».

Бібліотека «windows.h» потрібна для доступу до функцій Windows API, що забезпечують створення вікон, роботу з повідомленнями, таймерами, графічними пристроями, обробку подій від клавіатури та інших системних функцій необхідних для реалізації інтерактивного графічного застосунку в середовищі Windows.

Бібліотека «GL/gl.h» є основною бібліотекою OpenGL і містить базові функції для побудови тривимірної графіки: малювання геометричних фігур,

налаштування освітлення, матеріалів, кольорів, трансформацій об'єктів у просторі.

Бібліотека «GL/glu.h» є допоміжною бібліотекою OpenGL і забезпечує додаткові інструменти, такі як зручне створення перспективної проекції, автоматична генерація міпмапів для текстур, побудова складніших об'єктів та підтримка роботи з камерами через gluLookAt().

Бібліотека «cmath» використовується для виконання математичних розрахунків, необхідних у графіці, таких як обчислення кутів, синусів, косинусів, коренів, степенів тощо. Це необхідно для обчислення напрямків руху об'єктів, анімацій, побудови траєкторій або поворотів у просторі.

Після чого було реалізовано основні елементи інтерфейсу користувача та графічної сцени з використанням C++/CLI у середовищі Windows Forms з OpenGL. Головна форма містить елементи керування, такі як кнопки для додавання та видалення верстатів, очищення списку готових деталей, виклику налаштувань та перегляду керуючих команд. Також було додано вертикальну, горизонтальну прокрутку та обробник прокручування колеса миші, що дозволяє оглядати віртуальне середовище під різними кутами (рис. 3.2).

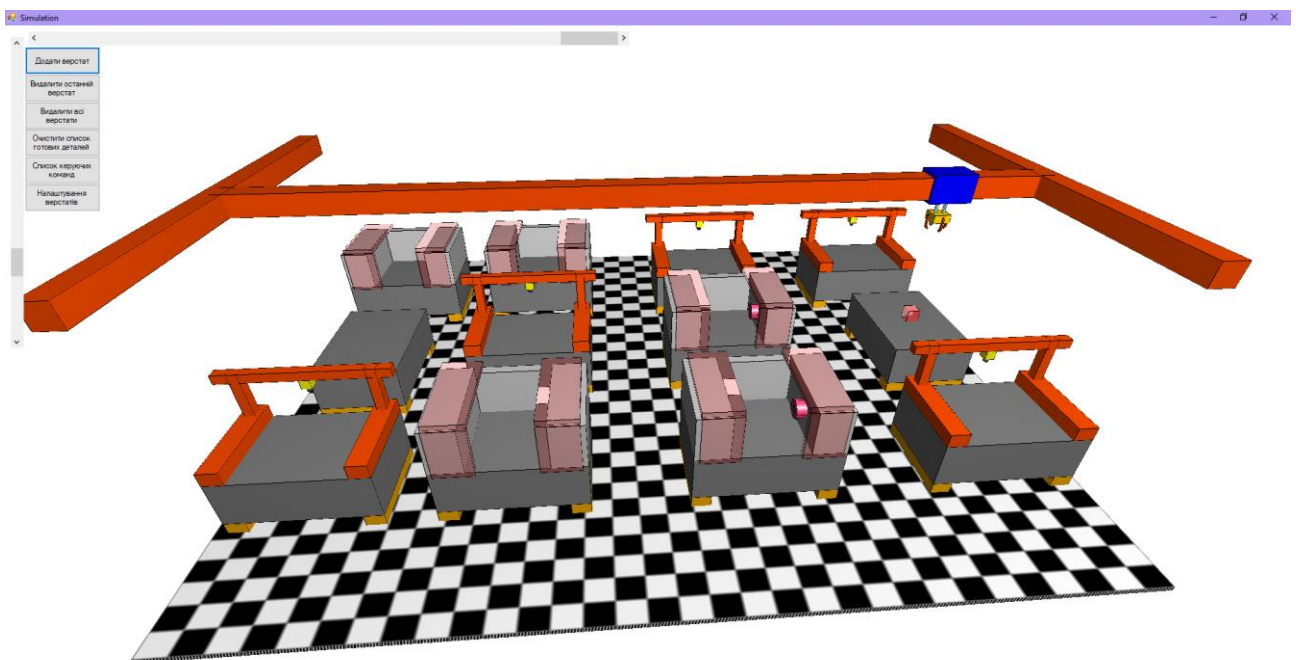


Рисунок 3.2 – Вигляд головної форми проекту

У проєкті створено кілька таймерів кожен із яких відповідає за окремі анімації такі як відкривання дверей верстату, обертання шпинделя фрезерного верстату, рух кран балки, опускання клешні крана, горизонтальне переміщення кран балки та відлік часу до завершення обробки деталі (рис. 3.3).

```
Timer^ renderTimer;  
Timer^ countTimer;  
Timer^ doorTimer;  
Timer^ spindleTimer;  
Timer^ craneTimer;  
Timer^ clawsTimer;  
Timer^ cranePillarTimer;
```

Рисунок 3.3 – Таймери для роботи анімацій

Основною функцією для налаштування роботи OpenGL є «InitializeOpenGL» та «SetupProjection», які налаштовують піксельний формат для відображення тривимірної сцени, активують контекст рендерингу, вмикають глибину, базове освітлення, кольорові матеріали та згладжування, що дозволяє досягти більш реалістичного вигляду сцени. Також встановлюється перспективна проєкція, щоб сцена відображалась з урахуванням глибини та перспективи.

Центральним елементом є функція рендерингу сцени RenderScene, яка циклічно виконується за допомогою таймера та відповідає за виведення усіх графічних об'єктів на екран. У цій функції оновлюється сцена. Задається положення камери, залежно від значень прокрутки масштабу, після чого відбувається візуалізація кожного графічного об'єкта. Завершення рендеру виконується через glFinish і SwapBuffers, що дозволяє забезпечити безперервне та плане оновлення зображення на екрані.

Усі графічні елементи, які можна побачити на сцені були зроблені за допомогою двох основних функцій, це «Cube» та «Celinder».

Функція «Cube» відповідає за побудову куба заданого розміру, з урахуванням координат вершин, освітлення та текстуровання.

Функція «Celinder» створює циліндричні об'єкти з круглими основами, які можна використовувати для зображення валів, шпинделів або інших частин. Обидві функції дозволяють додавати до сцени геометричні елементи, які є частиною загальної візуалізації виробничого середовища.

За допомогою цих функцій та основних функцій бібліотеки OpenGL були побудовані два верстати («DrawBaseStanok» та «DrawBaseStanok2»), мостовий кран, його рухома теліжка, клешні для захвату об'єкта («Crane», «Crane_cart» і «Crane_claws»), об'єкт («object»), підлога на якій все знаходиться («Floor»), місце отримання заготовки та місце для вивантаження готової деталі («Elements» та «Cage»).

Для підлоги («Floor») було реалізовано текстуру в вигляді шахматної дошки та ініціалізовано її (рис. 3.4).

```

void makeCheckerboard() {
    for (int i = 0; i < CHECKERBOARD_SIZE; i++) {
        for (int j = 0; j < CHECKERBOARD_SIZE; j++) { ... }
    }
}

void InitializeFloorTexture() {
    makeCheckerboard();
    glGenTextures(1, &floorTextureID);
    glBindTexture(GL_TEXTURE_2D, floorTextureID);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, CHECKERBOARD_SIZE, CHECKERBOARD_SIZE, 0, GL_RGB, GL_UNSIGNED_BYTE, checkerboardImage);
    glBuild2DMipmaps(GL_TEXTURE_2D, GL_RGB, CHECKERBOARD_SIZE, CHECKERBOARD_SIZE, GL_RGB, GL_UNSIGNED_BYTE, checkerboardImage);
}

```

Рисунок 3.4 – Реалізація та ініціалізація текстури шахматної дошки

Для кожного верстата та балочного мостового крану були реалізовані анімації за допомогою таймерів, які роблять їх інтерактивними.

Анімація для першого верстата була розділена на три фази. Перша фаза – закривання дверцят верстату, друга – очікування оброблення об'єкта, третя – відкривання дверцят після чого зміна кольору об'єкта, щоб було зрозуміло, що об'єкт був оброблений (рис. 3.5).

```

void DoorTimer_Tick(System::Object^ sender, System::EventArgs^ e) {
    MachineInstance^ targetMachine = FindMachineWithObject();
    if (targetMachine == nullptr) {
        doorTimer->Stop();
        return;
    }

    if (targetMachine->doorPhase == 0) {
        targetMachine->door_scale += door_scale_change;
        if (targetMachine->door_scale >= 0.3f) {
            targetMachine->door_scale = 0.3f;
            targetMachine->doorPhase = 1;
        }
    }

    else if (targetMachine->doorPhase == 1) {
        targetMachine->pauseCounter += doorTimer->Interval;
        if (targetMachine->pauseCounter >= pauseLimit) {
            targetMachine->doorPhase = 2;
        }
    }

    else if (targetMachine->doorPhase == 2) {
        targetMachine->door_scale -= door_scale_change;
        if (targetMachine->door_scale <= 0.0f) {
            targetMachine->door_scale = 0.0f;
            doorTimer->Stop();
            targetMachine->isDoorAnimationActive = false;
        }
    }

    cubeColor[0] = 0.0f;
    cubeColor[1] = 1.0f;
    cubeColor[2] = 0.0f;
    cubeColor[3] = 0.5f;
}

this->RenderScene(nullptr, nullptr);
}

```

Рисунок 3.5 – Функція анімації роботи першого верстата

У функції анімація другого верстата реалізовано покрокову анімацію роботи шпинделя. Усе відбувається поетапно відповідно до поточної фази анімації. Спочатку шпиндель поступово рухається до робочої позиції, коли він досягає її починається наступна фаза (рис. 3.6).

```

void SpindleTimer_Tick(System::Object^ sender, System::EventArgs^ e) {
    MachineInstance^ targetMachine = FindMachineWithSpindleAnimation();
    if (targetMachine == nullptr || !targetMachine->spindleAnimationActive) {
        spindleTimer->Stop();
        return;
    }

    if (targetMachine->spindlePhase == 0) {
        bool arrivedAtOperatingX = true;
        if (targetMachine->spindle_Translate[0] < OPERATING_X_TARGET) {
            targetMachine->spindle_Translate[0] += move_speed;
            if (targetMachine->spindle_Translate[0] > OPERATING_X_TARGET) targetMachine->spindle_Translate[0] = OPERATING_X_TARGET;
            arrivedAtOperatingX = false;
        }
        else if (targetMachine->spindle_Translate[0] > OPERATING_X_TARGET) {
            targetMachine->spindle_Translate[0] -= move_speed;
            if (targetMachine->spindle_Translate[0] < OPERATING_X_TARGET) targetMachine->spindle_Translate[0] = OPERATING_X_TARGET;
            arrivedAtOperatingX = false;
        }
        targetMachine->crossbar_Translate[0] = targetMachine->spindle_Translate[0];
        targetMachine->pillar_Translate[0] = targetMachine->spindle_Translate[0];

        targetMachine->spindle_Translate[2] = 0.0f;
        targetMachine->crossbar_Translate[2] = 0.0f;
        targetMachine->pillar_Translate[2] = 0.0f;

        if (arrivedAtOperatingX) {
            targetMachine->spindlePhase = 1;
        }
    }
}

