

Санкт–Петербургский государственный университет

ТАШКИНОВ Владимир Андреевич

Выпускная квалификационная работа

Задача динамической ходьбы андроида

Уровень образования: магистратура

Направление 01.04.02 «Прикладная математика и информатика»

Основная образовательная программа ВМ.5517.2020 «Методы прикладной математики и информатики в задачах управления»

Научный руководитель:

доцент, кафедра механики управляемого движения,
к.ф. - м.н., Шиманчук Дмитрий Викторович

Рецензент:

доцент, кафедра общенаучных дисциплин,
«Военная академия материально-технического
обеспечения имени генерала армии А.В.Хрулёва»,
к.т.н., Волков Юрий Александрович

Санкт-Петербург

2022 г.

Содержание

Актуальность	3
Обзор литературы	5
Описание модели	7
Постановка задачи	8
Глава 1. Описание датчиков двуногих шагающих роботов . .	9
Глава 2. Описание статического и динамического равновесия двуногих шагающих роботов	12
2.1. Статическое равновесие	12
2.2. Динамическое равновесие	13
2.2.1 Метод точки нулевого момента (ZMP)	15
Глава 3. Расчёт положения точки нулевого момента	20
Заключение	25
Список литературы	26
Приложение 1	29
Приложение 2	31

Актуальность

Человечество никогда не стоит на месте, научно-технический прогресс движется просто огромными шагами. Уже давно в фильмах и книгах пользуется популярностью тема научной фантастики. Люди задумываются о том как будет выглядеть их будущее и чего сможет достичь человечество в области технологий. Одним из довольно частых представлений является робототехника, в частности, антропоморфные роботы. Причину такой направленности гадать особо не нужно, люди стремятся облегчить себе жизнь, следовательно нужно сделать так, чтобы робот выполнял задачи совместно или вместо человека, а для этого удобно, чтобы робот обладал такими же возможностями как и человек или даже лучше. Антропоморфные роботы уже давно перестали быть фантастикой, они создаются для различных целей от простого домашнего компаньона до курьера или роботизированного пожарного. В скором будущем их использование может стать совершенно обычным делом, множество компаний работают над созданием антропоморфного робота, и добились уже немалых успехов. Одной из основных проблем этой задачи представляется как раз имитация подвижности человека, движение рук и ног.

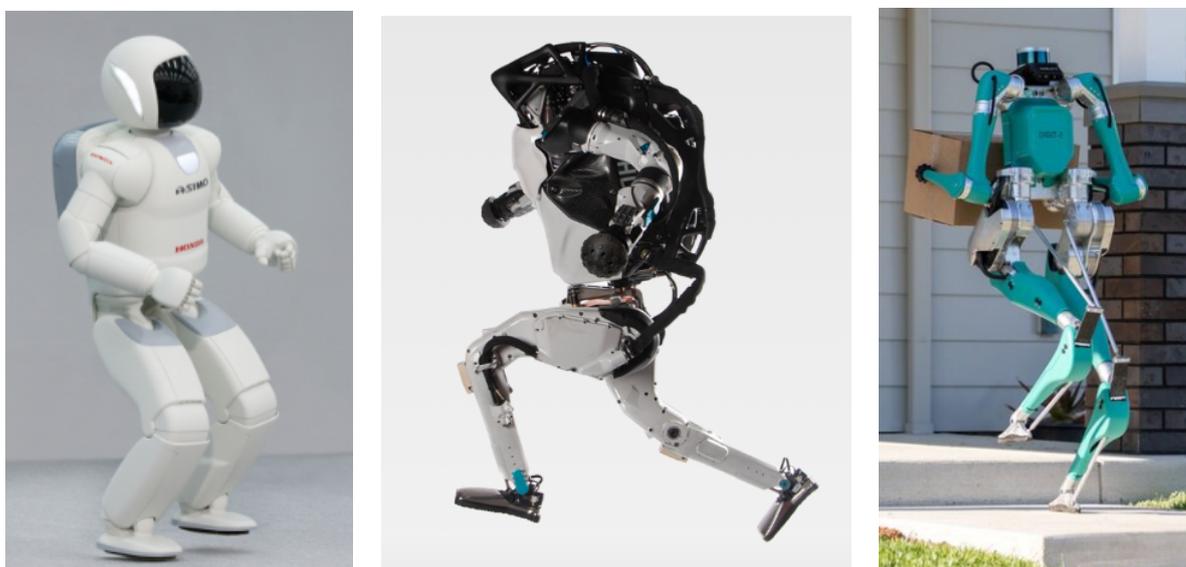


Рис. 1: Роботы ASIMO от Honda, ATLAS от Boston Dynamics и DIGIT от Agility Robotics.

На рис. 1 показаны роботы ASIMO от Honda (Япония), ATLAS от

Boston Dynamics (США), показывающий лучшие результаты в подвижности, и DIGIT от Agility Robotics (США), прототип робота-курьера.

В работе будет рассматриваться движение ног при динамической ходьбе двуногого робота с обеспечением условия балансировки. Считаю, что это важнейшая часть имитации реальных движений человека в задаче динамической ходьбы двуногого шагающего робота.

Обзор литературы

Данная работа является продолжением моей ВКР, указанной в [1], также была использована следующая литература:

В работах [2] и [8] рассматривается метод Денавита-Хартенберга, используемый для решения обратной задачи кинематики, также из работы [8] взята идея использовать систему разгон-торможение.

В работе [3] рассмотрены плоские модели ходьбы двуногого робота.

В работах [4], [7] решается ряд проблем по кинематике и динамике антропоморфных систем, их моделированию, управлению походкой и поддержанию равновесия при ходьбе.

В работе [5] проводится исследование по динамике и управлению двуногими и четырёхногими шагающими аппаратами.

В статье [11] рассмотрен порядок выбора конструкции суставов пространственной манипуляционной системы в зависимости от типа двуногого робота и типа ходьбы, который позволяет составить алгоритм управления движением рук.

В статье [12] создан прототип робота, который используется для тестирования управления движениями робота, такими как движение вперед, назад, поворот влево и вправо, подъём со спины и назад, перекатывание влево и вправо.

В работе [13] представлены исследования и разработки в области мобильной робототехники и связанных с ними технологий поддержки, большое внимание уделяется роботам вне заводских условий.

В работах [14] и [15] описывается метод точки нулевого момента, который используется для динамического равновесия двуногих роботов.

В статьях [16] и [17] представлены модели обратных маятников, которые также могут использоваться для поддержания динамического равновесия.

В книгах [18] и [19] представлены исследования в области управления двуногими роботами, и устройство некоторых из уже существующих роботов.

В работах [20] - [24] рассматриваются нейросети и их использование

в задаче динамической двуногой ходьбы.

В источниках [25] и [26] раскрыты особенности моделирования робототехнических систем в программе CoppeliaSim(V-REP).

Книги [27] и [28] описывают примеры реализации искусственной жизни в программном и, соответственно аппаратном обеспечении.

Описание модели

Дан пятизвенный двуногий робот. Каждая нога которого имеет два звена, длины верхнего и нижнего равны l_1 и l_2 соответственно, каждая нога имеет по два поворотных сочленения. Туловище робота также имеет свой поворотный шарнир. Углы q_5 между туловищем и горизонтальной

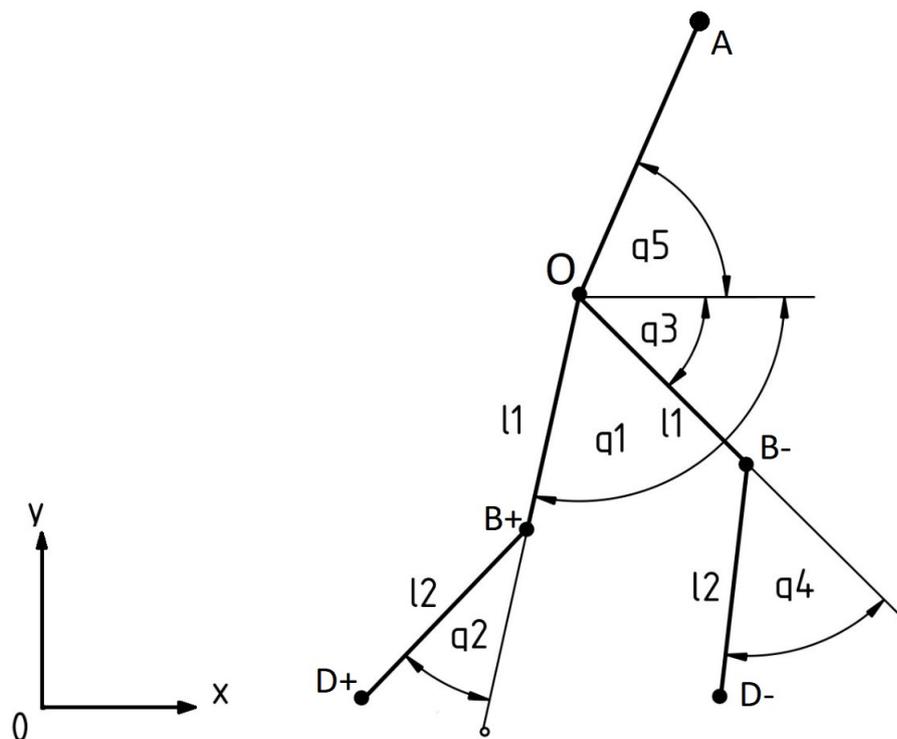


Рис. 2: Кинетическая модель робота.

