**Кыргызский государственный Технический Университет им. И. Раззакова**

На правах рукописи

УДК: 681.5.013: 625.031 (043.3)

**Кадыркулова Кыял Кудайбердиевна**

**Синтез СИСТЕМЫ управления траекторным движением объекта**

Специальность 05.13.06 – Автоматизация и управление технологическими процессами и производствами

Диссертация на соискание учёной степени кандидата наук

**Научный руководитель:**

доктор технических наук, профессор

**Батырканов Жеңиш Исакунович**

Бишкек 2019

**ОГЛАВЛЕНИЕ**

**ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ** …………………………………..4

**ВВЕДЕНИЕ** …………………………………………………………………………5

**ГЛАВА 1.**  **АНАЛИЗ ПРОБЛЕМЫ СИНТЕЗА ЗАКОНОВ УПРАВЛЕНИЯ ПО ОСУЩЕСТВЛЕНИЮ ТРАЕКТОРНОГО ДВИЖЕНИЯ ОБЪЕКТА** ...11

* 1. Анализ современного состояния проблемы управления траекторным движением объекта …………………………………………………………11
  2. Постановка задачи синтеза управлений по осуществлению движения управляемого объекта по предписанной траектории ……………………16
     1. Математические модели управляемых объектов …………………….…..16
     2. Предписанные траектории движения ……………………………………..17
     3. Задачи синтеза ………………………………………………………………18

**ВЫВОДЫ** …………………………………………………………………...19

**ГЛАВА 2.** **СИНТЕЗ ЗАКОНОВ УПРАВЛЕНИЯ ПО ОСУЩЕСТВЛЕНИЮ ДВИЖЕНИЯ ОБЪЕКТА ПО ПРЕДПИСАННОЙ ТРАЕКТОРИИ** …….….20

* 1. Подход синтеза законов управления в случаях задания предписанной траектории движения в аналитической форме …………………………...20
  2. Подход синтеза законов управления в случаях задания предписанной траектории движения в табличной форме …………………….…………..47
  3. Адаптивное управление траекторным движением …………………..…..53

**ВЫВОДЫ** ……………………………………………………………….…..79

**ГЛАВА 3. ПРАКТИЧЕСКИЕ ПРИЛОЖЕНИЯ ТЕОРЕТИЧЕСКИХ РЕЗУЛЬТАТОВ ИСЛЕДОВАНИИ** …………………………………………….80

* 1. Исследование и разработка системы управления траекторным движением трёхзвенного манипулятора ………………………………………..………80
  2. Разработка и исследование системы управления приводами 3D-принтера ………………………………………………………………………………...87
  3. Методика синтеза управляющих законов для шаговых двигателей 3D-принтера ……………………………………………………………………..89
  4. Некоторые вопросы технической реализации системы управления шаговыми приводами прототипа 3 D- принтера …………………….…...95
     1. Функциональная схема системы управления шаговым приводом ………95
     2. Техническая реализация на основе микроконтроллера Arduino ……….100
  5. Программное обеспечение системы управления 3D—принтера ……….110
     1. Руководство по эксплуатации 3D – принтера…………………………….110

**ВЫВОДЫ** ………………………………………………………………….117

**ЗАКЛЮЧЕНИЕ** …………………………………………………………..118

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** ………………….119

**ПРИЛОЖЕНИЯ** . ………………………………………………..……….131

**ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ**

АСУ – Автоматизированная система управления

АСУ ТП – Автоматизированная система управления технологическим процессом

CAD – Система автоматизированного проектирования

LPT – Line Print Terminal (порт принтера, параллельный порт)

ДОС – Датчик обратной связи

ИС – Информационная система

ММ – Математическая модель

ОС – Операционная система

ОУ – Объект управления

ПО – Программное обеспечение

ППП – Пакет прикладных программ

РО – Регулирующий орган

ШД – Шаговый двигатель

**ВВЕДЕНИЕ**

На сегодняшний день во многих областях автоматизации технологических процессов ставится задача управления различными механизмами по заданным предписанным программам движения.

Это, в первую очередь, касается управления робототехническим устройствам, когда рабочие органы (рука, захватывающие устройства) промышленных манипуляторов должны отрабатывать заданные предписанные траектории движения.

Эффективность любого производства на современном этапе развития немыслима без перехода к комплексной автоматизации.

Эффективным средством решения многих задач комплексной автоматизации являются построение автоматизации на основе робототехнических комплексов. Применение робототехнических комплексов очень актуально и перспективно в условиях перехода к рыночной экономике, что характерно для Кыргызстана.

В условиях рыночной экономики осуществляется быстрый переход от выпуска одной номенклатуры изделий к другим, при этом осуществляется мелкосерийное и даже единичное производство, такие технологии организации производства появились благодаря применению роботехнических комплексов.

В последнее время бурно развиваются и внедряются в различные сферы машиностроения различные 3D – технологии.

В машиностроении до 3D – технологий, применялась классическая, так называемая субстрактивная технология изготовления деталей, в отличие от классической технологии в 3D – технологиях используется аддитивная технология изготовления деталей.

Прогресс в области микроэлектроники, вычислительной техники, программирования, математического моделирования обеспечили возможность организации производства и технологических процессов на новых принципах в виде гибких производственных систем, на основе робототехнических комплексов и широкого применения 3D – технологий.

На сегодняшний день отсутствуют универсальные и в то же время эффективные методы синтеза управления по осуществлению движения управляемых объектов по произвольно заданным предписанным траекториям движения. К настоящему времени более или менее решены задачи осуществления предписанных движений, когда рассматривается линейный объект управления и когда предписанная траектория движения задаётся (описывается) в аналитической форме.

В большинстве случаев на практике, как случаях построения систем управления с промышленными роботами, так и в случаях применения 3D – принтеров предписанную траекторию движения описать аналитически очень сложно.

Это связано с тем, что форму изготавливаемых деталей в 3D – технологиях в подавляющем большинстве случаев аналитически описать невозможно. В робототехнике, когда применяется роботы последних поколений (контурно - адаптивные, интеллектуальные) описать аналитически траекторию движения руки манипулятора, также очень трудно.

Поэтому, стоит очень **актуальная как в практическом, так и в теоретическом аспекте** разработка новых эффективных методов синтеза управлений по осуществлению движений объектов по заранее заданной предписанной траектории движения.

В связи с вышесказанным, в данном диссертационном исследовании разрабатывается новый подход синтеза законов управлений по осуществлению движений объектов по заданным предписанным траекториям движения.

**Цель и задачи исследования.**

Целью диссертационной работы является разработка нового подхода синтеза управлений осуществляющих движение объекта по заданной предписанной траектории, на основе применения концепции обратных задач динамики.

**Задачи исследований:**

* анализ современного состояния проблемы управления траекторным движением объекта;
* разработка метода синтеза управления траекторным движением, в случае, когда предписанная траектория задаётся в аналитической форме;
* разработка метода синтеза управления траекторным движением, в случае, когда предписанная траектория задаётся в табличной форме;
* разработка математических моделей приводов для промышленных манипуляторов и 3D – принтеров.

**Новизной научных результатов является:**

* расширение и усовершенствование ранее известных методик синтеза управления траекторным движением в случае, когда предписанная траектория задаётся аналитически;
* впервые, совместно с научным руководителем Ж.И. Батыркановым, предложена новая методика синтеза управлений, когда предписанная траектория движения задаётся в табличной форме;
* получена новая математическая модель шагового привода 3D – принтера.

**Практическая значимость полученных научных результатов заключается в следующем:**

* разработанные методики синтеза управлений осуществляющих движение объекта по заданной траектории в достаточной степени формализованы и позволяют эффективно, конструктивно определять искомые законы управления;
* разработанные методики синтеза позволяют проводить процедуру синтеза по таблично заданным предписанным траекториям движения, когда предписанную траекторию невозможно описать аналитически;
* разработанные методики синтеза очень успешно работают, в случаях, когда использование известных методик синтеза затруднительно, это в первую очередь касается задач синтеза управлений для приводов промышленных роботов и 3D – принтеров. В связи с тем, что практическое применение ­3D – принтеров, только начинается, разработанные методики синтеза практически очень значимы.

**Основные положения диссертации, выносимые на защиту:**

* подход синтеза управлений, осуществляющих движение объекта по предписанной траектории движения;
* подход синтеза управлений в случае задания предписанной траектории движения в табличной форме;
* разработанная математическая модель шагового привода 3D–принтера, как объекта управления в задачах осуществления траекторного движения.
* программное и техническое обеспечение системы управления экспериментального 3D – принтера.

**Личный вклад соискателя:**

Все научно-технические результаты работы получены диссертантом под руководством научного руководителя.

В опубликованных работах, в соавторстве с научным руководителем, постановка задач и общий подход исследований принадлежит научному руководителю, конкретная процедура синтеза и моделирования принадлежит диссертанту.

**Апробация результатов диссертации.**

Результаты диссертационной работы докладывались на следующих международных симпозиумах, республиканских, межвузовских конференциях:

* международной конференции «Актуальные проблемы теории управления, топологии и операторных уравнений», Чолпон-Ата, 2013 г.;
* научно - технической конференции КСТУ, 2014 г.;
* международной научно-технической конференции молодых учёных, аспирантов и студентов «Молодёжь в инновационных исследованиях, Бишкек 2016 г.;
* международной телеконференции, 2018 г. (МГУ им. Н.П. Огарева, МЭИ и КГТУ им. И. Раззакова).
* в ежегодных научно-технических конференциях молодых учёных аспирантов и студентов КГТУ им. И. Раззакова, 2007-2017 гг.

**Внедрение результатов диссертационных исследований.**

1. Результаты диссертационных исследований использовались в грантовых научных проектах МО и Н КР (2011, 2012, 2013, 2014, 2015, 2016).
2. Результаты диссертационных исследований внедрены в учебный процесс КГТУ им. И. Раззакова по дисциплинам: «Основы робототехники и электропривода», «Теория автоматического управления», часть II.

**Полнота отражения результатов диссертации в публикациях.**

Основные научные результаты диссертации опубликованы в научных периодических изданиях Кыргызстана (Известия КГТУ им. И.Раззакова), Казакстана (Вестник науки Костанайского социально - технического Университета), России (Прикаспийский научно-технический журнал: «Управление и высокие технологии, Известия науки Мордовского государственного университета им. Н.П. Огарёва»).

**Структура и объем диссертации.**

Диссертация состоит из введения, трёх глав, приложения, и списка литературы. Объем диссертации составляет 176 страниц машинописного текста, включая 37 рисунка и 4 таблиц, список литературы, содержащий 115 наименований печатных изданий и 15 наименований электронных источников информации.

В первой главе диссертации анализируются существующие подходы и методы синтеза законов управления по осуществлению траекторного движения объекта. Отмечаются достоинства и недостатки, области целесообразного применения рассматриваемых методов.

Во второй главе описываются подходы синтеза управления в случае задания предписанной траектории, как в аналитической, так и в табличной форме.

Разработанные подходы синтеза рассмотрены на модельных примерах. Эти подходы позволяют эффективно проводить процедуру синтеза для достаточно широкого класса управляемых объектов.

В третьей главе рассматривается применение разработанной теории синтеза управления траекторным движением объекта для некоторых прикладных задач управления. Разработанная теория применяется: для построения системы управления движением рабочих органов 3х-звенного манипулятора по заданной траектории; для построения системы управления шаговыми приводами 3D-принтера.

Диссертант выражает искреннюю признательность научному руководителю профессору Батырканову Ж.И. за оказанную помощь при выполнении работы.

**ГЛАВА 1. АНАЛИЗ ПРОБЛЕМЫ СИНТЕЗА ЗАКОНОВ УПРАВЛЕНИЯ ПО ОСУЩЕСТВЛЕНИЮ ТРАЕКТОРНОГО ДВИЖЕНИЯ ОБЪЕКТА**

В этой главе проводится анализ существующих подходов и методов синтеза законов управления по осуществлению траекторного движения объекта. Отмечаются достоинства и недостатки, области целесообразного применения рассматриваемых методов.

В конце главы приведена постановка задачи синтеза управлений траекторным движением объекта.

* 1. **Анализ современного состояния проблемы управления траекторным движением объекта**

В классической теории управления рассматривается целый ряд задач:

* задачи стабилизации;
* задачи слежения;
* задачи программного регулирования;
* задачи оптимального управления;
* задачи адаптивного управления;
* задачи робастного управления и другие.

Для решения этих задач разработаны самые разнообразные подходы и методы решения этих задач.

К ним относятся: метод структурных схем и передаточных функций, частотные методы, методы модального управления, методы анализа и синтеза, на основе функций Ляпунова, методы оптимального и адаптивного управления, подходы на основе спектрального анализа и многие другие [2, 3, 13, 24, 33, 41, 59, 60, 70, 77, 108, 111].

В отличие от классических задач управления, до настоящего времени, задачи осуществления траекторного движения управляемых объектов были достаточно слабо разработаны. Но в последнее время, в связи с потребностями практики, резко возрос интерес к решению задач по осуществлению траекторного движения управляемых объектов. Это в первую очередь связано с решением задач синтеза законов управления манипуляторами промышленных роботов. Во-вторых, это касается проблем управления рабочими органами 3D- принтеров. Кроме этих задач, это задачи связанные с управлением летательных объектов.

В теории управления разработано большое количество подходов и методов построения и создания систем управления. Анализируя все эти подходы и методы известный ученый П.Д. Крутько [55, 56] заметил, что в основе каждого метода прямо или косвенно заложены концепция обратных задач динамики. Действительно, во всех этих методах прямо или косвенно используется эталонная система, с помощью которой задаётся предписанное движение, которое требуется реализовать синтезируемыми законами управления.

Данный вывод имеет важное практическое значения для построения принципиально новых и эффективных приёмов синтеза соответствующих законов управления.

Для задач управления П.Д. Крутько предложил следующую формулировку обратной задачи динамики: обратными задачами динамики управляющих систем называются задачи определения законов управления движением динамических систем и их параметров из условия осуществления движения по предписанной траектории [55].

Если коснуться истории, обратные задачи динамики впервые были сформулированы и решены в аналитической механике И. Ньютоном, А. Пуанкаре, Ж. Лагранжем и другими [27]. Далее, концепция обратных задач были применены к управляемому движению. Это относятся к задачам, которые рассматривались в работах И.В. Мещерского [68].

Позднее, в конце 50-х годов прошлого столетия концепцию обратных задач динамики применил советский учёный Е.А. Барбашин для осуществления движения управляемого объекта по заданной временной программе [4]. Полученные результаты Е.А. Барбашина имеют существенный недостаток, заключающийся в том, что синтезируется разомкнутая, а не замкнутая система управления.

Н.П. Еругин, А.С. Галиулин, Д. Фурасов [26, 27, 28, 41] и ихние последователи рассматривали задачу построения множества управлений, когда движения происходит по заданной траектории. Недостатком подхода, которые предлагались Н.П. Еругиным, А.С. Галиулиным, Д. Фурасовым является неоднозначность возникающаяся в выборе управляющих функции.

Достаточно интересные результаты получены Л.М. Бойчуком [21]. Здесь предлагается подход по структурному синтезу программного движения. В подходе Л.М. Бойчука производится исключение высшей производной управляемой переменной из дифференциального уравнения программы движения и подстановки его в математическую модель управляемого объекта. Подход, предложенный Л.М. Бойчуком, ограничивается классом одномерных управляемых систем.

С.В. Емельянов и его ученики [37, 39] предлагают подход синтеза, при котором траектория движения объекта следует за траекторией движения системы сравнения и обладает свойством асимптотической устойчивости. Такие системы они назвали бинарными.

Очень интересные и достаточно эффективные процедуры синтеза законов управления по осуществлению движения управляемого объекта по заданной траектории получены П.Д. Крутько и его последователям [55, 59, 60].

В случае параметрического задания предписанной траектории движения закон управления сперва, определяется программная функция управления далее строится закон управления на основе обратной связи по переменным состояния.

В случае задания предписанной траектории в виде решения соответствующего дифференциального уравнения искомый закон управления определяется путем приравнивая старших производных уравнений объекта и дифференциального уравнения предписанной траектории.

Узким местом в подходе П. Д. Крутько является то, что эта процедура хорошо работает только для одномерных систем.

Для многомерных систем, процедура синтеза строится по схеме последовательных приближений.

Достаточно эффективные конструктивные методы синтеза управлений траекторным движением предложены Ж.И. Батыркановым [6, 7, 8, 9]. Подходы синтеза предложены, как для случаев, когда предписанная траектория движения управляемого объекта задана в аналитической форме, так и для случаев когда, предписанная траектория задаётся в табличной форме. Последний случай, особо ценен, когда рассматриваются задачи управления рабочими органами промышленных манипуляторов и 3D – принтеров.

Определённые хорошие результаты получены в исследованиях А. П. Крищенко [54].

А. П. Крищенко для решения задачи нахождения соответствующих законов управления предлагает процедуру синтеза, когда математическая модель управляемого объекта записывается в переменных вход-выход, затем осуществляется преобразование аффинных систем, чтобы математическая модель имела каноническую структуру. Далее из канонической структуры определяется искомый закон управления. Узким местом в этом подходе является то, что не всегда можно легко перейти к канонической структуре.

Интересные результаты получены в исследованиях И. В. Мирошника, А. Н. Шалаева по управлению траекторным движением автономных роботов [70].

Но здесь полученные результаты имеют ограниченную область применения, это связана с тем, что авторы рассматривают только узкий класс роботов так называемых автономных роботов, кроме того предписанная траектория задаётся только на плоскости.

Определённые достаточно интересные результаты получены в исследованиях Кыргызстанских учёных: Ж.Ш. Шаршеналиева, В.П. Живоглядова; Т.Т. Оморова, У.Н. Бримкулова, Р.О. Оморова, Е.Л. Миркина и других [42, 43, 69, 72].

Интересные подходы и результаты получены Казахстанскими учеными: М.Ф. Баймухамедовым, Д.Ж. Сыздыковым, Б.Х. Айтчановым, М.А. Бейсенби и другими [1, 7, 17, 85, 99].

Выше приведённый анализ современного состояния проблемы синтеза законов управления, по осуществлению движения объекта по заданной предписанной траектории позволяет сделать следующее выводы:

* решение проблемы синтеза на данное время до конца не решены, это связано с трудностями при рассмотрении нелинейных и многомерных систем;
* процедуры синтеза на основе применения концепции обратных задач динамики позволяют задавать желаемые процессы, движения в прямом неформальном виде;
* рассмотренные немногочисленные методы синтеза основанные на концепциях обратных задач динамики, в основном, ограничены классом одномерных систем;
* в подавляющем большинстве случаев существующие методы синтеза работают только в случаях, когда предписанные траектории движения задаются в аналитической форме.
* к сегодняшнему дню во многих прикладных задач, как управление рабочими органами промышленных манипуляторов, рабочими органами 3D – принтеров, управление движением крылатых ракет и в других ставятся задачи осуществления движения по заданным предписанным траекториям.

Для вышеотмеченных задач, достаточно очевидно, решение задачи синтеза более удобно вести на основе концепции обратных задач динамики. В этом случае структура и параметры алгоритмов управления вполне определяются структурами и параметрами математической модели управляемого объекта и предписанной траектории движения.

В силу вышеперечисленных проблем синтеза, а также широким и глубоким использование промышленных роботов и 3D - принтеров для различных сфер практики, все это приводит к постановке **актуальной,** как в теоретическом, так и практических аспектах задачи разработки конструктивных методов синтеза законов управления по осуществлению движения по заданной предписанной траектории движения с позиций обратных задач динамики.

* 1. **Постановка задачи синтеза управлений по осуществлению движения управляемого объекта по предписанной траектории**
     1. **Математические модели управляемых объектов**

В данной диссертационной работе осуществляется решение задачи как для классов линейных объектов управления, так и для отдельных классов нелинейных объектов:

, (1.1)

(1.2)

) (1.3)

Здесь: x вектор переменных состояния;

 – числовые матрицы соответствующих размерностей;

– векторная функция;

– матричная функция;

– нелинейная непрерывная векторная функция; символ текущего времени;

символ времени в –й дискретный момент времени;

– значение вектора состояния в – тый дискретный момент времени.

* + 1. **Предписанные траектории движения**

Цель управления, в данной работе, заключается в осуществлении движения управляемого объекта по предписанной траектории.

В данной работе предписанные траектории задаются в следующих формах.

1. В параметрической

, (1.4)

где заданные функции времени

1. В форме функциональных уравнений

(1.5)

1. В табличной форме. Таблица задаёт контрольные точки, через которые проходит объект в процессе движения.
   * 1. **Задачи синтеза**

Для заданных классов управляемых объектов, в данной диссертации решаются задачи осуществления движения управляемых объектов, в случаях:

* параметрического задания предписанной траектории;
* задания предписанной траектории в виде функциональных уравнений;
* задания предписанной траектории в табличной форме.

**ВЫВОДЫ**

Анализ существующих методов синтеза позволяет сделать следующие выводы:

* применение существующих как традиционных, так и не многочисленных методов синтеза, основанных на концепциях обратных задач динамики, ограничиваются в основном, классом одномерных систем и линейных систем;
* многие современные прикладные задачи приводят к постановке нетрадиционных задач управления таких как задачи траекторного управления;
* важной особенностью методов синтеза с позиций обратных задач динамики это возможность решать многие задачи синтеза, в таких практически значимых и перспективных областях, где применяются промышленные роботы и 3D – принтеры.

Из вышеуказанных моментов, однозначно следует, что разработка новых, конструктивных подходов синтеза по осуществлению движения управляемого объекта по предписанной траектории - является **практически значимой и актуальной** в теоретическом аспекте задачей.

**ГЛАВА 2. СИНТЕЗ ЗАКОНОВ УПРАВЛЕНИЯ ПО ОСУЩЕСТВЛЕНИЮ ДВИЖЕНИЯ ОБЪЕКТА ПО ПРЕДПИСАННОЙ ТРАЕКТОРИИ**

На сегодняшний день проблема управления движением объекта по предписанной траектории (программе) более или менее решена для линейных одномерных объектов в случае, когда предписанная траектория задаётся в аналитической форме. Когда предписанную траекторию нельзя описать аналитически, то возникают большие затруднения. В данной главе описываются подходы синтеза управления в случае задания предписанной траектории, как в аналитической, так и в табличной форме. Суть предлагаемых подходов синтеза заключается, во-первых во введении в рассмотрении функции отклонения фактической траектории движения от предписанной траектории; во-вторых рассматривается дифференциальное уравнение динамики переходного процесса для отклонения. Затем, назначая для управляемого объекта заданную динамику переходного процесса для отклонения на траекториях движения, мы находим алгебраические соотношения для отыскания необходимых законов управления. При этом, используется аппарат пространства со скалярным произведением и один из результатов Р. Беллмана, относящийся к линейной алгебре [18]. В итоге, получаем новые конструктивные, эффективные процедуры синтеза искомых законов управления.

**2.1. Подход синтеза законов управления в случаях задания предписанной траектории движения в аналитической форме**

Задачи осуществления движения управляемого объекта по предписанной программе (траектории) относятся к неклассическим задачам автоматического управления и на сегодняшний день полная теория анализа и синтеза таких систем отсутствует. Однако практика ставит множество задач, которые относятся к задачам обеспечения движения управляемых объектов по предписанным программам. Это задачи траекторного управления летательными аппаратами; лазерным лучом; рабочими органами роботов; управления рабочими органами, механизмами в системах, использующих 3D-технологии.

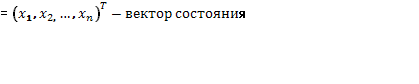
На сегодняшний день в случае, когда предписанная траектория движения объекта описывается аналитически, существуют различные методы, подходы к решению таких задач. Это, в частности, подходы, предложенные В.Д. Фурасовым, П.Д. Крутько, Л.М. Бойчуком и другими авторами [12, 21, 26, 27, 41, 55, 91]. Однако они позволяют решать задачи только для определённых классов управляемых объектов – в основном для линейных одномерных систем.

В данной диссертационной работе разрабатывается подход синтеза законов управления по осуществлению движения объектов по произвольным предписанным траекториям: сначала для траекторий, задаваемых в аналитической форме, затем – в табличной. Эти подходы позволяют эффективно проводить процедуру синтеза для достаточно широкого класса управляемых объектов.

Предлагаемый подход синтеза сначала рассмотрим для случая, когда предписанная траектория задаётся в виде одного алгебраического соотношения, затем для случая, когда используется несколько алгебраических соотношений.

**Пусть математическая модель управляемого объекта представлена в стандартной векторно-дифференциальной форме**

**,**  (2.1)

где − вектор состояния*,*

− вектор управления,

*T* − символ транспонирования.

