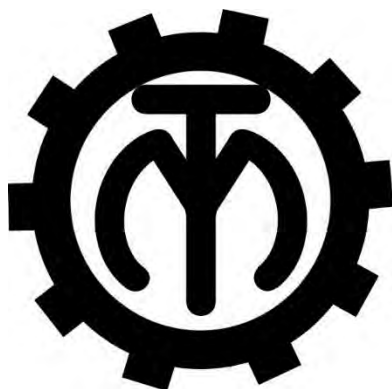


ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Технология машиностроения»

ПРОЕКТИРОВАНИЕ РОБОТОВ И РОБОТОТЕХНИЧЕСКИХ СИСТЕМ

*Методические рекомендации к лабораторным работам
для студентов направления подготовки
15.03.06 «Мехатроника и робототехника»
дневной формы обучения*



Могилев 2018

УДК 621.01
ББК 34.41
П 75

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Технология машиностроения» «28» мая 2018 г.,
протокол № 11

Составитель канд. техн. наук, доц. А. В. Капитонов

Рецензент канд. техн. наук, доц. В. В. Кутузов

Методические рекомендации содержат указания для выполнения лабораторных работ в соответствии с рабочей программой дисциплины «Проектирование роботов и робототехнических систем», а также необходимые справочные данные.

Учебно-методическое издание

ПРОЕКТИРОВАНИЕ РОБОТОВ И РОБОТОТЕХНИЧЕСКИХ СИСТЕМ

Ответственный за выпуск	В. М. Шеменков
Технический редактор	А. А. Подошевка
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 46 экз. Заказ №

Издатель и полиграфическое исполнение:
Государственное учреждение высшего профессионального образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 24.01.2014.
Пр. Мира, 43, 212000, Могилев.

© ГУ ВПО «Белорусско-Российский
университет», 2018



Содержание

Лабораторная работа № 1. Проектирование механической системы роботов. Лабораторная работа № 2. Проектирование конструкции и расчет схватов манипулятора.	4
Лабораторная работа № 3. Проектирование механизмов вращения в робототехнических устройствах. Лабораторная работа № 4. Проектирование механизмов поступательного движения в робототехнических устройствах.	5
Лабораторная работа № 5. Проектирование конструктивных модулей. Лабораторная работа № 6. Проектирование динамических моделей манипуляторов. Лабораторная работа № 7. Проектирование роботов с рекуператорами энергии	26

Лабораторная работа № 1. Проектирование механической системы роботов. Лабораторная работа № 2. Проектирование конструкции и расчет схватов манипулятора

Цель лабораторных работ – изучение методик проектирования и анимации (движения) механизмов цифровых прототипов технологического оборудования и оснастки в среде системы RobotStudio.

1 Техническое обеспечение.

- 1 Персональная ЭВМ.
- 2 Накопитель на жестких магнитных дисках.
- 3 Видеомонитор.

2 Программное обеспечение.

- 1 Операционная система Windows.
- 2 Пакет программ моделирования и программирования роботизированных технологических комплексов RobotStudio.

Общие сведения о проектировании и анимации цифровых прототипов технологического оборудования при моделировании и программировании технологических комплексов

При выполнении операций загрузки и выгрузки технологического оборудования необходимо синхронизировать работу робота, схвата и автоматизированной транспортной системы (конвейера, тактового стола, позиционера и др.).

Для решения этой задачи создаются специальные трехмерные цифровые прототипы, которые состоят из нескольких частей, например, загружаемых деталей, паллет и самого автоматизированного транспортного устройства.

Задание движения схватов и транспортных устройств позволяет оценить правильность взаимной ориентации схвата, перемещаемых объектов в процессе их движения на конвейере. Кроме того, анимация роботизированного технологического процесса дает возможность создать видеоролик для эффектной презентации проекта заказчику РТК.

3 Основные этапы выполнения работ.

- 1 Создание компоновки РТК из библиотеки цифровых прототипов.
- 2 Создание груза инструментами рисования RobotStudio.
- 3 Создание механизма схвата.
- 4 Задание перемещений звеньев механизма схвата.
- 5 Выполнение индивидуального задания и составление отчета по работе.

4 Методика выполнения лабораторных работ.

4.1 Создание РТК (станции).

Загрузим систему **RobotStudio** и создадим РТК на базе шабло-



на IRB 660_250kg_3.15m.

Для использования оригинальных цифровых прототипов скопируем файлы цифровых прототипов оборудования из папки **Лаб3_Геометрия** в папку **Geometry** по следующему пути: **C:\Program Files\ABB Industrial IT\Robotics IT\RobotStudio 5.13\ABB Library\Geometry**.

Из библиотеки цифровых прототипов **Geometry** на площадку РТК установим конвейеры **conv_box_top**, **conv_pallets_top**.

Установим силовой блок и контроллер IRC5 в пространстве станции на безопасном расстоянии от робота. Рабочая зона робота приведена на рисунке 14.

Из библиотеки цифровых прототипов **Geometry** на площадку РТК установим **Стол** (это пьедестал для робота). Командой **Set Position** определим положение стола с координатами $X = 450$, $Y = 350$, $Z = 540$, углы поворота вокруг осей X и $Y - 90$ град.

Переместим робота и его систему координат на пьедестал. Командой **System Configuration** (Конфигурирование системы) на вкладке **Offline** зададим новое положение робота и его системы координат, как показано на рисунке 1.

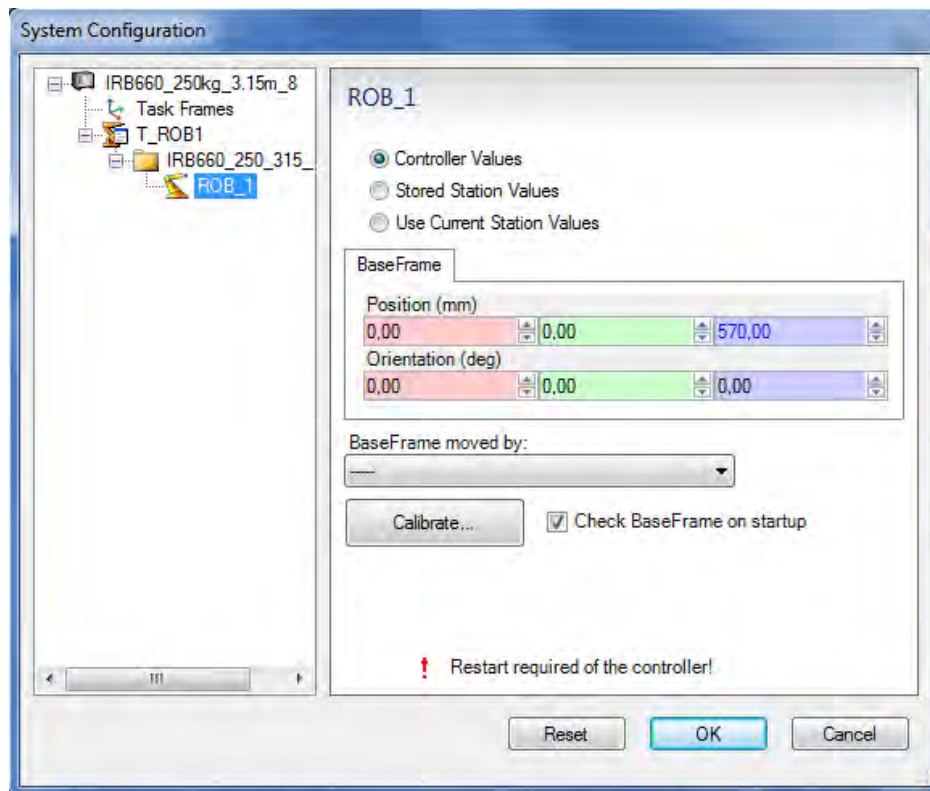


Рисунок 1 – Перемещение робота и его системы координат

В процессе изменения конфигурации появляется запрос на рестарт контроллера. Необходимо подтвердить перезагрузку контроллера нажатием кнопки **Да**.

Во время перезагрузки контроллера возможно появление запроса на обновление данных (рисунок 2). Необходимо подтвердить обновление данных нажатием кнопки **Yes**. После завершения создания новой конфигурации системы робот разместится на пьедестале.

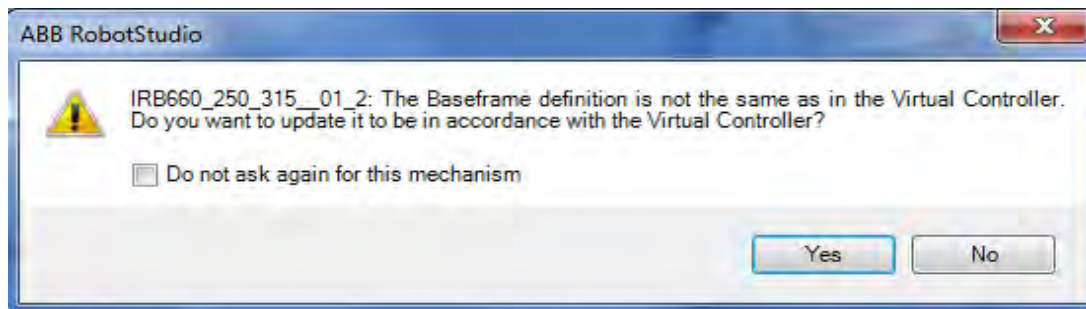


Рисунок 2 – Запрос на обновление данных для виртуального контроллера

Создадим объект **Груз**. На вкладке **Modeling** в меню **Solid** выберем объект **Box** (Ящик) и зададим его габаритные размеры:

Length (длина): 210 мм;

Width (ширина): 1000 мм;


Height (высота): 200 мм.

Командой **Set Position** определим положение груза, задавая координаты $X = 2000$, $Y = 110$, $Z = 1400$, угол поворота вокруг оси $X - 180$ град.

Из библиотеки цифровых прототипов **Geometry** на площадку РТК установим паллету (файл **Pallet.sat**). Командой **Set Position** зададим координаты ее положения $X = 1700$, $Y = -350$, $Z = 415$, угол поворота вокруг оси $Y - 90$ град.

4.2 Создание механизма открытия-закрытия схвата.

Загрузим файлы цифровых прототипов деталей схвата **tool_part1** и **tool_part2** из библиотеки **Geometry**. Переместим **tool_part2**, задавая командой **Set Position** его координаты $X = 0$, $Y = 180$, $Z = 55$, угол поворота вокруг осей – 0 град.

В верхней части окна программы выберем вкладку **Modeling** и нажмем кнопку  «**Create Mechanism**» (Создать механизм).

В окне **Create Mechanism** (рисунок 3) введем название механизма **My_Gripper** (Мой схват) для его идентификации на дереве проекта. Выберем тип механизма **Tool** (Инструмент).

Выделим узел **Links** (Связи) и вызовем его контекстное меню нажатием правой клавиши мыши. В контекстном меню объекта **Links** щелкнем **Add link** (Добавить связь). Открывается окно **Create Link** для создания связей между подвижными элементами схвата.

В диалоговом окне **Create Link** (Создайте связь) (рисунок 4) в поле **Selected Part** (Выберите звено) выберем **tool_part1** (главная неподвижная часть схвата) и поставим галочку напротив **Set as BaseLink** (Установить как основную связь).

Для выбора связываемого звена схвата нажмем кнопку с зеленой стрелкой. Название связанной части (звена) **tool_part1** появится в поле списка справа. Нажмем кнопку **Apply** (Применить). В окне появляется поле **Link Name** (Имя связи) со значением **L1**. В поле **Selected Part** (Выбрать звено) из списка выбираем **tool_part2**, нажмем на кнопку с зеленой стрелкой. Звено **tool_part2** появляется в правом поле.

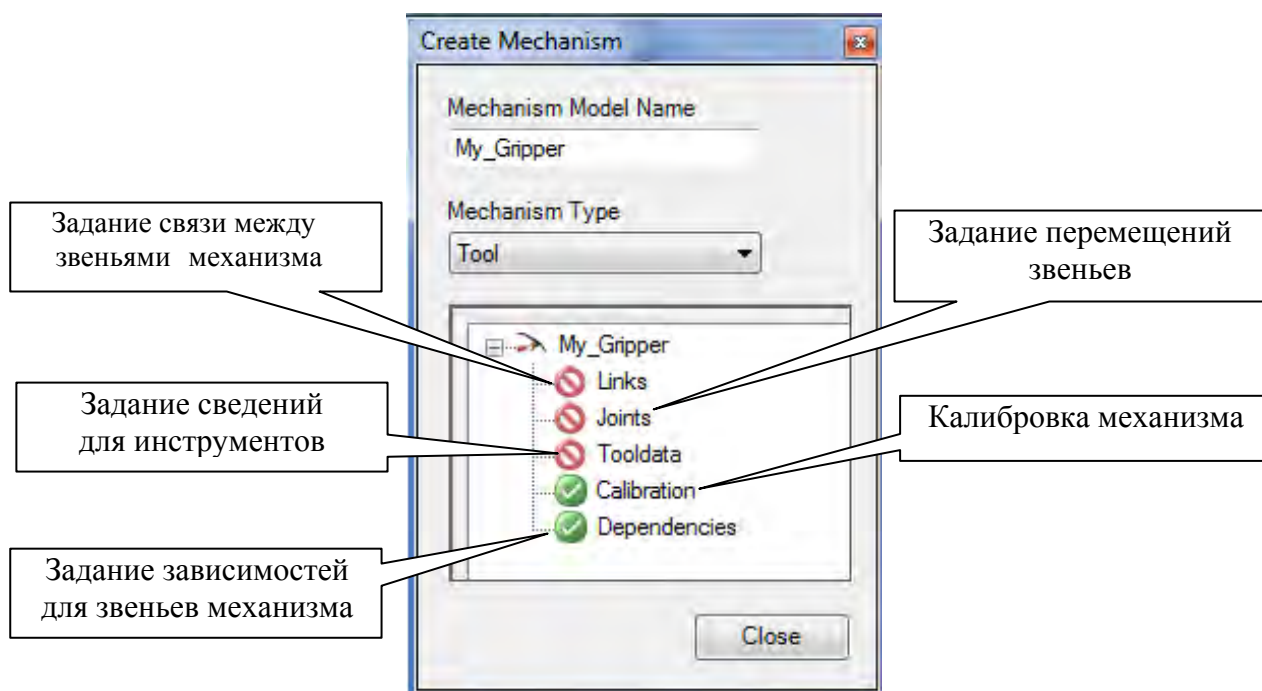


Рисунок 3 – Окно Create Mechanism

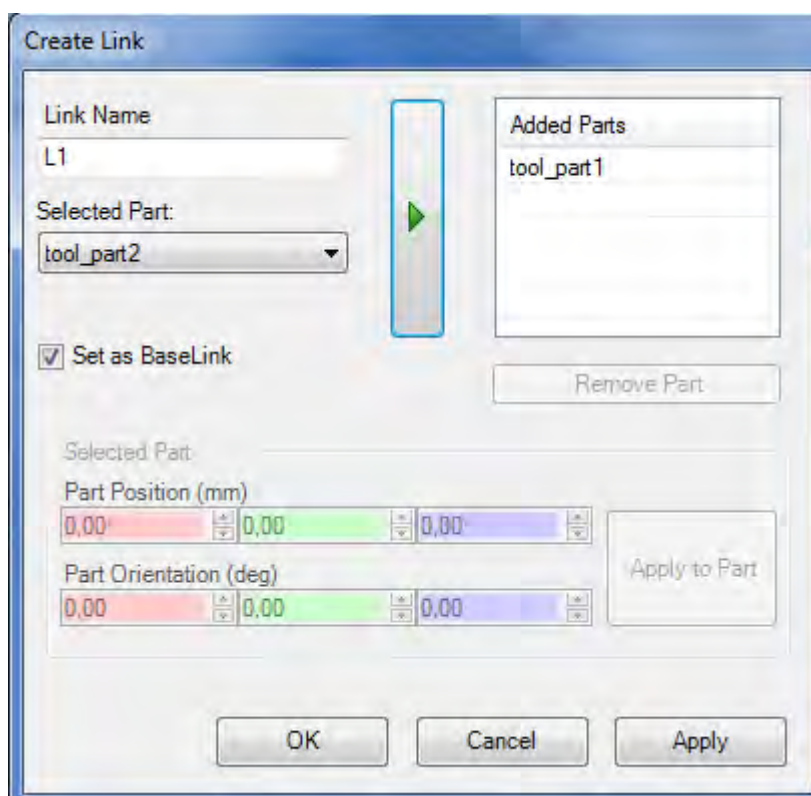


Рисунок 4 – Окно создания связей звеньев схвата Create Link

В результате выполнения этих операций созданы связи между подвижными частями схвата. При необходимости положения звеньев могут быть изменены путем задания значений координат и углов поворота в полях **Part Position** (Положение звена) и **Part Orientation** (Ориентация звена, части). Нажимаем кнопку **OK**.

4.3 Задание перемещений подвижных элементов схвата.

