

## **АННОТАЦИЯ**

Целью данной работы является разработка доступного и дешевого четырехногого робота для образовательных и исследовательских целей. Доступность достигается путем проектирования робота таким образом, чтобы его можно было собрать без каких-либо специальных инструментов, а все необходимые детали были либо напечатаны на 3D-принтере, либо легко изготовлены. Для снижения стоимости, вместо дорогих бесщеточных двигателей и контроллеров двигателей используются доступные сервоприводы. Для того чтобы конструкция робота была полезна для образования и исследований, она должна обладать большинством возможностей более крупных и дорогих экземпляров. Это достигается путем изучения существующих конструкций четырехногих роботов и выбора возможных компромиссов.

Конечная цель — разработать подходящую структуру управления, которая обеспечит автономную работу робота для демонстрации возможностей аппаратного обеспечения.

В данной работе достигнуты следующие цели:

1. Решены задачи траекторного движения на основе численного решения обратной задачи кинематики.
2. Спроектированы твердотельные чертежи и произведена сборка физической единицы робота.
3. Разработана модель движения робота с учетом обратной связи, основанной на данных с инерционного датчика.

Результат работы представлен в виде работоспособного физического прототипа шагающего робота с возможностью дистанционного управления.

# Содержание

<b>ВВЕДЕНИЕ . . . . .</b>	<b>5</b>
<b>ГЛАВА 1 . . . . .</b>	<b>8</b>
1.1. Анализ существующих моделей шагающих роботов . . . . .	8
1.2. Задачи работы . . . . .	11
1.3. Исследование особенностей конструкции робота . . . . .	12
<b>ГЛАВА 2 . . . . .</b>	<b>14</b>
2.1. Общая схема робота . . . . .	14
2.2. Построение рабочей области ноги робота . . . . .	15
<b>ГЛАВА 3 . . . . .</b>	<b>18</b>
3.1. Общие требования к физической модели четырехногого робота	18
3.2. Задача синтеза походки четырехногого шагающего робота .	20
3.3. Решение задачи для одной ноги. Опорное положение . . . . .	22
3.4. Решение задачи для одной ноги. Свободное положение . . . . .	28
<b>ГЛАВА 4 . . . . .</b>	<b>33</b>
4.1. Проектирование ног . . . . .	33
4.2. Описание корпуса робота . . . . .	36
4.3. Выбор комплектующих . . . . .	37
4.3.1. Выбор сервоприводов . . . . .	37
4.3.2. Выбор управляющей электроники . . . . .	40
4.3.3. Выбор инерционного датчика . . . . .	42
4.3.4. Выбор источника питания . . . . .	44
4.3.5. Результаты проектирования и сборки . . . . .	45
<b>ГЛАВА 5 . . . . .</b>	<b>48</b>
5.1. Общая архитектура . . . . .	48
5.2. Обратная связь с помощью гироскопа . . . . .	49

5.3. Движение робота вперед с обратной связью . . . . .	53
<b>ЗАКЛЮЧЕНИЕ</b> . . . . .	55
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b> . . . . .	56
<b>ПРИЛОЖЕНИЕ А</b> . . . . .	58
<b>ПРИЛОЖЕНИЕ Б</b> . . . . .	59
<b>ПРИЛОЖЕНИЕ В</b> . . . . .	62

## **ВВЕДЕНИЕ**

Особенное место в робототехнике занимает подкласс шагающих роботов. Шагающими роботами принято называть класс роботов, которые имитируют походку людей или животных. Данный вид появился не просто так, их главной особенностью является как особенность походки, так и в принципе уникальная модель движения. Она позволяет быстро и, что главное, эффективно приспосабливаться к работе в неровных поверхностях [1]. По сути, оптимизация движения шагающих роботов является главной проблемой всех ведущих исследователей в данной области.

Все известные производители шагающих роботов не используют четко описанные законы движения для управления в своих продуктах. Вместо этого, они используют заранее сформированные модели глубокого обучения, основанные на больших пластиах информации, собранной с реальных экземпляров. Например, для управления (обучения) четырехногих шагающих роботов использовали массивы данных, полученных с датчиков, которые в свою очередь были закреплены на собаке, схожей по физическим размерам конечностей и тела [2].

Такие методы, как правило показывают себя более выгодными как с точки зрения точности позиционирования робота, скорости адаптации в окружении, так и выигрывают по энергоэффективности в сравнении с классическими методами, но имеют существенный недостаток в скорости разработки и в особенности требуют больших денежных затрат.

## **Актуальность**

Задача разработки шагающих роботов не теряет актуальности по сей день. Начиная с 2004 года, когда Boston Dynamics впервые представила четырехногого шагающего робота BigDog [3], промышленность в данной области не сбавляет обороты и постоянно вносит новые технологии, методики в разработку, а также находит новые применения данным роботам. Прогресс можно заметить при сравнении с современными моделями Boston Dynamics или, например LaikaGo и Strelka от UniTree [4] (рисунок 1), которые используют на кафедре робототехники в МГТУ им. Н. Э Баумана [5].



Рис. 1. BigDog и Spot от Boston Dynamics, Laika от UniTree

Сегодня такие мобильные роботы используются для составления строительных планов здания по технологии BIM (Building Information Modeling) (рисунок 2), отслеживания хода строительства на площадках Pomerlau, а также на заводах Ford. Такие продукты стали востребованы для мониторинга оборудования в опасных условиях, например, нефтепромышленная компания AkerBP использовала робота Spot для снятия показаний и отслеживания утечек на судне FPSO в Норвегии [6, 7].

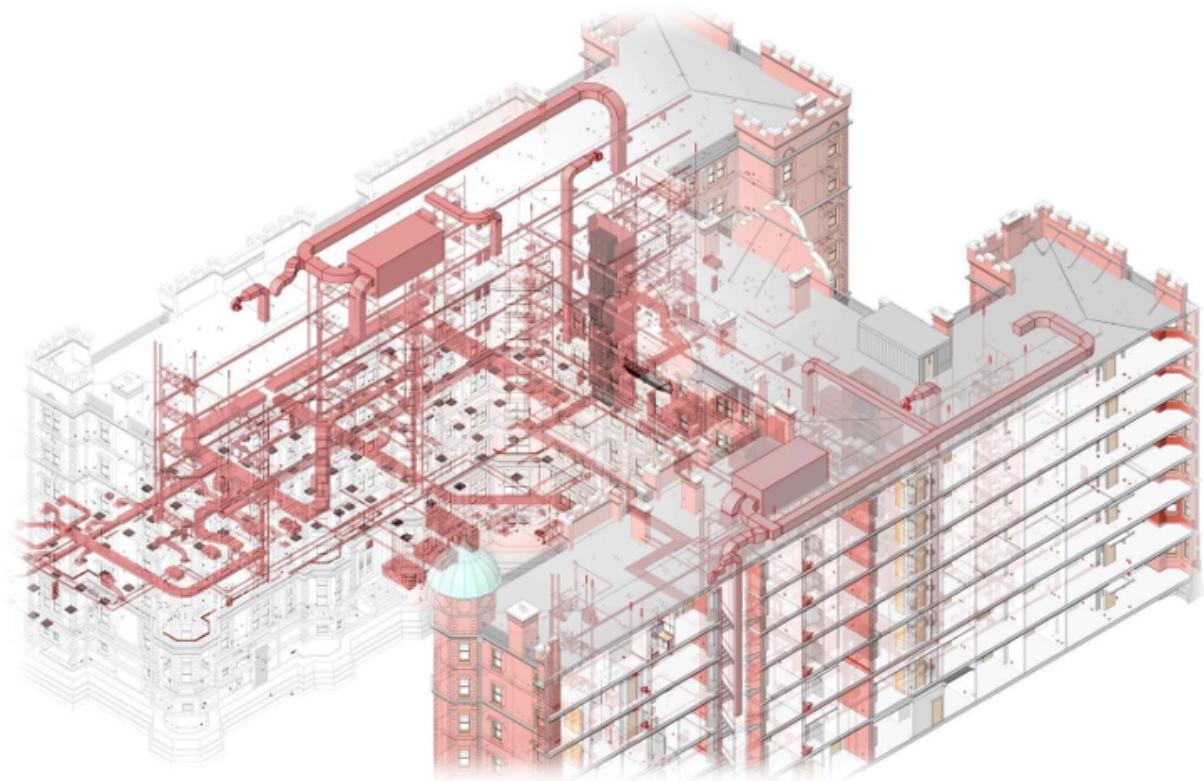


Рис. 2. Строительный план по технологии BIM

# **ГЛАВА 1**

## **ПОСТАНОВКА ЗАДАЧИ И ПЛАН РЕШЕНИЯ**

### **1.1. Анализ существующих моделей шагающих роботов**

Автономных роботов можно условно разделить на две категории: стационарные роботы и мобильные роботы. Одним из типов стационарных робототехнических систем является манипулятор, который размещает поверхностно установленные компоненты с чрезвычайно высокой точностью [8], т.е. в сборочной линии, рука робота способна перемещаться с высокой скоростью и точностью для выполнения повторяющихся задач, таких как точечная сварка [9] и покраска [10].

Другим важным типом стационарных роботизированных систем является устройство схватка [11], которое обычно используется для захвата нужных объектов. В настоящее время разработано множество роботизированных механических устройств имитирующие руки и пальцы, которые широко применяются, например, в сельском хозяйстве [12]. К сожалению, эти стационарные роботы страдают от недостаточной подвижности и ограниченного диапазона движения. В отличие от них, мобильные роботы способны передвигаться по пересеченной местности. Благодаря гибкости мобильных роботов, они широко применяются во многих областях, таких как строительство, охрана объектов, сбор информации. В этих сценариях применения мобильные роботы могут быть классифицированы на наземные, воздушные и подводные. Кроме того, в соответствии с различными механическими структурами, наземных роботов можно разделить на колесные, гусеничные и шагающие роботы, соответственно.

На неровной местности для перемещения больше подходят шагающие роботы, в силу их способности адаптироваться к местности. Кроме того, они также могут применяться областях, далеких от промышленности, например медицина [13, 14] и образование [15]. Таким образом, шагающие роботы стали актуальной темой для исследований в области робототехники, благодаря их

преимуществу перед колесными и гусеничными роботами.

Рассмотрим несколько моделей шагающих роботов, на основе которых проведем анализ конструкции (раздел 1.3).

Mini Cheetah [16] — это четырехногий робот, разработанный в Массачусетском технологическом институте (рисунок 1.1). Основное внимание в роботе уделяется его модульным приводам высокой мощности. Mini Cheetah обладает двенадцатью одинаковыми приводами, которые состоят из бесщеточного двигателя, планетарного редуктора с передаточным числом 6:1, энкодера и контроллера двигателя. Все приводы Mini Cheetah обмениваются данными по единой шине CAN-bus.



Рис. 1.1. Mini Cheetah, разработанный в МИТ

Stanford Doggo [17] — квази-прямоходящий четырехногий робот, разработанный в Стэнфорде. Квази-прямоходящий - термин, описывающий движение, которое почти прямолинейно, но с некоторой степенью отклонения от прямой линии. Данный робот обладает очень высокой выходной мощностью по сравнению с его весом и низкой ценой. Такой высокий крутящий момент достигается за счет квазипрямого привода и меньшего количества степеней свободы на каждой ноге.

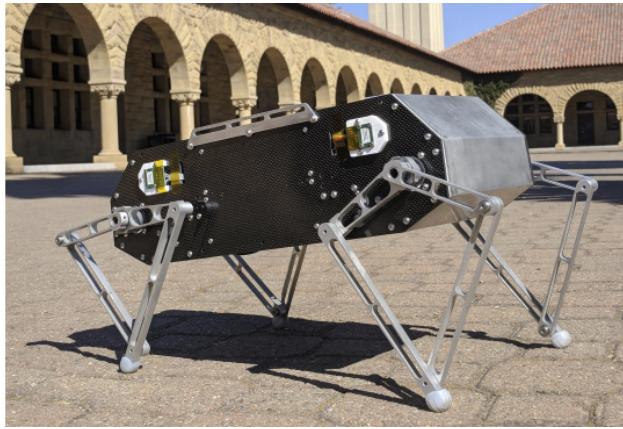


Рис. 1.2. Stanford Doggo, разработанный в Стэнфордском университете

Spot [18]—шагающий робот, разработанный компанией Boston Dynamics, представляет из себя четвероногого робота, предназначенного для промышленного рынка. По этой причине он обладает более широкими возможностями, чем другие четвероногие роботы, так как создан для выполнения бизнес задач. Он обладает хорошей подъемной силой, имеет возможность установки дополнительных датчиков и инструментов на верхней части корпуса, а также использует камеру для технического зрения с целью сканирования окружающей обстановки.



Рис. 1.3. Spot, разработанный в компании Boston Dynamics

## 1.2. Задачи работы

Создание функционирующего прототипа шагающего робота, приближенного к промышленным моделям, является существенно сложным и дорогостоящим процессом. В связи с этим, для достижения поставленных целей, в данной работе будут применены упрощения, которые окажут влияние на некоторые технические аспекты, однако не повлекут за собой потерю общей ценности исследования - разработки действующего прототипа с готовым управлением, полученным путем решения задач механики и управления без использования методов искусственного интеллекта.

Для выполнения данного проекта потребуется выполнить следующие работы:

1. Разработать кинематическую схему ноги робота.
2. Разработать общую кинематическую схему робота.
3. Решить обратную задачу кинематики для положения ноги.
4. Решить задачу описания модели походки робота.
5. Разработать твердотельную модель ног и корпуса робота.
6. Подобрать необходимые сервоприводы в сочленения ног робота.
7. Подобрать комплектующие, связанные с реализацией контроля сервоприводами.
8. Разработать программное обеспечение для дистанционного управления роботом.
9. Автоматизировать внутренние процессы прототипа.
10. Создать графический интерфейс для управления.

В настоящей работе будут тщательно рассмотрены пункты **1-6** и описаны пункты **7-8**, пункты **9-10** рассматриваться не будут.

### **1.3. Исследование особенностей конструкции робота**

Среди всех компонентов робота, ноги являются наиболее важными составляющими. Такие свойства, как вес, размеры и приведение в действие суставов, непосредственно влияют на производительность движения ног и, следовательно, точность позиционирования робота.

У упомянутых ранее моделей роботов схожи комплекции как корпуса, так и ног, даже общие объемы роботов будут схожи. Каждая нога состоит из тазобедренного сустава, бедра, коленного сустава, голени и стопы для контакта с землей, то есть все ноги обладают тремя степенями свободы, а принципы расположения двигателей в ногах имеют минимальное расхождение. Это следствие того, что промышленные модели сделаны из металла и действительно требуют расположения тяжелых двигателей как можно ближе к корпусу для большей устойчивости и меньших затрат на управление.

В данном проекте робот не обладает существенным весом, а также большими размерами корпуса и ног, в следствие чего будут следующие упрощения:

- На одну ногу приходится две степени свободы, вместо распространенной схемы в виде трех степеней.
- Двигатели располагаются в области бедра (у корпуса) и колена (в сочленении ног), так как робот обладает небольшой массой и малыми габаритами для классического решения (рисунок 1.4).

Указанные упрощения могут привести к утрате возможности контролировать отклонения бедра, что в свою очередь подчеркивает необходимость разработки методов компенсации устойчивости при изменениях углов ориентации корпуса в плоскостях крена и тангажа.