Предписанная программа движения объекта описывается в виде аналитического соотношения

 (2.2)

Для отыскания управления рассматривается полная производная по времени функции  на движениях системы. В соответствии с (2.2) имеем

 (2.3)

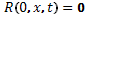
С учётом уравнений движения системы имеем

 (2.4)

Разрешая это соотношение относительно искомой функции , находим искомый закон управления, при котором движение объекта осуществляется по предписанной траектории. Однако с практической точки зрения задача синтеза будет решена в том случае, если закон управления обеспечивает также возвращение фазовой траектории , объекта, на предписанную траекторию  если по каким-либо причинам она окажется вне нее. Другими словами закон управления должен обеспечивать не только процесс движения по предписанной траектории, но и устойчивость требуемого движения при возмущениях. Для такого закона управления, очевидно, должно выполняться соотношение

 (2.5)

где - произвольная функция, обращающаяся в нуль на кривой (2.2), т.е.

 (2.6)

Действительно, если точка  окажется вне предписанной траектории (2.2), то  

Поэтому  представляет собой отклонение фактической траектории от предписанной, а переходный процесс для отклонения описывается уравнением

  (2.7)

Из постановки задачи синтеза закона управления непосредственно вытекает, что должно выполняться условие  при . Это условие накладывает определённые требования на выбор произвольной функции . Например, функция  подбирается из условия заданного времени отработки начального рассогласования  и условий физической реализуемости. На этих моментах сейчас не будем останавливаться.

Таким образом, искомый закон управления с учётом (2.5) определяется из соотношения

 (2.8)

Для определения искомого уравнения из этого соотношения, представим (2.8) в пространстве со скалярным произведением, в следующем равносильном виде:

 (2.9)

где ( . , .) − символ скалярного произведения.

**В частности, для линейного объекта**

 (2.10)

Основное соотношение (2.9) в данном случае представится как



Отсюда

 (2.11)

Для определения искомого управления из полученного скалярного уравнения используем результат Р. Беллмана из линейной алгебры в работе, где решается задача нахождения вектора, лежащего в заданной гиперплоскости

 (2.12)

и имеющего минимальную норму .

Решение данной задачи определяется выражением

 (2.13)

Используя теперь аналогию между выражениями (2.11) и (2.12) на основании (2.13), искомый закон управления определяется в виде

 (2.14)

где функция   определяется выражением

 (2.15)

При этом, синтезированный закон управления обладает свойством 

В конце данного пункта сделаем следующее замечание. Закон управления существует только для тех требуемых траекторий, для которых вектор **** не является нулевым. Это утверждение вытекает из соотношения (2.11).

**Модельный пример синтеза закона управления при описании траектории в аналитической форме**

Пусть управляемое движение описывается уравнением

 (2.16)

Требуется построить управляющую функцию исходя из условия, чтобы переменная  изменялась по гармоническому закону  где амплитуда и  − частота. При гармонических колебаниях между переменными  и  существует зависимость

 (2.17)

Для решения задачи уравнение (2.16) представим в виде системы



или в стандартном векторно-матричном виде

 (2.18)

где

 , 

Предписанная траектория в новых обозначениях принимает вид

 (2.19)

Произвольную функцию выберем исходя из следующих соображений. Для нашей задачи отклонение фактической траектории от предписанной, т.е. ошибка выполнение заданной программы движения имеет вид



При этом, как мы показали ранее, переходный процесс для отклонения определяется уравнением



Потребуем, чтобы процесс отработки начального отклонения происходил за заданное время и по заданной кривой. Такое требование можно удовлетворить, если выберем функцию в виде

 (2.20)

где коэффициент подберём из условий физической реализуемости.

Для нахождения искомого закона управления, согласно выражения (2.14), предварительно определим:

;

;



С учётом найденных выражений искомый закон управления, согласно (2.14), определяется в виде

 (2.21)

где 

Перепишем выражение (2.21) в следующем виде:



или в исходных обозначениях переменных

 (2.22)

Уравнение замкнутой системы с учётом (2.16) и (2.22) определяется уравнением

 (2.23)

Если точка  находится на кривой , то отклонение (ошибка, рассогласование) .

В этом случае из (2.23) следует известное уравнение гармонических колебаний

.

Если же точка  не скользит по кривой  то управляющая функция имеет составляющую которая стремится вывести систему на заданную траекторию движения.



Вычисление величины связано с необходимостью деления на . При наличии колебаний переменная  может принимать нулевые значения. Поэтому реализация закона управления (2.22) связана с известными затруднениями.

Чтобы их избежать, подбор коэффициента  осуществим в виде

  (2.24)

Тогда выражение (2.22) для закона управления представится как

 (2.25)

где 



Реализация полученного закона (2.25) уже не вызывает никаких затруднений. Выбор коэффициента  в виде (2.24) мы осуществили исходя из следующих соображений. Переходный процесс отработки ошибки выполнения предписанной программы, как было показано выше, выражается в виде (2.7) и при этом требуется выполнение условия  при ∞. Данному требованию удовлетворяет, например, функция





которую мы выбрали выше. Действительно, при этом имеем

 (2.26)

Применяя к уравнения (2.26) метод функций Ляпунова с функцией Ляпунова вида   имеем  Отсюда вытекает выполнение условия  при ∞.

Рассмотренный выше пример показывает, что процедура синтеза не вызывает никаких затруднений и единственный произвол при этом, заключается в выборе функции . Такой произвол частично устраняется путём подбора этой функции из условий заданной динамики отработки ошибки выполнения предписанной программы движения и условий физической реализуемости.

На рис. 2.1 и 2.2 показаны результаты моделирования синтезированной замкнутой системы управления на ПЭВМ с помощью ППП MATLAB. При этом на этих рисунках пунктиром показаны возмущённые движения, которые затем переходят в предписанные требуемые движения.

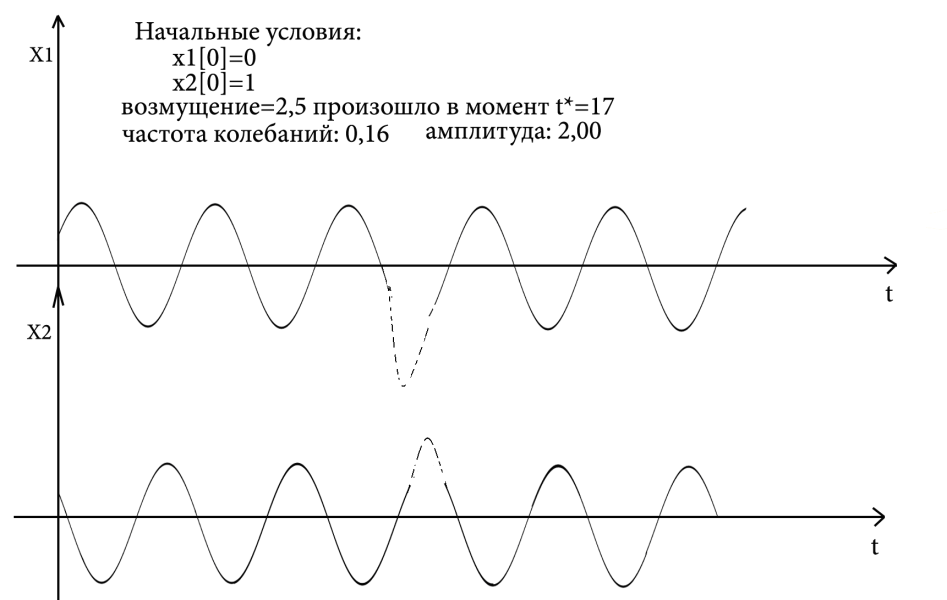


Рис. 2.1. Предписанные движения со стабилизацией

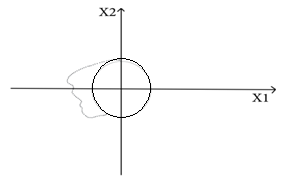


Рис. 2.2. Фазовый портрет синтезированной системы

Из этих рисунков видно, что независимо от начальных условий движение объекта выходит на предписанную траектории движения, т.е. на гармонический закон колебаний. Кроме того, показана, об устойчивости движения на этих предписанных траекториях движения. В момент времени  мы подали возмущающий импульс, движение системы возмутилось, т.е. вышло за рамки предписанной (гармонической) траектории, а затем движения системы возвратилось к предписанному гармоническому закону движения. Таким образом, мы видим, что синтезированный закон управления обеспечивает устойчивость движения по предписанной траектории.

**Процедура синтеза в общем случае**

**Рассмотрим процедуру синтеза для нелинейного объекта**

(2.27)

когда предписанная траектория движения управляемой системы задаётся в виде системы аналитических выражений

*.* (2.28)

Беря полные производные от интегральных кривых (2.28) с учетом уравнений движения (2.27) и записывая через скалярные произведения, имеем

(2.29)

Требуя выполнения заданной динамики стабилизации

(2.30)

предписанной траектории из (2.29) и (2.30), окончательно получаем систему для определения закона управления со стабилизацией

. (2.31)

В зависимости от линейной зависимости или линейной независимости векторов , процедура синтеза будет осуществляться разными путями.

В том случае, когда векторы представляют собой линейно независимую систему, процедура синтеза осуществляется на основе

(2.32)

которая представляет несколько иную запись системы (2.31).

Отсюда, в соответствии с результатами, которые приведены в приложении 2, вектор управления определяется

(2.33)

где коэффициенты C1 , определяется из следующей системы:

( 2.34)

В том случае, когда система векторов не является линейно независимой, процедура синтеза осуществляется с использованием следующей системы:

( (2.35)

Из этой системы, в соответствии с приложением 2, первоначально определяется

(2.36)

где коэффициенты С1 определяются из системы

(2.37)

Далее искомый вектор управления определяется из (2.36). Но здесь искомый вектор управления для требуемой предписанной траектории существует только в том случае, когда выполняется условие

(2.38)

Если данное условие не выполняется, то это говорит о невозможности осуществления движения управляемой системы по заданной (предписанной) траектории.

**Рассмотрим процедуру синтеза для линейного объекта**

Пусть объект описывается:

(2.39)

предписанная траектория описывается в виде системы

(2.40)

Процедура синтеза аналогична той процедуре, которую мы рассмотрели выше.

Беря полные производные от интегральных кривых (2.40) с учётом уравнений движения (2.39) и записывая через скалярные произведения, имеем

(2.41)

Требуя выполнения заданной динамики стабилизации

(2.42)

предписанной траектории из (2.41) и (2.42), окончательно получаем систему для определения закона управления со стабилизацией

. (2.43)

В зависимости от линейной зависимости или линейной независимости векторов , процедура синтеза будет осуществляться разными путями. В том случае, когда векторы представляют собой линейно независимую систему, процедура синтеза осуществляется на основе

(2.44)

которая представляет несколько иную запись системы (2.43).

Отсюда, в соответствии с результатами, которые приведены в приложении 2, вектор управления определяется

(2.45)

где коэффициенты C1 , определяется из следующей системы:

( 2.46)

В том случае, когда система векторов не является линейно независимой, процедура синтеза осуществляется с использованием следующей системы:

( (2.47)

Из этой системы, в соответствии с приложением 2, первоначально определяется

(2.48)

где коэффициенты С1 определяются из системы

(2.49)

Далее искомый вектор управления определяется из (2.48). Но здесь искомый вектор управления для требуемой предписанной траектории существует только в том случае, когда выполняется условие

(2.50)

Если данное условие не выполняется, то это говорит о невозможности осуществления движения управляемой системы по заданной (предписанной) траектории.

**Рассмотрим модельный пример синтеза**

Уравнения углового движения искусственного спутника Земли при действии управляющего момента относительно главных осей инерции спутника с одной осью симметрии задаются уравнениями [56] в виде

(2.51)

Осуществим колебательное движение объекта вокруг первых двух осей, при постоянной угловой скорости вращения относительно третьей оси, т.е. предписанную траекторию зададим в виде

(2.52)

Определим, в соответствии с (2.51) и (2.52), необходимые выражения:

Векторы в нашем случае линейно независимы. Действительно,

На основании формул (2.32 ), (2.33) и (2.34), имеем

где коэффициенты С1, С2 определяются из

Подставляя числовые выражения, имеем

Отсюда

Подставляя найденные коэффициенты в выражение (2.53), окончательно имеем

Для физической реализуемости синтезированного закона (2.54) достаточно положить

где

При этом динамика стабилизации определяется в виде

где , и путём выбора постоянных а1, а2 осуществляется требуемая динамика

Как видно из синтезированного закона (2.54) на предписанной траектории, в силу

R1 (0,x,t)=0, R2 (0,x,t)=0 имеем U1=0; U2 =0; U3=0.

Тогда уравнение замкнутой системы на предписанной траектории имеет вид

(2.55)

Из интегрирования этой системы имеем

(2.56)

Система (2.56) равносильна системе (2.52), т.е. предписанной программе движения, что и требовалось доказать.

**Модельный пример синтеза алгоритмов управления по предписанной траектории**

Рассмотрим управляемый объект, описываемый дифференциальным уравнением

(2.57)

где – n – мерный вектор состояний, u - m – мерный вектор управляющих функций.

Принимается, что вектор-функция есть непрерывная ограниченная функция, имеющая частные производные по совокупности переменных в области определения уравнения (2.57).

Предписанная траектория движения системы задаётся в виде уравнений

(2.58)

Здесь непрерывные функции своих аргументов, имеющие частные производные

Равен “s” в каждый момент времени .

Рассмотрим следующую задачу синтеза. Требуется синтезировать такой закон управления, при котором движение системы (2.57) проходит по предписанной программе (траектории) (2.58).

В качестве **примера** рассмотрим. Пусть управляемый объект описывается системой

(2.59)

или

(2.60)

### где A = , B =

Поставим задачу осуществления движения системы по прямой, описываемой системой

(2.61)

Для решения задачи синтеза предварительно выясним вопрос линейной независимости векторов

Для этого определим

;

Из найденных выражений следует линейная независимость рассматриваемых векторов. В соответствии с полученным результатом решение задачи должно осуществляться на основании (2.62), (2.63), (2.64).

(2.62)

(2.63)

(2.64)

Для решения задачи синтеза также определим:

Согласно формуле (2.64), искомый вектор управления определяется в виде

(2.65)

где коэффициент C1 и C2, согласно формуле (2.62) определяется из системы

(2.66)

Подставляя конкретные выражения, имеем

Решая систему, находим

(2.67)

Подставляя найденные значения для этих коэффициентов в выражение (2.63), имеем

= (2.68)

Итак

;

; (2.69)

В найденных выражениях функция удовлетворяют условиям

; (2.70)

Для удовлетворения условия стабилизации заданной траектории произвольные функции R1 и R2 выбирают в виде

(2.71)

где Действительно, в этом случае динамика стабилизации описывается системой

(2.72)

где - oшибка выполнения заданной программы движения.

В связи с условиями как нетрудно видеть, обеспечивается процесс стабилизации заданной траектории

На заданной траектории функции и поэтому управление на заданной траектории имеет вид

(2.73)

Покажем, что действительно, при таком управлении обеспечиваются движение по заданной траектории (2.61) для этого рассмотрим управление замкнутой системы при управлении (2.73). Имеем

(2.74)

Интегрируя систему (2.74), имеем

;

; (2.75)

;

Теперь, поставив эти найденные функции в уравнений траектории (2.61), покажем выполнение тождества. Действительно, подставляя выражения , имеем

Отсюда

В выражения, характеризующие управляющие функции (2.69), подставим значения функций по (2.71), предварительно их рассчитав.

При расчёте примем следующее .

)

Учитывая (2.61) вычисляем окончательные значения функций

Рассчитав начальные значения , приняв используя выражение (2.75)

(2.76)

Для объекта управления, описываемого системой (2.59), по полученному закону управления, моделируем замкнутую систему управления с помощью ППП Матлаб.

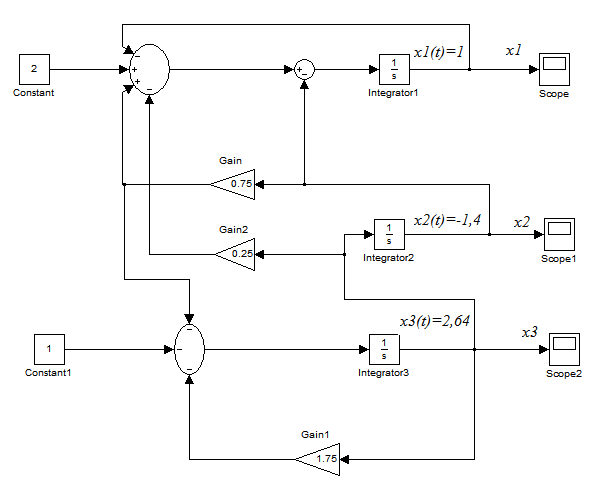


Рис. 2.3. Моделирование замкнутой системы управления

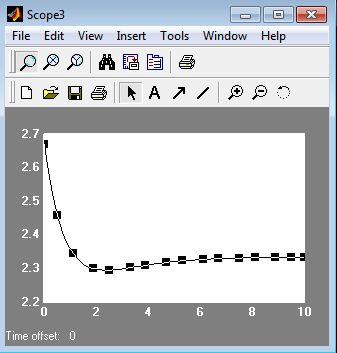


Рис.2.4. Процесс на выходе x1 (t)

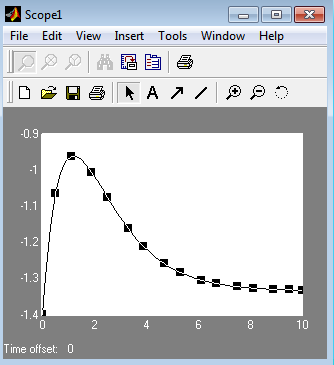


Рис. 2.5. Процесс на выходе x2 (t)

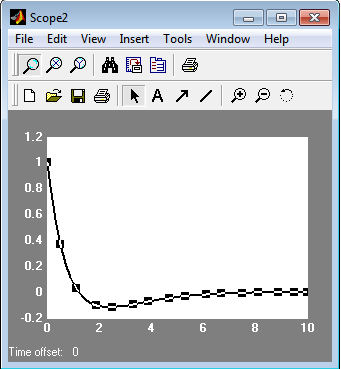


Рис. 2.6. Процесс на выходе x3 (t)

Итак, видно, что синтезированный закон в виде (2.73) обеспечивает движение системы по предписанной траектории (2.61). Физическая реализация синтезированных функций не вызывает никаких затруднений.

**2.2. Подход синтеза законов управления в случаях задания предписанной траектории движения в табличной форме**

Во многих практических важных случаях, например, в задачах робототехники, при автоматизации раскроя и конструирования одежды, использования лазерных технологий микроэлектроники, в системах 3D технологий и др. аналитическое представление предписанной программы движения вызывает большие затруднения. В этих случаях предписанную траекторию движения можно представлять в виде конечномерного множества дискретных контрольных точек, в частности в виде таблицы. В этих случаях предлагается следующий подход к синтезу.

**Пусть управляемый объект описывается уравнением**

  (2.77)

где, 



Требуется синтезировать закон управления по осуществлению движения управляемой системы по предписанной траектории, заданной в табличной форме.

Таблица 2.1- Предписанная траектория движения, заданная в табличной форме

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | *…* |
|  |  |  |  |  | *…* |
|  |  |  |  |  | *…* |
| *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  |  |  | *…* |

Для удобства принимается ,  то есть вводится абстрактные дискретные моменты времени.

Уравнение системы после дискретизации имеет следующий вид:



или  (2.78)

 (2.79)

где  - текущее состояние, - состояние на следующем шаге. Это выражение связывает текущее состояние , текущее управление  и состояние  в следующий момент времени.

Из выражения (2.79) можно определить :



Однако такой способ не гарантирует устойчивого осуществления движения по требуемым предписанным дискретным точкам движения.

Поэтому, искомое управление будем находить путём минимизации квадрата невязки между требуемыми и текущими дискретными значениями состояния управляемого объекта.

Таким образом управление будем искать из условия:

 (2.80)

где  − табличное значение,  − текущее значение.

Распишем выражение (2.80) подробно

  
затем взяв частную производную по *u(k)* из этого выражения, тем самым найдём необходимое управление



Распишем эту процедуру для линейного объекта

;

или

; (2.81)

Подставив конкретные выражения, возьмём скалярные произведения, и получим общее выражение:



Возьмём частную производную по 



Отсюда, окончательно, управление определяется в виде

 (2.82)

**Модельный пример синтеза управления в случае предписанной программы движения, заданной в табличной форме**

Рассмотрим модельный пример синтеза по таблично заданной предписанной программе движения. Пусть математическая модель движения системы описывается системой уравнений

 (2.83)

При этом задана следующая предписанная программа движения в табличном виде.

Таблица 2.2 - Предписанная программа движения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 |
|  | 0.5 | 0.7 | 0.8 | 0.4 |
|  | 0.5 | 0.71 | 0.6 | 0.91 |

Для нахождения искомых управляющих функций в дискретные моменты времени применим полученное выражение (2.82).

Используя (2.82), проведём 3 итерации по определению искомых законов управления, а также значения вектора состояния при действии этих найденных значений для управляющих функций.

Итак, определим управление , т.е. значение управляющего вектора в момент времени , при начальном положении 

Для этого запишем дискретную модель объекта с использованием (2.83)

 (2.84)

В векторно-матричной записи матрицы *А* и *В* имеют вид

; 

Согласно (2.83) имеем

 (2.85)

Подставим в выражение (2.83), с учётом .

Тогда будем иметь

 (2.86)

При ****, имеем

 (2.87)

При этом значения компонент вектора состояния

 (2.88)

При , имеем

(2.89)

и вектор состояния принимает значения

 (2.90)

Сравнивая выражения (2.86), (2.88), (2.90) с последними 3-мя столбцами таблицы (2), мы видим что, синтезированные законы управления (2.85), (2.87), (2.89) обеспечивают осуществление движения управляемого объекта (2.83) по таблично заданной траектории.

**2.3. Адаптивное управление траекторным движением**

В начале, рассмотрим процедуру синтеза для линейного объекта, затем рассмотрим процедуру синтеза для одного класса нелинейных объектов.

Рассмотрим линейный объект управления, описываемый уравнением

x, (2.91)

где A, B – числовые детерминированные матрицы размерностей nn; nm соответственно;

∆A – матрица неизвестных изменяющихся по времени параметров объекта (матрица параметрических возмущений).

Матрица неизвестных параметров объекта удовлетворяет так называемому условию квазистационарности, при котором предполагается, что параметры объекта изменяются намного медленнее, чем переменные состояния в переходных процессах.

В соответствии с этим принимается

. (2.92)

Предписанная программа движения задаётся в виде уравнений

(2.93)

Предполагается при этом, что ранг матрицы равен “**s**”.

Задачу синтеза адаптивного управления программным движением сформулируем следующим образом. Требуется синтезировать адаптивный закон управления в классе

(2.94)

где С – матрица настраиваемых параметров регулятора, при котором движение системы (2.91) осуществляется по предписанной программе (2.93).

# Построение адаптивного управления в частном случае

Рассмотрим случай, когда предписанная программа движения задана одним уравнением в неявной форме

(2.95)

Для решения выше поставленной задачи применим аппарат функций Ляпунова. Для формирования функции Ляпунова предварительно заметим, что цель адаптивного управления в данном случае заключается в обеспечении движения управляемой системы по предписанной траектории и в обеспечении возвращения движения системы на предписанную траекторию, в случае выхода системы из неё вследствие действия параметрических возмущений и возмущений по переменным состояния. Другими словами, цель управления заключается в выполнении условия устранения ошибки выполнения предписанной программы движения.

(2.96)

Исходя из этого замечания, функция Ляпунова определяется как функция от ошибки выполнения предписанной программы. Она должна, также, учитывать процесс компенсации параметрических возмущений путём настройки параметров регулятора. Применительно к выше поставленной задаче и исходя из структуры управляемого объекта (2.91), структуры регулятора (2.94) и предписанной программы движения (2.95) функцию Ляпунова возьмём в виде

V(,) = + (2.97)

где – 1-я строка матрицы (∆A – BC), т.е.

(2.98)

Очевидно, что рассматриваемая функция Ляпунова является положительно определённой функцией по отношению к переменным

Для нахождения искомого закона управления вычислим полную производную функции Ляпунова с учётом рассматриваемых уравнений.

Имеем

(2.99)

Заметим, что выражение , с учётом обозначений (2.98), можно представить в виде:

(2.100)

С учётом этого выражение (2.99) представится как:

(2.101)

В соответствии с поставленной целью управления (2.96) потребуем, чтобы полная производная функции Ляпунова удовлетворяла условию:

*,*  (2.102)

где – знакоотрицательная функция;

– произвольная положительно определённая (относительно) функция, удовлетворяющая условию

. (2.103)

С учётом (2.102) выражение (2.101) представится:

(2.104)

Если существует адаптивный закон управления (2.94), который удовлетворяет соотношению (2.104), то, очевидно, вышепоставленная цель управления (2.96) достигнута. Действительно, при выполнении соотношения (2.104) выполняется известные теоремы В.В. Румянцева об устойчивости по части переменных. В рассматриваемом случае это происходит по отношению к переменной:

.