В окне **Create Mechanism** (рисунок 3) выделим узел **Joints** (Звенья) и вызываем его контекстное меню нажатием правой клавиши мыши. В контекстном меню объекта **Joints** щелкнем **Add Joint** (Добавить звено). Появляется новое окно **Create Joint** (Создание звена) (рисунок 5).

Рисунок 5 – Окно Create Joint

Назначение параметров, задаваемых в окне **Create Joint**, приведено ниже.

Joint Name – определяет имя звена схвата. Имя предлагается системой.

Joint Type – определяет тип перемещения звена схвата **Rotational** – **вращательное**, **Prismatic** – **линейное**. Выбор по умолчанию является **Вращательным**. Изменение типа очищает данные, расположенные в полях ниже.

Parent Link – определяет родительскую связь деталей механизма, обычно родителем является первый сустав механизма.

Child Link – определяет связь с дочерней частью механизма.

Active – эта опция делает звено активным. Активное звено то, которое может переместить пользователь, в то время как бездействующее звено является по отношению к нему подчиненным.

Joint Axis – эта группа полей служит для определения начальной и конечной

точек оси, вокруг которой вращается дочернее звено или отрезок, вдоль которого дочернее звено перемещается.

First Position – определяет точку начала вектора оси.

Second Position – определяет конечную точку вектора оси.

Jog Axis – определяет, каким образом расположена дочерняя связь относительно оси.

Limit Type – определяет тип пределов перемещения звеньев в каждом направлении, в котором может двигаться звено. Варианты: **Constant – Постоянные, Variable – Переменные и No – Нет.**

Joint Limits – эта группа полей для задания численных значений пределов перемещения звена.

Min Limit – определяет минимальное перемещение звена.

Max Limit – определяет максимальное перемещение звена.

Зададим значения параметров так, как это показано на рисунке 5.

В окне **Create Mechanism** (см. рисунок 3) выделим узел **Tooldata** (Данные инструмента, в данном случае схвата) и вызовем его контекстное меню нажатием правой клавиши мыши. В контекстном меню объекта **Tooldata** щелкаем **Add Tooldata** (Добавить данные инструмента). Появляется новое окно **Create Tooldata** (Создание звена) (рисунок 6).

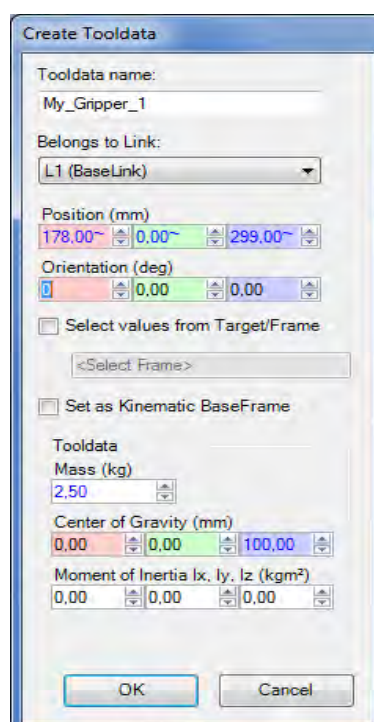


Рисунок 6 – Окно Create ToolData

Назначение параметров, задаваемых в окне **Tooldata**, приведено ниже.

Frame/Tool Data name – определяет название группы данных инструмента.

Belongs to Link – определяет связь, к которой принадлежит группа данных.

Position – определяет новое положение инструмента в пространстве.

Orientation – определяет новую ориентацию инструмента в пространстве.

Select values from target/Frame – флаг устанавливается для определения данных из поля расположенного ниже.

Tool Data – группа данных для задания массоинерционных характеристик схвата или другого инструмента. Эта группа данных видима, если механизм – инструмент.

Mass – определяет массу инструмента.

Center of Gravity – определяет положение центра тяжести инструмента.

Moment of Inertia I_x , I_y , I_z – определяет моменты инерции инструмента относительно осей.

Назначение массоинерционных характеристик инструмента необходимо для выбора скоростей перемещений, при которых не будет создаваться опасных суммарных гравитационных и инерционных нагрузок на механизмы привода отдельных звеньев руки робота.

Введем данные в соответствии с рисунком 6 и нажмем кнопку **ОК**. Окно **Create Mechanism** приобретает вид, показанный на рисунке 7.

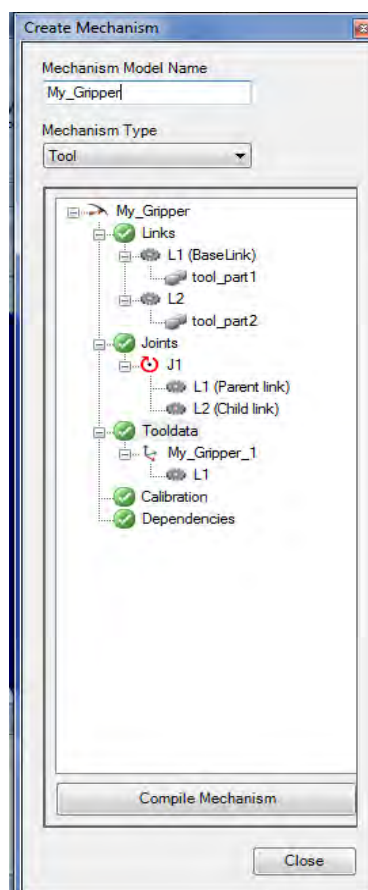


Рисунок 7 – Вид окна Create Mechanism после ввода данных об инструменте

Нажмем кнопку **Compile Mechanism** (Сборка механизма) в нижней части окна **Create Mechanism** (см. рисунок 7).

Под окном **Poses** (Поза, положение) нажмем кнопку **Add**. В появившемся окне **Modifi Pose** (Изменение позы) для задания положения схвата в состоянии

«Закрыт» введем данные, как на рисунке 8. Нажмем кнопку **OK**.

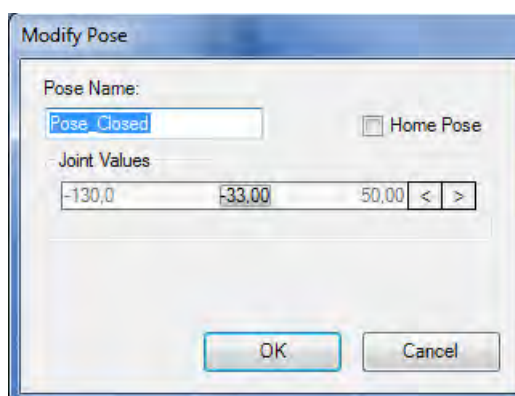


Рисунок 8 – Создание позы схвата в положении «Закрыт»

Задаем еще одно положение схвата для состояния «Открыт» (рисунок 9).

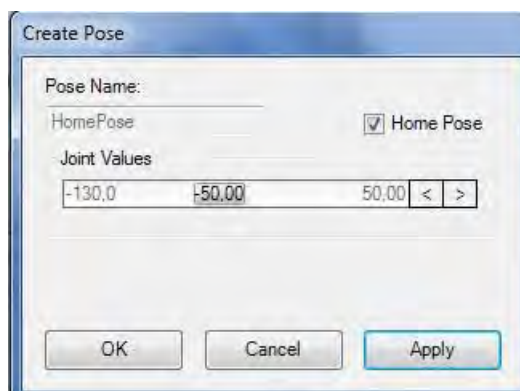


Рисунок 9 – Создание позы схвата в положении «Открыт»

На следующем этапе моделирования работы механизма схвата задается время, которое необходимо на его открытие и закрытие. Нажимаем кнопку **Set Transmission Time** (Задать время перемещения) (рисунок 10).

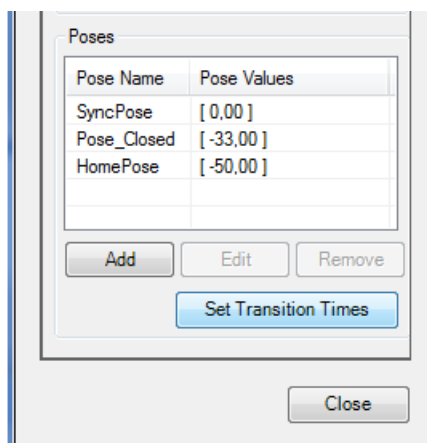


Рисунок 10 – Кнопка Set Transmission Time

В окне **Set Transmission Time** (рисунок 11) отображаются имена ранее определенных положений звеньев механизма. В столбце **To Pose** указываются стартовые положения механизма, в строках **From Pose** – конечные положения. Для каждой комбинации исходного и конечного положения механизма может быть задано время срабатывания механизма (в секундах) при переходе его из одного положения в другое.

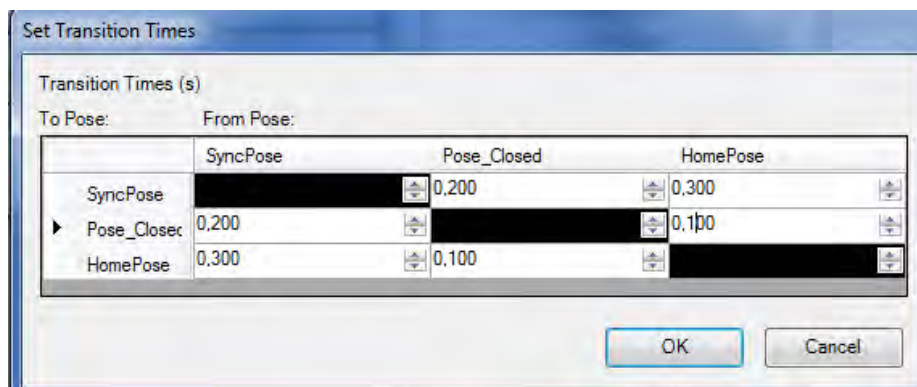


Рисунок 11 – Окно Set Transmission Time

Введем данные, показанные на рисунке 11, и нажмем кнопку **ОК**.

После ввода данных, которые определяют время срабатывания механизма, создание механизма считается завершенным и система выводит сообщение с запросом на подтверждение завершения процедуры моделирования механизма (рисунок 12). Для подтверждения завершения процедуры моделирования нажмем кнопку **Да**.

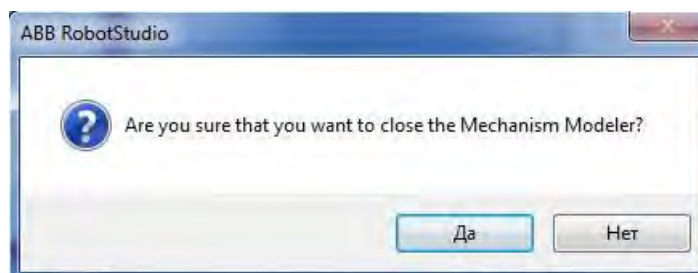


Рисунок 12 – Завершение процедуры моделирования механизма открытия и закрытия схвата

После завершения процедуры моделирования работы механизма схвата на дереве проекта в закладке **Layout** появится ветвь **My_Gripper**.

4.4 Установка схвата на фланец руки робота.

Для точной установки схвата на фланец робота на вкладке **Layout** выберем ветвь **My_Gripper**. Откроем контекстное меню и выберем команду **Attach to** (присоединить к ...). В открывшемся списке нажмем пункт меню **IRB 660_250_315_01**. Появляется окно с запросом к пользователю «Следует ли сохранить текущую позицию инструмента» (рисунок 13).

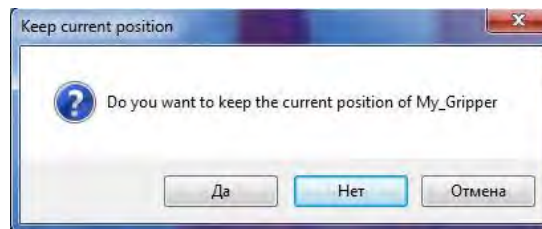


Рисунок 13 – Окно запроса на сохранение прежнего положения объекта

Нажмем кнопку **Нет**. В графическом окне видим, что схват установлен в точку начала системы координат фланца. Вращением, увеличением и перемещением изображения оцениваем точность установки инструмента на фланец.

Индивидуальное задание

Загрузить систему **RobotStudio** и создать РТК на базе шаблона **по заданному варианту**.

Установить силовой блок и контроллер IRC5 в пространстве станции на безопасном расстоянии от робота. Рабочая зона роботов приведена на рисунок 14.

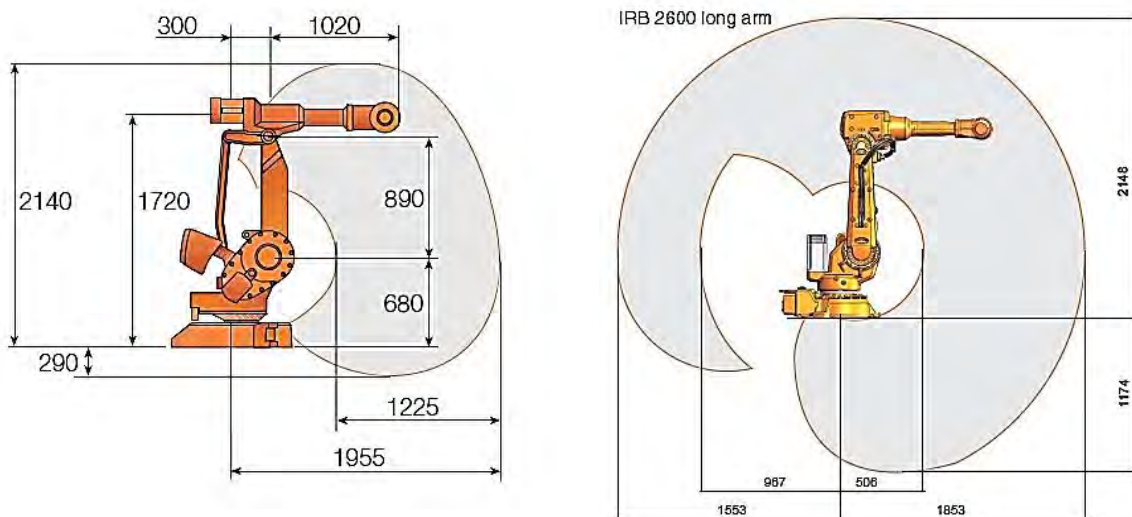


Рисунок 14 – Схема рабочего пространства роботов IRB 4400 и IRB 2600

Из папки **Equipment** выбрать и импортировать конвейер **по заданному варианту**. Командой **Set position** развернуть конвейер на -90 град и перенести на 1000 мм по оси X.

Из папки **Equipment** выбрать и импортировать паллету **Euro Pallet**.

Командой **Set position** развернуть и перенести паллету таким образом, чтобы при перегрузке груза с конвейера на паллету робот совершал минимальные перемещения в пределах своей рабочей зоны и не мешал подъезду погрузчика для забора паллеты. Возможное расположение робота, конвейера, паллеты и направление движения погрузчика показаны на рисунке 15.

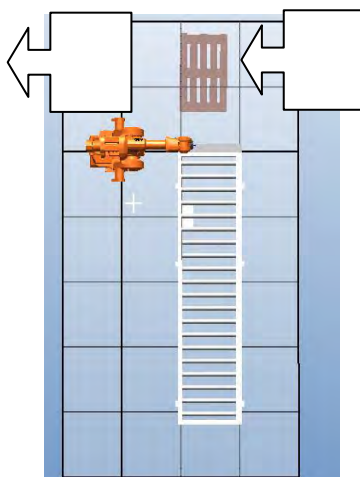


Рисунок 15 – Вариант компоновки РТК

Установить ограждение для создания зоны безопасности робота и конвейера. Если необходимо, изменить положение контроллера и силового блока.

Создать груз с размерами по заданному варианту и установить его на крайней позиции конвейера.

Импортировать детали схвата из библиотеки Geomtry. С учетом размеров груза задать необходимые параметры перемещения звеньев механизма схвата (рисунок 16).

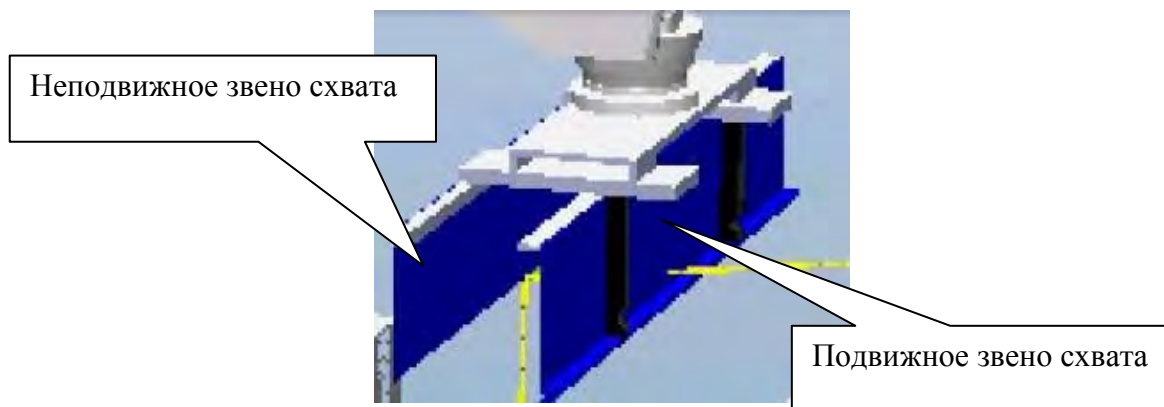


Рисунок 16 – Конструкция схвата для перемещения грузов прямоугольной формы.