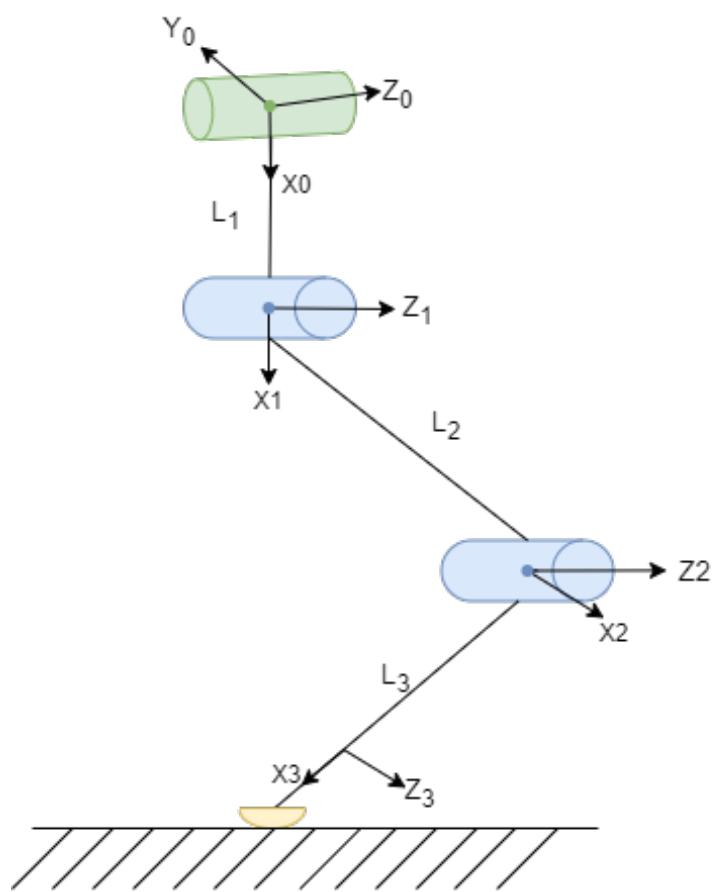


Рис. 1.4. Расположения двигателей для случая трех степеней свободы на ногу

## ГЛАВА 2

# КИНЕМАТИЧЕСКАЯ СХЕМА РОБОТА

### 2.1. Общая схема робота

В разделе 1.3 были определены требования к кинематической конфигурации робота. В целях создания прототипа ноги предложена двухзвенная механическая система, оснащенная двумя двигателями. Первый двигатель расположен ближе к корпусу робота и обеспечивает ориентацию положения первого звена - имитации бедра робота. Второй двигатель связывает первое и второе звено, которое, в свою очередь, имитирует голень робота и завершается элементом, аналогичным стопе. Результирующая кинематическая схема ноги представлена на рисунке 2.1.

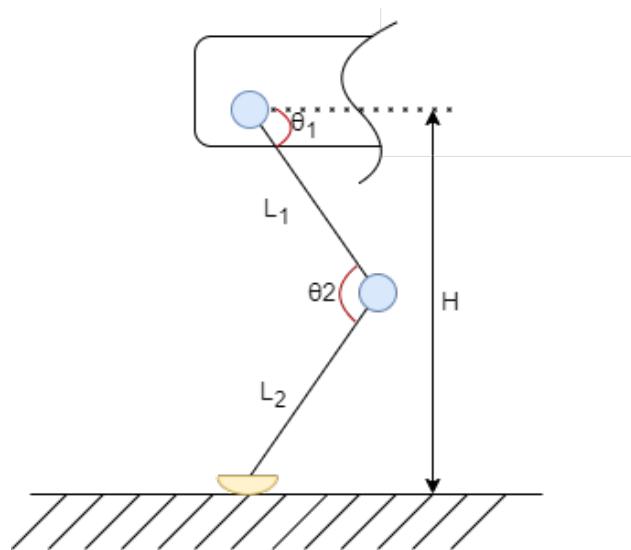


Рис. 2.1. Кинематическая схема ноги робота

Таким образом, робот имеет четыре одинаковые ноги, каждая из которых имеет две степени свободы. Шарниры передних и задних ног направлены в противоположные стороны и закреплены на туловище, как показано на рисунке 2.2. Степени свободы распределены между плечевой частью (тазобедренный сустав) и коленным суставом.

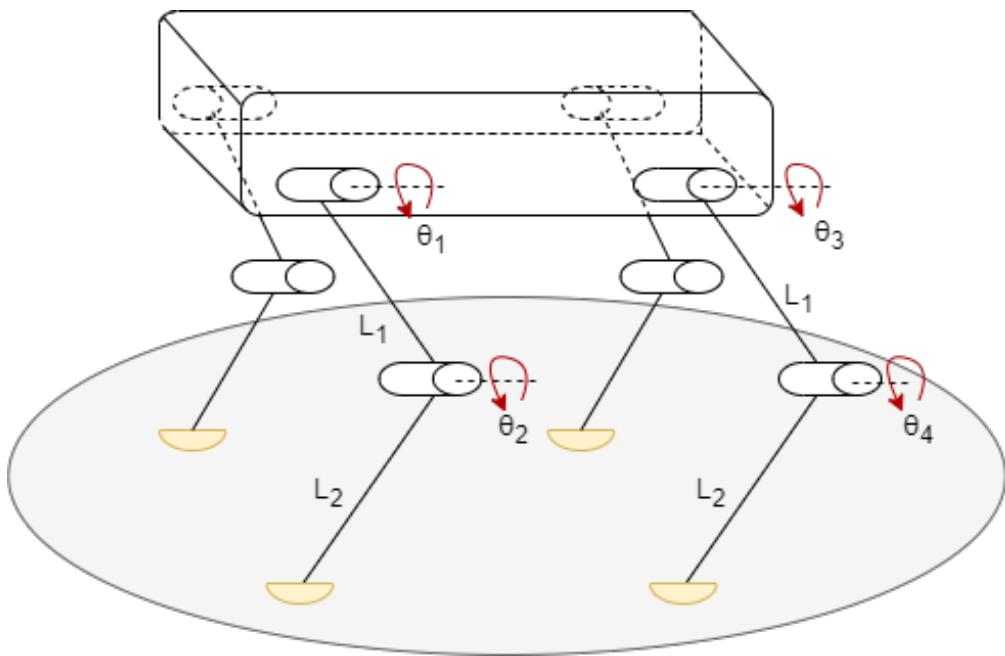


Рис. 2.2. Полная кинематическая схема робота

## 2.2. Построение рабочей области ноги робота

Для того, чтобы решить обратную задачу кинематики, необходимо определять положения углов с течением времени для совершения шага по некоторой траектории, следовательно, необходимо понимать рабочую область исследуемого объекта, в данном случае ноги. Общие характеристики ноги сведены в таблице 2.1

*Таблица 2.1*  
**Характеристики ног робота**

Общий вес	0.2 кг
Длина первого звена	100 мм
Длина второго звена	100 мм
Диапазон поворота двигателя (бедро)	$\pm 180^\circ$
Диапазон поворота двигателя (колено)	$\pm 180^\circ$

Чтобы описать рабочую область, составим кинематическую схему, представленную на рисунке 2.3 и решим задачу о положении стопы ноги. Точку  $O$  будем считать зафиксированной на корпусе.

Тогда получим уравнения для положения точки  $B$  относительно корпуса робота:

$$\begin{aligned} X_B &= l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2), \\ Y_B &= l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2). \end{aligned} \quad (2.1)$$

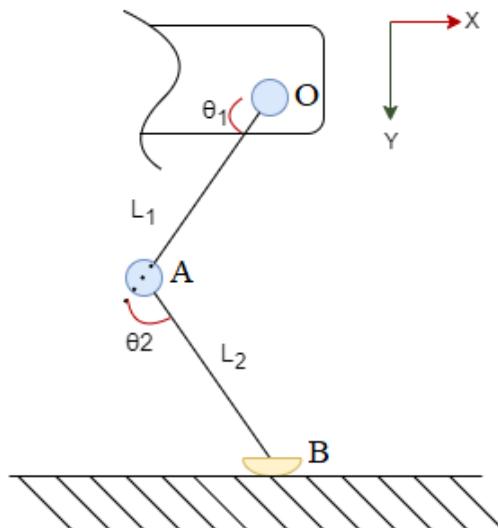


Рис. 2.3. Кинематическая схема для определения рабочей области ноги

Используя данные уравнения, которые описывают положение точки  $B$  относительно корпуса, а также используя информацию об ограничениях на углы поворота двигателя из таблицы 2.1, определим рабочую область ноги (рисунок 2.4). Данная информация позволит нам определить доступные границы для ширины шага робота. Программа реализована в математическом пакете Wolfram Mathematica (Приложение А).

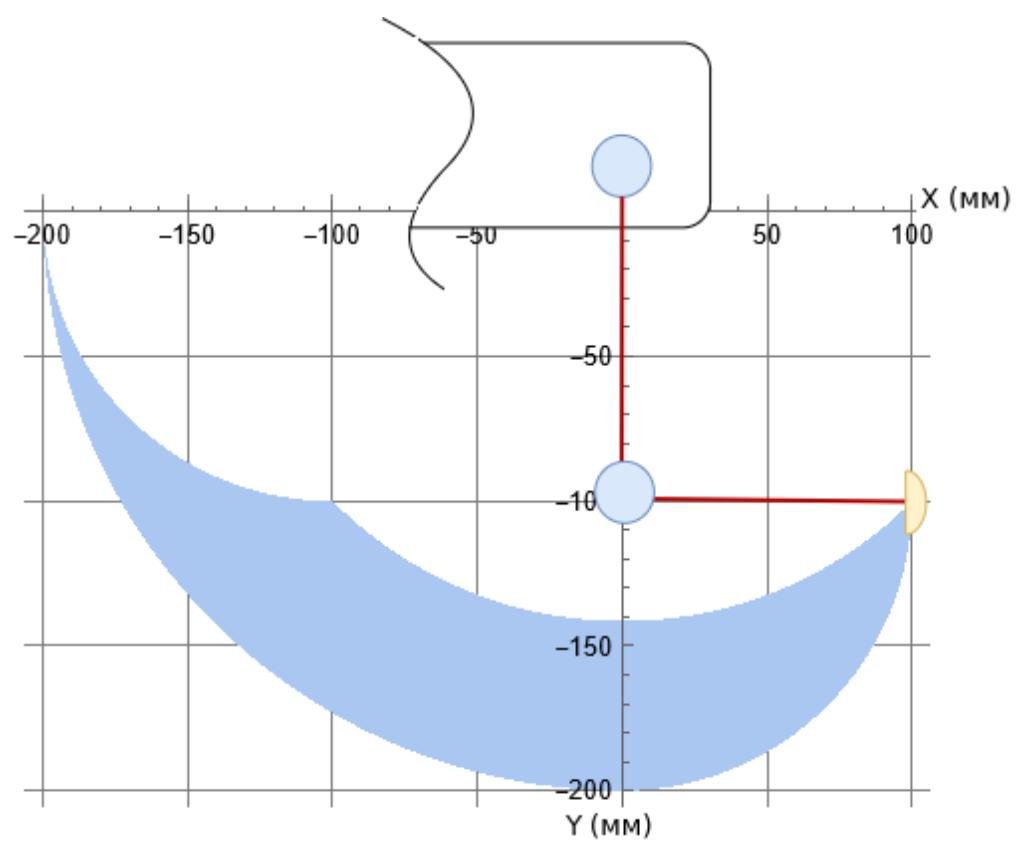


Рис. 2.4. Рабочая область ноги робота относительно корпуса

## ГЛАВА 3

# РЕШЕНИЕ ОБРАТНОЙ ЗАДАЧИ КИНЕМАТИКИ

### 3.1. Общие требования к физической модели четырехногого робота

Все четвероногие животные при движении сохраняют равновесие почти исключительно за счет динамической устойчивости. Изначально план чередования опорных и свободных положений ног заключался в переключении фаз по принципу ноги 1-4 опорная фаза, ноги 2-3 свободная, как показано на рисунке 3.1. Зелеными стрелками обозначены фазы свободного положения, а красными опорного. Однако, такой подход не совсем точен, хоть и вполне логичен.

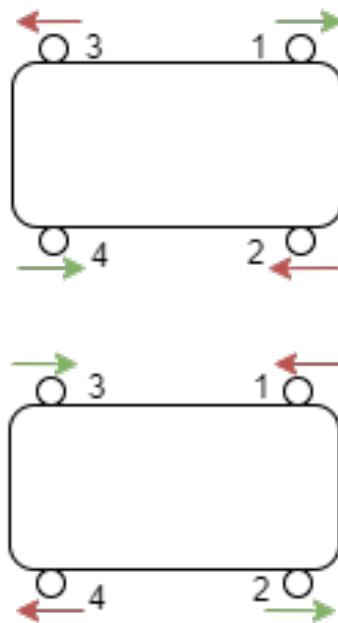


Рис. 3.1. Смена опорных и свободных фаз.

В случае искусственных шагающих аппаратов походка может быть определена таким образом, чтобы центр тяжести аппарата постоянно находился

внутри треугольника, вершинами которого являются конечности, находящиеся на данный момент времени в опорном положении (рисунок 3.2). На практике эта возможность была реализована д-ром Такути из Научно-исследовательского центра проблем механики [19]. Под его руководством был разработан шагающий аппарат с четырьмя конечностями, у которых скорость движения в фазе восстановления подобрана так, что длительность этой фазы втрое меньше длительности каждой рабочей фазы. В результате в данный момент времени лишь одна нога робота находится в воздухе, а корпус опирается на три остальные, сохраняя тем самым статическую устойчивость.

Показанные на рисунке 3.2 заштрихованные треугольники (так называемые опорные треугольники) образованы вершинами, которые соответствуют текущим точкам касания опорной поверхности какими-либо тремя из четырех ног робота. Например, в состоянии, соответствующем рисунку 3.2, опоры касаются три ноги - А, В, С, а четвертая нога - D, будучи в фазе восстановления, находится в воздухе. Для обеспечения статической устойчивости принципиально важное значение имеет правильный выбор порядка чередования (сдвиг фаз) четырех ног в процессе движения аппарата. В момент времени, отраженный на диаграмме 1, нога В только что коснулась земли и сейчас занимает опорное положение, нога А также касается земли, но находится уже во второй половине своей рабочей фазы, а нога С - еще в первой половине собственной рабочей фазы. Нога D, находящаяся в момент наблюдения в воздухе, быстро заканчивает фазу восстановления, и к моменту времени, которому соответствует диаграмма 2, она уже опускается на землю, переходя в опорное положение. К этому моменту нога А завершила свою рабочую фазу и, оторвавшись от земли, переходит в фазу восстановления; нога В находится в первой половине, а нога С - во второй половине своих рабочих фаз. Аналогично в следующие моменты времени, которым соответствуют диаграммы 3 и 4, в фазу восстановления переходят друг за другом нога С и нога В.

При таком порядке чередования ног в любой момент времени, когда одна из ног робота находится в фазе восстановления, центр тяжести аппарата обязательно будет лежать внутри треугольника, образованного тремя ногами, находящимися не в рабочей фазе (опирающимися на землю).

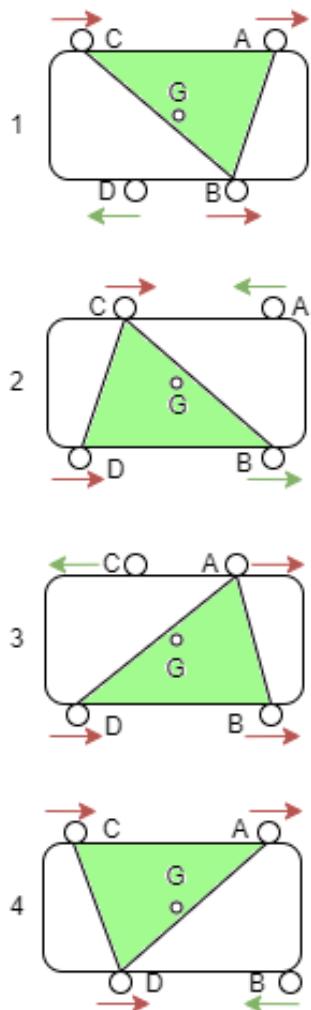


Рис. 3.2. Смена опорных и свободных фаз с учетом тяжести

### 3.2. Задача синтеза походки четырехногого шагающего робота

Процесс движения шагающего аппарата, в котором осуществляется работа ног, можно определить как ходьбу. В ходе данного процесса каждая из ног может находиться в одном из двух основных состояний, отличающихся друг от друга принципиально.

