ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени М.В.ЛОМОНОСОВА»

МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА ОБЩИХ ПРОБЛЕМ УПРАВЛЕНИЯ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

(ДИПЛОМНАЯ РАБОТА)

специалиста

**Построение оптимального синтеза в одной задаче оптимального управления для функционала быстродействия и его реализация в виде мобильного робота на платформе Arduino**

Выполнил студент

632 группы

Кондрашкин Степан Сергеевич

Научный руководитель:

Профессор

Локуциевский Лев Вячеславович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва

2019

Оглавление

[Введение 3](#_Toc8609653)

[Глава I. Постановка задачи и ее решение. 5](#_Toc8609654)

[Глава II. Физическая реализация управления. 15](#_Toc8609655)

[2.1. Модули, используемые в реализации. 15](#_Toc8609656)

[2.2. Вычисление угла отклонения маятника. 17](#_Toc8609657)

[2.3. Энкодер. 20](#_Toc8609658)

[2.4. Первые проблемы и их решения. 22](#_Toc8609659)

[2.5. Бесколлекторный двигатель. 24](#_Toc8609660)

[2.6. Схема подключения. 26](#_Toc8609661)

[2.7. Программное обеспечение. 27](#_Toc8609662)

[Выводы. 28](#_Toc8609663)

[Заключение 29](#_Toc8609664)

[Список использованной литературы. 30](#_Toc8609665)

# Введение

Настоящее исследование посвящено реализации оптимального управления для платформы, управляемой микроконтроллером Arduino.

Проблема поиска оптимального управления рассматривалась многими ведущими специалистами. В настоящее время решено большое количество задач оптимального управления. Однако стоит отметить, что далеко не все они реализованы в физических, программных или иных практических решениях. Одной из таких задач является стабилизация обратного маятника с одним или несколькими звеньями. Возможность реализовать задачу с несколькими звеньями никто не исследовал. Это определяет **актуальность** нашего исследования.

**Цель** данной работы заключается в попытке реализовать оптимальное управление леанизированной задачи для тележки с однозвенным обратным маятником из общедоступных компонентов, модулей.

Данная цель требует решения следующих исследовательских **задач**:

1. найти управление, позволяющее стабилизировать перевёрнутый маятник на тележке за минимально возможное время;
2. собрать платформу, разработать коммутацию между всеми модулями и микроконтроллерами;
3. найти программное обеспечение для микроконтроллеров.

Для выполнения поставленных задач и достижения цели настоящей дипломной работы были использованы аналитические методы.

Теоретической базой исследования послужили исследовательские труды, научные труды С.А. Беляев (Современные подходы к решению задач стабилизации перевернутого маятника) [1], А.А. Колесников, И.В. Кондратьев (Синергетическое управление механической системой «перевернутый маятник на тележке» линейное преобразование координат)[2].

**Научная значимость** работы состоит в попытке реализации оптимального управления для тележки с перевёрнутым маятником.

**Практическая значимость** работы состоит в оценки возможностей микроконтроллеров, датчиков, а также моторов, для реализации оптимального управления.

**Структура работы**. Настоящая дипломная работа состоит из введения, двух глав, выводов, заключения, списка использованной литературы.

# Глава I. Постановка задачи и ее решение.

Рисунок 1. Чертёж физической системы

В рамках настоящего исследования мы располагаем следующими данными.

Мы имеем тележку с жёстко закрепленным стержнем на шарнире. При этом стержень мы считаем невесомым. На его конце размещена масса m. В начальный момент времени система находится в подвижном состоянии. Наша задача заключается в том, чтобы найти управление, которое позволит стабилизировать перевёрнутый маятник на тележке за минимально возможное время Т. Для этого мы вводим следующие обозначения:

M – Масса тележки

m – Масса маятника

T – Минимальное время стабилизации перевернутого маятника

l – Длина стержня

u – Управление тележки (ускорение тележки)

– Угол отклонения маятника от вертикали

v – Скорость центра тяжести

– Угловое ускорение маятника

- задача стабилизации.

Решение задачи выглядит следующим образом.

Запишем Лагранжиан:

Пусть - скорость тележки, – скорость центра масс маятника относительно тележки.