прямой проходящей через точку O , q_1, q_3 , между той же прямой и звеньями длины l_1 , q_2, q_4 между продолжениями звеньев длины l_1 и звеньями длины l_2 (см. рис. 2). Обобщёнными координатами модели являются q_1, q_2, q_3, q_4, q_5 , массы сконцентрированы в точках A, O, B_+, B_-, D_+, D_- и соединены невесомыми стержнями.

Постановка задачи

В ходе работы необходимо:

- рассмотреть виды датчиков двуногих шагающих роботов, используемых для определения его положения,
- определить ситуации статического и динамического равновесия для заданной модели робота,
- при помощи методов динамической ходьбы создать алгоритм расчёта управляющего сигнала при заданных параметрах,
- разработать скрипт для передачи управляющего сигнала приводам модели робота в программе для симуляции,
- рассчитать положения точки нулевого момента с помощью алгоритмов минимизации для сокращения длины стопы робота, при сохранении динамического равновесия.

Целью работы является применение методов динамической ходьбы для большего соответствия процесса передвижения робота к передвижению реального человека, а также его программная реализация.

Глава 1. Описание датчиков двуногих шагающих роботов

В двуногих роботах могут использоваться следующие датчики:

- **Тензорезистивный датчик** (датчик деформации) обычно представляет собой специальную упругую конструкцию с закреплённым на ней тензорезистором и другими вспомогательными деталями. После калибровки, по изменению сопротивления тензорезистора можно вычислить степень деформации, которая будет пропорциональна силе, приложенной к конструкции. Если использовать конструкцию ступней как у робота Andy [19], с тремя пальцами, то установив тензодатчик на каждый палец можно вычислить крутящие моменты в ступне, включая точку нулевого момента (ZMP)(См. 2.2.1).
- **Оптические датчики.** Оптические датчики работают с помощью фоторезистора. Они состоят из излучателя и приёмника, по отражённому лучу можно различить чёрные и белые части поверхности, это используется для движения робота по нарисованной на поверхности линии или определять близость установленного препятствия. Но движение с такими сенсорами эффективно только в заранее оборудованных местах, в иных же случаях, при помещении робота в неподготовленное окружение, их использование будет иметь мало пользы. Камера в роботе может служить также для поддержания равновесия, выделив контрастным цветом искусственный горизонт, в зависимости от его положения на изображении камеры, робот может распознавать находится он в равновесии или в какую сторону он наклоняется, из чего уже совершать необходимые движения для поддержки равновесия.
- **Звуковые датчики.** Звуковые датчики предназначены для определения расстояния до препятствий по принципу эхолокации – динамик прибора издает УЗ импульс на определенной частоте и замеряет время до момента его возвращения на микрофон. Звуковые локаторы из-

лучают направленные звуковые волны, которые отражаются от объектов, и часть этого звука снова поступает в датчик. При этом время поступления и интенсивность такого возвратного сигнала несут информацию о расстоянии до ближайших объектов. В основном используются для предотвращения столкновения на ближней дистанции.

- **Датчики с использованием лазерного луча.** К ним относятся радары, лазеры и лидары. Вместо звука, в этом типе дальномеров используется отраженный от препятствия лазерный луч. Эти датчики получили более широкое применение в разработке автономных автомобилей, так как позволяют транспортному средству более эффективно справляться с дорожным движением.
- **Датчики положения и наклона.** Этот вид датчиков используется в основном в беспилотных транспортных средствах, промышленных роботах, а также устройствах, требующих самобалансировки. К датчикам положения относятся GPS (система глобального позиционирования), ориентиры (исполняют роль маяка), гироскопы (определение угла вращения) и акселерометры. GPS – это спутниковая система навигации, обеспечивающая измерение расстояния, времени и определяющая местоположение робота в пространстве. GPS позволяет беспилотным наземным, воздушным и водным транспортным средствам находить свой маршрут и без труда двигаться от одной точки к другой. Гироскопы в робототехнике также распространенная вещь. Они отвечают за балансировку и стабилизацию любого устройства. А за счет того, что эта деталь относительно недорогая, её можно установить в любой самодельный робот. Акселерометр - это датчик, позволяющий роботу измерять ускорение тела под действием внешних сил. Этот прибор похож на массивное тело, способное передвигаться вдоль некоторой оси и соединено с корпусом прибора пружинами. Если такой прибор толкнуть вправо, то груз сместится по направляющей влево от центра оси. Акселерометры с двумя измерительными осями (датчики ускорения) позволяют вычислить динамические силы, действующие на робота. Эта информация используется для управления

балансом робота. Пьезогироскопы являются альтернативой датчикам ускорения, так как последние подвержены высокочастотному шуму от тряски сервоприводов.

- **Инфракрасные датчики.** Самый доступный и простой вид датчиков, которые применяются в роботах для определения приближения. Инфракрасный датчик самостоятельно посылает инфракрасные волны и, поймав отраженный сигнал, определяет наличие препятствия перед собой. ИК-датчики близости могут использоваться в ступнях, для определения на какой части ступни стоит робот в данный момент, а также ИК-датчики расстояния могут применяться для обнаружения препятствий.

Помимо датчиков самого робота в исследовании могут также быть использованы датчики для сбора данных о движении человека в реальных условиях, или для обратной связи робота. Для этой задачи подходит сторонний сенсорный контроллер позволяющий записывать движения человека (или робота) с некоторого расстояния или несколько IMU - сенсоров. Сбор таких данных может использоваться для обучения нейросетей, управляющих походкой робота.

Также имеются методы для вычисления положения человека используя изображения с двух ракурсов, но данный способ сильно ограничивает область сбора данных.

Глава 2. Описание статического и динамического равновесия двуногих шагающих роботов

Существует два типа передвижения двуногих роботов:

- С поддержкой статического равновесия – когда проекция центра масс робота на поверхность всегда находится в области покрываемой его ступнями.
- С поддержкой динамического равновесия – понятие динамического равновесия трактуется по разному в зависимости от источников, в данном же случае будем предполагать что равновесие называется динамическим, когда проекция центра масс может выходить за пределы области покрываемой ступнями.

2.1 Статическое равновесие

Для модели робота на рис.3 определим положение центра масс:

$$\vec{r}_C = \frac{1}{M}(m_O\vec{r}_O + m_A\vec{r}_A + m_B(\vec{r}_{B+} + \vec{r}_{B-}) + m_D(\vec{r}_{D+} + \vec{r}_{D-})), \quad (1)$$

где m_O, m_A, m_B, m_D – массы соответствующих точек, M – сумма масс точек, $\vec{r}_C, \vec{r}_O, \vec{r}_A, \vec{r}_{B+}, \vec{r}_{B-}, \vec{r}_{D+}, \vec{r}_{D-}$ – вектор-функции определяющие положение соответствующей точки. При заданном движении ног регулировать положение центра масс можно при помощи управления углом наклона корпуса q_5 . Для статического равновесия не важно на какой высоте будет находиться центр масс, поэтому можно рассмотреть только составляющую по x :

$$x_C = \frac{1}{M}(m_Ox_O + m_Ax_A + m_B(x_{B+} + x_{B-}) + m_D(x_{D+} + x_{D-})).$$

Определим условия, при которых статическое равновесие будет достигнуто. Обозначим длину стопы переменной s . Для случая когда обе ноги на земле:

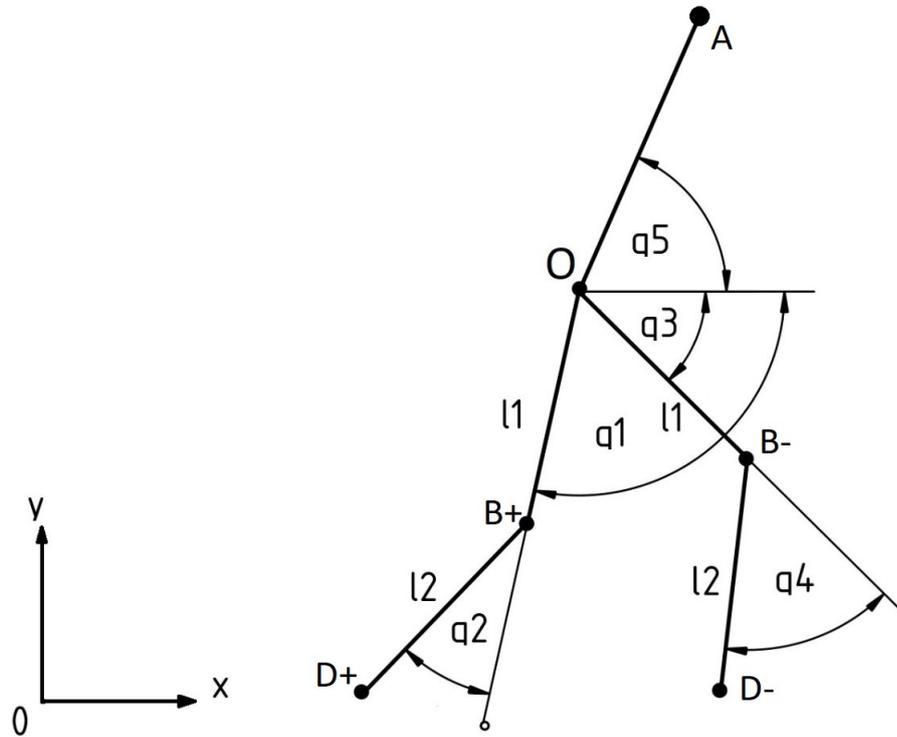


Рис. 3: Кинетическая модель робота.

$$x_{D+} \leq x_C \leq x_{D-} + s,$$

для $x_{D+} \leq x_{D-}$, в обратном случае аналогично. Для случая когда робот стоит на одной ноге:

$$x_{D+} \leq x_C \leq x_{D+} + s,$$

для $y_{D+} = 0$, для второй ноги аналогично. Если условия выше выполнены, то робот будет находиться в состоянии статического равновесия.

2.2 Динамическое равновесие

Движение с использованием статического равновесия малоэффективно. Этот метод требует использовать довольно большие ступни для робота, а также сильно ограничивает скорость его передвижения. Использование динамического равновесия позволяет уменьшить длину ступней или же совсем от них избавиться, оставив лишь одну точку контакта с поверхностью, также динамическое равновесие может поддерживаться даже при беге ро-

бота.

Существует несколько методов динамической ходьбы:

- Метод точки нулевого момента (ZMP) – один из стандартных методов динамического равновесия, для его использования необходимо знать все силы действующие на робота, эти данные можно получить при помощи датчиков или высчитать из модели робота. Суть метода состоит в нахождении точки нулевого момента, которая является динамическим эквивалентом центра масс.
- Перевернутый маятник – задачу равновесия двуногого робота можно рассматривать как задачу стабилизации обратного маятника. Динамическое равновесие в таком случае может быть достигнуто постоянным наблюдением за наклоном робота и адаптацией соответствующих движений ног.
- Нейронные сети – используются для изменения походки в реальном времени. Такой подход делает возможным обучение новым походкам, которые не являются взвешенными комбинациями определённых заранее.
- Генетические алгоритмы – группа виртуальных роботов генерируется с изначально случайными настройками управления. Наиболее эффективные роботы воспроизводятся с использованием генетических алгоритмов следующего поколения.
- ПИД-регулятор – используется для управления наклоном робота вперед/назад и влево/вправо, аналогично статическому равновесию. Однако здесь на этапе обучения мы задаём желаемый наклон тела робота вперёд и в стороны на всех этапах его ходьбы.
- Искусственный горизонт – выделив контрастным цветом искусственный горизонт в окружении робота, робот может распознавать, в зависимости от положения горизонта на изображении камеры, находится он в равновесии или в какую сторону он наклоняется, из чего уже совершать необходимые движения для поддержки равновесия.

2.2.1 Метод точки нулевого момента (ZMP)

Использование метода точки нулевого момента позволяет центру масс робота заходить за пределы площади покрываемой его ступнями. В основе метода стоит нахождение точки нулевого момента (ZMP) – динамического эквивалента центра масс, и если точка лежит внутри площади, покрываемой стопами, то робот находится в состоянии динамического равновесия.

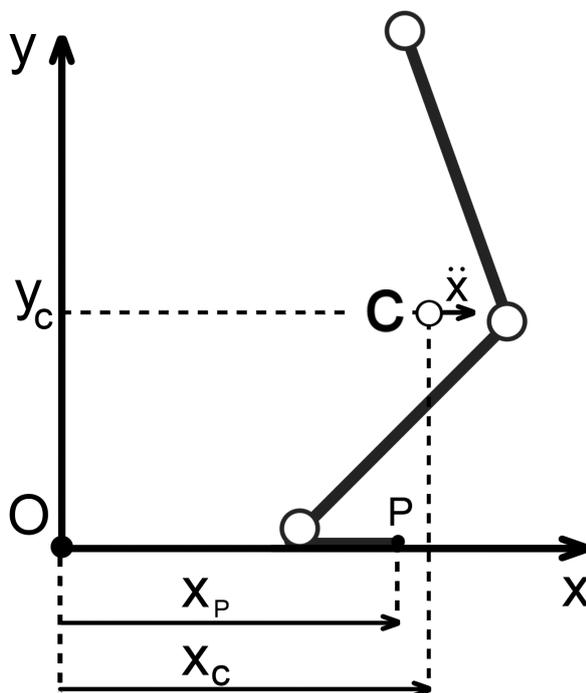


Рис. 4: Точка нулевого момента.

Определим крутящий момент в некоторой точке стопы P :

$$\tau = -Mg(x_C - x_P) + M\ddot{x}_C y_C,$$

где g – ускорение свободного падения, а M – общая масса робота. Предположим что P и есть искомая точка нулевого момента, тогда при $\tau =$

0, $x_P = x_{ZMP}$ получаем:

$$x_{ZMP} = x_C - \frac{y_C}{g} \ddot{x}_C.$$

Составляющие x_C, y_C можно найти из уравнения (1), а \ddot{x}_C находится из заданной параметризации движения.

Рассмотрим движение механизма из [1]. Такой же способ передвижения рассматривается в [7]. Точка D переносимой ноги двигается по заранее заданной траектории в виде верхней части эллипса, а точка подвеса ног O , так как нога на которую опирается модель при переносе остаётся прямой, движется по дуге окружности. На разных шагах во время движения будут

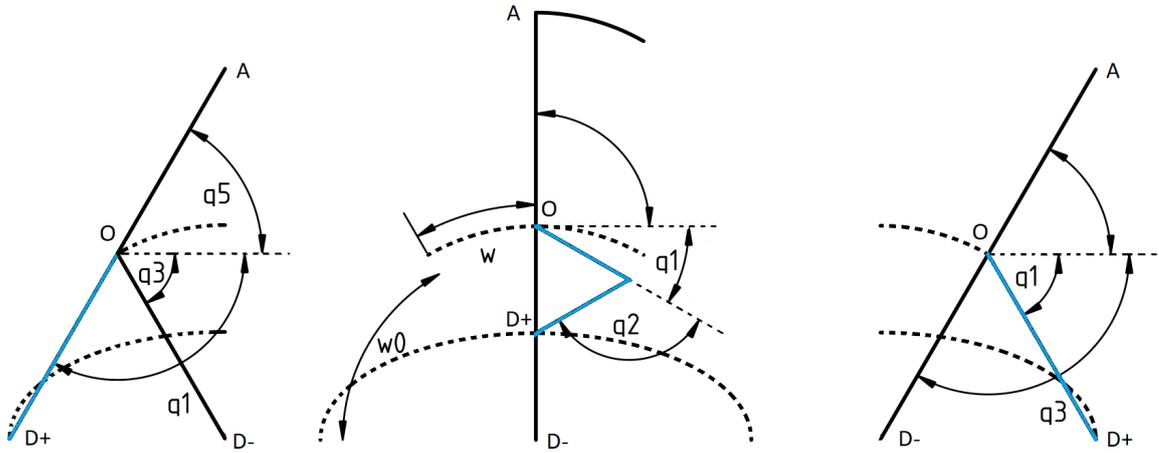


Рис. 5: Схема движения.

различия лишь в том какая нога переносится, а какая действует в качестве опоры, в остальном же движение будет периодическим, поэтому можно рассмотреть только один шаг, следовательно можем рассматривать ногу регулируемую углами q_1, q_2 как переносимую, а вторую как опору.

