

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя  
(повне найменування вищого навчального закладу)  
Факультет прикладних інформаційних технологій та електроінженерії  
(назва факультету)  
Кафедра автоматизації технологічних процесів та виробництв  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

**магістр**

(освітній рівень)

на тему: Автоматизація конструювання робототехнічної комірки  
обробки циліндричних деталей

Виконавли: студенти 6 курсу, групи КАМ-61

Спеціальність 151

*“Автоматизоване управління технологічними процесами”*

(шифр і назва спеціальності)

Загоруйко О.А.

(підпис)

(прізвище та ініціали)

Керівник

Михайлишин Р.І.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Козбур І.Р.

(підпис)

(прізвище та ініціали)

Завідувач кафедри

Савків В.Б.

(підпис)

(прізвище та ініціали)

Рецензент

Золотий Р.З.

(підпис)

(прізвище та ініціали)

м. Тернопіль – 2020

## **А н о т а ц і я**

В дипломній роботі використані такі терміни: інструмент, захоплюючий пристрій, промисловий робот, зварювання, програмування, циліндричний об'єкт.

Об'єктом дослідження є процес автоматичної обробки циліндричних об'єктів промисловим роботом з різними параметрами.

Мета роботи – є розробка та тестування програми автоматичного конструювання робототехнічної станції обробки циліндричних деталей, яка використовується для полегшення програмування робота АВВ, що включає різні елементи: апаратне та програмне забезпечення.

Дана дипломна робота полягала в розробці різних інструментів для підтримки швидкого розвитку концептуальних робототехнічних станцій. Доцільність нових деталей є невід'ємною частиною конструкції роботизованих зварювальних комірок і вимагає концепції станції. Багато незалежних розробників використовують віртуальне моделювання для зменшення ризиків, пов'язаних з роботою з новими деталями в робототехнічних системах. Ці розробники бажають швидкої та точної програми для тестування над робототехнічними станціями, обладнаними різними зварювальними аксесуарами та позиціонерами роботів. Поточне програмне забезпечення, яке використовують ці розробники, вимагає підвищення кваліфікації та великих часових вкладень. З цих причин був створений швидкий і точний спосіб створення нових концептуальних комірок. Це було зроблено завдяки розробці запрограмованих надбудов та інтелектуальних компонентів у АВВ RobotStudio.

The diploma thesis uses the following terms: tool, capturing device, industrial robot, welding, programming, cylindrical object.

The object of research is the process of automatic processing of cylindrical objects by an industrial robot with different parameters.

The purpose of the work is to develop and test a program for automatic design of a robotic station for processing cylindrical parts, which is used to facilitate the programming of the ABB robot, which includes various elements: hardware and software.

This thesis was to develop various tools to support the rapid development of conceptual rootechnical stations. The feasibility of new parts is an integral part of the design of robotic welding cells and requires the concept of the station. Many independent developers use virtual simulation to reduce the risks associated with working with new parts in robotic systems. These developers want a fast and accurate program for testing robotic stations equipped with various welding accessories and robot positioners. The current software used by these developers requires training and a lot of time. For these reasons, a fast and accurate way to create new conceptual cells has been created. This was done through the development of programmed add-ons and intelligent components in ABB RobotStudio.

## ЗМІСТ

ВСТУП.....	6
1 АНАЛІТИЧНА ЧАСТИНА.....	9
1.1 Аналіз сучасного стану проблем управління ..... промисловими роботами.....	9
1.1.1 Виконавчі механізми.....	11
1.1.2 Приводи та передавальні механізми .....	12
1.1.3 Інформаційно вимірювальні системи.....	13
1.1.4 Системи управління .....	13
1.2 Інтеграція та нові велики в робототехніці, автоматизоване моделювання робототехнічних комірок.....	15
1.3 Позиціонування маніпуляторів з ангулярною системою координат.....	24
1.4 Адаптивне програмування робототехнічних комірок.....	29
2 ТЕХНОЛОГІЧНА ЧАСТИНА.....	36
2.1 Поточна реалізація інтелектуальних компонентів та надбудов.....	36
2.2 Технічні характеристики промислових роботів .....	42
2.2.1 Частина роботизованого маніпулятора.....	42
2.2.2 Програми .....	45
2.2.3 Кінематика робота .....	49
2.2.4 Ітеративний контроль навчання.....	54
3 КОНСТРУКТОРСЬКА ЧАСТИНА .....	57
3.1 Розробка автоматичної огрожі.....	57
3.2 Автоматичний конструктор доріжок.....	63
3.3 Конструювання автоматичних легких завісів робототехнічної комірки ..	68
3.4 Конструювання додатку.....	72
4 НАУКОВО-ДОСЛІДНА ЧАСТИНА.....	76
4.1 Створення автоматичного конструктора робототехнічної комірки.....	76
4.2 Створення робототехнічної комірки без надбудови .....	80
4.3 Створення робототехнічної комірки з надбудовою .....	92
4.4 Створення робототехнічної комірки автоматично .....	100
5 СПЕЦІАЛЬНА ЧАСТИНА .....	105
5.1 Створення розумної робототехнічної комірки .....	105
6 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	130
6.1 Значення охорони праці в забезпеченні здорових умов праці .....	130
6.2 Охорона праці як система заходів щодо гармонізації використання комп'ютерних технологій.....	131
6.3 Аналіз потенційних небезпек та шкідливостей виробничого середовища .....	133

6.4 Розрахунок освітленості робочого місця при розробці криптографічних систем.....	138
6.5 Електромагнітний імпульс ядерного вибуху і захист від нього радіоелектронних засобів .....	139
6.6 Забезпечення нормальних умов праці.....	142
6.6.1 Вибір приміщення .....	142
6.6.2 Забезпечення нормальних санітарно- гігієнічних умов на робочому місці.....	143
ВИСНОВКИ.....	148
ПЕРЕЛІК ПОСИЛАНЬ .....	150
ДОДАТКИ .....	<b>Ошибка! Закладка не определена.</b>

## ВСТУП

Перехід народного господарства України на шлях інтенсивного розвитку – об’єктивна вимога часу. Збільшення випуску продукції машинобудування і покращання її якості при одночасному зниженні виробничих затрат забезпечується за рахунок удосконалення існуючих і впровадження нових видів обладнання і технологічних процесів, засобів їх механізації і автоматизації, за рахунок підвищення ефективності організації і управління виробництвом.

Для збільшення швидкості офлайн-програмування робото технічної комірки була створена серія розумних компонентів та надбудов. Кожен розумний компонент містив необхідну геометрію для створення графічної складової. Розумні компоненти мають можливість створювати та модифікувати механізми, завдяки чому графічні компоненти візуально відображатимуть функцію, яку виконує механізм. Розумні компоненти також мають можливість завантажувати несуттєву геометрію та прикріплювати її до належного кадру, щоб зробити станцію якомога точнішою під час досліджень охоплення та автономного програмування. Кожен інтелектуальний компонент був запрограмований у Microsoft Visual Studio C # та взаємодівав з RobotStudio за допомогою візуального програмування, увімкненого творцем інтелектуального компонента. Змінні, передані між станцією та інтелектуальним компонентом .dll, були скомпільовані в програму через документ .xml.

Надбудова була створена в Microsoft Visual Studio C #, щоб дозволити користувачеві швидко завантажити позиціонер та бажаний райзер. Позиціонер створений у правильному положенні для калібрування віртуального

контролера. Головний запас та хвостовий запас вибираються на основі бажаної потужності та висоти стояка; також доступна можливість поєднання запасу голови та хвоста з підйомом. SkyHooks вибираються на основі потужності, падіння та кидка. Відповідний стояк завантажується на основі більших вимірювань падіння і кидка; це вірно лише тоді, коли SkyHook з підйомом не вибрано. Коли завантажується Drop Center, користувачеві надається можливість вибрати стояк.

Серія розумних компонентів складається з трьох частин. Перший - це автоматичний будівельник огорожі, який дозволяє користувачеві вибрати бажану довжину та висоту огорожі та побудувати його на станції. Другий розумний компонент - це конструктор доріжок. Цей інтелектуальний компонент завантажує бажану довжину колії, козлової частини на основі вибраної висоти та довжини стріли. Користувач також має можливість завантажувати аксесуари та автоматично розміщувати та прикріплювати їх до належного положення на козлі. Цей розумний компонент також складає козловий механізм з висоти вибраної вежі та довжини колії проїзду.

Останнім розумним компонентом є автоматичний компонент світлової завіси. Цей компонент дозволяє користувачеві створити кілька світлових завіс та автоматично побудувати віртуальну лазерну огорожу, яка розміщується між двома стовпами світлових завіс.

Потім компонент Automatic Station Builder був побудований для використання раніше розроблених інтелектуальних компонентів. Цей компонент отримує вагу, довжину та діаметр деталі та самостійно створює робототехнічну зварювальну станцію. Кожна станція побудована з наступних частин: огороження, світлової завіси, позиціонера робота та позиціонерів

деталей. Кожен компонент правильно розміщений відповідно до стандартів очищення.

Кожен розумний компонент має індивідуальний посібник користувача. Посібник користувача охоплює встановлення та використання кожного компонента. Посібники користувача були створені з метою пришвидшити процес навчання нових компонентів, хоча вони досить інтуїтивні. Також були створені посібники з технічного обслуговування для підтримки оновлень інтелектуальних компонентів протягом усього їх життєвого циклу. Також було створено посібник користувача для надбудови; однак жодного посібника з технічного обслуговування не було створено через складність побудови надбудови. Поки не складено жодного посібника з технічного обслуговування, бібліотеки можна оновити до новіших моделей САПР, якщо не змінюються основні розміри.



# 1 АНАЛІТИЧНА ЧАСТИНА

## 1.1 Аналіз сучасного стану проблем управління промисловими роботами

Робототехнічна система (РТС) - це технічна система, в якій перетворення і зв'язку енергії, маси та інформації відображені з використанням роботів різного призначення [1].

РТС володіють основними ознаками системності: цілісністю системи машин, що виявляється в тому, що вона має властивості, відмінними від властивостей складових її елементів; ієрархічністю структури, що дається ознаки в наявності декількох рівнів ієрархії і можливістю опису РТС з різним ступенем деталізації, яка визначається розглянутим рівнем залежно від розв'язуваних завдань. Так, окремі складові РТС, виконують самостійні функції, можна розглядати при диференціюванні в якості самостійних складних систем і, навпаки, в процесі інтегрування можна прийти від системи нижчого рівня до вищого.

Існують різні класифікації роботів, прийнято виділяти два основні класи маніпуляційні роботи та мобільні роботи.

Маніпуляційні РТС отримали найбільший розвиток і практичне застосування в промисловості, де їх називають промисловими роботами. Промисловий робот (ПР) - автоматична машина, стаціонарна або пересувна, що складається з виконавчого пристрою у вигляді маніпулятора, що має кілька ступенів рухливості, і перепрограмовуваного пристрою програмного управління для виконання у виробничому процесі рухових і керуючих функцій (по ДСТУ 25686-85). У технічній літературі часто зустрічається і більш

коротке визначення: ПР - перепрограмований автоматичний маніпулятор промислового застосування. Характерними признаками ПР є автоматичне керування; перепрограмовуваність - здатність до швидкої автоматичної або за допомогою людини-оператора заміні керуючої програми (до зміни послідовності, системи та змісту команд); здатність до виконання трудових дій. Існують різні класифікації роботів, наприклад роботи маніпулятори можуть бути класифіковані залежно від компонентів з яких складаються, конфігурації та використання, для роботів маніпуляторів найбільш поширені три класифікації за типами систем управління, типам приводів, і формі робочої зони [2].

Мобільні (рухомі) РТС забезпечують автоматичне переміщення корисного навантаження в просторі. Можуть мати запрограмований маршрут руху і (або) автоматичне адресування мети. Оснащуються рушіями різних типів: колісними, гусеничними, крокуючими, водометними, гвинтовими, ракетними і т.п. У промисловості застосовуються для обслуговування складів, міжцехового і внутрішньоцехового транспортування матеріалів, деталей, інструменту та оснастки. Такі рухливі РТС часто оснащуються маніпуляторами.

В літературі про крокуючих роботів часто можна зустріти терміни "локомоція" синергія ". Локомоція - пересування крокуючого пристрою будь-яким способом, шляхом ходьби, стрибків або будь-якого їх поєднання. Синергія - упорядкована послідовність дій приводить до локомоції [3].

Серед робототехнічних систем, тобто керованих систем твердих тіл, помітну роль відіграють крокуючі пристрою або, як їх ще часто називають, локомоційні системи (ЛС). На відміну від маніпуляційних локомоційні пристрої являють собою сукупність відкрито-замкнених ланцюгів (кінцівок),

за допомогою яких здійснюється пересування ЛЗ в просторі. Крокуючі роботи мають ширші можливості для переміщення по пересіченій місцевості ніж колісні або гусеничні машини. Обмеженням крокуючих машин є складна конструкція рушія і як наслідок складна система управління рухами робота.

Не залежно від галузі застосування і використовуваних виконавчих механізмів РТС складається і наступних компонентів:

- Виконавчі механізми;
- Приводи та передавальні механізми;
- Інформаційні системи, що складаються їх різних; датчиків і вимірювальних систем;
- Системи управління.

Важливо відзначити що сучасні РТС можуть обладнуватися сотнями ступінь свободи, а частоти роботи систем управління 200 Гц [4, 5] при управління рухом таких складних систем необхідно враховувати безліч умов. Тому завдання розробки систем управління рухом таких роботів є актуальною.

### **1.1.1 Виконавчі механізми**

Маніпулятори роботів і деталі рушіїв приводяться в дію за допомогою виконавчих приводів, які забезпечують рухливість системи та виконання цільових функцій.

У біологічних організмах функції виконавчих органів покладаються на м'язи, які складаються з пучків скорочуються волокон.

Середній час скорочення одного волокна сто мілі-секунд, а створюване зусилля може досягати двохсот грамів. Посилення, створюване всім м'язом,

визначається числом задіяних волокон, а плавність рухів забезпечується їх послідовним підключенням. В цілому м'яз являє своєрідний біологічний двигун, що включає кілька тисяч Простіший двигунів імпульсного типу. Всі вони об'єднуються в пучки, а потім - в рухові одиниці.

Приводи роботів класифікуються по ряду характерних ознак. Залежно від типу енергії, використовуваної двигуном, вони можуть бути електричними, гідравлічними і пневматичними.

### **1.1.2 Приводи та передавальні механізми**

Пристроїв передачі механічної енергії від двигуна до ланок механічним елементам РТС досить багато:

- Кулько-гвинтова передача;
- Зубчато-рейкова передача;
- Хвильової редуктор;
- Ланцюгові передачі, стрічки та приводні ремені;
- Шарнірно-важільні передачі.

Часто в якості приводів використовують механотронні модулі, які частково об'єднують в собі виконавчі та інформаційні компоненти РТС. Механотронні модулі - це базові функціональні компоненти механотронних систем і машин з комп'ютерним управлінням, призначені для виконання рухів, як правило, по одній керованій координаті. Головними перевагами механотронних систем є виняток багатоступінчастого перетворення енергії та інформації, спрощення кінематичних ланцюгів і отже висока точність і

поліпшені динамічні характеристики, конструктивна компактність модулів і отже поліпшені масо-габаритні характеристики. Можливість об'єднання мехатронних модулів в складні механотронні системи, і комплекси, що допускають швидку реконфігурацію, відносно низька вартість установки, настройки і обслуговування системи, завдяки модульності конструкції, уніфікації апаратних і програмних засобів, здатність виконувати складні рухи, завдяки застосуванню методів адаптивного та інтелектуального управління.

### **1.1.3 Інформаційно вимірювальні системи**

Головна задача інформаційно вимірювальної системи, це забезпечення інформацією систему управління. Можна виділити два типи даних одержуваних від інформаційно системи. Відповідно до рівнів ієрархії системи управління для тактичного рівня це інформація про об'єкт маніпулювання і про навколишнє робочому середовищі, для тактичного рівня це дані про стан виконавчої системи, переміщення в приводах, стан різних систем, гальмівних муфт, положення кінцевих вимикачів і т.д. Інформація про стан виконавчої системи необхідна для організації зворотного зв'язку.

### **1.1.4 Системи управління**

Об'єктом управління є різні виконавчі механізми - маніпуляційні, пересування і системи управління. Загалом механічна система як об'єкт управління може бути охарактеризована: типом і числом переносних і орієнтуючих робочих органів ступенів рухливості (переносні - поступальний і обертальні, що орієнтують - обертальні), типом і розміром робочої зони.

Виконавчі механізми складаються з механічної системи і приводів. Для маніпуляторів механічна система буде закінчуватися яким або робочим органом або інструментом. Для систем пересування також можуть мати робочий орган, наприклад бульдозерний ніж, підйомник, лебідка і т.д. також в якості робочого органи можуть виступати елементи крокуючого рушія, наприклад стопи [6] [7].

Система управління має ієрархічну організацію в якій можна виділити такі рівні [3]:

1. Формулювання завдання. Система управління апаратом отримує завдання від оператора (наприклад пройти з пункту А в пункт В).

2. Планування поведінки та прийняття рішення. Алгоритми цього рівня виконують аналіз ситуації і здійснюють підбір оптимального режиму руху, що забезпечує досягнення поставленої мети з урахуванням перешкод і конструктивних обмежень.

3. Рівень побудови руху та управління приводами. Алгоритми цього рівня будують програмні руху окремих ланок апарат т.п. закон зміни їх узагальнених координат.

Частіше перші два рівня ієрархії об'єднують в один рівень і тоді задачу управління рухами робота можна розділити на два рівня тактичний і виконавчий. На тактичному рівні визначається траєкторії і цілі руху. Виконавчий рівень вирішує задачу генерації управляючих сигналів забезпечують необхідний рух [8].

Основне завдання виконавчої частини системи управління мобільним роботом - це управління рухом. Сучасний підхід полягає в тому щоб управляти рухом в кожному зчленуванні за допомогою вбудованого механізму [9, 10].

Окремо варто виділити управління локомоційним рухом. Локомоційний рух-це пересування динамічних керованих систем в просторі. Проблема синтезу локомоцій головним чином досліджувалася стосовно до крокуючим роботам. Складні завдання організації систем управління крокуючих робототехнічних пристроїв становлять значний інтерес з точки зору моделювання та конструювання різних засобів протезування кінцівок (екзоскелетони), проникнення в важкодоступні, небезпечні або заражені місця, освоєння інших планетарних просторів [11].

## **1.2 Інтеграція та нові велики в робототехніці, автоматизоване моделювання робототехнічних комірок**

За останні сорок років, промислові роботи пройшли шлях від планування – розуміння – розробки до більш надійного, адаптованого управління роботами: зокрема, інтеграція тіла, датчиків і програмного забезпечення за допомогою штучного інтелекту (AI) [Kawamura, 2006]. Штучний інтелект запровадили не тільки в передових промислових роботах, але і не в промислових роботах. Починаючи від роботів в розважальній сфері і до роботів, які спостерігають за станом здоров'я, усі вони використовують штучний інтелект і очікується, що ця тенденція продовжиться. У цій статті автор описує і показує, які потрібно докласти зусилля для вирішення проблеми досягнення інтеграції тіла і розуму. Це досягається шляхом реалізації пізнавального робота за допомогою когнітивного контролю: уваги, емоцій і адаптивної системи робочої пам'яті. За останні роки, різновидності промислової робототехніки та автоматизації також отримала багато

нововведень. Реалізація роботів з когнітивними навичками буде проблемою не тільки для наукових кіл, але і для інженерів-технологів.

Зокрема часто описується когнітивні архітектури, які вивчаються в США. Вчені вивчають людське пізнання шляхом побудови моделей, які допомагають пояснити функції мозку, використовуючи психологічні і нейронні дослідження. За останні десятиліття різні когнітивні архітектури та системи були розроблені американськими вченими, щоб краще зрозуміти людське пізнання. Далі коротко приведена характеристика когнітивної архітектури. Перші дві архітектури були обрані через їх популярність в США. Третя архітектура була обрана, як приклад системи, що включає людське сприйняття і моторні аспекти в більш конкретних способах аналізу складних когнітивних завдань, таких як керування роботом і т.п.

Також пояснюється, що набули широкого застосування багатоагентні системи. У робототехніці, термін "агент" зазвичай використовується для позначення автономного об'єкта, який здатний діяти в середовищі і з іншими агентами. Це може бути робот, людина або навіть програмний модуль. Термін «агент» вперше використав вчений Minsky [Minsky, 1985]. Термін «Багатоагентна система» (MAS) - система з багатьма агентами - стає все більш і більш популярним в області штучного інтелекту. Також система набуває популярності у спільнотах, які займаються мобільними роботами (де часто називають цю систему мультироботом). За основу гуманоїда було прийнято багатоагентну систему, до переваг якої відносилась простота модульного розвитку. Додавши кілька датчиків і виконавши механізми і необхідні інтеграції таких як людських, отримали єдиного робота гуманоїда [Kawamura та ін, 2000].



Створена когнітивна архітектура багатоагентної системи, яка інкапсулює: функції обладнання, регулятори низького рівня і сенсори основного обробітку. Ця абстракція деталей дозволяє програмуванню відбуватися на рівні цілеспрямованих дій та екологічних характеристик де: приводи витісняються діями, а сировинні сенсорні дані замінені функціями. Кожна функція має багато абстракційних ключів. Ці абстракційні ключі здібностей реалізовані з використанням програми (ІМА) агентів. Кожен агент взаємодіє з відповідним апаратним ресурсом і забезпечує цей ресурс інтерфейсом для управління іншими агентами. Система сприйняття включає в себе ряд датчиків. Кожному датчику призначений агент ІМА, який обробляє сенсорні входи і зберігає інформацію в залежності від типу сприйняття агента. Для візуальних входів є візуальні засоби, які виконують кодування, сприйняття, локалізацію об'єкта, виявлення руху або розпізнавання осіб. Інші входи включають в себе звукові локалізації та агентів, що розпізнають звук. На більш високому рівні пізнавальні здібності реалізовані за допомогою двох складових агентів самоагентування і людського фактору. Самоагентування це структура агент в агенті. Самоагентування несе відповідальність в основному за виконання завдання, а людський фактор, який представляє собою людину, пояснює дії, що взаємодіють в даний час.

Структури пам'яті використовуються, щоб допомогти підтримувати інформацію, необхідну для нагальних завдань і збереження подій, які можуть бути використані під час прийняття рішень. Довготривала пам'ять (LTM) зберігає інформацію, таку як отримання навичок, семантичні знання і минулий досвід (епізоди) для пошуку в майбутньому. В рамках LTM, процедурна пам'ять (PM) запам'ятовує примітивні рухи та поведінку, необхідну для дій, наприклад, як дістатися до кінцевого пункту призначення. Також існує

поведінка отримана за допомогою методу просторово - часових карт Iso [Jenkins & Mataric, 2003]. Семантична пам'ять (SM) являє собою структуру даних про об'єкти навколишнього середовища. Епізодична пам'ять (EM) зберігає минулий досвід в тому числі цілей і дій, які здійснювались в минулому. Система робочої пам'яті (WMS) моделюється після робочої пам'яті, як у людини, яка зберігає обмежене число інформації, необхідної для виконання завдання, наприклад, номер телефону під час його набору. Це дозволяє роботіві приділити увагу найбільш актуальним особливостям поточного завдання, яке тісно пов'язане з навчанням і виконанням завдань.

Крім того звертається увагу, що когнітивний контроль в організмі людини є частиною функцій виконавчої волі (таким, як планування і абстрактне мислення) під час роботи лобових часток людського мозку [Stuss, 2002]. Когнітивне управління це "здатність свідомо керувати думками і поведінкою за допомогою уваги із суперечливими цілями і вимогами" [O'Reilly та ін, 1999] [MacLeod і Sheehan, 2003]. Когнітивний контроль дозволяє людям перешкоджати відволіканню і зосередитися на завданні, включаючи зміну різних завдань. На думку дослідників в галузі неврології, людський когнітивний контроль здійснюється через оперативну пам'ять в лобовій корі (PFC) [O'Reilly та ін, 1999]. Когнітивний контроль під час виконання завдань і перемикавання зобов'язує мозок виконувати виконавчі функції, включаючи:

- Звернення уваги на інформацію про завдання, пов'язану з введенням й оновленням інформації
- Блокування відволікання
- Перемикання між різного рівнями пізнання під час рутинних дій
- Комплекс обговорення
- Дізнавання нових відповідей під час нової ситуації

Вказується, що когнітивні роботи повинні мати можливість обробляти непередбачувані ситуації і навчитися виконувати нові завдання. Крім того, когнітивний контроль, який намагаються запровадити у робототехніці повинен діяти відповідно до стимулів в динамічному середовищі. Реалізація когнітивних функцій управління в даний час вирішується шляхом забезпечення інформації про завдання, пов'язаних з системою робочої пам'яті через стробування інформації, яка може бути передана в механізм прийняття рішень. Забезпечення інформації про завдання називається центральний виконавчий агент.

Крім того проводиться аналіз і зводить все до того, що будь – який когнітивний робот повинен бути в змозі використати як зовнішні, так і внутрішні стимули, щоб свідомо організувати свою поведінку. Відповідно до цього, емоції можуть стати одним з основних факторів під час прийняття рішень. Для того, щоб увага та емоції на основі організації процесу відбору були більш реалістичними, впроваджують змінювач часу, що відображає зміну внутрішніх станів з плином часу. У цьому типі процесу діє відбір, система не обов'язково повинна виконати ті ж дії щоразу з однаковим набором зовнішніх подразників. У когнітивній архітектурі, агент несе відповідальність за мету керування і його внутрішній стан [Sloman та ін., 2005]. Емоції агента будуть замінені, щоб вони були більш динамічними, для кращого відстежування внутрішнього стану системи.

В подальшому переходить до теми модульного робота. Модульний робот, що повторно конфігурується, складається з колекції індивідуальних зв'язків і об'єднує компоненти, які можуть асемблюватися в геометрії різних роботів. Порівняно з обумовленим промисловим роботом з фіксованою геометрією, така система може забезпечити гнучкість для користувача, щоб

покрити широкий спектр різних завдань через належний вибір і реконфігурацію великого інвентаря функціональних компонентів. Декілька макетів такої системи було продемонстровано у різних дослідницьких установах (Cohen і ін. 1992; Fukuda & Nakagawa 1988; Schmitz, і ін. 1988). Додатки модульних систем були запропоновані в швидких системах розгорнутого робота для ризикованої матеріальної обробки (Paredis і ін. 1995), де в просторі розташовані автономні системи (Ambrose 1995), у виробничих системах (Chen 2000; 2001). У контролі і моделюванні модульної автоматичної системи, що повторно конфігурується, потрібні точні кінематичні і динамічні моделі робота. Тобто класична кінематична і динамічна моделююча техніка для автоматичних маніпуляторів, або для робота з фіксованою геометрією. Ці моделі доводиться отримати вручну і індивідуально зберігати у автоматизованому диспетчері перед моделюванням і управлінням роботом. Комерційне програмне забезпечення автоматичного моделювання зазвичай забезпечує кінцевому користувачеві бібліотеку моделей існуючих роботів. Моделі будь – яких нових роботів, яких немає у бібліотеці доводиться отримувати виключно з параметрів і команд в пакеті. Для модульної автоматичної побудованої системи на стандартних модульних компонентах, можлива автоматична геометрія і за допомогою цього свобода руху стає величезною. Як показав Chen (1994), число роботизованих - конфігурацій зростає показово, коли модульний набір стає великим і модульне проектування ускладнюється, щоб отримати базу де зберігаються усі ці моделі. Оскільки бібліотечні функції вимагають не лише величезного зусилля, але і дуже великої кількості дискового простору: у таких випадках це непрактично і майже неможливо.

Представлено структуру, щоб полегшити модельне покоління для контролю і моделювання модульних роботів. Структура складається з трьох частин: складова база даних, представлення геометрії робота і незалежна від геометрії моделюючої техніки для кінематики, динаміки, і калібрування. Складова база даних підтримує опис і специфікації стандартних автоматичних компонентів, як наприклад приводи, жорсткі зв'язки, датчики, і крайні виконавчі елементи. Автоматичне представлення вказує типи і залежність автоматичних компонентів, що сполучаються. Геометрично незалежні моделюючі алгоритми потім роблять належні моделі з автоматичними описами. Також автор подає на розгляд граф для базування техніки. Граф для базування техніки, ним названо кінематичний граф, представлений для представлення послідовності зборів моделей і автоматичної геометрії. Модулі в ньому встановлюються або відокремлюються від робота. Також може бути вказаний близько додаючий або видаляючий вузол графа. Реалізація цього графа здійснюється через матрицю (Chen 1994; Chen & Burdick 1998). Модульний робот може запрацювати згідно з цією задачею без знання деяких параметрів, як наприклад об'єднані кути і вихідні позиції.

Зокрема дається інформація про попередню спробу працювати з автоматичним зразковим поколінням для модульних роботів, яку почали Denavit-Hartenburg (D-H) параметризацію автоматичного покоління (Kelmar & Khosla 1988; Benhabib і ін. 1989). Проте, метод D-H не забезпечує ясної відмінності між договірною послідовністю модулів в автоматичних ланцюгах і їх просторовими зв'язками. Також, це залежить від вихідної позиції робота: той же робот можливо має різні набори параметрів D-H, просто через різні вихідні або нульові позиції. Оцінюючи завдання виконання модульного робота відносно його відповідної геометрії, мають бути визначені набори параметрів,

щоб ідентифікувати унікальність автоматичної геометрії (Chen & Burdick 1998). Формулювання кінематики і динаміки засноване на теорії «Брехні» групи і алгебри «Брехні». Автоматична кінематика йде за місцевим представленням як експоненціальна (РОЕ) формула, в якій з'єднана як члени SE так і алгебри брехні Евкліда. Асоційована структура алгебри брехні може спростити конструкцію з диференціалів кінематичної функції до потрібного числового зворотного рішення.

Представлення РОЕ може також уникати оригінальності умови, яка часто відбувається в кінематичному калібруванні. Отже, це забезпечує метод для обробки зворотної кінематики відкаліброваної і не відкаліброваної автоматичної системи. Формулювання з динамічної моделі запускається за рекурсивним алгоритмом Ньютона - Ейлера (Hollerbach 1980; Rodriguez і ін. 1991). Узагальнена швидкість, прискорення, і сили, які виражаються в термінах лінійних дій над SE (Murray і ін. 1994). Відмічено, що усе запропоноване моделювання техніки може боротися з модульною автоматичною конфігурацією без надмірності.

За переконаннями автора, щоб візуалізувати і симулювати виконання завдання робота, що асемблює, наприклад досяжності робота і робочої області, пояснює, що знадобиться додаткове програмне забезпечення автоматичного моделювання. Моделювання довкілля для модульної автоматичної системи (SEMORS) - заснований в NT Windows орієнтований додаток програмного забезпечення, що розвивається для цієї мети. Додаток моделює і націлює дані структури. SEMORS пропонує узагальнювати і проводити автоматичне навантаження зразкової конструкції (кінематика, динаміка і калібрування) через комп'ютерне моделювання і в реальному часі контролювати довільні автоматичні конфігурацій (Chen і ін. 1999). Основа

графічного інтерфейсу користувача SEMORS призначена, щоб стати однорідним інтерфейсом для усіх модульних роботів і портативних модульних автоматичних систем від різних постачальників. Це дозволить використати систему для моделювання і для виконання он-лайн завдання, незважаючи на те, чи робот виконує операції, чи симулюється, щоб виконувати. Отже, це дозволяє користувачеві швидко об'єднувати апаратні компоненти в модульних роботах і, щоб управляти їх діями, що повторно конфігуруються. Ключові особливості SEMORS включають:

- Моделювання і автоматизування будівельника
- графічне завдання моделювання
- "Універсальну" зворотню кінематику
- Повну динаміку моделювання
- Траєкторію і завдання планування
- Прозору можливість з'єднання мережі модуля

Також представлено основні особливості технічного забезпечення і програмне забезпечення нещодавно задуманого робототехнічного модуля. Коротко розглядаються АІМ, асоційованої матриці і матриці шляху. Сформульовано і виконано незалежну геометрично діючу модель для модульних роботів. На додаток до автоматизованого зразкового покоління, було ототожнено зразок оптимальної модульної геометрії автоматичних зборів для специфіки завдання при великій базі даних. Мета представленого ототожнення полегшує процес пошуку і оптимізації під час користування генетичним алгоритмом наближення. Ставиться мета дати орієнтовану оптимальну геометрію виходів для модульних роботів, що повторно конфігуруються, і вирішення їх проблем. Запропонований автоматичний метод модельного покоління, що здійснює свою роботу в Windows Базузерах. Цей

додаток для модульної робототехнічної автоматизаційної системи, названий SEMORS (Моделювання Довкілля для Модульної Автоматичної Системи). Прототипи модульної робототехнічної автоматизаційної системи розмістили як серійно, так і паралельно геометрії заснованій на дії SEMORS.

### **1.3 Позиціонування маніпуляторів з ангулярною системою координат**

Промислові автоматичні маніпулятори являються по суті просторовими зв'язками, які представляються жорсткими тілами сполученими з'єднаннями. Хоча багато видів з'єднань доступні для використання в механіці, але тільки два типи використовуються для автоматичних маніпуляторів. Вони мають повторний завиток, чи ротацію з'єднань (R) і призматичний, або ковзальне з'єднання (P). Ці специфічні типи з'єднань дозволяють створити єдину степінь свободи. Тобто відносний рух між сусідніми тілами дозволяє рухатися і контролювати ці кінематичні пари. Кожне з'єднання на маніпуляторі самостійно управляється приданим йому двигуном. Також в маніпулятор застосовується схема, яка центрально під'єднується до кінематичного контролера маніпулятора, щоб вичислити набори моторів для водотонажного з'єднання у виконавчому кінцевому елементі. Математична процедура, яка спрямована на виконання цих дій часто вмикається до того як задіюється зворотній аналіз позиції. У цього аналізу є деякі труднощі у сфері роботехніки, оскільки він застосовує заплутану просторову геометричну техніку. Складність аналізу зростає істотно числу ротаційних з'єднань в структурі маніпулятора, тому значна частина виданих баз даних включає реєстр об'єднаних повторних завитків маніпуляторів.



Автор розкажує про видану літературу, що в ній були запропоновані різні методи для вирішення зворотної проблеми позиціювання маніпуляторів. Проте, самі промислові роботи проектуються з геометричними особливостями, отже зробити зворотне позиціювання можливим можна лише в закритій формі і там воно формується відповідним чином. Інший геометричний аспект приводить до спрощення зворотнього рішення. Наприклад до спрощення сферичного дизайну зап'ястка, який призводить до того, що останні три з'єднання на структурі маніпулятора перетинаються в одному місці. У цьому випадку є можливість оцінити маніпулятор, який складається з двох окремих частини, де перша частина (рука) складається з трьох з'єднань в стаціонарній структурі маніпулятора. Завдання руки повинно стати розміщення її в певному положенні в просторі. Рішення першої проблеми можна отримати, завдяки знаходженню кутів нахилу при з'єднаннях, які надаватимуть виконавчому елементу бажаної просторової орієнтації. Представлений в цій книзі зап'ясток використовує цю стратегію, щоб надати математичну процедуру, для вирішення проблеми протилежної руки, яка заснована на призначенні місцевих координат у кожному з'єднанні. Використовуючи обертання ця процедура може дати відношення до цих координат. Кінематичне обговорення синтезу потім представляється для структури руки. Воно буде показувати, що концепт перетину просторових кругів пропонується добре ґрунтувати для досягнення загальної кінематики об'єднаних завиток маніпуляторів. Окрім того, моделі представляються, щоб структура зап'ястка отримала повне зворотне кінематичне рішення автоматичного маніпулятора. Вирішений приклад продемонстрований, щоб довести обґрунтованість з представленого методу.

Роботи з кабельною основою будуються на механізмах, які не лише користуються жорсткими зв'язками їх структури, але і використовують односторонні силові елементи, наприклад кабелі. Кабелі також з'єднуються з активними лебідками, забезпечуючи змінну довжину, щоб приводити в дію механізм, щоб забезпечити кінематичне обмеження і не включати небажаний рух робочого органу. Маніпулятори, в яких кабелі мають змінну довжини, зазвичай називають кабельнокерованими або телекеруючими маніпуляторами. Роботи з кабельною основою мають переваги від кабельно вмонтованих маніпуляторів.

Автор відмічає, що користування кабельними структурами в автоматичних маніпуляторах супроводжується теоретичними і технічними труднощами. Одностороння сила кабелів ускладнює кінематику і аналіз динаміки. Обмеження для розтяжної сили в усіх кабелях повинно бути запроектовано і управлятися процедурою керування, інакше маніпулятор зруйнується. Також, низька жорсткість кабелів порівнянних з жорсткими зв'язками призведе до небажаних вібрацій, що прийдеться виправляти компенсаційною контрольною схемою.

Автор представляє загальну структуру цих маніпуляторів і критичні концепти жорсткості і можливі способи визначення напруги у в'язях. Виведена теорема, щоб спростити вивчення можливих напруг в цих маніпуляторах. Представлені два просторові маніпулятори на кабельній основі і доведена їх жорсткість.

Автор констатує, що планування руху - первинне завдання в автоматизації, де метою є визначення безпечного шляху для робота, який працює в довкіллі, що містить деякі перешкоди (Latombe, 1991; Fugimura, 1991; Tzafestas, 1999). Перешкода, що рухається це жорсткий об'єкт, або об'єкт

із з'єднаннями як наприклад промисловий маніпулятор. Змінюючись і частково непередбачуючи довкілля, автоматичне планування руху відбувається дуже повільно. Планувальник отримує безперервний потік інформації про появу подій і генерує нові команди, поки попередні сплановані рухи виконуються. – лінія автоматично планує рух, тобто заряджають обчисленнями раніше виконання будь-якого руху, і вимагається, щоб усі відповідні дані були доступні заздалегідь. Автоматичний планувальник руху і відповідні пристрої, що забезпечують чутливість робота можуть пристосуватись швидко до несподіваних змін в довкіллі і до моделювання помилки робочої області. Основна особливість розумних робототехнічних систем це здатність виконувати автономно безліч завдань без повного володіння інформацією, пристосовувавшись до безперервних змін в робочому довкіллі.

Автор пояснює, що робототехнічні маніпулятори, так як і мобільні роботи (також їх комбінація, тобто мобільним маніпулятори (Seraji, 1998; Tzafestas & Tzafestas, 2001)) залежні від алгоритму, який планує рух. Для робототехнічних маніпуляторів, планування руху це критичний аспект, через те, що шляхи крайнього виконавчого елемента мають завжди деякі обмеження. Наприклад, в дузі, що зварює можливо доведеться проходити за складним шляхом, що буде мати 3 виміри впродовж зварювального процесу. Визначення вручну таких шляхів може бути дуже нудним і на це потрібно витратити багато часу. Для руху мобільних роботів з плануючою і автономною навігацією критичним моментом можна вважати те, що під час їх роботи можуть працювати інші елементи які незалежні від системи, наприклад автоматичні прибиральники і т.п.

Автор ставить мету в цій главі показати рішення проектування планувальника руху без проблемного використання. По-перше, розглядається випадок в промислових робототехнічних маніпуляторах, а потім клас мобільних роботів. Прийнята методологія передусім ґрунтується на деяких недавніх результатах винаходів (Moustris & Tzafestas, 2005; Zavlangas & Tzafestas, 2000; 2001; 2002), щоб допомогти читачеві оцінити важливість техніки представленої в главі. Автор у короткому огляді уперше включає загальний плануючий автоматичний рух, разом з деякими недавніми результатами досліджень. Показаний хороший багатообіцяючий практичний підхід, який користується нечіткою логікою впродовж шляху системного проектування (Izumi & Watanabe, 2000; Topalov & Tzafestas, 2001; Watanabe і ін., 1996; Watanabe і ін., 2005).

Автор проводить короткий огляд плануючого автоматичного руху для промислового робота і мобільного робота. В подальшому розказується про техніку планування для індустріальних маніпуляторів і їх навігація. Продовження розширює цю техніку на мобільні роботи і обговорюється інтеграцію глобального і місцевого планування траєкторії і навігація робота. Встановлений глобальний шлях, при плануванні топологічної карти, що використовується для представлення довкілля роботів. Далі представлений набір експериментальних результатів для робототехнічного маніпулятора «SCARA Adept 1» і автофургона Robuter III (який може бути обладнаний робототехнічною рукою). Виникає як місцевий, так і глобальний шлях, для планування і навігації, включаючи робота Robuter III. Нарешті автор завершує загальним обговоренням результатів справжнього технічного забезпечення, для майбутніх напрямів дослідження.

#### **1.4 Адаптивне програмування робототехнічних комірок**

В даний час багато виробників інтелектуальних компонентів є приватними компаніями, які вільно не передають свої інтелектуальні компоненти та надбудови приватним особам за межами компанії. Враховуючи цей факт, вони також не публікують свої роботи в журналах для академічних цілей, що ускладнює пошук академічних досліджень розумних компонентів. Замість цього було взято тангенс, щоб побачити, що раніше було розроблено, приводячи технологію до поточного стану інтелектуальних компонентів та надбудов.

Один з найбільш подібних розкритих джерел літератури походить від Технологічного університету Чалмерса. Карл-Йохан Рутгерссон продовжив попередню роботу з розробки супервізора, який запобігав би зіткненню роботів всередині комірки [1]. На першому етапі проекту виконано віртуальне моделювання оригінальної програми. Це моделювання було досягнуто з певним успіхом, але його потрібно було протестувати в реальному середовищі. Рутгерссон зміг придбати тест-комірку і запрограмувати контролер робота IRC5 для маніпулювання роботами. Його тести були успішними; проте надійність програми викликала помітне занепокоєння.

Ще одним успіхом у галузі розвитку роботизованої конструкції комірок було оптимальне розміщення робота. Стаття, написана Сяолунгом Фенгом, наближається до оптимального розміщення робота із заданим завданням у робочій області [2]. Ця програма була розроблена як доповнення до RobotStudio. Проаналізувавши шляхи роботи робота і поставивши робота в оптимальне положення, вони змогли збільшити продуктивність на 37%.

Більше розробок у галузі програмування роботів також можна знайти у програмуванні рук. Навчання простим завданням роботів зайняло більшу частину часу роботизованих програмістів. На відміну від сучасних віртуальних моделей, роботи програмувались за допомогою навчальних підвісок із повністю побудованою роботою комірки. Щоб збільшити швидкість навчання робота новому завданню, було розроблено систему, здатну визначати рухи людської руки [3]. Цей проект був досягнутий з обмеженим успіхом, але показав необхідність віртуального представлення роботизованих комірок.

ABB просунувся на шляху роботизованого програмування, створивши програмні пакети, які розширюють можливості інженерів, спрощуючи завдання роботизованого програмування [4]. Створення RobotStudio надало людям, які практично не мають досвіду програмування, можливість створювати симуляції за лічені хвилини. Це програмне забезпечення різко скоротило час, необхідний для програмування робототехнічних станцій. Це досягло обриву завдяки доданню запрограмованих користувачем надбудов та розумних компонентів.