Установить схват на фланец руки робота. Оценить получившийся результат и при необходимости выполнить корректировку заданных параметров звеньев механизма схвата.

Таблица 1 – Варианты индивидуальных заданий

Номер варианта	Модель робота	Модель конвейера	Длина груза	Ширина груза	Высота груза
1	IRB 4400	950 4000 h2	150	800	150
2	IRB 2600	1200 4000 h2	250	1200	180

Содержание отчета по лабораторным работам

- 1 Титульный лист.
- 2 Цель работы.
- 3 Протокол построения модели РТК и создания механизма (последовательность действий и команды, используемые при выполнении индивидуального задания).
- 4 Копия экрана (скриншот) геометрической модели РТК.

Лабораторная работа № 3. Проектирование механизмов вращения в робототехнических устройствах. Лабораторная работа № 4. Проектирование механизмов поступательного движения в робототехнических устройствах

Цель лабораторных работ – изучение методики задания траектории и моделирование движения рабочих инструментов робота в среде системы RobotStudio.

1 Техническое обеспечение.

- 1 Персональная ЭВМ.
- 2 Накопитель на жестких магнитных дисках.
- 3 Видеомонитор.

2 Программное обеспечение.

- 1 Операционная система Windows.
- 2 Пакет программ моделирования и программирования роботизированных технологических комплексов RobotStudio.

Общие сведения по заданию путей и моделированию движения рабочих инструментов робота при моделировании и программировании роботизированных технологических комплексов

Для автоматизированного программирования перемещений рабочих инструментов робота (схватов, сварочных головок и клещей, краскопультов и др.) необходимо задать пути их движения. Пути перемещения рабочих инструментов робота могут быть заданы различными способами:

- в режиме обучения робота;
- точным определением опорных точек пути движения путем непосредственного задания их координат в трехмерном пространстве;
- привязкой траектории движения к поверхностям цифровых прототипов изделий.

Задание путей перемещения в режиме обучения робота наиболее простой способ задания пути, но он не обеспечивает высокой точности расположения рабочих инструментов. При выполнении операций складирования грузов, когда не



требуется высокая точность их размещения в пространстве, этот способ вполне приемлем.

Если требуется определить с высокой точностью положение рабочих инструментов, например, при выполнении операций загрузки и выгрузки технологического оборудования, выполнении технологических операций роботом, необходимо точное определение опорных точек траектории движения путем непосредственного задания их координат в трехмерном пространстве.

При выполнении операций дуговой сварки, зачистки поверхностей, заусенцев и т. п. для задания пути перемещения целесообразно использовать различные элементы цифровых прототипов изделий, например, рёбра, поверхности и др. элементы их трехмерных моделей.

3 Основные этапы выполнения работ.

- 1 Создание системы координат WorkObject.
- 2 Создание целевых точек для создания пути инструмента робота.
- 3 Создание пути перемещения груза.
- 4 Оптимизация пути перемещения груза.
- 5 Выполнение индивидуального задания и составление отчета по работе.

4 Методика выполнения лабораторных работ.

4.1 Создание системы координат WorkObject.

Загрузим систему **RobotStudio**, а затем файл модели РТК, который был создан при выполнении первого этапа лабораторной работы № 3 (рисунок 1).

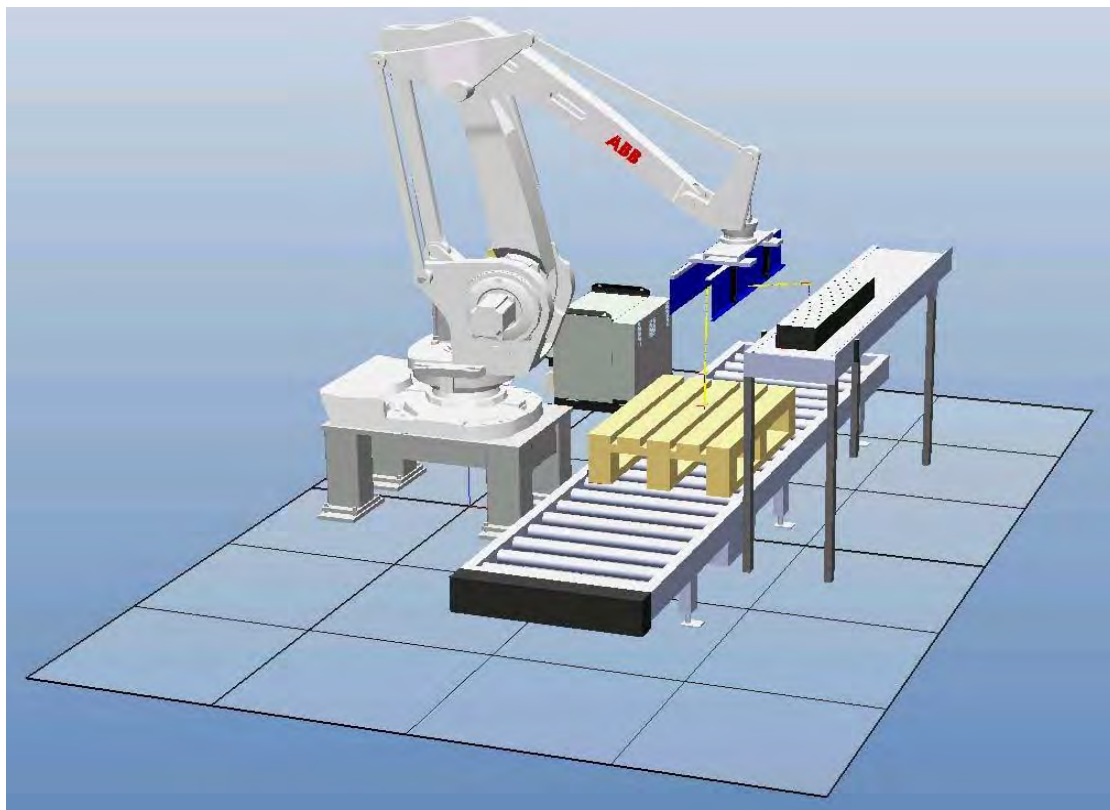


Рисунок 1 – Модель РТК, используемая для программирования перемещений груза

Создадим специальные системы координат, относительно которых будут заданы целевые точки. В RobotStudio такая система координат называется **WorkObject** (Объект работы).

Для создания объекта работы на вкладке **Home** в подразделе **Path Programming** нажмем кнопки **Other** → **Create WorkObject**.

В появившемся окне **Create WorkObject** (Создание объекта работы) в поле **Name** (Имя) присвоим имя **Wobj_box** создаваемому объекту работы (рисунок 2).

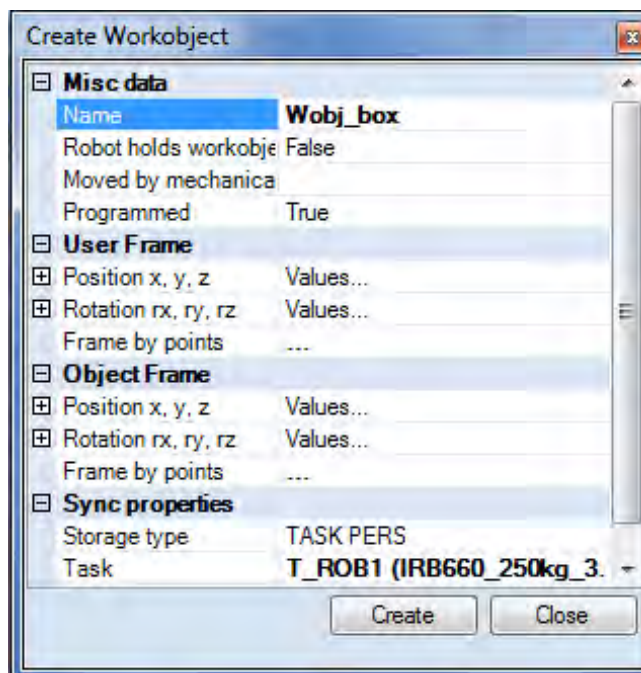


Рисунок 2 – Окно создания объекта работы

Повернем станцию в графическом окне так, чтобы ее конвейер был ориентирован так, как показано на рисунке 3.

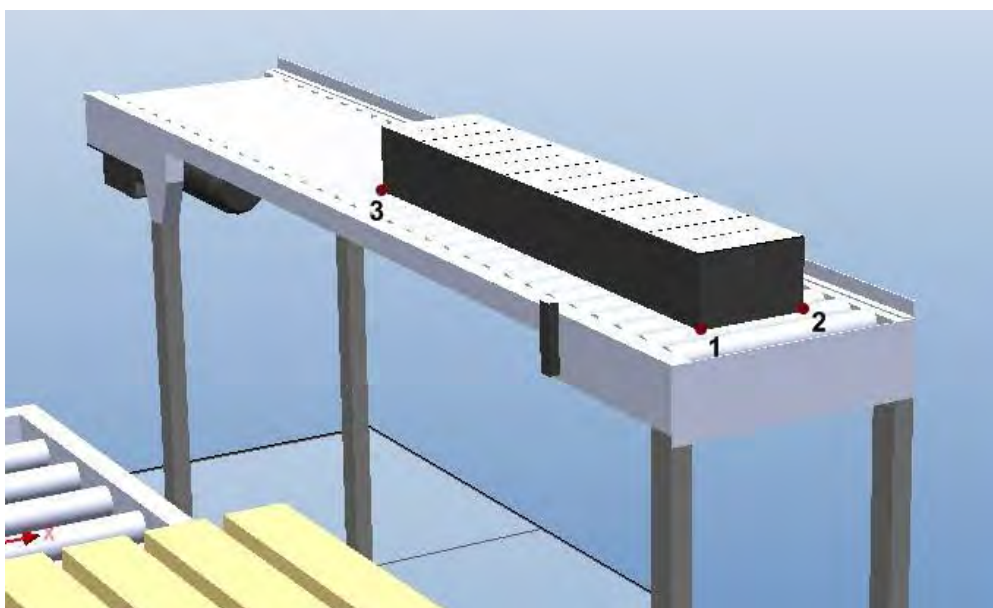



Рисунок 3 – Вид станции для создания Workobject

Для точного определения системы координат объекта работы включим привязку к конечным точкам. Для этого нажмем кнопку **Snap End**  в верхней части графического окна системы.

Далее в разделе **User Frame** (Пользовательская Структура) в окне **Create Workobject** выберем пункт **Frame by points** (Структура из точек) и нажмем на поле справа от него. Появится окно ввода координат точек (рисунок 4).

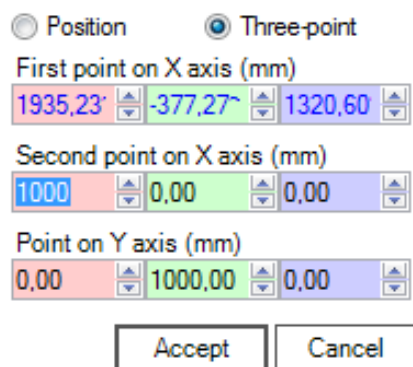


Рисунок 4 – Окно для ввода координат точек при создании Workobject

Выберем опцию **Three-point** (По трем точкам). Затем щелкнем поле ввода координат первой точки, с помощью привязки наведем курсор на точку 1 (см. рисунок 3) и щелкнем по ней. В полях, соответствующих первой точке, появляются значения координат. Аналогично вводим координаты для второй и третьей точек (см. рисунок 3). Нажмем кнопку **Accept** (Принять). В окне **Create WorkObject** нажмем кнопку **Create** (Создать).

Аналогично создадим **WorkObject** с названием **Wobj_pallet** на основе точек, показанных на рисунке 5.

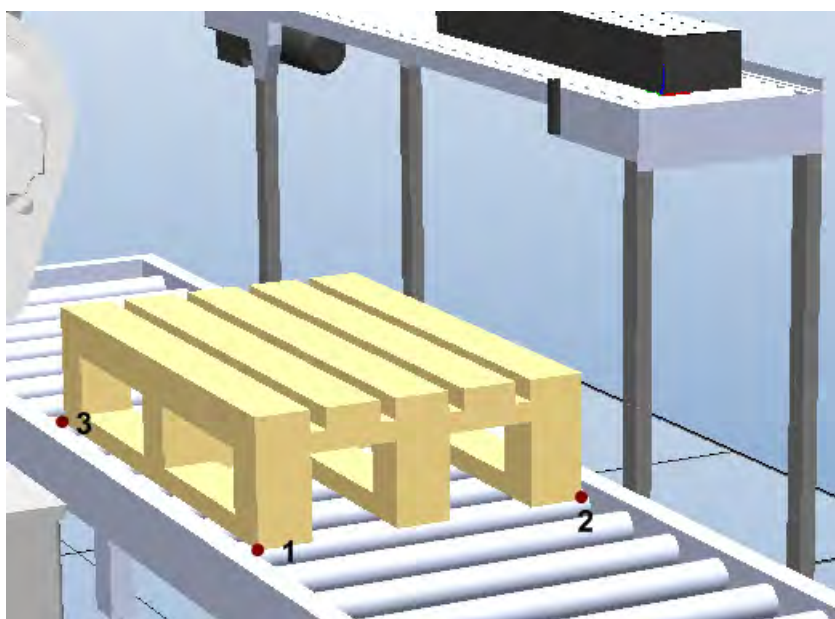


Рисунок 5 – Вид станции для задания Workobject с названием Wobj_pallet

4.2 Создание целевых точек для создания пути инструмента робота.

Для создания пути инструмента необходимо в рабочем пространстве задать точки, которые в дальнейшем будут определять направления перемещения инструмента. На рисунке 6 показан путь и целевые точки перемещения груза с конвейера на паллету. Рассмотрим методику определения целевых точек путем задания их координат X, Y, Z.

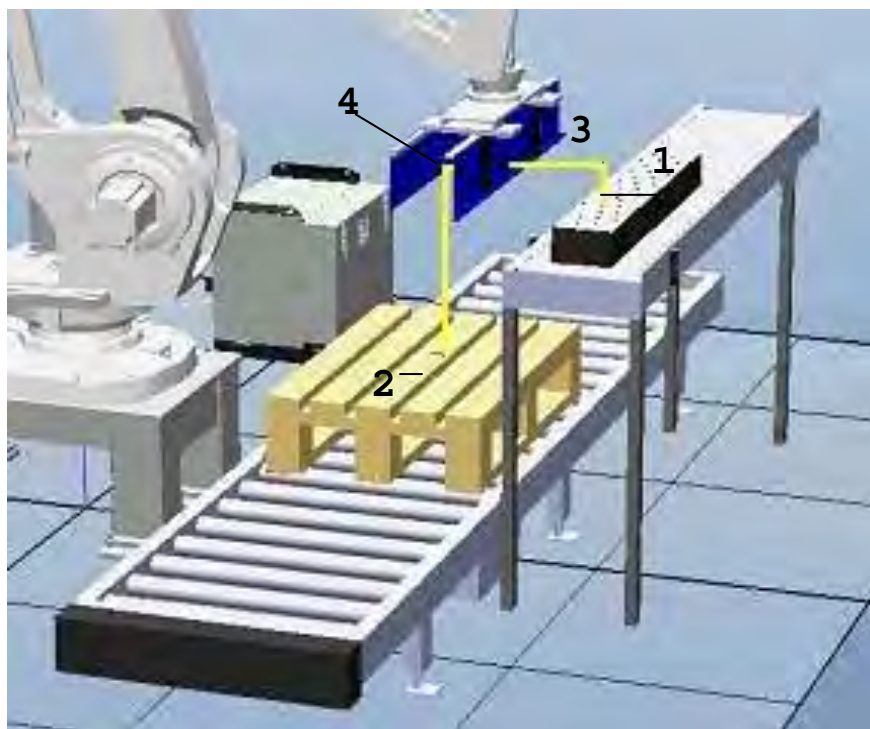


Рисунок 6 – Целевые точки и путь инструмента при перемещении груза на паллету

Для задания точек координатами по осям X, Y, Z откроем вкладку **Home** главного меню системы и в разделе **Path Programming** и нажмем кнопки **Target** → **Create Target** (Цель, целевая точка) (рисунок 7).