Таким образом, уравнение кинетической энергии приобретает вид:

Так как , мы получаем:

Потенциальная энергия системы имеет вид:

Таким образом мы получаем:

Воспользуемся уравнениями Эйлера-Лагранжа для и :

Линеаризуя данную систему в окрестности точки , мы получаем следующую систему:

, и получим задачу оптимизации:

Граничные условия:

Воспользуемся принципом максимума Понтрягина:

, что функционал где , достигает своего максимума на оптимальной траектории управления из множества всех управлений

В нашем случае функционал выглядит следующим образом:

Запишем Гамильтонову систему:

Из четвёртого уравнения получаем: .  
Подставляя из шестого в пятое, получим:

Дифференцируя , получим , и таким образом наша система примет следующий вид:  
Поиск максимума гамильтониана по эквивалентен поиску максимума той части, которая зависит от управления :

Отсюда следует, что есть 3 случая для :  
1)  
2)  
3)

Заметим, что случай под номером 3 не возможен так, как из него получаем, что . Следовательно, из Гамильтоновой системы мы находим, что   
Подставим это в :   
. Так как M, значит C=0 , что противоречит принципу максимума.

Найдёмто есть, когда =0:   
   
Используя замену: ,

Таким образом, мы можем рассмотреть 3 случая:   
1) - В этом случае 2 переключения мощности.

2) - Одно переключение.

3) – Нет переключений.

Решим начальную систему:

Решение этой системы выглядит следующим образом:

В общем случае будет 2 переключения. Соответственно, можно построить поверхность переключения и в зависимости от того, с какой стороны будет находиться точка состояния системы, двигаться с управлением и при достижении поверхности переключения изменить управление на .

При достижении кривой переключения сменить управление на U и достичь точки (0,0,0).

Для построении поверхности переключения, решим задачу в обратном времени. Пусть в начальный момент времени:

Подставив t=0, получим следующую систему:

Подставим данные константы в решения и получим следующую кривую:

Предположим, что мы двигались с управлением в течение времени и совершили переключение, т.е. сменили на . Таким образом, нам надо найти константы из граничных условий:

Решив данную систему, получим следующие константы:

Подставим данные константы в уравнения для

Значит поверхность, заданная параметрически, имеет следующий вид:

б

В фазовой плоскости данная поверхность выглядит следующим образом, где синие – это траектории с управлением U1 и U2, а красные – U2 и U1.

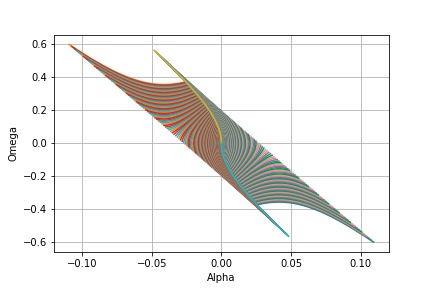


Рисунок 2. Поверхность на фазовой плоскости (α,ω)

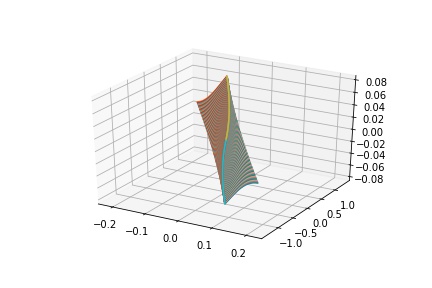
В пространстве () построенная поверхность представлена ниже:

Рисунок 3. Поверхность в пространстве (α,ω,v)

# Глава II. Физическая реализация управления.

## 2.1. Модули, используемые в реализации.

В качестве основы была выбрана платформа от компании DFRobots, Pirate 4WD. В комплект входит 4 мотора малой мощности, корпус, аккумуляторный блок. Моторы и плата управления питается от 5 батареек размера ААА.