Шаг в данном случае рассматривался в два этапа: на первой половине шага центр масс находился над точкой D_- опорной ноги, а на второй половине над точкой D_+ переносимой ноги. Посмотрим теперь где будет находиться точка x_{ZMP} относительно точки x_C .

Для x_C в данном случае справедливо выражение 2.1, y_C же выразим

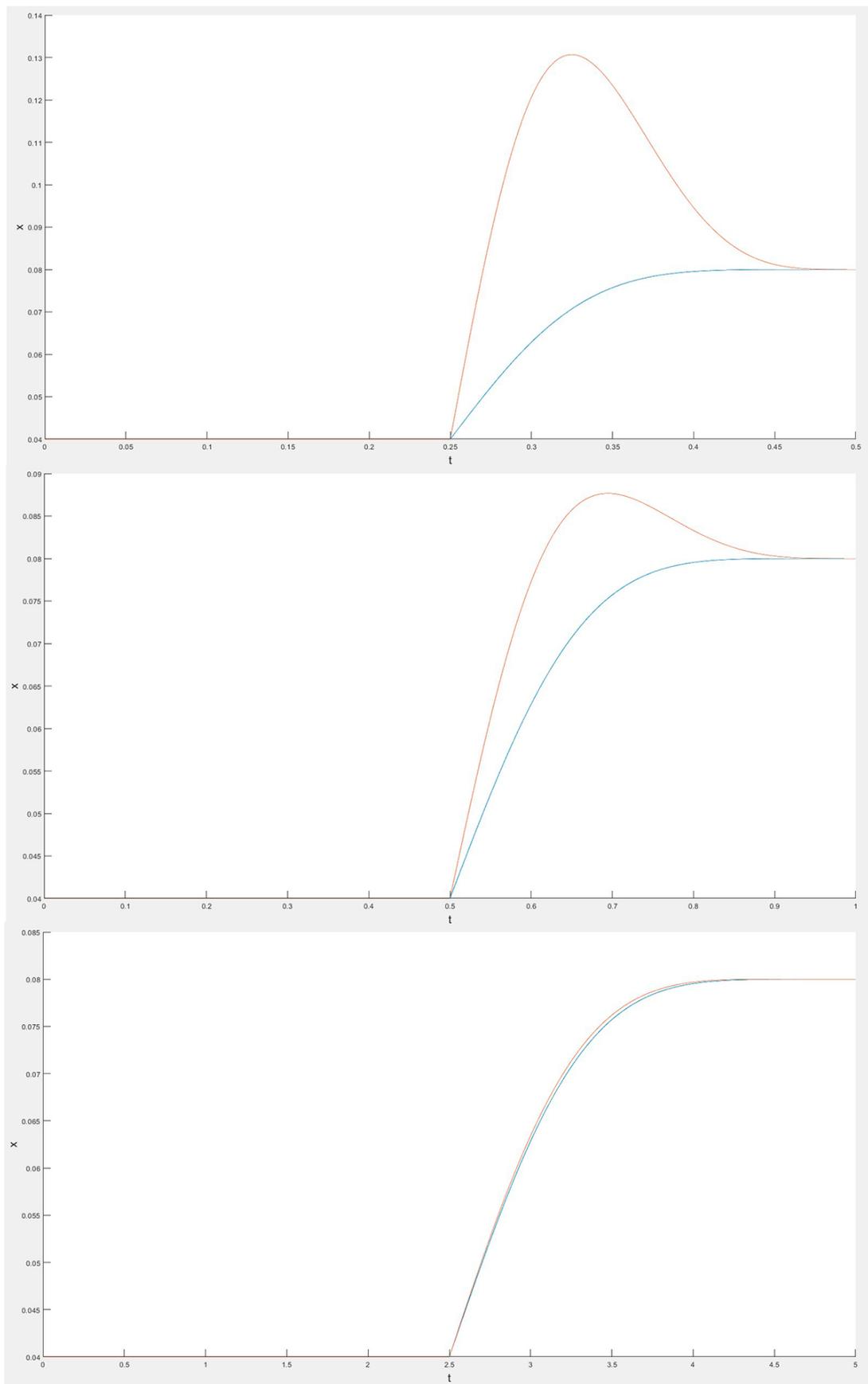


Рис. 6: Графики изменения по x точки нулевого момента (красным) и точки центра масс (синим), при различной длительности шага (0.5, 1.5 с.).

через угол q_5 , зависящий от t и x_A :

$$q_5 = \arccos \frac{x_A - x_O}{OA},$$

$$y_A = \sin q_5 OA,$$

$$y_C = \frac{1}{M}(m_O y_O + m_A y_A + m_B(y_{B+} + y_{B-}) + m_D(y_{D+} + y_{D-})).$$

Моделирование проводилось со следующими параметрами:

$$a = 0.04\text{м}, b = 0.02\text{м}, l_1 = l_2 = 0.1\text{м},$$

$$OA = 0.15\text{м}, m_D = 0.1\text{кг}, m_B = 0.1\text{кг}, m_O = 0.2\text{кг}, m_A = 0.6\text{кг}.$$

Графики построены с помощью программы Matlab R2022a.

На рис. 6. можно увидеть, что в первую половину шага центр масс, как и точка нулевого момента покоятся в начальном положении, но как только центр масс начинает перемещаться, то и точка нулевого момента начинает своё движение, при том чем меньше время шага, тем больше их отклонение. По этому отклонению можно судить какой длины должна быть стопа. Для времени 0.5, 1, 5 с. отклонения соответственно – 0.0507, 0.0077, 3.6995e-6 м.

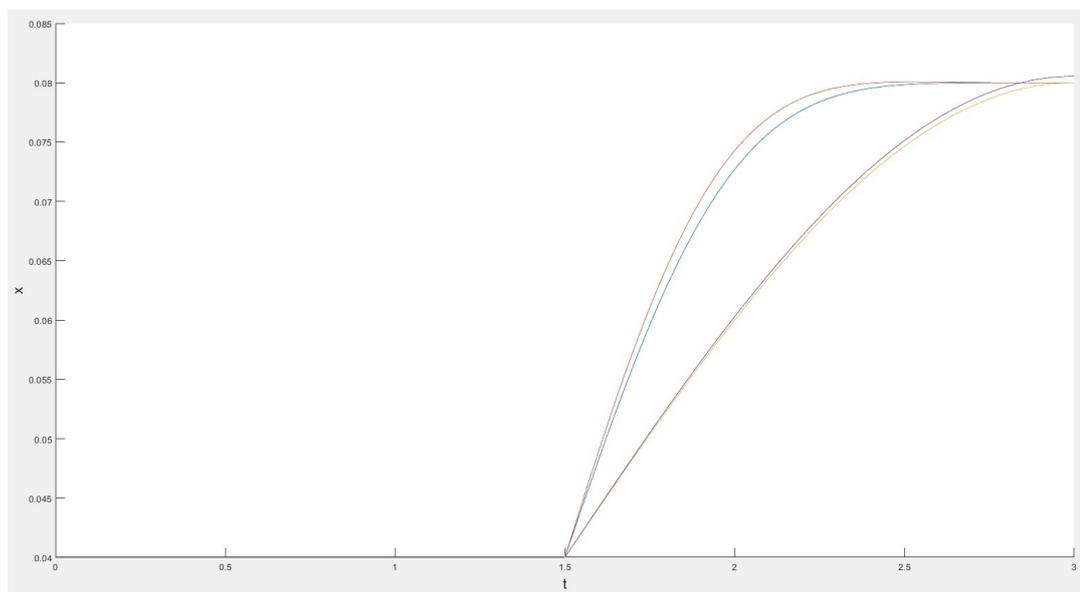


Рис. 7: Работа системы разгон-торможение.

Также, по положению точки нулевого момента можно определить по-

лезность системы разгона и торможения, на рис. 7. можно видеть что при времени шага 3 с. точка нулевого момента при равномерном движении (на графике фиолетовым) выходит за конечную точку положения центра масс (жёлтым), что приведёт к наклону робота. Красным и синим изображены соответственно изменение координаты x точки нулевого момента и точки центра масс с использованием разгона и торможения. По максимальному отклонению можно увидеть, что в случае разгона-торможения проекция точки нулевого момента тоже выходит за пределы стопы, но на несколько меньшее расстояние, для равномерного движения максимальное отклонение составляет $5.4635e-4$ м., а для разгона и торможения – $6.3566e-5$ м.