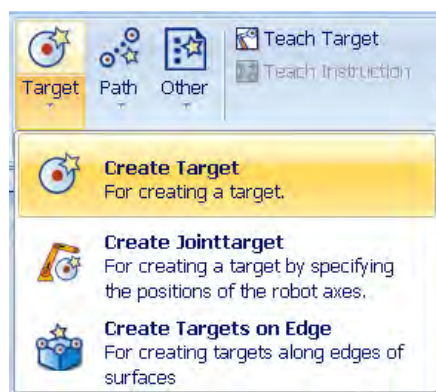


Рисунок 7 – Меню для выбора способа создания целевой точки

Появляется окно ввода данных (рисунок 8). Внизу окна выберем рабочий объект (WorkObject) **Wobj_box**. Заполним поля **Position** значениями координат

целевой точки **1** $X = 1930$, $Y = 210$, $Z = 1330$. Зададим в поле **Orientation** угол ориентации инструмента в этой точке относительно оси Y, равный 180 град. Нажмем кнопку **Add** (Добавить), а затем кнопку **Create** (Создать).

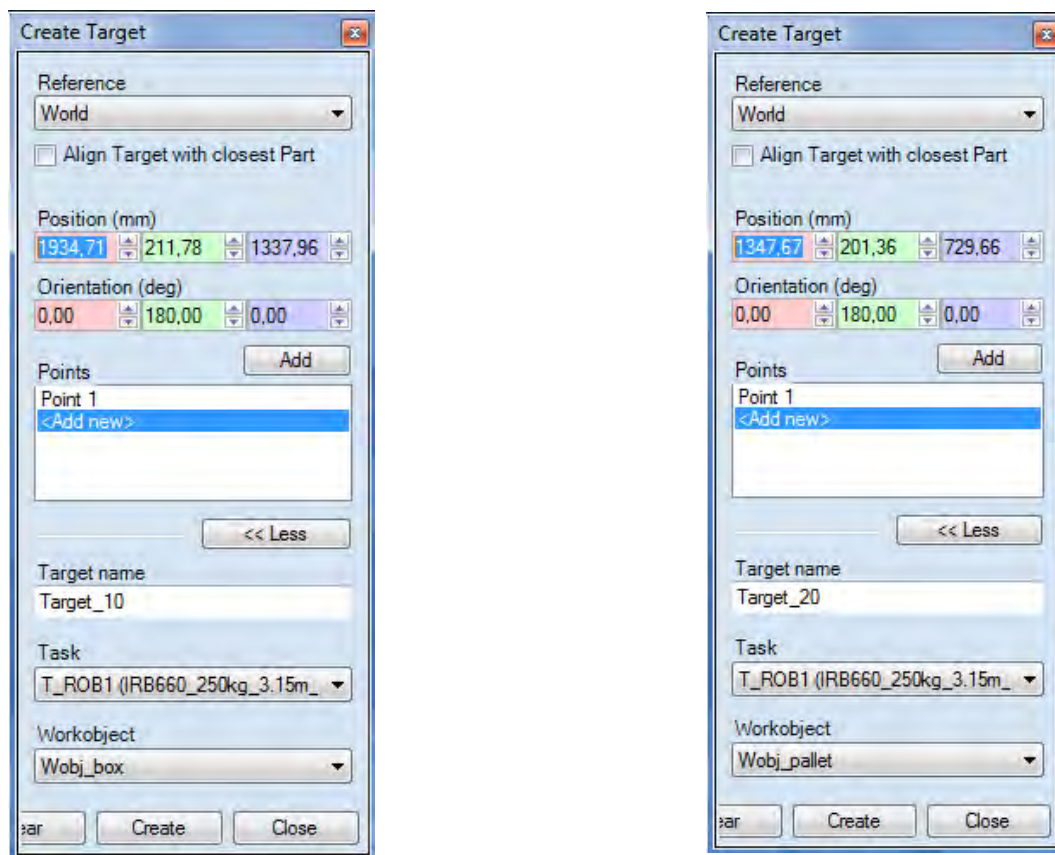


Рисунок 8 – Окна задания координат целевых точек 1 и 2

Создадим целевую точку **2** над рабочим объектом **Wobj_pallet** с координатами $X = 1200$, $Y = 200$, $Z = 730$ и углом ориентации инструмента в этой точке относительно оси Y, равным 180 град. Закроем окно **Create Target** нажатием кнопки **Close**. Откроем меню **Paths&Targets** (Пути и цели) (рисунок 9).

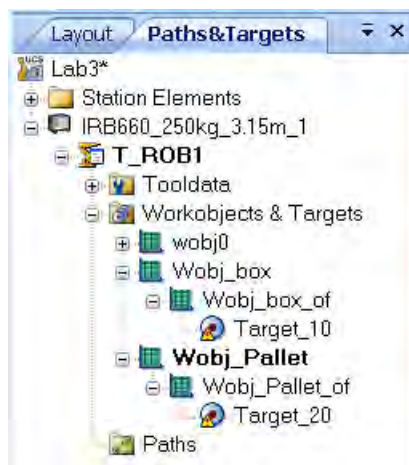


Рисунок 9 – Дерево путей и точек перемещения инструмента

Точку **Target_10**, принадлежащую **Wobj_box**, используя команду контекстного меню **Rename**, переименуем в **take_box** (Взять ящик), а точку **Target_20**, принадлежащую **Wobj_pallet** – в **put_box** (Положить ящик).

Выберем точку **put_box** в меню **Paths&Targets**. В контекстном меню щелкнем **Jump To Target** (Перейти к точке назначения). Появится окно конфигурирования звеньев руки робота (задание углов поворота звеньев) в выбранной точке (рисунок 10). Выбираем **Cfg 1 (0,0,0,0)** и нажимаем кнопку **Apply**. В графическом окне видим, что манипулятор переместился в точку **put_box**. Повторим эти команды для точки **take_box**.

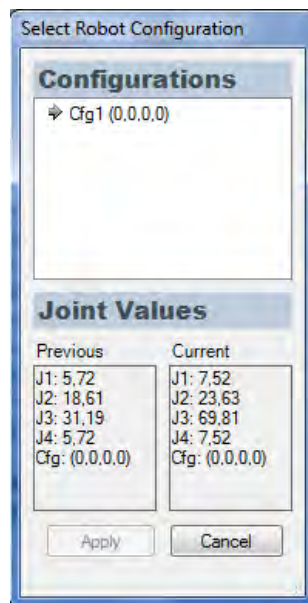


Рисунок 10 – Окно задания конфигурации звеньев руки робота

При необходимости координаты целевых точек могут быть изменены командой **Modify Target** и проверены на корректность задания путем выполнением команды **Jump To Target**.

Создадим точку **3** путем копирования точки **take_box**. Выделим на дереве путей точку **take_box** и из контекстного меню выполним команду **Copy**. Выделим на дереве путей и точек ветвь **Wobj_box_of**, в контекстном меню выберем пункт **Paste** (Вставить). Появится новая точка с названием **take_box_2**. Переименуем ее в **take_box_offset** (Поднять ящик). Выделим точку **take_box_offset** и в контекстном меню выберем пункт **Modify Target** → **Set Position**. В появившемся окне (рисунок 11) введем координаты точки **3** $X = 0$, $Y = 0$, $Z = -200$, которые определяют высоту подъема груза над конвейером, и нажмем кнопку **Apply**. Аналогичным образом создадим точку **4 put_box_offset** (Положить ящик).

Теперь добавим промежуточную точку между позициями **put_box_offset** и **take_box_offset** для того, чтобы при возврате в точку **1** по кратчайшему пути манипулятор не сталкивался с конвейером и грузом. В этой же точке манипулятор будет останавливаться в конце траектории в ожидании поступления на конвейер следующего груза.

На вкладке **Home** в разделе **Settings** поменяем систему координат рабочих

объектов **Workobject** на систему координат робота **wobj0**. Выведем руку робота в нужную точку вручную с помощью команды **Jog Linear** из раздела меню **Freehand**. Затем в разделе меню **Path Programming** (Программирование пути) нажмем кнопку **Target** (Цель) и выберем пункт **Teach Target** (Передать цель). Переименуем на дереве путей и целей полученную точку в **Home_position**.

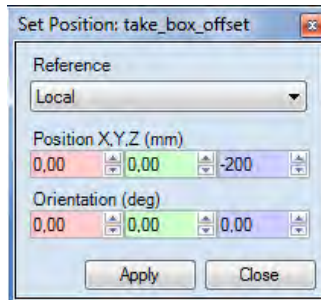


Рисунок 11 – Окно задания координат точки 3

4.3 Создание пути перемещения груза.

Для создания пути инструмента по ранее определенным точкам нажмем в разделе меню **Path Programming** кнопки **Path** → **Empty Path** (Пустой путь). На дереве путей и точек **Paths&Targets** в разделе **Path** появляется пустой путь **Path_10**. Добавим в пустой путь **Path_10** целевые точки. Для этого выделим в меню **Paths&Targets** целевую точку **Home_position**. В контекстном меню выбранной точки нажмем пункт меню **Add to path** → **Path 10** → **<First>**. На дереве путей и точек **Paths&Targets** в разделе **Path_10** появляется участок пути к заданной точке **MoveL Home_position**. Создадим полный путь командой **Add to path** → **Path 10** → **<Last>**, добавляя другие точки в последовательности, показанной на рисунке 12.

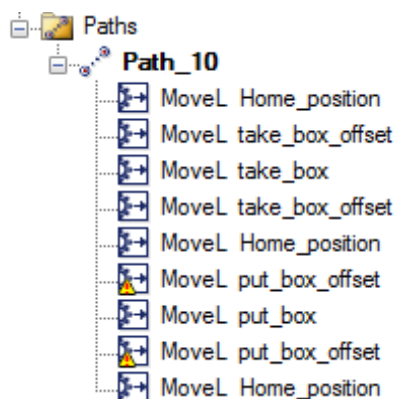


Рисунок 12 – Полный путь инструмента, заданный отрезками прямых MoveL

На дереве проекта выберем путь **Path_10** и из контекстного меню выполним команду **Auto Configuration**.

Для проверки созданного пути инструмента выберем вкладку **Simulation**

(Моделирование), раздел меню **Configure** (Формирование) и нажмем пункт меню **Simulation Setup** (Установки моделирования).

Появляется окно **Setup Simulation** (рисунок 13). В списке **Available Procedures** (Доступные процедуры) выделим путь **Path_10** и нажмем на стрелку влево, а затем кнопки **Apply** и **OK**.

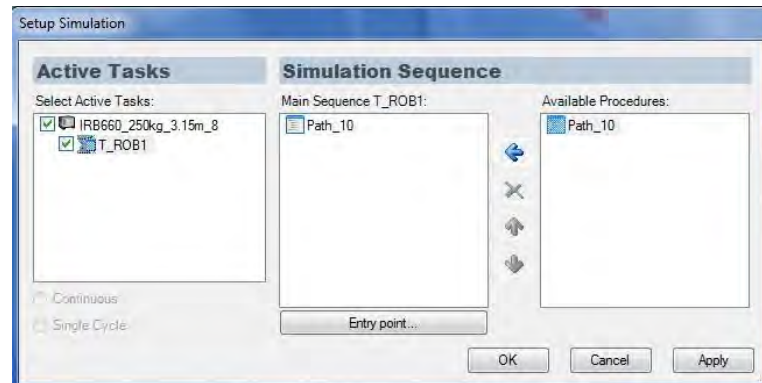



Рисунок 13 – Окно настройки моделирования Setup Simulation

В результате выполнения этих команд будет сформирована программа, управляющая движением робота вдоль заданной траектории.

Для моделирования движения робота в меню **Simulation Control** (Контроль моделирования) нажмем кнопку  **Play**. В графическом окне системы наблюдаем, как робот перемещается по заданному пути.

4.4 Оптимизация пути перемещения груза.

Созданный путь состоит из отрезков, расположенных под прямыми углами. Путь, созданный таким способом, не является оптимальным. Для того чтобы путь был оптимальным, необходимо, чтобы перемещение происходило при минимальных движениях отдельных звеньев руки робота.

Оптимизируем путь перемещения груза путем редактирования управляющей программы, автоматически составленной системой на языке **Rapid**. Язык **Rapid** разработан исследовательским центром робототехники **ABB Group** для программирования роботов собственного производства.

Для открытия автоматически созданной программы выберем вкладку **Offline**. В окне **Offline** на дереве программных модулей выделим ветвь **Module1** (рисунок 14) и в меню **Controller Tools** (Инструментарий контроллера) нажмем кнопку **Rapid Editor** (Текстовый редактор Rapid).

Если в левой части окна закладка не появляется, то необходимо выполнить команды настройки быстрого доступа к меню **Offline Windows Layout** → **Default Layout** или **Windows Lay-out** → **Windows** → **Offline** (см. рисунок 14).

Произведем изменения в тексте программного модуля **PROC Path_10()**. В строке программного модуля **Path_10**, которая определяет начало перемещения

MoveL Home_position,v1000,z100, My_Gripper_1\WObj:=wobj0;

и в подобной строке в конце программного модуля Path_10, которая определяет конец перемещения

MoveL Home_position,v1000,z100,My_Gripper_1\WObj:=wobj0;

заменим команды **MoveL** на **MoveJ**.

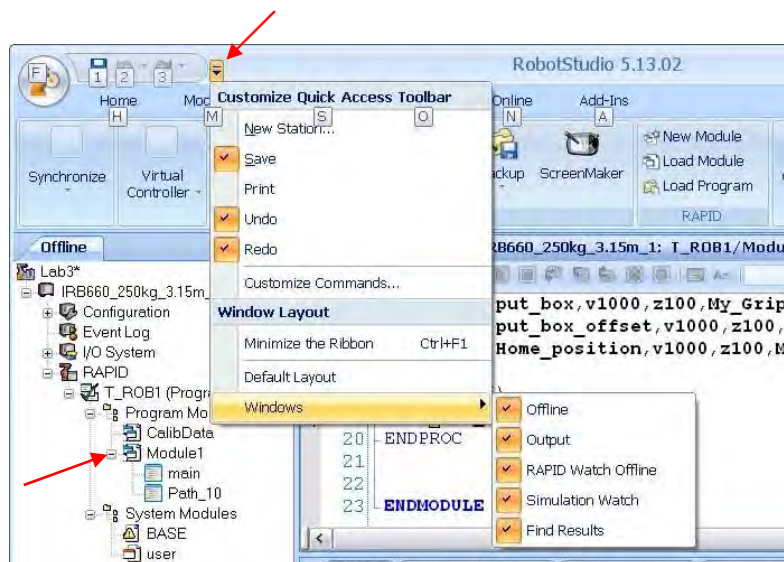


Рисунок 14 – Окно дерева программных модулей Offline

Это означает, что манипулятор будет перемещаться в эти точки, таким образом, чтобы движение звеньев его руки было минимальным.

В строках программного модуля **Path_10**, которые определяют перемещение в точки **take_box**

MoveL take_box,v1000,z100,My_Gripper_1\WObj:=Wobj_box;


и **put_box**

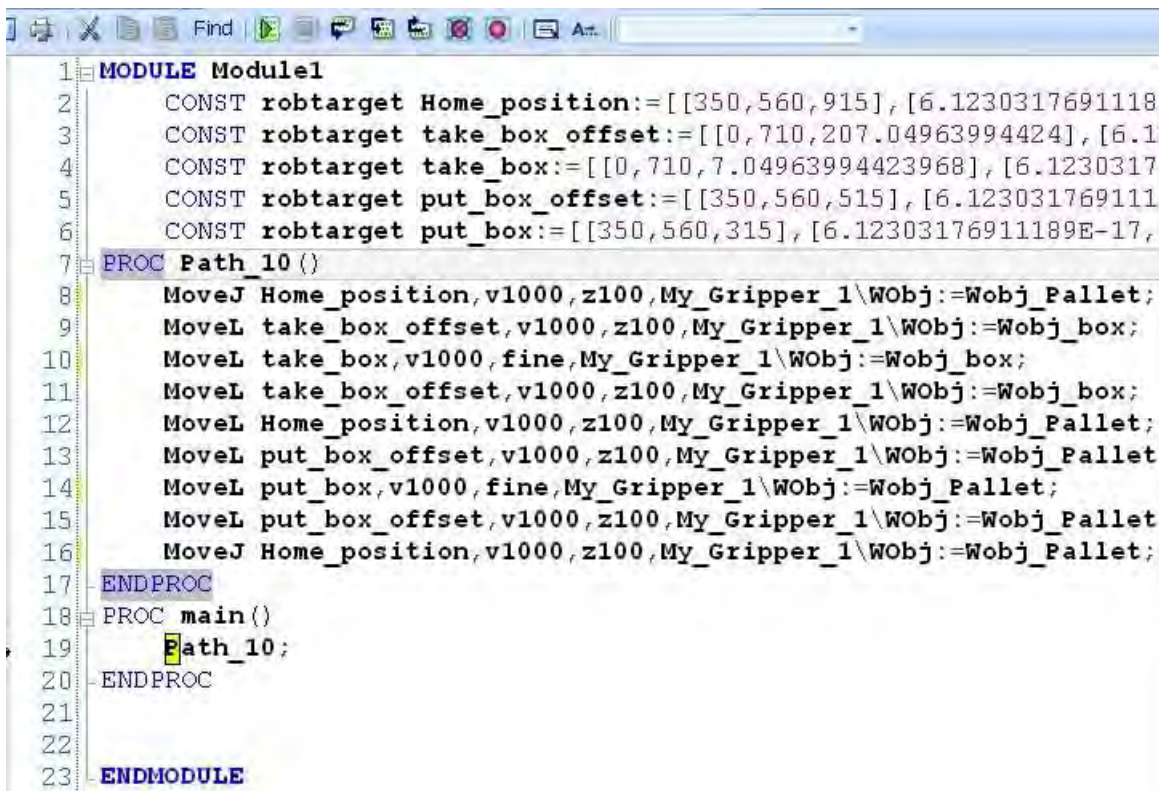
MoveL put_box,v1000,z100,My_Gripper_1\WObj:=Wobj_pallet;

заменим **z100** на **fine** (точно). Это значит, что робот должен переместить груз в эти точки точно.