Хоча існує велика спільнота робототехнічних розробників, є чотири основні гравці: ABB, KUKA, Fanuc та Motoman. Ці троє розробників не тільки створюють роботів і контролери, але й створили засоби автономного програмування. Кожен з них створив свій власний інструмент програмування в режимі офлайн для обробки підтримуваних ними виробничих процесів.

ABB створив RobotStudio, який використовує сучасний графічний інтерфейс користувача та Комплект розробки програмного забезпечення для персональних комп'ютерів (PC SDK), який безпосередньо взаємодіє з RobotStudio [5]. Підвищена гнучкість RobotStudio PC SDK призвела до створення незалежних інтелектуальних компонентів та надбудов зовнішніми



роботизованої комірки та переконатися, що всередині комірки знаходиться відповідне приміщення для частинного завантаження, розвантаження та експлуатації. З цих причин розробники програмного забезпечення включили засоби вимірювання, які можуть проводити віртуальні вимірювання комірки; це може бути цінним як у віртуальному макеті, так і під час виготовлення цих комірок. Обов'язково, щоб віртуальна комірка відповідала фактичній комірці. Якщо комірка не є точною, віртуальні шляхи не можуть бути взяті безпосередньо з віртуальної комірки та скомпільовані у фактичний код робота для використання в реальній моделі.

В надії пришвидшити розміщення комірок, розробники програмного забезпечення створили стандартні бібліотеки для використання. Сюди входять людські машинні інтерфейси, позиціонери деталей, різні зварювальні матеріали та обладнання безпеки, таке як огороження та легкі завіси. Незважаючи на те, що ці бібліотеки є цінним інструментом економії часу, не завжди бажано використовувати надані компоненти. У цьому випадку можна інтегрувати власну геометрію з програмним забезпеченням для автоматизованого складання. Потім ці файли можуть бути змінені на станції, використовуючи обмежену кількість інструментів для модифікації геометрії, таких як забарвлення та видалення поверхні. Ця функція важлива не тільки при викладі початкової концепції комірки, але й при імпорті геометрії замовника, оскільки ця геометрія часто велика і надзвичайно складна.

Щоб розташувати ці бібліотеки всередині віртуальної комірки, користувач може або перетягнути бібліотеку у певне розташування, або встановити розташування за допомогою діалогового вікна. При встановленні координат бібліотеки вона може бути встановлена щодо світової системи координат, оголошеної станцією, створеного користувачем кадру або базового



кадру іншого компонента. Одиниці довжини можна змінювати між стандартними та англійськими, а одиниці кутів - між радіанами та градусами.

Через розмір та складність деталей замовника часто доводиться розміщувати деталі в різних орієнтаціях для різних виробничих процесів. Для створення бажаного ступеня свободи потрібні різні позиціонери. Ці виробники не тільки забезпечують роботу та контролери, але вони також пропонують стандартні позиціонери, які можуть взаємодіяти безпосередньо з контролером роботи, що постачається.

Це дозволяє координувати роботизовані рухи. Ці бібліотеки позиціонерів постачаються з програмним забезпеченням для автономного програмування. Якщо бажаний спеціальний позиціонер, є конструктори механізмів, які можна використовувати для створення власних позиціонерів. Завдяки цій функціональності користувачі можуть створювати власні механізми, побудовані з імпортованої геометрії. Вибравши положення суглоба і тіла, на яке він буде діяти, можна створити призматичний або обертальний суглоб. Потім ці позиціонери можна зберегти та використовувати на майбутніх станціях.

Після того, як деталь була розміщена у бажаній орієнтації, наступним кроком є проведення дослідження охоплення за допомогою робота. Хоча деталь може знаходитися всередині зони досяжності робота, важливо переконатися, що центральна точка робочого інструменту (ТСП) може бути розміщена в правильній орієнтації на деталі; наприклад, його можна орієнтувати перпендикулярно поверхні для фарбування. Програмне забезпечення кожної компанії запровадило функціональність досліджень охоплення.

Наступним кроком техніко-економічного обґрунтування є планування шляху. Цей процес повідомляє роботу, під яким кутом і зсувом має бути ТСР робота, щоб пройти бажаний шлях. Це може бути здійснено різними способами. Перший - це вибір початкових точок, кінцевих точок, кутів між ними, а також програмне забезпечення інтерполює лінійний шлях. Це також можна виконати на відрізках кругової дуги, вибравши початкову точку, кінцеву точку та ще одну точку вздовж кривої; це дозволяє програмі визначати орієнтацію, радіус кола та довжину дуги. Багато разів просте коло чи лінійний шлях не може виконати певний шлях. У цьому випадку виявлення краю можна використовувати для створення бажаного шляху.

Завершальним етапом техніко-економічного обґрунтування є моделювання. На цьому кроці робот запускає віртуальний контролер, щоб забезпечити належне виконання всіх команд. Щоб це вдалося, ТСР робота завжди повинен бути в доступному положенні. Робот також повинен мати змогу виконати інструкцію з переміщення, не змінюючи конфігурації важеля і не проходячи сингулярність. Під час цього процесу станція може також виконувати виявлення зіткнень та гарантувати, що робот не наїде на деталь або не зіткнеться з будь-яким із компонентів станції. Моделювання також може забезпечити прогнозовані терміни процесу, і ці часи процесу можна використовувати для оцінки кількості часу, необхідного для виготовлення всієї деталі.



## 2 ТЕХНОЛОГІЧНА ЧАСТИНА

### 2.1 Поточна реалізація інтелектуальних компонентів та надбудов

Завдяки додаванню надбудов та розумних компонентів до RobotStudio від АББ багато людей почали програмувати прості розумні компоненти. Ці розумні компоненти різняться між собою, але всі вони додають базу знань, яка швидко розвивається навколо RobotStudio. У розділі нижче висвітлено деякі програми, які є новаторськими надбудовами та розумними компонентами.

Розумний компонент Parametric Fence, створений Anders Spaak, будує коробчату огорожу [9]. Цей розумний компонент дозволяє користувачеві змінювати довжину та напрямок будь-якої частини огорожі. Використання цього розумного компонента також дозволяє користувачеві змінити колір огороження, яке вони завантажили. На рисунку 2.1 показано користувальницький інтерфейс зі смарт-компонентом.

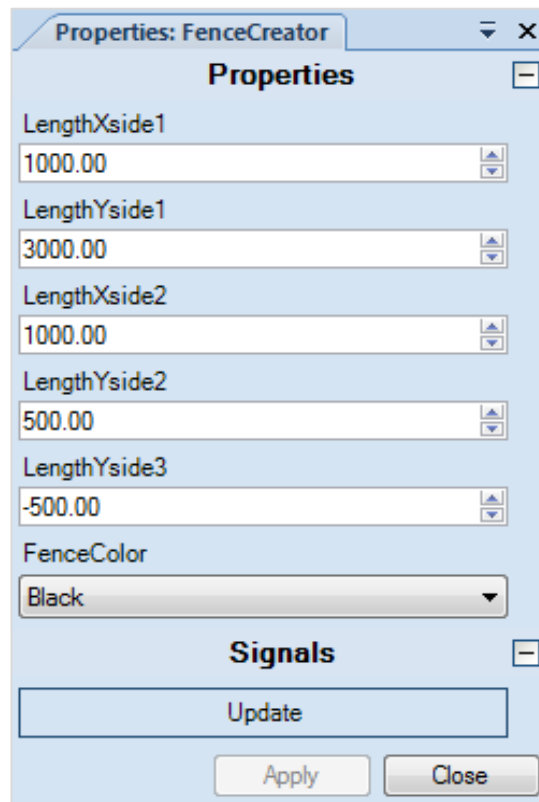


Рис. 2.1 Інтерфейс користувача з Parametric Fence Creator

На цьому малюнку показано, як користувач може змінити довжину будь-якої ділянки огорожі та встановити колір. На рисунку 2.2 показаний графічний вихід, створений на станції розумним компонентом.

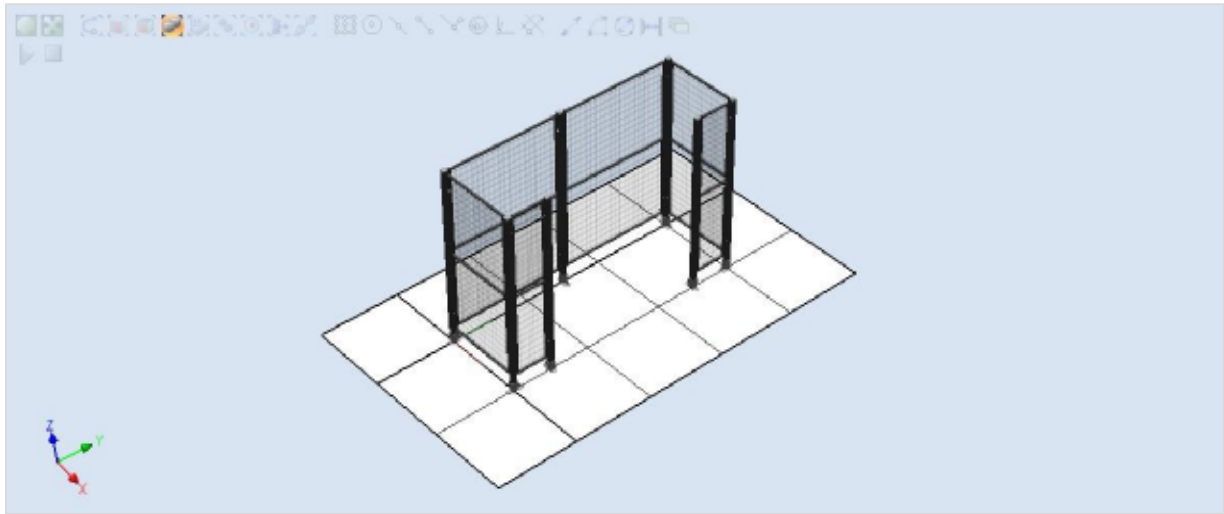


Рис. 2.2 Графічний вихід інтелектуального компонента параметричної огорожі

Parametric Robot Stand, створений Діном Томасом, є розумним компонентом, який використовує здатність RobotStudio створювати прості геометричні тіла та прикріплювати їх для створення деталі [10]. Цей розумний компонент дозволяє користувачеві вибрати верхній діаметр і висоту стенду. У міру зміни верхнього діаметра розмір основи та головного циліндра змінюється пропорційно вибраному верхньому діаметру. При зміні висоти змінюється розмір головного циліндра в напрямку  $z$ . Це також змінює положення верхньої пластини, тому вона все ще сидить на верхній частині головного циліндра. На рисунку 2.3 показано користувацький інтерфейс, сформований інтелектуальним компонентом. У цьому діалоговому вікні користувач може вибрати діаметр і висоту та оновити стенд робота. На рисунку 2.4 показаний графічний вивід, створений на станції розумним компонентом Parametric Robot Stand.

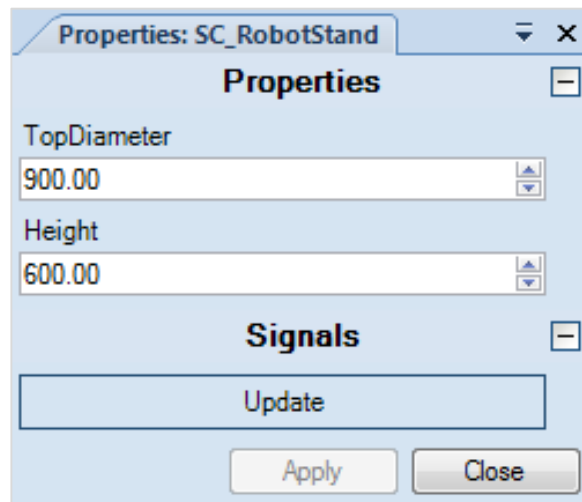


Рис. 2.3 Інтерфейс користувача з параметричною підставкою робота

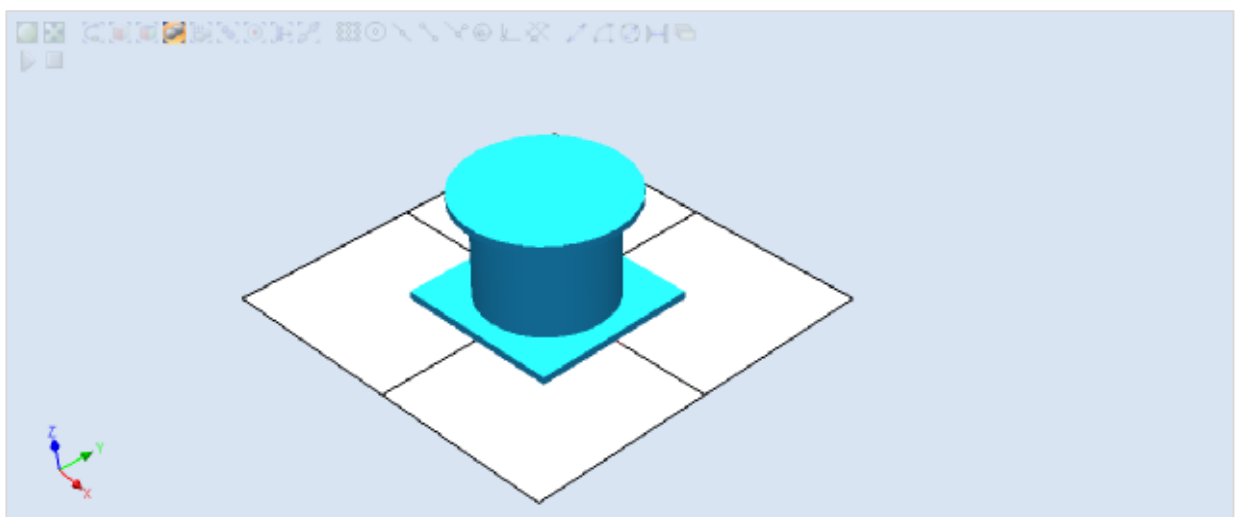


Рис. 2.4 Графічний вихід інтелектуального компонента параметричної підставки для роботів

Schunk Gripper, автор Річард Рамос, - це захоплювач, який автоматично кріпить і від'єднує деталь, яку він відчуває, у своєму захоплювачі [11]. Коли ви поміщаєте деталь у захоплювачі та вибираєте кнопку doGripper, губки захоплювача закриваються, а деталь рухатиметься за допомогою захоплювача. Щелепи відкриваються та відпустять об'єкт у поточному положенні, коли буде

вибрана кнопка diGripper, як показано на рисунку 2.5. Біля поточного стану захоплення з'являється жовтий глобус. Вихід на станції зображений на рисунку 2.6.

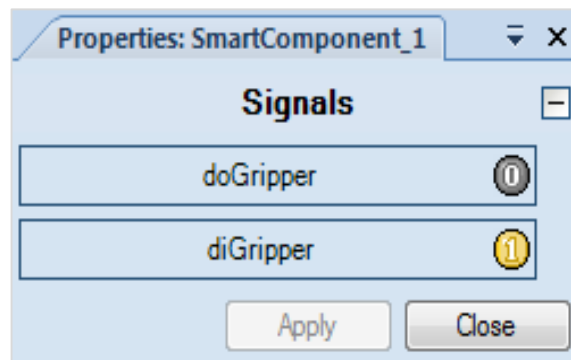


Рис. 2.5 Інтелектуальний компонентний інтерфейс Schunk Gripper

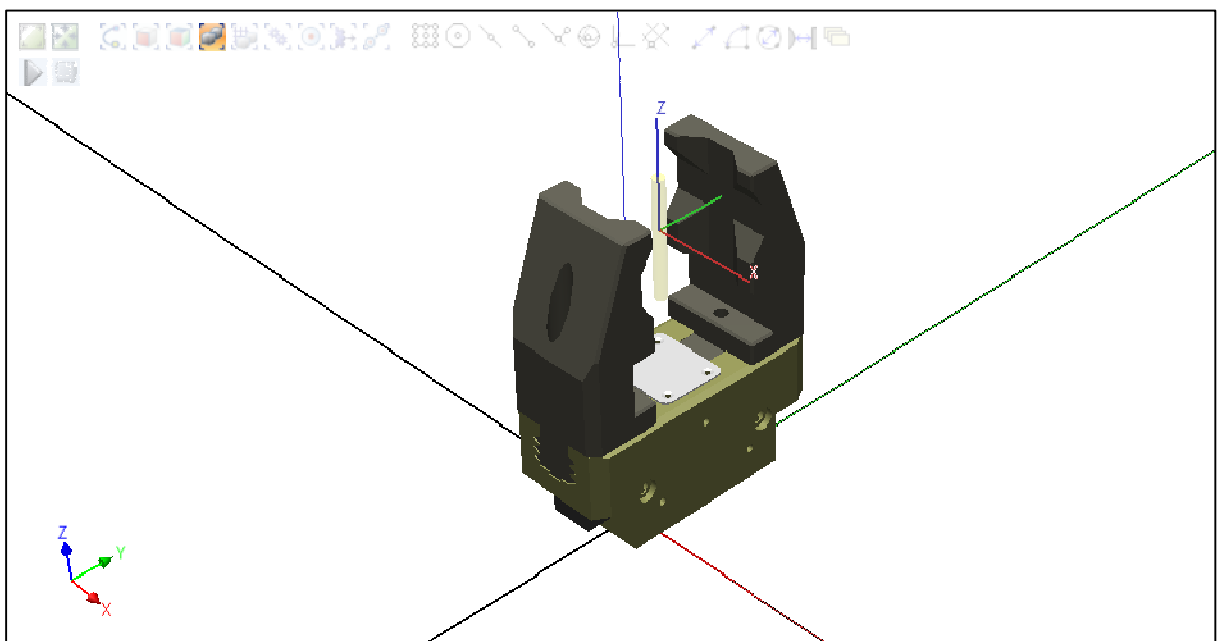


Рис. 2.6 Графічне представлення інтелектуального компонента Schunk Gripper



Надбудову для імпорту файлу координат створив Саймон Фогбрінг [12]. Цей надбудова містить деякі помилки в найновішій версії RobotStudio. Раніше ця надбудова дозволяла користувачеві додавати цілі до шляху із підтримуваного типу файлу. Це сприяло програмуванню роботизованих шляхів. Підтримувані формати файлів: nc (числовий контроль), csv (експорт Robcad), crd (файли координат ABB S3), cls, aptsource (експорт Catia) та aml (експорт Manz). Ця надбудова відхиляє координати, які ближче мінімально встановленої відстані. Він також має можливість виводити координати, коли кут відхиляється більше, ніж максимальний кут; це відносно попередньої успішно імпортованої координати. Він автоматично створював би кругові інструкції, якщо радіус був меншим за максимальний радіан.

Додаток до бібліотеки користувачів RS був створений ABB, щоб продемонструвати деякі можливості надбудов та створити зручну для доступу бібліотеку користувачів [13]. Цей надбудова створить кнопку бібліотеки користувачів на домашній вкладці. Ця вкладка автоматично заповнює бібліотеки, що містяться в певній структурі файлів. Файли повинні міститися в папці проектів користувача та вказувати на папку "Бібліотеки викликів". Окремі папки під Бібліотеки створювали б сегменти у спливаючому вікні. Кожен сегмент містив би зображення бібліотек, доступних за цією структурою файлів, як показано на рисунку 2.7. Це спочатку розглядалося для бібліотек Wolf Robotics; однак завдяки рекурсивній геометрії та обсягу бібліотек Вовка було обрано інший маршрут.

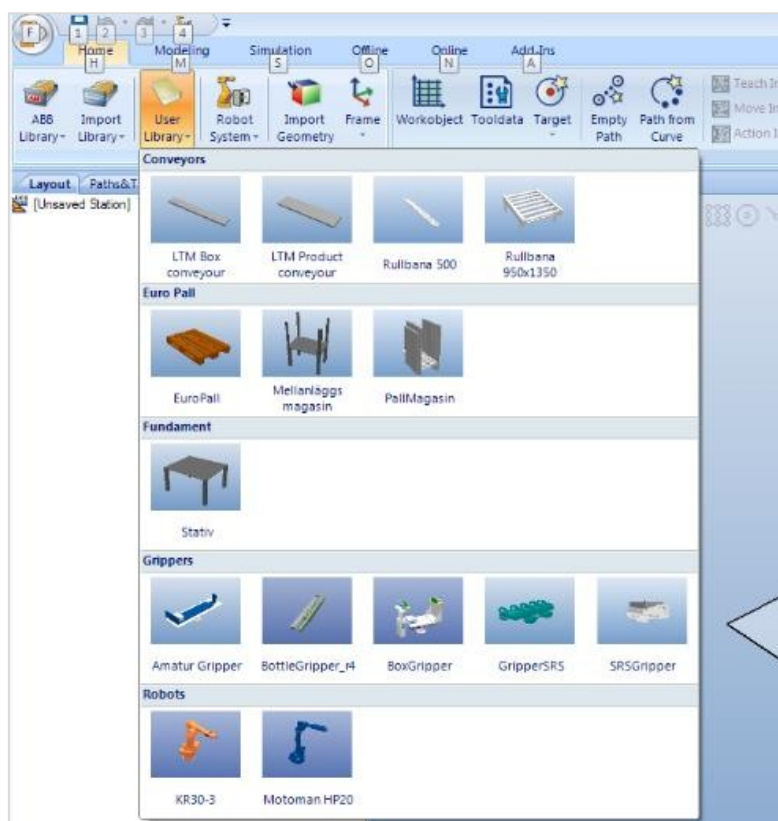


Рис. 2.7 Надбудова для бібліотеки користувачів RobotStudio

## 2.2 Технічні характеристики промислових роботів

### 2.2.1 Частини роботизованого маніпулятора

Робот - це все, що працює з певною мірою автономності, як правило, під контролем комп'ютера. Зазвичай термін робот (або робот-маніпулятор) називають лише його механічною частиною, тоді як він - просто компонент. Насправді робота-система складається з таких частин:

## 1. Механічна рука (робот-маніпулятор)

Це збирає всі механічні частини робота, ті, які без контролера - це просто прості матеріали. Робототехнічний кронштейн побудований таким чином, щоб нагадувати людський, і складається з комплексу жорстких ланок та гнучких з'єднань.

- Ланки: жорсткі ділянки, що складають механізм. (грудна клітка, надпліччя, передня рука)

- Суглоби: зв'язок між двома ланками. (плече, лікоть, зап'ястя)

## 2. Кінцевий інструмент (Кінцевий ефектор)

Кінцевий ефектор або інструмент кінця руки (ЕОАТ) - це частина, підключена до кінця роботизованої руки (до зап'ястя), що взаємодіє з навколишнім середовищем, і тому відома як найважливіша периферія робота. Залежно від завдання, яке повинен розробити робот, його структура та характер програмування та обладнання будуть різними. Для основних робіт зазвичай використовують нестандартні компоненти, оскільки вони дешевші та скорочують час підготовки. Натомість для спеціальних робіт кінцеві ефектори виготовляються на замовлення, тобто кожен із них розробляється та виготовляється з урахуванням конкретних характеристик та вимірів завдання, яке потрібно розробити.

Деякі з зазвичай використовуваних кінцевих ефекторів перелічені нижче.

- Захвати (найпоширеніші; для операцій вибору та розміщення)
- Щітки
- Магніти
- Шліфувальні машини
- Викрутки

- Інструменти для видалення матеріалу (наприклад, ріжучий інструмент або свердла)

- Пістолети-розпилювачі
- Вакуумні чашки
- Зварювальні пістолети
- Змінювачі інструментів

### 3. Зовнішні та внутрішні датчики

Датчики дозволяють обмінюватися інформацією між роботом та навколишнім середовищем. Вони передають роботів різні виміри від навколишнього середовища, що є важливим завданням для роботи належної роботи. Існує три типи інформації, яку можна надіслати:

- Різні типи вимірювань (наприклад, датчики сили і крутного моменту використовуються для вимірювання сили та крутного моменту при частинній вставці або складанні)
- Попередження про безпеку або помилки (наприклад, датчики проти зіткнення запобігають пошкодженню робочого інструменту або об'єкта, яким маніпулюють)
- Інформація в реальному часі про саме завдання (наприклад, встановлення камер поблизу робота дозволяє порівняти теоретичне з практичним рухом)

### 4. Зовнішнє джерело живлення

На даний момент цілком звично використовувати батареї як джерело енергії. Конструкція робота, що працює від акумулятора, повинна враховувати такі фактори, як безпека, тривалість циклу та вага. Іншим джерелом, яке використовується досить часто, є генератор, наприклад, двигун внутрішнього

згорання. Однак недоліки, такі як механічна складність, потреба у відведенні газу чи тепла та вага, призводять до зменшення його використання.

### 5. Інтерфейс комп'ютера (контролер робота)

Це може переходити від простих дискретних операцій вводу-виводу до протоколів промислового зв'язку.

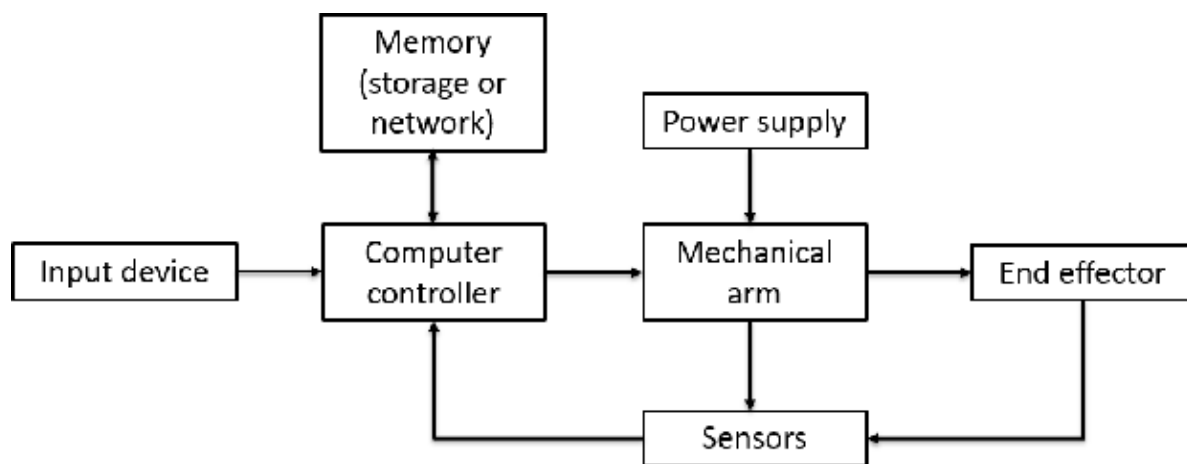


Рисунок 2.8 Компоненти роботизованої системи

### 2.2.2 Програми

З наукової точки зору роботизовані маніпулятори - це велика область досліджень, що дає змогу застосувати на практиці всю свою теорію. У минулому роботи сприймалися як цікавість, але незабаром вони почали отримувати широке визнання в промисловості, перебуваючи зараз у точці, де вони є незамінною частиною для автоматизації промислових процесів і, отже, стають звичайним інструментом у промисловому виробництві арсенал. Вони використовуються в найрізноманітніших галузях промисловості завдяки своїм потужним можливостям опору, точності функції або гнучкості для автоматизації та адаптації до робочого середовища. Окрім цього, їх

використання приносить такі переваги, як зниження витрат, підвищення продуктивності праці, поліпшення якості продукції або зменшення екологічних проблем.

В даний час роботи широко використовуються в промисловості, будучи необхідним елементом у більшій частині виробничих процесів. Промисловий робот є багатофункціональним, тобто теоретично його можна використовувати для необмеженої кількості додатків. Однак на практиці доведено, що його використання є оптимальним лише в певних процесах.

Найпоширеніші програми:

- Зварювання
- Живопис
- Збірка
- Виберіть і розмістіть
- Упаковка та маркування
- Палетизація
- Перевірка товару
- Тестування

Хоча звичайно пов'язують використання цього типу роботів із масовим виробництвом, у міру вдосконалення програмування та технологій стає все більш поширеним використання їх у робочих місцях / завданнях, які в минулому були надто небезпечними або неможливими для досягнення людиною, або коли якість результатів дуже висока. Прикладом цього є те, що роботів запускають у космос для завершення наступних етапів позаземних та позасонячних досліджень. На сьогоднішній день роботи займають місце в таких сферах, як іграшки та розваги, військова зброя, помічники в пошуку та резерві та багато інших.

Існує три різні методи програмування будь-якої роботизованої руки. Нижче буде зроблений опис кожного з них, а також переваги та недоліки кожного з методів.

### **Автономне програмування**

Цей метод полягає у використанні віртуального контролера (див. Розділ 3.4.2) для програмування робота, тобто моделювання та програмування всіх шляхів або процедур, які повинен повторити робот, і, закінчивши, відправити код справжньому роботу. Цей метод має широке застосування.

Перш за все, він використовується в робототехнічних дослідженнях, коли впроваджуються вдосконалені алгоритми управління. Таким чином, правильна робота коду може бути забезпечена перед тим, як спробувати його в реальному роботі.

Окрім цього, він також використовується у промисловості, оскільки це найкращий спосіб підвищити економічну ефективність роботизованих систем. Найважливішими перевагами цього методу є наступні:

- Скорочення простою (наприклад, завдяки можливості перевірити багато різних підходів до однієї і тієї ж проблеми)
- Підвищення ефективності
- Простий у використанні (Він дуже інтуїтивний, і, як правило, не потрібні глибокі навички програмування)
- Збільшення продуктивності (тобто це дозволяє формування, програмування та оптимізацію програм без зміни / зупинки виробництва)

- Зниження ризику (наприклад, за допомогою таких інструментів, як виявлення зіткнень, поломки можна виявити та виправити)

Але є й деякі недоліки, порівняно з іншими методами:

- Отримано не 100 % точності
- Помилки програмного забезпечення (тобто замість того, щоб просто зосередитись на коді або процесі, програмне забезпечення може також привести до деяких помилок, що призведе до втрати часу; виправлення самого програмного забезпечення, а також марно витрачання ресурсів; як людина мав би бути там, щоб це виправити).

### **Навчальний пульт**

На сьогодні це найпопулярніший метод, коли промисловим роботом керують безпосередньо за допомогою навчального пульта (блок керування для програмування опцій робота). За допомогою цього методу роботом керують поетапно (що також називають рухом точка-точка), тобто за раз надсилається одна команда. Позитивними моментами при використанні цього методу є:

- Знайомство серед техніків (Сьогодні високий відсоток промислових роботів має кулон для навчання)

- Точне позиціонування

І, навпаки, основними недоліками є:

- Потрібна підготовка (Навіть якщо це знайоме технічним працівникам, інші люди, які не звикли працювати з роботами, потребуватимуть навчання, щоб навчитися користуватися ним та програмувати з цим)



- Операції зупинені під час програмування (тобто робот повинен бути переведений у "режим навчання", поки він запрограмований, що змушує зупинити решту операцій)

### **Навчання шляхом демонстрації**

Застосовуючи цей метод, оператор зберігає кожну позицію в робочому комп'ютері. Сьогодні багато роботів застосували цей метод, оскільки операторам легко негайно розпочати роботу з використанням робота у своїх додатках. Переваги такі:

- Швидше
- Немає необхідності в знаннях щодо концепцій програмування

І навпаки, основними недоліками цього методу є те, що він не скорочує простої та позиціонування не так просто.

### **2.2.3 Кінематика робота**

Термін кінематика в робототехніці передбачає вивчення руху робота-маніпулятора, не дивлячись на сили та моменти, що впливають на рух. Зазвичай його поділяють на пряму та зворотну кінематику. Тут буде представлена пряма кінематика роботизованого маніпулятора.

Розрахунок прямої кінематики робота є важливим етапом при вивченні роботизованого маніпулятора, оскільки вони використовуються для отримання положення та орієнтації кінцевого ефектора для заданого набору значень спільних змінних за допомогою рівнянь кінематики. Не тема даної дипломної роботи - пояснювати теорію або детально пояснювати, як отримуються ці

рівняння кінематики, саме тому тут будуть висвітлені лише найбільш відповідні поняття; тих, хто вважається необхідним для застосування прямої кінематики до IRB140. Матеріал для цього розділу взято з [10]; де можна прочитати подальшу тему.

Кінематичний ланцюг - це група твердих тіл (їх також називають ланками), з'єднаних між собою стиками. Простим прикладом цього є робот-маніпулятор, який визначається як простий ланцюг, утворений жорсткими тілами, з'єднаними послідовно.

З'єднання можуть бути простими (революційні або призматичні) або більш складними (кульові або розеткові); але відтепер і для полегшення розрахунків будуть вивчатися лише кінематичні ланцюги з простими з'єднаннями, і, отже, завжди буде справедливим таке припущення:

"Кожен суглоб має один ступінь свободи, і, отже, дія кожного з них буде описуватися лише одним параметром. (Див. Рівняння 2.1) "

$$q_i = \begin{cases} \theta_i & \text{if joint } i \text{ is revolute} \\ d_i & \text{if joint } i \text{ is prismatic} \end{cases} \quad (2.1)$$

- Робот-маніпулятор, що складається з n суглобів, матиме n + 1 ланок.
- З'єднання пронумеровані від 1 до n, а ланки від 0 до n.
- З'єднання i з'єднує посилення i-1 з посиленням i, і воно фіксується щодо посилення i-1.
- Кожна ланка має жорстко прикріплений координатний каркас. (Координатна рамка o<sub>i</sub>x<sub>i</sub>y<sub>i</sub>z<sub>i</sub> додається до посилення i)
- Інерційний кадр (i = 0) - це той, який прикріплений до основи робота.
- Однорідна матриця перетворення A<sub>i</sub>:

- Дає інформацію про положення та орієнтацію  $o_i x_i y_i z_i$  щодо  $o_{i-1} x_{i-1} y_{i-1} z_{i-1}$ .

$$A_i = \begin{bmatrix} R_i^{i-1} & o_i^{i-1} \\ 0 & 1 \end{bmatrix} \quad (2.2)$$

- Відповідно до припущення, зробленого раніше, це лише функція змінної,  $q_i$ . (Рівняння 2.1)

• Розробляючи наступне множення, обчислюється положення та орієнтація кінцевого ефектора щодо інерційного кадру: (Він містить загальний обертоматриця  $R_0$  і загальна довжина  $o$ )

$$H = T_n^0 = A_1(q_1) \cdots A_n(q_n) = \begin{bmatrix} R_n^0 & o_n^0 \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (2.3)$$

Підсумовуючи, для визначення прямої кінематики робота-маніпулятора потрібно виконати два основні кроки:

1. Встановіть координатну рамку для кожного з'єднання
2. Заповніть таблицю параметрів ДН
3. Обчисліть кожен  $A_i(q_i)$
4. Помножте їх разом, щоб отримати  $H$

Але, навіть якщо це здається простим завданням, доведено, що розробка кінематичного аналізу маніпулятора з  $n$ -ланкою може бути досить складною. Для того, щоб максимально спростити аналіз та спробувати зробити його універсальним, протягом багатьох років були запроваджені такі конвенції, як Денавіт-Гартенберг (ДН).

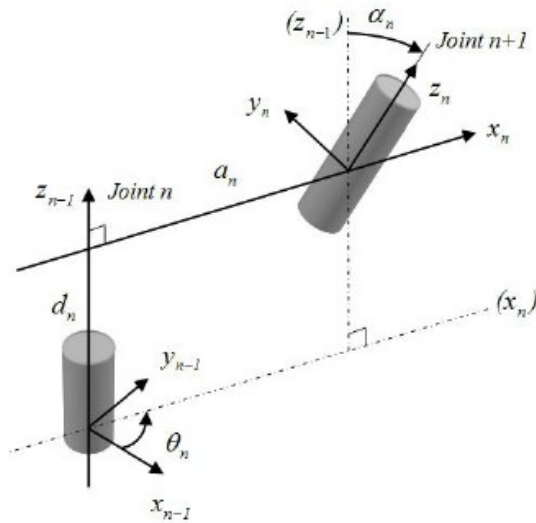


Рисунок 2.9 Призначення кадру ДН, взято з [3]

### Конвенція ДН

Це системний підхід до обчислення однорідної матриці перетворень (рівняння 2.3) кінематичного ланцюга. Загалом, будь-яка однорідна матриця перетворення має шість відповідних чисел (наприклад, три, що визначають положення, і три кути Ейлера). Натомість у конвенції ДН потрібні лише чотири параметри, але для цього потрібно зробити розумний вибір осі координат і координат.

Вся процедура, як визначати системи координат кожного з'єднання, описана на р72-74 [10]. Існує два важливих припущення, які необхідно виконати, роблячи вибір кожного координатного кадру, щоб можна було використовувати рівняння, викладені в конвенції ДН (показано нижче). Ці припущення забезпечують існування та унікальність однорідної матриці перетворень для обраних координатних кадрів і викладаються як:

**DH1) The axis  $x_1$  is perpendicular to the axis  $z_0$ .**

**(DH2) The axis  $x_1$  intersects the axis  $z_0$ .**

При використанні цього домовленості кадри призначаються, як показано на рисунку 2.9, і кожен  $A_i$  матриця обчислюється наступними основними перетвореннями:

$$A_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \quad (2.4)$$

Де чотири параметри широко відомі як:

- $a_i$ : довжина посилення
- $\alpha_i$ : поворот ланки
- $d_i$ : зміщення посилення
- $\theta_i$ : кут з'єднання

Запам'ятавши припущення з 2.1.5; можна легко зробити висновок, що три з чотирьох змінних будуть постійними, так що кожна матриця  $A_i$  є лише функцією однієї змінної.

Будучи наступними виразами для кожної з тих матриць, з яких складається  $A_i$

$$Rot_{z,\theta_i} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Trans_{x,a_i} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Trans_{z,d_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Rot_{x,\alpha_i} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Результат створення добутку вищевказаних рівнянь у порядку, описаному в рівнянні 2.4 - це наступне:

$$A_i = T_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_{i,i+1} & \sin\theta_i \sin\alpha_{i,i+1} & a_{i,i+1} \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_{i,i+1} & -\cos\theta_i \sin\alpha_{i,i+1} & a_{i,i+1} \sin\theta_i \\ 0 & \sin\alpha_{i,i+1} & \cos\alpha_{i,i+1} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

### 2.2.4 Ітеративний контроль навчання

Існує багато методів та методологій контролю, які можна застосувати до контрольних маніпуляторів. Вибраний конкретний метод управління, а також спосіб його реалізації можуть мати значний вплив на продуктивність маніпулятора і, отже, на діапазон його можливих застосувань.

У цьому розділі буде представлена теорія методу управління Iterative Learning Control (ILC). Роздумуючи над доповіддю про дисертацію, спочатку думали розробити огляд літератури з різних схем контролю, щоб усі вони могли бути вивчені. Але врешті-решт буде вивчено лише ILC, оскільки саме цей найкраще підходить для цього типу технологій.

Концепція ILC описує техніку, що використовується для поліпшення реакції (компенсації помилки) системи, яка працює повторно. Усі системи, що працюють у повторюваному режимі, зазнають однакового перевитрати, часу наростання, часу встановлення та помилок стаціонарного стану при кожному повторенні. ILC намагається покращити цей вихід, оновлюючи сигнал управління, поки не знайде вхід, який перетворює вихідну помилку на успішні результати [20]. Однією з найактуальніших мотивацій розвитку цієї концепції є промисловий робот, завдяки тому, що він, як правило, розробляється і

використовується для повторення одного і того ж завдання ще раз у промисловому процесі [21].

Давно це поняття не використовується, ще до того, як воно було офіційно відоме під назвою «Ітеративний контроль навчання». Перше використання цієї ідеї може бути датовано в 1967 р. (І запатентовано в 1971 р.), Коли було опубліковано «Навчання керуванню виконавчим механізмом в системах управління». У цій роботі вивчалася ідея зберігання "командного сигналу" в пам'яті, а потім оновлення його ітеративним способом з використанням помилки між фактичною реакцією та бажаною реакцією [22]. Шукаючи точну концепцію ІЛС, вона була вперше представлена в 1978 році Учіямою, але він опублікував її японською мовою, тож лише в 1984 році вона стала активною сферою досліджень. Це сталося з публікацією "Поліпшення роботи роботів шляхом навчання" Арімото, Кавамури та Міядзакі [23]. Він був опублікований англійською мовою та вказував метод, що використовував просте правило ітерації, яке автономно генерує поточний ввід приводу краще, ніж попередній [23].

У ІЛС враховано деякі припущення, які чітко розмежують цей метод із стандартною проблемою відстеження [24]. При  $t = T$  (один період / повторення):

- Стан  $x$  системи скидається до того самого початкового стану  $x_0$ , тобто  $x(T) = x_0$ .
- Після перезавантаження система знову вимагає відстеження того самого опорного сигналу  $r(t)$ .

Стандартна схема ІЛС така:

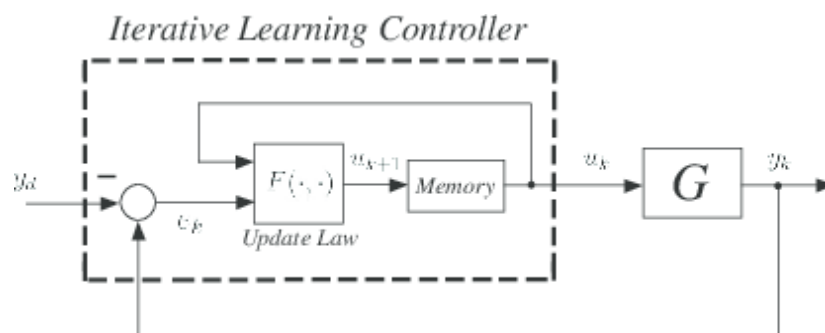


Рисунок 2.10 Стандартна схема ІЛС, взята з [10]



## **3 КОНСТРУКТОРСЬКА ЧАСТИНА**

Хоча бібліотеки, надані розробниками, корисні, користувацькі бібліотеки кожного розробника є загальними. Цей процес включає введення бажаних геометрій, забарвлення поверхонь геометрії та побудову декількох геометрій у механізм, якщо такий потрібен. Компанія ABB RobotStudio працювала над тим, щоб подолати ці короткі надходження, створивши набір розробників програмного забезпечення. Сюди входять різні команди, які можуть маніпулювати віртуальною станцією з урахуванням набору обмежень. SDK спочатку підтримує Visual Studio C # від Microsoft; використовуючи надані шаблони програм, налагоджувач може контролювати надбудову або інтелектуальний компонент, коли він працює всередині RobotStudio. Це дозволило полегшити пошук помилок під час програмування.

Перш ніж можна було створити розумні компоненти та надбудову, позиціонери та деталі Wolf Robotics потрібно було перетворити на бібліотеки RobotStudio. Стандарти Wolf Robotics були спочатку розібрані, щоб визначити, з яких основних деталей складається кожен компонент; відзначаючи повторювані геометрії між компонентами. Потім бібліотеки графічних компонентів були створені з понад 500 окремих файлів CAD. Кожна частина була ретельно розміщена в окремих бібліотеках і пофарбована у відповідний колір. Потім ці бібліотеки були розділені на ієрархічну структуру папок для завантаження інтелектуальних компонентів та надбудови.