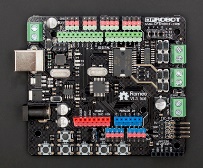
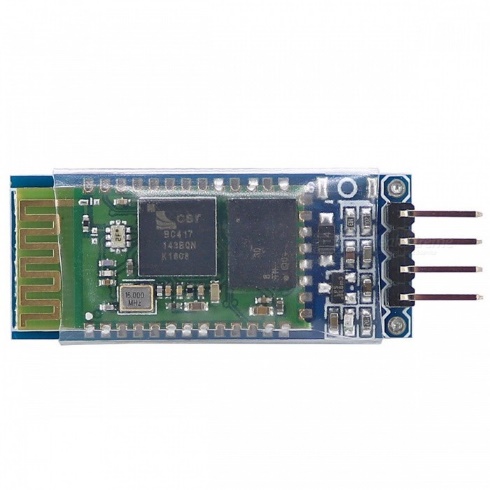
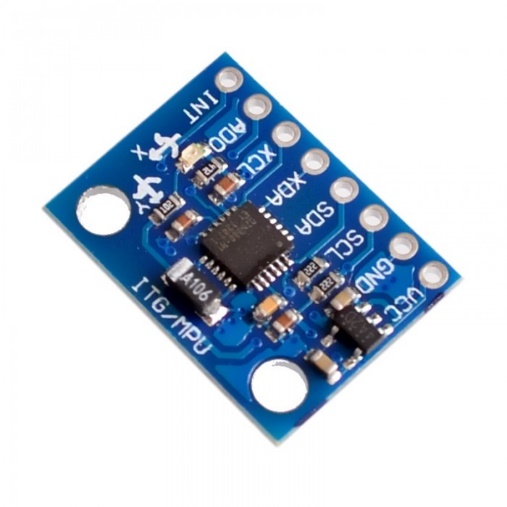
Также была куплена плата DFRobots, Romeo для управления данной платформой. Эта плата является усовершенствованным вариантом популярной версии микроконтроллеров Arduino Leonardo. Отличие Rameo от Leonardo заключается в наличии встроенного драйвера моторов L298SO20, который позволяет управлять мощностью на моторах. На Rameo стоит 16MHz процессор, EEPROM (энергонезависимая память) размером 1kB, Flash память (программа и константы) размером 32 kB, и памятью SPAM (переменные) 2 kB.

Рисунок 6 Модуль MPU6050

Рисунок 7 Bluetooth модуль HC-06

Рисунок 4 DFRobots Pirate

Рисунок 5 DFRobots Rameo

Для измерения угла был выбран акселерометр mpu6050. Данный модуль содержит в себе акселерометр и гироскоп, что позволяет более точно считать отклонение акселерометра от вертикального положения. Модуль работает по I2C – последовательной асимметричной шине для связи. Частота данного модуля 1кГц, что позволяет опрашивать его не чаще чем 1 раз в 1 миллисекунду.

Для связи с телефоном или компьютером и для возможности удалённого управления был куплен модуль Bluetooth HC-06.

## 2.2. Вычисление угла отклонения маятника.

Имеется датчик MPU6050, закреплённый в верхней точке перевёрнутого маятника. Необходимо с помощью этого датчика определить угол отклонения мятника относительно вертикали.

Гироскоп выдает значения мгновенной угловой скорости с разрешением, заданным в настройках, например, 2000 градусов в секунду.

Акселерометр представляет собой устройство, которое измеряет проекцию кажущегося ускорения, то есть, разницу между истинным ускорением объекта и гравитационным ускорением.

Измерения и акселерометра, и гороскопа имеют систематические ошибки. Акселерометр обеспечивает точные данные в течение длительного времени, но в краткосрочной перспективе является шумным.

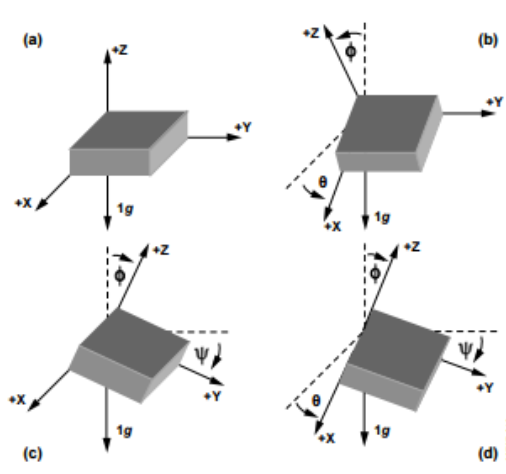
**Акселерометр****.** Одним из методов определения наклона тела с тремя осями является определение каждого угла отдельно для каждой оси акселерометра от исходного положения. За исходное положение принимается ситуация, когда проекция силы тяжести на оси X и Y равны 0, проекция силы тяжести на ось Z равна − g .

