

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Инженерно-строительный институт
Кафедра строительных материалов и технологии строительства

УТВЕРЖДАЮ
Заведующий кафедрой

подпись

«_____» _____ 2020г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

**«Совершенствование рабочего процесса строительного работа при
монтаже строительных конструкций»**

08.04.01 «Строительство»

08.04.01.07 «Комплексная механизация и автоматизация строительства»

Научный руководитель _____ доц., канд. техн. наук А.П. Прокопьев

подпись

Выпускник _____ ст. гр. СФ18-07М Н.Ю. Андреев

подпись

Рецензент _____ проф., д-р. техн. наук В.И. Иванчура

подпись

Нормоконтролер _____ доц., канд. техн. наук Т.С. Турышева

Красноярск 2020

РЕФЕРАТ

Выпускная квалификационная работа по теме «Совершенствование рабочего процесса строительного робота при монтаже строительных конструкций» содержит 101 страниц текстового документа, 50 иллюстрации, 1 приложение и 39 использованных источников.

СТРОИТЕЛЬНЫЙ РОБОТ, РОБОТИЗАЦИЯ, РАБОЧИЙ ПРОЦЕСС, МОДЕЛИРОВАНИЕ, ДИНАМИКА СЛЕДЯЩИХ ПРИВОДОВ, ДВИГАТЕЛЬ ПОСТОЯННОГО ТОКА, СИНТЕЗ РЕГУЛЯТОРА, ARDUINO, SIMULINK.

Объект исследования – приводы исполнительных механизмов строительных роботов.

Предмет исследования – переходные процессы в приводе исполнительного механизма строительных роботов.

Целью работы является исследование и обоснование способов усовершенствования следящих приводов строительных роботов с учетом динамики рабочего процесса, для повышения их эффективности.

В результате были поставлены и решены следующие задачи:

- выполнен анализ существующих конструкций и научных работ по темам строительных роботов, обоснованы варианты для исследования;

- получена математическая модель динамики строительного робота-монтажника;

- разработана имитационная модель строительного робота-монтажника с применением программной среды MATLAB/Simulink;

- подготовлена методика компьютерного моделирования и экспериментального исследования объекта;

- выполнено компьютерное моделирование рабочего процесса и эксперимент с объектом исследования;

- проведен анализ полученных в магистерской диссертации результатов и предложены варианты совершенствования строительного робота-монтажника.

Научная новизна заключается в подходах к изучению, с использованием актуальных методов и программных комплексов.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Анализ роботизации в строительстве.....	8
1.1 Строительные процессы и задачи роботизации.....	10
1.2 Анализ конструкций строительных манипуляторов.....	13
1.3 Основные тенденции развития строительных манипуляторов.....	20
2 Теоретические исследования.....	22
2.1 Моделирование динамики следящих приводов.....	26
2.2 Проектирование модели системы управления следящим приводом строительного робота.....	36
2.3 Математическое описание модели двигателя постоянного тока.....	45
2.4 Синтез регулятора в следящей системе постоянного тока робота.....	50
3 Экспериментальные исследования.....	55
3.1 Методика.....	55
3.2 Экспериментальный стенд.....	58
3.3 Технические средства автоматизации.....	60
3.4 Обработка результатов экспериментальных исследований.....	65
4 Реализация результатов работы.....	68
4.1 Реализация системы управления приводом захватного устройства на базе Arduino.....	69
4.2 Элементы объектно-ориентрованного программирования.....	73
4.3 Запуск моделей Simulink на Arduino.....	74
ЗАКЛЮЧЕНИЕ.....	77
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	79
ПРИЛОЖЕНИЕ А. ПУБЛИКАЦИИ.....	83
ПРИЛОЖЕНИЕ Б.....	84

ВВЕДЕНИЕ

Строительная отрасль – одна из самых консервативных, сложно воспринимающих инновации, но в тоже время одна из ведущих сфер современной экономики. С каждым годом новые строительные проекты становятся еще масштабнее и сложнее. Сами строительные работы отличаются повышенной сложностью, опасностью и монотонностью, что в совокупности усложняет задачу рабочих. При этом успех строительного проекта зависит от трех факторов – стоимость, сроки, качество. Для успешной реализации проектов нужно совершенствовать технологии и организацию строительного производства с помощью перспективных отечественных и зарубежных технологий.

На данное время одной из наиболее прогрессивных технологий является роботизация, которая позволяет ускорить сроки строительства, повысить безопасность и уменьшить количество ручного труда. В данной работе рассматривается опыт применения строительных роботов и обзор их современных разновидностей. При применении роботизированных систем роль человека заключается в подготовке исходных данных, выборе алгоритма работы и в анализе полученных результатов.

Широкое распространение в современной технике получили автоматизированные комплексы, в состав которых входят и строительные роботы. Развитие научно-исследовательских и опытно - конструкторских работ в этом направлении проводятся в сторону создания качественно новых робототехнических систем, например, роботов с высоким искусственным интеллектом и мобильных роботов различного назначения.

С середины 80-х годов XX-го столетия задачи роботизации и автоматизации строительных работ становятся в центре внимания ученых и специалистов научно-исследовательских и строительных организаций. Среди них ведущую роль занимают ЦНИИОМТП, ВНИИстройдормаш. МИСИ. ЮРГТУ (НПИ), Спецжелезобетонстрой, Мюнхенский технологический

университет. Решение проблемы автоматизации и роботизации строительных операций базируется на трудах ученых Макарова И.М., Фролова К.В., Попова Е.Л., Юревича Е.И., Кулешова А.И., Локоты Н.А., Ющенко А.С., Крутько П.Д., Медведева В.С., Подураева Ю.В., Зенкевич С.А., Лохина В.М., Бурдакова С.Ф., Кореидясева А.И., Тугенгольда А.К., Жавпер В.Л., Каляева И.А., Петракова В.А. и др., внесших значительный вклад в становление и развитие современных теоретических основ робототехники и мехатроники.

Существенный вклад в решение задач роботизации строительных операций внесли ученые Евдокимов В.А., Вильман Ю.А., Воробьев В.А., Загороднюк В.Т., Булгаков А.Г., Максимычев О.И., Френкель Г.Ю., Гудиков Г.Г., Бок Т. и др. Однако, несмотря на широкий спектр научных и проектных разработок в области строительной робототехники уровень автоматизации и роботизации строительных операций остается достаточно низким. Это связано с необходимостью систематизации выполненных исследований и разработок, проведения комплексных научно-исследовательских и опытно-конструкторских разработок. Необходимо решать задачи структурной организации строительных роботов и робототехнических комплексов, искать оптимальные алгоритмы решения задач кинематики и динамики манипуляторов, заниматься разработкой методов планированием траекторий движения роботов при выполнении различных строительных операций, разрабатывать алгоритмы управления, обеспечивающие функционирование строительных роботов в условиях стохастической и недетерминированной среды.

Моделирование динамики роботов-монтажников является одной из сложных задач. Это связано с проблемой составления динамических уравнений для пространственных механизмов, а также их интегрированием. С появлением систем компьютерной алгебры упростился процесс получения уравнений, и появились эффективные аналитические алгоритмы для кинетического и динамического анализа таких систем.

Настоящая работа посвящена совершенствованию рабочего процесса строительного робота при монтаже строительных конструкций.

Таким образом, целью работы является разработка методического инструментария, обеспечивающего возможность расчёта рабочих режимов строительного робота для его совершенствования.

Используя опыт внедрения манипуляторов и строительных роботов в стройиндустрии, анализируя технологические особенности производства в этой отрасли, можно сформулировать следующие основные задачи совершенствования роботизации технологических процессов в строительстве:

- анализ существующих конструкций и научных работ по темам строительных роботов, обосновать вариант для исследования;
- получить математическую модель динамики строительного робота;
- разработать имитационную модель строительного робота с применением программной среды MATLAB/Simulink;
- подготовить методику компьютерного моделирования и экспериментального исследования объекта;
- выполнить компьютерное моделирование рабочего процесса и эксперимент с объектом исследования;
- предложить варианты совершенствования строительного робота.

Методы исследования. Теоретические и экспериментальные исследования и практические результаты основываются на системном подходе, законах классической механики, теоретических основах робототехники и мехатроники, методах математического анализа, классической и современной теории управления, методах компьютерного моделирования, теории планирования эксперимента. Полученные результаты проверялись компьютерным моделированием и экспериментальными исследованиями.

Обоснование достоверности научных положений выводов и результатов обусловлена корректным использованием фундаментальных законов физики, механики, робототехники, мехатроники, классической теории управления, теории электропривода, корректными допущениями при составлении

математических моделей, удовлетворительной сходимостью теоретических и экспериментальных результатов и подтверждается результатами компьютерного моделирования, при этом максимальная ошибка отклонения расчетных и экспериментальных данных не превышает 7,8%.

Научное значение результатов исследований. Совокупность разработанных в диссертации структур, моделей, методов планирования и управления движением представляет собой методологические основы построения роботизированных и мехатронных комплексов усовершенствования рабочего процесса строительного робота при монтаже строительных конструкций, которые расширяют теорию робототехники и мехатроники и могут рассматриваться как новое самостоятельное направление прикладной робототехники. Теоретическая значимость научных результатов состоит в том, что предложены, обоснованы и экспериментально подтверждены методы структурного построения и анализа, моделирования динамики и управления движением для нового класса строительных роботов, отличающихся структурой, технологией взаимодействия с рабочей средой, требованиями к управлению.

Апробация работы.

Публикации. Основные положения диссертации опубликованы в 5 научных трудах, в том числе по материал работы опубликовано 4 статьи, получено 1 авторское свидетельство на изобретение.

1 Анализ роботизации в строительстве

Современные тенденции развития отрасли строительства диктуют свои условия: сроки строительных работ должны сокращаться, а прибыль увеличиваться. Поэтому современным строительным предприятиям, которые хотят оставаться конкурентно способными на мировом рынке, необходим переход к новым технологиям. Одной из передовых технологий является применение роботов-манипуляторов.

Манипулятор – управляемая машина, для выполнения двигательных функций, аналогично движениям руки человека. Схема манипулятора представлена на рисунке 1.

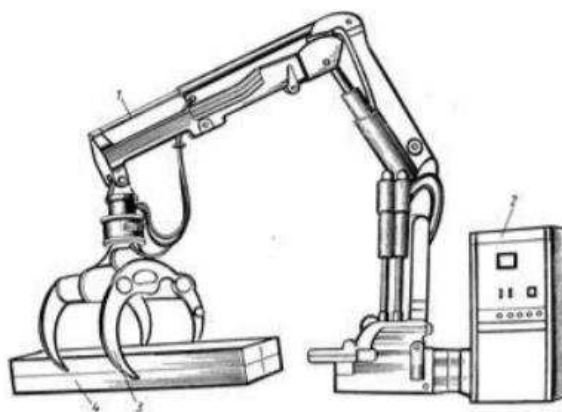


Рисунок 1 – Схема робота-манипулятора

На рынке строительной техники в России, появился новый вид машин – строительные роботы. По прогнозам специалистов, уже в скором будущем дистанционно управляемая и автоматическая техника займет место обычной строительной техники. Малые габариты и весовые характеристики подобной техники делают удобной и ее транспортировку. Роботом при помощи пульта может управлять всего один квалифицированный оператор. Поэтому подобная технология уместна в ситуациях, потенциально опасных для здоровья рабочих. Анализируя все многообразие строительных манипуляторов и роботов, представленных сейчас на рынке, в зависимости от назначения,

технологической и структурной особенности можно разделить на несколько групп.

Первая группа – манипуляторы, роботы для выполнения монтажных работ. Средства роботизации этой группы характеризуются широким диапазоном грузоподъемности, мобильности, повышенной точностью позиционирования. В состав этой группы включаются манипуляторы и роботы для монтажа элементов зданий, перегородок, монтажа оборудования и строительных лесов.

Вторая группа – образуются из строительных манипуляторов и роботов для бетонных работ. Эта группа подразделяется на три подгруппы: манипуляторы и роботы для укладки и уплотнения бетонной смеси, средства роботизации арматурных работ на объекте, манипуляционное оборудование для монтажа и демонтажа строительных опалубок.

Третья группа – строительных манипуляторов и роботов составляют средства роботизации отделочных работ. В состав этой группы включаются манипуляторы и роботы для выполнения малярных, штукатурных, облицовочных работ, нанесения изоляционных мастик на кровли, а также для устройства монолитных полов. Эта группа отличается небольшой грузоподъемностью механизмов, мобильностью конструкций, использования программного управления и необходимостью сенсорных устройств.

Четвертую группу – составляют манипуляторы и роботы для производства земляных работ. В нее входят многофункциональное манипуляционное оборудование, предназначенное для устройства котлованов, засыпки и уплотнения грунта, прокладки коммуникаций. Оборудование этой группы снабжается дистанционным, программным или комбинированным управлением.

Пятая группа – это манипуляторы и роботы для погрузочно-разгрузочных работ. В составе этой группы манипуляторы и роботы различной грузоподъемности для погрузки и разгрузки строительных конструкций и элементов, контейнеров и пакетов, а также мелкоштучных грузов. Такие

манипуляторы могут устанавливаться на транспортных средствах, самоходных шасси. На настоящий момент на рынке роботов представлено достаточное количество моделей, выполняющих множество операций при строительстве, реконструкции зданий и сооружений. Эти промышленные образцы способны заменить ручной труд на опасных утомительных работах.

1.1 Строительные процессы и задачи роботизации

Строительные работы – это совокупность сложных и простых строительных процессов, в результате продвижения, которых во времени и пространстве, создаётся строительная продукция. Строительные работы можно разделить на общестроительные и специальные.

Рассмотрим процессы, связанные с общестроительными работами.

По названиям основных видов работ, можно выделить:

- 1) технологические процессы при производстве земляных и свайных работ;
- 2) технологические процессы при бетонировании, монтаже, каменной кладке;
- 3) технологические процессы при устройстве защитных и отделочных покрытий.

В связи с возрастающими требованиями потенциального потребителя продукции строительного сектора возрастает сложность и увеличиваются требования к срокам и качеству строительных работ. Для повышения эффективности операционной деятельности и успешной конкуренции.

Использование роботизированных технологии, позволит ускорить процесс возведения зданий, улучшить качество выполняемых операций и снизить затраты на рабочую силу.

За последние годы автоматизация успешно внедряется в строительную индустрию и за человеком остается только наблюдение за правильностью работы специальных приборов.

Проведен анализ использования строительных роботов и автоматизированных машин, используемых в строительном деле и получаемые положительные аспекты от их деятельности.

К общестроительным работам, а это земляные, монтажные, железобетонные и т. д. предъявляются определенные требования. Они, как правило, отличаются повышенной сложностью и монотонностью, что представляет особую трудность для рабочего с учетом того, что для получения прибыли необходимо выполнение поставленной задачи в минимальные сроки без потери в качестве. К примеру, снос зданий, их подрыв, уборка территории от мусора – все эти операции вполне доступны для автоматике. И при этом, людям не приходится работать на опасных участках, что сводит к минимуму травматизм.

Земляные работы являются одними из самых сложных строительных работ, где обойтись без специального оборудования. Роботы в этой сфере представляют собой обычные экскаваторы, управление которыми обходится без участия человека. Имеется возможность быстрой смены оборудования, это дает возможность их использования для решения различных задач - устройство ломает огнеупорные материалы, железобетонные и металлические конструкции, выгребают мусор, проводят погрузочные и аварийно-спасательные работы.

В настоящее время в различных областях человеческой жизни широко используются манипуляционные роботы (МР). Их применение позволяет повысить производительность технологических процессов, устранить присутствие людей на опасных участках, проводить операции в агрессивных средах.

Надо признать, что, несмотря на множество экспериментов в области строительной робототехники, пока накоплен небольшой опыт практического применения роботов в строительном секторе, сдерживающими факторами роботизации строительства являются:

– каждый проект уникален и требует применения стандартных и высокотехнологичных решений;

– использование роботов, например, при создании бетонных каркасов связано с необходимостью применения дорогих строительных смесей с высокой скоростью твердения, безопалубочных и безарматурных строительных технологий, иначе эффективность использования сложной техники будет утеряна;

– стоимость роботов пока очень высока.

Специалисты уверены, что в будущем роботы будут активно использоваться в малоэтажном строительстве. Применение подобной техники позволяет сократить временные и денежные затраты в несколько раз, а также обеспечивает высокую точность строительства. Например, возведение коттеджей по монолитной технологии уже роботизировано. Такие операции, как обустройство фундамента, подготовка бетонного раствора, его подача и укладка в стены и перекрытия будут производиться без участия человека, главной задачей которого станет обеспечение бесперебойной подачи компонентов бетонной смеси.

Самые роботизируемые страны на сегодняшний день согласно данным International Federation of Robotics Южная Корея, число роботов на 10 тыс. рабочих: 631; Сингапур, число роботов на 10 тыс. рабочих: 488 и Германия, число роботов на 10 тыс. рабочих: 309.

В 2015 г. в мире было 66 установленных промышленных роботов на 10 тыс. рабочих. В 2016 г. их число достигло 74 на 10 тыс. рабочих.

При этом в Америке и Азии их число было 84 и 63 соответственно, а в Европе – 99.

Главные задачи заключаются в создании и внедрении в строительную отрасль разного рода роботов и манипуляторов, целых строительных комплексов оборудования, в том числе анализ социально-экономических проблем связанных с их использованием.

Ученые многих развитых стран мира, осуществляют работы и исследования, которые связаны с внедрением средств робототехники в строительство разного рода объектов и выводом ее на передовые технические позиции. Эти работы, способствуют формированию строительной робототехники.

Вне зависимости от того, что многие исследования и цели роботизации строительной отрасли являются общими для всей прикладной робототехники, их решения определяются спецификой строительного производства и отличаются индивидуальностью и неповторимостью.

Сегодня исследуется и разрабатывается робототехника, которая будет применяться в строительстве и промышленности, в виде совершенно нового класса машин и автоматов, которые будут обладать огромным количеством механических действий и универсальностью выполняемых операций. Такая многофункциональная техника, у которой искусственный интеллект и которая в процессе выполнения разного рода строительных работ в состоянии воспроизводить, что-то похожее на интеллектуальные или двигательные функции человека, называется роботами. Сегодня, самое широкое применение при автоматизации производственных операций получили, так называемые промышленные роботы.

Обобщая вышеизложенное можно отметить, что научно-технический прогресс в области строительства не стоит на месте. Роботы и автоматизированные машины внедряются в разнообразные строительные процессы.

1.2 Анализ конструкции строительных манипуляторов

Строительный манипулятор представляет собой перепрограммируемый многофункциональный манипулятор, предназначенный для осуществления определенных, заранее заданных перемещений материалов, деталей, инструментов или специальных приспособлений с целью выполнения

различных работ. Важным компонентом СМ является сам манипулятор – устройство для выполнения двигательных функций, аналогичных функциям руки человека при перемещении объектов в пространстве, оснащённое рабочим органом [19].

По своей структуре СМ – многозвенная машина, между отдельными элементами которой существуют механические связи. Большинство существующих в настоящее время СМ используют электрические двигатели для выполнения перемещений. Для управления рабочим органом могут использоваться пневматические и гидравлические приводы.

Все промышленные роботы имеют два физически отдельных базовых элемента - манипулятор и контроллер. Основная архитектура роботов-манипуляторов состоит из цифровых сервоуправляемых электромоторных приводов на кинематических машинах с последовательным подключением, обычно с не более чем шестью осями (степенями свободы). Все поставляются с собственным контроллером.

Практически все приложения роботов требуют значительные усилия инженеров и техников по проектированию и реализации. Уникальность каждого робота заключается в том, как компоненты собираются вместе, чтобы достичь производительности, которая даст конкурентоспособный продукт.

В зависимости от области применения могут использоваться различные схемы построения механической части манипулятора. Следует разделять структуру его «руки» и «запястья».

Основная конструкция «руки» представляет собой последовательность звеньев, соединённых между собой вращательными и поступательными связями. По характеру и количеству связей можно выделить следующие категории организации пространственных перемещений [20, 21]:

- манипуляторы с декартовой (прямоугольной) системой координат (рисунок 2);
- манипуляторы с цилиндрической системой координат (рисунок 3);

– манипуляторы со сферической (полярной) системой координат (рисунок 4);

– манипуляторы с вращательной (шарнирной) системой координат (рисунок 5).

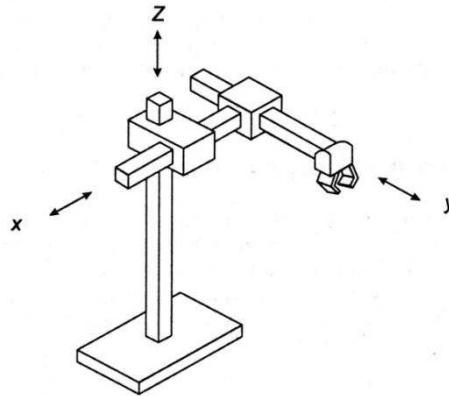


Рисунок 2 – Манипулятор с декартовой (прямоугольной) системой координат

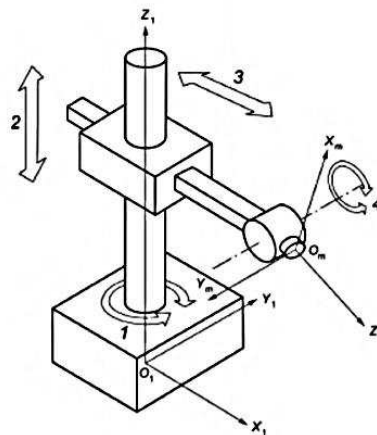


Рисунок 3 – Манипуляторы с цилиндрической системой координат

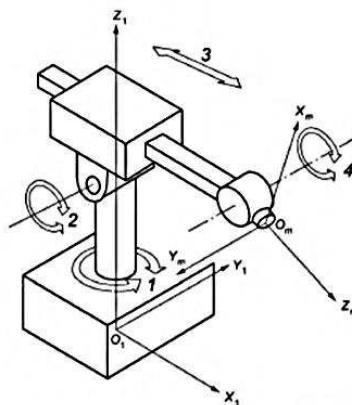


Рисунок 4 – Манипуляторы со сферической (полярной) системой координат

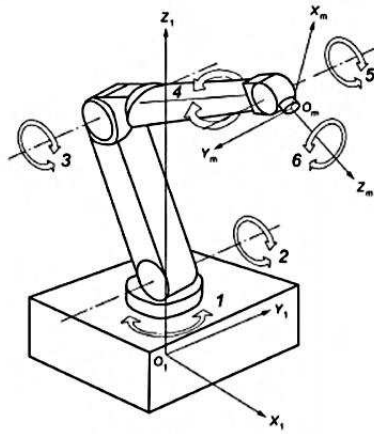


Рисунок 5 – Манипуляторы с вращательной (шарнирной) системой координат

Кроме того, в настоящее время развиваются манипуляторы, использующие параллельные связи (рисунок 6), и SCARA-манипуляторы (Selective Compliant Assembly Robot Arm) (рисунок 7) [22].

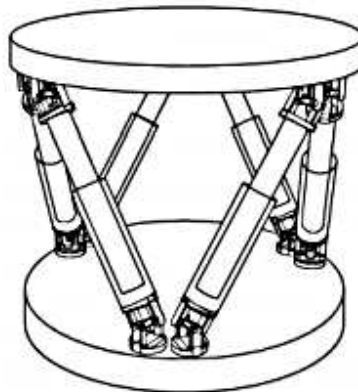


Рисунок 6 – Конструкция манипулятора с параллельными связями

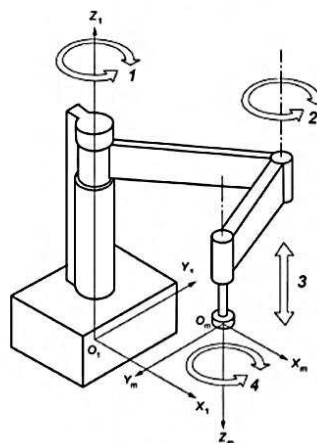
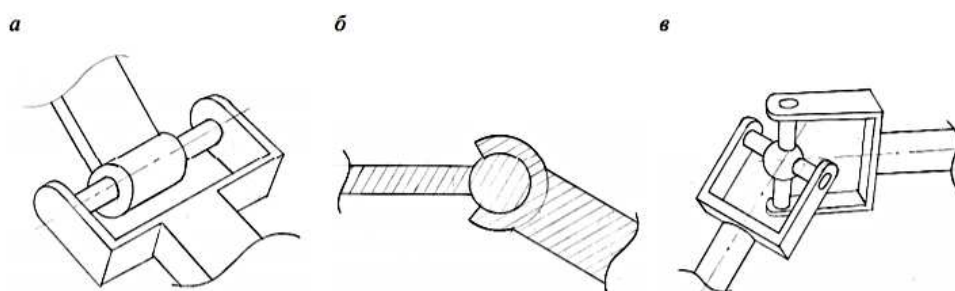


Рисунок 7 – Манипулятор типа SCARA

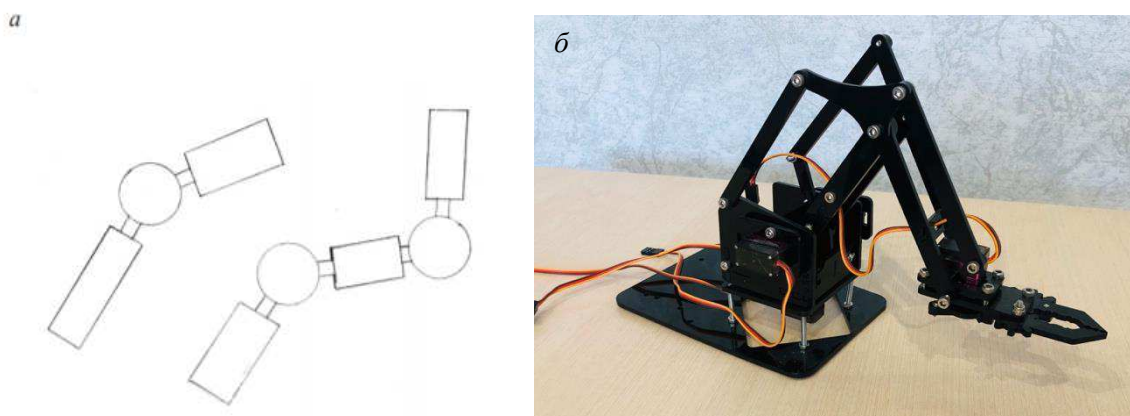
Шарнир также является примером кинематической пары в манипуляторе.