```

Рисунок 3.6 – Перша фаза роботи фрезерного верстата

Друга фаза включає в себе опускання шпинделя та пов'язаних з ним елементів (рис. 3.7).

```

else if (targetMachine->spindlePhase == 1) {
    bool allLowered = true;
    if (targetMachine->spindle_Translate[1] > (INITIAL_SPINDLE_Y - dropDistance)) {
        targetMachine->spindle_Translate[1] -= move_speed;
        allLowered = false;
    }
    else {
        targetMachine->spindle_Translate[1] = INITIAL_SPINDLE_Y - dropDistance;
    }

    if (targetMachine->crossbar_Translate[1] > (INITIAL_CROSSBAR_Y - dropDistance)) {
        targetMachine->crossbar_Translate[1] -= move_speed;
        allLowered = false;
    }
    else {
        targetMachine->crossbar_Translate[1] = INITIAL_CROSSBAR_Y - dropDistance;
    }

    if (targetMachine->pillar_Translate[1] > (INITIAL_PILLAR_Y - dropDistance)) {
        targetMachine->pillar_Translate[1] -= move_speed;
        allLowered = false;
    }
    else {
        targetMachine->pillar_Translate[1] = INITIAL_PILLAR_Y - dropDistance;
    }

    if (allLowered) {
        targetMachine->spindlePhase = 2;
    }
}

```

Рисунок 3.7 – Друга фаза роботи фрезерного верстата

Після цього в третій фазі шпindelь обертається по колу наче обробляє деталь (рис. 3.8).

```

else if (targetMachine->spindlePhase == 2) {
    targetMachine->spindleAngle += 0.05f;
    if (targetMachine->spindleAngle >= 2 * 3.14159f) {
        targetMachine->spindleAngle = 0.0f;
        targetMachine->spindlePhase = 3;
    }

    GLfloat x = OPERATING_X_TARGET + spindleRadius * cos(targetMachine->spindleAngle);
    GLfloat z = spindleRadius * sin(targetMachine->spindleAngle);
    targetMachine->spindle_Translate[0] = x;
    targetMachine->spindle_Translate[2] = z;
    targetMachine->crossbar_Translate[0] = x;
    targetMachine->crossbar_Translate[2] = z;
    targetMachine->pillar_Translate[0] = x;
    targetMachine->pillar_Translate[2] = z;
}

```

Рисунок 3.8 – Третя фаза роботи фрезерного верстата

Коли оберт завершено в четвертій фазі об'єкт змінює свій колір і шпindelь фрезерного станка повертається в початкову точку (рис. 3.9).

```

else if (targetMachine->spindlePhase == 3) {
    cubeColor[0] = 0.0f;
    cubeColor[1] = 0.0f;
    cubeColor[2] = 1.0f;
    cubeColor[3] = 0.5f;

    bool allReset = true;
    if (targetMachine->spindle_Translate[1] < INITIAL_SPINDLE_Y) {
        targetMachine->spindle_Translate[1] += move_speed;
        if (targetMachine->spindle_Translate[1] > INITIAL_SPINDLE_Y) targetMachine->spindle_Translate[1] = INITIAL_SPINDLE_Y;
        allReset = false;
    }
    if (targetMachine->crossbar_Translate[1] < INITIAL_CROSSBAR_Y) {
        targetMachine->crossbar_Translate[1] += move_speed;
        if (targetMachine->crossbar_Translate[1] > INITIAL_CROSSBAR_Y) targetMachine->crossbar_Translate[1] = INITIAL_CROSSBAR_Y;
        allReset = false;
    }
    if (targetMachine->pillar_Translate[1] < INITIAL_PILLAR_Y) {
        targetMachine->pillar_Translate[1] += move_speed;
        if (targetMachine->pillar_Translate[1] > INITIAL_PILLAR_Y) targetMachine->pillar_Translate[1] = INITIAL_PILLAR_Y;
        allReset = false;
    }
    if (targetMachine->spindle_Translate[0] > REST_X_TARGET) {
        targetMachine->spindle_Translate[0] -= move_speed;
        if (targetMachine->spindle_Translate[0] < REST_X_TARGET) targetMachine->spindle_Translate[0] = REST_X_TARGET;
        allReset = false;
    }
    else if (targetMachine->spindle_Translate[0] < REST_X_TARGET) {
        targetMachine->spindle_Translate[0] += move_speed;
        if (targetMachine->spindle_Translate[0] > REST_X_TARGET) targetMachine->spindle_Translate[0] = REST_X_TARGET;
        allReset = false;
    }
}

```

Рисунок 3.9 – Четверта фаза роботи фрезерно верстата

Робота мостового кран балки поділена на три анімації:

– рух кран балки по горизонталі між рядами верстатів;

– рух верхньої теліжки яка рухається горизонтально від верстата до верстата;

– рух по вертикалі клешнів, які піднімають та опускають об’єкт.

У функції анімації руху теліжки реалізується рух крана по осі Z. Коли від оператора надходить команда руху, починається поступове наближення теліжки до заданої цільової координати. Щоразу при виклику таймера позиція теліжки та об’єктів, які прив’язані до неї змінюється в бік заданої координати. Коли теліжка досягла цілі її положення фіксується, рух припиняється і таймер зупиняється (рис. 3.10).

```
void CraneTimer_Tick(System::Object^ sender, System::EventArgs^ e) {
    if (!craneMoving) return;
    float speed = 0.1f;
    float diff = crane_TargetZ - crane_Translate[2];
    if (fabs(diff) < speed) {
        crane_Translate[2] = crane_TargetZ;
        claws_Translate[2] = crane_TargetZ;
        craneMoving = false;
        craneTimer->Stop();
    }
    else {
        crane_Translate[2] += (diff > 0 ? speed : -speed);
        claws_Translate[2] += (diff > 0 ? speed : -speed);
    }

    this->RenderScene(nullptr, nullptr);
}
```

Рисунок 3.10 – Анімація переміщення теліжки

У функції анімації руху клешень, коли перша фаза активна, клешні рухаються до низу до певної координати, після чого перевіряється чи знаходиться об’єкт достатньо близько для того щоб його захопити, якщо так – об’єкт рухається разом з клешнями, потім якщо було вказано, що об’єкт можна відпускати його позиція фіксується на тому місці, де зараз знаходяться клешні і об’єкт вважається відпущеним. Далі запускається друга фаза в якій клешні піднімають до гори. Коли вони досягають початкової висоти анімація закінчується. Також під час виконання анімації оновлюється довжина тросу, який опускає та піднімає клешні (рис. 3.11).

```

void ClawsTimer_Tick(System::Object^ sender, System::EventArgs^ e) {
    float clawsSpeed = 0.05f;

    if (clawsPhase == 1) {
        if (claws_Translate[1] > -2.3f) {
            claws_Translate[1] -= clawsSpeed;
            if (claws_Translate[1] < -2.3f) claws_Translate[1] = -2.3f;
        }
        else {
            clawPauseCounter++;

            if (clawPauseCounter >= 2 && !triedToAttach) {
                float dx = fabs(claws_Translate[0] - object_Pos[0]);
                float dz = fabs(crane_Translate[2] - object_Pos[2]);
                float dy = fabs((claws_Translate[1] + 3.6f) - object_Pos[1]);

                if (dx < 0.3f && dy < 0.3f && dz < 0.3f) {
                    objectAttached = true;
                }

                triedToAttach = true;
                clawsPhase = 2;

                if (objectAttached && shouldReleaseObject) {
                    objectAttached = false;
                    object_Pos[1] = 1.35f;
                    object_Pos[0] = claws_Translate[0];
                    object_Pos[2] = crane_Translate[2];
                    shouldReleaseObject = false;
                }
            }
        }
    }
    else if (clawsPhase == 2) {
        if (claws_Translate[1] < 0.0f) {
            claws_Translate[1] += clawsSpeed;
            if (claws_Translate[1] > 0.0f) claws_Translate[1] = 0.0f;
        }
        else {
            clawsPhase = 0;
            clawsMoving = false;
            triedToAttach = false;
            clawPauseCounter = 0;
            clawsTimer->Stop();
        }
    }
    ropeLength = fabs(claws_Translate[1]);
    this->RenderScene(nullptr, nullptr);
}

```

Рисунок 3.11 – Анімація руху клешень кран балки

У функції руху кран балки відбувається анімація горизонтального переміщення крана. Якщо було отримано вказівку від оператора про рух крана, він поступово рухається в позицію вказану оператором. Разом з ним рухаються теліжка та клешні, які прикріплені до неї. Як тільки координата була досягнута, рух припиняється (рис. 3.12).

```

void cranePillarTimer_Tick(System::Object^ sender, System::EventArgs^ e) {
    if (!cranePillarMoving) return;

    float currentX = crane_pillar_Translate[0];
    float targetX = cranePillar_TargetX;
    float moveSpeed = 0.1f;

    if (currentX < targetX) {
        currentX += moveSpeed;
        if (currentX > targetX) currentX = targetX;
    }
    else if (currentX > targetX) {
        currentX -= moveSpeed;
        if (currentX < targetX) currentX = targetX;
    }
    crane_pillar_Translate[0] = currentX;
    crane_Translate[0] = currentX;
    claws_Translate[0] = currentX;
    if (fabs(currentX - targetX) < 0.001f) {
        crane_pillar_Translate[0] = targetX;
        crane_Translate[0] = targetX;
        claws_Translate[0] = targetX;
        cranePillarMoving = false;
        cranePillarTimer->Stop();
    }
    this->RenderScene(nullptr, nullptr);
}