### **3.1 Розробка автоматичної огорожі**

Компонент для введення вибраної довжини огорожі був бажаним, щоб одиничні панелі огорожі не повинні були розташовуватися та орієнтуватися

окремо. Також було бажано, щоб довжина або висота огорожі могла бути легко оновлена, якщо станція зміниться в майбутньому. Цей компонент був побудований для використання матричного ретранслятора компонентів RobotStudio. Це дозволило б повторити одиничний об'єкт лінійно вздовж обраного напрямку, обмежуючи тим самим кількість необхідних приєднаних геометрій. Приклад цього розумного компонента можна знайти на рисунку 3.1, наведеному нижче.

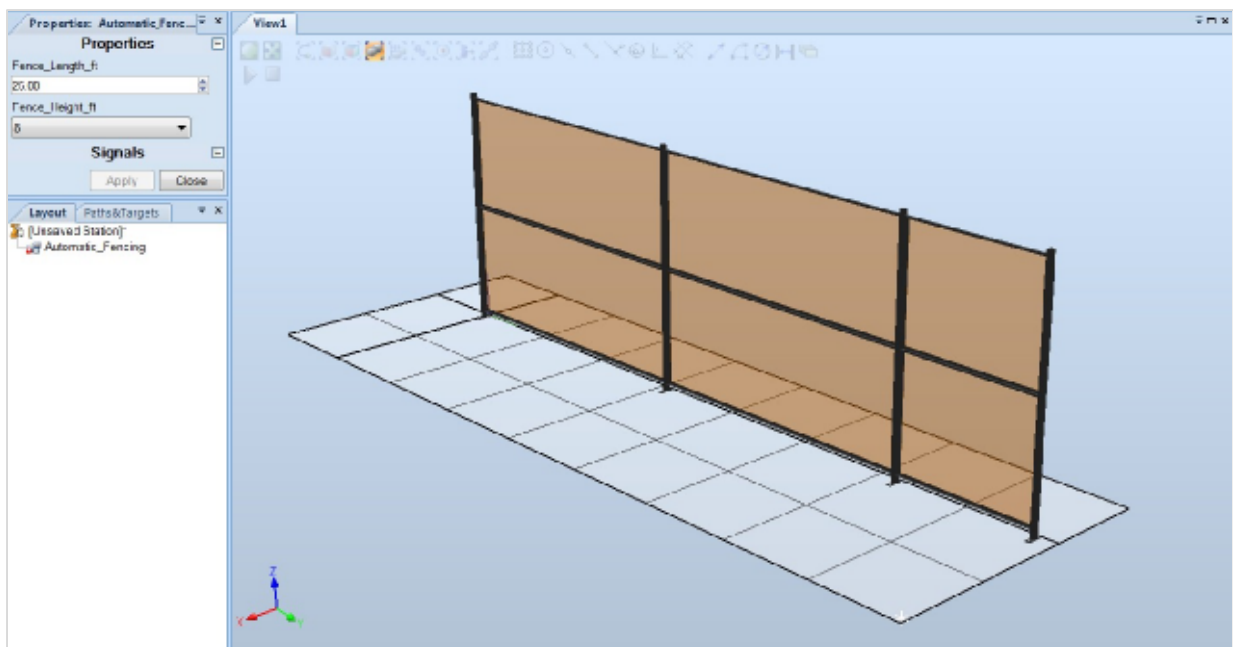


Рисунок 3.1 Розумний компонент автоматичного забору

Першим кроком було оголошення змінних таким чином, щоб користувач RobotStudio міг встановити змінні, а станція відображатиме зміни. Це було здійснено за допомогою XML-файлу, який створюється при створенні надбудови RobotStudio; цей XML-файл компілюється в бібліотеці RobotStudio (.rslib). Сегмент коду Fence\_Calculator можна побачити на рисунку 3.2; цей сегмент коду оголошує одну змінну, яку можна прочитати або записати в

RobotStudio та три змінні, які можна прочитати з RobotStudio. Три значення лише для читання використовуються в матричному ретрансляторі RobotStudio: count, distance та offset. Граф визначає кількість необхідних панелей огорожі 4 метри на 10 метрів, відстань визначає інтервал, на якому будуть розміщені ці сегменти огорожі, а зміщення визначає, де слід розміщувати останню панель огорожі.

```
<DynamicProperty name="Fence_Height_ft" valueType="System.Int32" value="0">
  <Attribute key="Allowed" value="4;8;10;12"/>
</DynamicProperty>

<DynamicProperty name="count_4ft_10ft" valueType="System.Double" value="0" readOnly="true">
  <Attribute key="MinValue" value="0"/>
  <Attribute key="Quantity" value="None"/>
</DynamicProperty>
<DynamicProperty name="dist_4ft_10ft" valueType="System.Double" value="0" readOnly="true">
  <Attribute key="MinValue" value="0"/>
  <Attribute key="Quantity" value="Length"/>
</DynamicProperty>
<DynamicProperty name="offset_4ft" valueType="System.Double" value="0" readOnly="true">
  <Attribute key="MinValue" value="0"/>
  <Attribute key="Quantity" value="Length"/>
</DynamicProperty>
```

Рисунок 3.2 Smart-компонентний XML-калькулятор паркану міститься в Додатку А.

Наступним кроком було форматування змінних у форму, яку можна використовувати у файлі CodeBehind для розрахунків. Файл CodeBehind включається до бібліотеки RobotStudio під час компіляції. Рисунок 3.3 - це сегмент коду з Fence\_Calculator, який зчитує значення зі станції RobotStudio і встановлює його цілим числом, яке може використовуватися SmartComponent.

```
int Orig_Fence_Height = (int)component.Properties["Fence_Height_ft"].Value;
```

Рисунок 3.3 Код калькулятора паркану знайдений у Додатку А.

Нарешті, кількість сегментів 10 метрів обчислюється за допомогою простого алгоритму. Довжина огорожі ділиться на кожне ціле число від десяти до одиниці, і ціле число зберігається в масиві, усіваючи залишок. Вищеописаний код можна знайти на рисунку 3.4.

```
fence[9] = Orig_fencelength / 10;
fence[9] = (int)Math.Floor(fence[9]);
temp = Orig_fencelength % 10;
x--;

while (x >= 0)
{
    fence[x] = temp / (x + 1);
    fence[x] = (int)Math.Floor(fence[x]);
    temp = temp % (x + 1);
    x--;
}
```

Рисунок 3.4 Код калькулятора паркану знайдений у Додатку А.

Потім масив, створений алгоритмом, використовується для визначення необхідного зміщення та кількості необхідних сегментів. Відстань між панелями огорожі (зсув) відома з довжини огорожі плюс довжина опорного стовпа. Потім старі значення очищаються, так що якщо оновити панель, неправильні сегменти не будуть створені. Це було реалізовано в C #, сегмент коду, що це показує, можна знайти нижче на рисунку 3.5.

```
component.Properties["count_4ft_10ft"].Value = (fence[9]);
component.Properties["dist_4ft_10ft"].Value = (3.0988);
component.Properties["offset_4ft"].Value = (fence[9] * 3.0988);

component.Properties["count_8ft_10ft"].Value = (0);
component.Properties["dist_8ft_10ft"].Value = (0);
component.Properties["offset_8ft"].Value = (0);
component.Properties["count_10ft_10ft"].Value = (0);
component.Properties["dist_10ft_10ft"].Value = (0);
component.Properties["offset_10ft"].Value = (0);
component.Properties["count_12ft_10ft"].Value = (0);
component.Properties["dist_12ft_10ft"].Value = (0);
component.Properties["offset_12ft"].Value = (0);
```

Рисунок 3.5 Код калькулятора паркану знайдений у Додатку А.

Після того, як CodeBehind буде скомпільований, його можна буде завантажити в RobotStudio і використовувати за допомогою візуального програмування RobotStudio. На рисунку 3.6 показано, як код Fence\_Calculator взаємодіє з візуально запрограмованим інтелектуальним компонентом. Візуальне програмування використовує дроти для встановлення змінних, рівних між собою. Тут видно, що змінні Fence\_Length\_ft і Fence\_Height\_ft, визначені користувачем на станції, вливаються в інтелектуальний компонент Fence\_Calculator; це те, що було створено вище в С #.

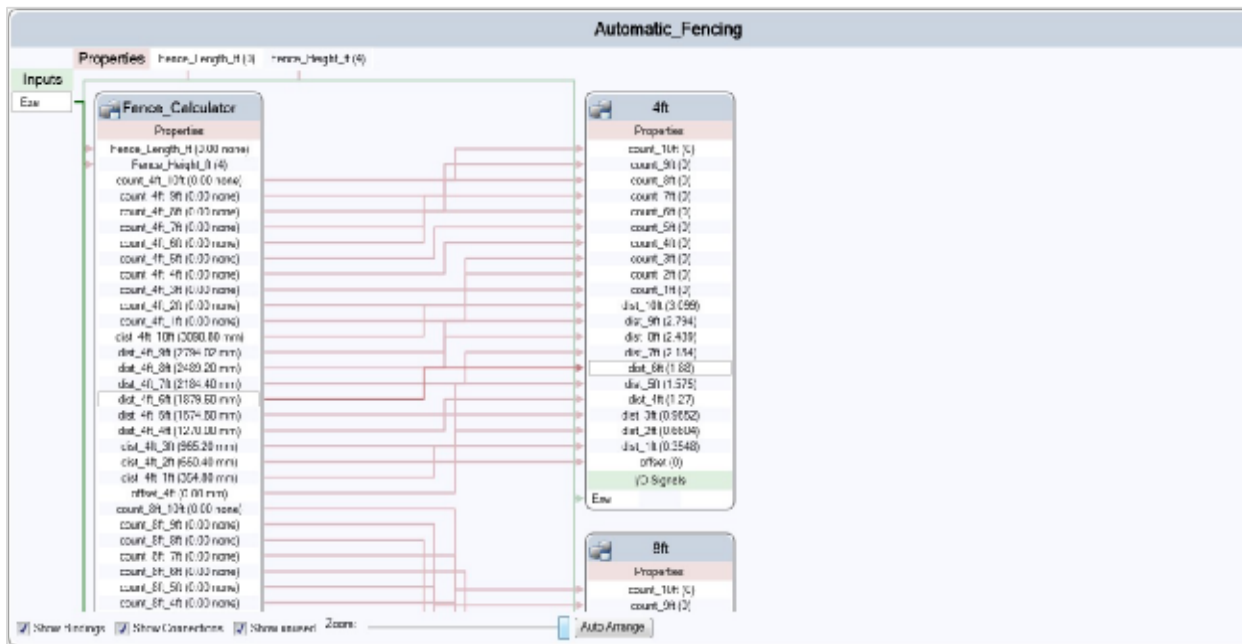


Рисунок 3.6 Калькулятор парканів RobotStudio в Додатку А.

Вихід від цього розумного компонента потім надходить до додаткового розумного компонента, який називається 4ft, 8ft, 10ft і 12ft.

Інтелектуальний компонент для висот огорожі чотирьох футів можна побачити нижче на рисунку 3.7. Цей інтелектуальний компонент містить компонент приховування, матричний повторювач та вираз перетворення. Компонент `hide` відповідає за приховування кореневого екземпляра графічного компонента - в даному випадку єдиної довжини огорожі в чотири фути. Для кожного із 10 стандартних довжин, що використовуються, існує один компонент приховування. Це оновлюється щоразу, коли змінюється основний компонент; це гарантує, що небажані сегменти огорожі приховані від очей користувачів. Матричний ретранслятор відповідає за створення відповідної кількості сегментів огорожі та зміщення їх на відповідну відстань. Цей компонент також пов'язаний з кореневим графічним компонентом, тому правильна геометрія продубльована. Кінцевий компонент був створений як вираз, цей вираз перекладає матричний ретранслятор; це дозволяє використовувати кілька довжин огорожі одночасно. Також було створено три додаткові компоненти для управління восьми футовими сегментами огорожі, десяти футовими сегментами огорожі та дванадцятьма сегментами паркану; їх можна знайти в Додатку A Fence\_Calculator RobotStudio.

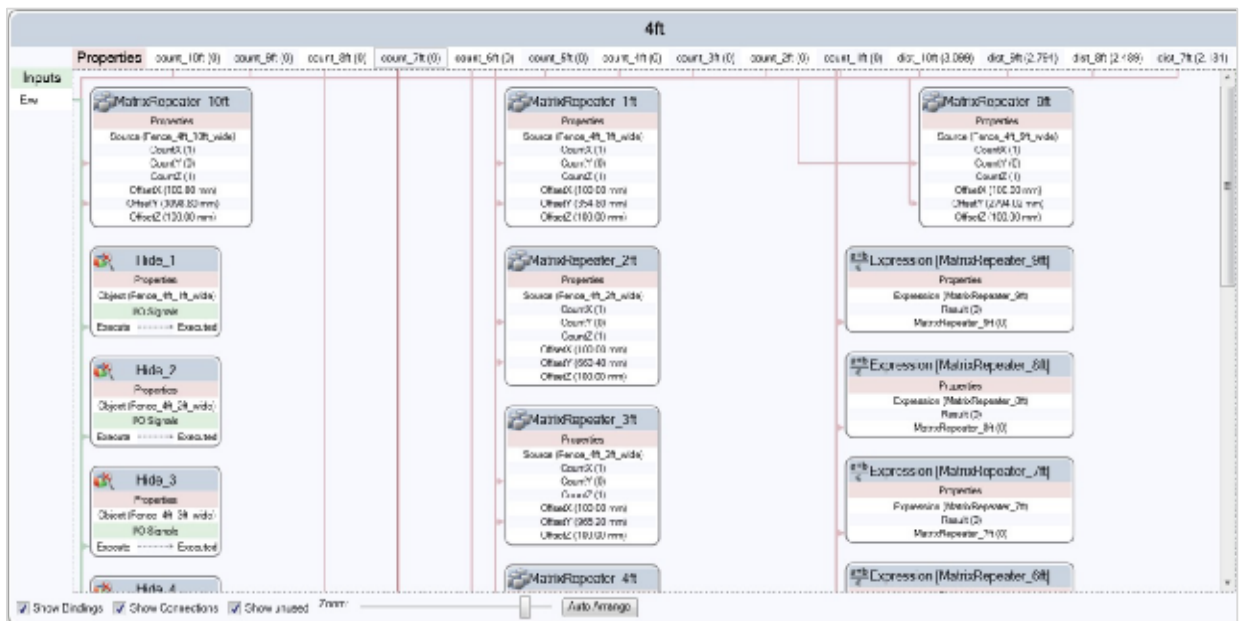


Рисунок 3.7 Калькулятор парканів RobotStudio в Додатку А.

### 3.2 Автоматичний конструктор доріжок

Для спрощення процесу введення робочої колії, стріли та необхідного зварювального обладнання було створено компонент для завантаження необхідної геометрії та оновлення існуючого механізму. Обмеження цього компонента повинні відповідати довжині колії та висоті вежі. Це можна знайти на малюнку нижче, рисунку 3.8. Першим кроком у цьому процесі був розгляд стандартних треків подорожей Wolf. Є 20 стандартних розмірів, які можна використовувати; кожна містить свою комбінацію частин. Додаток А. Порядок сегментів Track\_Builder показує порядок деталей, який потрібно використовувати для створення композицій різного розміру.

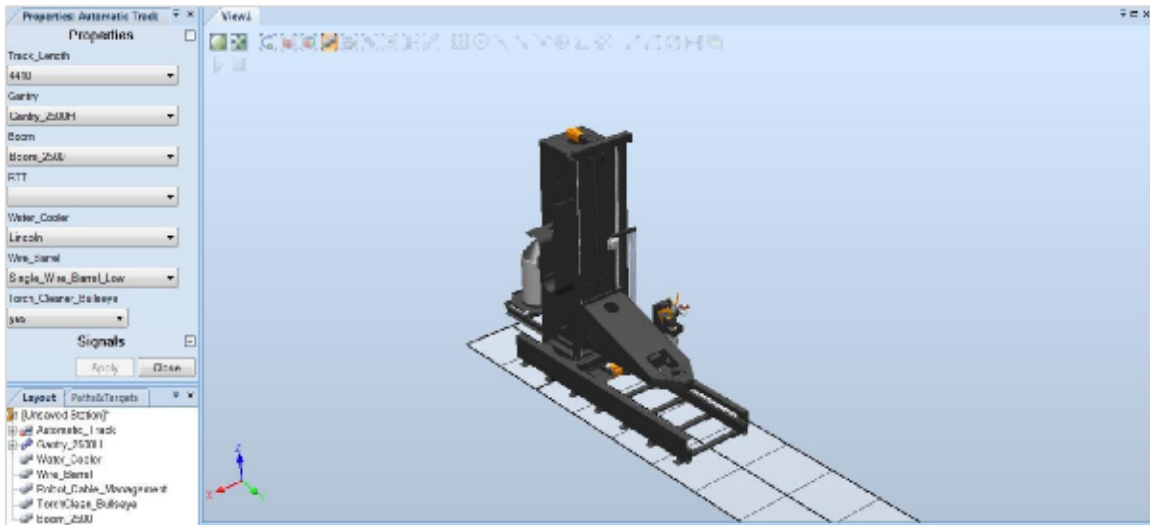


Рисунок 3.8 Розумний компонент Автоматичний конструктор доріжок

Спочатку змінні створюються та завантажуються аналогічно методу, що використовується в автоматичному паркані.

Був бажаний механізм, що представляє стандартні довжини доріжок, що використовуються у Wolf Robotics, тому доріжка на віртуальній станції побудована відповідно до специфікацій, викладених у таблицях, наведених у Додатку А. Порядок сегментів калькулятора паркану. Були побудовані великі масиви, щоб зберігати кількість кожного сегмента, що використовується на певній довжині колії. Далі компонент Track\_Builder шукає попередню геометрію станції та видаляє її, тому нові компоненти не створюються поверх старих. Цей спосіб завантаження та видалення поточних компонентів станції використовується у багатьох інтелектуальних компонентах і видно на рисунку 3.9.



```

GraphicComponent temp;
Station station = Project.ActiveProject as Station;
station.GraphicComponents.TryGetGraphicComponent("Carriage", out temp);
if (temp != null)
{
    station.GraphicComponents.Remove(temp);
}

```

Рисунок 3.9 Код побудови автоматичної доріжки з Додатку А.

Після видалення старої геометрії зі станції новий механізм завантажується в пам'ять на основі бажаної висоти башти. Потім обмеження швів встановлюються на довжину побудованої колії. Потім новий механізм додається до станції та відключається від бібліотеки, щоб майбутні зміни в механізмі не зберігалися в бібліотеці механізмів. Цей процес видно на рисунку 3.10.

```

//Load mechanism tower
Station station = Project.ActiveProject as Station;
GraphicComponentLibrary mylib;
mylib =
GraphicComponentLibrary.Load("S:\\LibraryBuild\\Custom_Lib\\Gantry\\Gantry_2500H.rslib", true);
Mechanism Gantry = (Mechanism)mylib.RootComponent.CopyInstance();
Gantry.Name = "Gantry_2500H";
Gantry.SetJointLimits(0, 0, ((Orig_Track_Length - 1667.05) / 1000));
station.GraphicComponents.Add(Gantry);
Gantry.DisconnectFromLibrary();

```

Рисунок 3.10 Код побудови автоматичної доріжки з Додатку А.

Потім обрана довжина стріли завантажується на станцію і розміщується відносно вежі, яка була додана до станції. Потім козловий механізм розсікають, щоб отримати належний шарнір механізму. Стріла пов'язана з цим зв'язком матрицею, що описує її орієнтацію щодо світової системи координат - це видно на рисунку 3.11.

```

mylib = GraphicComponentLibrary.Load("S:\\Library
Build\\Custom_Lib\\Gantry\\Boom\\Boom_1500.rslib", true);
Part Boom = (Part)mylib.RootComponent.CopyInstance();
Boom.Name = "Boom_1500";
station.GraphicComponents.Add(Boom);

//Place boom in correct position
Boom.Transform.X = .708271;
Boom.Transform.Y = .666345;
Boom.Transform.Z = .946033;

//Attach boom to mechanism
GraphicComponent L4;
Gantry.GetParentLink(3, out L4);
Part temp = (Part)L4;
temp.Attach(Boom, false, new Matrix4(new Vector3(0, 0, 0)));

```

Рисунок 3.11 Код побудови автоматичної доріжки з Додатку А.

Знову CodeBehind компілюється та завантажується в RobotStudio, щоб візуальне програмування RobotStudio могло використовувати новостворений інтелектуальний компонент. У цьому інтелектуальному компоненті користувацькі змінні від станції подаються в інтелектуальний компонент Track\_Builder за допомогою проводів. Потім цей компонент формує описані вище розрахунки та подає їх на чотири частини, що складають колію: Rail\_Weldment, Rail\_Cover, Rack та Star\_Rail - як показано на рисунку 3.12.

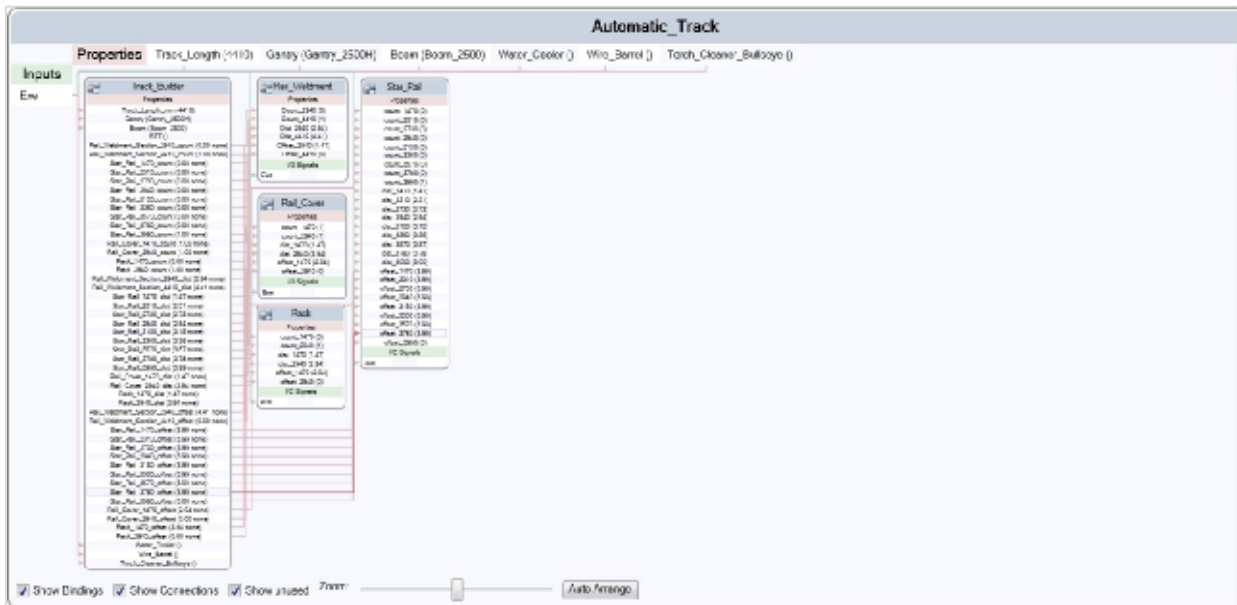


Рисунок 3.12 Автоматичний трек-конструктор RobotStudio в Додатку А.

Щоб дати загальне уявлення про те, як функціонує кожен із цих розумних компонентів, буде детально обговорено компонент Rail\_Weldment. Візуальне програмування цього розумного компонента видно на рисунку 3.13.

Це показує три основні компоненти цього розумного компонента: компонент приховування, ретранслятор матриці та вираз перекладу. Як і раніше, прихований компонент видаляє відображення кореневого графічного компонента, який видно на станції. Матричний ретранслятор відповідає за створення видимих екземплярів прихованого компонента, коли вони доречні. Матричний ретранслятор також відповідає за зміщення кожної рейки. Переклад матричного ретранслятора контролюється виразом. Цей вираз бере зсув, заданий компонентом Track\_Builder, і переводить повторювач матриці.

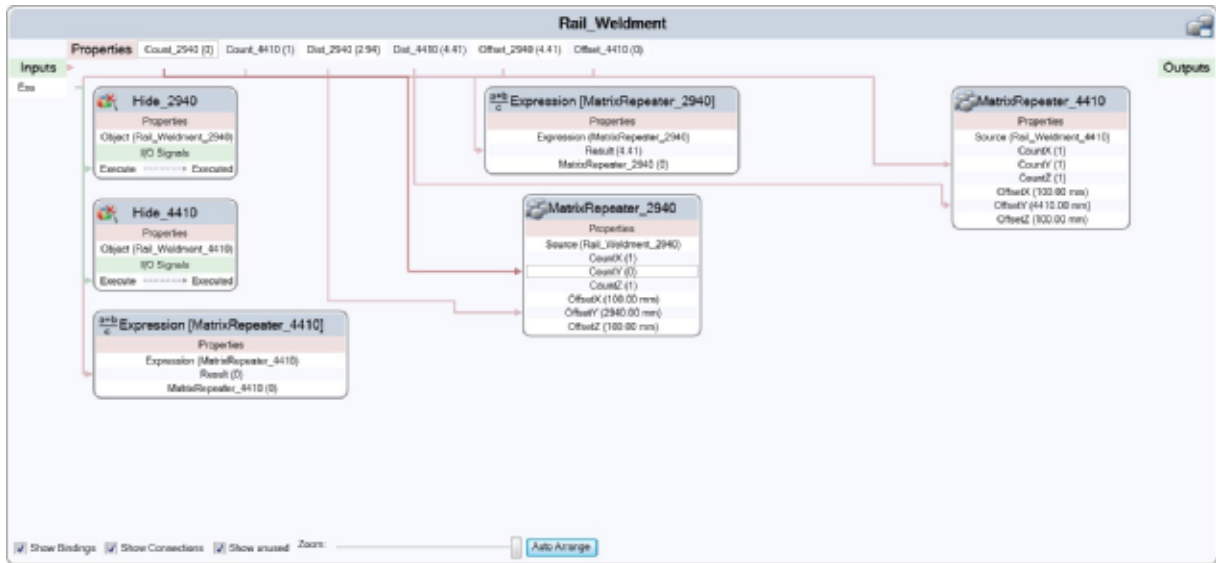


Рисунок 3.13 Автоматичний трек-конструктор RobotStudio в Додатку А.

### 3.3 Конструювання автоматичних легких завісів робототехнічної комірки

Розумний компонент автоматичної світлової завіси був запрограмований динамічно, щоб на станцію можна було одночасно імпортувати кілька світлових завіс. Цей компонент створює напівпрозоре зображення лазерів, що поширюються по отвору клітини. Це видно на малюнку, включеному нижче, рисунок 3.14.

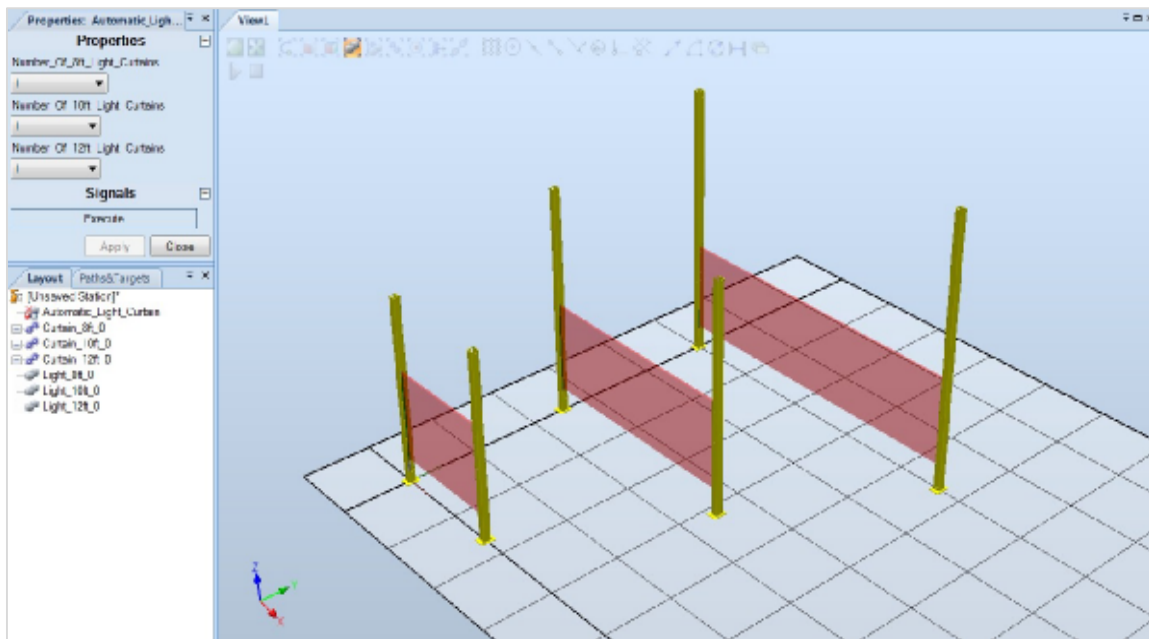


Рисунок 3.14 Автоматичні світлові завіси Smart Component

Три петлі керують програмою і базуються на кількості штор кожної висоти, як показано на рисунку 3.15. Це дозволило змінити кожну окрему світлову завісу. Спочатку рядок оголошується, а потім редагується на кожній ітерації циклу, щоб компонент можна було викликати правильно. Наведений нижче код очищає станцію старої геометрії, тому можна створити нову оновлену геометрію.

```
System.Text.StringBuilder sb = new System.Text.StringBuilder("Curtain_8ft_0");
System.Text.StringBuilder sb_light = new System.Text.StringBuilder("Light_8ft_0");
```

**LOOP:**

```
sb.Remove(12, 1);
sb_light.Remove(10,1);
sb.Append(count);
sb_light.Append(count);
```

Рисунок 3.15 Автоматичний код світлової завіси, наведений у Додатку А.

Шукається вже існуюча світлова завіса, і якщо жодної не буде знайдено, буде створена. Спільні значення, розташування та орієнтація витягуються зі станції та використовуються для побудови відповідної лазерної завіси. У наведеному нижче коді показано спільні значення, що використовуються для створення вікна належної довжини, а також розташування та орієнтація світлової завіси, що використовується для встановлення положення коробки. Після встановлення всіх цих змінних і додавання вікна до станції встановлюється колір і прозорість новоствореного вікна. Це було реалізовано в коді, зображеному на рисунку 3.16.

```
//Get Current Joint Values
double[] CurtainLength;
CurtainLength = Curtain.GetJointValues();
Curtain.Frames.ToArray();

//Create Light Curtain
Part Light = new Part();
Light.Name = sb_light.ToString();

//Move Light Curtain to Correct Location
Matrix4 rot = Curtain.Transform.GlobalMatrix;
Vector3 size = new Vector3(CurtainLength[0] + .13096, .03048, 1.20371);

//Create Body and set Color and Transparency
Body square = Body.CreateSolidBox(rot, size);
square.Transform.Z = .33439;
square.Name = "box";
Light.Bodies.Add(square);
VSTABridge.SetColor(Light, 255, 0, 0, 126);
```

Рисунок 3.16 Автоматичний код світлової завіси, наведений у Додатку А.

Потім лазерна завіса прикріплюється до основи механізму світлової завіси так, що вона змінює положення зі світловою завісою, якщо користувач вирішує перемістити або переорієнтувати механізм світлової завіси. Якщо при початковому пошуку виявлена світлова завіса, пов'язана лазерна завіса видаляється і перебудовується на основі нових спільних значень,

розташування та орієнтації світлової завіси. Це було реалізовано в коді, зображеному на рисунку 3.17.

```
//Attach box to base of mechanism
GraphicComponent home;
Curtain.GetParentLink(0, out home);
Part home1 = (Part)home;
home1.Attach(Light, false, new Matrix4(new Vector3(0, 0, 0)));
```

Рисунок 3.17 Автоматичний код світлової завіси, наведений у Додатку А.

Потім CodeBehind для автоматичної світлової завіси компілюється і завантажується в RobotStudio. Візуальне програмування, відповідальне за цей компонент, набагато простіше, ніж компоненти раніше. Цей компонент завантажує всю необхідну геометрію і, отже, вимагає лише введення від користувача. Цей вхід був створений на станції та підключений до інтелектуального компонента. Це видно на рисунку 3.18.

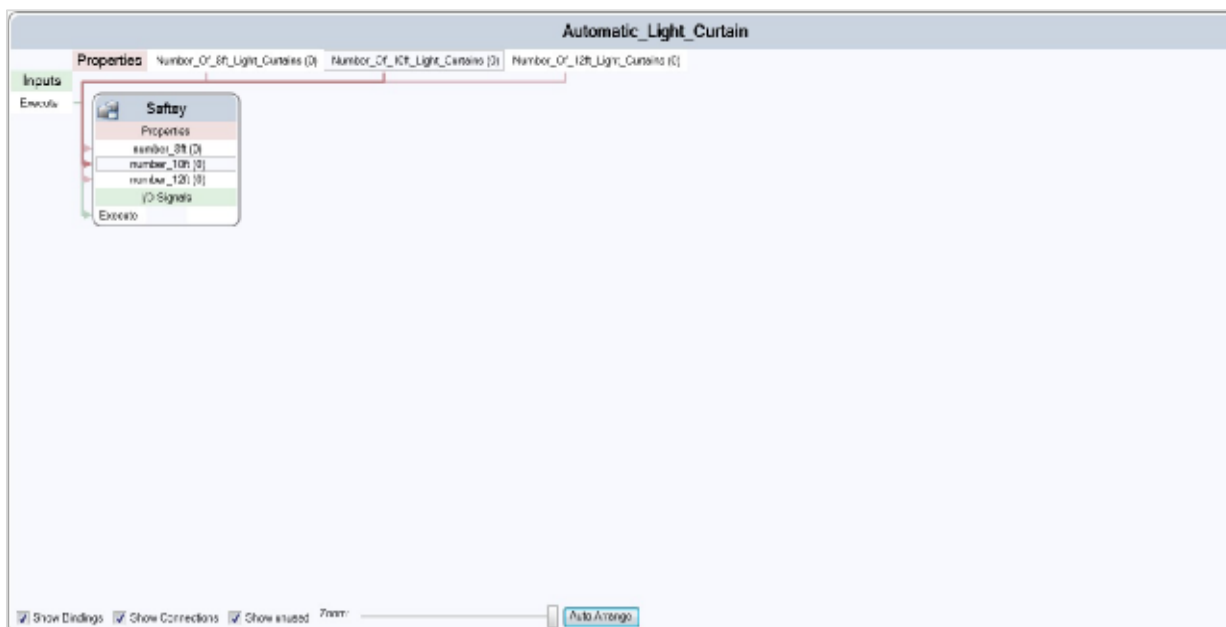


Рисунок 3.18 Автоматична світлова завіса RobotStudio знаходиться в Додатку

А.

### **3.4 Конструювання додатку**

Надбудова позиціонерів була створена для зовнішнього розгортання для використання іншими особами, що не належать до компанії. З цієї причини він був побудований у простій у використанні надбудові, яку можна було встановити на віддаленому комп'ютері з необхідними зовнішніми бібліотеками. Цей код був розроблений для створення позиціонера SkyHook, позиціонера SkyHook з підйомом, комбінованого запасу головного запасу, запасу головного запасу з підйомом або Центру падіння за допомогою спільних геометрій для обмеження необхідного розміру зовнішньої бібліотеки.

Програма починається зі створення власної вкладки та встановлення цієї вкладки для поточного подання. Усередині вкладки він створює свою першу кнопку Створити SkyHook і створює необхідний рівний обробник. Зображення на кнопці встановлено як внутрішній ресурс під назвою SkyHook. Потім вкладка та кнопка додаються до програми RobotStudio. Розділ коду, показаний нижче, рисунок 3.19, показує, як кнопка SkyHook створюється та додається на вкладку Wolf Positioners.



```

//Create a tab
RibbonTab ribbontab = new RibbonTab("Wolf Positioners", "Wolf Positioners");
UIEnvironment.RibbonTabs.Add(ribbontab);

//Make tab active
UIEnvironment.ActiveRibbonTab = ribbontab;

//Create a button Group
RibbonGroup ribbongroup = new RibbonGroup("SkyHook", "SkyHook");

//Create skyhook button
CommandBarButton button = new CommandBarButton("Create SkyHook");
button.Caption = "Create SkyHook";
button.HelpText = "Create a SkyHook";
button.Enabled = true;
button.Image = Properties.Resources.SkyHook;

//Add button event
button.ExecuteCommand += new ExecuteCommandEventHandler(button_ExecuteCommand);

//Add button to ribbon group
ribbongroup.Controls.Add(button);

//Add ribbongroup to tab
ribbontab.Groups.Add(ribbongroup);

```

Рисунок 3.19 Позиціонери Class1

Потім станція чекає натискання кнопки. Коли кнопка SkyHook активована, вона створює форму, докладно описану у Form1 та Form1.Designer. Ця форма з'являється у вікні користувача та чекає, коли буде змінено один із вибраних варіантів. Коли вибір змінюється, він оновлює інші поля вибору, щоб відобразити параметри, доступні у стандартних бібліотеках Wolf. На рисунку 3.20 показані варіанти кидання comboBox2 та падіння comboBox3, якщо вибрано ємність 1000 кг.

```

if (comboBox1.Text == "1000Kg")
{
    comboBox2.Items.AddRange(new object[] { "400", "500", "600", "700", "800", "900", "1000"
});
    comboBox3.Items.AddRange(new object[] { "400", "500", "600", "700", "800", "900",
"1000", "1100", "1200" });
}

```

Рисунок 3.20 Позиціонери Form1

Коли користувач закінчує вибір і натискає кнопку ОК, ємність, підйом і кидок зберігаються в класі і виконується функція побудови. Опинившись у функції побудови, програма завантажує геометрії основи, важеля та пластини та розміщує їх відносно вибраних методів кидання та падіння. Потім вибирається стояк із більшого падіння і кидка. На наведеному нижче сегменті коду, рисунок 3.21, показано завантажуваний стояк, якщо падіння більше, ніж кидок. Drop має додаткове зміщення 800 мм для компенсації двигуна, який обертає пластину.

```
if ((Convert.ToInt32(settings.skyhook_drop) + 800) > Convert.ToInt32(settings.skyhook_throw))
{
    string riser_path = "S:\\Library Build\\Custom Lib\\Riser\\WTU5000 Risers\\";
    mylib = GraphicComponentLibrary.Load(riser_path + "Riser_WTU5000_" +
Convert.ToString(Convert.ToInt32(settings.skyhook_drop) + 800) + "H.rslib", true);
    Part Riser = (Part)mylib.RootComponent.CopyInstance();
    Riser.Name = "Riser";
    station.GraphicComponents.Add(Riser);
    Riser.Transform.RY = -90 * (Math.PI / 180);
    Riser.Transform.RZ = -180 * (Math.PI / 180);
    Riser.Transform.Z = (Convert.ToInt32(settings.skyhook_drop)) * .001;
    Riser.Transform.X = (-1811.439 - (Convert.ToInt32(settings.skyhook_throw) - 1500))
* .001;

    GraphicComponent L1;
    mech.GetParentLink(0, out L1);
    Part attacher = (Part)L1;
    attacher.Attach(Riser, false, new Matrix4(new Vector3(0, 0, 0)));
}
```

Рисунок 3.21 Позичіонери Class1

Після завантаження та розміщення відповідної геометрії створюється механізм SkyHook. Наведений нижче сегмент коду, рисунок 3.22, показує завантажену геометрію, додану до ланок та стиків, доданих до ланок. Два вектори диктують, де слід розмістити суглоб і по якій осі він рухатиметься - обидва суглоби встановлені в обертальний режим. Після встановлення спільних обмежень механізм компілюється, додається до станції та називається.

```

MechanismBuilder mb = new MechanismBuilder(MechanismType.ExternalAxis);
mb.ModelName = "SkyHook";
mb.AddLink("Base", Base);
mb.AddLink("Link1", Arm);
mb.AddLink("Link2", Platter);
mb.BaseLink = "Base";
mb.AddJoint("J1", "Base", "Link1", new Vector3(1, 0,
(Convert.ToInt32(settings.skyhook_drop) * .001)), new Vector3(0, 0,
(Convert.ToInt32(settings.skyhook_drop) * .001)), JointType.Rotational, true);
mb.AddJoint("J2", "Link1", "Link2", new Vector3(0, 0, 0), new Vector3(0, 0, 1),
JointType.Rotational, true);
mb.SetJointLimit("J1", -180 * (Math.PI / 180), 180 * (Math.PI / 180));
mb.SetJointLimit("J2", -360 * (Math.PI / 180), 360 * (Math.PI / 180));
Mechanism mech = mb.CompileMechanism();
station.GraphicComponents.Add(mech);
mech.Name = "SkyHook";

```

Рисунок 3.22 Позичіонери Class1

## 4 НАУКОВО-ДОСЛІДНА ЧАСТИНА

### 4.1 Створення автоматичного конструктора робототехнічної комірки

Компонент Automatic Station Builder був побудований для подальшого використання розроблених інтелектуальних компонентів. На рисунку 4.1 показано загальний компонент компонента Builder Station. Для підтримки модульності був розроблений користувальницький інтерфейс RobotStudio. Створюючи цей загальний компонент, окремі інтелектуальні компоненти можуть бути додані або відредаговані без повторного підключення блоків коду RobotStudio. Цей графічний користувальницький інтерфейс був підключений до автоматичного конструктора станцій, щоб важливі змінні могли передаватися компоненту "Конструктор станцій" і за необхідності проводити розрахунки. Цей компонент діяв як головний мозок усього проекту. Цей компонент відповідає за завантаження станції відповідними компонентами, видалення старих компонентів та розміщення всіх необхідних компонентів. Потім цей компонент підключається безпосередньо до огороження підкомпонентів та автоматичного конструктора шляхів. Кожен із цих компонентів має понад 40 окремих блоків коду та провідів - не показано.