Кинематические пары – это два соединенных между собой звена, допускающие какое-либо относительное перемещение. Кинематические пары образуют кинематические цепи, которые бывают замкнутыми и разомкнутыми (рисунок 8). Замкнутой цепью называют ту, все звенья которой входят в состав как минимум двух кинематических пар. Кинематическая цепь, в которой есть звенья, входящие лишь в одну пару, называется разомкнутой [2].



а) цилиндрический шарнир; б) – сферический шарнир; в) – шарнир угловых осей

Рисунок 8 – Примеры замкнутых кинематических пар



а – пример разомкнутой и замкнутой кинематических пар; б – замкнутые кинематические пары нашего манипулятора

Рисунок 9 – Схемные решения робота

Однако, вне зависимости от вида кинематической цепи, ее замыкает одно условное неподвижное звено. Один его конец неподвижно крепится к основанию робота, а его вторая половина соединена со следующим звеном, от

которого идет еще одно, и так продолжается до самого «рабочего органа» манипулятора – его ключевой части.

Рабочий орган – это исполнительная система манипулятора, предназначенная для выполнения различных действий. Она может представлять собой сверло, схват, фрезу, что угодно, в зависимости от целевого назначения робота. В нашем проекте – это схват, который сжимается, захватывая предмет, поворачиваясь, переносит его на другое место и разжимается (рисунок 10).

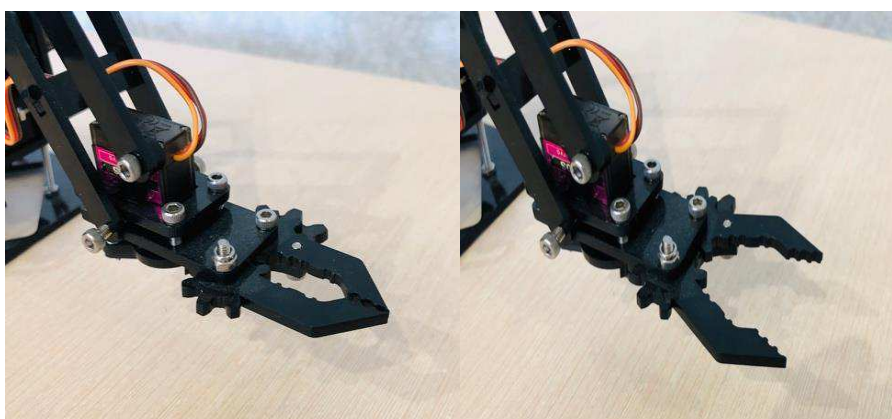


Рисунок 10 – Пример работы манипулятора

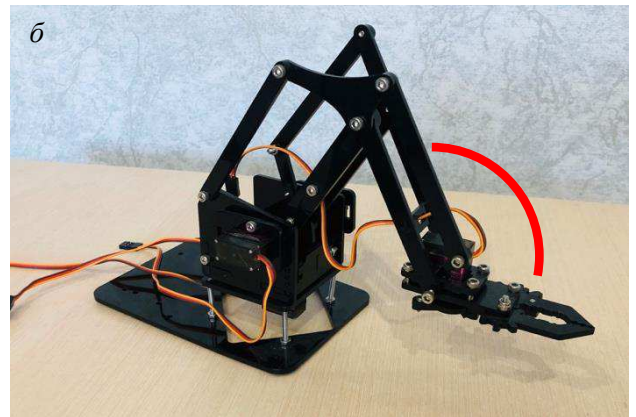
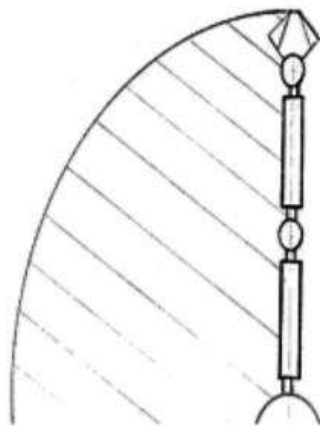
Рабочая зона манипулятора предполагает изменение положения, но каждый робот имеет границы, предел возможностей. Это своеобразный контур, за пределами которого он не сможет работать, так как попросту туда не дотягивается. Пространство, находящееся внутри этого контура называется рабочей зоной манипулятора [3]. На рисунке 11 это заштрихованная область, а линия, ограничивающая ее – траектория движения рабочего органа, когда угол между всеми звеньями равен 180° .

Рабочей зоны не было бы, не будь манипулятор подвижен. Движение роботу обеспечивают специальные приводы, расположенные внутри него или, иногда, на поверхности. Базовый состав привода – двигатель и устройство управления данным приводом. Главное качество частей привода – легкость. В движение исполнительную часть и звенья приводят двигатели. Их может быть

несколько и расположены они в разных местах, так как расположение двигателя определяется рациональностью его нахождения в данной части робота.

Различают пневматические, гидравлические и электрические приводы. Они различаются мощностью и стоимостью, зависящей от сложности рабочего принципа. Большинство производимых в настоящее время манипуляторов относится к числу роботов с вращательной системой координат.

а



а – схема рабочей области; б – схема рабочей области нашего манипулятора
Рисунок 11 – Схема робота

Они обеспечивают наибольший объём рабочей зоны, в которой может осуществляться движение. Их структура позволяет достигать заданного положения и ориентации рабочего органа, в том числе и при наложении ограничений на возможные перемещения, возникающих при наличии препятствий в рабочей зоне или требуемых для обеспечения безопасности эксплуатации.

Выбор конкретной реализации и конструкции МР определяется, прежде всего, областью его непосредственного применения. Основными направлениями их использования в настоящее время являются:

- промышленность;
- сельское хозяйство;
- космическая техника;

– медицина.

Из указанных областей применения наибольшее распространение получила промышленность.

В вышеперечисленных направлениях мы не указали строительство, так как применение МР в нем, составляет очень низкий процент.

Факторами, сдерживающими применение промышленных роботов, являются:

– отсутствие у предприятий не только собственного опыта применения роботов, но даже общего представления о технических и экономических основах роботизированных технологий;

– отсутствие у предприятий квалифицированных кадров, способных обеспечить эксплуатацию роботов;

Робототехника, находящаяся ныне на передовой научно-технического прогресса – это прикладная наука, занимающаяся разработкой автоматизированных технических систем и являющаяся важнейшей основой интенсификации производства. Использование роботов позволяет значительно повысить производительность труда, улучшить качество продукции, оптимизировать процессы управления, отстранить человека от опасных для здоровья производств.

1.3 Основные тенденции развития строительных манипуляторов

Системный и экономический анализ строительных процессов показывает, что строительные манипуляторы должны быть специализированы по виду работ и иметь достаточный уровень многофункциональности для выполнения группы операций определенного вида [4; 5; 21]. При этом к ним предъявляются требования расширенной зоны обслуживания, повышенной надежности работы, подвижности в процессе выполнения операций. Постоянное перемещение строительной площадки предъявляет определенные требования к мобильности и компактности строительных манипуляторов. Кроме того, при

роботизации строительных работ должны учитываться ограниченные рабочие зоны.

Наряду с перечисленными особенностями необходимо также отметить разнообразие операций, выполняемых на одном рабочем месте, недетерминированность окружающей среды и ее высокую вариативность. Температурный диапазон работы манипуляторов, предназначенных для использования на строительных площадках, выбирается от $-40\text{ }^{\circ}\text{C}$ до $+40\text{ }^{\circ}\text{C}$ при влажности до 90 %. В недетерминированной среде при выполнении строительных технологических процессов функционирование автономных манипуляционных роботов даже с высокоорганизованной системой автоматического управления невозможно без участия человека. Поэтому возникает необходимость сочетания дистанционного и автоматического управления манипуляторами и приспособление этого управления к изменяющейся среде.

Большое внимание уделяется совершенствованию кинематической цепи манипуляторов [1]. Целью исследований в этом направлении является улучшение динамических и точностных характеристик.

Совершенствование математического обеспечения и алгоритмов управления играет решающую роль в придании строительным манипуляторам необходимых качеств. Именно соответствующим построением системы управления и рабочих программ можно добиваться устранения влияния погрешностей механизмов манипулятора, высокой точности позиционирования, плавности движений, быстродействия, отсутствия колебаний и перерегулирования при остановках.

Все перечисленное позволяет сформулировать основные тенденции дальнейшего развития и совершенствования строительных манипуляторов:

- повышение грузоподъемности;
- расширение рабочей зоны;
- совершенствование кинематики рабочего оборудования;

- совершенствование математического обеспечения и алгоритмов управления;
- обеспечение плавного движения манипулятора вдоль всей траектории.

2 Теоретические исследования

С точки зрения механики, современный робот представляет собой управляемую систему достаточно сложной кинематической конструкции из множества звеньев, сопряженных посредством линейно или/и вращательно подвижных сочленений и взаимодействующих между собой и с окружающими объектами, приводимую в движение посредством различных типов актюаторов и перемещающиеся по произвольным траекториям под действием внешних, консервативных и диссипативных сил, включая инерционные, кориолисовы и центробежные, гравитационные силы, силы трения, упругой и неупругой деформаций, аэродинамического и гидродинамического сопротивления среды и т.д.

Суть динамического анализа заключается в определении движения манипулятора, находящегося под воздействием динамических и статических нагрузок, действующих в каждой степени подвижности, движение манипулятора всегда осуществляется с какой-нибудь целью, например, привести схват в заданную точку или отработать заданную траекторию в зоне его обслуживания и т.д.

Вопросы, связанные с построением движений и с оценкой динамического поведения манипулятора являются достаточно сложными для исследования, поскольку исполнительный орган состоит не только из одного механизма, но и силовых приводов, действующих совместно с системой управления. Поэтому полное исследование исполнительного органа манипулятора является трудной задачей даже при использовании вычислительной техники. Следовательно, обычно от реальной динамической системы переходят к некоторой ее упрощенной динамической модели, учитывающей главные особенности

исследуемого объекта и пренебрегающей менее существенными факторами. Такая модель, во-первых, позволяет значительно упростить математическое описание динамических свойств системы, а во - вторых дает возможность более наглядно исследовать динамические характеристики системы, ее особенности.

Методы аналитического исследования динамики манипуляторов излагаются в работах [1, 9], но эти методы применяются для упрощенных линейных моделей.

Методика экспериментального исследования динамических характеристик манипуляторов [10] дает возможность оценить динамику и анализировать качества переходных процессов следящего электрогидравлического привода. Методика заключалась, в основном, в нахождении коэффициентов характеристического уравнения, а также аналитическом построении (аппроксимации) переходной функции, что позволяет получить численные значения характеристик данного привода манипулятора, необходимые для анализа его работы.

В ряде случаев для моделирования уравнений движения используется численные методы [11], которая автоматизирует этот процесс. Следует отметить, что в основном моделирование многозвенных органов манипуляторов сопровождается весьма громоздкими преобразованиями сложных аналитических выражений. Общеизвестным подходом при этом является представление всех уравнений и параметров в матричном виде [12, 13], который дает возможность создания общей теории и решения задач управления манипуляторами независимо от числа звеньев. Однако применение такого аппарата требует значительного объема памяти для хранения необходимой информации в оперативном запоминающем устройстве ПЭВМ. В работе [11] разработана методика упрощения математической модели с целью минимизации времени счета на ПЭВМ. Для манипуляторов с небольшим числом звеньев более эффективным является использование непосредственных алгебраических соотношений, связывающих кинематические и динамические параметры.

Численно-аналитические методы исследования динамики манипулятора предлагаются в работах [14] также здесь излагаются системы аналитических вычислений (компьютерной алгебры) к получению и анализу уравнений движения с применением численных алгоритмов и программ.

Вопросам динамического анализа манипуляторов на базе МВК посвящены работы [6, 7, 8]. Решение дифференциальных уравнений движения МВК производится совместно с системой тригонометрических уравнений кинематики.

В работах [7] предложены аналитические методы анализа динамики манипуляторов на базе МВК основанные на уравнениях Даламбера - Лагранжа и Лагранжа 2-го рода, позволяющие учитывать динамику привода простым присоединением его уравнений движения к уравнениям движения манипулятора. Методы также позволяют применять для ее решения численные методы интегрирования, обойти трудности, связанные с проблемой «жесткости» и с не единственностью решения системы тригонометрических уравнений при анализе динамики манипулятора на ПЭВМ.

В последнее время при исследовании динамики манипуляторов используются отображения систем обыкновенных уравнений, в виде математических структурных схем [5, 2, 3] и передаточных функций, что позволяет при моделировании широко использовать методы теории автоматического управления. При этом для проведения исследований были использованы разные методы моделирования, основанные на применении цифровых вычислительных машин [4, 11, 13], аналоговых вычислительных машин [15, 16], а также гибридных вычислительных систем [13], состоящих из аналоговых вычислительных машин, ПЭВМ со средствами сопряжения.

Помимо вышеназванных методов исследования динамики манипуляторов применяется метод имитационного моделирования [4, 17], который позволяет достаточно просто учесть особенности сложных систем при различных воздействиях. Решение задачи моделирования связано с методами численного интегрирования на ПЭВМ, так как динамические модели трудно поддаются

аналитическому исследованию. По сравнению с аналитическими методами метод имитационного моделирования позволяет достаточно просто учесть особенности сложных систем при различных воздействиях.

Для моделирования поведения динамических систем, к которым относится система следящих приводов, используются ПЭВМ. Существует большое количество алгоритмических языков, на которых может быть выполнено решение задачи. Выбор того или иного языка программирования зависит от многих условий. Часто решающую роль оказывает удобство программирования, наличие проверенных математических методов, легкость представления результатов моделирования. Такими особенностями обладает программный пакет MATLAB, содержащий в своем составе инструмент визуального моделирования Simulink [18].

Simulink сочетает в себе наглядность аналоговых вычислительных машин и точность ПЭВМ. Он обеспечивает пользователю доступ ко всем возможностям пакета MATLAB, в том числе к большой библиотеке численных методов.

При моделировании с использованием Simulink реализуется принцип визуального программирования, в соответствии с которым, пользователь на экране из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты.

Наиболее целесообразным методом анализа динамики манипулятора с параметрическими регуляторами является метод математического моделирования, основанный на составлении модели, учитывающий особенности исследуемого объекта и исследования ее с помощью аналоговых вычислительных машин и ПЭВМ (в частности персональных компьютеров (ПК)). Именно этим методом воспользуемся в дальнейшем для проведения работы, направленной на исследование динамики манипуляторов.

Приведенный выше обзор работ показывает, что к существующим современным манипуляторам и системам приводов их ведущих звеньев предъявляется большое количество требований. Но одним из основных

требований является получение заданных динамических характеристик манипулятора. С точки зрения повышения точности позиционирования, грузоподъемности и быстродействия наиболее перспективным представляется применение манипуляторов, созданных на базе замкнутых кинематических цепей. Все вышеизложенные работы, посвященные как манипуляторам на базе замкнутых кинематических цепей, так и манипуляторам на базе незамкнутых кинематических цепей свидетельствуют о преимуществе и эффективности применения первого из них по сравнению со вторым. Но их количественного сравнительного между собой, анализа, как в частности их динамики, до сих пор не было проведено.

Существующие ныне системы следящих приводов манипуляторов не отвечают в полной мере требованиям к их динамике. Наиболее перспективным представляется принцип применения оптимального параметрического регулятора (ПР), обеспечивающего стабильность динамических характеристик манипуляторов при изменении его конфигурации. Систему, в которой наряду с координатным регулированием используется параметрический регулятор, назовем системой с координатно-параметрическим регулированием (КПР).

2.1 Моделирование динамики следящих приводов

В целях устранения недостатков системы следящих приводов была проведена работа, направленная на проектирование более совершенствованной системы следящих приводов. Как известно, для обеспечения требуемых показателей качества систем следящих приводов обычно используют обратные, последовательные и параллельные корректирующие связи. В соответствии с этим, задача синтеза заключается в выборе параметров коэффициентов усиления скорректированного привода. Таким образом, настоящий параграф посвящен совершенствованию описанного в работе [8] принципа параметрического регулирования и исследованию его эффективности за счет

улучшения качества протекания динамических процессов в системе следящих приводов манипулятора, рисунок 12.

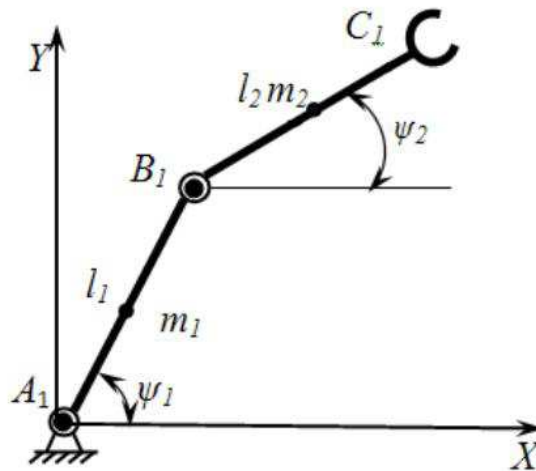
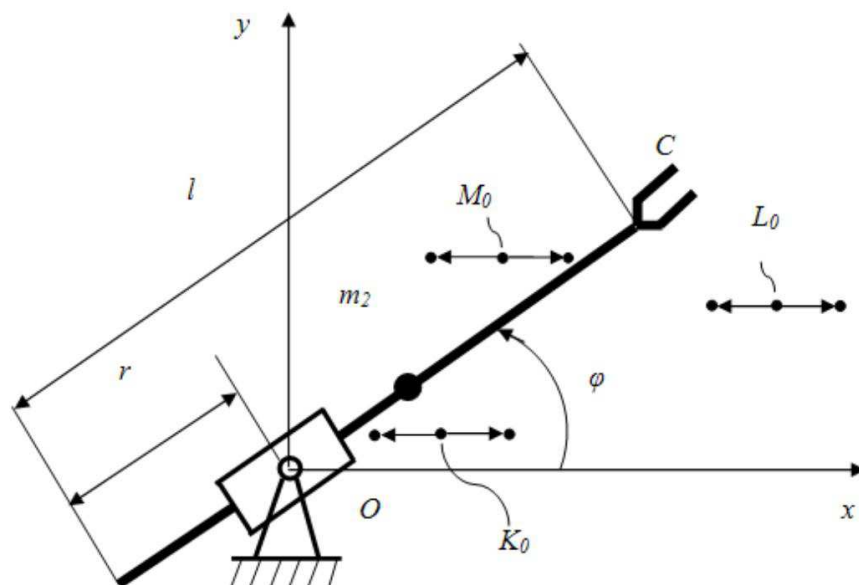


Рисунок 12 – Кинематическая схема манипулятора

Так задачу выбора параметров рассмотрим на примере плоского манипулятора, работающего в полярной системе координат (рисунок 13).



m_2 – масса 2-го звена; l – длина руки; r – линейное перемещение руки; φ – угловое перемещение руки; K_0, L_0, M_0 – характерные точки перемещения схвата

Рисунок 13 – Кинематическая схема манипулятора

Динамическое взаимовлияние следящих приводов для соответствующих степеней подвижности проявляется в том, что приведенный момент инерции

звеньев относительно оси является переменной величиной и зависит от линейного выдвижения r руки:

$$J_{п\varphi} = J_{п\varphi(r)} = J_1 + J_{2c} + m_2 \left(\frac{l}{2} - r \right)^2, \quad (1)$$

где J_1 – момент инерции звена I манипулятора относительно оси вращения ($J_1 = \text{const}$);

J_{2c} – момент инерции звена 2 относительно центральной оси, параллельной оси вращения;

m_2 – масса звена 2;

l – длина руки;

r – линейное перемещение руки.

Тогда в соответствии с выражением приведенный к валу двигателя момент инерции примет вид:

$$J_{пр\varphi} = J_{д\varphi} + J_{р\varphi} = \frac{J_1 + J_{2c} + m_2 \left(\frac{l}{2} - r \right)^2}{z_\varphi^2 \eta_\varphi}, \quad (2)$$

где $J_{д\varphi}$ – момент инерции ротора исполнительного двигателя;

$J_{р\varphi} = (0,05 \div 0,25) \times J_{д\varphi}$ – момент инерции редуктора, приведенный к валу двигателя.

Обычно, при решении технических задач, выбирается та совокупность приемов, которая обеспечивает достижения требуемой цели наиболее простыми средствами. Именно такой подход использован при разработке новой структурной схемы управления манипуляторами. Так как $k_y = k_{ос}$, то можно их комбинировать в один блок, с коэффициентом усиления $k_y = k_\varphi$, на выходе которого получаем:

$$u_{\varepsilon\varphi} = k_y\varphi^0 - k_{oc}\varphi = k_\varphi(\varphi^0 - \varphi). \quad (3)$$

Таким образом, в соответствии с принципом, приведенной в работе [2], и с учетом (2) оптимальное значение переменного коэффициента усиления в прямой цепи следящего привода углового перемещения руки, в зависимости от ее линейного выдвигения r , определяются следующим выражением:

$$k_\varphi^{\text{опт}}(r) = \left[C\sqrt{E^3} + \frac{D}{\sqrt{E}} \right]^2, \quad (4)$$

где

$$E = A + B \left(\frac{l}{2} - r \right)^2, \quad A = J_{д\varphi} + J_{p\varphi} + \frac{J_1 + J_{2c}}{z_\varphi^2 \eta_\varphi}, \quad B = \frac{m_2}{z_\varphi^2 \eta_\varphi},$$

$$C = \frac{2\alpha_1}{c_{e\varphi}\alpha_2 \sqrt{\frac{R_{я\varphi}^3 z_\varphi}{c_{м\varphi}^3 k_{ум\varphi}}}}, \quad D = \frac{c_{e\varphi}}{2} \sqrt{\frac{c_{м\varphi} z_\varphi}{R_{я\varphi} k_{ум\varphi}}}.$$

Так как, приведенный к валу двигателя момент инерции для следящего привода с линейным перемещением r является постоянным, т.е.:

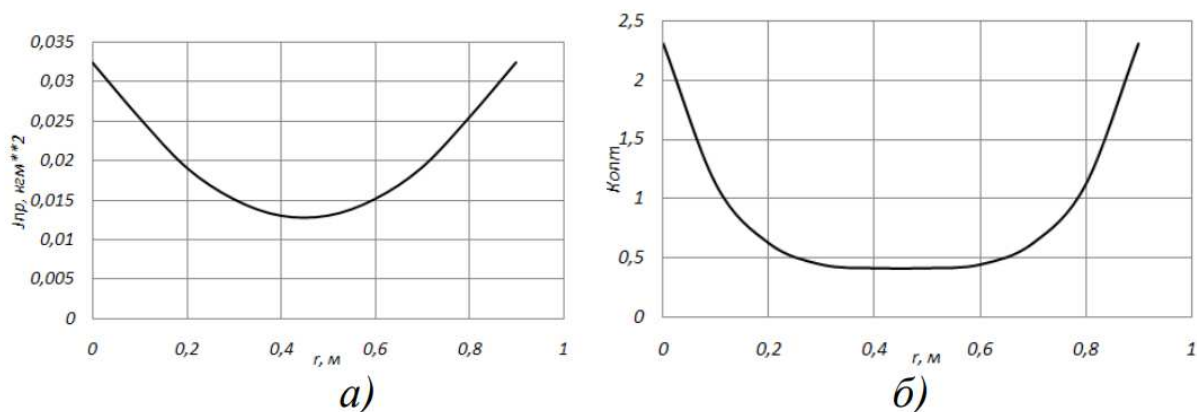
$$J_{прr} = J_{двр} + J_{pr} + \frac{m_2}{z_r^2 \eta_r} \quad (5)$$

то соответствующее значение коэффициента усиления может быть сразу вычислено по формуле

$$k_r^{\text{опт}} = \left[C\sqrt{J_{прr}^3} + \frac{D}{\sqrt{J_{прr}}} \right]^2, \quad (6)$$

Выбор мощности электродвигателя и оптимального передаточного числа редуктора производились согласно стандартным методикам (см. например, [3]).

Для каждого изменения r в интервале $[0; 0,9]$ определяем значения приведенного момента инерции $J_{\text{пр}\varphi}$ по выражению (1) и оптимальный коэффициент усиления по (3) и построим графики зависимостей $J_{\text{пр}\varphi}(r)$ и $k_{\varphi}^{\text{опт}}(r)$ (рисунок 14).