Для определения искомого адаптивного закона управления используется (2.104). Предварительно (2.104) преобразуем к виду:

(2.105)

Из последнего соотношения программную часть регулятора определим из соотношения:

(2.106)

а алгоритм настройки параметров регулятора определяется в виде

или

(2.107)

С учётом обозначений (2.98) и условия квазистационарности параметров объекта (2.92) окончательно имеем

, , (2.108)

где ( обозначает i - ю строку матрицы.

Программную часть регулятора определяем из соотношения (2.106) по аналогичной процедуре, как это делали во второй главе. Опуская промежуточные выкладки, окончательно имеем:

(2.109)

Итак, вышепоставленная задача адаптивного управления программным движением решается на основании (2.94), (2.108) и (2.109).

**Рассмотрим теперь процедуру синтеза для нелинейного объекта заданной системой**

(2.110)

где, F(x) – нелинейная вектор функция;

– матрица параметрических возмущений;

Предписанная программа движения задана одним уравнением в неявной форме

(2.111)

Для решения выше поставленной задачи применим аппарат функций Ляпунова. Для формирования функции Ляпунова предварительно заметим, что цель адаптивного управления в данном случае заключается в обеспечении движения управляемой системы по предписанной траектории и в обеспечении возвращения движения системы на предписанную траекторию, в случае выхода системы из неё вследствие действия параметрических возмущений и возмущений по переменным состояния. Другими словами, цель управления заключается в выполнении условия устранения ошибки выполнения предписанной программы движения.

(2.112)

Исходя из этого замечания, функция Ляпунова определяется как функция от ошибки выполнения предписанной программы. Она должна также учитывать процесс компенсации параметрических возмущений путём настройки параметров регулятора. Применительно к выше поставленной задаче и исходя из структуры управляемого объекта (2.110), структуры регулятора (2.108) и предписанной программы движения (2.111) функцию Ляпунова возьмём в виде

V(,) = + (2.113)

где – 1-я строка матрицы (∆A – BC), т.е.

(2.114)

Очевидно, что рассматриваемая функция Ляпунова является положительно определённой функцией по отношению к переменным

Для нахождения искомого закона управления вычислим полную производную функции Ляпунова с учётом рассматриваемых уравнений.

Имеем

(2.115)

Заметим, что выражение , с учётом обозначений (2.114), можно представить в виде:

(2.116)

С учетом этого выражение (2.115) представится как:

(2.117)

В соответствии с поставленной целью управления (2.112) потребуем, чтобы полная производная функции Ляпунова удовлетворяла условию:

*,*  (2.118)

где – знакоотрицательная функция;

– производная положительно определенная (относительно ) функция, удовлетворяющая условию

. (2.119)

С учётом (2.118) выражение (2.117) представится:

(2.120)

Если существует адаптивный закон управления (2.94), который удовлетворяет соотношению (2.104), то, очевидно, вышепоставленная цель управления (2.112) достигнута. Действительно, при выполнении соотношения (2.104) выполняется известные теоремы В.В. Румянцева об устойчивости по части переменных. В рассматриваемом случае это происходит по отношению к переменной:

.

Для определения искомого адаптивного закона управления используется (2.120). Предварительно (2.120) преобразуем к виду:

(2.121)

Из последнего соотношения программную часть регулятора определим из соотношения:

(2.122)

а алгоритм настройки параметров регулятора определяется в виде

или

(2.123)

С учётом обозначений (2.114) и условия квазистационарности параметров объекта (2.92) окончательно имеем

, , (2.124)

где ( обозначает i-ю строку матрицы.

Программную часть регулятора определяем из соотношения (2.122) по аналогичной процедуре, как это делали во второй главе. Опуская промежуточные выкладки, окончательно имеем:

(2.125)

Итак, вышепоставленная задача адаптивного управления программным движением решается на основании (2.84), (2.124) и (2.125).

**Рассмотрим модельный пример синтеза адаптивного закона управления программным движением**

Пусть объект управления описывается системой

(2.126)

где,

При этом элементы матрицы параметрических возмущений удовлетворяют условию квазистационарности.

Пусть предписанная программа движения описывает гармонический процесс или

= + = =0 (2.127)

Для нахождения , согласно выражения (2.109), выберем в качестве положительно определённой функции

,х) = (x,t) . (2.128)

Тогда имеет место

(2.129)

С учётом того, что = 0 ,

имеем

(2.130)

При отсутствии параметрических возмущений. Повторяя те же рассуждения, выбор а(х) осуществляем из условий физической реализуемости (в данном случае это устранение в (2.124) деления на ).

Окончательно имеем

=( ) . (2.131)

где положительное число.

Алгоритм самонастройки параметров регулятора определяется согласно (2.124). Для рассматриваемого примера уравнение (2.124) распишется в виде

=

или отсюда имеем

(2.132)

С учётом выражения для

(2.133)

Окончательно уравнение замкнутой системы в данном примере с учетом структуры регулятора вида.

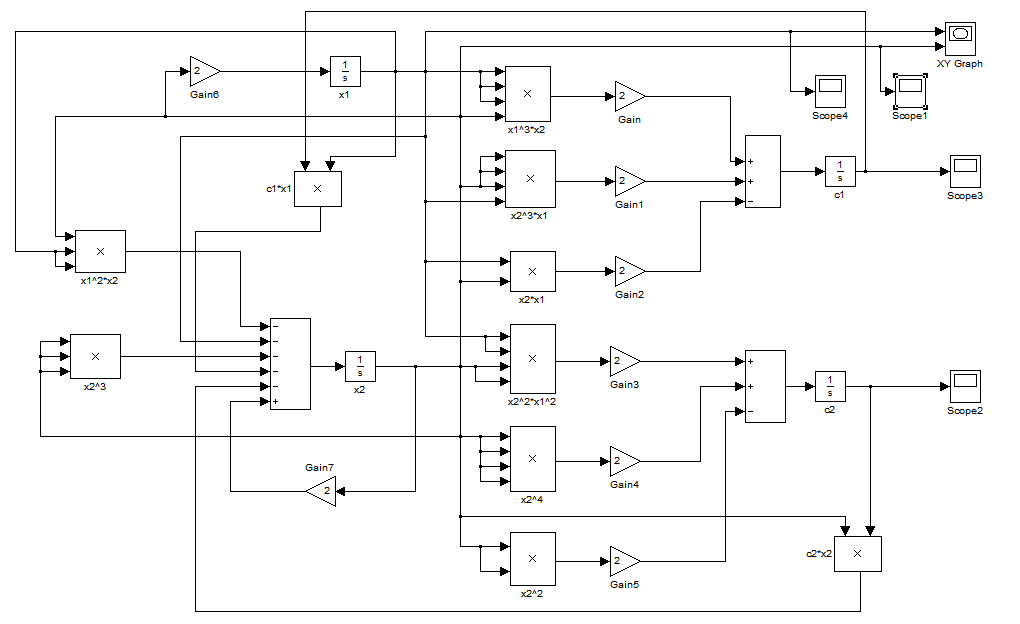
представится в виде

2.134)

Проинтегрировав (2.134) со значениями параметров ,

получим

Далее исследуем процессы и пронаблюдаем фазовый портрет графического выхода “XY- Graph” с помощью Матлаб. Результаты моделирования показаны на Рис. 2.8., 2.9., 2.10., 2.11., 2.12.

****Рис. 2.7. Моделирование системы управления

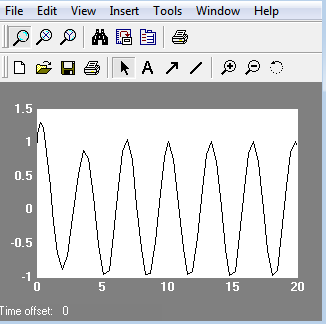


Рис. 2.8. Процесс на выходе x1(t)

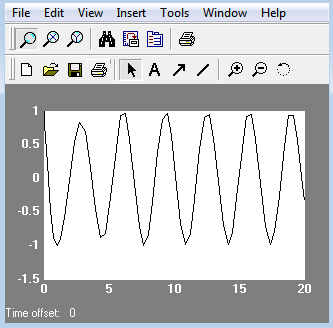


Рис. 2.9. Процесс на выходе x2 (t)

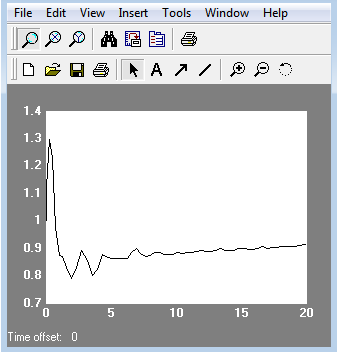


Рис. 2.10. Процесс на выходе C1

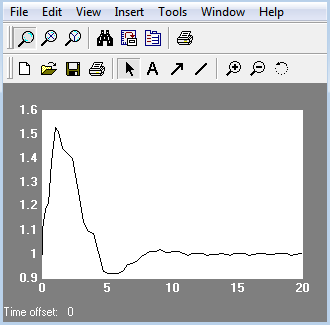


Рис. 2.11. Процесс на выходе C2

# 

Рис. 2.12. Фазовый портрет выходных сигналовx1(t) и x2 (t)

**Построение адаптивного управления в общем случае**

Рассмотрим случай, когда предписанная программа движения описывается системой уравнений

(2.135)

а объект описывается системой

.

Целью управления в рассматриваемом случае является выполнение условий

(2.136)

Функция Ляпунова формируем в виде

(2.137)

где – обозначает, i – ю строку матрицы (.

Для отыскания искомого закона сперва, вычисляем полную производную функции Ляпунова на движениях системы, затем приравняем его к произвольной знакоопределенной относительно переменных функций.

Рассмотрим эту схему построения искомого закона управления. Вычисляем полную производную функции Ляпунова на движениях системы. Имеем

(2.138)

(2.139)

Учитывая последнее соотношение выражения (2.138), запишем

+

(2.140)

В последнем выражении произведём следующие преобразования:

= (2.141)

С учётом последнего выражения (2.140) представим

+

(2.142)

Потребуем выполнения условия

(2.143)

где a(x) – знакоотрицательная функция выбираемая из условия физической реализуемости синтезируемого закона управления;

– произвольная положительно определённая по переменным функция.

Если закон управления определить из системы (2.142), (2.143), то очевидно, согласно известным теоремам В.В. Румянцева, цель управления (2.116) будет достигнута. Таким образом, из систем (2.142), (2.143) определяем, во-первых, закон настройки параметров регулятора в виде

(2.144)

и во-вторых, определяем программную часть закона управления. Для определения последнего имеем

(2.145)

где

Запишем последнее выражение в следующем равносильном виде:

(2.146)

из уравнение (2.146) окончательно имеем

(2.147)

С учётом обозначений для квазистационарности алгоритм настройки (2.144) окончательно запишется в виде

, (2.148)

где через , обозначается i-я строка матрицы .

Итак, адаптивный закон управления определяется через соотношения (2.147), (2.148) и имеет структуру:

(2.149)

**Рассмотрим теперь процедуру синтеза для одного класса нелинейных объектов**

(2.150)

Рассмотрим процедуру синтеза для одного класса нелинейных случай, когда предписанная программа движения описывается системой уравнений

(2.151)

Целью управления в рассматриваемом случае является выполнение условий

(2.152)

Функция Ляпунова формируем в виде

(2.153)

где – обозначает, i – ю строку матрицы (.

Для отыскания искомого закона сперва, вычисляем полную производную функции Ляпунова на движениях системы, затем приравняем его к произвольной знакоопределенной относительно переменных функций.

Рассмотрим эту схему построения искомого закона управления. Вычисляем полную производную функции Ляпунова на движениях системы. Имеем

(2.154)

(2.155)

Учитывая последнее соотношение выражения (2.154), запишем

+

(2.156)

В последнем выражении произведём следующие преобразования:

= (2.157)

С учётом последнего выражения (2.156) представим

+

(2.158)

Потребуем выполнения условия

(2.159)

где a(x) – знакоотрицательная функция выбираемая из условия физической реализуемости синтезируемого закона управления;

– произвольная положительно определённая по переменным функция.

Если закон управления определить из системы (2.158), (2.159), то очевидно, согласно известным теоремам В.В. Румянцева, цель управления (2.132) будет достигнута. Таким образом, из систем (2.158), (2.159) определяем, во-первых, закон настройки параметров регулятора в виде

(2.160)

и во-вторых, определяем программную часть закона управления. Для определения последнего имеем

(2.161)

где

Запишем последнее выражение в следующем равносильном виде:

(2.162)

из уравнение (2.162) окончательно имеем

(2.163)

С учётом обозначений для квазистационарности алгоритм настройки (2.160) окончательно запишется в виде

, (2.164)

где через , обозначается i-я строка матрицы .

Итак, адаптивный закон управления определяется через соотношения (2.163), (2.164) и имеет структуру:

(2.165)

**Рассмотрим модельный пример синтеза адаптивного закона управления**

Рассмотрим объект управления, описываемый системой

(2.166)

### где,

A = , B = , A = ,

.

Поставим задачу осуществления движения системы (2.166) по прямой, описываемой уравнениями:

(2.167)

Управление строим в виде:

.

Для определения по формуле (2.163) предварительно установим:

;

;

;

;

;

В качестве выражения возьмём

) =

С учётом вышеопределённых выражений, согласно выражения (2.163), имеем

; (2.168)

Структура выражения (2.168) показывает, что она вполне физически реализуема.

Найдём алгоритм настройки параметров регулятора. Алгоритм настройки определяем по формуле (2.164).

Предварительно определим:

= ;

= ;

= ;

B =

Подставляя найденные выражения в формулу (2.164), окончательно определим:

(2.169)

Уравнение замкнутой системы, согласно найденным выражениям (2.168), (2.169) представится в виде

(2.170)

**ВЫВОДЫ**

В данной главе описывается три подхода к синтезу законов управления по осуществлению движения управляемого объекта по предписанной траектории.

В первом, предлагается подход к синтезу, в случае, когда предписанная траектория задаётся аналитически. Во втором - когда предписанная траектория задаётся таблично.

В третьем подходе рассматривается методика синтеза адаптивного управления в случае действия параметрических возмущений.

Предлагаемые подходы к синтезу позволяют решать задачи управления не только для класса линейных объектов, но и для широкого класса нелинейных объектов.

В предлагаемых подходах синтеза законы управления, определяются с учётом их физической реализуемости.

Определены (выведены) критерии по осуществляемости, для рассматриваемых объектов, движения по заданным предписанным траекториям.

Приведённые модельные примеры синтеза показывают, что предлагаемые методики синтеза конструктивны и эффективны.

**ГЛАВА 3. ПРИКЛАДНЫЕ ЗАДАЧИ УПРАВЛЕНИЯ ТРАЕКТОРНЫМ ДВИЖЕНИЕМ**

В данной главе рассматривается применение разработанной теории синтеза управления траекторным движением объекта для некоторых прикладных задач управления.

Разработанная теория применяется: для построения системы управления движением рабочих органов 3х-звенного манипулятора по заданной траектории; для построения системы управления шаговыми приводами 3D-принтера.

Для решения поставленных задач, вначале, разработаны математические модели рассматриваемых управляемых объектов, далее используя разработанные во 2-й главе методики синтеза синтезированы искомые законы управления. Кроме этих вопросов рассмотрены вопросы технической реализации синтезированных систем управления.

**3.1.** **Исследование и разработка системы управления траекторным движением трёхзвенного манипулятора**

В данном разделе рассматривается проблема синтеза законов управления шаговыми электроприводами кинематических звеньев манипулятора по осуществлению движения по предписанным траекториям.

Вопрос синтеза соответствующих законов управления шаговыми приводами манипуляторов наталкивается на то обстоятельство, что к настоящему времени нет универсальной методики синтеза законов управления по осуществлению движения рабочих органов манипулятора по произвольной предписанной программе движения. Решаются вопросы синтеза законов управления, только, для определённых предписанных движений отдельными частными методами.

Для синтеза искомых законов управления шаговыми приводами, вначале выведена математическая модель шагового привода, как объекта управления, далее приведена процедура синтеза.

Сегодня на промышленных предприятиях крайне востребованы автоматизированные системы построенные на применении различных, роботов и роботизированных комплексов, актуальны также инновационные технологические решения, которые помогают наладить эффективный производственный процесс и в то же время минимизировать отрицательное воздействие производства на работников.

Все это способствовало внедрению на предприятия промышленных роботов, отличающихся своей высокой производительностью, не требующих время на отдых, исключающих из своей работы ошибки.

Любой промышленный робот состоит из 3-х подсистем: манипулятора; информационно-измерительной подсистемы; управляющей подсистемы. Существуют различные типы манипуляторов, которые характеризуются различной кинематической структурой. Кинематическая структура определяет степени подвижности и траекторию движения кинематических звеньев в пространстве.

В настоящее время в манипуляторах промышленных роботов в качестве привода для кинематических звеньев широко начинает использоваться электроприводы на шаговых двигателях.

В связи вышесказанными предлагаемый, достаточно, универсальный, подход синтеза законов управления движением кинематических звеньев манипулятора по предписанным траекториям является актуальной практически значимой проблемой.

Итак, рассмотрим 3-х звенный манипулятор, кинематическая структура, которого представлена на рис.3.1.

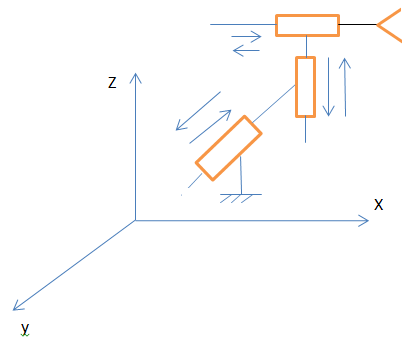


Рис. 3.1. Кинематическая структура 3-х звенного манипулятора

Каждое звено манипулятора приводится в движение своим шаговым электроприводом.

В работе решается задача осуществления движения рабочего органа манипулятора (захватного устройства) по заданной предписанной программе движения. В качестве примера, рассмотрим простейшую предписанную программу, заданную в табличной форме в виде табл. 3.1.

Геометрически, предписанная траектория движения отображается в виде движения рабочего органа манипулятора по вершинам пирамиды (рис. 3.2).

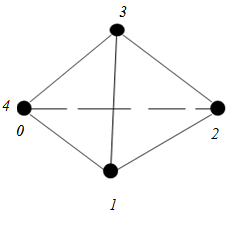


Рис. 3.2. Контрольные точки предписанной программы движения

Движение начинается с вершины 1 далее осуществляется переход к вершинам 2,3,4 и 5. Программа движения отображается в виде таблицы 3.1.

Таблица 3.1. – Предписанная заданная программа

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| tk | t0=0 | t1=1 | t2=2 | t3=3 | t4=4 |
| Хк | 0 | 7 | 8 | 2.2 | 0 |
| Ук | 0 | 7 | 8 | 2.2 | 0 |
| Zk | 0 | 7 | 8 | 8 | 0 |

где Xk, Уk Zk – координаты вершин в дискретные моменты времени tk;

Для синтеза искомых законов управления шаговыми приводами по осуществлению движения по предписанной программе, заданной в табличной форме в виде таблицы 3.1. нами выведена [10] математическая модель дискретного движения 3-х шаговых приводов по осям X, У, Z в виде системы (3.1)

(3.1)

где Xk, Уk,  Zk – координаты рабочего органа по осям x,у,z в дискретный момент tk;

Xk+1, Уk+1, Zk+1  - координаты к моменту времени tk+1;

Lшх, Lшу, Lшz - соответствующие величины линейных перемещений рабочего органа по осям х, у, z от действия одиночных импульсов.

nkx, nkу, nkz – необходимое количество управляющих импульсов подаваемых на отрезке времени [ tk, tk+1] на шаговые двигатели для осуществления движения по осям х, у, z, из точки Xk, Уk,  Zk к точке Xk+1, Уk+1,  Zk+1.

По таблице 3.1. по формуле (3.2) компьютер вычисляет необходимые количества импульсов:

(3.2)

При выводе системы (3.1) учитывалось, что на вал роторов шаговых двигателей посажены зубчато-винтовые механизмы.

Величины Lшх , Lшу, Lшz определяются конструктивными параметрами зубчато-винтового передаточного механизма. В частности если используется шестерня с радиусом R и угловым расстоянием между зубьями [град], то Lшх, определяется по формуле

(3.3)

В дальнейшем при расчётах нужно перевести в радианы по формуле

(3.4)

Такой перевод необходим, чтобы все расчёты производились в системе СИ.

С применением ППП Matlab для конкретных параметров передаточного механизма и конкретных числовых значений таблицы 3.1. проведено компьютерное моделирования траекторного движения рассматриваемой системы ниже приведена соответствующая программа моделирования [31, 35].

Результаты моделирования показывают, что действительно, движение происходит по заданной предписанной траектории движения по таблице 3.1.

**Листинг программы:**

clc;clear all;close all;

%tetraedr

xt=[0 7 8 2.2 0 0 0 0];

yt=[0 7 8 2.2 0 0 0 0];

zt=[0 7 8 8 0 0 0 0];

%%traektoria

x=[0 7 8 2.2 0];

y=[0 7 8 2.2 0];

z=[0 7 8 8 0];

len=length(x)-1;

%%impulsi upravleniya

lx=1.7; % sag privoda po x

ly=1.7; % sag privoda po y

lz=1.7; % sag privoda po z

%% impulsi upravleniya

for i=1:len

nx(i)=(x(i+1)-x(i))/lx;

ny(i)=(y(i+1)-y(i))/ly;

nz(i)=(z(i+1)-z(i))/lz;

nx

ny

nz

end

%% nachalnaya tochka

X=0;

Y=0;

Z=0;

for i=2:len+1

X(i)=X(i-1)+nx(i-1)\*lx;

Y(i)=Y(i-1)+ny(i-1)\*ly;

Z(i)=Z(i-1)+nz(i-1)\*lz;

end

%tochki

xx=[];

yy=[];

zz=[];

for i=1:len

x1=linspace(x(i),x(i+1),1000);

xx=[xx x1];

y1=linspace(y(i),y(i+1),1000);

yy=[yy y1];

z1=linspace(z(i),z(i+1),1000);

zz=[zz z1];

end

%%graph

plot3(X,Y,Z) % optimalnaya taektoria

grid on

figure

plot3(xt,yt,zt,'\*') % predpisannaya traektoria

grid on

hold on

comet3(xx,yy,zz)

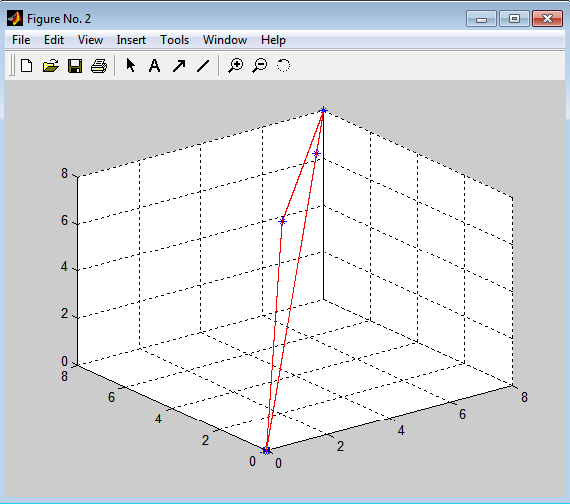
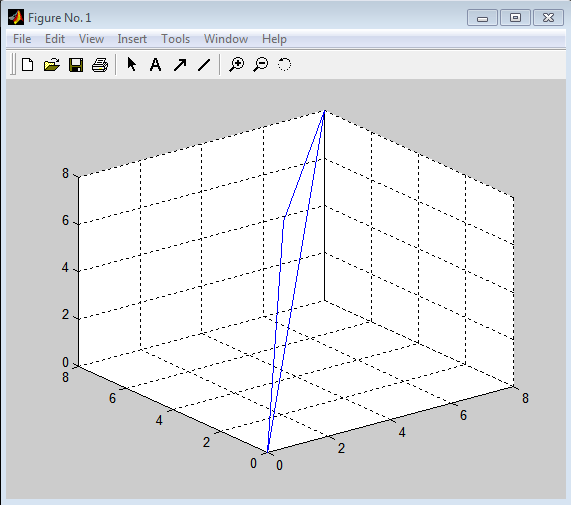
 

Рис. 3.3. Результаты компьютерного моделирования движения объекта по предписанной траектории

**3.2. Разработка и исследование системы управления приводами 3D – принтера**

3D печать (объёмное производство) есть процесс изготовления твёрдого объекта в трёхмерном пространстве из цифрового макета. Построение печатаемого в трёхмерном пространстве объекта достигается посредством использования аддитивных процессов, при которых объект создаётся нанесением последовательных слоёв материала, пока не будет построен весь объект. Каждый из этих слоёв может рассматриваться как тонкий срез конечного объекта в горизонтальном поперечном сечении.