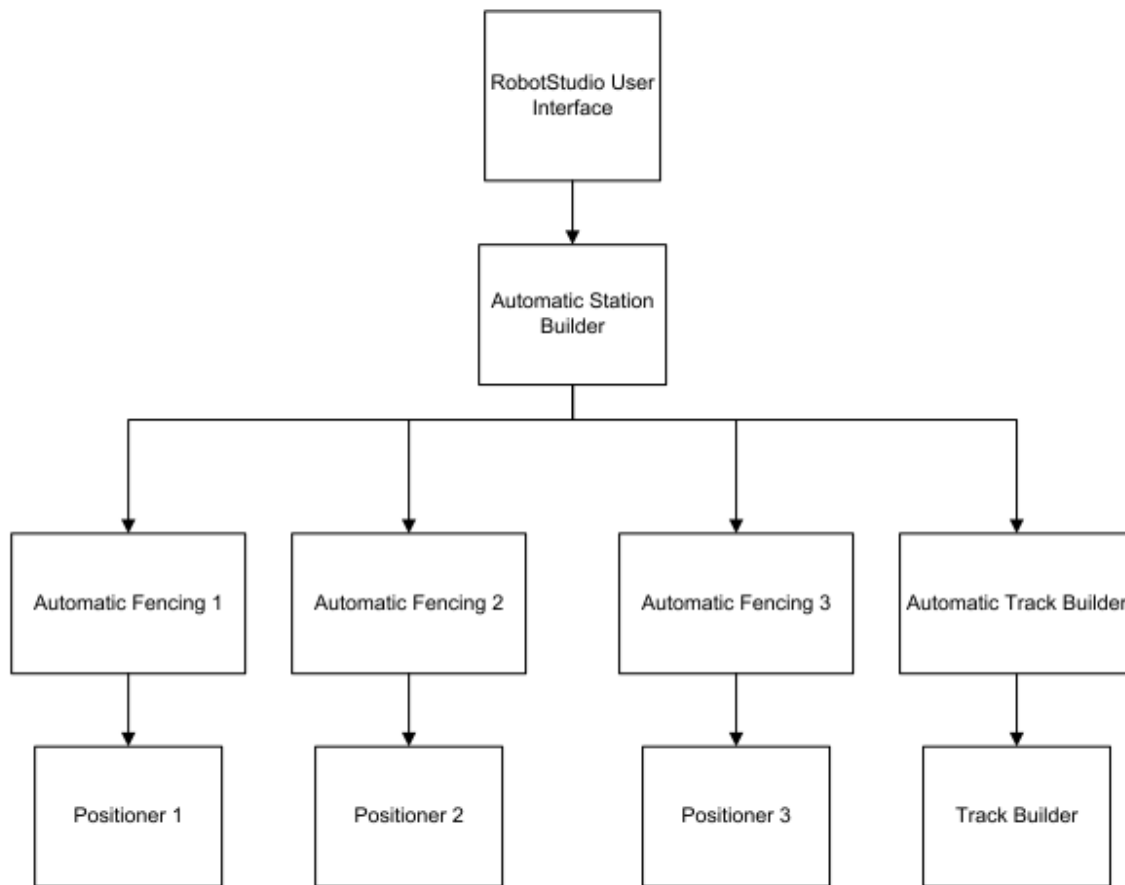


Рисунок 4.1 Макет програми автоматичного побудови станції

Після того, як змінні передаються в Автоматичний конструктор станцій, виконуються розрахунки, щоб переконатися, що створена станція здатна зварювати потрібну деталь. Наведений нижче сегмент коду, рисунку 4.2, показує, як обирається стояк та потужність HSTS. Видно, якщо вага деталі менше максимальної ємності комбінації HSTS, але перевищує попередню HSTS, в даному випадку 0 кг, ця потужність обрана. Потім перераховується список стояків для обраної потужності HSTS, використовуючи половину діаметра, щоб визначити відповідну висоту центру повороту.

```

//Select HSTS
if (Convert.ToInt32(Orig_Weight) >= 0 & Convert.ToInt32(Orig_Weight) < 3000)
{
    CodeBehind.settings.HSTS_capacity = "3000Kg";

    //Select Riser
    string[] temp = { "0", "1000", "1100", "1200", "1300", "1400", "1500", "1600", "1700", "1800",
"1900", "2000", "2100", "2200", "2300", "2400", "2500" };
    for (int i = 0; i < 15; i++)
    {
        if (Convert.ToInt32(Orig_Diam)/2 >= Convert.ToInt32(temp[i]) & Convert.ToInt32(Orig_Diam)/2 <
Convert.ToInt32(temp[i + 1]))
        {
            CodeBehind.settings.HSTS_riser_height = temp[i + 1];
        }
    }
}
}

```

Рисунок 4.2 Код побудови автоматичної станції

На рисунку 4.3 показано, як висота огорожі та висота порталу вибираються з висоти стояка, яка була обрана на основі діаметра потрібної деталі. Цей код показує, якщо стояк піднімає центр повороту деталі на висоту менше 2500 мм, слід завантажити 2500 козлів.

```

//Select Gantry Height and Fence Height
if (Convert.ToInt32(CodeBehind.settings.HSTS_riser_height) < 2500)
{
    component.Properties["Track_Gantry"].Value = ("Gantry_2500H");
    component.Properties["Fence_Height"].Value = (8);
}

```

Рисунок 4.3 Код побудови автоматичної станції

Подальші розрахунки для визначення довжини огорожі та розміщення кожного компонента можна побачити на рисунку 4.4. Цей код показує зміщення, використані для комбінації HSTS на 20000 кг. Тут була використана функція `Math.Ceiling` для обліку помилок округлення в різних розумних компонентах.

```

if (settings.HSTS_capacity == "20000Kg")
{
    component.Properties["Fence_Length1"].Value = Math.Ceiling((4000 + CodeBehind.settings.RTT) *
3.28084 * .001);
    component.Properties["Fence_Length2"].Value = Math.Ceiling(3.28084 *
((Convert.ToInt32(CodeBehind.settings.HSTS_riser_height) + 4000) * .001));
    component.Properties["Fence_Length3"].Value = Math.Ceiling(3.28084 *
((Convert.ToInt32(CodeBehind.settings.HSTS_riser_height) + 4000) * .001));

    component.Properties["Fence_Pos3"].Value = (new Vector3((-2000 * .001), (Math.Ceiling((4000 +
CodeBehind.settings.RTT) * 3.28084 * .001) * 309.48 - 2951.24) * .001, 0));

    CodeBehind.settings.Curtain_Transform_Y = -3095.887 * .001;
    CodeBehind.settings.Curtain_Transform_X = (Math.Ceiling(3.28084 * .001 *
Convert.ToInt32(CodeBehind.settings.HSTS_riser_height)) * 309.48 + 1961.08 + 295.44) * .001;
    CodeBehind.settings.Curtain_Len = (Math.Ceiling((4000 + CodeBehind.settings.RTT) * 3.28084 *
.001) * 309.48 - 2160.69);
}

```

Рисунок 4.4 Код побудови автоматичної станції

Кореневе рівняння, що використовується у зміщеннях огорожі та зміщенні світлової завіси, було визначено експериментально. Залежно від бажаної довжини огорожі потрібна різна кількість сегментів огорожі. Додаткові сегменти огорожі також створюють додатковий зсув від доданих стовпів огорожі. Були завантажені різні довжини огорожі та виміряно центр останнього стовпа від стартового кута. Ці результати були зображені на рисунку 4.5, щоб визначити, чи можна знайти простий зв'язок. На щастя, було знайдено абсолютно лінійну залежність, яку зафіксував  $R^2=1$ . Потім були потрібні додаткові зсуви, щоб компенсувати різні розміри позиціонерів та їх розміщення.

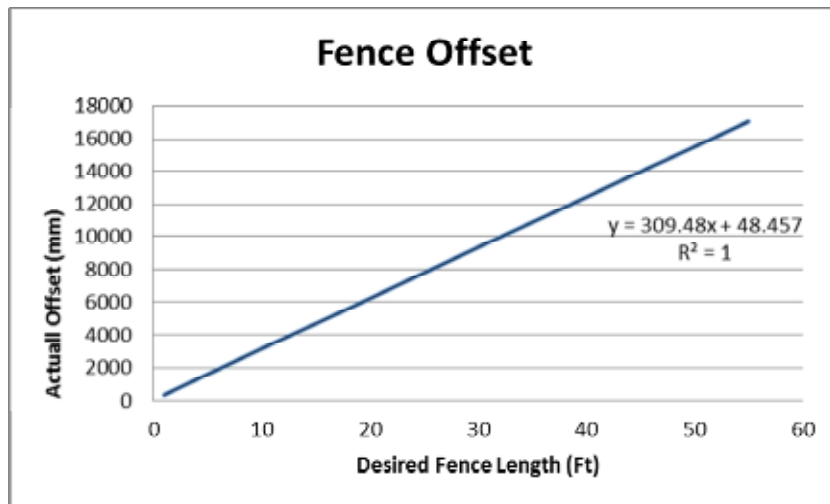
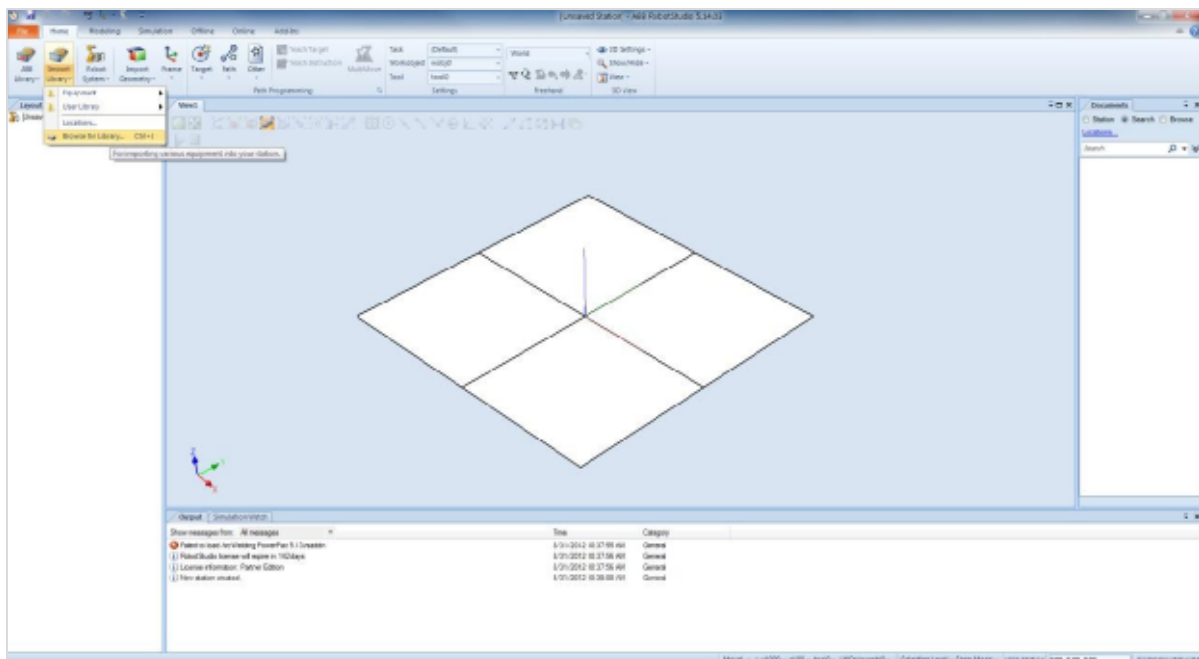


Рисунок 4.5 Розрахунок зміщення огорожі

#### 4.2 Створення робото технічної комірки без надбудови

На вкладці Домашня сторінка RobotStudio виберіть Імпорт бібліотеки, а потім Огляд бібліотеки. Це видно на рисунку 4.6. Перейдіть до папки, яка містить попередньо побудовані бібліотеки Head Stock. Виберіть запас голівки 30000 кг і виберіть Load.





#### Рисунок 4.6 Бібліотека імпорту RobotStudio

Для того, щоб побачити запас голови, може знадобитися деяка репозиція. Це може бути здійснено за допомогою інструменту переміщення або перестановки головного запасу за допомогою системи координат, як показано на рисунку 4.7. Важливо відсунути його від світових координат (0,0,0), оскільки саме тут бути завантаженим.

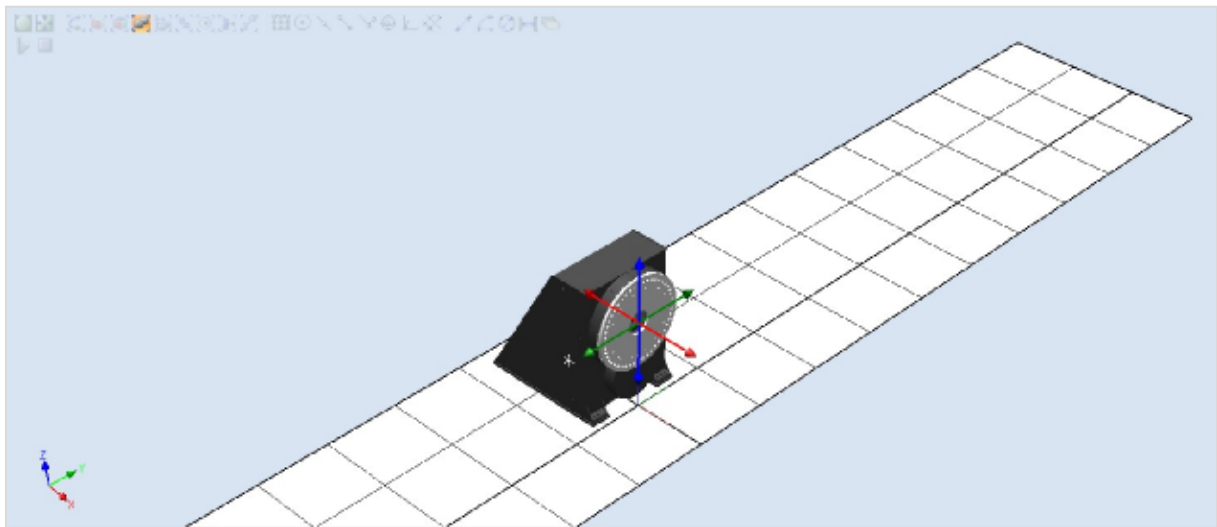


Рисунок 4.7 RobotStudio Position Head Stock

Знову ж таки, використовуйте кнопку Імпортувати бібліотеку, щоб завантажити відповідний запас хвоста. Переорієнтуйте та розташуйте хвостовий запас напроти головного запасу, підкладку цього можна знайти на рисунку 4.8. На даний момент не потрібно встановлювати хвостовий запас точно навпроти головного запасу.

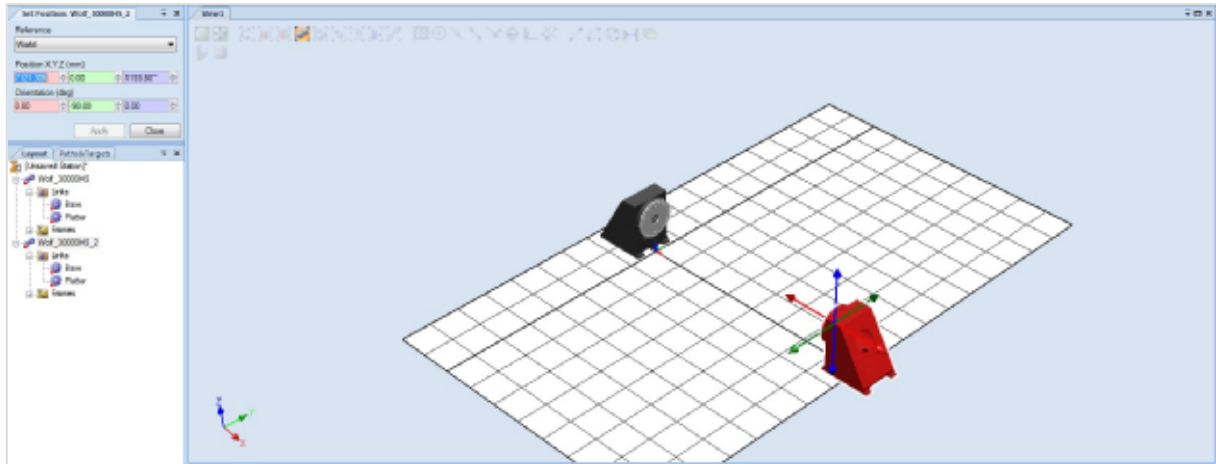


Рисунок 4.8 RobotStudio Position Tail Stock

Використовуйте бібліотеку імпорту, щоб завантажити два стояки на 2500 мм для запасу головки на 30000 кг і запасного хвоста на 30000 кг. Переорієнтуйте стояки так, щоб отвори для болтів були зверху, і розташуйте їх на підлозі. Перемістіть головку в таке положення, щоб було видно отвори для болтів внизу, як показано на рисунок 4.9. Використовуйте функцію розміщення однієї точки і виберіть один із отворів для болтів. Для другого пункту виберіть відповідне отвір для болта на стояку. Це перемістить головку в правильне положення на стояку.

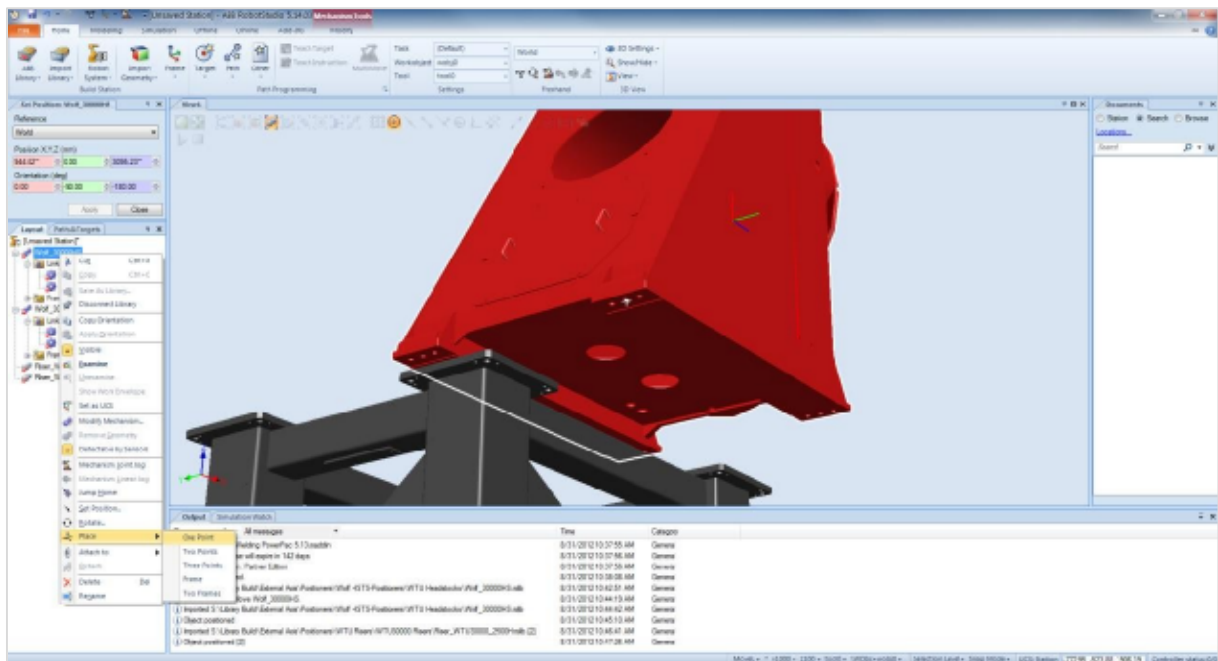


Рисунок 4.9 RobotStudio Place Headstock on Riser

Тепер прикріпіть головку до її стояка. Це дозволить користувачеві переміщувати компоненти разом. Це повинно виглядати схоже на приклад, зображений на рисунку 4.10. Використовуйте цю процедуру ще раз, щоб прикріпити хвостовий запас до його стояка.

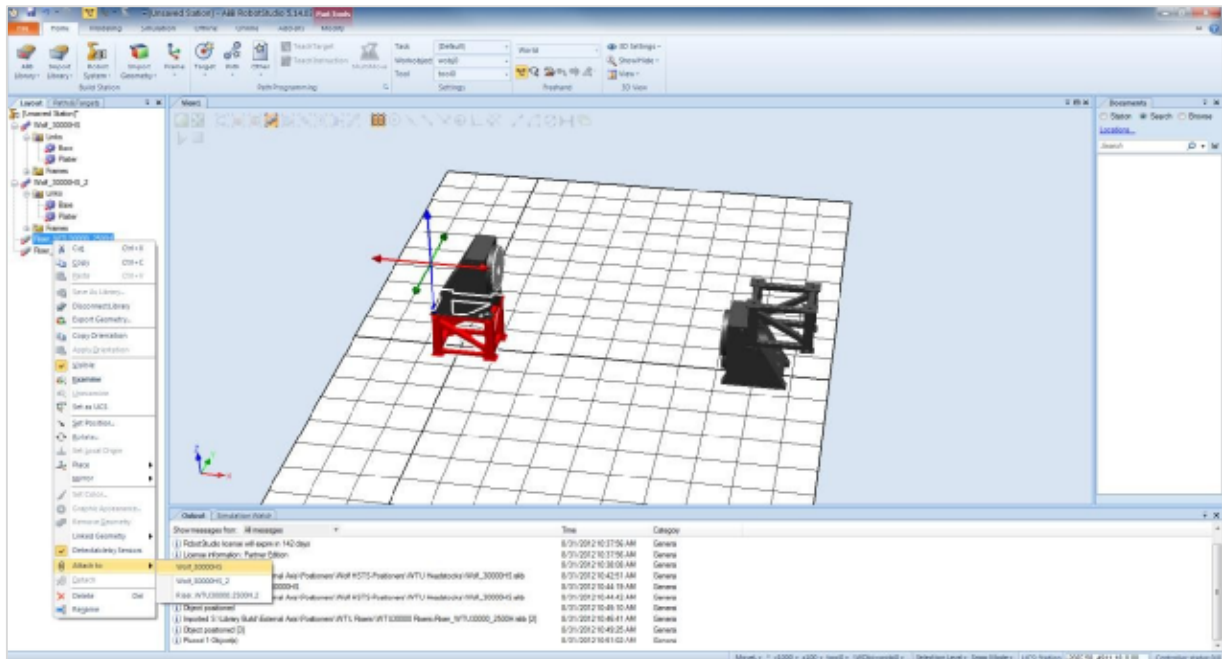
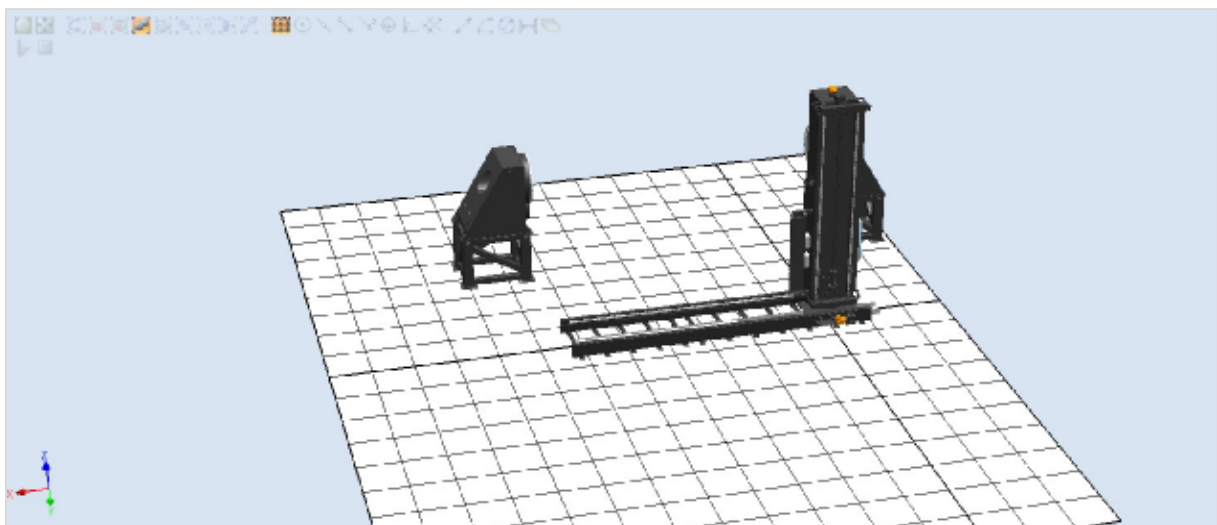


Рисунок 4.10 Attach Head Stock to Riser

Навантажуйте важку колію довжиною 7350 мм. Помістіть доріжку за світовими координатами (0,0,0). Завантажте до станції 3500-мм портал і розмістіть його за світовими координатами (0,0,0) При завантаженні він повинен мати портал і колію 7350 мм, як показано на рисунку 4.11.



#### Рисунок 4.11 RobotStudio Load Robot Travel Track

Завантажте стрілу 2500 мм і орієнтуйте її у правильному положенні обертання відповідно до порталу. Знову ж таки, використовуйте функцію розміщення однієї точки та отвори для болтів, щоб розмістити стрілу. Прикріпіть стрілу до ланки 4 козлової. Коли станція запитає, чи слід зберігати поточне положення, виберіть так. Прикріплену стрілу можна побачити на рисунку 4.12.

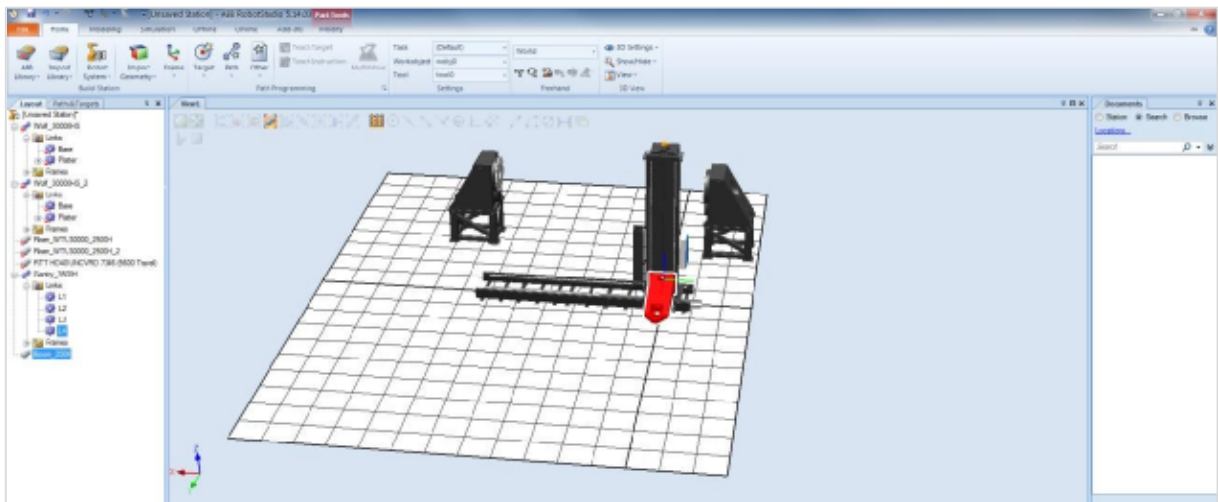


Рисунок 4.12 RobotStudio Load Boom

Зверніть увагу, що при використанні механізму шарнірного пробігу можливе переміщення порталу повністю з колії, як показано на рисунку 4.13. Ця стріла встановлюється на будь-яку довжину колії.

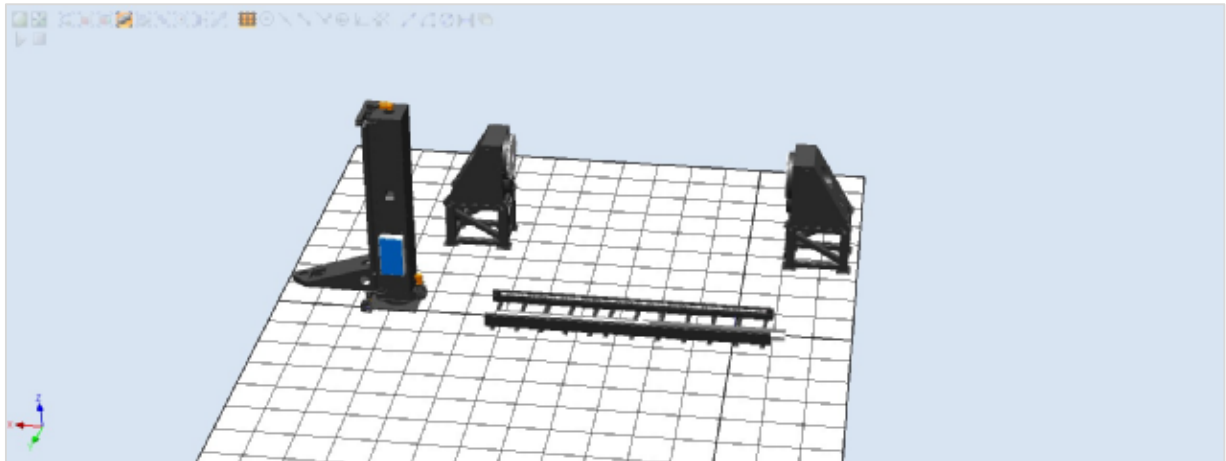


Рисунок 4.13 RobotStudio Modifying Gantry Mechanism

Завантажте кулер для води Lincoln, однопровідну бочку з низьким кріпленням, очищувач пальника та бичаче вушко Wolf. Перемістіть ці частини, щоб усі вони були на виду. Приклад цього можна побачити на рисунок 4.14.

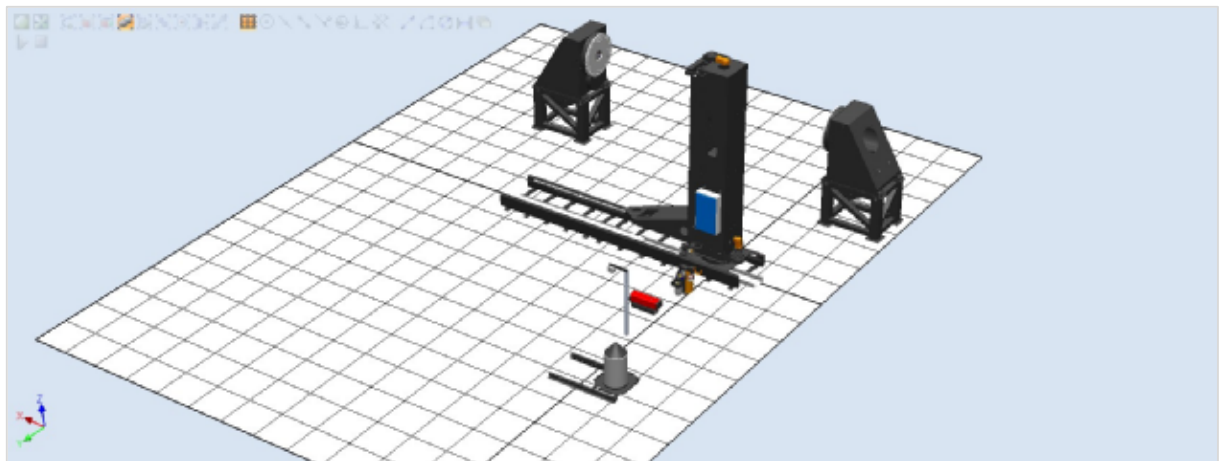


Рисунок 4.14 RobotStudio Load Additional Tooling

Індивідуально перемістіть кожну деталь у відповідне місце на козлі. Опинившись на місці, прикріпіть його до відповідної форми. Для Wolf Bullseye та управління дротом виберіть посилення чотири. Для дротяної бочки

та охолоджувача води виберіть зв'язок три. Остаточна збірка повинна виглядати приблизно так, як показано на рисунку 4.15.

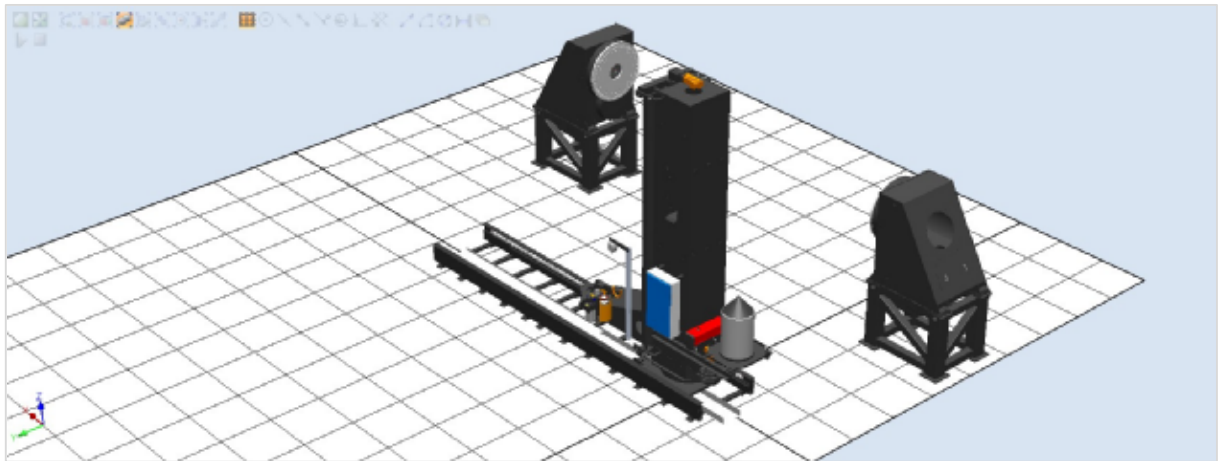


Рисунок 4.15 RobotStudio Attach Tooling

Тепер огорожа повинна бути завантажена. Потрібно завантажити вісім секцій огорожі 10x40 і дві секції 10x5. Обертайте кожну деталь, доки вздовж осі x не залишиться чотирьох секцій, а вздовж осі y - шість секцій. Два навантажені шматки огорожі видно на рисунку 4.16.

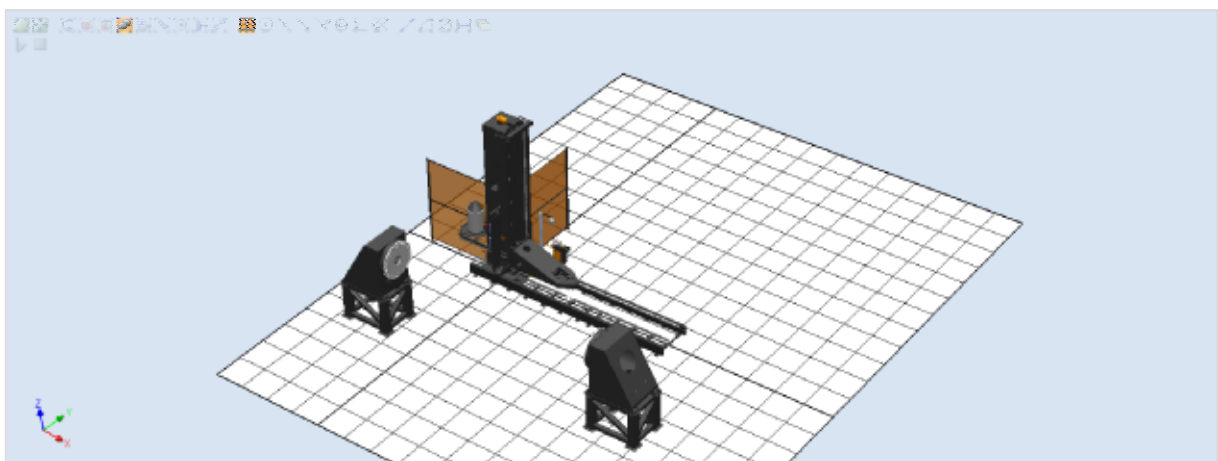


Рисунок 4.16 RobotStudio Load Fencing

Кожен шматок огорожі повинен бути розміщений індивідуально. Це можна зробити точно, використовуючи систему координат, або використовуючи інструмент переміщення та підбираючи кожний стовп паркану. Помилка на багатьох ділянках огорожі буде ускладнюватися, тому найкраще, якщо для розміщення використовується система координат. На рисунку 4.17 показано два шматки огорожі, які вирівняні між собою.

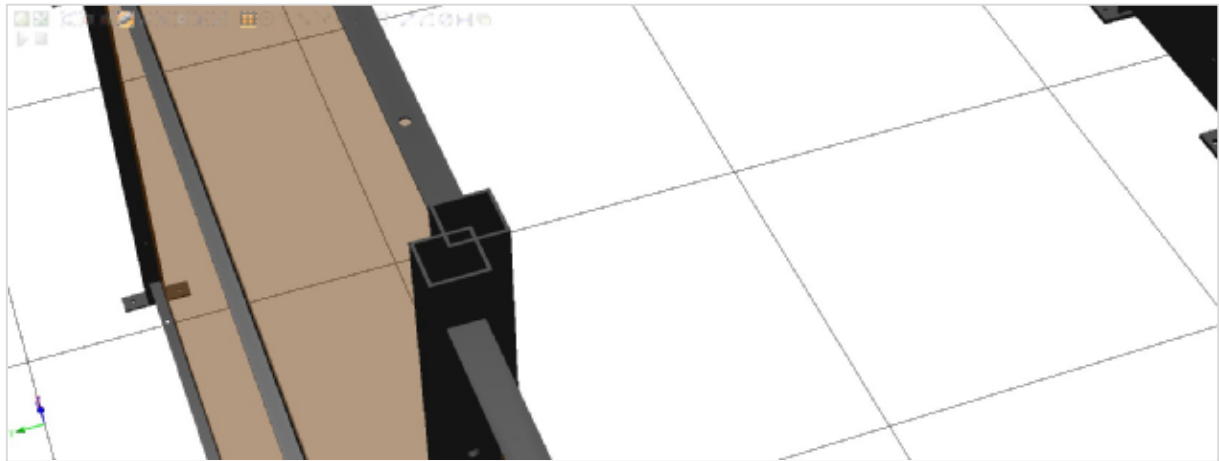


Рисунок 4.17 RobotStudio Align Fencing

Тепер можна помітити, що встановлено недостатньо огорож, Рисунок 4.18. Для того, щоб скорегувати дефіцит, деякі огорожі слід видалити окремо і занести довші шматки огорожі. Знову зорієнтуйтеся і розташуйте їх у потрібному положенні. Якщо потрібно змінити висоту огорожі, кожен шматок огорожі потрібно видалити і завести нову огорожу належної висоти. Знову кожен шматок повинен бути індивідуально орієнтований та розміщений.



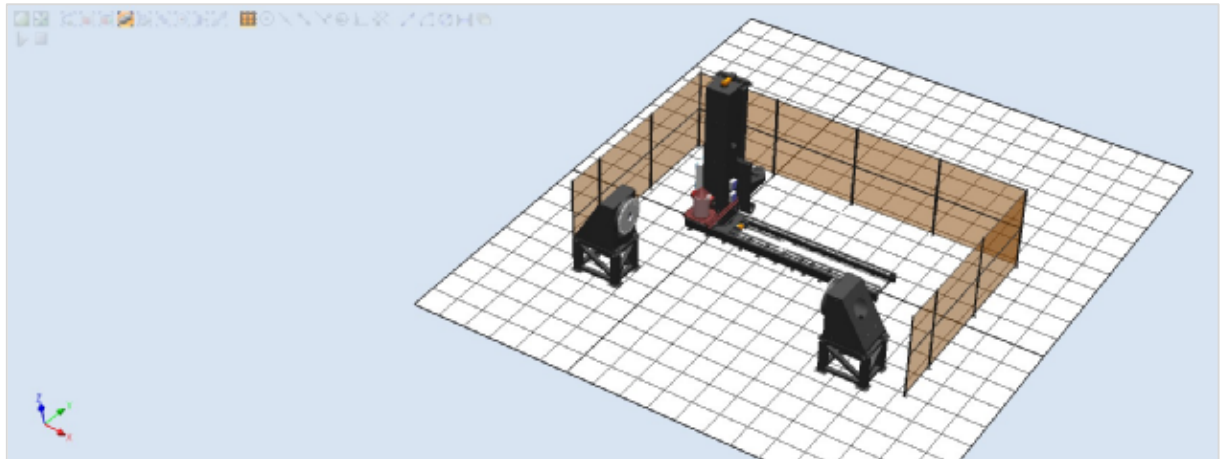


Рисунок 4.18 RobotStudio Load Additional Fencing

Після того, як були розміщені правильні сегменти огорожі, занесіть стовп легкої завіси. Орієнтуйтеся та розмістіть стовп так, щоб обидві світлові завіси були навпроти один одного та на відповідному стовпі огорожі. Будь-яку помилку в розміщенні огорожі можна легко побачити, коли світлові завіси не збігаються. Правильно розміщені огорожувальні та легкі завіси можна побачити на рисунку 4.19.

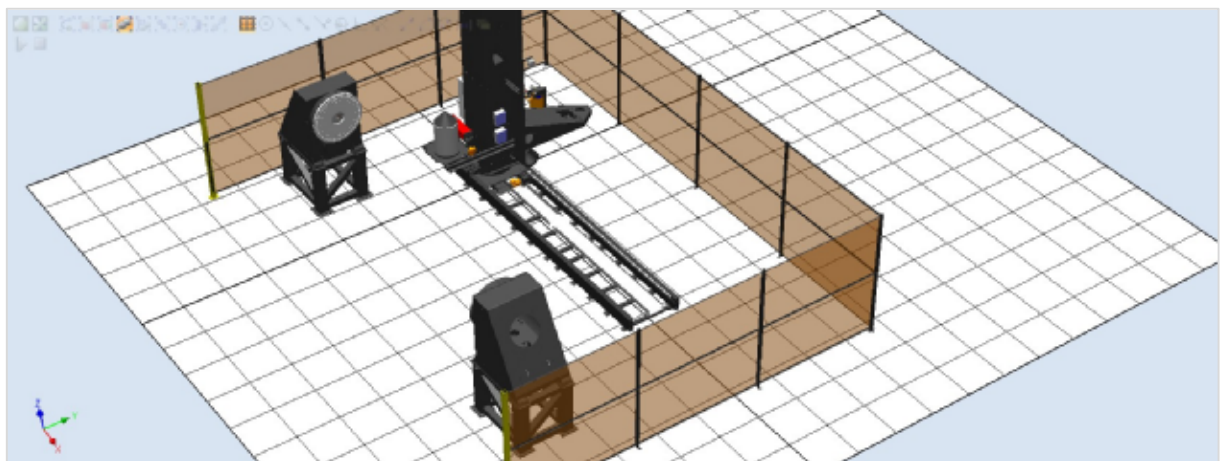


Рисунок 4.19 RobotStudio Load and Place Light Curtain

Наступним кроком є побудова штори. У моделюванні виберіть тверду, а потім коробку з трьох точок - як показано на рисунку 4.20. Виберіть правий верхній кут світлової завіси, а потім нижній правий кут протилежного стовпа огорожі. Знову ж, виберіть нижній правий кут, щоб визначити зазор, яким буде охоплювати штору огорожі.

