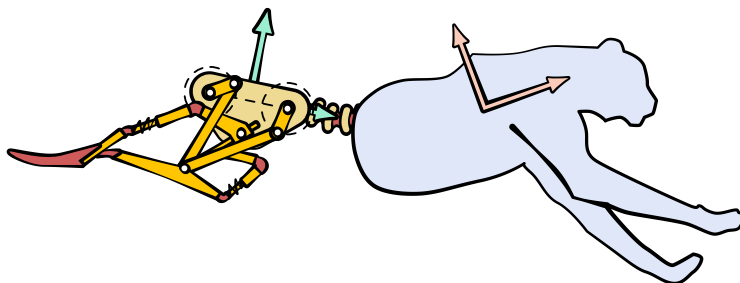


И.И. Борисов, С.А. Колюбин

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ  
МЕХАТРОННЫХ СИСТЕМ



Санкт-Петербург  
2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

**И.И. Борисов, С.А. Колюбин**

**ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ  
МЕХАТРОННЫХ СИСТЕМ**

**УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ**

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ  
ИТМО

по направлениям 15.04.06 "Мехатроника и робототехника"  
и 27.04.03 "Системный анализ и управление"

в качестве учебного пособия для реализации основных  
профессиональных образовательных программ высшего образования  
магистратуры.



Санкт-Петербург  
2020

Борисов И.И., Колобин С.А., Имитационное моделирование мехатронных систем — СПб: Университет ИТМО, 2020. — 103 с.

**Рецензент:**

Шаветов Сергей Васильевич, кандидат технических наук, доцент факультета систем управления и робототехники, Университета ИТМО.

Учебное пособие на различных примерах рассматривает способы построения имитационных моделей мехатронных и робототехнических систем. Имитационное моделирование – метод исследования, при котором изучаемая система заменяется моделью, с достаточной точностью описывающей реальную систему, для проведения экспериментов с целью проверки, оценки и поддержки конструктивных решений, изучения взаимодействия между составными элементами робота и с объектами окружающей среды, анализа кинематических и динамических характеристик, настройки алгоритмов управления и получения прочей информации о рассматриваемой системе. Пособие предназначено для студентов, обучающихся по направлениям 15.04.06 Мехатроника и робототехника и 27.04.03 Системный анализ и управление.

Пособие подготовлено при поддержке гранта Российского научного фонда (проект №17-79-20341).



**Университет ИТМО** – ведущий ВУЗ России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2020  
© Борисов И.И., Колобин С.А., 2020

# Оглавление

<b>Вступление</b>	<b>5</b>
<b>1 Основы моделирования мехатронных систем в среде MATLAB</b>	<b>10</b>
1.1 Структура Simscape . . . . .	12
1.2 Интерфейс . . . . .	14
1.3 Модель маятника . . . . .	15
1.4 Управление однозвенным маятником . . . . .	21
<b>2 Моделирование многозвенных механизмов</b>	<b>24</b>
2.1 Структурный анализ . . . . .	25
2.2 Кинематический анализ . . . . .	26
2.3 Имитационное моделирование . . . . .	31
<b>3 Моделирование контактного взаимодействия</b>	<b>37</b>
3.1 Теоретические основы контактного взаимодействия . . . . .	38
3.2 Моделирование контакта в Simscape . . . . .	43
3.3 Моделирование контактного взаимодействия плоского робота с объектами окружающей среды . . . . .	45
<b>4 Моделирование составных элементов</b>	<b>53</b>
4.1 Двигатель постоянного тока . . . . .	53
4.1.1 Аналитическая модель . . . . .	53
4.1.2 Имитационная модель . . . . .	56
4.2 Ремни, тросы, шкивы и катушки . . . . .	59

<b>5</b>	<b>Комплексное моделирование мехатронных систем</b>	<b>63</b>
5.1	Установка Simscape Multibody Link и экспорт моделей САПР . . . . .	64
5.2	Моделирование захватных устройств . . . . .	66
5.2.1	Описание механизма . . . . .	66
5.2.2	Описание задач моделирования . . . . .	67
5.2.3	Построение имитационной модели . . . . .	68
5.2.4	Результаты моделирования . . . . .	72
5.3	Моделирование галопирующих роботов . . . . .	75
5.3.1	Описание механизмов . . . . .	75
5.3.2	Описание задач моделирования . . . . .	77
5.3.3	Имитационное моделирование . . . . .	77
5.3.4	Моделирование привода ноги . . . . .	78
5.3.5	Моделирование редуцированного механизма ноги . . . . .	81
5.3.6	Моделирование полного механизма ноги . . . . .	82
	<b>Приложение А. Задания для курсового проекта</b>	<b>89</b>
A.1	Шагающий робот на основе механизма Кланна . . . . .	91
A.2	Прыгающий робот Минитаур . . . . .	92
A.3	Шагающий робот на основе механизма Янсена . . . . .	93
A.4	Неполноприводное захватное устройство . . . . .	94
A.5	Стопоходящая машина Чебышева . . . . .	95
A.6	3R робот . . . . .	96
A.7	Механизм Хайна с гибким звеном . . . . .	97
A.8	Кулисно-рычажный механизм убирающегося шасси самолёта . . . . .	98
A.9	Энергоэффективный галопирующий робот с гибкой спиной . . . . .	99
	<b>Литература</b>	<b>103</b>

# Введение

## Что есть мехатронная система?

Конструирование мехатронных систем (МС) является сложной комплексной задачей, требующей от инженеров знаний в разных областях науки: теоретической механике, теории механизмов и машин, сопромата, деталей машин, электроники, электротехники, теории автоматического управления, не говоря о базовых курсах физики, линейной алгебры, математического анализа и других.

Процесс проектирования сложен и трудоёмок, при этом качество разрабатываемой технической системы определяется в первую очередь качеством проектирования, поэтому неудачные решения на этом этапе не всегда могут компенсироваться на последующих. Затраты на качественное проектирование окупаются впоследствии за счёт экономии средств в процессе эксплуатации [1].

Процесс проектирования мехатронной системы – результат творческого труда, зачастую, большого числа высококвалифицированных специалистов разных направлений, при этом для успешного взаимодействия между собой специалисты должны понимать друг друга. Как дирижёр умеет играть на всех инструментах оркестра, так и специалист в области мехатроники и робототехники должен разбираться во всех аспектах работы разрабатываемой мехатронной системы. Помочь инженеру разработать мехатронную систему, убедиться в её работоспособности и способности обеспечивать требуемый характер работы может имитационное моделирование, виртуальный эксперимент, позволяющий верифицировать работу устройства или системы целиком.

Цель данного методического пособия – помочь инженерам в освоении имитационного моделирования мехатронных систем.

