




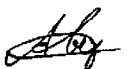
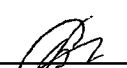
Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет энергетический
Кафедра Автоматизации производственных процессов и электротехники
Направление подготовки 15.03.04 - Автоматизация технологических
процессов и производств
Профиль Автоматизация технологических процессов и производств в
энергетике

ДОПУСТИТЬ К ЗАЩИТЕ
И.о. зав. кафедрой
 О.В. Скрипко
« 02 » июля 2020 г.

БАКАЛАВРСКАЯ РАБОТА

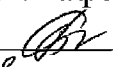
на тему: Разработка имитационных моделей систем управления движением
(комплексная выпускная квалификационная работа)

Исполнитель студент группы 641об	 02.07.2020 (подпись, дата)	М.А. Алеко
Руководитель доцент, канд. техн. наук	 02.07.2020 (подпись, дата)	А.Н. Рыбалёв
Консультант по безопасности и экологичности доцент, канд. физ.-мат. наук	 02.07.2020 (подпись, дата)	В.Н. Аверьянов
Нормоконтроль профессор, д-р техн. наук	 07.07.2020 (подпись, дата)	О.В. Скрипко

Благовещенск 2020

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет Энергетический
Кафедра автоматизации производственных процессов и электротехники

УТВЕРЖДАЮ
И.о. зав. кафедрой
 О.В. Скрипко
« 08 » июня 2020 г.

ЗАДАНИЕ

К выпускной квалификационной работе студента 641 Алеко Михаила Александровича

1. Тема выпускной квалификационной работы: Разработка имитационных моделей систем управления движением (комплексная выпускная квалификационная работа)

(утверждена приказом от 30.04.2020. № 810-уч)

2. Срок сдачи студентом законченной работы: 1 июля 2020 года.

3. Исходные данные к выпускной квалификационной работе: 1) ФГОС направления подготовки бакалавров 15.03.04 Автоматизации технологических процессов и производств; 2) Техническое задание на проектирование.

4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов):

- 1) Исследование и описание физических моделей;
- 2) Построение имитационных моделей;
- 3) Разработка программ управления и экранов визуализации.

5. Перечень материалов приложения (наличие чертежей, таблиц, графиков, схем, программных продуктов, иллюстративного материала и т.п.):

ПРИЛОЖЕНИЕ А – Техническое задание

ПРИЛОЖЕНИЕ Б – Код подпрограммы Kont_ri

ПРИЛОЖЕНИЕ В – Код подпрограммы r_ург

ПРИЛОЖЕНИЕ Г – Код подпрограммы a_ург

Лист 1 – Разработка 3D-модели в SolidWorks

Лист 2 – Экспорт 3D-модели в Simscape Multibody Link

Лист 3 – 3D модель крана в SolidWorks

Лист 4 – Разработка Simulink модели крана

Лист 5 – Разработка программы управления краном в CoDeSys

Лист 6 – Демонстрация работы имитационной модели систем управления движением крана

6. Дата выдачи задания 10.03.2020

Руководитель выпускной квалификационной работы Рыбалёв Андрей Николаевич, доцент кафедры АПП и Э, канд. техн. наук.

Задание принял к исполнению (дата): 10.03.2020



(подпись студента)

РЕФЕРАТ

Бакалаврская работа содержит 90 с., 63 рисунка, 1 таблицу, 12 источников.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ
ТЕХНОЛОГИЧЕСКИМ ПРОЦЕССОМ, СИСТЕМА ИМИТАЦИОННОГО
МОДЕЛИРОВАНИЯ, СТАНДАРТ ОРС.

Объекты для разработки проекта представлены в виде математических моделей, которые практически полностью реализуют необходимые свойства реальных объектов. Цель работы заключается в проектировании имитационных моделей систем управления АСУ ТП. Такая система включает в себя: модель объекта; управляющую программу; экран визуализации. Метод, рассмотренный в выпускной квалификационной работе, позволяет отказаться от физического объекта и программируемого логического контроллера, они реализованы в виде программ, разных классов и типов.

СОДЕРЖАНИЕ

Введение	7
1 Технология имитационного моделирования систем управления движением	9
1.1 Концепция	9
1.2 Разработка 3D-модели в SolidWorks	10
1.2.1 Создание деталей	10
1.2.2 Создание сборки	15
1.2.3 Наглядный пример 3D модели объекта	18
1.3 Пакет Simscape Multibody Link	19
1.4 Экспорт 3D-модели в Simscape Multibody Link	23
1.5 Разработка программы CodeSys для виртуального пакета	25
1.6 Запуск OPC сервера и его настройка	29
2 Имитационное моделирование системы управлени движением козлового крана	36
2.1 Создание 3D-модели крана	37
2.2 Создание Simulink модели	39
2.2.1 Основные блоки и их описание	40
2.3 Создание управляющей программы	43
2.4 Запуск и проверка работы системы	48
3 Безопасность жизнедеятельности	50
3.1 Организация рабочего места оператора	50
3.1.1 Расчет искусственного освещения	51
3.2 Инструкция по технике безопасности при выполнении работ на стенде	57
3.3 Общие требования безопасности к лабораторному стенду	59
3.3.1 Требования безопасности перед началом работы	60
3.3.2 Требования безопасности во время работы	61
3.3.3 Требования безопасности по окончанию работы	62

3.3.4 Требования безопасности в аварийных ситуациях	63
Заключение	64
Библиографический список	65
Приложение А	67
Приложение Б	70
Приложение В	74
Приложение Г	77

ВВЕДЕНИЕ

Создание АСУ новейших технологических процессов – это непростая и ответственная задача, получение результата которой осуществляется в несколько этапов: от создания математической модели до проектирования человеко-машинного интерфейса.[1] В данной работе будет фигурировать имитационное управление систем движения. Имитационное моделирование – это метод изучения объектов, основанный на том, что изучаемый объект заменяют имитирующим его объектом. Имитирующий объект исследуют (не прибегая к экспериментам на реальном объекте) и в результате получают нужную информацию исследуемого объекта. Основная задача имитационного моделирования – получение приближенных знаний о требуемом параметре объекта, не прибегая к непосредственному измерению его значений. Очевидно, что это требуется тогда и только тогда, когда проведение измерений невозможно, или оно стоит дороже, чем проверка на имитационной модели. При этом для исследования данного параметра можно воспользоваться другими известными параметрами объекта и моделью его конструкции. С вероятностью, что имитационная модель достаточно точно описывает объект, предполагается, что полученные в ходе имитации статистические распределения значений параметра моделирующего объекта будут в той или иной степени совпадать с распределением значений параметра реального объекта.

Задачи, которые необходимо решить:

- имитационное моделирование технологического процесса в различных режимах работы при воздействиях, программно формируемых управляющей аппаратурой и средствами человеко-машинного интерфейса;
- отладка технологических программ;
- выбор наиболее удобных для пользователя средств визуализации технологического процесса и способов формирования управляющих воздействий.

Оперативный персонал задействует программный комплекс на этапе настройки АСУ ТП, а также в целях обучения.

Кроме того, разрабатываемая система, безусловно, будет весьма полезна в учебном процессе по образовательным программам, предусматривающие изучение дисциплин, связанных с проектированием АСУ ТП.

В рамках единого комплекса предлагается задействовать программные средства разных производителей и классов:

- система имитационного моделирования – для построения моделей технологического процесса;

- система класса PC-based controller – для программной реализации алгоритмов управления на языках программирования промышленных контроллеров;

- SCADA-система (supervisory control and data acquisition – система диспетчерского управления и сбора данных) – для визуализации технологических процессов и оперативного управления.

Перечисленные программные средства предназначены для исследования и разработки компонентов АСУ ТП, но, используемые по отдельности, не могут решать перечисленные выше задачи.

Для построения прототипа было решено использовать следующие программные продукты:

- MathWorks[®] MATLAB[®], Simulink[®] (среда имитационного моделирования);[8]

- 3S-Smart Software[®] CODESYS[®] (PC-эмулятор ПЛК SP PLCWinN, OPC-сервер);

Выбор данных программ обусловлен опытом их применения в учебном процессе.

1 ТЕХНОЛОГИЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ СИСТЕМ УПРАВЛЕНИЯ ДВИЖЕНИЕМ

Понятие «компьютерного моделирования» в сфере информационных технологий относительно новое и связано со становлением и выделением относительно традиционного моделирования с помощью ЭВМ (последние - это, как правило, функционально-ориентированные автоматизированные системы поддержки математического и других видов моделирования, реализуемые обычно как системы библиотечного типа) двух современных видов компьютерного моделирования: - структурно-функционального и имитационного.[2] Компьютерное моделирование – эффективный метод решения задач анализа и синтеза сложных систем. Методологической основой компьютерного моделирования является системный анализ (в то время, как у моделирования на ЭВМ – те или иные разделы теории математических моделей). В мире информационных технологий имитационное моделирование переживает второе рождение.

Особое внимание к этому виду компьютерного моделирования проявило себя в связи значительного технологического развития систем моделирования, которые в нынешних реалиях являются мощным аналитическим средством, объединившим в себе весь арсенал современных информационных технологий, содержащие улучшенные графические оболочки для целей конструирования моделей и интерпретации выходных результатов моделирования, мультимедийные средства и видео, поддерживающие анимацию в масштабе реального времени, объектно-ориентированное программирование.

1.1 Концепция

Концепция состоит в том, что у нас имеется объект, имитационную модель движения которого нужно создать. Для этого в программе SOLIDWORKS создаем 3D-модель каждого элемента этой системы и в последующем объединить их в сборку. После с помощью Simscape

Multibody Link мы экспортируем 3D-модель объекта (в виде сборки ее элементов) в MatLab и получаем Simulink модель, настраиваем ее и в приложении CoDeSys пишем для нее программу, с помощью которой будем управлять движением имитационной модели объекта. Для того что бы заставить работать вместе Simulink модель и программу из CoDeSys, используем OPC сервер. Так же программу из CoDeSys нужно занести в ПЛК (в нашем случае мы используем PC-эмулятор ПЛК SP PLCWinN). Схема концепции изображена на рисунке 1.

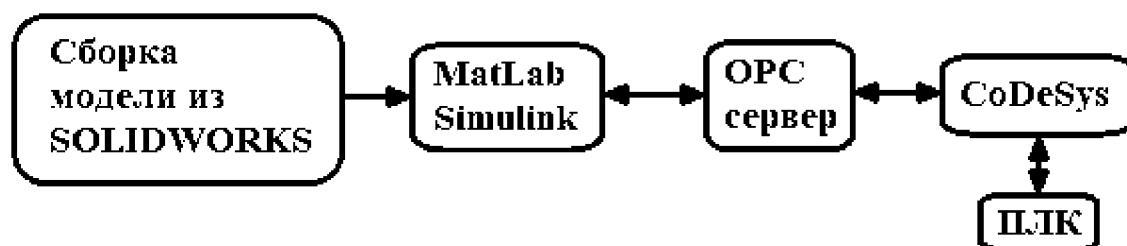


Рисунок 1 – Схема концепции

1.2 Разработка 3D-модели в SolidWorks

Невзирая на обширные возможности Simscape Multibody Link[9], смоделировать многозадачную механическую систему очень сложно. Связано это с необходимостью определять тензоры моментов инерции элементов, учитывать координаты их расположения и т.д. Для облегчения моделирования в среде Simscape Multibody Link, компания MATHWORKS создала специализированный CAD-транслятор, делающий возможным создание динамических моделей механизмов в среде Simscape Multibody Link на основе их твердотельных моделей разработанных в CAD-системах (напр., SolidWorks[3]). При применении CAD-транслятора тензоры моментов инерции и присоединительные размеры отправляются из CAD-системы в Simscape Multibody Link без изменений, при этом, работоспособность моделей проверяется в CAD-системе посредством установления верных связей между деталями механизма. Такой подход гораздо облегчает задачу и увеличивает возможности имитационного моделирования мехатронных систем.

Solidworks – программный комплекс САПР ориентированный для автоматизации этапов подготовки производства. Главной задачей Solidworks является работа с 3D моделями, именно о них и будет идти речь.

1.2.1 Создание деталей

Для создания 3D модели в открывшемся окне программы SolidWorks выбираем пункт создать новый документ (рисунок 2).[4]



Рисунок 2 – Создание нового документа

Чтобы получить правильную модель нашего объекта нам необходимо разбить его на отдельные детали и в первую очередь начертить их, поэтому выбираем тип документа «Деталь» (рисунок 3).

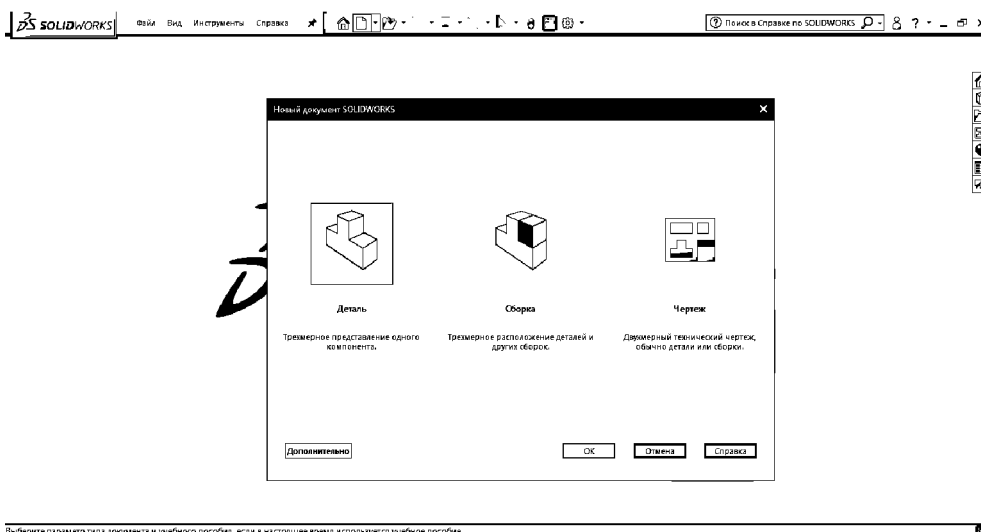


Рисунок 3 – Создание нового документа типа «Деталь»

После чего перед нами появляется рабочая область, пример которой изображен на рисунке 4.

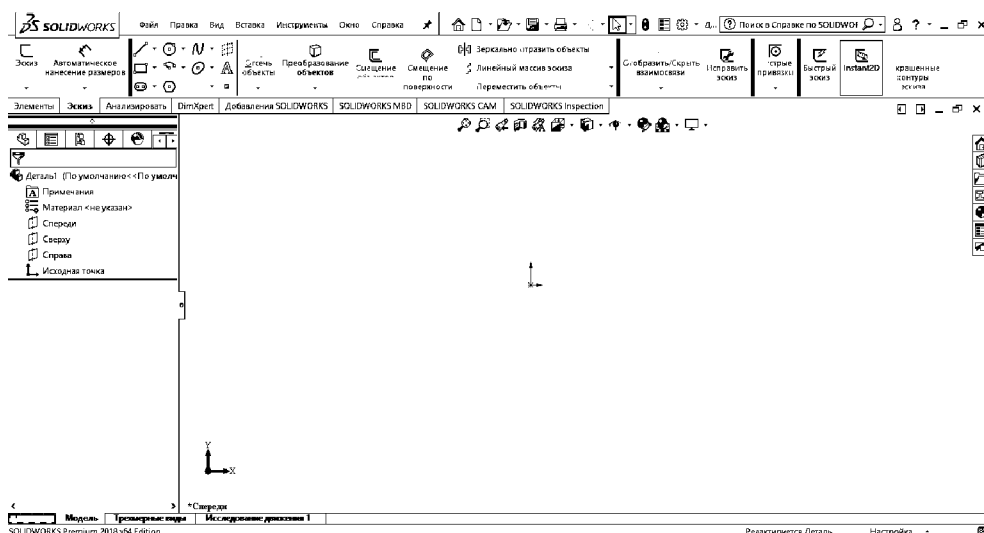


Рисунок 4 – Рабочая область документа типа «Деталь»

Далее выбираем плоскость, с которой начнем постройку эскиза детали, в нашем случае выберем вид спереди (рисунок 5). После чего можно сразу приступить к постройке детали.

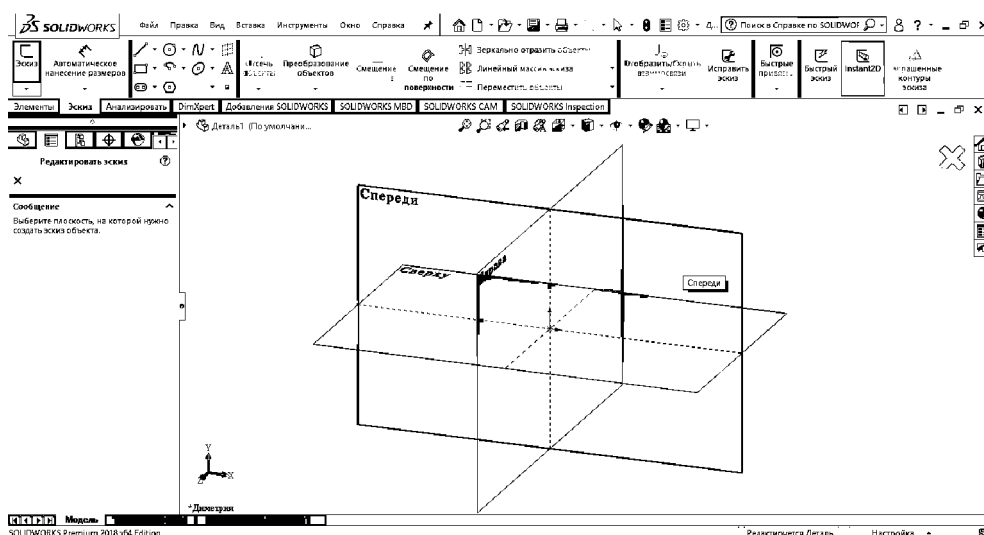


Рисунок 5 – Выбор плоскости

Для того чтобы получить эскиза будущей детали пользуемся панелью инструментов, во вкладке эскиз (рисунок 6).

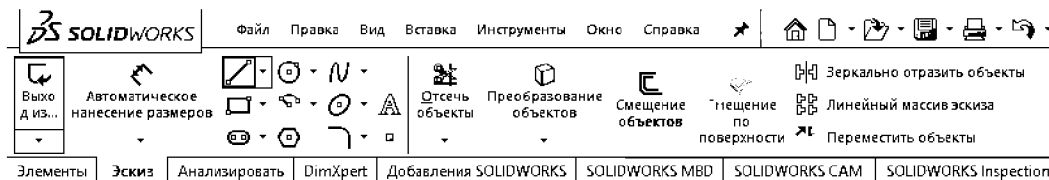


Рисунок 6 – Панель инструментов

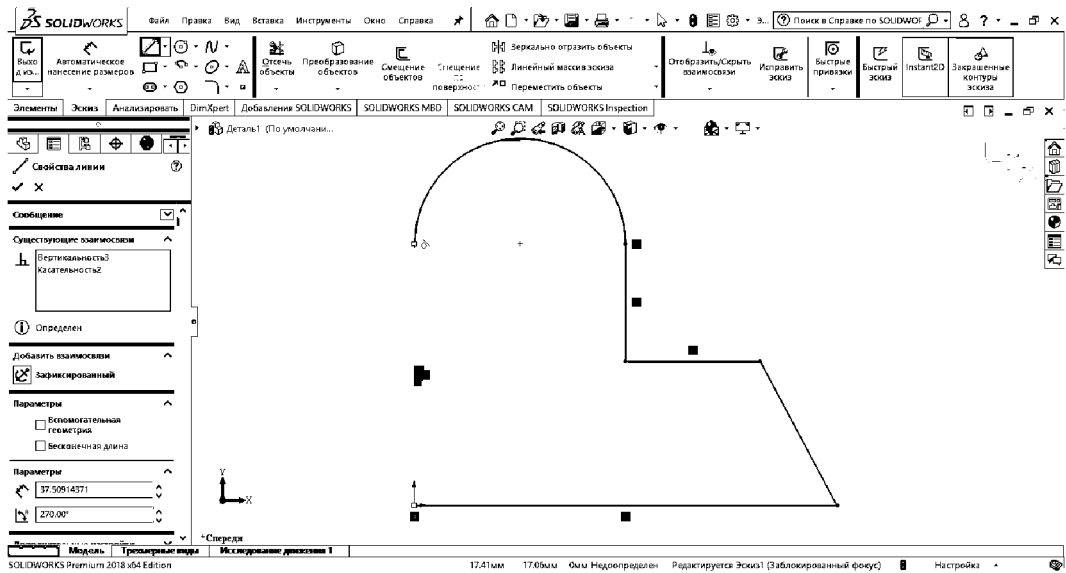


Рисунок 7 – Готовый эскиз детали

Когда эскиз готов, заходим на вкладку элементы, выбираем пункт «вытянутая бобышка» (рисунок 8).

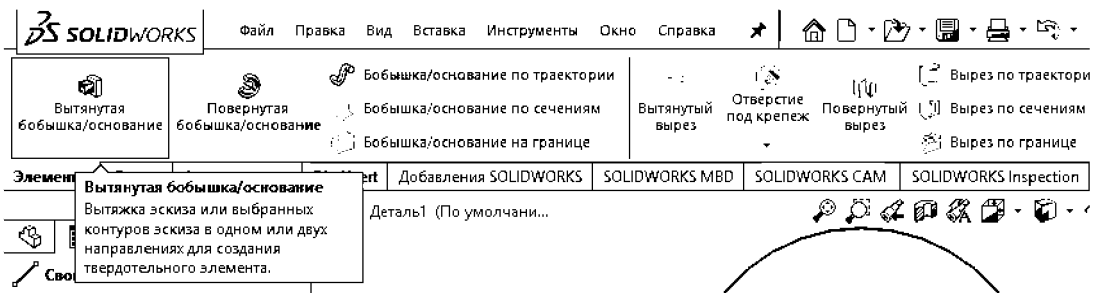


Рисунок 8 – Пункт «Вытянутая бобышка»

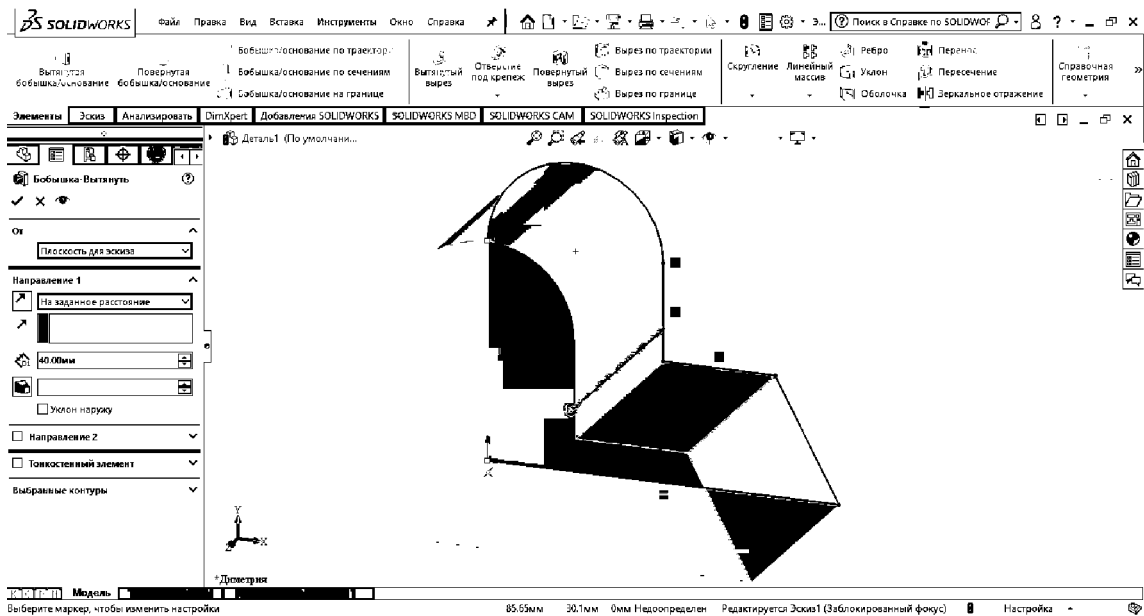


Рисунок 9 – Эскиз после нажатия на пункт «Вытянутая бобышка»

В появившемся окне, в настройках (рисунок 10), вводим необходимые характеристики и ждем галочку, чтобы подтвердить настройки.

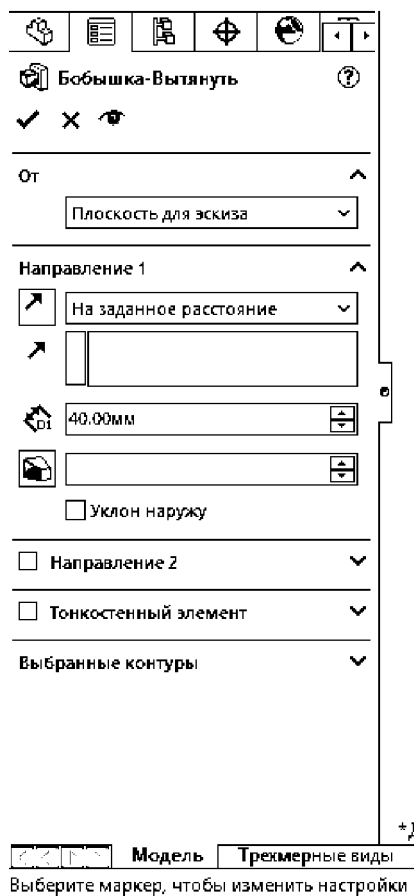


Рисунок 10 – Настройки

В итоге получаем требуемую деталь (рисунок 11).

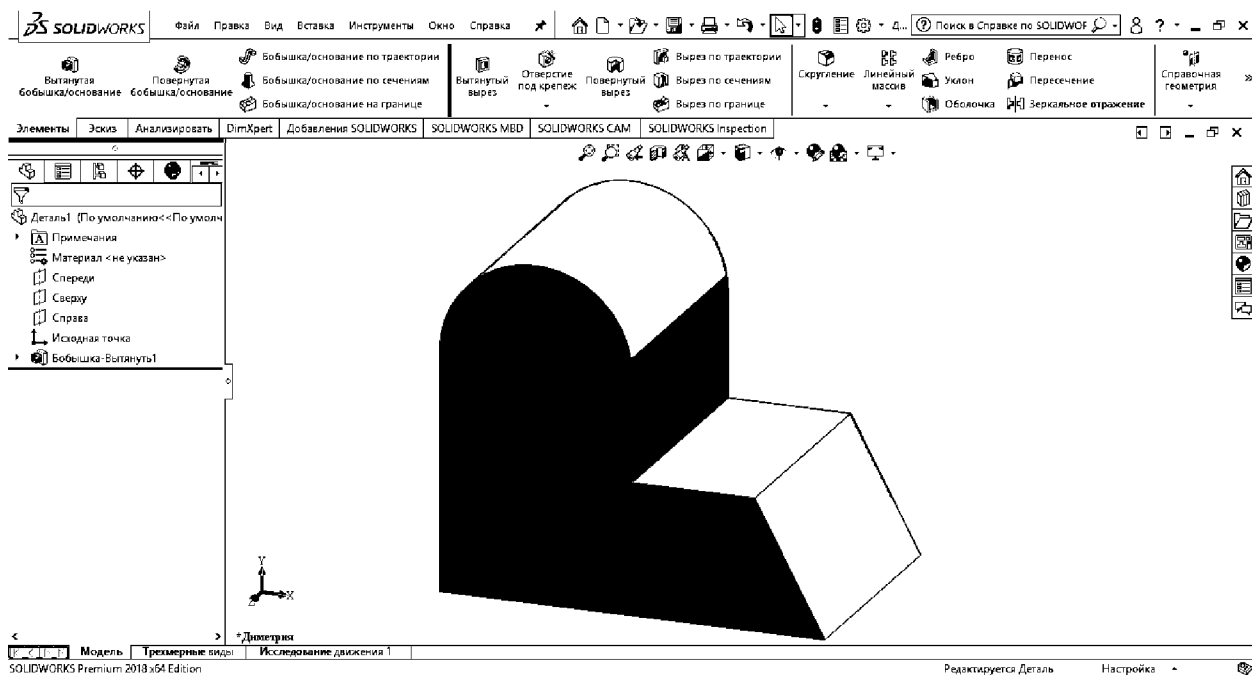


Рисунок 11 – Готовая деталь

1.2.2 Создание сборки

После того как все необходимые детали объекта готовы можно переходить к его сборке, по итогу которой мы получим точную 3D модель требуемого объекта. Для этого создаем новый документ и теперь выбираем тип документа «Сборка» и нажимаем ОК (рисунок 12).[5]

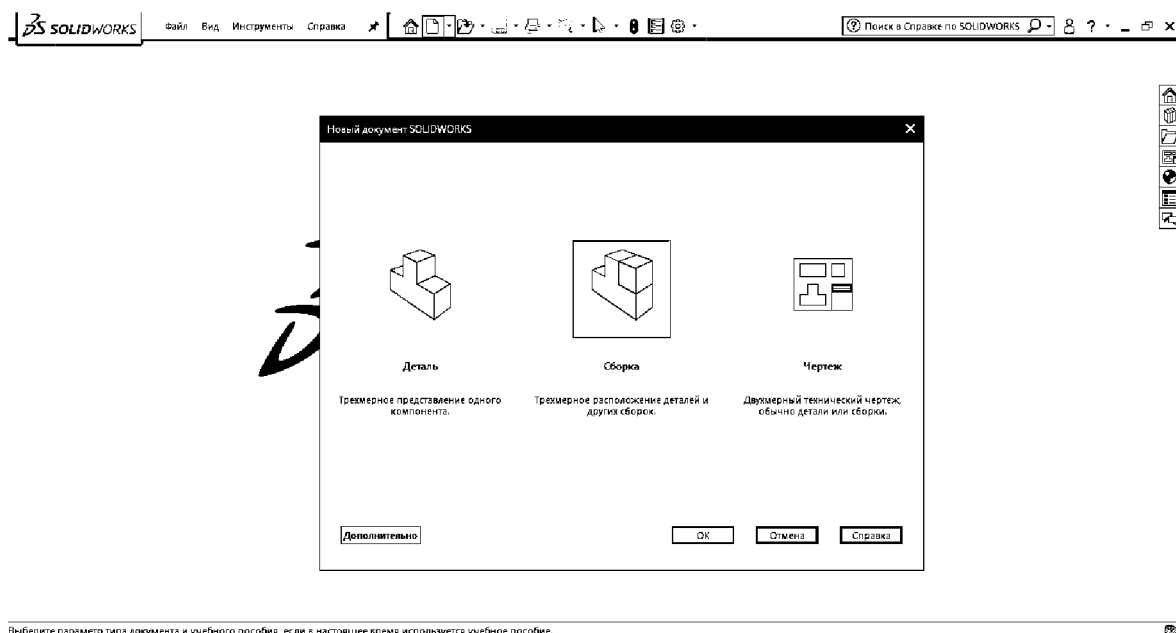


Рисунок 12 – Создание нового документа типа «Сборка»

Перед нами появится рабочая область для сборки (рисунок 13), которая отличается от той, что была в документе типа «Деталь».

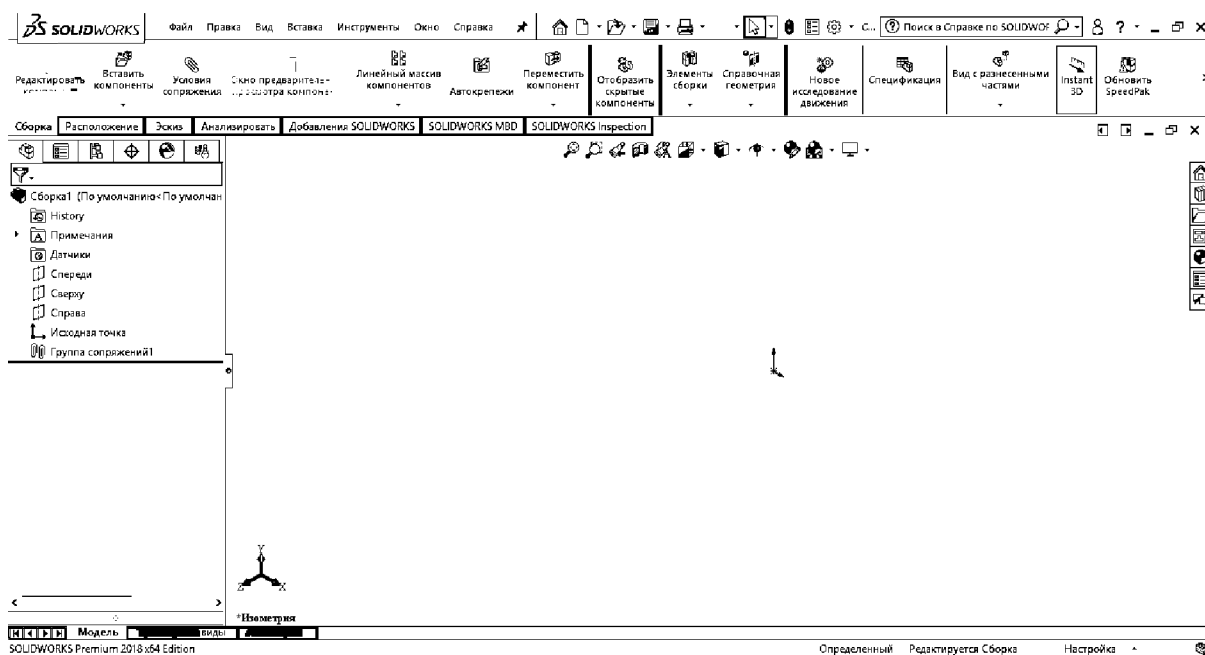


Рисунок 13 – Рабочая область документа типа «Сборка»

Для того чтобы добавить детали выбираем в сборке пункт «вставить компоненты» (рисунок 14).

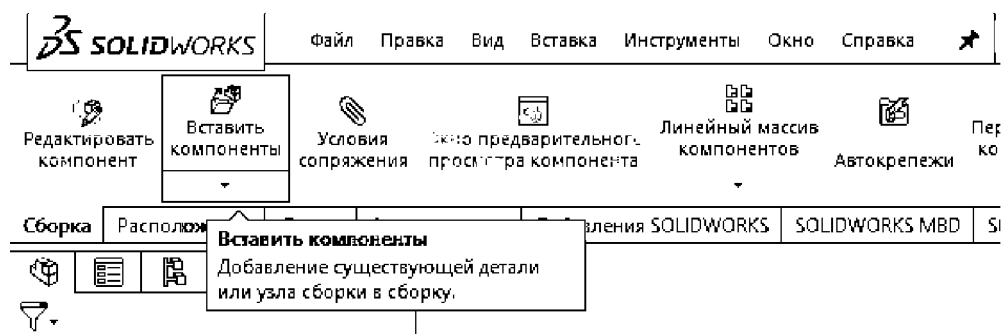


Рисунок 14 – Пункт «Вставить компоненты»

После того как нажали на «Вставить компоненты» перед нами появится окно (рисунок 15) с помощью которого выбираем требуемую деталь и вставляем её, данную операцию повторяем для каждой требуемой детали.

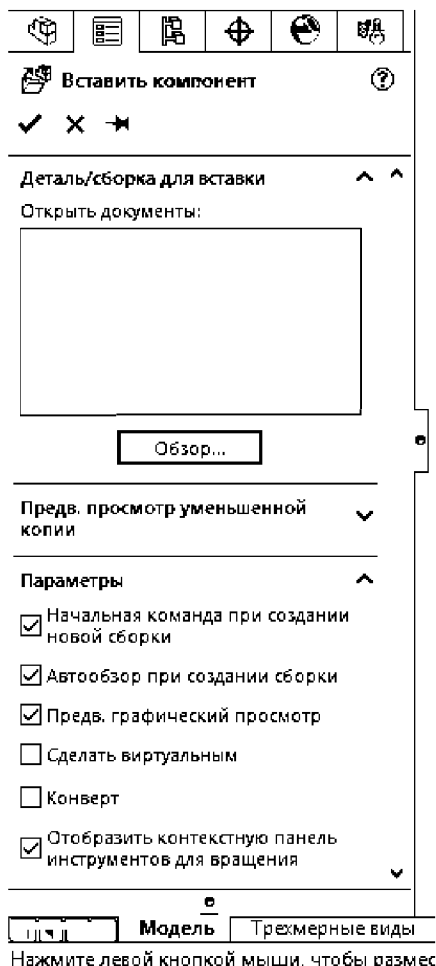


Рисунок 15 – Окно «Вставить компоненты»

Разместив все детали на рабочей области (рисунок 16) их нужно соединить между собой, так как это необходимо. Соединение производим

при помощи пункта «Условия сопряжения» (рисунок 17) и его настроек (рисунок 18).

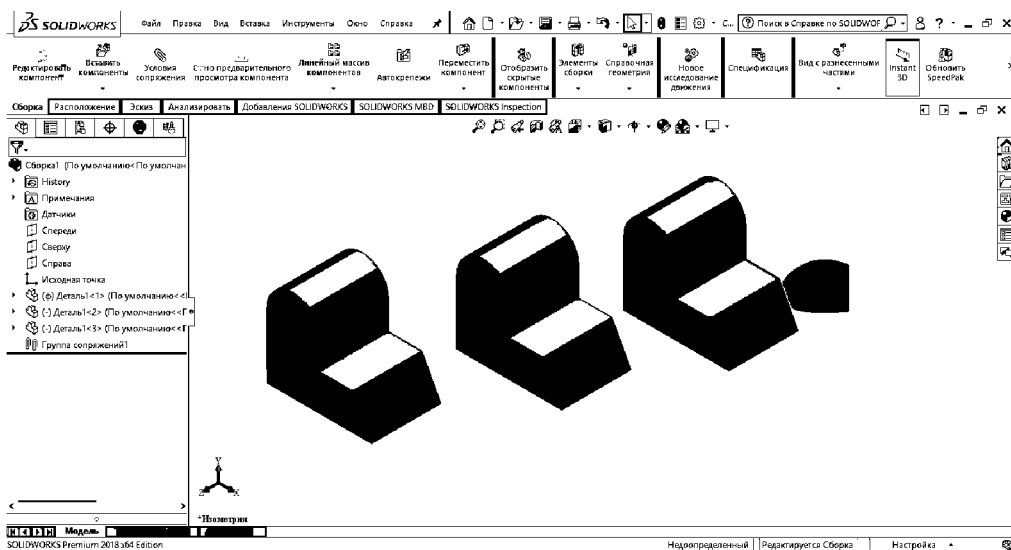


Рисунок 16 – Детали на рабочей области

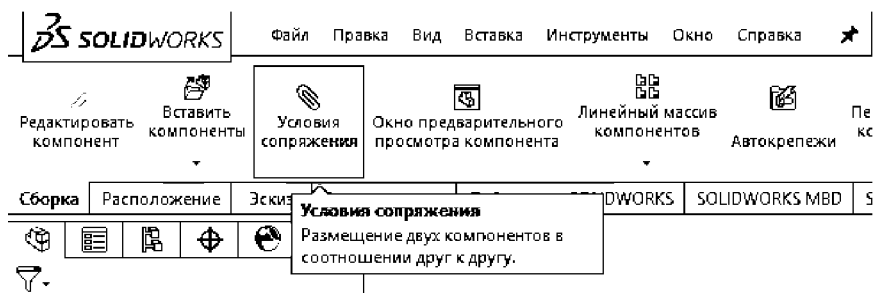


Рисунок 17 – Пункт «Условия сопряжения»

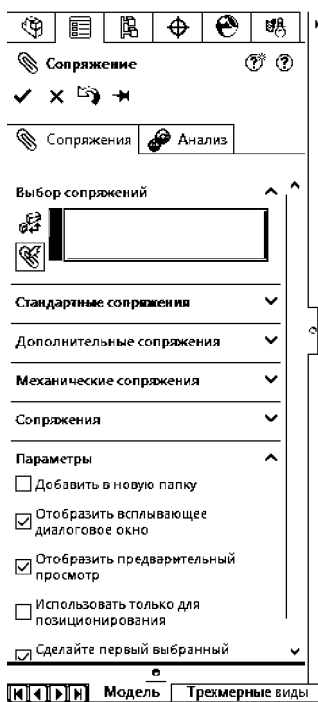


Рисунок 18 – Настройки пункта «Условия сопряжения»

После того как все детали объекта соединены между собой так как это необходимо, получаем готовую 3D модель объекта (рисунок 19).