Рисунок 8. Углы отклонения от вертикальной   
оси

Обозначим за θ угол между горизонтом и осью X, за угол между горизонтом и осью Y, за φ угол между вектором ~g и осью Z. Тогда формулы для определения θ, и φ имеют вид:

**Гироскоп**. Гироскоп вычисляет угловые ускорения относительно осей X, Y, Z соответственно. Поэтому необходимо задать начальные положения , найденные, например, акселерометром. Тогда если ∆t – это некоторый интервал времени, a – одна из угловых скоростей, то новый угол вычисляется по формуле = + ∆t .

**Ошибка.** Для минимизации ошибки был использован фильтр Калмана. Пусть динамическая система записывается следующей системой уравнений: , где – показания датчика; – истинное значение наблюдаемой величины; F – матрица перехода между состояниями (динамическая модель системы); B – матрица применения управляющего воздействия; – управляющее воздействие;  
 H – матрица измерений, отображающая отношение измерений и состояний;  
 – шум процесса, при этом считаем, что все независимы.   
Датчик имеет некоторую погрешность измерения , являющуюся случайной величиной; все считаем независмыми.   
Будем предполагать, что = N(0; ) и = N(0; )

Ведём необходимые обозначения:

– предсказание состояния системы в текущий момент времени.  
 – состояние системы в прошлый момент времени.  
Q – ковариация шума процесса.  
 – ошибка в прошлый момент времени.  
 – ошибка в прошлый момент времени.  
 – измерение в текущий момент времени.  
 – усиление Калмана.  
R – ковариация шума измерения.  
I – единичная матрица.

В каждый момент времени алгоритм делится на два этапа:  
**I. Предсказание**  
Предсказание состояния системы: Предсказание ошибки ковариации:   
**II. Корректировка**  
1. Вычисление усиления Калмана: 2. Обновление оценки с учётом измерения :3. Обновление ковариации ошибки:

**Теорема [1]:** Если шумы системы являются гауссовскими случайными величинами, то фильтр Калмана даёт наилучшуюлинейную несмещённую оценку состояния .

## 2.3. Энкодер.

При использовании модуля MPU6050 было выявлено, что ошибка, выдаваемая датчиком, может существенно повлиять на значения в около вертикальном положении, а также может произойти накопление ошибки и получится сильная зависимость от стабильности напряжения. После консультации со специалистом из института механики было решено использовать инкрементный оптический энкодер.

Энкодер представляет собой устройство, предназначенное для преобразования угла поворота вращающегося объекта (вала) в цифровые или аналоговые сигналы, позволяющие определить угол его поворота.  
 Энкодеры делятся по способу выдачи информации на накапливающие (инкрементные) и абсолютные (позиционные), а также делятся по принципу действия на оптические, резистивные, магнитные, индуктивные, механические, и по допустимому углу поворота вала ограниченным диапазоном, и с неограниченным диапазоном работы.

Инкрементные энкодеры на выходе формируют импульсы, по которым принимающее устройство определяет текущее положение вала путём подсчёта числа импульсов счётчиком. В начальный момент число импульсов неизвестно. К минусам данного типа энкодера можно отнести невозможность определить начальное положение вала, а также невозможность определить пропуск импульсов по каким-либо причинам.

Оптические энкодеры имеют стеклянный диск с оптическим растром, жёстко закреплённый на валу. При вращении вала растр перемещается относительно неподвижного растра, при этом модулируется световой поток, принимаемый фотодатчиком.

Механические энкодеры содержат диск из диэлектрика или стекла с нанесёнными выпуклыми, проводящими или непрозрачными участками.

На рисунке 10 изображён принцип работы инкрементного энкодера.