Строка программы для задания перемещения объекта содержит следующие данные:




Нажмем кнопку  **Apply Changes** (Принять изменения), которая расположена на панели над текстом программы. Внизу в окне Output должно появиться сообщение об отсутствии ошибок. Закроем окно **Rapid Editor**.



```

1  MODULE Module1
2      CONST robtarget Home_position:=[ [350,560,915], [6.1230317691118
3      CONST robtarget take_box_offset:=[ [0,710,207.04963994424], [6.1
4      CONST robtarget take_box:=[ [0,710,7.04963994423968], [6.1230317
5      CONST robtarget put_box_offset:=[ [350,560,515], [6.123031769111
6      CONST robtarget put_box:=[ [350,560,315], [6.12303176911189E-17,
7
8  PROC Path_10 ()
9      MoveJ Home_position,v1000,z100,My_Gripper_1\WObj:=Wobj_Pallet;
10     MoveL take_box_offset,v1000,z100,My_Gripper_1\WObj:=Wobj_box;
11     MoveL take_box,v1000,fine,My_Gripper_1\WObj:=Wobj_box;
12     MoveL take_box_offset,v1000,z100,My_Gripper_1\WObj:=Wobj_box;
13     MoveL Home_position,v1000,z100,My_Gripper_1\WObj:=Wobj_Pallet;
14     MoveL put_box_offset,v1000,z100,My_Gripper_1\WObj:=Wobj_Pallet;
15     MoveL put_box,v1000,fine,My_Gripper_1\WObj:=Wobj_Pallet;
16     MoveL put_box_offset,v1000,z100,My_Gripper_1\WObj:=Wobj_Pallet;
17     MoveJ Home_position,v1000,z100,My_Gripper_1\WObj:=Wobj_Pallet;
18
19  ENDPROC
20
21 PROC main()
22     Path_10;
23
24 ENDPROC
25
26 ENDMODULE
  
```

Рисунок 15 – Окно текстового редактора системы программирования Rapid

Для моделирования движения робота по измененной программе в меню **Simulation Control** (Контроль моделирования) нажмем кнопку  **Play**. В графическом окне системы наблюдаем, как робот перемещается по оптимальному пути.

Индивидуальное задание

Создать путь перемещения груза в РТК.

Выполнить оптимизацию пути путем редактирования программного модуля.

Содержание отчета по лабораторной работе

- 1 Титульный лист.
- 2 Цель работы.
- 3 Протокол построения пути для перемещения груза (последовательность действий и команды, используемые при выполнении индивидуального задания).
- 4 Копия экрана (скриншот) геометрической модели РТК со следом пути перемещения груза.
- 5 Текст программы управления роботом на языке Rapid.



Лабораторная работа № 5. Проектирование конструктивных модулей. Лабораторная работа № 6. Проектирование динамических моделей манипуляторов. Лабораторная работа № 7. Проектирование роботов с рекуператорами энергии

Цель лабораторных работ – изучение методики создания цифровых прототипов (модулей и моделей) основного и вспомогательного технологического оборудования в среде CAD-систем для моделирования и программирования роботизированных технологических процессов.

1 Техническое обеспечение.

- 1 Персональная ЭВМ.
- 2 Накопитель на жестких магнитных дисках.
- 3 Видеомонитор.

2 Программное обеспечение.

- 1 Операционная система Windows.
- 2 Пакет программ трехмерного геометрического моделирования.

Цифровой прототип и технология цифровых прототипов

Цифровой прототип – это цифровой макет изделия, используемый для анализа формы и функциональности изделий до их изготовления. Цифровой прототип становится все более совершенным по мере того, как интегрируются все концептуальные, механические и электрические проектные данные.

Полный цифровой прототип является виртуальным опытным образцом готового изделия и служит для его оптимизации и проверки. Это снижает потребность в изготовлении дорогостоящих физических опытных образцов.

Технология цифровых прототипов дает дизайнерским, проектным и производственным подразделениям возможность на практике изучить изделие, прежде чем оно станет реальностью. С помощью данной технологии производители могут создавать проекты, проверять, оптимизировать их и управлять ими с момента создания концепции до стадии изготовления.

Использование цифровых моделей во время разработки проекта помогает конструкторам обмениваться информацией с заинтересованными сторонами, быстрее выводить продукцию на рынок, делать ее все более совершенной. Практически отпадает необходимость создания дорогостоящих опытных образцов, т. к. все испытания и тесты выполняются в виртуальном режиме. Использование технологии цифровых прототипов позволяет создавать изделия более высокого качества за меньшее время.

В последние годы цифровые прототипы изделий, технологического оборудования используются в системах моделирования и программирования производственных процессов (Delmia, RobotStudio, RoboCim, ROBOCAD др.).

3 Цифровое прототипирование технологического оборудования РТК.

3.1 Цифровое прототипирование металлорежущих станков.

Все наиболее распространенные в практике машиностроительного проектирования CAD-системы (AutoCAD, Inventor, SolidWorks, и др.) обеспечивают создание трехмерных цифровых прототипов изделий и располагают функциональными возможностями их всестороннего анализа. При моделировании роботизированных операций загрузки технологического оборудования его трехмерная модель должна правильно отражать:

- положение и размеры рабочего пространства станка;
- положение и размеры элементов для установки и закрепления заготовок;
- положение узлов станка, которые могут создавать проблемы при установке и снятии заготовки.

В связи с этим трехмерная модель станка может быть изображена без излишних деталей с целью сокращения затрат времени на ее создание.

Рассмотрим пример создания цифрового прототипа токарно-винторезного станка с ЧПУ в среде системы AutoCAD.

Загружаем систему AutoCAD из главного меню Windows и создаем новый чертеж на основе шаблона acadiso.dwt.

Выполним необходимые настройки пространства модели. Командой **Сервис** → **Лимиты** задаем Нижний левый угол – 0,0, Правый верхний угол – 3000, 2000.

Выполним настройки инструментов для повышения точности и производительности черчения. Включаем и настраиваем сетку и шаг курсора 100x100мм, включаем динамический ввод (в нижней строке экрана).

Открываем панели инструментов для быстрого доступа к командам черчения, управления видами и редактирования чертежа (рисунок 1).

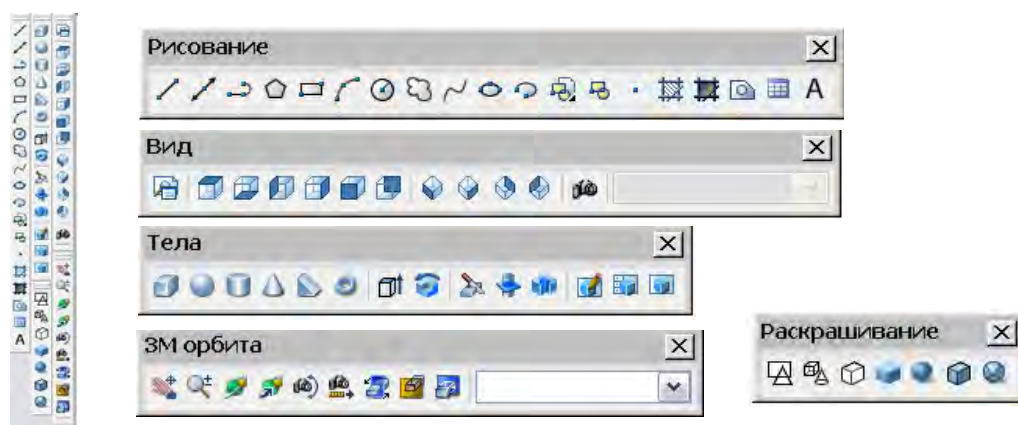


Рисунок 1 – Панели инструментов для быстрого доступа к командам

Командой **Файл** → **Сохранить как** сохраняем файл чертежа в своей рабочей папке под именем **Lathe.dwg** (Токарный станок).

На первом этапе моделирования на основе размеров отдельных узлов станка создаем путем выдавливания контура 3D-модель передней бабки.

Командой **Полилиния**  из панели инструментов **Рисование** (рисунок 4),

начиная из исходной точки в начале координат пространства модели 0,0, строим замкнутый прямоугольный контур по размерам, показанным на рисунке 2.

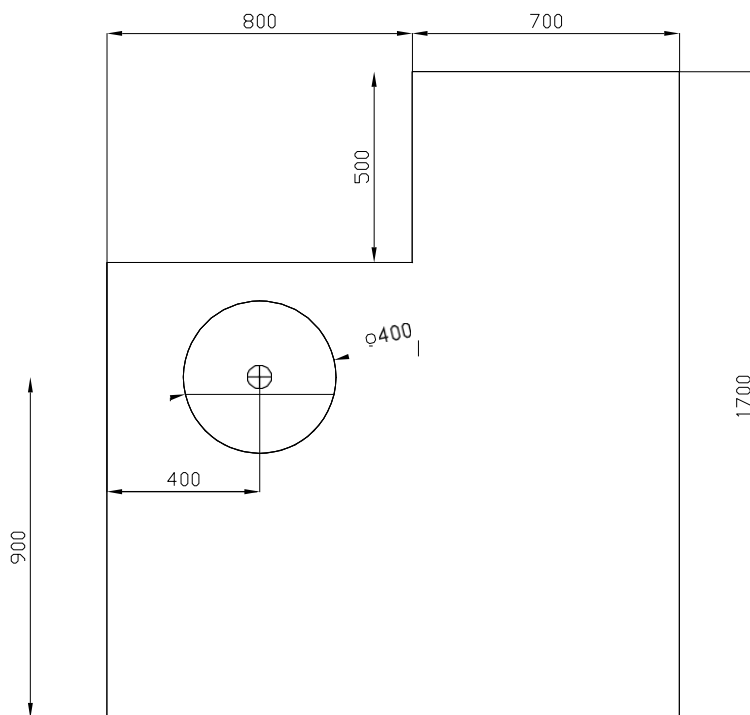







Рисунок 2 – Набор контуров для создания 3D-модель передней бабки станка



Командой **Выдавить**  из панели инструментов **Тела** (см. рисунок 1), выдавливаем контур на высоту 800 мм без задания сужения в направлении выдавливания.

Для создания упрощенной модели шпинделя и поводкового патрона командой **Цилиндр**  строим цилиндр высотой 1000 мм по размерам, показанным на рисунке 2.

Для построения модели жесткого центра, установленного в поводковом патроне, поднимаем плоскость ХОУ в направлении оси Z командой **Уровень** (набираем имя команды в командной строке) на 1000 мм (высота 0), командой **Конус**  строим конус с радиусом основания 30 мм, высотой 52 мм и углом сужения 30 град.

В результате выполнения приведенных выше операций получаем трехмерную модель передней бабки токарно-винторезного станка.

Для оценки формы полученной модели передней бабки станка выполняем раскрашивание модели по **Гуро с кромками**  и вращаем модель в пространстве командой **3М-Орбита** . На рисунке 3 показан результат выполнения этих команд.

Для продолжения моделирования устанавливаем **Вид сверху**  и отменяем раскрашивание командой **2М-каркас** .

Для моделирования наклонной станины станка строим командой **Полилиния** контур, начиная из точки с координатами 600,0 вправо по размерам, которые

показаны на рисунке 4, и выполняем его выдавливание на высоту 2500 мм.

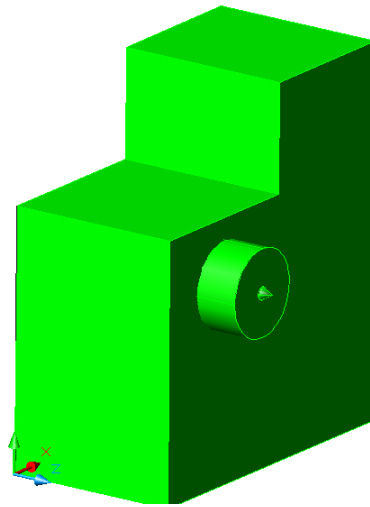


Рисунок 3 – 3D-модель передней бабки станка и поводкового патрона с центром

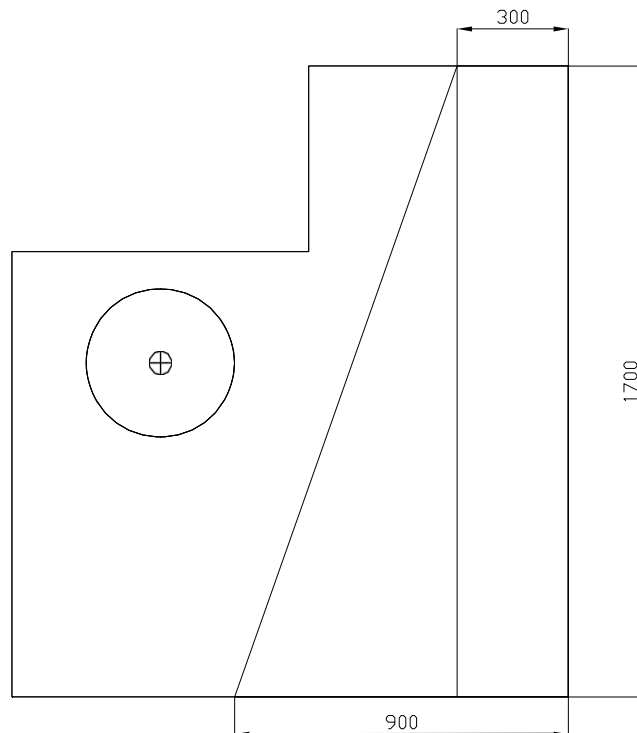


Рисунок 4 – Форма и размеры контура для выдавливания объема станины станка

Для построения модели инструментального магазина станка, поднимаем плоскость XOY в направлении оси Z командой **Уровень** (набираем имя команды в командной строке) на 1400 мм (высота 0). Командой **Цилиндр** строим цилиндр с радиусом основания 300 мм и высотой 100 мм с центром в точке, положение которой показано на рисунке 5.

Поднимаем плоскость XOY в направлении оси Z командой **Уровень** на 1550 мм (высота 0). На свободном месте чертежа командой **Полилиния** строим прямоугольный контур привода револьверной головки и суппорта по

размерам, показанным на рисунке 6. Выполняем выдавливание контура на высоту 300 мм.

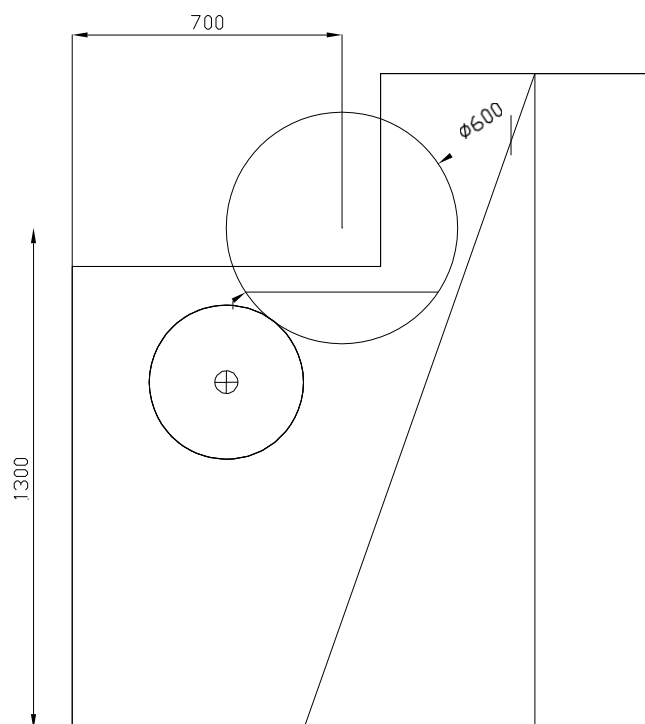


Рисунок 5 – Положение и размеры модели инструментального магазина станка

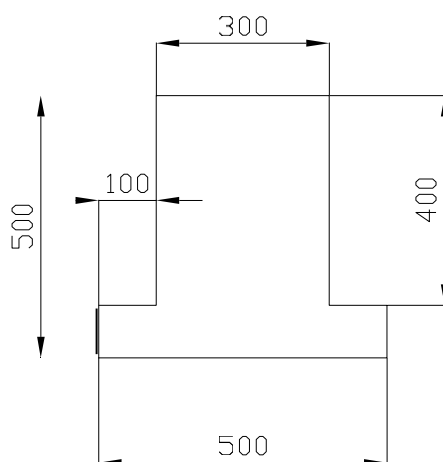


Рисунок 6 – Размеры контура привода револьверной головки и суппорта

Для размещения модели привода револьверной головки и суппорта на плоскости направляющих станины выполняем командой **Повернуть** из раздела меню Редактирование поворот модели относительно оси Z на 71 град относительно нижней левой точки основания. Командой **Перенести** из раздела меню Редактирование переносим модель таким образом, чтобы плоскость основания совпала с плоскостью станины, а привод разместился симметрично оси инструментального магазина.