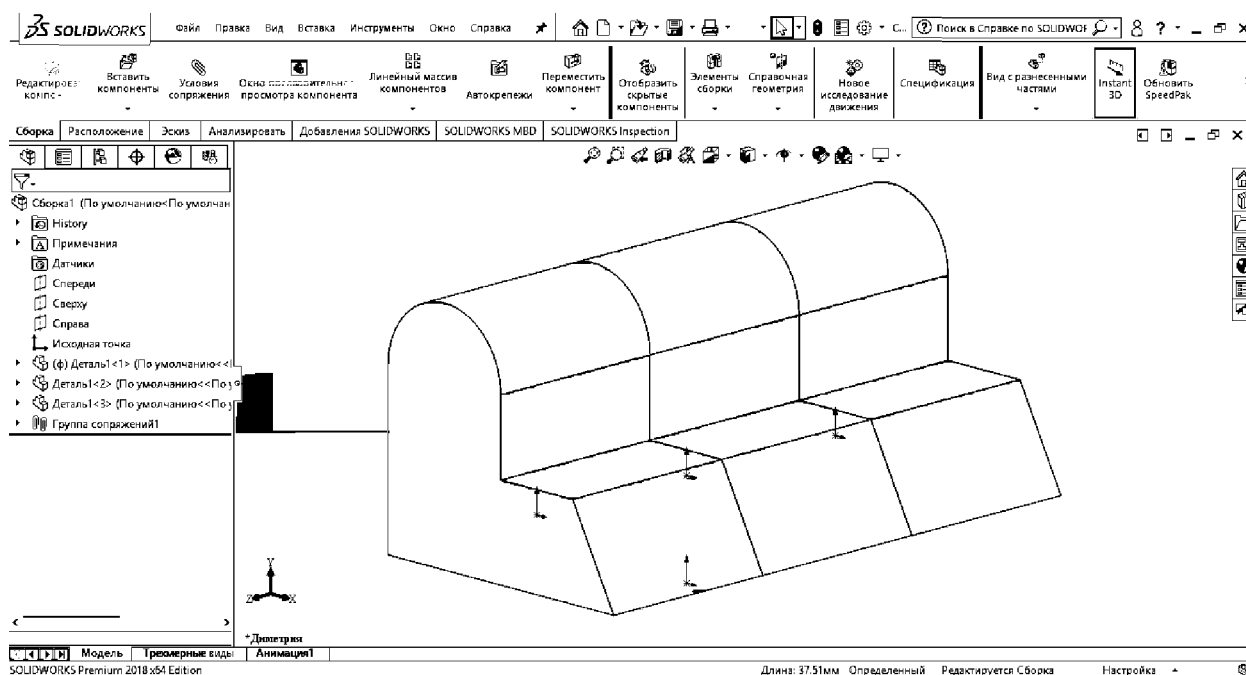


Рисунок 19 – Готовая 3D модель объекта

1.2.3 Наглядный пример 3D модели объекта

Для примера представим, что у нас есть какое-то основание и на нем куб, который двигается по двум осям X и Y. Разделив модель объекта по частям, имеем два объекта куб (рисунок 20) и основание (рисунок 21), из которых состоит модель. Эти объекты объединяем в сборку (рисунок 22).

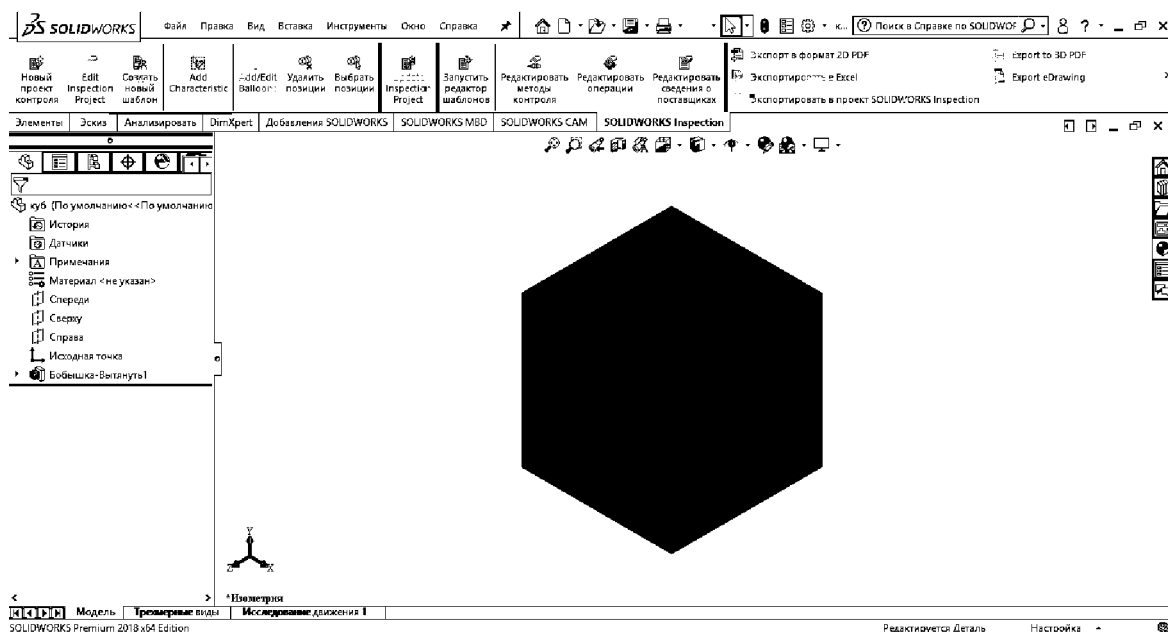


Рисунок 20 – 3D-модель куба

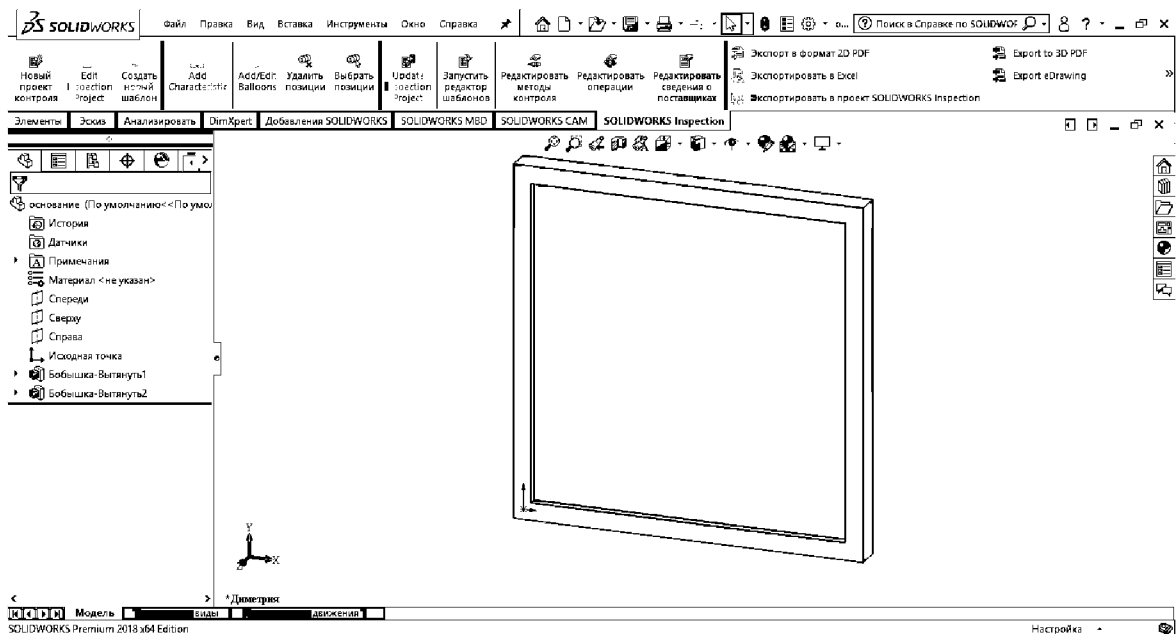


Рисунок 21 – 3D-модель основания

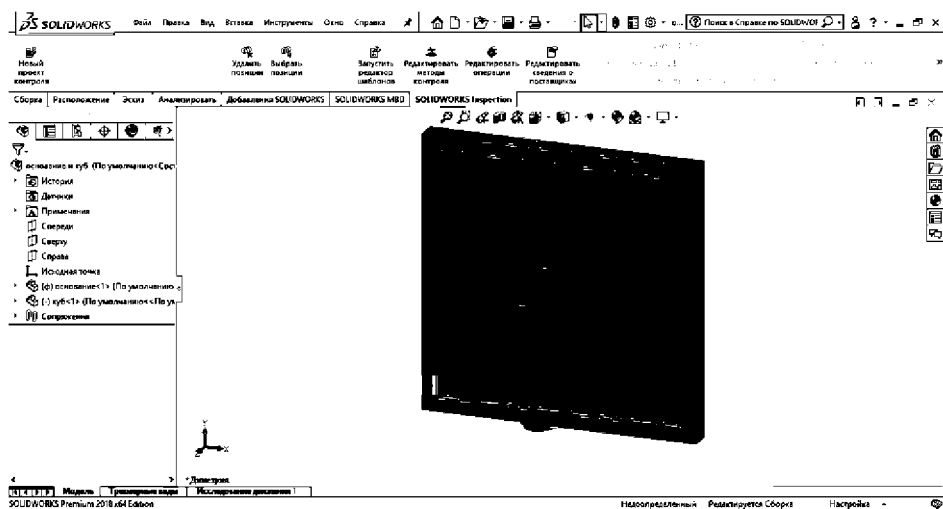


Рисунок 22 – 3D-модель сборки

1.3 Пакет Simscape Multibody Link

Simscape Multibody Link является частью программного комплекса имитационного моделирования Simulink системы MATLAB и даёт возможность моделировать сложные физические системы с помощью ненаправленных сигнальных графов.[1] При этом в инструментальной базе Simscape Multibody Link в наличии имеются блоки, посредством которых можно осуществить обмен информацией разработанной модели (передачу и прием сигналов) с другими элементами библиотеки Simulink. Это дает допустимым моделировать мехатронные и робототехнические системы,

транспортные средства, летательные аппараты, производственное оборудование, до того, как будет начато их производство.

1) С официального сайта MATHWORKS необходимо скачать дистрибутив Simscape Multibody Link – CAD транслятор для системы SolidWorks. Версия дистрибутива, при этом, должна соответствовать версии Matlab. Также необходимо скопировать установочный файл `install_addon.m`;

2) Открыть Matlab. Запуск Matlab должен производиться с правами администратора. Поэтому в свойствах файла (ярлыка на рабочем столе) MATLAB.exe на вкладке «Совместимость» необходимо выбрать опцию «Выполнять эту программу от имени администратора»;

3) В MatLab открыть установочный m-файл – `install_addon.m`. После чего запустится программа «Editor», ориентированная на создания и редактирования m-файлов среды Matlab и вылезет диалоговое окно редактирования функции `install_addon.m`;

4) В главном меню «Editor» выбрать: Debug – Run Configuration for `install_addon.m` – Edit Run for `install_addon.m`;

5) В появившемся диалоговом окне «Edit M-File Configurations» написать:

```
% Modify expression to add input arguments. % Example: % a = [1 2 3; 4  
5 6]; % foo(a);
```

```
install_addon(указать полный путь архива, например: 'C:\Documents and  
Settings\student\Рабочий стол\Матлаб\smlink31.win32.zip')
```

6) Запустить M-File, путём нажатия кнопки Run;

7) После в окне Windows нажать кнопку Пуск – Выполнить – ввести `cmd`;

8) В командной оболочке ввести: `matlab-regservers`;

9) В окне консоли MatLab Command Window ввести: `regmatlabserver`;

10) В окне Command Window MatLab ввести: `enable_service('AutomationServer',true)`. Ответ должен быть 1, если `ans = 0`, то необходимо ещё раз ввести `enable_service('AutomationServer',true)`;

11) В окне Command Window MatLab ввести: `smlink_linksw`. В результате, если все получилось, должно появиться сообщение об успешной регистрации.

В Simscape Multibody Link есть блоки, находящиеся в библиотеке Simulink (рисунок 23), с помощью которых наша модель может работать.[10]

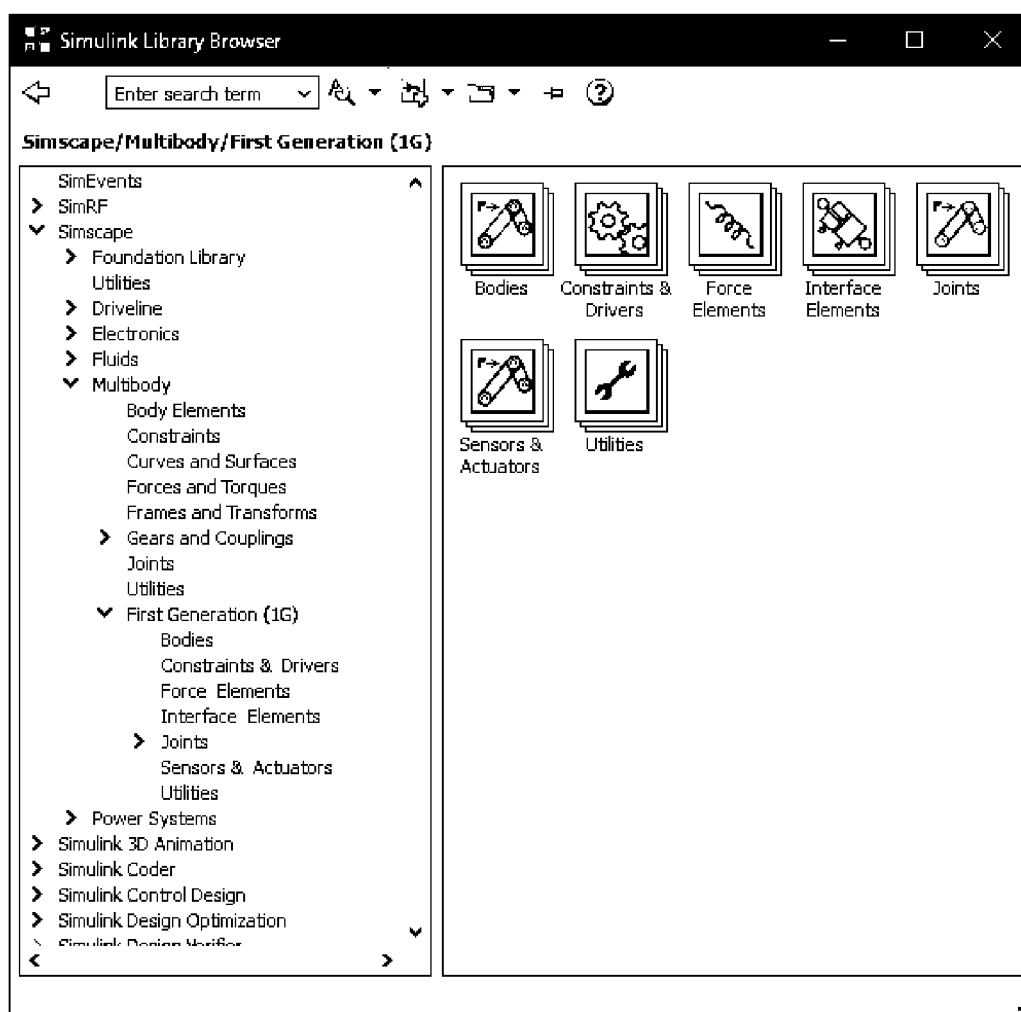
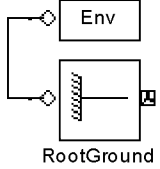
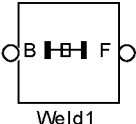
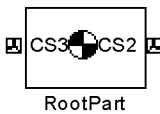
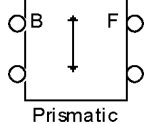
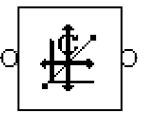


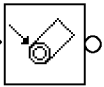
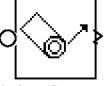
Рисунок 23 – Библиотека элементов Simulink

В таблице 1 описаны блоки Simscape Multibody Link.

Таблица 1 – Описание блоков Simscape Multibody Link подсистемы Robots

	<p>Блок Ground, представляет собой неподвижную заземленную точку, подключенный к его входу блок, задает механические параметры для одного механизма.</p>
	<p>Блок Wade, представляет неподвижное соединение звеньев.</p>
	<p>Блок Body, представляет собой твердое настраиваемое тело. В свойствах блока указывается:</p> <ul style="list-style-type: none"> – Масса тела и момент инерции; – Координаты центра тяжести(CG); – Один или более систем координат тела(CSS); – Дополнительная геометрия, цвет.
	<p>Блок Prismatic, представляет собой поступательную кинематическую пару. В настройках блока указывается координата, относительно которой выполняется движение.</p>
	<p>Блок Custom Joint, представляет общее пользовательское соединение с несколькими степенями свободы. Соединяет два тела с комбинацией призматических, вращающихся и/или сферических примитивов. Этот блок ограничен максимум шестью степенями свободы: до трех вращательных степеней свободы и до трех поступательных степеней свободы. Первый примитив, прикрепленный к базе (B). Последний примитив, прикрепленный к последователю (F).</p>

Продолжение таблицы 1

 <p>Joint Actuator1</p>	<p>Соединение между двумя телами представляет относительные степени свободы, блок Joint Actuator приводит в поступательное движение, либо вращательное движение эти тела с точки зрения линейной позиции.</p>
 <p>Joint Sensor</p>	<p>Блок Joint Sensor измеряет положение блока в пространстве.</p>

1.4 Экспорт 3D-модели в Simscape Multibody Link

Процесс передачи модели из SolidWorks в Simscape Multibody Link осуществляется следующим образом, берем модель сборки из SolidWorks и экспортируем Simscape Multibody Link (рисунок 24).

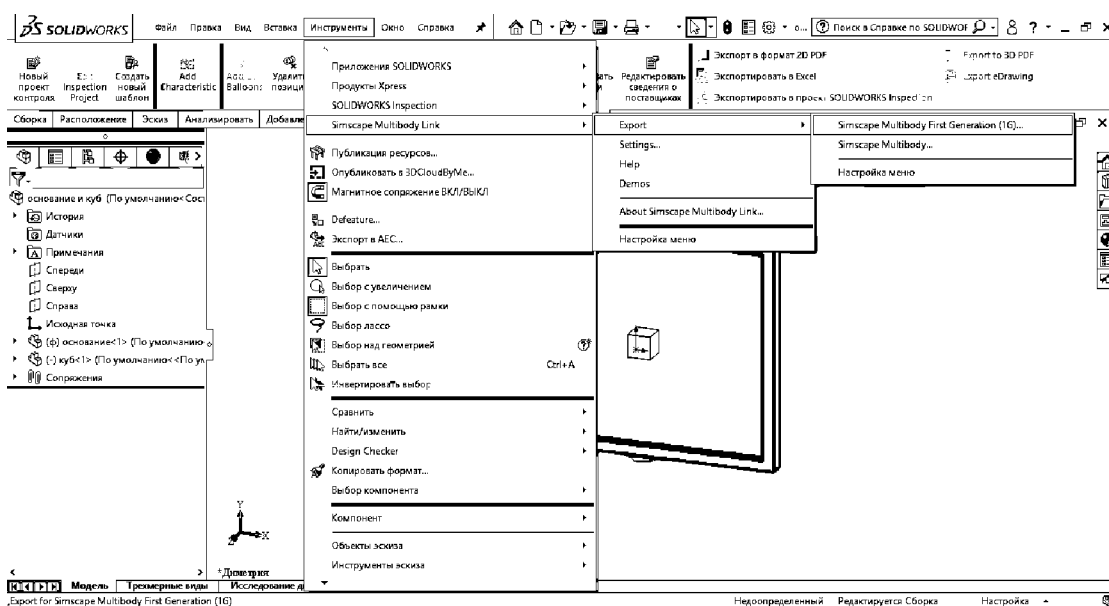


Рисунок 24 – Экспорт из SolidWorks в Simscape Multibody Link

Сохраняем файл в появившемся окне в формате .XML в удобное для вас место (рисунок 25).

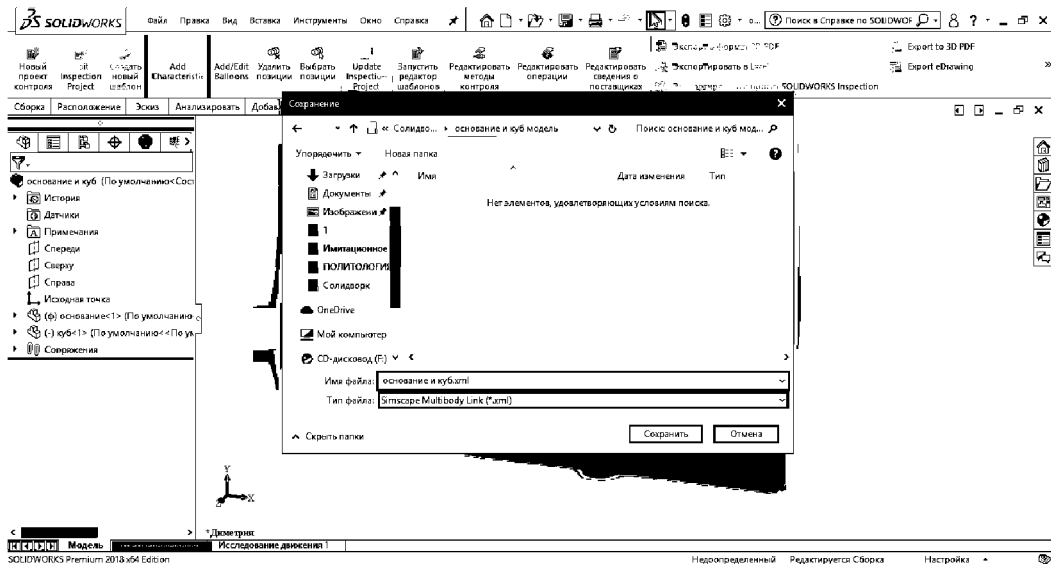


Рисунок 25 – Сохранение файла в .XML

В появившемся командном окне MatLab (Рисунок 26) пишем команду `mech_import` ('путь\имя файла.xml').

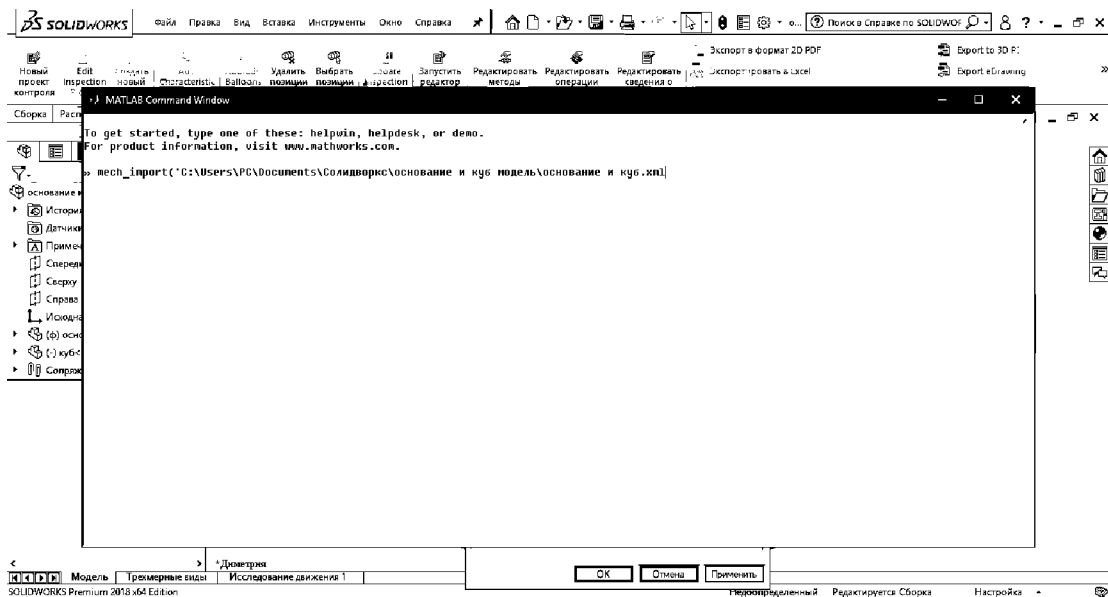


Рисунок 26 – Командное окно MatLab

После того как ввели команду и нажали кнопку ввод, создалась Simulink модель (рисунок 27) с которой можно в дальнейшем работать.

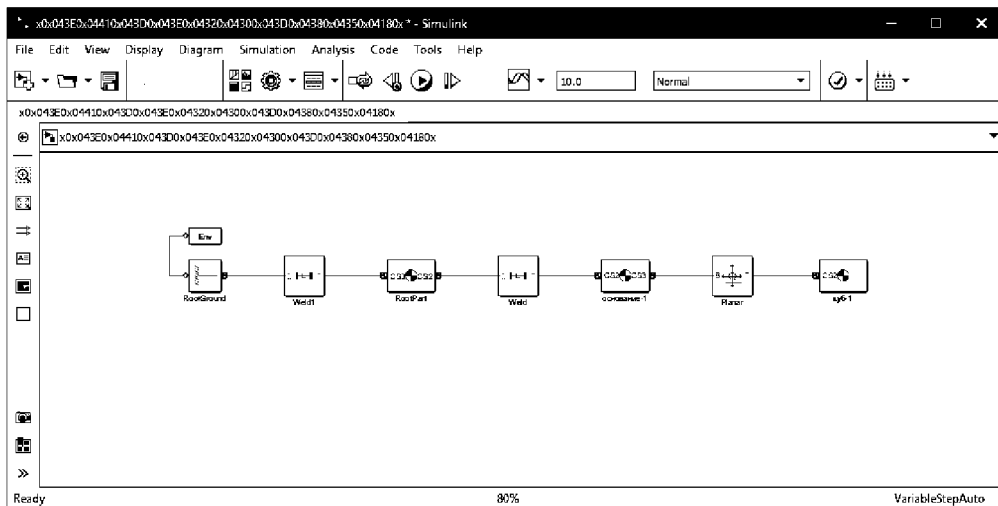


Рисунок 27 – Simulink модель

1.5 Разработка программы CodeSys для виртуального пакета

CodeSys – инструментальный программный комплекс промышленной автоматизации. Производится и распространяется компанией 3S-Smart Software Solutions GmbH.[6]

CodeSys – один из самых развитых инструментов для разработки программ ПЛК. CodeSys поддерживает 5 языков МЭК (IT, LD, FBD, SFC и ST) и еще один язык – CFC. То есть если разрабатывать программу в CodeSys можно на любом из этих языков. Кроме того, допускается комбинирование языков программирования в одной программе.

Прежде чем разрабатывать программу, необходимо создать новый проект. Делается это с помощью главного меню файл – создать (рисунок 28).

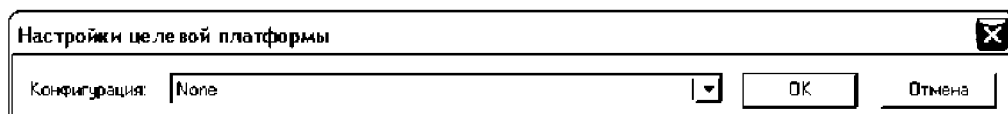


Рисунок 28 – Настройки целевой платформы

Проект является машинно-независимым, то есть программирование можно не привязывать к какой-либо конкретной модели контроллера. Если же вы хотите разрабатывать программу для конкретной модели, то в окне настройки целевой платформы нужно выбрать эту модель в поле КОНФИГУРАЦИЯ. Однако вашего контроллера в списке может не оказаться. Тогда нужно будет найти целевой файл (target-file) и установить

его, а потом создавать новый проект. В любом случае выбрать модель контроллера можно позже, уже после создания проекта.

Мы же всё это затеяли в учебных целях, поэтому оставим значение None и нажмём кнопку ОК, после чего появится окно.

Любой проект состоит из одного или нескольких модулей. В однозадачном проекте обязательно должен присутствовать главный программный модуль с именем PLC_PRG.

В этом окне мы устанавливаем следующие настройки:

1.Имя нового POU – PLC_PRG

2.Тип POU – программа

3.Язык реализации – ST. Язык реализации – это язык программирования, на котором будет написан программный модуль (POU) (рисунок 29).

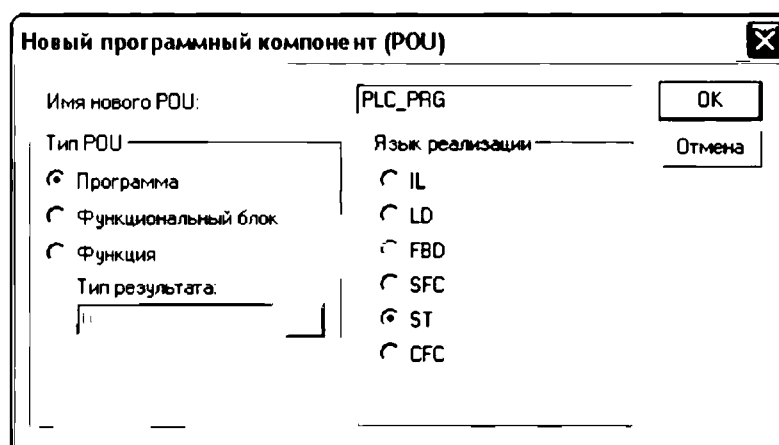


Рисунок 29 – Новый программный компонент

Нажимаем ОК. Всё, проект готов. Осталось только сохранить его. Как и во всех рассмотренных ранее случаях, сохраняем его в отдельном каталоге. Сохранение выполняется через меню ФАЙЛ – СОХРАНИТЬ. Присвоим файлу имя TEST и сохраним проект.

После сохранения в выбранном каталоге появится файл TEST.PRO. Главное окно CoDeSys с новым проектом показано на рисунке 30.

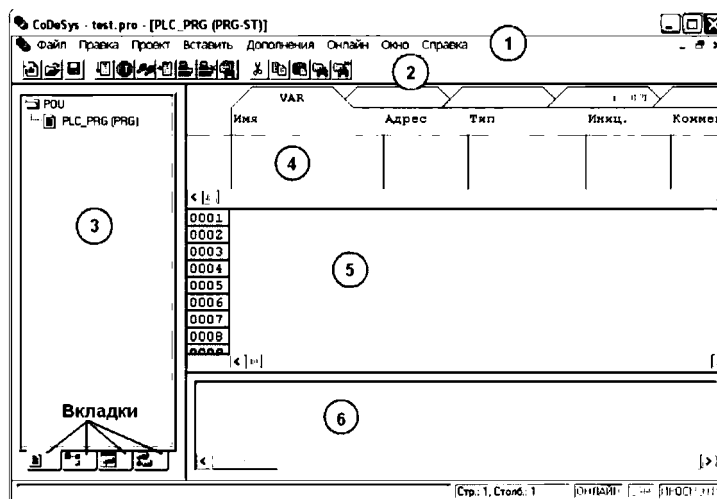


Рисунок 30 – Главное окно CoDeSys

1. Главное меню. Практически все команды можно выполнить через меню.

2. Панель инструментов. Дублирует основные команды меню.

3. Организатор объектов. Здесь отображаются и создаются все объекты проекта, такие как программные модули, визуализации и т.п. В нижней части организатора объектов можно переключаться между его страницами (вкладками). Всего имеется четыре вкладки:

а. POU – список программных модулей;

б. Типы данных;

в. Визуализации;

г. Ресурсы – глобальные переменные, конфигуратор задач, настройки целевой платформы и т.п.

4. Окно объявления переменных. В вашем случае оно будет выглядеть, скорее всего, по-другому, то есть переменные будут отображаться в виде списка. Чтобы переменные отображались в виде таблицы, необходимо изменить настройки программы следующим образом: в организаторе объектов перейти на вкладку РЕСУРСЫ, выбрать там раздел РАБОЧАЯ ОБЛАСТЬ, в появившемся окне выбрать категорию РЕДАКТОР, установить флажок напротив надписи ОБЪЯВЛЕНИЯ ТАБЛИЦЕЙ и нажать кнопку ОК.

5. Редактор исходного кода. Здесь и будем писать программу.

6. Окно сообщений. Здесь выводятся сообщения о ходе компиляции.

Для объявления глобальных переменных перейдем на вкладку РЕСУРСЫ, дважды щёлкнем

по разделу ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ, затем также двойным щелчком выберем раздел Global_Variables.

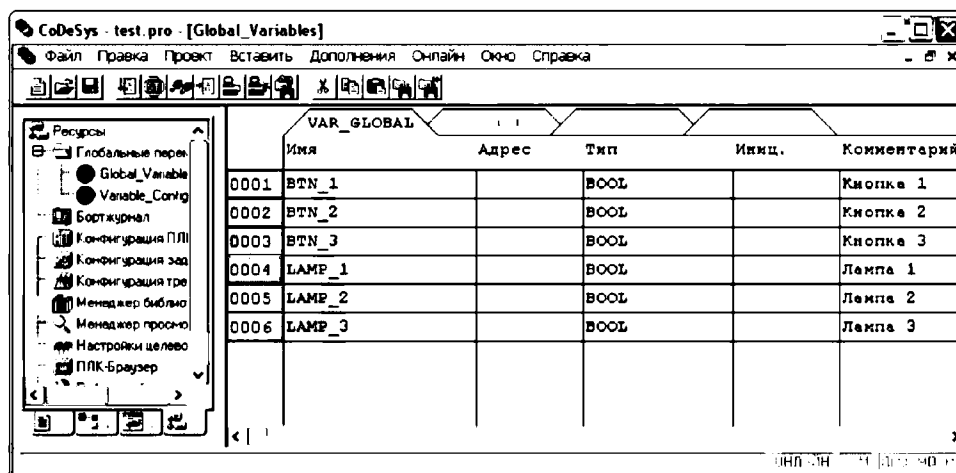


Рисунок 31 – Глобальные переменные

В верхней части таблицы расположены четыре вкладки (рисунок 31), имена которых соответствуют типу объявляемых данных:

1. VAR_GLOBAL – глобальные переменные;
2. CONSTANT – глобальные константы;
3. RETAIN – глобальные переменные, хранящиеся в ПЗУ контроллера.

Если вы объявите переменную в этом разделе, то её значение будет сохранено даже после выключения питания контроллера (правда, для этого нужно иметь контроллер);

4. INFO – информация о программе.

Каждая вкладка таблицы (кроме INFO) имеет следующие столбцы:

1. ИМЯ – идентификатор переменной;
2. АДРЕС – адрес переменной. За исключением случаев связывания переменных с входами/выходами ПЛК адрес указывать нет необходимости. Принципы адресации в данной книге не рассматриваются;

3. ТИП – тип данных, к которому принадлежит переменная;

4. ИНИЦИАЛИЗАЦИЯ – значение, которое будет присвоено переменной при старте программы ПЛК. Это не относится к переменным, хранящимся в ПЗУ, если только программа не работает в режиме эмуляции;

5. КОММЕНТАРИЙ – комментарий.

Чтобы добавить в таблицу переменную, нужно щёлкнуть правой кнопкой по таблице и выбрать команду НОВОЕ ОБЪЯВЛЕНИЕ. В таблице появится новая строка со значениями по умолчанию. Чтобы изменить какое-либо значение, нужно щёлкнуть по нужной ячейке таблицы и ввести с клавиатуры новое значение.

Все наши переменные будут иметь тип BOOL – логический тип, эквивалентный типу Boolean в Паскале. После ввода всех переменных, у вас должно получиться примерно так, как показано на рисунке 31.

Теперь можно переходить к написанию исходного кода программы.

1.6 Запуск OPC сервера и его настройка

Аббревиатура OPC (сейчас – это Open Platform Communications, взаимодействие открытых платформ) ранее означала OLE for Process Control, OLE для управления процессами.[7]

OPC использует технологию COM/DCOM для решения задачи обмена данными в системах промышленной автоматизации. OPC-клиент (любое приложение, например SCADA-система), вызывая определенные функции объекта OPC-сервера, подписывается на получение определенных данных с определенной частотой. OPC-сервер, получив каким-то образом эти данные (чаще всего, опросив оборудование), вызывает известные функции клиента и вручает ему данные. Таким образом, используются как прямые COM-вызовы (от клиента к серверу), так и обратные (callback, от сервера к клиенту).

OPC-серверы конкретных аппаратных устройств поставляются многими производителями аппаратуры. Связь сервера с аппаратурой может осуществляться через какой-либо физический интерфейс компьютера: последовательный порт, USB, Ethernet, плату расширения с выходом на промышленную сеть и т.д. Для приложения-клиента параметры физического

подключения совершенно не интересны, поскольку при настройке соединения с сервером они не задействуются (они используются только при первоначальном конфигурировании самого сервера). Настройка клиента сводится к выбору нужного сервера из списка зарегистрированных в системе, соединения с ним и выбору переменных для чтения и записи из предоставляемого сервером списка переменных. Таким образом, OPC-сервер создает абстракцию аппаратуры, позволяя любому OPC-клиенту записывать и считывать данные с устройства.

Состав и схема взаимодействия программ-элементов имитационной системы показана на рисунке 32.

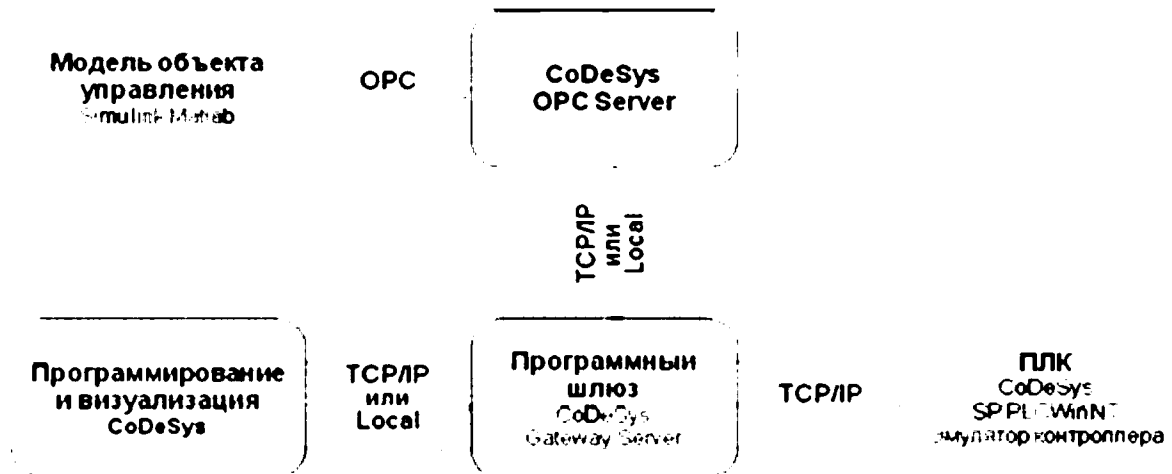


Рисунок 32 – Схема взаимодействия программ

Объект управления представлен своей имитационной Simulink-моделью. Говоря упрощенно, модель «пересчитывает» входные (управляющие) сигналы в выходные, несущие информацию о состоянии объекта. Для ввода-вывода используются специальные блоки из пакета OPC Toolbox, наличие которых в Simulink-диаграмме автоматически обеспечивает «работу» модели в реальном времени.

Частота пересчета модели, конечно, не совпадает с частотой обмена по протоколу OPC, она существенно выше, что позволяет объекту изменять свое состояние и в период между обменами. Однако блоки препятствуют тому, чтобы модельное время изменялось быстрее, чем реальное, как это обычно имеет место в Simulink-моделях. Конечное время расчета в наших системах

мы всегда будем устанавливать равным бесконечности (в Matlab за бесконечность «отвечает» константа $\text{inf} - \text{infinity}$, бесконечность). Таким образом, остановить расчет можно будет только вручную.