Началом процесса является подготовка виртуального проекта объекта. Этот виртуальный проект, который мы хотим создать, выполняется в файле CAD (системы автоматизированного проектирования) с помощью программных средств 3D - моделирования (для создания нового объекта) или с использованием 3D - сканера (для копирования существующего объекта). 3D-сканер делает цифровую 3D-копию объекта. 3D-сканеры используют различные методы построения 3D-модели: пролётное время, модулированное/ структурированное световое излучение, объёмное сканирование и многие другие. Недавно компании Microsoft и Google включились в представление своего оборудования для 3D-сканирования, ярким примером является Microsoft Kinect. Это явный признак того, что будущие портативные устройства, такие как смартфоны, будут иметь встроенные 3d-сканеры. Оцифровка реальных объектов в 3D-модели будет настолько лёгкой, как обычное фотографирование. Цены на 3D-сканеры варьируются от очень дорогих профессиональных промышленных устройств до 30 - ти долларовых любительских устройств, которыми сможет пользоваться любой человек даже в домашних условиях.

Для подготовки цифрового файла к печати, «слои» программного обеспечения 3D-моделирования окончательной модели наносятся сотнями и тысячами горизонтальных слоёв. Когда файлы слоёв загружаются в 3D-принтер, объект может создаваться слой за слоем. 3D-принтер считывает каждый слой (или 2D-изображение) и строит в результате трёхмерный объект, смешивая каждый слой с едва заметным следом предыдущих слоёв. Эта технология не является единственной, кроме нее используются аддитивные методы с различными способами наслоения, при построении окончательного объекта. Часть этих методов при изготовлении слоев используют плавление или размягчение материала. Кроме методов селективного лазерного и наплавленного осаждения (FDM), которые наиболее распространены эффективно используется стереолитография (SLA), метод отвердевания фотореактивной смолы ультрафиолетовым лазером подобным ему излучателем мощности послойно один за другим.

В истории производства субтрактивные методы (для получения детали от заготовки удаляют все «лишнее») зачастую имеют приоритет. Область машинной обработки (создающей высокоточные формы) была, в общем, субтрактивным делом, от простого точения до фрезерования и шлифования.

Первое применение аддитивного производства стало периодом окончания широкого использования механообработки в спектре производства. Например, быстрое прототипирование было одним из самых ранних аддитивных вариантов, и его миссия заключалась в том, чтобы уменьшить время выполнения и стоимость разработки прототипов новых частей и устройств, что выполнялось ранее только субтрактивными методами механообработки (как правило, медленно и дорого). С развитием технологий аддитивные методы движутся все дальше внутрь конечного продукта производства. Части детали, которые ранее могли быть изготовлены только с помощью субтрактивных методов, могут быть теперь более выгодно изготовлены с помощью аддитивных методов.

Тем не менее, реальная интеграция новых технологий в промышленном производстве по существу дополняет субтрактивные методы, а не полностью вытесняет их [63, 88, 97, 108].

В условиях Кыргызстана будет функционировать и развиваться только мелкосерийное машиностроительное производство. Номенклатура вышесказанной продукции будет иметь короткий жизненный цикл. В связи с вышесказанным становится актуальным сокращение времени проектирования. 3D-технологии как раз направлены на сокращение времени проектирования и внедрение новых технологий производства объёмных деталей.

В подавляющем числе случаев в 3D-принтерах применяются шаговые приводы (ШП). Поэтому в данном исследовании ставится и решается задача управления движением рабочего органа 3D-принтера по заданной (предписанной) пространственной траектории движения. В качестве рабочего органа в данной диссертации по умолчанию предполагается головка экструдера (печатающая головка) и платформа, где происходит формообразование деталей.

Технологические вопросы (решения) управления приводами подачи полимерной нити на экструдер 3D- принтера, управления температурным режимом экструдера, давлением в нем и другие (решения) в данной работе не рассматриваются.

**3.3. Методика синтеза управляющих законов для шаговых двигателей 3D-принтера**

Результаты данной работы по синтезу управляющих алгоритмов для трёх совместно используемых ШД могут быть использованы в 3D-принтерах, упрощённая конструкция которых показана на рис. 3.4.

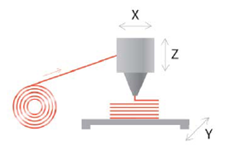


Рис. 3.4. Упрощенная конструкция 3D- принтера, использующего метод послойного наплавления (FDM)

Ниже детально рассматривается проблема синтеза законов управления ШД 3D-принтера, имеющая целью осуществить движение его рабочих органов по предписанным пространственным траекториям.

Прежде всего, приведём «базовую» информацию о ШД.

Для всех типов ШД с помощью электронного коммутатора вырабатываются импульсы напряжения, которые подаются на обмотки управления, расположенные на статоре ШД. При этом каждый импульс обеспечивает поворот на определённый угол, зависящий от конструкции ШД. В настоящее время в сфере машиностроения наибольшее распространение получили высоко моментные двухфазные гибридные шаговые электродвигатели с угловым перемещением 1,8°/шаг (200 шагов/оборот) или 0,9°/шаг (400 шагов/оборот).

Закон поворота ротора ШД во времени определяется последовательностью, скважностью и частотой поступления на него управляющих импульсов, а также типом и конструктивными параметрами ШД: число фаз (обмоток управления и схема их соединения) – различают однофазные и многофазные двигатели; тип ШД - с активным или пассивным ротором; одиночный шаг ротора (угол поворота ротора при единичном импульсе); номинальное напряжение питания; номинальный вращающей момент и.т.д. Управление ШД обеспечивается электронным блоком управления.

Для ШД, в силу особенностей, вопросы «люфта» ротора [49] при вращении, являются не актуальными. Зато важна точность установки ротора в очередное заданное положение, скорость поворота (установки) ротора в это положение и пр. Недостаточная скорость установки ШД в процессе работы 3D-принтера может привести к погрешностям воспроизведения формы детали.

Для синтеза законов управления рассмотрим ШД с зубчатым передаточным механизмом на валу двигателя. При помощи этого механизма вращательное движение ротора двигателя преобразуется в поступательное движение механизмов (в конечном счёте – печатающей головки) 3D- принтера.

Пусть на роторном валу ШД посажено (установлено) зубчатое колесо (шестерня) с радиусом *R* и угловым расстоянием между двумя зубьями  градусов, т.е. угол поворота ротора ШД, при действии единичного импульса составляет  градусов.

В дальнейшем требуется представить  в радианах. Перевод из градусов в радианы осуществляется по формуле

 (3.1)

При действии единичного импульса на ШД с посаженной на валу его ротора шестерней с радиусом *R*, одиночный угловой шаг поворота ротора преобразуется в поступательное движение шестерни на расстояние, равное

 (3.2)

где  выражена в радианах; - величина линейного шага шестерни при действии одного импульса на ШД.

Так как в ШД шаги осуществляются в дискретные моменты времени, то математическая модель (ММ) ШД как объекта управления описывается с использованием конечно-разностного выражения

 (3.3)

где,  - текущее угловое положение ротора ШД на  – ом моменте времени;  - шаг поворота ротора ШД при подаче единичного импульса;  - количество управляющих импульсов, подаваемых на ШД за отрезок времени  ;  -угловое положение ротора ШД в  - ый момент времени.

Формула (3.3) используется тогда, когда требуемые перемещения рабочих органов 3D - принтера описаны в угловых координатах. Однако в большинстве случаев необходимые перемещения описываются в линейных координатах. При этом, аналогично (3.3), предлагается следующая ММ управления движениями рабочих органов 3D-принтера, которая описана и обоснована в нашей работе [8, 9, 10] .

 (3.4)

где - текущие координаты рабочих органов принтера по осям  в  - ый момент времени;  – координаты рабочих органов по осям  в  момент времени;  - количества импульсов, поданных на ШД приводов по осям  за указанный промежуток времени;  – величины линейных шагов рабочего органа по осям  от действия одиночных импульсов на соответствующие ШД.

Отметим, что если количество подаваемых на ШД импульсов за «шаг по времени» для каждой оси движения печатающей головки разное, то для обеспечения лучшего воспроизведения формы детали необходимо решать задачу оптимизации распределения во времени импульсов, подаваемых на ШД с меньшей частотой.

Для ШД как объекта управления, целесообразно использовать математическую модель (3.4), содержащую выражения, связывающие управляемые величины с управляющими . Подчеркнём, что эта модель формально соответствует «независимому» управлению ШД: вопросы «синхронизации» импульсов управления при разном их количестве за шаг по времени непосредственно в данной модели не учитываются.

Таким образом, в практическом плане задача управления ШД сводится к нахождению количества управляющих импульсов из формул (3.4) и подаче их на соответствующие ШД на отрезке времени . При этом значения  берутся из таблицы предписанных (требуемых) контрольных точек пространственной траектории движения рабочих органов 3D-принтера (она может строиться в автоматическом или полуавтоматическом режиме на основе компьютерных моделей объектов).

Отметим, что в ряде случаев при планировании использования 3D-принтера приходится учитывать и инерционные характеристики ШД и рабочих органов, т.е. возможность корректной «отработки» ими всех поступивших импульсов управления за заданный промежуток времени.

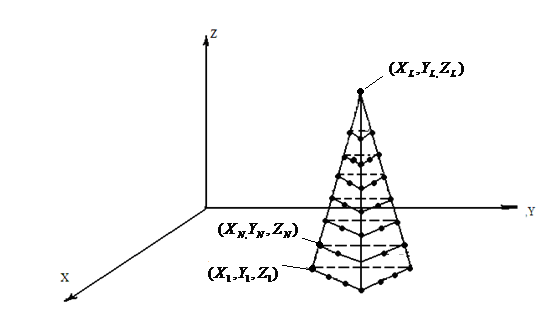
Проиллюстрируем построение предписанной пространственной траектории движения рабочих органов 3D-принтера с помощью рисунка 3.5.

Рис. 3.5. Пространственная модель детали с нанесёнными на неё

контрольными точками

По требуемой пространственной модели объекта с нанесёнными на неё контрольными точками строим таблицу для предписанной траектории движения рабочего органа 3D - принтера (таблица 3.2).

Таблица 3.2. - Предписанные координаты контрольных точках траектории

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | … |  |  | … |  |  |
|  |  |  | … |  |  | … |  |  |
|  |  |  | … |  |  | … |  |  |
|  |  |  | … |  |  | … |  |  |

Начало движения печатающей головки осуществляется с момента  и начального положения . Затем выполняется переход в положение , соответствующее моменту времени . Далее движение осуществляется аналогично, и в конце первого цикла, соответствующего «первому слою» воспроизводимого объекта, печатающая головка приходит в начальное положение, т.е. . Из этого положения осуществляется переход в положение  и затем движение осуществляется по второму слою объекта, воспроизводимого 3-D принтером. Далее движение выполняется по 3-ему слою и т.д. В конце работы осуществляется переход к конечному положению .

Отметим, что подсчёт количества необходимых импульсов по формулам (3.4) для обеспечения перемещений, задаваемых таблицей 3.1, может приводить к тому, что расчётное количество импульсов (для определённости – по «Х – оси») будет выражаться не целым числом.

 (3.5)

В этом случае, берётся ближайшее целое число (по стандартным правилам округления), а максимальная погрешность позиционирования ШД составит

 (3.6)

Следовательно, чтобы уменьшить погрешность позиционирования ШД, нужно уменьшить . А для этого, в свою очередь, необходимо уменьшить величину единичного шага (поворота ротора) , путём выбора ШД с необходимым количеством полюсов (в том числе - эквивалентных полюсов). Необходимая скорость движения ШД обеспечивается уменьшением или увеличением длины отрезка по времени  при фиксированных количествах импульсов, подаваемых на ШД за этот отрезок. Уменьшение отрезка по времени увеличивает скорость движения (перемещения) ШД и, как следствие, печатающей головки.

**3.4. Некоторые вопросы технической реализации системы управления шаговыми приводами прототипа 3 D- принтера**

В данном разделе решается вопрос технической реализации на основе применение микроконтроллерной техники.

Разрабатываемая система может работать как на основе применения параллельного, так и последовательного выходных портов компьютера.

**3.4.1. Функциональная схема системы управления шаговым приводом**

По сравнению с обычными двигателями постоянного тока, шаговые двигатели требуют более сложных схем управления, выполняющих коммутацию обмоток при работе. В зависимости от конфигурации обмоток двигатели делятся на биполярные и униполярные. Для управления биполярным двигателем требуется более сложный драйвер (силовая часть электросхемы управления шаговым двигателем, не путать с программным драйвером). На рис. 3.9 показана функциональная схема системы управления тремя шаговыми двигателями. Электрическую схему управления шаговыми двигателями с помощью компьютера можно разбить на две части: контроллер (буфер) и драйвер.

**Контроллер.** Контроллер шагового двигателя строит автономное управление системой различного рода технологическими процессами: управление линейными и вращательными движениями, упаковочные роботы управление дозаторами сыпучих продуктов и жидкостей, в качестве устройства управления открытием-закрытием окон, дверей и других целей.

**LPT порт компьютера.** IEEE 1284 (порт принтера, параллельный порт, [англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Line Print Terminal, LPT) — международный стандарт [параллельного интерфейса](http://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%BB%D0%BB%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D1%81%D0%BE%D0%B5%D0%B4%D0%B8%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%28%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%29) для подключения периферийных устройств к персональному компьютеру. В основном используется для подключения к компьютеру [принтера](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BD%D1%82%D0%B5%D1%80), [сканера](http://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D0%B0%D0%BD%D0%B5%D1%80) и других внешних устройств (часто использовался для подключения внешних устройств хранения данных), однако может применяться и для других целей (организация связи между двумя компьютерами, подключение каких-либо механизмов телесигнализации и [телеуправления](http://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BB%D0%B5%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5)). В основе данного стандарта лежит интерфейс Centronics и его расширенные версии (ECP, EPP). Внешний вид устройства представлен на рис.3.6.



Рис.3.6. Внешний вид LPT-порта.

Порт на стороне управляющего устройства (компьютера) имеет 25-контактный 2-рядный разъём DB-25-female («мама») (IEEE 1284-A). На периферийных устройствах обычно используется 36-контактный [микроразъем HYPERLINK ленточного типа](http://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BA%D1%80%D0%BE%D1%80%D0%B0%D0%B7%D1%8A%D0%B5%D0%BC_%D0%BB%D0%B5%D0%BD%D1%82%D0%BE%D1%87%D0%BD%D0%BE%D0%B3%D0%BE_%D1%82%D0%B8%D0%BF%D0%B0) Centronics (IEEE 1284-B), поэтому [кабели](http://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%B1%D0%B5%D0%BB%D1%8C) для подключения периферийных устройств к компьютеру по параллельному порту обычно выполняются с 25-контактным разъёмом DB-25-male на одной стороне и 36-контактным IEEE 1284-B на другой (AB-кабель).

Соединительный кабель должен быть длиной не более 3 метров. Структура кабеля: витые пары в общем либо в индивидуальных экранах. Ленточные кабели встречаются желательно реже.

Для подключения сканера и некоторых других устройств используется кабель с разъемом DB-25-male. Схемотехника сканера выполнена, опираясь на условие: при работе с принтером сканер прозрачно должен передавать данные с одного интерфейса на другой. Базовый интерфейс Centronics является однонаправленным параллельным [интерфейсом](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81), содержащие характерные для такого интерфейса сигнальные линии (8 для передачи данных, [строб](http://ru.wikipedia.org/w/index.php?title=%D0%A1%D1%82%D1%80%D0%BE%D0%B1&action=edit&redlink=1), линии состояния устройства).

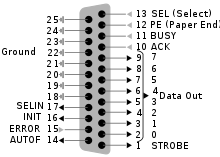


Рис.3.7. Интерфейс LPT-порта.

На рис. 3.8 показано возможное подключение шагового двигателя к LPT-порту.

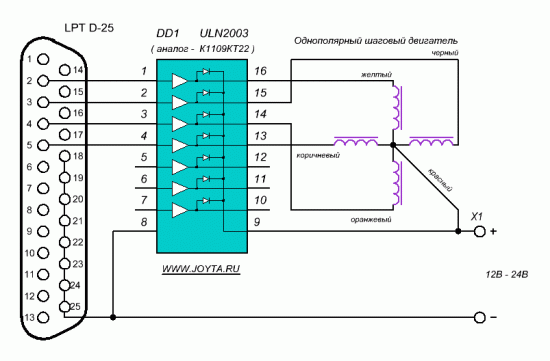
[](http://www.joyta.ru/uploads/2009/02/sh_d_shema.gif)

Рис.3.8. Схема контроллера шагового двигателя

Контроллер, схема которого представлена рис. 3.8, даст возможность через параллельный порт точно управлять однополярным шаговым двигателем.

LPT порт компьютера

КОНТРОЛЛЕР

Драйвер 1

Драйвер 2

Драйвер 3

ШД1

ШД2

ШД3

РО1

РО2

РО3

ДОС1

ДОС2

ДОС3

Рис.3.9. Функциональная схема системы управления 3-мя шаговыми двигателями

ШД - шаговые двигатели; РО – регулирующий орган; ДОС – датчик обратной связи

**3.4.2. Техническая реализация на основе микроконтроллера Arduino**

Рассмотрим техническую реализацию на основе применения микроконтроллера Arduino Uno, который необходим, для управления двигателем (Рис.3.10).

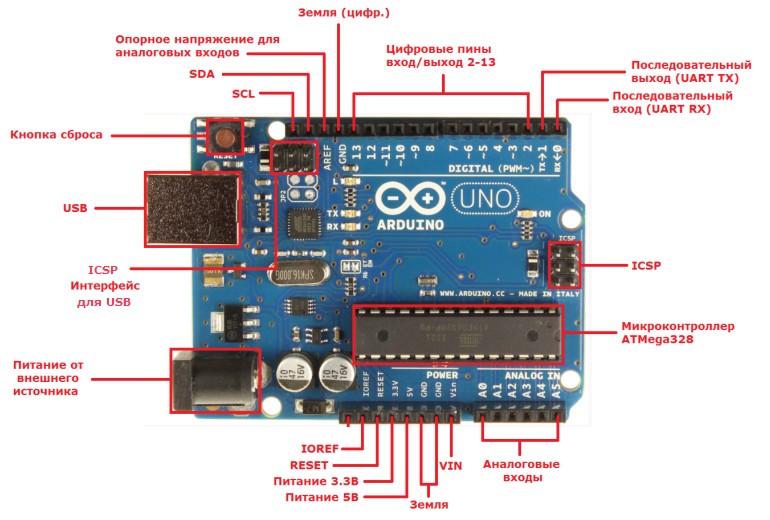


Рис. 3.10. Аппаратная платформа Arduino Uno

Arduino - это комплекс аппаратно-программных средств, с помощью которых создаются простые электронные системы автоматики и робототехники. У него полностью открытая архитектура, он ориентирован даже на «чайников». 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 кнопка перезагрузки.

Платформа подключается к компьютеру через кабель или USB, и питается при помощи адаптера AC/DC или батареи.

Программная часть Arduino состоит из интегрированной программной среды (IDE) и позволяет выполнять следующее: писать, компилировать программы, а также загружать их в аппаратуру.

Аппаратная часть – это электронные платы с микроконтроллером, сопутствующими элементами (стабилизатор питания, кварцевый резонатор, блокировочные конденсаторы и т.п.), портом для связи с персональным компьютером, разъёмами для сигналов ввода-вывода и т.п.

Arduino – по сути маленький компьютер, отличающийся от привычного в понимании компьютера тем, что работает с физическими электрическими сигналами. Основное назначение микроконтроллера это взаимодействие с физической средой (рис.3.11).

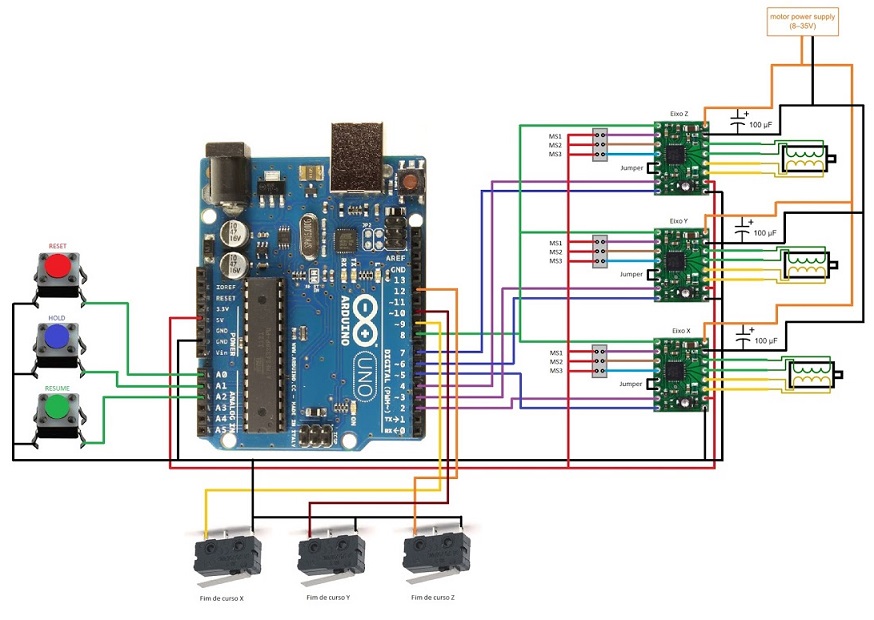
****

Рис.3.11. Структурно - функциональная схема системы управления 3D – принтера с использованием ARDUINO.

**Для управления ШД использован драйвер A4988***.* Модуль позволяет управлять одним биполярным двигателем. Максимальный выходной ток при напряжении 24 В составляет 2 А (может потребоваться теплоотвод). Модуль имеет простой интерфейс управления и подключается напрямую к системам с напряжением 3 – 5 В. Для управления шаговыми двигателями используются драйверы шаговых двигателей. Драйвер двигателя построен на базе микросхемы A4988, схематическое изображение которой представлено на рисунке 3.12. Драйвер шагового двигателя А4988 работает от напряжения 8 - 35 В и может обеспечить ток до 1 А на фазу без радиатора (и до 2 A с радиатором). Модуль A4988 имеет защищён от перегрузки и перегрева. ШД характеризуется количеством шагов на один оборот 360°. Так, для шаговых двигателей Nema17 это 200 шагов на оборот, т.е 1 шаг соответсвует 1.8°. Засчет драйвера A4988 можно увеличить это значение благодаря возможности управления промежуточными шагами. Он имеет пять режимов микрошага: 1 (полный), 1/2, 1/4, 1/8 и 1/16.

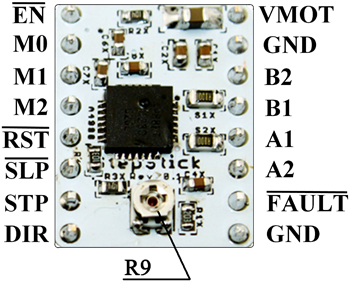


Рис.3.12. Драйвер А4988

**Технические характеристики A4988**

* напряжение питания: 8-35 В
* режим микрошага: 1, 1/2, 1/4, 1/8, 1/16
* напряжение логики: 3-5.5 В
* защита от перегрева
* максимальный ток на фазу: 1 А без радиатора; 2 А с радиатором
* размер: 20 х 15 мм

**Назначение контактов драйвера A4988**

* ENABLE – включение/выключение драйвера
* MS1, MS2, MS3 – контакты для установки микрошага
* RESET - cброс микросхемы
* STEP - генерация импульсов для движения двигателей (каждый импульс – шаг), можно регулировать скорость двигателя
* DIR – установка направление вращения
* VMOT – питание для двигателя (8 – 35 В)
* GND – общий
* 2B, 2A, 1A, 1B – для подключения обмоток двигателя
* VDD – питание микросхемы (3.5 –5В)

**USB порт**

Самый распространенный и встречаемый интерфейс в компьютерном оборудовании – **USB порт**. USB порт предназначен для подключения различного дополнительного оборудования с целью передачи цифровых данных с высокой скоростью. Современные USB порты также позволяют передавать электроэнергию, например, подключая смартфон к компьютеру, через USB можно передавать данные в обоих направлениях и при этом заряжать аккумулятор устройства.

[](https://xsoid.ru/wp-content/uploads/2014/03/usb-222222.jpg)

Рис. 3.13. Виды USB - портов

Существует несколько USB  стандартов, отличающихся скоростью передачи информации, в настоящий момент это [стандарты версий 1.0, 1,1,2,0 и 3,0](https://xsoid.ru/kak-uznat-tip-usb-porta-2-0-ili-3-0).  Также есть несколько типов интерфейсов, отличающихся конструкцией коннектора. Существуют 4 вида USB  портов, на которых показан на рисунке 3.13.

**Питание микроконтроллера Arduino**

**Arduino Uno**  получает питание или подключением USB, или от внешнего источника питания, которые выбираются автоматически.