*« Мехатроника<sup>a</sup>, с точки зрения инженера теории управления, представляет собой синергетический подход к использованию передовых методов управления, измерения данных и приведения в движение в таких областях науки и техники, как робототехника и автоматизация, теория автоматического управления, автомобильные, биомедицинские, микро- и наноразмерные, энергетические и прочие системы.»*

<sup>a</sup>IFAC, TC 4.2 Mechatronic Systems, <https://tc.ifac-control.org/4/2>

## Имитационное моделирование

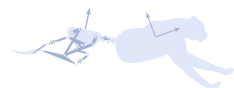
Имитационное моделирование<sup>1</sup> – метод, позволяющий строить модели, описывающие процессы так, как они проходили бы в действительности. Такую модель можно «проиграть» во времени как для одного испытания, так и заданного их множества. Имитационное моделирование можно рассматривать как разновидность экспериментальных испытаний.

Имитационное моделирование часто используется в качестве замены экспериментов на реальных системах, что связано с рядом преимуществ:

- По сравнению с реальными экспериментами, виртуальные требуют значительно меньших финансовых и временных затрат, поскольку моделирование значительно дешевле, чем создание реальных прототипов;
- Все аспекты виртуальных экспериментов повторяемы, что не может быть гарантировано для реальной системы, либо требует значительных затрат;
- Имитационные модели полностью управляемы. Таким образом, все входные переменные и параметры системы могут

---

<sup>1</sup><http://sewiki.ru/>



быть predeterminedены, что обычно не свойственно реальной системе;

- Имитационные модели полностью контролируются. Все выходные переменные и внутренние состояния известны, тогда как в реальной системе для измерения чего-либо необходим специальный датчик;
- В некоторых случаях «постоянные времени» эксперимента и наблюдателя несовместимы, например, при исследовании элементарных частиц или галактик;
- В некоторых случаях эксперимент исключается по моральным соображениям, например эксперименты на людях в области медицинских технологий.

Однако моделированию свойственны и недостатки, например, точность и скорость моделирования ограничиваются мощностью компьютера, но главным недостатком модели является её суть.

### Модель всегда остаётся моделью!

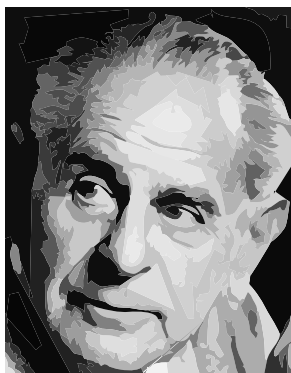
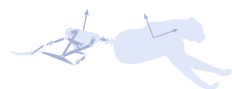


Рисунок 1. Карл Поппер

Судить о работе системы и её характеристиках можно только в рамках её работоспособности, модель всегда есть и должна быть фальсифицируемой, т.е. удовлетворять критерию научности **Карла Поппера**<sup>2</sup>.

Теория удовлетворяет критерию Поппера (является фальсифицируемой и, соответственно, научной) в том случае, если существует возможность её экспериментального или иного опровержения. Согласно критерию Поппера, научная теория не может быть принципиально непроверяемой [2]. Аналогично, имитационная модель имеет свои рамки применимости, и инженер должен понимать их.

<sup>2</sup>[https://en.wikipedia.org/wiki/Karl\\_Popper](https://en.wikipedia.org/wiki/Karl_Popper)





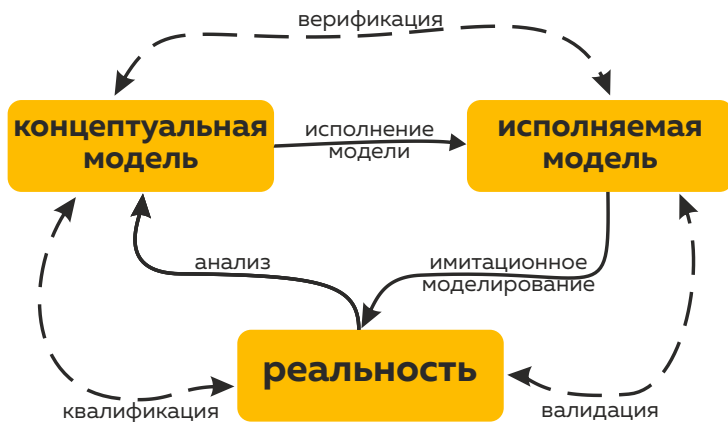
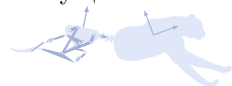


Рисунок 2. Схема процедуры моделирования

Тем не менее, во многих случаях преимущества имитационного моделирования перевешивают недостатки. На рисунке 2 изображена схема процедуры моделирования [2]. *Реальность* – сущность, ситуация или система, подлежащая исследованию с помощью моделирования. Само моделирование можно разбить на два этапа. На первом этапе *реальность* анализируется и представляется в виде уравнений, физических законов либо словесных описаний, которые составляют *концептуальную модель*.

Затем для этой *концептуальной модели* должна быть определена область применения, в рамках которой модель должна обеспечивать приемлемое представление *реальности*. Кроме того, должна быть определена степень соответствия между *концептуальной моделью* и *реальностью*. Концептуальная модель должна удовлетворять *квалификации* заданной области применения модели для обеспечения требуемого уровня соответствия *реальности*.

На втором этапе моделирования *концептуальная модель* преобразуется в *исполнительную модель*. Моделирование – представление *реальности* в виде математической модели, имитационное моделирование или симуляция – представление *реальности* в виде кода, скрипта или набора инструкций, описывающих, как модель должна реагировать на внутренние и внешние возмущения.



Разработка моделей – сложный процесс, склонный к ошибкам. Надёжность моделирования в решающей степени зависит от качества модели. Требуются методы и инструменты, позволяющие проверять модели. *Верификация* модели исследует, отражает ли исполняемая модель концептуальную модель в заданных пределах точности. Верификация переносит область применения концептуальной модели в исполняемую модель. *Валидация* модели, с другой стороны, должна подсказать нам, пригодна ли исполняемая модель для выполнения поставленной задачи в рамках своей области применения. Другими словами: *верификация* гарантирует, что система правильно смоделирована, в то время как *валидация* означает моделирование правильной системы.

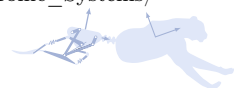
## Структура методического пособия

Данное методическое пособие на различных примерах рассматривает способы построения имитационных моделей мехатронных и робототехнических систем:

1. Моделирование плоских механизмов с замкнутой кинематической цепью и позиционным управлением с помощью ПИД регулятора;
2. Моделирование плоских механизмов с открытой кинематической цепью и управлением по механическому импедансу;
3. Моделирование контактного взаимодействия твёрдых объектов методом штрафов (penalty based approach);
4. Моделирование электродвигателей, редукторов, тросов, гибких тел и прочих составляющих мехатронных систем;
5. Импорт моделей из SolidWorks в MATLAB;
6. Проектирование адаптивных захватных устройств;
7. Проектирование энергоэффективных галопирующих роботов, способных осуществлять динамическую локомоцию.