```

Рисунок 3.12 – Анімація руху кран балки

Також в проєкті було реалізовано запис та відтворення дій, які були вказані оператором. Якщо було ввімкнено запис дій, то кожна вказівка оператора записується в список для подальшого відтворення. Відтворення відбувається по порядку в залежності від списку відтворених дій, кожна дія має своє умовне позначення, наприклад, «А» – запуск анімації роботи першого верстата, «В» – запуск анімації другого верстата, «С» – рух клешень кран балки, «1», «2», «3», «4» – переміщення теліжки кран балки в певну позицію, «Q» та «E» – переміщення кран балки по горизонталі. Також є окремий запис для скидання об'єкта до початкових налаштувань (рис. 3.13 - 3.14).

```

void recordAction(String* action) {
    if (isRecording && recordedActions != nullptr) {
        recordedActions->Add(action);
    }
}

void playbackTimer_Tick(System::Object* sender, System::EventArgs* e) {
    if (recordedActions->Count == 0) {
        isPlaying = false;
        countTimer->Stop();
        MessageBox::Show("Відтворення закінчено (дів відсутні)", "Інформація");
        return;
    }

    if (playbackIndex >= recordedActions->Count) {
        playbackIndex = 0;
    }

    String* action = recordedActions[playbackIndex];
    if (action->Equals("A")) {
        MachineInstance* targetStanok1 = FindMachineWithObject();
        if (targetStanok1 != nullptr && !targetStanok1->isStanok1 && !targetStanok1->isDoorAnimationActive) {
            targetStanok1->isDoorAnimationActive = true;
            targetStanok1->doorPhase = 0;
            targetStanok1->pauseCounter = 0;
            doorTimer->Start();
        }
    }
    else if (action->Equals("B")) {
        MachineInstance* targetStanok2 = FindMachineWithObject();
        if (targetStanok2 != nullptr && !targetStanok2->isStanok1 && !targetStanok2->spindleAnimationActive) {
            targetStanok2->spindleAnimationActive = true;
            targetStanok2->spindlePhase = 0;
            targetStanok2->spindleAngle = 0.0f;
            spindleTimer->Start();
        }
    }
    else if (action->Equals("C")) {
        if (!clawsMoving) {
            clawsMoving = true;
            clawsPhase = 1;
            clawsTimer->Start();
        }
    }
    else if (action->Equals("1")) {
        crane_TargetZ = -7.0f;
        craneMoving = true;
        craneTimer->Start();
    }
    else if (action->Equals("2")) {
        crane_TargetZ = -2.0f;
        craneMoving = true;
        shouldReleaseObject = true;
        craneTimer->Start();
    }
    else if (action->Equals("3")) {
        crane_TargetZ = 3.0f;
        craneMoving = true;
        shouldReleaseObject = true;
        craneTimer->Start();
    }
    else if (action->Equals("4")) {
        crane_TargetZ = 7.0f;
        craneMoving = true;
        shouldReleaseObject = true;
        craneTimer->Start();
    }
}

```

Рисунок 3.13 – Функції запису та відтворення дій

```

    else if (action->Equals("E")) {
        cranePillar_TargetX = crane_pillar_Translate[0] + 4.0f;
        cranePillarMoving = true;
        shouldReleaseObject = true;
        cranePillarTimer->Start();
    }
    else if (action->Equals("Q")) {
        cranePillar_TargetX = crane_pillar_Translate[0] - 4.0f;
        cranePillarMoving = true;
        shouldReleaseObject = true;
        cranePillarTimer->Start();
    }
    else if (action->Equals("R")) {
        object_Pos[0] = 0.0f;
        object_Pos[1] = 1.35f;
        object_Pos[2] = -7.0f;
        cubeColor[0] = 1.0f;
        cubeColor[1] = 0.0f;
        cubeColor[2] = 0.0f;
        cubeColor[3] = 0.5f;
        shouldReleaseObject = false;
        objectAttached = false;
        clawPauseCounter = 0;
        finishedPartsCount++;
        MessageBox::Show(String::Format("Кількість готових деталей: {0}", finishedPartsCount),
            "Готові деталі",
            MessageBoxButtons::OK,
            MessageBoxIcon::None);
    }
    playbackIndex++;
    this->RenderScene(nullptr, nullptr);
}

```

Рисунок 3.14 – Кінець функції відтворення дій

Для того щоб анімації запускалися тільки на тому верстаті на якому знаходиться об'єкт було реалізована дві функції де здійснюється пошук верстату за певними умовами.

У першій функції «FindMachineWithObject» здійснюється пошук верстата, який знаходиться найближче до об'єкта. Перевіряється відстань між об'єктом і верстатом по горизонталі та вертикалі і якщо відстань знаходиться в межах заданих порогів то ця функція повертає відповідний екземпляр верстата.

У другій функції «FindMachineWithSpindleAnimation» відбувається пошук верстатів другого типу, у якого активована анімація. Якщо такий верстат знайдено то функція повертає його (рис. 3.15).

```
MachineInstance^ FindMachineWithObject() {
    float horizontalProximityThreshold = 1.5f;
    float verticalProximityThreshold = 2.0f;
    for each (MachineInstance ^ machine in placedMachines) {
        float dx = fabs(machine->x - object_Pos[0]);
        float dz = fabs(machine->z - object_Pos[2]);
        float dy = fabs(machine->y - object_Pos[1]);
        if (dx < horizontalProximityThreshold && dz < horizontalProximityThreshold) {
            if (dy < verticalProximityThreshold) {
                return machine;
            }
        }
    }
    return nullptr;
}

MachineInstance^ FindMachineWithSpindleAnimation() {
    for each (MachineInstance ^ machine in placedMachines) {
        if (!machine->isStanok1 && machine->spindleAnimationActive) {
            return machine;
        }
    }
    return nullptr;
}
```

Рисунок 3.15 – Функція пошуку необхідного верстату для анімації

Також були реалізовані дві допоміжні форми. У першій формі («MachineQuantityDialog») було реалізовано діалогове вікно, яке використовується для того, щоб користувач міг вибрати кількість та тип верстатів перед їх розміщенням на сцені (рис. 3.16).

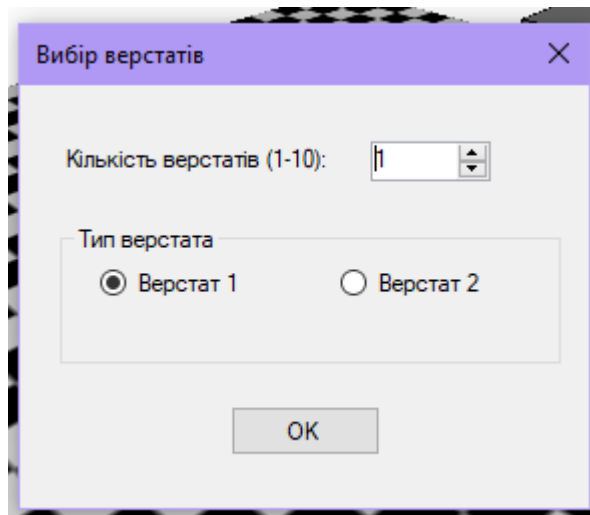


Рисунок 3.16 – Форма для вибору типу та кількості верстатів

Друга допоміжна форма («SettingsForm») слугує для налаштування параметрів верстатів, зокрема швидкості роботи. Вона надає користувачеві зручний інтерфейс у вигляді повзунків, які дозволяють візуально та інтуїтивно встановлювати значення, а також одразу бачити числовий результат кожного параметра (рис. 3.17).

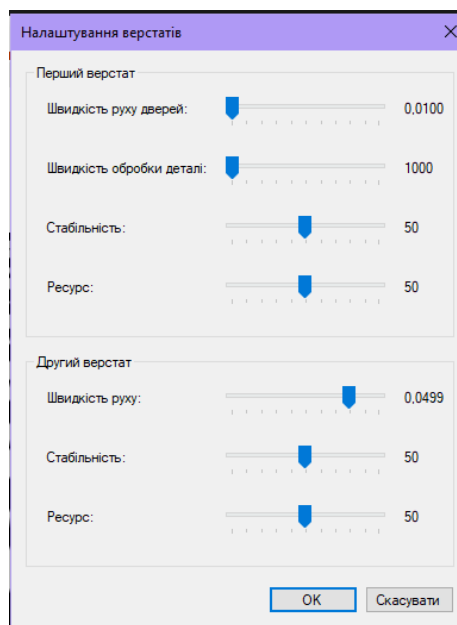


Рисунок 3.17 – Форма налаштування верстатів

3.2 Дослідження роботи програми

Як було згадано раніше, додавання верстатів відбувається за допомогою форми «MachineQuantityDialog». Після натискання на кнопку «Додати верстат» з'являється форма яка представлена на рис. 3.16. В цій формі можна обрати тип та кількість верстатів. Кількість верстатів, які можуть одночасно знаходитися на сцені, штучно обмежена і дорівнює 10 (рис. 3.18 - 3.19).