Внешнее питание (не USB) подаётся двояко: или 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.

Рекомендуемый диапазон работы платформы 7-12 В, так как при напряжении питания ниже 7 В вывод 5V выдает менее 5 В, что приводит к нестабильной работе платформы. Использование напряжения выше 12 В может привести к перегреву, и повреждению платы.

**Дополнительный блок питания**

На рисунке 3.14. приведена схема блока питания. В нее включены гасящий конденсатор и гальваническая развязка от сети.

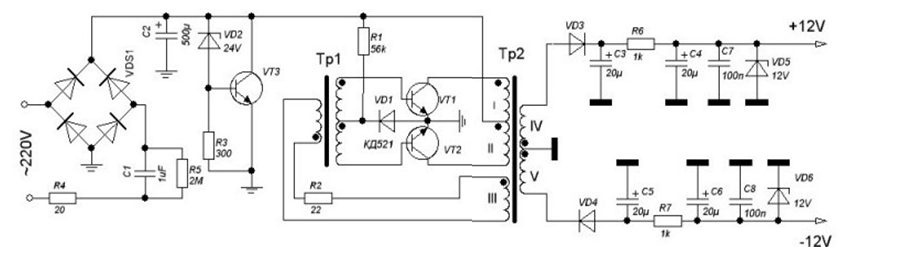
****

Рис. 3.14. Схема блока питания с гасящим конденсатором и гальванической развязкой от сети

Схема была многократно применена (не один десяток раз) и показала отличные результаты. Так как схема имеет подробное описание, поэтому здесь остановимся только на следующих характеристиках: сетевое напряжение через гасящий конденсатор C1 выпрямляется мостом VD1 и стабилизируется на уровне 24В стабилизатором на транзисторе VT3. От этого стабилизатора питание подается на генератор, выполненный на транзисторах VT1, VT2. На ферритовом кольце диаметром 20 мм. выполнен «Силовой» трансформатор Тр2, который на частоте 40…50 КГц способен выдать в нагрузку мощность до 7 ватт. Этой мощности достаточно для питания схемы. Электрическая безопасность схемы обеспечивается отсутствуем гальванической связи питаемой нагрузки с сетью благодаря присутствию развязывающего трансформатора VD5, VD6.

Примечание, оборудование изображено на схеме справа от сердечника трансформатора Тр2, находится под потенциалом сети требует аккуратно и осторожного обращения. На рисунке 3.15. представлена схема безопасного блока питания с гасящим конденсатором.

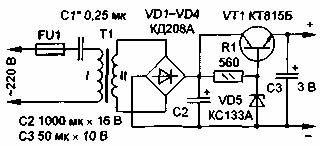


Рис. 3.15. Схема безопасного блока питания с гасящим конденсатором

Первичная обмотка трансформатора малогабаритных блоков питания содержит несколько (четыре…семь) тысяч витков сверхтонкого провода,- 0,05…0,06мм. Чтобы такую обмотку не мотать, предлагается с помощью гасящего конденсатора снизить напряжение на первичной обмотке до 30…40В. В этом случае первичная обмотка содержит не более 600…700 витков достаточно толстого провода (0,1…0,15мм), а вторичная обмотка стандартно рассчитывается на необходимое напряжение.

Трансформатор можно намотать на магнитопроводе Ш12\*15 от абонентского громкоговорителя, при помощи конденсатора C1 можно более точно подобрать значение напряжений. Использование трансформатора дает результат: гальванически развязан от сети. Мощности подобного блока питания достаточно для запитки элементарного микроконтроллера, чтобы Arduino работал. Напряжение питания - 9 В сила тока до 2 А.

**Ниже приведен код программы управления ШД.**

**Код для управления шаговым двигателем**

//простое подключение A4988

//пины reset и sleep соединены вместе

//подключите VDD к пину 3.3 В или 5 В на Arduino

//подключите GND к Arduino GND (GND рядом с VDD)

//подключите 1A и 1B к 1 катушке шагового двигателя

//подключите 2A и 2B к 2 катушке шагового двигателя

//подключите VMOT к источнику питания (9В источник питания + term)

//подключите GRD к источнику питания (9В источник питания - term)

int stp = 14; //подключите 14 пин к step

int dir = 13; //подключите 13 пин к dir

int a = 0;

void setup()

{

pinMode(stp, OUTPUT);

pinMode(dir, OUTPUT);

}

void loop()

{

if (a < 200) // вращение на 200 шагов в направлении 1

{

a++;

digitalWrite (stp, HIGH);

delay (11);

digitalWrite (stp, LOW);

delay (10);

}

else { digitalWrite(dir, HIGH);

a++;

digitalWrite (stp, HIGH);

delay (11);

digitalWrite (stp, LOW);

delay (10);

if (a>400) // вращение на 200 шагов в направлении 2

{

a = 0;

digitalWrite(dir, LOW);

}

}

}

При изготовлении опытного прототипа 3D-принтера взят ШД марки ДС4-6В и винтовой редуктор с длиной винта 84 мм и диаметром 3 мм.

При эксперименте установлен, что винтовой редуктор за одиночный импульс перемещает нагрузку на 1,6 мм. Значит, .

На фотографии (Рис 3.16) показан фото внешний вид разработанного прототипа 3D-принтера.

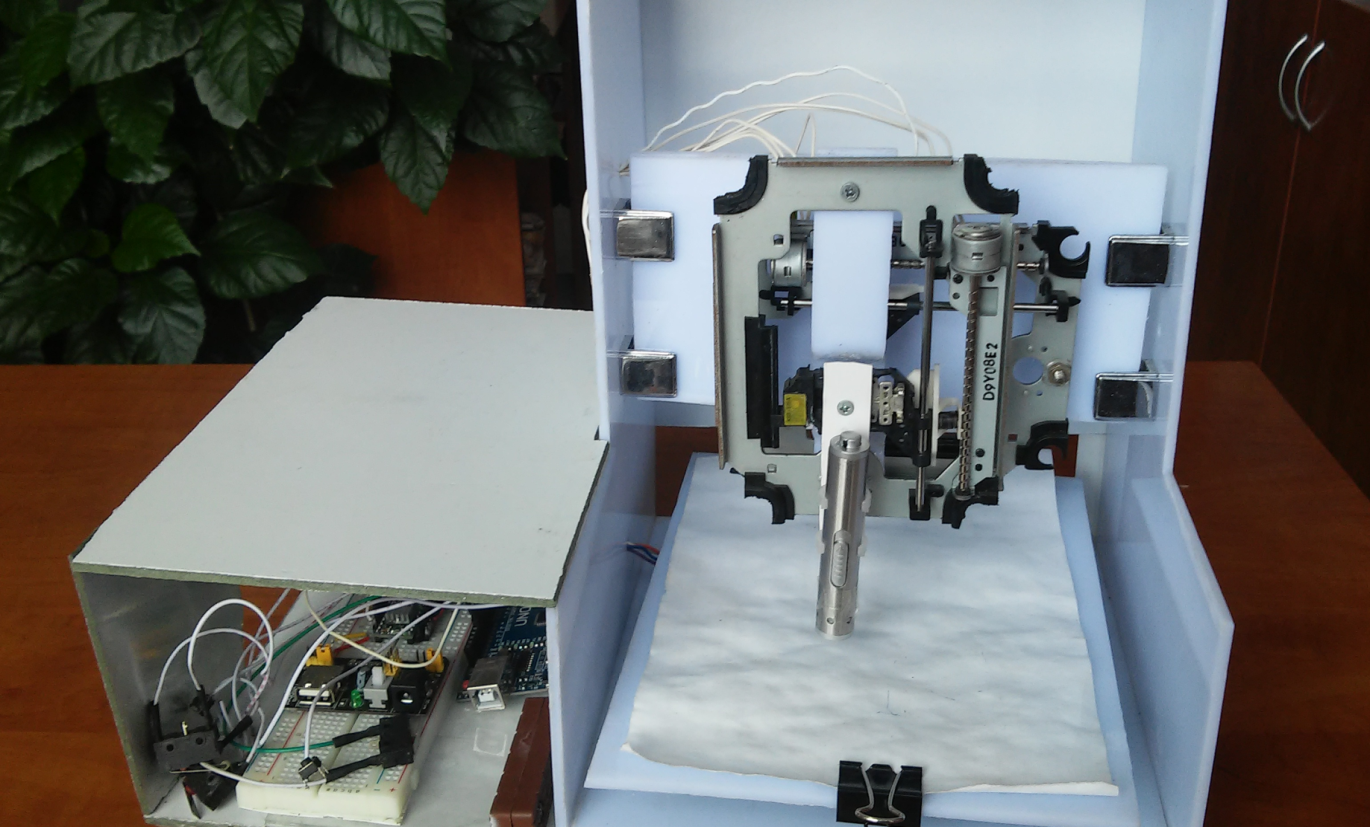


Рис. 3.16. Внешний вид макета прототипа 3D-принтера

Работоспособность опытного прототипа 3D-принтера доказана проведёнными экспериментальными испытаниями.

**3.5. Программное обеспечение системы управления 3D—принтера**

Учитывая, что среда Delphi имеет ряд преимуществ простата синтаксиса, удачная демонстрация парадигмы процедурного программирования и объектно – ориентированного программирования, проста и понятность архитектуры стандартных библиотек она была использована в качестве языке программирования [109]. Ниже приведено руководство по созданию 3D- модели в программе “JCNC”, а в приложении 4 - листинг программы для управления 3D-принтером.

**3.5.1. Руководство по эксплуатации 3D-принтера**

После установки драйвера открываем папку Arduino и запускаем установщик драйвера (драйвер 64).

Далее открываем программу c названием Хloader, интерфейс которой представлен на рисунке 3.17.

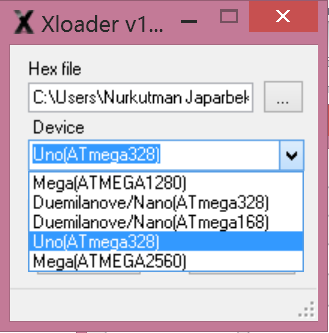
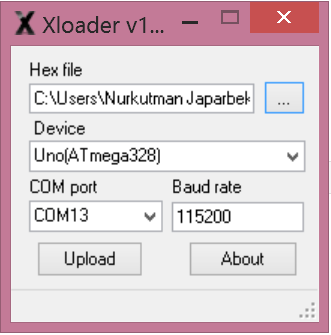


Рис.3.17. Интерфейс Хloader

Далее указываем путь к зашифрованному драйверу в папке Arduino с названием «SVOI.hex». Во вкладке «устройство» (device) выбираем устройство ( в нашем случае arduino Uno atmega 328) и указываем порт, к которомом был подсоединен наш 3D - принтер.

Для определения порта нужно зайти в компьютере меню «диспетчер устройств» и посмотреть, на каком порте была подсоединена плата Arduino.

Затем вводим частоту для подключения (введено автоматически 115200) и нажимаем на кнопку Upload (загрузить). Ждем, успешной загрузки драйвера.

Для управления 3D - принтером устанавливаем программу “JCNC” (выбор на эту программу, как наиболее подходящую). Далее открываем саму программу (Рисс. 3.18).

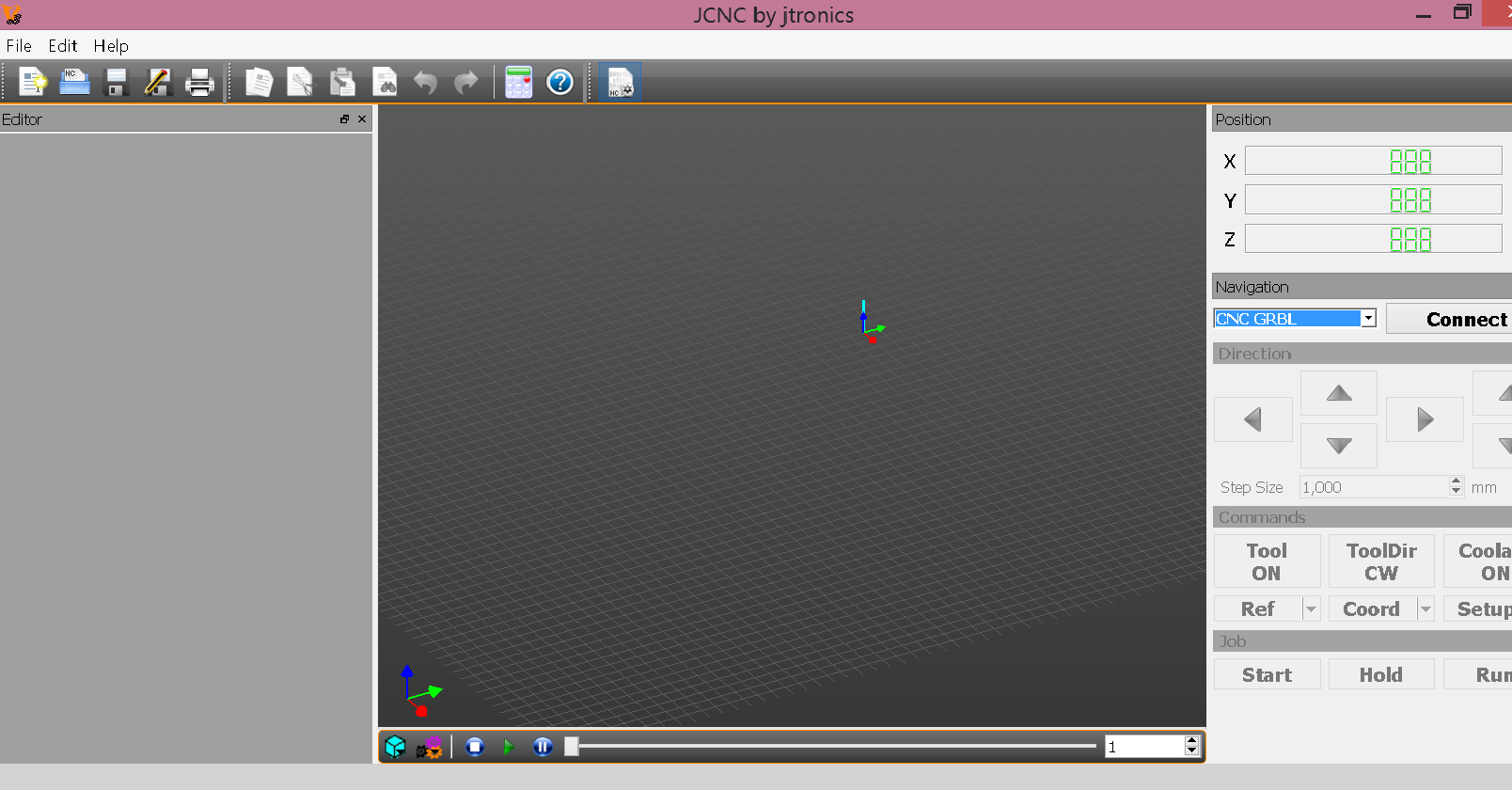


Рис. 3.18. Рабочее окно программы “JCNC”

На правой стороне вкладке навигации (navigation) подключаем наш 3D-принтер. После выбора типа подключения «CNC GRBL» нажимаем на кнопку подключения «connect».

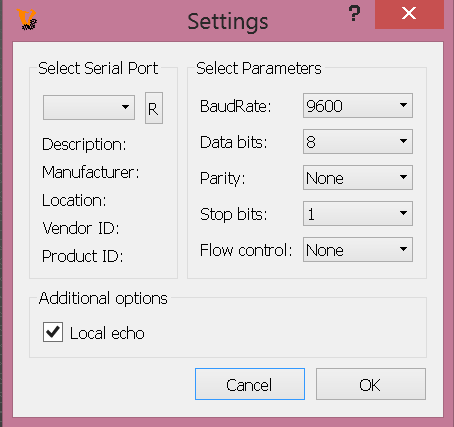


Рис. 3.19. Окно настройки параметров

Появляется окно надстройки параметров, подключения и вкладка выбора серийного порта. Отсюда мы выбираем нужную нам плату, зарегистрированный тип которой можно посмотреть в диспетчере устройств. Также в окне «настройка параметров» можно установить частоту подключения, разрядность подключения, поддержку контроля (Рис. 3.19). После нажатия «ОК» 3D - принтер подключён к компьютеру.

Далее создаём 3D - модель для печати. Можно использовать любое программное обеспечение, но для примера используем программу “Blender”, интерфейс которой изображён на рисунке. 3.20.

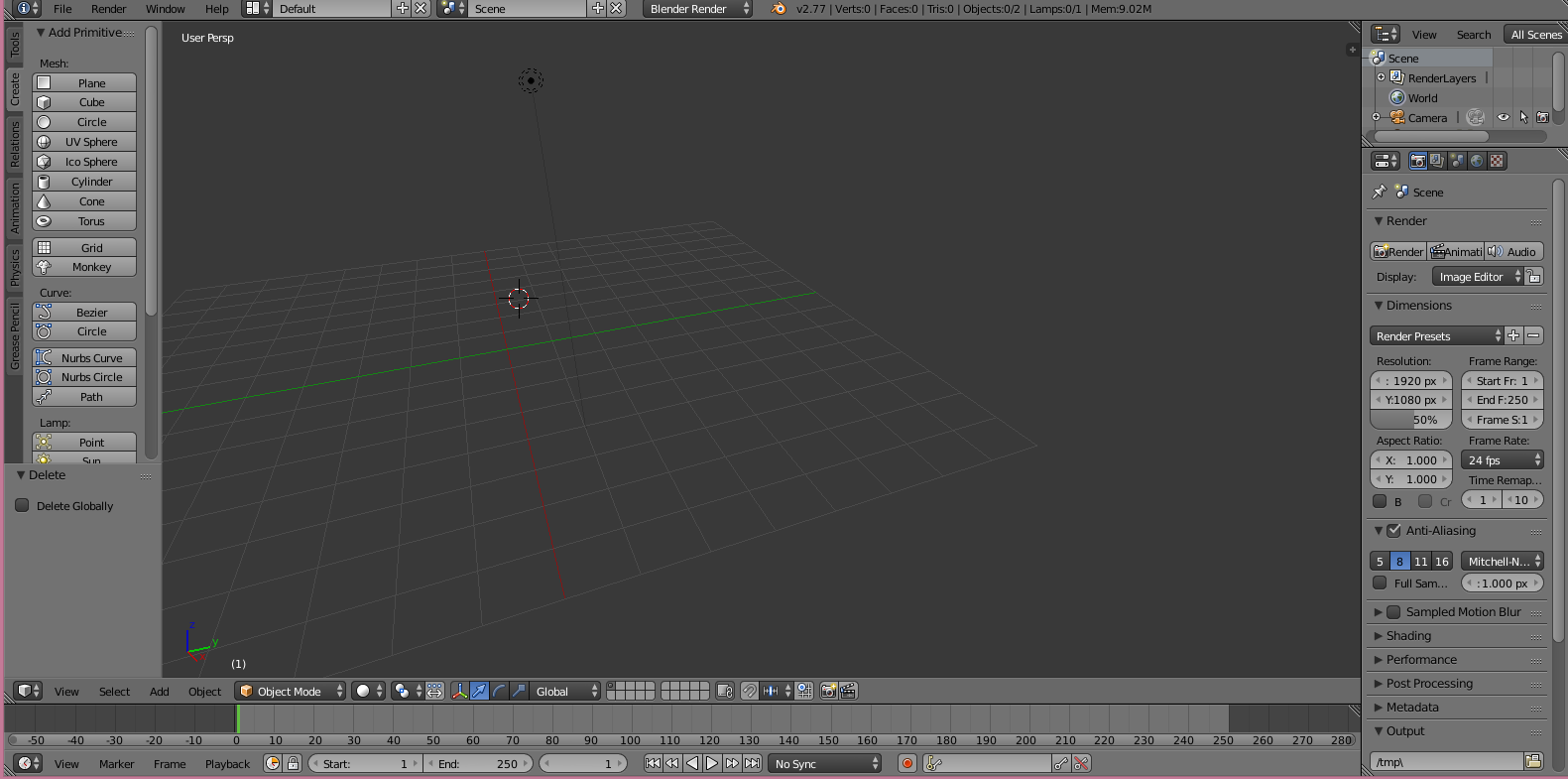


Рис. 3.20. Интерфейс “Blender”

Заходим во вкладку “File” -> “New”: создали новое 3D пространство. В слева в меню нужно выбрать “Create” для создания детали или можно выбрать нужный экземпляр 3D моделей из существующих (Рис. 3.21).

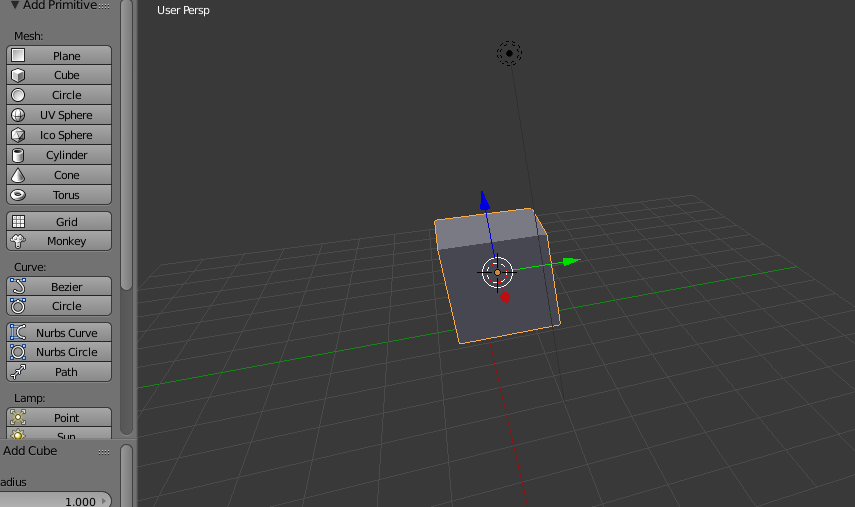


Рис. 3.21. Интерфейс 2 “Blender”

В нашем случае, мы выбираем куб и видим готовую 3D - модель из меню вкладки “File” -> ”Export” выбираем нужный формат для экспортирования файла. В нашем случае это - .stl (Рис. 3.22).

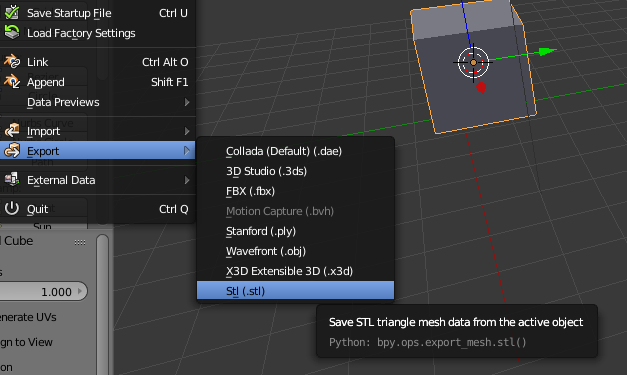


Рис. 3.22. Интерфейс экспортирование файла

Сохраняем формат, выбрав предварительно путь сохранения.

Теперь открываем программу “slic3r” (Рис. 3.23).

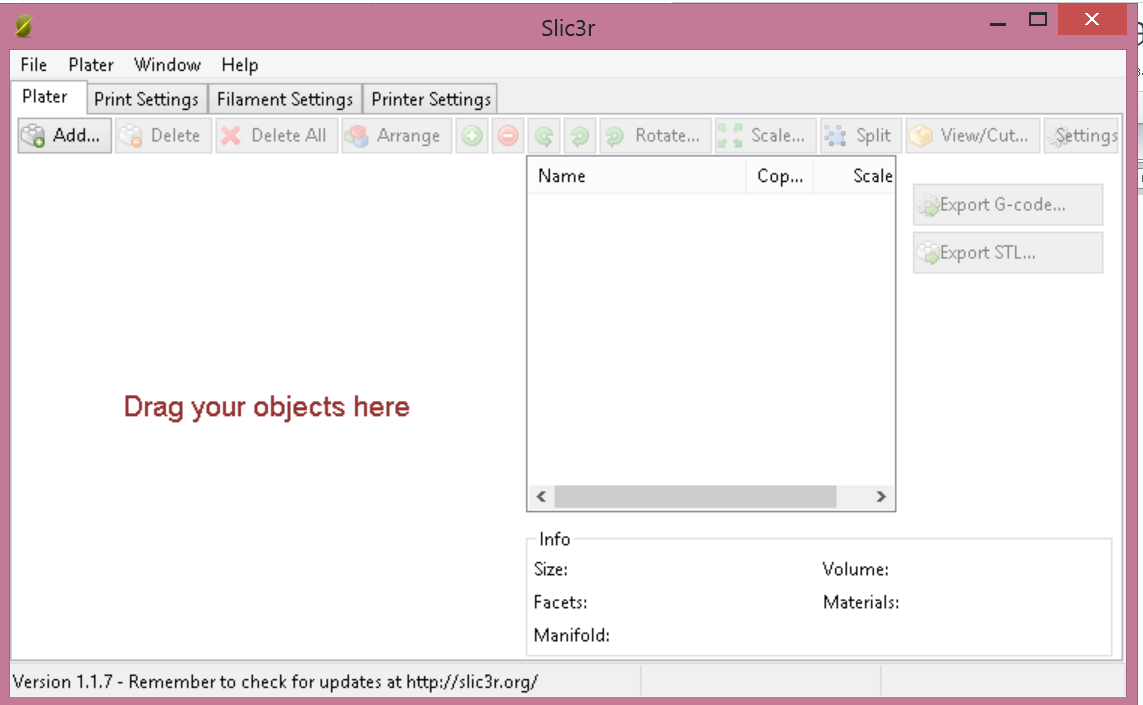


Рис. 3.23. Интерфейс программу “slic3r”.