В процессе ходьбы, ноги шагающего аппарата попеременно занимают то опорное, то свободное положения, причем в течение одного цикла каждая нога занимает то и другое положение один раз. Последовательность, в которой при ходьбе происходит чередование опорного и свободного положе-

жений всех ног аппарата, определяет конкретный способ его передвижения или его походку. Последовательность чередований ног за один период называется циклом ходьбы, а расстояние, которое проходит аппарат за один цикл, — шагом [19]. Движения, совершаемые любой из ног при чередовании опорного и свободного положений в процессе ходьбы, можно упрощенно разделить на фазу поступательного перемещения и фазу восстановления. В простейшем случае переход от первой фазы ко второй может осуществляться за счет движений ноги вверх-вниз строго по вертикали. Однако энергия шагающего аппарата будет расходоваться более эффективно, если в течение фазы восстановления нога будет перемещаться по некоторой кривой. Одна из возможных форм криволинейной траектории стопы относительно корпуса показана на рис 3.3.

Разделим предстоящую задачу формирования походки на два этапа:

1. Опорное положение (перенос корпуса) — в это время нога касается поверхности и служит опорной для корпуса аппарата;
2. Свободное положение (перенос ноги) — в это время нога находится над поверхностью и готовится к выполнению опорных функций на следующем шаге.

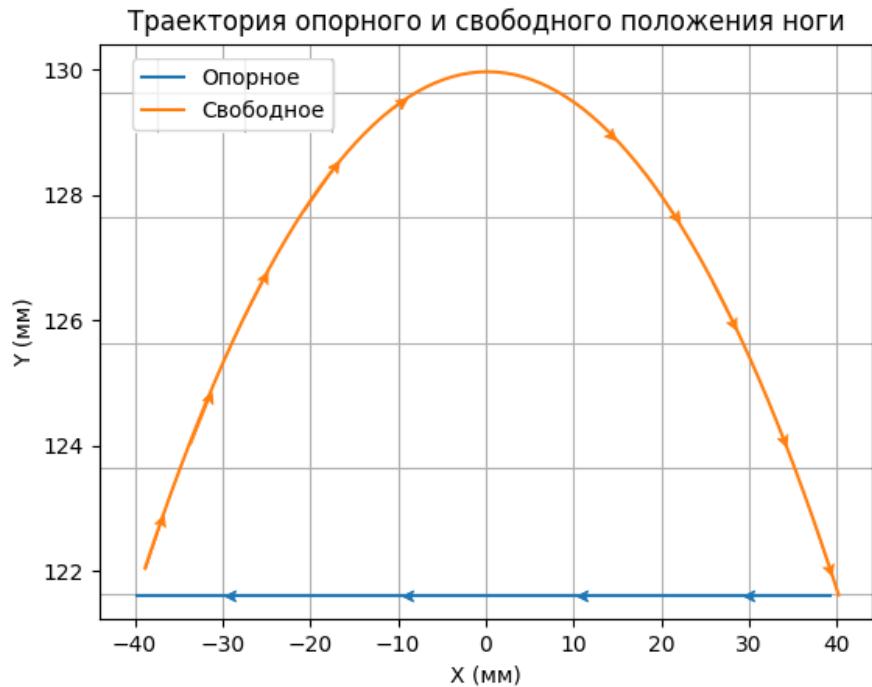


Рис. 3.3. Траектория относительно корпуса в фазах опорного и свободного положения ноги во время движения

### 3.3. Решение задачи для одной ноги. Опорное положение

Рассмотрим подробно первый этап. Нам известны длины звеньев  $L_1$  и  $L_2$ , а также высота от поверхности земли до корпуса  $h_{\text{корп}}$ . Начальные условия для шага:

$$\begin{aligned} X &= +X_{\text{ш}}, \\ Y &= 0. \end{aligned} \tag{3.1}$$

Конечные условия для шага:

$$\begin{aligned} X &= -X_{\text{ш}}, \\ Y &= 0, \end{aligned} \tag{3.2}$$

где  $X_{\text{ш}}$  — максимальное значение для переноса корпуса во время опорной фазы,  $-X_{\text{ш}}$  — максимальное значение для переноса ноги во время свободной фазы. Отрезок от  $(-X_{\text{ш}})$  до  $X_{\text{ш}}$  — полная ширина шага (см. Приложение А и

рисунок 2.4).

Таким образом, положение ноги в начальный момент времени можно отобразить согласно рисунку 3.4.

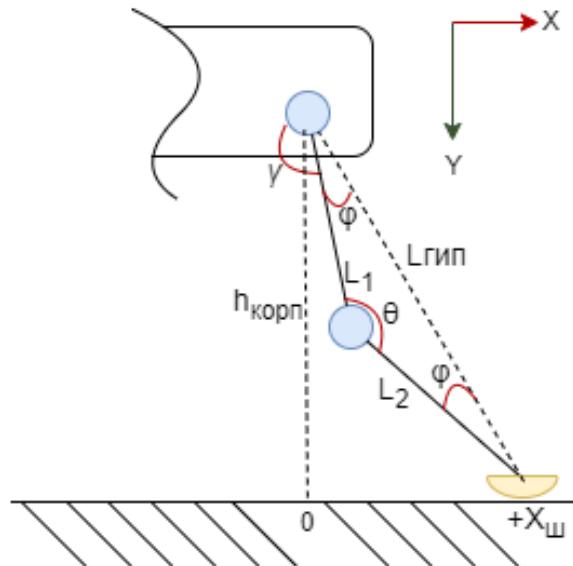


Рис. 3.4. Положения ноги относительно корпуса в начальный момент времени

В данной задаче необходимо определить закон изменения углов движателей в бедре  $\gamma$  и колене  $\theta$  для положительного переноса корпуса по оси  $X$ . Исходя из рисунка 3.4, получим следующие уравнения для  $\gamma$ :

$$\begin{aligned} \phi + \gamma - \frac{\pi}{2} &= \operatorname{arcctg} \left( \frac{+X_{ш}}{h_{корп}} \right), \\ \gamma &= \frac{\pi}{2} + \operatorname{arcctg} \left( \frac{+X_{ш}}{h_{корп}} \right) - \phi. \end{aligned} \quad (3.3)$$

Соответственно для  $\theta$  получим:

$$\theta = 2 \arcsin \left( \frac{l_{гип}}{2l_1} \right), \quad (3.4)$$

где вспомогательный угол  $\phi$  и длину  $l_{\text{гип}}$  можно найти как:

$$\begin{aligned}\phi &= \arccos \left( \frac{l_{\text{гип}}}{2l_1} \right), \\ l_{\text{гип}} &= \sqrt{x_{\text{ш}}^2 + h_{\text{корп}}^2}.\end{aligned}\quad (3.5)$$

Далее рассмотрим конечный момент времени, где условия шага описаны формулой 3.2. Конфигурация ноги в таком случае примет положение, которое отображено на рисунке 3.5.

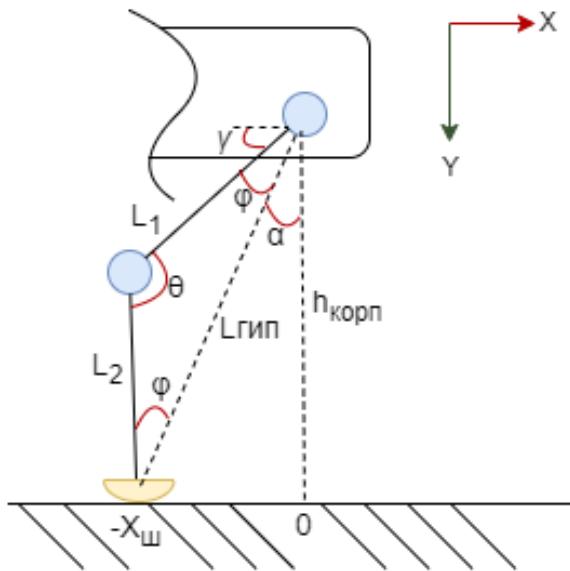


Рис. 3.5. Положения ноги относительно корпуса в конечный момент времени

Определим значения, которые принимают  $\gamma$ ,  $\theta$  и вспомогательный угол  $\alpha$  в конечный момент времени переноса корпуса.

$$\begin{aligned}\gamma &= \frac{\pi}{2} - \phi - \arctg \left( \frac{-X_{\text{ш}}}{h_{\text{корп}}} \right) = \\ &= \frac{\pi}{2} - \arccos \left( \frac{l_{\text{гип}}}{2l_1} \right) + \arctg \left( \frac{-X_{\text{ш}}}{h_{\text{корп}}} \right), \\ \theta &= 2 \arcsin \left( \frac{l_{\text{гип}}}{2l_1} \right), \\ \alpha &= \arctg \left( \frac{-X_{\text{ш}}}{h_{\text{корп}}} \right).\end{aligned}\quad (3.6)$$

Заметим, что законы изменения углов двигателей отличаются только знаком  $X_{ш}$ . Таким образом, определим общий закон изменения углов в зависимости от  $X_{ш}$ . Иначе говоря получим решение обратной задачи кинематики для одного шага робота, но без переноса стопы:

$$\begin{aligned}\theta(x_{ш}) &= 2 \arcsin \left( \frac{\sqrt{x_{ш}^2 + h_{корп}^2}}{2l_1} \right), \\ \gamma(x_{ш}) &= \frac{\pi}{2} - \arccos \left( \frac{\sqrt{x_{ш}^2 + h_{корп}^2}}{2l_1} \right) + \arctg \left( \frac{X_{ш}}{h_{корп}} \right), \\ l_{тип} &= \sqrt{x_{ш}^2 + h_{корп}^2}.\end{aligned}\tag{3.7}$$

Напишем небольшой скрипт на языке Python, чтобы построить графики изменения углов в двигателях в зависимости от ширины шага  $X_{ш}$ :

```
def calculate_step(x_range_step, height):
    y_step = 0
    height_slew = np.sqrt(x_range_step**2 + (height - y_step)**2)
    hip_step = 0.5 * np.pi - np.arccos(height_slew / (2 * .11)) + np.arctan(
        x_range_step / (height - y_step))
    knee_step = 2 * np.arcsin(height_slew / (2 * .11))

    return np.rad2deg(hip_step), np.rad2deg(knee_step)
```

Получим следующие результаты для значений  $X_{ш}$  от  $-40$  мм до  $40$  мм с шагом 1 мм. Отобразим их на рисунке 3.6. Для этого вызовем функцию в исполнительном файле:

```

import numpy as np
import matplotlib.pyplot as plt

x_range_step = np.arange(40,-40,-10)
hip_step = np.zeros(np.size(x_range_step))
knee_step = np.zeros(np.size(x_range_step))

for i in range(x_range_step):
    hip,knee = callculate_step(x_range_step[i])
    hip_step[i] = hip
    knee_step[i] = knee

plt.plot(hip_step)
plt.plot(knee_step)

```

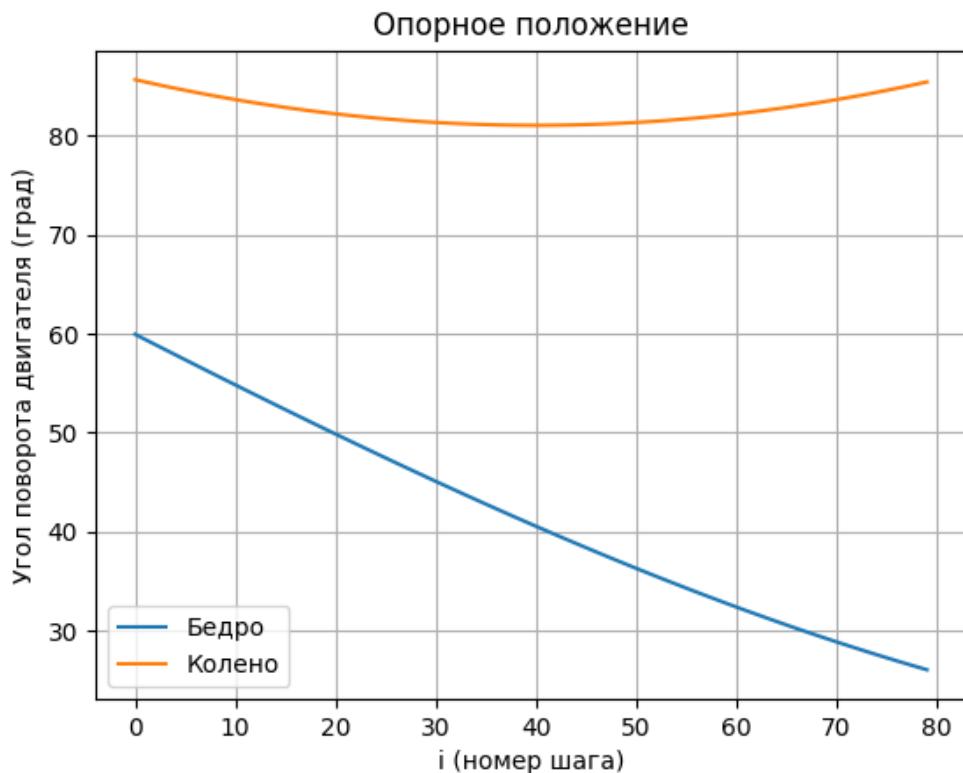


Рис. 3.6. График изменения углов бедра и колена во времени

Проверим, что в момент фазы опорного положения ноги, ее высота не изменялась и задача о положении ноги для переноса корпуса решена верно. Если задача решена верно, то результатом уравнения 3.8, при построении графика должна быть горизонтальная прямая (рисунок 3.7 ), по значению равная  $h_{\text{корп}}$ , которая означает, что крайняя точка ноги (стопа) находилась на максимальном заданном расстоянии от корпуса и не изменилась в течение фазы опорного положения:

$$H = l_1 \cos(\frac{\pi}{2} - \gamma) + l_2 \sin(\theta - \gamma). \quad (3.8)$$

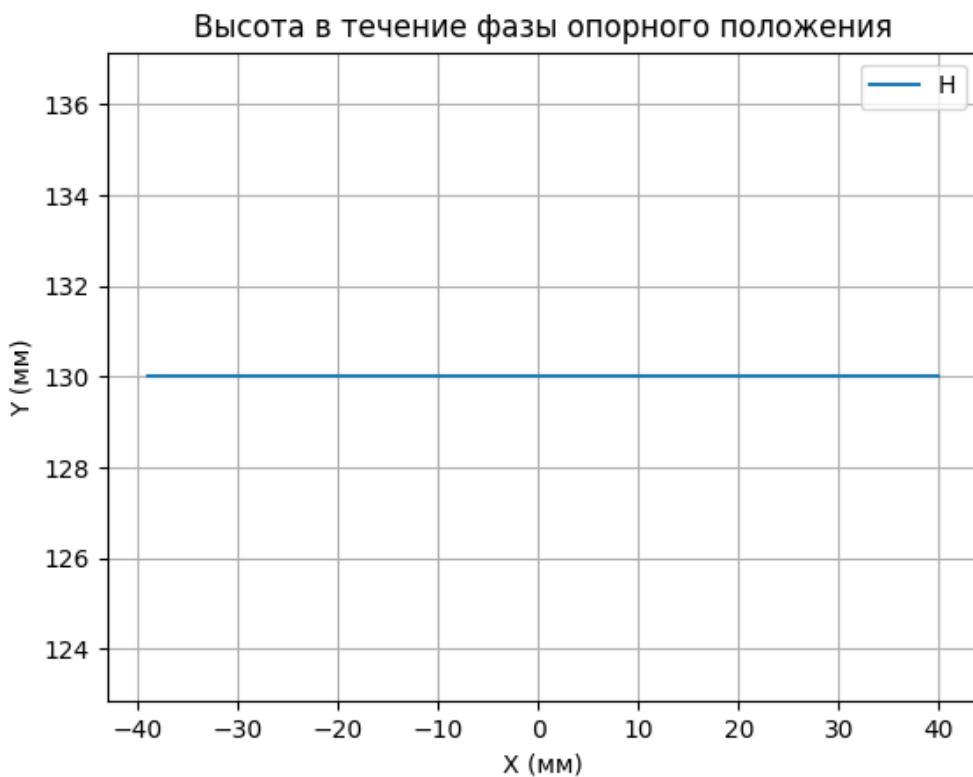


Рис. 3.7. График высоты стопы при опорном положении

### 3.4. Решение задачи для одной ноги. Свободное положение

Фаза переноса — это один из ключевых этапов движения четырехногого робота, который происходит между фазами опоры. Во время переноса конечность робота находится в воздухе и перемещается в новую позицию для следующего шага. Эта фаза требует от робота сбалансированности и точности движений, чтобы избежать падений и обеспечить плавность передвижения. Иначе говоря, фазу переноса можно рассматривать как переходный этап между двумя фазами опоры, в котором робот готовится к следующему шагу.