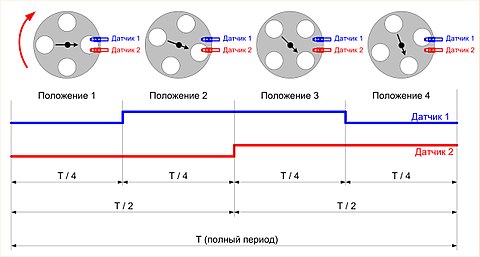
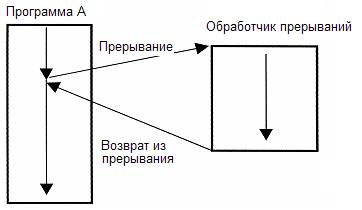
Если пропустить 2 положения энкодера, то программа некорректно распознает изменение положение энкодера. Так как программа распознает это изменение как вращение вала энкодера в обратном направлении, то это приведёт к ошибке, равной одному импульсу. С целью не допустить возникновения данной проблемы выводы энкодера подключены к пинам прерывания на платформе Ардуино. При изменении сигнала на пине прерывания, микроконтроллер прерывает выполнение текущих инструкций и передаёт управление обработчику прерывания. После выполнения прерывания обработчик возвращается на место прерывания и продолжает выполнение кода основной программы. В связи с тем, что каждый энкодер требует по 2 пина прерывания, нам необходимо 4 пина прерывания на платформе Ардуино.

Рисунок 9. Принцип работы прерывания

Рисунок 10. Принцип работы инкриментного энкодера

Для определения начального положения было выбрано начальное положение маятника в сторону передней части платформы, где расположен акселерометр.

Для определения угла отклонения маятника был выбран инкрементный оптический энкодер со 128 импульсами на один оборот, что соответствует тому, что один «щелчок» составляет ~ 0.7 градуса. А для определения положения тележки и её скорости мы выбрали механический энкодер с 24 импульсами на оборот, что соответствует 3.75 градусам на один «щелчок».

Помимо угла энкодер должен выдавать угловую скорость. Данную скорость можно рассчитать из положения энкодера в данный момент и некоторый момент dt. Если выбрать dt очень маленьким, то значение скорости будет скачкообразным, что приведёт к некорректному управлению. Если выбрать dt слишком большим, то возрастёт точность определения угловой скорости, но возникнет лаг системы, который также может привести к некорректному определению управления. Данная проблема представлена на графике ниже, где синим цветом изображена угловая скорость с расчётом в 20мс, а зелёном цветом скорость, рассчитанная за последние 80мс.  
По зелёному графику видно, что появился лаг системы.

## Энкодер и Скорость**2.4.** Первые проблемы и их решения**.**

Рисунок 11. Угол и рассчитанные угловые скорости

В ходе проведения нашего исследования мы столкнулись со следующими ошибками:

1. Моторы, установленные в данную платформу, оказались слабыми и смогли стабилизировать маятник только в очень маленьких углах.
2. Bluetooth модуль потребляет большой ток, и если аккумуляторные батареи немного разряжены, то в момент старта или переключения направления моторов, он выключается, и для дальнейшего его использования необходимо переподключиться.
3. Акселерометр очень чувствительный к напряжению, и в силу второго пункта он также при просадке напряжения отключается и выдаёт нулевые значения. Для того чтобы он начал выдавать значения, после включения его надо вывести из режима сна.

Для решения первой проблемы было принято решение использовать двигатель с большей мощностью. В наличии оказался бесколлекторный двигатель от радиоуправляемой машинки, который был установлен в данную платформу.

Для решения второго пункта было решено заменить штатное питание в виде батарейного блока, на литий-ионный аккумулятор с платой защиты/зарядки и модулем, повышающим напряжение. Данный аккумулятор будет питать платформы Arduino.

Для решения третьего пункта используется энкодер, который не будет зависеть от перепадов напряжений или ошибки.

## 2.5. Бесколлекторный двигатель.

Отличие бесколлекторных двигателей от коллекторных заключается в следующих двух пунктах:

1. Бесколлекторные двигатели питаются трёхфазным переменным током, поэтому для их работы необходим специальный контроллер (регулятор скорости), преобразующий постоянный ток от аккумуляторной батареи в переменный.
2. Ротор с магнитами вращается вокруг неподвижного статора с электромагнитной катушкой.

В процессе реализации проекта был использован регулятор хода, который был в комплекте с радиоуправляемой машинкой.