Примеры моделей, представленных в данном учебном пособии, находятся на [GitHub](#)<sup>3</sup>.

<sup>3</sup>[https://github.com/iiborisov/Simulation\\_of\\_Mechatronic\\_Systems/](https://github.com/iiborisov/Simulation_of_Mechatronic_Systems/)



# Глава 1

## ОСНОВЫ МОДЕЛИРОВАНИЯ МЕХАТРОННЫХ СИСТЕМ В СРЕДЕ MATLAB

**MATLAB**<sup>1</sup> – мультипарадигменная вычислительная среда и язык программирования, которые позволяют осуществлять матричные вычисления, построение графиков функций и данных, реализацию алгоритмов, создание пользовательских интерфейсов и взаимодействие с программами, написанными на других языках и многое другое.

**MathWorks**<sup>2</sup> разделяет непосредственно сам MATLAB, позволяющий анализировать данные, разрабатывать алгоритмы и создавать математические модели, и Simulink – программу для моделирования систем, представленных графически в виде мультипарадигменных блок-схем.

Данное учебное пособие нацелено на помощь в освоении построения имитационных моделей физических систем с помощью блоков Simscape в программе Simulink. Simscape позволяет быстро моделировать физические системы в среде Simulink, например электродвигатели, мостовые выпрямители, гидравлические приводы или холодильные системы, собирая компоненты из библиотеки Simscape в блок-схему. Созданные модели можно переместить в подсистемы, чтобы их можно было повторно использовать в других моделях [3].

---

<sup>1</sup><https://en.wikipedia.org/wiki/MATLAB>

<sup>2</sup><https://www.mathworks.com/>

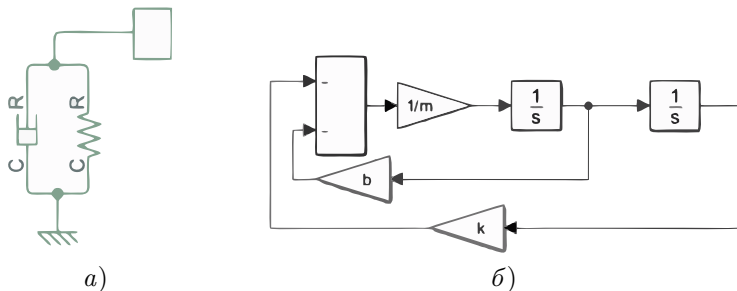


Рисунок 1.1. Блок-схемы массы на пружине и демпфере в Simscape *a)* и Simulink *б)*

Моделирование в Simscape основано на физических взаимосвязях. Например, чтобы смоделировать массу на пружине и демпфере, необходимо соединить компоненты *Mass*, *Spring* и *Damper* вместе, после чего Simscape самостоятельно выведет математическое описание системы (рисунок 1.1, *a*). Это гораздо проще, чем выводить уравнения и реализовывать их в коде или собирать из блоков ввод-вывод Simulink (рисунок 1.1, *б*). По мере усложнения системы метод моделирования, применяемый Simscape, становится ещё более мощным.

Есть возможность создавать пользовательские компоненты с помощью языка Simscape на базе MATLAB. Пользователь может изменить код, который предоставляется MATLAB, чтобы включить в него дополнительные эффекты. Электрические, гидравлические, пневматические и другие компоненты могут быть интегрированы в имитационную модель, используя компоненты из семейства продуктов Simscape, который предлагает набор фундаментальных компонентов во многих физических областях и предоставляет технологию для моделирования и анализа физических систем.

Модели Simscape интегрируются непосредственно с блок-схемами в Simulink, автоматами состояний в Stateflow и функциями MATLAB. Вся система может быть смоделирована в одной среде. Это ускоряет выполнение таких задач, как проектирование систем управления и оптимизация производительности на уровне системы.



## 1.1 Структура Simscape

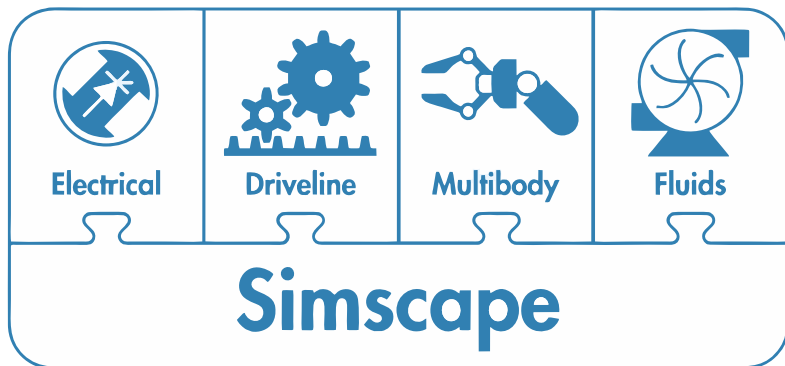
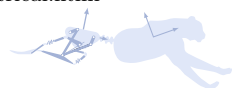


Рисунок 1.2. Структура Simscape

### Simscape Electrical

**Simscape Electrical**<sup>3</sup> предоставляет библиотеки компонентов для моделирования электронных, мехатронных и электроэнергетических систем. Библиотека Simscape Electrical [4] включает в себя модели электрических схем и их компонентов, датчиков, источников энергии, актуаторов, двигателей, энергосистем и прочих компонентов для мехатронных систем. Эти компоненты могут быть использованы для оценки архитектуры аналоговых схем, разработки мехатронных систем с электроприводами, а также анализа генерации, преобразования, передачи и потребления электрической энергии. Simscape Electrical помогает разрабатывать системы управления и тестировать производительность на уровне системы.

<sup>3</sup><https://www.mathworks.com/products/simscape-electrical.html>



## Simscape Driveline

**Simscape Driveline**<sup>4</sup> предоставляет библиотеки компонентов для моделирования вращательных и поступательных механических систем, передающих и преобразующих движение. Simscape Driveline [5] включает в себя множество типов компонентов: шестерни, редукторы, передачи, трансмиссии, ремни, шкивы, муфты, моторы, гидротрансформаторы, трансмиссии, шины, датчики, идеальные источники сил и моментов и многое другое.

## Simscape Multibody

**Simscape Multibody**<sup>5</sup> предоставляет среду для имитационного моделирования многосвязных трёхмерных механических систем, которые собираются из блоков, представляющих собой тела, кинематические соединения, а также силовые элементы, в блок-схему [6]. Сборки моделей могут быть импортированы из САПР, включая все значения масс, инерций, соединения и детализированную визуализацию. Кроме того, тела могут быть созданы из фундаментальных примитивов или определены с помощью данных MATLAB. Далее к кинематическим парам прикладывается сила/момент или профиль движения, после чего система готова к моделированию.