Рассмотрим подробно второй этап. Нам известны длины звеньев  $L_1, L_2$  и высота от поверхности земли до корпуса  $h_{\text{корп}}$ . Условия для свободного положения включают в себя два краевых: начальное и конечное, а также серединное.

Начальные условия:

$$\begin{aligned} X &= -X_{\text{ш}}, \\ Y &= 0. \end{aligned} \tag{3.9}$$

Конечные условия:

$$\begin{aligned} X &= +X_{\text{ш}}, \\ Y &= 0. \end{aligned} \tag{3.10}$$

Серединные условия:

$$\begin{aligned} X &= 0, \\ Y &= h_{\text{под}}, \end{aligned} \tag{3.11}$$

где  $X_{\text{ш}}$  — максимальное значение для переноса корпуса во время опорной фазы,  $-X_{\text{ш}}$  — максимальное значение для переноса ноги во время свободной фазы. Отрезок от  $-X_{\text{ш}}$  до  $X_{\text{ш}}$  — полная ширина шага (см. Приложение А и рисунок 2.4).  $h_{\text{под}}$  — высота подъема ноги, ограничения на которую необходимо найти.

Рассмотрим кинематическую схему на рисунке 3.8. На нем изображена конфигурация ноги в трех положениях, которые описывают условия 3.9, 3.10, 3.11.

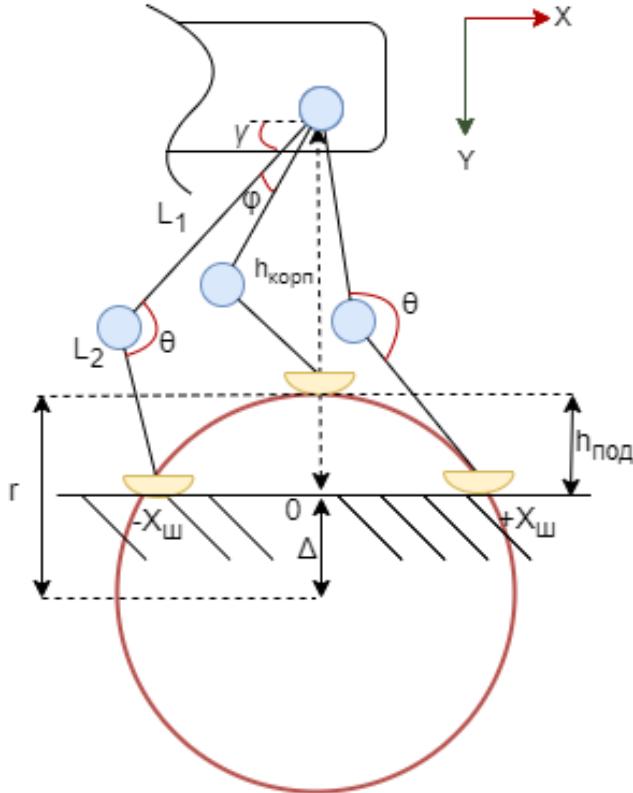


Рис. 3.8. Последовательность переноса стопы в свободном положении

Перенос стопы по дуге окружности не является оптимальным решением, однако нам неизвестны динамические характеристики робота, так как это прототип. Следовательно, на начальном этапе достаточно выбрать удобную для реализации траекторию без оптимизации. В таком случае выполнять оптимизацию предпочтительнее на завершающем этапе работы. Конечно, существуют алгоритмы поиска таких траекторий, где расчет переноса стопы выполняется с опорой на машинное обучение с подкреплением, что требует отдельного исследования [20]. В настоящей работе движение происходит на ровной поверхности, таким образом с помощью регулирования радиуса  $r$  создадим механизм, позволяющий регулировать высоту переноса стопы  $h_{\text{под}}$ , которая будет реализуема для заданной конфигурации робота, с учетом известных длин звеньев  $l_1$ ,  $l_2$  и высоты  $h_{\text{корп}}$ . Исходя из известных данных, вполне справедливо, что радиус можно найти как:

$$r = \sqrt{x_{\text{ш}}^2 + \Delta^2} = \Delta + h_{\text{под}}, \quad (3.12)$$

где  $x_{ш}$  - задаваемая ширина шага от  $-X_{ш}$  до  $+X_{ш}$ ,  $\Delta$  — расстояние, которым контролируется положение центра окружности по высоте.

Таким образом высота подъема стопы  $h_{под}$  определяется из уравнения:

$$\begin{aligned} h_{под}(x_{ш}) &= \sqrt{r^2 - x_{ш}^2} - \Delta, \\ \Delta &= \sqrt{r^2 - x_{ш}^2}, \end{aligned} \quad (3.13)$$

где ограничение на высоту подъема ноги вводится как:

$$r > x_{ш}$$

Тогда с учетом найденного  $h_{под}$ , определим законы изменения углов в двигателях для бедра и колена соответственно:

$$\begin{aligned} \theta(x_{ш}) &= 2 \arcsin \left( \frac{\sqrt{x_{ш}^2 + h_{корп}^2 - h_{под}^2}}{2l_1} \right), \\ \gamma(x_{ш}) &= \frac{\pi}{2} - \arccos \left( \frac{\sqrt{x_{ш}^2 + h_{корп}^2 - h_{под}^2}}{2l_1} \right) + \arctg \left( \frac{X_{ш}}{h_{корп}} \right), \\ l_{гип} &= \sqrt{x_{ш}^2 + h_{корп}^2}, \\ h_{под}(x_{ш}) &= \sqrt{r^2 - x_{ш}^2} - \Delta, \\ \Delta &= \sqrt{r^2 - x_{ш}^2}. \end{aligned} \quad (3.14)$$

Аналогично с пунктом 3.3 напишем скрипт на языке Python, чтобы получить графики изменения углов в двигателях в зависимости от ширины шага  $X_{ш}$ :

```
def calculate_skip(x_range_skip, min_delta, height, r):
    delta = np.sqrt(r**2 - min_delta**2)
    y_skip = np.sqrt((r**2) - (x_range_skip**2)) - delta
    height_slew = np.sqrt(x_range_skip**2 + (height - y_skip)**2)

    hip_skip = 0.5 * np.pi - np.arccos(height_slew / (2 * l1)) + np.arctan(
        x_range_skip / (height - y_skip))
    knee_skip = 2 * np.arcsin(height_slew / (2 * l1))
```

```
    return np.rad2deg(hip_skip), np.rad2deg(knee_skip)
```

С учетом результатов на рисунке 3.6, получим графики углов  $\theta$  и  $\gamma$  при переносе стопы для значений  $X_{ш}$  от  $-40$  мм до  $40$  мм с шагом 1 мм, высотой  $h_{корп} = 130$  мм и  $r = 100$  мм. Отобразим их на рисунке 3.9. Для этого добавим функцию в исполнительный файл, описанный выше в пункте 3.3:

```
import numpy as np
import matplotlib.pyplot as plt

x_range_step = np.arange(40, -40, -10)
x_range_skip = np.arange(-40, 40, 10)
r = 100
height = 130
min_delta = x_range_skip[0]
hip = []
knee = []

for i in range(x_range_step):
    hip, knee = calculate_step(x_range_step[i], height)
    hip[i] = hip
    knee[i] = knee

for i in range(x_range_skip):
    hip, knee = calculate_skip(x_range_skip, min_delta, height, r)
    hip[i] = hip
    knee[i] = knee

plt.plot(hip)
plt.plot(knee)
```

Проведем проверку, аналогичную в пункте 2.2. Если задача решена верно, то мы получим результат схожий с рисунком 3.3. Для этого воспользуемся формулой 3.8. Полученный график отображен на рисунке 3.10

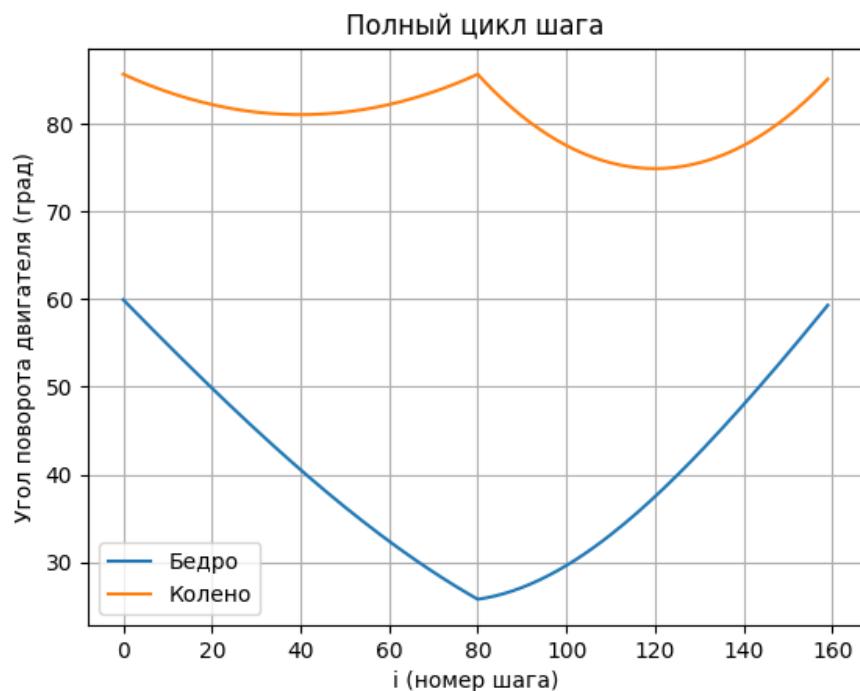


Рис. 3.9. Изменение углов двигателей за полный цикл одного шага

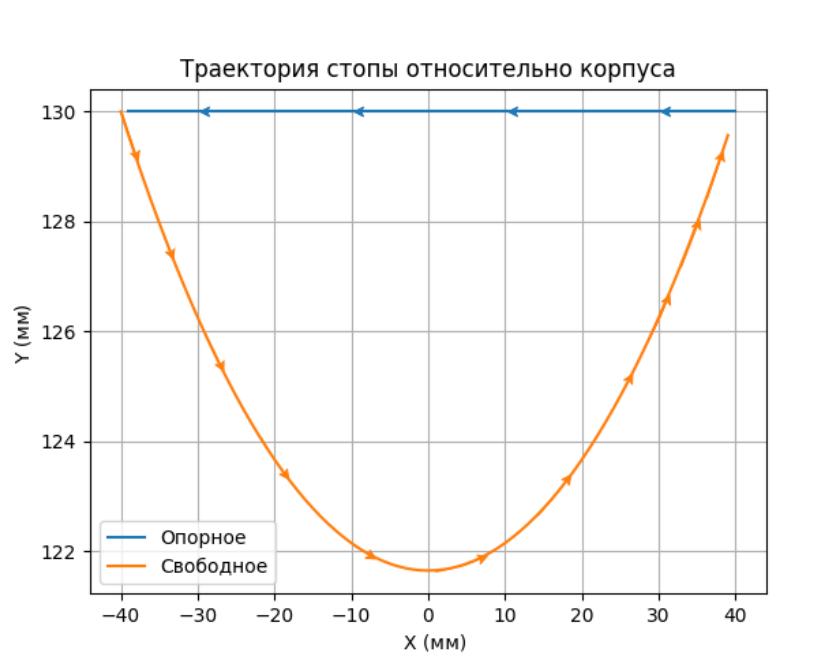


Рис. 3.10. Траектория стопы относительно корпуса за полный шаг

## **ГЛАВА 4**

### **МОДЕЛИРОВАНИЕ И СБОРКА**

#### **4.1. Проектирование ног**

В разделе 1.3 были представлены особенности конструкции и основные проблемы, которые необходимо решить перед производством деталей.

Для минимизации проблем, связанных с соосностью отверстий, отношениями и масштабами между конструкциями перед этапом печати, были использовали трехмерные твердотельные чертежи при проектировании ног в формате 3D (рисунок 4.1). В качестве материала для изготовления деталей ног был выбран пластик, который используется при печати с помощью 3D-принтера, поскольку уникальная конструкция ног не позволяет использовать доступные металлические изделия для сборки. Кроме того, использование пластика значительно снижает вес конструкции, что позволяет использовать менее мощные двигатели для создания прототипа. Напечатанные и собранные конструкции ног отображены на рисунках 4.3, 4.4. Характерные размеры сведены в таблице 4.1.

*Таблица 4.1*

#### **Характерные размеры робота**

Длина первого звена	100 мм
Длина второго звена	100 мм
Длина лонжерона	210 мм
Длина корпуса	210 мм
Ширина корпуса	160 мм

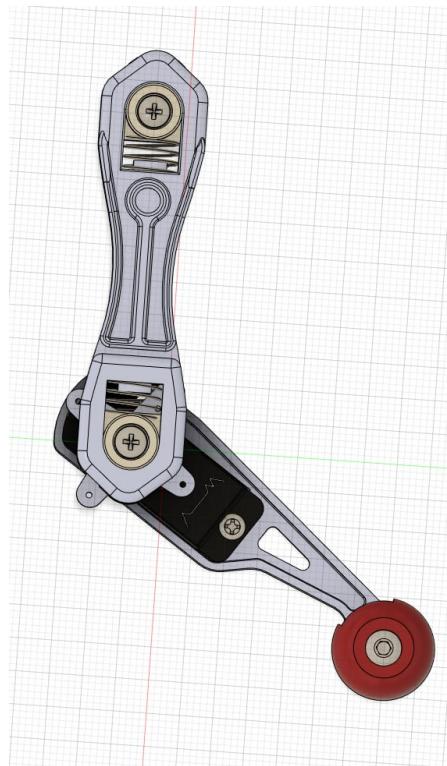


Рис. 4.1. Твердотельный чертеж ноги робота

Сервоприводы, как в сочленении ног, так и в месте соединения с корпусом, закреплены к ползуну, который в свою очередь подпирается пружиной. Ползун обладает “Т” образной формой (рисунок 4.2), которая поддерживает его в плоскости ноги и не дает покинуть область конструкции.

Пружина служит в качестве демпфера, который гасит ударные воздействия на вал двигателя при движении робота, а также оказывает некоторое полезное сопротивление: при резком изменении угла поворота двигателя, не дает сервоприводу развить максимальную скорость, что не раз спасало прототип при натурном тестировании походки.