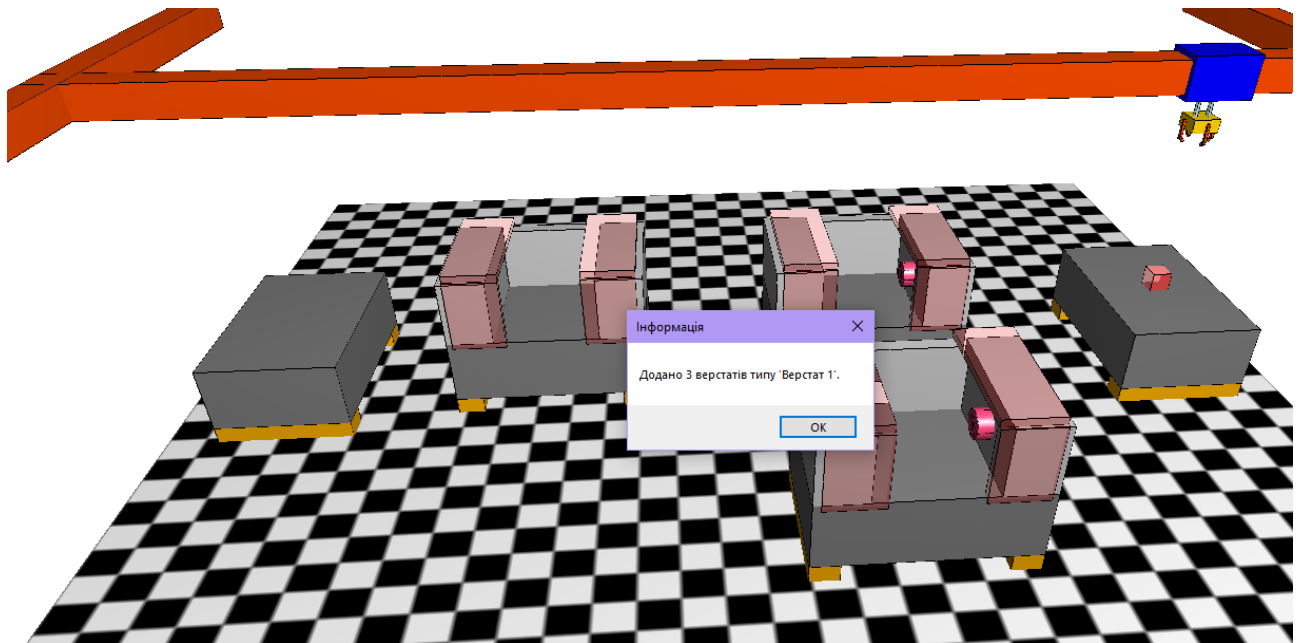


Рисунок 3.18 – Виставлення трьох верстатів одного типу

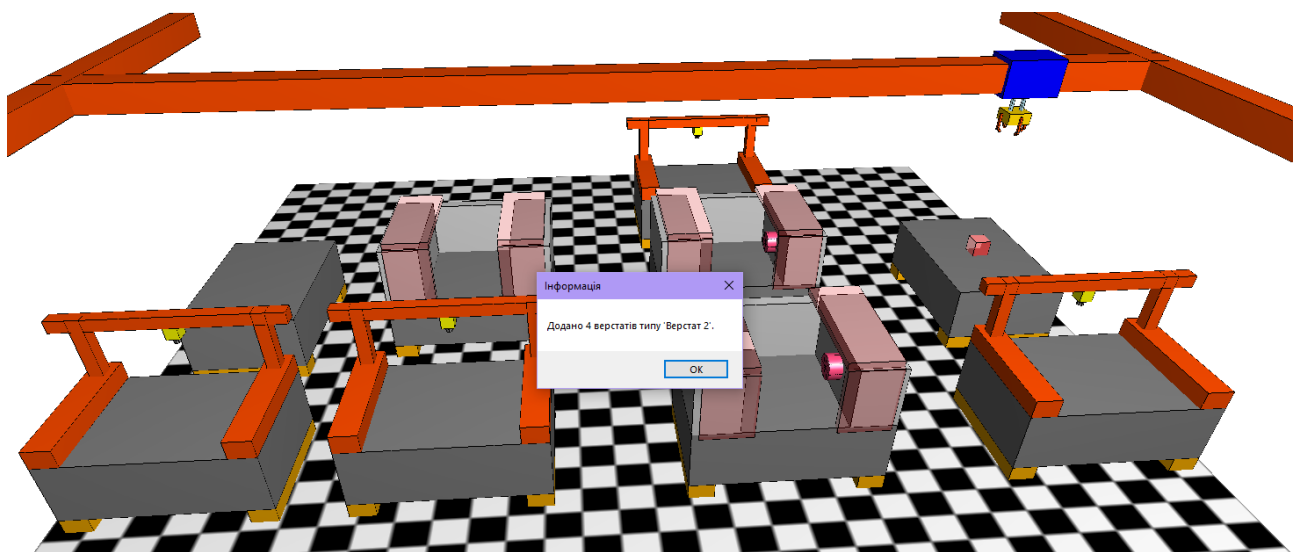


Рисунок 3.19 – Виставлення ще чотирьох верстатів другого типу

Якщо кількість верстатів для виставлення на сцену більша, за кількість вільних місць для їх розміщення, з'являється повідомлення, яке інформує, що досягнуто максимальної кількості верстатів, після чого виставляється та кількість верстатів, яка дорівнює кількості вільних місць (рис. 3.20 - 3.21).

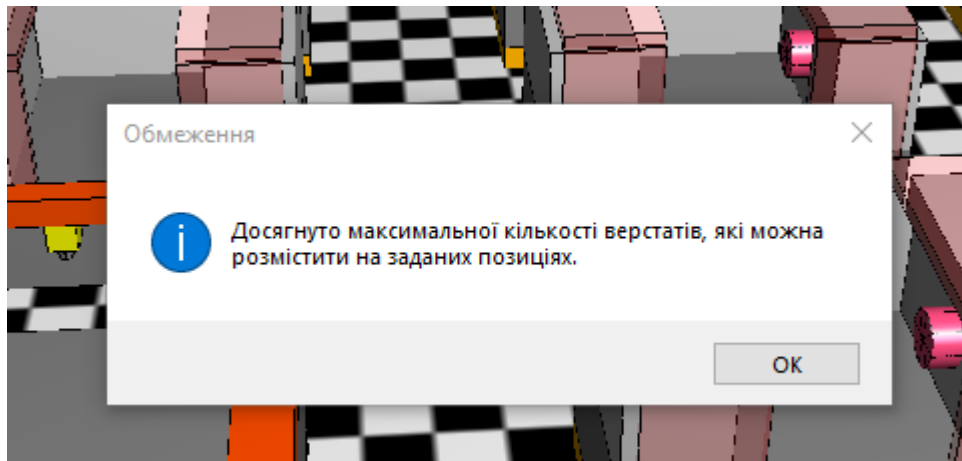


Рисунок 3.20 – Інформаційне вікно обмеження кількості верстатів

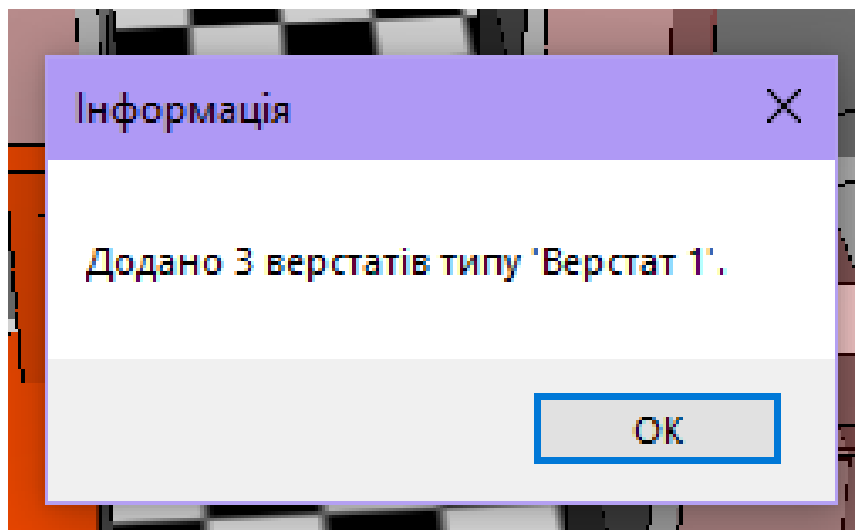


Рисунок 3.21 – Кількість доданих верстатів

У програмі розроблено два типи видалення верстатів зі сцени, це «Видалити останній верстат» та «Видалити всі верстати». Після натискання на кнопку «Видалити останній верстат», видалається останній виставлений на сцену верстат. Якщо натиснути на кнопку «Видалити всі верстати», з'явиться форма, де

буде уточнюватись чи точно ви хочете видалити всі верстати, після надання позитивної відповіді верстати видаляються (рис. 3.22).

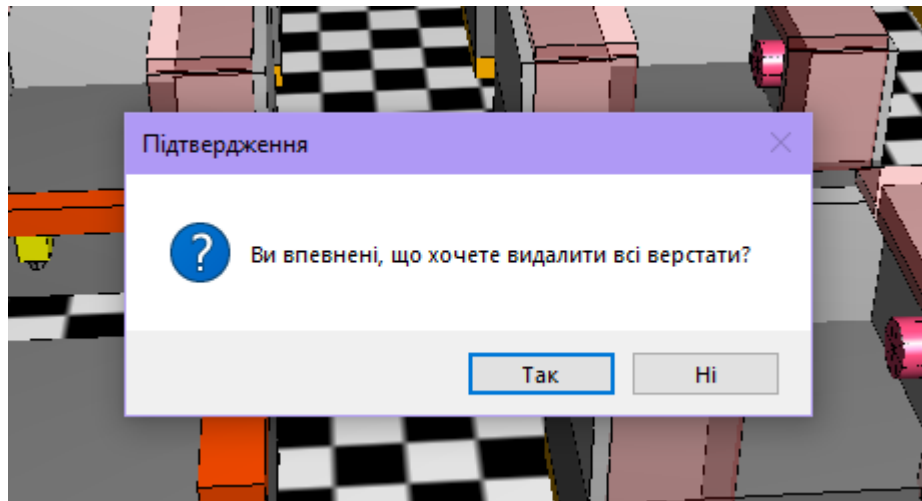


Рисунок 3.22 – Підтвердження видалення всіх верстатів

Для кожного типу верстату та балочного крану реалізовано власні анімації рухів (рис. 3.23 - 3.26).