## Simscape Fluids

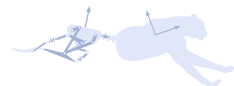
**Simscape Fluids**<sup>6</sup> предоставляет библиотеки компонентов для моделирования и имитации жидкости. Данные компоненты могут быть использованы для моделирования приводов экскаваторов, систем охлаждения двигателей, сетей водоснабжения и других приложений. Simscape Fluids [7] включает в себя множество типов компонентов: модели гидравлических насосов, клапанов, приводов, трубопроводов и теплообменников и многое другое.

---

<sup>4</sup><https://www.mathworks.com/videos/simscape-driveline-overview-117981.html>

<sup>5</sup><https://www.mathworks.com/videos/simscape-multibody-overview-117986.html>

<sup>6</sup><https://www.mathworks.com/videos/simscape-fluids-overview-117985.html>



## 1.2 Интерфейс

Все примеры имитационного моделирования в данном методическом пособии реализованы с помощью пакета MATLAB Simscape в среде Simulink. Методическое пособие не преследует цели рассказать о всех функциях и возможностях MATLAB и Simscape. В данном разделе будут даны краткие комментарии к интерфейсу MATLAB и Simulink. На рисунке 1.3 представлен стартовый интерфейс MATLAB 2018b.

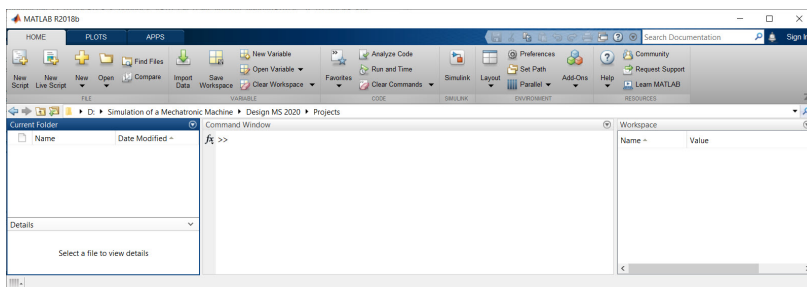
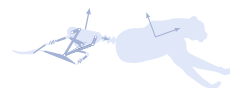


Рисунок 1.3. Стартовый интерфейс MATLAB 2018b

- *Current Folder* – текущая папка, в которой находятся активные файлы, папки, скрипты, модели;
- *Command Window* – поле для введения команд;
- *Workspace* – поле, где отображаются параметры и переменные системы.

Для входа в Simulink необходимо ввести в *Command Window* команду `smnew`. Появится окно с заранее установленными блоками, необходимыми для построения имитационной модели:

- *Solver Configuration* – определяет параметры решателя, используемые для моделирования.
- *World Frame* – обеспечивает доступ к «мировой» уникальной, неподвижной, ортогональной, правосторонней системе координат, являющейся основой в модели.



- *Mechanism Configuration* – устанавливает механические и имитационные параметры, которые применяются ко всей модели.
- *Solid* – представляет собой твёрдое тело, объединяющее геометрию, инерцию, массу, графическую составляющую и жёстко закреплённые системы координат в единое целое. Порт *R* – система координат в геометрическом центре.
- *Rigid Transform* – однородная матрица преобразования между двумя системам координат. Две составляющие независимо определяют поступательную и вращательную части преобразования. Порты *B* и *F* – это порты систем координат, которые представляют базовую «*base*» и последующую «*follower*» системы координат соответственно.
- *Simulink-PS Converter* – преобразователь сигналов в физическое измерение;
- *PS-Simulink Converter* – преобразователь физического измерения в сигнал;
- *Scope* – визуализация сигналов;
- *Open Simscape Multibody Library* – ярлык библиотеки Simscape.

### 1.3 Модель маятника

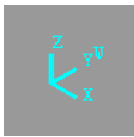

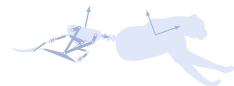


Рисунок 1.4.  
Мировая система координат

Если запустить модель (нажать *Run*, ), то в окне *Mechanics Explorer* будет видна только мировая система координат, если выбрать *World Frame* в списке слева (Рисунок 1.4). Отметим, что вертикальной осью является ось *Z*, горизонтальной ось – *X*, а перпендикулярно плоскости монитора сориентирована ось *Y*. Соберём самую простую модель физической системы – вращательный маятник, состоящий из одного подвижного звена, стойки и вращательного сочленения. Схема, отображающая подвижное звено и необходимые системы координат, изображена на рисунке 1.5.





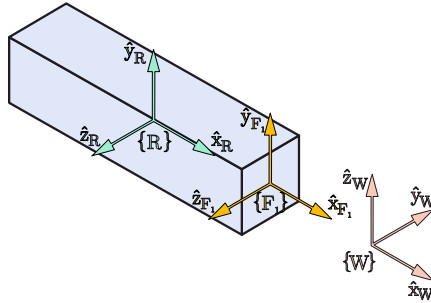


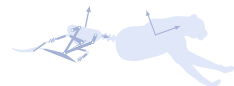
Рисунок 1.5. Схема моделируемого звена с системами координат мира  $\{W\}$ , центра масс звена  $\{R\}$  и созданного бокового порта звена  $\{F_1\}$

Если подключить *Solid* портом R к *World Frame* напрямую, твёрдое тело будет отображаться в *Mechanics Explorer*, однако присоединённое тело не будет двигаться. Соберём схему, изображённую на рисунке 1.6. Через *Rigid Transform* и *Revolute Joint* присоединим *Solid* к *World Frame*. Определим размеры звена, вписав  $[10\ 1\ 1]$  в графу *Dimensions* в блоке *Solid*, и выберем систему измерения – сантиметры, см. Это означает, что тело будет длиной 10 см вдоль оси X и 1 см вдоль Y и Z. Однако, даже если сейчас запустить модель, то вращения не произойдёт.

***Revolute Joint* работает только вокруг оси Z!**

*Revolute Joint* работает только вокруг оси Z, а так как вектор гравитации тоже направлен вдоль оси Z, то на тело не действует ни одной возмущающей силы. Вращение маятника будет иметь место, если:

- Развернуть всю сцену на  $90^\circ$  относительно оси X, используя *Rigid Transform* блок. В нем необходимо выбрать *Rotation* → *Method: Standard Axis* → *Axis: +X* → *Angle: 90 deg*. На рисунке 1.5  $\{F_1\}$  изображает систему координат тела, которая совмещена с системой координат, созданной блоком *Rigid Transform*, где видно, что вращение вокруг оси  $\hat{z}_{F1}$  приведёт к движению тела в плоскости  $\hat{x}_{F1}\hat{y}_{F1}$ .