Выполняя раскрашивание и вращение модели в пространстве, оцениваем результаты моделирования (рисунок 7).

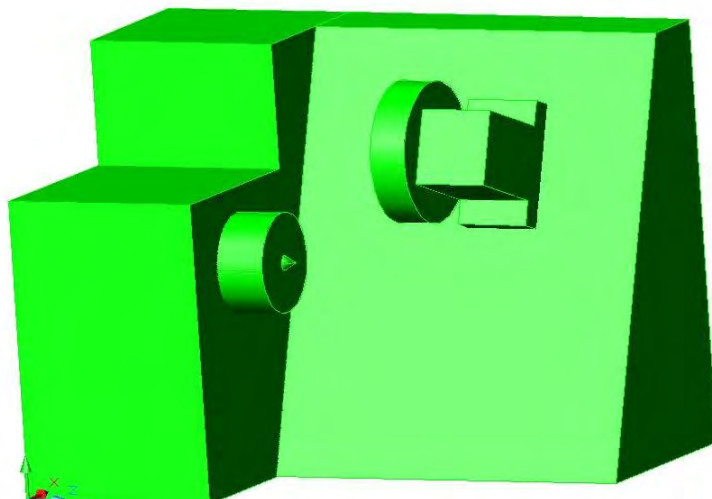


Рисунок 7 – Результаты моделирования привода револьверной головки и суппорта

Для продолжения моделирования устанавливаем **Вид сверху** и отменяем раскрашивание командой **2М-каркас**.

Создадим модель задней бабки, пиноли и вращающегося центра. Поднимаем плоскость XOY в направлении оси Z командой **Уровень** на 1500 мм (высота 0). На свободном месте чертежа командой **Полилиния** строим контур задней бабки по размерам, показанным на рисунке 8. Выполняем выдавливание контура на высоту 500 мм.

Для размещения модели задней бабки на плоскости направляющих станины командой **Повернуть** из раздела меню **Редактирование** выполняем поворот модели вокруг оси Z на 71 град относительно нижней левой точки основания контура задней бабки станка. Командой **Перенести** из раздела меню **Редактирования** переносим модель таким образом, чтобы плоскость основания совпала с наклонной плоскостью станины, а задняя бабка разместились симметрично оси шпинделя станка.

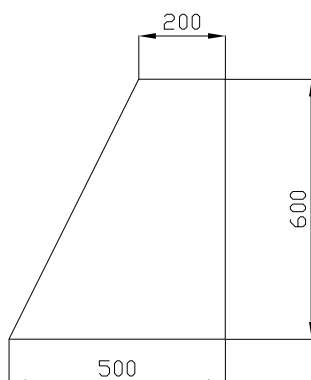


Рисунок 8 – Размеры контура задней бабки станка

Создаем модель пиноли в виде цилиндра радиусом 75 мм и длиной 800 мм, который расположен на оси центров станка. Поднимаем плоскость XOY

в направлении оси *Z* командой **Уровень** на 1400 мм (высота 800) и командой **Цилиндр** строим на оси центров станка цилиндр заданных размеров. Для построения модели вращающегося центра строим еще один цилиндр радиусом 50 мм и длиной 100 мм, предварительно задав уровень плоскости построения по оси *Z* 1300 мм (высота 100 мм). Далее строим модель конуса вращающегося центра. Для построения модели конуса, поднимаем плоскость *XOY* в направлении оси *Z* командой **Уровень** (набираем имя команды в командной строке) на 1300 мм (высота 0), командой **Конус** строим конус с радиусом основания 20 мм, высотой – 34,5 мм и углом сужения 30 град.

Выполняя раскрашивание и вращение модели в пространстве, оцениваем результаты моделирования (рисунок 9).

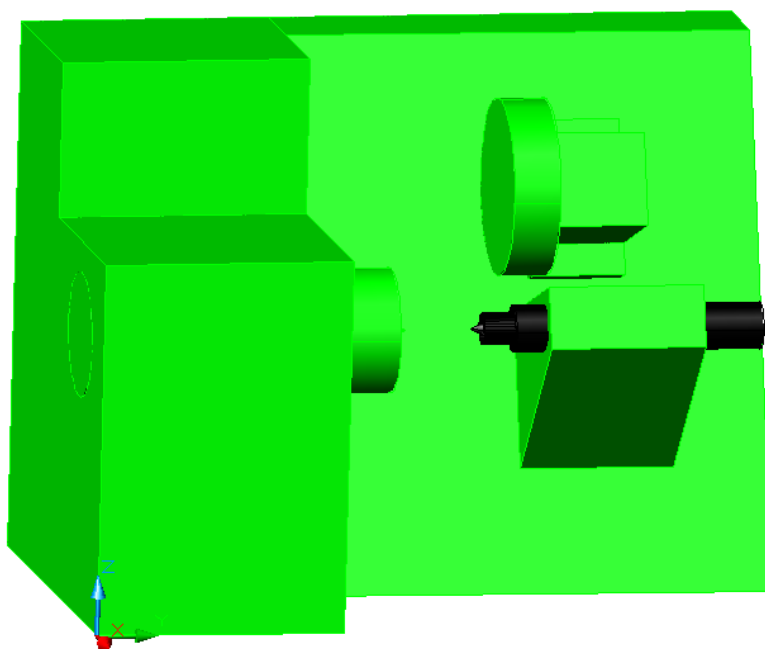


Рисунок 9 – Результаты моделирования задней бабки станка

Для продолжения моделирования устанавливаем **Вид сверху** и отменяем раскрашивание командой **2М-каркас**. Устанавливаем шаг курсора 50 мм и переходим к созданию модели кабины станка.

Командой **Полилиния** строим контур кабины в положении и по размерам, показанным на рисунке 10. Выполняем выдавливание контура на высоту 2500 мм. Для создания окна в кабине, через которое выполняется загрузка станка, устанавливаем шаг курсора 25 мм, поднимаем плоскость *XOY* в направлении оси *Z* командой **Уровень** на 800 мм (высота 0), строим контур (рисунок 11) и выдавливаем его на высоту 1000 мм.

Вырезаем окно в кабине путем вычитания из объема кабины объема окна с помощью команды **Редактирование** → **Редактирование тел** → **Вычитание**. Сначала указываем объем, из которого будет выполняться вычитание, а затем объем, который будет вычитаться.

Выполняя раскрашивание и вращение модели в пространстве, оцениваем результаты моделирования (рисунок 12).

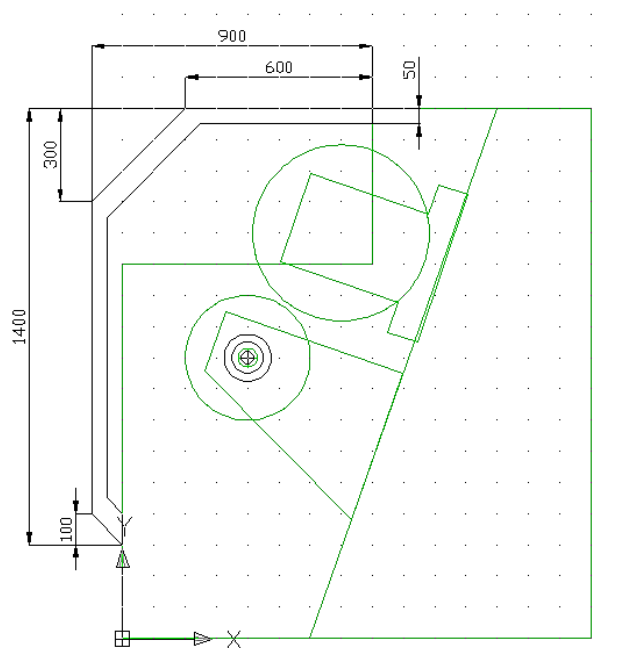


Рисунок 10 – Положение и размеры контура кабины станка

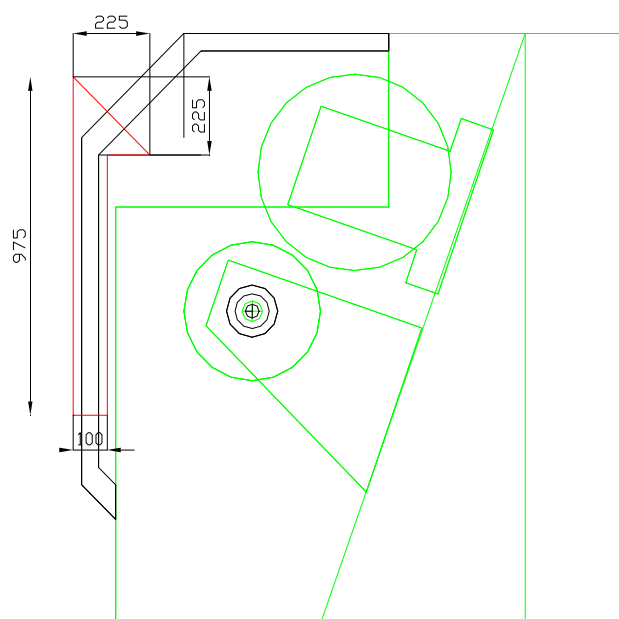


Рисунок 11 – Положение и размеры контура для создания окна кабины станка

Для моделирования сигналов управления контроллера робота в среде RobotStudio создаем модель двери, которая будет открываться и закрываться по командам управления.

Для продолжения моделирования устанавливаем **Вид сверху** и отменяем раскрашивание командой **2М-каркас**.

Поднимаем плоскость XOY в направлении оси Z командой **Уровень** на 700 мм (высота 0). Командой **Полилиния** строим контур кабины в положении и по размерам, показанным на рисунке 13. Выполняем выдавливание контура на высоту 1200 мм.

Выполняя раскрашивание и вращение модели в пространстве, оцениваем

результаты моделирования (рисунок 14). Сохраним модель в файл **Lathe.dwg**.

При моделировании механизмов в среде RobotStudio необходимо, чтобы подвижные и неподвижные части модели хранились в отдельных файлах.

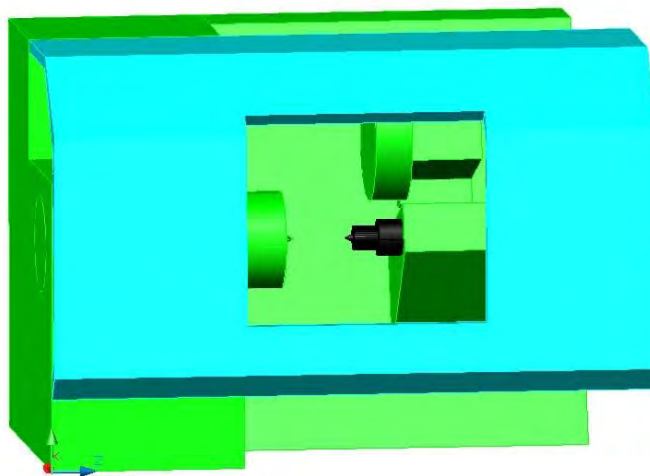


Рисунок 12 – Результаты моделирования кабины станка

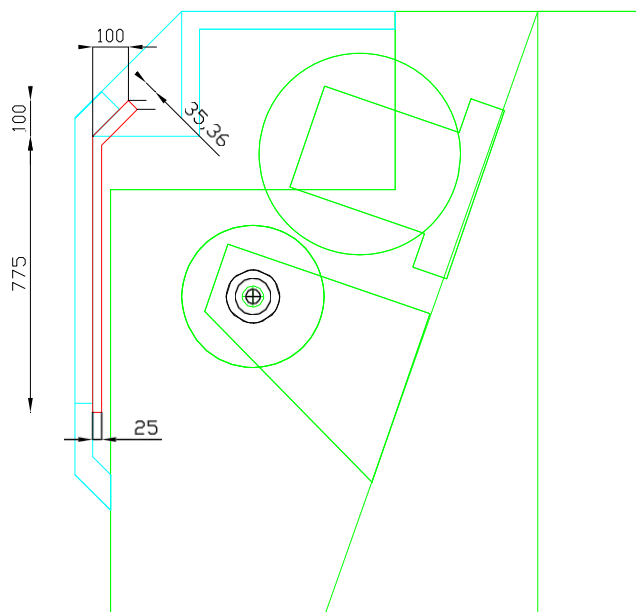



Рисунок 13 – Положение и размеры контура для создания двери окна кабины станка

Создаем новый файл и сохраняем его под именем **Lathe_Bed.dwg**. Командой **Вставка** → **Блок** → **Обзор** вставляем файл **Lathe.dwg** как внешний блок. На запрос **Точка вставки** вводим в командной строке координаты **0,0,0**. Выполняем команду **Редактирование** → **Расчленить** для разделения блока на отдельные элементы модели. На загруженной модели станка удаляем модель двери и пинноли с вращающимся центром. Для импорта модели станины в среду системы RobotStudio создаем файл экспорта командой **Файл** → **Экспорт**. Указываем

имя файла **Lathe_Bed** и задаем тип файла ***.sat**. Командой **Отменить**  восстанавливаем все элементы модели станка, а затем удаляем все элементы модели, кроме модели пиноли с вращающимся центром (рисунок 15).

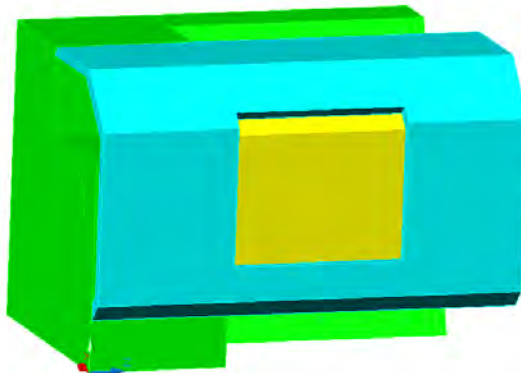


Рисунок 14 – Результаты моделирования двери кабины станка



Рисунок 15 – Модели пиноли и вращающегося центра

Сохраняем командой **Файл** → **Экспорт** модели пиноли и вращающегося центра в файл под именем **Pinole**, тип файла ***.sat**.


Командой **Отменить** восстанавливаем все элементы модели станка, а затем удаляем все элементы модели, кроме модели двери (рисунок 16).

Сохраняем модель двери командой **Файл** → **Экспорт** в файл под именем **Door**, тип файла ***.sat**.



Рисунок 16 – Модель двери кабины станка

3.2 Цифровое прототипирование обрабатываемой заготовки.

В рабочей папке создаем файл **Rolling.dwg**. Для создания цифрового прототипа заготовки из проката с центром в точке начала координат командой **Цилиндр**  создаем цилиндр радиусом 30 мм и высотой 200 мм (рисунок 17).

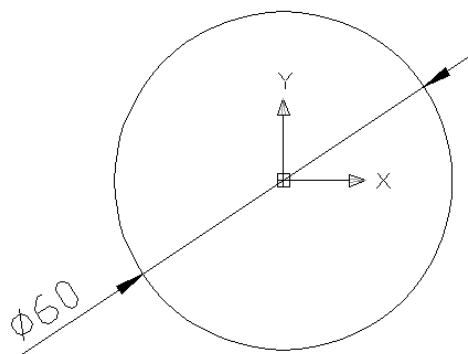



Рисунок 17 – К построению модели заготовки

Для создания на модели центровых отверстий устанавливаем шаг курсора и сетки 2 мм. Из точки начала координат строим командой **Полилиния**  контур поперечного сечения центрового отверстия по размерам, показанным на рисунке 18.

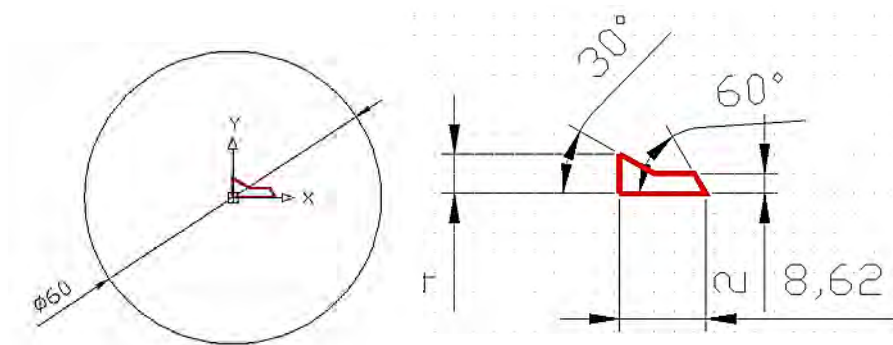



Рисунок 18 – К построению модели центровых отверстий

Командой **Вращать**  вращаем контур вокруг оси X на 360 град, а затем командой **Редактирование** → **3М операции** → **3М поворот** поворачиваем модель вокруг оси Y на -90 град (рисунок 19).