Данный регулятор хода в радиоуправляемой модели подключён к приёмнику сигнала с радиоуправляемого пульта. Поэтому для общения с ним необходимо моделировать частоту, которая соответствует тому или иному положению рычага пульта управления. Так как моделировать данную частоту необходимо непрерывно, основная плата не могла одновременно управлять мотором и обсчитывать управление. Для решения этой проблемы используется дополнительная плата Ардуино (Arduino Nano), которая будет использоваться в качестве модулятора и будет отвечать только за управление моторами. Данные Ардуины совмещены тремя проводами:

1. Сигнальный отвечает за индикацию отправки сигнала.
2. Вперёд соответствует отправки сообщения «вперёд».
3. Назад соответствует отправки сообщения «назад».

Данный метод взаимодействия позволяет реализовать управление с переключениями с максимального на максимальное движение в обратном направлении, а также устойчив к помехам.

Опытным путём мы нашли минимальные и максимальные частоты, соответствующие движению вперёд, а также нейтральное положение, и частоты, равные отклонению рычага в обратном положении (тормоз и назад). Для того чтобы после движения вперёд регулятор начал движение в противоположном направлении, необходимо перевести рычаг в нейтральное положение, затем перевести в зону назад. Регулятор начинает тормозить двигатель. После небольшой задержки, равной 60 мс, необходимо перевести в нейтральное положение и подождать такую же задержку, а затем переключить частоту, соответствующую движению назад. В связи с тем, что на данном переключении необходимо выполнить 3 задержки по 60 мс, возникает разница между математическим переключением и фактическим.

Для корректного взаимодействия Arduino и регулятора хода необходимо соединить земли, а после этого необходимо соединить с Ардуино, которая выполняет вычисления для корректного общения между ними. Во время большого потребления тока мотором возникают помехи по земле. В результате данных помех модуль акселерометра перезагружается и выходит в режим сна, в связи с чем, это делает невозможным его использование во время работы мотора.

Мощность мы ограничили 15% в связи с тем, что радиоуправляемая модель могла разгоняться до 60км/ч, что в случае потери контроля или ошибки могло привести к непоправимым последствиям. В дальнейшем мощность считается относительно этих 15%.

При замене двигателей на бесколлекторный и изменении цепи питания плата Arduino Romeo пришла в негодность и была заменена на плату Arduino Mega, которая имеет большее количество памяти, пинов и пинов прерывания с той же частотой процессора.

Для реализации одного из управлений необходимо передавать не переключение с максимального на противоположную мощность двигателя, а непрерывное (пропорциональное), то есть первая шина передачи данных не подходит. По этой причине была реализована вторая шина передачи данных, в последовательный порт (Serial) которой передаётся число от -100 до 100, что соответствует процентной мощности мотора.