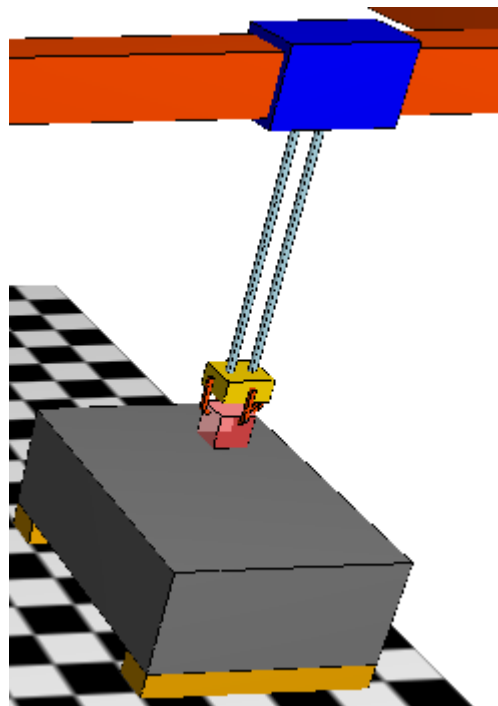


Рисунок 3.23 – Піднімання об'єкта

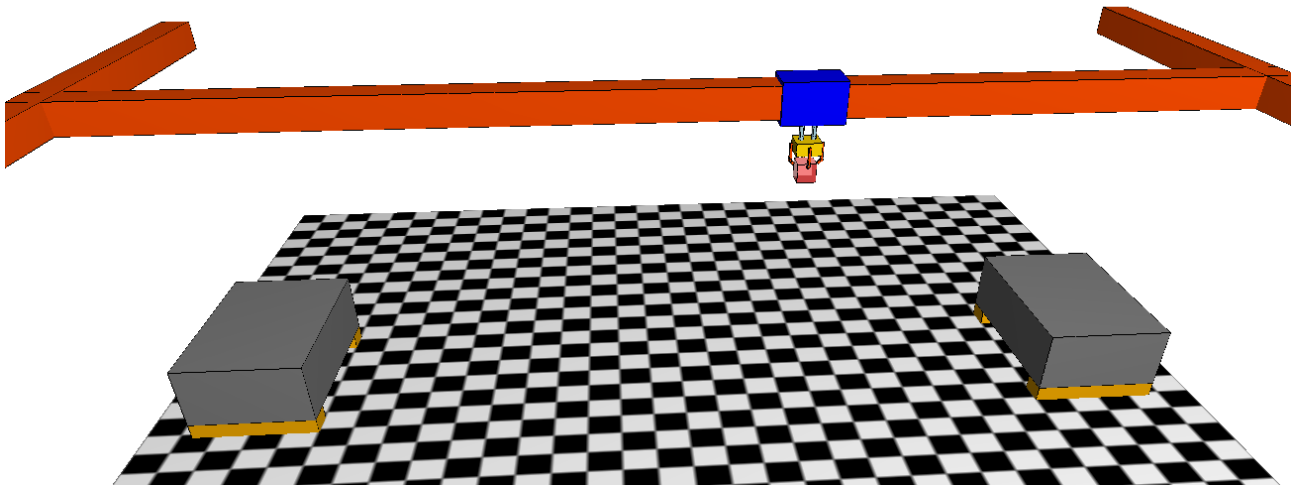


Рисунок 3.24 – Транспортування деталі за допомогою кран балки

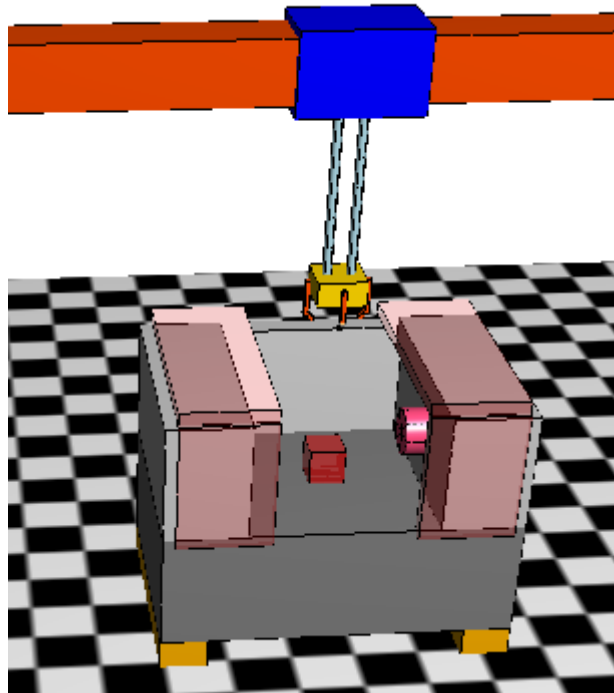


Рисунок 3.25 – Опускання об'єкта

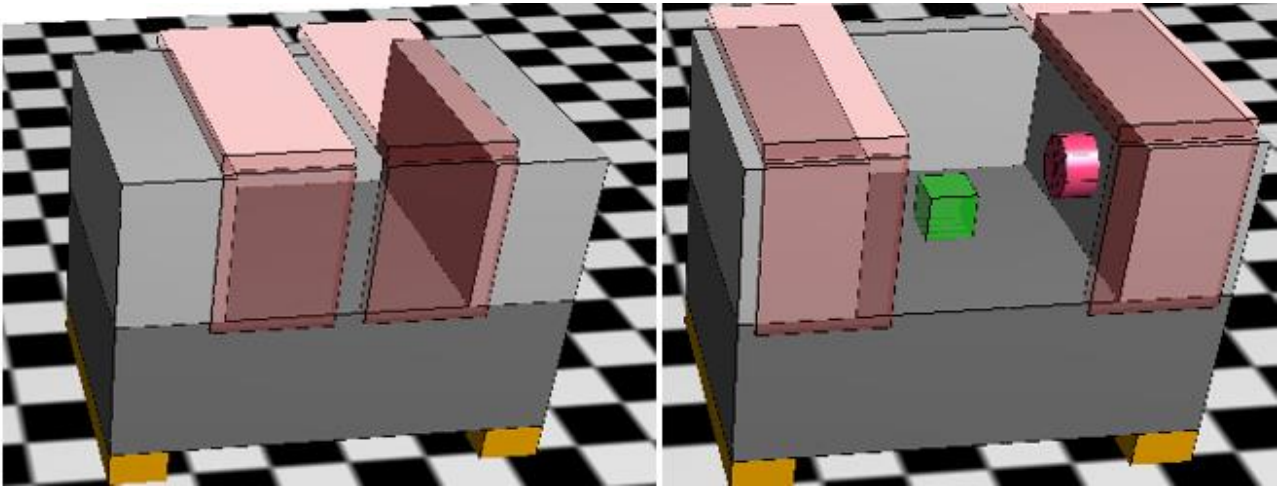


Рисунок 3.26 – Анімація обробки для першого типу верстата

Після закінчення обробки на першому типі верстатів об'єкт змінює свій колір на зелений, для того щоб користувач міг розуміти, що оброблення деталі пройшло успішно. Якщо оброблення відбувалося на другому типі верстатів, то об'єкт змінює свій колір на синій (рис. 3.27).

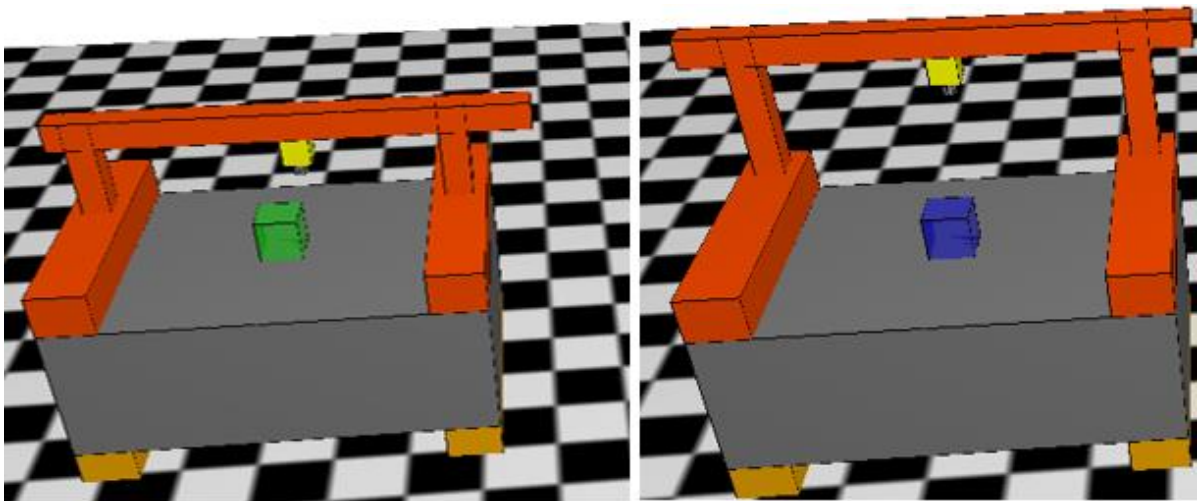


Рисунок 3.27 – Анімація обробки для другого типу верстатів

Якщо користувач подає сигнал для початку обробки деталі, а її на верстаті не має, або клешні кран балки ще не встигли відпустити об'єкт то з'являється інформаційна форма (рис. 3.28).

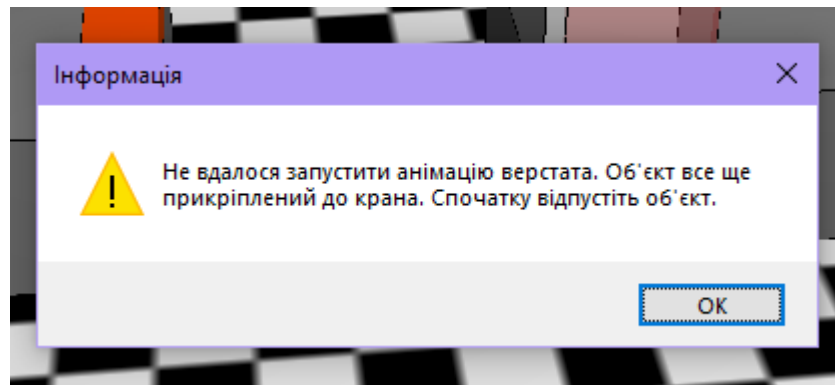


Рисунок 3.28 – Інформація про не наявність об'єкта в зоні обробки

В програмі також реалізований список керуючих команд, які допоможуть користувачеві краще орієнтуватися в виконавчих командах (рис. 3.29).

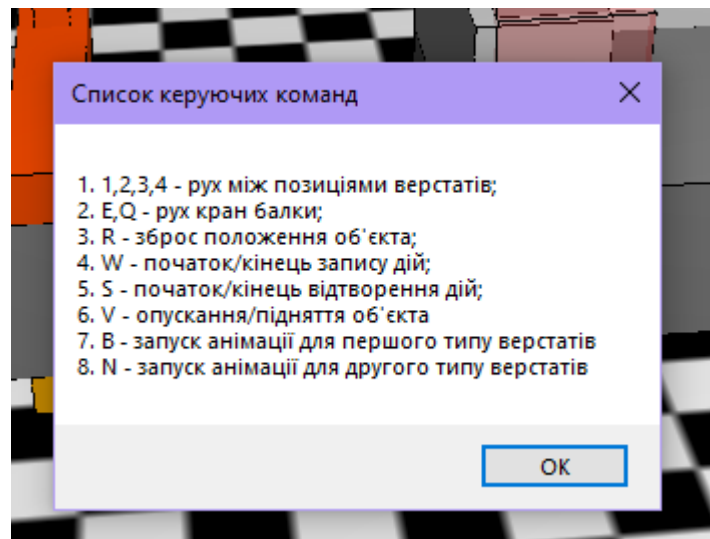


Рисунок 3.29 – Список керуючих команд

Під час початку та кінця запису та відтворення дій з'являються форми, які інформують користувача про початок/кінець цих операцій (рис. 3.30).

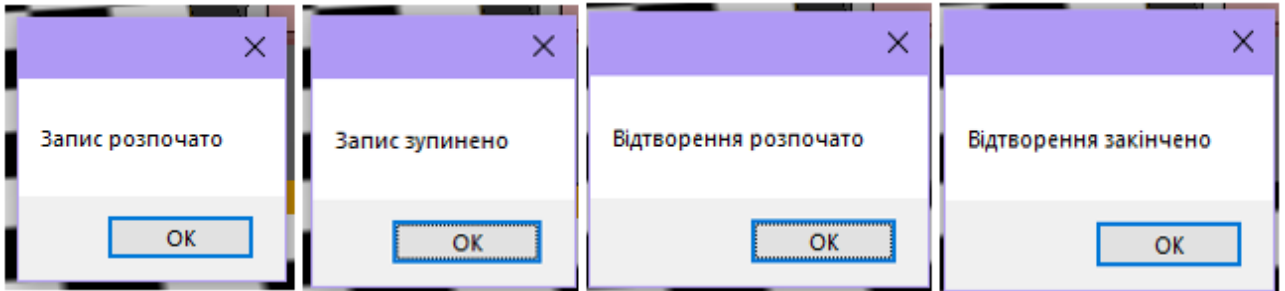


Рисунок 3.30 – Початок/кінець відтворення/запису дій

Користувач може змінювати налаштування верстатів, тим самим змінюючи виробничу ситуацію (рис. 3.31). Для кожного типу верстата є можливість налаштувати швидкість, стабільність та ресурс.



Рисунок 3.31 – Вікно налаштування верстатів

4 РОЗРАХУНОК СТІЙКОСТІ СИСТЕМИ УПРАВЛІННЯ ГРАВІРОВАЛЬНО-ФРЕЗЕРНОГО ВЕРСТАТА MSF6090

Сервоприводи, що використовуються у верстатах з ЧПК, зазвичай реалізують триконтурну каскадну структуру управління. Ця архітектура дозволяє досягти точності, жорсткості та динаміки, послідовно вирішуючи завдання управління на різних рівнях. Кожен з трьох вкладених контурів регулювання має власні унікальні атрибути, що забезпечує гарну продуктивність.