Для создания второго центрового отверстия командой **Редактирование** → **Копировать** копируем модель центрового отверстия. Для переноса второго центрового отверстия на противоположный торец заготовки выполняем команды **Редактирование** → **Перенести** и вводим координаты базовой точки **0,0,0** и второй точки **0,0,200**. Поворачиваем перенесенную копию центрового отверстия командой **Редактирование** → **3М операции** → **3М поворот** вокруг оси Y на 180 град.

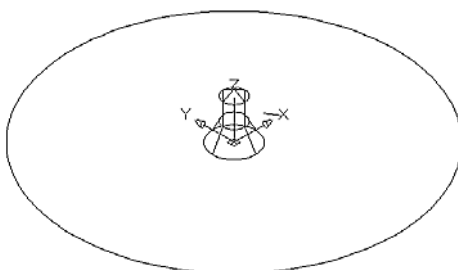


Рисунок 19 – Модель центрового отверстия

Для создания отверстий выполняем вычитание из объема цилиндра объемов двух центровых отверстий с помощью команды **Редактирование** → **Редактирование тел** → **Вычитание**. Сначала указываем объем, из которого будет выполняться вычитание, а затем объемы двух вычитаемых центровых отверстий.

Выполняя раскрашивание и вращение модели в пространстве, оцениваем результаты моделирования (рисунок 20).

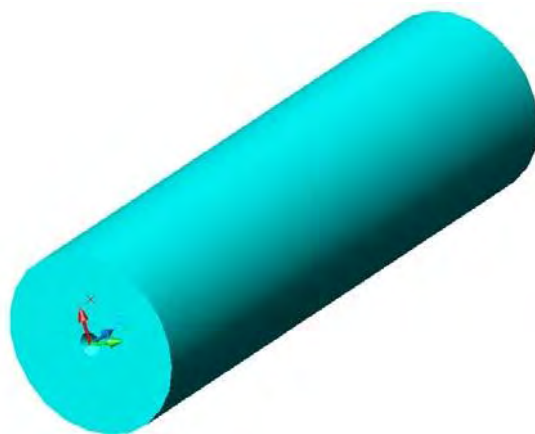


Рисунок 20 – Модель заготовки с центровыми отверстиями

Сохраняем в своей рабочей папке файл *Заготовка.dwg*, а затем командой **Файл** → **Экспорт** создаем файл экспорта под именем **Rolling.sat**.

3.3 Цифровое прототипирование кассеты для заготовки.

Создаем цифровой прототип кассеты для размещения заготовки на шаговом конвейере. Общий вид кассеты приведен на рисунке 21.

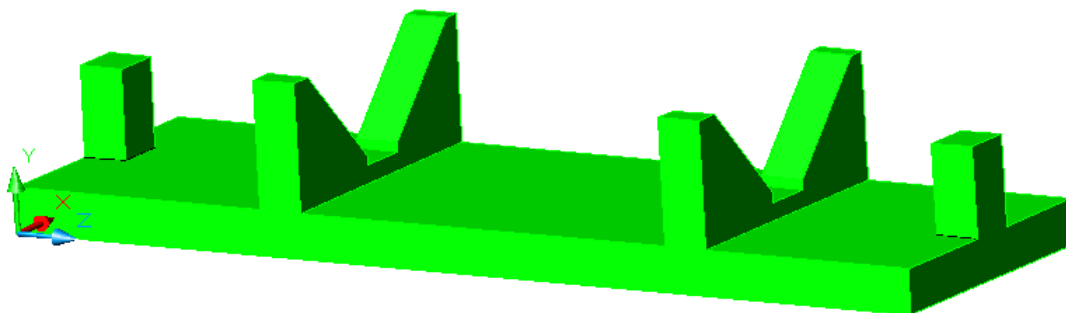


Рисунок 21 – Кассета для размещения заготовки на шаговом конвейере

В рабочей папке создаем файл **Cartridge.dwg**. Из точки начала координат строим командой **Полилиния** контур поперечного сечения основания кассеты по размерам, показанным на рисунке 22, и выдавливаем основание на высоту 220 мм.

Командой **Полилиния** строим контур поперечного сечения упора кассеты по размерам, показанным на рисунке 23, и выдавливаем его на высоту 10 мм.

Командой **Полилиния** строим контур поперечного сечения призмы кассеты по размерам, показанным на рисунке 24, и выдавливаем его на высоту 10 мм.

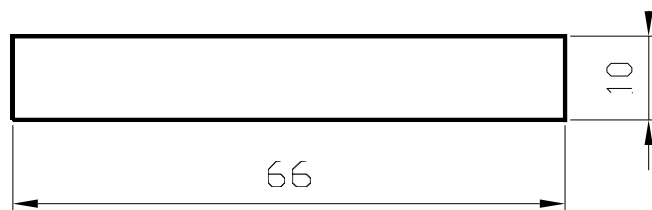


Рисунок 22 – Контур поперечного сечения основания кассеты

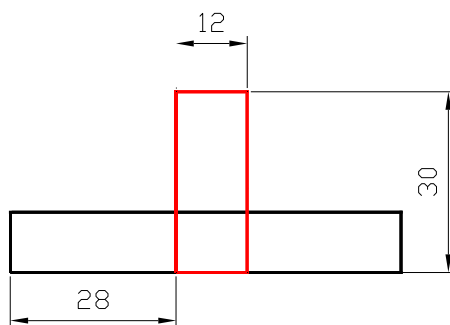


Рисунок 23 – Контур поперечного сечения упора кассеты

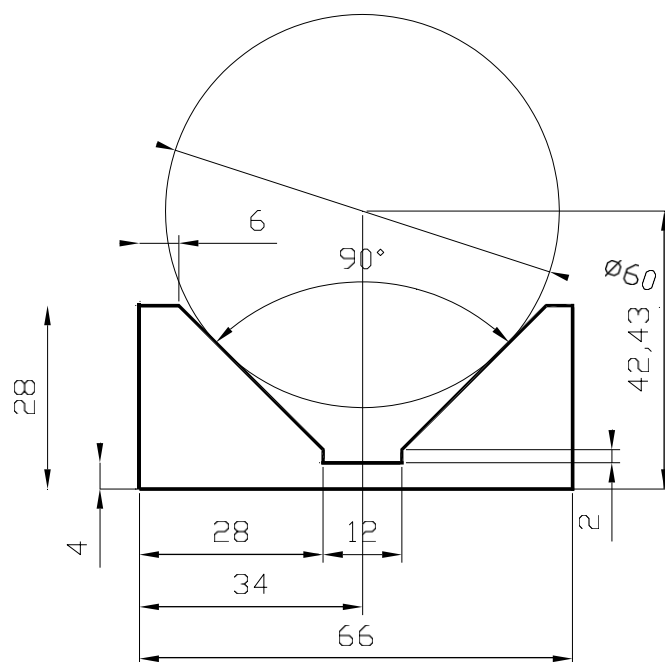


Рисунок 24 – Контур поперечного сечения призмы кассеты

Переносим призму в положительном направлении по оси Z на 60 мм командой **Редактирование** → **Перенести** и вводим координаты базовой точки **0,0,0** и второй точки **0,0,60**.

Копируем призму с одновременным переносом в положительном направлении по оси Z на 150 мм командой **Редактирование** → **Копировать**, вводим координаты базовой точки **0,0,0** и второй точки **0,0,150** и нажимаем клавишу **Enter**

для завершения команды **Копировать**.

Аналогично копируем упор с одновременным переносом в положительном направлении по оси Z на 210 мм.

Выполняя раскрашивание и вращение модели в пространстве, оцениваем результаты моделирования (рисунок 25).

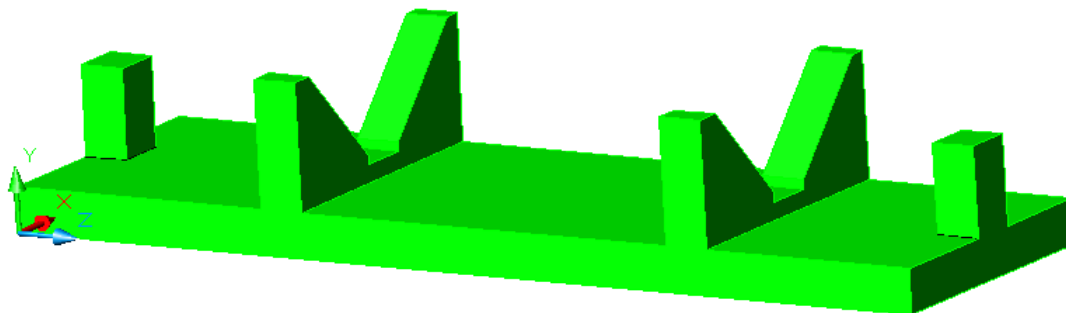


Рисунок 25 – Модель кассеты для размещения заготовки на шаговом конвейере

Сохраняем в своей рабочей папке файл **Cartridge.dwg**, а затем командой **Файл → Экспорт** создаем файл экспорта под именем **Cartridge.sat**.

3.3 Цифровое прототипирование схвата.

В рабочей папке создаем файл **Gripper.dwg**. Из произвольной точки командой Полилиния строим контур поперечного сечения левого призматического рычага схвата по размерам, показанным на рисунке 26.

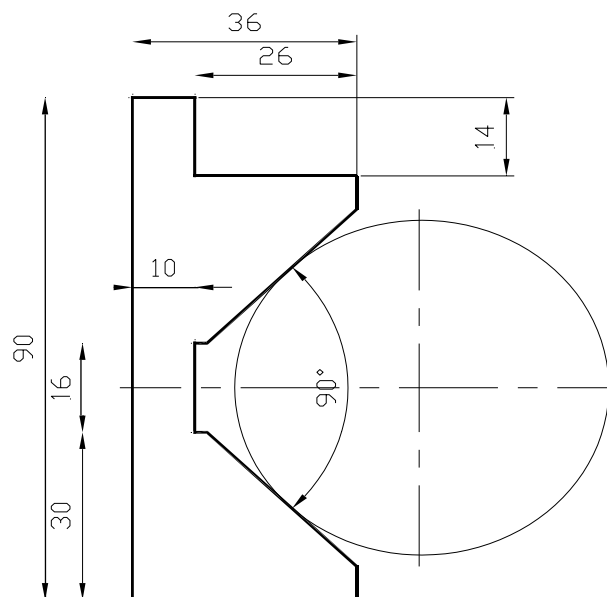


Рисунок 26 – Контур поперечного сечения левого призматического рычага схвата

Правый рычаг схвата создаем как зеркальное отображение левого. Предварительно определяем положение оси симметрии путем построения окружности диаметром 60 мм, касательной к двум отрезкам, с использованием объектной

привязки (см. рисунок 26). Командой **Редактирование** → **Зеркало** выполняем построение зеркального отображения левого рычага схвата (рисунок 27).

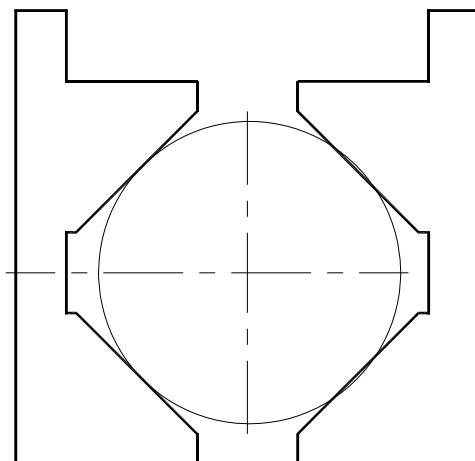


Рисунок 27 – Построение зеркального отображения левого рычага схвата

Командой **Полилиния** на свободном месте пространства модели строим контур корпуса схвата по размерам, показанным на рисунке 28.

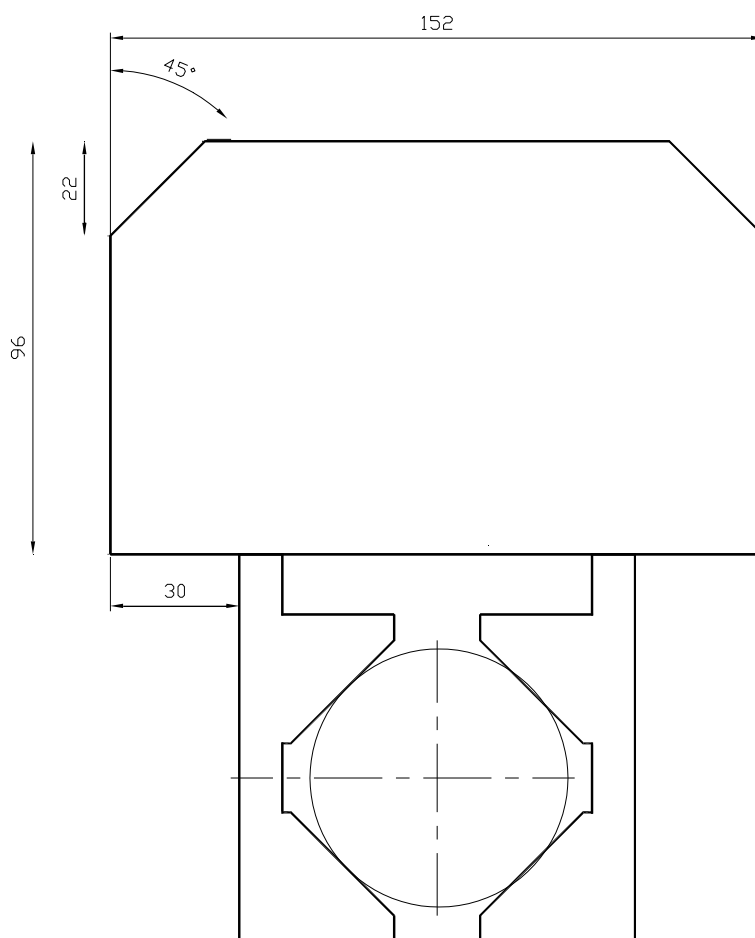


Рисунок 28 – Контур корпуса схвата

Последовательно выдавливаем корпус схвата на высоту 60 мм и затем два рычага на высоту 40 мм.

Переносим два рычага в положительном направлении по оси Z на 10 мм командой **Редактирование** → **Перенести** и вводим координаты базовой точки **0,0,0** и второй точки **0,0,10**.

Командой **Полилиния** строим половину контура поперечного сечения фланца схвата по размерам, показанным на рисунке 29. Начало и конец контура должны лежать на оси симметрии корпуса схвата.

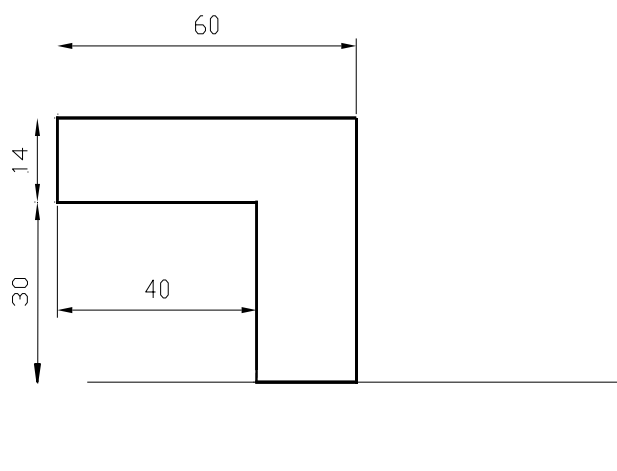


Рисунок 29 – Контур для построения фланца корпуса схвата

Командой **Вращать** создаем трехмерную модель фланца. Переносим фланец в положительном направлении по оси Z на 30 мм командой **Редактирование** → **Перенести** и вводим координаты базовой точки **0,0,0** и второй точки **0,0,30**.

Объединяем фланец и корпус в одно целое командой **Редактирование** → **Редактирование тел** → **Объединение**.

Выполняя раскрашивание и вращение модели в пространстве, оцениваем результаты моделирования (рисунок 30).

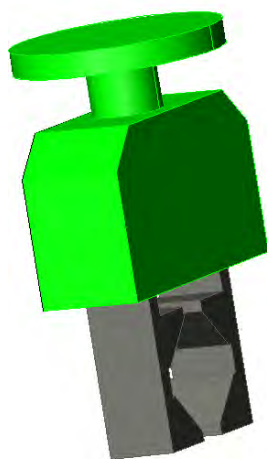


Рисунок 30 – Результат моделирования схвата

Для того чтобы присоединение схвата к фланцу робота происходило автоматически, перенесем схват в начало координат и расположим его в направлении оси Z путем поворота вокруг оси X (рисунок 31).