а) $J_{\text{пр}\varphi}(r)$; б) $k_{\varphi}^{\text{опт}}(r)$

Рисунок 14 – Графики зависимостей

Таким образом, были получены оптимальные значения $k_{\varphi}^{\text{опт}}(r)$, которые реализуются с помощью блока ПР, который осуществляет параметрическое регулирование не в цепи обратной связи по положению (с одновременной стабилизацией коэффициента усиления системы), а путем непосредственного изменения коэффициента передачи регулятора следящего привода, что вызывает упрощение системы с ПР и ее технической реализации [4].

Структурная схема системы управления следящего электропривода с ПР приведена на рисунок 15.

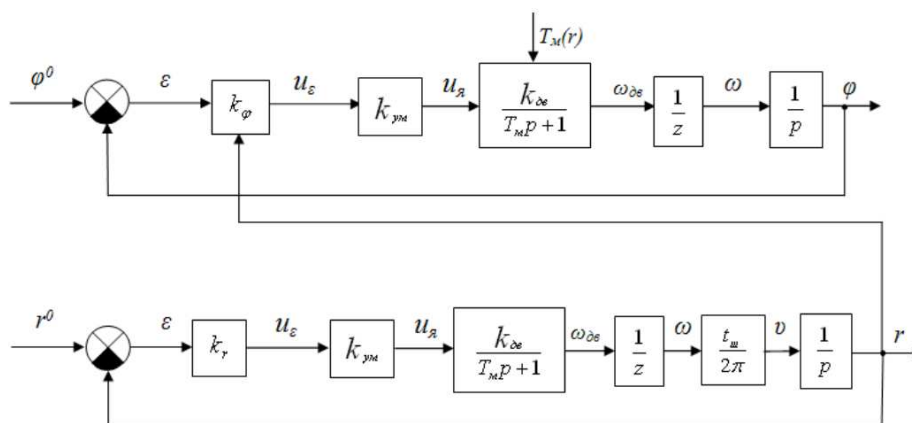


Рисунок 15 – Структурная схема системы управления следящих приводов с параметрическим регулированием

Соответствующие уравнения элементов схемы для следящего электропривода, соответствующей угловой координате φ записываются в следующем виде:

$$\varphi = \frac{\omega_{\text{дв}}^{\varphi}}{z_{\varphi} p},$$

$$\frac{\omega_{\text{дв}}^{\varphi}(p)}{u_{\text{я}}^{\varphi}(p)} = \frac{k_{\text{дв}}}{T_{\text{м}}(r)p + 1}, \quad (7)$$

$$\omega_{\text{дв}}^{\varphi} = \frac{k_{\text{дв}\varphi}}{T_{\text{м}\varphi}p + 1} u_{\text{я}\varphi},$$

$$u_{\text{я}\varphi} = k_{\text{ум}\varphi} u_{\varepsilon\varphi},$$

$$u_{\varepsilon\varphi} = k_{\varphi} \varepsilon_{\varphi},$$

$$\varepsilon_{\varphi} = \varphi^0 - \varphi,$$

где $\omega_{\text{дв}}^{\varphi}$ – угловая скорость вала двигателя;

k_{φ} – коэффициент усиления регулятора;

φ^0 – входное (регулирующее) воздействие, соответствующее требуемому угловому положению манипулятора;

φ – регулируемая величина, соответствующая угловому положению звена манипулятора.

Уравнения, соответствующие элементам схемы по линейной координате r записываются аналогичным образом:

$$\begin{aligned}
r &= \frac{\omega_{ДВ}^r}{z_r p}, \\
\omega_{ДВ}^r &= \frac{k_{ДВr}}{T_{Mr} p + 1} u_{Яr}, \\
u_{Яr} &= k_{умr} u_{\varepsilon r}, \\
u_{\varepsilon r} &= k_r \varepsilon_r, \\
\varepsilon_r &= r^0 - r.
\end{aligned} \tag{8}$$

Общие дифференциальные уравнения системы следящего электропривода могут быть получены исключением промежуточных переменных из систем дифференциальных уравнений (7) и (8):

$$\begin{aligned}
\frac{z_\varphi R_{Я\varphi} J_{пр\varphi}}{c_{м\varphi} k_{ум\varphi} k_\varphi^{опт}} \ddot{\varphi} + \frac{c_e z_\varphi}{k_{ум\varphi} k_\varphi^{опт}} \dot{\varphi} + \varphi &= \varphi^0, \\
\frac{z_r R_{Яr} J_{прr}}{c_{mr} k_{умr} k_r^{опт}} \ddot{r} + \frac{c_e z_r}{k_{умr} k_r^{опт}} \dot{r} + r &= r^0,
\end{aligned} \tag{9}$$

или

$$\begin{aligned}
T_\varphi^2 \ddot{\varphi} + 2T_\varphi \xi_\varphi \dot{\varphi} + \varphi &= \varphi^0, \\
T_r^2 \ddot{r} + 2T_r \xi_r \dot{r} + r &= r^0,
\end{aligned} \tag{10}$$

где

$$\begin{aligned}
T_\varphi &= \sqrt{\frac{z_\varphi R_{Я\varphi} J_{пр\varphi}}{c_{м\varphi} k_{ум\varphi} k_\varphi^{опт}}}, & \xi_\varphi &= \frac{c_e}{2} \sqrt{\frac{c_{м\varphi} z_\varphi}{R_{Я\varphi} J_{пр\varphi} k_\varphi^{опт} k_{ум\varphi}}}, \\
T_r &= \sqrt{\frac{z_r R_{Яr} J_{прr}}{c_{mr} k_{умr} k_r^{опт}}}, & \xi_r &= \frac{c_e}{2} \sqrt{\frac{c_{mr} z_r}{R_{Яr} J_{прr} k_r^{опт} k_{умr}}}.
\end{aligned} \tag{11}$$

По формулам (11) и полученным значениям $k_\varphi^{опт}(r)$ по (4) для каждого положения r в интервале $[0; 0,9]$ определим $T_\varphi(r)$ и $\xi_\varphi(r)$. По результатам расчета построены графики (рисунки 16 и 17, где кривая a – соответствует к КР, а кривая b – к ПР). Из этих графиков видно, что полученные значения

коэффициента усиления обеспечивают оптимальный динамический режим при управлении манипулятором.

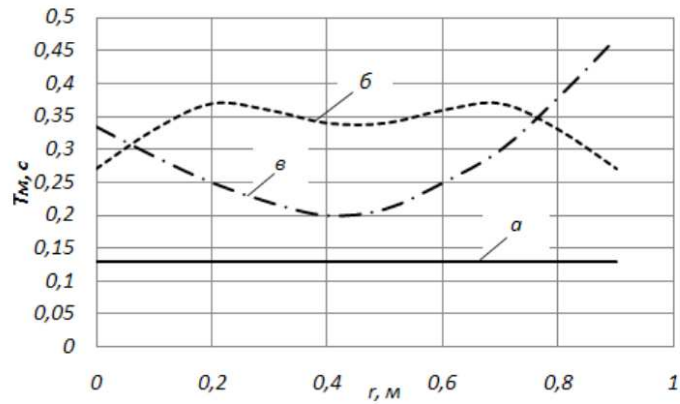


Рисунок 16 – График зависимости $T_{\varphi}(r)$

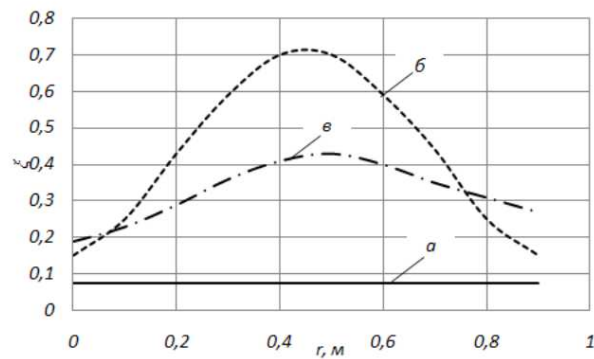


Рисунок 17 – График зависимости $\xi_{\varphi}(r)$

Наиболее просто реализуется в системах автоматического регулирования при его линейном изменении в зависимости от значения r . Произведем линеаризацию выражения (4) по методу наименьших квадратов. Используя этот метод, получена следующая линеаризованная функция [4]:

$$k_{\varphi}^{\text{л}}(r) = ar + b \tag{12}$$

где коэффициенты линеаризации a и b определяются для различных j -х линейных положений r_j руки по следующим формулам

$$a = \frac{\sum_j^N r_j \sum_j^N k_{\varphi j}^{\text{опт}} - N \sum_j^N r_j k_{\varphi j}^{\text{опт}}}{[\sum_j^N r_j]^2 - N \sum_j^N r_j^2},$$

$$b = \frac{1}{N} [\sum_j^N k_{\varphi j}^{\text{опт}} - a \sum_j^N r_j], \quad (13)$$

где $k_{\varphi j}^{\text{опт}}(r)$ – значение коэффициента усиления для значения r_j ;
 N – число различных значений r_j .

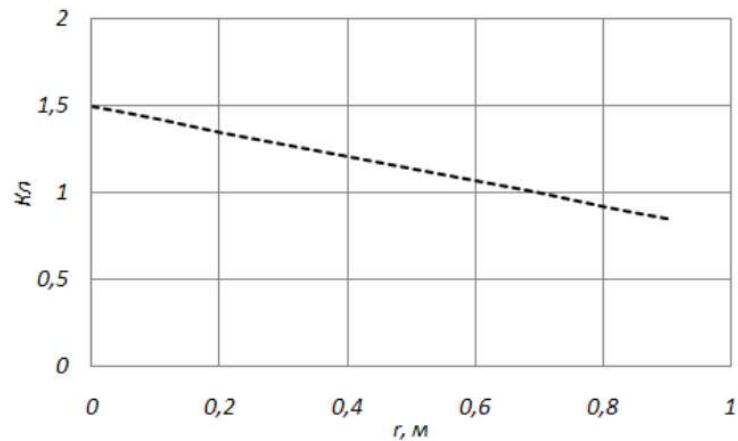


Рисунок 18 – График зависимости $k_{\varphi}^l(r)$

При исследовании динамики следящего привода манипулятора в динамическом режиме в его зоне обслуживания требовалось решить задачу позиционирования схвата из одной заданной точки в другую, где начальное и конечное положения схвата определялись из уравнения кинематики.

$$x_c = (l - r) \cos \varphi, \quad (14)$$

$$y_c = (l - r) \sin \varphi.$$

Путем решения обратной задачи кинематики определялись соответствующие им значения координат углового и линейного положения.

$$r = l - \sqrt{x_c^2 + y_c^2}$$

(15)

$$\varphi = \operatorname{arctg} \frac{x_c}{y_c}.$$

Построение переходных процессов заключалась в численном интегрировании. Но прежде всего систему (10) приведем к следующей форме:

$$\begin{aligned} \dot{\varphi} &= q_1, & \dot{r} &= q_2, \\ T_\varphi^2 \ddot{\varphi} + 2T_\varphi \xi_\varphi \dot{\varphi} + \varphi &= \varphi^0, \\ T_r^2 \ddot{r} + 2T_r \xi_r \dot{r} + r &= r^0 \\ \dot{q}_1 &= \frac{2\xi_r}{T_r} q_1 - \frac{r}{T_r^2} - \frac{k_r r^0}{T_r^2} \\ \dot{q}_1 &= \frac{2\xi_\varphi}{T_\varphi} q_1 - \frac{\varphi}{T_\varphi^2} - \frac{k_\varphi \varphi^0}{T_\varphi^2}. \end{aligned} \tag{16}$$

Используя метод Рунге-Кутты 4-го порядка, проинтегрируем систему, решением которой являются значения угловых и линейных координат φ и r . Когда эти обобщенные координаты примут установившиеся значения конечной заданной величины, процесс интегрирования остановится.

В процессе интегрирования системы (10) по текущим значениям обобщенных координат φ и r путем решения (14) определяются соответствующие им значения координат схвата манипулятора (x_c , y_c). Для оценки влияния блока ПР усовершенствованного следящего привода на качество протекания динамических процессов при выполнении манипулятором операции позиционирования рассматривалась система следящих приводов с ПР в сравнении с традиционно используемой системой с координатным регулированием КР.

В целом же анализ расчета показывает, что быстродействие системы с ПР увеличивается примерно на 20-30% по сравнению с системой с КР. Выполнены аналогичные процедуры для случаев с КПП и с КР систем следящих приводов с учетом линеаризованного коэффициента усиления $k_\varphi^l(r)$.

Моделирование динамики системы следящих приводов с ПР манипуляторов на базе незамкнутых кинематических цепей показало, что по сравнению с динамикой обычной системы координат регулируемого привода, здесь происходит улучшение характера протекания переходных процессов. Сравнительный анализ динамики систем следящих приводов манипуляторов на базе механизмов с замкнутыми и незамкнутыми кинематическими цепями с ПР и КР позволяет сделать вывод о том, что первый из них имеет преимущество по показателям качества переходных процессов как при ПР, так и при КР по сравнению со вторым при ПР. Это объясняется тем, что момент инерции манипулятора I имеет малую величину по сравнению с величиной момента инерции манипулятора II.

2.2 Проектирование модели системы управления следящим приводом строительного робота

Целью проектирования является определение переходных характеристик системы следящих приводов манипулятора при номинальных значениях параметров ее элементов, а также определение влияния изменения положения объекта регулирования 2-го канала управления (линейного перемещения механизма манипулятора) на качество регулирования выходной координаты 1-го канала управления (углового положения механизма манипулятора). При этом в качестве критериев оценки влияния 2-го канала управления на свойства 1-го канала управления можно принять время переходного процесса и величину перерегулирования.

При исследовании качества процесса управления математическое описание системы следящих приводов манипулятора будем осуществлять в форме структурных схем систем автоматического управления (САУ). Эти схемы наглядно и полно отражают основные части системы, их взаимосвязь, динамические свойства, а также прохождение сигналов управления. При этом, как известно, наибольшее распространение для описания САУ получили схемы,

представляющие систему в виде соединения звеньев направленного действия, характеристики «вход-выход» которых заданы передаточными функциями САУ.

В математических моделях систем следящих приводов манипулятора для учета динамического взаимовлияния приводов достаточно рассматривать упрощенные передаточные функции исполнительных двигателей приводов, которые представляют собой аperiodические звенья САУ [6,7], т.к. в качестве исполнительных двигателей используются двигатели постоянного тока независимого возбуждения (ДПТ НВ).

Передаточная функция ДПТ НВ следящего привода углового перемещения руки манипулятора с переменным моментом инерции нагрузки представляет собой передаточную функцию аperiodического звена с изменяющейся постоянной времени:

$$\frac{\omega_{дв}(p)}{u_{я}(p)} = \frac{k_{дв}}{T_{м}(r)p+1}. \quad (17)$$

где $u_{я}(p)$, $\omega_{дв}(p)$ – соответственно операторные выражения напряжений на якорной обмотке двигателя и скорости вала двигателя;

$T_{м}(r)$ – электромеханическая постоянная времени ДПТ НВ следящего привода углового положения руки манипулятора.

$$T_{м}(r) = \frac{(J_{дв} + J_{п} + \frac{J_{л}(p)}{z^2 \eta}) R_{я}}{c_e c_{м}} \quad (18)$$

С учетом выражения (18) получим в общем виде выражение, характеризующее динамическое влияние следящего привода линейного перемещения на следящий привод углового перемещения манипулятора:

$$T_{м}(r) = \alpha r^2 + \beta r + \gamma, \quad (19)$$

где $\alpha = \frac{m_2 R_{Я}}{z^2 \eta c_e c_M}$, $\beta = \frac{2 l m_2 R_{Я}}{z^2 \eta c_e c_M}$, $\gamma = \frac{(J_{ДВ} + J_p + J_1 + J_{2c}) c_e c_M}{z^2 \eta}$.

Оно характеризует динамическое влияние следящего привода линейного перемещения на следящий привод углового перемещения руки манипулятора.

Структурная схема системы следящего привода манипулятора, с учетом динамического влияния следящего привода линейного перемещения на следящий привод углового перемещения руки манипулятора представлена на рисунке 19.

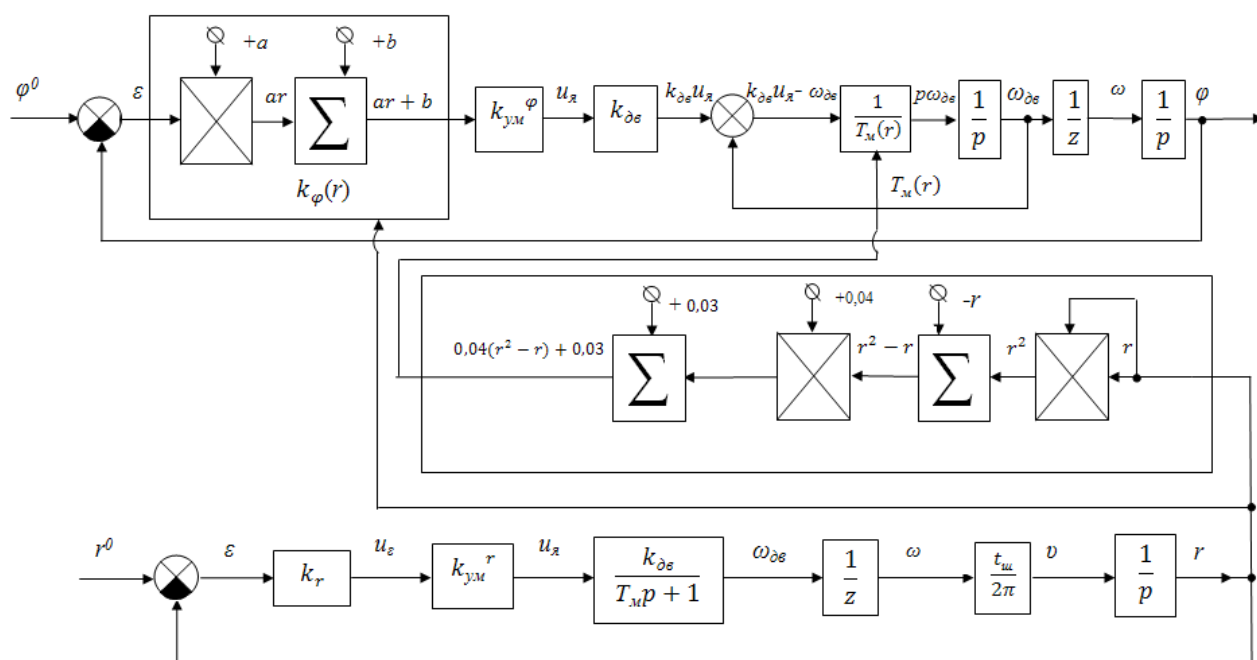


Рисунок 19 – Структурная схема системы следящих приводов манипулятора с ПР

На схеме учитывается параметрическое регулирование значения коэффициентов усиления главной обратной связи следящего привода углового перемещения руки манипулятора. Усилительно-преобразовательные элементы приводов здесь описывались как безынерционные звенья, исполнительные двигатели (ДПТ НВ) – как апериодические звенья САУ. Механические передачи считаются абсолютно жесткими.

В этой структурной схеме введены следующие обозначения:

φ^0 – входное (регулирующее) воздействие, соответствующее требуемому угловому положению манипулятора;

φ – регулируемая величина, соответствующая угловому положению манипулятора,

r^0 – входное воздействие, соответствующее требуемому линейному положению руки манипулятора,

r – регулируемая величина, соответствующая линейному положению руки манипулятора;

v – скорость линейного перемещения руки манипулятора.

На рисунке 19 в части структурной схемы следящего привода углового перемещения с учетом динамического влияния на него следящего привода линейного перемещения руки манипулятора $T_m(r)$ при параметрическом регулировании коэффициента усиления регулятора $k_{\text{ропт}}(r)$, не раскрываются полностью внутренние связи динамических звеньев с изменяющимися параметрами, например, апериодическое звено с изменяющейся постоянной времени. Это затрудняет составление схемы моделирования для этой части структурной схемы системы следящих приводов манипулятора способом структурного моделирования. Поэтому по структурной схеме (рисунок 19) системы следящих приводов манипулятора для следящего привода углового положения руки манипулятора составим детализированную структурную схему, т. е. схему, состоящую только из интегрирующих, дифференцирующих и безынерционных (в том числе нелинейных) звеньев с полностью вскрытыми связями между ними. Для составления детализированной структурной схемы апериодического звена с изменяющейся постоянной времени перейдем от ее передаточной функции к дифференциальному уравнению, разрешенному относительно старшей производной, с выделением (вынесением за скобку) изменяющейся постоянной времени $T_m(r)$. Используя оператор дифференцирования $p=d/dt$, получим следующее выражение:

$$p\omega_{дв}(p) = \frac{1}{T_m(r)}(k_{дв}u_{я}(p) - \omega_{дв}(p)). \quad (20)$$

Структурная схема на рисунке 20, составленная по полученному выражению (20), является детализированной структурной схемой и позволяет получить схемы моделирования аperiodического звена с изменяющейся постоянной времени способом структурного моделирования.

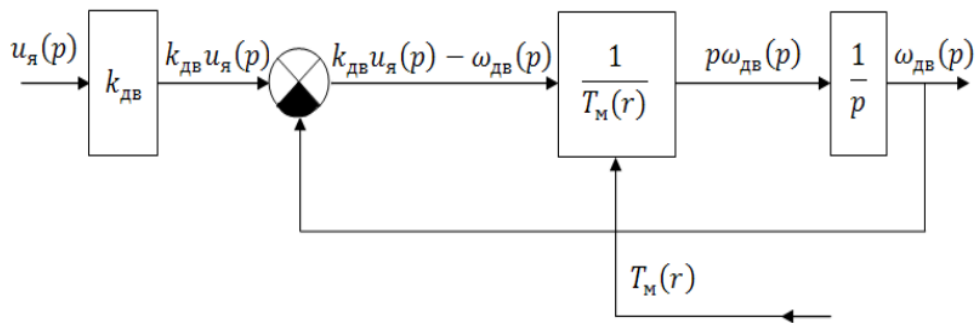


Рисунок 20 – Детализированная структурная схема

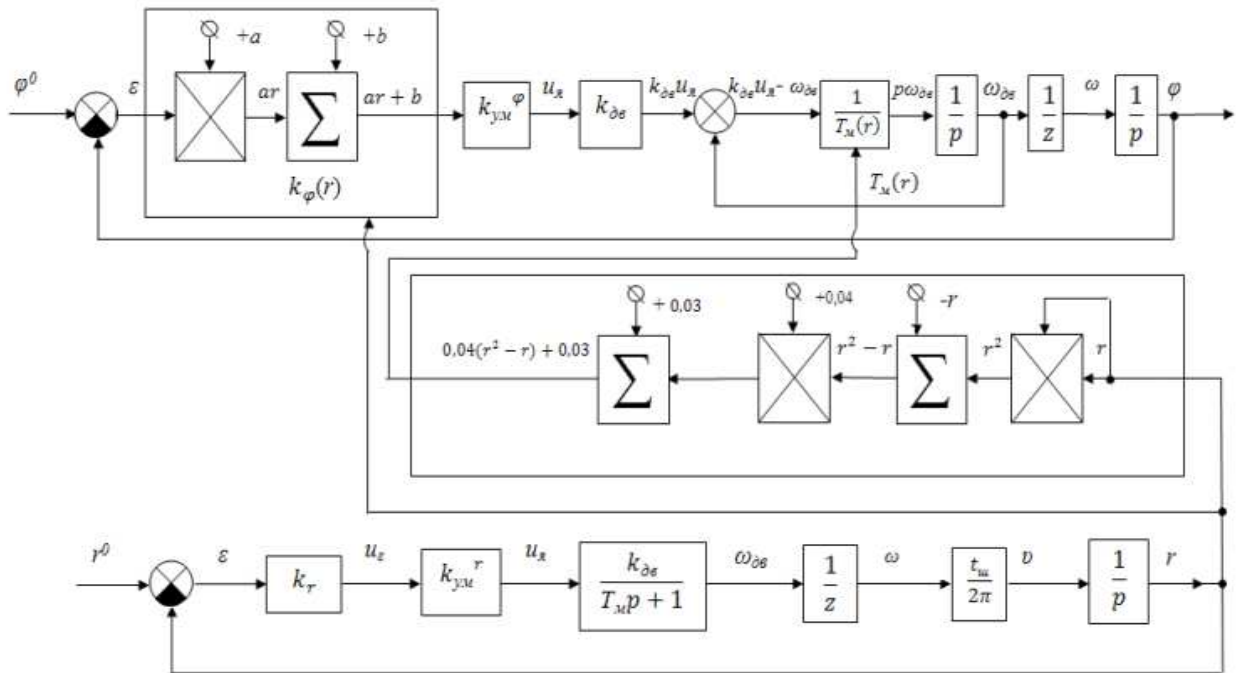


Рисунок 21 – Детализированная структурная схема системы следящих приводов манипулятора с ПР

Вычисленные основные параметры структурной схемы исследуемой системы следящих приводов манипулятора при принятых значениях, являются следующими:

$$k_r = 1 \left[\frac{\text{В}}{\text{рад}} \right], \quad k_\varphi(r) = ar + b, \text{ где } a = 0,205 \left[\frac{\text{В} \times \text{рад}}{\text{м}} \right],$$

$$b = 0,44 \left[\frac{\text{В}}{\text{рад}} \right], \quad k_{\text{ум}}^r = k_{\text{ум}}^\varphi = 10, \quad k_{\text{дв}}^r = 2,209 \left[\frac{\text{рад} \times \text{с}}{\text{В}} \right],$$

$$k_{\text{дв}}^\varphi = 1,09 \left[\frac{\text{рад} \times \text{с}}{\text{В}} \right], \quad z^\varphi = 240, \quad z^r = 60, \quad T_M = 0,03 \text{ [с]},$$

$$\alpha = 0,0412 \left[\frac{\text{с}}{\text{м}^2} \right], \quad \beta = -0,0412 \left[\frac{\text{с}}{\text{м}} \right], \quad \gamma = 0,0311 \text{ [с]},$$

где $k_{\text{ум}}$ – коэффициент усиления усилителя мощности;

$k_{\text{дв}}$ – коэффициент усиления двигателя;

k_φ – коэффициента усиления;

a и b – коэффициенты линеаризации;

$\varphi(t)$ – переходной процесс углового перемещения;

$r(t)$ – переходной процесс линейного перемещения;

$T_M(r)$ – электромеханическая постоянная времени ДПТ НВ следящего привода углового положения руки манипулятора.