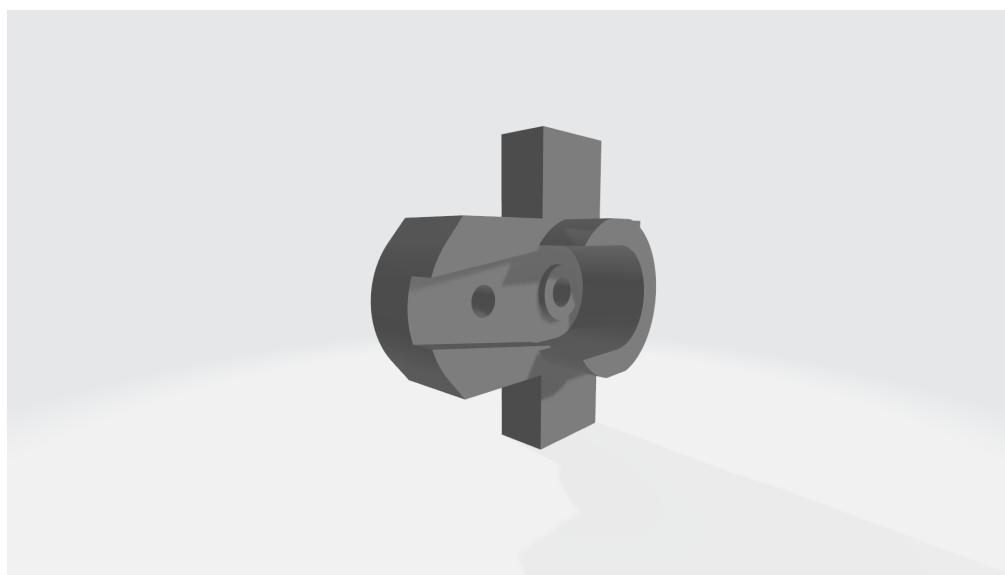


Рис. 4.2. “Т” - образный ползун

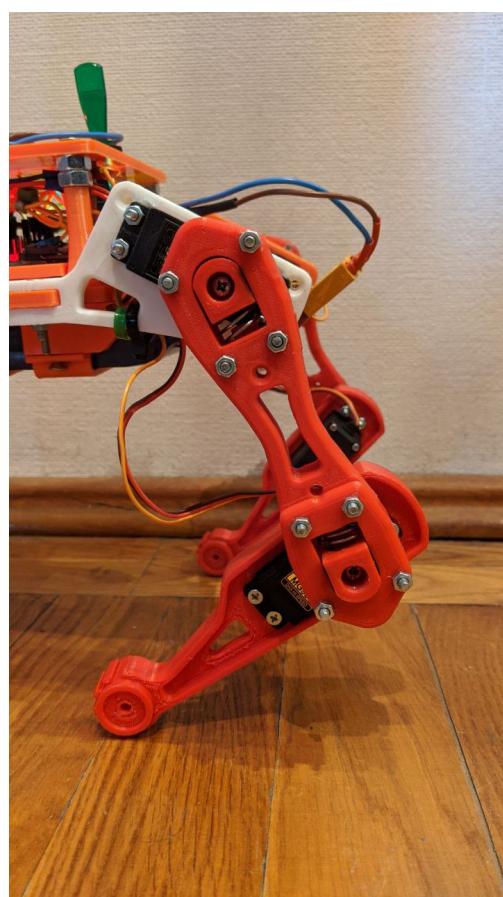


Рис. 4.3. Напечатанная нога в сборке

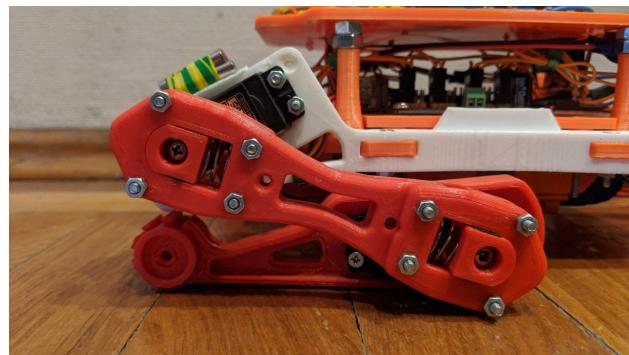


Рис. 4.4. Робот в исходном положении до инициализации

## 4.2. Описание корпуса робота

Корпус для робота представляет из себя поперечные пластины (рисунок 4.5), соединенные с лонжеронами. Данные пластины несут функцию размещения электронных компонентов и создания жесткости конструкции.

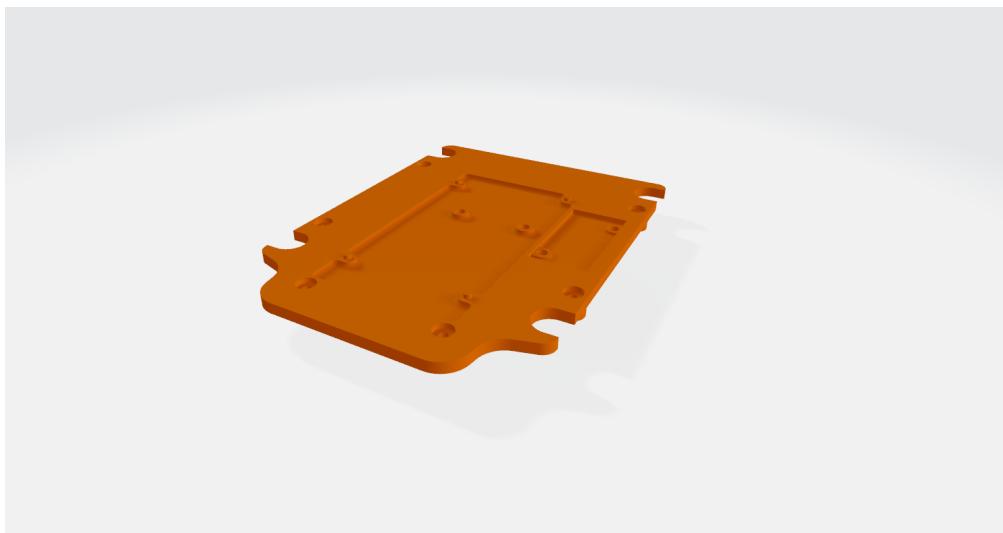


Рис. 4.5. Твердотельный чертеж пластины корпуса

В качестве детали к которой крепятся ноги робота, были выбраны лонжероны, также изготовленные из пластика (рисунок 4.6). Данное решение было принято из-за легкости материала и достаточной жесткости, необходимой для удержания закрепленных деталей.

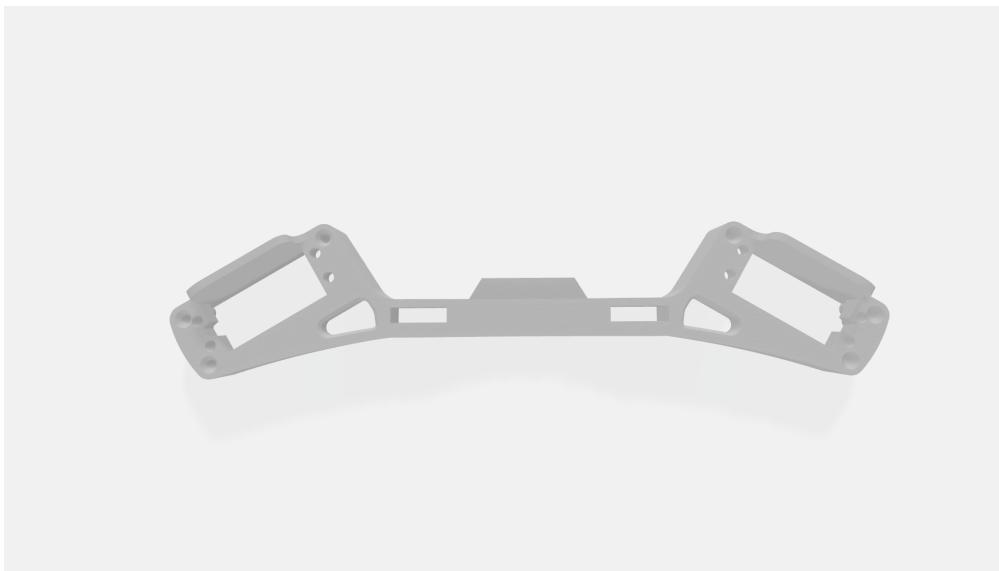


Рис. 4.6. Твердотельный чертеж лонжерона корпуса

В отличие от ног, к корпусу были только минимальные требования, а именно небольшой вес и удобное расположение модулей для последующего подключения(подробные чертежи размещены в Приложении Б). На верхнем уровне корпуса расположены DC/DC преобразователи и трехосевой гироскоп акселерометр MPU6050. На нижнем уровне располагается миникомпьютер OrangePi 3 LTS и ШИМ контроллер PCA9685, в то время как с обратной стороны пластины закреплен источник питания в виде Li-Po аккумулятора (более подробно о подборе комплектующих в разделе 4.3).

## 4.3. Выбор комплектующих

### 4.3.1. Выбор сервоприводов

Сервопривод - это механический привод, который содержит датчик (для измерения положения, скорости, усилия и других параметров) и блок управления приводом (который может быть электронной схемой или механической системой тяг), что позволяет автоматически поддерживать необходимые параметры на датчике в соответствии с заданным внешним значением (например, положение ручки управления или численное значение от других систем).

Сервопривод можно рассматривать как автоматического точного исполнителя, который получает на вход значение управляющего параметра в режиме реального времени и используя показания датчика, стремится создать и поддерживать это значение на выходе исполнительного элемента.

В данной работе предлагается рассчитать максимальный крутящий момент при худшем случае конфигурации робота. При выборе сервопривода данный параметр особенно значим, так как все остальные характеристики вроде габаритов и мест размещения крепежей у этих типов двигателей мало отличаются. Для того чтобы рассчитать худший случай нагрузки и выбрать подходящий сервопривод, необходимо знать следующие параметры:

- Общий вес робота;
- Длины ног.

Худшей конфигурацией является случай, когда все ноги робота вытянуты в "струнку" (рисунок 4.7). В таком случае серводвигателям необходимо приложить максимальный крутящий момент для возврата в рабочую конфигурацию.

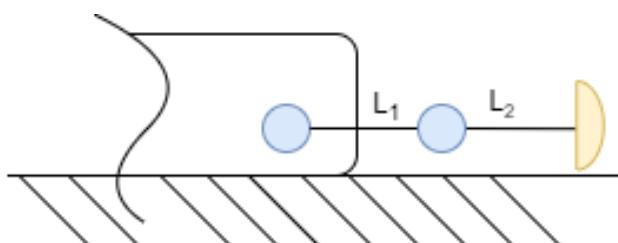


Рис. 4.7. Худшая конфигурация для статического случая

Учитывая параметры из таблицы 2.1, а также взвесив корпус, с заранее установленными комплектующими, вес робота составляет 1.6 кг. Будем считать, что движения ног для возвращения в исходную конфигурацию будет происходить синхронно, то есть общую нагрузку можно разделить на четыре ноги, что составляет 0.4 кг на одну ногу. Предлагается следующая формула для вычисления максимального момента на одну ногу в случае худшей

конфигурации:

$$M_{\max}^{\text{крут}} = \frac{m_{\text{роб}}g}{4}(l_1 + l_2). \quad (4.1)$$

Исходя из формулы 4.1 был получен крутящий момент равный 0.8 Нм. В настоящей работе выбрана модель сервопривода MG995, которая полностью удовлетворяет требованиям. Основные характеристики данного привода сведены в таблице 4.2.

*Таблица 4.2*  
**Характеристики сервопривода MG995**

Общий вес	55г
Размеры (ШxВxГ)	54x47.2x20мм
Крутящий момент	8.5кгс/см (4.8В), 10кгс/см (6В)
Рабочая скорость	0.2с/60° (4.8В), 0.16с/60° (6В)
Рабочее напряжение	4.8В - 7.2В
Рабочая температура	0 С° - 55 С°
Зона нечувствительности ШИМ	5мкс

Одним из самых главных минусов при работе с сервоприводами является гибкость управления валом двигателя. Резкие изменения угла в серводвигателях являются следствием специфики их конструкции, а если быть точнее, то данная проблема связана с принципом работы потенциометра, который выполняет роль энкодера внутри двигателя. Таким образом принцип регулирования угла поворота состоит в подаче ШИМ сигнала посредством написанного программного обеспечения, сам сигнал преобразуется в напряжение благодаря работе потенциометра и приводит вал в заданное положение с максимальной скоростью. Из этого можно сделать вывод, что в используемых сервоприводах не имеется обратной связи, и при использовании с тяжелыми объектами будут создаваться лишние нагрузки, которые будут приводить к потреблению тока, особенно это важно по отношению к стартовым токам. В нашем случае такое поведение потенциально приведет к уничтожению редуктора внутри двигателя (рисунок 4.8), так как движимые объекты инерционные и такое управление для них неприемлемо. Решение этой проблемы подробной разобрано в главе 5.2.

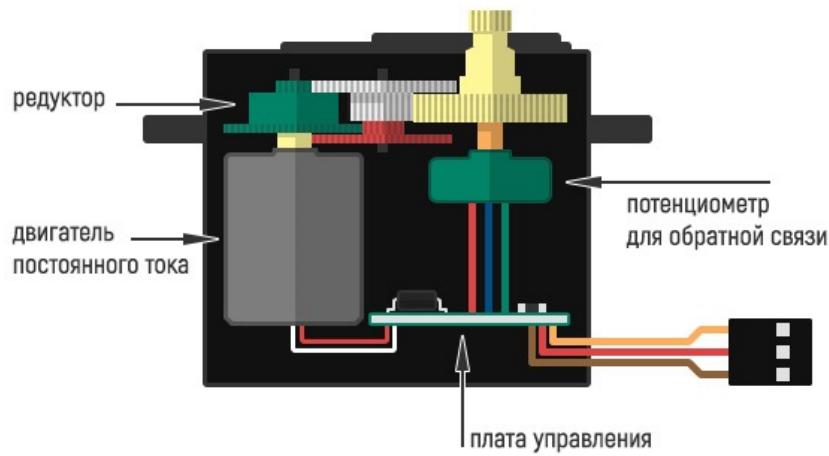


Рис. 4.8. Схема сервопривода

Еще одним последствием такого принципа работы выступает усложнение корректной сборки, так как для правильной конфигурации ноги на прототипе сервоприводам требуется калибровка. Иначе говоря, необходимо выставлять крайние положения ног и рассчитывать диапазон ШИМ сигнала и принимаемого угла, чтобы как можно точнее задавать положение ноги в будущем. Это достаточно неприятное следствие, так как каждая единица серводвигателя уникальна и требует отдельной калибровки. Методы калибровки в данной работе разбираются не будут.

#### 4.3.2. Выбор управляющей электроники

В качестве управляющей платы миникомпьютера была выбрана OrangePi 3 LTS (рисунок 4.9). Данный миникомпьютер покрывает все технические требования прототипа сверх нормы, что позволяет не задумываться о методах опрашивания датчиков или скорости обработки информации. Также весомым преимуществом такого компьютера является автономность и наличие встроенной ОС Linux. Такие особенности позволяют использовать плату в привычном для разработчика режиме и не терять время на настройку программного окружения.



Рис. 4.9. OrangePi 3 LTS

Так как OrangePi не поддерживает управление восемью серводвигателями одновременно, было предложено использовать ШИМ контроллер. В качестве данного контроллера был выбран модуль PCA9685 (рисунок 4.10). PCA9685 - это высокопроизводительный драйвер сервоприводов, разработанный фирмой NXP Semiconductors. Он обладает шестнадцатью независимыми каналами широтно-импульсной модуляции (PWM), каждый из которых может управлять сервоприводом. Также PCA9685 имеет встроенную функцию автоматического обновления адресов, что позволяет подключать несколько драйверов к однойшине I<sub>2</sub>C. Такой подход позволяет назначить уникальный адрес для каждого подключения, чтобы добиться одновременного управления большим количеством сервоприводов. Драйвер также поддерживает частоту ШИМ до 1 кГц, что обеспечивает высокую точность управления сервоприводами.

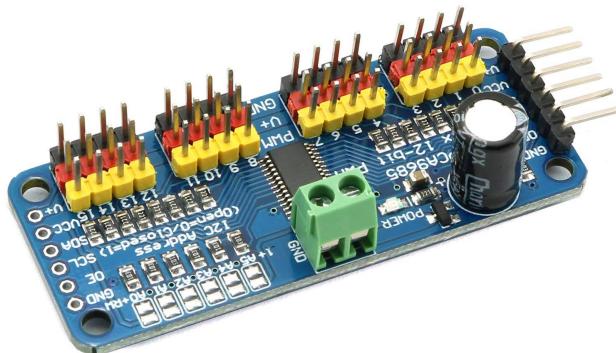


Рис. 4.10. PCA 9685

### 4.3.3. Выбор инерционного датчика

В ходе разработки прототипа возникла проблема с отсутствием данных о реальном угле поворота двигателя. Главной причиной послужило отсутствие обратной связи у сервоприводов, о чём уже упоминалось в разделе 4.3.1. Было предложено реализовать обратную связь с помощью использования блока гироскопов-акселерометров MPU6050 (рисунок 4.12).