Управляющая программа выполняется на программном эмуляторе ПЛК SP PLC WinNT, входящем в состав пакета программ CoDeSys. Необходимость в «отдельном» программном эмуляторе состоит в следующем. Конечно, сама среда программирования CoDeSys дает возможность запуска открытых в ней проектов без подключения к ПЛК в режиме симуляции (Simulation Mode). Однако в данном режиме «общаться» с запущенной программой, т.е. задавать входные и наблюдать выходные переменные, можно только из среды программирования. В этом случае следует забыть об использовании, каких бы то ни было внешних программ и всю имитацию АСУ ТП внедрить непосредственно в проект CoDeSys. В принципе такой подход вполне имеет право на существование, поскольку CoDeSys предоставляет полный набор возможностей для его реализации: можно «оформить» программную модель объекта в виде одной или нескольких программных единиц (программ или функциональных блоков), а для имитации человеко-машинного интерфейса задействовать достаточно развитые средства визуализации самой системы CoDeSys. Но, на наш взгляд, такой «бюджетный» вариант имитационной системы все же проигрывает варианту с использованием разных программ.

Подробно последовательность действий, осуществляемых при подключении контроллеров системы CoDeSys через OPC-сервер к компьютеру, описана в [7]. Ниже перечислены шаги, которые необходимо сделать для организации обмена с виртуальным контроллером SP PLCWinNT V2.4.

1. Создать программу (проект) в среде CoDeSys с целевой платформой 3S CoDeSys SP PLCWinNT V2.4. При настройке целевой платформы следует установить параметр Download symbol file вкладки General (рисунок 33).

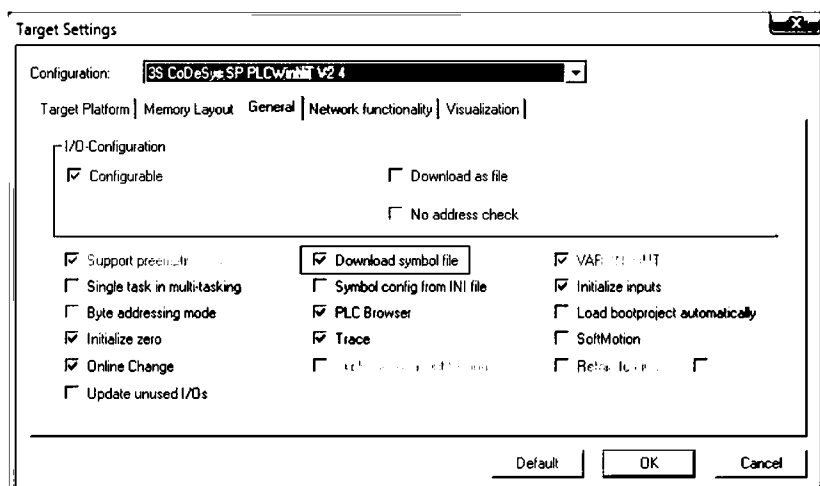


Рисунок 33 – Настройка целевой платформы

2. Объявить переменные для обмена по OPC. В общем случае такие переменные могут быть объявлены в любой из программ, однако для упрощения доступа лучше все их сделать глобальными.

3. Написать программу PLC_PRG. В программе должен быть, по меньшей мере, один оператор, поскольку без этого она не компилируется.

4. Сохранить проект под осмысленным именем в отдельную папку.

5. Запустить PLCWinNT (Пуск □ Все программы □ 3S Software □ CoDeSys SP PLCWinNT □ CoDeSys SP PLCWinNT V2.4) (рисунок 34), установить связь и загрузить программу в «контроллер».

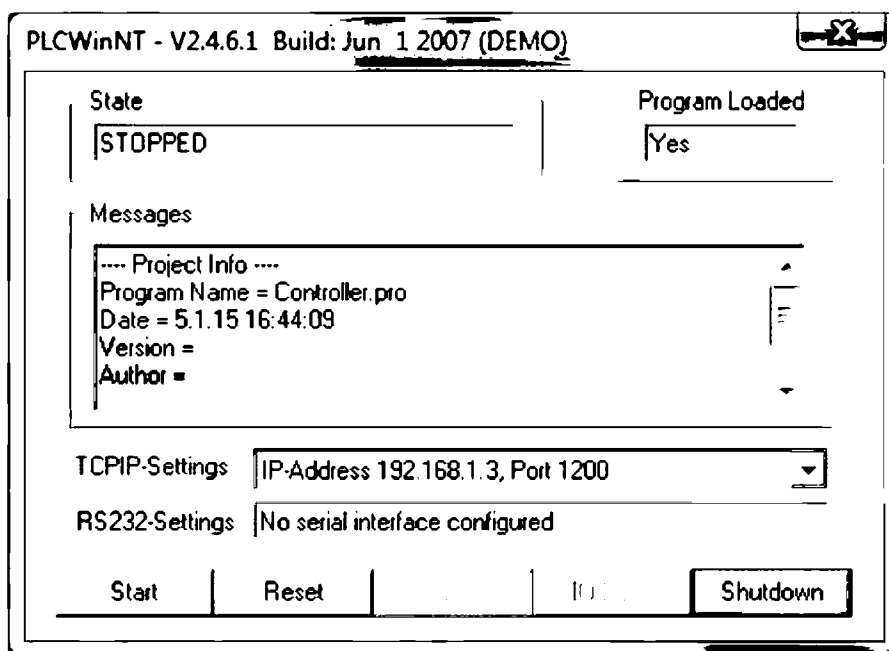


Рисунок 34 – Окно PLCWinNT

В случае если связь с контроллером установить не удалось, следует настроить коммуникационные параметры в меню Online (рисунке 35).

6. Отключить CoDeSys от PLCWinNT и перейти в Опции (Options) в меню Project. Выбрать Symbol Configuration и установить галочку Dump symbol entries (рисунок 36).

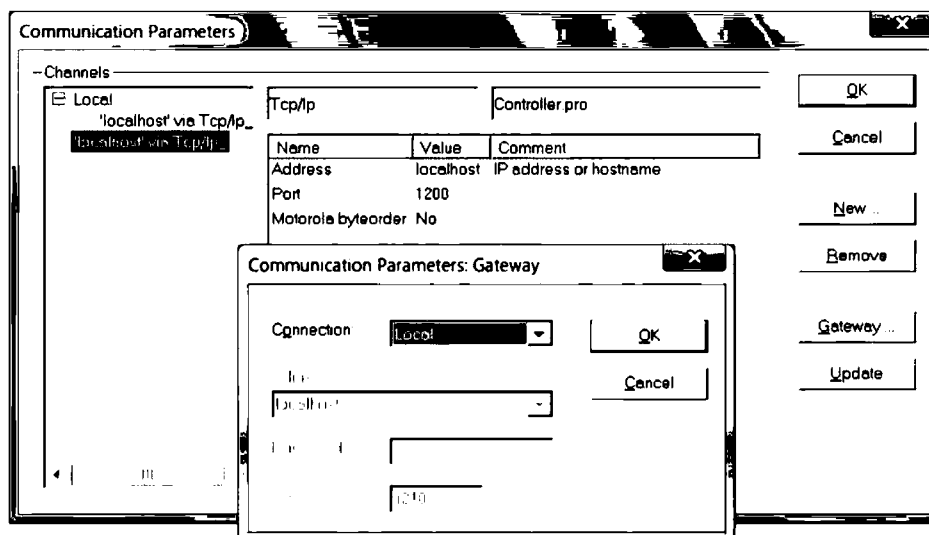


Рисунок 35 – Настройка коммутационных параметров

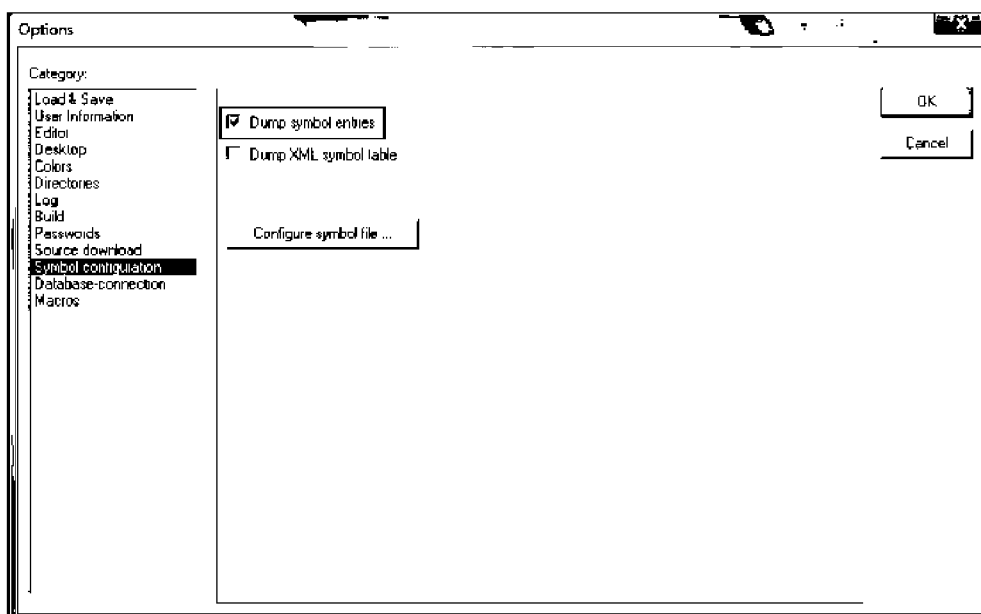


Рисунок 36 – Установка опций

7. Сконфигурировать «символьный файл», выбрав переменные для обмена по OPC (рисунок 37).

8. Настроить параметры OPC-сервера. Для этого необходимо запустить конфигуратор OPC-сервера (Пуск □ Все программы □ 3S Software □

Communication □ CoDeSys OPC Configurator). В окне конфигуратора требуется добавить PLC (Append PLC) и настроить соединение (Connection). По существу достаточно выбрать из списка соединение, настроенное в проекте CoDeSys (рисунок 38).

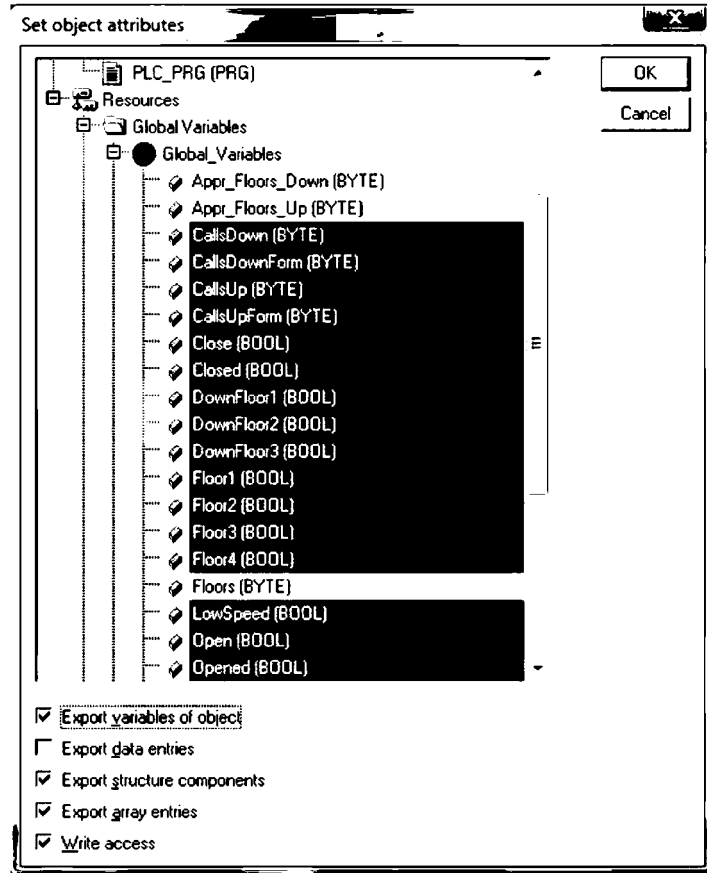


Рисунок 37 – Выбор переменных для обмена по OPC

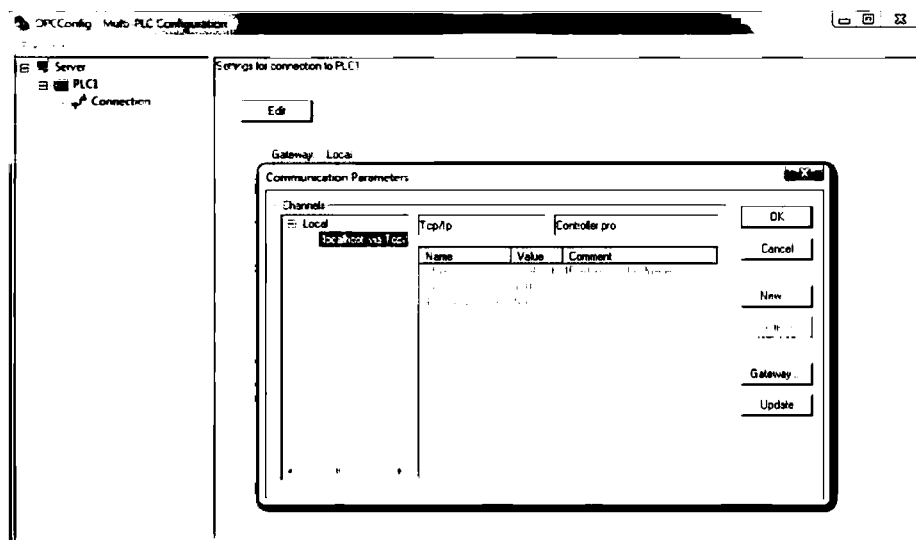


Рисунок 38 – Настройка OPC сервера

9. «Перестроить» программу (проект) CoDeSys, включив в нее все изменения, сделанные после загрузки в контроллер. Для этого требуется вызвать команду Clean all из меню Project, затем команду Rebuild all из того же меню. CoDeSys перекомпилирует программу и перезагрузит проект при следующем подключении к ПЛК.

10. Подключиться к PLCWinNT, согласившись на перезагрузку проекта. С помощью любой программы OPC-клиента, проверить доступность OPC сервера CoDeSys и переменных для обмена. Рекомендации по решению возможных проблем приведены в предыдущем пункте.

2 ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ ДВИЖЕНИЕМ КОЗЛОВОГО КРАНА

В качестве примера создадим имитационную модель козлового крана (рисунок 39).



Рисунок 39 – Козловой кран

Козловой кран (КК) – многофункциональное оборудование для различных площадок и объектов, где грузоподъемные работы ведутся постоянно. Козловой кран – это агрегат мостового типа. Горизонтальная балка, по которой перемещается груз – конструктивно относится к мостовой группе подъемных механизмов. Основным отличием является то, что КК может сам передвигаться по рельсам. Усредненный срок эксплуатации козловых кранов – около 20 лет, наработка на отказ $\approx 3\ 000$ циклов. Козловой кран состоит из следующих элементов:

1. Ригель (ферма, балка) – пролетное строение. Мост легких и средних КК – одноблочный, тяжелых – двухблочный;

Балка бывает решетчатая или сплошностенная разного сечения. Она может иметь 1 или 2 консоли, или не иметь их совсем (консольные и бесконсольные краны);

2. Опоры – жесткие, пространственные, гибкие или плоские. Если пролеты < 25 м, то все опоры – жесткие, внизу каждой зафиксированы ходовые тележки. При больших расстояниях между рельсами – одна нога

фиксируется жестко, а 2-я – шарнирно. Каждая опора включает 1 или 2 стойки;

3. Крановые (грузовые) тележки (или электроталь), оборудованные грузоподъемным механизмом, на который крепится груз. Грузовые тележки бывают монорельсовыми или двухрельсовыми. Двухрельсовые тележки передвигаются по обоим поясам – верхнему и нижнему, а однорельсовые – только по нижнему;

4. Электрооборудование. Энергоснабжение КК – от внешней сети по троллеям или с помощью гибкого кабеля. Кабель наматывается на кабельный барабан, зафиксированный на одной из опор крана;

5. Подкрановые рельсовые пути – наземные, рельсы Р43/Р24 (Р50/Р65) устанавливаются на деревянные (ж/б) шпалы на гравийной подушке. Рельсы должны быть с заземлением и тупиковыми упорами. Движение происходит благодаря двухребордным колесам, установленным на опорах. Размер колеи равен пролету крана. Помимо конструктивных элементов в устройство козлового крана входят механизмы передвижения всего крана и тележки, грузоподъемный механизм. Управлять подъемом и перемещением грузов КК можно с земли или из кабины.

Электрический способ – из кабины крановщика. С высоты четко видно груз любых размеров, кабина может быть оборудована бортовым компьютером для более точного размещения грузов.

2.1 Создание 3D-модели крана

Для того чтобы создать 3D-модель козлового крана, необходимо определить его основные подвижные части. Всего их получилось три: ноги крана (рисунок 40), кабина (рисунок 41) и крюк (рисунок 42). Для каждой из них в SolidWorks создаем модель и после собираем их вместе в сборке.

Модель ног крана будет перемещаться по оси Z и вместе с ней все остальные части.

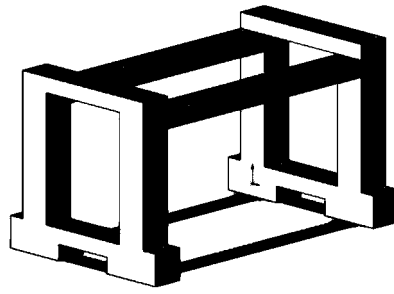


Рисунок 40 – Трехмерная модель ног крана

Так как кабина будет расположена на ногах крана, ее перемещение осуществляться по двум осям X и Z .

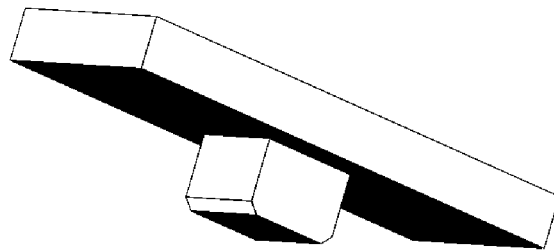


Рисунок 41 – Трехмерная модель кабины крана

Перемещение крюка в свою очередь осуществляется по всем осям, так как его положение вплотную зависит от перемещений ног и кабины крана.

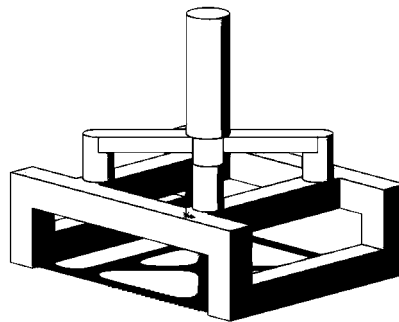


Рисунок 42 – Трехмерная модель крюк

Помимо основных элементов нам необходимо какое-то основание (рисунок 43), по которому кран будет совершать движение. В качестве его возьмем плоскую поверхность и расположим на ней рельсы, по которым будет двигаться кран.

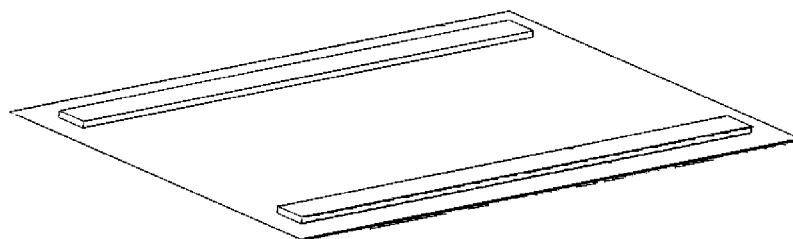


Рисунок 43 – Модель основания

Для демонстрации работы крана создадим 3D-модель контейнера (рисунок 44), играющего роль груза, который перемещает кран с одного места на другое.

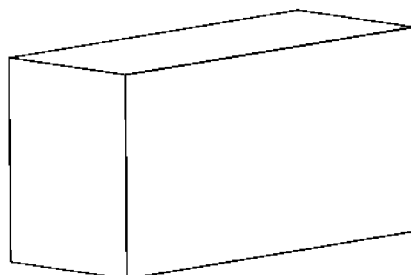


Рисунок 44 - 3D-модель контейнера

Собрав все элементы вместе, получим полную модель козлового крана (рисунок 45), которую экспортируем с помощью Simscape Multibody Link в MatLab Simulink.

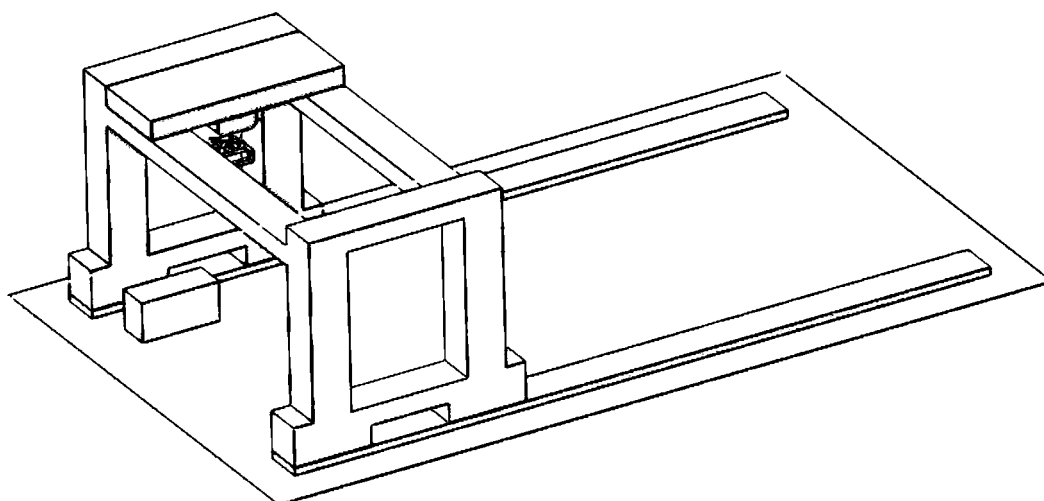


Рисунок 45 – Трехмерная модель козлового крана

2.2 Создание Simulink модели

После переноса трехмерной модели из SolidWorks в Simulink, была получена модель которую нужно дорабатывать для того чтобы ей можно было задавать позиции объектам системы (тем самым управлять ею). Готовая система изображена на рисунке 46.

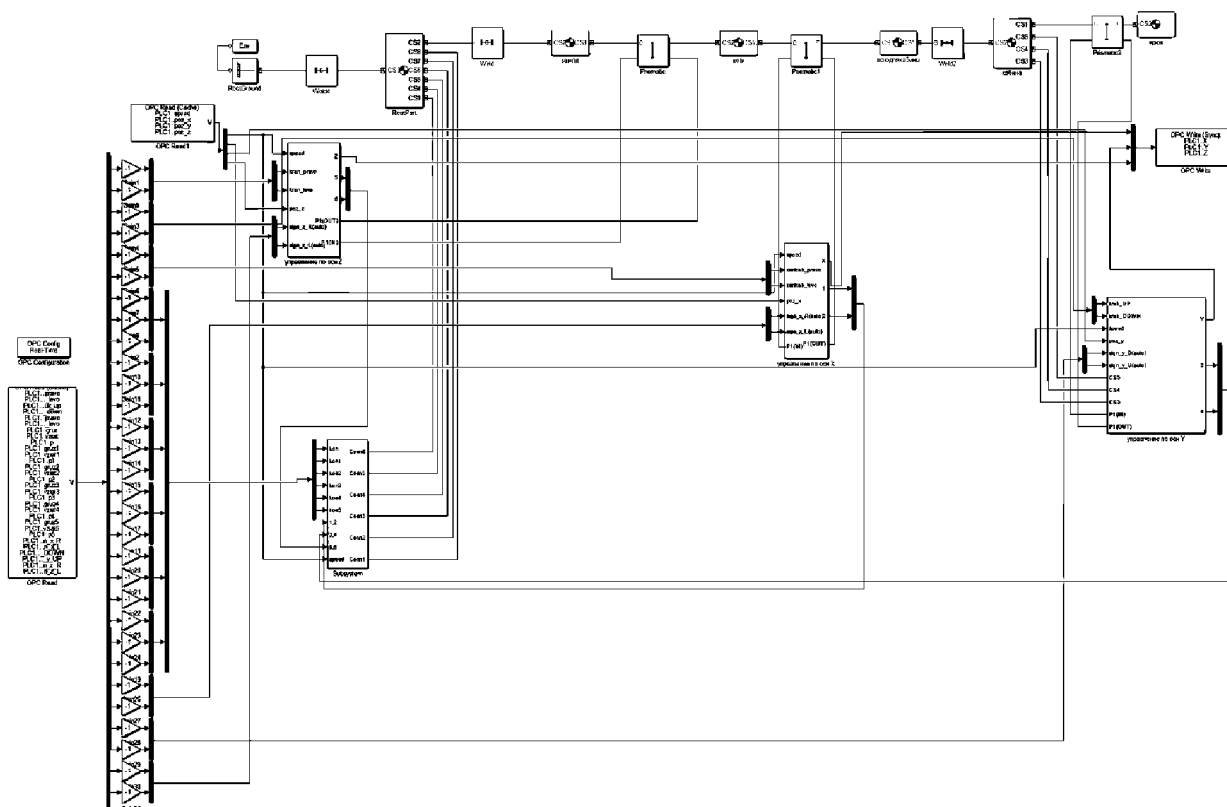


Рисунок 46 – Simulink модель

Саму систему можно поделить на четыре основные части, такие как: блок управление по оси X; блок управления по оси Y; блок управления по оси Z; блок задания положения контейнерам (Subsystem).

2.2.1 Основные блоки и их описание

Все блоки между собой практически идентичны. Некоторые блоки плотно взаимодействуют друг с другом, но при этом у каждого своя задача и объект управления.

Блок управление по оси X (рисунок 47) отвечает за передачу положения основанию кабины крана, которое совершает движение по оси X.

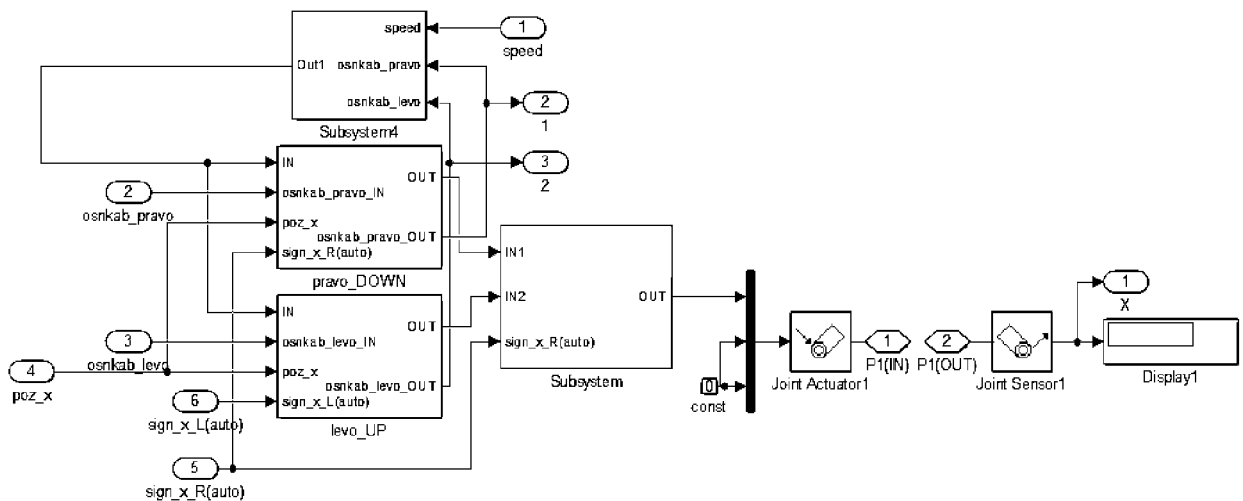


Рисунок 47 – Блок управление по оси X

Состоит он из следующих элементов: Subsystem4 (рисунок 48) – в данном блоке в основе лежит интегратор, в котором увеличивается или уменьшается число (в зависимости от того какой сигнал был подан, osnkab_pravo или osnkab_levo), тем самым это число будет служить позицией для основания кабины по оси X; pravo_DOWN (рисунок 49), levo_UP (рисунок 50) и Subsystem (рисунок 51) – служат для того чтобы можно было задать положение и занять именно его, независимо от скорости.

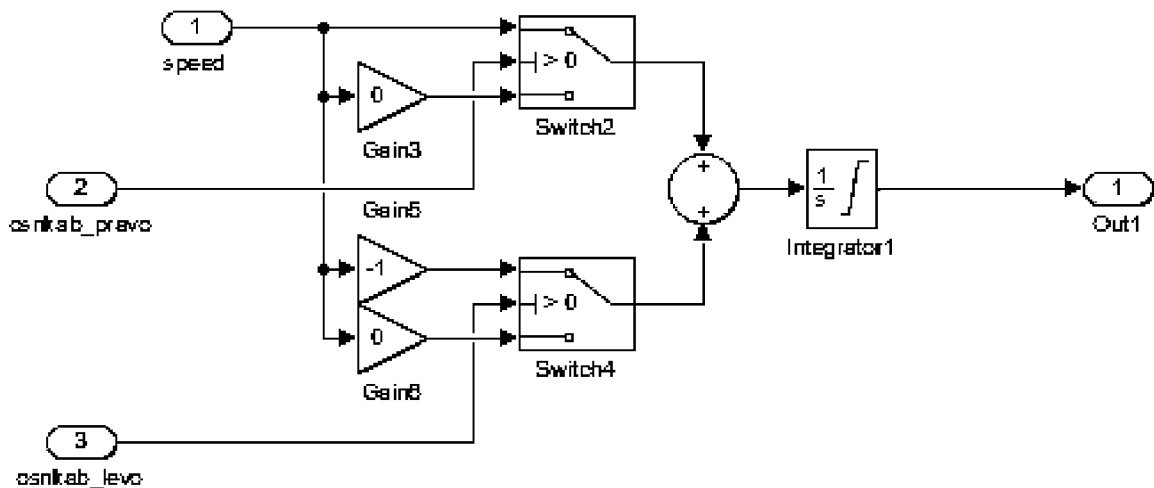


Рисунок 48 – Блок Subsystem4

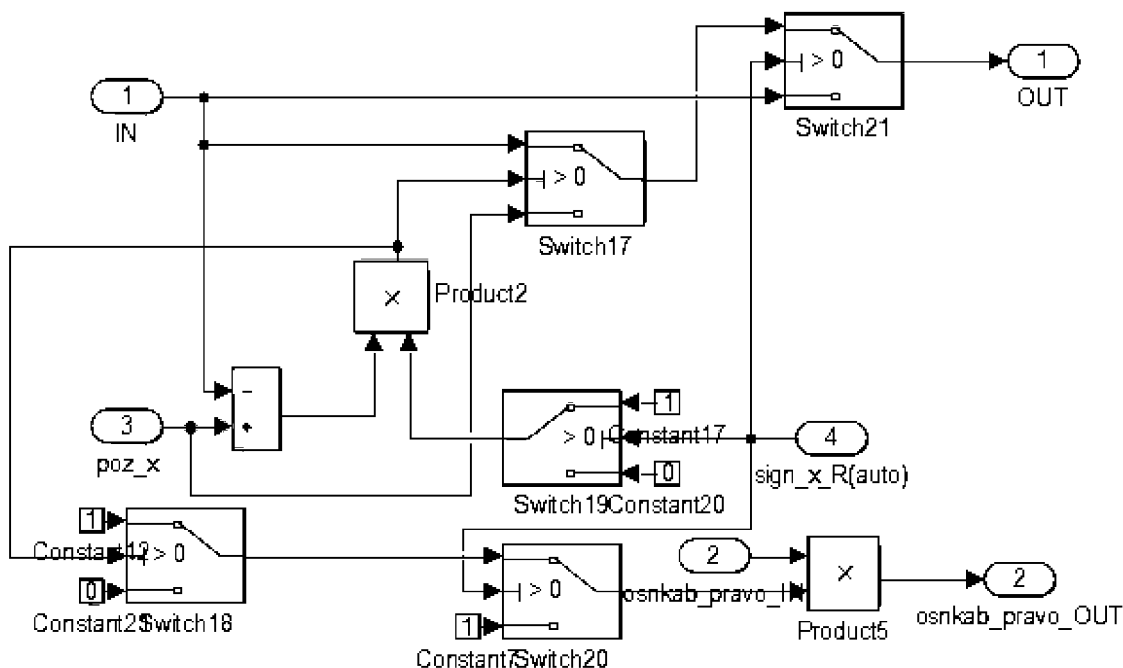


Рисунок 49 – Блок pravo_DOWN

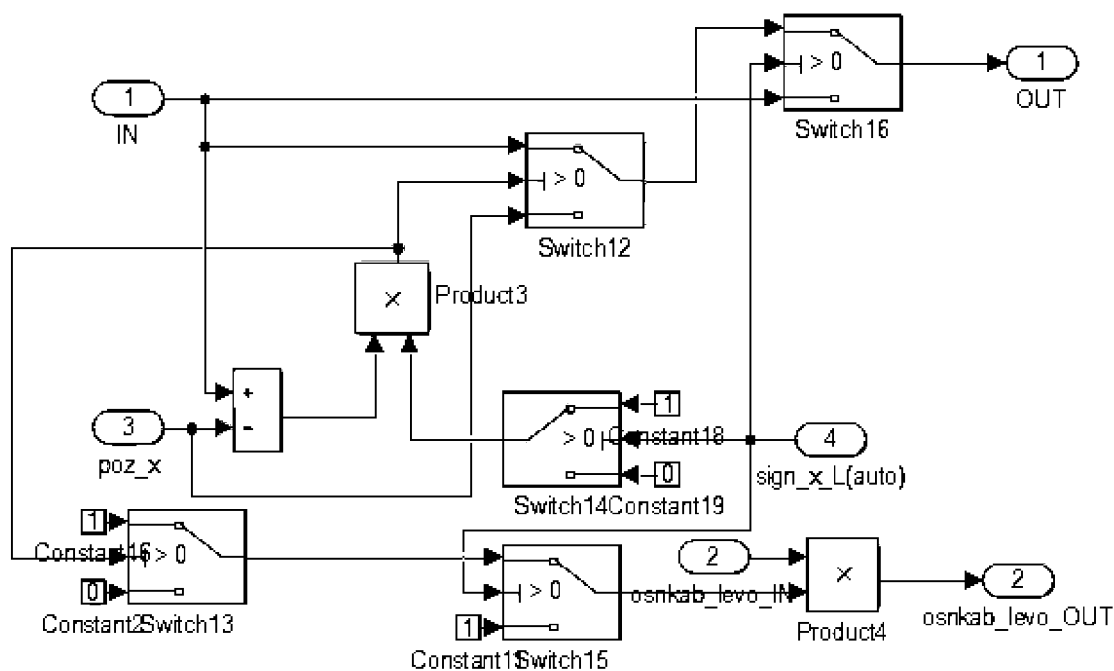


Рисунок 50 – Блок levo_UP

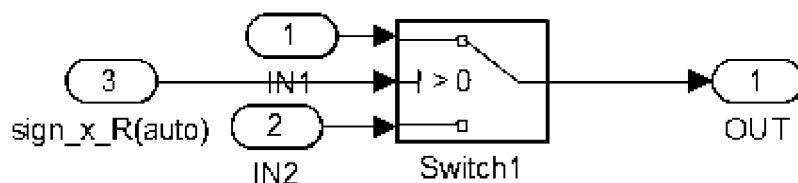


Рисунок 51 – Блок Subsystem

Аналогично устроены и остальные элементы системы, разница лишь в том, что блок управления по оси Y отвечает за крюк, блок управления по оси

Z отвечает за основание крана, а блок задания положением контейнерам (Subsystem) отвечает за управление положением контейнеров в пространстве.

2.3 Создание управляющей программы

После того как разработали Simulink модель крана, переходим к разработке управляющей программы:

- 1) Описание визуализации панели управления движением модели.

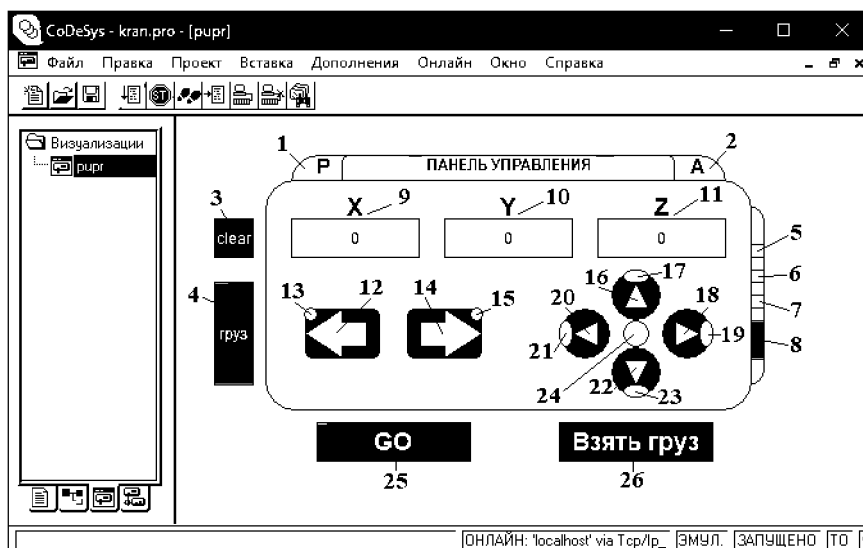


Рисунок 52 – Панель ручного и автоматического управления

- 1 – кнопка ручного режима управления;
- 2 – кнопка автоматического режима управления;
- 3 – кнопка убрать все контейнеры;
- 4 – кнопка появления груза;
- 5 – кнопка включения высокой скорости движения модели;
- 6 – кнопка включения средней скорости движения модели;
- 7 – кнопка включения обычной скорости движения модели;
- 8 – кнопка задания начального положения;
- 9 – координаты крана по оси X;
- 10 – координаты крана по оси Y;
- 11 – координаты крана по оси Z;
- 12 – кнопка управления краном по оси Z (основание крана) влево;
- 13 – кнопка фиксации кнопки 12;
- 14 – кнопка управления краном по оси Z (основание крана) вправо;

- 15 – кнопка фиксации кнопки 14;
 - 16 – кнопка управления краном по оси Y (крюк крана) вверх;
 - 17 – кнопка фиксации кнопки 16;
 - 18 – кнопка управления краном по оси Y (крюк крана) вниз;
 - 19 – кнопка фиксации кнопки 18;
 - 20 – кнопка управления краном по оси X (кабина крана) влево;
 - 21 – кнопка фиксации кнопки 20;
 - 22 – кнопка управления краном по оси X (кабина крана) вправо;
 - 23 – кнопка фиксации кнопки 22;
 - 24 – кнопка задания позиции для взятия груза;
 - 25 – кнопка начала автоматической работы;
 - 26 – кнопка захвата груза крюком.
- 2) Описание программы PLC_PRG(ST):

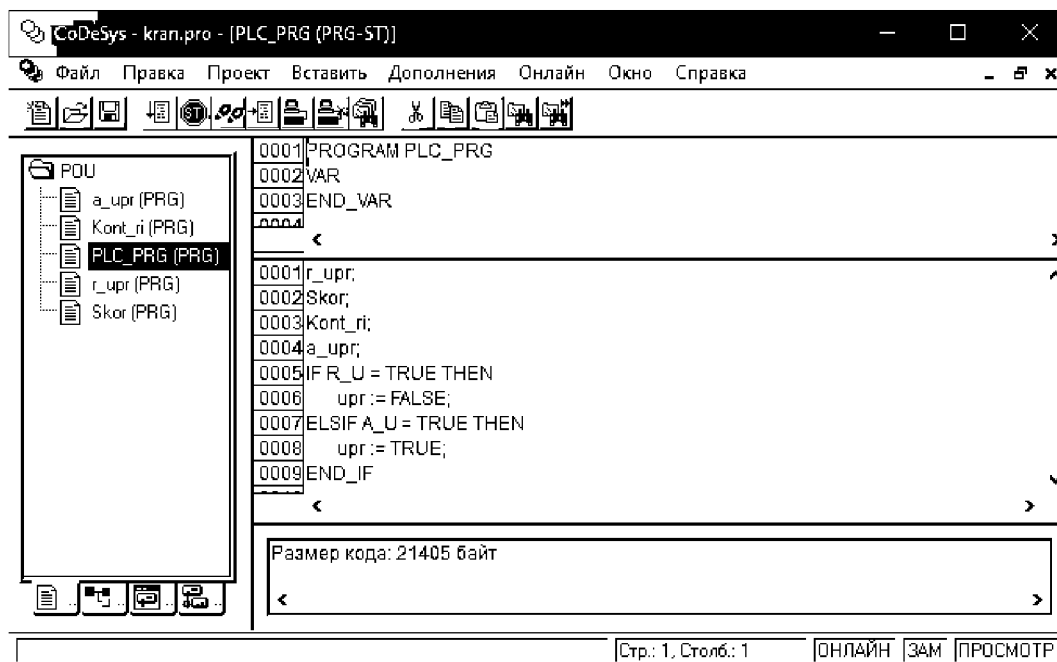
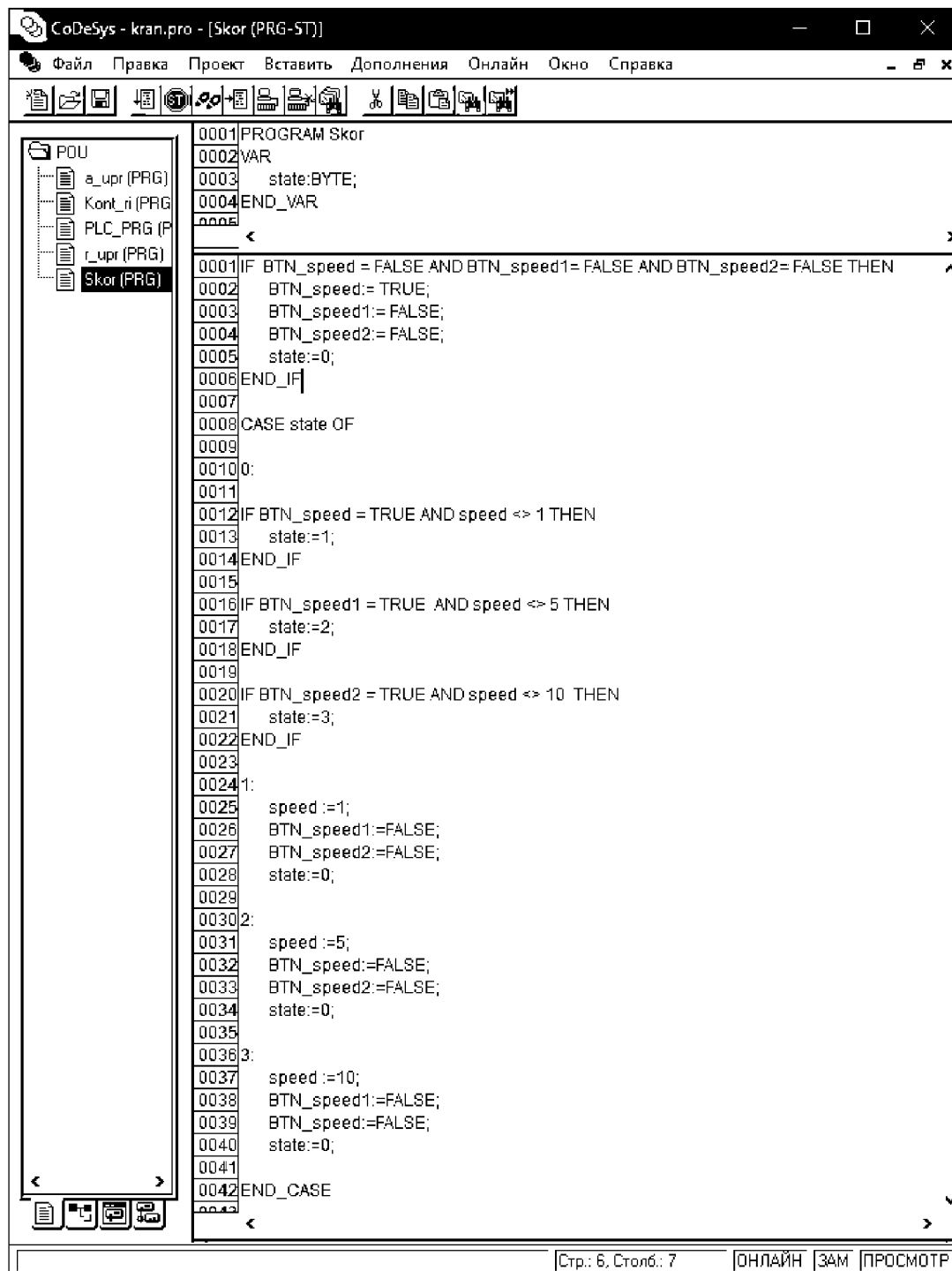


Рисунок 53 – Основная программа PLC_PRG

В данной программе осуществляется запуск подпрограмм для работы системы и реализована работа кнопок для переключения между ручным и автоматическим режимом управления.