С помощью кнопки добавить “add” находим прежде сохраненный .stl файл. Теперь мы должны изменить размеры куба в соответсвии с размером 3D - принтера.

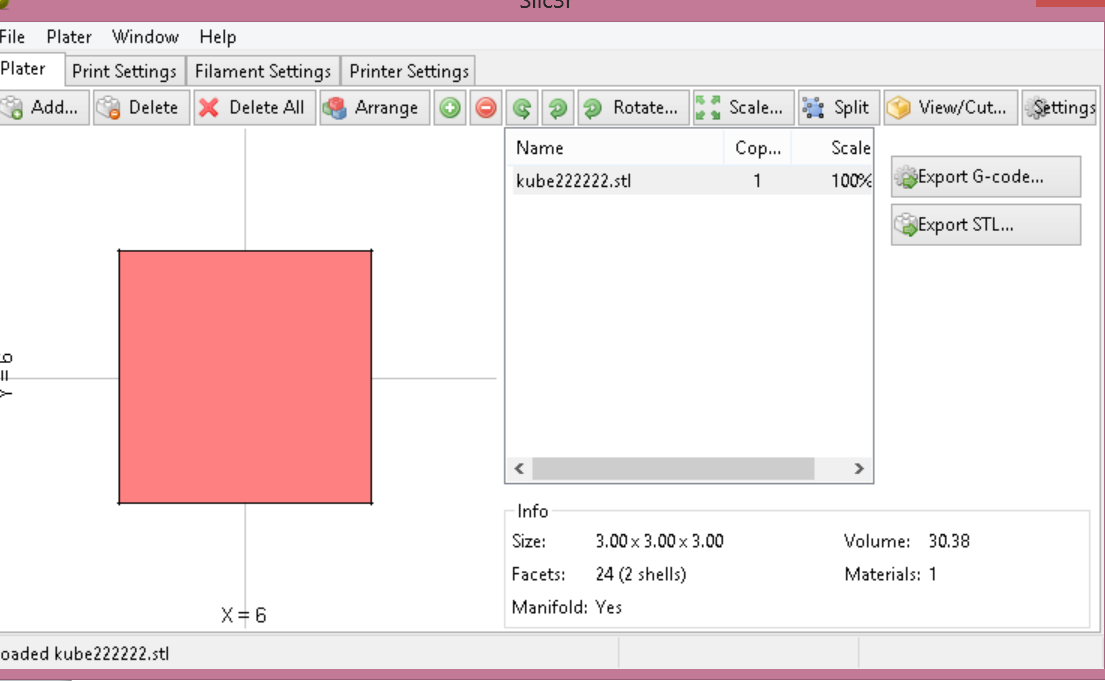


Рис. 3.24. Интерфейс экспорта в G-код.

После надстроек мы экспортируем данную готовую модель в машинный язык G-код (Рис. 3.24). Для этого нажимаем на кнопку “Export G-code” и выбираем путь для сохранения.

После выполненных выше описнных процедур в программе «JCNC» открываем скомпилированный G-code файл (Рис. 3.25).

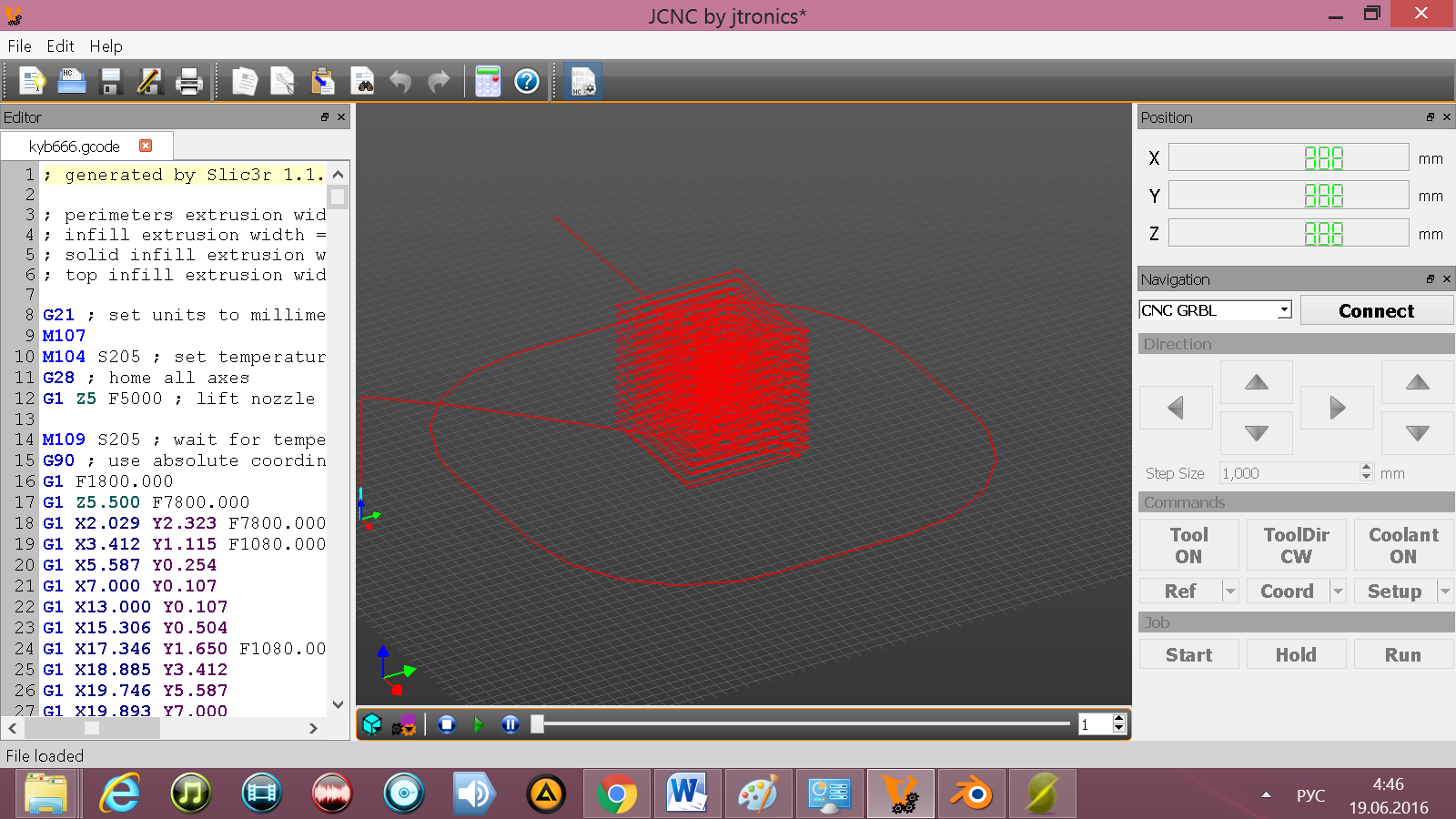


Рис. 3.25. Интерфейс «JCNC»

Во вкладке “Job” нажимаем на кнопку старт («start») и запускаем («Run») программу.

**ВЫВОДЫ**

В этой главе приведены обоснованные методики синтеза законов управления шаговыми приводами по осуществлению движения рабочего органа 3-х звенного манипулятора по предписанной траектории, заданной в табличной форме, и методика синтеза алгоритмов управления ШД в прототипе 3D-принтера.

Описана конкретная техническая реализация синтезированных алгоритмов управления ШД на примере прототипа 3D-принтера.

Представлены программное обеспечение разработанной системы управления и руководство по эксплуатации 3D-принтера.

Программное обеспечение разработанной системы управления приведено в Приложении 4.

**ЗАКЛЮЧЕНИЕ**

В результате диссертационных исследований разработана теория и конструктивные процедуры синтеза управления по осуществлению движения объекта по предписанной траектории:

* подход синтеза в случаях задания предписанной траектории движения в аналитической форме;
* подход синтеза в случаях задания предписанной траектории движения в табличной форме;
* для линейного многомерного объекта, подверженного параметрическим возмущениям, разработан подход к синтезу адаптивного управления траекторным движением;
* для рассматриваемых классов управляемых объектов разработан критерий осуществимости траектории движения на языке ранга соответствующих матриц;
* разработаны математические модели шаговых двигателей, применяемых в 3D-принтерах, как объектов управления.

Разработанные теоретические положения апробированы на множестве модельных задач, а также при решении конкретных прикладных задач.

Результаты моделирования на ПЭВМ, а также натурные испытания разработанного прототипа 3D-принтера полностью подтверждают положения разработанной теории.

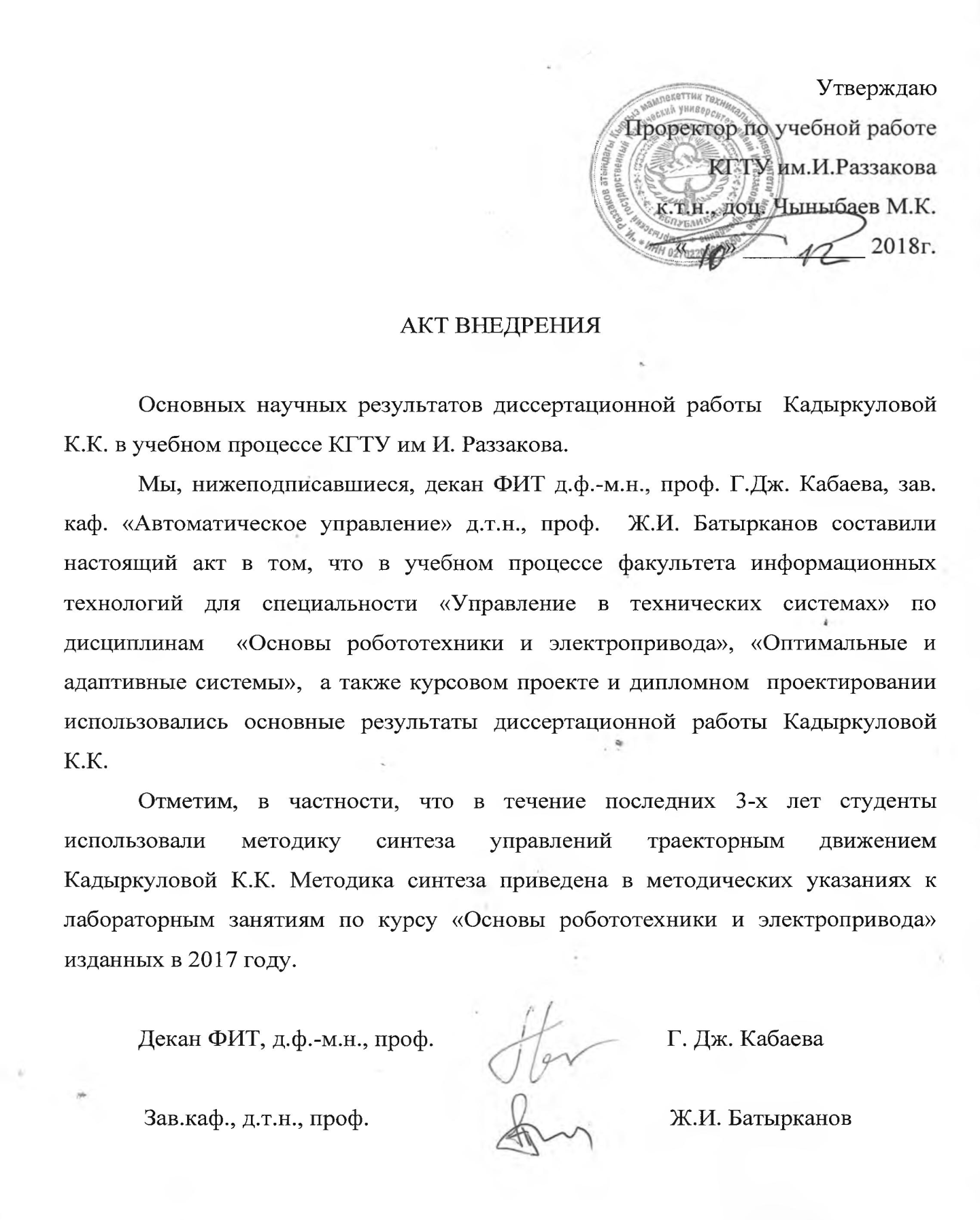
Разработанный подход к синтезу в случае задания предписанной траектории в табличной форме позволяет синтезировать законы управления приводами 3D-принтера для деталей любых форм, изготавливаемых на 3D-принтере.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Айтчанов Б.Х. Технологический процесс вытяжки оптических стержней как объект автоматизированного управления [Текст] / Б.Х. Айтчанов // Доклады Национальной академии наук РК. – Алматы. – 2017. – №2. – С. 91-95.
2. Александров А.Г. Синтез регуляторов многомерных систем [текст] / А.Г. Александров. – М.: Машиностроение, 1985. – 272 с.
3. Андриевский Б. Р.Анализ систем в пространстве состояний [текст] / Б.Р. Андриевский. – СПб.: ИПМаш РАН, 1997. – 267 с.
4. Барбашин Е.А. Введение в теорию устойчивости [текст] / Е.А. Барбашин. – М.: Наука, 1967. – 224 с.
5. Барбашин Е.А. Функции Ляпунова [текст] / Е.А. Барбашин. – М.: Наука, 1970. – 240с.
6. Батырканов Ж.И., Кадыркулова К.К., Белялов Ш.А. Управление шаговым приводом 3d – принтера: математическая модель и вопросы технической реализации [текст] / Ж.И. Батырканов, К.К. Кадыркулова, Ш.А. Белялов // Прикаспийский журнал: управление и высокие технологии. Научно-технический журнал, Издательский дом «Астраханский университет». – 2016. – № 1 (33) . – С. 128-138.
7. Батырканов Ж.И., баймухамедов М.Ф., Кадыркулова К.К. Об одном подходе адаптивного управления траекторным движением [текст] / Ж.И. Батырканов, М.Ф. баймухамедов, К.К. Кадыркулова //Известия КГТУ им. И. Раззакова. – 2017. – № 1(41). – С. 110-117.
8. Батырканов Ж.И., Кадыркулова К.К. Синтез законов управления для осуществлению движения объекта по предписанной программе [текст] / Ж.И. Батырканов, К.К. Кадыркулова // Прикаспийский журнал: управление и высокие технологии. Научно-технический журнал, Издательский дом «Астраханский университет». – 2015. – №1 (29). – С. 143-154.
9. Батырканов Ж.И., Кадыркулова К.К. Синтез законов управления по осуществлению движения по предписанной программе [текст] / Ж.И. Батырканов, К.К. Кадыркулова // Вестник науки КСТУ им. академика З. Алдамжар, Казакстан. – 2014. – №3. – С. 40-46.
10. Батырканов Ж.И., Кадыркулова К.К., Маматбеков И.М. Разработка системы управления шаговыми приводами 3-х звенного манипулятора [Электронный ресурс] / Ж.И. Батырканов, К.К. Кадыркулова, И.М. Маматбеков// Огарев-online. – 2018. – №13. – Режим доступа: http://journal.mrsu.ru/arts/razrabotka-sistemy-upravleniya-shagovymi-privodami-trexzvennogo-manipulyatora
11. Батырканов Ж.И., Кармышаков А.К. Синтез алгоритма траекторного управления нелинейным электроприводом постоянного тока [Приборы и системы. Управление, контроль, диагностика] / Ж.И. Батырканов, А.К. Кармышаков // – М.: ООО Издательство «Научтехлитиздат». – 2006. – №10. – С. 24-27.
12. Батырканов Ж.И., Об одном методе оптимальной стабилизации нелинейных автономных систем управления [текст] / Ж.И. Батырканов // Тезисы докл. II Всесоюзного межвуз. Науч. техн. конф. Математическое, алгоритмическое и техническое обеспечение АСУТП. – Ташкент. – 1980. – С. 101-102.
13. Батырканов Ж.И., Шаршеналиев Ж.Ш. Синтез систем управления с заданными показателями качества [текст] / Ж.И. Батырканов, Ж.Ш. Шаршеналиев. – Бишкек: Илим, 1991. – 174с.
14. Башарин А.В., Голубев Ф.Н., Кепперман В.Г. Примеры расчётов автоматизированного электропривода [текст] / А.В. Башарин, Ф.Н. Голубев, В.Г. Кепперман. – Л.: Энергия, 1971. – 440 с.
15. Башарин А.В., Новиков В.А., Соколовский Г.Г. Управление электроприводами [текст] / А.В. Башарин, В.А. Новиков, Г.Г Соколовский. – Л.: Энергоатомиздат, 1982. – 392 с.
16. Башарин А.В., Постников Ю.В. Примеры расчета автоматизированного электропривода на ЭВМ [текст] / А.В. Башарин, Ю.В. Постников. – Л.: Энергатомиздат, 1990. – 512 с.
17. Бейсенби М.А., Ускенбаева Г. А. [Метод функции АМ Ляпунова в исследовании робастной устойчивости линейных систем управления с одним входом и одним выходом](http://scholar.google.com/scholar?cluster=4525266813783576338&hl=en&oi=scholarr) [текст] / М.А.Бейсенби, Г.А. Ускенбаев // Общество с ограниченной ответственностью «Центр развития научного сотрудничества». – 2013. – С. 27-32.
18. Беллман Р. Введение в теорию матриц [текст] / Р. Беллман. – М.: Наука, 1969. – 368 с.
19. Бесекерский В.А., Изранцев В.В. Системы автоматического управления с микроЭВМ [текст] / В.А. Бесекерский, В.В. Изранцев. – М.: Наука, 1987. – 320 с.
20. Бесекерский В.А., Попов Е.П. Теория систем автоматического регулирования [текст] / В.А. Бесекерский, Е.П. Попов. – М.: Наука, 1975. – 768 с.
21. Бойчук Л.М. Синтез координирующих систем автоматического управления [текст] / Л.М. Бойчук. – М.: Энергоатомиздат, 1992. – 160 с.
22. Болтянский В.Г. Математические методы оптимального управления [текст] / В.Г. Болтянский. – М.: Наука, 1966. – 307 с.
23. Булычев Ю.Г., Бурлый И.В., Манин А.А. Аналитическое конструирование систем управления на основе метода опорных интегральных кривых [текст] / Ю.Г. Булычев, И.В. Бурлый, А.А. Манин // Автоматика и телемеханика. – 1994. – №7, – С. 37- 48.
24. Вавилов А.А. Частотные методы расчёта нелинейных систем [текст] / А.А. Вавилов. – Л.: Энергия, 1970. – 324 с.
25. Воронов А. А. Основы теории автоматического управления [текст] / А. А. Воронов. – М. – Л.: Энергия, 1965. – 396 с.
26. Галиулин А.С. Методы решения обратных задач динамики [текст] / А.С. Галиулин. – М.: Наука, 1986. – 224 с.
27. Галиулин А.С. Обратные задачи динамики [текст] / А.С. Галиулин. – М.: Наука, 1981. – 143 с.
28. Галиулин А.С., Мухаметзянов И.А., Мухарлямов Р.Г., Фурасов В.Д. Построение систем программного движения [текст] / А.С. Галиулин, И.А. Мухаметзянов, Р.Г. Мухарлямов, В.Д. Фурасов. – М.: Наука, 1971. – 352 с.
29. Герман-Галкин С.Г. Компьютерное моделирование полупроводниковых систем в Matlab 6.0: Уч. пособие [текст] / Герман-Галкин С.Г. – СПб.: Питер, 2001. – 320 с.
30. Григорьев В.В., Дроздов В.Н., Ушаков А.В. Синтез дискретных регуляторов при помощи ЭВМ [текст] / В.В. Григорьев, В.Н. Дроздов, А.В Ушаков. – Л.: Машиностроение, 1983. – 245 с.
31. Гультяев А. Визуальное моделирование в среде MATLAB: учебный курс [текст] / А. Гультяев. – СПб.: Питер, 2000.
32. Джолдошев Б.О. Синтез управляющих устройств для нелинейных систем с учётом заданных инженерных показателей качества [текст] / Б.О. Джолдошев // Известия КГТУ. – 2009. – №19. – С.136-140.
33. Дмитриев А.Н., Егупов Н.Д., Шаршеналиев Ж.Ш. Спектральные методы анализа, синтеза и идентификации систем управления [текст] / А.Н. Дмитриев, Н.Д. Егупов, Ж.Ш. Шаршеналиев. – Фрунзе: Илим, 1986. – 229 с.
34. Душин С.Е., Имаев Д.Х., Моисеев С.С. Структурно-параметрический синтез нелинейной системы управления по заданному движению [Текст] / С.Е. Душин, Д.Х. Имаев, С.С. Моисеев. – С. Петербург, 1991.
35. Дьяконов В. П., Круглов В. Математические пакеты расширения MATLAB [Текст] / В. П. Дьяконов, В. Круглов. Специальный справочник. – СПб.: Питер, 2001.
36. Егоренков Д.Л., Фрадков А.Л., Харламов В.Ю. Основы математического моделирования [Текст]: Учебное пособие / Д.Л. Егоренков, А.Л. Фрадков, В.Ю. Харламов. – СПб.: БГТУ, 1994.
37. Емельянов С.В. Бинарные системы автоматического управления [Текст] / С.В. Емельянов. – М.: МНИИПУ, 1984.
38. Емельянов С.В. Системы автоматического управления с переменной структурой [Текст] / С.В. Емельянов. – М.: Наука, 1967. – 336 с.
39. Емельянов С.В., Коровин С. К., Сизиков В.И. Бинарные системы управления нестандартными процессами с применением адаптивных и мультипликативных связей [Текст] / С.В. Емельянов, С. К. Коровин, В.И. Сизиков. – М.: МНИИПУ, 1983.
40. Ермолин Н.П. Электрические машины малой мощности [Текст] / Н.П. Ермолин. – М.: Высш. школа, 1967. – 503 с.
41. Еругин Н.П. Построение всего множества системы дифференциальных уравнений, имеющих заданную интегральную кривую [текст] / Н.П. Еругин // ПММ. – 1952. – Вып.2. – С. 651-658.
42. Живоглядов В.П. Адаптация в автоматизированных систем управления технологическими процессами [текст] / В.П. Живоглядов. – Фрунзе: Илим, 1974. – 228 с.
43. Живоглядов В.П. Интегрированные и многоуровневые системы управления производством [текст] / В.П. Живоглядов. – Фрунзе: Илим, 1980. – 146 с.
44. Зубов В.А. Лекции по теории управления [текст] / В.А. Зубов. – М.: Наука, 1976.
45. Иванов В.А., Фалдин Н.В. Теория оптимальных систем автоматического управления [текст] / В.А. Иванов, Н.В. Фалдин. – М.: Наука, 1981.
46. Иващенко Н.Н. Автоматическое регулирование. Теория и элементы систем [текст] / Н.Н. Иващенко – М.: Машиностроение, 1978. – 736 с.
47. Ильин Ю.П. и др. Структуры системы управления автоматизированными электроприводами [текст] / Ю.П. Ильин и др. – М.: Наука и техника, 1978. – 368с.
48. Карпенко Б.К., Ларченко В.И., Прокофьев Ю.А. Шаговые электродвигатели [текст] / Б.К. Карпенко, В.И. Ларченко, Ю.А. Прокофьев. – Киев: Изд-во «Технiка», 1972. – 216 с.
49. Кенио Т. Шаговые двигатели и их микропроцессорные системы управления [текст]: Пер. с англ. / Т. Кенио. – М.: Энергоатомиздат, 1987. – 200 с.
50. Клиначёв Н. В. Теория систем автоматического регулирования и управления [электронный ресурс]: Учебно-методический комплекс / Н. В. Клиначёв – Offline версия 1.8. – Челябинск, 2002. – 550 файлов.
51. Ковчин.С.А., Сабинин Ю.А. Теория электропривода [текст] / С.А. Ковчин, Ю.А. Сабинин. – С. Петербург: Энергоатомиздат,1994.
52. Кондратьева Л.А. Аппроксимационная обратная задача для предельных циклов [текст] / Л.А. Кондратьева // Качественное и численное исследование математических моделей динамических систем. Межвуз. сб. научн. тудов. М.: Изд-во РГОТУПС, – 2006. – С.72-75.
53. Красовский Н.Н. Теория управления движением [текст] / Н.Н. Красовский. – М.: Наука, 1978. – 476 с.
54. Крищенко А.П. Метод обратной задачи динамики в теории управления [текст] / А.П. Крищенко // ВСПУ, Москва. –2014. – С. 431-437.
55. Крутько П.Д. Обратные задачи динамики управляемых систем: линейные модели [текст] / П.Д. Крутько. – М.: Наука, 1987. – 304 с.
56. Крутько П.Д. Обратные задачи динамики управляемых систем: нелинейные модели [текст] / П.Д. Крутько. – М.: Наука, 1988. – 328 с.
57. Крутько П.Д. Оптимизация многомерных динамических систем по критерию минимума энергии ускорения [текст] / П.Д. Крутько // Изв. РАН. Техническая кибернетика. – 1994. – №1. – С. 32-47.
58. Крутько П.Д. Управление исполнительными системами роботов [текст] / П.Д. Крутько. – М.: Наука, 1991. – 405 с.
59. Крутько П.Д., Лакота Н.А. Метод обратных задач динамики в теории конструирования алгоритмов управления манипуляционных роботов. Осуществление назначенных траекторий [текст] / П.Д. Крутько, Н.А. Лакота // Изв. АН СССР. Техн. Кибернетика. – 1987. – №4.
60. Крутько П.Д., Максимов А.И., Скворцов Л.М. Алгоритмы и программы проектирования автоматических систем [текст] / П.Д. Крутько, А.И. Максимов, Л.М. Скворцов. – М.: Радио и связь, 1988. – 306 с.
61. Кузовков Н.Т. Модальные уравнения и наблюдающие устройства [текст] / Н.Т. Кузовков. – М.: Машиностроение, 1976. – 184 с.
62. Кунцевич В.М., Лычак М.М. Синтез систем автоматического управления с помощью функций Ляпунова [текст] / В.М. Кунцевич, М.М. Лычак. – М.: Наука, 1977. – 400с.
63. Лейбов А.М., Каменев Р.В., Осокина О.М. Применение технологий 3D- прототипирования в образовательном процессе [текст] / А.М. Лейбов, Р.В. Каменев, О.М. Осокина // Современные проблемы науки и образования. – 2014. №5. – С.93.
64. Лихоманов А.М., Панин С.Ю., Писарев А.Ю., Мякишев В.В. Синтез систем стабилизации скорости на основе частотного подхода к решению обратных задач динамики [текст] / А.М. Лихоманов, С.Ю. Панин и.др. // Изв. вузов. Приборостроение. – 2000. – №7. – С. 24-27.
65. Ломака М.В., Медведев И.В. Микропроцессорное управление приводами промышленных роботов [текст] / М.В. Ломака, И.В. Медведев. – М.: Машиностроение, 1990. – 96 с.
66. Ляпунов А.М. Общая задача об устойчивости движения [текст] / А.М. Ляпунов. – М. -Л.: ГИТТЛ, 1950. – 472 с.
67. Медведев В.С. Управляющие и вычислительные устройства роботизированных комплексов на базе микроЭВМ / В.С. Медведев и др. – М.: Высш. шк., 1990. – 239 с.
68. Мещерский И.В. Механика тел переменной массы. Избранные труды [текст] / И.В. Мещерский. – М.: Издательство Юрайт, 2017. – 233с.
69. Миркин Е.Л, Шаршеналиев Ж.Ш. Адаптивные системы управления сложными объектами [текст] / Е.Л. Миркин Е., Ж.Ш. Шаршеналиев. – LAP LAMBERT, Academic Publishing, Германия, 2012. –184 с.
70. Мирошник И.В., Шалаев А.Н. управление траекторным движением автономных роботов [текст] / И.В. Мирошник, А.Н. Шалаев // ТОМ 18.– 2018 . –– № 5. – С. 243-248.
71. Москаленко В.В. Системы автоматизированного управления электропривода [текст] / В.В. Москаленко. – М.: ИНФРА-М, 2004. – 208 с.
72. Оморов Т.Т., Шаршеналиев Ж.Ш. управление многомерными объектами на основе концепции допустимости [текст] / Т.Т. Оморов, Ж.Ш. Шаршеналиев. – Бишкек: Илим, 1996. – 160 с.
73. Пестерев А.В. Синтез стабилизующего управления в задаче следования колёсного робота вдоль заданной кривой / А.В. Пестерев // Автоматика и Телемеханика. – 2012. – №7. – С. 25-39.
74. Петров Б.Н. Теория автоматического управления: [текст] / Избранные труды/ Б.Н. Петров. – Т1. – М.: Наука, 1983. – 430 с.
75. Петров Ю.П. Оптимальное управление с электрическим приводом с учётом ограничений по нагреву [текст] / Ю.П. Петров. – Л.: Энергия, 1971. – 144 с.
76. Прангишвили И.В. Микропроцессоры и локальные сети микроЭВМ в распределённых системах управления [текст] / И.В. Прангишвили. – М.: Энергоатомиздат, 1985. – 272 с.
77. Пятницкий Е.С., Дунская Н.В. Стабилизация управляемых механических и электромеханических систем [текст] / Е.С. Пятницкий, Н.В. Дунская // Автоматика и телемеханика, – 1988. – № 12, – С. 40-51.
78. Смирнов Е.Я., Управление движением механических систем [текст] Учебное пособие / Е.Я. Смирнов. – Л., изд-во ЛГУ. – 1988. – 81 с.
79. Солодовников В.В. Синтез корректирующих устройств систем автоматического регулирования [текст] / В.В. Солодовников // техническая кибернетика. Теория автоматического регулирования. – М.: 1967. – С. 303-347.
80. Соломенцева Ю.М. Теория автоматического управления [текст] / В. Н. Брюханов, М.Г. Косов и др.; Под ред. Соломенцева Ю.М. – М.: Высшая школа, 2000 г.
81. Справочник по автоматизированному электроприводу/ Под ред. В.А. Елисеева и А.В. Шинянского. – М.: Энергоатомиздат, 1983. – 616 с.
82. Справочник по теории автоматического управления / Под ред. А.А. Красовского. – М.: Наука, 1987. – 712 с.
83. Справочник проектировщика АСУ ТП / Под ред. Г.Л. Смилянского. М.: Машиностроение, 1983. – 527 с.
84. Срибнер Л.А. Программируемые контроллеры, их особенности и тенденция развития [текст] / Л.А. Срибнер // Управляющие системы и машины, 1978. – №2. – С.86-91.
85. [Сыздыков Д.Ж.](https://istina.msu.ru/workers/92379192/), [Байбатышев М.Ш.](https://istina.msu.ru/workers/92379193/), [Бейсембаев А.А.](https://istina.msu.ru/workers/92379195/) Синтез программной траектории манипуляционного робота на основе описания рабочих зон с применением R-функций [текст] / Д.Ж. [Сыздыков](https://istina.msu.ru/workers/92379192/), М.Ш [Байбатышев](https://istina.msu.ru/workers/92379193/), А.А. [Бейсембаев](https://istina.msu.ru/workers/92379195/) // МИРЭиА, Москва. – 1990. – С. 21-29.
86. Сю Д., Мейер А. Современная теория автоматического управления и ее применение [текст] / Д. Сю, А. Мейер.– М.: Машиностроение, 1972.
87. Тимофеев А.В. Построение адаптивных систем управления программным движением [текст] / А.В.Тимофеев. – Л.: Энергия, 1988. – 88 с.
88. Тольга А.В., Воронина А.Д Развитие информационных технологий. Трёхмерная печать объектов [текст] / А.В. Тольга, А.Д. Воронина // В сборнике: Перспективы развития науки и образования сборник научных трудов по материалам Международной научно-практической конференции. Тамбов. – 2014. – С. 150-151.
89. Уткин В.И. Скользящие режимы в задачах оптимизации и управления [текст] / В.И. Уткин. – М.: Наука, 1981.
90. Федорков Б.Г., Телец В.А. Микросхемы ЦАП и АЦП: функционирование, параметры, применение [текст] / Б.Г. Федорков, В.А. Телец. – М.: Энергоатомиздат, 1990. – 320 с.
91. Фурасов В.Д. Устойчивость движения, оценки и стабилизация [текст] / В.Д. Фурасов. – М.: Наука и техника, 1987.
92. Чаки Ф. Современная теория управления. Нелинейные, оптимальные и адаптивные системы [текст]: пер. с англ. / Ф. Чаки. – М.: Мир, 1975. – 424 с.
93. Чиликин М.Г., Ключев В.И., Сандлер А.С. Теория автоматизированного электропривода [текст] / М.Г. Чиликин, В.И. Ключев, А.С. Сандлер. – М.: Энергия, 1979. – 619 с.
94. Шаршеналиев Ж.Ш., Батырканов Ж.И., Кармышаков А.К. Траекторное управление электромеханическим объектом [текст] / Ж.Ш. Шаршеналиев, Ж.И. Батырканов, А.К. Кармышаков // Материалы международной научной конференции «Образование через науки». – Бишкек, КТУ им. И.Раззакова. –2004. – С. 64-66.
95. Юрков А.В. Стабилизация программных движений управляемых систем [текст] / А.В. Юрков // Известия РАЕН, серия ММИУ. – 1999. – том 3. – С.107-120.
96. Янтовский А.В., Беленький Э.И. Управление автоматическими линиями с гибкими и жесткими связями на базе программируемых контроллеров [текст] / А.В. Янтовский, Э.И. Беленький. – М.: НИИМАШ, 1983. – 68 с.
97. [Aaron Council](http://www.amazon.com/Aaron-Council/e/B00IDYVDTC/ref=dp_byline_cont_ebooks_1),  [Michael Petch](http://www.amazon.com/Michael-Petch/e/B00IPRWZJY/ref=dp_byline_cont_ebooks_2). 3D Printing: Rise of the Third Industrial Revolution. – Publisher: gyges3d.com. – 2014. – 116 p.
98. Batyrkanov Zh., Kadyrkulova K., Belialov Sh. Management stepper of 3d – printer [electronic resource] / Zh. Batyrkanov, K. Kadyrkulova, Sh. Belialov // India. – 2016.
99. Baymuhamedov M.F. Information systems [text] / M.F. Baymuhamedov. – «Бастау». – Almaty. – 2016. – 288 p.
100. Brogan W.L. Modern Control Theory [text] / W.L. Brogan. – 3rd ed. Englewood Cliffs, NJ: Prentice Hall. –1991.
101. Canessa E., Fonda C., Zennaro M. Low – cost 3D Printing for Science, Education and Sustainable Development [text] / E Canessa, C. Fonda, M. Zennaro. – Low Publisher: [ICTP – The Abdus Salam International Centre for Theoretical Physics](https://www.livelib.ru/publisher/28674). – 2013 – 119 p.
102. Charles L. Phillips, Royce D. Harbor. Feedback Control Systems [text] / L.P. Charles, R.D. Phillips. – Harbor – New Jersey. – 2000.
103. Dixon W., Dawson D. M., Zargeloglu E., Behal A. Nonlinear control of wheled mobile robot [text] / W. Dixon, D. M. Dawson, E. Zargeloglu, A. Behal. – Springer – Verlag, London. – 2001.
104. Franklin G.F., Powell J.D., Workman M. Digital Control of Dynamic Systems [text] / G.F. Franklin, J.D. Powell, M. Workman. – 3rd ed. Reading, MA: Addison – Wesley. – 1998.
105. Freidland B. Control System Design [text] / B. Freidland. – New York: McGraw-Hill. – 1986.
106. Hordeski M.F. Design of Microprocessor Sensor & Control Systems. [text] / M.F. Hordeski. – VA: Reston Publishing Company. – 1985. – 392 p.
107. Isidori A. Nonlinear control systems [text] / A. Isidori. – London: Springer. – 1995. – 587 p.
108. Jacobs P.F. Rapid Prototyping and Manufacturing: Fundamentals of Stereolithography [text] / P.F. Jacobs. – Publisher: Society of Manufacturing Engineers. – 1992. – 434 p.
109. John Barrow, Linda Millar, Katherina Malan. Introducing DELPHI programming [text] / J. Barrow, L. Millar, K. Malan// Oxford University Press. – 2005. P. – 544.
110. Karj J.A., Richard M.M. Feedback Systems: An Introduction for Scientists and Engineers [text] / J.A. Karj, M. M. Richard – DRAFT v2.4a. –2006.
111. Liu Y.H., Chen C.L., Tu R.J. A novel space - vector current regulation scheme for a field-oriented-controlled induction motor drive [text] / Y.H. Liu, C.L.,Chen, R.J. Tu // IEEE. –1998. – P. 730-737.
112. Marino R., Tomei P. Global adaptive output feedback control of non-linear systems [text] / R. Marino, P. Tomei // Part I: Linear parameterization. IEEE Trans. Automat. Control. – 1993. V. 38. – № 1. – P. 17-32.
113. Miroshnik. I.V., Nikiforov V.O. Trajectory motion control and coordination of multilink robots [text] / I.V. Miroshnik, V.O. Nikiforov // Prepr. 13th IFAC world Congress. San-Francisco. Vol. A. – P. 361- 366.
114. Murray. R.M., Zexiang L.I., Sastry S.S. A mathematical Introduction to robotic Manipulation [text] / R.M. Murray, L.I. Zexiang, S.S. Sastry. – Boca Ration: CRS Press. –1993.
115. Paraskevopoulas P.N., Trafestas S.G. New results in feedback modal – controllers design [text] / P.N. Paraskevopoulas, S.G. Trafestas // Int. J.Control. – 1975. V21. – № 6. P. – 911- 928.
116. [http:/www.advantech.com](http://www.microsoft.com/isapi/redir.dll?prd=ie&pver=6&ar=msnhome)
117. [http:/www. prosoft. ru](C:\\Documents and Settings\\Nurlan\\Рабочий стол\\km2.swf)
118. <http://www.ispu.ru/library/>
119. <http://www.rosinf.ru/editions/pmea/issues/>
120. <http://microcon.euro.ru/>
121. <http://ets.ifmo.ru:8101/>
122. <http://www.tgizdat.ru/mag/oforml.shtml>
123. <http://www.promavto.ru/>
124. <http://nauka.vprod.ru/2000/sr22_06/0009.htm>
125. <http://zhurnal.ape.relarn.ru/articles/2001/>
126. <http://www.maxonmotor.com>
127. <http://journal.mrsu.ru/>
128. <https://ntv.ifmo.ru/ru/journal/83/journal_83.htm>
129. <http://apriori-journal.ru/>
130. <http://www.lektorium.tv/>