Принцип работы такого модуля заключается в преобразовании аналогового напряжения, считываемого с емкостных датчиков, которые затем преобразуются в цифровой сигнал в диапазоне от 0 до 32750 значений. Эти значения составляют единицы измерения для блока гироскопов-акселерометров. MPU6050 распределяет свои единицы измерения, создавая четыре уровня чувствительности, как показано на рисунке 4.11. Выбранный уровень чувствительности зависит от требований к точности датчика. Например, если робот собирается выполнять высокоскоростные вращения более чем на  $1000^\circ$  в секунду (167 об / мин), то следует установить чувствительность гироскопа на  $2000^\circ$ . В таком режиме гироскопу приходится преодолевать большую площадь вращения за очень короткий промежуток времени, ему необходимо

## Understanding MPU6050 Raw Values

Gyro measurement units = 32,750

$32,750 / \text{degrees} = \text{sensitivity}$

e.g.  $32,750 / 250^\circ = 131$  measurement units per degree

Accelerometer measurement units = 32,750

$32,750 / g = \text{sensitivity}$

e.g.  $32,750 / 2g = 16,384$

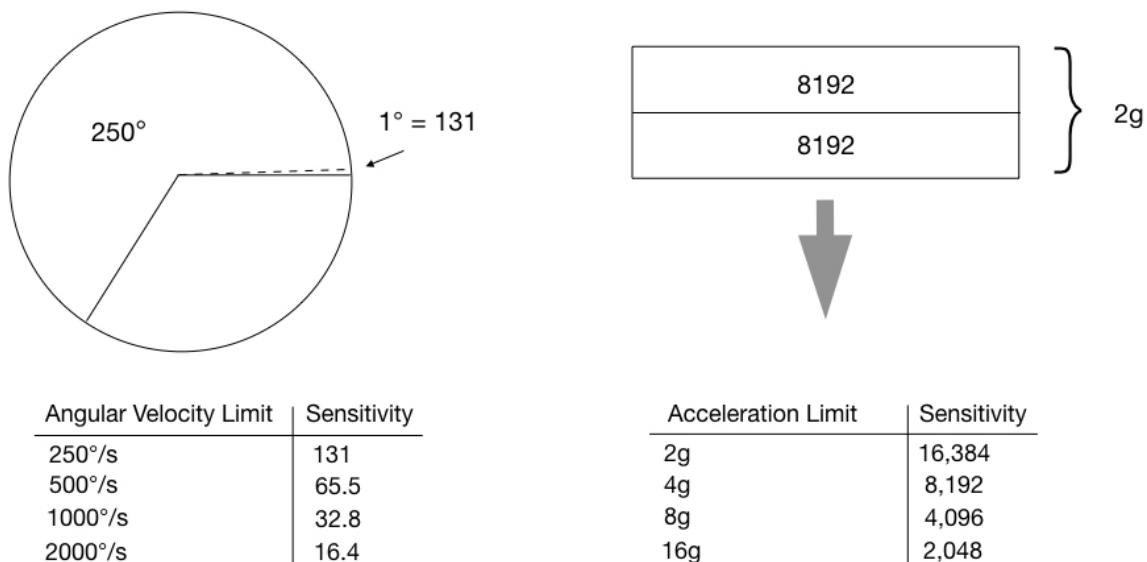


Рис. 4.11. Принцип преобразования чувствительности в MPU6050

экономно распределять единицы измерения. Однако для большинства задач роботов вряд ли будет вращаться так быстро, поэтому необходимый и достаточный уровень чувствительности в настоящей работе -  $250^\circ$ , который является настройкой по умолчанию, обеспечивая очень высокий уровень точности.

Еще одним преимуществом данного модуля является наличие DMP (Digital Motion Processor) — это встроенный процессор MPU6050, который объединяет данные, поступающие с акселерометра и гироскопа. С помощью данного процессора имеется возможность легко преобразовывать выходные данные сразу в углы крена и тангла. Такой способ получения информации с датчика намного точнее, так как в процессе снятия данных происходит внутренняя фильтрация, что убирает потребность сложной фильтрации шумов в реальном времени, а также позволяет минимизировать пост-обработку с помощью

комплементарного фильтра, который сильно уступает по точности DMP.



Рис. 4.12. MPU 6050

#### 4.3.4. Выбор источника питания

Источник питания необходимо выбирать опираясь на определенные критерии. Опишем их в порядке значимости:

- Высокая токоотдача для нормальной работы сервоприводов.
- Большая емкость для продолжительной работы.
- Малый вес, чтобы не увеличивать нагрузку на двигатели.
- Быстрая зарядка.

Таким свойствам удовлетворяет большинство Li-Po аккумуляторов, в настоящей работе была выбрана модель Spard Li-Po 3200mAh (рисунок 4.13). Этот тип аккумулятора получил широкое распространение благодаря своим высоким характеристикам производительности и компактности. Тип аккумуляторов Li-Po может хранить больше энергии, чем другие типы компактных



Рис. 4.13. Li-По аккумулятор емкостью 3200 мАч

источников питания, это означает, что он может работать дольше на одном заряде, что делает его идеальным для устройств, которые требуют высокой производительности и долгого времени работы. Однако, как и все, Li-По аккумуляторы имеют некоторые недостатки. В частности, он может быть более чувствителен к перезарядке, чем другие типы аккумуляторов. Также при возможном коротком замыкании, подключенные элементы имеют высокий шанс на полный выход из строя из-за высокого порога токоотдачи.

#### 4.3.5. Результаты проектирования и сборки

Финальный итог сборки робота по описанным выше твердотельным чертежам с перечисленными комплектующими представлен на рисунках 4.14, 4.15, 4.16. Полная схема подключения электронных компонентов (см. Приложение В).

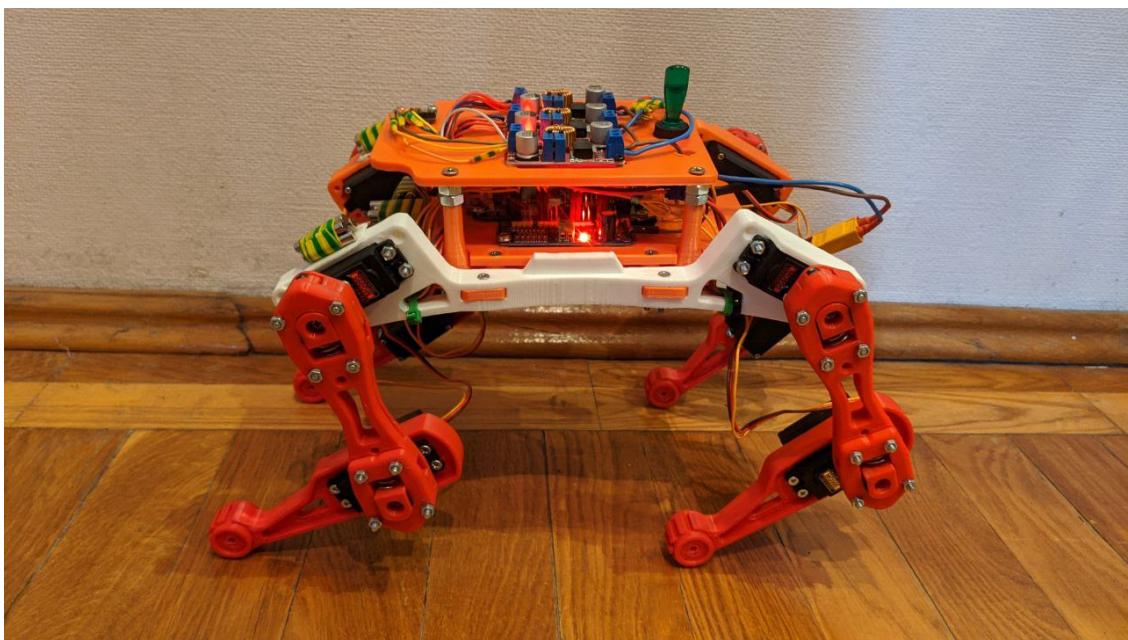


Рис. 4.14. Собранный прототип робота - вид сбоку

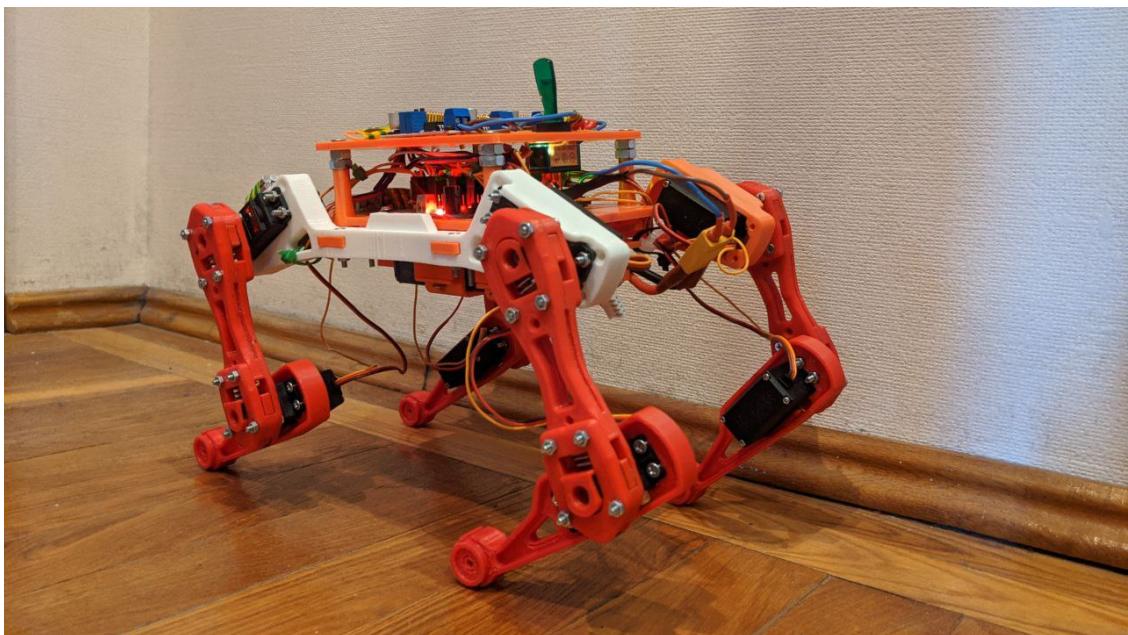


Рис. 4.15. Собранный прототип робота - вид сзади

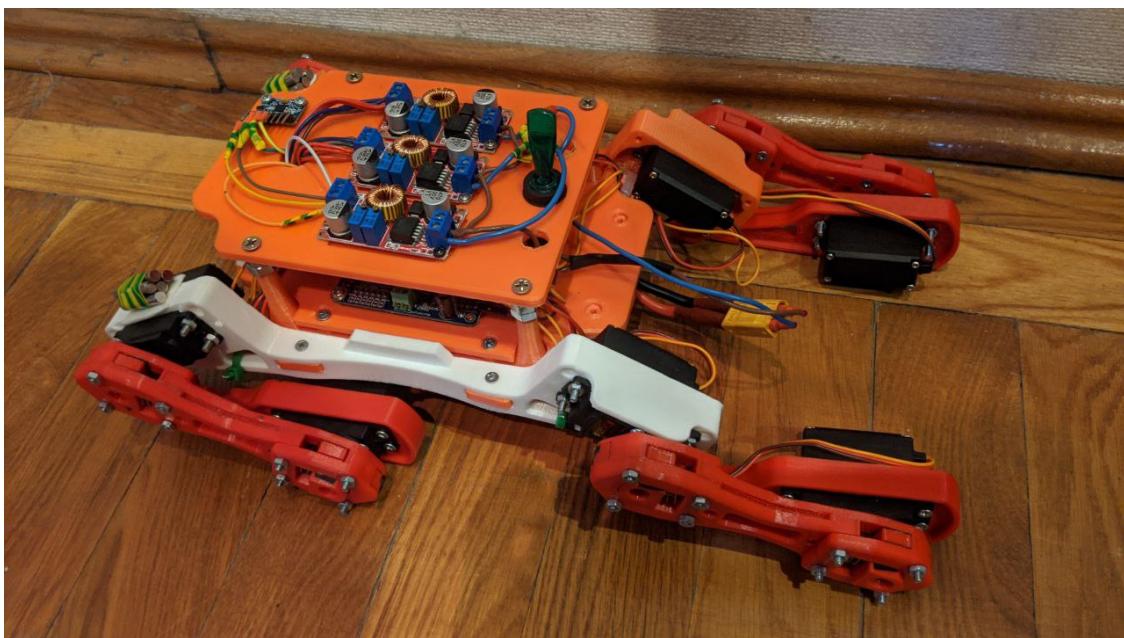


Рис. 4.16. Собранный прототип робота - вид сверху

# ГЛАВА 5

## Разработка ПО

### 5.1. Общая архитектура

Все взаимодействия комплектующих из пунктов 4.3.1 - 4.3.5 происходят путем отправки низкоуровневых команд из миникомпьютера Orange Pi в соответствующие модули. Команды формируются с помощью программного обеспечения, написанного на высокуюровневом языке Python 3. Данный язык был выбран в связи с наличием динамической типизации, богатой библиотекой для математических вычислений и большой существующей кодовой базой, что позволяет быстро писать прототипы для тестирования. На рисунке 5.1 показана структурная схема управления роботом.

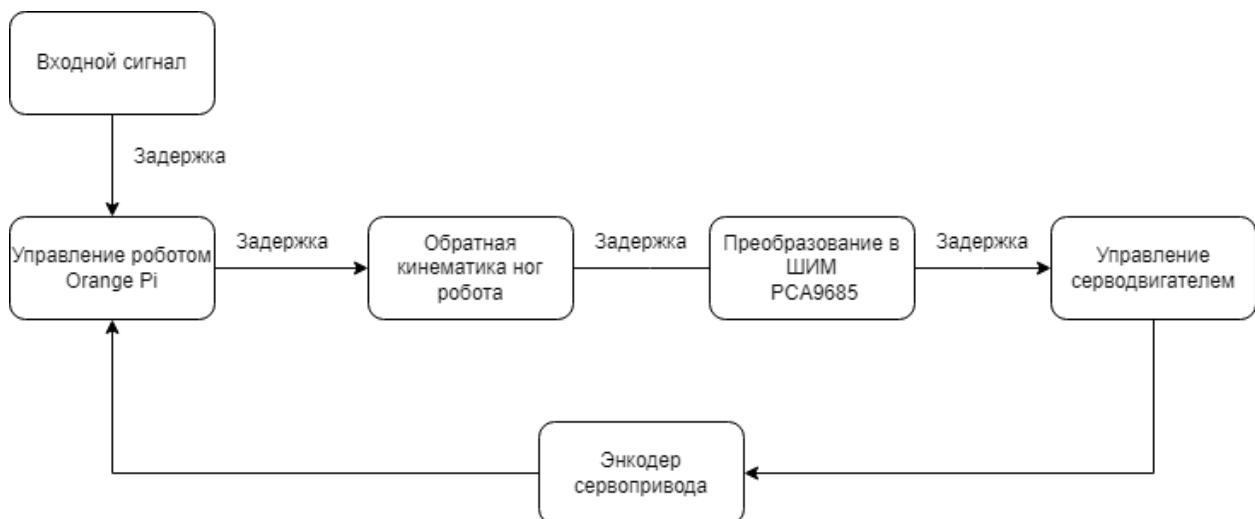


Рис. 5.1. Структурная схема управления роботом

Система управления роботом интерпретирует входные данные от устройства ввода и преобразует их в траекторию ходьбы для последующего позиционирования тела робота. Блоки обратной кинематической преобразуют координаты ноги в положения сервопривода.

## 5.2. Обратная связь с помощью гироскопа

Главным недостатком такой структуры управления — недоступность реальных значений углов поворота двигателя, как уже упоминалось в разделе 4.3.1, в следствие чего невозможно контролировать ошибку между идеальными и реальными значениями углов вала серводвигателя. Были предложены несколько решений данной проблемы:

1. Модификация каждой платы управления сервоприводом;
2. Подключение гироскопа акселерометра к каждому сервоприводу;
3. Установка гироскопа акселерометра на корпусе робота.

Первый вариант заключается в демонтаже каждого привода и прямого подключения в выходной сигнал управляющей платы.

Преимущества:

- Получение реальных углов поворота, которые принимают валы двигателей.
- Нет необходимости корректировать уравнения кинематики.
- Нет необходимости дополнительных вычислений, достаточно учитывать ошибку между реальным и идеальным значением на каждом шаге.

Недостатки:

- Восемь аналоговых подключений.
- OrangePi не поддерживает аналоговые порты.
- Требуются дополнительные преобразовательные модули.
- Необходимо фильтровать данные.

Вывод: Данное решение, не смотря на его простоту, не является подходящим, так как нет возможности его корректной реализации.