## 2.6. Схема подключения.

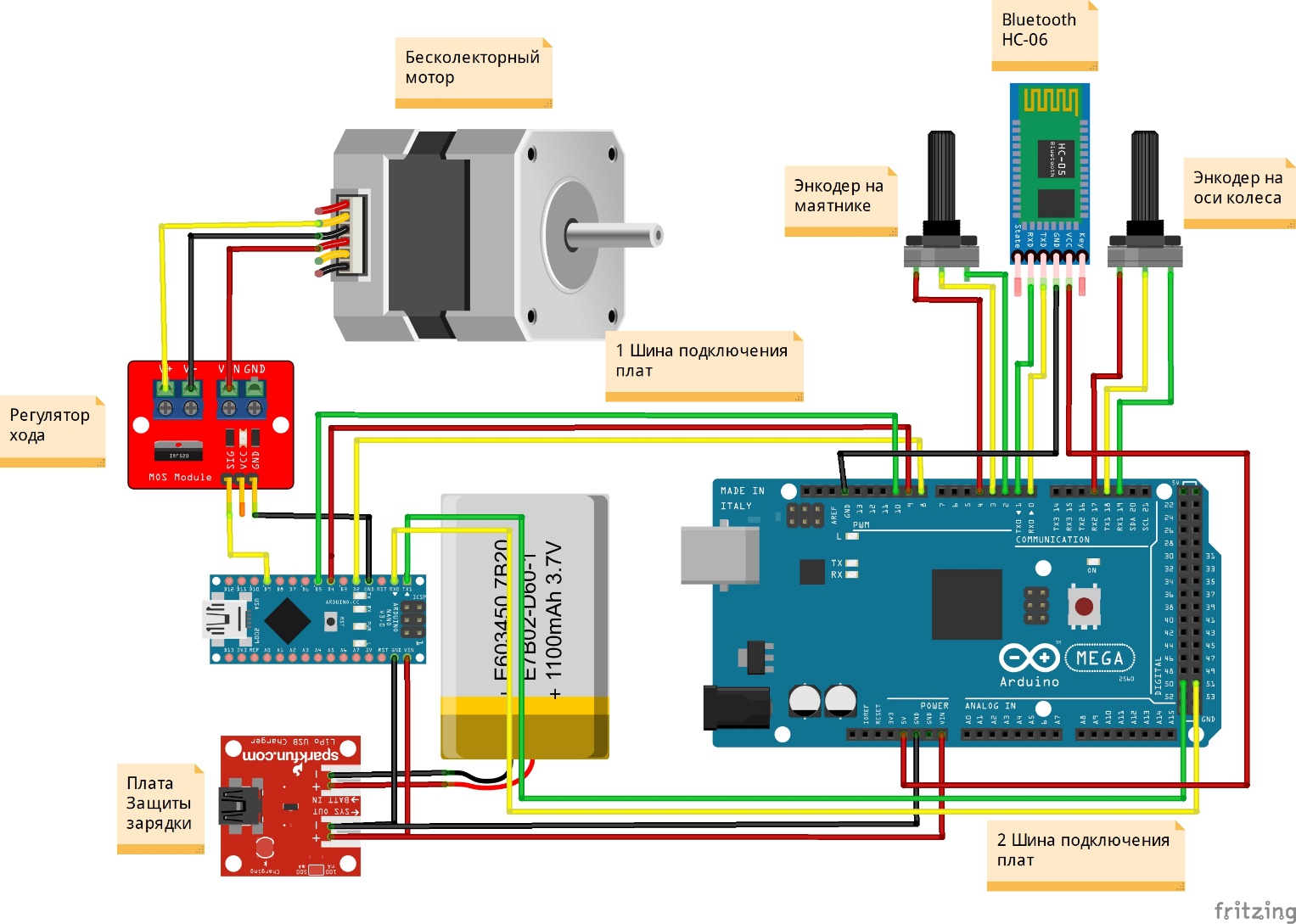
Общее построение электронной составляющей робота изображено на следующей схеме.

Рисунок 12. Принципиальная схема подключения модулей.

## 2.7. Программное обеспечение.

Для реализации найденного управления нужно найти, с какой стороны от поверхности находится система, а затем включить нужное управление. Поверхность задана параметрически, и аналитическое определение расположения точки относительно него невозможно. Для определения положения было решено нанести сетку точек на данную поверхность и определить положение по триангуляции по трём ближайшим точкам к заданной.

В Arduino есть возможность сохранения констант в Flash память, но при этом нельзя будет использовать 2 шину подключения плат.   
 Flash память Arduino Mega 2560 составляет 252 kB, из которых 40 kB занимает сама программа, а остальное можно выделить на переменные. В Arduino переменная типа float и double равны и занимают 4 байта памяти. Соответственно мы можем записать 16000 точек.

Сначала мы записали 1000 точек, что составляет примерно 30 измерений на каждую из переменных T1, Т2. Мы посчитали скорость выполнения обсчёта расстояний, триангуляции и определение расположения точки относительно поверхности. Расчёт расстояний до 1000 точек, поиск 10 ближайших, поиск самой малой выпуклой оболочки, содержащей точку, и вычисление расположения относительно выпуклой оболочки занимает 100 мс. За 100 мс робот может уехать на 10 см, так как ускорение на 50% мощности равно . Если учесть тот факт, что переключение занимает 180 мс, то управление на мотор будет подано спустя ~ 285 мс, что является большим лагом системы, который не позволяет подставить данное управление в робота.