3) Описание подпрограммы Skor (ST):

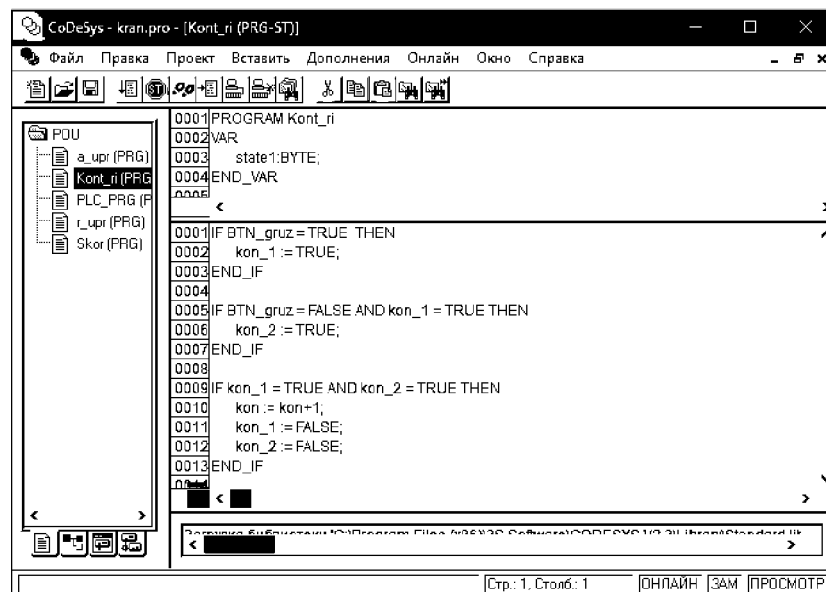


```
0001 PROGRAM Skor
0002 VAR
0003   state:BYTE;
0004 END_VAR
0005
0006 <
0007 >
0008
0009 IF BTN_speed = FALSE AND BTN_speed1= FALSE AND BTN_speed2= FALSE THEN
0010   BTN_speed:= TRUE;
0011   BTN_speed1:= FALSE;
0012   BTN_speed2:= FALSE;
0013   state:=0;
0014 END_IF
0015
0016 CASE state OF
0017 0:
0018   IF BTN_speed = TRUE AND speed <= 1 THEN
0019     state:=1;
0020   END_IF
0021   IF BTN_speed1 = TRUE AND speed <= 5 THEN
0022     state:=2;
0023   END_IF
0024   IF BTN_speed2 = TRUE AND speed <= 10 THEN
0025     state:=3;
0026   END_IF
0027 1:
0028   speed :=1;
0029   BTN_speed1:=FALSE;
0030   BTN_speed2:=FALSE;
0031   state:=0;
0032 2:
0033   speed :=5;
0034   BTN_speed:=FALSE;
0035   BTN_speed2:=FALSE;
0036   state:=0;
0037 3:
0038   speed :=10;
0039   BTN_speed1:=FALSE;
0040   BTN_speed:=FALSE;
0041   state:=0;
0042 END_CASE
0043 <
0044 >
```

Рисунок 54 – Код подпрограммы Skor

В данной подпрограмме реализован выбор скорости движения объектов системы. По главному условие если ни одна из кнопок не была нажата, то включается минимальный режим скорости. А дальше в скорость зависит от того какая кнопка была нажата (малая скорость, средняя или большая).

4) Описание подпрограммы Kont_ri (ST):

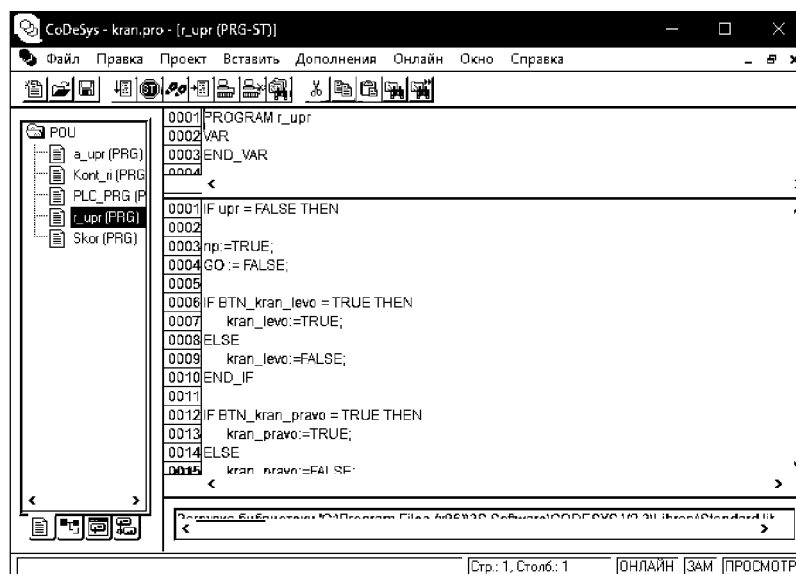


```
0001 PROGRAM Kont_ri
0002 VAR
0003   state1:BYTE;
0004 END_VAR
0005
0006
0007
0008
0009 IF BTN_gruz = TRUE THEN
0010   kon_1 := TRUE;
0011 END_IF
0012
0013 IF BTN_gruz = FALSE AND kon_1 = TRUE THEN
0014   kon_2 := TRUE;
0015 END_IF
0016
0017 IF kon_1 = TRUE AND kon_2 = TRUE THEN
0018   kon := kon+1;
0019   kon_1 := FALSE;
0020   kon_2 := FALSE;
0021 END_IF
0022
0023
0024
```

Рисунок 55 – Подпрограммы Kont_ri (полный код подпрограммы в приложении Б)

Данная подпрограмма отвечает за появление груза и его взаимодействие с краном, а именно подбор контейнера. В нашем случае контейнеров шесть, каждый имеет свой номер. По условию контейнер можно подобрать только в том случае, когда крюк находится прямо над ним и касается его сверху.

5) Описание подпрограммы r_upr (ST):

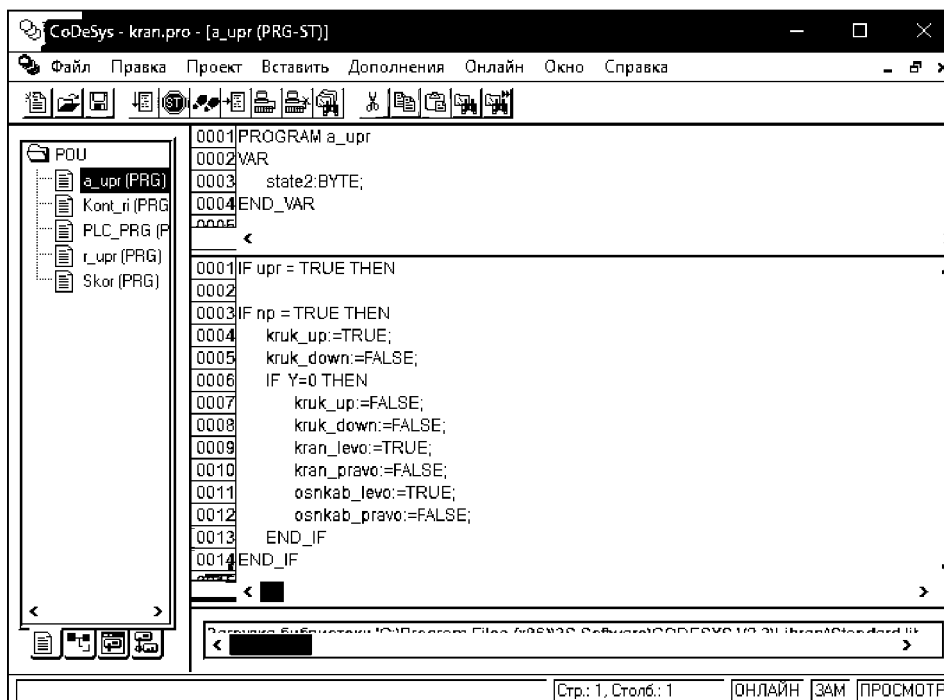


```
0001 PROGRAM r_upr
0002 VAR
0003 END_VAR
0004
0005
0006
0007
0008 IF upr = FALSE THEN
0009   nr := TRUE;
0010   GO := FALSE;
0011
0012 IF BTN_kran_leva = TRUE THEN
0013   kran_leva := TRUE;
0014 ELSE
0015   kran_leva := FALSE;
0016 END_IF
0017
0018 IF BTN_kran_prava = TRUE THEN
0019   kran_prava := TRUE;
0020 ELSE
0021   kran_prava := FALSE;
0022
0023
```

Рисунок 56 – Подпрограмма r_upr (полный код подпрограммы в приложении Б)

В данной подпрограмме реализовано ручное управление системой, если данный режим выбран. С его помощью можно лично направить кран в то место, которое необходимо, а также подобрать груз и поставить его там, где необходимо. Так же эта подпрограмма содержит код для управления кнопками задания начального положения и задания положения для взятия груза.

6) Подпрограмма a_upr (ST):



```
0001 PROGRAM a_upr
0002 VAR
0003   state2:BYTE;
0004 END_VAR
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
END_PROGRAM
```

Рисунок 57 – Подпрограммы a_upr (полный код подпрограммы в приложении Г)

Подпрограмма реализует автоматическое управление системой, если данный режим выбран. В автоматическом режиме кран самостоятельно расставляет все грузы по своим местам при нажатии на кнопку 25 (начало автоматической работы). После нажатия на кнопку (если имеется груз) кран занимает позицию для подъема контейнера, подцепляет его и переносит на определенное ему место, поставив груз на место, он отпускает его и возвращается в начальное положение, в котором ожидает нового груза и повторного нажатия на кнопку.

2.4 Запуск и проверка работы системы

В первую очередь открываем все программы, а именно MatLab и Simulink модель описанную пунктом выше, CoDeSys и его программу для управления движением системы, PC-эмулятор ПЛК SP PLCWinN.

Загружаем программу из CoDeSys в PC-эмулятор ПЛК и запускаем её. В Simulink, во вкладке simulation stop time пишем inf для того чтобы наша модель работала без остановки, после запускаем модель и если все сделано правильно и не возникла проблема с подключением OPC – сервера, то откроется окно с трехмерной моделью нашего крана (рисунок 58).

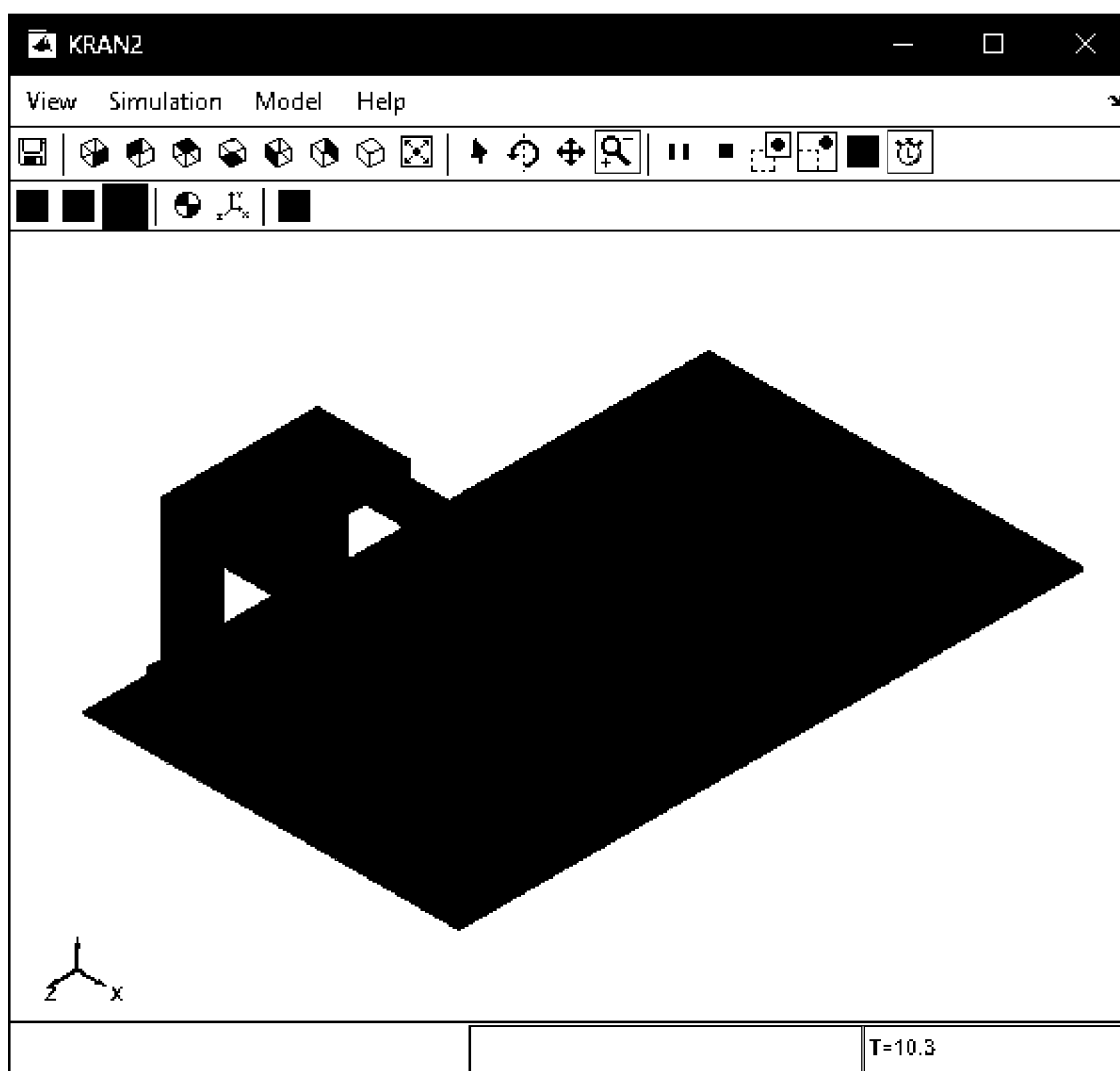


Рисунок 58 – Трехмерная модель козлового крана

Проверим управление движением системы с помощью визуализации в CoDeSys (рисунок 59).

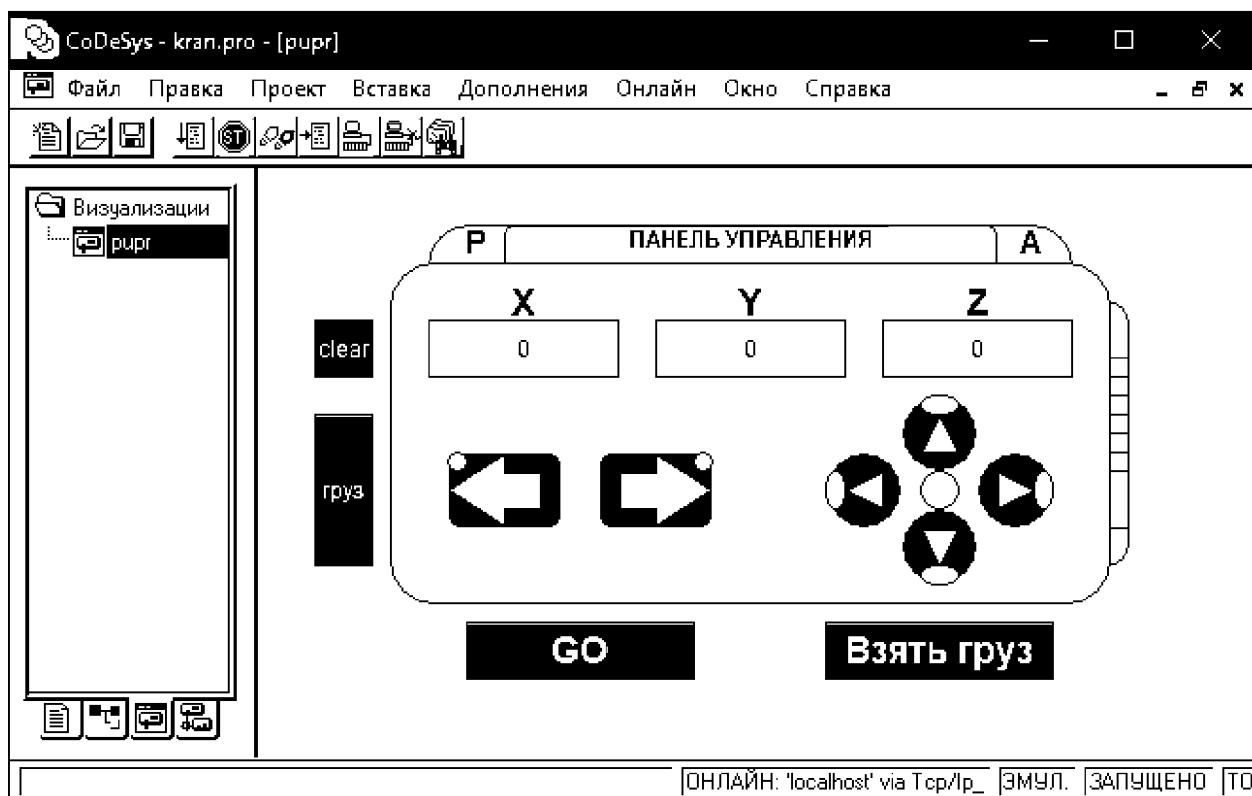


Рисунок 59 – Визуализация панель управления в CoDeSys

Если все кнопки и режимы работают, как задумывалось, то можно сделать вывод что имитационная модель системы управлением движением козлового крана работает правильно.

3.1 Организация рабочего места оператора

Для дипломного проекта было выбрано делать кабинет для одной подгруппы. При расчете количества рабочих мест и их размещении в помещении необходимо учитывать требования, изложенные в СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».[11]

- размеры помещения – 6х7х3,5м;
- количество рабочих мест – не известно;
- ориентация световых проемов – восток;
- коэффициенты отражения стен и потолка – 70%;
- размеры световых проемов – 6х1,5 м.

В соответствие с требованиями СанПиН площадь одного рабочего места с ПВЭМ должно быть не менее 6 м². По выбранным размерам помещения, получается $6 \times 7 = 42$ м² площадь помещения, из этого следует количество рабочих мест 5. Высота потолков 3.5м. Исходя из этих данных, был спроектирован план помещения (см. рис. 60).

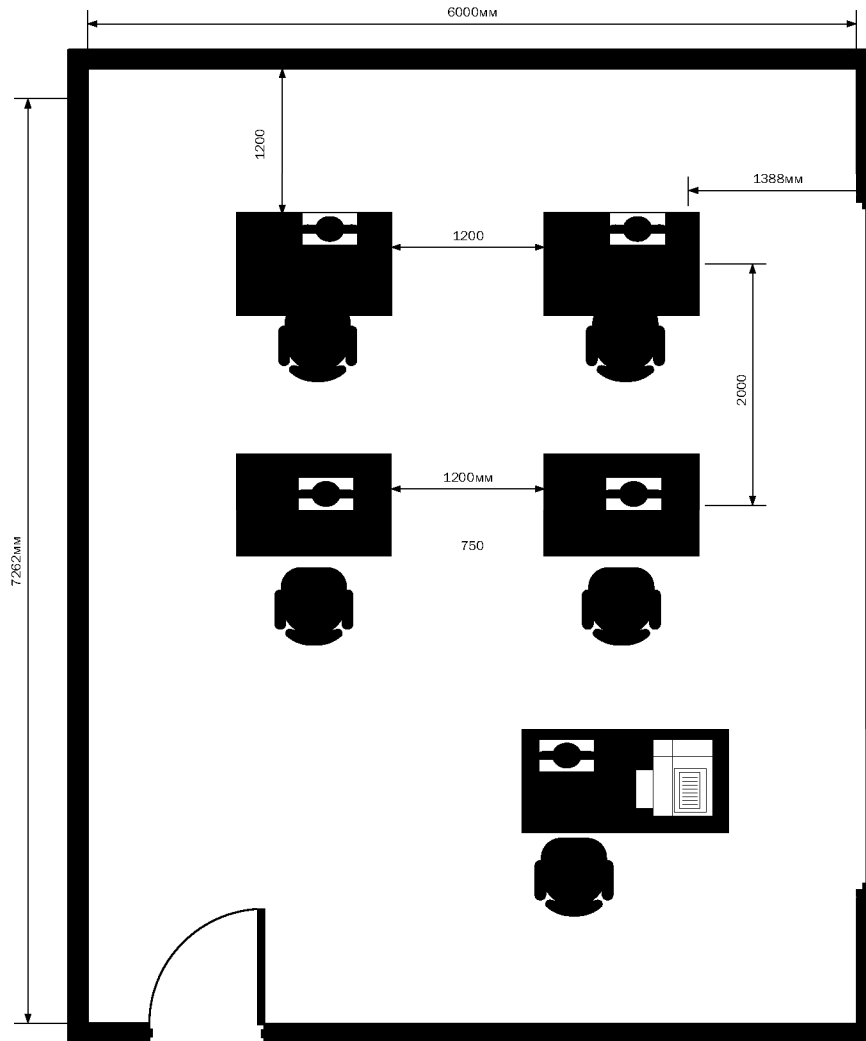


Рисунок 60 – План помещения

3.1.1 Расчет искусственного освещения

Исходные данные:

- высота потолка 3,5 м;
- высота рабочей поверхности 0,8 м.

Расчет новой системы искусственного освещения (реконструкция имеющейся системы искусственного освещения) для создания требуемой освещенности на рабочем месте для данного вида работ требуется выполнение следующих задач:

- 1) Выбор вида системы искусственного освещения. Для нашего случая выбираем общие освещения.

- 2) Выбор источника света. Выбираем люминесцентные лампы.
- 3) Выбор типа светильника их мощности и определение высоты их подвеса над рабочей поверхностью.

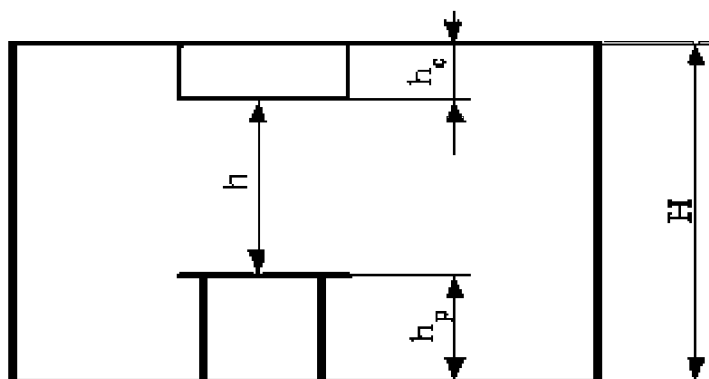


Рисунок 61 – Расчет высоты подвеса над рабочей поверхностью

Был выбран светильник ШОД-2-80 с двумя лапами по 80 Вт. Длина светильника 1530 мм, его ширина составляет 284 мм, а высота 155 мм.

Расчетам высоту подвеса светильника h :

$$h = H - h_p - h_c = 3,5 - 0,8 - 0,155 = 2,545 \text{ м}, \quad (1)$$

где H – высота помещения; h_p – высота рабочей поверхности; h_c – высота до светового центра светильника (в нашем случае возьмем высоту светильника).

4) Определение нормативной освещенности на рабочих местах для заданного вида работ. $E_n = 300 - 500 \text{ Лк}$.

5) Определение коэффициента запаса для данных производственных условий. Для ШОД ламп коэффициент запаса в помещениях с малым выделением пыли равен $K_3 = 1,5$.

6) Выбор рационального расположения светильников и определение необходимого количества.

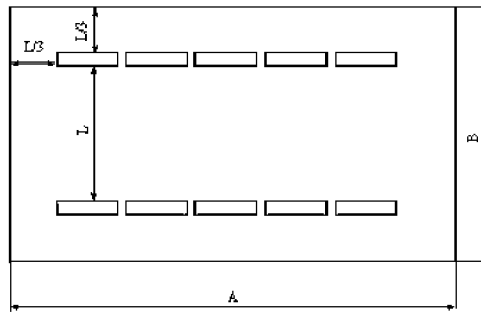


Рисунок 62 – Определения количества светильников и их размещение

Определим расстояние между рядами L:

$$L = \lambda \times h = 1,2 \times 2,545 = 3,054\text{м}, \quad (2)$$

где λ – наивыгоднейшее относительное расстояние между светильниками (справочные данные).

Определим количество светильников в ряду:

$$n_{\text{СВРЯД}} = \frac{A - 2 \frac{L}{3}}{l_{\text{СВ}}} = \frac{7 - 2 \frac{3,054}{3}}{1,53} = 3,2 \text{ шт}, \quad (3)$$

где A – длина помещения; $l_{\text{СВ}}$ – длина светильника.

Количество светильников получилось дробным округлим в меньшую сторону, тогда расстояние между светильниками рассчитаем по формуле:

$$\Delta = \frac{0,xxx \times l_{\text{СВ}}}{n'_{\text{СВРЯД}} - 1} = \frac{0,7 \times 1,53}{3 - 1} = 0,54 \text{ м}, \quad (4)$$

где 0,xxx – отброшенный остаток при округлении до значения $n'_{\text{СВРЯД}}$.

Рассчитаем количество светильников в ряду:

$$n_{\text{ряд}} = \frac{B}{L} = \frac{6}{3,054} = 1,96 \approx 2 \text{ шт}, \quad (5)$$

где B – ширина помещения.

Общее количество светильников:

$$N = n'_{\text{СВРЯД}} \times n_{\text{ряд}} = 3 \times 2 = 6 \text{ шт} \quad (6)$$

По результатам расчета получаем значение:

$$B' = 2 \frac{L}{3} + (n_{\text{ряд}} - 1) \times L + n_{\text{ряд}} \times \text{Ш}_{\text{СВ}}, \quad (7)$$

где $\text{Ш}_{\text{СВ}}$ – ширина светильника.

Подставим значения:

$$B' = 2 \times 1,018 + (2 - 1) \times 3,054 + 2 \times 0,284 = 6,7 \text{ м} \quad (8)$$

Расчетное значения B' не равно ширине помещения B . Чтобы выполнить условие $B = B'$ необходимо изменить размеры L и $L/3$ в пределах $\pm 10\%$.

Для того чтобы определить, на сколько нужно уменьшить или увеличить размеры L и $L/3$, сначала определим, насколько расчетное значение B' отличается от реальной ширины помещения B :

$$\delta = B' - B = 6,7 - 6 = 0,7 \quad (9)$$

Знак перед значением говорит о том, что необходимо сделать с B' увеличить или уменьшить. Значение ΔL , которое необходимо прибавить или вычесть из L , можно определить из выражения:

$$\Delta L = \frac{3\delta}{2+3 \times (n_{\text{ряд}}-1)} = \frac{3 \times 0,7}{2+3 \times (2-1)} = 0,42 \text{ м} \quad (10)$$

Скорректированное значение будет равно $L' = L + \Delta L = 3,474$ и $\frac{L'}{3} = 1,158$.

Итог окончательный план размещения светильников представлен на рисунке 63.

7) Определения расчетного значения светового потока одной лампы:

$$F = \frac{E \times K_3 \times S \times z}{n \times \eta} \quad (11)$$

где E – нормативная (требуемая) освещенность, лк;

K_3 – коэффициент запаса;

S – площадь помещения, м²;

η – коэффициент использования светового потока (в долях единицы);

$z=0,9$ – коэффициент неравномерности освещения;

n – количество ламп.

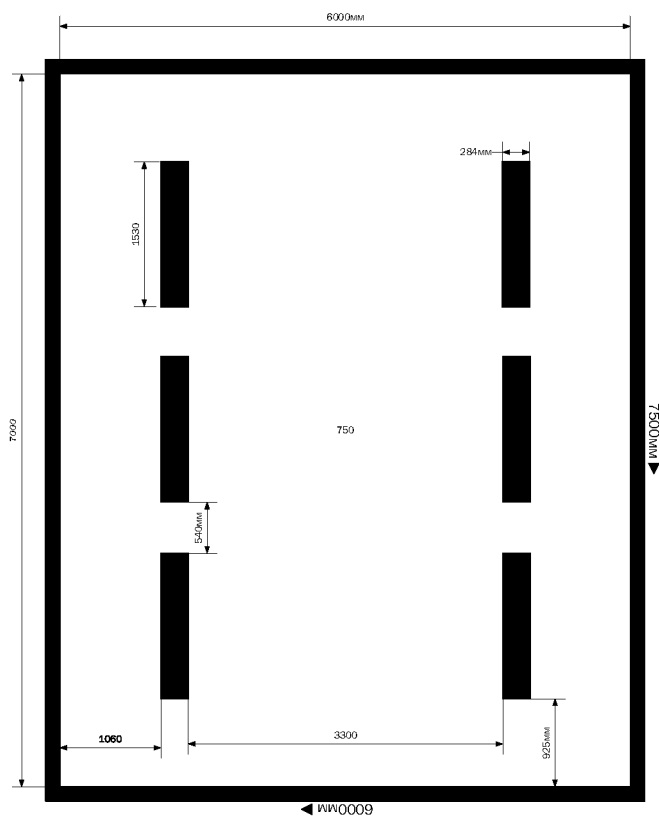


Рисунок 63 – План размещения светильников

Площадь помещения рассчитываем по формуле $S=AB$. Количество ламп в помещении $n = n_{\text{ряд}} n_{\text{свряд}} a$, где $a=2$ количество ламп в светильнике.

$$n = 2 \times 3 \times 2 = 12 \text{ ламп} \quad (12)$$

Для определения коэффициента использования светового потока необходимо знать индекс помещения i , значение коэффициентов отражения стен ρ_c , потолка $\rho_{\text{п}}$ и тип светильника.

$$i = \frac{S}{h(A+B)} = 1,269 \quad (13)$$

$\rho_c = 70\%$ – свежепобеленные стены.

$\rho_{\text{п}} = 70\%$ – свежепобеленный потолок.

По таблице определили $\eta=51\%$.

Тогда световой поток равен:

$$F = \frac{300 \times 1,5 \times 45 \times 0,9}{12 \times 51} = 2977,94 \text{ лм} \quad (14)$$

Отталкиваясь от силы светового потока, была выбрана ЛД лампа на 80Вт имеющая силу светового потока $F_{\text{п}}=3440$ лм. Для правильности выбора:

$$-10\% \leq \frac{F_{\text{п}} - F}{F_{\text{п}}} \times 100\% \leq 20\% \quad (15)$$

Подставим значения:

$$-10\% \leq 13,43\% \leq 20\% \quad (16)$$

На основании этого делаем вывод, что лампа подобрана правильно.

3.2 Инструкция по технике безопасности при выполнении работ на стенде

В едином перечне продукции, в отношении которой устанавливаются обязательные требования в рамках Таможенного союза, присутствуют пункты: «оборудование, работающее под избыточным давлением» и «сосуды, работающие под давлением». Соответствующие требования установлены ТР ТС 032/2013 «О безопасности оборудования, работающего под избыточным давлением».[12]

Сосуды под давлением являются техническими устройствами, эксплуатация которых делают производственный объект опасным. С авариями сосудов под давлением связано большое количество несчастных случаев, поэтому на их проектирование, устройство, изготовление, реконструкцию, наладку, монтаж, ремонт, техническое диагностирование и эксплуатацию в большинстве стран мира накладывается ряд ограничений.

В России обязательны Федеральные нормы и правила в области промышленной безопасности «Правила промышленной безопасности опасных производственных объектов, на которых используется оборудование, работающее под избыточным давлением», а также ряд других отраслевых документов, действие которых ограничено своей специфической областью (например «Правила устройства и безопасной эксплуатации оборудования и трубопроводов атомных энергетических установок», «Правила устройства и безопасной эксплуатации сосудов, работающих под давлением, для объектов использования атомной энергии» и др.). Эти и другие государственные документы устанавливают границы параметров содержащихся в сосуде веществ, превышение которых причисляет сосуд к опасным, в общем случае, как:

- вода с температурой выше 115°C или другие нетоксичные, не взрывопожароопасные жидкости при температуре, превышающей температуру кипения при давлении 0,07 МПа;

- пар, газ или токсичные взрывопожароопасные жидкости с давлением свыше 0,07 МПа;

- сжатых, сжиженных и растворенных газов под давлением свыше 0,07 МПа.

Требования к оснащению

Для управления работой и обеспечения безопасных условий эксплуатации сосуда в зависимости от назначения должны быть оснащены:

- запорной или запорно-регулирующей арматурой;
- приборами для измерения давления;
- приборами для измерения температуры;
- предохранительными устройствами;
- указателями уровня жидкости.

Контроль сварных соединений

Организация-изготовитель (доизготовитель), монтажная или ремонтная организация обязаны применять такие виды и объёмы контроля своей продукции, которые гарантировали бы выявление недопустимых дефектов, её высокое качество и надёжность в эксплуатации. Контроль качества сварки и сварных соединений должен включать:

- проверку аттестации персонала;
- проверку сборочно-сварочного, термического и контрольного оборудования, аппаратуры, приборов и инструментов;
- контроль качества основных материалов;
- контроль качества сварочных материалов и материалов для дефектоскопии;
- операционный контроль технологии сварки;
- неразрушающий контроль качества сварных соединений;
- разрушающий контроль качества сварных соединений;
- контроль исправления дефектов.

Государственный надзор

Сосуды, на которые распространяются российские государственные правила, до пуска их в работу должны быть зарегистрированы в органах Ростехнадзора России, кроме специально оговоренных случаев, на основании письменного заявления владельца сосуда; при перестановке сосуда на новое место или передаче сосуда другому владельцу, а также при внесении изменений в схему его включения сосуд до пуска в работу должен быть перерегистрирован в органах Ростехнадзора России.

Кроме того сосуды, на которые распространяется действие государственных правил, должны подвергаться техническому освидетельствованию после монтажа, до пуска в работу, периодически в процессе эксплуатации и в необходимых случаях – внеочередному освидетельствованию с участием специалиста организации, имеющей лицензию Ростехнадзора России (если сосуд зарегистрирован). Объем, методы и периодичность технических освидетельствований сосудов (за исключением баллонов) должны быть определены изготовителем и указаны в руководстве по эксплуатации. В случае отсутствия таких указаний техническое освидетельствование должно проводиться в соответствии с требованиями государственных правил.

3.3 Общие требования безопасности к лабораторному стенду

1. Лаборатории кафедры автоматизации производственных процессов и электротехники АмГУ служат для учебных целей, а также для проведения научно-исследовательских работ.

2. Ответственными за технику безопасности в лабораториях во время учебных занятий являются преподаватели, ведущие лабораторные работы со студентами, а при выполнении научно-исследовательских работ руководители соответствующих тем работ.

3. К работе на учебных и научно-исследовательских установках допускаются лица, изучившие настоящую Инструкцию, о чем должна быть дана расписка в журнале инструктажа.

4. Пользователи лабораторным оборудованием обязаны соблюдать правила пожарной безопасности, знать места расположения первичных средств пожаротушения. Помещение с лабораторным оборудованием должно быть оснащено двумя углекислотным огнетушителем и автоматической системой пожарной сигнализации.

5. О каждом несчастном случае с работающим на оборудовании пострадавший или очевидец несчастного случая обязан немедленно сообщить администрации АмГУ.

При выявлении неисправности оборудования, работающие на нем обязаны прекратить работу и сообщить администрации АмГУ о такой неисправности.

6. В процессе работы пользователи лабораторным оборудованием должны соблюдать правила использования средств индивидуальной и коллективной защиты, соблюдать правила личной гигиены, содержать в чистоте рабочее место.

7. В процессе работы неукоснительно выполнять все указания преподавателя, руководящего занятиями, и обслуживающего персонала лаборатории в части соблюдения правил техники безопасности и эксплуатации электроустановок.

3.3.1 Требования безопасности перед началом работы

Работающие на оборудовании обязаны:

1. Тщательно проветрить помещение с лабораторным оборудованием, убедиться что микроклимат в помещении находится в допустимых пределах: температура воздуха в холодный период года 22-24⁰С, в теплый период года 23-25⁰С, относительная влажность воздуха 40-60%.