Взаємозалежність цих контурів та їхній вплив на загальну продуктивність системи є ключовим аспектом. Кожен контур у каскадній структурі має власні атрибути та безпосередньо впливає на загальну продуктивність системи.

Для сучасних ЧПК верстатів, таких як MSF6090, зазвичай використовуються синхронні серводвигуни з постійними магнітами завдяки їх високій ефективності, щільності потужності, високому співвідношенню крутного моменту ді інерції та високій точності [24 - 28].

4.1 Електрична частина, рівняння напруги

Рівняння напруги статора в d-q координатах описують електричну поведінку двигуна:

$$U_d = R_s i_d + L_d \frac{di_d}{dt} - \omega_d L_q i_q, \quad (4.1)$$

$$U_q = R_s i_q + L_q \frac{di_q}{dt} - \omega_e L_d i_d + \omega_e \psi_f, \quad (4.2)$$

де U_d, U_q – напруги статора по осях d-q, що подаються на обмотки двигуна;

i_d, i_q – струми статора по осях d-q, що протікають через обмотки;

R_s – опір обмоток статора;

L_d, L_q – індуктивності обмоток статора по осях d-q;

ω_e – електрична кутова швидкість ротора;

ψ_f – потік постійних магнітів, що проходить через обмотки статора.

4.2 Механічна частина, рівняння руху ротора

Механічна поведінка двигуна описується рівнянням руху ротора, яке пов'язує крутний момент, інерцію та швидкість:

$$J_{\text{двигун}} \frac{d\omega_m}{dt} = T_e - T_L - B_m \omega_m, \quad (4.3)$$

де $J_{\text{двигун}}$ – момент інерції ротора двигуна;

ω_m – механічна кутова швидкість ротора;

T_e – електромагнітний крутний момент, що розвивається двигуном;

T_L – крутний момент навантаження, приведений до валу двигуна;

B_m – коефіцієнт в'язкого тертя двигуна.

Електромагнітний крутний момент T_e визначається за формулою:

$$T_e = \frac{3}{2} P (\psi_f i_d + (L_d - L_q) i_d i_q). \quad (4.4)$$

Точне математичне моделювання динаміки приводів подачі є критично важливим кроком у проєктуванні високопродуктивних систем ЧПК. Математична модель дозволяє не лише краще розуміти вплив кожної змінної на динаміку машини, але й розробити відповідну систему управління, яка забезпечить бажану точність та швидкість.

4.3 Передавальні функції основних елементів сервоприводу

Для аналізу системи управління в частотній області, що є типовим підходом у теорії автоматичного управління (ТАУ), зручно використовувати передавальні функції. Це дозволяє представити динаміку кожного елемента системи та їх взаємодію в системі з зворотним зв'язком. Передавальна функція серводвигуна відображає відношення Лапласа вихідної змінної до вхідної, визначаючи як двигун перетворює електричну енергію в механічну.

4.3.1 Передавальна функція контуру струму

Завдяки своїй високій швидкості відгуку, контур струму часто може бути змодельований як інерційна ланка першого порядку. Це спрощення припустиме оскільки його постійна часу значно менша за постійну часу інших контурів:

$$W_i(s) = \frac{K_I}{T_1 s + 1}, \quad (4.5)$$

де K_I – коефіцієнт підсилення контуру струму;

T_1 – постійна часу контуру струму, яка в основному залежить від індуктивності та опору обмоток двигуна.

4.3.2 Передавальна функція контуру швидкості

Після того, як контур струму забезпечує точне управління крутним моментом, контур швидкості може бути змодельований. Він отримує задану швидкість і через регулятор швидкості генерує команду струму для внутрішнього контуру:

$$W_\omega(s) = \frac{K_\omega}{J_{\text{екв}} s + B_{\text{екв}}}, \quad (4.6)$$

де K_{ω} – коефіцієнт підсилення контуру швидкості;

$J_{\text{екв}}$ – еквівалентний момент інерції, приведений до валу двигуна;

$B_{\text{екв}}$ – еквівалентний коефіцієнт в'язкого тертя, приведений до валу двигуна.

4.3.3 Передавальна функція контуру положення

Зовнішній контур положення отримує задане положення i , за допомогою регулятора положення, генерує команду швидкості для контуру швидкості. Механічна передача перетворює обертовий рух двигуна в лінійний рух робочого органу:

$$W_{pos}(s) = \frac{K_{pos}}{s}, \quad (4.7)$$

де K_{pos} – коефіцієнт підсилення контуру положення;

s – оператор Лапласа, що відображає інтегруючу ланку, оскільки положення є інтегралом швидкості.

4.3.4 Передавальна функція механічного навантаження

Це критично важлива частина моделі, оскільки вона пов'язує обертовий рух двигуна з лінійним рухом робочого столу. Кульково-гвинтова пара (КГП) є механізмом, що перетворює обертовий рух у поступальний.

Сумарний момент інерції, приведений до валу двигуна ($J_{\text{сумарний}}$), є ключовим параметром, що впливає на динаміку системи. Він складається з моменту інерції ротора двигуна ($J_{\text{двигун}}$), моменту інерції самої кульково-гвинтової пари ($J_{\text{КГП}}$) та приведенного моменту інерції лінійного навантаження ($J_{\text{навантаження}}$):

$$J_{\text{сумарний}} = J_{\text{двигун}} + J_{\text{КГП}} + J_{\text{навантаження}} \quad (4.8)$$

4.4 Момент інерції кульково-гвинтової пари

Гвинт КГП можна апроксимувати як циліндр. Його момент інерції розраховується за формулою:

$$J_{\text{КГП}} = \frac{1}{2} m_{\text{КГП}} r_{\text{КГП}}^2 = \frac{1}{8} \pi \rho_{\text{сталь}} L_{\text{гвинта}} d_{\text{гвинта}}^4, \quad (4.9)$$

де $\rho_{\text{сталь}}$ – густина матеріалу гвинта, для сталі це 7850 кг/м³;

$L_{\text{гвинта}}$ – довжина гвинта;

$d_{\text{гвинта}}$ – діаметр гвинта.

4.5 Приведений момент інерції лінійного навантаження

Лінійна маса перетворюється на еквівалентний обертовий момент інерції на валу двигуна за формулою:

$$J_{\text{навантаження_приведений}} = M_{\text{рухома_частина}} \cdot \left(\frac{P}{2\pi} \right)^2, \quad (4.10)$$

де $M_{\text{рухома_частина}}$ – маса рухомих частин;

P – крок гвинта КГП.

Окрім інерційних сил на вісь подачі діють і сили тертя. Тертя в системах подачі верстатів з ЧПК є складним нелінійним явищем, що включає статичне, кулонівське та в'язке тертя. В математичних моделях тертя часто включається як частина демпфуючого члена в передавальній функції. Точне моделювання динаміки приводів подачі є критичним кроком у проектуванні високопродуктивних систем ЧПК. Аналіз показує, що тертя, зазори та крок кульково-гвинтової пари впливають на резонансні частоти та точність відстеження. Тому важливо враховувати всі тертя та інерційні ефекти.

4.6 Розрахунок заданих параметрів для верстата MSF6090

Для проведення розрахунків та аналізу системи управління верстата MSF6090 необхідно визначити або оцінити ключові механічні та електричні параметри, що впливають на динаміку. Характеристики верстата можна побачити в таблиці 4.1.

Таблиця 4.1 – Технічні характеристики верстата MSF6090

Назва характеристики	Значення
Робоче поле	600 мм×900 мм×160 мм
Напрямні	квадратні + кульково-гвинтові пари
Максимальна швидкість переміщення	10000 мм/хв
Максимальна робоча швидкість	6000 мм/хв
Приводи	сервоприводи
Шпиндель	2,2 кВт, 0-24000 об/хв
Вага верстату	250 кг
Маса рухомої частини	80 кг
Діаметр гвинта КГП	20 мм
Крок гвинта КГП	10 мм
Ефективність КГП	0,9
Номінальний крутний момент	15 Нм
Момент інерції ротора двигуна	$1 \cdot 10^{-4}$ кг·м ²
Опір обмотки статора	0,5 Ом
Індуктивність обмотки статора	5 мГн
Кількість пар полюсів	4
Потік постійних магнітів	0,49 Вб

Для розрахунків візьмемо одну з горизонтальних осей, наприклад Y-вісь, як найдовшу, оскільки розрахунки для неї будуть найбільш показовими з точки зору інерції та швидкості.

Для оцінки динамічних характеристик системи необхідно розрахувати сумарний момент інерції, приведений до валу двигуна.

Момент інерції кульково-гвинтової пари (4.9):

$$J_{\text{КГП}} = \frac{1}{8} \pi \rho_{\text{сталь}} L_{\text{гвинта}} d_{\text{гвинта}}^4,$$

$$J_{\text{КГП}} = \frac{1}{8} \pi \cdot 7850 \cdot 0,9 \cdot (0,02)^4 \approx 4,43 \cdot 10^{-5} \text{ кг} \cdot \text{м}^2.$$

Приведений момент інерції лінійного навантаження (5.10):

$$J_{\text{навантаження_приведений}} = M_{\text{рухома_частина}} \cdot \left(\frac{P}{2\pi} \right)^2,$$

$$J_{\text{навантаження_приведений}} = 80 \cdot \left(\frac{0,01}{2\pi} \right)^2 \approx 2,02 \cdot 10^5 \text{ кг} \cdot \text{м}^2.$$

Сумарний момент інерції, приведений до валу двигуна (5.8):

$$J_{\text{сумарний}} = J_{\text{двигун}} + J_{\text{КГП}} + J_{\text{навантаження_приведений}},$$

$$J_{\text{сумарний}} = 1 \cdot 10^{-4} + 4,43 \cdot 10^{-5} + 2,02 \cdot 10^5,$$

$$J_{\text{сумарний}} = 1,645 \cdot 10^{-4} \text{ кг} \cdot \text{м}^2.$$

Коефіцієнт інерції (співвідношення інерції навантаження до інерції двигуна):

$$\text{Коефіцієнт інерції} = \frac{J_{\text{навантаження_приведений}}}{J_{\text{двигун}}} = \frac{2,02 \cdot 10^5}{1 \cdot 10^{-4}} = 0,20. \quad (4.11)$$

Цей коефіцієнт значно менший від рекомендованого максимуму 30:1, що вказує на дуже хороші динамічні характеристики системи та потенціал для швидкого відгуку.

Для визначення необхідного крутного моменту двигуна необхідно врахувати всі сили, що діють на рухому вісь. Розглянемо горизонтальну вісь, де сила гравітації не впливає на рух.

Сила інерції виникає під час прискорення або гальмування рухомих мас.

$$F_i = M_{\text{рухома_частина}} \cdot a.$$

Максимальна швидкість переміщення $V_{\text{max}} = 1000 \text{ мм/хв} = 0,1667 \text{ м/с}$.