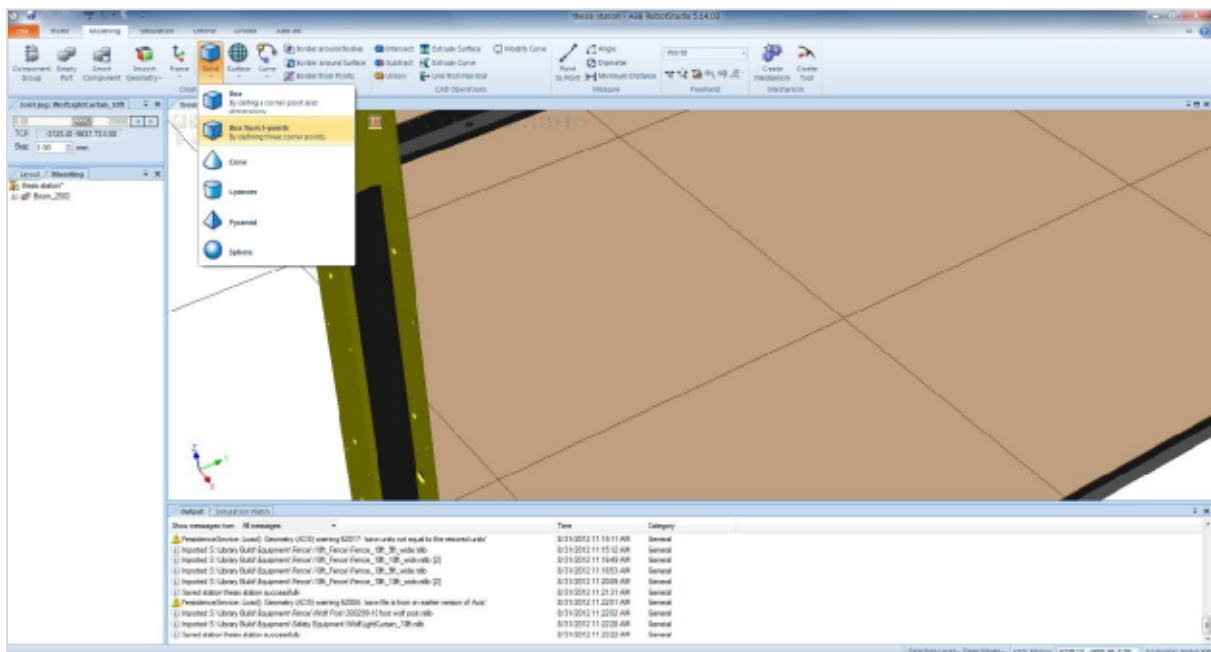


Рисунок 4.20 RobotStudio Building the Curtain Box

Виділіть щойно створене поле, і з'явиться вкладка модифікації; виділіть щойно створене вікно та оберіть графічний вигляд. У діалоговому вікні змініть колір на червоний, а прозорість - на 50%. Станцію з завершеною лазерною завісою можна побачити на рисунку 4.21.

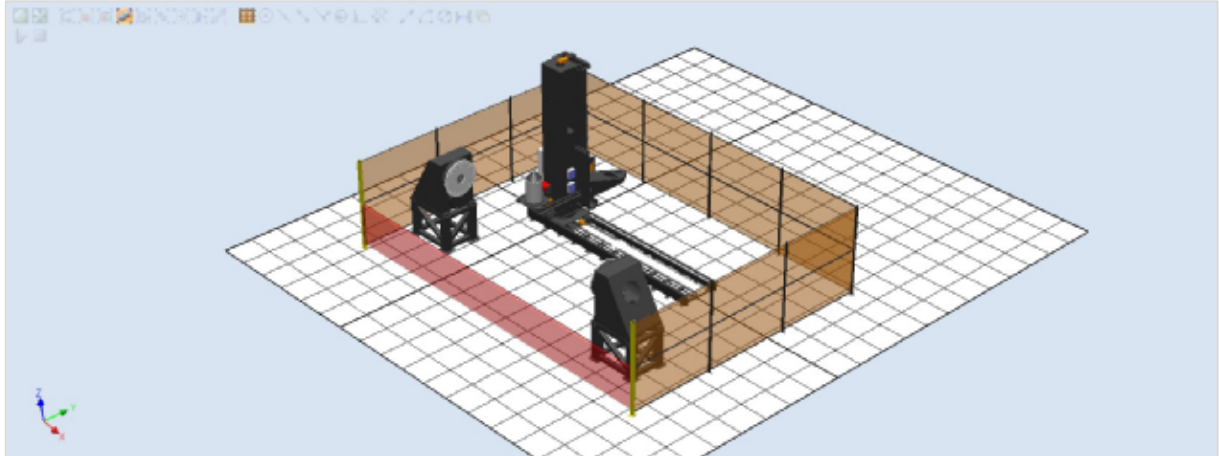


Рисунок 4.21 RobotStudio Laser Curtain Complete

Тепер можна вибрати робота, який можна завести на станцію. Використовуйте інструмент переміщення, щоб розташувати робота на стрілі. Приєднайте робота, щоб зв'язати чотири ворота. Готову до використання концептуальну станцію тепер можна побачити на рисунку 4.22.

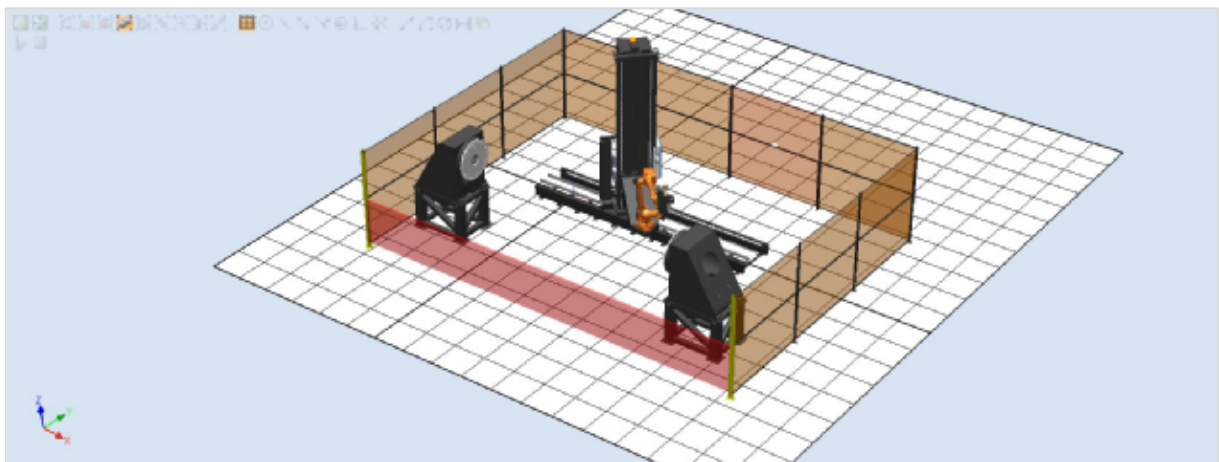


Рисунок 4.22 RobotStudio Final Concept Cell

### 4.3 Створення робототехнічної комірки з надбудовою

Це тематичне дослідження охопить основне використання надбудови та інтелектуальної серії компонентів. Почніть із створення нової станції RobotStudio. Перейдіть на вкладку Позичіонери Wolf і натисніть кнопку Створити HSTS. У новому вікні, що відкриється, виберіть ємність 30000 кг і висоту стояка 2500 мм. Клацніть на кнопку завантаження. Приклад діалогового вікна можна побачити на рисунку 4.23.

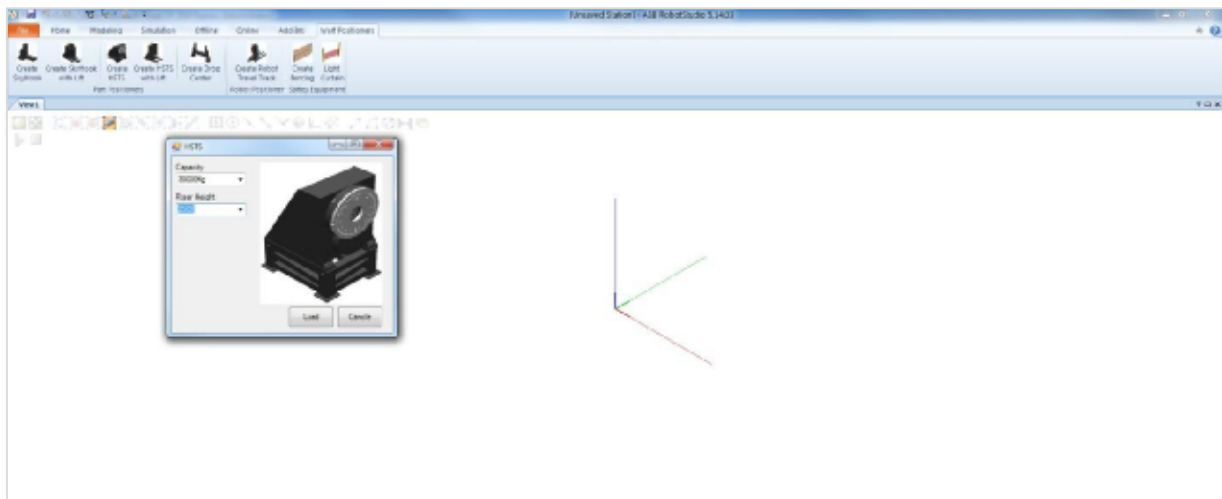


Рисунок 4.23 Add-in Load Riser

Тепер, коли позиціонер знаходиться на станції, додайте серію інтелектуальних компонентів. Це включає автоматичну колію, автоматичну світлову завісу та три автоматичні огорожі. Виберіть компонент автоматичної доріжки; для довжини доріжки виберіть 7350, для Gantry - Gantry\_3500H, для Boom - Boom\_2500. Крім того, нехай цей компонент автоматично вводить

охолоджувач води Lincoln, однопровідну бочку з низьким кріпленням, очищувач пальника та бичаче вушко Wolf. Виберіть Застосувати, щоб внести нові зміни. Це діалогове вікно видно на рисунку 4.24.

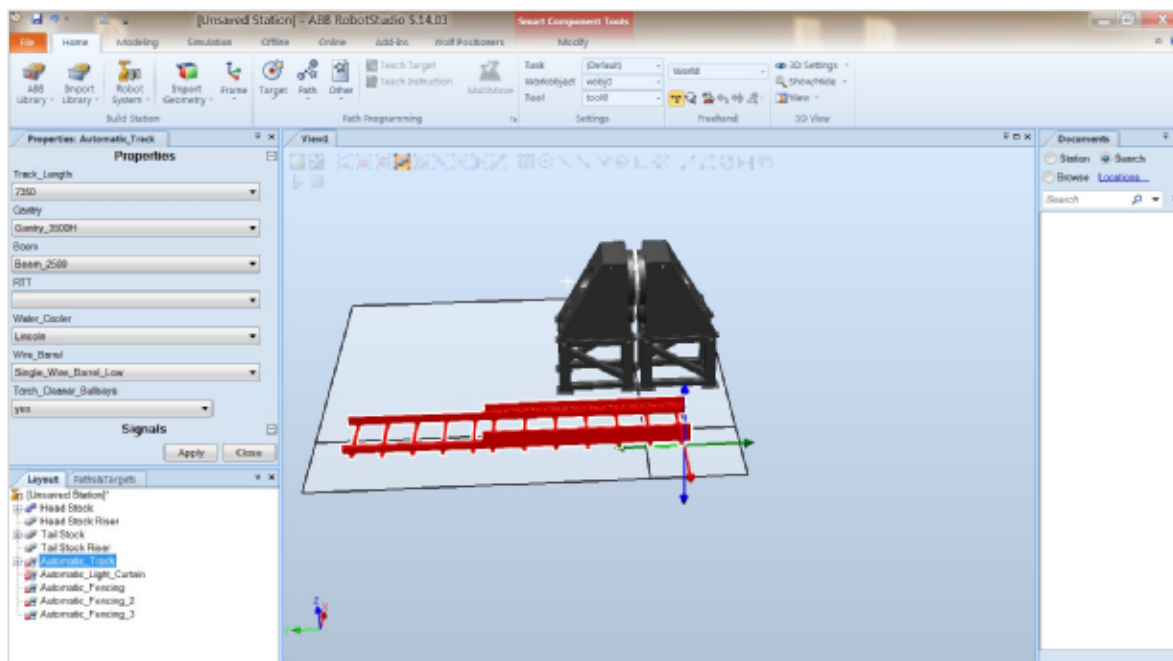


Рисунок 4.24 Add-in Load Robot Travel Track

Тепер перемістіть позиціонери до кінця дорожньої доріжки. За допомогою трьох компонентів автоматичного фехтування створіть дві секції розміром 10x40 та одну секцію 10x 40. Розмістіть 40-дюймовий паркан уздовж тильної сторони станції. Розмістіть дві секції 35 уздовж бортів станції. Це можна зробити, ввівши довжину у футах і вибравши 10 у полі вибору висоти. Натисніть кнопку застосувати, щоб завершити зміни. Після того, як усі ці зміни були застосовані, станція повинна виглядати, як показано на рисунку 4.25.

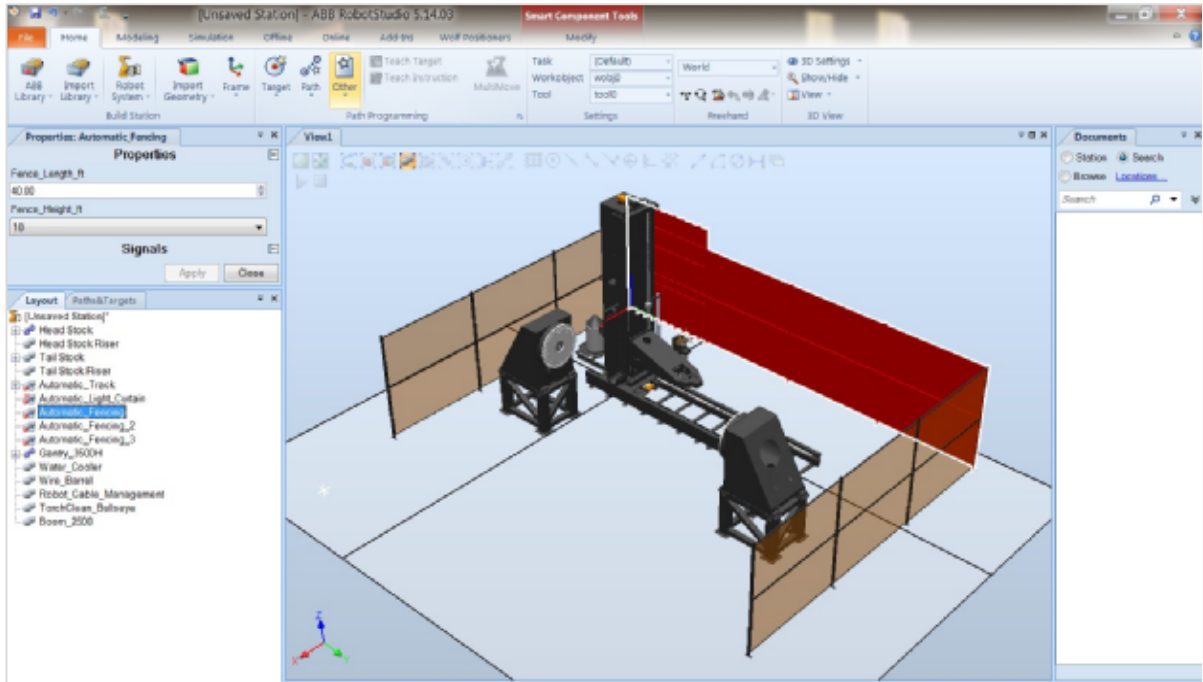


Рисунок 4.25 Add-in Load Fencing

Виберіть компонент Автоматична світлова завіса. Виберіть одну 10 метрів світлову завісу та виберіть Застосувати, а потім Виконати. Перемістіть світлову завісу з нульового світу туди, де її можна побачити. Поставте одну стійку світлової завіси на останню стійку огорожі. За допомогою механізму суглобової пробіжки перемістіть інший стовп світлової завіси так, щоб він знаходився зверху протилежного стовпа огорожі. Світлова завіса тепер повинна бути розташована подібно до тієї, що видно на рисунку 4.26.

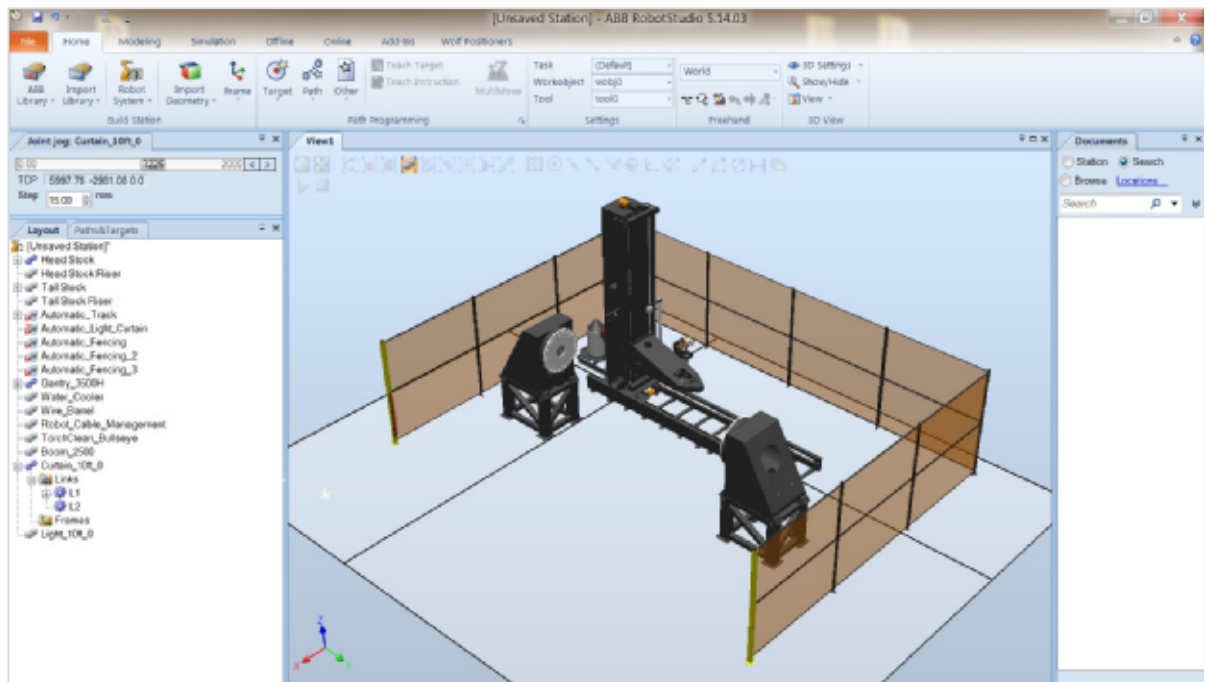


Рисунок 4.26 Add-in Load Light Curtain

Тепер, коли позиціонер знаходиться на станції, додайте серію інтелектуальних компонентів. Це включає автоматичну колію, автоматичну світлову завісу та три автоматичні огорожі. Виберіть компонент автоматичної доріжки; для довжини доріжки виберіть 7350, для Gantry - Gantry\_3500H, для Boom - Boom\_2500. Крім того, нехай цей компонент автоматично вводить охолоджувач води Lincoln, однопровідну бочку з низьким кріпленням, очищувач пальника та бичаче вушко Wolf. Виберіть Застосувати, щоб внести нові зміни. Це діалогове вікно видно на рисунок 4.27.

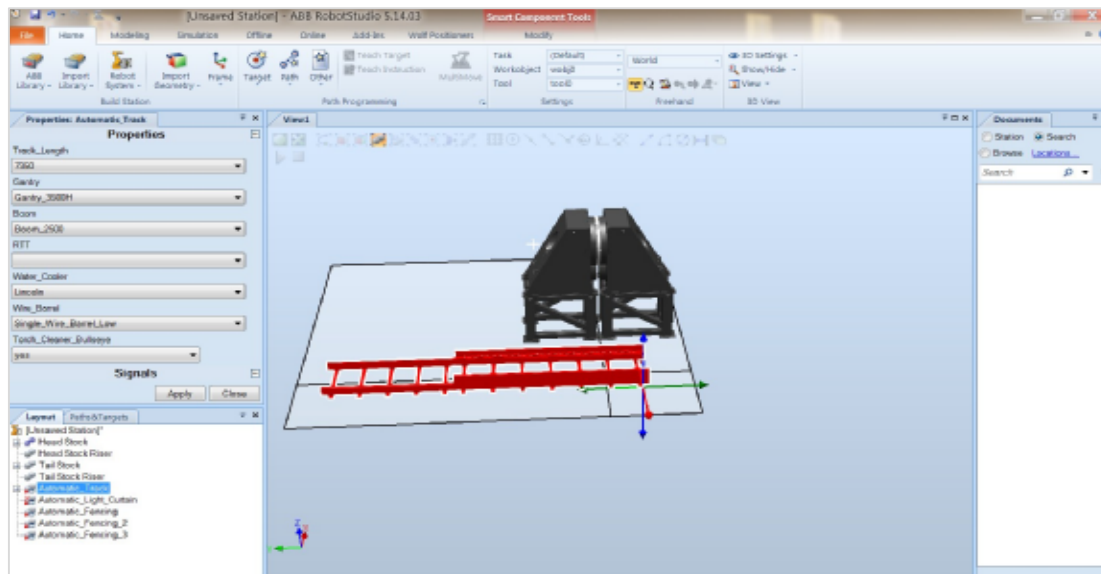
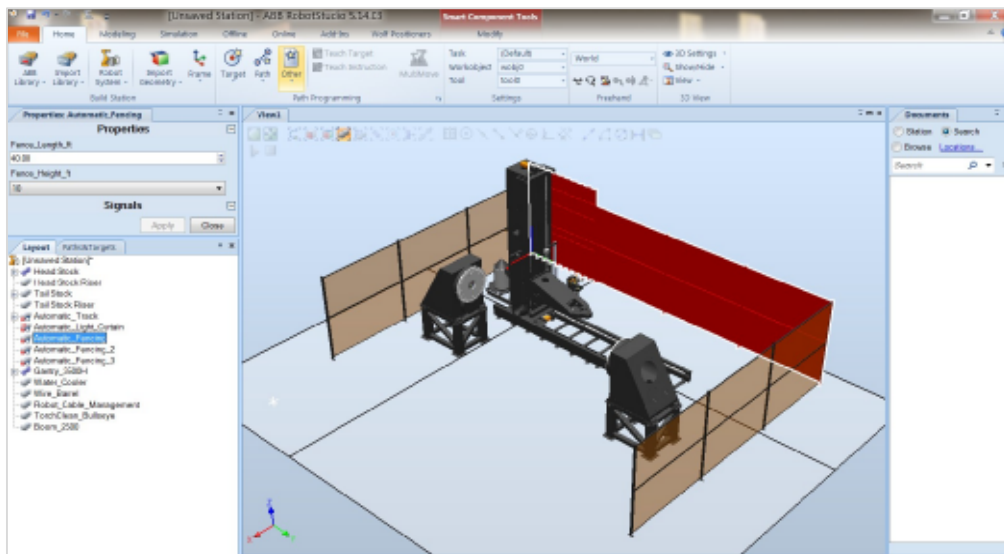


Рисунок 4.27 Add-in Load Robot Travel Track

Тепер перемістіть позиціонери до кінця дорожньої доріжки. За допомогою трьох компонентів автоматичного фехтування створіть дві секції розміром 10x40 та одну секцію 10x40. Розмістіть 40-метровий паркан уздовж тильної сторони станції. Розмістіть дві секції 35 уздовж бортів станції. Це можна зробити, ввівши довжину у футах і вибравши 10 у полі вибору висоти. Натисніть кнопку застосувати, щоб завершити зміни. Після того, як усі ці зміни були застосовані, станція повинна виглядати, як показано на рисунку 4.28.





Рисунк 4.28 Add-in Load Fencing

Виберіть компонент Автоматична світлова завіса. Виберіть одну 10-метрову світлову завісу та виберіть Застосувати, а потім Виконати. Перемістіть світлову завісу з нульового світу туди, де її можна побачити. Поставте одну стійку світлової завіси на останню стійку огорожі. За допомогою механізму суглобової пробіжки перемістіть інший стовп світлової завіси так, щоб він знаходився зверху протилежного стовпа огорожі. Світлова завіса тепер повинна бути розташована подібно до тієї, що видно на рисунку 4.29.

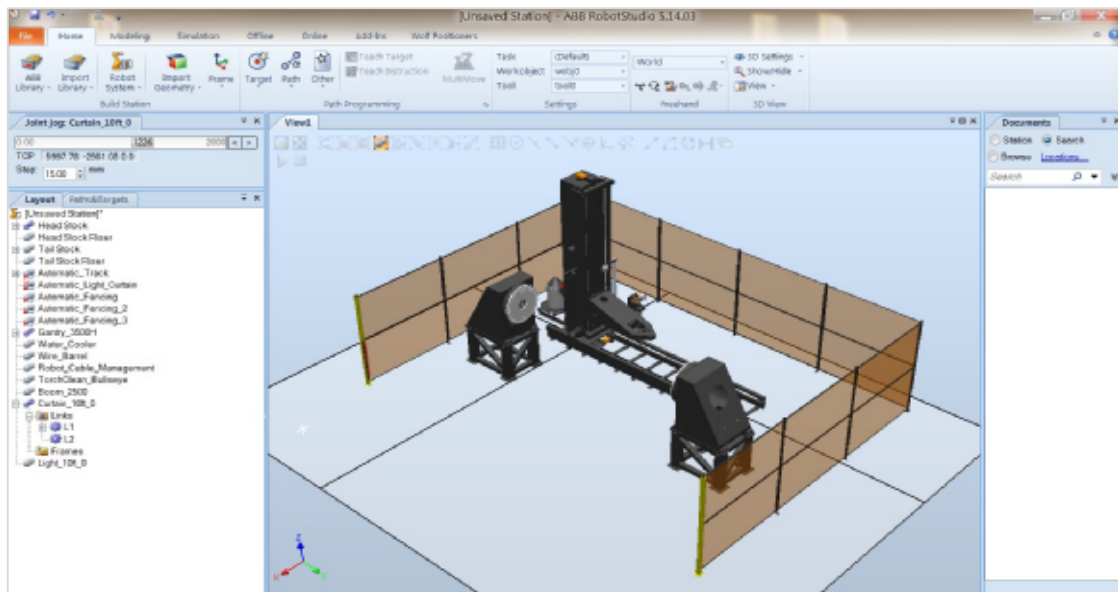


Рисунок 4.29 Add-in Load Light Curtain

Поверніться до компонента Автоматична світлова завіса та виберіть Виконати, щоб оновити світлову завісу. Це створить візуальне зображення лазера, який покриває отвір комірки, як показано на рисунку 4.30.

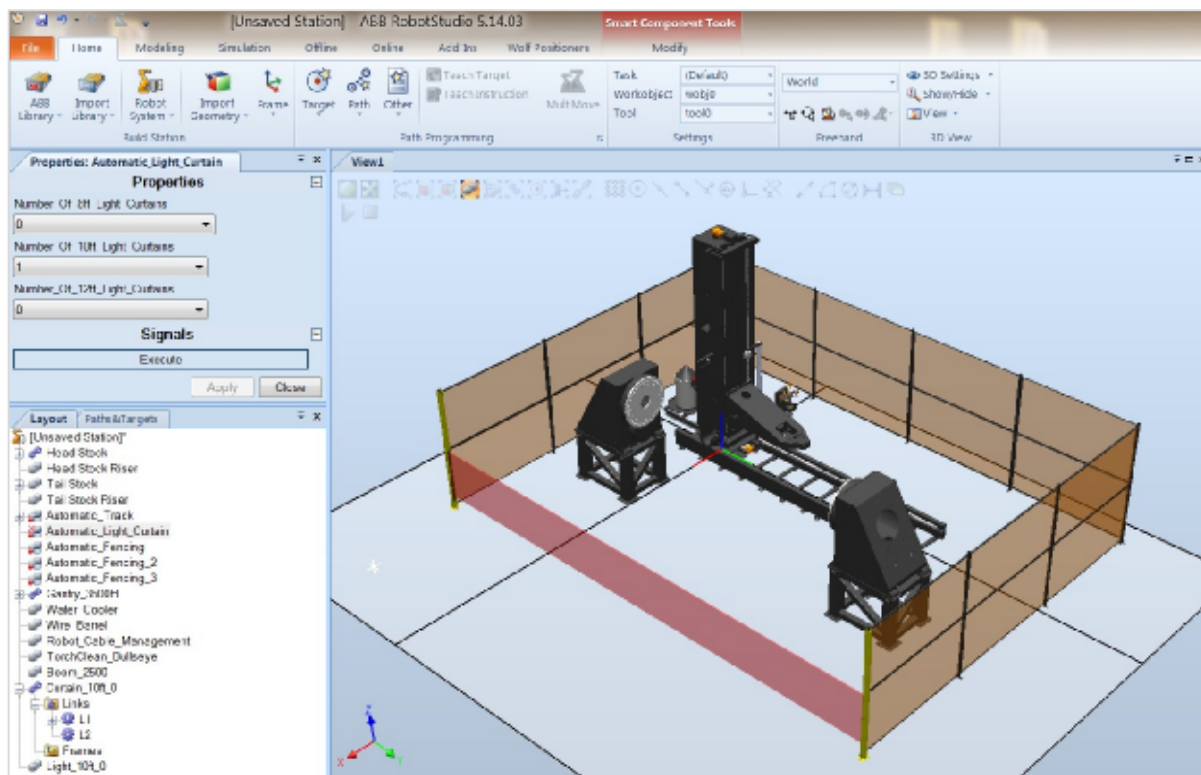


Рисунок 4.30 Add-in Update Laser Curtain

Нарешті принесіть робота і прикріпіть його до порталу. Зараз станція готова залучити частину клієнтів та прикріпити її до головного фонду. Після цього контролер, який відповідає механізмам станції, може бути завантажений. Зараз станція готова до програмування. Завершену станцію тепер можна побачити на рисунку 4.31.

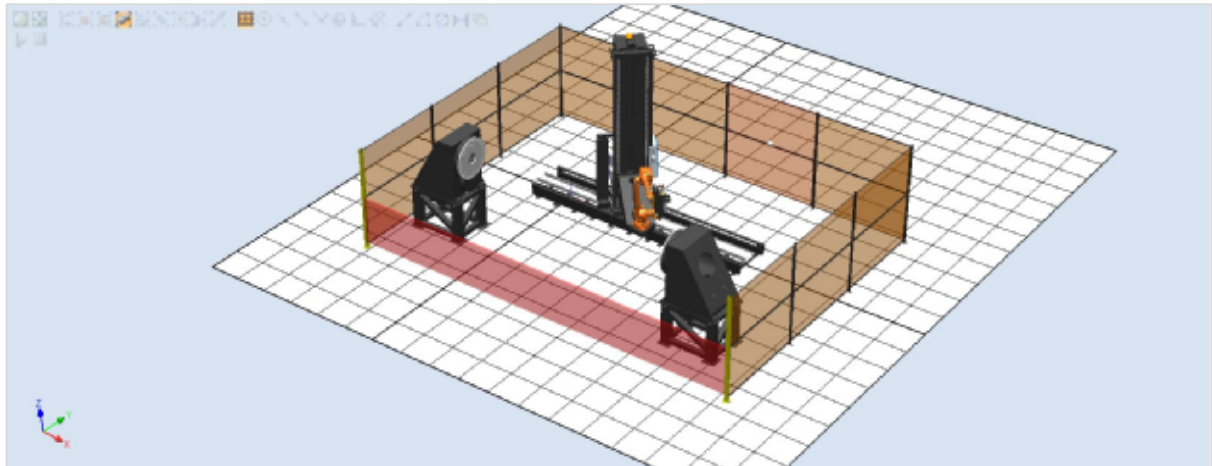


Рисунок 4.31 Add-in Final Concept Cell

#### 4.4 Створення робототехнічної комірки автоматично

Після завантаження компонента Automatic Station Builder, двічі клацнувши піктограму у вашому браузері файлів, завантажиться екран, зображений на рисунку 4.32. У лівій частині екрана з'являється діалогове вікно для введення ваги в кг, діаметра в мм і довжини деталі в мм. Натискання "Застосувати" збереже ці виділення. Після збереження вибору натисніть кнопку Виконати.

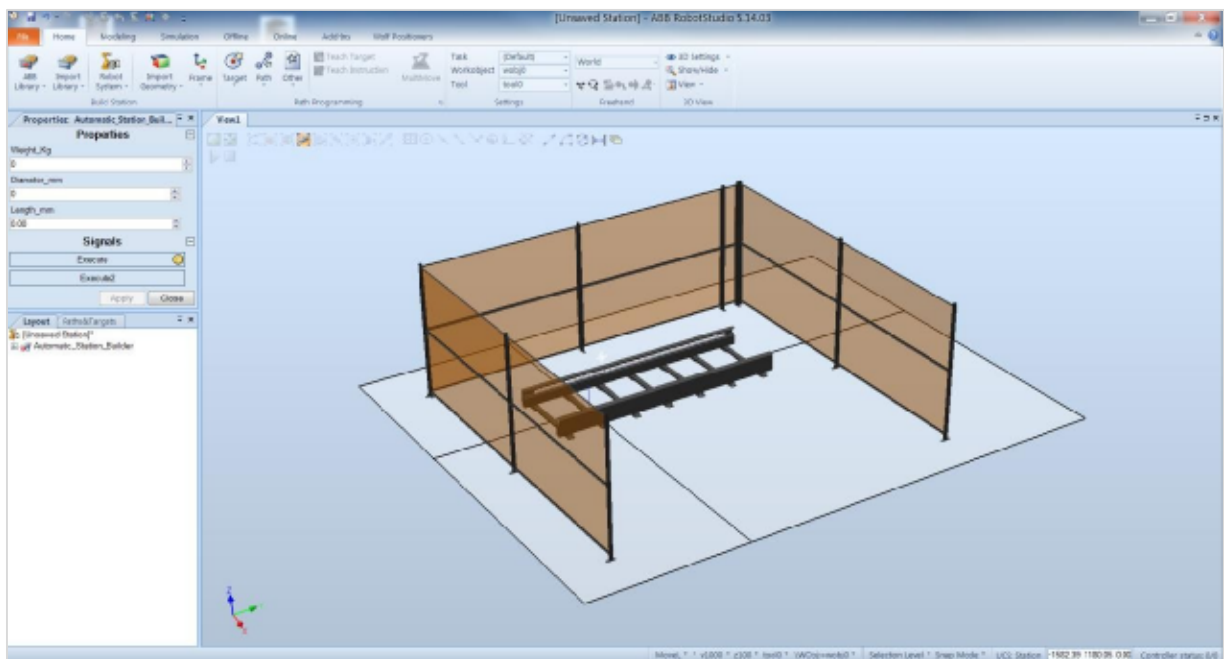


Рисунок 4.31 Automatic Station Builder Interface

Натискання кнопки Виконати активує перший етап компонента, і станція заповниться компонентами, які найкраще відповідають обмеженням вашої деталі. На рисунку 4.32 показаний приклад натискання кнопки «Виконати» вагою 50 кг, діаметром 1550 мм і довжиною 5580 мм.

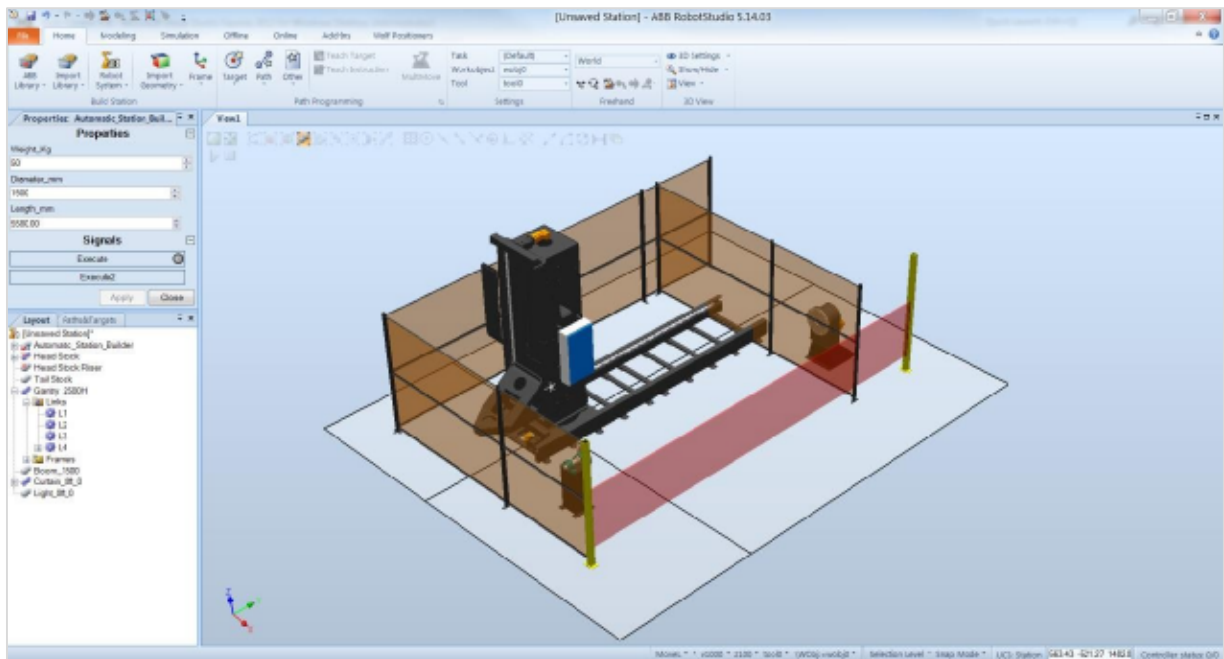


Рисунок 4.32 Automatic Station Builder Execute

Тепер натисніть кнопку Execute2. Ця кнопка належним чином розмістить решту компонентів у створеній роботизованій зварювальній камері. На рисунку 4.33 показана та сама комірка після розміщення компонентів.

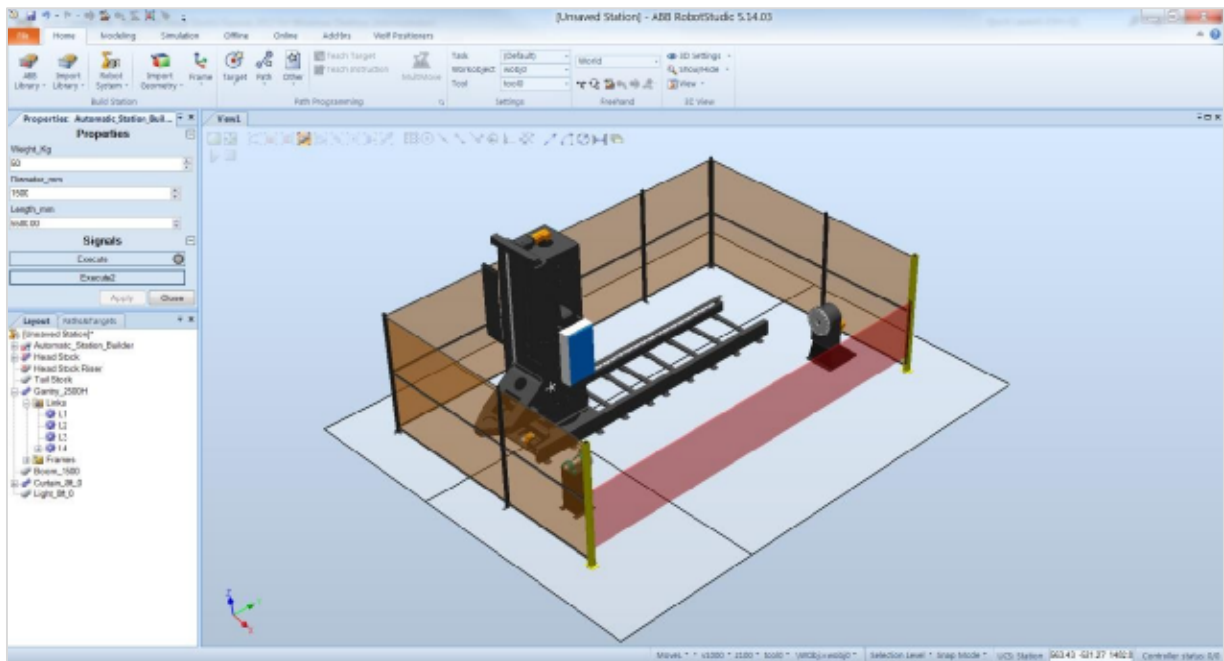


Рисунок 4.33 Automatic Station Builder Execute 2

Після побудови роботизованої зварювальної комірочки користувач тепер може розміщувати деталь, що зварюється. Станція, побудована на рисунку 4.34, тепер має робота, розміщеного на позиціонері робота, і деталь із описаними раніше атрибутами. На цьому етапі роботизовані шляхи зварювання комірок тепер можна програмувати.

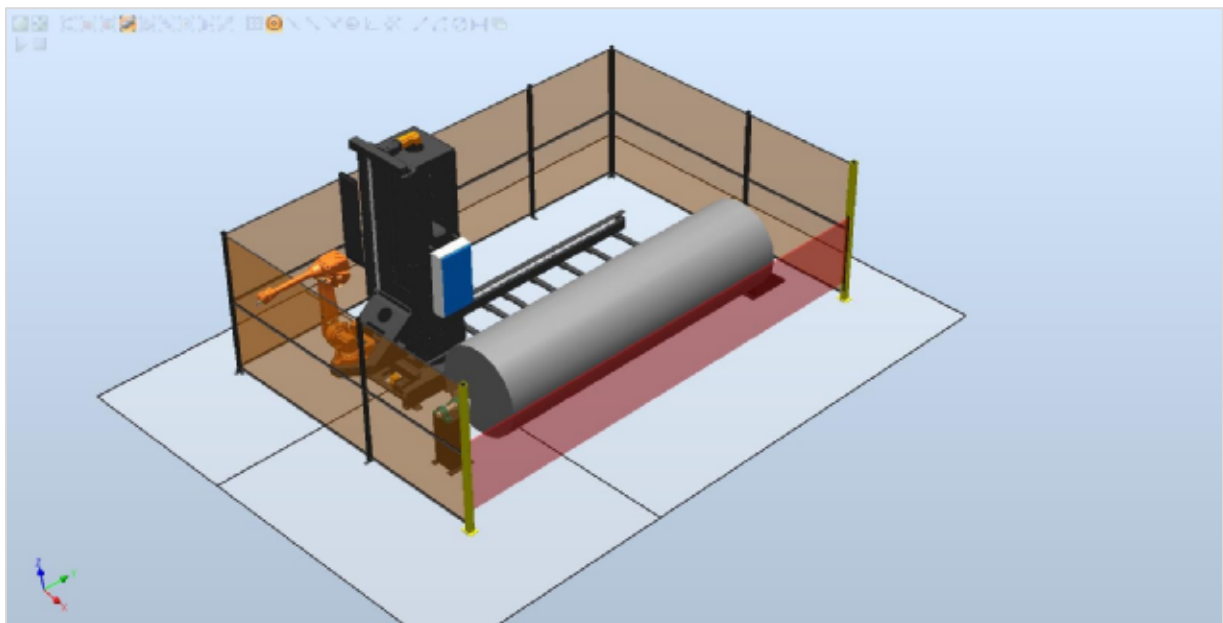


Рисунок 4.34 Automatic Station Builder Example 1

Інший приклад сформованої станції можна побачити на рисунку 4.35. Ця станція була створена для частини 30000 кг діаметром 2150 мм і довжиною 14 000 мм. Ця клітина відрізняється від попередньої кількома основними способами. Спочатку завантажено довшу доріжку, яка відповідає довжині нової частини. Цей механізм також був змінений, тому портал може переміщати загальну довжину колії. Нові запаси головки та хвостового запасу були завантажені, щоб відповідати вазі. Головка та хвостовий шток також були розміщені на вищих стояках, щоб забезпечити вільне обертання деталі. Ця додаткова висота забезпечила потребу в більшій козловій частині, щоб робот міг виконувати зварювання у верхній частині деталі. Вища козлова частина також створила потребу у вищих світлових завісах та захисних огорожах, які також були завантажені.