2. Убедиться в наличии защитного заземления оборудования.

3. Вначале занятия ознакомиться с приборами и оборудованием, необходимым для выполнения лабораторной работы.

4. Получив разрешение преподавателя, собрать схему. При сборке схемы все соединительные провода необходимо надежно закреплять, чтобы

после включения в схему напряжения не произошло обрыва или отсоединения провода.

5. Пользоваться только исправными проводами, с хорошо закрепленными наконечниками.

6. Предъявить собранную схему на проверку преподавателю.

7. Подать напряжение в схему студенты имеют право только с разрешения преподавателя. За самовольное включение схемы под напряжением студент отстраняется от дальнейшего выполнения лабораторных работ и несет дисциплинарное взыскание.

3.3.2 Требования безопасности во время работы

1. Выполнять работу в строгом соответствии с указанием соответствующего методического пособия.

2. Запрещается передвигать приборы и электротехнические устройства, находящиеся под напряжением.

3. Не допускать перегрузки агрегатов приборов, строго следить за их показаниями, которых не должны отличаться от номинальных, указанных в методическом пособии.

4. В случае появления неисправности (перегрев, искрение \pm отклонение показания прибора от нормы и т.д.) немедленно отключить установку от сети и поставить в известность преподавателя.

5. Когда схема находится под напряжением, ни в коем случае нельзя касаться токоведущих частей (клемм, проводов и т.д.).

6. Если в процессе лабораторной работы требуется проводить какие-либо присоединения в схеме, то необходимо сначала обесточить схему, убедиться что все приборы находятся в отключенном состоянии, а затем приступить к работе.

7. После выполнения требуемых присоединений повторно представить схему на проверку преподавателя, который после проверки подает в схему напряжение. Студентам включать схему самостоятельно после вынужденных соединений категорически запрещается.

8. При использовании измерительным прибором, оборудованным специальными щупами (например: щуповой вольтметр на лабораторных работах по трехфазным цепям), щупы надо держать за изоляционные детали, ни в коем случае не дотрагиваться до оголенных концов, щупов, которые при присоединении их к токоведущим частям схемы, находятся под напряжением.

9. Если в процессе выполнения работы появляется необходимость сделать перерыв н работе (например, для вычисления промежуточных результатов, выявления дальнейшего хода лабораторной и кд.), необходимо сначала обесточить схему, отключив ее от сети, а затем заниматься нужными вычислениями. Перед последующей подачей в схему напряжения студент, включающему стенд в сеть необходимо обратить на это внимание всех членов бригады и затем со словами «включаю» произвести включение цепи.

10. Студентам, выполняющим лабораторные работы запрещается переходить от одного стенда к другому, заниматься посторонними разговорами и другими делами, пе касающимися выполнение лабораторной работы.

3.3.3 Требования безопасности по окончанию работы

1. После проверки преподавателем полученных результатов, разобрать схемы.

2. Установить на положенные места приборы, измерительные средства, мебель (табуреты), навести порядок и чистоту на рабочем месте.

3. Сдать лабораторную работу, сдать методические указания и монтажные провода.

3.3.4 Требования безопасности в аварийных ситуациях

1. В случаях возгорания сетевых и монтажных проводов, агрегатов и приборов немедленно:

- отключить установку от сети (выключатели, автоматы, рубильники);
- приступить к тушению пожара имеющимися средствами;
- при необходимости вызвать пожарную команду по телефону 02.

2. При поражении пользователя электрическим ТОКОМ:

Освободить пострадавшего от действия тока, отключив установку или отделив электрический контакт изолированным предметом от тела пострадавшего. Отключение произвести с помощи ближайшего выключателя, рубильника или иного отключающего аппарата. При этом необходимо следить за тем, чтобы самому не оказаться в контакте с токоведущей частью или с телом пострадавшего, а также под шаговым напряжением.

3. При термических ожогах необходимо наложить на рану стерильную повязку не вскрывать пузырей, не удалять приставшую к обожженному месту канифоль, мастику и т.д. При ожогах глаз, промыть глаза борной кислотой и срочно показаться врачу.

4. Уложить пострадавшего, освободить от стесняющей движение одежды обеспечить приток свежего воздуха, а потом положить грелку.

5. Оказать первую помощь пострадавшему, производить искусственное дыхание любым из известных методов и массаж сердца известным методом.

6 Срочно вызвать врача или скорую помощь по телефону 03. При необходимости отправить его в ближайшее лечебное учреждение.

ЗАКЛЮЧЕНИЕ

В данной работе подробно изучили имитационное моделирование систем управления движением. Была проведена работа в таких программах как SolidWorks, MatLab Simulink и CoDeSys. В качестве примера разработана рабочая модель козлового крана, которая похожа на оригинал и в точности повторяет его движения, а также создана программа управления им в ручном и автоматическом режиме. Имитационное моделирование можно использовать в разных сферах, особенно в учебном процессе, например при работе с лабораторным стендом, что позволит сначала проверить готовую программу на модели, а уже после запустить её на стенде. Таким образом, лабораторный стенд, при правильной эксплуатации, прослужит гораздо дольше.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Рыбалев, А.Н. Разработка и эмулирование АСУ ТП с использованием программ разных производителей и типов / А.Н. Рыбалев, Ф.А. Николаец // Вестник Амурского государственного университета. – 2014. – Вып. 65: Сер. Естеств. и экон. науки. – С. 73–82.
2. Современные технологии имитационного моделирования и их применение в информационных бизнес-системах [Электронный ресурс]. Режим доступа: <https://www.anylogic.ru/upload/iblock/049/0498c3885e7d7b5dc8ac3dd4f261bca0.pdf>. – 10.05.2020.
3. Каплун, С.А. SolidWorks. Оформление чертежей по ЕСКД: учебное пособие / С.А. Каплун, Т.Ф. Ходякова, И.В. Щекин - SolidWorksRussia. 2009 – 190 с.
4. Создание простейшей модели в Solidworks с помощью эскиза [Электронный ресурс]. Режим доступа: http://sapr-journal.ru/uroki-solidworks/sozдание-prostejshej-modeli-v-solidworks-s-pomoshhyu-eskiza/#disqus_thread. – 22.05.2020.
5. Создание сборки | SOLIDWORKS [Электронный ресурс]. Режим доступа: <https://autocad-lessons.ru/sozдание-sborki/>. – 22.05.2020.
6. Руководство пользователя по программированию ПЛК в CoDeSys 2.3 [Электронный ресурс], – 2006. – 158 с. Режим доступа: http://www.kipshop.ru/CoDeSys/steps/codesys_v23_ru.pdf. – 19.05.2020.
7. Использование OPC-сервера 3SSoftware для подключения контроллеров системы CoDeSys к компьютеру. Методическое пособие. Режим доступа: <https://forum-ru.codesys.com/download/file.php?id=130>. – 15.05.2020.
8. Ануфриев, И.А. MATLAB7 Наиболее полное руководство [Текст] / И.А. Ануфриев, А.Б. Смирнов, Е.Н. Смирнова // Спб.: БХВ-Петербург, 2005. – 1104 с.

9. Мусалимов, В.М Моделирование мехатронных систем в среде MATLAB (Simulink / SimMechanics): учебное пособие для высших учебных заведений / В.М Мусалимов, Г.Б. Заморуев, И.И. Калапышина, А.Д. Перечесова, К.А. Нуждин // Спб.: НИУ ИТМО, 2013. – 114 с.

10. Блинов, О.В. Исследование механических систем в среде SimMechanics (MatLab) с использованием возможностей программ трехмерного моделирования / О.В. Блинов, В.Б. Кузнецов / Методические указания для лабораторного практикума по дисциплинам «Теория механических цепей», «Технология системного моделирования», «Моделирование систем», «Динамические модели сложных систем». Иваново, 2012. – 19 с.

11. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. Режим доступа: <http://docs.cntd.ru/document/901865498>. – 19.06.2020.

12. Рыбалев, А.Н. Инструкция по охране труда при работе па лабораторном оборудовании кафедры автоматизации производственных процессов и электротехники: инструкция/ А.Н. Рыбалев Ин от 10-2013. – 7 с.

ПРИЛОЖЕНИЕ А

Техническое задание на разработку

Техническое задание разрабатывается согласно требованиям ГОСТ 34.602-89 «Информационная технология».

Техническое задание содержит следующие разделы:

- 1) введение;
- 2) назначение и цели создания (развития) системы;
- 3) характеристика объектов автоматизации;
- 4) требования к системе;
- 5) состав и содержание работ по созданию системы;
- 6) порядок контроля и приемки системы;
- 7) требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие;
- 8) требования к документированию;
- 9) источники разработки.

1) Введение

Данный курсовой проект будет посвящен имитационному моделированию управления движения систем.

Объектом данной разработки является козловой кран.

Модель – это объект, который был создан искусственно с целью упрощенного представления о реальном объекте, процессе или явлении и отражает существенные стороны объекта, который изучается, с точки зрения цели моделирования.

Моделирование – исследование объектов познания на их моделях; построение и изучение моделей реально существующих объектов, процессов или явлений с целью получения объяснений этих явлений, а также для предсказания явлений, интересующих исследователя.

Объект, для которого создают его модель, называют оригиналом или прототипом. Модель не является абсолютной копией своего прототипа, а

Продолжение приложения А

лишь отражает основные его качества и свойства, которые являются наиболее существенными для выбранной цели исследования. При создании модели всегда имеют место определенные допущения и гипотезы. С помощью системного подхода можно создавать полноценные модели. Исполнители: Алеко М.А, Дорофеева Т.А.

2) Назначение и цели создания (развития) системы

Имитационное моделирование управления систем движения нужно для того что бы изучить движение объекта, как он работает, из чего состоит. Разработка прототипа, на основе которого будут создаваться реальные модели объектов, которые будут заменять реальные объекты.

3) Характеристика объектов автоматизации

Козловой кран (КК) – многофункциональное оборудование для различных площадок и объектов, где грузоподъемные работы ведутся постоянно. Козловой кран – это агрегат мостового типа. Горизонтальная балка, по которой перемещается груз – конструктивно относится к мостовой группе подъемных механизмов. Основным отличием является то, что КК может сам передвигаться по рельсам. Усредненный срок эксплуатации козловых кранов – около 20 лет, наработка на отказ $\approx 3\ 000$ циклов.

4) Требование к системе

Нужно что бы моделируемая система была проста и понятна, максимально соответствовала движениям и вела себя, так как объект, который мы моделируем. Система должна реализовать ручное и автоматическое управление.

5) Состав и содержание работ по созданию системы

Разработка трехмерной модели в Solidworks, перенос этой модели и разработка управляющей программы в Simulink, разработка программы визуализации, апробации, настройка.

6) Порядок контроля и приемки системы

Модель козлового крана, которая должна двигаться по трех осям, поднимать предметы. Модель должна двигаться при помощи команд, которые будут задаваться. Руководитель проекта раз в неделю проводит контроль системы, а приемка системы будет непосредственно на защите.

7) Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

Установка программного обеспечения (MatLab (с пакетом Simscape Multibody Link), Codesys), настройка связи, к составу модели есть Solidworks.

8) Требования к документированию

Описание компонентов систем управления, инструкция к запуску модели, общее описание, описание режимов работ (ручного, автоматического).

9) Источники разработки

1. Рыбалев, А.Н. Разработка и эмулирование АСУ ТП с использованием программ разных производителей и типов / А.Н. Рыбалев, Ф.А. Николаец // Вестник Амурского государственного университета. – 2014. – Вып. 65: Сер. Естеств. и экон. науки. – С. 73–82.

2. Ануфриев, И.А. MATLAB7 Наиболее полное руководство [Текст] / И.А. Ануфриев, А.Б. Смирнов, Е.Н. Смирнова // Спб.: БХВ-Петербург, 2005. – 1104 с.

3. Руководство пользователя по программированию ПЛК в CoDeSys 2.3 [Электронный ресурс], –2006. – 158 с. Режим доступа: http://www.kipshop.ru/CoDeSys/steps/codesys_v23_ru.pdf, свободный. – Загл. с экрана.

ПРИЛОЖЕНИЕ Б

Kont_ri

```
IF BTN_gruz = TRUE THEN
    kon_1 := TRUE;
END_IF

IF BTN_gruz = FALSE AND kon_1 = TRUE THEN
    kon_2 := TRUE;
END_IF

IF kon_1 = TRUE AND kon_2 = TRUE THEN
    kon := kon+1;
    kon_1 := FALSE;
    kon_2 := FALSE;
END_IF

IF kon = 7 THEN
    kon := 0;
END_IF

IF BTN_clear THEN
    gruz := FALSE;
    gruz1 := FALSE;
    gruz2 := FALSE;
    gruz3 := FALSE;
    gruz4 := FALSE;
    gruz5 := FALSE;
    p := FALSE;
    p1 := FALSE;
    p2 := FALSE;
    p3 := FALSE;
    p4 := FALSE;
    p5 := FALSE;
    kon := 0;
END_IF

IF kon = 0 THEN
    state1 := 0;
END_IF
IF kon = 1 THEN
    state1 := 1;
END_IF
IF kon = 2 THEN
    state1 := 2;
END_IF
IF kon = 3 THEN
    state1 := 3;
END_IF
IF kon = 4 THEN
```

Продолжение приложения Б

```
state1 := 4;
END_IF
IF kon = 5 THEN
state1 := 5;
END_IF
IF kon = 6 THEN
state1 := 6;
END_IF

CASE state1 OF

1:(*-----1-----*)
IF kon = 1 THEN
gruz := TRUE;
END_IF

IF (kon = 1 AND v = TRUE AND BTN_vziat) OR (upr = TRUE AND vziat = TRUE) THEN
vziat := TRUE;
p := TRUE;
ELSE
vziat := FALSE;
END_IF

IF X = 15 AND Y = 14.5 AND Z = 0 AND kon = 1 THEN
v := TRUE;
ELSIF vziat = FALSE AND kon = 1 AND NOT (X = 15 AND Y = 14.5 AND Z = 0) THEN
v := FALSE;
END_IF

2:(*-----2-----*)
IF kon = 2 THEN
gruz1 := TRUE;
END_IF

IF (kon = 2 AND v1 = TRUE AND BTN_vziat) OR (upr = TRUE AND vziat1 = TRUE) THEN
vziat1 := TRUE;
p1 := TRUE;
ELSE
vziat1 := FALSE;
END_IF

IF X = 15 AND Y = 14.5 AND Z = 0 AND kon = 2 THEN
v1 := TRUE;
ELSIF vziat1 = FALSE AND kon = 2 AND NOT (X = 15 AND Y = 14.5 AND Z = 0) THEN
v1 := FALSE;
END_IF

3:(*-----3-----*)
```

Продолжение приложения Б

```
IF kon = 3 THEN
    gruz2 := TRUE;
END_IF

IF (kon = 3 AND v2 = TRUE AND BTN_vziat) OR (upr = TRUE AND vziat2 = TRUE) THEN
    vziat2 := TRUE;
    p2 := TRUE;
ELSE
    vziat2 := FALSE;
END_IF

IF X = 15 AND Y = 14.5 AND Z = 0 AND kon = 3 THEN
    v2 := TRUE;
ELSIF vziat2 = FALSE AND kon = 3 AND NOT (X = 15 AND Y = 14.5 AND Z = 0) THEN
    v2 := FALSE;
END_IF

4:(*-----4-----*)
IF kon = 4 THEN
    gruz3 := TRUE;
END_IF

IF (kon = 4 AND v3 = TRUE AND BTN_vziat) OR (upr = TRUE AND vziat3 = TRUE) THEN
    vziat3 := TRUE;
    p3 := TRUE;
ELSE
    vziat3 := FALSE;
END_IF

IF X = 15 AND Y = 14.5 AND Z = 0 AND kon = 4 THEN
    v3 := TRUE;
ELSIF vziat3 = FALSE AND kon = 4 AND NOT (X = 15 AND Y = 14.5 AND Z = 0) THEN
    v3 := FALSE;
END_IF

5:(*-----5-----*)
IF kon = 5 THEN
    gruz4 := TRUE;
END_IF

IF (kon = 5 AND v4 = TRUE AND BTN_vziat) OR (upr = TRUE AND vziat4 = TRUE) THEN
    vziat4 := TRUE;
    p4 := TRUE;
ELSE
    vziat4 := FALSE;
END_IF

IF X = 15 AND Y = 14.5 AND Z = 0 AND kon = 5 THEN
    v4 := TRUE;
ELSIF vziat4 = FALSE AND kon = 5 AND NOT (X = 15 AND Y = 14.5 AND Z = 0) THEN
```


Продолжение приложения Б

```
        v4 := FALSE;
    END_IF

    6:(*-----6-----*)
    IF kon = 6 THEN
        gruz5 := TRUE;
    END_IF

    IF (kon = 6 AND v5 = TRUE AND BTN_vziat) OR (upr = TRUE AND vziat5 = TRUE) THEN
        vziat5 := TRUE;
        p5 := TRUE;
    ELSE
        vziat5 := FALSE;
    END_IF

    IF X = 15 AND Y = 14.5 AND Z = 0 AND kon = 6 THEN
        v5 := TRUE;
    ELSIF vziat5 = FALSE AND kon = 6 AND NOT (X = 15 AND Y = 14.5 AND Z = 0) THEN
        v5 := FALSE;
    END_IF

    END_CASE
```

ПРИЛОЖЕНИЕ В

r_upr

```
IF upr = FALSE THEN

np:=TRUE;
GO := FALSE;

IF BTN_kran_levo = TRUE THEN
    kran_levo:=TRUE;
ELSE
    kran_levo:=FALSE;
END_IF

IF BTN_kran_pravo = TRUE THEN
    kran_pravo:=TRUE;
ELSE
    kran_pravo:=FALSE;
END_IF

IF BTN_osnkab_levo = TRUE THEN
    osnkab_levo:=TRUE;
ELSE
    osnkab_levo:=FALSE;
END_IF

IF BTN_osnkab_pravo = TRUE THEN
    osnkab_pravo:=TRUE;
ELSE
    osnkab_pravo:=FALSE;
END_IF

IF BTN_kruk_up = TRUE THEN
    kruk_up:=TRUE;
ELSE
    kruk_up:=FALSE;
END_IF

IF BTN_kruk_down = TRUE THEN
    kruk_down:=TRUE;
ELSE
    kruk_down:=FALSE;
END_IF

IF X=0 THEN
    BTN_osnkab_levo:=FALSE;
ELSIF X=30 THEN
    BTN_osnkab_pravo:=FALSE;
END_IF
```

Продолжение приложения В

```

IF Y=0 THEN
    BTN_kruk_up:=FALSE;
ELSIF Y=14.5 THEN
    BTN_kruk_down:=FALSE;
END_IF

IF Z=0 THEN
    BTN_kran_levo:=FALSE;
ELSIF Z=-60 THEN
    BTN_kran_pravo:=FALSE;
END_IF

(*////////////////////////////////////*)

IF BTN_npol = TRUE THEN
    kruk_up:=TRUE;
    kruk_down:=FALSE;
    IF BTN_npol = TRUE AND Y=0 THEN
        kruk_up:=FALSE;
        kruk_down:=FALSE;
        kran_levo:=TRUE;
        kran_pravo:=FALSE;
        osnkab_levo:=TRUE;
        osnkab_pravo:=FALSE;
    END_IF
END_IF

IF X=0 AND Y=0 AND Z=0 AND BTN_npol = TRUE THEN
    kruk_up:=FALSE;
    kruk_down:=FALSE;
    kran_levo:=FALSE;
    kran_pravo:=FALSE;
    osnkab_levo:=TRUE;
    osnkab_pravo:=FALSE;
    BTN_npol := FALSE;
END_IF

(*////////////////////////////////////*)

IF BTN_pol = TRUE THEN
    poz_x := 15;
    poz_z := 0;
    poz_y :=14.5;
    kruk_up := TRUE;
    kruk_down:=FALSE;
    IF Y=0 AND BTN_pol =TRUE THEN
        kruk_up := FALSE;
        kruk_down:=FALSE;
        IF BTN_pol =TRUE AND poz_x - X > 0 THEN
            osnkab_pravo := TRUE;
            osnkab_levo := FALSE;
        END_IF
    END_IF
END_IF

```

Продолжение приложения В

```
    sign_x_R := TRUE;
    sign_x_L := FALSE;
  ELSIF BTN_pol = TRUE AND poz_x - X < 0 THEN
    osnkab_pravo := FALSE;
    osnkab_levo := TRUE;
    sign_x_L := TRUE;
    sign_x_R := FALSE;
  ELSIF BTN_pol = TRUE AND poz_x - X = 0 THEN
    osnkab_pravo := FALSE;
    osnkab_levo := FALSE;
    sign_x_L := FALSE;
    sign_x_R := FALSE;
  END_IF

  IF BTN_pol = TRUE AND Z <> 0 THEN
    kran_levo := TRUE;
    kran_pravo := FALSE;
    sign_z_R := TRUE;
    sign_z_L := FALSE;
  ELSIF BTN_pol = TRUE AND Z = 0 THEN
    kran_levo := FALSE;
    kran_pravo := FALSE;
    sign_z_R := FALSE;
    sign_z_L := FALSE;
  END_IF
END_IF

END_IF

IF BTN_pol = TRUE AND X = 15 AND Y = 0 AND Z = 0 THEN
  kran_levo := FALSE;
  kran_pravo := FALSE;
  sign_z_R := FALSE;
  sign_z_L := FALSE;
  osnkab_pravo := FALSE;
  osnkab_levo := FALSE;
  sign_x_L := FALSE;
  sign_x_R := FALSE;
  kruk_up := FALSE;
  kruk_down := FALSE;
  sign_y_UP := FALSE;
  sign_y_DOWN := FALSE;
  BTN_pol := FALSE;
END_IF

END_IF
```

ПРИЛОЖЕНИЕ Г

a_upr

IF upr = TRUE THEN

IF np = TRUE THEN

kruk_up:=TRUE;

kruk_down:=FALSE;

IF Y=0 THEN

kruk_up:=FALSE;

kruk_down:=FALSE;

kran_levo:=TRUE;

kran_pravo:=FALSE;

osnkab_levo:=TRUE;

osnkab_pravo:=FALSE;

END_IF

END_IF

IF X=0 AND Y=0 AND Z=0 AND np = TRUE THEN

kruk_up:=FALSE;

kruk_down:=FALSE;

kran_levo:=FALSE;

kran_pravo:=FALSE;

osnkab_levo:=FALSE;

osnkab_pravo:=FALSE;

np := FALSE;

END_IF

END_IF

IF kon = 1 AND GO = TRUE AND np = FALSE THEN

state2 := 10;

END_IF

IF kon = 2 AND GO = TRUE AND np = FALSE THEN

state2 := 20;

END_IF

IF kon = 3 AND GO = TRUE AND np = FALSE THEN

state2 := 30;

END_IF

IF kon = 4 AND GO = TRUE AND np = FALSE THEN

state2 := 40;

END_IF

IF kon = 5 AND GO = TRUE AND np = FALSE THEN

state2 := 50;

END_IF

IF kon = 6 AND GO = TRUE AND np = FALSE THEN

Продолжение приложения Г

```
    state2 := 60;
END_IF

CASE state2 OF

10:
    poz_x := 15;
    state2 := 11;

11:
    IF poz_x - X > 0 THEN
        osnkab_pravo := TRUE;
        osnkab_levo := FALSE;
        sign_x_R := TRUE;
        sign_x_L := FALSE;
    ELSIF poz_x - X = 0 THEN
        osnkab_pravo := FALSE;
        osnkab_levo := FALSE;
        sign_x_L := FALSE;
        sign_x_R := FALSE;
    END_IF

    IF X = 15 AND Y=0 AND Z=0 THEN
        state2 := 12;
    END_IF

12:
    IF Y <> 14.5 THEN
        kruk_down := TRUE;
        kruk_up := FALSE;
    ELSE
        kruk_down := FALSE;
        kruk_up := FALSE;
        state2 := 13;
    END_IF

13:
    vziat := TRUE;
    kruk_down := FALSE;
    kruk_up := TRUE;
    IF Y = 0 THEN
        kruk_down := FALSE;
        kruk_up := FALSE;
        state2 := 14;
    END_IF

14:
    poz_x := 5;

    IF poz_x - X < 0 THEN
```

Продолжение приложения Г

```
    osnkab_pravo := FALSE;
    osnkab_levo := TRUE;
    sign_x_R := FALSE;
    sign_x_L := TRUE;
ELSIF poz_x - X = 0 THEN
    osnkab_pravo := FALSE;
    osnkab_levo := FALSE;
    sign_x_L := FALSE;
    sign_x_R := FALSE;
END_IF

kran_levo := FALSE;
kran_pravo := TRUE;

IF Z = - 60 THEN
    kran_levo := FALSE;
    kran_pravo := FALSE;
END_IF

IF X = 5 AND Y=0 AND Z=-60 THEN
    kruk_down := TRUE;
    kruk_up := FALSE;
END_IF

IF Y = 14.5 THEN
    kruk_down := FALSE;
    kruk_up := TRUE;
    vziat := FALSE;
    state2 := 15;
END_IF
```

15:

```
IF Y=0 THEN
    kruk_up:=FALSE;
    kruk_down:=FALSE;
    kran_levo:=TRUE;
    kran_pravo:=FALSE;

    poz_x := 15;

    IF poz_x - X > 0 THEN
        osnkab_pravo := TRUE;
        osnkab_levo := FALSE;
        sign_x_R := TRUE;
        sign_x_L := FALSE;
    ELSIF poz_x - X = 0 THEN
        osnkab_pravo := FALSE;
        osnkab_levo := FALSE;
        sign_x_L := FALSE;
```

Продолжение приложения Г

```
        sign_x_R := FALSE;
    END_IF

    IF X = 15 AND Y=0 AND Z=0 THEN
        state2 := 0;
    END_IF
END_IF

20:
    poz_x := 15;
    state2 :=21;

21:
    IF poz_x - X > 0 THEN
        osnkab_pravo := TRUE;
        osnkab_levo := FALSE;
        sign_x_R := TRUE;
        sign_x_L := FALSE;
    ELSIF poz_x - X = 0 THEN
        osnkab_pravo := FALSE;
        osnkab_levo := FALSE;
        sign_x_L := FALSE;
        sign_x_R := FALSE;
    END_IF

    IF X = 15 AND Y=0 AND Z=0 THEN
        state2 := 22;
    END_IF

22:
    IF Y <> 14.5 THEN
        kruk_down := TRUE;
        kruk_up := FALSE;
    ELSE
        kruk_down := FALSE;
        kruk_up := FALSE;
        state2 := 23;
    END_IF

23:
    vziat1 := TRUE;
    kruk_down := FALSE;
    kruk_up := TRUE;
    IF Y = 0 THEN
        kruk_down := FALSE;
        kruk_up := FALSE;
        state2 := 24;
    END_IF

24:
```


Продолжение приложения Г

```
kran_levo := FALSE;
kran_pravo := TRUE;

IF Z = - 60 THEN
    kran_levo := FALSE;
    kran_pravo := FALSE;
END_IF

IF X = 15 AND Y=0 AND Z=-60 THEN
    kruk_down := TRUE;
    kruk_up := FALSE;
END_IF

IF Y = 14.5 THEN
    kruk_down := FALSE;
    kruk_up := TRUE;
    vziat1 := FALSE;
    state2 := 25;
END_IF
```

25:

```
IF Y=0 THEN
    kruk_up:=FALSE;
    kruk_down:=FALSE;
    kran_levo:=TRUE;
    kran_pravo:=FALSE;

    poz_x := 15;

    IF poz_x - X > 0 THEN
        osnkab_pravo := TRUE;
        osnkab_levo := FALSE;
        sign_x_R := TRUE;
        sign_x_L := FALSE;
    ELSIF poz_x - X = 0 THEN
        osnkab_pravo := FALSE;
        osnkab_levo := FALSE;
        sign_x_L := FALSE;
        sign_x_R := FALSE;
    END_IF

    IF X = 15 AND Y=0 AND Z=0 THEN
        state2 := 0;
    END_IF
END_IF
```

30:

```
poz_x := 15;
state2 :=31;
```

Продолжение приложения Г

31:

```
IF poz_x - X > 0 THEN
    osnkab_pravo := TRUE;
    osnkab_levo := FALSE;
    sign_x_R := TRUE;
    sign_x_L := FALSE;
ELSIF poz_x - X = 0 THEN
    osnkab_pravo := FALSE;
    osnkab_levo := FALSE;
    sign_x_L := FALSE;
    sign_x_R := FALSE;
END_IF

IF X = 15 AND Y=0 AND Z=0 THEN
    state2 := 32;
END_IF
```

32:

```
IF Y <> 14.5 THEN
    kruk_down := TRUE;
    kruk_up := FALSE;
ELSE
    kruk_down := FALSE;
    kruk_up := FALSE;
    state2 := 33;
END_IF
```

33:

```
vziat2 := TRUE;
kruk_down := FALSE;
kruk_up := TRUE;
IF Y = 0 THEN
    kruk_down := FALSE;
    kruk_up := FALSE;
    state2 := 34;
END_IF
```

34:

```
poz_x := 25;

IF poz_x - X > 0 THEN
    osnkab_pravo := TRUE;
    osnkab_levo := FALSE;
    sign_x_R := TRUE;
    sign_x_L := FALSE;
ELSIF poz_x - X = 0 THEN
    osnkab_pravo := FALSE;
    osnkab_levo := FALSE;
    sign_x_L := FALSE;
    sign_x_R := FALSE;
```

Продолжение приложения Г

END_IF

```
kran_levo := FALSE;  
kran_pravo := TRUE;
```

```
IF Z = - 60 THEN  
    kran_levo := FALSE;  
    kran_pravo := FALSE;  
END_IF
```

```
IF X = 25 AND Y=0 AND Z=-60 THEN  
    kruk_down := TRUE;  
    kruk_up := FALSE;  
END_IF
```

```
IF Y = 14.5 THEN  
    kruk_down := FALSE;  
    kruk_up := TRUE;  
    vziat2 := FALSE;  
    state2 := 35;  
END_IF
```

35:

```
IF Y=0 THEN  
    kruk_up:=FALSE;  
    kruk_down:=FALSE;  
    kran_levo:=TRUE;  
    kran_pravo:=FALSE;  
  
    poz_x := 15;  
  
    IF poz_x - X < 0 THEN  
        osnkab_pravo := FALSE;  
        osnkab_levo := TRUE;  
        sign_x_R := FALSE;  
        sign_x_L := TRUE;  
    ELSIF poz_x - X = 0 THEN  
        osnkab_pravo := FALSE;  
        osnkab_levo := FALSE;  
        sign_x_L := FALSE;  
        sign_x_R := FALSE;  
    END_IF  
  
    IF X = 15 AND Y=0 AND Z=0 THEN  
        state2 := 0;  
    END_IF  
END_IF
```

40:

Продолжение приложения Г

```
poz_x := 15;  
state2 := 41;
```

41:

```
IF poz_x - X > 0 THEN  
    osnkab_pravo := TRUE;  
    osnkab_levo := FALSE;  
    sign_x_R := TRUE;  
    sign_x_L := FALSE;  
ELSIF poz_x - X = 0 THEN  
    osnkab_pravo := FALSE;  
    osnkab_levo := FALSE;  
    sign_x_L := FALSE;  
    sign_x_R := FALSE;  
END_IF  
  
IF X = 15 AND Y=0 AND Z=0 THEN  
    state2 := 42;  
END_IF
```

42:

```
IF Y <> 14.5 THEN  
    kruk_down := TRUE;  
    kruk_up := FALSE;  
ELSE  
    kruk_down := FALSE;  
    kruk_up := FALSE;  
    state2 := 43;  
END_IF
```

43:

```
vziat3 := TRUE;  
kruk_down := FALSE;  
kruk_up := TRUE;  
IF Y = 0 THEN  
    kruk_down := FALSE;  
    kruk_up := FALSE;  
    state2 := 44;  
END_IF
```

44:

```
poz_x := 5;  
poz_z := 40;  
  
IF poz_x - X < 0 THEN  
    osnkab_pravo := FALSE;  
    osnkab_levo := TRUE;  
    sign_x_R := FALSE;  
    sign_x_L := TRUE;  
ELSIF poz_x - X = 0 THEN
```

Продолжение приложения Г

```
osnkab_pravo := FALSE;
osnkab_levo := FALSE;
sign_x_L := FALSE;
sign_x_R := FALSE;
END_IF

IF poz_z + Z > 0 THEN
    kran_levo := FALSE;
    kran_pravo := TRUE;
    sign_z_R := TRUE;
    sign_z_L := FALSE;
ELSIF poz_z + Z = 0 THEN
    kran_levo := FALSE;
    kran_pravo := FALSE;
    sign_z_R := FALSE;
    sign_z_L := FALSE;
END_IF

IF X = 5 AND Y=0 AND Z=-40 THEN
    kruk_down := TRUE;
    kruk_up := FALSE;
END_IF

IF Y = 14.5 THEN
    kruk_down := FALSE;
    kruk_up := TRUE;
    vziat3 := FALSE;
    state2 := 45;
END_IF
```

45:

```
IF Y=0 THEN
    kruk_up:=FALSE;
    kruk_down:=FALSE;
    kran_levo:=TRUE;
    kran_pravo:=FALSE;

    poz_x := 15;

    IF poz_x - X > 0 THEN
        osnkab_pravo := TRUE;
        osnkab_levo := FALSE;
        sign_x_R := TRUE;
        sign_x_L := FALSE;
    ELSIF poz_x - X = 0 THEN
        osnkab_pravo := FALSE;
        osnkab_levo := FALSE;
        sign_x_L := FALSE;
        sign_x_R := FALSE;
```

Продолжение приложения Г

```
END_IF

IF X = 15 AND Y=0 AND Z=0 THEN
    state2 := 0;
END_IF
END_IF

50:
    poz_x := 15;
    state2 :=51;

51:
    IF poz_x - X > 0 THEN
        osnkab_pravo := TRUE;
        osnkab_levo := FALSE;
        sign_x_R := TRUE;
        sign_x_L := FALSE;
    ELSIF poz_x - X = 0 THEN
        osnkab_pravo := FALSE;
        osnkab_levo := FALSE;
        sign_x_L := FALSE;
        sign_x_R := FALSE;
    END_IF

    IF X = 15 AND Y=0 AND Z=0 THEN
        state2 := 52;
    END_IF

52:
    IF Y <> 14.5 THEN
        kruk_down := TRUE;
        kruk_up := FALSE;
    ELSE
        kruk_down := FALSE;
        kruk_up := FALSE;
        state2 := 53;
    END_IF

53:
    vziat4 := TRUE;
    kruk_down := FALSE;
    kruk_up := TRUE;
    IF Y = 0 THEN
        kruk_down := FALSE;
        kruk_up := FALSE;
        state2 := 54;
    END_IF

54:
    poz_z := 40;
```

Продолжение приложения Г

```
IF poz_z + Z > 0 THEN
    kran_levo := FALSE;
    kran_pravo := TRUE;
    sign_z_R := TRUE;
    sign_z_L := FALSE;
ELSIF poz_z + Z = 0 THEN
    kran_levo := FALSE;
    kran_pravo := FALSE;
    sign_z_R := FALSE;
    sign_z_L := FALSE;
END_IF
```

```
IF X = 15 AND Y=0 AND Z=-40 THEN
    kruk_down := TRUE;
    kruk_up := FALSE;
END_IF
```

```
IF Y = 14.5 THEN
    kruk_down := FALSE;
    kruk_up := TRUE;
    vziat4 := FALSE;
    state2 := 55;
END_IF
```

55:

```
IF Y=0 THEN
    kruk_up:=FALSE;
    kruk_down:=FALSE;
    kran_levo:=TRUE;
    kran_pravo:=FALSE;

    poz_x := 15;

    IF poz_x - X > 0 THEN
        osnkab_pravo := TRUE;
        osnkab_levo := FALSE;
        sign_x_R := TRUE;
        sign_x_L := FALSE;
    ELSIF poz_x - X = 0 THEN
        osnkab_pravo := FALSE;
        osnkab_levo := FALSE;
        sign_x_L := FALSE;
        sign_x_R := FALSE;
    END_IF

    IF X = 15 AND Y=0 AND Z=0 THEN
        state2 := 0;
    END_IF
```

Продолжение приложения Г

END_IF

60:

```
poz_x := 15;  
state2 := 61;
```

61:

```
IF poz_x - X > 0 THEN  
    osnkab_pravo := TRUE;  
    osnkab_levo := FALSE;  
    sign_x_R := TRUE;  
    sign_x_L := FALSE;  
ELSIF poz_x - X = 0 THEN  
    osnkab_pravo := FALSE;  
    osnkab_levo := FALSE;  
    sign_x_L := FALSE;  
    sign_x_R := FALSE;  
END_IF
```

```
IF X = 15 AND Y=0 AND Z=0 THEN  
    state2 := 62;  
END_IF
```

62:

```
IF Y <> 14.5 THEN  
    kruk_down := TRUE;  
    kruk_up := FALSE;  
ELSE  
    kruk_down := FALSE;  
    kruk_up := FALSE;  
    state2 := 63;  
END_IF
```

63:

```
vziat5 := TRUE;  
kruk_down := FALSE;  
kruk_up := TRUE;  
IF Y = 0 THEN  
    kruk_down := FALSE;  
    kruk_up := FALSE;  
    state2 := 64;  
END_IF
```

64:

```
poz_x := 25;  
poz_z := 40;  
  
IF poz_x - X > 0 THEN  
    osnkab_pravo := TRUE;  
    osnkab_levo := FALSE;
```


Продолжение приложения Г

```
    sign_x_R := TRUE;
    sign_x_L := FALSE;
ELSIF poz_x - X = 0 THEN
    osnkab_pravo := FALSE;
    osnkab_leva := FALSE;
    sign_x_L := FALSE;
    sign_x_R := FALSE;
END_IF

IF poz_z + Z > 0 THEN
    kran_leva := FALSE;
    kran_pravo := TRUE;
    sign_z_R := TRUE;
    sign_z_L := FALSE;
ELSIF poz_z + Z = 0 THEN
    kran_leva := FALSE;
    kran_pravo := FALSE;
    sign_z_R := FALSE;
    sign_z_L := FALSE;
END_IF

IF X = 25 AND Y=0 AND Z=-40 THEN
    kruk_down := TRUE;
    kruk_up := FALSE;
END_IF

IF Y = 14.5 THEN
    kruk_down := FALSE;
    kruk_up := TRUE;
    vziat5 := FALSE;
    state2 := 65;
END_IF
```

65:

```
IF Y=0 THEN
    kruk_up:=FALSE;
    kruk_down:=FALSE;
    kran_leva:=TRUE;
    kran_pravo:=FALSE;

    poz_x := 15;

    IF poz_x - X < 0 THEN
        osnkab_pravo := FALSE;
        osnkab_leva := TRUE;
        sign_x_R := FALSE;
        sign_x_L := TRUE;
    ELSIF poz_x - X = 0 THEN
        osnkab_pravo := FALSE;
```