Глава 3. Расчёт положения точки нулевого момента

Для заданных выше параметров моделирования определим каким значениям должна соответствовать точка X_{ZMP} для сохранения равновесия, время шага - T , для большей стабильности, возьмём равным 2.5 с. Для достижения этой цели предположим, что точка нулевого момента большую часть шага находится над стопой опорной ноги, но в конце периода перемещается к положению второй ноги, которая впоследствии станет опорной. Определим положение X_{ZMP} равным положению текущей опорной ноги большую часть шага, а в качестве перехода будем использовать параболу второй степени, тогда желаемое значение точки нулевого момента равно:

$$\hat{X}_{ZMP} = a - (\beta(t - \alpha)^2)H(\alpha - t) + (\beta(t - T + \alpha)^2)H(t - T + \alpha),$$

где $H(\zeta)$ - функция Хевисайда[9], $t \in [0, T]$, а α и β - параметры параболы удовлетворяющие выражению:

$$\beta\alpha^2 = a.$$

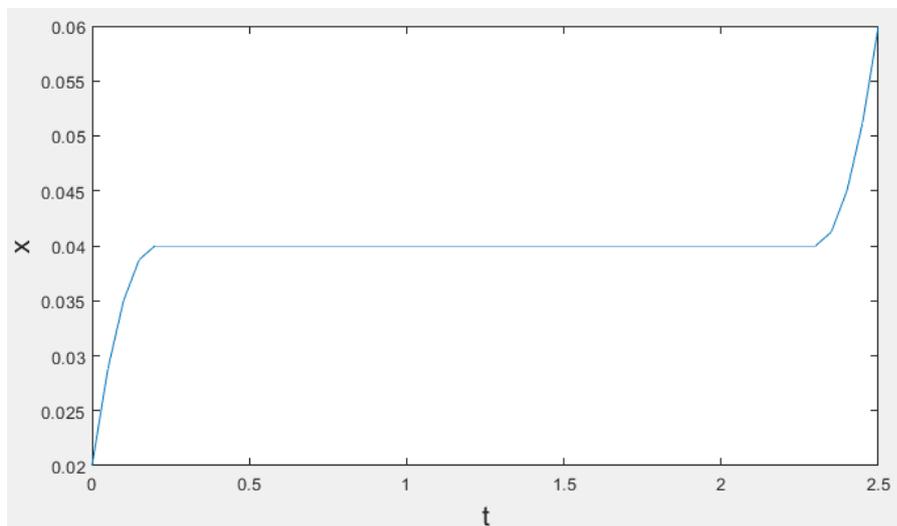


Рис. 8: Вид зависимости \hat{X}_{ZMP} от t .

Для плавного и быстрого перехода определим что: $\alpha = 0.2$, тогда $\beta = 0.5$.

В целях достижения достаточной точности и скорости расчёта зададим X_A как массив из 51 точки – значения X_A на промежутке $t \in [0, T]$ с шагом 0.05 с. Используем алгоритм минимизации функции многих переменных – метод внутренней точки, для подбора соответствующих значений X_A . В качестве минимизируемой функции определим:

$$h = \max_{i=1,51} |X_{ZMP}(i) - \hat{X}_{ZMP}(i)|.$$

За 26 итераций значение удалось сократить до 0.00268275. Полученный

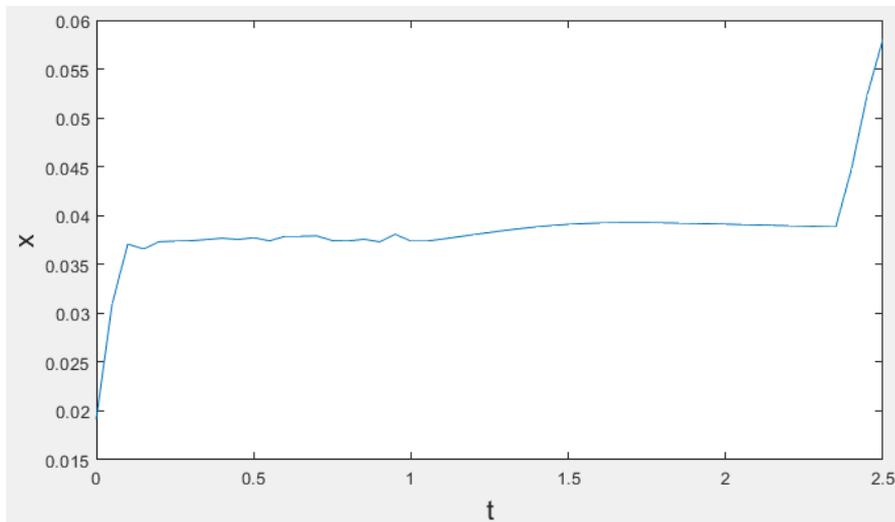


Рис. 9: X_{ZMP} полученная из результата алгоритма.

вектор значений записан в Приложении 1. На рис. 9. можно увидеть поведение X_{ZMP} для этого решения.

На графике видно что практически на всём шаге положение X_{ZMP} немного отстаёт от требуемого, соответственно в процессе симуляции именно это значение необходимо покрыть длиной стопы, добавив также погрешность вычислений. Это расстояние можно покрыть продолжив минимизацию функции, но как видно на рис. 10. это займёт довольно большое количество времени.

Чтобы оценить стабильность походки робота проверяем полученные значения на модели робота. Симуляция выполнена в CoppeliaSim, поло-

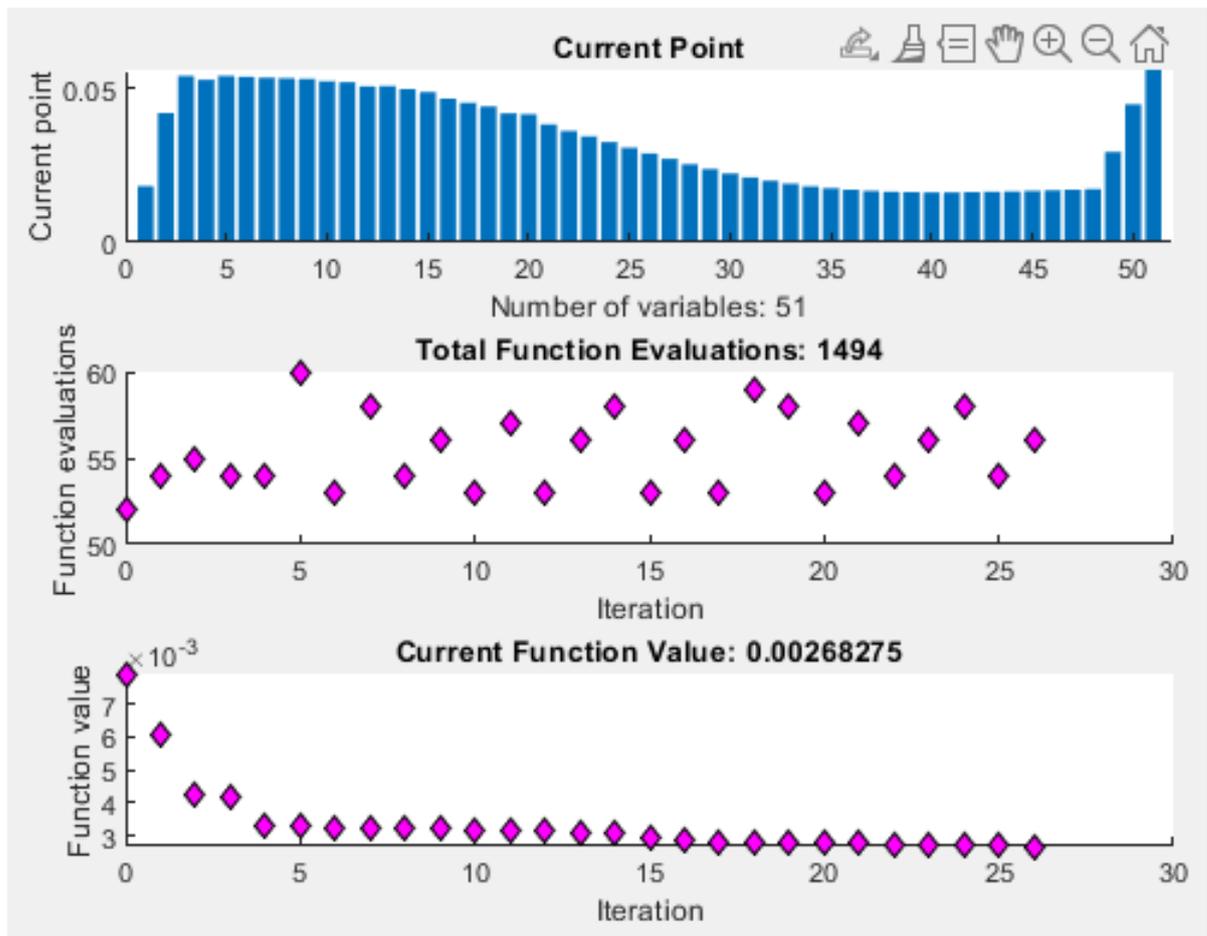


Рис. 10: Вывод алгоритма внутренней точки.