Исследование качества процесса управления манипулятором произведем с помощью имитационного моделирования системы следящих приводов в среде MATLAB (Simulink), который сочетает в себе наглядность аналоговой вычислительной машины и точность ПК.

Процесс построения модели Simulink включает в себя компоновку модели и задание необходимых параметров. Компоновка заключается в выборе из библиотек Simulink необходимых блоков, их размещение в открывшемся окне и соединение между собой. Далее для каждого блока устанавливаются соответствующие параметры, отвечающие требованиям моделируемой системы.

Схему моделирования в Simulink составим согласно структурной схеме системы следящих приводов исследуемого манипулятора. При исследовании рассматривалась модель систем следящих приводов манипулятора с ПР в сравнении с моделью традиционной системы координатного регулирования (КР) для манипулятора с взаимовлиянием степеней подвижности. Структурная схема, набранная в системе MATLAB (Simulink), представлена на рисунке 22.

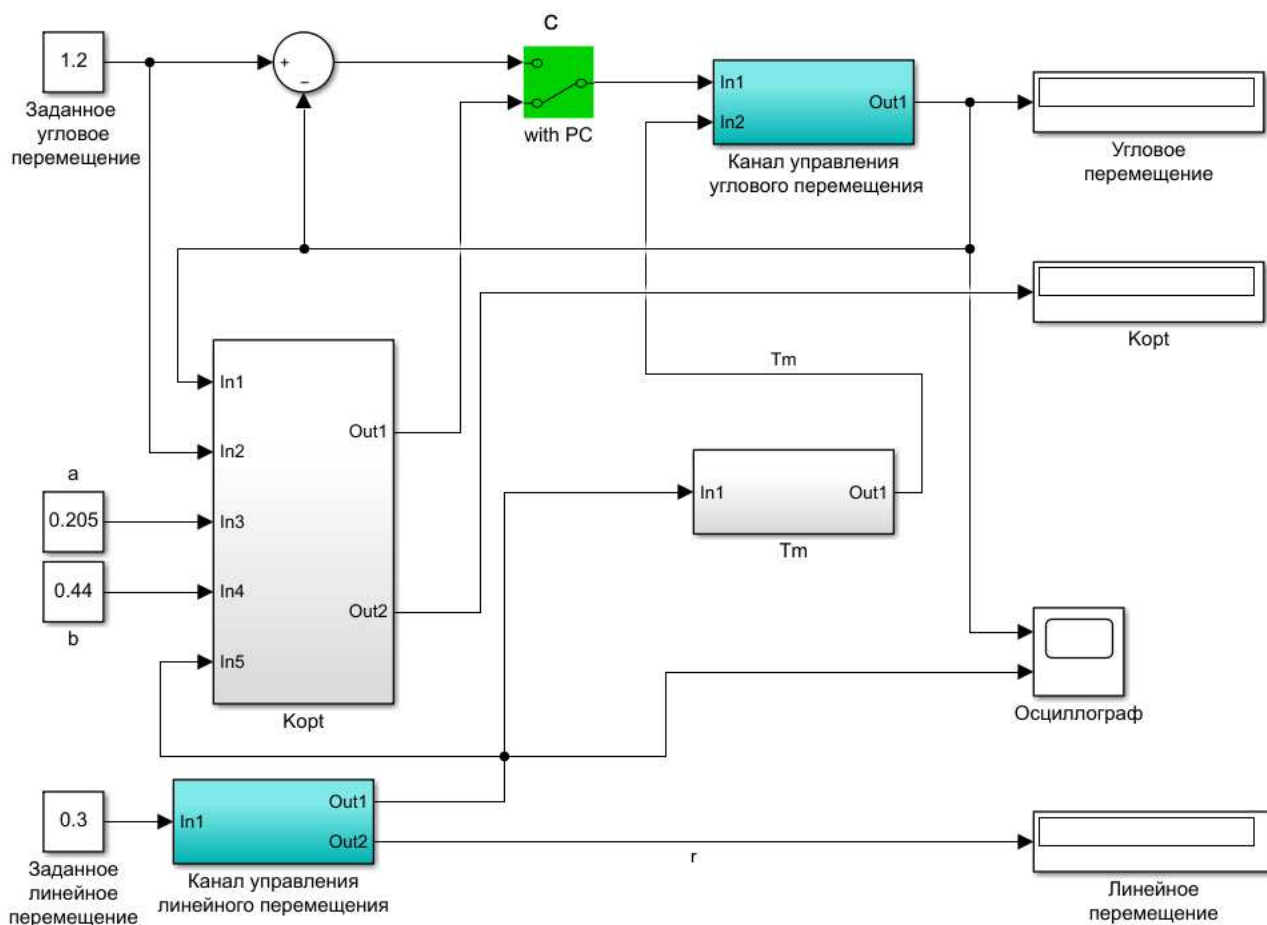


Рисунок 22 – Имитационная модель системы следящих приводов манипулятора

Для повышения наглядности модели сложные функциональные зависимости были объединены в отдельные подсистемы (Subsystems):

- подсистема канала управления по угловому перемещению манипулятора (рисунок 23);
- подсистема канала управления по линейному перемещению манипулятора (рисунок 24);

– подсистема, характеризующая динамическое влияние следящего привода линейного перемещения на следящий привод углового перемещения (рисунок 25);

– подсистема параметрического регулирования значения коэффициента усиления главной обратной связи следящего привода углового перемещения манипулятора (рисунок 26).

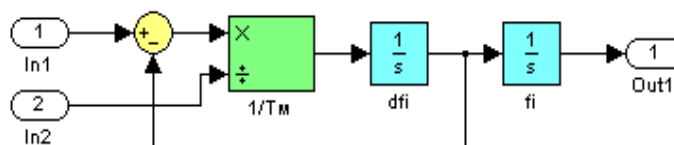


Рисунок 23 – Подсистема канала управления по угловому перемещению манипулятора системы следящих приводов

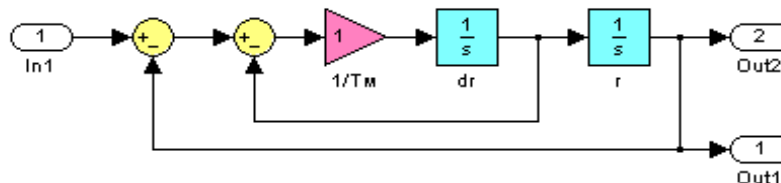


Рисунок 24 – Подсистема канала управления по линейному перемещению манипулятора системы следящих приводов

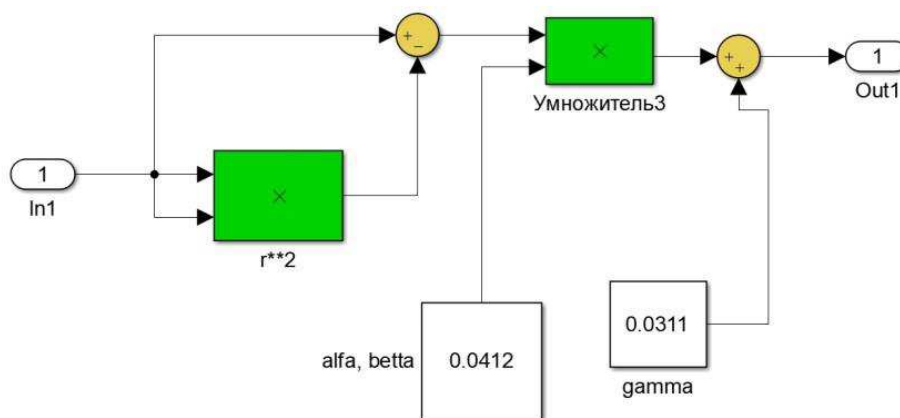


Рисунок 25 – Подсистема, характеризующая динамическое влияние следящего привода линейного перемещения на следящий привод углового перемещения манипулятора

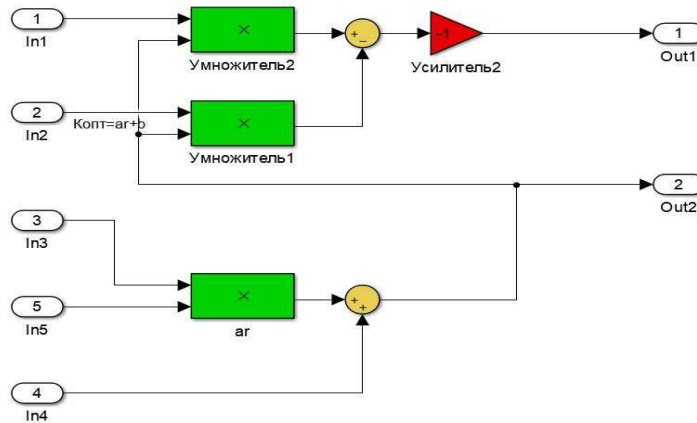


Рисунок 26 – Подсистема параметрического регулирования значения коэффициента усиления главной обратной связи следящего привода углового перемещения манипулятора

При исследовании рассматривалась модель системы следящих приводов манипулятора с ПР (рисунок 22, положение ПР) с взаимовлиянием степеней подвижности.

На выходе системы управления имеются два сигнала:

- переходной процесс углового перемещения $\varphi(t)$;
- переходной процесс линейного перемещения $r(t)$;

Для отображения графиков выходных сигналов в модели использован виртуальный осциллограф (Scope). Получены осциллограммы переходных процессов (рисунок 27, рисунок 28) при отработке системами следящих приводов с ПР и с ПР в окрестностях характерных точек рабочей плоскости манипулятора.

В качестве критериев оценки влияния ПР на динамические свойства системы принимались величина перерегулирования σ и время переходного процесса t_n .

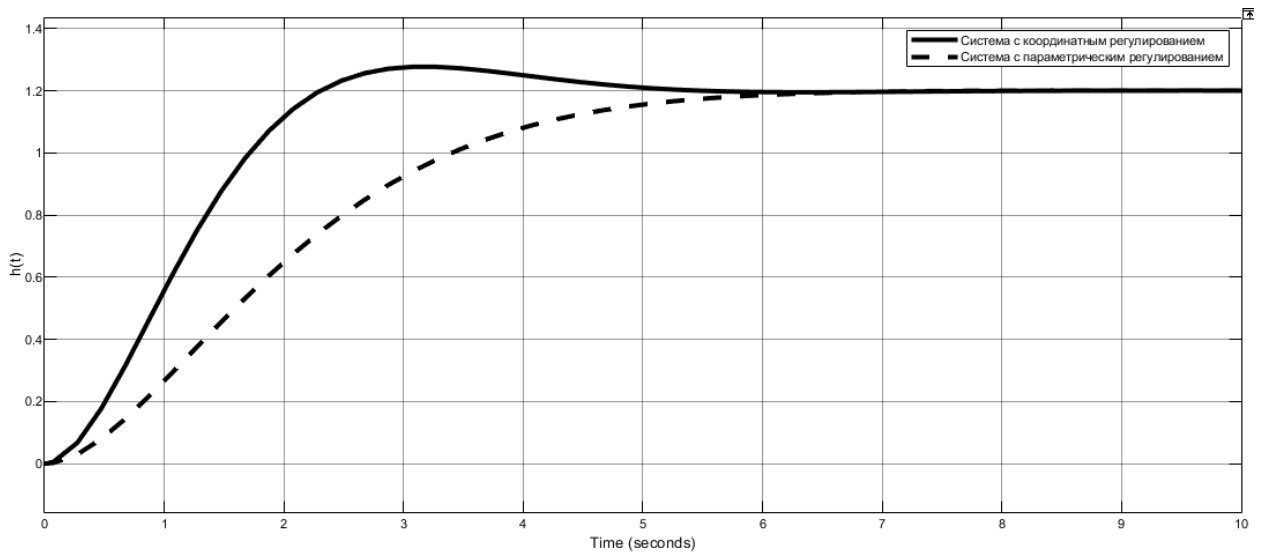


Рисунок 27 – Осциллограммы переходных процессов углового перемещения

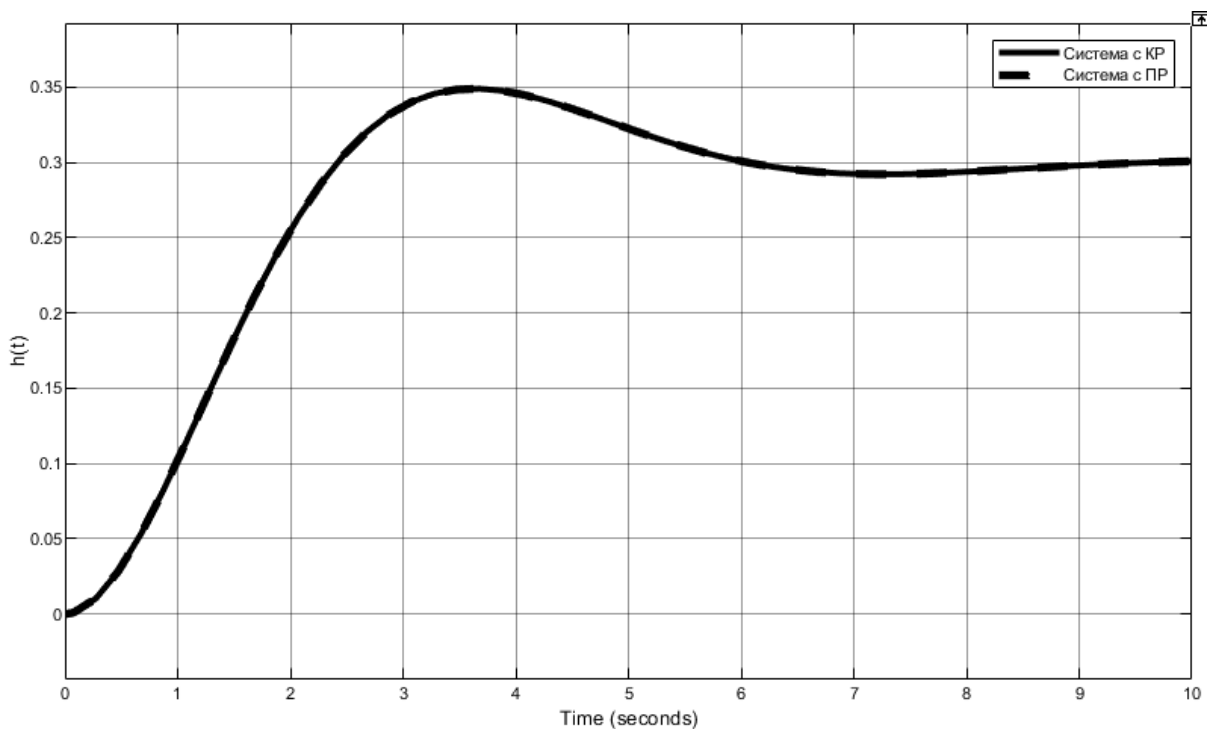


Рисунок 28 – Осциллограммы переходных процессов линейного перемещения

Как видно из осциллограмм, переходные процессы в следящем приводе углового перемещения φ с ПР без перерегулирования по сравнению со следящим приводом с КР (перерегулирование 6,4%). Характеристики переходных процессов в следящем приводе линейного перемещения r руки манипулятора в обоих случаях аналогичны.

2.3 Математическое описание модели двигателя постоянного тока

В системах привода роботов имеют большое распространение двигатели постоянного тока с независимым возбуждением или магнитоэлектрические. Управление этими двигателями, осуществляется по цепи якоря. Электромагнитные и электромеханические процессы в двигателе постоянного тока с независимым возбуждением могут быть описаны следующими уравнениями:

$$\begin{aligned}u_{\text{я}} &= R_{\text{я}} \left(T_{\text{я}} \frac{di_{\text{я}}}{dt} + i_{\text{я}} \right) + e_{\text{я}}, \\ J \frac{d\omega_m}{dt} &= M - M_{\text{H}}, \\ \omega_m &= \frac{d\theta_m}{dt}, \\ e_{\text{я}} &= k_E \omega_m, \\ M &= k_M i_{\text{я}}.\end{aligned}\tag{21}$$

где $u_{\text{я}}$, $i_{\text{я}}$, $e_{\text{я}}$ – напряжение, ток и противо-ЭДС якоря;

$L_{\text{я}}$, $R_{\text{я}}$, $T_{\text{я}} = \frac{L_{\text{я}}}{R_{\text{я}}}$ – индуктивность, сопротивление и электромагнитная постоянная времени якоря;

ω_m , M , M_{H} , θ_m – механическая угловая скорость, электромагнитный момент, момент нагрузки и механический угол поворота вала;

J – момент инерции ротора;

коэффициенты k_E , k_M являются конструктивными постоянными.

Параметры двигателя, входящие в уравнения (21) рассчитываются на основе паспортных данных, помещенных в справочных материалах. В таблице 1 приведены параметры малоинерционных двигателей постоянного тока с электромагнитным возбуждением, разработанные для робототехнических комплексов.

Таблица 1 – Параметры малоинерционных двигателей постоянного тока с электромагнитным возбуждением.

$P_H, \text{кВт}$	$U_{\text{я}} = U_B, \text{В}$	$n_H, 1/\text{мин}$	$I_{\text{я}}, \text{А}$	$R_{\text{я}}, \text{Ом}$	$R_B, \text{Ом}$	$J, \text{кгм}^2$
0,12	110	3000	1.53	1.48	642	0.06
0.20	110	3000	2.46	0.762	560	0.08
0.25	110	3000	3.05	0.945	827	0.14
0.37	110	3000	4.40	0.546	934	0.16
0.45	110	3000	5.60	0.585	400	0.36
1.00	110	3000	9.96	0.561	500	0.80

В таблице 1 использованы следующие обозначения:

P_H – номинальная мощность двигателя, *кВт*;

$U_{\text{я}}$ – номинальное напряжение питания якоря, *В*;

U_B – номинальное напряжение возбуждения, *В*;

$I_{\text{я}}$ – номинальный ток якоря, *А*;

n_H – номинальная скорость вращения якоря, *1/мин*;

$R_{\text{я}}$ – сопротивление обмотки якоря, *Ом*;

R_B – сопротивление обмотки возбуждения, *Ом*.

Номинальный момент двигателя в (*Н м*), номинальный ток возбуждения в (*А*), конструктивные постоянные и индуктивность якоря рассчитываются из уравнений

$$\begin{aligned}
 M_H &= \frac{P_H}{\omega_H} = \frac{30P_H}{\pi n_H}, \\
 I_B &= \frac{U_B}{R_B}, \\
 k_M &= \frac{M_H}{I_{\text{я}}}, \\
 k_E &= \frac{30(U_{\text{я}} - R_{\text{я}}I_{\text{я}})}{\pi n_H}, \\
 L_{\text{я}} &\approx \frac{30}{\pi} \times \frac{U_{\text{я}} c_X}{n_H I_{\text{я}}}.
 \end{aligned} \tag{22}$$

где $c_x = 0,3-0,4$ – эмпирический коэффициент.

Передаточные функции скоростной части ДПТ по управляющему и возмущающему воздействиям могут быть представлены соответственно выражениями:

$$W(s) = \frac{\omega_m(s)}{u_{\text{я}}(s)} = \frac{1/k_E}{\frac{T_{\text{Я}}}{K_1 K_2 k_E} s^2 + \frac{1}{K_1 K_2 k_E} s + 1}, \quad (23)$$

$$W^*(s) = \frac{\omega_m(s)}{M_H(s)} = \frac{\frac{(T_{\text{Я}} s + 1) k_M}{k_E}}{\frac{T_{\text{Я}}}{K_1 K_2 k_E} s^2 + \frac{1}{K_1 K_2 k_E} s + 1}. \quad (24)$$

В выражениях (23), (24) введены обозначения:

$$K_1 = \frac{1}{R_{\text{я}}}, \quad K_2 = \frac{k_M}{J}. \quad (25)$$

Корни характеристического уравнения определяются выражением:

$$s_{1,2} = -\frac{1}{2T_{\text{Я}}} \pm \frac{1}{2T_{\text{Я}}} \sqrt{1 - 4T_{\text{Я}} K_1 K_2 k_E}. \quad (26)$$

В зависимости от соотношения параметров $T_{\text{Я}} K_1 K_2$ и k_E корни характеристического уравнения могут быть либо вещественными, либо комплексно-сопряженными. При выполнении условия $4T_{\text{Я}} K_1 K_2 k_E \leq 1$ корни характеристического уравнения являются вещественными, эквивалентная передаточная функция ДПТ по управляющему воздействию может быть представлена в виде двух последовательно включенных апериодических звеньев первого порядка с постоянными T_1, T_2 и коэффициентом передачи $1/k_E$.

$$W(s) = \frac{x(s)}{u_{\text{я}}(s)} = \frac{1/k_E}{(T_1 s + 1)(T_2 s + 1)}, \quad (27)$$

где $T_1 = -1/s_1, T_2 = -1/s_2,$

В качестве исходной информации рассмотрим двигатель мощностью 0,45 кВт (таблица 1) параметры структурной схемы, рассчитанные по выражениям (8), представлены в таблице 2.

Таблица 2 – Параметры системы электрического привода робота

$K_1, 1/Ом$	$K_2, 1/Амс^2$	$k_M, Нм/А$	$k_E, Вс$	$T_я, с$	$T_1, с$	$T_2, с$
1.72	0.72	0.26	0.34	0.043	0.044	2.15

В соответствии с уравнением (7) и применением программы Simulink получена имитационная модель электрического двигателя, рисунок 29.

Для получения динамических характеристик использован пакет расширения программы Simulink – Control System с его графическим интерфейсом. Эталонным (задающим) входным сигналом является напряжение 1 В. Динамические характеристики по управлению и по возмущению показаны на рисунках 30 и 31.

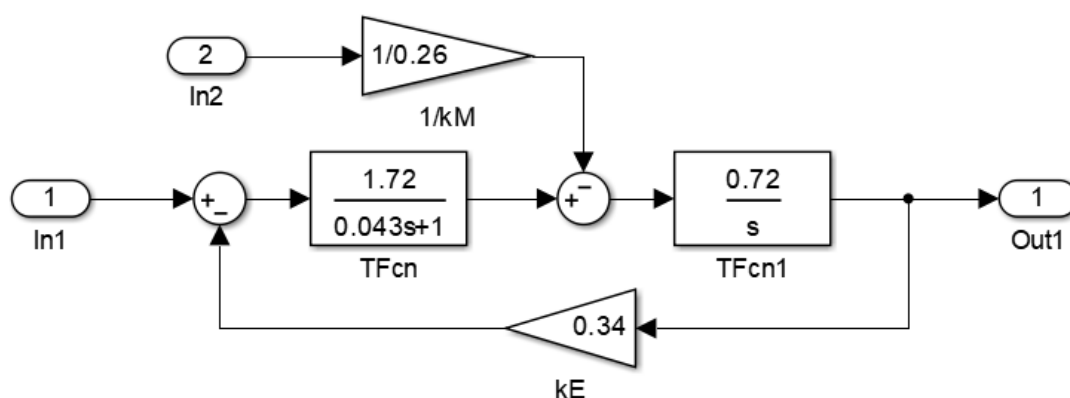


Рисунок 29 – Имитационная модель электрического двигателя

Из графиков динамических характеристик (рисунки 30, 31) видно, что по управлению и возмущению ДПТ с данными параметрами системы представляет

собой апериодическое звено. Коэффициент передачи по возмущению более чем в 2 раза больше, чем по управлению.