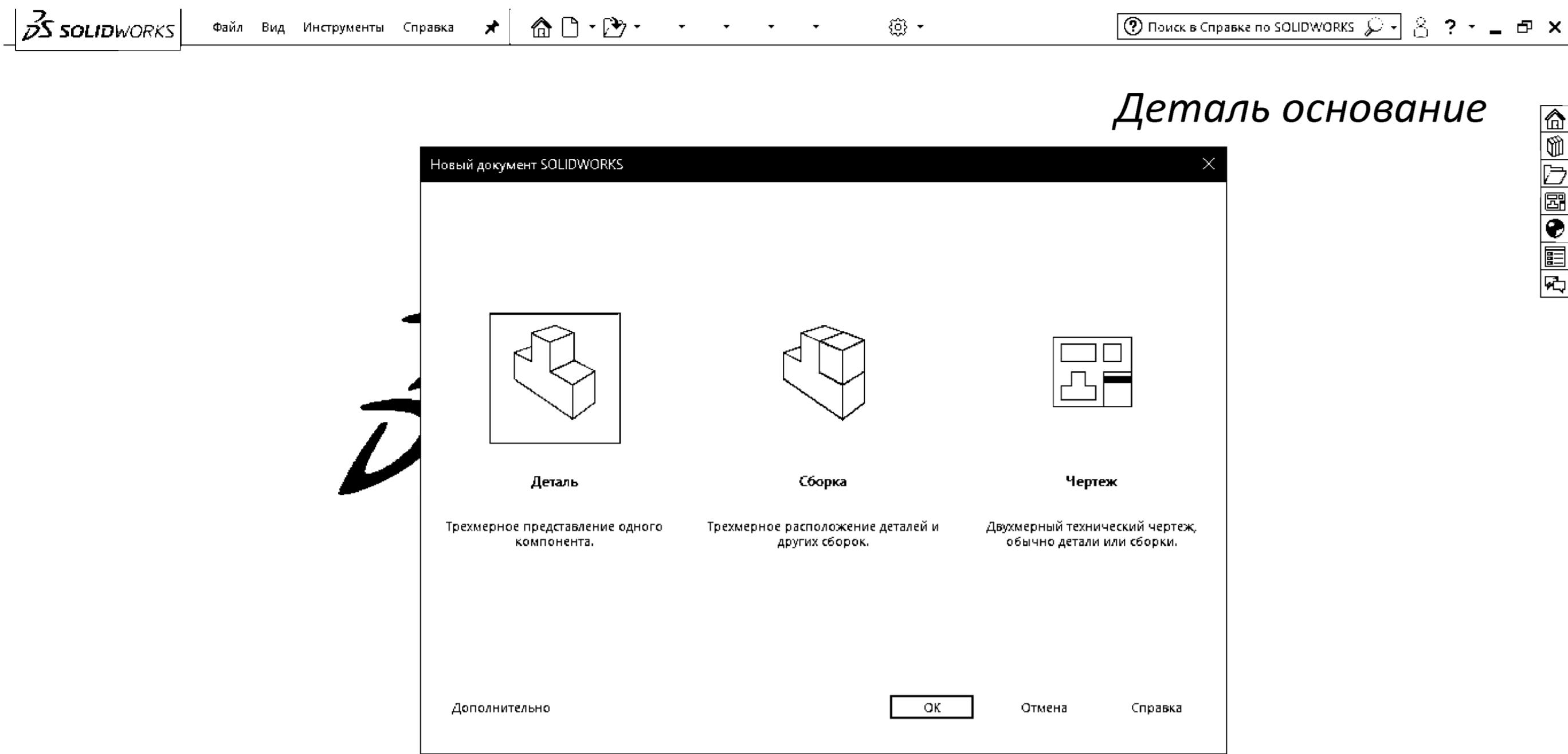
Продолжение приложения Г

```
osnkab_levo := FALSE;
sign_x_L := FALSE;
sign_x_R := FALSE;
END_IF

IF X = 15 AND Y=0 AND Z=0 THEN
    state2 := 0;
END_IF
END_IF

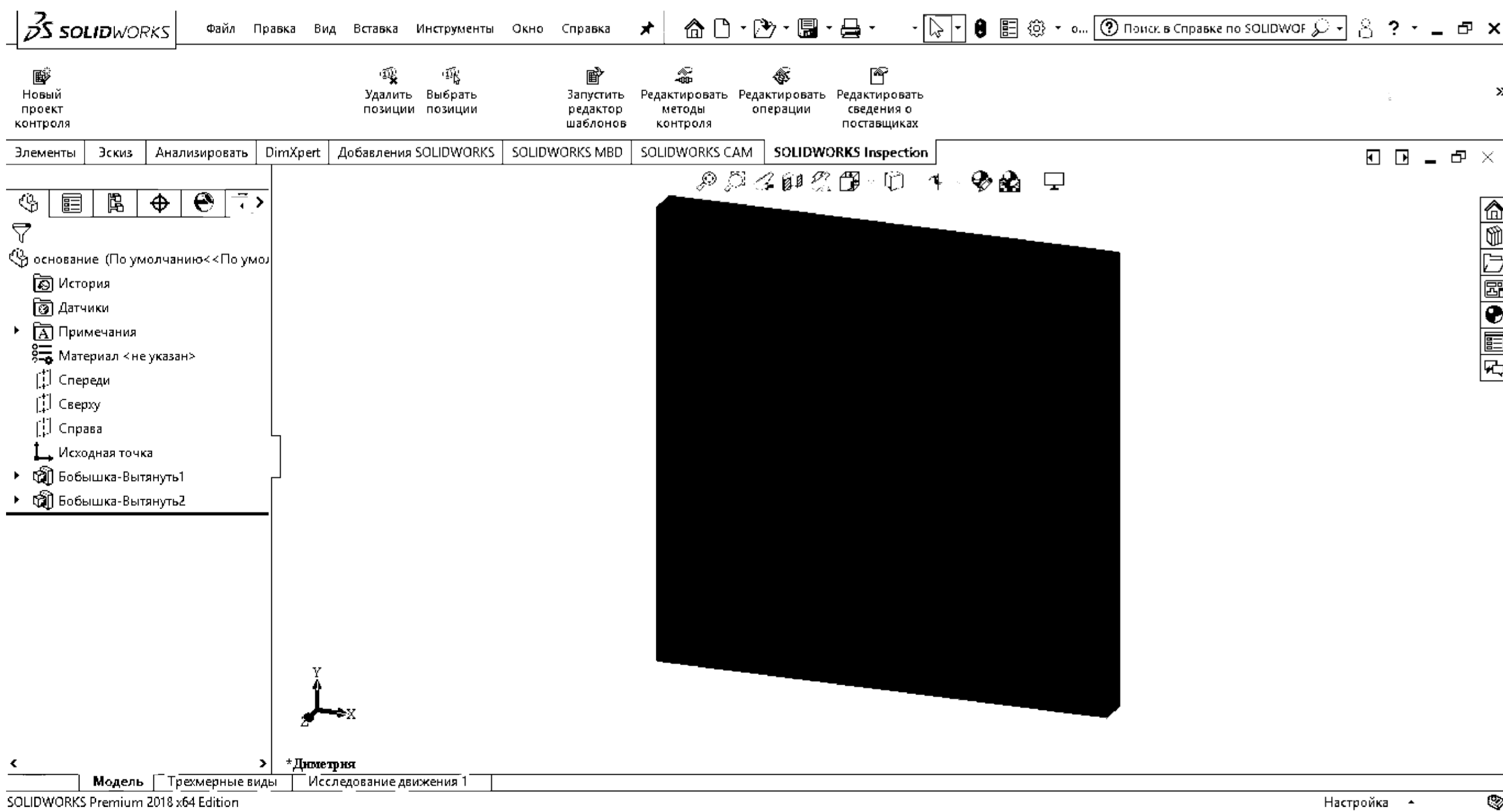
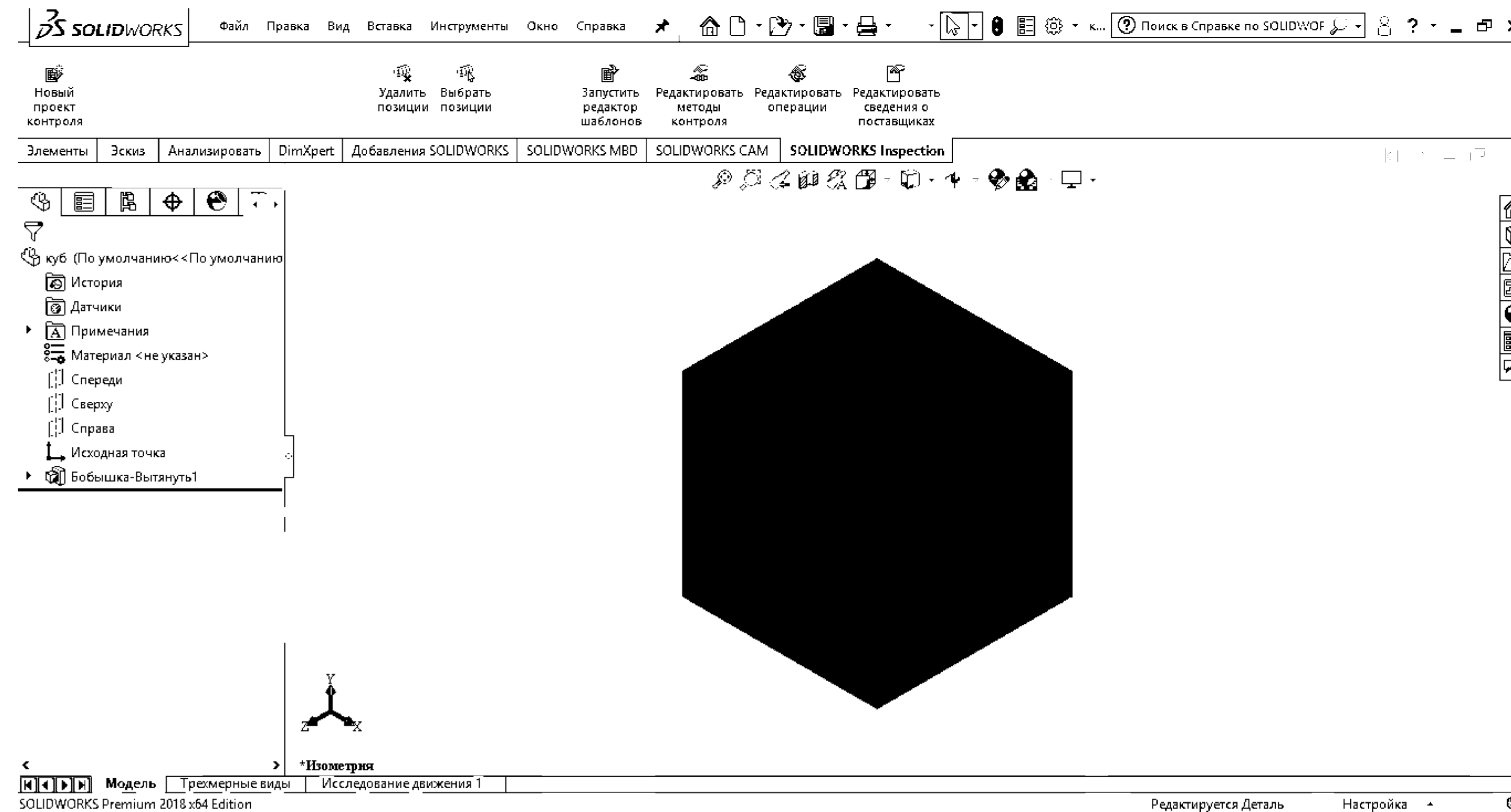
END_CASE
```

Создание нового документа

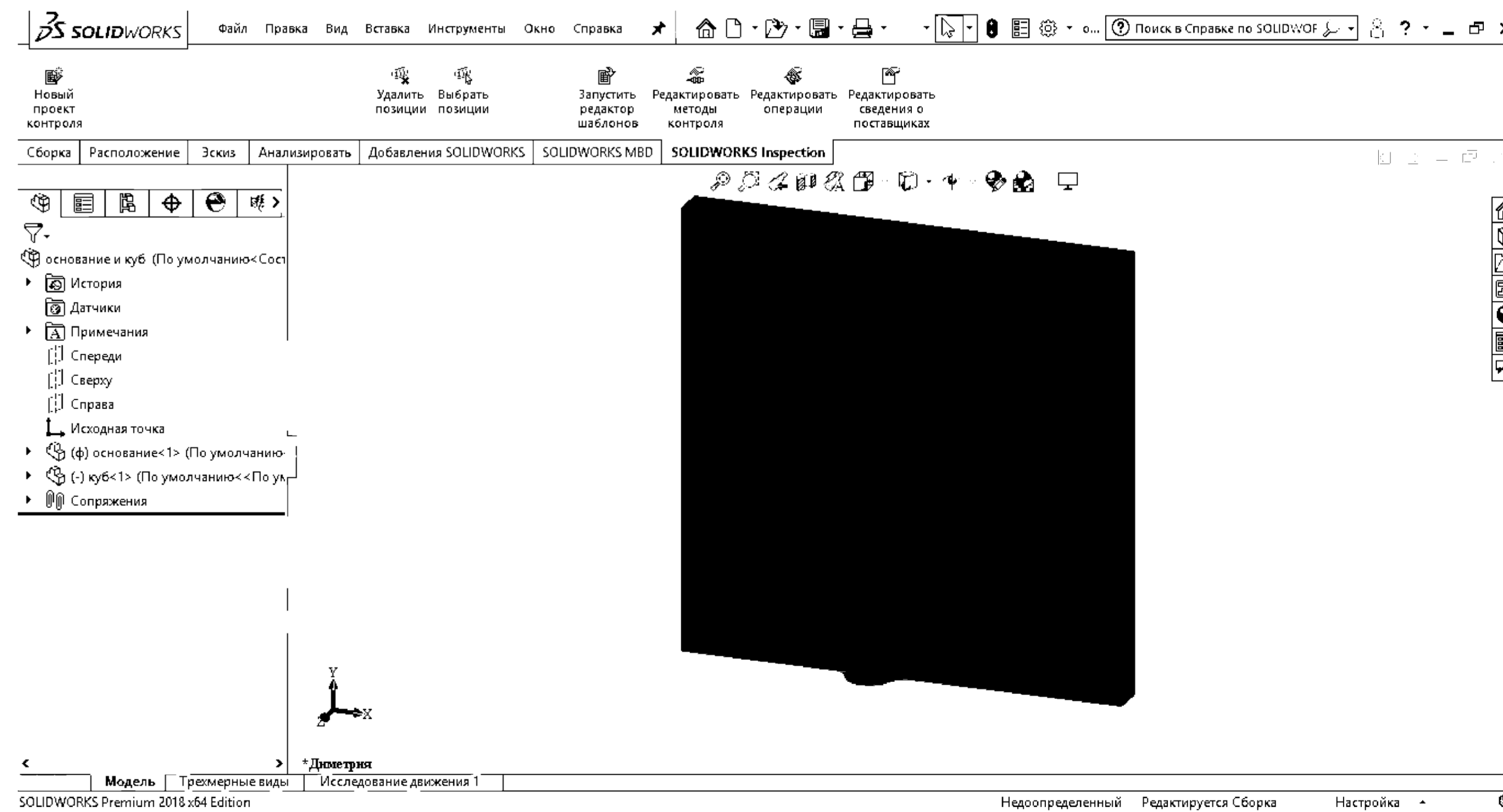


Выберите параметр типа документа и учебного пособия, если в настоящее время используется учебное пособие.

Деталь куб



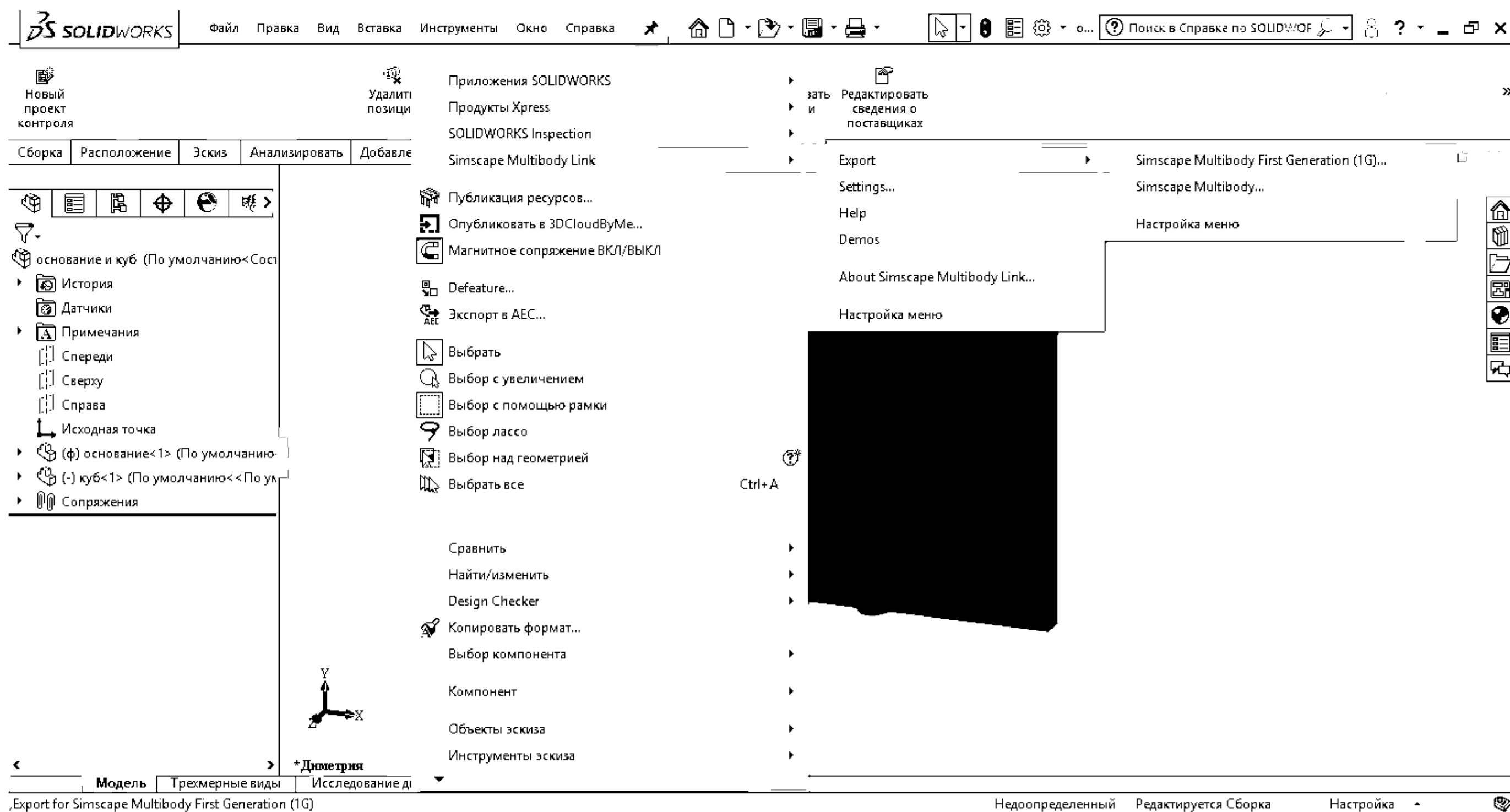
Деталь основание



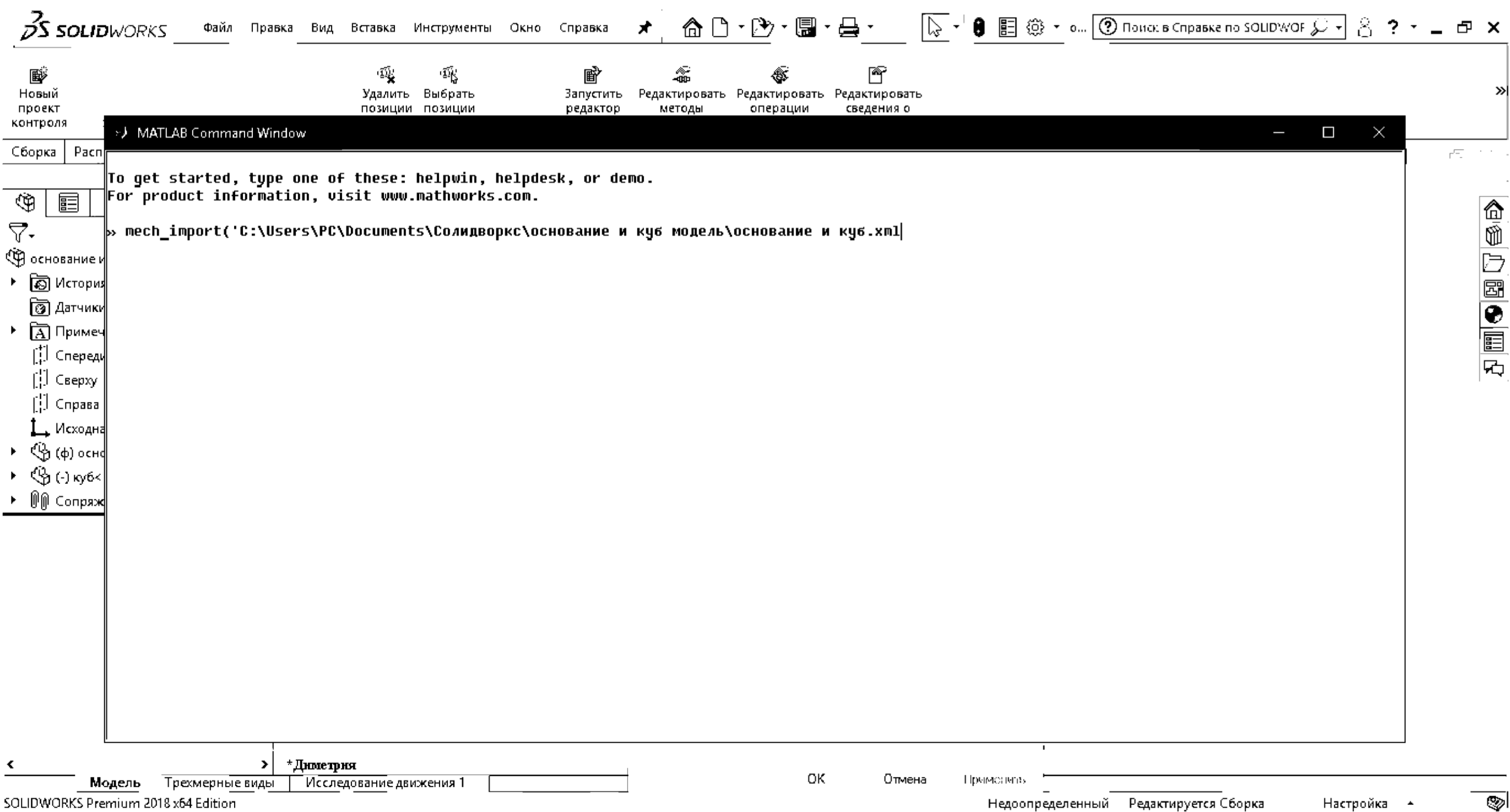
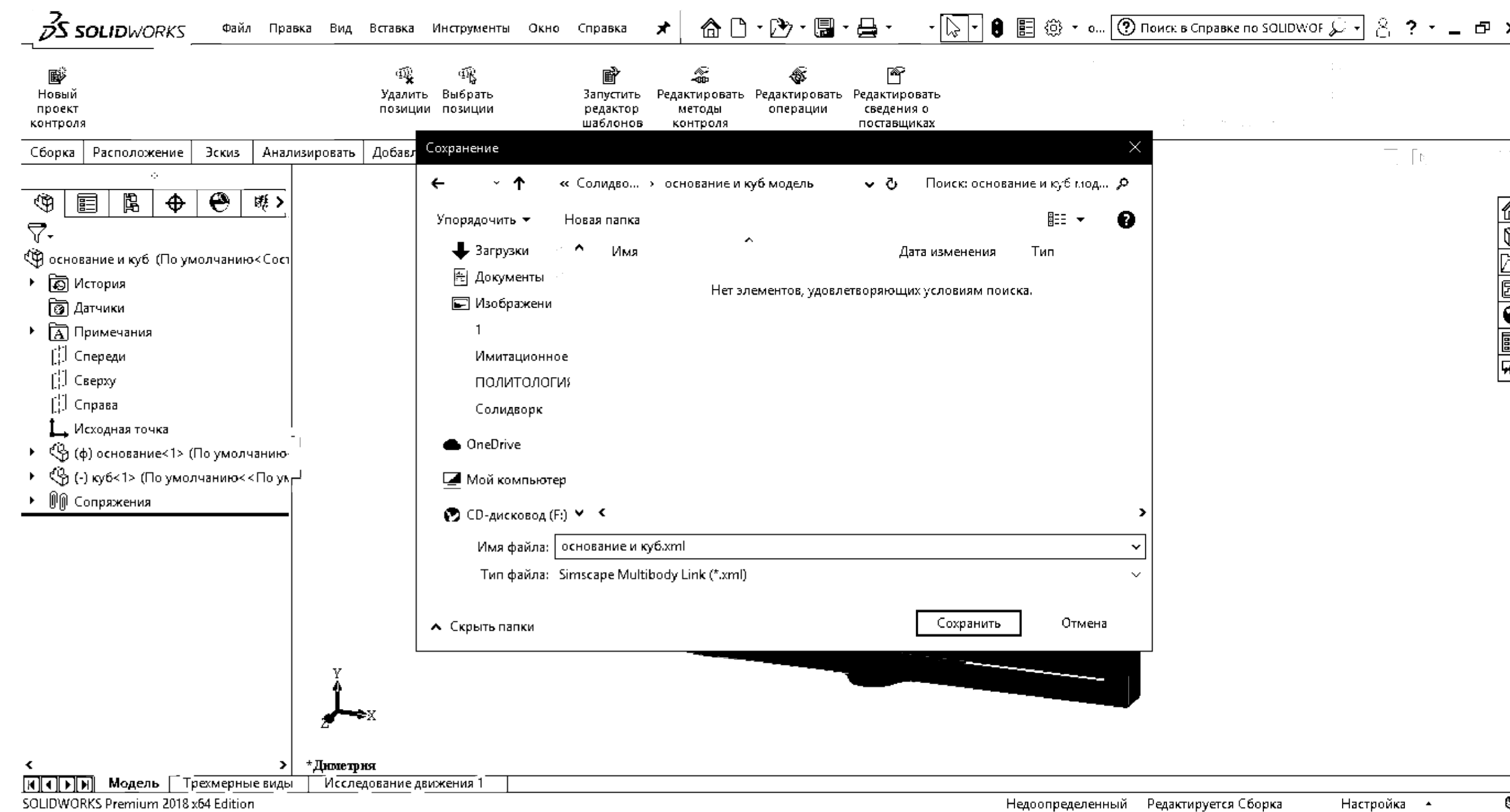
Сборка основание и куб

				ВКР.164023.15.03.04.CX						
Изм	Лист	№ докум.	Подп.	Дата	Разработка 3D-модели в SolidWorks	Литера	Масса	Масштаб		
Разраб.		Алеко М.А.				у		1:1		
Провер.		Рыбалева А.Н.				Лист 1	Листов 6			
Т.Контр.		Рыбалева А.Н.				Разработка имитационных моделей систем управления движением (комплексная выпускная квалификационная работа)				
Рецензент		Скрипко О.В.			АМГУ					
Утвержд.		Скрипко О.В.			Кафедра АППиЭ					

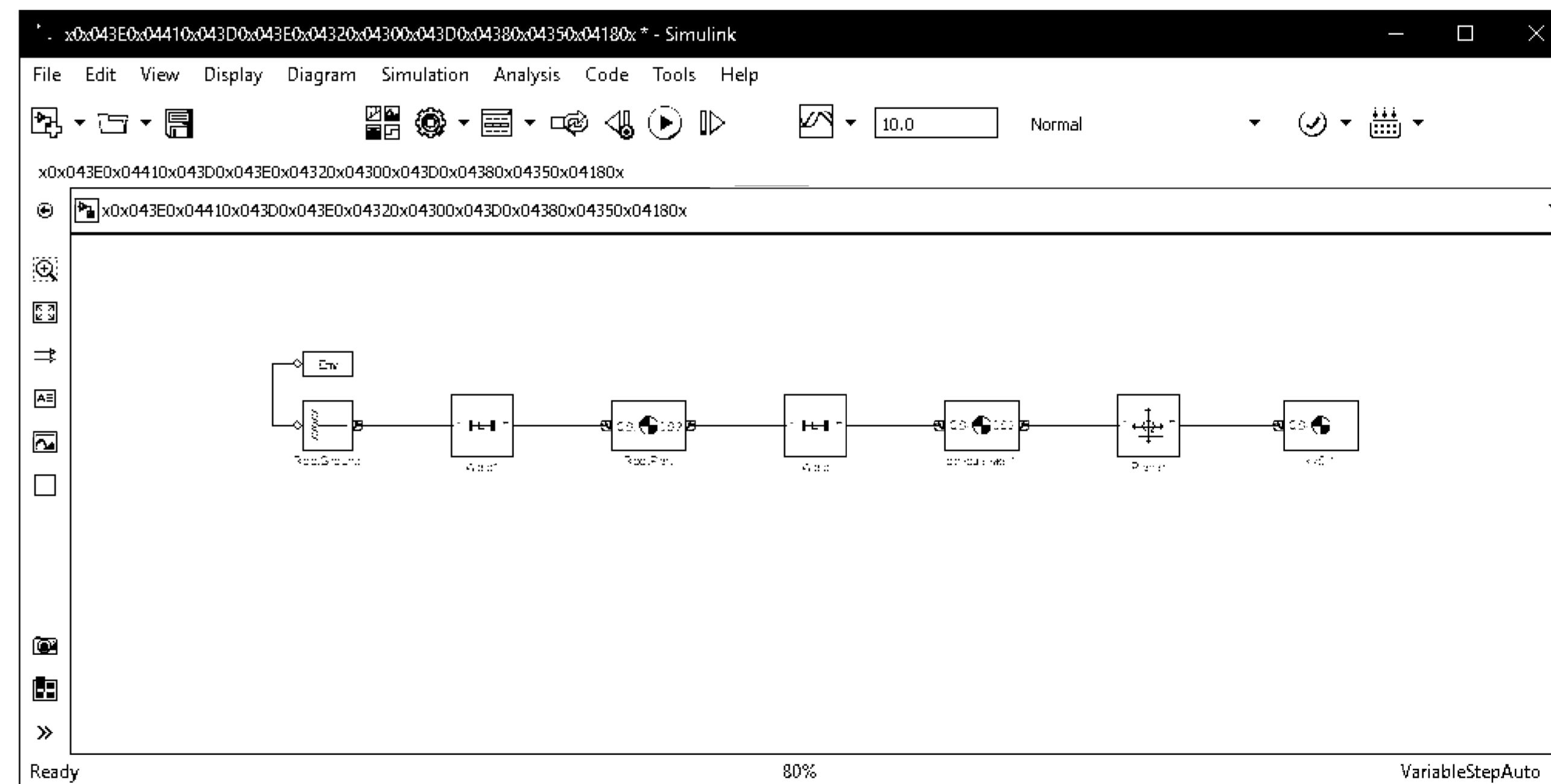
Экспорт в Simscape Multibody Link



Сохранение в формате XML

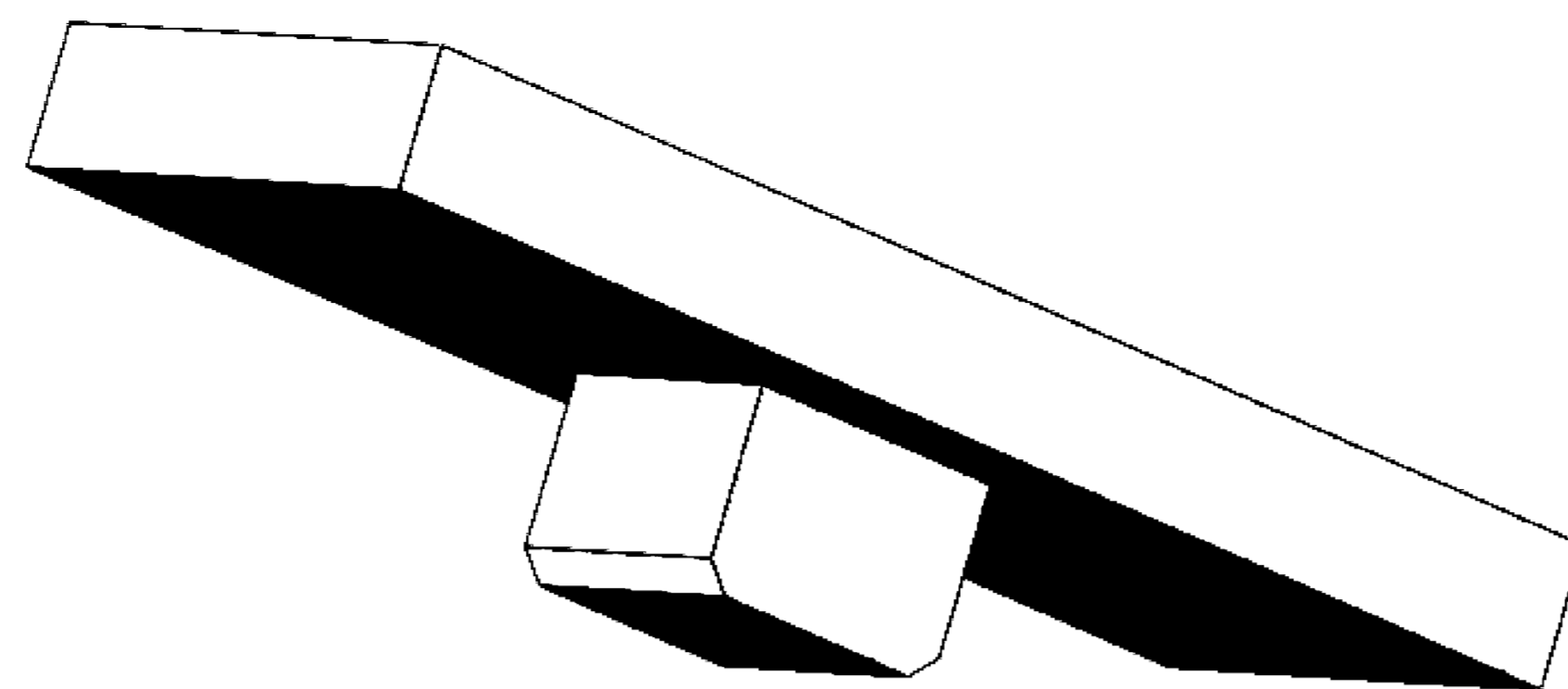


Командное окно MATLAB

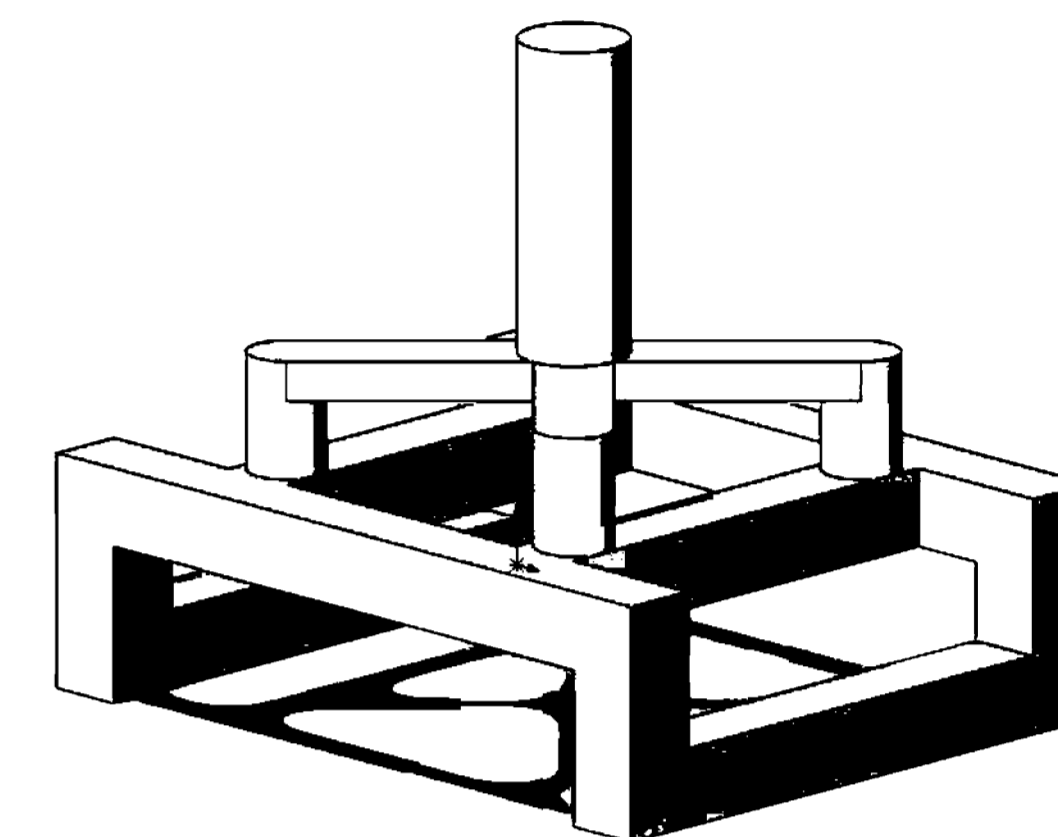


Simulink модель

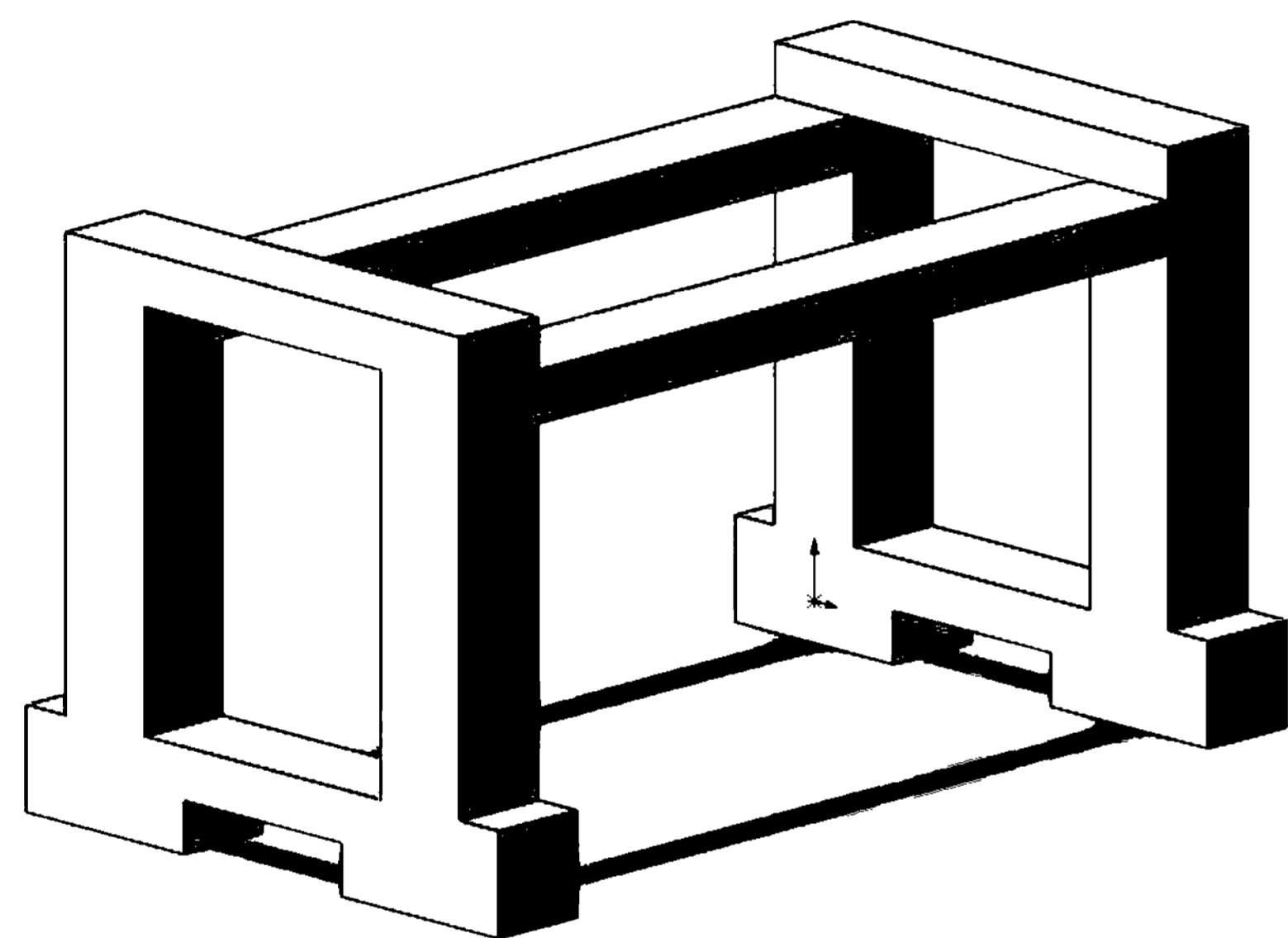
				BKP.164023.15.03.04.CX		
Изм.	Лист	№ докум.	Подп.	Дата	Литера	Масштаб
Разраб.	Алеко М.А.				у	1:1
Провер.	Рыбалева А.Н.					
Т.Контр.	Рыбалева А.Н.				Лист 2	Листов 6
Рецензент	Н.Контр.	Скрипко О.В.			АМГУ Кафедра АППиЭ	
Утвержд.	Скрипко О.В.					
					Разработка имитационных моделей систем управления движением (комплексная выпускная квалификационная работа)	