жения приводов подаются через скрипт, написанный на языке Lua, код которого прикреплён в Приложении 2. При времени шага в 2.5 с. робот не заваливается, несмотря на то что центр масс всё же выходит за пределы области покрываемой стопами, в данном случае выход за область обосновывается тем что переносимая нога ещё не приблизилась к земле. Определить завалился робот или нет можно по сигналу инклинометра, датчику положения по высоте, установленному в корпусе робота, или датчиком приближения, направленным под углом к земле.

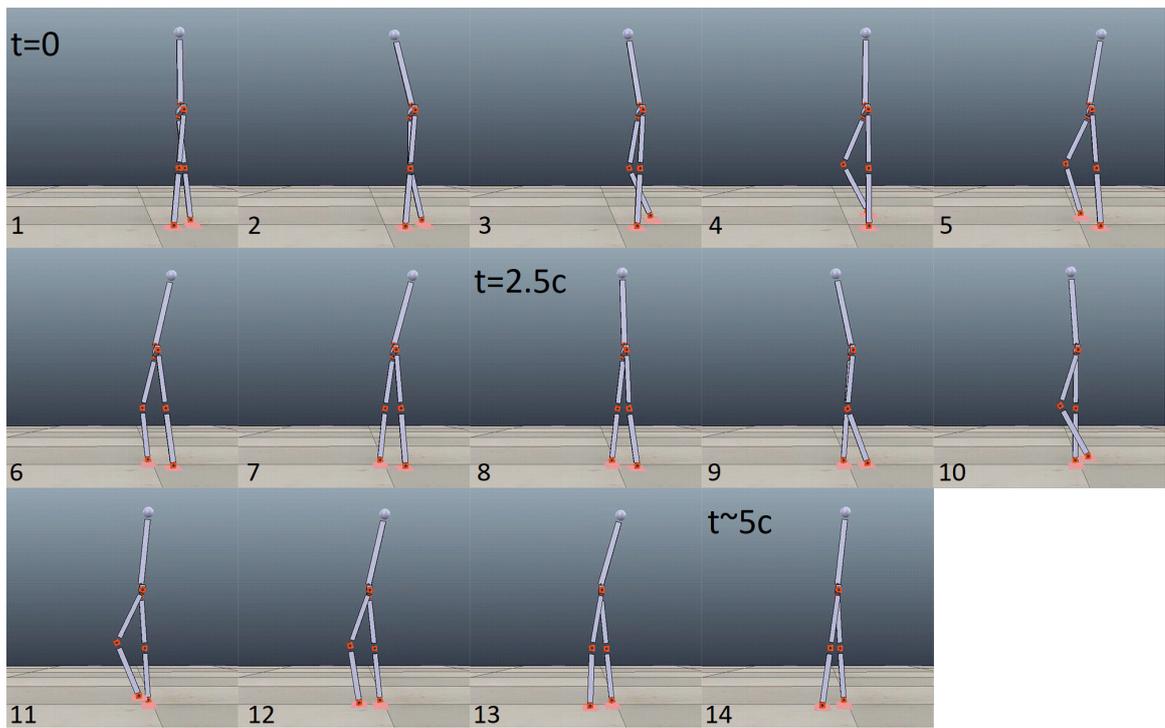


Рис. 11: Симуляция в CoppeliaSim.

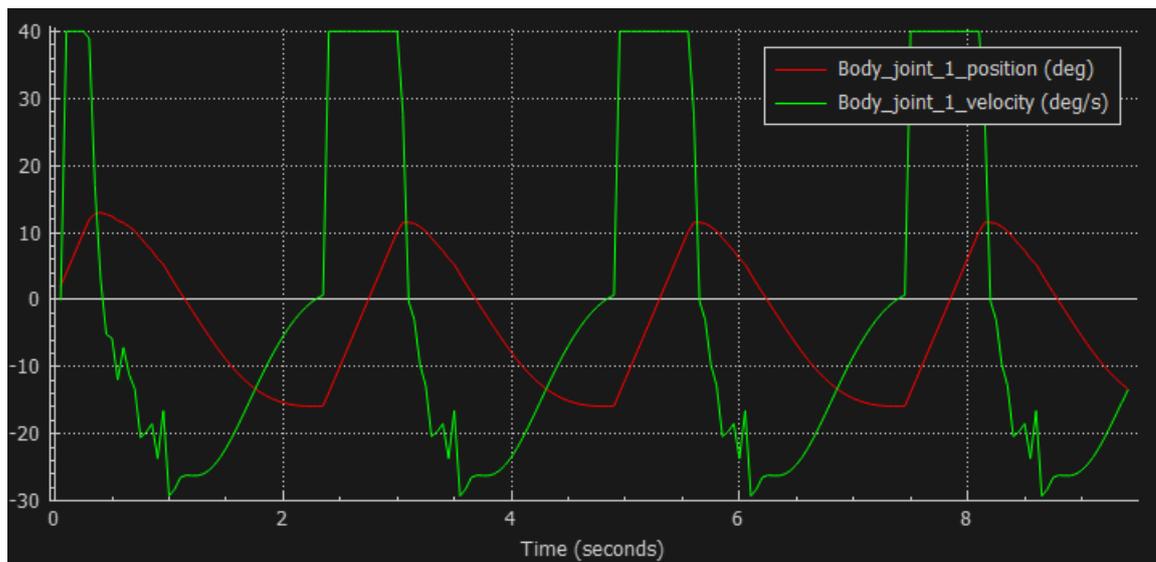


Рис. 12: Вывод значений с приводов.

На рис. 12. по обратной связи от привода можно видеть каким образом на них подаётся управляющий сигнал. Как заметно по графику положение корпуса робота повторяется с периодом равным времени шага, погрешности вычислений X_{ZMP} сильно сказываются на графике скоростей привода, изображённого на рисунке зелёным цветом.

На рис. 13-14 показаны выходные значения q_1 и q_2 соответственно.

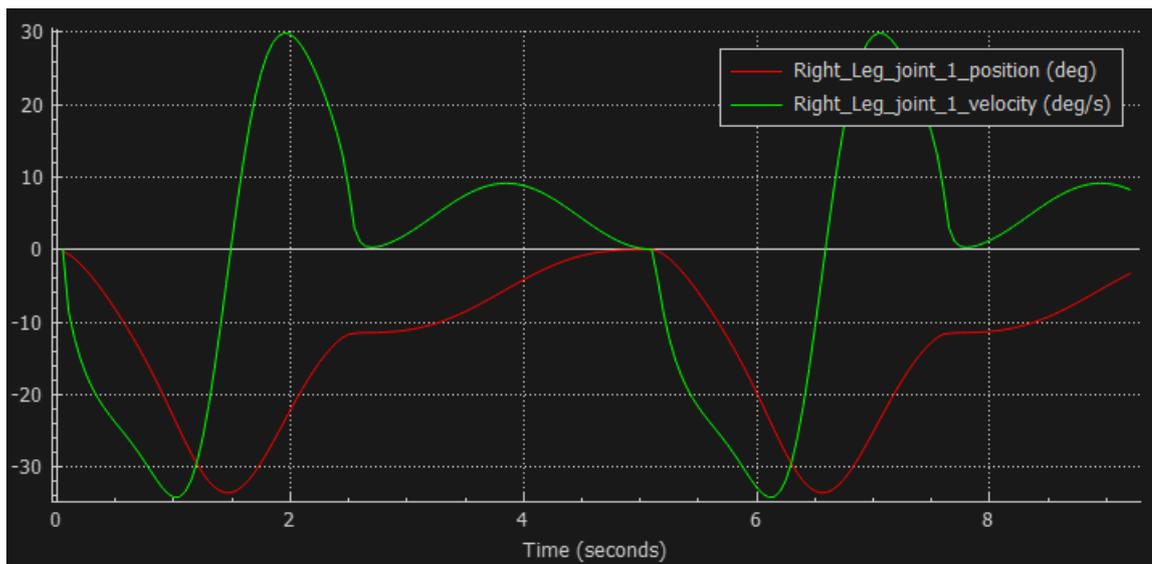


Рис. 13: Вывод значений с приводов.