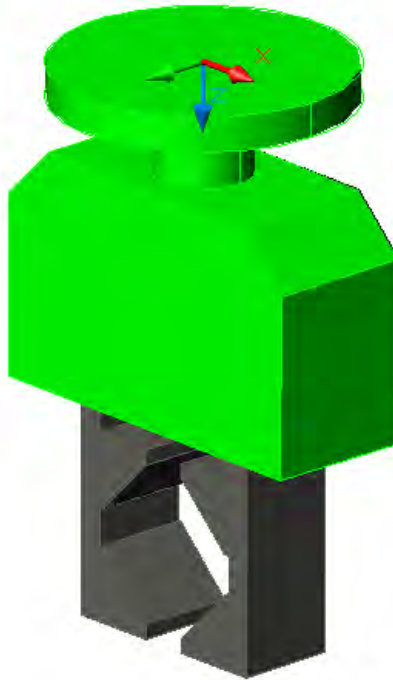



Рисунок 31 – Результат переноса начала координат схвата

Переносим схват в начало координат командой **Редактирование** → **Перенести**. Выбираем объекты для переноса командой **Все**. Для точного определения координат базовой точки в центре фланца включаем объектную привязку **Центр** и выбираем центр окружности на верхней плоскости фланца, координаты второй точки задаем значением **0,0,0**.

Располагаем схват в направлении оси Z командой **Редактирование** → **3D операции** → **3D поворот**. Выбираем объекты для поворота командой **Все**, координаты базовой точки задаем значением **0,0,0**, указываем ось поворота **X** и угол поворота **–90** град (см. рисунок 31).

Выполняя раскрашивание и вращение модели в пространстве, оцениваем результаты моделирования (см. рисунок 31). Сохраняем в своей рабочей папке файл **Gripper.dwg**.

Для последующего моделирования работы механизма схвата сохраняем подвижные и неподвижные элементы схвата в разных файлах экспорта.

Удаляем призматические рычаги схвата и командой **Файл** → **Экспорт** создаем файл экспорта под именем **Mounting.sat**. Командой **Отменить**  восстанавливаем все ранее удаленные элементы схвата, а затем удаляем корпус и правый рычаг схвата. Создаем файл экспорта под именем **Lever_L.sat**. Восстанавливаем командой **Отменить** видимость правого схвата. Удаляем левый рычаг схвата и создаем файл экспорта под именем **Lever_R.sat**.

3.4 Создание механизмов «Дверь» и «Пиноль».

Создаем пустую станцию командой **Empty Station**. Командой **Import Geometry** (меню Home) загружаем файл **Lathe_Bed.sat**, а затем файлы **Door.sat**, **Pinole.sat** из своей рабочей папки. Командой **Set Position** последовательно поворачиваем станину, дверь и пиноль на 90 град вокруг оси X и 180 град вокруг оси Z.

Из меню **Modeling** выполняем команду **Create Mechanism** (создать механизм). В окне Create Mechanism (рисунок 32) в поле **Mechanism Model Name** (Имя модели механизма) вводим **Lathe** (Токарный станок). Выбираем из выпадающего списка **Mechanism Type** строку **Device** (Устройство).

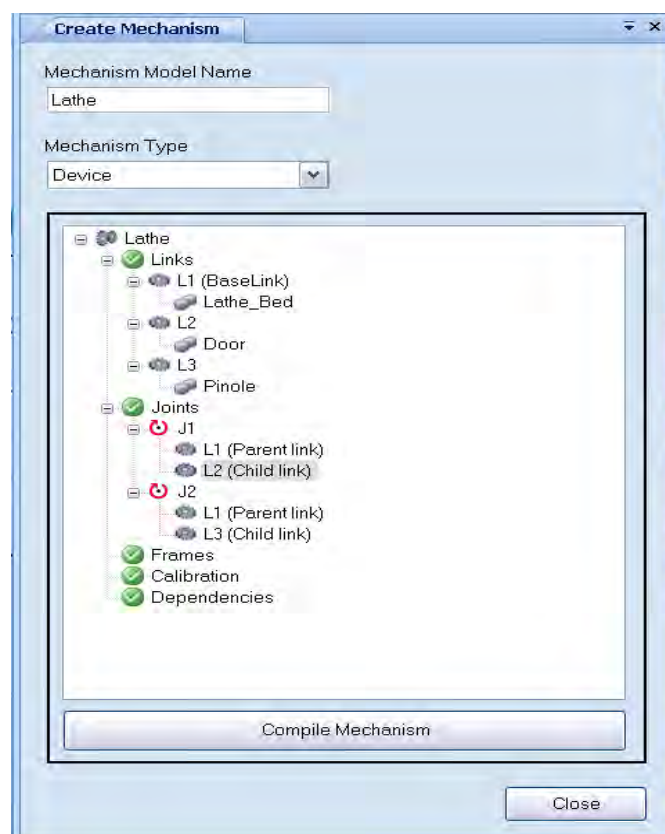


Рисунок 32 – Результат создания связей и перемещений

На дереве **Lathe** в окне **Create Mechanism** выбираем узел **Links** (Связи) и из контекстного меню узла выполняем команду **Add Link** (Добавить связь). В окне **Create Link** (Создать связь) в поле **Selected Part** (Выбрать часть) выбираем **Lathe**, устанавливаем флаг **Set as BaseLink** (База, неподвижная часть) и жмем зеленую стрелку. Нажимаем кнопку **Apply** (Применить). В поле **Selected Part** выбираем **Door**, жмем зеленую стрелку. Нажимаем кнопку **Apply**. В поле **Selected Part** выбираем **Pinole**, жмем зеленую стрелку и последовательно кнопки **Apply**, **Cancel**. Таким образом, определен состав подвижных и неподвижных деталей механизма.

Для задания перемещения двери на дереве **Lathe** выбираем узел **Joints** (Перемещения). Из контекстного меню узла выполняем команду **ADD Joint** (Добавить перемещение). В окне (рисунок 33) **Create Joint** (Создать перемещение) выбираем тип перемещения **Prismatic** (Линейное). В поле **Second Position** (Вторая

позиция) вводим координату Y 1 мм (определяем направление перемещения вдоль оси Y). В поле **Jog Axis** (перемещение по оси) перемещаем движок и наблюдаем перемещение двери в окне кабины станка. В поле **Min Limit** (Минимальный предел перемещения) записываем 0, в поле **Max Limit** (Максимальный предел перемещения) записываем 1000 мм. Нажимаем кнопку **Apply** (Применить).

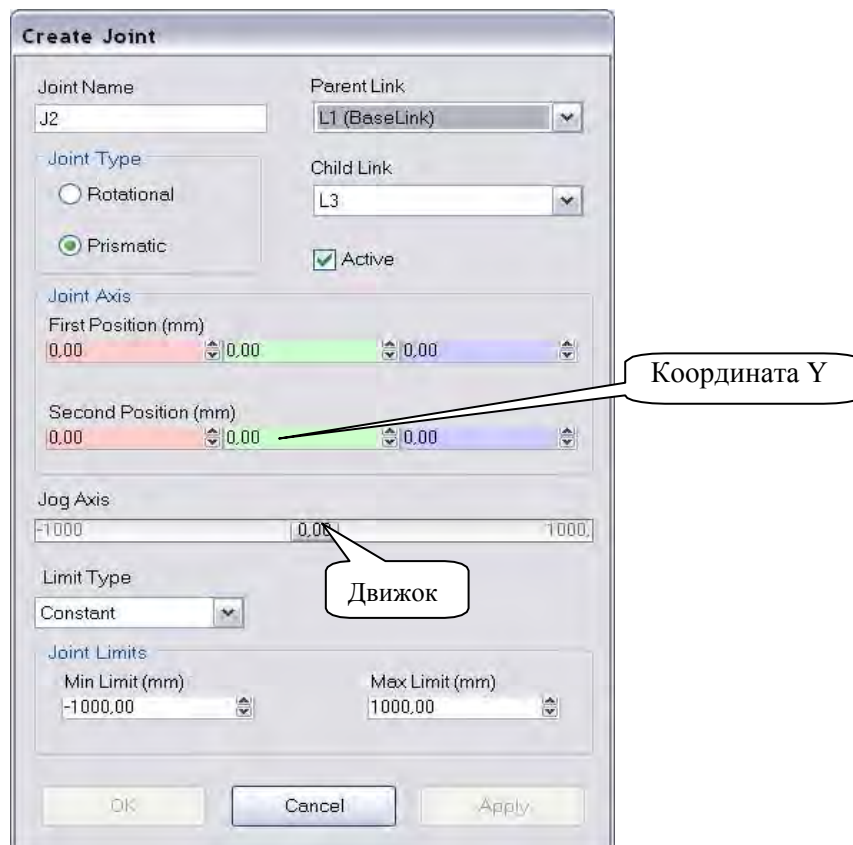


Рисунок 33 – Окно задания величины перемещений

Для задания перемещения пиноли в поле **Child Link** (Дочерняя связь) выбираем связь L3. В окне **Create Joint** выбираем тип перемещения **Prismatic**. В поле **Second Position** вводим координату Y 1 мм (определяем направление перемещения вдоль оси Y). В поле **Min Limit** (Минимальный предел перемещения) записываем 0, в поле **Max Limit** (Максимальный предел перемещения) записываем 16 мм. Последовательно нажимаем кнопки **Apply**, **Cancel**. Внизу окна **Create Mechanism** (см. рисунок 32) нажимаем кнопку **Compile Mechanism** (Собрать механизм).

Открываем закладку **Layout**, выбираем узел **Door**. В окне **Create Mechanism** щелкаем кнопку **Add**. В окне (рисунок 34) **Create Pose** (Создать положение) в поле **Pose Name** (Имя положения) вводим имя положения двери (движок стоит в положении 00) **Door Closed** (Дверь закрыта) и нажимаем кнопку **Apply**. В окне **Create Pose** вводим имя положения двери (движок стоит в положении 1000) **DoorOpened** (Дверь открыта) и нажимаем кнопку **OK**.

Аналогичным образом задаем положения пиноли задней бабки станка **Pino-leOn** (Пиноль подведена), **PinoleOff** (Пиноль отведена), используя движок для

задания положения пиноли. При моделировании РТК может быть установлена длительность цикла выполнения операции. Для того чтобы это значение было корректно определено, задается время перемещения частей моделируемого механизма. Для этого нажимаем кнопку **Set Transition Times** (Установить время перемещения) и в соответствующих полях (рисунок 35) **Door Closed**, **DoorOpened** устанавливаем время срабатывания механизма на открытие и закрытие дверей, например, 2 с. Аналогичным образом задаем время отвода **PinoleOff**, подвода **PinoleOn** пиноли задней бабки станка – 1 с. Закрываем последовательно окна **Set Transition Times**, **Create Mechanism** командой **Close**. На запрос «Are you sure that you want to close Mechanism Modeler?» (вы действительно желаете завершить моделирование механизма) отвечаете «Да».

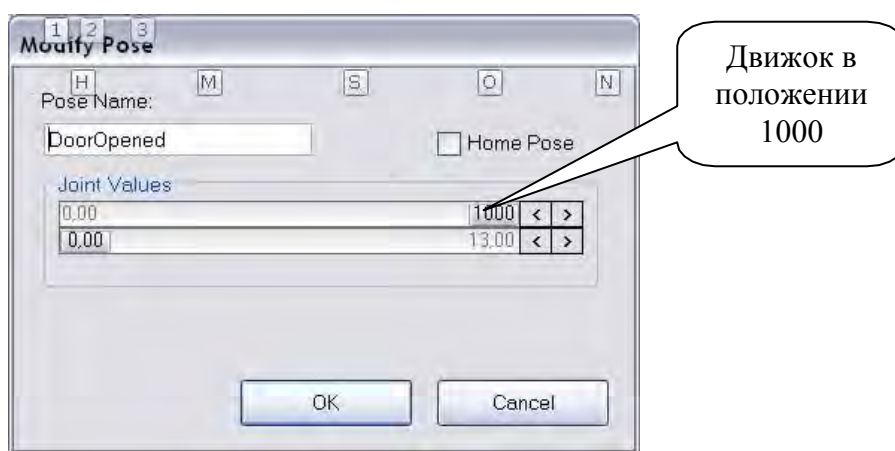


Рисунок 34 – Окно задания положения подвижных элементов механизма

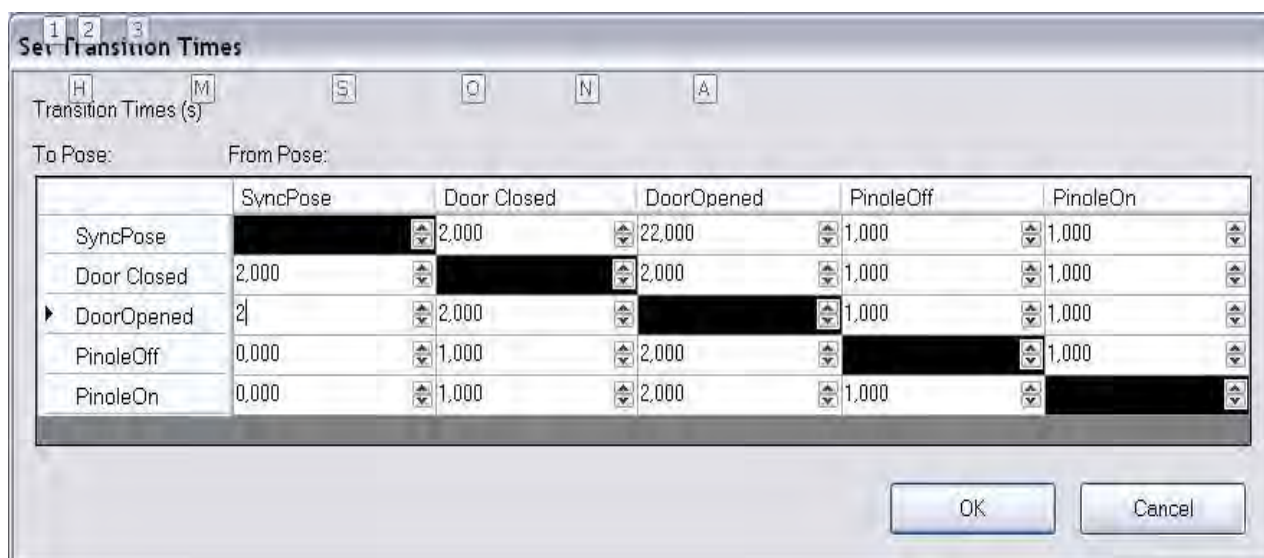


Рисунок 35 – Задание длительности перемещения подвижных частей механизма

На дереве проекта (закладка **Layout**) выбираем **Lathe**, открываем закладку меню **Modify** и открываем меню **Move to Pose** (Переместить в положение). Поочередно выбираем пункты **Door Closed**, **DoorOpened**, **PinoleOff**, **PinoleOn**

наблюдаем закрытие, открытие двери и перемещение пиноли.

На дереве проекта выбираем узел **Door** затем из контекстного меню команду **Mechanism Joint Jog**, на поле окна перемещаем движок, наблюдаем открытие закрытие двери.

На дереве проекта выбираем узел **Lathe** и из контекстного меню узла выполняем команду **Save as Library** (сохранить в библиотеке), вводим имя файла **Lathe** и сохраняем файл с расширением *.rslib, как библиотечный файл Robot Studio. В дальнейшем созданный цифровой прототип токарного станка может быть использован при создании различных РТК и их программировании.

3.6 Создание механизма «Схват».

Создадим пустую станцию командой **Empty Station**. Командой **Import Geometry** (меню Home) загрузим файл корпуса схвата **Mounting.sat**, а затем файлы рычагов схвата **Lever_R.sat**, **Lever_L.sat** из своей рабочей папки.

Далее в соответствии с методикой, изложенной в п. 3.5, создадим механизм **Gripper** (схват) с типом Tool (инструмент), который состоит из неподвижной части **Mounting** (корпус) и подвижных частей, правого и левого рычагов **Lever_R**, **Lever_L**.

В ходе создания механизма **Gripper** для правого и левого рычагов **Lever_R**, **Lever_L** должны быть созданы положения **Lever_R_Off** (закрыт), **Lever_R_On** (открыт), **Lever_L_Off** (закрыт), **Lever_L_On** (открыт). Величину перемещения рычагов схвата установим, перемещая движок положения моделируемой подвижной части механизма (см. рисунки 34 и 35).

При создании на дереве механизма определим данные инструмента **ToolData**. Масса схвата с заготовкой – 7,5 кг. Время срабатывания подвижных частей схвата установим равным 1 с.

После завершения создания механизма и проверки его работы сохраним его в файл под именем **Gripper** с расширением *.rslib как библиотечный файл **RobotStudio**.

В результате цифрового прототипирования основного и вспомогательного технологического оборудования созданы цифровые прототипы, которые отражают не только геометрические параметры моделируемых объектов, но и такие их характеристики, как величины перемещений подвижных элементов оборудования, скорости их перемещения, массоинерционные свойства.

Содержание отчета по лабораторным работам

1 Титульный лист.

2 Цель работы.

3 Скриншоты 3D-моделей основного и вспомогательного технологического оборудования с размерами, необходимыми в дальнейшем для задания путей перемещения руки робота со схватом.

4 Список файлов цифровых прототипов основного и вспомогательного технологического оборудования с указанием их назначения.