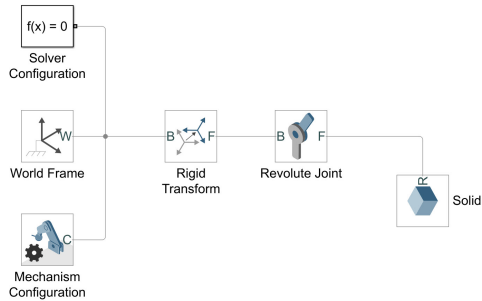


Рисунок 1.6. Модель вращательного маятника

- Создать дополнительную систему координат на теле *Solid* сборки  $\{F_1\}$  на рисунке 1.5). Для этого зайти в *Solid* → *Frames* → *New Frame* и нажать → выбрать плоскость, в которой необходимо создать систему координат → *Based on Geometric Feature* → *Use Selected Feature* → *Save*. На рисунке 1.7 изображён интерфейс создания новой системы координат.

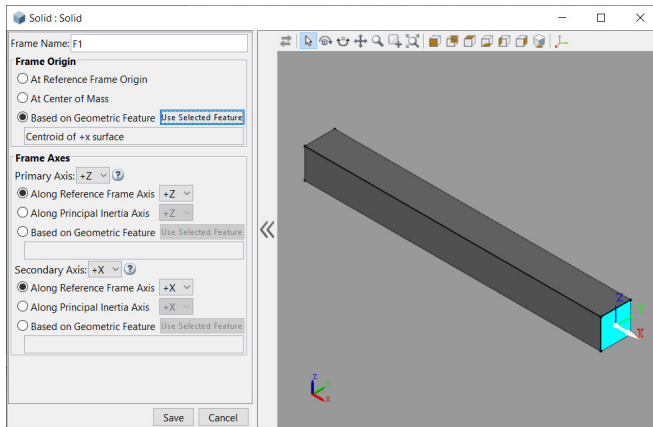
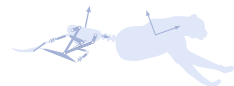


Рисунок 1.7. Создание новой системы координат



После этого *Solid* необходимо присоединить к *Revolute Joint*, используя только что созданный порт *F1*. В результате получим имитационную модель маятника с одним вращающимся звеном, которая может быть использована для проведения кинематического и динамического анализов.

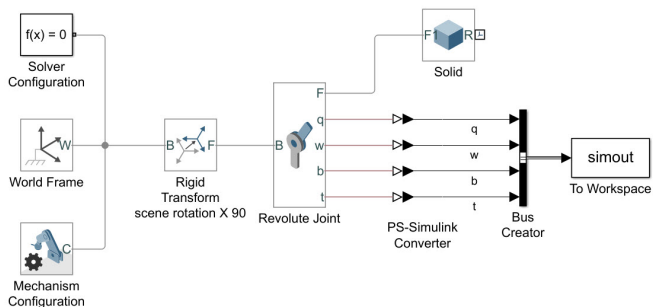


Рисунок 1.8. Рабочая модель вращательного маятника

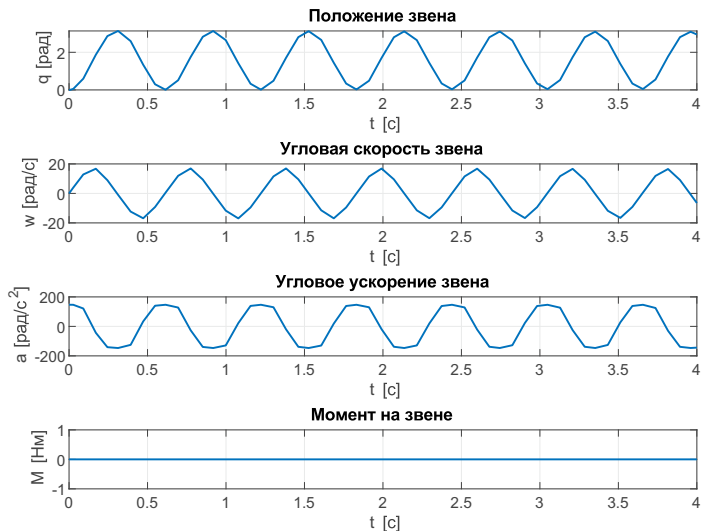
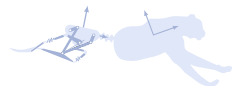


Рисунок 1.9. Графики моделирования вращательного маятника



Зайдя в *Revolute Joint* и поставив в графе *Sensing* галочки напротив *Position, Velocity, Acceleration, Actuator Torque*, можно получить графики изменения углового положения, угловой скорости, углового ускорения, а также момента, который должен быть приложен к звену для получения промоделированного движения. На рисунке 1.8 изображена модель маятника. Через блоки *PS-Simulink Converter* физические измерения с прописанными единицами измерения преобразуются в сигналы Simulink и собираются с помощью блока *Bus Creator* в один выходной сигнал, уходящий на блок *To Workspace*, данные с которого передаются в *Workspace* MATLAB, благодаря чему можно построить графики изменения изучаемых параметров.

На рисунке 1.9 изображены результаты моделирования вращательного маятника. Представлено, как изменяется угловое положение маятника под действием силы тяжести. Т.к. в кинематической паре не прописано трение колебание звена не угасает, что также видно на графиках угловой скорости  $\omega$  и углового ускорения  $a$ . Управляющее воздействие на звене  $M$  равно нулю, т.к. маятник движется только из-за силы тяжести.

Скрипт для вывода графиков представлен ниже. Для создания нового окна графика необходимо использовать *figure*; для вывода нескольких графиков на одном окне – *subplot(3,1,1)*, где первое число – количество строк, второе число – количество столбцов (в данном примере 3 графика в 3 строки и в 1 столбик), третье – порядковый номер графика; *plot(x, y)* – построение графика, на первом месте указывается значение по оси *ox*, на втором месте – *oy*, *linewidth* – настройка толщины линии графика, *1* – толщина линии; *title, xlabel, ylabel* – используются для подписи названия графика, осей *ox* и *oy* соответственно; *grid on* – отображение сетки на графике.

```
figure;
subplot(3,1,1);
plot(simout.q.time, simout.q.data, 'linewidth',1);
title('Link1position', 'fontname','CMU_Serif');
grid on;
xlabel('\it1t1[s]', 'fontname','CMU_Serif');
```



```

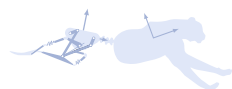
ylabel('\it{q}\_ [rad]', 'fontname', 'CMU_Serif');

subplot(3,1,2);
plot(simout.w.time, simout.w.data, 'linewidth',1);
title('Angular\_velocity', 'fontname', 'CMU_Serif');
grid on;
xlabel('\it{t}\_ [s]', 'fontname', 'CMU_Serif');
ylabel('\it{w}\_ [rad/s]', 'fontname', 'CMU_Serif');

subplot(3,1,3);
plot(simout.b.time, simout.b.data, 'linewidth',1);
title('Angular\_acceleration', 'fontname', 'CMU_Serif');
grid on;
xlabel('\it{t}\_ [s]', 'fontname', 'CMU_Serif');
ylabel('\it{a}\_ [rad/s^2]', 'fontname', 'CMU_Serif');

```

Подробнее о построении графиков рассказывается в справочной документации MATLAB.