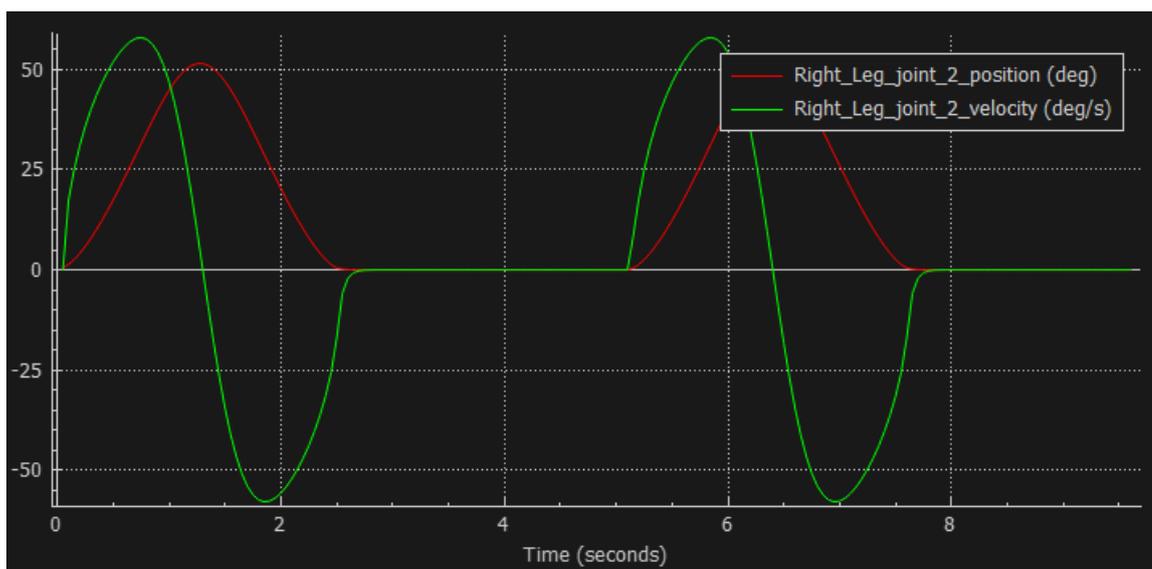


Рис. 14: Вывод значений с приводов.

Заключение

В результате работы было выполнено:

- Изучена тематическая литература.
- Составлено описание датчиков двуногих шагающих роботов.
- Описано статическое и динамическое равновесие двуногих шагающих роботов.
- Рассмотрены методы динамической ходьбы.
- Создан алгоритм расчёта управляющего сигнала при заданных параметрах.
- Создан скрипт для передачи управляемого сигнала приводам модели робота.
- С помощью алгоритма внутренней точки рассчитано положение точки нулевого момента для сокращения длины стопы робота.

Список литературы

- [1] Ташкинов В.А. «Динамическая ходьба двуногого робота». Выпускная квалификационная работа бакалавра. - СПбГУ, ПМ-ПУ, 2020.
- [2] Борисов О.И., Громов В.С., Пыркин А.А. «Методы управления робототехническими приложениями». Учебное пособие. — СПб.: Университет ИТМО, 2016. — 108 с.
- [3] Белецкий В. В. «Плоские линейные модели двуногой ходьбы». М.: Институт прикладной математики АН СССР, 1973, Препринт № 95, - 69с.
- [4] Вукобратович М. «Шагающие роботы и антропоморфные механизмы». М.: Мир, 1976. - 541 с.
- [5] Ларин В. Б. «Управление шагающими аппаратами». Киев: Наук. думка, 1980. - 168 с.
- [6] Маркеев А. П. «Теоретическая механика». Учебник для университетов. - Москва: ЧеРо, 1999, 572 стр.
- [7] Формальский А. М. «Перемещение антропоморфных механизмов». М.: Наука. Главная редакция физико-математической литературы, 1982. - 368 с.
- [8] Шиманчук Д.В. «Введение в современную робототехнику». – Санкт-Петербург, 2018. – 203 с.
- [9] Сетевой ресурс: <https://dic.academic.ru/dic.nsf/ruwiki/913201>.
- [10] Сетевой ресурс: <https://www.mathworks.com/help/> – документация MATLAB
- [11] С.В. Шаныгин «Методика расчета и проектирования многозвенной пространственной манипуляционной системы». - Москва: Известия высших учебных заведений, 2012. - 5с.

- [12] Seong Chiun Lim, Gik Hong Yeap «The Locomotion of Bipedal Walking Robot with Six Degree of Freedom». Published by Elsevier Ltd. Procedia Engineering 41 (2012) 8 – 14.
- [13] Mohammad Osman Tokhi, Noah J Cowan, Abul K M Azad, Gurvinder S Virk, Roger D Eastman «Adaptive Mobile Robotics». World Scientific Publishing Company, 2012. - 904.
- [14] Ложкин П. В. «Метод точки нулевого момента для задачи динамического равновесия РТК AR-600. Движение на месте», Вестник Балтийского федерального университета им. И. Канта. Серия: Физико-математические и технические науки, no. 10, 2013, pp. 81-84.
- [15] Kajita, S. et al. «Biped walking pattern generation by using preview control of zero-moment point.»2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422) 2 (2003): 1620-1626 vol.2.
- [16] М. Г. Огур, Е. А. Некрасова, Ф. Б. Тебуева, Е. Н. Ласкина, «Разработка алгоритма интегрального контроля динамического баланса антропоморфного робота», Информационные технологии интеллектуальной поддержки принятия решений (ITIDS'2018) : Труды VI Всероссийской конференции (с приглашением зарубежных ученых), Уфа-Ставрополь, 28–31 мая 2018 года. – Уфа-Ставрополь: ГОУ ВПО "Уфимский государственный авиационный технический университет 2018. – С. 178-182.
- [17] Поляхов, Н.Д. «Управление перевернутым двухзвенным маятником», Н.Д. Поляхов, Р.И. Галиулин, Известия СПбГЭТУ «ЛЭТИ». – 2015. – № 6. – С. 65–69.
- [18] Nenchev, Dragomir N., Atsushi Konno, and Teppei Tsujita. «Humanoid robots: Modeling and control». Butterworth-Heinemann, 2018.
- [19] B. Braunl, «Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems», Germany, Berlin:Springer-Verlag, 2006.
- [20] Ю. Ф. Голубев, «Нейронные сети в мехатронике, Фундамент. и прикл. матем.», 2005, том 11, выпуск 8, 81–1.

- [21] Richard S. Sutton, Andrew G. Barto, «Reinforcement learning: An introduction.», MIT press, 2018.
- [22] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, Daan Wierstra, «Continuous control with deep reinforcement learning.», 2015.
- [23] Geng T, Porr B, Wörgötter F. «A reflexive neural network for dynamic biped walking control.», eural Comput. 2006 May;18(5):1156-96. doi: 10.1162/089976606776241057. PMID: 16595061.
- [24] Сетевой ресурс: <https://weightagnostic.github.io/>
- [25] Сетевой ресурс: <https://www.coppeliarobotics.com/helpFiles/> - документация CoppeliaSim
- [26] Бжихатлов И.А. «Моделирование робототехнических систем в программе V-REP. Учебно-Методическое пособие.»— СПб: Университет ИТМО, 2018. – 59с.
- [27] Maciej Komosinski and Andrew Adamatzky (2009). «Artificial Life Models in Software.»New York: Springer. ISBN 978-1-84882-284-9.
- [28] Andrew Adamatzky and Maciej Komosinski (2009). «Artificial Life Models in Hardware.»New York: Springer. ISBN 978-1-84882-529-1.