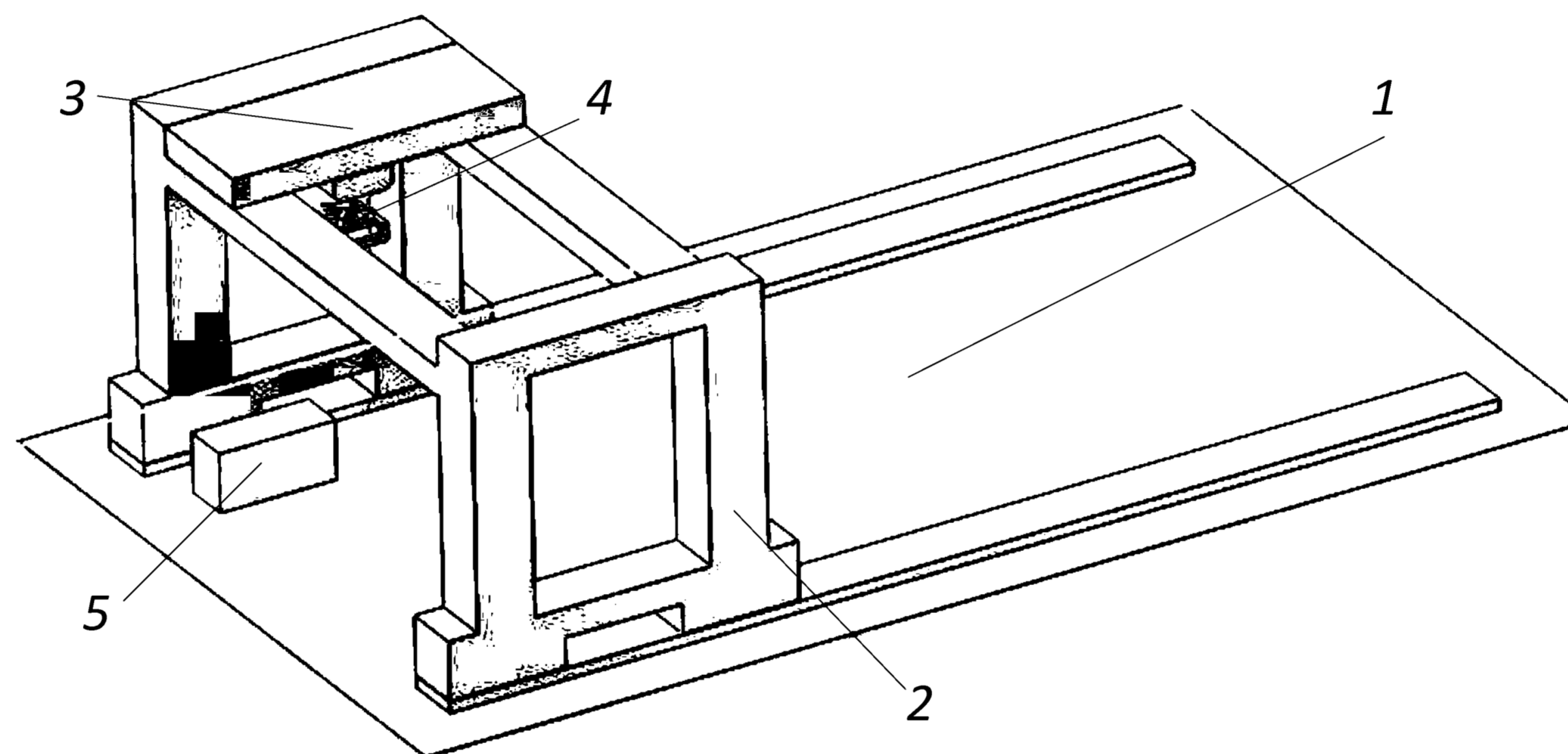
3. Деталь кабина



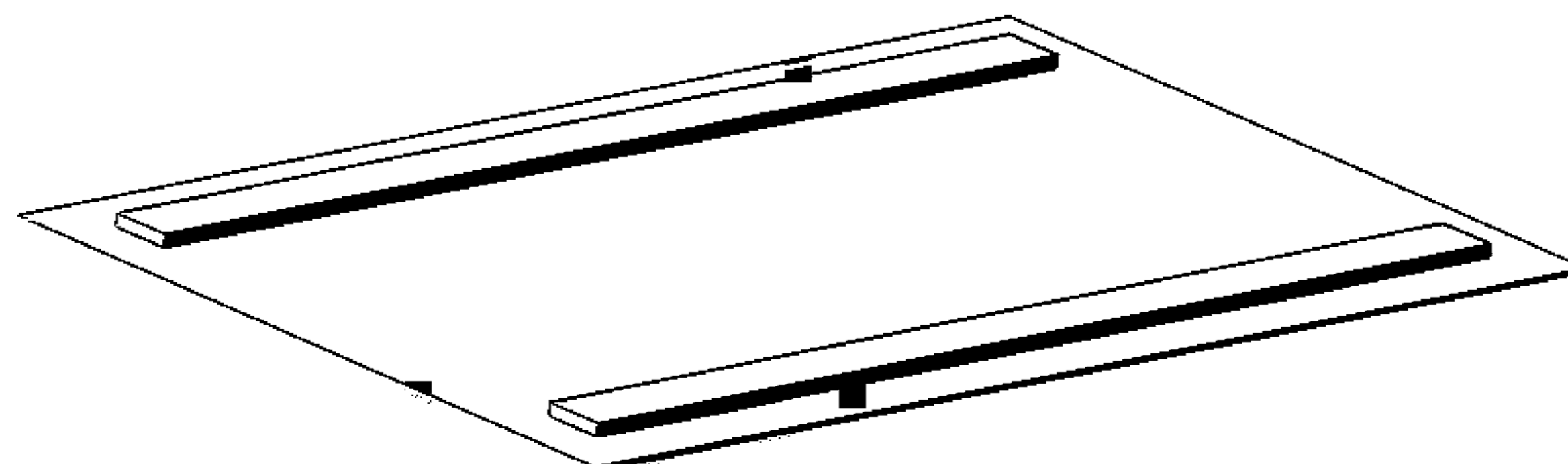
4. Деталь крюк



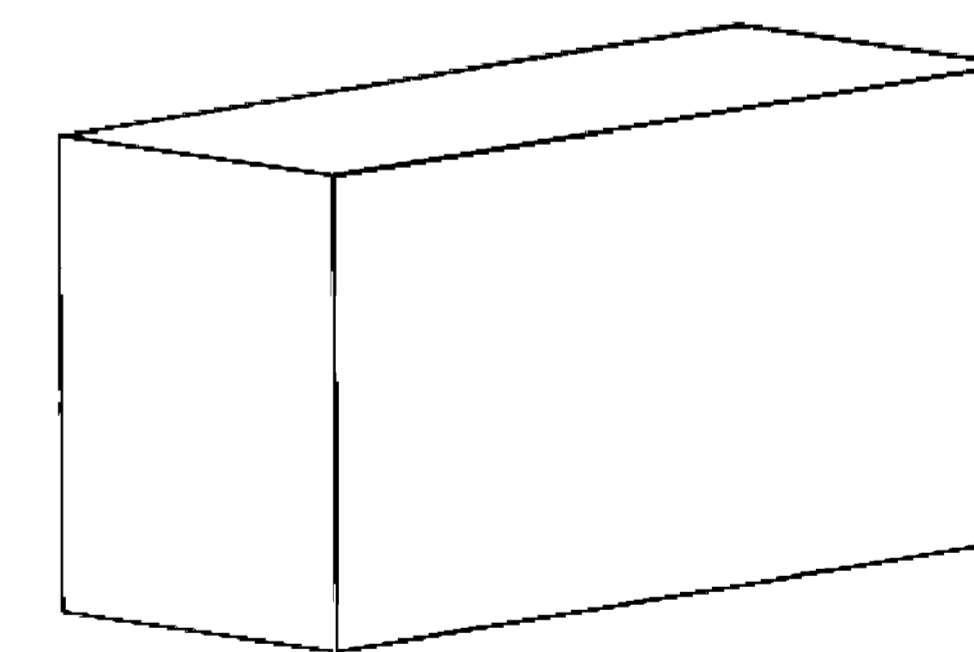
2. Деталь ноги



Сборка (3D Модель крана)



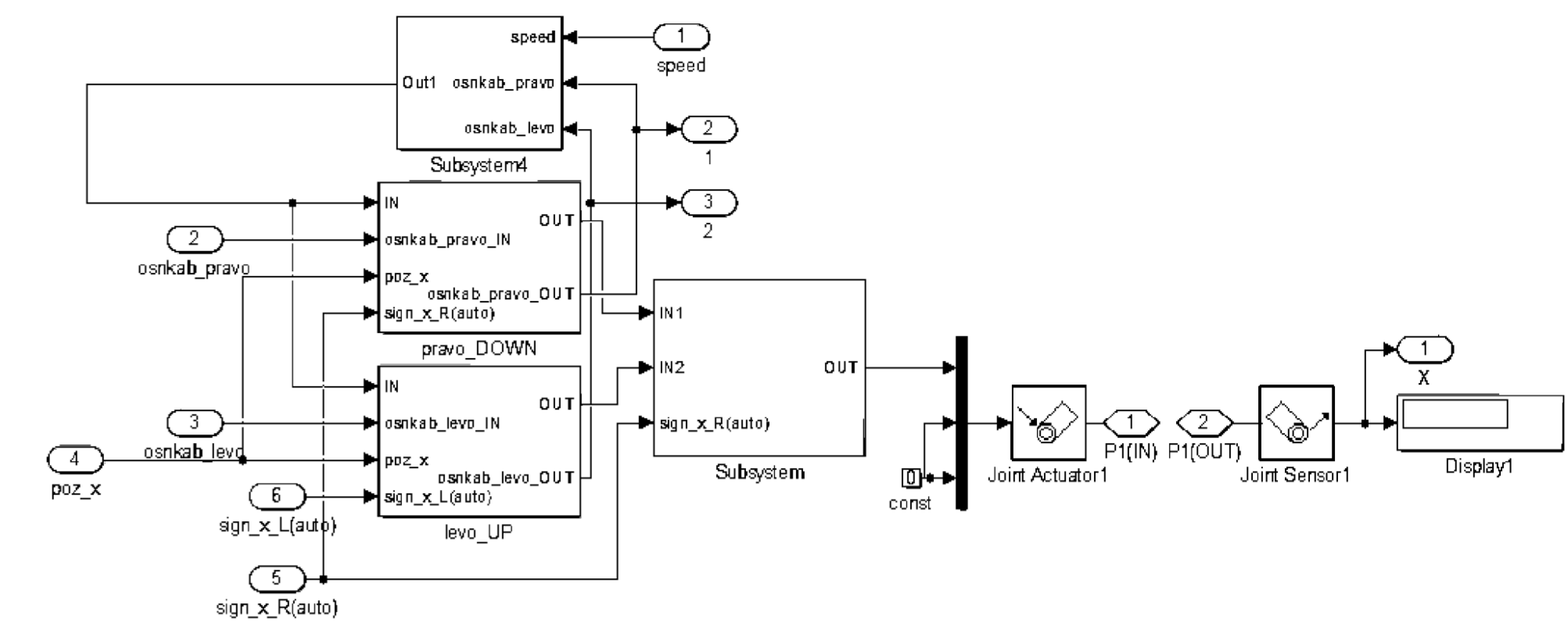
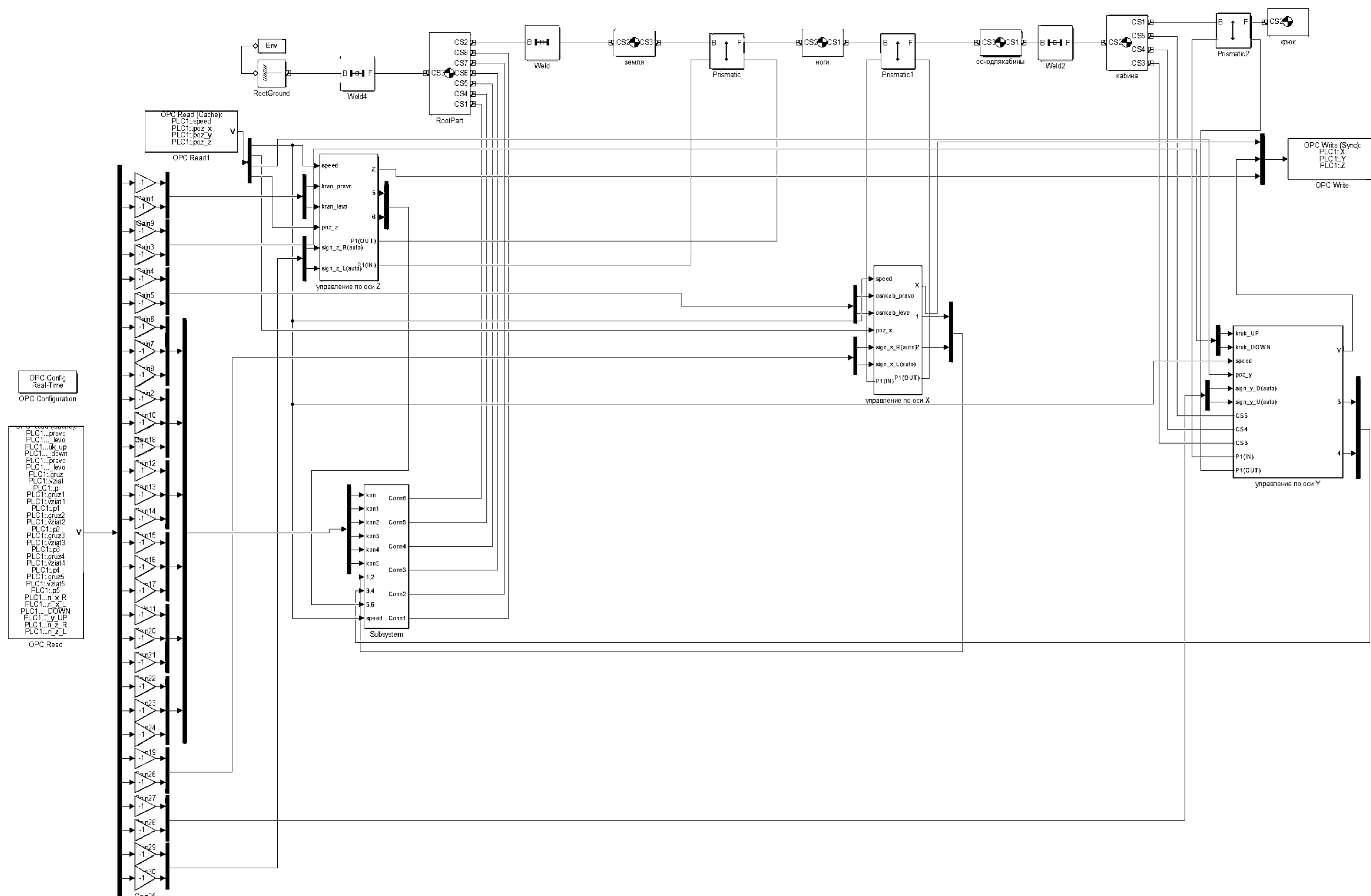
1. Деталь основание



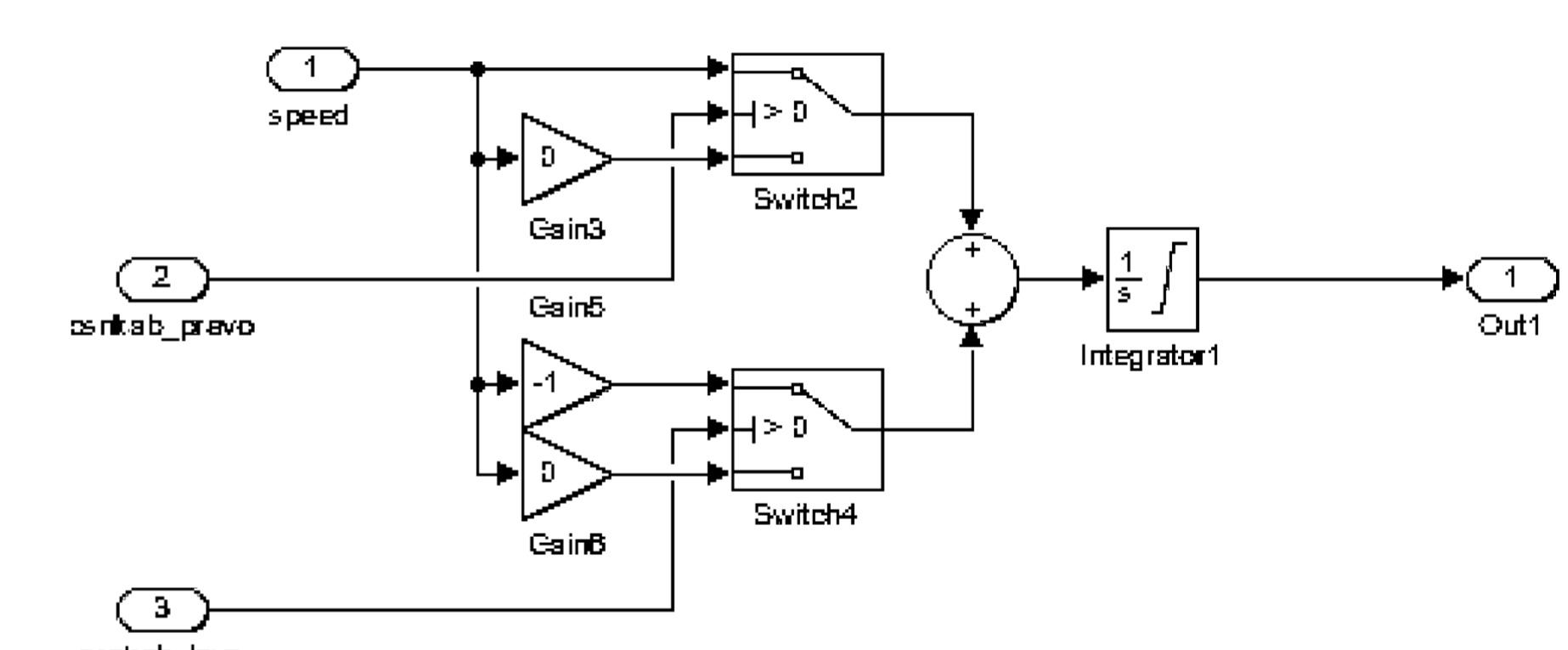
5. Деталь контейнер (груз)

				ВКР.164023.15.03.04.СХ			
Изм	Лист	№ докум.	Подп.	Дата	Литера	Масса	Масштаб
		Алеко М.А.			у		1:1
Разраб.		Рыбалева А.Н.					
Провер.		Рыбалева А.Н.					
Т.Контр.		Рыбалева А.Н.			Лист 3	Листов 6	
Рецензент		Скрипко О.В.			Разработка имитационных моделей систем управления движением (комплексная выпускная квалификационная работа)		АМГУ Кафедра АППиЭ
Н.Контр.		Скрипко О.В.					
Утвержд.		Скрипко О.В.					

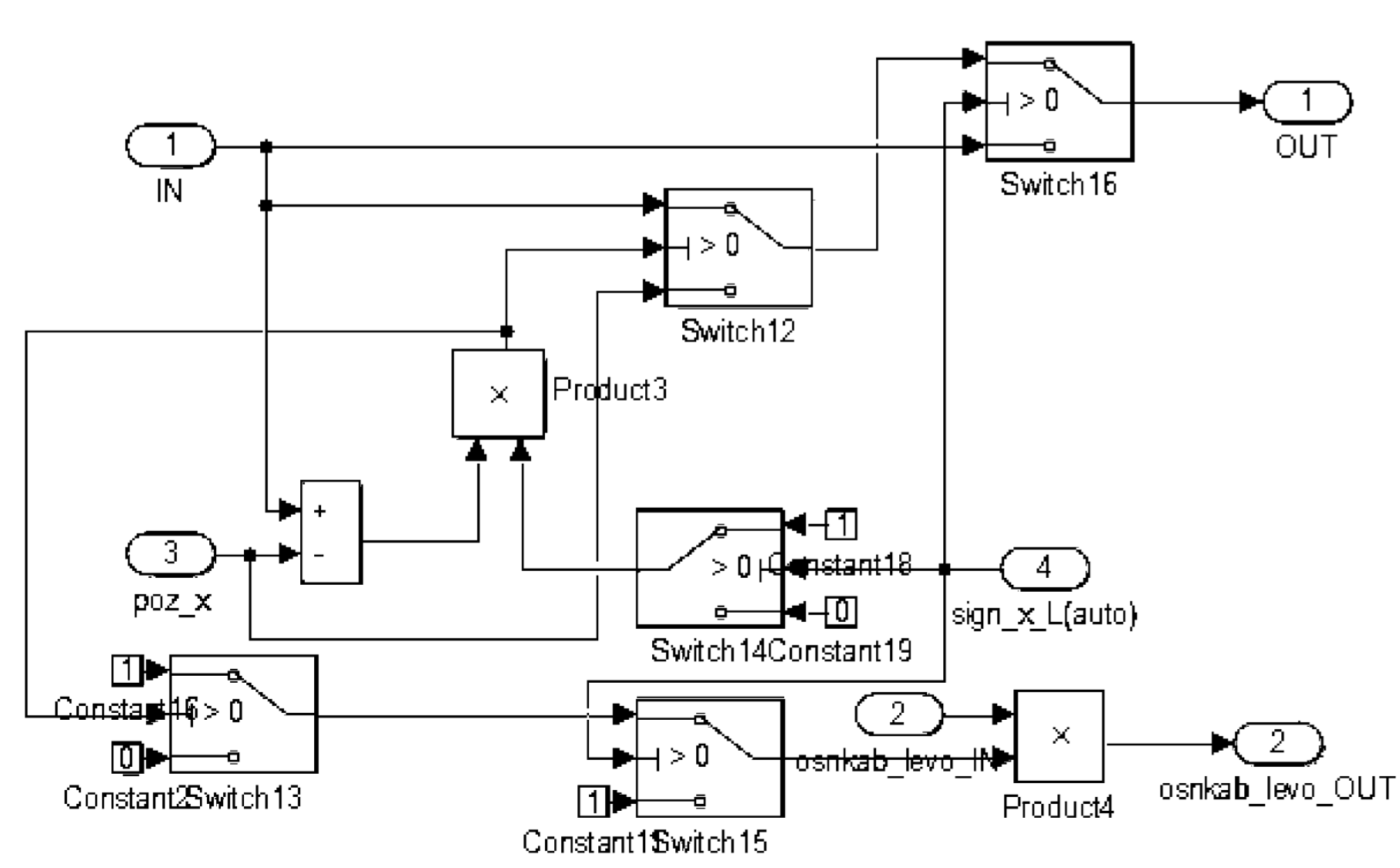
Simulink модель крана



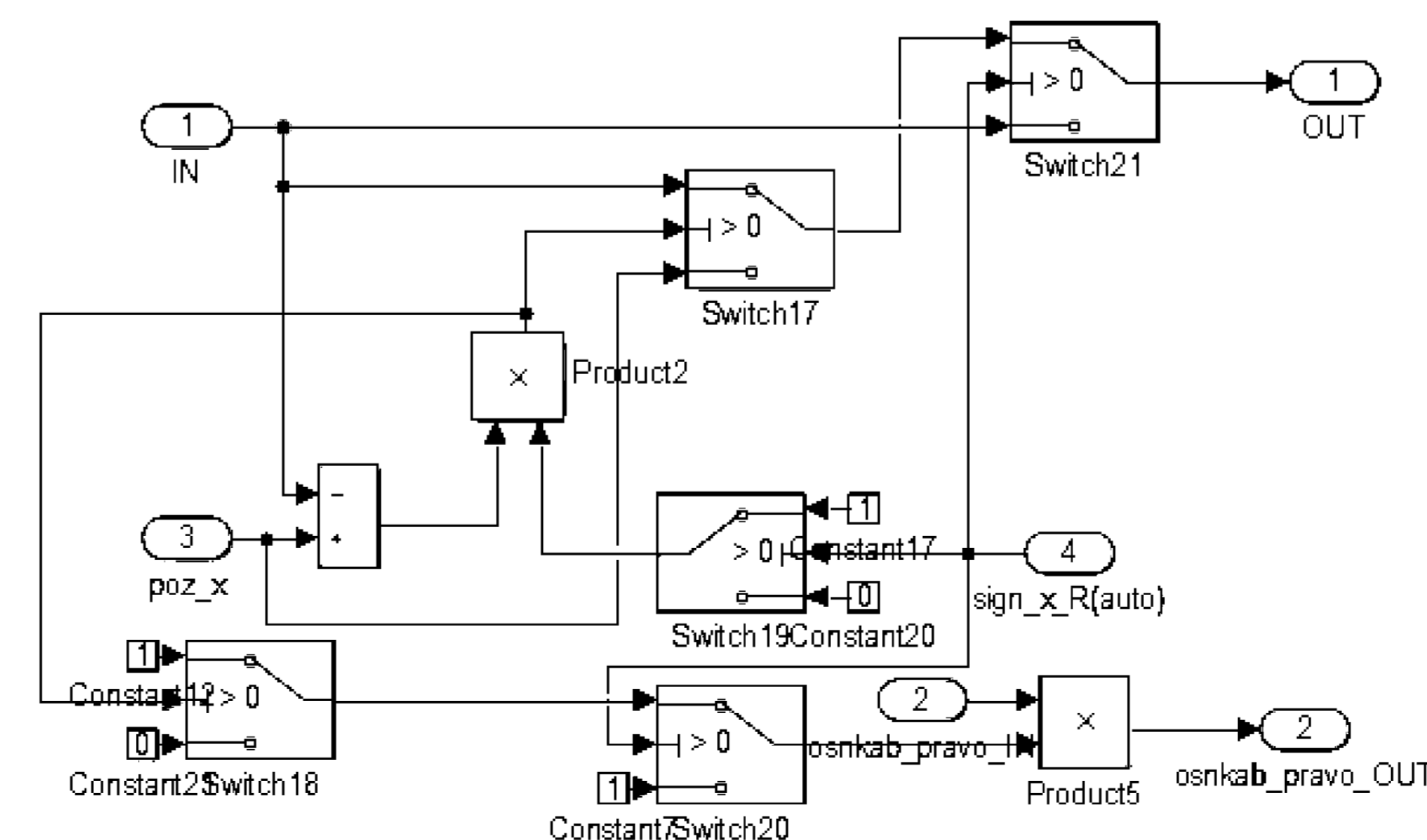
Блок управление по оси X



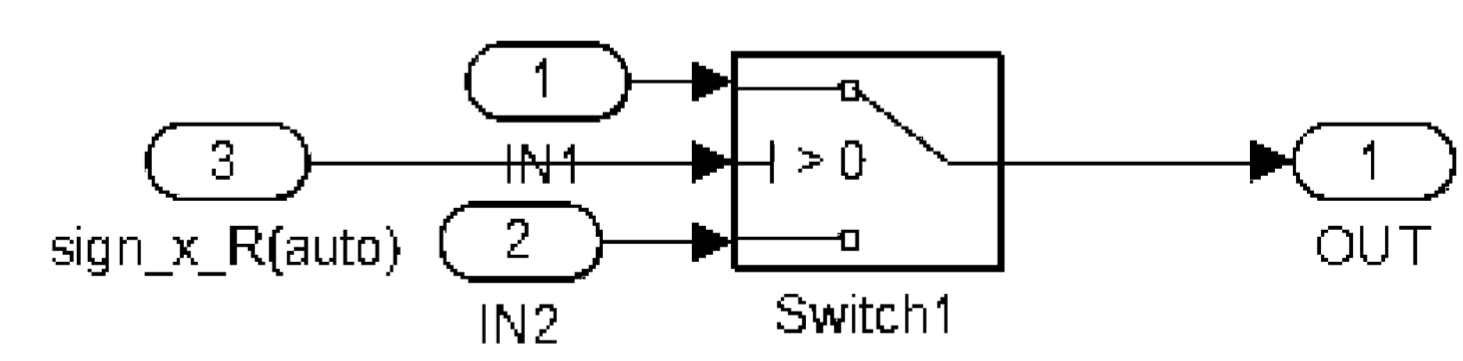
Блок Subsystem4



Блок levo_UP



Блок право_DOWN



Блок Sysubsystem

				BKP.164023.15.03.04.CX			
Изм	Лист	№ докум.	Подп.	Дата	Литера	Масса	Масштаб
		Разраб.	Алеко М.А.		у		1:1
		Провер.	Рыбалева А.Н.				
		Т.Контр.	Рыбалева А.Н.		Лист 4	Листов 6	
		Рецензент			Разработка имитационных моделей систем управления движением (комплексная выпускная квалификационная работа)		
		Н.Контр.	Скрипко О.В.		АМГУ		
		Утвержд.	Скрипко О.В.		Кафедра АППиЭ		

```

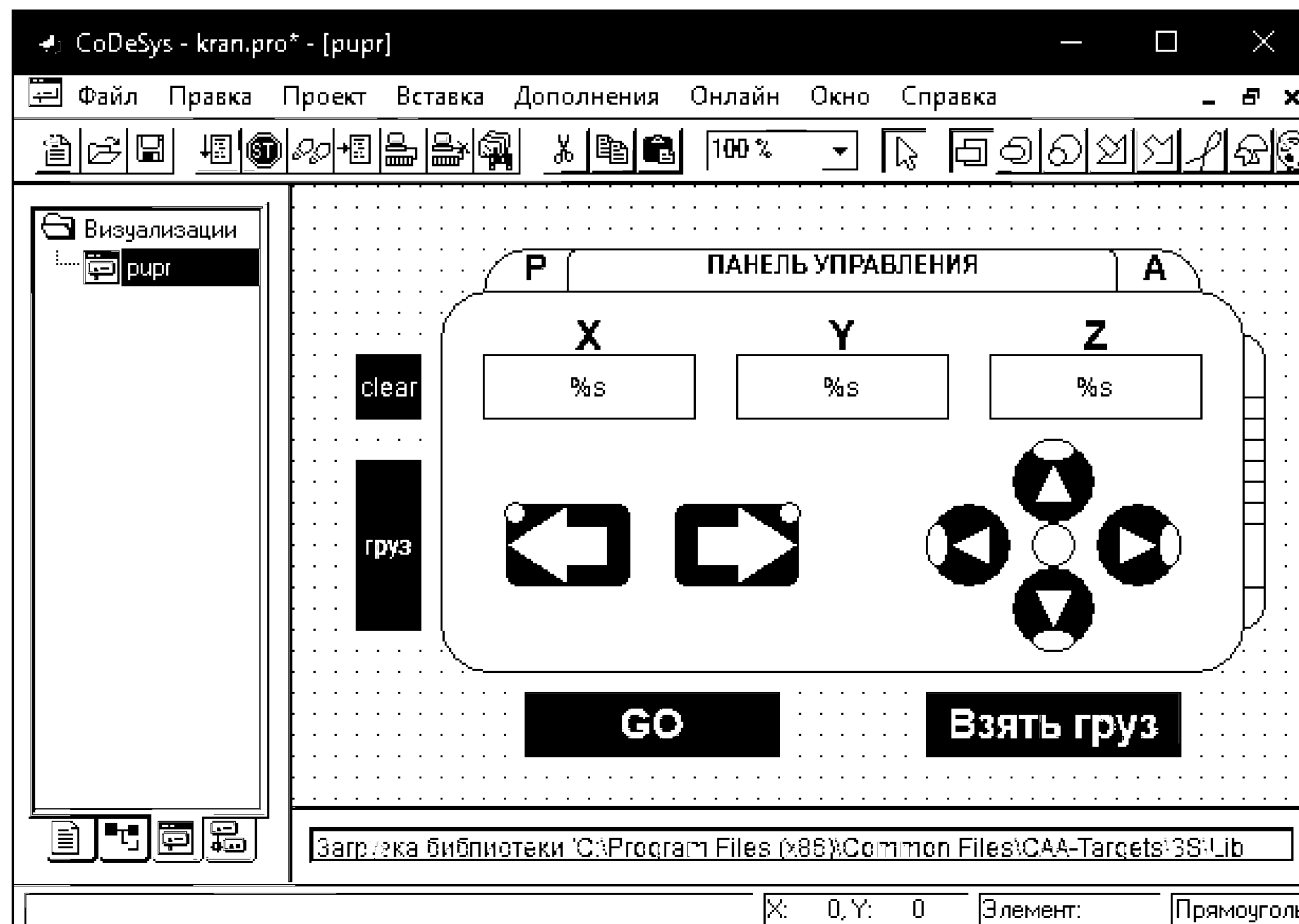
0001 PROGRAM PLC_PRG
0002 VAR
0003 END_VAR
0004
0005
0006
0007
0008
0009
0010
0011
0001 r_upr;
0002 Skor;
0003 Kont_ri;
0004 a_upr;
0005 IF R_U = TRUE THEN
0006   upr := FALSE;
0007 ELSIF A_U = TRUE THEN
0008   upr := TRUE;
0009 END_IF
0010
0011
    
```

Программы

```

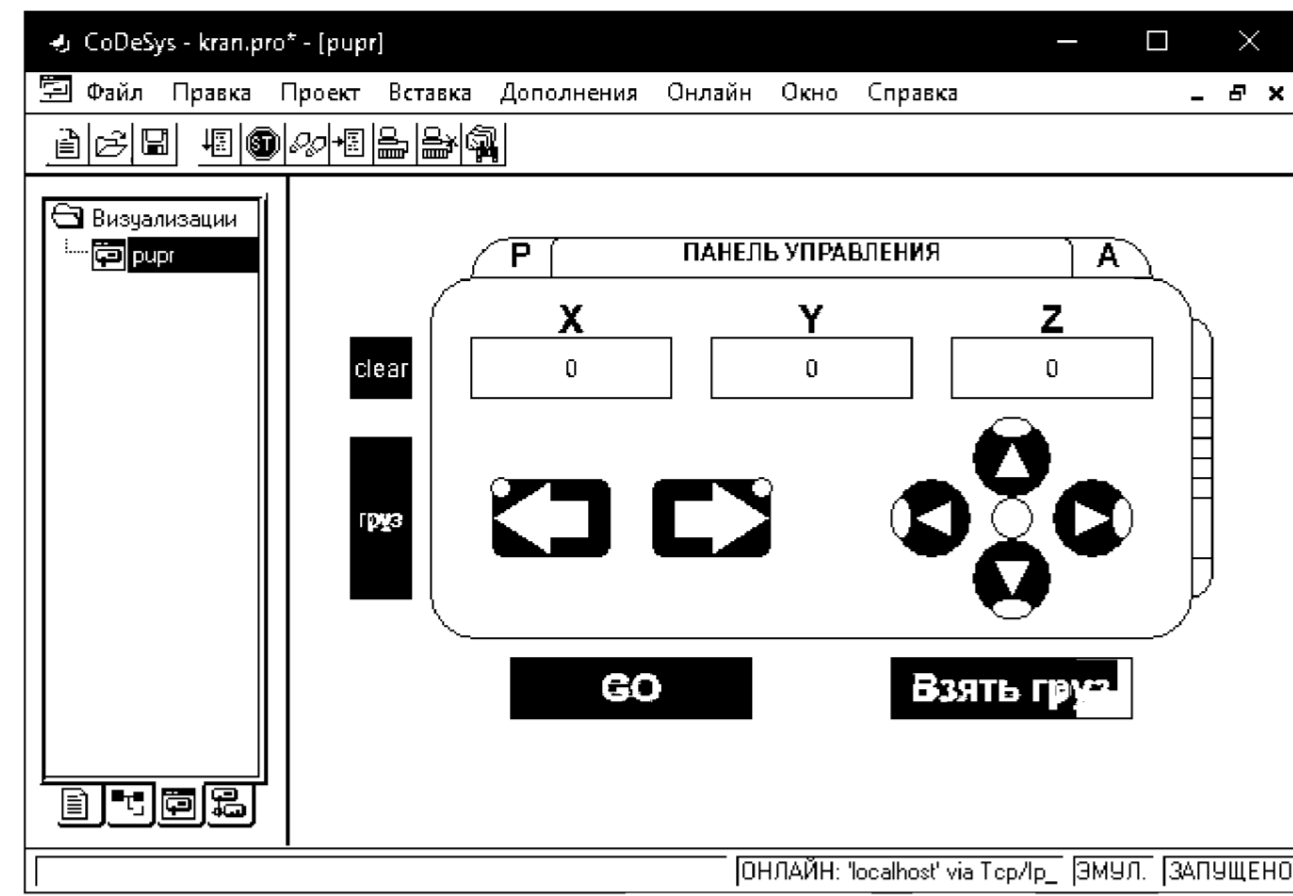
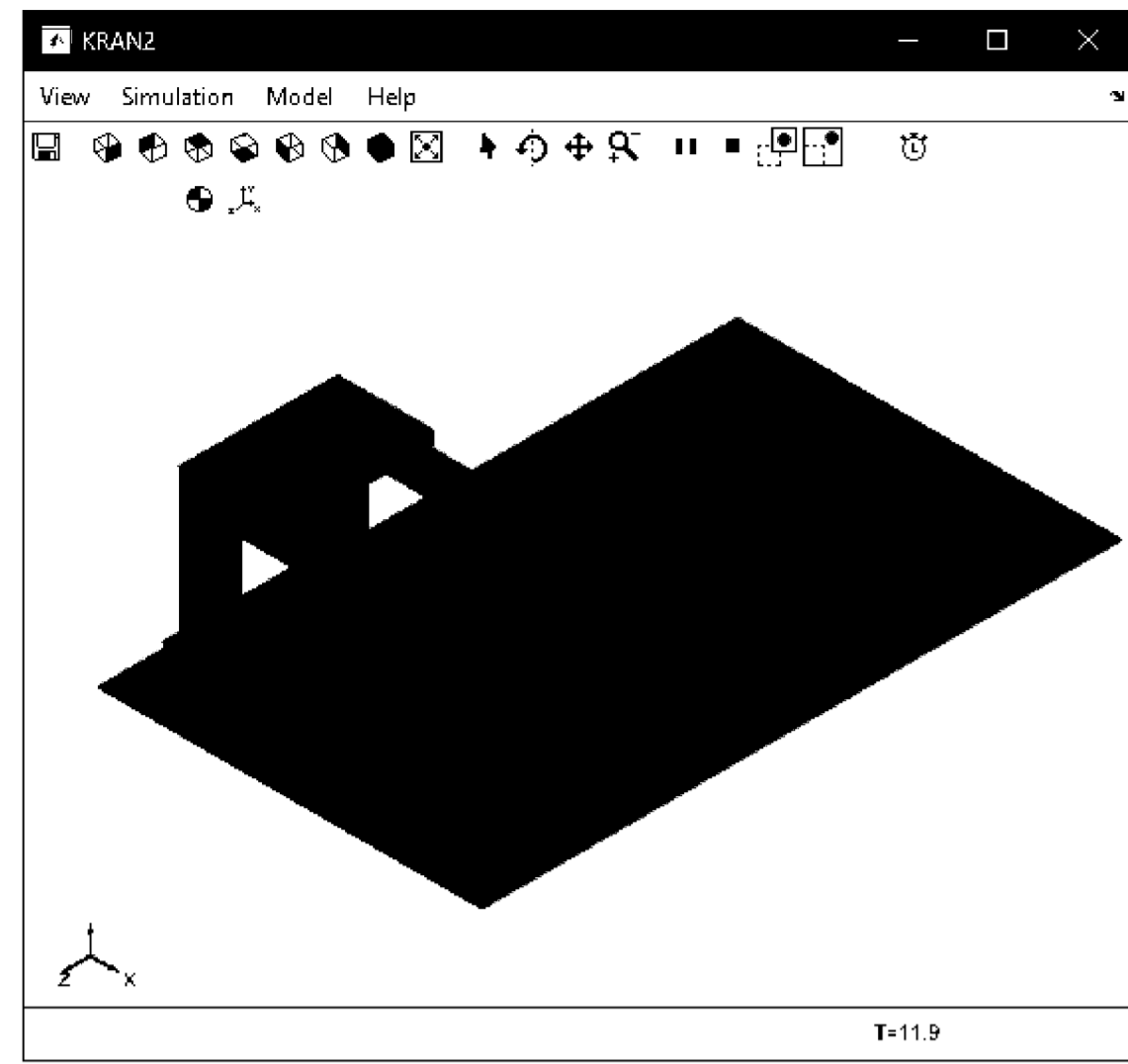
0001 VAR_GLOBAL
0002   kran_levo, kran_pravo, osnkab_levo, osnkab_pravo, kruk_up, kruk_down:BOOL;
0003   BTN_kran_levo, BTN_kran_pravo, BTN_osnkab_levo, BTN_osnkab_pravo, BTN_kruk_up, BTN_kruk_down, BTN_gruz, BTN_vzjat:BOOL;
0004   BTN_speed, BTN_speed1, BTN_speed2, BTN_pol, BTN_npol, BTN_clear:BOOL;
0005   gruz, vzjat, v, p, gruz1, vzjat1, v1, p1, gruz2, vzjat2, v2, p2, gruz3, vzjat3, v3, p3, gruz4, vzjat4, v4, p4, gruz5, vzjat5, v5, p5, kon_1, kon_2:BOOL;
0006   sign_x_R, sign_x_L, sign_y_UP, sign_y_DOWN, sign_z_R, sign_z_L:BOOL;
0007   R_U, A_U, upr, GO, np:BOOL;
0008
0009   kon:REAL;
0010   X, Y, Z:REAL;
0011   speed, poz_x, poz_y, poz_z:REAL;
0012 END_VAR
0013
0014
0015
0016
    
```