## 1.4 Управление однозвенным маятником

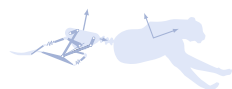
Рассмотрим, как описанное выше звено маятника может быть приведено в движение. Есть два варианта способа задания движения: подать на вход блока, моделирующего звено маятника, желаемое значение его положения как функции времени, либо непосредственно задающие (управляющие) силы или моменты, в зависимости от того, с призматическим или цилиндрическим сочленением мы имеем дело.

Для выбора типа управления необходимо зайти в *Revolute Joint* и в графе *Actuation* выбрать один из двух вариантов конфигурации:

- *Torque* → *Automatically Computed*, *Motion* → *Provided by Input* для подачи на вход положения;
- *Torque* → *Provided by Input*, *Motion* → *Automatically Computed* для подачи на вход момента (либо силы для призматической кинематической пары).

Рассмотрим первый способ, при котором на вход подаётся зависимость между занимаемым положением и временем. На рисунке 1.10 изображена схема подключения источника сигнала *Ramp* к *Revolute Joint* через *Simulink-PS Converter*, в котором прописано, что системой измерения является положение в радианах *rad*, и настроена фильтрация сигнала с фильтром второй степени *Second-order filtering*. В качестве сигнала может быть использован любой источник сигнала, который может быть найден в библиотеке ( → *Simulink* → *Sources*). В таком случае исполнения регулятор движения реализован в блоке кинематической пары, и значение момента можно снять с её выхода.

Однако характер переходного процесса может не удовлетворять требуемым характеристикам. Реализовать регулятор можно самим, используя второй тип управления, в котором на вход кинематической пары подаётся момент.



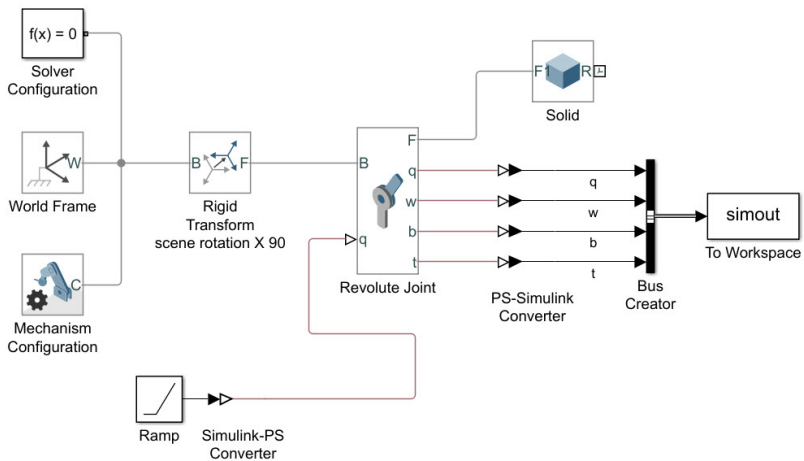


Рисунок 1.10. Управление по сигналу положения

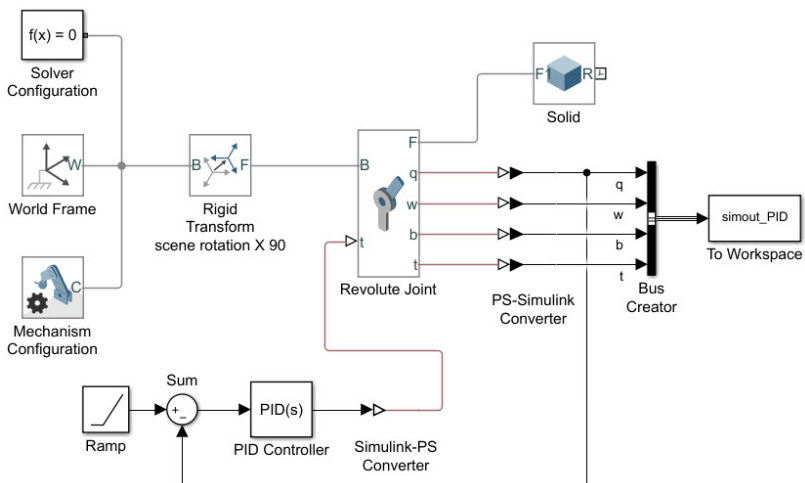
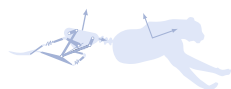


Рисунок 1.11. Управление по сигналу момента



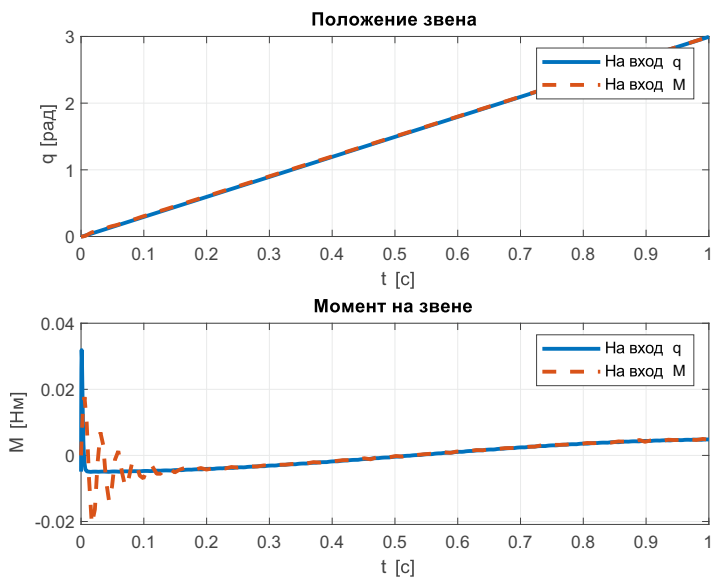
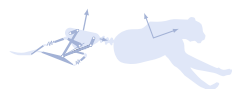


Рисунок 1.12. Графики положения звена и приложенного момента с управлением по положению  $q$  и по моменту  $M$

На рисунке 1.11 изображена схема подключения ПИД регулятора для управления звеном по ошибке позиционирования. На рисунке 1.12 изображены переходные процессы изменения положения звена и приложенного к нему управляющего момента. В случае управления по моменту был использован ПД регулятор с коэффициентами  $\Pi=1$ ,  $D=0,01$ . На графике видно перерегулирование, которое может быть ликвидировано, если увеличить  $D$  составляющую. В случае управления по положению параметры регулятора по умолчанию могут не удовлетворять требуемым показателям качества, например на рисунке 1.12 сплошной синей линией отображён график переходного процесса приложенного момента, которому свойственно резкое изменение, которое вполне может быть не отработано реальным двигателем.