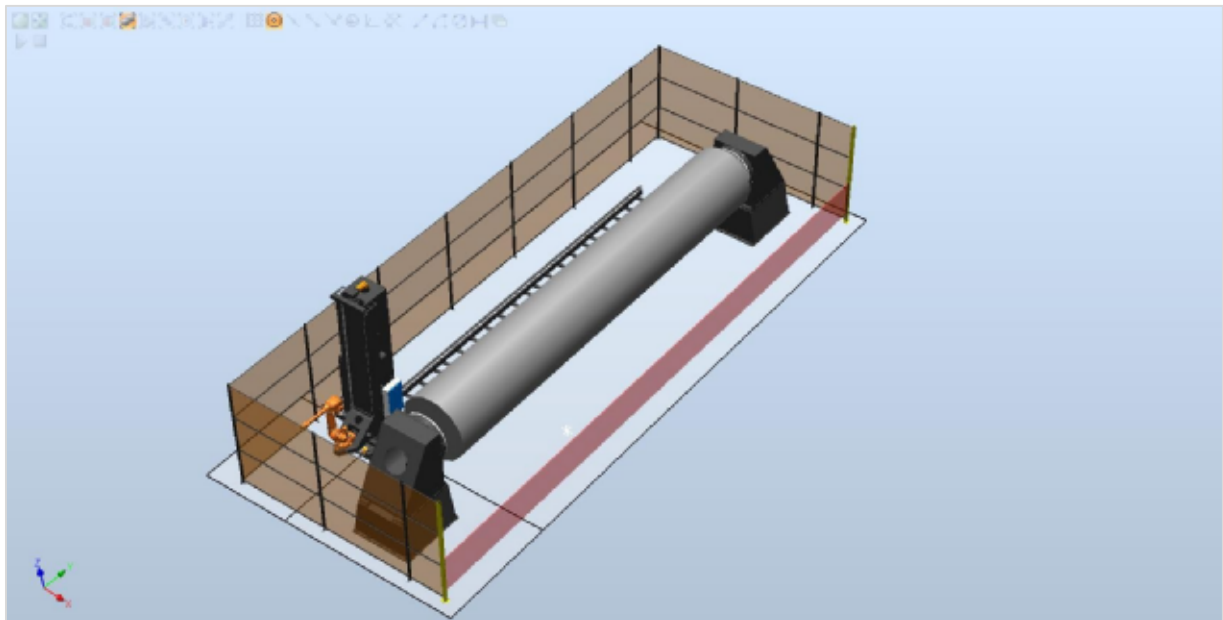


Рисунок 4.35 Automatic Station Builder Example 2

Остаточну демонстраційну станцію можна побачити на рисунку 4.36. Ця станція була створена для частини 60000 кг діаметром 3450 мм і довжиною деталі 9000 мм. Ця частина має набагато більший діаметр, ніж попередні деталі. Робототехнічна зварювальна камера вміщує частину великого діаметру, розміщуючи головку та хвостовий шток далі від порталу. Це також призвело до того, що станція завантажилася на два довші сегменти огорожі та поставила світлову завісу далі.

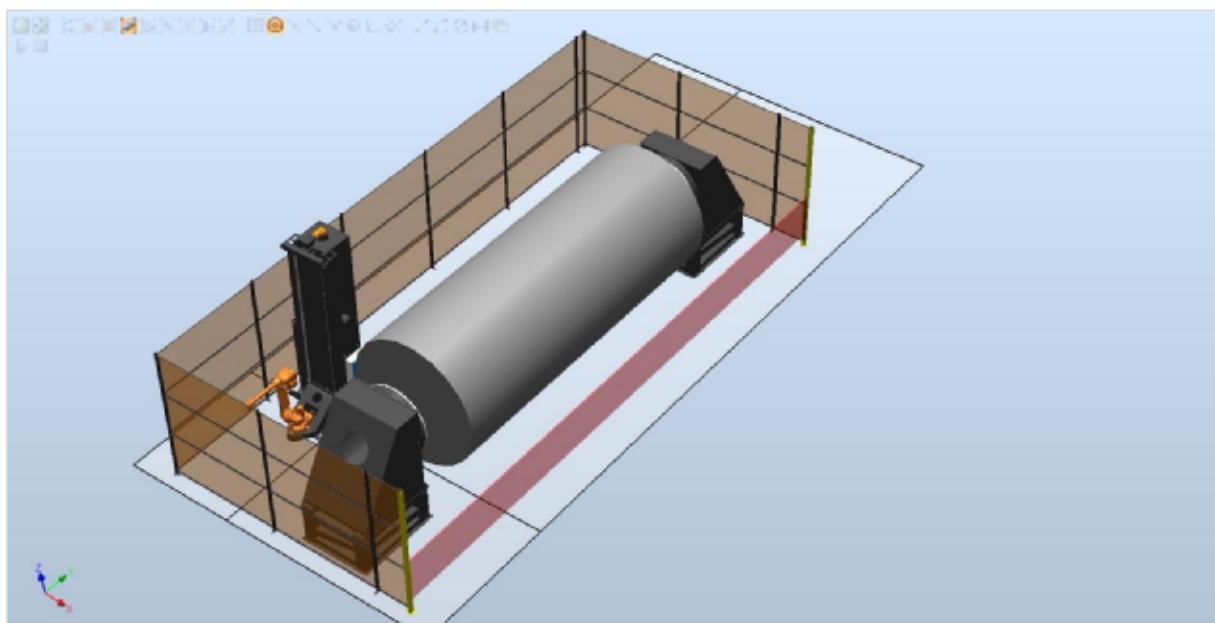


Рисунок 4.36 Automatic Station Builder Example 3



## 5 СПЕЦІАЛЬНА ЧАСТИНА

### 5.1 Створення розумної робототехнічної комірки

*Автоматичне загородження.*

1. Імпортуйте інтелектуальний компонент як бібліотеку. Останньою частиною цієї роботи було побудова робоча лінія, яка б імітувала та перевіряла різні елементи, розроблені протягом роботи.

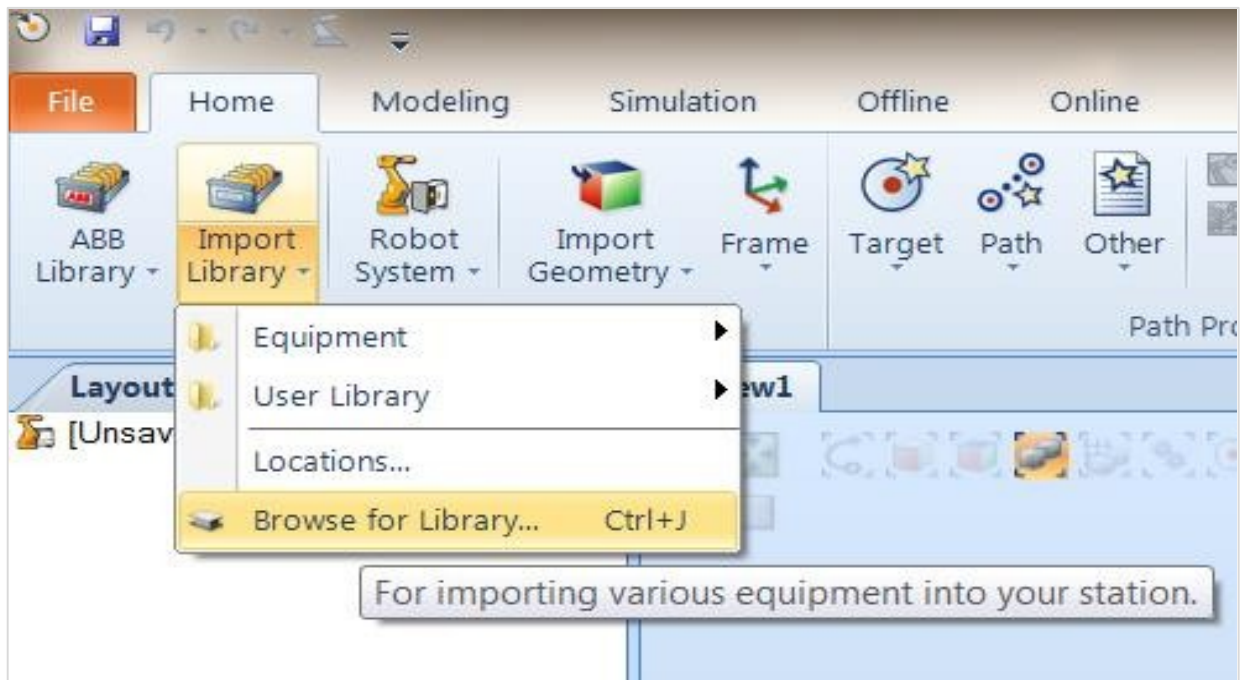


Рисунок 5.1 Керівництво користувача Імпорт загородження

2. З'явиться повідомлення з проханням перевірити інтелектуальний компонент - натисніть Так.

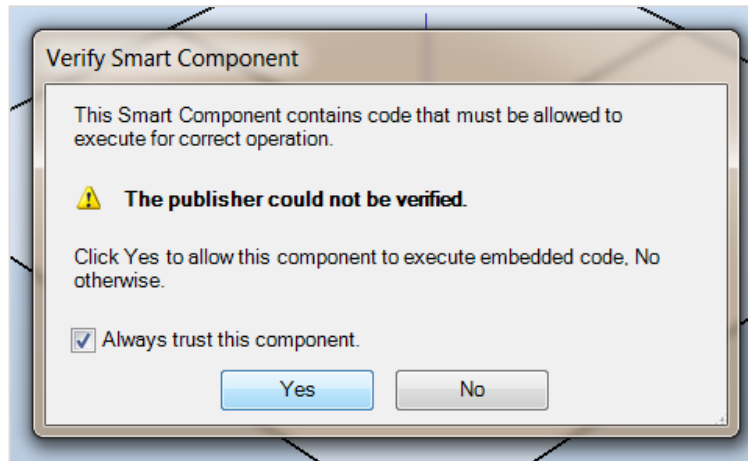


Рисунок 5.2 Посібник користувача Перевірка компонента

3. На вкладці «Домашня сторінка» під макетом двічі клацніть «Automatic\_Fencing», щоб відкрити меню вибору.

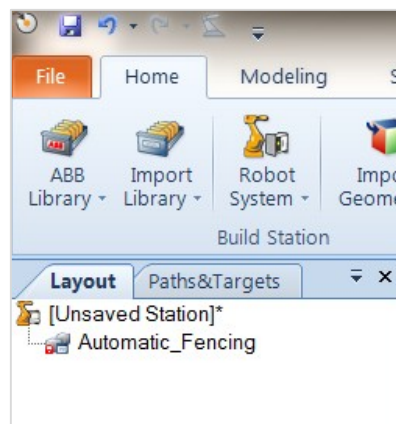


Рисунок 5.3 Керівництво користувача Вибір розумного компонента

4. У меню вибору виберіть бажану висоту та довжину огорожі.

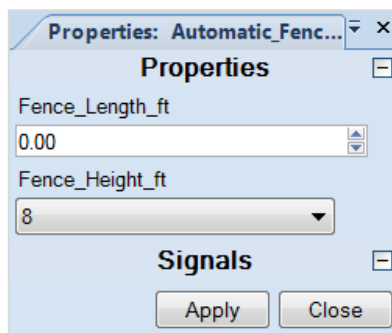


Рисунок 5.4 Посібник користувача Редагувати висоту та довжину огорожі

5. У меню вибору натисніть Застосувати, щоб активувати розумний КОМПОНЕНТ.

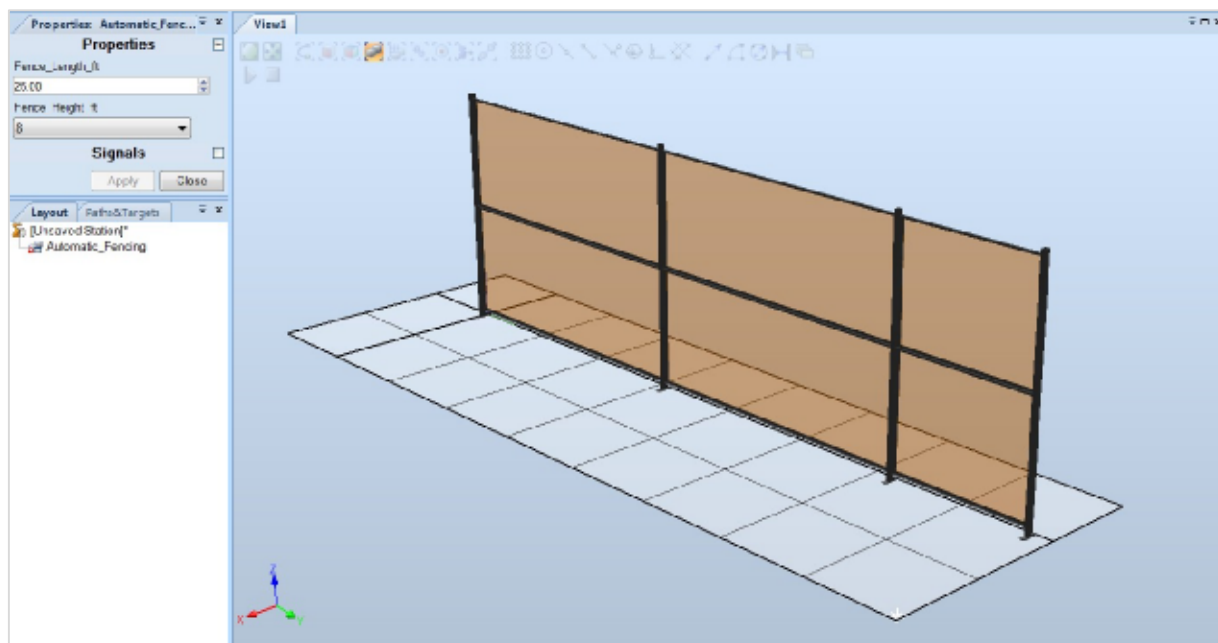


Рисунок 5.5 Керівництво користувача кінцевий елемент

*Автоматичний конструктор доріжок.*

1. Імпортуйте інтелектуальний компонент як бібліотеку.

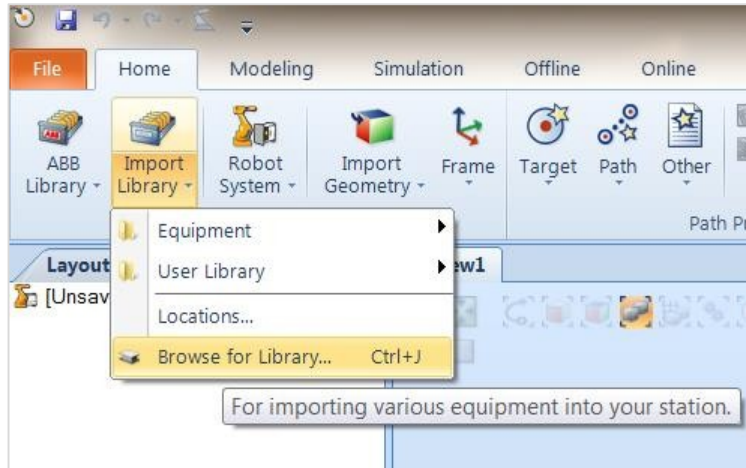


Рисунок 5.6 Посібник користувача Імпорт дорожньої доріжки

2. З'явиться повідомлення з проханням перевірити інтелектуальний компонент - натисніть Так.

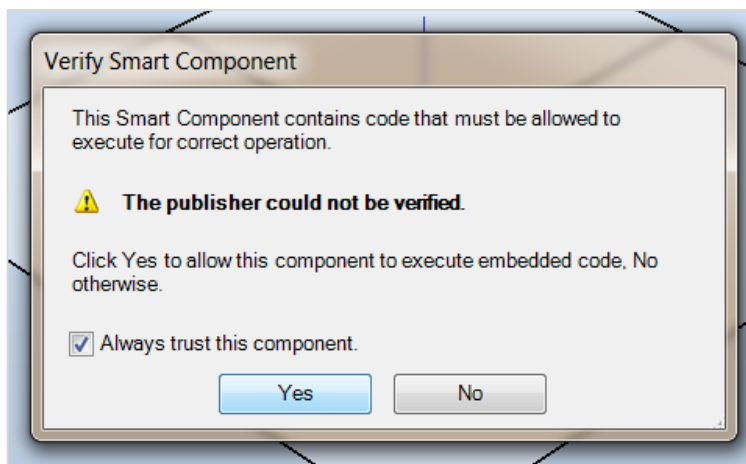


Рисунок 5.7 Посібник користувача Перевірка компонента

3. На вкладці Домашня сторінка під макетом двічі клацніть Automatic\_Track, щоб відкрити меню вибору.

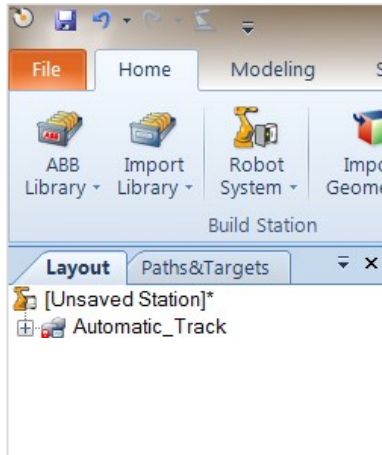


Рисунок 5.8 Керівництво користувача Вибір розумного компонента

4. У меню вибору виберіть Довжина доріжки, Тип порталу, Довжина стріли або Довжина доріжки та RTT (Стандартна карета робота).

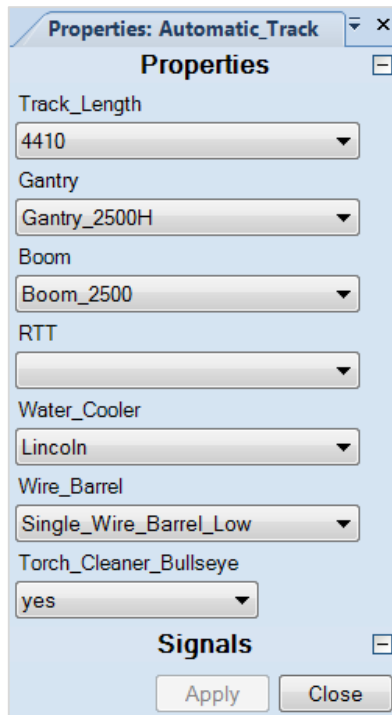


Рисунок 5.9 Керівництво користувача Редагувати параметри дорожньої доріжки

5. У меню вибору натисніть Застосувати, щоб активувати розумний КОМПОНЕНТ.

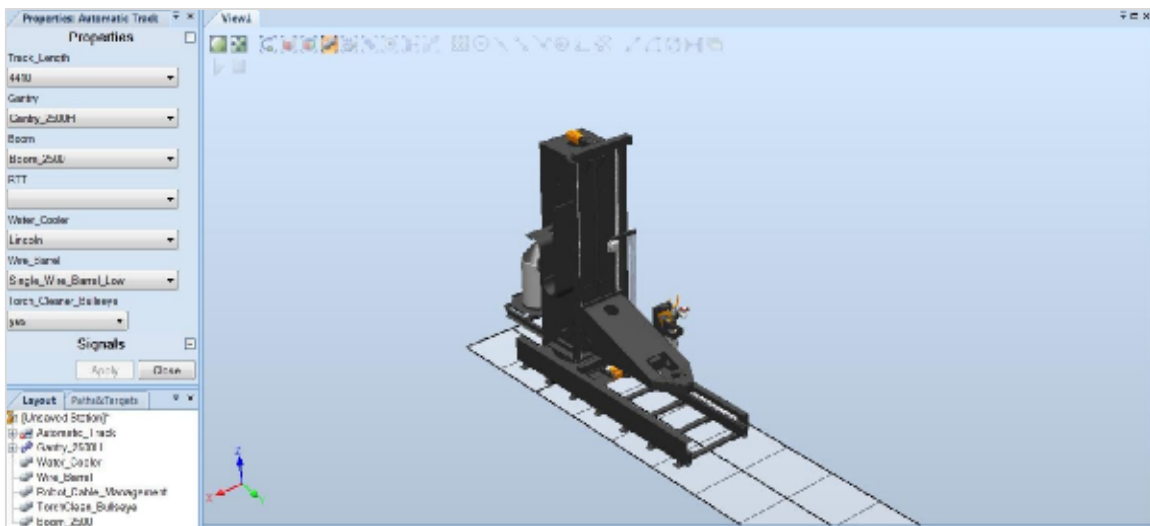


Рисунок 5.10 Кінцева доріжка користувача

6. Користувач тепер може вибрати козловий механізм та прокрутити будь-яку з трьох осей, використовуючи шарнірний пробіг механізму

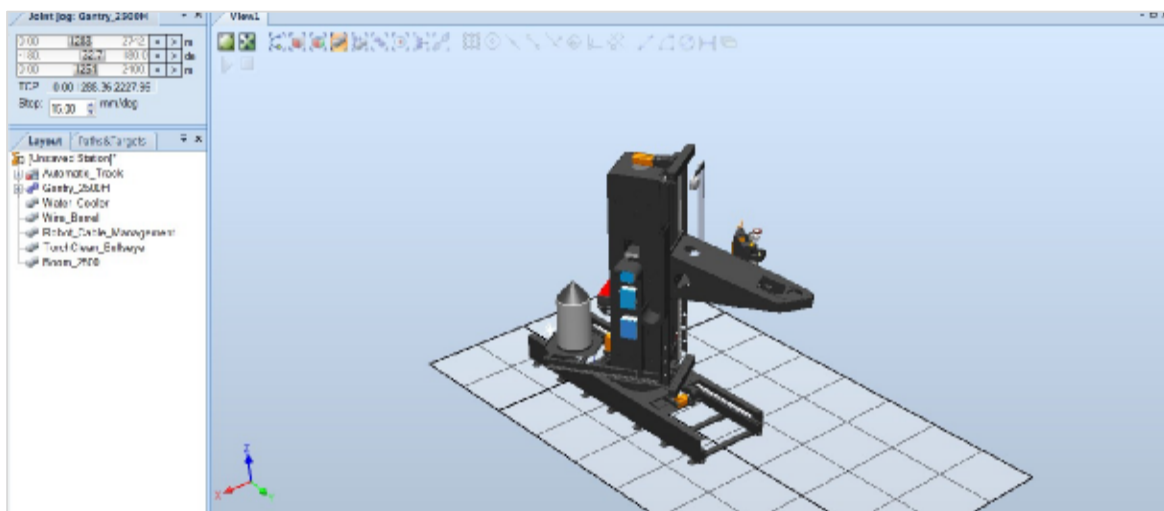


Рисунок 5.11 Керівництво користувача Механізм пробіжки

## Автоматична легка завіса

1. Імпортуйте інтелектуальний компонент як бібліотеку.

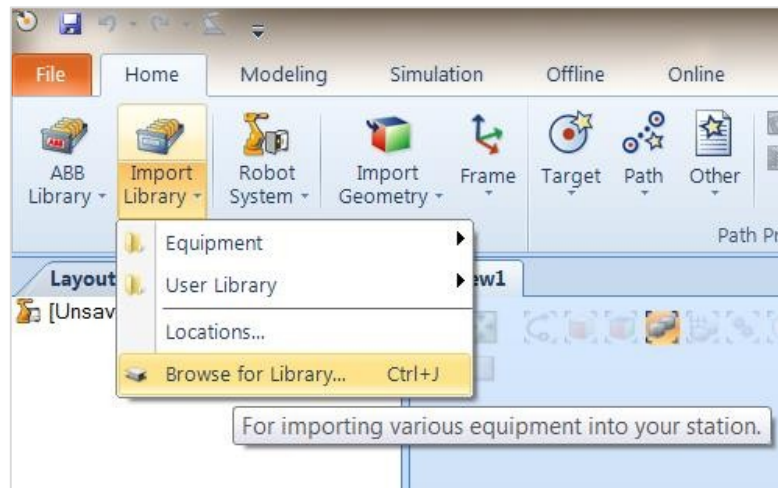


Рисунок 5.12 Керівництво користувача Імпорт світлової завіси

2. З'явиться повідомлення з проханням перевірити інтелектуальний компонент - натисніть Так

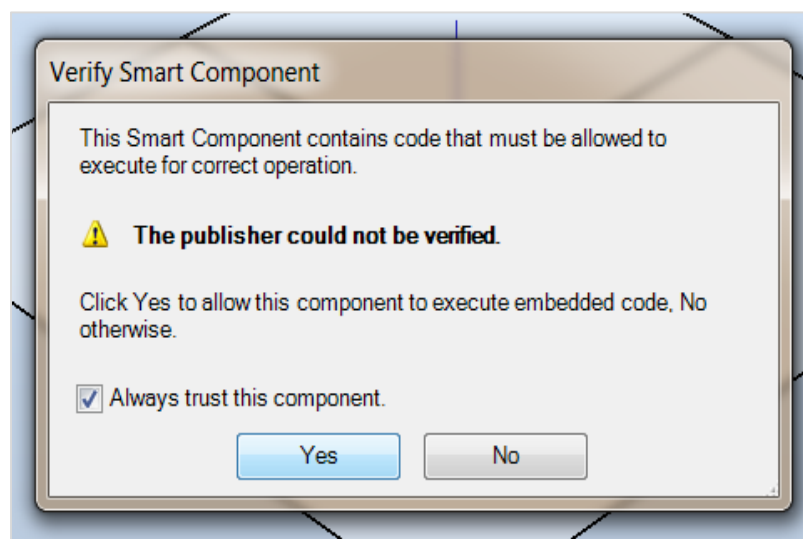


Рисунок 5.13 Посібник користувача Перевірка компонента

3. На вкладці Домашня сторінка під макетом двічі клацніть Automatic\_Track, щоб відкрити меню вибору.

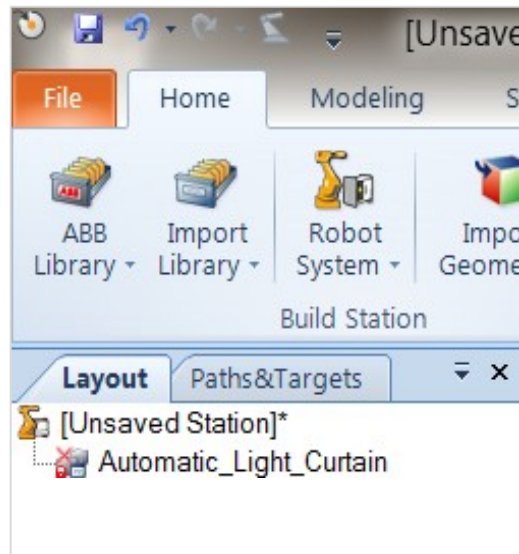


Рисунок 5.14 Керівництво користувача Вибір розумного компонента

4. У меню вибору виберіть значення висоти світлової завіси (до 9).

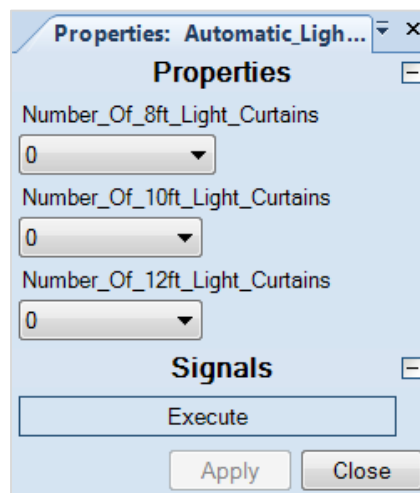




Рисунок 5.15 Посібник користувача Редагування параметрів світлової завіси

5. У меню вибору натисніть Застосувати, а потім Виконати, щоб активувати розумний компонент.
6. Тепер користувач може вибрати механізм і за допомогою механізму шарнірного повороту збільшити відстань між полюсами світла.
7. Щоб оновити довжину, положення або орієнтацію світлової завіси, натисніть кнопку Виконати в меню вибору.

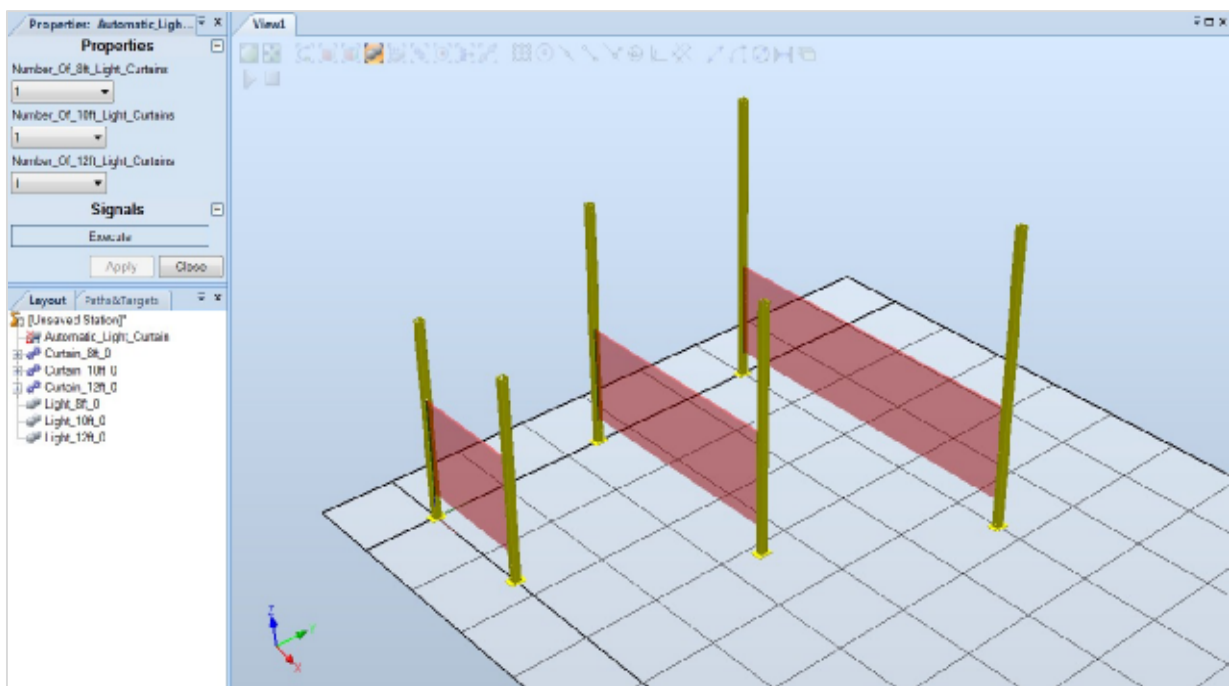


Рисунок 5.16 Посібник користувача Світлова завіса Фінал

*Встановлення надбудови позиціонерів.*

1. Почніть із запуску програми встановлення для позиціонерів Wolf.

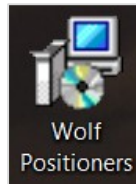


Рисунок 5.17 Встановити піктограму

2. У вікні привітання для налаштування позиціонерів Wolf - натисніть Далі.

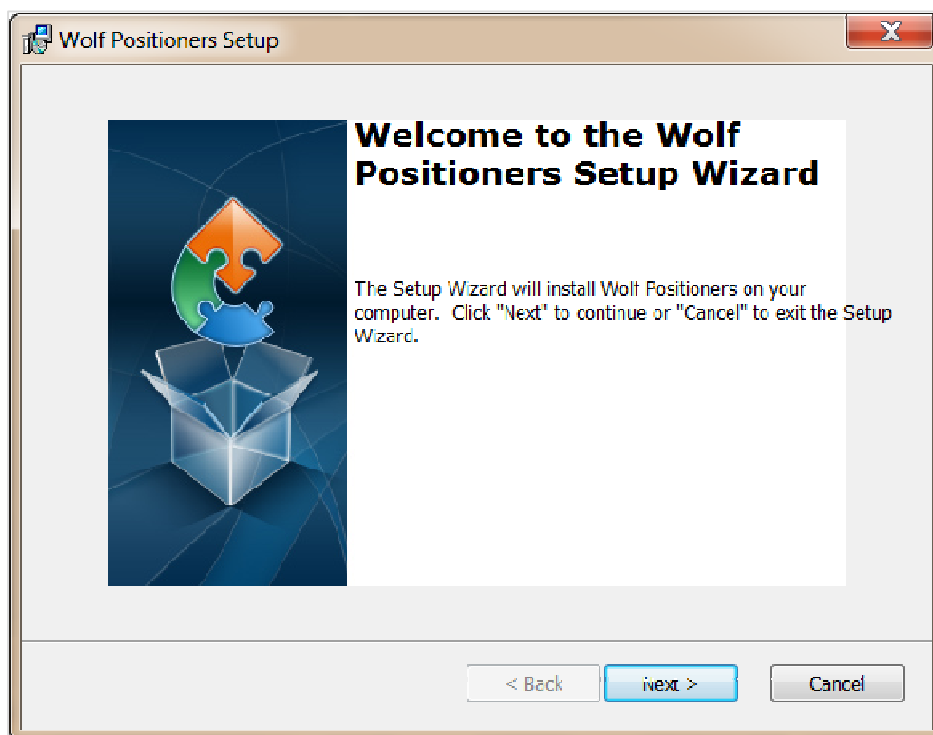


Рисунок 5.18 Встановіть екран привітання

3. Залиште папку встановлення за замовчуванням у наступному вікні та натисніть Далі.

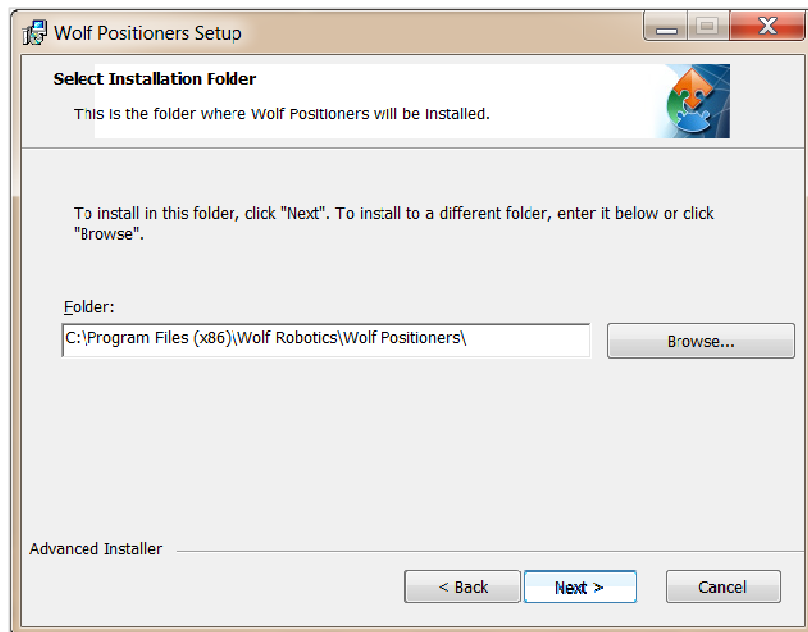


Рисунок 5.19 Встановіть шлях до файлу

4. У вікні Ready to Install натисніть кнопку Install.

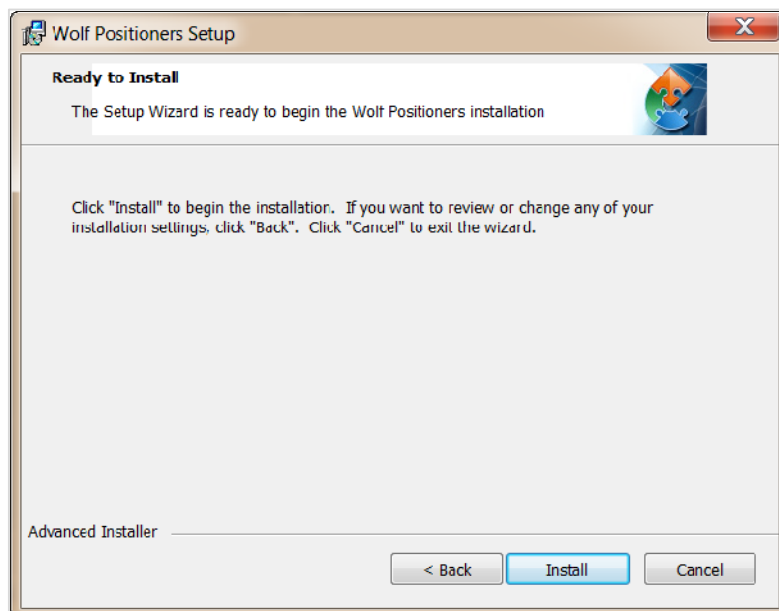


Рисунок 5.20 Почніть встановлювати

## SkyHook

1. Після встановлення надбудови Positioners буде доступна нова вкладка Wolf Positioners. Перейдіть на цю вкладку та натисніть кнопку Створити SkyHook.

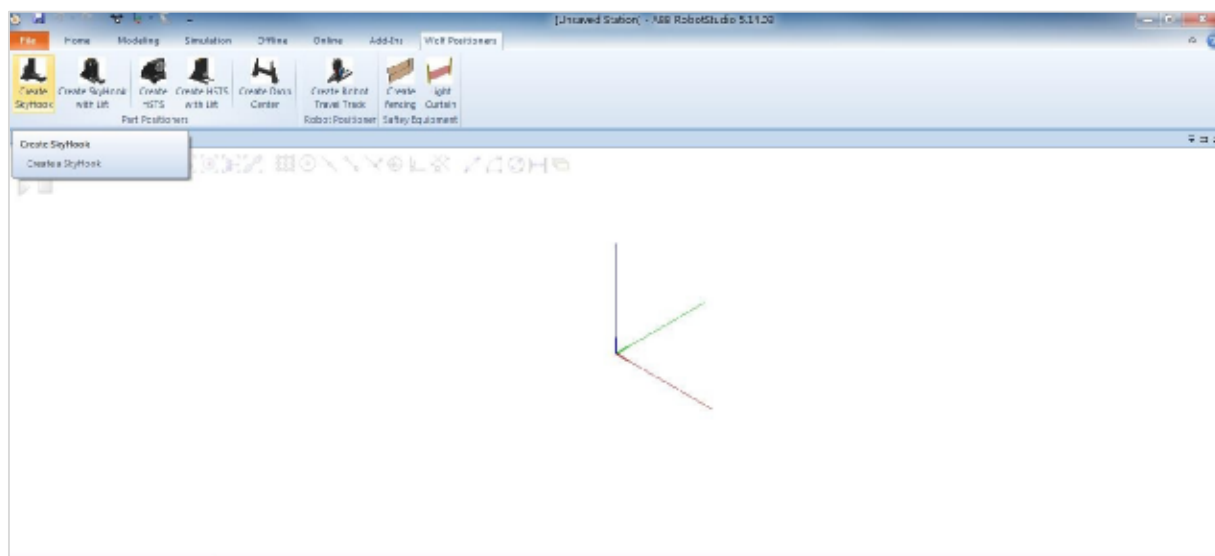


Рисунок 5.21 Посібник користувача Створення SkyHook

2. Після цього відкриється нове вікно для вибору параметрів SkyHook. Кидок і падіння автоматично оновлюватимуться зі стандартними бібліотеками, як тільки буде обрано потужність. Після того, як користувач вибрав ємність, киньте та опустіть, натисніть кнопку Завантажити.

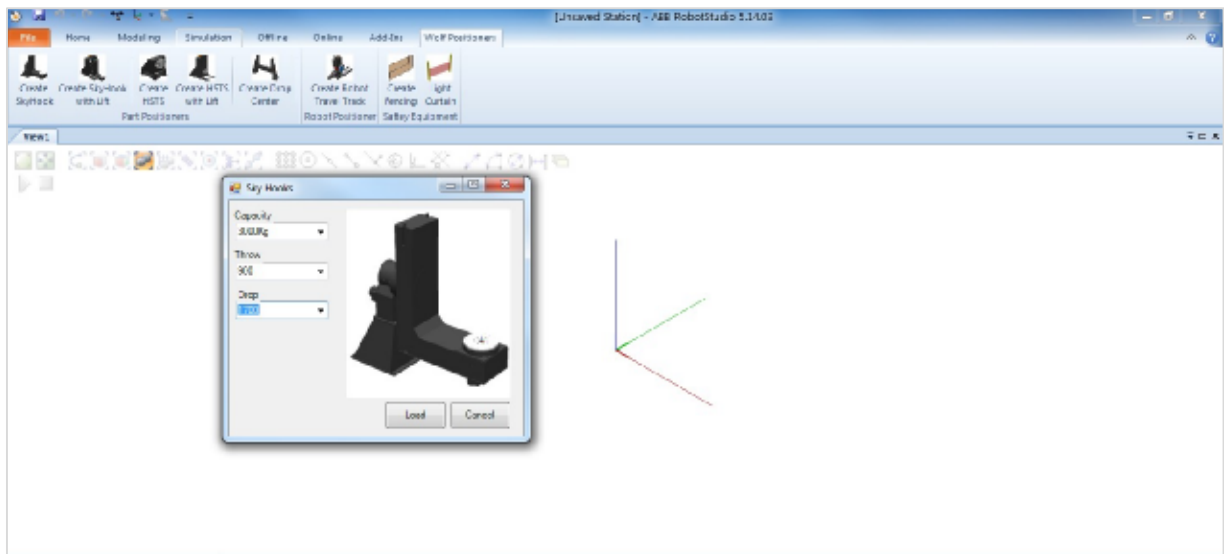


Рисунок 5.22 Вікно вибору SkyHook

3. Це завантажить бажаний SkyHook на станцію. Тепер SkyHook можна підключити до контролера або перемістити вручну за допомогою командної механізованої команди jog.

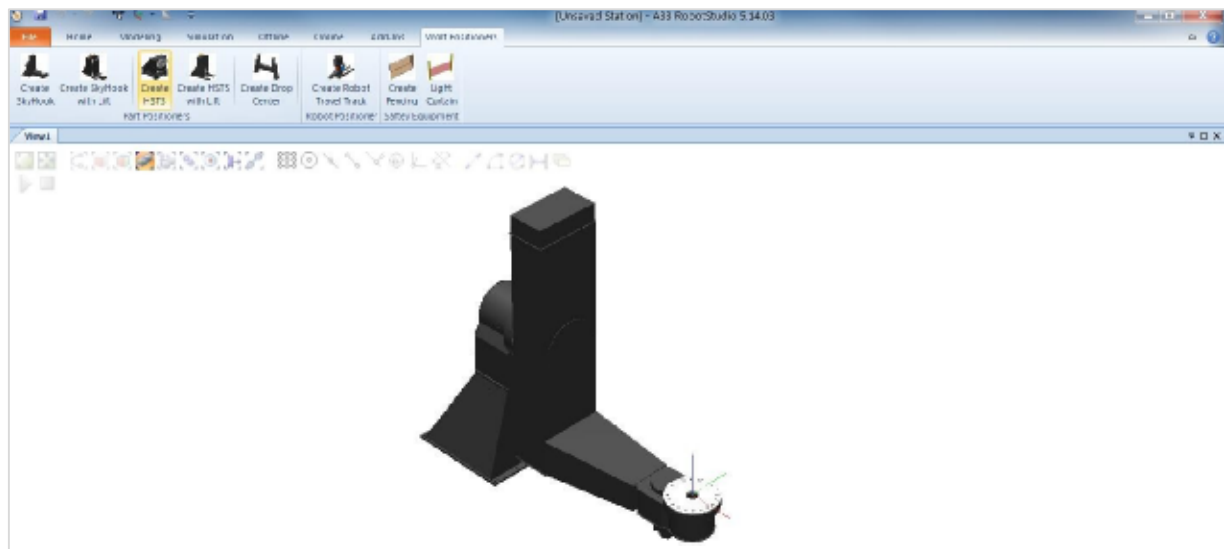


Рисунок 5.23 Посібник користувача SkyHook Final

## *SkyHook з підйомником*

1. Після встановлення надбудови Positioners буде доступна нова вкладка Wolf Positioners. Перейдіть на цю вкладку та натисніть кнопку Створити SkyHook with Lift.