Второй метод представляет из себя снятие отклонения углов напрямую, преобразуя снятые данные с гироскопа акселерометра. Далее учитывать полученную ошибку на каждом шаге.

Преимущества:

- Получение реальных углов поворота, которые принимают голень и бедро робота.
- Нет необходимости корректировать уравнения кинематики.
- Нет необходимости лишних вычислений, достаточно учитывать ошибку между реальным и идеальным значением на каждом шаге.
- Получаемые данные цифровые. Это позволяет их легко обрабатывать.

Недостатки:

- Гироскоп акселерометр имеет свойство накапливать ошибку. В данном случае их 8 штук, как следствие, низкая точность.
- Восемь цифровых подключений.
- Сложность калибровки каждого датчика отдельно.
- Требуется дополнительный модуль “сплиттер” для подключения более двух гироскопов акселерометров.
- Необходимо фильтровать данные.

Вывод: Данное решение не является подходящим, так как необходимы дополнительные модули для работы с гироскопами акселерометрами в количестве более двух штук. Также существенной сложностью является калибровка большого количества датчиков и поддержание необходимой точности вычислений.

Третий метод заключается в креплении гироскопа акселерометра на корпусе робота и последующего регулирования его отклонений в плоскостях тангажа и крена, тем самым соблюдая условие комфортабельности походки, то есть реализация поступательного, равномерного и прямолинейного движения корпуса.

Преимущества:

- Подключение всего одного датчика.
- Малые затраты на реализацию.
- Компенсация отклонений корпуса в плоскостях тангажа и рысканья.
- Получаемые данные цифровые. Это позволяет их легко обрабатывать.

Недостатки:

- Необходимо корректировать уравнения кинематики.
- Необходимо фильтровать данные
- Получаемые данные цифровые. Это позволяет их легко обрабатывать.

В настоящей работе выбран третий вариант, в таком случае структурная схема управления примет вид (см.рисунок 5.2).

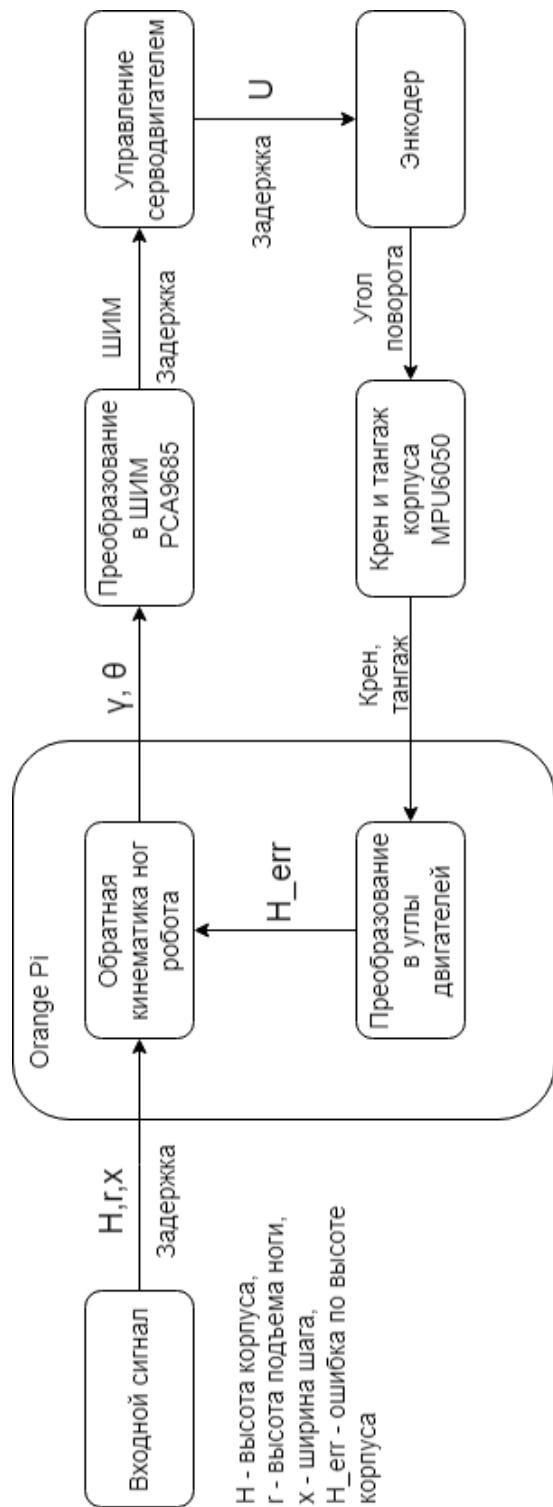


Рис. 5.2. Структурная схема управления роботом с обратной связью

## 5.3. Движение робота вперед с обратной связью

Преобразование углов крена и тангажа в углы серводвигателей происходит по алгоритму, описанному ниже:

Для опорной фазы:

```
def calculate_step(x_range_step):  
  
    roll_ang, elev_ang = get_pitch_roll() # get data from MPU6050  
    elev = np.sin(elev_ang)*body_l//2 # recalculate pitch angle to height  
    roll = np.sin(roll_ang)*body_w//2 # recalculate roll angle to height  
    y_step=0  
    height = drop + elev # include height error (pitch component)  
    hip_offset = np.arcsin(elev / body_l)  
    height_slew = np.sqrt(x_range_step**2+(height-y_step)**2)+roll #  
                           include height error (roll component)  
    hip_step = 0.5*np.pi - np.arccos(height_slew/(2*self.l1))+np.arctan(  
                           x_range_step/(height-y_step)) +  
                           hip_offset # conversion to hip servo  
                           angle  
    knee_step = 2*np.arcsin(height_slew/(2*self.l1)) # conversion to knee  
                           servo angle  
  
    return np.rad2deg(hip_step), np.rad2deg(knee_step)
```

Для свободной фазы аналогично:

```
def calculate_skip(x_range_skip,min_delta):  
  
    roll_ang,elev_ang = get_pitch_roll() # get data from MPU6050  
    elev = np.sin(elev_ang)*body_l//2 # recalculate pitch angle to height  
    roll = np.sin(roll_ang)*body_w//2 # recalculate roll angle to height  
  
    delta = np.sqrt(self.r**2-min_delta**2)  
    y_skip = np.sqrt((self.r**2) - (x_range_skip**2)) - delta  
    height = drop + elev # include height error (pitch component)  
    hip_offset = np.arcsin(elev / 105)  
    height_slew = np.sqrt(x_range_skip**2+(height-y_skip)**2)+roll #  
                           include height error (roll component)  
    hip_skip = 0.5*np.pi - np.arccos(height_slew/(2*self.l1))+np.arctan(  
                           x_range_skip/(height-y_skip)) +  
                           hip_offset # conversion to hip servo  
                           angle
```

```

knee_skip = 2*np.arcsin(height_slew/(2*self.l1)) # conversion to knee
servo angle

return np.rad2deg(hip_skip),np.rad2deg(knee_skip)

```

Используя данные алгоритмы, получим метрики при движении робота для значений  $X_{ш}$  от  $-50$  мм до  $50$  мм с шагом  $1$  мм, высотой  $h_{корп} = 150$  мм и  $r = 125$  мм. Сравним значения идеальных и реальных углов, полученных с помощью обратной связи.

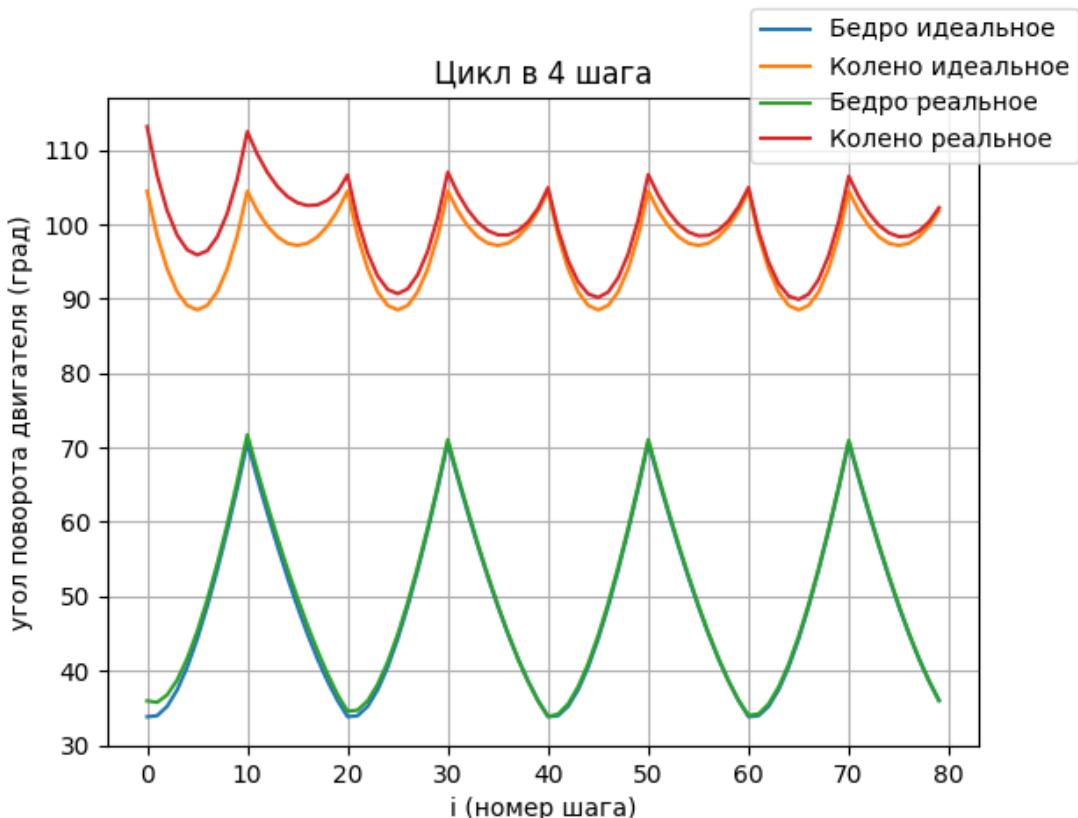


Рис. 5.3. Изменение углов двигателей за четыре шага с обратной связью

Исходя из полученного графика можно сделать вывод, что в течение четырех шагов, значения углов на серводвигателях стабилизируются и стремятся к идеальным, что означает работоспособность выбранного метода обратной связи.

## ЗАКЛЮЧЕНИЕ

В рамках работы разработан прототип шагающего робота для образовательных и исследовательских целей. Были составлены чертежи и 3D модели всех составляющих робота, подобраны электрические и логические компоненты, рассчитан худший случай статической нагрузки, исходя из которого была выбрана модель сервоприводов. Исследовано движение стопы шагающего робота, описан процесс решения обратной задачи кинематики для свободного и опорного случая стопы в проблеме о положениях. Произведен синтез походки робота, решена проблема обратной связи реальных углов двигателей с помощью блока гироскопов-акселерометров. Разработано программное обеспечение на языке *Python* 3 для управления роботом в любой доступной конфигурации, которая ограничена заранее найденной рабочей областью. В кодовой базе описаны протоколы снятия данных и последующего их преобразования, в качестве результата приведены графики реальных и идеальных значений углов поворота двигателей.

Все поставленные цели работы достигнуты, а результат проекта представлен в качестве рабочего прототипа с дистанционным управлением.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

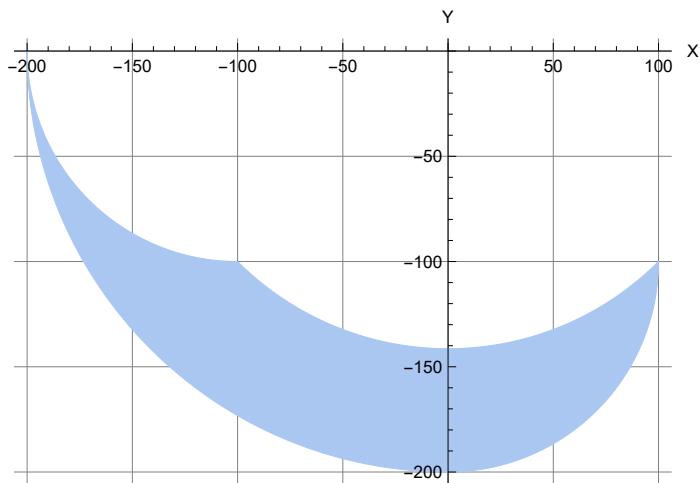
1. J. Estremera P. G. Santos, J. A. G' alvez and E. Garcia. A true walking robot for the comparative study of walking machine techniques. *Robotics and Automation Magazine*, 10(4):23–32, 2007.
2. Matt Simon. How a real dog taught a robot dog to walk. *Wired*, 2020.
3. G. Nelson R. Playter M. Raibert, Kevin Blankespoor. Bigdog, the rough-terrain quadruped robot. *IFAC Proceedings Volumes*, 41:10822–10825, 2008.
4. Unitree. Laikago: a four leg robot is coming to you. *IEEE Spectrum*, 2020.
5. Презентация платформы для разработки робособак unitree robotics прошла в МГТУ. <https://bmstu.ru/news/prezentacia-platformy-dla-razrabotki-robosobak-unitree-robotics-prosl>  
Дата посещения: 20.02.2023.
6. Aker BP. Exploring the potential of robotics in the oil and gas industry. *Aker BP News*, 2020.
7. Aker BP. Approved development plan for the hod field. *Aker BP News*, 2020.
8. W. L. Cleghorn G. Piras and J. K. Mills. Dynamic finite-element analysis of a planar high-speed, high-precision parallel manipulator with flexible links. 40:1061, 2005.
9. Z. Huang Z. Li and Y. Huang. Design of spot welding robot. *Indonesian Journal of Electrical Engineering*, 11(11):6267 – 6273, 2014.
10. M. Ruchanurucks S. Kudoh, K. Ogawara and K. Ikeuchi. Painting robot with multi-fingered hands and stereo vision. *Robotics and Autonomous Systems*, 57(3):279 – 288, 2009.
11. J. J. Steil F. Rothling G. A. Fink S. Wachsmuth G. Sagerer P. McGuire, J. Fritsch and H. Ritter. Multi-modal human-machine communication for instructing robot grasping tasks. *International Conference on Intelligent Robots and Systems*, 2:1082 – 1088, 2012.
12. A. Bechar and C. Vigneault. Agricultural robots for field operations. part 2:operations and systems. *Biosystems Engineering*, 153:110 – 128, 2017.
13. K. Shimomura H. Kondo A. Morishima H. O. Lim Y. Ogura, H. Aikawa and A. Takanishi. Development of a new humanoid robot wabian-2, 2016.

14. Y. Sugahara H. Lim and A. Takanishi. Development of a biped locomotor applicable to medical and welfare fields, 2013.
15. Y. J. Lee and S. Hirose. Three-legged walking for fault tolerant locomotion of a quadruped robot with demining mission, 2010.
16. J. D. Carlo B. Katz and S. Kim. Mini cheetah: A platform for pushing the limits of dynamic quadruped control. pages 6295 – 6301, 2019.
17. N. Ferrante N. Kau, A. Schultz and P. Slade. Stanford doggo: An open-source, quasi-directdrive quadruped. *International Conference on Robotics and Automation*, page 6309 – 6315, 2019.
18. E. Guizzo. Boston dynamics’ spot robot dog goes on sale. *IEEE Spectrum*, 2019.
19. Накано Э. *Введение в робототехнику*. Мир, 1988.
20. Zhuochen Liu Guillaume Bellegarda, Yiyu Chen and Quan Nguyen. Robust high-speed running for quadruped robots via deep reinforcement learning. pages 1–7, 2023.