## Глава 2

# Моделирование многозвенных механизмов

Рассмотрим построение имитационной модели четырехзвенного шарнирно-рычажного лямбда-механизма Пафнутия Чебышёва, преобразующего вращательное движение кривошипа в приближенное к прямолинейному движение выходной точки с приближённо постоянной скоростью. Проанализируем структуру и кинематику механизма аналитически, создадим имитационную модель устройства и проанализируем динамические параметры, в том числе значения сил реакций в кинематических парах.

Механизм Чебышёва был изобретён в XIX веке русским математиком Пафнутием Чебышёвым (рисунок 2.1), проводившим исследования теоретических проблем кинематических механизмов. Одной из таких проблем была проблема преобразования вращательного движения в приближённое к прямолинейному движению.

Механизм Чебышёва выглядит как греческая буква лямбда, поэтому также известен как лямбда-механизм. На основании описанного механизма Чебышёв изготовил первый в мире шагающий механизм, который пользовался большим успехом на Всемирной выставке в Париже в 1878 году. Стопоход П. Л. Чебышёва представляет собой четыре объединённых в две пары лямбда-механизмов Чебышёва, смонтированных на одном общем основании.

## 2.1 Структурный анализ

Задачей структурного анализа механизма является определение  $n$  – количества подвижных звеньев,  $P$  – количества кинематических пар, входящих в его состав, а также нахождение  $W$  – степени подвижности механизма. В задачу структурного анализа входит также последовательное разделение механизма на структурные группы.

Такая структурная декомпозиция механизма значительно упрощает его геометрическое, кинематическое и динамическое исследование, поскольку структурные группы, как правило, описываются независимыми системами уравнений небольшого порядка [8].

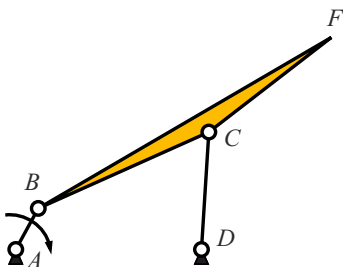


Рисунок 2.1. Лямбда механизм Чебышёва

Связи, накладываемые на относительные движения звеньев, могут быть определены при помощи структурной схемы – схемы механизма, выполненной с учётом условных обозначений кинематических пар и звеньев. Некоторые из связей или подвижностей могут быть выявлены только на кинематической схеме механизма, где учитываются геометрические размеры звеньев. На рисунке 2.2 изображён разбитый на группы Ассур лямбда-механизм Чебышёва, который состоит из 1 входного звена и диады Сильвестра. Механизм принадлежит к II классу, так как наивысший класс относится к структурной группе, состоящий из 2 подвижных звеньев и 3 кинематических пар.



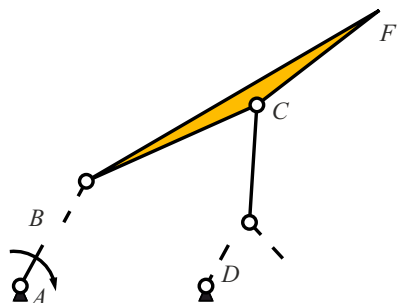


Рисунок 2.2. Разбитый на группы Ассура лямбда механизм Чебышева

Рассчитаем степень свободы  $W$  плоского механизма с помощью формулы Сомова-Малышева:

$$W = 3n - 2P_5 - P_4 = 3 \cdot 3 - 2 \cdot 4 = 1 \quad (2.1)$$

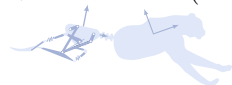
где  $n$  – число подвижных звеньев ( $BCF$  и  $CD$ ),  $P_5$  – число кинематических пар пятого класса ( $A, B, C$  и  $D$ ),  $P_4$  – число кинематических пар четвертого класса.

## 2.2 Кинематический анализ

Кинематические свойства проектируемого механизма определяются его геометро-кинематическими характеристиками, связывающими параметры движения на входе механизма и на выходе из него. Основные геометро-кинематические характеристики механизмов: функция положения, определяющая связь координат выходного и входного звеньев, и кинематическая передаточная функция, являющаяся первой производной от функции положения [1].

Составим уравнение замкнутого векторного многоугольника, согласно рисунку 2.3:

$$\vec{AF} = \vec{AB} + \vec{BC} + \vec{CF} = \vec{AD} + \vec{DC} + \vec{CF}. \quad (2.2)$$



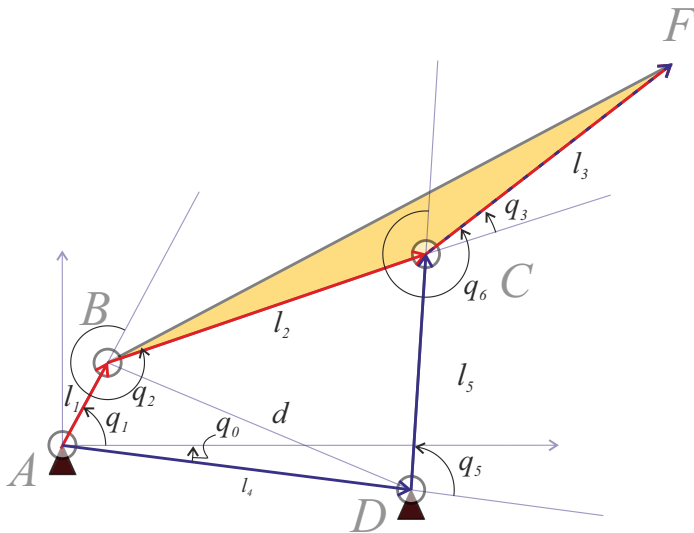


Рисунок 2.3. Кинематический анализ лямбда механизма Чебышёва

Спроецируем (2.2) на оси  $ox$  и  $oy$ :

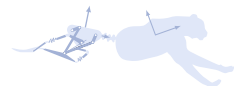
$$\begin{aligned}
 l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) + l_3 \cos(q_1 + q_2 + q_3) &= \\
 = l_4 \cos(q_4) + l_5 \cos(q_4 + q_5) + l_3 \cos(q_4 + q_5 + q_6) &
 \end{aligned} \quad (2.3)$$

$$\begin{aligned}
 l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) + l_3 \sin(q_1 + q_2 + q_3) &= \\
 = l_4 \sin(q_4) + l_5 \sin(q_4 + q_5) + l_3 \sin(q_4 + q_5 + q_6) &
 \end{aligned} \quad (2.4)$$

Траектория точки  $F$ :

$$\begin{bmatrix} x_F \\ y_F \end{bmatrix} = \begin{bmatrix} l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) + l_3 \cos(q_1 + q_2 + q_3) \\ l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) + l_3 \sin(q_1 + q_2 + q_3) \end{bmatrix} \quad (2.5)$$

где  $q_1$  – входной угол,  $q_3 = 20^\circ$  – постоянный угол, обусловленный формой детали  $BCF$ .