**ПРИЛОЖЕНИЯ**

**ПРИЛОЖЕНИЕ 1**

****

**ПРИЛОЖЕНИЕ 2**

**Задача минимизации на гиперплоскости**

В работе [18] рассматривается задача о нахождении вектора минимальной длины, принадлежащего заданной гиперплоскости. Уравнения гиперплоскости в эвклидовом пространстве задаётся уравнением

(,b)=α (П.1)

где символ (‘ , ‘) обозначает знак скалярного произведения, b – фиксированный вектор, α – скаляр.

Ставится задача о нахождении вектора , лежащего в данной гиперплоскости и имеющего минимальную норму (длину) . Решение этой задачи даётся в виде

(П.2)

для доказательства данного утверждения показывается, во-первых, что х0 лежит в данной гиперплоскости. Действительно, подставляя (П.2) и (П.1), имеем тождество

Далее показывается, что любой другой вектор из этой гиперплоскости имеет большую норму, т.е. . Действительно, если (, b)=α, тогда . Домножая это равенство н имеем

(П.3)

Вместе с тем, по определению скалярного произведения имеет место неравенство

0≤(

Из этого неравенства с учётом (П.3) имеем т.е. норма вектора больше нормы вектора , что и требовалось показать.

**ПРИЛОЖЕНИЕ 3**

**Задача минимизации с ограничениями**

Рассмотрим задачу минимизации норм вектора , удовлетворяющего условиям

, (П.4)

где аi – линейно независимые векторы;

– скаляры.

Известно, что неравенство

(П.5)

Выполняется для любого вектора . Это неравенство можно преобразовать к виду

(П.6)

так как определитель |ai, aj| положителен вследствии линейной независимости векторов ai .

Величина, стоящая в правой части неравенства (П.6), является минимальным значением () при наложенных ограничениях, так как равенство в формуле (П.5) достигается, если выбрать х равным некоторой линейной комбинации векторов aI :

(П.7)

Чтобы удовлетворить уравнениям (П.4), коэффициенты Сj , должны образовывать решение линейной системы уравнений

(П.8)

Так как определитель |ai, aj|≠0, то эта система имеет единственное решение, которое и даёт нам единственный минимизирующий вектор .

**ПРИЛОЖЕНИЕ 4**

**Листинг программы системы управления**

#include <Servo.h>

#include <AFMotor.h>

#define LINE\_BUFFER\_LENGTH 512

// Servo position for Up and Down

const int penZUp = 130;

const int penZDown = 83;

// Servo on PWM pin 6

const int penServoPin = 14;

// Should be right for DVD steppers, but is not too important here

const int stepsPerRevolution = 20;

// create servo object to control a servo

Servo penServo;

AF\_Stepper myStepperY(stepsPerRevolution,1);

AF\_Stepper myStepperX(stepsPerRevolution,2);

/\* Structures, global variables \*/

struct point {

float x;

float y;

float z;

};

// Current position of plothead

struct point actuatorPos;

// Drawing settings, should be OK

float StepInc = 1;

int StepDelay = 0;

int LineDelay =40;

int penDelay = 150;

// Motor steps to go 1 millimeter.

// Use test sketch to go 100 steps. Measure the length of line.

// Calculate steps per mm. Enter here.

float StepsPerMillimeterX = 100.0;

float StepsPerMillimeterY = 100.0;

// Drawing robot limits, in mm

// OK to start with. Could go up to 50 mm if calibrated well.

float Xmin = 0;

float Xmax = 40;

float Ymin = 0;

float Ymax = 40;

float Zmin = 0;

float Zmax = 1;

float Xpos = Xmin;

float Ypos = Ymin;

float Zpos = Zmax;

// Set to true to get debug output.

boolean verbose = false;

// Needs to interpret

// G1 for moving

// G4 P300 (wait 150ms)

// M300 S30 (pen down)

// M300 S50 (pen up)

// Discard anything with a (

// Discard any other command!

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* void setup() - Initialisations

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void setup() {

// Setup

Serial.begin( 9600 );

penServo.attach(penServoPin);

penServo.write(penZUp);

delay(200);

// Decrease if necessary

myStepperX.setSpeed(600);

myStepperY.setSpeed(600);

Serial.println("Mini CNC Plotter alive and kicking!");

Serial.print("X range is from ");

Serial.print(Xmin);

Serial.print(" to ");

Serial.print(Xmax);

Serial.println(" mm.");

Serial.print("Y range is from ");

Serial.print(Ymin);

Serial.print(" to ");

Serial.print(Ymax);

Serial.println(" mm.");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* void loop() - Main loop

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void loop()

{

delay(200);

char line[ LINE\_BUFFER\_LENGTH ];

char c;

int lineIndex;

bool lineIsComment, lineSemiColon;

lineIndex = 0;

lineSemiColon = false;

lineIsComment = false;

while (1) {

// Serial reception - Mostly from Grbl, added semicolon support

while ( Serial.available()>0 ) {

c = Serial.read();

if (( c == '\n') || (c == '\r') ) { // End of line reached

if ( lineIndex > 0 ) { // Line is complete. Then execute!

line[ lineIndex ] = '\0'; // Terminate string

if (verbose) {

Serial.print( "Received : ");

Serial.println( line );

}

processIncomingLine( line, lineIndex );

lineIndex = 0;

}

else {

// Empty or comment line. Skip block.

}

lineIsComment = false;

lineSemiColon = false;

Serial.println("ok");

}

else {

if ( (lineIsComment) || (lineSemiColon) ) { // Throw away all comment characters

if ( c == ')' ) lineIsComment = false; // End of comment. Resume line.

}

else {

if ( c <= ' ' ) { // Throw away whitepace and control characters

}

else if ( c == '/' ) { // Block delete not supported. Ignore character.

}

else if ( c == '(' ) { // Enable comments flag and ignore all characters until ')' or EOL.

lineIsComment = true;

}

else if ( c == ';' ) {

lineSemiColon = true;

}

else if ( lineIndex >= LINE\_BUFFER\_LENGTH-1 ) {

Serial.println( "ERROR - lineBuffer overflow" );

lineIsComment = false;

lineSemiColon = false;

}

else if ( c >= 'a' && c <= 'z' ) { // Upcase lowercase

line[ lineIndex++ ] = c-'a'+'A';

}

else {

line[ lineIndex++ ] = c;

}

}

}

}

}

}

void processIncomingLine( char\* line, int charNB ) {

int currentIndex = 0;

char buffer[ 64 ]; // Hope that 64 is enough for 1 parameter

struct point newPos;

newPos.x = 0.0;

newPos.y = 0.0;

// Needs to interpret

// G1 for moving

// G4 P300 (wait 150ms)

// G1 X60 Y30

// G1 X30 Y50

// M300 S30 (pen down)

// M300 S50 (pen up)

// Discard anything with a (

// Discard any other command!

while( currentIndex < charNB ) {

switch ( line[ currentIndex++ ] ) { // Select command, if any

case 'U':

penUp();

break;

case 'D':

penDown();

break;

case 'G':

buffer[0] = line[ currentIndex++ ]; // /!\ Dirty - Only works with 2 digit commands

// buffer[1] = line[ currentIndex++ ];

// buffer[2] = '\0';

buffer[1] = '\0';

switch ( atoi( buffer ) ){ // Select G command

case 0: // G00 & G01 - Movement or fast movement. Same here

case 1:

// /!\ Dirty - Suppose that X is before Y

char\* indexX = strchr( line+currentIndex, 'X' ); // Get X/Y position in the string (if any)

char\* indexY = strchr( line+currentIndex, 'Y' );

if ( indexY <= 0 ) {

newPos.x = atof( indexX + 1);

newPos.y = actuatorPos.y;

}

else if ( indexX <= 0 ) {

newPos.y = atof( indexY + 1);

newPos.x = actuatorPos.x;

}

else {

newPos.y = atof( indexY + 1);

indexY = '\0';

newPos.x = atof( indexX + 1);

}

drawLine(newPos.x, newPos.y );

// Serial.println("ok");

actuatorPos.x = newPos.x;

actuatorPos.y = newPos.y;

break;

}

break;

case 'M':

buffer[0] = line[ currentIndex++ ]; // /!\ Dirty - Only works with 3 digit commands

buffer[1] = line[ currentIndex++ ];

buffer[2] = line[ currentIndex++ ];

buffer[3] = '\0';

switch ( atoi( buffer ) ){

case 300:

{

char\* indexS = strchr( line+currentIndex, 'S' );

float Spos = atof( indexS + 1);

// Serial.println("ok");

if (Spos == 30) {

penDown();

}

if (Spos == 50) {

penUp();

}

break;

}

case 114: // M114 - Repport position

Serial.print( "Absolute position : X = " );

Serial.print( actuatorPos.x );

Serial.print( " - Y = " );

Serial.println( actuatorPos.y );

break;

default:

Serial.print( "Command not recognized : M");

Serial.println( buffer );

}

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw a line from (x0;y0) to (x1;y1).