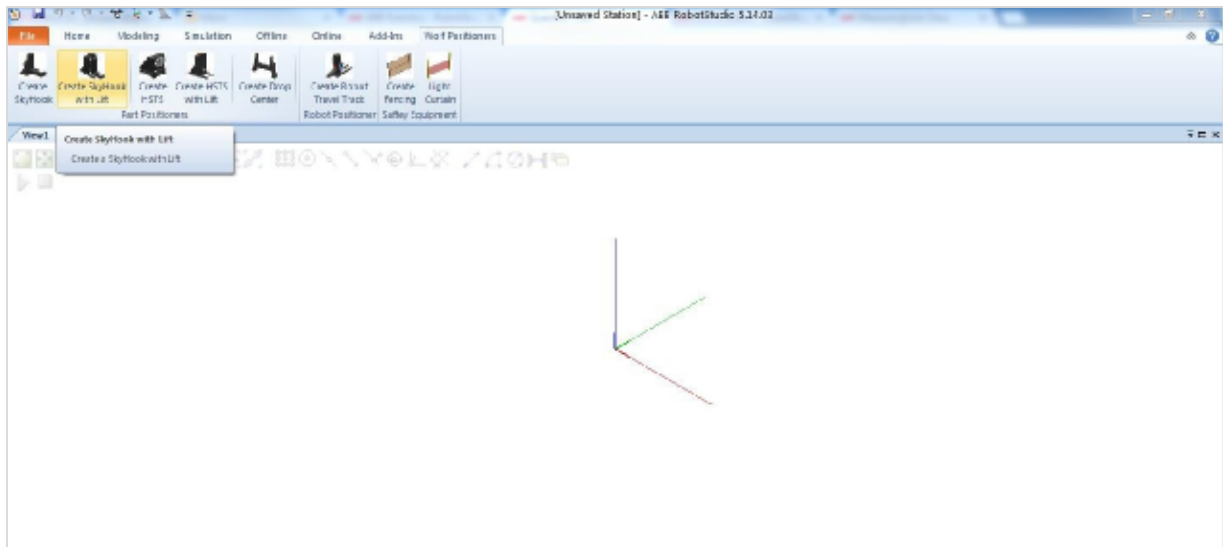


Рисунок 5.24 Посібник користувача Створення SkyHook з ліфтом

2. Це відкриє нове вікно для вибору SkyHook with Lift options. Кидок і падіння автоматично оновлюватимуться зі стандартними бібліотеками, як тільки буде обрано потужність. Після того, як користувач вибрав ємність, киньте та опустіть, натисніть кнопку Завантажити.

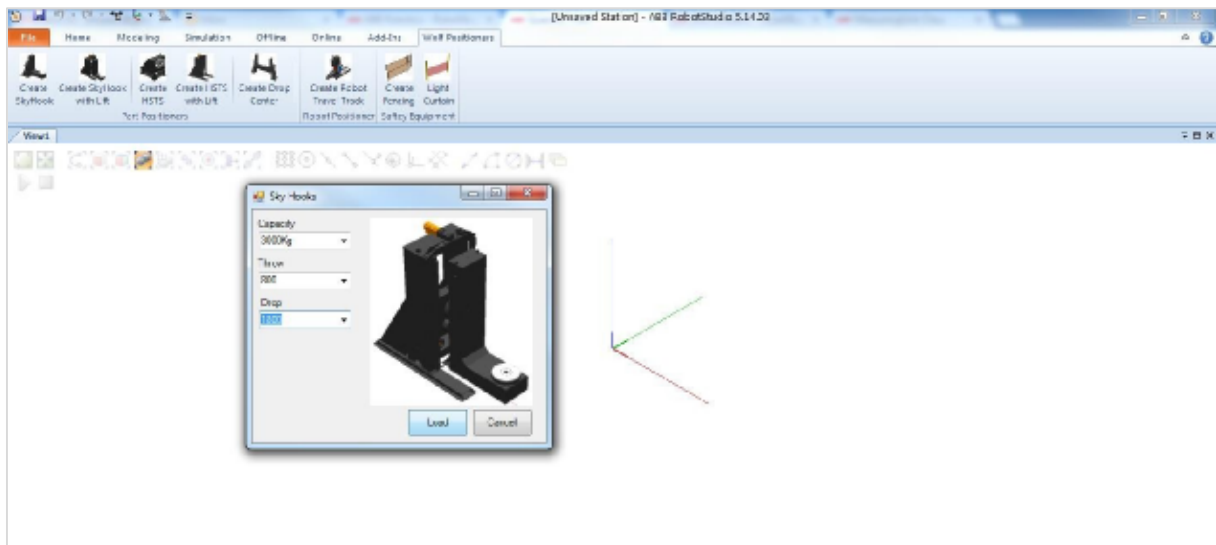


Рисунок 5.25 Посібник користувача SkyHook з вікном вибору ліфта

3. Це завантажить бажаний SkyHook на станцію. Тепер цей SkyHook можна підключити до контролера або перемістити вручну за допомогою командної механізованої команди jog.

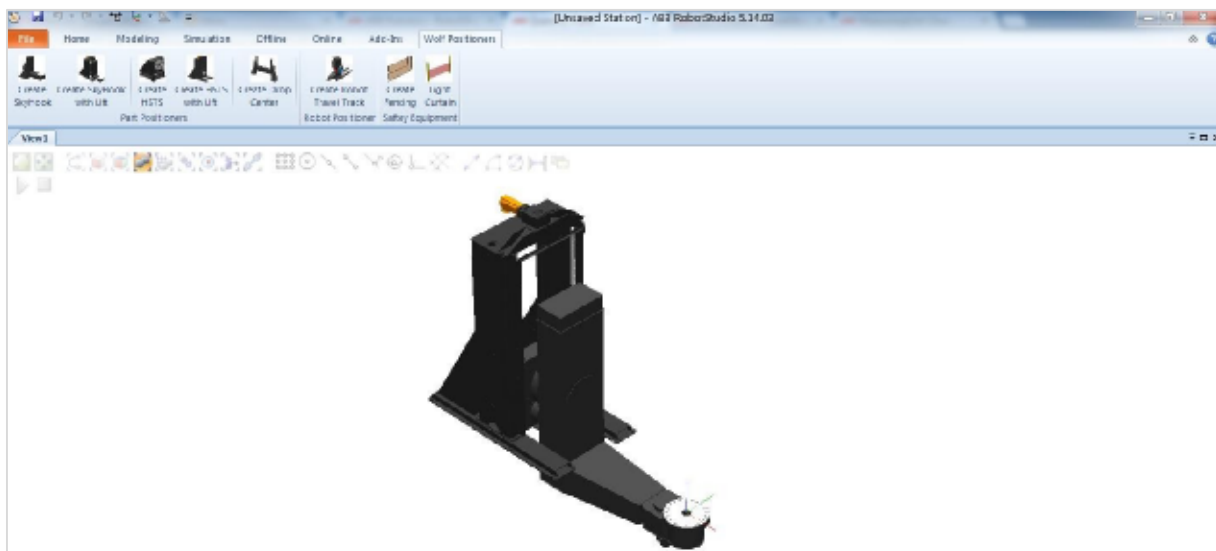


Рисунок 5.26 Посібник користувача SkyHook with Lift Final

*Головний запас хвоста.*

1. Після завершення встановлення надбудови Позиціонери буде доступна нова вкладка «Позиціонери Вовка». Перейдіть на цю вкладку та натисніть кнопку Створити HSTS.

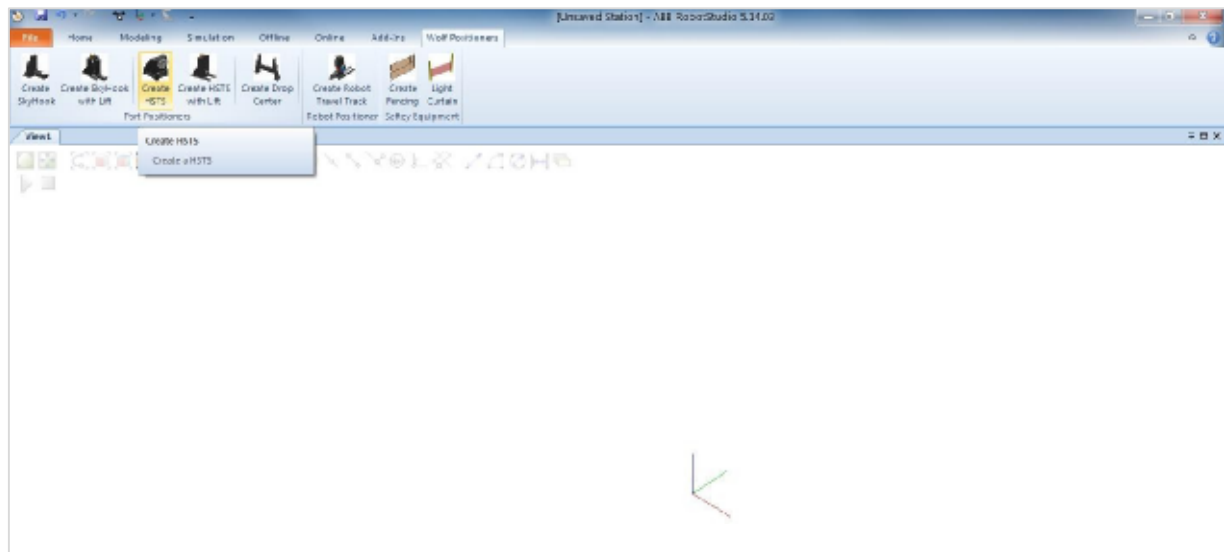


Рисунок 5.27 Посібник користувача Створення HSTS

2. Відкриється нове вікно, де можна вибрати параметри головного запасу та хвостового запасу.

Після того, як користувач вибрав потужність та висоту стояка, натисніть кнопку Завантажити.



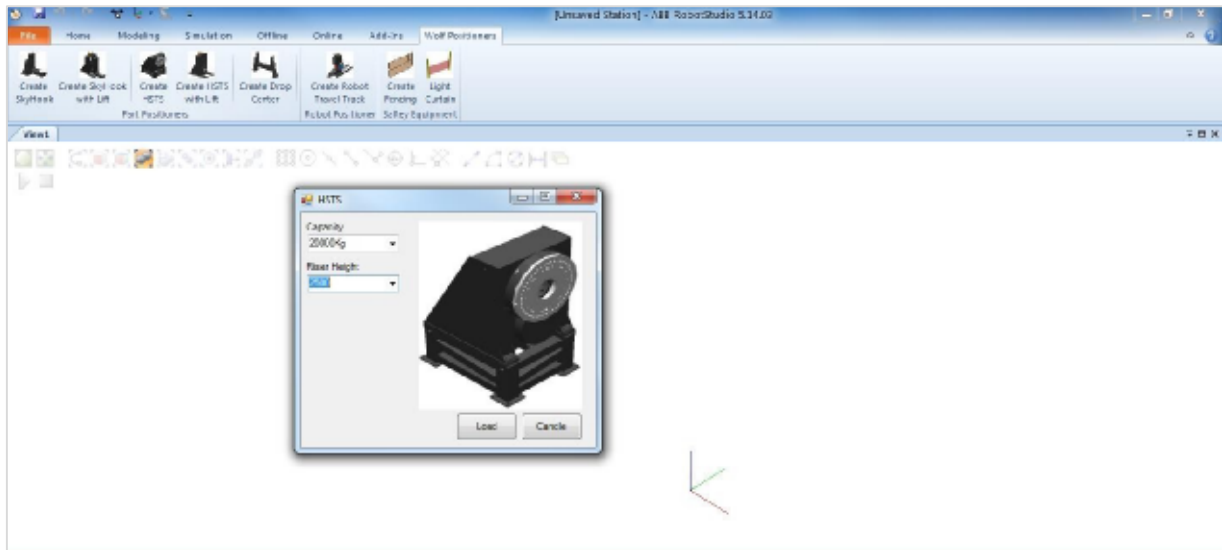


Рисунок 5.28 Вікно вибору HSTS

3. Це завантажить бажаний запас голови та хвостовий запас на станцію. Цей запас головки та хвостового запасу тепер можна вручну перемістити на відповідну відстань.

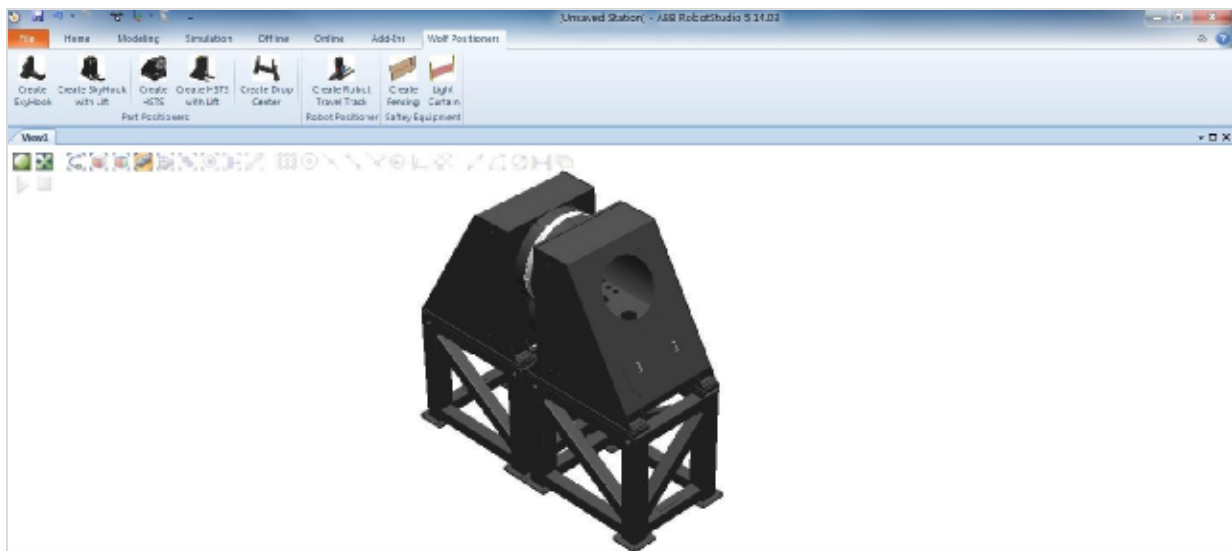


Рисунок 5.29 Посібник користувача Завантажений HSTS

#### 4. Head Stock тепер можна підключити до віртуального контролера.

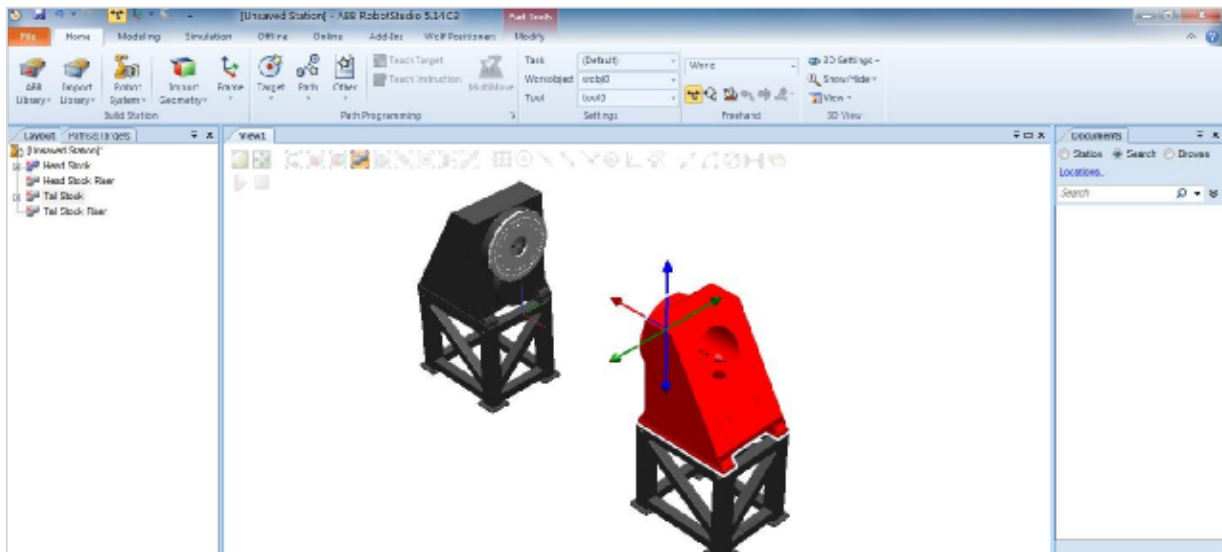


Рисунок 5.30 Позиція керування користувача HSTS

*Головний запас хвостового запасу з підйомником.*

1. Після завершення встановлення надбудови Позиціонери буде доступна нова вкладка «Позиціонери Вовка». Перейдіть на цю вкладку та натисніть кнопку Створити HSTS за допомогою ліфта.

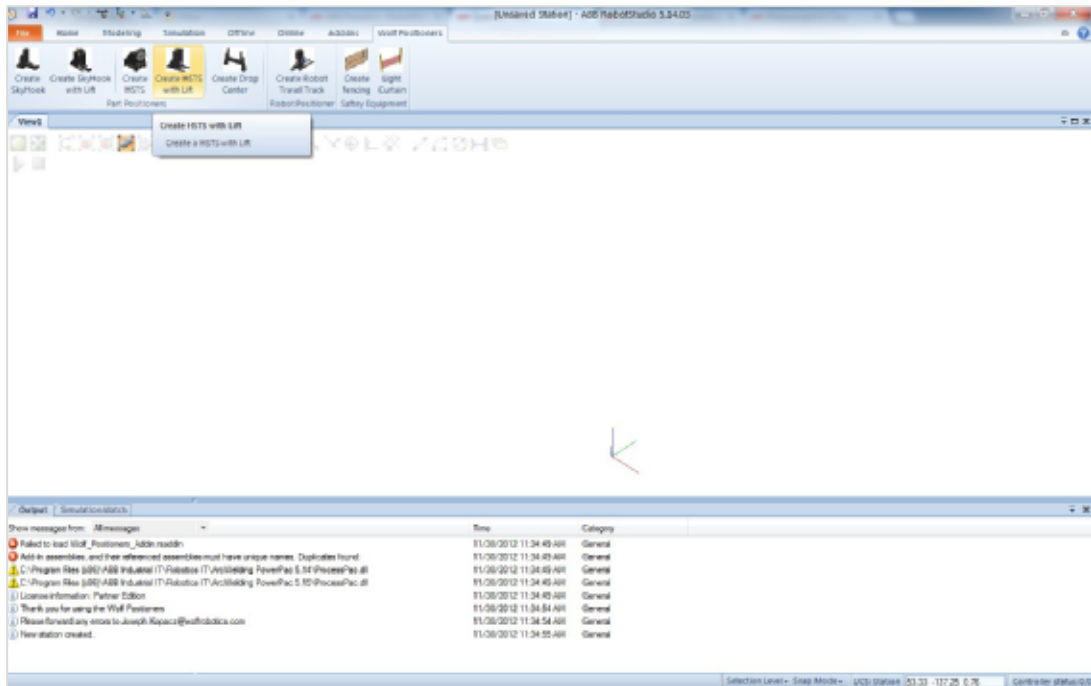


Рисунок 5.31 Посібник користувача Створення HSTS за допомогою ліфта

2. Відкриється нове вікно, де можна вибрати параметри головного запasu та хвостового запasu.

Після того, як користувач вибрав ємність, натисніть кнопку Завантажити.

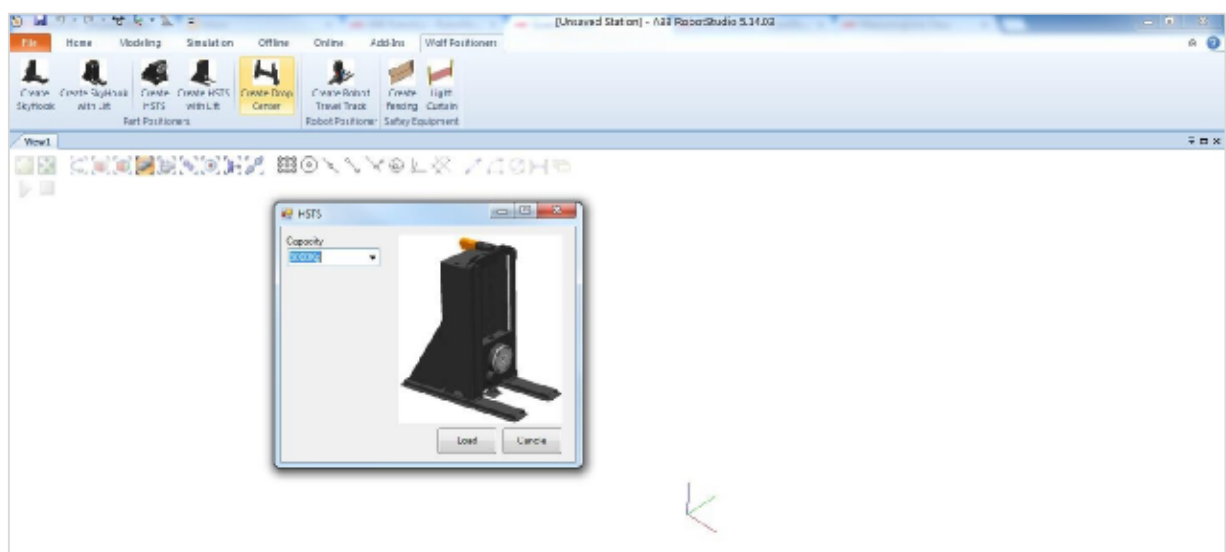


Рисунок 5.32 Посібник користувача HSTS з вікном вибору ліфта

3. Це завантажить бажаний запас голови та хвостовий запас на станцію. Цей запас головки та хвостового запасу тепер можна вручну перемістити на відповідну відстань.

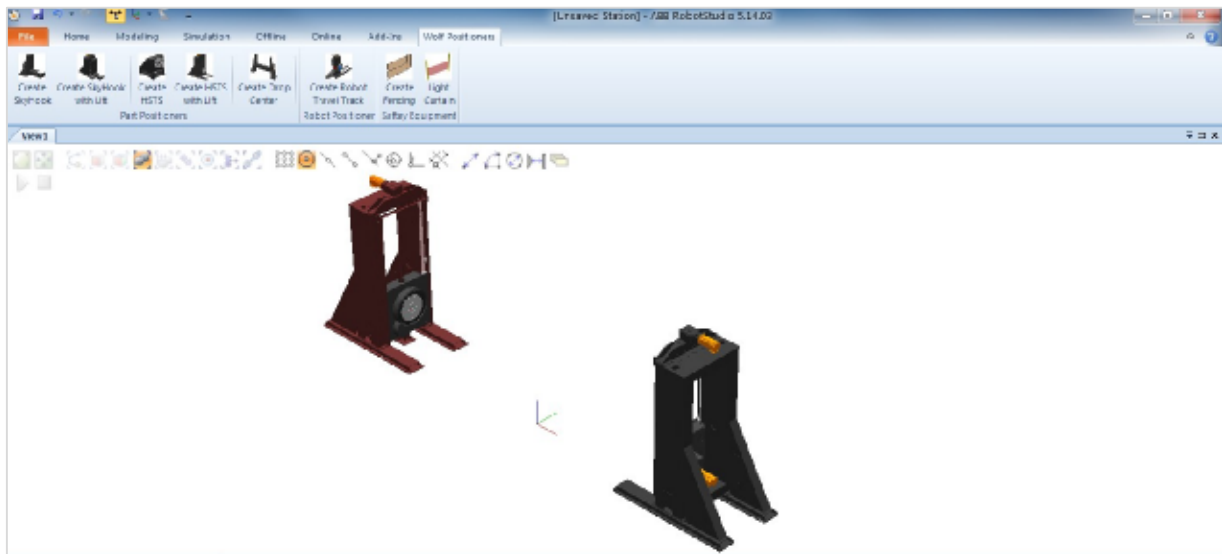


Рисунок 5.33 Посібник користувача HSTS з ліфтом Фінал

4. Head Stock тепер можна підключити до віртуального контролера.

#### *Drop Center.*

1. Після встановлення надбудови Positioners буде доступна нова вкладка Wolf Positioners. Перейдіть на цю вкладку та натисніть кнопку Створити центр падіння.

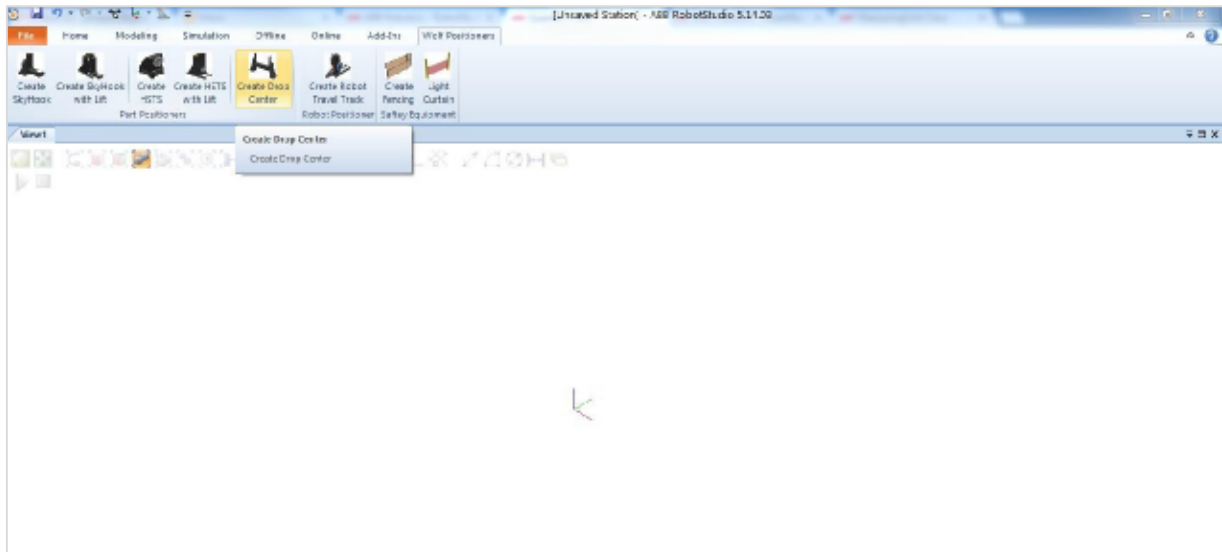


Рисунок 5.34 Посібник користувача Створення Drop Center

2. Після цього відкриється нове вікно, де можна вибрати параметри Drop Center. Після того, як користувач вибрав місткість, метання, падіння та висоту стояка, натисніть кнопку Завантажити.

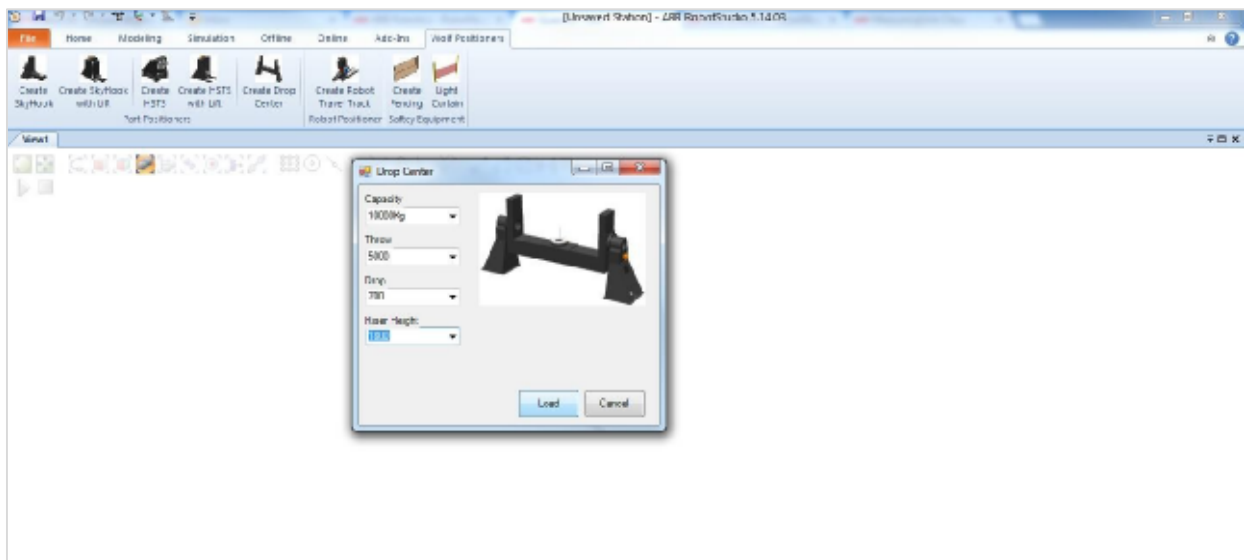


Рисунок 5.35 Вікно вибору центру Drop Center

3. Це завантажить вибраний Drop Center на станцію.

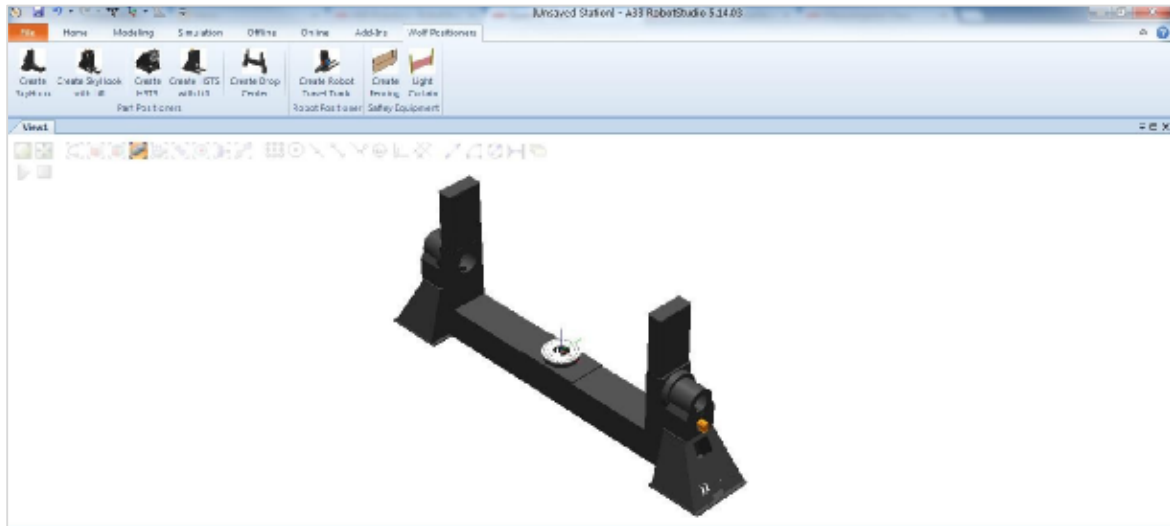


Рисунок 5.36 Посібник користувача Drop Center Final

*Автоматичний конструктор станцій.*

1. Імпортуйте інтелектуальний компонент як бібліотеку.

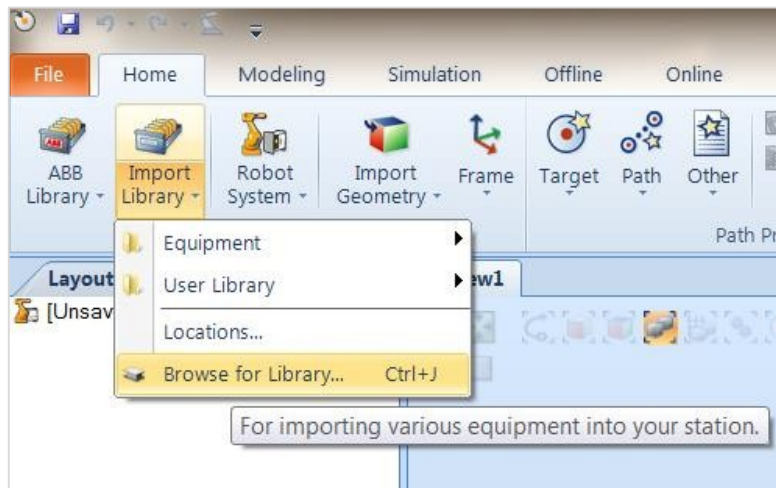


Рисунок 5.37 Керівництво користувача Імпорт фехтування

2. З'явиться повідомлення з проханням перевірити інтелектуальний компонент - натисніть Так.

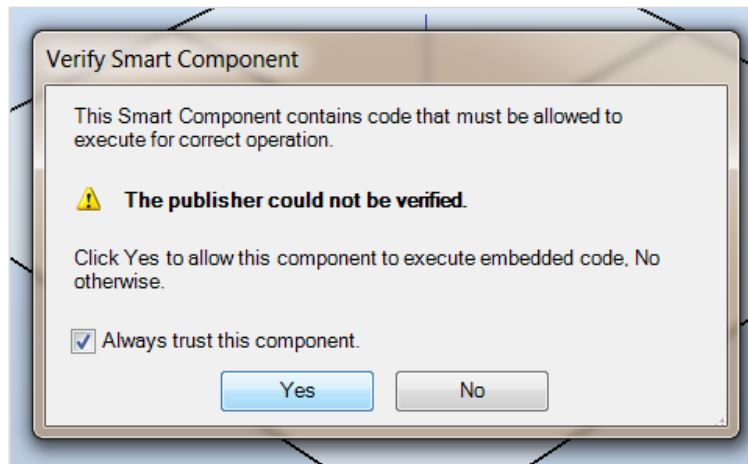


Рисунок 5.38 Посібник користувача Перевірка компонента

3. На вкладці Домашня сторінка під макетом двічі клацніть Automatic\_Station\_Builder, щоб відкрити меню вибору.

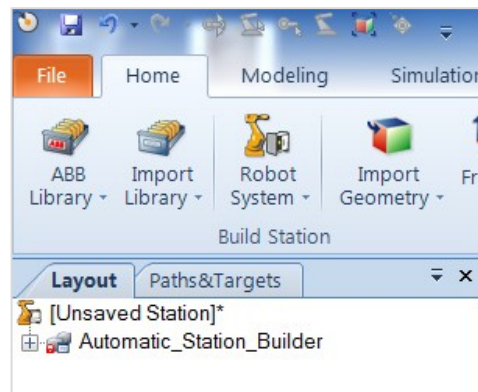


Рисунок 5.38 Керівництво користувача Вибір розумного компонента

4. У меню вибору виберіть деталі Вага в кг, Діаметр в мм і Довжина в мм.

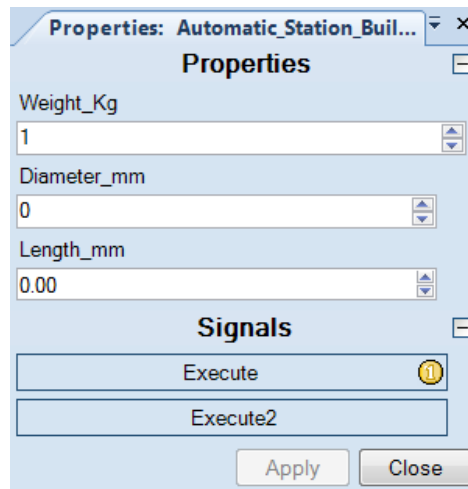


Рисунок 5.39 Посібник користувача Редагування ваги, діаметра та довжини

5. У меню вибору натисніть Застосувати, а потім Виконати, щоб активувати розумний компонент. Це призведе до компонентів, необхідних для побудови станції.

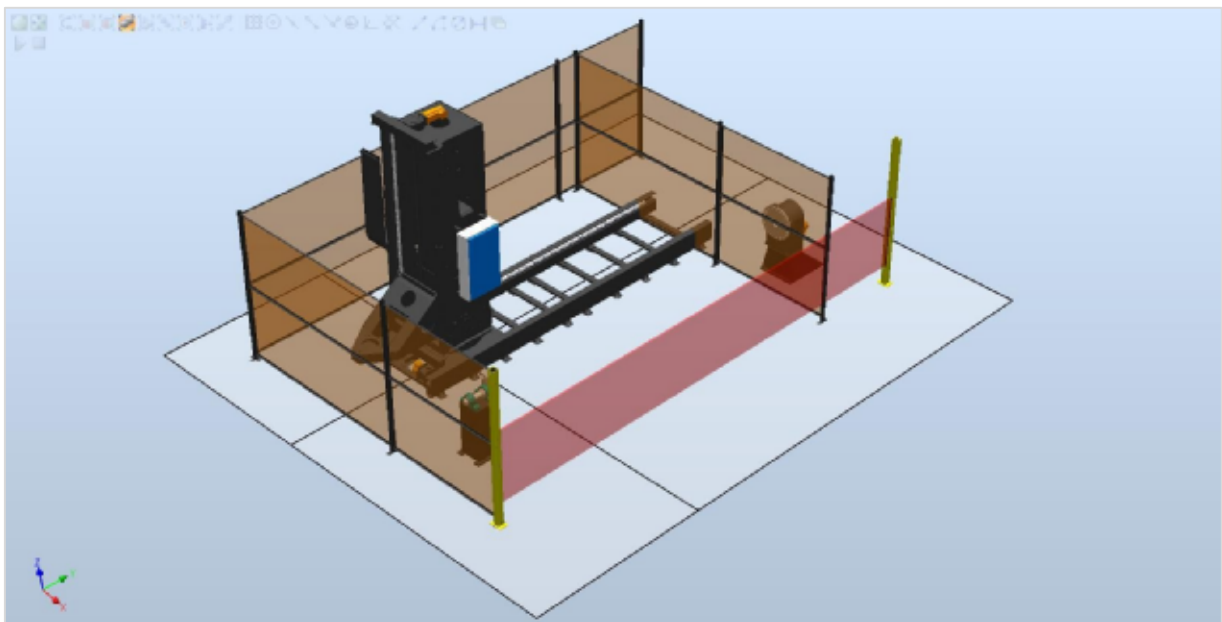


Рисунок 5.40 Керівництво користувача Виконання1



6. У меню вибору натисніть Execute2, щоб розмістити всі завантажені компоненти.

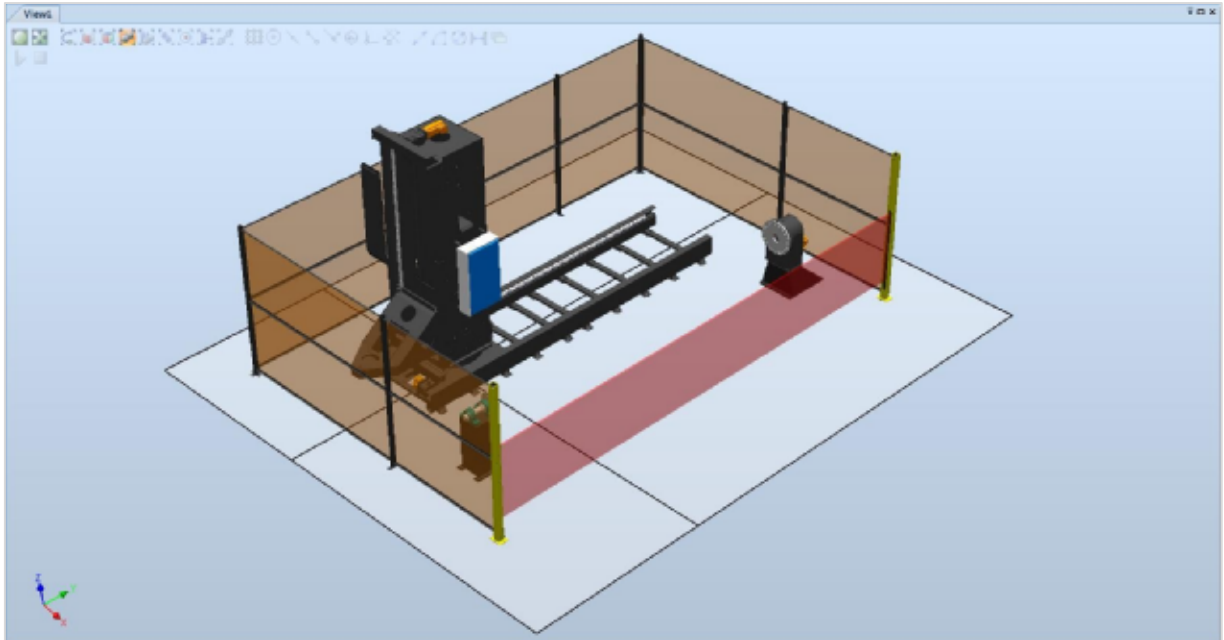


Рисунок 5.40 Керівництво користувача Виконання2

## **6 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ**

### **6.1 Значення охорони праці в забезпеченні здорових умов праці**

Під охороною праці розуміється система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людини в процесі праці.

Охорона праці - прикладна наука правового, санітарно-гігієнічного і технічного напрямку, що виявляє і вивчає виробничі небезпеки, розробляє методи їх попередження і послаблення з метою усунення виробничого травматизму, професійних захворювань робочих, попередження аварій, пожеж і вибухів, забезпечує оздоровлення умов праці і захист населених пунктів від шкідливих і небезпечних впливів виробництва.

За останні роки всі сфери промисловості автоматизувались, що привело до зменшення нещасних випадків на виробництві, але в той самий час виникло ряд інших шкідливих факторів пов'язаних з використанням ЕОМ і інших обчислювальних засобів.

Найбільш повні дані про вплив комп'ютера на людину можна отримати, зібравши і проаналізувавши дані за період, як мінімум, до 60 років, оскільки така середня тривалість життя людини. В Харківському інституті професійних захворювань і гігієни праці є лабораторія, яка уже на протязі 12 років поглиблено займається цими питаннями.

## **6.2 Охорона праці як система заходів щодо гармонізації використання комп'ютерних технологій**

Науково-технічний прогрес призводить до корінної зміни характеру та засобів трудової діяльності. Він створює для людини велику кількість благ: зменшує тяжкість праці (фізичне навантаження), робить її інтелектуальною, цікавою, різноманітною, розвиває творчі здібності людини, сприяє удосконаленню її професійних навичок тощо.

Проте сучасна техніка є потенційним джерелом високої небезпеки для життя та здоров'я працівника, оточуючих його людей, а також навколишнього середовища. Особливо це стосується сучасних технологій, що активно використовують комп'ютери. Мільйони користувачів відеодисплейних терміналів (ВДТ) персональних комп'ютерів працюють практично у всіх галузях народного господарства.