Глобальные переменные

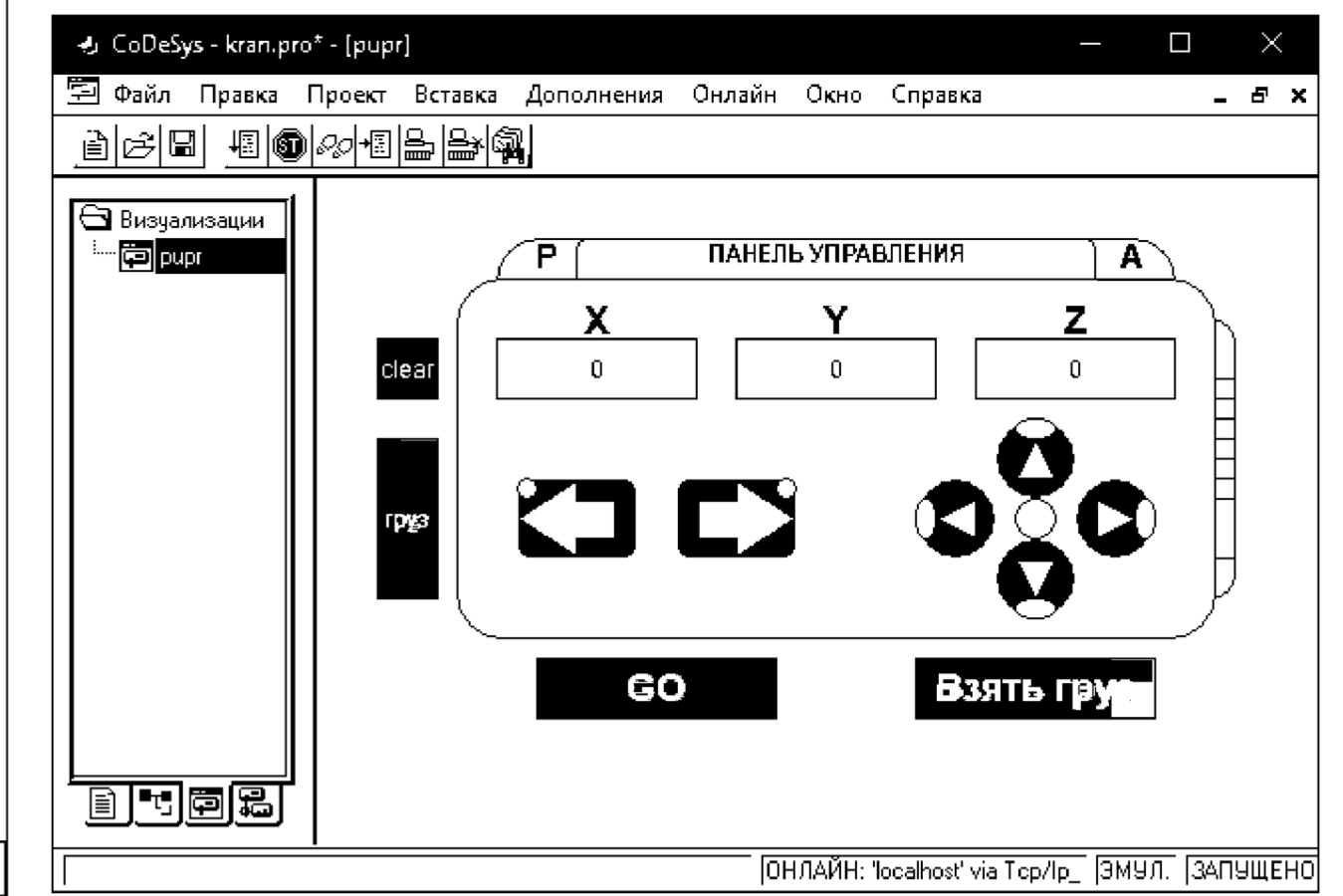
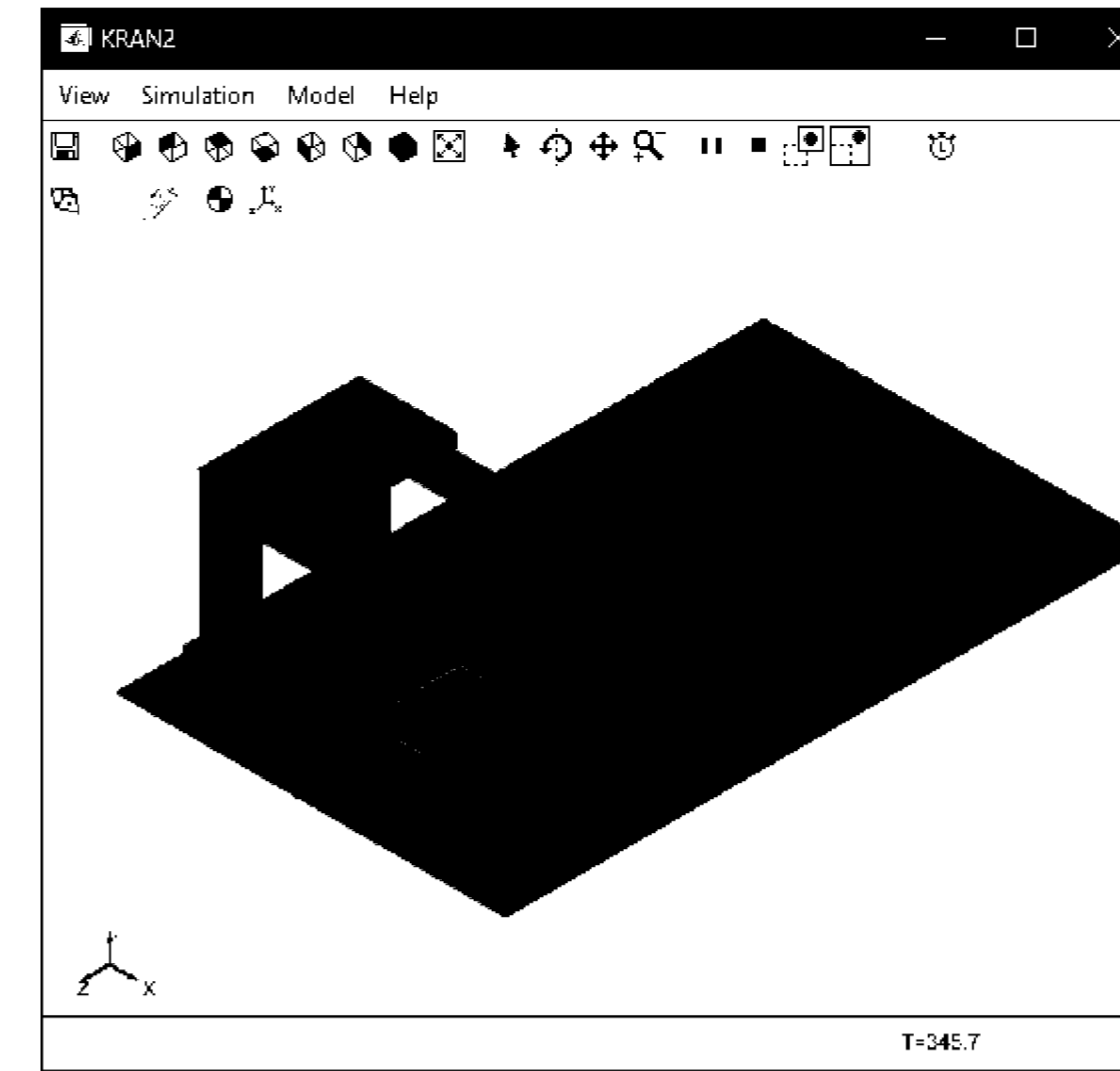


Визуализация панели управления движением крана

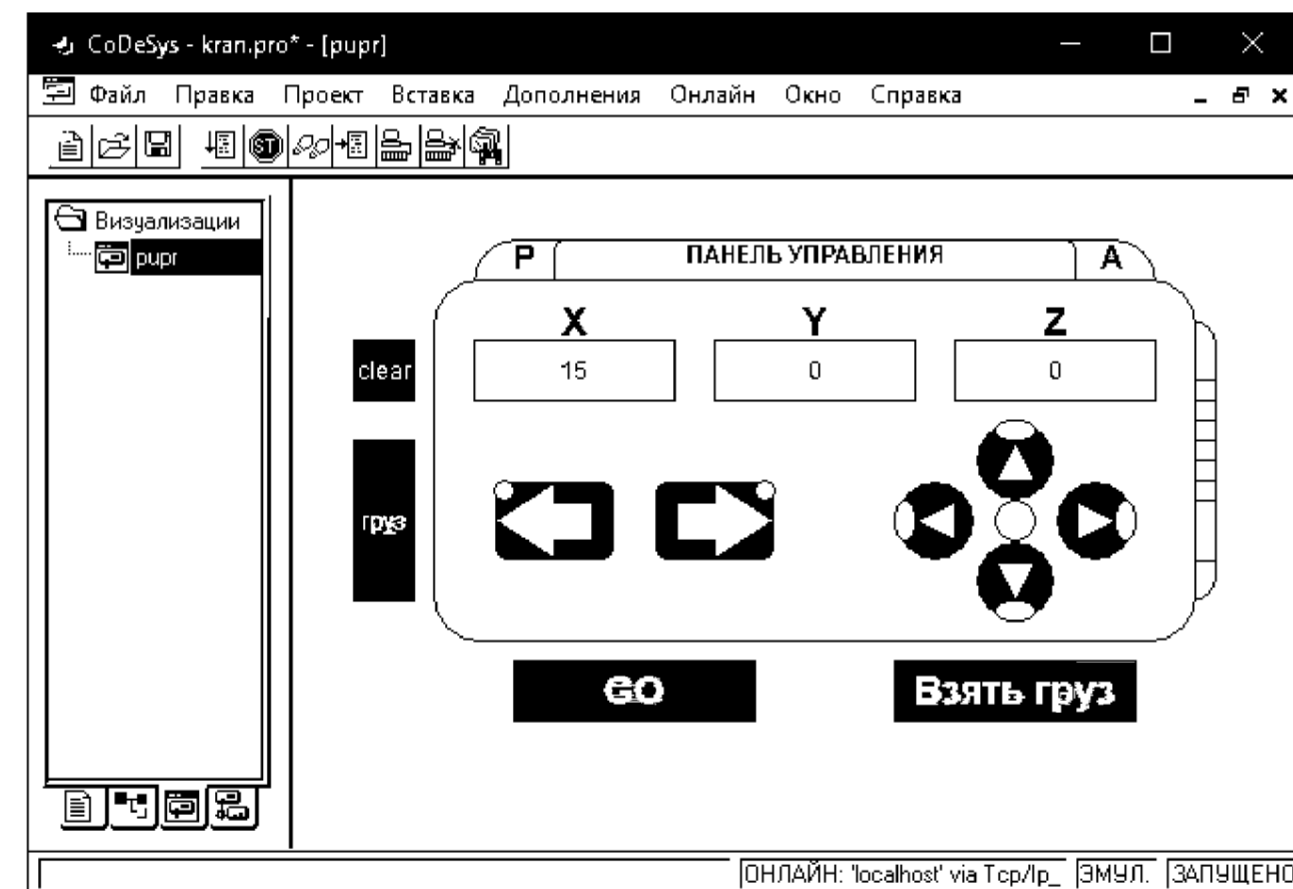
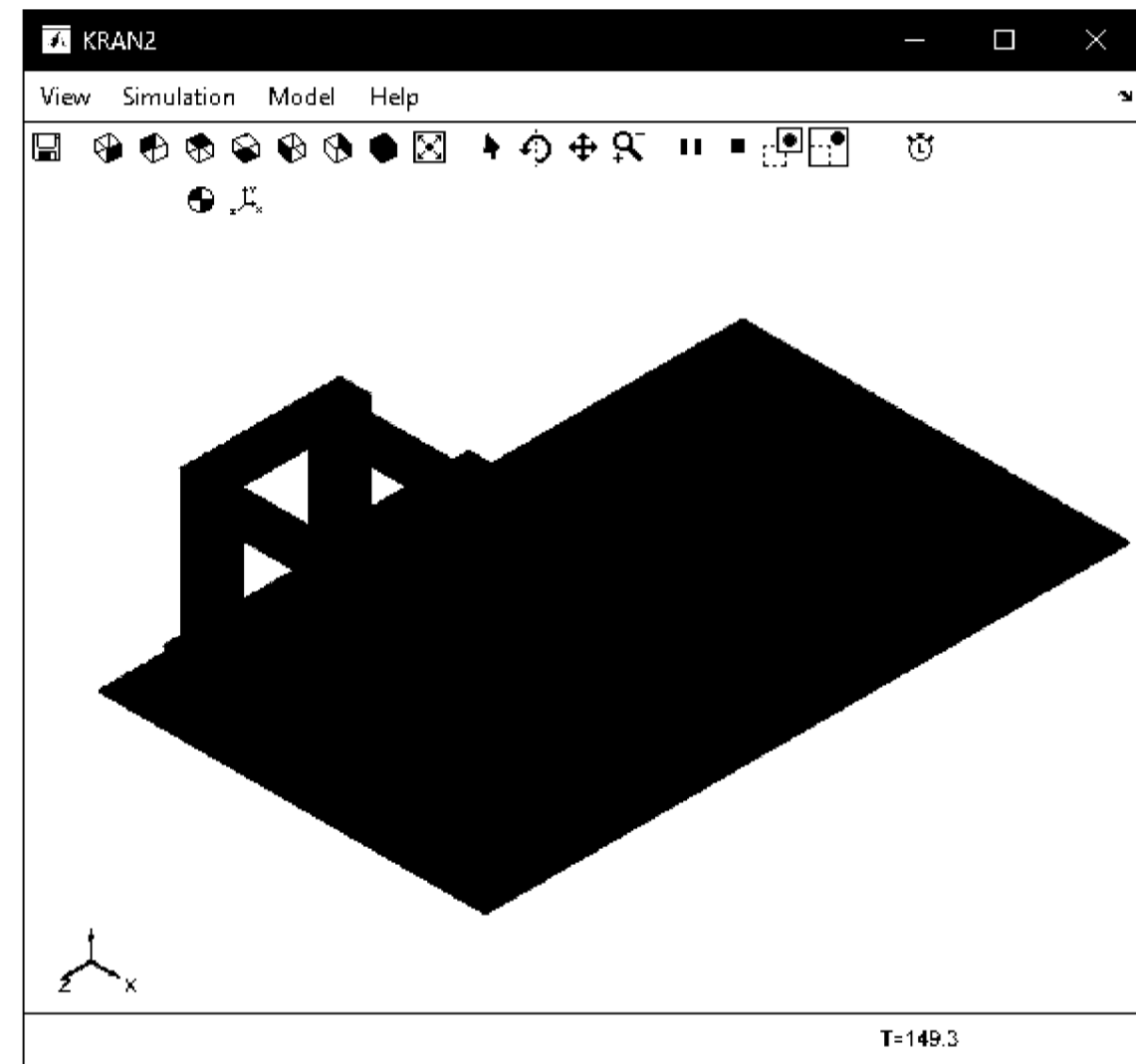
				ВКР.164023.15.03.04.CX			
Изм.	Лист	№ докум.	Подп.	Дата	Разработка программы управления краном в CoDeSys		
Разраб.	Алеко М.А.						
Провер.	Рыбалева А.Н.				Литера	Масса	Масштаб
Т.Контр.	Рыбалева А.Н.				у		1:1
Рецензент					Лист 5	Листов 6	
Н.Контр.	Скрипко О.В.				Разработка имитационных моделей систем управления движением (комплексная выпускная квалификационная работа)		
Утвержд.	Скрипко О.В.				АМГУ Кафедра АППиЭ		



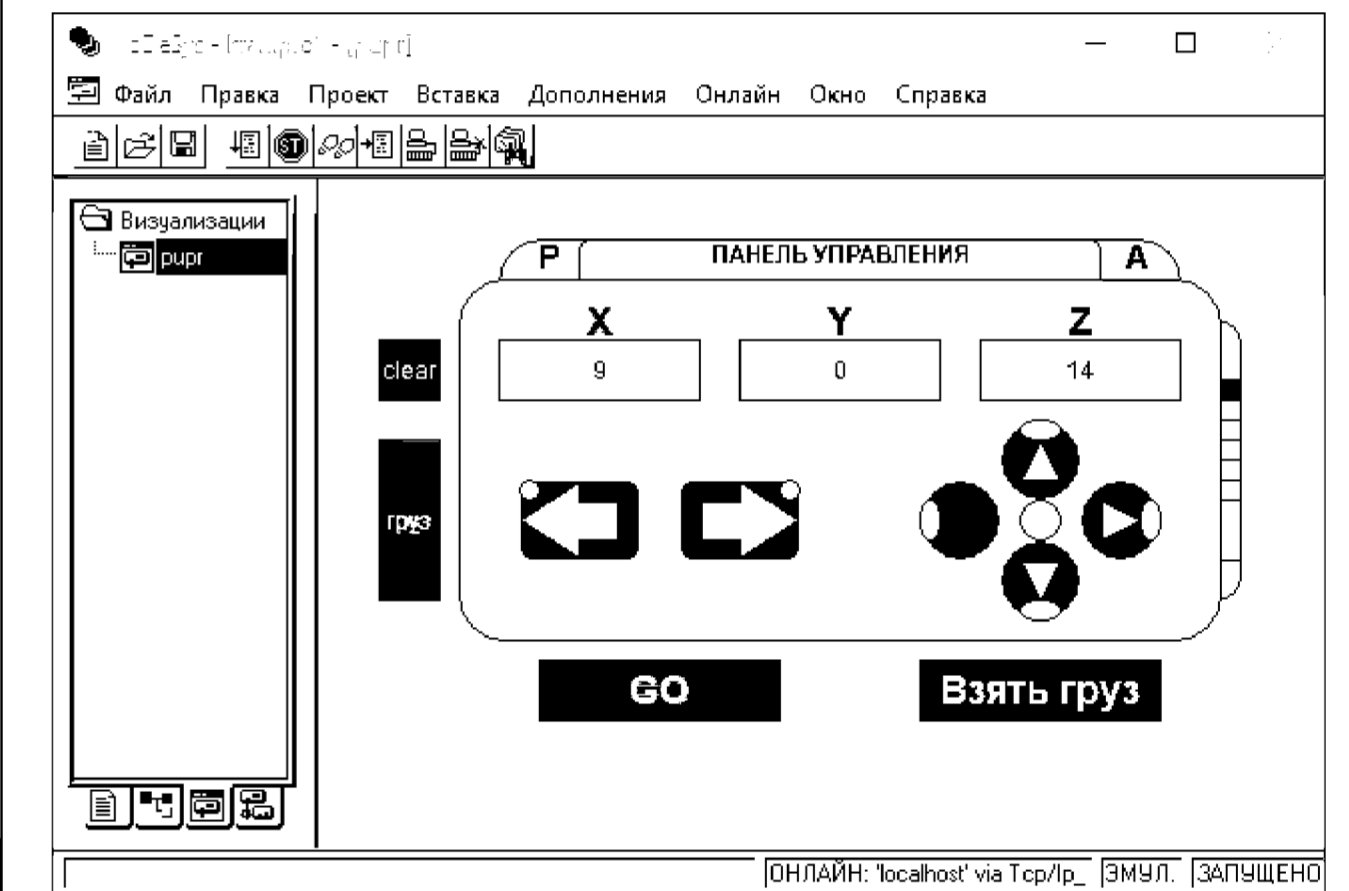
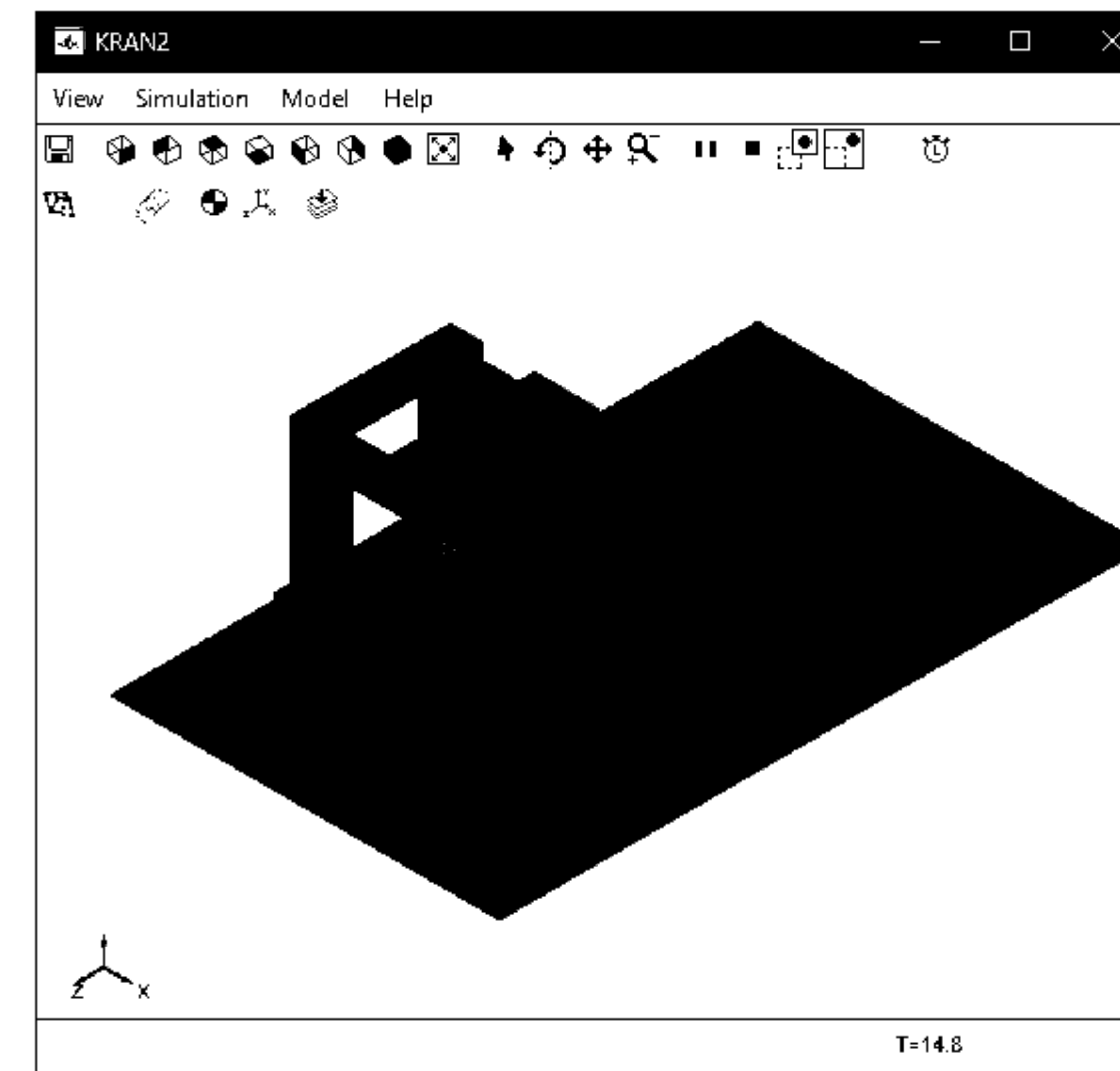
Начальное положение



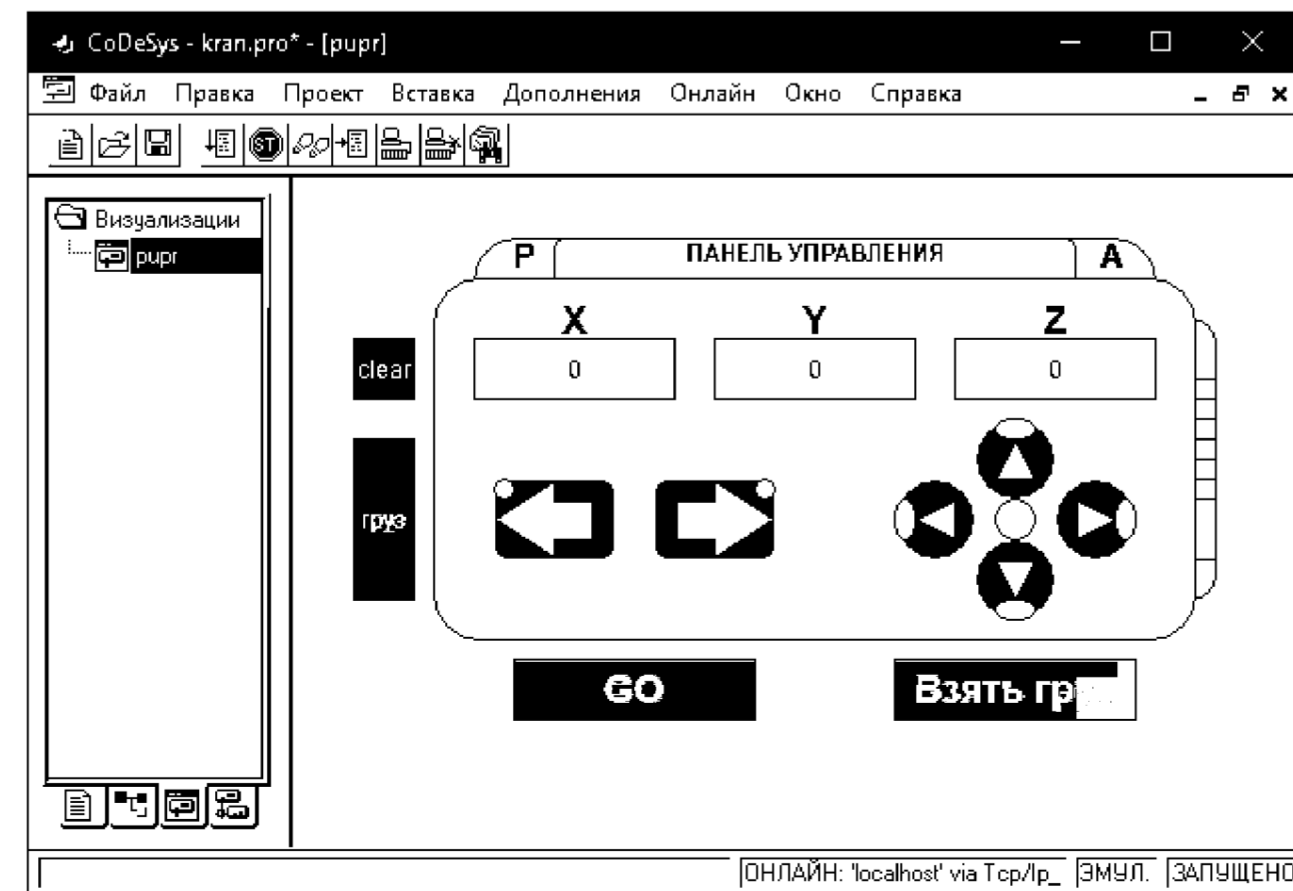
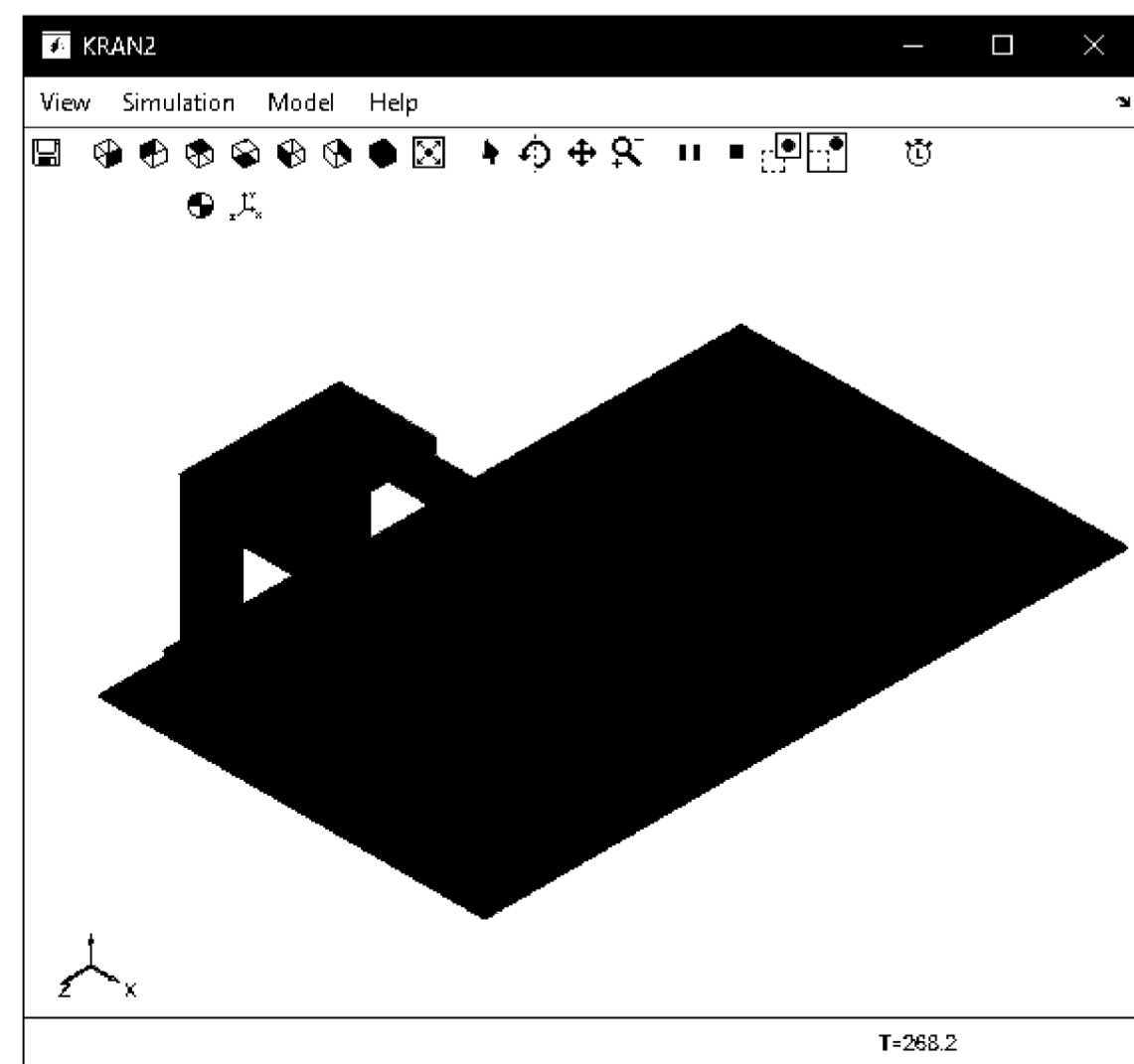
Груз



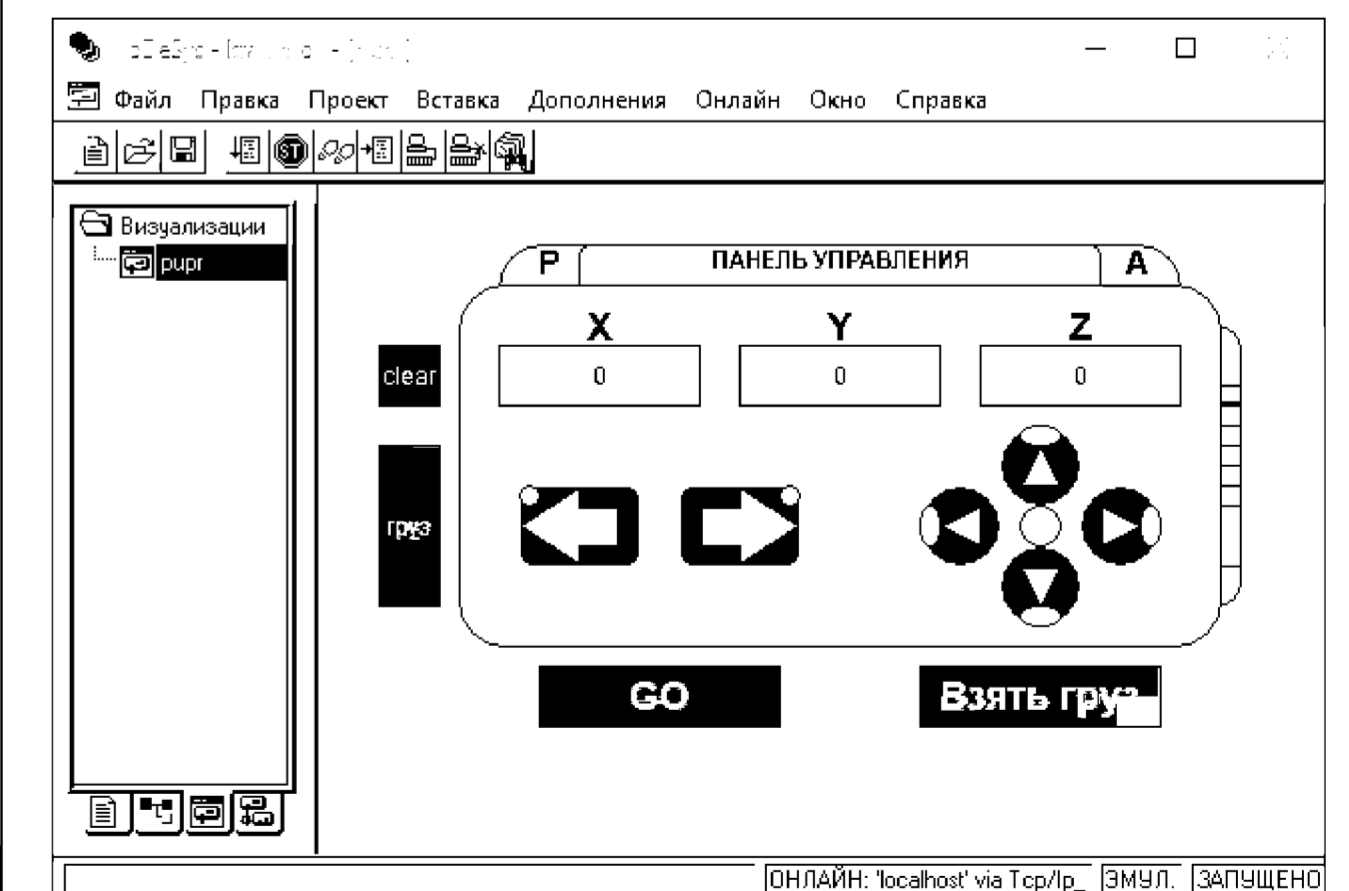
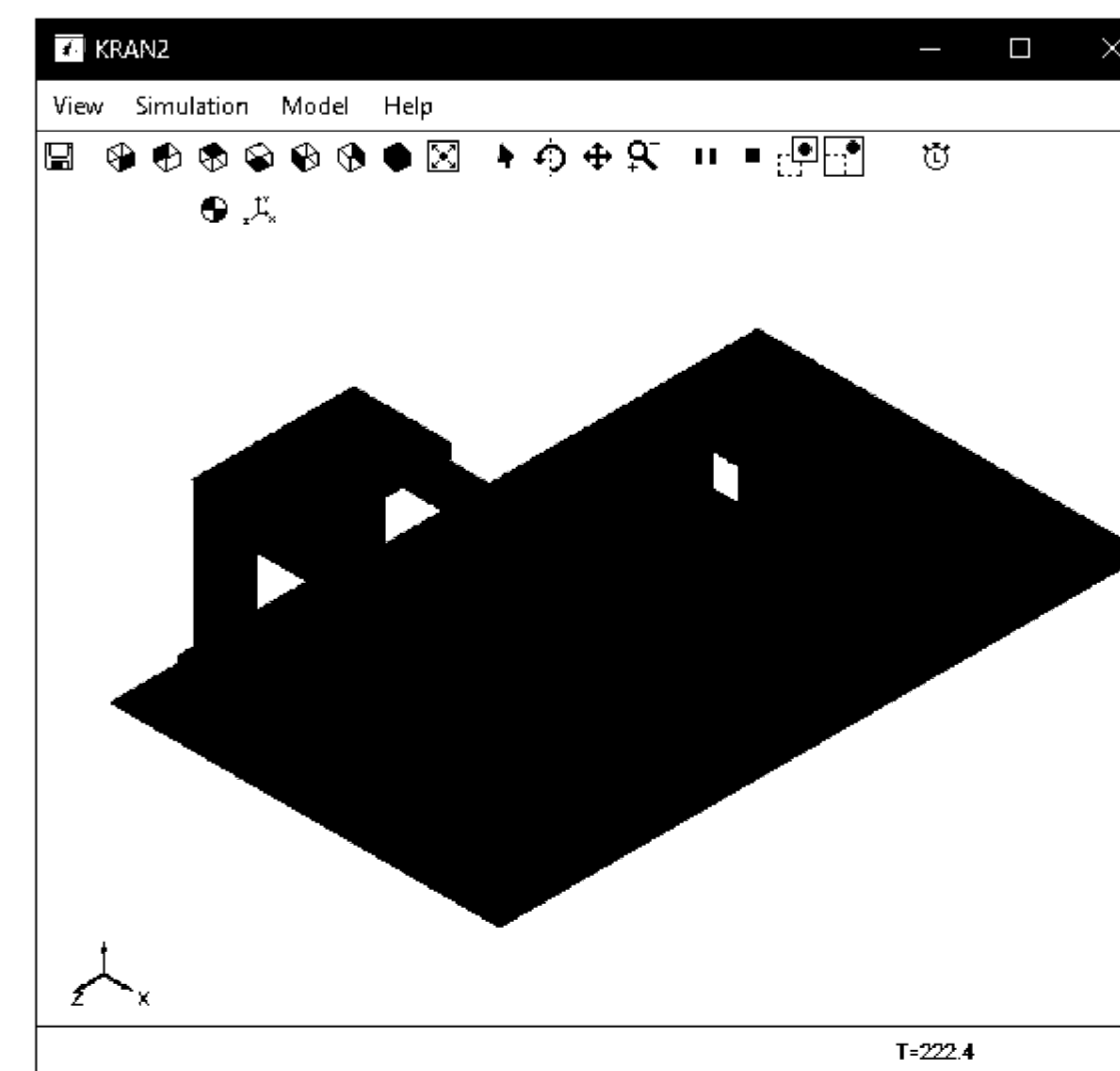
Позиция для взятия груза



Перенос груза



Автоматический режим



Груз на своем месте

				ВКР.164023.15.03.04.СХ		
Изм.	Лист	№ докум.	Подп.	Дата	Демонстрация работы имитационной модели системы управления движением крана	
		Разраб.	Алеко М.А.		Литера	Масса
		Провер.	Рыбалева А.Н.		у	1:1
		Т.Контр.	Рыбалева А.Н.		Лист 6	Листов 6
		Рецензент			АМГУ	
		Н.Контр.	Скрипко О.В.		Кафедра АППиЭ	
		Утвержд.	Скрипко О.В.		Разработка имитационных моделей систем управления движением (комплексная выпускная квалификационная работа)	