Припустимо, що час розгону $t_{\text{розгону}} = 0,1 \text{ с}$, це типовий значення для станків ЧПК з швидким відгуком.

Прискорення:

$$a = \frac{V_{\text{max}}}{t_{\text{розгону}}} = \frac{0,1667}{0,1 \text{ с}} = 1,667 \frac{\text{м}}{\text{с}^2},$$

$$F_i = 80 \cdot 1,667 \approx 133,36 \text{ Н.}$$

Сила тертя включає тертя в лінійних напрямних та кульково-гвинтовій парі. Для лінійних напрямних коефіцієнт тертя μ може бути дуже малим, близько 0,005-0,01. Нормальна сила $N = M_{\text{рухомої_частини}} \cdot g = 80 \cdot 9,8 \approx 784,8 \text{ (Н)}$.

$$F_T = \mu \cdot N = 0,005 \cdot 784,8 \approx 3,924 \text{ Н.}$$

Сила різання залежить від матеріалу, інструменту та режимів різання. Для фрезерного верстата це може бути значна сила. Прийmemo типові значення для невеликого фрезерного верстата 200 Н.

Сила гравітації для горизонтальних осей не впливає на рух, тому дорівнює нулю. Для вертикальної осі $F_g = M_{\text{рухомої_частини}} \cdot g$.

Сумарна сила навантаження, це сума всіх сил, що діють на вісь:

$$F_{\text{навантаження}} = F_i + F_T + F_{\text{різання}}, \quad (4.12)$$

$$F_{\text{навантаження}} = 133,6 + 3,924 + 200 = 337,284 \text{ Н.}$$

На основі розрахованих сил можна визначити необхідний крутний момент, який має розвивати серводвигун. Спочатку треба розрахувати необхідний крутний момент для подолання навантаження, з урахуванням ефективності КГП:

$$T_{\text{навантаження}} = \frac{F_{\text{навантаження}} \cdot R}{2\pi\eta_{\text{КГП}}}, \quad (4.13)$$

$$T_{\text{навантаження}} = \frac{337,284 \cdot 0,01}{2\pi \cdot 0,9} \approx 0,596 \text{ Н} \cdot \text{м}.$$

Далі розрахуємо необхідний крутний момент для прискорення:

$$T_{\text{прискорення}} = J_{\text{сумарний}} \cdot \frac{d\omega_m}{dt}, \quad (4.14)$$

де $\frac{d\omega_m}{dt}$ – кутове прискорення двигуна (рад/с²).

$$\frac{d\omega_m}{dt} = \frac{V_{\text{max}/P}}{t_{\text{розгону}}} = \frac{(0,1667) (0,01)}{0,1} = 166,7 \text{ рад/с}^2, \quad (4.15)$$

$$T_{\text{прискорення}} = 1,645 \cdot 10^{-4} \cdot 166,7 \approx 0,0274 \text{ Н} \cdot \text{м}.$$

Піковий крутний момент – це максимальний крутний момент, який потрібен двигуну під час прискорення з навантаженням.

$$T_{\text{піковий}} = T_{\text{навантаження}} + T_{\text{прискорення}}, \quad (4.16)$$

$$T_{\text{піковий}} = 0,596 + 0,0274 = 0,6234 \text{ Н} \cdot \text{м}.$$

Порівняння розрахованого пікового крутного моменту з номінальним крутним момент обраного двигуна показує, що серводвигун потужністю 2,2 кВт

має значний запас потужності. Це свідчить про те, що обраний є більш ніж достатнім для забезпечення необхідних динамічних характеристик осі подачі верстата MSF6090, навіть з урахуванням пікових навантажень під час прискорення та різання.

4.7 Визначення стійкості за критерієм Гурвіца.

Критерій Гурвіца є алгебраїчним методом, який дозволяє визначити стійкість системи за коефіцієнтами її характеристичного полінома. Система вважається стійкою, якщо всі корені її характеристичного рівняння лежать у лівій півплощині комплексної площини.

Для застосування критерію Гурвіца необхідно отримати характеристичне рівняння зімкненої системи управління. Як було зазначено в попередньому розділі, сервопривід ЧПК має каскадну триконтурну структуру:

а) контур струму;

б) контур швидкості, його замкнена передавальна функція може бути представлена як:

$$W_{\omega} = \frac{K_{P_{\omega}s} + K_{I_{\omega}}}{J_{\text{сумарний}}s^2 + K_{P_{\omega}s} + K_{I_{\omega}}}, \quad (4.17)$$

де $K_{P_{\omega}}$ – коефіцієнт пропорційності регулятора швидкості;

$K_{I_{\omega}}$ – коефіцієнт інтегрування регулятора швидкості;

$J_{\text{сумарний}}$ – сумарний момент інерції;

s – оператор Лапласа;

в) контур положення, в якому передавальна функція КГП, що перетворює кутову швидкість в лінійне положення має вигляд:

$$W_{\text{КГП}} = \frac{P}{2\pi s}, \quad (4.18)$$

де P – крок гвинта КГП (0,01 м).

Розглянемо замкнений контур положення, де вхідний сигнал, це задане положення, вихідний – фактичне положення. Регулятор положення генерує команду швидкості, яка подається на замкнений контур швидкості. Вихід контуру швидкості є фактичною кутовою швидкістю, яка через КГП перетворюється на лінійне положення. Зворотний зв'язок по положенню зазвичай рівний 1.

Передавальна функція розімкненого контуру положення:

$$G_{open_{pos}}(s) = G_{open_{pos}}(s) \cdot W_{\omega}(s) \cdot W_{КГП}(s), \quad (4.19)$$

$$G_{open_{pos}}(s) = K_{P_{pos}} \cdot \left(\frac{K_{P_{\omega}S} + K_{I_{\omega}}}{J_{сумарний}s^2 + K_{P_{\omega}S} + K_{I_{\omega}}} \right) \cdot \left(\frac{P}{2\pi S} \right).$$

Характеристичне рівняння замкненої системи має вигляд:

$$1 + K_{P_{pos}} \cdot \frac{K_{P_{\omega}S} + K_{I_{\omega}}}{J_{сумарний}s^2 + K_{P_{\omega}S} + K_{I_{\omega}}} \cdot \frac{P}{2\pi S} = 0, \quad (4.20)$$

$$s(J_{сумарний}s^2 + K_{P_{\omega}S} + K_{I_{\omega}}) + K_{P_{pos}} \frac{P}{2\pi} (K_{P_{\omega}S} + K_{I_{\omega}}) = 0,$$

$$J_{сумарний}s^3 + K_{P_{\omega}}s^2 + K_{I_{\omega}}s + K_{P_{pos}} \frac{P}{2\pi} K_{P_{\omega}}s^2 + K_{P_{pos}} \frac{P}{2\pi} K_{I_{\omega}}s = 0,$$

$$J_{сумарний}s^3 + K_{P_{\omega}}s^2 + \left(K_{I_{\omega}} + K_{P_{pos}} \frac{P}{2\pi} K_{P_{\omega}} \right) s + K_{P_{pos}} \frac{P}{2\pi} K_{I_{\omega}} = 0.$$

Це є характеристичний поліном системи третього порядку:

$$a_3s^3 + a_2s^2 + a_1s + a_0 = 0,$$

де коефіцієнти $a_3 = J_{сумарний} = 1,645 \cdot 10^{-4}$;

$$a_2 = K_{P_{\omega}} = 10;$$

$$a_1 = (K_{I_{\omega}} + K_{P_{pos}} \frac{P}{2\pi} K_{P_{\omega}} = 51,59;$$

$$a_0 = K_{ppos} \frac{P}{2\pi} K_{I\omega} = 7,96.$$

Отже характеристичний поліном системи з підставленими значення має вигляд:

$$1,645 \cdot 10^{-4} s^3 + 10 s^2 + 51,59s + 7,96 = 0.$$

Для стійкості системи необхідно, щоб виконувалося дві групи умов:

а) всі коефіцієнти характеристичного полінома мають бути додатними:

$$- a_3 = 1,645 \cdot 10^{-4} > 0 \text{ (умова виконується);}$$

$$- a_2 = 10 > 0 \text{ (умова виконується);}$$

$$- a_1 = 51,59 > 0 \text{ (умова виконується);}$$

$$- a_0 = 7,96 > 0 \text{ (умова виконується);}$$

б) визначники всіх головних діагональних мінорів матриці Гурвіца мають бути додатними.

Матриця Гурвіца для полінома третього порядку:

$$H = \begin{pmatrix} a_2 & a_0 & 0 \\ a_3 & a_1 & 0 \\ 0 & a_2 & a_0 \end{pmatrix}.$$

Головні діагональні мінори:

$$\Delta_1 = a_2;$$

$$\Delta_1 = 10 > 0 \text{ (умова виконується);}$$

$$\Delta_2 = \begin{pmatrix} a_2 & a_0 \\ a_3 & a_1 \end{pmatrix} = a_2 a_1 - a_3 a_0;$$

$$\Delta_2 = 10 \cdot 51,5915 - (1,645 \cdot 10^{-4}) \cdot 7,9575;$$

$$\Delta_2 = 515,91 > 0 \text{ (умова виконується);}$$

$$\Delta_3 = \begin{pmatrix} a_2 & a_0 & 0 \\ a_3 & a_1 & 0 \\ 0 & a_2 & a_0 \end{pmatrix} = a_0 (a_2 a_1 - a_3 a_0);$$

$$\Delta_3 = 7,95 \cdot 515,91;$$

$\Delta_3 = 4100,7$ (умова виконується).

На основі проведених розрахунків, з використанням обраних типових значень коефіцієнтів регуляторів, всі умови критерію Гурвіца виконуються, це свідчить про те, що система управління верстатом є стійкою.

4.8 Охорона праці

Впровадження гнучкої виробничої ділянки (ГВД) – це сучасний крок для оптимізації виробництва, але він також вимагає особливої уваги до питань охорони праці. Нові технології, обладнання та організація роботи створюють нові виклики та ризики, які необхідно враховувати для забезпечення безпеки та здоров'я працівників.

Охорона праці при впровадженні ГВД ґрунтується на загальних принципах трудового законодавства України. Основною метою є створення безпечних умов праці та мінімізація професійних ризиків. До основних нормативно-правових актів, що регулюють ці питання належать:

- закон України «Про охорону праці»;
- державні стандарти України (ДСТУ) та державні будівельні норми (ДБН), що стосуються проектування виробничих будівель, розміщення обладнання, організації робочих місць та системи безпеки;
- санітарні норми та правила, що регулюють мікроклімат, освітлення, шум, вібрацію та інші фактори виробничого середовища;
- правила безпеки при роботі з конкретними видами обладнання та технологій, що використовується в ГВД;
- нормативно-правові акти, що регулюють пожежну та електробезпеку на виробничих об'єктах.

Роботодавець несе відповідальність за забезпечення відповідних стандартів безпеки, навчання працівників, проведення інструктажів та оформлення необхідної документації.