Праця цих користувачів дуже різноманітна. Це працівники обчислювальних центрів, що обслуговують автоматизовані системи управління підприємствами, касири, працівники довідкових служб, оператори енергетичних, хімічних, газових, нафтових та інших підприємств, працівники поліграфічної промисловості, службовці редакцій журналів, газет, книжкових видань, працівники сфери освіти, перекладачі, особи, які працюють у агентствах новин, та багато інших категорій працівників. Всіх їх об'єднує використання у своїй роботі відеодисплейних терміналів персональних електронно-обчислювальних машин та розумовий характер праці.

Проте активне впровадження у практику персональних комп'ютерів має двоякий характер. З одного боку, підвищується результативність праці, а з другого — з'являються фактори, які несприятливо впливають на здоров'я

працюючої людини. У зв'язку з цим набуває актуальності вивчення фізіологічних, психологічних, соціальних та виробничих наслідків використання ВДТ ПЕОМ, розробка та активне застосування заходів, що нормалізують працю та зберігають здоров'я користувачів.

Збереження здоров'я користувачів ВДТ, підтримання ефективності та надійності їх праці на належному рівні є одним з аспектів застосування дисципліни — охорона праці. Ця важлива прикладна наука вивчає небезпечні та шкідливі виробничі фактори, визначає ступінь їх впливу на організм працюючих і розробляє організаційні та технічні заходи щодо усунення або у всякому разі зменшення їх несприятливого впливу на працюючих.

Разом з тим, у межах охорони праці вирішуються питання створення технічних засобів захисту від дії шкідливих факторів, розробляються заходи щодо попередження нещасних випадків, професійного травматизму, профілактики професійних та професійно зумовлених захворювань.

Охорона праці — системо законодавчих актів, соціально-економічних, організаційно-технічних, гігієнічних і лікувально-профілактичних заходів та засобів, що забезпечують безпеку, збереження здоров'я та працездатність людини у процесі праці.

Все різноманіття нормативних документів, що включають поняття охорона праці, передбачає створення таких умов праці, за яких виключено або значно ослаблено вплив на працюючих шкідливих та небезпечних факторів трудового середовища.

Під шкідливим виробничим фактором розуміємо такий фактор виробничого середовища, вплив якого на працюючого завдає шкоди його здоров'ю та працездатності.

### **6.3 Аналіз потенційних небезпек та шкідливостей виробничого середовища**

Прискорене впровадження ЕОМ практично у всі галузі діяльності веде до появи великої кількості робочих місць з візуальними дисплейними терміналами (ВДТ), що є основним засобом спілкування з ЕОМ. Вони поширюються як на виробництві у різних системах контролю та управління, так і в різних адміністративно-господарських приміщеннях, де розташовуються обчислювальні центри організацій та інститутів. За даними всесвітньої організації охорони здоров'я, професійна діяльність з ВДТ може в окремих випадках призводити до порушень з боку зорового аналізатора, кістково-м'язових (вимушена поза) та порушень, пов'язаних із стресовими ситуаціями та нервово-емоційними навантаженнями при роботі, захворювань шкіри та ін.

Зараз в нашій країні є комплекс розроблених організаційних заходів і технічних засобів захисту, накопичений передовий досвід роботи великої кількості обчислювальних центрів який показує, що є можливість досягнути значно більших успіхів в справі знешкодження дії шкідливих та небезпечних виробничих факторів на персонал. Але стан умов праці та її безпеки при роботі з ЕОМ ще не задовільняє сучасним вимогам. Працівники ще стикаються з дією на них таких фізично небезпечних та шкідливих виробничих факторів, як дія шкідливих випромінювань від монітора та комп'ютера, підвищений рівень шуму, підвищена температура середовища, відсутність або недостатня освітленість робочої зони, електричний струм, статична електрика та ін. В період роботи з ВДТ на електронно-променевих трубках на організм користувача впливає цілий ряд факторів фізичної

природи, але всі вони знаходяться в межах і значно нижче за нормовані величини відповідно до діючих зараз нормативних документів (таблиця 7.1).

**Таблиця 6.1 – Вид електромагнітних випромінювань ВДТ**

<b>Види випромінювань</b>	<b>Діапазон</b>	<b>Фактичні (середні) дані вимірів</b>	<b>Нормовані значення</b>
Рентгенівське випромінювання	понад 1.2 Кев	9-10-12 мкр/год	75.0 мкр/год
Ультрафіолетове випромінювання	220-280 нм 280-320 нм	0 0-0.02	0.1 Вт/м <sup>2</sup>
Видимий діапазон	320-400 нм 400-700 нм	0.1-0.2 2.5-4.0	10.0 Вт/м <sup>2</sup>
Яскравість		75-80 кд/м <sup>2</sup>	Не менше 35 кд/м <sup>2</sup>
α-випромінювання	700 нм-1 мм	0.05-4.0	100.0 Вт/м <sup>2</sup>
Електростатичне поле	0 Гц	15 кВ/м	20-60 кВ/м

Велика кількість працюючих з ЕОМ зв'язані з дією на них таких факторів, як розумове перевантаження, перевантаження слухових та зорових аналізаторів, монотонність праці, емоційні перевантаження. Вплив вказаних факторів приводить до зниження працездатності, що викликається втому. Поява та розвиток втоми пов'язана з змінами, які виникають під час роботи в центральній нервовій системі, з тормозними процесами в корі головного мозку.

Світлотехнічна специфіка робочих місць з ВДТ викликана такими унікальними особливостями:

1.Світлотехнічна різномірність об'єктів зорової роботи користувачів ЕОМ, що працюють з ВДТ, пов'язана з наявністю трьох об'єктів (екран,

клавіатура, документація), розташованих у різних зонах спостереження, що вимагає багаторазового переведення лінії зору від одного до іншого. Робоча документація частіше за все розміщена на столі у горизонтальній площині на відстані оптимальної зони видимості (приблизно 350 мм), об'єкти розрізнення мають негативний контраст, тобто об'єкти на світловому фоні. Об'єкти на клавіатурі визначаються більшим розміром і розташовані у похилій площині. Яскраві знаки на темному фоні майже вертикально орієнтованого екрану дисплея вимагають незвично горизонтальної орієнтації лінії зору. Таким чином умови роботи з ВДТ ускладнюються необхідністю постійної переадаптації від яскравих об'єктів з позитивним контрастом на темні-з негативним.

2. Робота з пульсуючим самосвітним об'єктом, який постійно знаходиться у центрі поля зору, що не відповідає нормативним вимогам щодо обмеження пульсації та засліпленості. Наявність пульсацій яскравості знаків виникає дискомфорт і втому, загальну і зорову.

3. На робочому місці несприятливо розподілена яскравість у полі зору, оскільки освітлені поверхні периферії поля зору можуть виявитись світлішими, ніж центр поля зору- темний, обмежено освітлений, а іноді заповнений знаками екран ВДТ. Такий розподіл яскравості у полі зору сприяє порушенню основних зорових функцій.

4. Засліплююча дія світильників, які освітлюють приміщення на робочому місці з ВДТ більша, ніж інших, що призводить до зменшення кута дії різних засліплюючих джерел і, відповідно, до зростання освітленості.

5. Наявність дзеркального відбиваючої і неплоскої зовнішньої поверхні не дає можливості повністю усунути з поля зору користувача всі відбиті відблиски.

В приміщеннях найчастіше використовується бокове природне освітлення. Як наслідок слабкої освітленості робочого приміщення, створюється різкий контраст між освітленістю моніторів та освітленістю навколишніх предметів, що дуже шкідливо для зору оператора. В тих випадках коли одного природнього освітлення недостатньо, встановлюється сумісне освітлення. При цьому додаткове штучне освітлення використовується не тільки в темноті, але й в світлий час доби.

Штучне освітлення по характеру виконуваних завдань ділиться на робоче, аварійне, евакуаційне.

Раціональне кольорове оформлення приміщення направлене на покращення санітарно-гігієнічних умов праці, підвищення його продуктивності та безпеки. Оформлення приміщень діє на нервову систему людини, її настрій та в кінцевому рахунку на продуктивність праці. Основні виробничі приміщення доречно фарбувати в відповідності з кольором технічних засобів. Освітленість приміщення і обладнання повинно бути м'яким, без відблисків.

Зниження шуму, що створюється на робочому місці внутрішніми джерелами, а також шуму, який проникає ззовні, є дуже важливим завданням. Зниження шуму в джерелі можна забезпечити використання пружних прокладок між основою машини, приладу і опорою поверхні. В якості прокладок використовується гума, войлок, пробка, різної конструкції амортизаторів. Під настільні апарати, які є джерелами шуму, можна підкладати м'які коврики з синтетичних матеріалів, а під ножки столу, на яких вони встановлені, прокладки з м'якої гуми, войлока, товщиною 6-8 мм. Кріплення прокладок можливе шляхом приклеювання їх до опорних частин.



Можливе також використання звукопоглинаючих кожухів, які не заважають технологічному процесу. Не менш важливим для зниження шуму в процесі експлуатації є питання правильного і своєчасного регулювання, змащення і заміни механічних вузлів обладнання, яке шумить.

Зниження рівня шуму, що надходить в виробниче приміщення ззовні, може бути досягнуто збільшенням звукоізоляції огорожуючих конструкцій, притисненням притворів вікон, дверей.

Таким чином для зниження шуму, що створюється на робочих місцях внутрішніми джерелами, а також шуму, що надходить ззовні:

- ослабити шум самих джерел (застосування екранів, звукоізолюючих кожухів);
- знизити ефект сумарної дії відбитих звукових хвиль (звукопоглинаючі поверхні конструкцій);
- використовувати архітектурно-планувальні та технологічні вирішення ізоляцій джерел шуму;
- використовувати раціональне розміщення обладнання.

Електричні установки, до яких відносяться практично всі складові ЕОМ, носять для людини велику потенційну небезпеку, так як в процесі експлуатації або проведення профілактичних робіт людина може доторкнутися до частин, які знаходяться під напругою. Специфічна небезпека електроустановок: струмоведучі провідники, корпуси стійок ЕОМ та іншого обладнання, які опинилися під напругою в результаті пошкодження (пробою) ізоляції, не подають будь-яких сигналів, які попереджують людину про безпеку. Реакція людини на електричний струм виникає тільки при протіканні останнього через тіло людини.

Для захисту персоналу від ураження електричним струмом широко використовуються заземлювачі.

#### **6.4 Розрахунок освітленості робочого місця при розробці криптографічних систем**

Правильне освітлення виробничих приміщень сприяє підвищенню комфортності праці, збереженню здоров'я, зниженню імовірності нещасного випадку.

Розрахунок системи висвітлення зводиться до вибору виду освітлення, визначенню типу і числа світильників.

Будемо використовувати метод розрахунку по світловому потоку.

Вибір параметрів освітлення робочого місця залежить від характеру виконуваної роботи. Об'єкт розрізнення визначається як найменший розмір предмета або його частини, яку необхідно розрізнити.

В залежності від розмірів об'єкта розрізнення і відстані предмета від очей працюючого, всі роботи поділяються на вісім розрядів. Наша робота – високої точності, яка відповідає третьому розряду. Найменший лінійний розмір об'єкта розрізнення цього розряду – 0.3-0.5 мм.

В приміщенні передбачається організація штучного освітлення комбінованого типу, яке включає місцеве і загальне освітлення.

В якості джерела світла використовуються люмінесцентні лампи типу ЛБ (лампи білі), включені у світильники АОД.

Останні встановлюються на висоті 3.5 м від підлоги.

Лампи включаються в мережу з напругою 220 В, їхня потужність 40 Вт.

Нормована міні освітленість:

$$E_{\min} = (F_{\text{л}} \cdot n \cdot \eta \cdot z) / (S \cdot k) \quad (6.1)$$

$F_{\text{л}}$  – світловий потік 1-ї лампи, лм;

$n$  – число ламп у приміщенні;

$\eta$  – коефіцієнт використання світлового потоку, тобто частка світлового потоку всіх ламп, які попадають на освітлену поверхню;

$z$  – коефіцієнт нерівномірності висвітлення;

$S = A \cdot B$  – площа поля освітленого приміщення;

$k$  – коефіцієнт запасу, який враховує зниження освітленості в процесі експлуатації.

Для обраного типу ламп:

$$F_{\text{л}} = 2120 \text{ лм}; \quad S = 7 \cdot 5 = 35 \text{ м}^2; \quad \eta = 0.5; \quad z = 0.9; \quad k = 1.5$$

Нормована  $E_{\min}$  для третього розряду точності - 500 лк.

З формули (6.1) визначаємо число ламп:

$$n = (E_{\min} \cdot S \cdot k) / (z \cdot \eta \cdot F_{\text{л}}) = (500 \cdot 35 \cdot 1.5) / (0.9 \cdot 0.5 \cdot 2120) = 27.5 = 28$$

Так як в кожному світильнику розташовано по 2 лампи, всього необхідно 14 світильників.

## **6.5 Електромагнітний імпульс ядерного вибуху і захист від нього радіоелектронних засобів**

На початку 90-х років у США стала зароджуватися концепція, відповідно до якої збройні сили країни повинні мати не тільки ядерні і звичайні озброєння, але і спеціальні засоби, що забезпечують ефективну участь у локальних конфліктах без нанесення супротивнику зайвих втрат у живій силі і матеріальних цінностях.

До цієї спеціальної зброї американські військові фахівці в першу чергу відносять:

- засоби створення електромагнітного імпульсу (ЕМІ);
- генератори інфразвуку;
- хімічні склади і біологічні рецептури, здатні змінювати структуру базових матеріалів основних елементів бойової техніки;
- речовини, що виводять з ладу змащення і гумові вироби, викликають загустення пального;
- лазери.

В даний час основні роботи з розвитку технологій зброї не смертельної дії (ЗНСД) проводяться в керуванні перспективних досліджень міністерства оборони, Ліверморській і Лос-Аламоській лабораторіях міністерства енергетики, центрі розробок озброєння міністерства армії і т.д.

Найбільш близькі до прийняття на озброєння різні типи лазерів для осліплення особового складу, хімічні засоби для його знерухомилення, генератори ЕМІ, що негативно впливають на роботу електронної техніки.

Генератори ЕМІ (супер ЕМІ), як показують теоретичні роботи і проведені за рубежом експерименти, можна ефективно використовувати для виводу з ладу електронної й електротехнічної апаратури, для стирання інформації в банках даних і псування ЕОМ.

За допомогою ЗНСД на основі генераторів ЕМІ можливий вивід з ладу

ЕОМ, ключових радіо й електротехнічних засобів, систем електронного запалювання й інших автомобільних агрегатів, чи підірвавши інактивація мінних полів. Вплив цієї зброї досить вибірково і політично цілком прийнятний, однак потрібна точна доставка його в райони поразення.

Незважаючи на визнання військово-політичним керівництвом США і НАТО неможливості перемоги в ядерній війні, різні аспекти вражаючого дії ядерної зброї продовжують широко обговорюватися. Так, в одному з розглянутих іноземними фахівцями сценаріїв початкового періоду ядерної війни особливе місце приділяється потенційної можливості висновку з ладу радіоелектронної техніки в результаті впливу на неї ЕМІ.

Вважається, що підірвавши на висоті близько 400 км тільки одних боєприпасів потужністю більш 10 Мт приведе до такого порушення функціонування радіоелектронних засобів у великому районі, при якому час їхнього відновлення перевищить припустимі терміни для вживання відповідних заходів.

По розрахунках американських експертів, оптимальною точкою підризу ядерних боєприпасів для поразки ЕМІ радіоелектронних засобів майже на всій території США була би точка в космосі з епіцентром у районі географічного центра країни, що знаходиться в штаті Небраска.

Теоретичні дослідження і результати фізичних експериментів показують, що ЕМІ ядерного вибуху може привести не тільки до виходу з ладу напівпровідникових електронних пристроїв, але і до руйнування металевих провідників кабелів наземних споруд. Крім того, можлива поразка апаратури, що знаходяться на низьких орбітах.

Для генерації ЕМІ ядерні боєприпаси можуть підриватися в космічному

просторі, що не приводить до виникнення ударної хвилі і випаданню радіоактивних опадів. Тому в закордонній пресі виголошуються наступні думки про "неядерний характер" такого бойового застосування ядерної зброї і про те, що удар з використанням ЕМІ не обов'язково приведе до загальної ядерної війни.

Небезпека цих заяв очевидна, тому що одночасно деякі закордонні фахівці не виключають можливість масової поразки за допомогою ЕМІ і живої сили. У всякому разі цілком очевидно, що наводимі під впливом ЕМІ в металевих елементах техніки струми і напруги будуть смертельно небезпечні для особового складу.

## **6.6 Забезпечення нормальних умов праці**

### **6.6.1 Вибір приміщення**

При виборі приміщення для робочих місць необхідно врахувати, що вони можуть давати відблиски на екранах дисплеїв і викликати значну освітленість у сидячих перед ними, особливо літом та в сонячні дні.

В машинних залах робочі місця операторів, які працюють з дисплеями, розміщують подалі від вікон і таким чином, щоб віконні пройоми виходили збоку. Якщо екран дисплея повернутий до віконного пройому, необхідні спеціальні екрануючі пристрої. Вікна рекомендується оснащувати світлорозсіюючими шторами, регулюючими жалюзями або сонцезахисною сіткою з металізованим покриттям. Площа засклення не повинна перевищувати 25 % від площі стіни з вікнами. Для мінімізації засвічування від

сонячних променів екранів ВДТ вікна можуть бути орієнтовані на північ (північний захід, північний схід).

Необхідно забезпечити відповідне оформлення інтер'єра, бо давати відблиски на екранах і сліпити працюючих можуть не тільки вікна, але й інші поверхні великої яскравості, в тому числі стеля, стіни, поверхні столів, шаф. Тому все повинно мати невисокі коефіцієнти віддзеркалювання: робочого столу, корпусу та клавіатури- 0.2-0.5; стелі- 0.6-0.7; стін- 0.2-0.5; підлоги- 0.1-0.2; шаф та стелажів- 0.25-0.35. Всі оздоблювальні матеріали приміщення повинні бути матового кольору.

#### **6.6.2 Забезпечення нормальних санітарно- гігієнічних умов на робочому місці**

Приміщення, їх розміри (площа, об'єм) повинні в першу чергу відповідати кількості працюючих і розміщеному в них комплекту технічних засобів. В них передбачаються відповідні параметри температури, освітлення, чистоти повітря, забезпечують ізоляцію, захист від виробничих шумів і т. п. Робочі місця з ВТД вимагають значної площі на одне робоче місце (6-7 м<sup>2</sup>), а при середній площі устаткування 0.8-1.2 м<sup>2</sup> навколо нього має залишатись вільний простір не менше 1 м з кожного боку. Для забезпечення нормальних умов праці санітарні норми СН245-71 встановлюються на одного працівника, об'єм виробничого приміщення не менше 15 м<sup>2</sup>, а площа приміщення огороженого глухими перегородками не менше 4.5 м<sup>2</sup>. В дисплейних класах температура повітря повинна становити 19-21 °С, відносна вологість 55-65 %, швидкість руху повітря не більше 0.1 м/с, рівні шуму та еквівалентні рівні звуку в приміщеннях не повинні перевищувати 50дБ .

Для експлуатації ЕОМ необхідно передбачити наступні приміщення:

- машинний зал;
- приміщення для розміщення сервісного і периферійного обладнання;
- приміщення для зберігання запасних деталей, інструментів приладів;
- приміщення для розміщення вентиляторів;
- приміщення для персоналу;
- приміщення для прийому- видачі інформації.

Ці приміщення розміщують в безпосередній близькості одне від одного. Їх обладнують загальнообмінною вентиляцією та штучним освітленням. До приміщень машинного залу і зберігання магнітних носіїв інформації ставляться особливі вимоги. Площа машинного залу повинна відповідати площі необхідної по заводським технічним умовам даного типу ЕОМ.

Висота залу над технічною підлогою до підвісної стелі повинна бути 3-5 м. Відстань між підвісною та основною стелями при цьому повинна бути 0,5-0,8 м. Висоту припідлогового простору приймають рівною 0,2-0,6 м.

При роботі з ЕОМ, як правило, застосовують бічне природне освітлення. Робочі кімнати і кабінети повинні мати природне освітлення. В решті приміщень допускається штучне освітлення. В таблиці 6.2 приведені рівні освітленості в приміщеннях з ВДТ при використанні різних типів дисплеїв.

В тих випадках, коли одного штучного освітлення не достатньо, встановлюється сумісне освітлення. При цьому додаткове штучне освітлення використовується як в темноті, так і в світлий час доби.

Раціональне планування приміщення, розміщення устаткування є важливим фактором, який дозволяє понизити шум при існуючому



устаткуванні. При плануванні машинний зал та приміщення для сервісної апаратури необхідно розташувати якнайдалі від устаткування, яке дуже шумить або вібрає.

**Таблиця 6.2 - Рівні освітленості в приміщеннях з ВДТ**

Характеристики дисплеїв зорової роботи		Нормована освітленість (лк) у площині столу, клавіатури (Ег), та екрану (Ев)			
Вид дисплея	Група напруженості зорової роботи	Загальне		Комбіноване	
		Ег, лк	Ев, лк	Ег, лк	Ев, лк
Одноколірний		300	150-100	400	150-100
		400	200-150	500	300-150
Багатоколірний	Будь-яка	-	-	200	100-75

Зниження рівня шуму, який проникає в виробниче приміщення ззовні може бути досягнуте збільшенням звукоізоляції огорожуючих конструкцій [23].

Для профілактики порушень та підтримання працездатності необхідно витримувати регламентовані перерви для відпочинку. У період роботи за дисплеєм у режимі праці та відпочинку необхідно передбачити через кожні 40-45 хв. три- або п'ятихвилинні перерви для відпочинку.

### **6.6.3 Розрахунок освітлення машинного залу**

Найбільш оптимальними для машинного залу є люмінісцентні лампи ЛБ (білого світла) і ЛТБ (тепло-білого світла) потужністю 20, 40 та 80 Вт.

Для виключення засвітки екранів дисплеїв прямими світловими потоками світильники загального призначення розміщують збоку від робочого місця, паралельно лінії зору оператора і стінці з вікнами. Таке розміщення світильників дозволяє проводити їх послідовне включення в залежності від

величини дійсної освітленості і виключає подразнення очей полосами світла і тіні, які виникають при поперечному розміщенні світильників.

Для освітлення машинного залу розмірами  $A=9$  м,  $B=6$  м і висотою  $H=3$  м передбачені настельні світильники типу УСП 3 з двома люмінесцентними лампами типу ЛБ-40. Коефіцієнт відображення світлового потоку від стелі, стін і підлоги відповідно  $\rho_{ст}=70\%$ ,  $\rho_{с}=50\%$ ,  $\rho_{м}=10\%$ . Необхідне число світильників визначається наступним чином.

Для машинних залів рівень робочої поверхні над підлогою складає 0.8 м.

Тоді

$$h = H - 0.8 = 2.2 \text{ м.} \quad (6.1)$$

У світильників УСП 35 умовна віддаль  $\xi = 1.4$

$$\frac{L}{h} = \xi = 1.4, \quad (6.2)$$

де  $L$ - відстань між рядами світильників,

$$L = \xi * h = 1.4 * 2.2 \approx 3 \text{ м.} \quad (6.3)$$

Розмістимо світильники вздовж довгої сторони приміщення. Відстань між стиками і крайніми рядами приймаємо рівною

$$l \approx (0.3 \dots 0.5) * L. \quad (6.4)$$

При ширині машинного залу  $B=6$  м маємо число рядів світильників

$$n = \frac{B}{L} = 2. \quad (6.5)$$

Для машинних залів встановлена норма освітленості  $E_n=400$  лк. З врахуванням  $\rho_{ст}$ ,  $\rho_{с}$ ,  $\rho_{м}$  при індексі приміщення з довідкових даних знаходимо коефіцієнт використання випромінюваного світильниками потоку, який приходить на розрахункову величину,  $\eta=0.45$ .

$$j = \frac{A * B}{(h * (A + B))} = \frac{54}{(2.2 * (9 + 6))} \approx 1.64 \quad (6.6)$$

Номінальний світловий потік лампи ЛБ-40  $\Phi_{л}=3120$  лм, тоді світловий потік, який випромінюється світильником складе

$$\Phi_{св} = 2 * \Phi_{л} = 2 * 3120 = 6240 \text{ лм.} \quad (6.7)$$

Необхідну кількість світильників знаходимо за наступною формулою

$$N = \frac{E_n * k_z * S * z}{n * \Phi_{св} * \eta * \gamma}, \quad (6.8)$$

де  $E_n$  - нормована мінімальна освітленість,  $E_n = 400 \text{ лк}$ ;

$k_z$  - коефіцієнт запасу, який враховує запилення світильників і знос джерел світла в процесі експлуатації,  $k_z = 1,4 \dots 1,5$ ;

$S$  - площа приміщення, м;

$z$  - коефіцієнт нерівномірності освітлення  $z = 1,1 \dots 1,2$ ;

$n$  - число рядів світильників;

$\eta$  - коефіцієнт затінення, який вводиться для приміщень з фіксованим положенням працюючих, а також при наявності крупногабаритних предметів,  $\gamma = 0,8 \dots 0,9$ .

$$N = \frac{(400 * 1.5 * 54 * 1.2)}{(2 * 6240 * 0.45 * 0.9)} = 7.692 \approx 8 \text{ ламп.}$$

Отже, необхідно 8 світильників.

## ВИСНОВКИ

У цій роботі для збільшення швидкості офлайн-програмування була створена серія розумних компонентів та надбудов. Кожен розумний компонент містив необхідну геометрію для створення графічної складової. Розумні компоненти мають можливість створювати та модифікувати механізми, завдяки чому графічні компоненти візуально відображатимуть функцію, яку виконує механізм. Розумні компоненти також мають можливість завантажувати несуттєву геометрію та прикріплювати її до належного кадру, щоб станція була якомога точнішою під час досліджень охоплення та автономного програмування.

Серія розумних компонентів складається з трьох частин. Перший - це автоматичний будівельник огорожі, який дозволяє користувачеві вибрати бажану довжину та висоту огорожі та буде його на станції. Другий розумний компонент - це конструктор доріжок. Цей розумний компонент завантажує бажану довжину колії; виходячи з обраної висоти; і довжини стріли. Користувач також має можливість завантажувати аксесуари та автоматично розміщувати та прикріплювати їх до належного положення на козлі. Цей розумний компонент також генерує козловий механізм на основі висоти вибраної вежі та довжини колії проїзду. Останній розумний компонент - це автоматичний компонент світлової завіси. Цей компонент дозволяє користувачеві створити кілька світлових завіс і автоматично побудувати віртуальний лазерний паркан, який знаходиться між двома стовпами світлових завіс.

Додатковий компонент був створений для повної автоматизації конструкції роботизованого зварювального елемента. Автоматичний конструктор станцій конструює зварювальну камеру з урахуванням основних атрибутів деталі: ваги в кг, діаметра в мм і довжини в мм. Цей компонент використовує інтелектуальну серію компонентів для створення відповідного робота та позиціонера деталей та включає в себе обладнання безпеки, таке як огорожа та легкі завіси. Всі компоненти мають автономний розмір і розміщені для побудови зварювальної комірки, здатної зварювати зазначену деталь.

Надбудова була створена в Microsoft Visual Studio C #, щоб дозволити користувачеві швидко завантажити позиціонер та бажаний райзер. Позиціонер завантажується у правильне положення для калібрування віртуального контролера. Головний запас та хвостовий запас вибираються на основі бажаної потужності та висоти стояка. SkyHooks вибираються залежно від ємності, падіння та кидка. Відповідний стояк завантажується на основі більших

вимірювань падіння та кидка. Крім того, Центри падіння завантажуються з урахуванням потужності, падіння, кидка та висоти стояка.

Серія автоматичних розумних компонентів, додатковий компонент автоматизації та надбудова позиціонера працюють за призначенням. Бібліотеки надзвичайно корисні та економічні за часом під час створення та оновлення концептуальної комірки в ABB RobotStudio. Спочатку деталі та позиціонери Wolf Robotics були розподілені між 190 окремими папками і займали майже 7 Гігабайт комп'ютерної пам'яті. Завдяки використанню інтелектуальних компонентів та надбудови, тепер можна отримати доступ до 2428 унікальних механізмів із 8 простих меню, за винятком опцій аксесуарів.

Створення механізмів таким чином не тільки економить час користувача при розробці концептуальної станції, але й економить значну пам'ять комп'ютера. Метод попередньої компіляції бібліотек, що застосовувався раніше, дублював геометрію в кожному механізмі. Новий метод використовує єдиний фрагмент геометрії, який можна розділити між різними механізмами. Пам'ять, яку використовують бібліотеки, зменшилась на 80%.

RobotStudio SDK надзвичайно потужний у створенні та модифікації станцій. Багато функцій, якими користувачі можуть скористатися для створення клітинок концепції, також доступні в SDK. Це дозволяє передавати знання зі створення стандартних комірок в SDK для створення програм для виконання завдань, які часто виконуються.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Bengtsson, Daniel, and Carl-Johan Rutgersson. Development of BoxSweeper and BoxSweeper PLC.
2. Göteborg, Sweden : Chalmers University of Technology, 2008.
3. Optimal Robot Placement Using Response Surface Method. Xiaolong Feng, et al. 2009, International Journal Of Advanced Manufacturing Technology 44.1/2 (2009), pp. 201-210.
4. MAN: Modern Applications News. Using CMM Arm Slashes Robot Programming 90%. 2004, Vol. 38, 10.
5. Mechanical Engineering. Loading Features on Palletizing Robots. 2011, Vol. 133, 9.
6. ABB. Operating Manual: Robot Studio 5.14. Document ID: 3HAC032104-001 Revision: D. 2008.
7. Release Notes: RobotStudio SDK 5.14. 2011.
8. Robotics, Fanuc. Accompanying Training Manual: Roboguide V6.40 Rev.B.
9. MotoMan. Instruction Manual MotoVisual Component Library. Reg No: ME00040EN-00. 2008.
10. Spaak, Anders. Parametric Fence Ver. 3. ABB RobotApps. [Online] May 29, 2012.  
<http://www.abb.com/product/ap/seitp327/5dd4fbc7afcb82d2c12579ad0054401d.aspx>.
11. Thomas, Dean. Parametric Robot Stand. ABB RobotApps. [Online] January 01, 2011. <https://robotapps.robotstudio.com/Details.aspx?fileId=367>.
12. Ramos, Richard. Schunk Gripper. ABB RobotApps. [Online] November 24, 2011. <https://robotapps.robotstudio.com/Details.aspx?fileId=365>.

13. Fogbring, Simon. Coordinate File Import 5.14 ver.2. Abb RobotApps. [Online] June 17, 2011.  
<http://www.abb.com/product/ap/seitp327/5dd4fbc7afcb82d2c12579ad0054401d.aspx>.
14. Admin, ABB. RS User Library Addin. ABB RobotApps. [Online] November 11, 2008.  
<http://www.abb.com/product/ap/seitp327/5dd4fbc7afcb82d2c12579ad0054401d.aspx>.
15. ABB. Application Manual: FlexPendant SDK RobotWare 5.14. Document ID: 3HAC036958-001 Revision: A. 2010.
16. Application Manual: PC SDK RobotWare 5.14. Document ID: 3HAC036957-001 Revision: A. 2010.
17. Application Manual: Robot Application Builder RobotWare 5.0. Document ID: 3HAC028083-001 Revision: D. 2008.
18. MotoMan. MotoVisual 3D Simulation Software Quick Start Guide 2007. 2007.
19. ABB System Programs Robots in Parallel. March 2006, Vol. 31, 3, pp. 45-45.
20. Orientation Modeling of Bernoulli Gripper Device with Off-Centered Masses of the Manipulating Object / V. Savkiv, R. Mykhailyshyn, O. Fendo, M. Mykhailyshyn // Procedia Engineering. – 2017. – № 187. – P. 264 – 271. – DOI: 10.1016/j.proeng.2017.04.374.
21. Justification of Design and Parameters of Bernoulli-Vacuum Gripping Device / V. Savkiv, R. Mykhailyshyn, F. Duchon, O. Fendo // International Journal of Advanced Robotic Systems. – 2017. – DOI: 1729881417741740.
22. Energy efficiency analysis of the manipulation process by the industrial objects with the use of Bernoulli gripping devices / V. Savkiv, R. Mykhailyshyn, F.

- Duchon, M. Mikhalishin // Journal of Electrical Engineering. – 2017. – № 68 (6). – P. 496 – 502. – DOI: 10.1515/jee-2017-0087.
23. Experimental Research of the Manipulation Process by the Objects Using Bernoulli Gripping Devices / R. Mykhailyshyn, V. Savkiv, M. Mikhalishin, F. Duchon // In Young Scientists Forum on Applied Physics and Engineering, International IEEE Conference. – 2017. – P. 8 – 11. – DOI: 10.1109/YSF.2017.8126583.
  24. Modeling of Bernoulli gripping device orientation when manipulating objects along the arc / V. Savkiv, R. Mykhailyshyn, F. Duchon, M. Mikhalishin // International Journal of Advanced Robotic Systems. – 2018. – DOI: 1729881418762670.
  25. Investigation of the energy consumption on performance of handling operations taking into account parameters of the grasping system / R. Mykhailyshyn, V. Savkiv, F. Duchon, V. Koloskov, I. Diahovchenko // 2018 IEEE 3rd International Conference on Intelligent Energy and Power Systems (IEPS) – IEEE, 2018. – P. 295 – 300. – DOI: 10.1109/ieps.2018.8559586.
  26. Analysis of frontal resistance force influence during manipulation of dimensional objects / R. Mykhailyshyn, V. Savkiv, F. Duchon, V. Koloskov, I. Diahovchenko // 2018 IEEE 3rd International Conference on Intelligent Energy and Power Systems (IEPS) – IEEE, 2018. – P. 301 – 305. – DOI: 10.1109/ieps.2018.8559527.
  27. Substantiation of Bernoulli Grippers Parameters at Non-Contact Transportation of Objects with a Displaced Center of Mass / R. Mykhailyshyn, V. Savkiv, F. Duchon, P. Maruschak, O. Prentkovskis // 22nd International Scientific Conference Transport Means 2018. – Klaipeda, 2018. – P. 1370 – 1375.



28. Gasdynamic analysis of the Bernoulli grippers interaction with the surface of flat objects with displacement of the center of mass / V. Savkiv, R. Mykhailyshyn, F. Duchon // *Vacuum*. – 2019. – № 159, P. 524 – 533. – DOI: 10.1016/j.vacuum.2018.11.005.
29. Protection of Digital Power Meters Under the Influence of Strong Magnetic Fields / R. Mykhailyshyn, V. Savkiv, I. Diahovchenko, R. Olsen, D. Danylchenko // 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering UKRCON-2019 – IEEE, 2019. – P. 314 – 320. – DOI: 10.1109/UKRCON.2019.8879985.
30. Research of Energy Efficiency of Manipulation of Dimensional Objects With the Use of Pneumatic Gripping Devices / R. Mykhailyshyn, V. Savkiv, I. Diahovchenko, F. Duchon, R. Trembach // 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering UKRCON-2019 – IEEE, 2019. – P. 527 – 532. – DOI: 10.1109/UKRCON.2019.8879957.
31. Diahovchenko, I., Lebedynskiy, I., Mykhailyshyn, R., & Savkiv, V. (2019, September). Methods to Improve the Accuracy of Power Meters through the Application of Nanomaterials and Calibration Techniques. In 2019 IEEE 9th International Conference Nanomaterials: Applications & Properties (NAP) (pp. 02NEE17-1). IEEE. doi: 10.1109/NAP47236.2019.216994.
32. Rogowsky coil applications for power measurement under non-sinusoidal field conditions / I. Diahovchenko, R. Mykhailyshyn, D. Danylchenko, S. Shevchenko // *Energetika*. – 2019. – 65(1), P. 14 – 20. – DOI: 10.6001/energetika.v65i1.3972.
33. Control of a small quadrotor for swarm operation / A. Trizuljak, F. Duchoň, J. Rodina, A. Babinec, M. Dekan, R. Mykhailyshyn // *Journal of Electrical Engineering*. – 70(1). – 2019. – P. 3-15. – DOI: 10.2478/jee-2019-0001.

34. Optimization of design parameters of Bernoulli gripper with an annular nozzle / V. Savkiv, R. Mykhailyshyn, P. Maruschak, L. Chovanec, E. Prada, I. Virgala, O. Prentkovskis // *Transport Means - Proceedings of the International Conference*. – 2019. – P. 423-428.
35. The analysis of influence of a nozzle form of the Bernoulli gripping devices on its energy efficiency / V. Savkiv, R. Mykhailyshyn, P. Maruschak, F. Duchon, L. Chovanec // *Proceedings of ICCPT 2019, May 28-29, 2019*. – Tern. : TNTU, Scientific Publishing House “SciView”, 2019. – P. 66–74. – DOI: 10.5281/zenodo.3387275.
36. Usage of Light-Emitting-Diode Lamps in Decorative Lighting / R. Mykhailyshyn, I. Belyakova, V. Medvid, V. Piscio, O. Shkodzinsky, M. Markovych // *IEEE 20th International Conference on Computational Problems of Electrical Engineering (CPEE)*. – IEEE, 2019. – DOI: 10.1109/CPEE47179.2019.8949154.
37. Justification of Influence of the Form of Nozzle and Active Surface of Bernoulli Gripping Devices on Its Operational Characteristics / V. Savkiv, R. Mykhailyshyn, P. Maruschak, F. Duchon // *TRANSBALTICA XI: Transportation Science and Technology. Lecture Notes in Intelligent Transportation and Infrastructure*. – Springer, 2020. — P. 263–272. – DOI: 10.1007/978-3-030-38666-5\_28.
38. Analysis of Operational Characteristics of Pneumatic Device of Industrial Robot for Gripping and Control of Parameters of Objects of Manipulation / V. Savkiv, R. Mykhailyshyn, P. Maruschak, F. Duchon, O. Prentkovskis, I. Diahovchenko // *TRANSBALTICA XI: Transportation Science and Technology. Lecture Notes in Intelligent Transportation and Infrastructure*. – Springer, 2020. — P. 504–510. – DOI: 10.1007/978-3-030-38666-5\_53.

39. Progress and Challenges in Smart Grids: Distributed Generation, Smart Metering, Energy Storage and Smart Loads / Diahovchenko, I., Kolcun, M., Čonka, Z., Savkiv, V., Mykhailyshyn, R. // Iranian Journal of Science and Technology, Transactions of Electrical Engineering, – 2020. – P. 1-15.
40. Михайлишин Р.І. Обґрунтування параметрів та орієнтації струминного захоплювача маніпулятора для автоматизації вантажно-розвантажувальних операцій: автореф. дис. на здобуття наук. ступеня канд. техн. наук : спец. 05.05.05 “Піднімально-транспортні машини” / Р.І. Михайлишин. – Тернопіль, 2018. – 21 с.
41. Михайлишин Р. І. Optimization of bernoulli gripping device’s orientation under the process of manipulations along direct trajectory / Р.І. Михайлишин, Я. І. Проць, В.Б. Савків // Вісник ТНТУ. – Тернопіль, 2016. – Том 81. – № 1. – С. 107 – 117.
42. Михайлишин Р. І. Аналіз методів планування траєкторій маніпуляторів / Р.І. Михайлишин, В.Б. Савків // Збірник наукових праць «Перспективні технології та прилади» Луцький НТУ. – Луцьк, 2016. – №8 (1). – С. 61 – 69.
43. Justification of the object of manipulation parameters influence on the optimal orientation and lifting characteristics of Bernoulli gripping device / В.Б. Савків, Р.І. Михайлишин, Ф. Духон, М.С. Михайлишин // Вісник Херсонського національного технічного університету. – Херсон, 2017. – № 2 (61). – С. 98 – 104.
44. «Ознайомлення з основними функціями програмного середовища RobotStudio» : методичні вказівки до лабораторної роботи № 1 з курсу “Гнучкі комп'ютеризовані системи та робототехніка” для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»

- / укл. Р.І. Михайлишин, В.Б. Савків. – Тернопіль : ТНТУ імені Івана Пулюя, 2019. – 45 с.
45. «Визначення базових точок та траєкторії промислового робота» : методичні вказівки до лабораторної роботи № 2 з курсу “Гнучкі комп’ютеризовані системи та робототехніка” для студентів спеціальності 151 «Автоматизація та комп’ютерно-інтегровані технології» / укл. : Р. І. Михайлишин, В. Б. Савків. – Тернопіль : ТНТУ імені Івана Пулюя, 2019. – 17 с.
  46. Методичні вказівки до лабораторної роботи № 3 «Імпорт тривимірних моделей та створення захоплювального пристрою в програмному середовищі RobotStudio» з курсу “Гнучкі комп’ютеризовані системи та робототехніка” для студентів спеціальності 151 «Автоматизація та комп’ютерно-інтегровані технології» / укл. : Р. І. Михайлишин, В. Б. Савків. – Тернопіль : ТНТУ імені Івана Пулюя, 2019. – 24 с.
  47. «Робота з віртуальним пультом управління FlexPendant в програмному середовищі RobotStudio» методичні вказівки до лабораторної роботи № 4 з курсу “Гнучкі комп’ютеризовані системи та робототехніка” для студентів спеціальності 151 «Автоматизація та комп’ютерно-інтегровані технології» / укл. : Р. І. Михайлишин, В. Б. Савків. – Тернопіль : ТНТУ імені Івана Пулюя, 2019. – 23 с.
  48. «Операції над об’єктами та контроль зіткнень в програмному середовищі RobotStudio» методичні вказівки до лабораторної роботи № 5 з курсу “Гнучкі комп’ютеризовані системи та робототехніка” для студентів спеціальності 151 «Автоматизація та комп’ютерно-інтегровані технології» / укл. : Р.І. Михайлишин, В.Б. Савків. – Тернопіль: ТНТУ імені Івана Пулюя, 2019. – 34 с.

49. «Розробка механізму конвеєра та програмування операцій MultiMove в програмному середовищі RobotStudio» методичні вказівки до лабораторної роботи № 6 з курсу “Гнучкі комп'ютеризовані системи та робототехніка” для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / укл. : Р. І. Михайлишин, В. Б. Савків. – Тернопіль : ТНТУ імені Івана Пулюя, 2019. – 39 с.
50. «Створення роботизованої станції в програмному середовищі RobotStudio» методичні вказівки до лабораторної роботи № 7 з курсу “Гнучкі комп'ютеризовані системи та робототехніка” для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / укл. : Р. І. Михайлишин, В. Б. Савків. – Тернопіль : ТНТУ імені Івана Пулюя, 2019. – 19 с.
51. «Розробка роботизованої лінії для автоматизації вантажно-розвантажувальних операцій в програмному середовищі RobotStudio» методичні вказівки до лабораторної роботи № 8 з курсу “Гнучкі комп'ютеризовані системи та робототехніка” для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / укл. : Р. І. Михайлишин, В. Б. Савків. – Тернопіль : ТНТУ імені Івана Пулюя, 2019. – 24 с.