\* Bresenham algo from https://www.marginallyclever.com/blog/2013/08/how-to-build-an-2-axis-arduino-cnc-gcode-interpreter/

\* int (x1;y1) : Starting coordinates

\* int (x2;y2) : Ending coordinates

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void drawLine(float x1, float y1) {

if (verbose)

{

Serial.print("fx1, fy1: ");

Serial.print(x1);

Serial.print(",");

Serial.print(y1);

Serial.println("");

}

// Bring instructions within limits

if (x1 >= Xmax) {

x1 = Xmax;

}

if (x1 <= Xmin) {

x1 = Xmin;

}

if (y1 >= Ymax) {

y1 = Ymax;

}

if (y1 <= Ymin) {

y1 = Ymin;

}

if (verbose)

{

Serial.print("Xpos, Ypos: ");

Serial.print(Xpos);

Serial.print(",");

Serial.print(Ypos);

Serial.println("");

}

if (verbose)

{

Serial.print("x1, y1: ");

Serial.print(x1);

Serial.print(",");

Serial.print(y1);

Serial.println("");

}

// Convert coordinates to steps

x1 = (int)(x1\*StepsPerMillimeterX);

y1 = (int)(y1\*StepsPerMillimeterY);

float x0 = Xpos;

float y0 = Ypos;

// Let's find out the change for the coordinates

long dx = abs(x1-x0);

long dy = abs(y1-y0);

int sx = x0<x1 ? StepInc : -StepInc;

int sy = y0<y1 ? StepInc : -StepInc;

long i;

long over = 0;

if (dx > dy) {

for (i=0; i<dx; ++i) {

myStepperX.onestep(sx,MICROSTEP);

over+=dy;

if (over>=dx) {

over-=dx;

myStepperY.onestep(sy,MICROSTEP);

}

delay(StepDelay);

}

}

else {

for (i=0; i<dy; ++i) {

myStepperY.onestep(sy,MICROSTEP);

over+=dx;

if (over>=dy) {

over-=dy;

myStepperX.onestep(sx,MICROSTEP);

}

delay(StepDelay);

}

}

if (verbose)

{

Serial.print("dx, dy:");

Serial.print(dx);

Serial.print(",");

Serial.print(dy);

Serial.println("");

}

if (verbose)

{

Serial.print("Going to (");

Serial.print(x0);

Serial.print(",");

Serial.print(y0);

Serial.println(")");

}

// Delay before any next lines are submitted

delay(LineDelay);

// Update the positions

Xpos = x1;

Ypos = y1;

}

// Raises pen

void penUp() {

penServo.write(penZUp);

delay(penDelay);

Zpos=Zmax;

digitalWrite(15, LOW);

digitalWrite(16, HIGH);

if (verbose) {

Serial.println("Pen up!");

}

}

// Lowers pen

void penDown() {

penServo.write(penZDown);

delay(penDelay);

Zpos=Zmin;

digitalWrite(15, HIGH);

digitalWrite(16, LOW);

if (verbose) {

Serial.println("Pen down.");

}

}

;;;;;;;;;;;;;;;;;;G-CODE

; perimeters extrusion width = 0.50mm

; infill extrusion width = 0.52mm

; solid infill extrusion width = 0.52mm

; top infill extrusion width = 0.52mm

G21 ; set units to millimeters

M107

M104 S205 ; set temperature

G28 ; home all axes

G1 Z5 F5000 ; lift nozzle

M109 S205 ; wait for temperature to be reached

G90 ; use absolute coordinates

G1 F1800.000

G1 Z5.500 F7800.000

G1 X2.029 Y2.323 F7800.000

G1 X3.412 Y1.115 F1080.000

G1 X5.587 Y0.254

G1 X7.000 Y0.107

G1 X13.000 Y0.107

G1 X15.306 Y0.504

G1 X17.346 Y1.650 F1080.000

G1 X18.885 Y3.412

G1 X19.746 Y5.587

G1 X19.893 Y7.000

G1 X19.893 Y13.000

G1 X19.496 Y15.306

G1 X18.350 Y17.346

G1 X16.588 Y18.885

G1 X14.413 Y19.746

G1 X13.000 Y19.893

G1 X7.000 Y19.893

G1 X4.694 Y19.496

G1 X2.654 Y18.350

G1 X1.115 Y16.588

G1 X0.254 Y14.413

G1 X0.107 Y13.000

G1 X0.107 Y7.000

G1 X0.504 Y4.694

G1 X1.650 Y2.654

G1 X1.973 Y2.372

G1 F1800.000

G1 X10.715 Y10.715 F7800.000

G1 X9.285 Y10.715 F540.000

G1 X9.285 Y9.285

G1 X10.715 Y9.285

G1 X10.715 Y10.640

G1 X11.607 Y11.607 F7800.000

G1 X8.393 Y11.607 F540.000

G1 X8.393 Y8.393

G1 X11.607 Y8.393

G1 X11.607 Y11.532

G1 X12.500 Y12.500 F7800.000

G1 X7.500 Y12.500 F378.000

G1 X7.500 Y7.500

G1 X12.500 Y7.500

G1 X12.500 Y12.425

G1 X12.067 Y12.250 F7800.000

M106 S255

M104 S200 ; set temperature

G1 F1800.000

G1 Z5.900 F7800.000

G1 X11.750 Y11.750 F7800.000

G1 X8.250 Y11.750 F1800.000

G1 X8.250 Y8.250

G1 X11.750 Y8.250

G1 X11.750 Y11.675

G1 X12.250 Y12.250 F7800.000

G1 X7.750 Y12.250 F1800.000

G1 X7.750 Y7.750

G1 X12.250 Y7.750

G1 X12.250 Y12.175

G1 X12.750 Y12.750 F7800.000

G1 X7.250 Y12.750 F1260.000

G1 X7.250 Y7.250

G1 X12.750 Y7.250

G1 X12.750 Y12.675

G1 X12.317 Y12.500 F7800.000

G1 X11.403 Y11.363 F7800.000

G1 X11.363 Y11.403 F3600.000

G1 X10.573 Y11.403

G1 X11.403 Y10.573

G1 X11.403 Y9.782

G1 X9.782 Y11.403

G1 X8.992 Y11.403

G1 X11.403 Y8.992

G1 X11.008 Y8.597

G1 X8.597 Y11.008

G1 X8.597 Y10.218

G1 X10.218 Y8.597

G1 X9.427 Y8.597

G1 X8.597 Y9.427

G1 X8.597 Y8.637

G1 X8.637 Y8.597

G1 F1800.000

G1 Z6.300 F7800.000

G1 X11.750 Y11.750 F7800.000

G1 X8.250 Y11.750 F1800.000

G1 X8.250 Y8.250

G1 X11.750 Y8.250

G1 X11.750 Y11.675

G1 X12.250 Y12.250 F7800.000

G1 X7.750 Y12.250 F1800.000

G1 X7.750 Y7.750

G1 X12.250 Y7.750

G1 X12.250 Y12.175

G1 X12.750 Y12.750 F7800.000

G1 X7.250 Y12.750 F1260.000

G1 X7.250 Y7.250

G1 X12.750 Y7.250

G1 X12.750 Y12.675

G1 X12.317 Y12.500 F7800.000

G1 X8.597 Y11.363 F7800.000

G1 X8.637 Y11.403 F3600.000

G1 X9.427 Y11.403

G1 X8.597 Y10.573

G1 X8.597 Y9.782

G1 X10.218 Y11.403

G1 X11.008 Y11.403

G1 X8.597 Y8.992

G1 X8.992 Y8.597

G1 X11.403 Y11.008

G1 X11.403 Y10.218

G1 X9.782 Y8.597

G1 X10.573 Y8.597

G1 X11.403 Y9.427

G1 X11.403 Y8.637

G1 X11.363 Y8.597

G1 F1800.000

G1 Z6.700 F7800.000

G1 X11.750 Y11.750 F7800.000

G1 X8.250 Y11.750 F1800.000

G1 X8.250 Y8.250

G1 X11.750 Y8.250

G1 X11.750 Y11.675

G1 X12.250 Y12.250 F7800.000

G1 X7.750 Y12.250 F1800.000

G1 X7.750 Y7.750

G1 X12.250 Y7.750

G1 X12.250 Y12.175

G1 X12.750 Y12.750 F7800.000

G1 X7.250 Y12.750 F1260.000

G1 X7.250 Y7.250

G1 X12.750 Y7.250

G1 X12.750 Y12.675

G1 X12.317 Y12.500 F7800.000

G1 X11.403 Y11.363 F7800.000

G1 X11.363 Y11.403 F3600.000

G1 X10.573 Y11.403

G1 X11.403 Y10.573

G1 X11.403 Y9.782

G1 X9.782 Y11.403

G1 X8.992 Y11.403

G1 X11.403 Y8.992

G1 X11.008 Y8.597

G1 X8.597 Y11.008

G1 X8.597 Y10.218

G1 X10.218 Y8.597

G1 X9.427 Y8.597

G1 X8.597 Y9.427

G1 X8.597 Y8.637

G1 X8.637 Y8.597

G1 F1800.000

G1 Z7.100 F7800.000

G1 X11.750 Y11.750 F7800.000

G1 X8.250 Y11.750 F1800.000

G1 X8.250 Y8.250

G1 X11.750 Y8.250

G1 X11.750 Y11.675

G1 X12.250 Y12.250 F7800.000

G1 X7.750 Y12.250 F1800.000

G1 X7.750 Y7.750

G1 X12.250 Y7.750

G1 X12.250 Y12.175

G1 X12.750 Y12.750 F7800.000

G1 X7.250 Y12.750 F1260.000

G1 X7.250 Y7.250

G1 X12.750 Y7.250

G1 X12.750 Y12.675

G1 X12.317 Y12.500 F7800.000

G1 X8.597 Y11.363 F7800.000

G1 X8.637 Y11.403 F3600.000

G1 X9.427 Y11.403

G1 X8.597 Y10.573

G1 X8.597 Y9.782

G1 X10.218 Y11.403

G1 X11.008 Y11.403

G1 X8.597 Y8.992

G1 X8.992 Y8.597

G1 X11.403 Y11.008

G1 X11.403 Y10.218

G1 X9.782 Y8.597

G1 X10.573 Y8.597

G1 X11.403 Y9.427

G1 X11.403 Y8.637

G1 X11.363 Y8.597

G1 F1800.000

G1 Z7.500 F7800.000

G1 X11.750 Y11.750 F7800.000

G1 X8.250 Y11.750 F1800.000

G1 X8.250 Y8.250

G1 X11.750 Y8.250

G1 X11.750 Y11.675

G1 X12.250 Y12.250 F7800.000

G1 X7.750 Y12.250 F1800.000

G1 X7.750 Y7.750

G1 X12.250 Y7.750

G1 X12.250 Y12.175

G1 X12.750 Y12.750 F7800.000

G1 X7.250 Y12.750 F1260.000

G1 X7.250 Y7.250

G1 X12.750 Y7.250

G1 X12.750 Y12.675

G1 X12.317 Y12.500 F7800.000

G1 X11.403 Y11.363 F7800.000

G1 X11.363 Y11.403 F3600.000

G1 X10.573 Y11.403

G1 X11.403 Y10.573

G1 X11.403 Y9.782

G1 X9.782 Y11.403

G1 X8.992 Y11.403

G1 X11.403 Y8.992

G1 X11.008 Y8.597

G1 X8.597 Y11.008

G1 X8.597 Y10.218

G1 X10.218 Y8.597

G1 X9.427 Y8.597

G1 X8.597 Y9.427

G1 X8.597 Y8.637

G1 X8.637 Y8.597

G1 F1800.000

G1 Z7.900 F7800.000

G1 X11.750 Y11.750 F7800.000

G1 X8.250 Y11.750 F1800.000

G1 X8.250 Y8.250

G1 X11.750 Y8.250

G1 X11.750 Y11.675

G1 X12.250 Y12.250 F7800.000

G1 X7.750 Y12.250 F1800.000

G1 X7.750 Y7.750

G1 X12.250 Y7.750

G1 X12.250 Y12.175

G1 X12.750 Y12.750 F7800.000

G1 X7.250 Y12.750 F1260.000

G1 X7.250 Y7.250

G1 X12.750 Y7.250

G1 X12.750 Y12.675

G1 X12.317 Y12.500 F7800.000

G1 X8.597 Y11.363 F7800.000

G1 X8.637 Y11.403 F3600.000

G1 X9.427 Y11.403

G1 X8.597 Y10.573

G1 X8.597 Y9.782

G1 X10.218 Y11.403

G1 X11.008 Y11.403

G1 X8.597 Y8.992

G1 X8.992 Y8.597

G1 X11.403 Y11.008

G1 X11.403 Y10.218

G1 X9.782 Y8.597

G1 X10.573 Y8.597

G1 X11.403 Y9.427

G1 X11.403 Y8.637

G1 X11.363 Y8.597

G1 F1800.000

G1 Z8.300 F7800.000

G1 X11.750 Y11.750 F7800.000

G1 X8.250 Y11.750 F1800.000

G1 X8.250 Y8.250

G1 X11.750 Y8.250

G1 X11.750 Y11.675

G1 X12.250 Y12.250 F7800.000

G1 X7.750 Y12.250 F1800.000

G1 X7.750 Y7.750

G1 X12.250 Y7.750

G1 X12.250 Y12.175

G1 X12.750 Y12.750 F7800.000

G1 X7.250 Y12.750 F1260.000

G1 X7.250 Y7.250

G1 X12.750 Y7.250

G1 X12.750 Y12.675

G1 X12.317 Y12.500 F7800.000

G1 X11.403 Y11.363 F7800.000

G1 X11.363 Y11.403 F3600.000

G1 X10.573 Y11.403

G1 X11.403 Y10.573

G1 X11.403 Y9.782

G1 X9.782 Y11.403

G1 X8.992 Y11.403

G1 X11.403 Y8.992

G1 X11.008 Y8.597

G1 X8.597 Y11.008

G1 X8.597 Y10.218

G1 X10.218 Y8.597

G1 X9.427 Y8.597

G1 X8.597 Y9.427

G1 X8.597 Y8.637

G1 X8.637 Y8.597

G1 F1800.000

G1 Z8.700 F7800.000

G1 X11.750 Y11.750 F7800.000

G1 X8.250 Y11.750 F1800.000

G1 X8.250 Y8.250

G1 X11.750 Y8.250

G1 X11.750 Y11.675

G1 X12.250 Y12.250 F7800.000

G1 X7.750 Y12.250 F1800.000

G1 X7.750 Y7.750

G1 X12.250 Y7.750

G1 X12.250 Y12.175

G1 X12.750 Y12.750 F7800.000

G1 X7.250 Y12.750 F1260.000

G1 X7.250 Y7.250

G1 X12.750 Y7.250

G1 X12.750 Y12.675

G1 X12.317 Y12.500 F7800.000

G1 X8.597 Y11.363 F7800.000

G1 X8.637 Y11.403 F3600.000

G1 X9.427 Y11.403

G1 X8.597 Y10.573

G1 X8.597 Y9.782

G1 X10.218 Y11.403

G1 X11.008 Y11.403

G1 X8.597 Y8.992

G1 X8.992 Y8.597

G1 X11.403 Y11.008

G1 X11.403 Y10.218

G1 X9.782 Y8.597

G1 X10.573 Y8.597

G1 X11.403 Y9.427

G1 X11.403 Y8.637

G1 X11.363 Y8.597

G1 F1800.000

G1 Z9.100 F7800.000

G1 X11.750 Y11.750 F7800.000

G1 X8.250 Y11.750 F1800.000

G1 X8.250 Y8.250

G1 X11.750 Y8.250

G1 X11.750 Y11.675

G1 X12.250 Y12.250 F7800.000

G1 X7.750 Y12.250 F1800.000

G1 X7.750 Y7.750

G1 X12.250 Y7.750

G1 X12.250 Y12.175

G1 X12.750 Y12.750 F7800.000

G1 X7.250 Y12.750 F1260.000

G1 X7.250 Y7.250

G1 X12.750 Y7.250

G1 X12.750 Y12.675

G1 X12.317 Y12.500 F7800.000

G1 X11.403 Y11.363 F7800.000

G1 X11.363 Y11.403 F3600.000

G1 X10.573 Y11.403

G1 X11.403 Y10.573

G1 X11.403 Y9.782

G1 X9.782 Y11.403

G1 X8.992 Y11.403

G1 X11.403 Y8.992

G1 X11.008 Y8.597

G1 X8.597 Y11.008

G1 X8.597 Y10.218

G1 X10.218 Y8.597

G1 X9.427 Y8.597

G1 X8.597 Y9.427

G1 X8.597 Y8.637

G1 X8.637 Y8.597

G1 F1800.000

G1 Z9.500 F7800.000

G1 X11.750 Y11.750 F7800.000

G1 X8.250 Y11.750 F1800.000

G1 X8.250 Y8.250

G1 X11.750 Y8.250

G1 X11.750 Y11.675

G1 X12.250 Y12.250 F7800.000

G1 X7.750 Y12.250 F1800.000

G1 X7.750 Y7.750

G1 X12.250 Y7.750

G1 X12.250 Y12.175

G1 X12.750 Y12.750 F7800.000

G1 X7.250 Y12.750 F1260.000

G1 X7.250 Y7.250

G1 X12.750 Y7.250

G1 X12.750 Y12.675

G1 X12.317 Y12.500 F7800.000

G1 X8.597 Y11.363 F7800.000

G1 X8.637 Y11.403 F3600.000

G1 X9.427 Y11.403

G1 X8.597 Y10.573

G1 X8.597 Y9.782

G1 X10.218 Y11.403

G1 X11.008 Y11.403

G1 X8.597 Y8.992

G1 X8.992 Y8.597

G1 X11.403 Y11.008

G1 X11.403 Y10.218

G1 X9.782 Y8.597

G1 X10.573 Y8.597

G1 X11.403 Y9.427

G1 X11.403 Y8.637

G1 X11.363 Y8.597

G1 F1800.000

G1 Z9.900 F7800.000

G1 X11.750 Y11.750 F7800.000

G1 X8.250 Y11.750 F1800.000

G1 X8.250 Y8.250

G1 X11.750 Y8.250

G1 X11.750 Y11.675

G1 X12.250 Y12.250 F7800.000

G1 X7.750 Y12.250 F1800.000

G1 X7.750 Y7.750

G1 X12.250 Y7.750

G1 X12.250 Y12.175

G1 X12.750 Y12.750 F7800.000

G1 X7.250 Y12.750 F1260.000

G1 X7.250 Y7.250

G1 X12.750 Y7.250

G1 X12.750 Y12.675

G1 X12.317 Y12.500 F7800.000

G1 X11.403 Y11.363 F7800.000

G1 X11.363 Y11.403 F3600.000

G1 X10.573 Y11.403

G1 X11.403 Y10.573

G1 X11.403 Y9.782

G1 X9.782 Y11.403

G1 X8.992 Y11.403

G1 X11.403 Y8.992

G1 X11.008 Y8.597

G1 X8.597 Y11.008

G1 X8.597 Y10.218

G1 X10.218 Y8.597

G1 X9.427 Y8.597

G1 X8.597 Y9.427

G1 X8.597 Y8.637

G1 X8.637 Y8.597

G1 F1800.000

G1 Z10.300 F7800.000

G1 X11.750 Y11.750 F7800.000

G1 X8.250 Y11.750 F1800.000

G1 X8.250 Y8.250

G1 X11.750 Y8.250

G1 X11.750 Y11.675

G1 X12.250 Y12.250 F7800.000

G1 X7.750 Y12.250 F1800.000

G1 X7.750 Y7.750

G1 X12.250 Y7.750

G1 X12.250 Y12.175

G1 X12.750 Y12.750 F7800.000

G1 X7.250 Y12.750 F1260.000

G1 X7.250 Y7.250

G1 X12.750 Y7.250

G1 X12.750 Y12.675

G1 X12.317 Y12.500 F7800.000

G1 X8.597 Y11.363 F7800.000

G1 X8.637 Y11.403 F3600.000

G1 X9.427 Y11.403

G1 X8.597 Y10.573

G1 X8.597 Y9.782

G1 X10.218 Y11.403

G1 X11.008 Y11.403

G1 X8.597 Y8.992

G1 X8.992 Y8.597

G1 X11.403 Y11.008

G1 X11.403 Y10.218

G1 X9.782 Y8.597

G1 X10.573 Y8.597

G1 X11.403 Y9.427

G1 X11.403 Y8.637

G1 X11.363 Y8.597

G1 F1800.000

G1 Z10.700 F7800.000

G1 X11.750 Y11.750 F7800.000

G1 X8.250 Y11.750 F1800.000

G1 X8.250 Y8.250

G1 X11.750 Y8.250

G1 X11.750 Y11.675

G1 X12.250 Y12.250 F7800.000

G1 X7.750 Y12.250 F1800.000

G1 X7.750 Y7.750

G1 X12.250 Y7.750

G1 X12.250 Y12.175

G1 X12.750 Y12.750 F7800.000

G1 X7.250 Y12.750 F1260.000

G1 X7.250 Y7.250

G1 X12.750 Y7.250

G1 X12.750 Y12.675

G1 X12.317 Y12.500 F7800.000

G1 X11.403 Y11.363 F7800.000

G1 X11.363 Y11.403 F3600.000

G1 X10.573 Y11.403

G1 X11.403 Y10.573

G1 X11.403 Y9.782

G1 X9.782 Y11.403

G1 X8.992 Y11.403

G1 X11.403 Y8.992

G1 X11.008 Y8.597

G1 X8.597 Y11.008

G1 X8.597 Y10.218

G1 X10.218 Y8.597

G1 X9.427 Y8.597

G1 X8.597 Y9.427

G1 X8.597 Y8.637

G1 X8.637 Y8.597

G1 F1800.000

G1 Z11.100 F7800.000

G1 X11.750 Y11.750 F7800.000

G1 X8.250 Y11.750 F1800.000

G1 X8.250 Y8.250

G1 X11.750 Y8.250

G1 X11.750 Y11.675

G1 X12.250 Y12.250 F7800.000

G1 X7.750 Y12.250 F1800.000

G1 X7.750 Y7.750

G1 X12.250 Y7.750

G1 X12.250 Y12.175

G1 X12.750 Y12.750 F7800.000

G1 X7.250 Y12.750 F1260.000

G1 X7.250 Y7.250

G1 X12.750 Y7.250

G1 X12.750 Y12.675

G1 X12.317 Y12.500 F7800.000

G1 F1800.000

G1 X8.597 Y11.363 F7800.000

G1 X8.637 Y11.403 F3000.000

G1 X9.427 Y11.403

G1 X8.597 Y10.573

G1 X8.597 Y9.782

G1 X10.218 Y11.403

G1 X11.008 Y11.403

G1 X8.597 Y8.992

G1 X8.992 Y8.597

G1 X11.403 Y11.008

G1 X11.403 Y10.218

G1 X9.782 Y8.597

G1 X10.573 Y8.597

G1 X11.403 Y9.427

G1 X11.403 Y8.637

G1 X11.363 Y8.597

G1 F1800.000

M107

M104 S0; turn off temperature

G28 X0; home X axis

M84; disable motors

; filament used = 1.0mm (0.0cm3)

; avoid\_crossing\_perimeters = 1

; bed\_size = 10,10

; bed\_temperature = 0

; bridge\_acceleration = 0

; bridge\_fan\_speed = 100

; bridge\_flow\_ratio = 1

; brim\_width = 0

; complete\_objects = 0

; cooling = 1

; default\_acceleration = 0

; disable\_fan\_first\_layers = 1

; duplicate\_distance = 6

; end\_gcode = M104 S0 ; turn off temperature\nG28 X0 ; home X axis\nM84 ; disable motors\n

; external\_perimeters\_first = 0

; extruder\_clearance\_height = 20

; extruder\_clearance\_radius = 20

; extruder\_offset = 0x0

; extrusion\_axis = E

; extrusion\_multiplier = 1

; fan\_always\_on = 0

; fan\_below\_layer\_time = 60

; filament\_diameter = 3

; first\_layer\_acceleration = 0

; first\_layer\_bed\_temperature = 0

; first\_layer\_extrusion\_width = 200%

; first\_layer\_speed = 40%

; first\_layer\_temperature = 206

; g0 = 0

; gcode\_arcs = 0

; gcode\_comments = 0

; gcode\_flavor = no-extrusion

; infill\_acceleration = 0

; infill\_first = 0

; layer\_gcode =

; max\_fan\_speed = 100

; min\_fan\_speed = 45

; min\_print\_speed = 10

; min\_skirt\_length = 0

; notes =

; nozzle\_diameter = 0.5

; only\_retract\_when\_crossing\_perimeters = 1

; ooze\_prevention = 0

; output\_filename\_format = [input\_filename\_base].gcode

; perimeter\_acceleration = 0

; post\_process =

; print\_center = 10,10

; resolution = 0

; retract\_before\_travel = 2

; retract\_layer\_change = 1

; retract\_length = 1

; retract\_length\_toolchange = 10

; retract\_lift = 0

; retract\_restart\_extra = 0

; retract\_restart\_extra\_toolchange = 0

; retract\_speed = 20

; skirt\_distance = 7

; skirt\_height = 1

; skirts = 1

; slowdown\_below\_layer\_time = 40

; spiral\_vase = 0

; standby\_temperature\_delta = -5

; start\_gcode = G28 ; home all axes\nG1 Z5 F5000 ; lift nozzle\n

; temperature = 300

; threads = 3

; toolchange\_gcode =

; travel\_speed = 130

; use\_firmware\_retraction = 0

; use\_relative\_e\_distances = 0

; vibration\_limit = 0

; wipe = 0

; z\_offset = 6

; dont\_support\_bridges = 2

; extrusion\_width = 0

; first\_layer\_height = 0.6

; infill\_only\_where\_needed = 0

; interface\_shells = 0

; layer\_height = 0.5

; raft\_layers = 0

; seam\_position = aligned

; support\_material = 0

; support\_material\_angle = 0

; support\_material\_enforce\_layers = 0

; support\_material\_extruder = 1

; support\_material\_extrusion\_width = 0

; support\_material\_interface\_extruder = 1

; support\_material\_interface\_layers = 3

; support\_material\_interface\_spacing = 0

; support\_material\_interface\_speed = 100%

; support\_material\_pattern = pillars

; support\_material\_spacing = 2.5

; support\_material\_speed = 60

; support\_material\_threshold = 0