# Выводы.

На основе проведенного исследования мы получили следующие результаты. Для вычисления угла и угловой скорости необходимо использовать энкодер, и в случае если энкодер имеет малое количество импульсов, то постоянная ошибка в показаниях энкодера может сильно повлиять на точность вычисления управления. Данное решение позволит избежать накопления ошибок, не будет требовать дополнительных вычислений, позволит узнавать положение маятника непрерывно, без задержек, в отличие от акселерометра, который имеет постоянную ошибку, а также ограничен по частоте считывания состояния (частота модуля 1kГц).

В связи с тем, что скорость рассчитывается численно из показаний энкодера, либо добавляется лаг в систему, либо дребезг в значение скорости. Оба этих нюанса приводят к дополнительной погрешности в вычислении.

Скорость расчёта положения системы относительно параметрически заданной поверхности занимает большое количество аппаратного времени на микроконтроллере на относительно небольшом количестве точек, что делает невозможным реализовать управление, где необходимо найти расположение до поверхности, которая является параметрической.

В случае использования бесколлекторных моторов с регулятором хода от радиоуправляемой машины, добавляется дополнительная задержка при переключении мощности мотора с направления вперёд на направление назад. Данная задержка вводит достаточно большое ограничение на управление.

# Заключение

В настоящей работе мы предприняли попытку реализовать оптимальное управление леанизированной задачи для тележки с однозвенным обратным маятником из общедоступных компонентов, модулей. Мы нашли управление, которое позволило бы  
стабилизировать перевёрнутый маятник на тележке за минимально возможное время, собрали платформу, создали коммутацию между всеми модулями и микроконтроллерами и разработали программное обеспечение для данных микроконтроллеров.

После запуска созданной нами модели мы пришли к выводу, что в использовании общедоступных модулей, плат и компонент от радиоуправляемых моделей есть целый ряд особенностей, которые надо предусмотреть в управлении, таких как задержка на переключение, задержка на вычисление управления и лаг в вычислении скоростей по энкодерам, в том числе и в вычислении ускорения двигателя для определения пределов максимального и минимального значения управления (ускорение платформы).

В данный момент микроконтроллеры рассчитаны на чтение данных с датчиков, управление разными модулями, двигателями, и другой атрибутикой. В случае использовании данных микроконтроллеров в обсчитывании большого количества точек, расстояний, триангуляций и иных сложных математических действий, возникает достаточно большая задержка, которая сказывается на точности рассчитанного управления, а также не позволяет использовать аналитические методы в их оригинальном виде. Следовательно, необходимо доработать аналитические решения или упростить их до линейных расчётов.

Примером таких расчётов является PID или LQR регуляторы, которые могут вывести систему в нужное положение, но имеют неоптимальное решение. PID регулятор требует дополнительный подбор необходимых коэффициентов для построения самого регулятора. LQR коэффициенты выводятся из матриц и в программе используются линейно, что позволяет очень быстро вычислять значение управления и управлять моторами непрерывно без каких-либо задержек.

Таким образом, мы считаем, что задачи нашего исследования выполнены, цель достигнута.

# Список использованной литературы.

1. Беляев С. А., Михнович А. Г. Современные подходы к решению задачи стабилизации перевернутого маятника //Программные продукты, системы и алгоритмы. – 2017. – №. 2. – С. 2-2.

2. Колесников А. А., Кондратьев И. В. Синергетическое управление механической системой «перевернутый маятник на тележке»: линейное преобразование координат //Известия Южного федерального университета. Технические науки. – 2001. – Т. 23. – №. 5.

3. Ллойд Э., Ледерман У. (ред.). Справочник по прикладной статистике. Том 2. М.: Финансы и статистика, 1990. - 526 с.

4. Сысоева С. Автомобильные датчики положения. Современные технологии и новые перспективы. Часть 5. Новые перспективы бесконтактных угловых измерений в диапазоне угла 360, снова датчики Холла-угловые магнитные энкодеры //Компоненты и технологии. – 2005. – №. 50.

5. Моряхин С. А., Астахова М. А., Андриенко Е. А. Создание автономного летательного аппарата типа «квадрокоптер» //Сборник трудов молодых ученых и сотрудников кафедры ВТ. – 2011. – С. 46.