Впровадження ГВД може створювати нові або посилювати існуючі ризики:

- механічні небезпеки;
- електричні небезпеки;
- термічні небезпеки;
- хімічні небезпеки;
- шум та вібрація;
- ергономічні ризики;
- психофізіологічні навантаження;
- пожежа та вибухонебезпека.

При організації робочих місць ключовим є безпека та ергономіка. Важливо забезпечити достатній простір для пересування та обслуговування обладнання. Освітлення має бути рівномірним, без відблисків, з можливістю регулювання. Необхідно утримувати оптимальний мікроклімат робочих місць та мінімізувати шум і вібрацію за допомогою ізоляції, надаючи працівникам відповідні засоби безпеки.

Кожен працівник має бути забезпечений відповідним ЗІЗ (засоби індивідуального захисту), а їхнє користування контролюватися. Регулярні медичні огляди є обов'язковими для тих, хто працює з потенційно шкідливими факторами. Необхідно розробити та відпрацювати аварійні процедури для НС (надзвичайна ситуація). Для зниження психоемоційного навантаження слід враховувати ротацію завдань та забезпечувати психологічну підтримку. Постійний моніторинг та контроль за дотриманням вимог охорони праці, регулярні перевірки та аудити є запорукою успішної та безпечної роботи персоналу.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було здійснено аналіз, проектування та реалізацію віртуального макета гнучкої виробничої дільниці (ГВД) як інтерактивної системи моделювання виробничих процесів.

Спочатку був проведений аналіз існуючих гнучких виробничих дільниць, їх класифікація, процес впровадження в системи виробництва, плюси та мінуси їх впровадження. Був здійснений вибір середовища, мови програмування та необхідних ресурсів для виконання поставленої задачі.

Після чого були сформульовані вимоги до функціональної структури макета, визначено його технічну та архітектурну елементу базу. Проведено обґрунтування вибору технічних компонентів, таких як однобалочний мостовий кран, модулі обробки деталей. Також були описані алгоритми взаємодії між компонентами системи, що відтворюють логіку роботи реальної виробничої дільниці. Значну увагу приділену моделюванню процесів транспортування, позиціонування, обробки деталей та автоматизованого контролю.

Далі було реалізовано програмну модель віртуального макета із використанням середовища Visual Studio та графічної бібліотеки OpenGL. Розроблену модульну архітектуру коду, яка включає в себе візуальні модулі, модулі логіки керування, інтерфейс взаємодії з користувачем та відображення стану компонентів систем. Візуалізація виробничих процесів забезпечена за допомогою анімацій та OpenGL-графіки, що дозволяє наочно продемонструвати роботу кран балки та обробних верстатів. Реалізовано інтерфейс керування, який дозволяє керувати симуляцією в режимі реального часу.

В процесі виконання кваліфікаційної роботи було досягнуто наступних результатів:

- проведено аналіз існуючих гнучких виробничих дільниць (ГВД);
- визначено вимоги до функціональної структури віртуального макету;
- підібрано відповідні технічні компоненти;

- розроблено алгоритми функціонування системи та взаємодії між компонентами;
- реалізовано графічне відображення виробничих процесів із використанням можливостей Visual Studio та графічної бібліотеки OpenGL;
- забезпечено інтерактивність за допомогою елементів керування та логіки обробки подій;
- проведено обчислення необхідного крутного моменту, який повинен розвивати серводвигун для нормальної роботи верстата MSF6090;
- визначена стійкість системи верстата MSF6090 за критерієм Гурвіца.

Таким чином, результати проведеної роботи підтверджують створення ефективного навчального засоба для ознайомлення з роботою гнучких виробничих систем та подальшого розширення програми.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. ДСТУ 3008:2015. Документація. Звіти у сфері науки та техніки. Структура та правила оформлення. – Введ. 2015–06–22. – К.: Держстандарт України, 2017. – 29 с..

2. Методичні вказівки з підготовки кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології освітньої програми «Системна інженерія» / Упоряд.: І.Ш. Невлюдов, О.М. Цимбал, О.В.Токарева, А.І. Бронніков. Харків: ХНУРЕ, 2022. 66 с.

3. Стеценко К. В. Функціонування гнучких виробничих систем. Збірник студентських наукових статей «Автоматизація та приладобудування» «Automation and Development of Electronic Devices». 2022. ADED-2020, (Вип. 2). С. 185–187.

4. Стоцько З. А. Автоматизація виробничих процесів у машинобудуванні та приладобудуванні. Academic Journals and Conferences |. URL: <https://science.lpnu.ua/sites/default/files/journal-paper/2018/feb/7785/1-2.pdf> (дата звернення: 18.04.2025).

5. Медведєв В., Тулупов В. І. Структура ГВС, 2020. URL: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSuY_7_p9FJqnodVrSYfrXjU3rpE3_EreQXsg&w=1000&h=1000 (дата звернення: 19.04.2025).

6. Підручники для вузів онлайн. URL: https://pidru4niki.com/imag/manag/kap_opt/image038.jpg (дата звернення: 18.04.2025).

7. Ю.Д. Абрашкевич, Л.Є. Пелевін, О.А. Марченко Процеси гнучких виробництв С. 43.

8. Берест Б. Р., Гурін Д. В. Актуальність віртуалізації гнучких виробничих ділянок на виробництві / МАТЕРІАЛИ ІІ Всеукраїнської конференції «Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки»

(Computer-integrated technologies, automation and robotics). Харків: ХНУРЕ, 2025. (с.30).

9. Берест Б. Р., Гурін Д. В. Дослідження використання гнучких виробничих систем та їх класифікація / Збірник студентських наукових статей «Автоматизація та приладобудування» «Automation and Development of Electronic Devices» ADED-2025 (Випуск 1). Харків: ХНУРЕ, 2025. (с.189).

10. Шафранська Т., Масленко К. С. Переваги застосування гнучких виробничих систем. Education and Science. URL: https://www.rusnauka.com/46_NIO_2016/Economics/10_220374.doc.htm (дата звернення: 19.04.2025).

11. Медведєв В., Тулупов В. І. Структура ГВС, 2020. URL: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSuY_7_p9FJqnodVrSYfrXjU3rpE3_EreQXsg&w=1000&h=1000&itd=1 (дата звернення: 19.04.2025).

12. Горизонтальна обробна гнучка автоматизована лінія DZHХ–001. Центр технічного обслуговування Кварц. URL: https://images.prom.ua/3176244489_w600_h600_3176244489.jpg (дата звернення: 19.04.2025).

13. Гнучка виробнича дільниця для збирання автомобілів, 2022. Wikimedia Commons. URL: https://upload.wikimedia.org/wikipedia/commons/thumb/2/28/Final_assembly_3.jpg/250px-Final_assembly_3.jpg (дата звернення: 19.04.2025).

14. Journaleco P. Завод airbus в США, 2025. JournalEco : les dernières actualités économiques et financières. URL: <https://www.journaleco.ma/wp-content/uploads/2025/03/aeronautique.jpg> (дата звернення: 19.04.2025).

15. RoboDK | Івано-Франківський національний технічний університет нафти і газу. ІФНТУНГ | Івано-Франківський національний технічний університет нафти і газу. URL: <https://nung.edu.ua/partner/robodk> (дата звернення: 19.04.2025).

16. Учасники проєктів Вікімедіа. Microsoft visual studio – вікіпедія. Вікіпедія.

URL: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio (дата звернення: 09.05.2025).

17. OpenGL - the industry standard for high performance graphics. OpenGL - The Industry Standard for High Performance Graphics. URL: <https://www.opengl.org/> (date of access: 19.04.2025).

18. Однобалкові крани Eot. DGCRANE. URL: <https://www.dgcrane.com/uk/products/single-girder-eot-cranes/> (дата звернення: 09.05.2025).

19. Універсальний токарний верстат CORMAK TORNADO 360 - токарні верстати по металу - cormak - kma.ua. КМА.ua. URL: https://kma.ua/tokarni-verstati-po-metalu/1511-universalnij-tokarnij-verstat-cormak-tornado-360.html?utm_source=google&utm_medium=cpc&utm_campaign=pmax_vid_20000_uah&utm_gad_source=1&utm_gad_campaignid=20675109495&utm_gbr_aid=0AAAAAC5qRJR2axDzBAymDtq1MIHvHr3y6&utm_gclid=Cj0KCQjwrPHABhCIARIsAFW2XBOIgm_pFOwL5LtZzJsVVfNWuO_dMxA05xhK2C7FQA7zKb5C32cuJYMaAjVUEALw_wcB (дата звернення: 09.05.2025).

20. Гравіровально-фрезерний верстат з ЧПУ MSF6090 на серводвигунах. MIRSTANKOV.COM. URL: <https://mirstankov.com/uk/catalog/hraviruvalno-frezernyy-verstat-z-chpu-msf6090-z-vakuumnym-stolom/> (дата звернення: 09.05.2025).

21. Однобалочний мостовий кран. Дошка безкоштовних приватних оголошень Salexy.kz. URL: https://salexy.kz/images/img_kz/124x124/aa/04/aa04c18c44132d18e2b0353f6e1bd73.webp (дата звернення: 09.05.2025).

22. Фрезерний верстат з ЧПУ MSF6090. MIRSTANKOV.COM. URL: https://mirstankov.com/wp-content/uploads/2021/11/Frezernyy_stanok_s_CHPU_MSF6090_s_vakuumnym_stolom.jpg (дата звернення: 09.05.2025).

23. Універсальний токарний верстат CORMAK TORNADO 360. Верстати по дереву та металу, будівельна та садова техніка світових брендів.

URL: https://kma.ua/12755-thickbox_default/universalnij-tokarnij-verstat-cormak-tornado-360.jpg (дата звернення: 08.05.2025).

24. Li Q., Wei L. Research and design of high-performance DSP control system based on frequency control. Highlights in science, engineering and technology. 2022. Vol. 7. P. 151–156. URL: <https://doi.org/10.54097/hset.v7i.1037> (date of access: 05.06.2025).

25. Velocity mode control for servo drives - ADVANCED motion controls. ADVANCED Motion Controls. URL: <https://www.a-m-c.com/experience/technologies/operating-mode/velocity-mode/> (date of access: 05.06.2025).

26. Серводвигун - знання. Китайський безщітковий двигун постійного струму, двигун без сердечника, постачальники, виробники, фабрика концентраційних двигунів - VSHIDA. URL: <https://ua.vsdmotor.com/info/what-is-a-servo-motor-102242807.html> (дата звернення: 05.06.2025).

27. Jones N. Servo motor guide. Anaheim Automation - Your source for Stepper Motor, Brushless DC Motor, DC Motor, and Planetary Gearbox Products. URL: <https://anaheimautomation.com/blog/post/servo-motor-guide> (date of access: 05.06.2025).

28. DC servo motor: construction, working & its applications - schneider electric. Schneider Electric. URL: <https://eshop.se.com/in/blog/post/dc-servo-motor-construction-working-its-applications.html> (date of access: 05.06.2025).