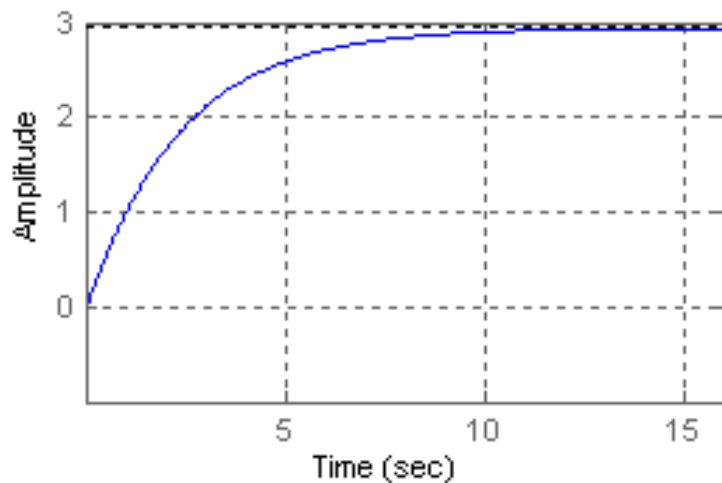


Рисунок 30 – Динамическая характеристика ДПТ по управлению

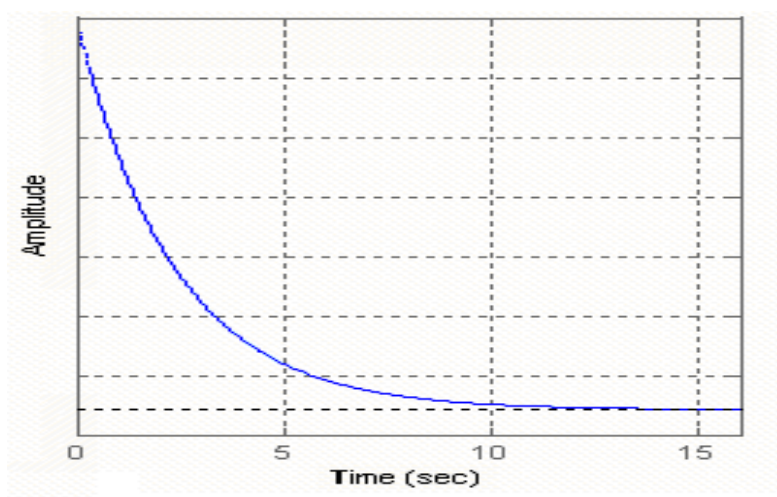


Рисунок 31 – Динамическая характеристика ДПТ по возмущению

Для обеспечения требуемых свойств динамической системы объекта управления необходимо проектирование (синтез) регулятора, структура и параметры которого зависят от параметров двигателя постоянного тока и требований к контуру скорости.

2.4 Синтез регулятора в следящей системе постоянного тока робота

Математическое описание привода звена захватного устройства робота имеет следующий вид.

$$J \frac{d\omega_m}{dt} + C_1 \omega_m + C_2 \sin \theta_m = k_M i_{\text{я}}, \quad (28)$$

где J – суммарный момент инерции якоря и механического звена робота;

$i_{\text{я}}$ – ток якоря;

$\theta_m = \int \omega_m dt$ – угол поворота звена робота.

Ток якоря двигателя постоянного тока находится из уравнения

$$T_{\text{я}} \frac{di_{\text{я}}}{dt} + i_{\text{я}} + \frac{k_E}{R_{\text{я}}} \omega_m = \frac{u_{\text{я}}}{R_{\text{я}}}. \quad (29)$$

По уравнениям (28) и (29) построена модель объекта управления, состоящая из двигателя мощностью 0,45 кВт и звена робота, рисунок 32.

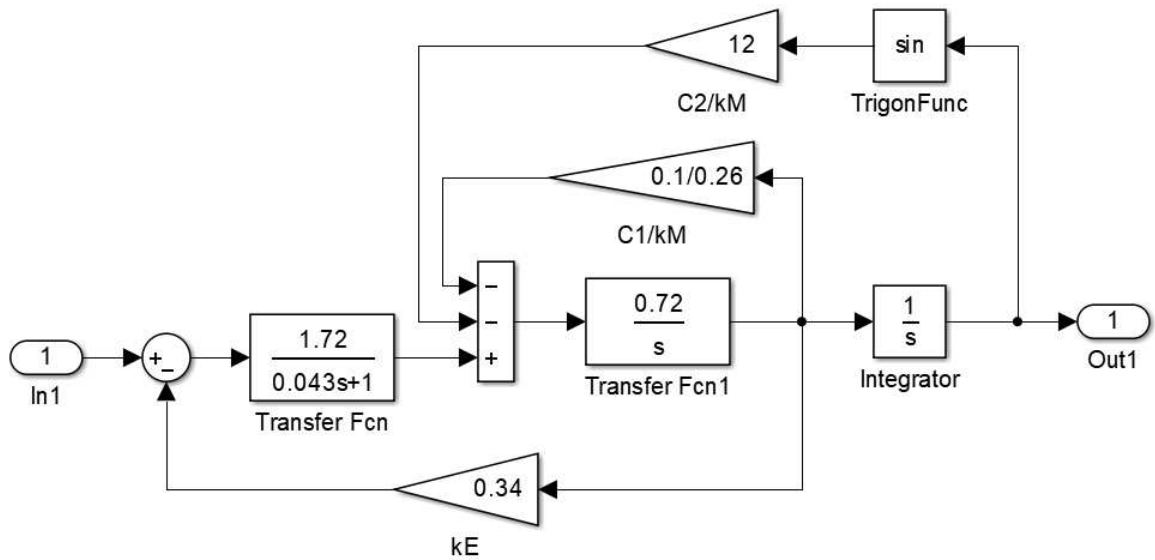


Рисунок 32 – Модель привода звена робота

Переходный процесс в системе при $C_1 = 0,1$, $C_2 = 3,2$ и малых углах поворота звена робота представлен на рисунке 33.

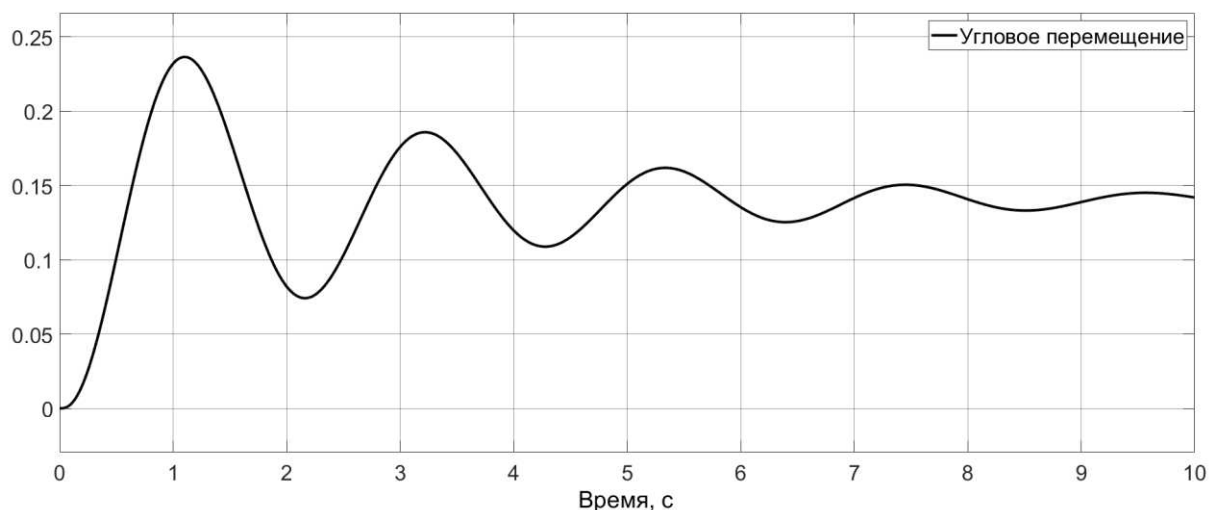


Рисунок 33 – Переходный процесс в приводе звена робота по управлению

Приведенный случай показывает, что двигатель с учетом нагрузки на его валу может быть колебательным динамическим звеном. Требуется проектирование регулятора системы управления приводом.

При проектировании регулятора в следящей робототехнической системе реализуется ограничитель диапазона для тока с использованием в контуре двигателя постоянного тока релейного регулятора. При этом контур тока считается безинерционным, а силовой регулятор – источником тока. Тогда звено робота вместе с двигателя постоянного тока описывается уравнением (30) из которого определим передаточную функцию:

$$W_{об}(s) = \frac{k_M}{C_2} \times \frac{1}{T^2 s^2 + 2\xi T s + 1}, \quad (30)$$

где $T^2 = \frac{J}{C_2}$, $2\xi T = \frac{c_1}{C_2}$, при заданных значениях $C_1 = 0,1$, $C_2 = 3,2$ параметры звена робота равны:

$T = 0,33$ с, $\xi = 0,047$, т.е. привод робота представляет собой колебательное звено.

Для такой системы используется ПИД-регулятор с реальным дифференцирующим звеном, параметры такого регулятора при настройке системы на оптимум по модулю находятся из уравнений

$$T_D < 2\xi T, \quad k_1 = \frac{C_2}{2T_D k_{oc} k_M},$$

$$\frac{k_D + T_D k_p}{k_I} = T^2, \quad \frac{k_p + T_D k_1}{k_1} = 2\xi T. \quad (31)$$

Модель двухконтурной робототехнической системы на рисунке 34.

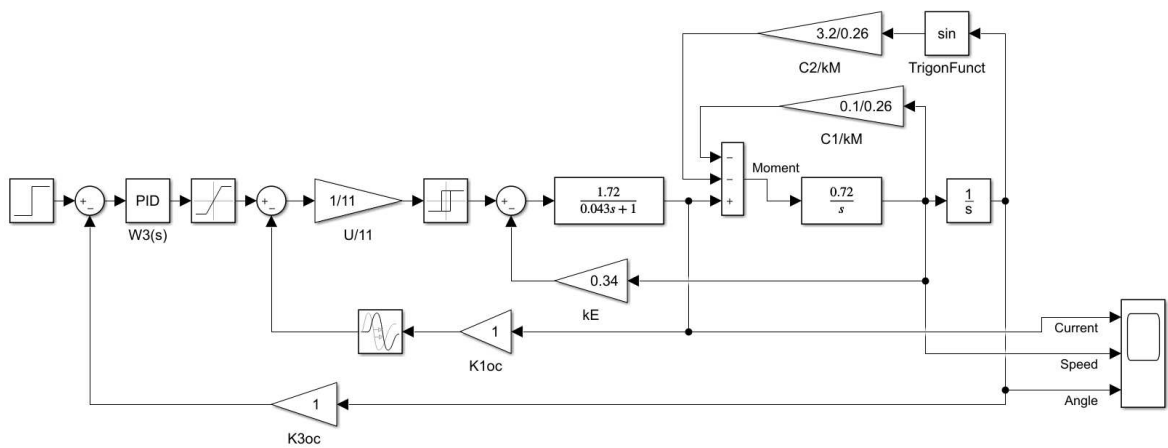


Рисунок 34 – Модель двухконтурной робототехнической системы

Динамические характеристики замкнутой системы при $T_D = 0,01$ с, $k_p = 19,25$, $k_1 = 615$, $k_D = 69,5$ показаны на рисунке 35.

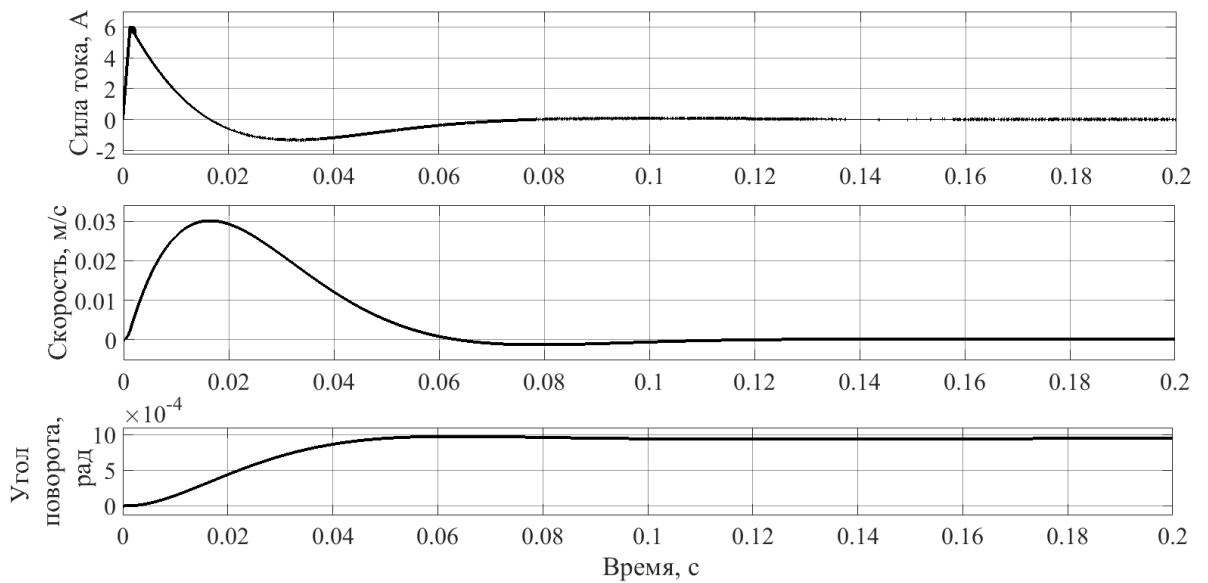


Рисунок 35 – Переходные процессы в двухконтурной робототехнической системе

Из кривых (рисунок 35) видно, что в такой системе колебаний выходной координаты (угла) не наблюдается. Поэтому такой переходной процесс можно считать удовлетворительным для проектирования.

Для исследования динамики системы получена имитационная модель экспериментального стенда звена робота, рисунок 36. Моделируется изменение входного сигнала в диапазоне от 0,001 рад до 0,003 рад.

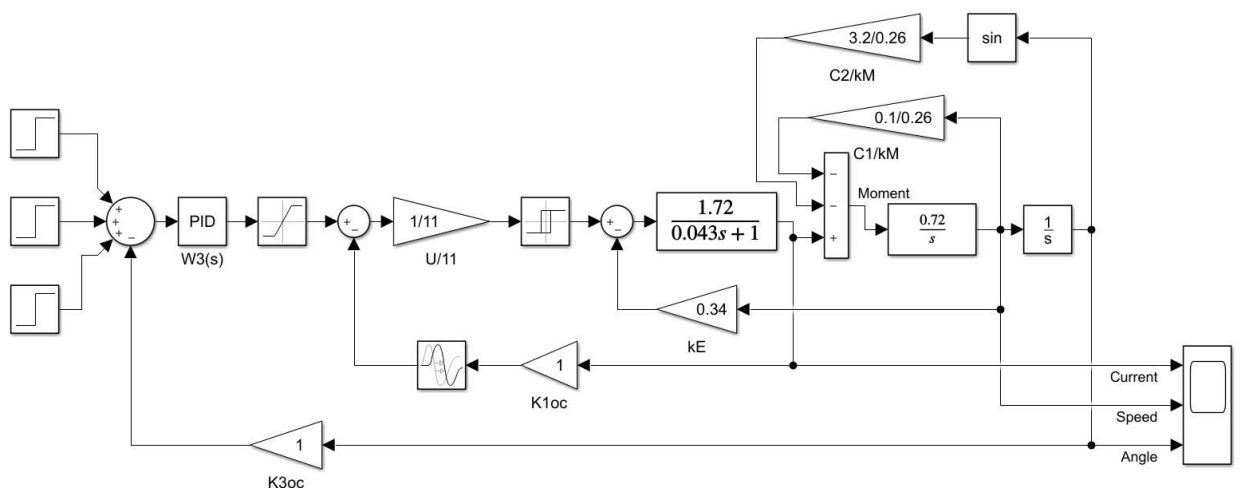


Рисунок 36 – Модель экспериментального стенда двухконтурной робототехнической системы

Переходные процессы модели экспериментального стенда показаны на рисунке 37.

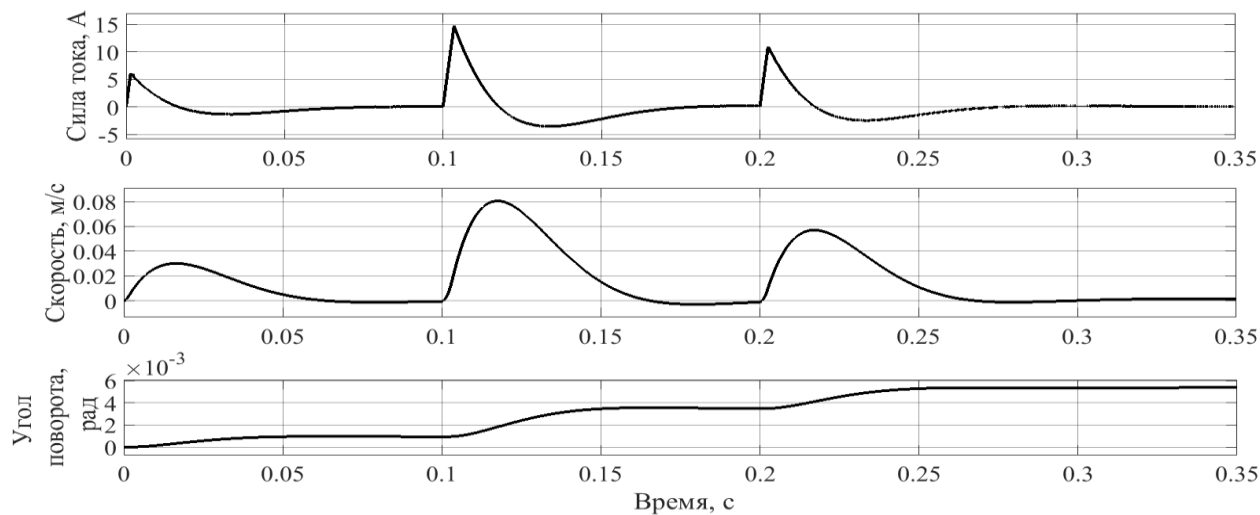


Рисунок 37 – Переходные процессы при изменении режимов работы в двухконтурной робототехнической системе

Из кривых (рисунок 37), при различных режимах работы системы, видно, что в такой системе при небольших и больших входных сигналах (при выходе системы в насыщение) колебаний выходной координаты (угла) не наблюдается. Поэтому такой переходной процесс может удовлетворить проектировщика.

3 Экспериментальные исследования

3.1 Методика

Работа посвящена моделированию системы с помощью аналоговой вычислительной машины, которая наглядно показывает физические процессы, протекающие в системе управления. Также, рассматривается моделирование системы следящих приводов в среде MATLAB (Simulink).

Исследование проводилось на примере плоского манипулятора, работающего в полярной системе координат. [9].

В процессе исследования составлена программа проведения моделирования, которая выглядит следующим образом:

- определение цели и задачи моделирования;
- системы следящих приводов исследуемого манипулятора с КПП;
- расчет начальных условий для схемы моделирования;
- расчет параметров коммутационной схемы моделирования;
- контроль соответствия соединений коммутационной схемы моделирования и проведение пробных решений, подтверждающих правильность выбора масштабов и коэффициентов передачи;
- проведение решения задачи моделирования для различных элементарных значений входных воздействий φ и γ для системы следящих приводов манипулятора с КПП и КР;
- просмотр результатов моделирования на универсальном осциллографе и регистрация их на светолучевом осциллографе;
- обработка осциллограмм и оценка результатов моделирования: определение параметров переходных процессов в системе следящих приводов манипулятора.

Компьютерное моделирование является одним из эффективных методов изучения сложных систем. Компьютерные модели проще и удобнее

исследовать, в тех случаях, когда реальные эксперименты затруднены из-за финансовых или физических препятствий или могут дать непредсказуемый результат. Логичность и формализованность компьютерных моделей позволяет выявить основные факторы, определяющие свойства изучаемого объекта. [X]

Моделирование играет ключевую роль в области роботостроения, потому что оно позволяет проводить эксперименты, которые в ином случае были бы дорогими и / или требовали больших затрат времени [X].

С помощью компьютерной динамической модели робота-манипулятора можно определить диапазон углов поворотов звеньев и рабочую зону робота с учётом ограничений его конструкции, а также определить различные возникающие моменты и силы в узлах конструкции. Разработка модели робота играет ключевую роль при его проектировании.

Существуют различные подходы к разработке динамической модели робота-манипулятора. Есть математические и компьютерные динамические модели.

Математическая динамическая модель манипулятора может быть построена на основе известных законов ньютоновской или лагранжевой механики. Результатом применения этих законов являются уравнения, связывающие действующие в сочленениях силы и моменты с кинематическими характеристиками и параметрами движения звеньев. Таким образом, уравнения динамики движения реального манипулятора могут быть получены традиционными методами Лагранжа-Эйлера или Ньютона-Эйлера.

Вывод уравнений динамики движения манипулятора методом Лагранжа-Эйлера отличается простотой и единством подхода и основан на следующем:

1) На описании взаимного пространственного расположения систем координат i -го и $(i-1)$ – го звеньев с помощью матрицы преобразования однородных координат.

2) На использовании уравнения Лагранжа-Эйлера

С целью получения более эффективных с вычислительной точки зрения алгоритмов расчета обобщенных сил и моментов используются уравнения

Ньютона-Эйлера. Вывод уравнений Ньютона-Эйлера прост по содержанию, но весьма трудоёмок. Результатом является система прямых и обратных рекуррентных уравнений, последовательно применяемых к звеньям манипулятора.

Уравнения Лагранжа-Эйлера обладают низкой вычислительной эффективностью, это обусловлено тем, что для описания кинематической цепи используются матрицы преобразования однородных координат. Уравнения Ньютона-Эйлера обладают большей вычислительной эффективностью, что связано с их векториальным представлением и их рекуррентной природой. [12]

Для построения компьютерной динамической модели робота используются программы, предназначенные для моделирования динамики и кинематики пространственных механических систем, например: MSC.ADAMS [17]; UM [17]; Unigraphics NX [17]; CATIA [17]; SolidWorks [17], MATLAB [17].

Создание модели подразумевает описание всех ее характеристик: геометрических размеров, физических свойств, способов соединения подвижных и неподвижных частей, задание действующих сил и моментов, начального положения элементов модели и их скоростей.

Этап тестирования модели включает в себя моделирование поведения частей модели под действием приложенных сил и заданных движений и выявление критических параметров, наиболее сильно влияющих на эффективность работы модели в целом.

В данной работе динамическая модель робота-манипулятора будет строиться в программной среде MATLAB, предназначенной для виртуального моделирования сложных механизмов. Широкие возможности программного пакета, высокая надёжность и малая трудоёмкость его использования позволяют исследовать множество вариантов механизмов, моделируя на компьютере реальные условия их работы, сравнивать и выбирать лучший вариант [17].

3.2 Экспериментальный стенд

Перед разработкой трёхмерной модели конструкции манипулятора необходимо составить его структурную кинематическую схему. Для начала необходимо определить тип манипулятора. Было решено разработать манипулятор ангулярного типа т.к. вращательные кинематические пары наиболее просты в конструкции и легко реализуемы, также такая конструкция имеет довольно обширную рабочую зону и своей гибкостью позволяет обходить некоторые препятствия. Также по заданию манипулятор должен иметь всего 3 степени подвижности: поворот основания вокруг своей оси и повороты 1-го и 2-го звеньев. Чтобы хват захватывал объекты правильно, он должен быть направлен в определённой ориентации, для этого в кинематическую схему добавлен специальный механизм для удержания хвата в неизменной ориентации.

Кинематическая схема разрабатывалась на основе промышленного робота Fanuc M-410iB/450 [39] (Рисунок 38), который имеет 4 степени подвижности и используется для переноски груза.

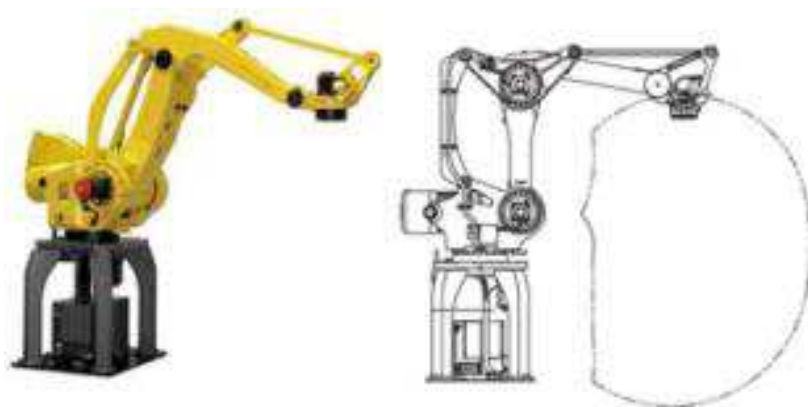


Рисунок 38 – Манипулятор Fanuc M-410iB/450 и его рабочая зона

Разработанная кинематическая схема трёхстепенного манипулятора указана на рисунке 39. На схеме указаны 1-ое и 2-ое звенья, которые

поворачиваются на углы q_1 и q_2 , основание, вокруг которого манипулятор поворачивается на угол q_0 , и схват, а также проставлены длины всех частей манипулятора.