## ПРИЛОЖЕНИЕ А

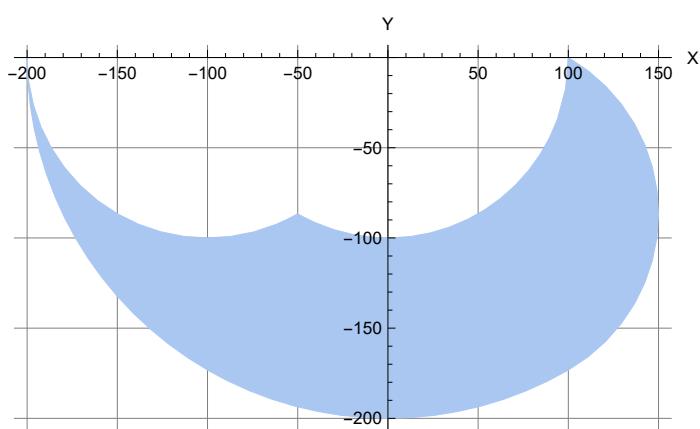
```
In[1]:= 11 = 100;
12 = 100;
Xb = 11 * Cos[\theta1] + 12 * Cos[\theta1 + \theta2];
 $\cos[\theta_1]$   $\cos[\theta_1 + \theta_2]$ 
Yb = 11 * Sin[\theta1] + 12 * Sin[\theta1 + \theta2];
 $\sin[\theta_1]$   $\sin[\theta_1 + \theta_2]$ 
area =
ParametricRegion[{Xb, Yb}, {{\theta1, 180 Degree, 270 Degree}, {\theta2, 0 Degree, 90 Degree}}];
 $\text{параметрический заданный геометрический ре...}$   $\text{градус}$   $\text{градус}$   $\text{градус}$   $\text{градус}$ 
Show[Region[area], GridLines -> Automatic,
 $\text{пок...}$   $\text{область}$   $\text{линии координат}$   $\text{автоматический}$ 
Axes -> True, AxesOrigin -> 0, AxesLabel -> {"X", "Y"}]
 $\text{оси}$   $\text{ист...}$   $\text{точка пересечения}$   $\text{обозначения на осях}$ 
```

Out[1]=

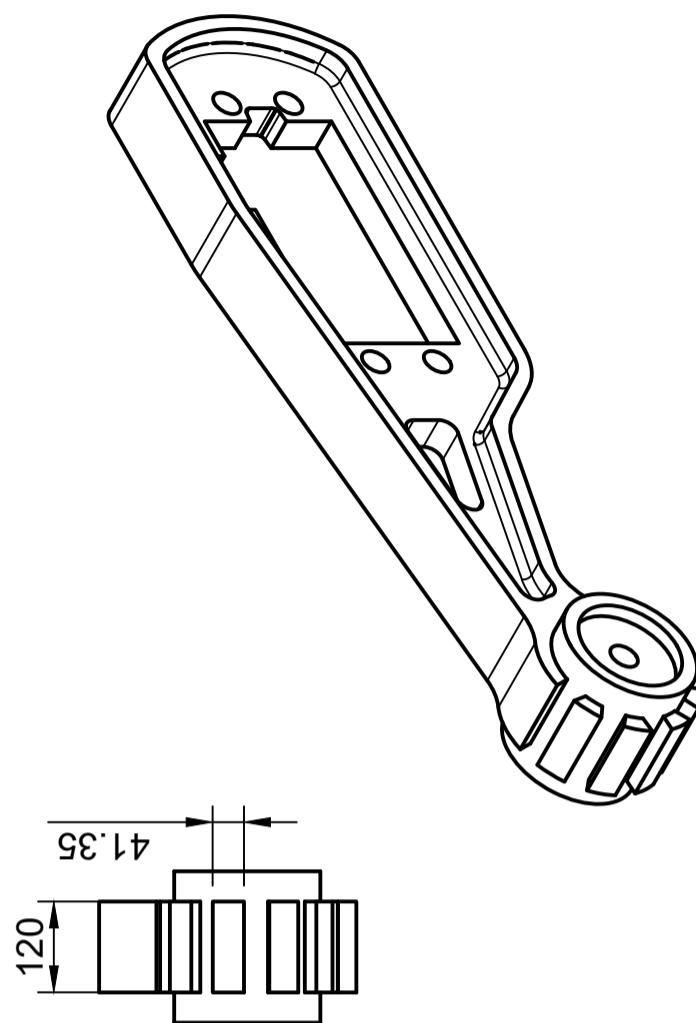
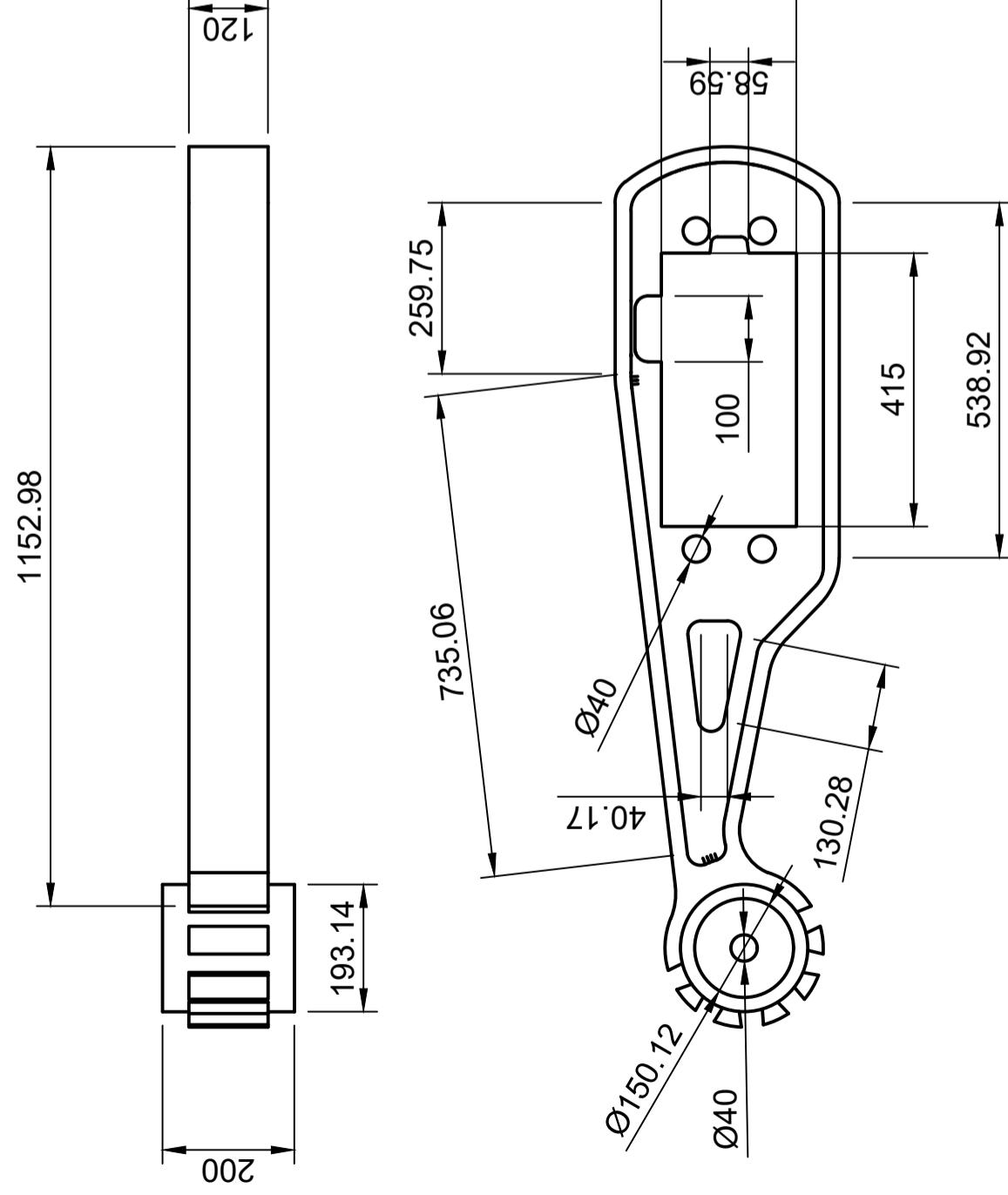


```
In[2]:= area = ParametricRegion[{Xb, Yb},
 $\text{параметрический заданный геометрический регион}$ 
{{\theta1, 180 Degree, (270 + 30) Degree}, {\theta2, 0 Degree, (90 + 30) Degree}}];
 $\text{градус}$   $\text{градус}$   $\text{градус}$   $\text{градус}$ 
Show[Region[area], GridLines -> Automatic,
 $\text{пок...}$   $\text{область}$   $\text{линии координат}$   $\text{автоматический}$ 
Axes -> True, AxesOrigin -> 0, AxesLabel -> {"X", "Y"}]
 $\text{оси}$   $\text{ист...}$   $\text{точка пересечения}$   $\text{обозначения на осях}$ 
```

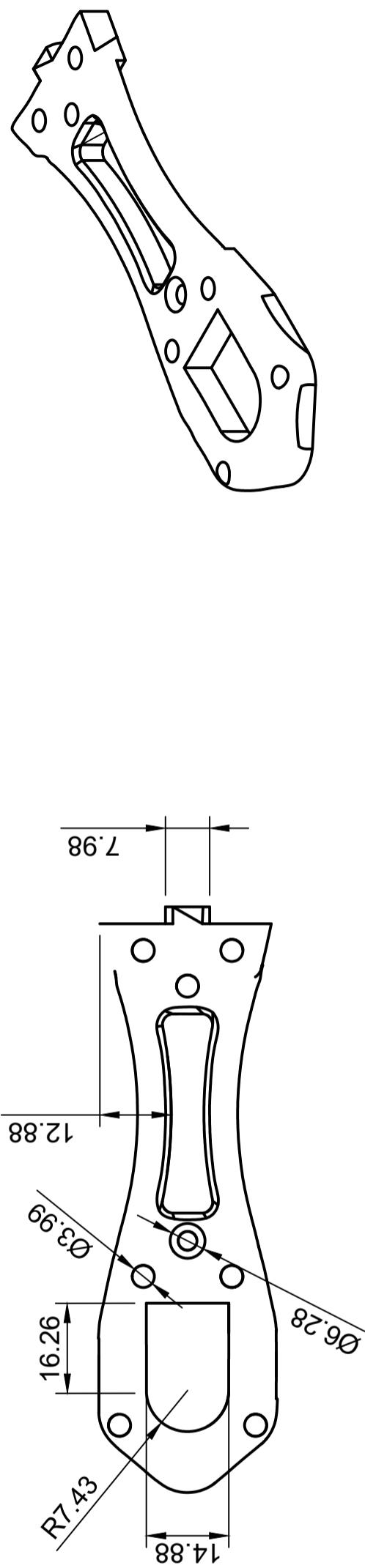
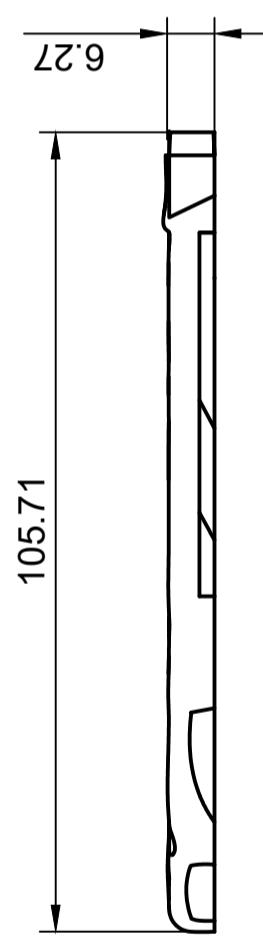
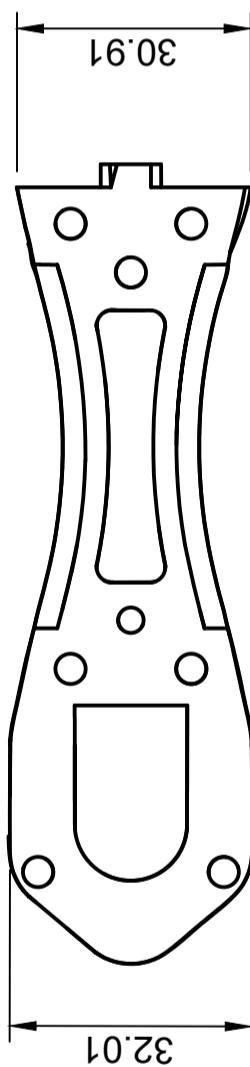
Out[2]=



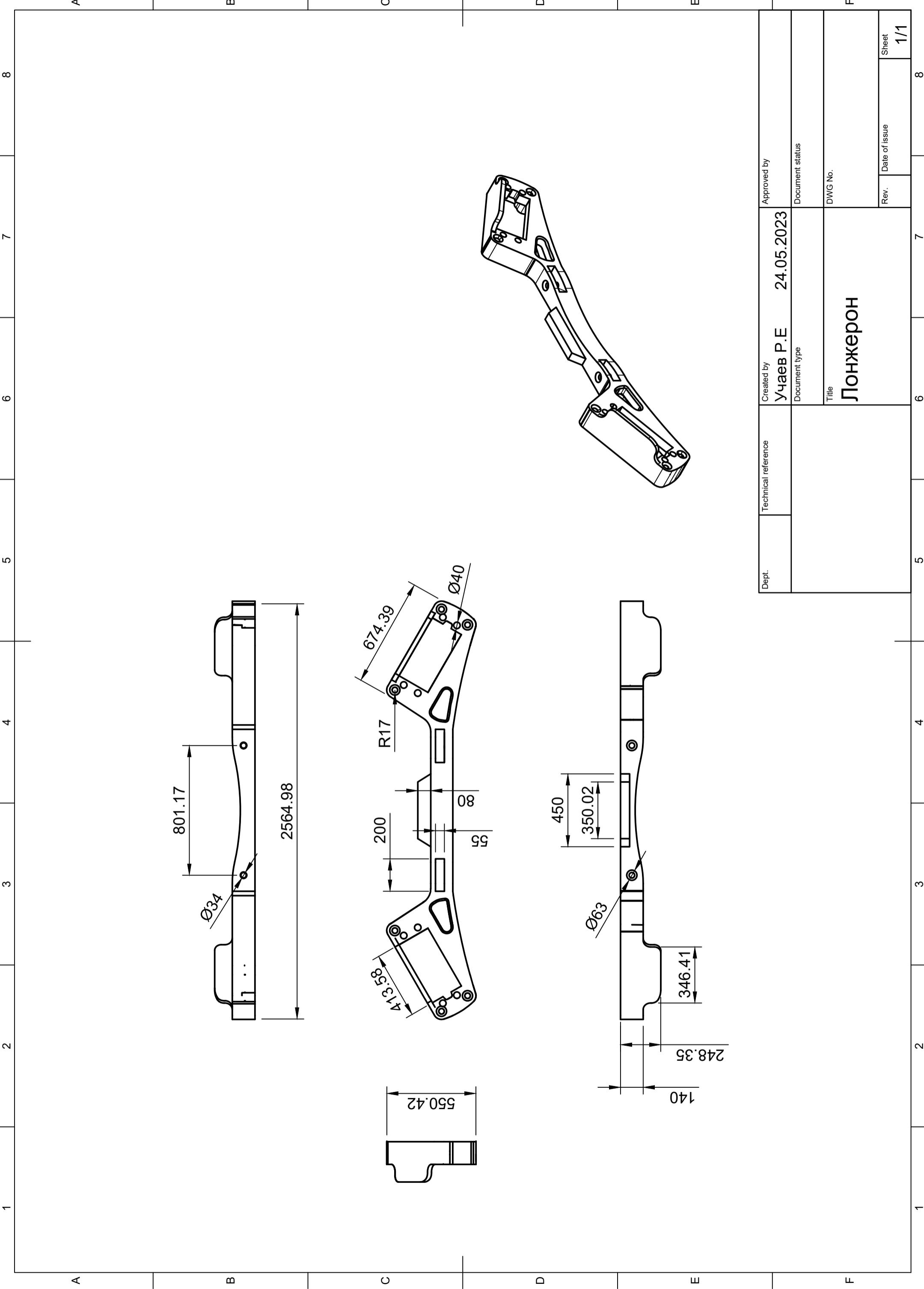
# Приложение Б



Dept.	Technical reference	Created by Учаев Р.Е	Approved by 24.05.2023
Document status			DWG No.
Title <b>Колено</b>			Rev. Date of issue
			Sheet <b>1/1</b>



Dept.	Technical reference	Created by Учаев Р.Е	Approved by 24.05.2023
Document status			DWG No.
Title <b>Бедро</b>			Rev. Date of issue
			Sheet 1/1



## ПРИЛОЖЕНИЕ В