Приложение 1

```
1 %initial values
2 md = 0.1; mb = 0.1; mo = 0.2; ma = 0.6;
3 M = 2*md + 2*mb + mo + ma;
4 a = 0.04; b = 0.02; l1 = 0.1; l2 = 0.1; psy0 = acos((a/2)/(l2+l1));
5 T = 2.5; OA = 0.15;
6 syms t
7 %angular coordinates
8 fi = pi*(t/T-(1/(2*pi)))*sin((2*pi*t)/T);
9 psy = psy0 - (2*psy0-pi)*(t/T-(1/(2*pi)))*sin((2*pi*t)/T);
10
11 %parametrization:
12 ro = [a-(l1+l2)*cos(psy), (l1+l2)*sin(psy)];
13 rd = [a-a*cos(fi), b*sin(fi)];
14 rd2 = [a, 0];
15 od = rd - ro;
16 od2 = rd2 - ro;
17 OD = norm(od);
18 gamma = acos((l1^2+l2^2-OD^2)/(2*l1*l2));
19 alfa = acos((l1^2-l2^2+OD^2)/(2*l1*OD));
20 q2 = gamma - pi;
21 q1 = alfa - acos(od(1)/OD);
22 q3 = -acos(od2(1)/(l1+l2));
23
24
25 rb = [cos(q1)*l1 + ro(1), ro(2) - sin(-q1)*l1];
26 rb2 = [(rd2(1)+ro(1))/2, ro(2)/2];
27
28
29 syms xa
30
31 xa_ca = [0.0183339082877097 0.0421047587800002 0.0542020000237644 ...
0.0529569713176412 0.0541564087600582 0.0539350802722694 ...
0.0536322847489616 0.0534500238001730 0.0532188308088906 ...
0.0523994170022735 0.0521233822678227 0.0507557583067255 ...
0.0508673502596844 0.0499477243420299 0.0489735291005645 ...
0.0468023824502753 0.0453844911576979 0.0442453481585130 ...
0.0420604857654064 0.0417409456724771 0.0383845675310047 ...
0.0362616423231464 0.0344605519131369 0.0326185904840757 ...
0.0307665390906899 0.0289371383374087 0.0271630841509686 ...
0.0254750378819290 0.0238998869654568 0.0224594252987798 ...
0.0211695366993519 0.0200398882303468 0.0190740860331863 ...
0.0182702132732414 0.0176216510213582 0.0171180737775894 ...
```

```

0.0167465112630811  0.0164923777678683  0.0163403889761456  ...
0.0162753107423876  0.0162825103985115  0.0163483046746972  ...
0.0164601163101068  0.0166064624867687  0.0167768018269209  ...
0.0169612619135583  0.0171502518269811  0.0173339173748389  ...
0.0294827885717279  0.0449132697435344  0.0561444774307089 ] ;
32
33 q5 = acos((xa-ro(1))/OA);
34 ansq5 = eval(subs(q5,t,(0:0.05:T)));
35 for i=1:51
36     ansq5(i) = subs(ansq5(i),xa,xa_ca(i));
37 end
38
39 ansq1 = rad2deg(eval(subs(q1,t,(0:0.05:T))));
40 for i=1:51
41     ansq1(i) = subs(ansq1(i),xa,xa_ca(i));
42 end
43 ansq2 = rad2deg(eval(subs(q2,t,(0:0.05:T))));
44 for i=1:51
45     ansq2(i) = subs(ansq2(i),xa,xa_ca(i));
46 end
47 ansq3 = rad2deg(eval(subs(q3,t,(0:0.05:T))));
48 for i=1:51
49     ansq3(i) = subs(ansq3(i),xa,xa_ca(i));
50 end

```

```

1 function crit = gen(xa_input)
2 %initial values
3 md = 0.1; mb = 0.1; mo = 0.2; ma = 0.6;
4 M = 2*md + 2*mb + mo + ma;
5 a = 0.04; b = 0.02; l1 = 0.1; l2 = 0.1; psy0 = acos((a/2)/(l2+l1));
6 T = 2.5; OA = 0.15;
7 syms t
8
9 %angular coordinates
10 fi = pi*(t/T-(1/(2*pi)))*sin((2*pi*t)/T);
11 psy = psy0 - (2*psy0-pi)*(t/T-(1/(2*pi)))*sin((2*pi*t)/T);
12
13 %parameterization:
14 ro = [a-(l1+l2)*cos(psy),(l1+l2)*sin(psy)];
15 rd = [a-a*cos(fi),b*sin(fi)];
16 rd2 = [a,0];
17 od = rd - ro;
18 od2 = rd2 - ro;
19 OD = norm(od);

```

```

20 gamma = acos((l1^2+l2^2-OD^2)/(2*l1*l2));
21 alfa = acos((l1^2-l2^2+OD^2)/(2*l1*OD));
22 q2 = gamma - pi;
23 q1 = alfa - acos(od(1)/OD);
24 q3 = -acos(od2(1)/(l1+l2));
25 rb = [cos(q1)*l1 + ro(1), ro(2) - sin(-q1)*l1];
26 rb2 = [(rd2(1)+ro(1))/2, ro(2)/2];
27
28 syms xa
29
30 q5 = acos((xa-ro(1))/OA);
31 ya = sin(q5)*OA;
32 yc = (1/M)*(mo*ro(2)+ ma*ya + mb*(rb(2)+rb2(2))+md*(rd(2)+rd2(2)));
33 xc = (1/M)*(mo*ro(1)+ ma*xa + mb*(rb(1)+rb2(1))+md*(rd(1)+rd2(1)));
34 xc2 = diff(xc, t, 2);
35 Step = a - (0.5*(t-0.2)^2)*heaviside(0.2-t) + ...
        (0.5*(t-T+0.2)^2)*heaviside(t-T+0.2);
36 Xz = subs(xc - xc2*(yc/9.8), t, (0:.05:T));
37 for i=1:51
38     Xz(i) = subs(Xz(i), xa, xa_input(i));
39 end
40 Xzmp = eval(Xz);
41 crit = max(abs(Xzmp - eval(subs(Step, t, (0:.05:T)))));

```

Приложение 2

```

1 if (sim_call_type==sim.syscb_init) then
2     graph=sim.getObject('/Graph')
3     joint1Vel=sim.addGraphStream(graph, 'Right Leg joint 2 ...
        position', 'deg', 0, {1,0,0})
4     joint2Vel=sim.addGraphStream(graph, 'Right Leg joint 2 ...
        velocity', 'deg/s', 0, {0,1,0})
5     switch = true
6     i = 1
7     local file = io.open("c:\\qs.txt")
8     local qs = {}
9     local p = 0
10    if file then
11        for line in file:lines() do
12            p = p + 1
13            qs[i] = line
14        end
15        file:close()

```

```

16     end
17     q1 = qs [1]
18     q2 = qs [2]
19     q3 = qs [3]
20     q5 = qs [4]
21     q1end = q1 [101]
22     q3end = q3 [101]
23     bodyJoint = sim.getObjectHandle ("Revolute_joint8")
24     leftLegJoints={sim.getObjectHandle ("Revolute_joint2"), ...
25     sim.getObjectHandle ("Revolute_joint3"), ...
26     sim.getObjectHandle ("Revolute_joint4")}
27     rightLegJoints={sim.getObjectHandle ("Revolute_joint1"), ...
28     sim.getObjectHandle ("Revolute_joint"), ...
29     sim.getObjectHandle ("Revolute_joint0")}
30     movingLegJoint = ...
31         {rightLegJoints [1], rightLegJoints [2], rightLegJoints [3]}
32     notMovingLegJoint = ...
33         {leftLegJoints [1], leftLegJoints [2], leftLegJoints [3]}
34     sensors = {sim.getObjectHandle ("Proximity_sensor1"), ...
35     sim.getObjectHandle ("Proximity_sensor2")}
36 end
37
38 if (sim_call_type==sim.syscb_actuation) then
39     i = i + 2
40     j = math.floor (i)
41     if j > 101 then
42         j = 101
43     end
44     sim.setJointTargetPosition (movingLegJoint [1], -q1 [j])
45     sim.setJointTargetPosition (movingLegJoint [2], -q2 [j])
46     sim.setJointTargetPosition (movingLegJoint [3], q1 [j]+q2 [j])
47     sim.setJointTargetPosition (notMovingLegJoint [1], -q3 [j])
48     sim.setJointTargetPosition (notMovingLegJoint [3], q3 [j])
49     sim.setJointTargetPosition (bodyJoint, -q5 [j])
50     if i > 101 then
51         if switch then
52             movingLegJoint = ...
53                 {leftLegJoints [1], leftLegJoints [2], leftLegJoints [3]}
54             notMovingLegJoint = ...
55                 {rightLegJoints [1], rightLegJoints [2], rightLegJoints [3]}
56             for k = 1,101 do
57                 q3 [k] = q3 [k] + q1end
58                 q1 [k] = q1 [k] + q3end
59             end
60         else
61             movingLegJoint = ...

```

```

58         {rightLegJoints [1] , rightLegJoints [2] , rightLegJoints [3]}
notMovingLegJoint = ...
        {leftLegJoints [1] , leftLegJoints [2] , leftLegJoints [3]}
59     for k = 1,101 do
60         q3[k] = q3[k] - q1end
61         q1[k] = q1[k] - q3end
62     end
63     end
64     switch = not(switch)
65     i = 1
66 end
67 end
68
69
70 if (sim_call_type==sim.syscb_sensing) then
71     sens = sim.readProximitySensor(sensors [1])
72     sim.setGraphStreamValue(graph , joint1Vel , 180 * ...
73     sim.getJointPosition(rightLegJoints [2]) / math.pi)
74     sim.setGraphStreamValue(graph , joint2Vel , 180 * ...
75     sim.getJointVelocity(rightLegJoints [2]) / math.pi)
76 end

```