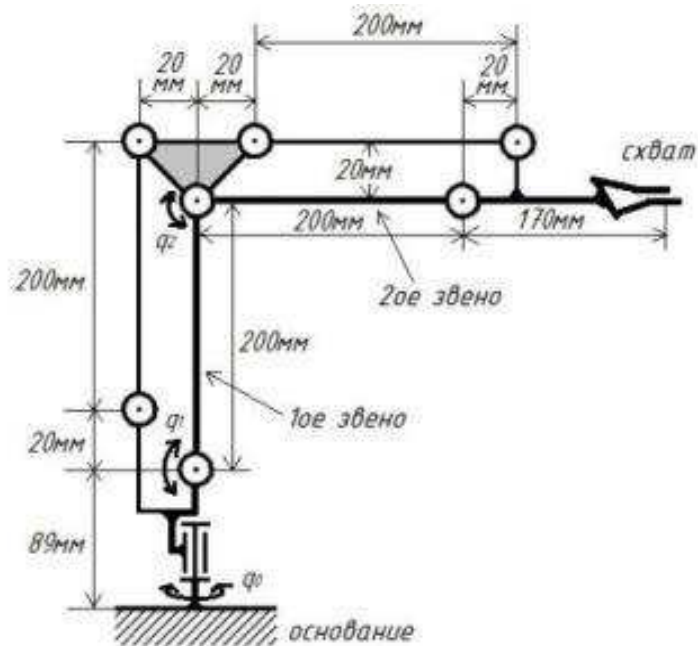


Рисунок 39 – Кинематическая схема манипулятора

В этой кинематической схеме используются некоторые дополнительные звенья и элементы, которые образуют параллельный механизм, который позволяет схвату манипулятора оставаться всегда параллельно основанию независимо от того, на какие углы q_1 и q_2 повернуться звенья манипулятора. На рисунке 40 (а, б, в, г) показан принцип работы параллельного механизма, на котором видно что при изменении углов поворота звеньев, схват робота сохраняет параллельное положение относительно основания. Данное свойство удобно для захвата, перемещения и установления объекта в новое положение, при этом сохраняется его вертикальная ориентация. Так же отпадает необходимость ставить 3е управляемое звено. Несмотря на множество шарнирных соединений в данной схеме для ориентации манипулятора используется всего 3 сервопривода, которые расположены в точках углов поворотов звеньев q_0 , q_1 , q_2 , ещё один сервопривод нужен для сжатия / разжатия схвата.

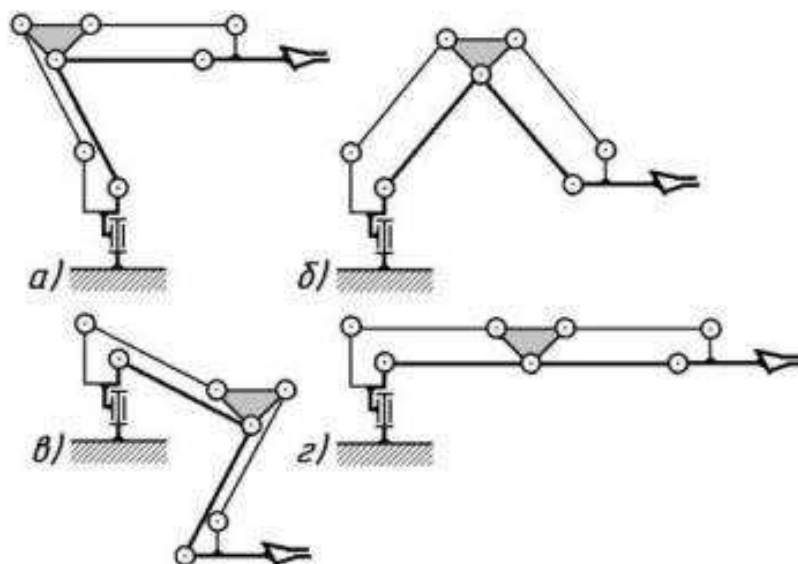


Рисунок 40 – Принцип работы параллельного механизма

3.3 Технические средства автоматизации

Рассматривается реализация системы управления приводом захватного устройства на базе аппаратно-программного комплекса Arduino. В состав экспериментального стенда входят:

- захватное устройство, рисунок 41;
- серводвигатель, рисунок 42;
- контроллер Arduino Uno, рисунок 43.

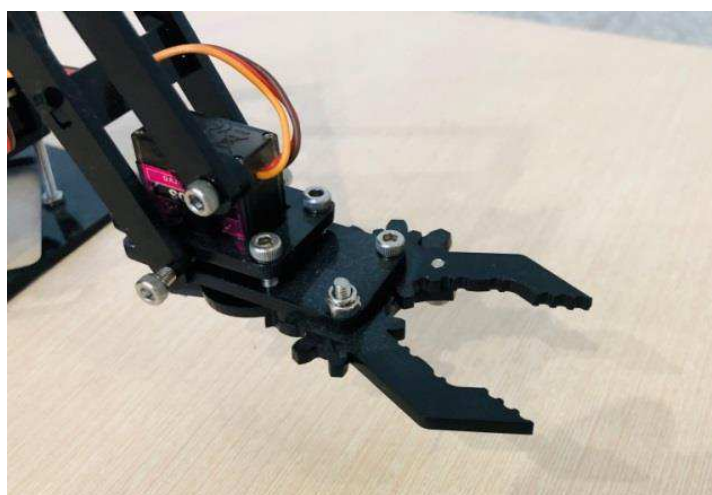


Рисунок 41 – Модель захватного устройства с серводвигателем



Рисунок 42 – Модель серводвигателя

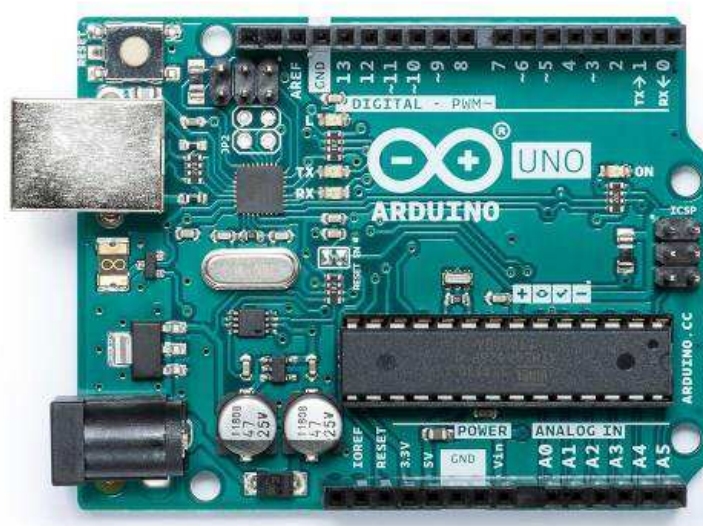


Рисунок 43 – Общий вид лицевой части контроллера Arduino UNO R3

Сервопривод Arduino устройство с электрическим мотором, которое можно повернуть на определенный угол и оставить в этом положении на определенное время.

Сервомоторы Arduino – устройства, которые могут поворачиваться в указанное положение и могут применяться в разных технических устройствах. Массово применяются в робототехнике. У сервопривода есть выходной вал, который может поворачиваться на 180 градусов. Используя Arduino можно задать сервомотору определенное положение в которое он перейдет.

Конструкция серводвигателя представлена на рисунке 44.

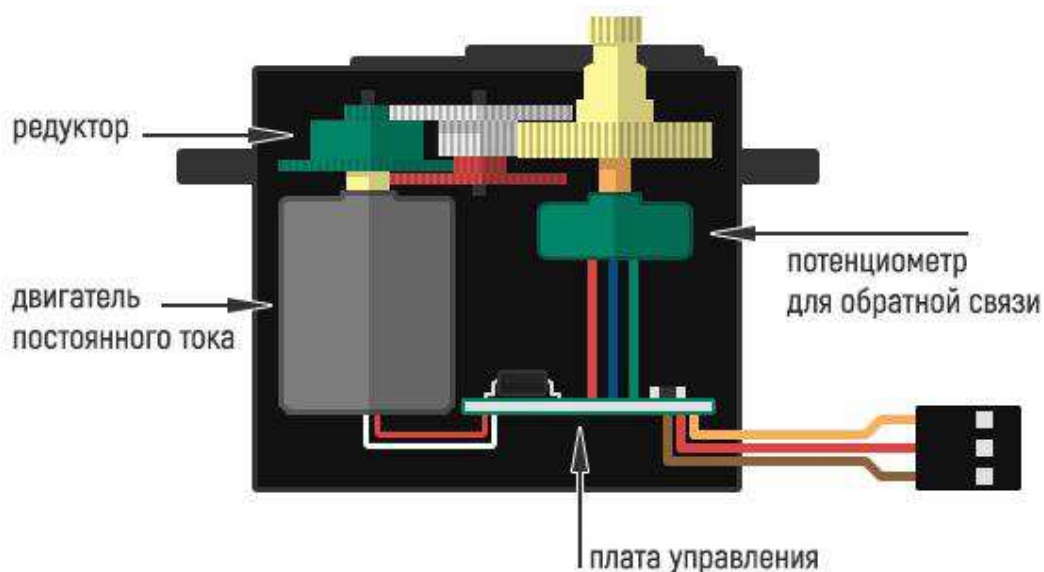


Рисунок 44 – Устройство сервопривода

Сервопривод состоит из датчика (скорости, положения и т.п.), блока управления приводом механической системы и электронной схемы. Редукторы выполняют из металла, карбона или пластика. Сервомотор имеет встроенный потенциометр, который соединен с выходным валом. Поворотом вала, сервопривод меняет значение напряжения на потенциометре (см. рисунок 40). Плата анализирует напряжение входного сигнала и сравнивает его с напряжением на потенциометре, исходя из полученной разницы, мотор будет вращаться до тех пор, пока не выровняет напряжение на выходе и на потенциометре.

Сервоприводы Arduino – это мехатронные устройства. Используя только один входной пин, они получают значения для позиционирования от микроконтроллера и переходят в это положение. На рисунке 44 сервопривод имеет электрический двигатель и цепь обратной связи, которая гарантирует, что вал сервопривода достигнет желаемого положения.

На входе серводвигатели получают сигнал «прямоугольная волна». Каждый цикл в сигнале длится 20 мс, и большая часть времени в значении LOW. В начале каждого цикла значение сигнала становится HIGH на время от 1

до 2 мс. При 1 мс составляет 0 градусов, а при 2 мс – 180 градусов, а в промежутке значение от 0 до 180, рисунок 45.

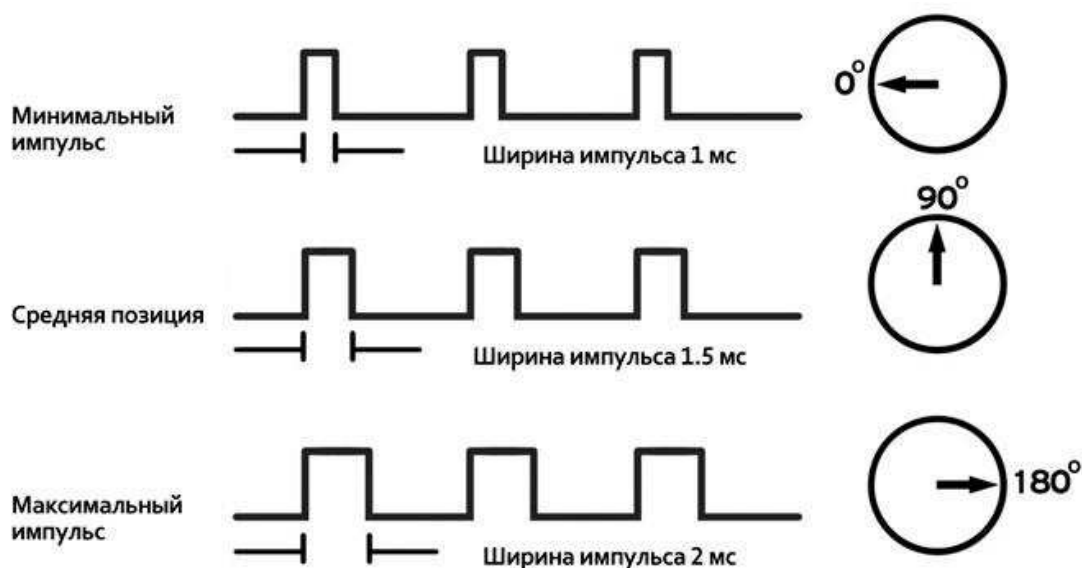


Рисунок 45 – Сигналы на входе сервопривода

Arduino – торговая марка аппаратно-программных средств для построения несложных систем автоматического управления и робототехники. В работе используется модель современной платы Arduino UNO R3, выполненная на базе микроконтроллера ATmega328. Внешний вид контроллера приведен на рисунке 43.

Кроме этой платы Arduino выпускает несколько разновидностей плат для конкретных проектов автоматизации. Выбор конкретной модели контроллера выполняется с учетом решаемых задач, доступности, стоимости, и других критериев, по которым Arduino UNO R3 превосходит другие варианты.

Arduino Uno контроллер построен на ATmega328. Платформа имеет 14 цифровых вход/выходов (6 из которых могут использоваться как выходы ШИМ), 6 аналоговых входов, кварцевый генератор 16 МГц, разъем USB, силовой разъем, разъем ICSP и кнопку перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB, либо подать питание при помощи адаптера AC/DC или батареи.

Таблица 3 – Характеристики Arduino Uno

Показатель	Характеристика
Микроконтроллер	ATmega328
Рабочее напряжение	5 В
Входное напряжение (рекомендуемое)	7-12 В
Входное напряжение (предельное)	6-20 В
Цифровые Входы/Выходы	14 (6 из которых могут использоваться как выходы ШИМ)
Аналоговые входы	6
Постоянный ток через вход/выход	40 мА
Постоянный ток для вывода 3.3 В	50 мА
Флеш-память	32 Кб (ATmega328) из которых 0.5 Кб используются для загрузчика
ОЗУ	2 Кб (ATmega328)
EEPROM	1 Кб (ATmega328)
Тактовая частота	16 МГц

Arduino Uno может получать питание через подключение USB или от внешнего источника питания. Источник питания выбирается автоматически.

Внешнее питание (не USB) может подаваться через преобразователь напряжения AC/DC (блок питания) или аккумуляторной батареей. Преобразователь напряжения подключается посредством разъема 2.1 мм с центральным положительным полюсом. Провода от батареи подключаются к выводам Gnd и Vin разъема питания.

Платформа может работать при внешнем питании от 6 В до 20 В. При напряжении питания ниже 7 В, вывод 5V может выдавать менее 5 В, при этом платформа может работать нестабильно. При использовании напряжения выше 12 В регулятор напряжения может перегреться и повредить плату. Рекомендуемый диапазон от 7 В до 12 В.

VIN. Вход используется для подачи питания от внешнего источника (в отсутствие 5 В от разъема USB или другого регулируемого источника питания). Подача напряжения питания происходит через данный вывод.

5V. Регулируемый источник напряжения, используемый для питания микроконтроллера и компонентов на плате. Питание может подаваться от вывода VIN через регулятор напряжения, или от разъема USB, или другого регулируемого источника напряжения 5 В.

3V3. Напряжение на выводе 3.3 В генерируемое встроенным регулятором на плате. Максимальное потребление тока 50 мА.

GND. Выводы заземления.

Микроконтроллер ATmega328 располагает 32 кБ флэш памяти, из которых 0.5 кБ используется для хранения загрузчика, а также 2 кБ ОЗУ (SRAM) и 1 Кб EEPROM.

3.4 Обработка результатов экспериментальных исследований

Платформа программируется посредством ПО Arduino. Микроконтроллер ATmega328 поставляется с записанным загрузчиком, облегчающим запись новых программ без использования внешних программаторов. Связь осуществляется оригинальным протоколом STK500.

В основу языка программирования, используемого в проектах Arduino, положен C++ – один из самых широко используемых языков программирования, поддерживающий как работу с низкоуровневыми командами, так и построение сложных объектов. Программирование контроллеров Arduino удобно осуществлять в специальной среде Arduino IDE, поскольку в нее включен основной функционал для работы с ними.

Arduino IDE можно скачать в Интернете по адресу основного сайта проекта: www.arduino.cc/en/Main/Software, программное обеспечение свободно распространяется.

До начала работы на компьютер необходимо установить программу Arduino IDE. Его надо запустить на выполнение с административными полномочиями, принять условия лицензии GNU LESSER GENERAL PUBLIC LICENSE и согласиться с предложенным вариантом установки. На все предупреждения Windows в процессе установки следует отвечать утвердительно (продолжать установку). Когда установщик предложит установить драйверы порта, также ответить утвердительно.

Среда разработки имеет интуитивно понятный русскоязычный интерфейс. При запуске установленной Arduino IDE откроется окно (рисунок 46), в котором уже содержится заготовка программы. Она состоит из двух функций: `setup` и `loop`. Функция `setup` содержит команды, выполняемые один раз при включении Arduino, – это установка номеров портов ввода/вывода для управления мониторами и установки скорости обмена данными между Arduino и компьютером.



```
sketch_may28a | Arduino 1.8.9
Файл Правка Скетч Инструменты Помощь
sketch_may28a
#include <Servo.h>

void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Рисунок 46 – Окно Arduino IDE

Функция `loop` выполняется бесконечное число раз – до тех пор, пока мы не отключим питание.

Подключение контроллера Arduino к компьютеру.

Программная среда позволяет в зависимости от типа контроллера Arduino использовать разные USB-кабели.

Контроллер Arduino UNO подключается к компьютеру через USB интерфейс и устанавливает связи между ним и оболочкой Arduino IDE. Для этого нужно задать номер порта, к которому подключен контроллер (рисунок 47).

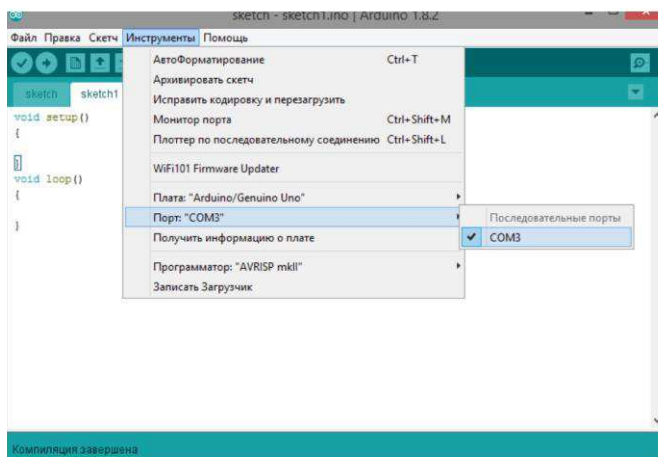


Рисунок 47 – Выбор нужного порта

Затем выбирают тип контроллера Arduino. На рисунке 48 показан выбор Arduino/Genuino Uno. Если контроллер выбран системой автоматически, необходимо проверить правильность этого выбора. Для некоторых контроллеров необходимо еще выбрать подвид микроконтроллера, на котором реализована плата Arduino.

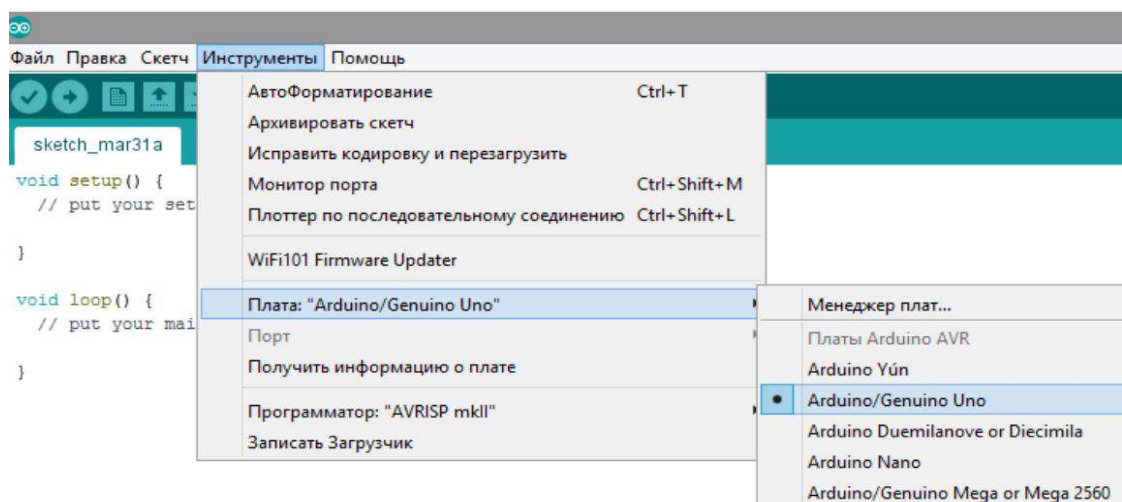


Рисунок 48 – Выбор типа контроллера Arduino

Написанная программа в Arduino IDE загружается в контроллер нажатием кнопки со стрелкой вправо, рисунок 48. Оболочка проверит программу на наличие ошибок, а затем переведет ее в двоичный код данных и команд выбранного микроконтроллера и запишет в Arduino.

Если после появления слова Загрузка в нижней строке окна Arduino IDE замигали светодиоды TX и RX на плате Arduino, то загрузка программы в плату началась успешно.

Функции являются поименованными блоками команд, которые по их имени можно вызвать из любого места программы. Функция — это блок кодов программы, у которых есть имя и ряд команд, которые запускаются при вызове функции. Например, функции `void setup` и `void loop`. Существуют встроенные функции — они уже присутствуют в библиотеке функций Arduino IDE — и функции, созданные в рамках текущей программы. Также существуют библиотеки, которые можно подключать к текущей программе и пользоваться имеющимися в них функциями. Подключаются библиотеки из пункта оболочки Скетч | Подключить библиотеку. После подключения той или иной библиотеки, можно использовать содержащиеся в ней готовые программные решения, написанные для различных устройств и датчиков.

4 Реализация результатов работы

На данный момент, на рынке существует множество микросхем, с помощью которых реализуется концепция, предложенная мною в разделе магистерской диссертации Arduino - это микроконтроллеры, а не полноценные компьютеры. На них нет операционной системы как таковой, Arduino просто выполняет код, интерпретируемый прошивкой. В данном случае вы не имеете в распоряжении базовых инструментов, предоставляемых операционной системой, но, с другой стороны, такое непосредственное выполнение несложного кода протекает проще, а при работе не возникает никаких издержек, связанных с операционной системой. Основное назначение платы Arduino – взаимодействие с сенсорами и устройствами, поэтому Arduino отлично подходит для аппаратных проектов, где требуется просто реагировать на различные сигналы сенсоров и ручной ввод. Может показаться, что в этом нет ничего особенного, однако на деле Arduino – сложная выверенная система, значительно облегчающая управление устройствами. Она отлично подходит именно для сочленения других устройств и исполнительных механизмов, где полноценная операционная система просто не требуется, так как речь идет просто о регистрации действий и реагировании на них.

4.1 Реализация системы управления приводом захватного устройства на базе Arduino

Серводвигатель подключается программно. Сначала подключается библиотека (`#include <Servo.h>`), а затем создается объект `servomotor`.

Контакт серводвигатель должен быть физически подключен к порту Arduino UNO R3, и на мотор подано электропитание. Серводвигатель можно подключить непосредственно к плате (рисунок 49), или через макетную плату (рисунок 50).

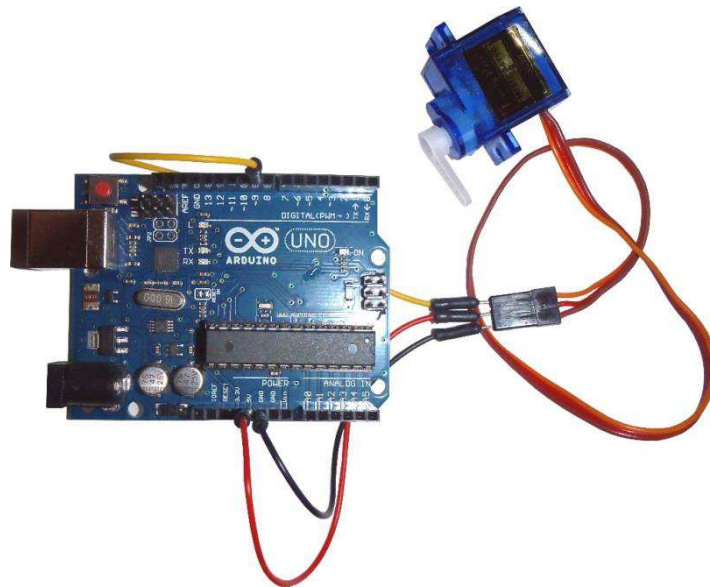


Рисунок 49 – Подключение серводвигателя к контроллеру Arduino

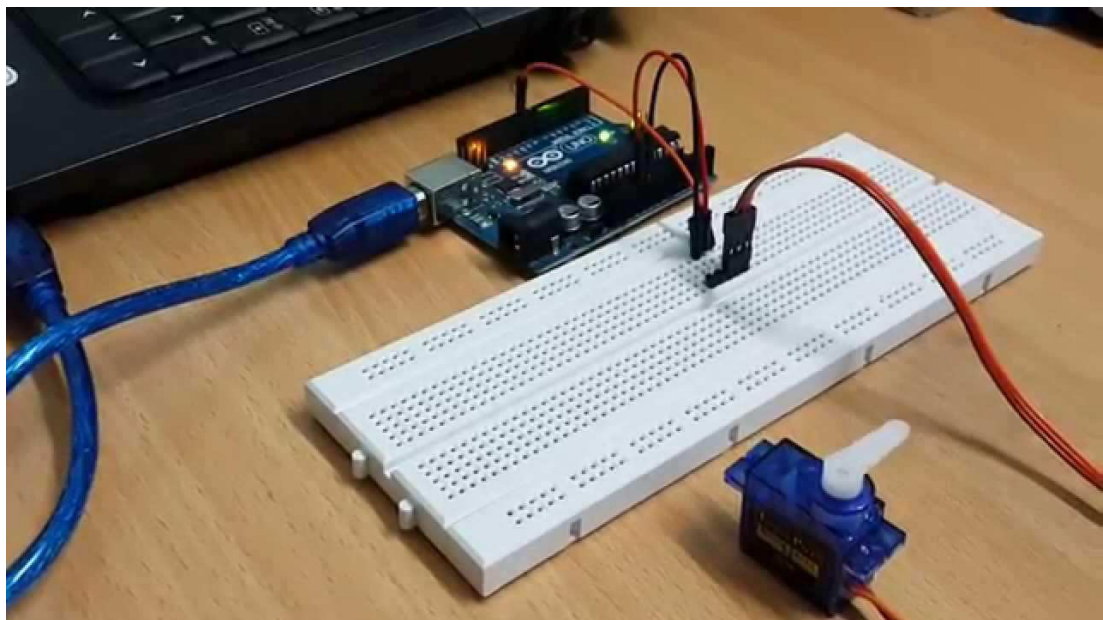


Рисунок 50 – Подключение серводвигателя к контроллеру Arduino через макетную плату

План физического эксперимента с сервоприводом и программная реализация – вариант 1:

```
#include <Servo.h> // Подключаем дополнительную библиотеку
Servo servo1; // Объект управления сервоприводом
void setup()
{
```

```

servo1.attach(9); // Назначается пин 9 для подключения серводвигателя
}
void loop()
{
int position;
servo1.write(90); // Команда серве повернуться на позицию 90 градусов
delay(1000);      // Пауза 1 мкс, чтобы дать время повернуться
servo1.write(180); // Команда серве повернуться на позицию 180 градусов
delay(1000);      // Пауза 1 мкс
servo1.write(0);  // Команда серве повернуться на позицию 0 градусов
delay(1000);      // Пауза 1 мкс
// Формируется команда сервоприводу повернуться на позицию 180 градусов,
// с шагом в два градуса:
for(position = 0; position < 180; position += 2)
{
servo1.write(position); // Команда перемещения на следующую позицию
delay(20);              // Короткая пауза
}
// Формируется команда сервоприводу повернуться на позицию 0 градусов,
// с шагом в 1 градус:
for(position = 180; position >= 0; position -= 1)
{
servo1.write(position); // Команда сервоприводу переместиться на
// следующую позицию
delay(20);              // Короткая пауза
}
}

```

План физического эксперимента с сервоприводом и программная реализация – вариант 2:

```
int servopin=9;// select digital pin 9 for servomotor signal line
```



```

int myangle;// initialize angle variable
int pulsewidth;// initialize width variable
int val;
void servopulse(int servopin,int myangle)// define a servo pulse function
{
pulsewidth=(myangle*11)+500;// convert angle to 500-2480 pulse width
digitalWrite(servopin,HIGH);// set the level of servo pin as “high”
delayMicroseconds(pulsewidth);// delay microsecond of pulse width
digitalWrite(servopin,LOW);// set the level of servo pin as “low”
delay(20-pulsewidth/1000);
}
void setup()
{
pinMode(servopin,OUTPUT);// set servo pin as “output”
Serial.begin(9600);// connect to serial port, set baud rate at “9600”
Serial.println("servo=o_seral_simple ready" );
}
void loop()// convert number 0 to 9 to corresponding 0-180 degree angle, LED blinks
corresponding number of time
{
val=Serial.read();// read serial port value
if(val>='0'&&val<='9')
{
val=val-'0';// convert characteristic quantity to numerical variable
val=val*(180/9);// convert number to angle
Serial.print("moving servo to ");
Serial.print(val,DEC);
Serial.println();
for(int i=0;i<=50;i++) // giving the servo time to rotate to commanded position
{

```

```
servopulse(servopin,val);// use the pulse function  
}  
}  
}
```

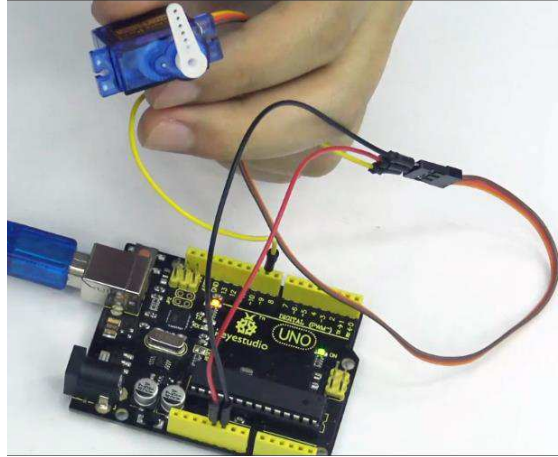


Рисунок 51 – Реализация варианта 2 эксперимента с серводвигателем

Результаты отработки рабочего процесса программного управления сервоприводом показан на рисунке 52.

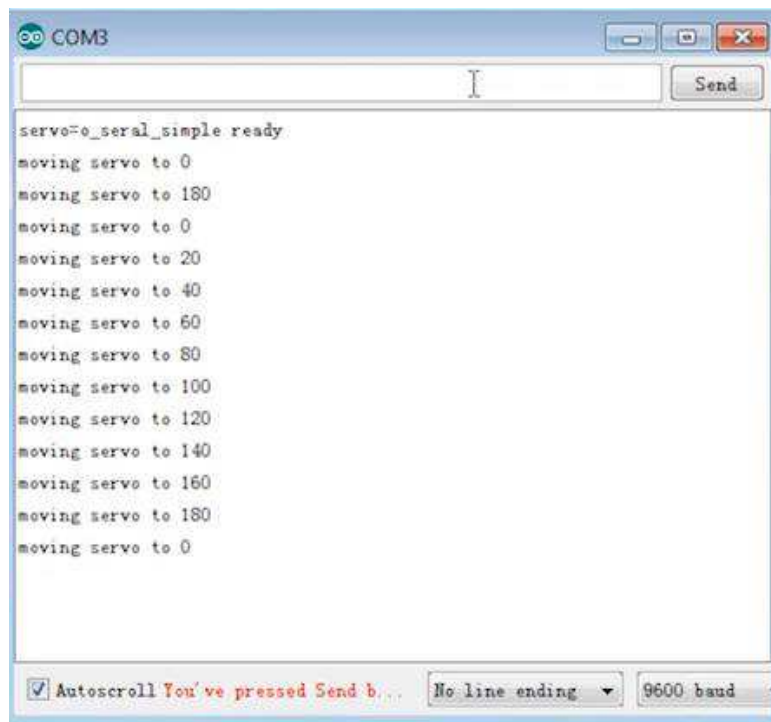


Рисунок 48 – Результаты рабочего процесса программного управления сервоприводом

4.2 Элементы объектно-ориентированного программирования

Язык C++, составляющий основу для программирования в проекте Arduino, является объектно-ориентированным. Объекты – это сложные структуры, которые имеют оригинальное название, могут включать в себя как переменные, так и функции. Фактически, объекты созданы для удобного обращения к сложным структурам.

4.3 Запуск моделей Simulink на Arduino

Подключение Arduino к SIMULINK официальная страница:
<http://www.mathworks.com/academia/arduino-software/arduino-simulink.html>

Требуются: MATLAB (R2010a или позже) Simulink.

Скачивается пакет Simulink Support Package for Arduino
<http://www.mathworks.com/matlabcentral/fileexchange/30277> и распаковывается его на диске компьютера, например в c:\arduino_simulink. Также скачивается среда Arduino IDE и распаковывается, к примеру, в c:\ArduinoTarget.

Выполняются команды для добавления путей:

```
>> cd c:\arduino_simulink
>> addpath(fullfile(pwd,'arduino'),fullfile(pwd,'blocks'),fullfile(pwd,'demos'))
>> savepath.
```

Далее, обновляются кастомизации:

```
>> sl_refresh_customizations.
```

Подключается плата Arduino к компьютеру. Прошивается. Указывается путь к среде Arduino:

```
>>arduino.Prefs.setArduinoPath('c:\ArduinoTarget')
```

Определяется текущая платформа командой

```
>> arduino.Prefs.setBoard('mega')
```

Список всех доступных платформ выводится командой

```
>> arduino.Prefs.setBoard
```

Specify one of the following board labels:

'uno' (Arduino Uno)

'atmega328' (Arduino Duemilanove or Nano w/ ATmega328)

'diecimila' (Arduino Diecimila, Duemilanove, or Nano w/ ATmega168)

'mega2560' (Arduino Mega 2560)

'mega' (Arduino Mega (ATmega1280))

'mini' (Arduino Mini)

'fio' (Arduino Fio)

'bt328' (Arduino BT w/ ATmega328)

'bt' (Arduino BT w/ ATmega168)

'lilypad328' (LilyPad Arduino w/ ATmega328)

'lilypad' (LilyPad Arduino w/ ATmega168)

'pro5v328' (Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328)

'pro5v' (Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168)

'pro328' (Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328)

'pro' (Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168)

'atmega168' (Arduino NG or older w/ ATmega168)

'atmega8' (Arduino NG or older w/ Atmega8)

Проверяются доступные порты:

```
>> comPorts=arduino.Prefs.searchForComPort
```

```
comPorts =
```

```
'COM3'
```

И подключаемся к нему:

```
>> arduino.Prefs.setComPort('COM3');
```

Библиотека блоков Simulink инсталлированная с официального сайта показана на рисунке 53.

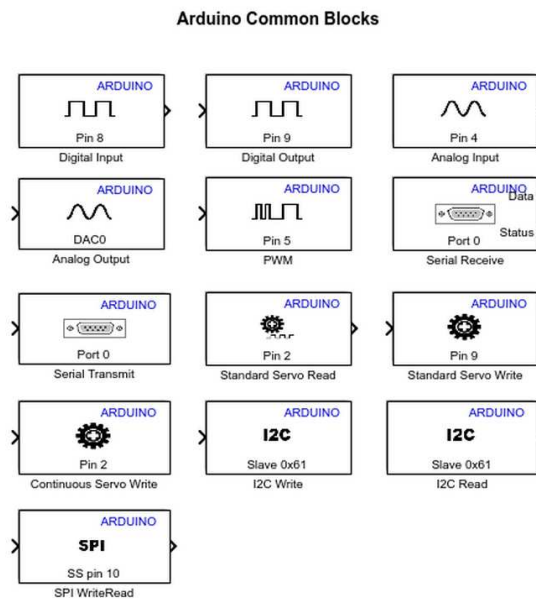


Рисунок 53 – Блоки Simulink/Arduino

Запуск проекта Simulink Servo Control системы управления сервоприводом показан на рисунке 54:

```
>> arduino_servocontrol_sweep
```

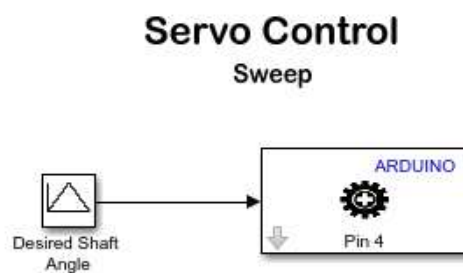


Рисунок 54 – Имитационная модель управления сервоприводом на языке Simulink

Достоинством использования MATLAB/Simulink и Arduino является сопряжение компьютерных моделей с реальными стендами.

ЗАКЛЮЧЕНИЕ

В результате проведенного исследования проанализировано сопоставление физического, структурного и математического моделирования с результатами имитационного моделирования двигателя постоянного тока с независимым возбуждением при помощи пакета программ MATLAB. Для этого были решены следующие задачи: разработана имитационная модель; рассчитаны дополнительные параметры двигателя постоянного тока, необходимые для моделирования; разработаны физическая, структурная и математическая модели с помощью библиотек блоков Simulink, а также непосредственно в окне Command Window среды MATLAB; полученные результаты моделирования сопоставлены с результатами имитационного моделирования; определены абсолютная и относительная погрешности в результатах.

В результате чего был сделан выбор в пользу робота-манипулятора, как наиболее перспективного объекта для применения в строительстве.

Следует отметить, что физическое моделирование является более сложным как при разработке модели, так и при анализе результатов моделирования по сравнению со структурным и математическим моделированием. Но в то же время физическое моделирование позволяет получить и проанализировать намного больше физических (в нашем случае электрических) параметров, которые важны для оценки работы электромеханической системы. Структурная модель в построении проще физической, но требует дополнительных расчетов постоянных времени и констант двигателя для адекватного моделирования. Преимущество структурной модели заключается в том, что блок измерителя (осциллограф) может быть подключен к любой точке модели, в отличие от физической модели, где измерители подключаются только к специальным блокам. Исходя из всех результатов, можно смело утверждать, что математическое моделирование для двигателя постоянного тока является наиболее

предпочтительным, благодаря малым погрешностям и высокой точности результатов. Также при математическом моделировании легко произвести оценку передаточной функции на устойчивость, не прибегая к дополнительным расчетам.

Проведен анализ технологических процессов и оборудования, использующих манипуляционные роботы.

Выделены важные исторические циклы развития роботизации.

Разработанный метод исследования динамики манипуляторов.

Таким образом, под управлением роботом понимают комплекс задач, связанных с выбором робота, его адаптацией к технологическому процессу, непосредственным программированием движений, а также синтезом и модификацией системы управления и ее программного обеспечения. Для решения этой задачи будем пользоваться математическим аппаратом, используемым для анализа кинематики и динамики манипуляторов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Попов, Е.П. Робототехника и гибкие производственные системы / Е.П. Попов – Москва; Наука, 1987. – 192 с.
- 2 ElMaraghy Hoda. Changable and Reconfigurable Manufacturing Systems. – Springer, 2009. – 405 p.
- 3 Красильникъянц, Е.В. Применение контроллеров движения для систем управления электромеханическими объектами/ Е.В. Красильникъянц, А.П. Бурков, В.А. Иванков // Мехатроника. Автоматизация. Управление. – 2008. – Вып. №2. – С. 45-50.
- 4 Фу, К. Робототехника / К. Фу, Р. Гонсалес, К. Ли – Москва; Мир, 1989. – 624 с.
- 5 Шахинпур, М. Курс робототехники / М. Шахинпур – Москва; Мир, 1990. – 527 с.
- 6 Ахромеев, Ж.П. Робототехника и гибкие автоматизированные производства/ под ред. И.М. Макарова, учеб. пособие для вузов. Кн. 2. Приводы робототехнических систем, Ж.П. Ахромеев, Н.Д. Дмитриева, В.М. Лохин и др.; под общ. ред. И.М. Макарова. – Москва; Высш. шк. 1986. – 175 с.
- 7 Козырев, Ю. Г. Роботизированные производственные комплексы / Ю. Г. Козырев, А. А. Кудинов, В. Э. Булатов и др.; Под ред. Ю. Г. Козырева, А. А. Кудинова – Москва; Машиностроение, 1987. – 270 с.
- 8 ГОСТ Р 60.0.0.2-2016 Роботы и робототехнические устройства. Классификация. – Введ.01.01.2018. Москва : Стандартинформ, 2018. – 15 с.
- 9 Corke Peter. The Unimation Puma Servo System. CSIRO. 1994.
- 10 Голубенцев, Н.Д. Некоторые вопросы оценки качества роботов-манипуляторов / Н.Д. Голубенцев, А.А.Халфен //: Робототехника. – 1976. – часть I – С.29-33.
- 11 Саблин, А.Д. Некоторые методы оценки точности позиционирования манипулятора. / А.Д. Саблин, В.М. Сорин // В кн: Робототехника. – 1976, С.50-54.

12 Кузовков, Н. Т. / Модальное управление и управляющие системы / Н. Т. Кузовков– М.: 1976. – 184 с.

13 Красильникъянц, Е. В. Цифровой асинхронный электропривод подачи/ Е. В Красильникъянц., А. П. Бурков, А. А. Смирнов, Г. А. Бурков // РИТМ. – 2012. - №4 – С. 44-49.

14 Verdonck Walter, Swevers Jan. Improving the Dynamic Accuracy of Industrial Robots by Trajectory Pre-Compensation. // Proceedings. 2002 IEEE International Conference on Robotics and Automation. 2002.

15 Somsawas Tungpataratanawong, Kiyoshi Ohishi, Toshimasa Miyazaki. Robust Motion Control of Industrial Robot Based on Robot Parameter Identification and Feedforward Control Considering Resonant Frequency. // IEEJ Trans. IA, Vol. 125. 2005.

16 Тютиков, В. В. Система управления манипуляционным роботом с компенсацией динамических моментов/ В. В Тютиков., Е. В. Красильникъянц, А. А. Варков// Автоматизация в промышленности – 2015. – С. 58-63.

17 Popov A., Farag A., Werner H. Tuning of a PID controller Using a Multiobjective Optimization Technique Applied to A Neutralization Plant. - 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05, 12-15 Dec. 2005, p.7139 - 7143.

18 Ротач, В.Я. К расчету оптимальных параметров реальных ПИД-регуляторов по экспертным критериям/ В.Я. Ротач // Промышленные АСУ и контроллеры. – 2006. – №2. – С. 22-29

19 ГОСТ 25686-85. Манипуляторы, автооператоры и промышленные роботы. Введ.01.01.1986. Москва : Стандартинформ, 1985. – 6 с.

20 Юревич Е.И. Управление роботами и робототехническими системами/ Е.И. Юревич – СПб; – 2001. – 168 с.

21 Канаев, Е.М Конструкции промышленных роботов: Учеб. Пособие для СПТУ/ Е.М.Канаев, Ю.Г. Козырев, Б.И. Черпаков, В.И. Царенко. – Москва; – 1987. – 95 с.

22 ElMaraghy Hoda. *Changable and Reconfigurable Manufacturing Systems*. – Springer, 2009. – 405 p.

23 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной и научной деятельности. СТО 4.2- 07-2014.

24. Лебедев С.К., Колганов А.Р. Регуляторы положения систем позиционирования с динамикой Бесселя для электромехатронных модулей // Вестник Ивановского государственного энергетического университета. – 2015. – № 6. – С. 41-47.

25. Gnezdov N.E., Lebedev S.K., Kolganov A.R. Structural and parametric synthesis of positioning systems of electromechatronic modules // 9th International Conference on Power Drives Systems (ICPDS). – 2016.

26. Радионов А.А., Карандаев А.С., Евдокимов А.С., Андрюшин И.Ю., Гостев А.Н., Шубин А.Г. Математическое моделирование взаимосвязанных электромеханических систем непрерывной подгруппы клеток прокатного стана. Часть 1. Разработка математической модели // Вестник Южно-Уральского государственного университета. Серия: Энергетика. – 2015. – Т. 15. – № 1. – С. 59-73.

27. Шевырёв Ю.В., Моргачёв Д.А. Исследование электромагнитной совместимости частотно-регулируемого электропривода буровой установки и источника электроэнергии соизмеримой мощности // Вестник Южно-Уральского государственного университета. Серия: Энергетика. – 2015. – Т. 15. – № 2. – С. 25-33.

28. Шрейнер Р.Т. Системы подчиненного регулирования электроприводов. Учебное пособие для ВУЗов. Екатеринбург, Изд-во Урал. гос. проф.-пед.ун-та, 1997.

29. Дьяконов В.П. MATLAB R2006/2007/2008 + SIMULINK 5/6/7. Основы применения. Учебное пособие. – М.: СОЛОН-Пресс, 2008. – 800 с.

30. Илюхин, А.В. Математическое описание объектов автоматизации строительного производства: учебное пособие / А.В. Илюхин, А.М. Колбасин, В.И. Марсов. – Москва: МАДИ, 2016.

31. Ковчин С.А., Сабинин Ю.А. Теория электропривода. Санкт-Петербург, ЭНЕРГОАТОМИЗДАТ, Санкт-Петербургское отделение, 1994.

32. Тихонов, А.Ф. Автоматизация строительных и дорожных машин [Электронный ресурс]: учебное пособие / А.Ф. Тихонов, С.Л. Демидов, А.Н. Дроздов. – Электрон. текстовые данные. – Москва: Московский государственный строительный университет, ЭБС АСВ, 2013. – Режим доступа: <http://www.iprbookshop.ru/23716>.

33. Фигаро Б.И., Павлячек Л.Б. Теория электропривода. Минск, ЗАО Техноперспектива, 2004.

34. Филипс Ч., Харбор Р. Системы управления с обратной связью. Москва. Лаборатория базовых знаний. 2001.

35. Цифровые электроприводы с транзисторными преобразователями/ С.Г. Герман-Галкин, В.Д. Лебедев, Б.А. Марков, Н.И. Чичерин. - Л.: Энергоатомиздат. Ленингр. отд-е, 1986.

36. Черных И.В. SIMULINK среда создания инженерных приложений. Москва, ДИАЛОГ-МИФИ, 2004.

37. Шрейнер Р.Т. Системы подчиненного регулирования электроприводов. Учебное пособие для ВУЗов. Екатеринбург, Изд-во Урал.гос. проф.-пед.ун-та, 1997.

38. Щербаков, В.С. Теория автоматического управления. Линейные непрерывные системы: учебное пособие / В.С. Щербаков, И.В. Лазута. – Омск: СиБАДИ, 2013.

39. Fanuc M-410iB Series [электронный ресурс]. URL: [http://www.fanucrobotics.com/cmsmedia/datasheets/M-410iB % 20Series_15.pdf](http://www.fanucrobotics.com/cmsmedia/datasheets/M-410iB%20Series_15.pdf) (дата обращения: 11.06.2015).

ПРИЛОЖЕНИЕ А

ПУБЛИКАЦИИ

1) Андреев Н.Ю. и др. [Моделирование динамики строительных манипуляторов](#) // [Математические методы в технике и технологиях - ММТТ](#). 2019. Т. 8. – СПб.: Изд-во Политехн. ун-та, 2019. – С. 132-135.

2) Разработка имитационной модели строительного робота мостового типа / Н.Ю. Андреев, В.В. Алешин // XV Международная конференция студентов, аспирантов и молодых ученых «Перспективны – 2019».

3) Манипуляторы в дорожном строительстве / Н.Ю. Андреев, В.В. Горобец, Э.И. Османов, С.В. Пиндур // Вестник магистратуры. 2019. № 6-2. – Йошкар-Ола: Изд-во ООО «Коллоквиум», 2019. – 106 с. (С. 90-93).

4) Андреев Н.Ю. [Программа моделирования системы следящих приводов манипулятора](#). Свидетельство о регистрации программы для ЭВМ RU 2019667144, 19.12.2019. Заявка № 2019666291 от 10.12.2019.

5) Андреев Н.Ю. Моделирование динамики строительного робота // XVI Международная конференция студентов, аспирантов и молодых ученых «Перспективны – 2020» в формате онлайн 23.05.2020 г. Секция: [Строительные материалы и инновационные технологии строительства](#). Красноярск: Сиб. федер. ун-т, 2020.

ПРИЛОЖЕНИЕ Б

Название программы для ЭВМ: Программа моделирования системы следящих приводов манипулятора.

Реферат: Программа предназначена для моделирования системы следящих приводов манипулятора. Моделирование процесса выполняется на основе параметров конструктивных элементов манипулятора и следящего привода. Обеспечивается удобство настройки исходных данных и отображения результатов моделирования. Программа предназначена для научных и инженерно-технических работников, занимающихся проектированием, разработкой и созданием систем управления, а также для аспирантов и студентов соответствующих специальностей.

Тип ЭВМ: IBM PC-совмест. ПК. ОС: Windows 7/8/8.1/10.

Язык программирования: MATLAB/Simulink.

Объем программы для ЭВМ: 76 Кб.

Листинг программы MAG язык программирования C/C++

File: ert_main.cpp

```
1 //
2 // File: ert_main.cpp
3 //
4 // Code generated for Simulink model 'mag2'.
5 //
6 // Model version : 1.8
7 // Simulink Coder version : 8.13 (R2017b) 24-Jul-2017
8 // C/C++ source code generated on : Wed Oct 16 20:56:56 2019
9 //
10 // Target selection: ert.tlc
11 // Embedded hardware selection: Intel->x86-64 (Windows64)
12 // Code generation objectives:
13 // 1. Execution efficiency
14 // 2. RAM efficiency
15 // Validation result: Not run
```

```

16 //
17 #include <stddef.h>
18 #include <stdio.h> // This ert_main.c example uses printf/fflush
19 #include "mag2.h" // Model's header file
20 #include "rtwtypes.h"
21
22 static mag2ModelClass rtObj; // Instance of model class
23
24 //
25 // Associating rt_OneStep with a real-time clock or interrupt service routine
26 // is what makes the generated code "real-time". The function rt_OneStep is
27 // always associated with the base rate of the model. Subrates are managed
28 // by the base rate from inside the generated code. Enabling/disabling
29 // interrupts and floating point context switches are target specific. This
30 // example code indicates where these should take place relative to executing
31 // the generated code step function. Overrun behavior should be tailored to
32 // your application needs. This example simply sets an error status in the
33 // real-time model and returns from rt_OneStep.
34 //
35 void rt_OneStep(void);
36 void rt_OneStep(void)
37 {
38     static boolean_T OverrunFlag = false;
39
40     // Disable interrupts here
41
42     // Check for overrun
43     if (OverrunFlag) {
44         rtmSetErrorStatus(rtObj.getRTM(), "Overrun");
45         return;
46     }
47
48     OverrunFlag = true;
49
50     // Save FPU context here (if necessary)
51     // Re-enable timer or interrupt here
52     // Set model inputs here
53
54     // Step the model for base rate
55     rtObj.step();
56
57     // Get model outputs here
58

```

```

59 // Indicate task complete
60 OverrunFlag = false;
61
62 // Disable interrupts here
63 // Restore FPU context here (if necessary)
64 // Enable interrupts here
65 }
66
67 //
68 // The example "main" function illustrates what is required by your
69 // application code to initialize, execute, and terminate the generated code.
70 // Attaching rt_OneStep to a real-time clock is target specific. This example
71 // illustrates how you do this relative to initializing the model.
72 //
73 int T main(int T argc, const char *argv[])
74 {
75 // Unused arguments
76 (void)(argc);
77 (void)(argv);
78
79 // Initialize model
80 rtObj.initialize();
81
82 // Simulating the model step behavior (in non real-time) to
83 // simulate model behavior at stop time.
84
85 while ((rtmGetErrorStatus(rtObj.getRTM()) == (NULL)) && !rtmGetStopRequested
86 (rtObj.getRTM())) {
87 rt_OneStep();
88 }
89
90 // Disable rt_OneStep() here
91 return 0;
92 }
93
94 //
95 // File trailer for generated code.
96 //
97 // [EOF]
98 //
99

```

File: mag2.cpp

```
1 //
2 // File: mag2.cpp
3 //
4 // Code generated for Simulink model 'mag2'.
5 //
6 // Model version : 1.8
7 // Simulink Coder version : 8.13 (R2017b) 24-Jul-2017
8 // C/C++ source code generated on : Wed Oct 16 20:56:56 2019
9 //
10 // Target selection: ert.tlc
11 // Embedded hardware selection: Intel->x86-64 (Windows64)
12 // Code generation objectives:
13 // 1. Execution efficiency
14 // 2. RAM efficiency
15 // Validation result: Not run
16 //
17 #include "mag2.h"
18
19 // Private macros used by the generated code to access rtModel
20 #ifndef rtmIsMajorTimeStep
21 # define rtmIsMajorTimeStep(rtm) (((rtm)->Timing.simTimeStep) ==
    MAJOR_TIME_STEP)
22 #endif
23
24 #ifndef rtmIsMinorTimeStep
25 # define rtmIsMinorTimeStep(rtm) (((rtm)->Timing.simTimeStep) ==
    MINOR_TIME_STEP)
26 #endif
27
28 #ifndef rtmGetTPtr
29 # define rtmGetTPtr(rtm) ((rtm)->Timing.t)
30 #endif
31
32 #ifndef rtmSetTPtr
33 # define rtmSetTPtr(rtm, val) ((rtm)->Timing.t = (val))
34 #endif
35
36 // private model entry point functions
37 extern void mag2_derivatives();
38
39 //
40 // This function updates continuous states using the ODE3 fixed-step
```



```

41 // solver algorithm
42 //
43 void mag2ModelClass::rt_ertODEUpdateContinuousStates(RTWSolverInfo *si )
44 {
45 // Solver Matrices
46 static const real_T rt_ODE3_A[3] = {
47 1.0/2.0, 3.0/4.0, 1.0
48 };
49
50 static const real_T rt_ODE3_B[3][3] = {
51 { 1.0/2.0, 0.0, 0.0 },
52
53 { 0.0, 3.0/4.0, 0.0 },
54
55 { 2.0/9.0, 1.0/3.0, 4.0/9.0 }
56 };
57
58 time_T t = rtsiGetT(si);
59 time_T tnew = rtsiGetSolverStopTime(si);
60 time_T h = rtsiGetStepSize(si);
61 real_T *x = rtsiGetContStates(si);
62 ODE3_IntgData *id = (ODE3_IntgData *)rtsiGetSolverData(si);
63 real_T *y = id->y;
64 real_T *f0 = id->f[0];
65 real_T *f1 = id->f[1];
66 real_T *f2 = id->f[2];
67 real_T hB[3];
68 int_T i;
69 int_T nXc = 4;
70 rtsiSetSimTimeStep(si,MINOR_TIME_STEP);
71
72 // Save the state values at time t in y, we'll use x as ynew.
73 (void) memcpy(y, x,
74 (uint_T)nXc*sizeof(real_T));
75
76 // Assumes that rtsiSetT and ModelOutputs are up-to-date
77 // f0 = f(t,y)
78 rtsiSetdX(si, f0);
79 mag2_derivatives();
80
81 // f(:,2) = feval(odefile, t + hA(1), y + f*hB(:,1), args:)(*);
82 hB[0] = h * rt_ODE3_B[0][0];
83 for (i = 0; i < nXc; i++) {

```

```

84 x[i] = y[i] + (f0[i]*hB[0]);
85 }
86
87 rtsiSetT(si, t + h*rt_ODE3_A[0]);
88 rtsiSetdX(si, f1);
89 this->step();
90 mag2_derivatives();
91
92 //f(:,3) = feval(odefile, t + hA(2), y + f*hB(:,2), args(:)(*));
93 for (i = 0; i <= 1; i++) {
94 hB[i] = h * rt_ODE3_B[1][i];
95 }
96
97 for (i = 0; i < nXc; i++) {
98 x[i] = y[i] + (f0[i]*hB[0] + f1[i]*hB[1]);
99 }
100
101 rtsiSetT(si, t + h*rt_ODE3_A[1]);
102 rtsiSetdX(si, f2);
103 this->step();
104 mag2_derivatives();
105
106 // tnew = t + hA(3);
107 // ynew = y + f*hB(:,3);
108 for (i = 0; i <= 2; i++) {
109 hB[i] = h * rt_ODE3_B[2][i];
110 }
111
112 for (i = 0; i < nXc; i++) {
113 x[i] = y[i] + (f0[i]*hB[0] + f1[i]*hB[1] + f2[i]*hB[2]);
114 }
115
116 rtsiSetT(si, tnew);
117 rtsiSetSimTimeStep(si, MAJOR_TIME_STEP);
118 }
119
120 // Model step function
121 void mag2ModelClass::step()
122 {
123 if (rtmIsMajorTimeStep((&rtM))) {
124 // set solver stop time
125 rtsiSetSolverStopTime((&rtM)->solverInfo, (((&rtM)->Timing.clockTick0+1)*
126 (&rtM)->Timing.stepSize0));

```

```

127 } // end MajorTimeStep
128
129 // Update absolute time of base rate at minor time step
130 if (rtmIsMinorTimeStep((&rtM))) {
131 (&rtM)->Timing.t[0] = rtsiGetT((&rtM)->solverInfo);
132 }
133
134 // Integrator: '<S1>/dfi'
135 rtDW.dfi = rtX.dfi_CSTATE;
136
137 // Product: '<S1>/1//Тм' incorporates:
138 // Constant: '<Root>/Заданное угловое перемещение'
139 // Constant: '<S4>/alfa, betta'
140 // Constant: '<S4>/gamma'
141 // Integrator: '<S1>/fi'
142 // Integrator: '<S2>/r'
143 // Product: '<S4>/r**2'
144 // Product: '<S4>/Умножитель3'
145 // Sum: '<Root>/Sum'
146 // Sum: '<S1>/Sum'
147 // Sum: '<S4>/Sum'
148 // Sum: '<S4>/Sum1'
149
150 rtDW.u = ((1.14 - rtX.fi_CSTATE) - rtDW.dfi) / ((rtX.r_CSTATE - rtX.r_CSTATE *
151 rtX.r_CSTATE) * 0.0412 + 0.0311);
152
153 // Integrator: '<S2>/dr'
154 rtDW.dr = rtX.dr_CSTATE;
155
156 // Sum: '<S2>/Sum1' incorporates:
157 // Constant: '<Root>/Заданное линейное перемещение'
158 // Integrator: '<S2>/r'
159 // Sum: '<S2>/Sum'
160
161 rtDW.Sum1 = (0.52 - rtX.r_CSTATE) - rtDW.dr;
162 if (rtmIsMajorTimeStep((&rtM))) {
163 rt_ertODEUpdateContinuousStates((&rtM)->solverInfo);
164
165 // Update absolute time for base rate
166 // The "clockTick0" counts the number of times the code of this task has
167 // been executed. The absolute time is the multiplication of "clockTick0"
168 // and "Timing.stepSize0". Size of "clockTick0" ensures timer will not
169 // overflow during the application lifespan selected.

```

```

170
171 ++(&rtM)->Timing.clockTick0;
172 (&rtM)->Timing.t[0] = rtsiGetSolverStopTime(&(&rtM)->solverInfo);
173
174 {
175 // Update absolute timer for sample time: [0.3s, 0.0s]
176 // The "clockTick1" counts the number of times the code of this task has
177 // been executed. The resolution of this integer timer is 0.3, which is the step size
178 // of the task. Size of "clockTick1" ensures timer will not overflow during the
179 // application lifespan selected.
180
181 (&rtM)->Timing.clockTick1++;
182 }
183 } // end MajorTimeStep
184 }
185
186 // Derivatives for root system: '<Root>'
187 void mag2ModelClass::mag2_derivatives()
188 {
189 XDot *_rtXdot;
190 _rtXdot = ((XDot *) (&rtM)->derivs);
191
192 // Derivatives for Integrator: '<S1>/fi'
193 _rtXdot->fi_CSTATE = rtDW.dfi;
194
195 // Derivatives for Integrator: '<S1>/dfi'
196 _rtXdot->dfi_CSTATE = rtDW.u;
197
198 // Derivatives for Integrator: '<S2>/r'
199 _rtXdot->r_CSTATE = rtDW.dr;
200
201 // Derivatives for Integrator: '<S2>/dr'
202 _rtXdot->dr_CSTATE = rtDW.Sum1;
203 }
204
205 // Model initialize function
206 void mag2ModelClass::initialize()
207 {
208 // Registration code
209 {
210 // Setup solver object
211 rtsiSetSimTimeStepPtr(&(&rtM)->solverInfo, &(&rtM)->Timing.simTimeStep);
212 rtsiSetTPtr(&(&rtM)->solverInfo, &rtmGetTPtr((&rtM)));

```

```

213 rtsiSetStepSizePtr(&(&rtM)->solverInfo, &(&rtM)->Timing.stepSize0);
214 rtsiSetdXPtr(&(&rtM)->solverInfo, &(&rtM)->derivs);
215 rtsiSetContStatesPtr(&(&rtM)->solverInfo, (real_T **) &(&rtM)->contStates);
216 rtsiSetNumContStatesPtr(&(&rtM)->solverInfo, &(&rtM)->Sizes.numContStates);
217 rtsiSetNumPeriodicContStatesPtr(&(&rtM)->solverInfo, &(&rtM)
218 ->Sizes.numPeriodicContStates);
219 rtsiSetPeriodicContStateIndicesPtr(&(&rtM)->solverInfo, &(&rtM)
220 ->periodicContStateIndices);
221 rtsiSetPeriodicContStateRangesPtr(&(&rtM)->solverInfo, &(&rtM)
222 ->periodicContStateRanges);
223 rtsiSetErrorStatusPtr(&(&rtM)->solverInfo, (&rtmGetErrorStatus((&rtM))));
224 rtsiSetRTModelPtr(&(&rtM)->solverInfo, (&rtM));
225 }
226
227 rtsiSetSimTimeStep(&(&rtM)->solverInfo, MAJOR_TIME_STEP);
228 (&rtM)->intgData.y = (&rtM)->odeY;
229 (&rtM)->intgData.f[0] = (&rtM)->odeF[0];
230 (&rtM)->intgData.f[1] = (&rtM)->odeF[1];
231 (&rtM)->intgData.f[2] = (&rtM)->odeF[2];
232 getRTM()->contStates = ((X *) &rtX);
233 rtsiSetSolverData(&(&rtM)->solverInfo, (void *)&(&rtM)->intgData);
234 rtsiSetSolverName(&(&rtM)->solverInfo, "ode3");
235 rtmSetTPtr(getRTM(), &(&rtM)->Timing.tArray[0]);
236 (&rtM)->Timing.stepSize0 = 0.3;
237
238 // InitializeConditions for Integrator: '<S1>/fi'
239 rtX.fi_CSTATE = 0.0;
240
241 // InitializeConditions for Integrator: '<S1>/dfi'
242 rtX.dfi_CSTATE = 0.0;
243
244 // InitializeConditions for Integrator: '<S2>/r'
245 rtX.r_CSTATE = 0.0;
246
247 // InitializeConditions for Integrator: '<S2>/dr'
248 rtX.dr_CSTATE = 0.0;
249 }
250
251 // Constructor
252 mag2ModelClass::mag2ModelClass()
253 {
254 }
255

```

```

256 // Destructor
257 mag2ModelClass::~mag2ModelClass()
258 {
259 // Currently there is no destructor body generated.
260 }
261
262 // Real-Time Model get method
263 RT_MODEL * mag2ModelClass::getRTM()
264 {
265 return (&rtM);
266 }
267
268 //
269 // File trailer for generated code.
270 //
271 // [EOF]
272 //
273

```

File: mag2.h

```

1 //
2 // File: mag2.h
3 //
4 // Code generated for Simulink model 'mag2'.
5 //
6 // Model version : 1.8
7 // Simulink Coder version : 8.13 (R2017b) 24-Jul-2017
8 // C/C++ source code generated on : Wed Oct 16 20:56:56 2019
9 //
10 // Target selection: ert.tlc
11 // Embedded hardware selection: Intel->x86-64 (Windows64)
12 // Code generation objectives:
13 // 1. Execution efficiency
14 // 2. RAM efficiency
15 // Validation result: Not run
16 //
17 #ifndef RTW_HEADER_mag2_h_
18 #define RTW_HEADER_mag2_h_
19 #include <string.h>
20 #ifndef mag2_COMMON_INCLUDES_
21 # define mag2_COMMON_INCLUDES_

```

```

22 #include "rtwtypes.h"
23 #include "rtw_continuous.h"
24 #include "rtw_solver.h"
25 #endif // mag2_COMMON_INCLUDES_
26
27 // Macros for accessing real-time model data structure
28 #ifndef rtmGetErrorStatus
29 # define rtmGetErrorStatus(rtm) ((rtm)->errorStatus)
30 #endif
31
32 #ifndef rtmSetErrorStatus
33 # define rtmSetErrorStatus(rtm, val) ((rtm)->errorStatus = (val))
34 #endif
35
36 #ifndef rtmGetStopRequested
37 # define rtmGetStopRequested(rtm) ((rtm)->Timing.stopRequestedFlag)
38 #endif
39
40 #ifndef rtmSetStopRequested
41 # define rtmSetStopRequested(rtm, val) ((rtm)->Timing.stopRequestedFlag = (val))
42 #endif
43
44 #ifndef rtmGetStopRequestedPtr
45 # define rtmGetStopRequestedPtr(rtm) (&((rtm)->Timing.stopRequestedFlag))
46 #endif
47
48 #ifndef rtmGetT
49 # define rtmGetT(rtm) (rtmGetTPtr((rtm))[0])
50 #endif
51
52 // Forward declaration for rtModel
53 typedef struct tag_RTM RT_MODEL;
54
55 // Block signals and states (auto storage) for system '<Root>'
56 typedef struct {
57 real_T dfi; // '<S1>/dfi'
58 real_T u; // '<S1>/I//Tm'
59 real_T dr; // '<S2>/dr'
60 real_T Sum1; // '<S2>/Sum1'
61 } DW;
62
63 // Continuous states (auto storage)
64 typedef struct {

```

```

65 real_T fi_CSTATE; // '<S1>/fi'
66 real_T dfi_CSTATE; // '<S1>/dfi'
67 real_T r_CSTATE; // '<S2>/r'
68 real_T dr_CSTATE; // '<S2>/dr'
69 } X;
70
71 // State derivatives (auto storage)
72 typedef struct {
73 real_T fi_CSTATE; // '<S1>/fi'
74 real_T dfi_CSTATE; // '<S1>/dfi'
75 real_T r_CSTATE; // '<S2>/r'
76 real_T dr_CSTATE; // '<S2>/dr'
77 } XDot;
78
79 // State disabled
80 typedef struct {
81 boolean_T fi_CSTATE; // '<S1>/fi'
82 boolean_T dfi_CSTATE; // '<S1>/dfi'
83 boolean_T r_CSTATE; // '<S2>/r'
84 boolean_T dr_CSTATE; // '<S2>/dr'
85 } XDis;
86
87 #ifndef ODE3_INTG
88 #define ODE3_INTG
89
90 // ODE3 Integration Data
91 typedef struct {
92 real_T *y; // output
93 real_T *f[3]; // derivatives
94 } ODE3_IntgData;
95
96 #endif
97
98 // Real-time Model Data Structure
99 struct tag_RTM {
100 const char_T *errorStatus;
101 RTWSolverInfo solverInfo;
102 X *contStates;
103 int_T *periodicContStateIndices;
104 real_T *periodicContStateRanges;
105 real_T *derivs;
106 boolean_T *contStateDisabled;
107 boolean_T zCCacheNeedsReset;

```



```

108 boolean_T derivCacheNeedsReset;
109 boolean_T CTOutputIncnstWithState;
110 real_T odeY[4];
111 real_T odeF[3][4];
112 ODE3_IntgData intgData;
113
114 //
115 // Sizes:
116 // The following substructure contains sizes information
117 // for many of the model attributes such as inputs, outputs,
118 // dwork, sample times, etc.
119
120 struct {
121 int_T numContStates;
122 int_T numPeriodicContStates;
123 int_T numSampTimes;
124 } Sizes;
125
126 //
127 // Timing:
128 // The following substructure contains information regarding
129 // the timing information for the model.
130
131 struct {
132 uint32_T clockTick0;
133 time_T stepSize0;
134 uint32_T clockTick1;
135 SimTimeStep simTimeStep;
136 boolean_T stopRequestedFlag;
137 time_T *t;
138 time_T tArray[2];
139 } Timing;
140 };
141
142 // Class declaration for model mag2
143 class mag2ModelClass {
144 // public data and function members
145 public:
146 // model initialize function
147 void initialize();
148
149 // model step function
150 void step();

```

```

151
152 // Constructor
153 mag2ModelClass();
154
155 // Destructor
156 ~mag2ModelClass();
157
158 // Real-Time Model get method
159 RT_MODEL * getRTM();
160
161 // private data and function members
162 private:
163 // Block signals and states
164 DW rtDW;
165 X rtX; // Block continuous states
166
167 // Real-Time Model
168 RT_MODEL rtM;
169
170 // Continuous states update member function
171 void rt_ertODEUpdateContinuousStates(RTWSolverInfo *si);
172
173 // Derivatives member function
174 void mag2_derivatives();
175 };
176
177 //-
178 // These blocks were eliminated from the model due to optimizations:
179 //
180 // Block '<Root>/Коэффициент усилений' : Unused code path elimination
181 // Block '<Root>/Линейное перемещение' : Unused code path elimination
182 // Block '<Root>/Осциллограф' : Unused code path elimination
183 // Block '<Root>/Угловое перемещение' : Unused code path elimination
184 // Block '<S2>/1//Тм' : Eliminated nontunable gain of 1
185 // Block '<Root>/КПР' : Eliminated due to constant selection input
186 // Block '<Root>/a' : Unused code path elimination
187 // Block '<Root>/b' : Unused code path elimination
188 // Block '<S3>/Sum' : Unused code path elimination
189 // Block '<S3>/Sum1' : Unused code path elimination
190 // Block '<S3>/ar' : Unused code path elimination
191 // Block '<S3>/Умножитель1' : Unused code path elimination
192 // Block '<S3>/Умножитель2' : Unused code path elimination
193 // Block '<S3>/Усилитель2' : Unused code path elimination

```

```

194
195
196 //-
197 // The generated code includes comments that allow you to trace directly
198 // back to the appropriate location in the model. The basic format
199 // is <system>/block_name, where system is the system number (uniquely
200 // assigned by Simulink) and block_name is the name of the block.
201 //
202 // Use the MATLAB hilite_system command to trace the generated code back
203 // to the model. For example,
204 //
205 // hilite_system('<S3>') - opens system 3
206 // hilite_system('<S3>/Kp') - opens and selects block Kp which resides in S3
207 //
208 // Here is the system hierarchy for this model
209 //
210 // '<Root>': 'mag2'
211 // '<S1>': 'mag2/1-й канал управлений'
212 // '<S2>': 'mag2/2-й канал управлений'
213 // '<S3>': 'mag2/Konm'
214 // '<S4>': 'mag2/Tm'
215
216 #endif // RTW_HEADER_mag2_h_
217
218 //
219 // File trailer for generated code.
220 //
221 // [EOF]
222 //
223

```

File: rtwtypes.h

```

1 //
2 // File: rtwtypes.h
3 //
4 // Code generated for Simulink model 'mag2'.
5 //
6 // Model version : 1.8
7 // Simulink Coder version : 8.13 (R2017b) 24-Jul-2017
8 // C/C++ source code generated on : Wed Oct 16 20:56:56 2019
9 //

```

```

10 // Target selection: ert.tlc
11 // Embedded hardware selection: Intel->x86-64 (Windows64)
12 // Code generation objectives:
13 // 1. Execution efficiency
14 // 2. RAM efficiency
15 // Validation result: Not run
16 //
17
18 #ifndef RTWTYPES_H
19 #define RTWTYPES_H
20
21 // Logical type definitions
22 #if (!defined(__cplusplus))
23 # ifndef false
24 # define false (0U)
25 # endif
26
27 # ifndef true
28 # define true (1U)
29 # endif
30 #endif
31
32 //=====
33 //=====*
34 // Target hardware information
35 // Device type: Intel->x86-64 (Windows64)
36 // Number of bits: char: 8 short: 16 int: 32
37 // long: 32 long long: 64
38 // native word size: 64
39 // Byte ordering: LittleEndian
40 // Signed integer division rounds to: Zero
41 // Shift right on a signed integer as arithmetic shift: on
42 //
43 //=====
44 //=====
45 //=====*
46 // Fixed width word size data types: *
47 // int8_T, int16_T, int32_T - signed 8, 16, or 32 bit integers *
48 // uint8_T, uint16_T, uint32_T - unsigned 8, 16, or 32 bit integers *
49 // real32_T, real64_T - 32 and 64 bit floating point numbers *
50 //
51 //=====

```

```

=====
49 typedef signed char int8_T;
50 typedef unsigned char uint8_T;
51 typedef short int16_T;
52 typedef unsigned short uint16_T;
53 typedef int int32_T;
54 typedef unsigned int uint32_T;
55 typedef long long int64_T;
56 typedef unsigned long long uint64_T;
57 typedef float real32_T;
58 typedef double real64_T;
59
60 //=====
    =====*
61 // Generic type definitions: boolean_T, char_T, byte_T, int_T, uint_T, *
62 // real_T, time_T, ulong_T, ulonglong_T. *
63 //
    =====
    =====
64 typedef double real_T;
65 typedef double time_T;
66 typedef unsigned char boolean_T;
67 typedef int int_T;
68 typedef unsigned int uint_T;
69 typedef unsigned long ulong_T;
70 typedef unsigned long long ulonglong_T;
71 typedef char char_T;
72 typedef unsigned char uchar_T;
73 typedef char_T byte_T;
74
75 //=====
    =====*
76 // Min and Max: *
77 // int8_T, int16_T, int32_T - signed 8, 16, or 32 bit integers *
78 // uint8_T, uint16_T, uint32_T - unsigned 8, 16, or 32 bit integers *
79 //
    =====
    =====
80 #define MAX_int8_T ((int8_T)(127))
81 #define MIN_int8_T ((int8_T)(-128))
82 #define MAX_uint8_T ((uint8_T)(255U))
83 #define MAX_int16_T ((int16_T)(32767))
84 #define MIN_int16_T ((int16_T)(-32768))
85 #define MAX_uint16_T ((uint16_T)(65535U))

```

```
86 #define MAX_int32_T ((int32_T)(2147483647))
87 #define MIN_int32_T ((int32_T)(-2147483647-1))
88 #define MAX_uint32_T ((uint32_T)(0xFFFFFFFFU))
89 #define MAX_int64_T ((int64_T)(9223372036854775807LL))
90 #define MIN_int64_T ((int64_T)(-9223372036854775807LL-1LL))
91 #define MAX_uint64_T ((uint64_T)(0xFFFFFFFFFFFFFFFFULL))
92
93 // Block D-Work pointer type
94 typedef void * pointer_T;
95
96 #endif // RTWTYPES_H
97
98 //
99 // File trailer for generated code.
100 //
101 // [EOF]
102 //
103 //
104 //
```

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Инженерно-строительный институт
Кафедра: «Строительные материалы и технологии строительства»

УТВЕРЖДАЮ
Заведующий кафедрой
Енджиевская И.Г.
подпись, инициалы, фамилия
« 09 » 07 2020 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

**«Совершенствование рабочего процесса строительного работа при
монтаже строительных конструкций»**

08.04.01 «Строительство»
08.04.01.07 «Комплексная механизация и автоматизация строительства»

Научный
руководитель


подпись

канд. техн. наук, доцент
ученая степень

А.П Прокопьев
И.О. Фамилия

Выпускник

подпись

СФ18-07М,
номер гр., зачетной книжки

Н.Ю. Андреев
И.О. Фамилия

Рецензент


подпись

докт. техн. наук, проф.
ученая степень

В.И. Иванчура
И.О. Фамилия

Нормоконтролер


подпись

канд. техн. наук, доцент
ученая степень

Е.С. Турышева
И.О. Фамилия

Красноярск 2020