Выразим формулу для  $q_2$ :

$$q_2 = \pi + \beta_1 + \beta_2, \quad (2.6)$$

где  $\beta_1$  – угол  $\angle ABD$ , а  $\beta_2$  – угол  $\angle DBC$ . Воспользуемся теоремами косинуса и синуса, чтобы найти значения обеих частей угла  $\beta$ .

С помощью теоремы косинусов найдём диагональ  $d$  четырехугольника  $ABCD$ , для треугольника  $\triangle ABD$  справедливо:

$$d = \sqrt{l_1^2 + l_4^2 - 2l_1l_4 \cos(q_0 + q_1)}. \quad (2.7)$$

Чтобы найти  $\beta_1$ , воспользуемся теоремой синусов, однако, так как  $\arcsin$  работает только для острых углов, сначала необходимо выразить значение угла  $\angle BDA = \delta_1$ :

$$\frac{\sin(q_0 + q_1)}{d} = \frac{\sin \delta_1}{l_1}, \quad (2.8)$$

откуда выразим значение для угла  $\angle BDA = \delta_1$ :

$$\delta_1 = \arcsin \left( \frac{l_1}{d} \sin(q_0 + q_1) \right) \quad (2.9)$$

Получим значение для угла  $\beta_1$ , зная все остальные углы в  $\triangle ABD$

$$\beta_1 = \pi - \delta_1 - q_0 - q_1. \quad (2.10)$$

Рассмотрим  $\triangle DBC$  и воспользуемся теоремой косинусов:

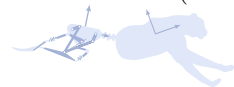
$$l_5^2 = d^2 + l_2^2 - 2dl_2 \cos \beta_2, \quad (2.11)$$

откуда выразим значение для  $\beta_2$ :

$$\beta_2 = \arccos \left( \frac{d^2 + l_2^2 - l_5^2}{2dl_2} \right). \quad (2.12)$$

Итоговое выражение для  $q_2$  примет вид:

$$q_2 = 2\pi - \arcsin \left( \frac{l_1}{d} \sin(q_0 + q_1) \right) + \arccos \left( \frac{d^2 + l_2^2 - l_5^2}{2dl_2} \right) - q_0 - q_1, \quad (2.13)$$



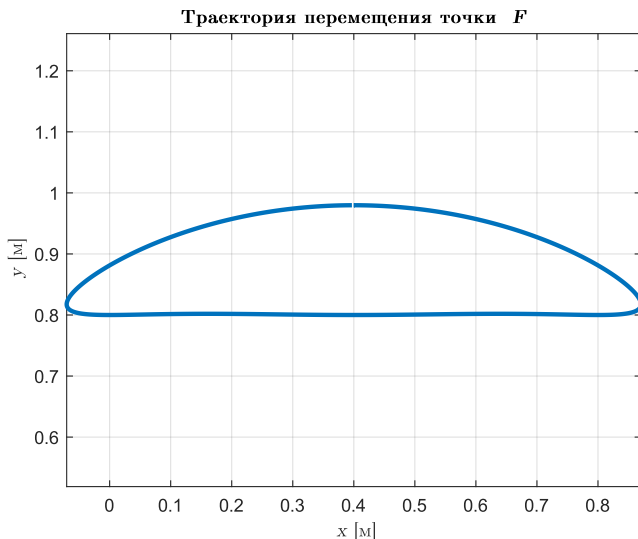


Рисунок 2.4. Траектория перемещения точки  $F$

Скорость точки  $F$  получим, продифференцировав (2.14):

$$\begin{aligned}
 \begin{bmatrix} \dot{x}_F \\ \dot{y}_F \end{bmatrix} &= \begin{bmatrix} -l_1 \dot{q}_1 \sin q_1 - l_2 (\dot{q}_1 + \dot{q}_2) \sin q_{12} - l_3 (\dot{q}_1 + \dot{q}_2 + \dot{q}_3) \sin q_{123} \\ l_1 \dot{q}_1 \cos q_1 + l_2 (\dot{q}_1 + \dot{q}_2) \cos q_{12} + l_3 (\dot{q}_1 + \dot{q}_2 + \dot{q}_3) \cos q_{123} \end{bmatrix} = \\
 &= \begin{bmatrix} -l_1 \sin q_1 - l_2 \sin q_{12} - l_3 \sin q_{123} & -l_2 \sin q_{12} - l_3 \sin q_{123} & -l_3 \sin q_{123} \\ l_1 \cos q_1 + l_2 \cos q_{12} + l_3 \cos q_{123} & l_2 \cos q_{12} + l_3 \cos q_{123} & l_3 \cos q_{123} \end{bmatrix} \cdot \\
 &\quad \cdot \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = J \cdot \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \tag{2.14}
 \end{aligned}$$

где  $q_{12} = q_1 + q_2$ ,  $q_{123} = q_1 + q_2 + q_3$ ,  $J$  – матрица Якоби.

Для получения траектории лямбда-механизма Чебышёва должно выполняться следующее условие:

$$AB = a; AD = 2a; q_0 = 0; BC = 2, 5a; CD = CF = 2, 5a.$$

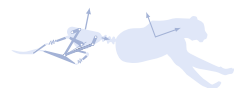
Для рассматриваемого примера допустим, что  $a = 0, 2$  м.



Ниже представлен скрипт MATLAB для построения графика траектории точки  $F$ .

```
n=2*pi;% 'конечное_положение_входного_звена'
theta=[0.0:0.01:n];% 'массив_положений_входного_звена'
a=0.2;%m 'длина_входного_звена'
AB=a; AD=2*a; BC=2.5*a; CD=2.5*a; CF=2.5*a; % 'длины_
всех_звеньев'
d=sqrt(AB^2+AD^2-2*AB*AD*cos(theta)); % диагональ
delta_1=asin(AB*sin(theta)./d); % 'дельта_1'
beta_1=pi-theta-delta_1; % 'бета_1'
beta_2=acos((d.^2+BC^2-CD^2)./(2*d*BC)); % 'бета_2'
beta=beta_1+beta_2; % 'бета'
x_F=AB*cos(theta)+2*BC*cos(theta+beta-pi);
y_F=AB*sin(theta)+2*BC*sin(theta+beta-pi);
plot(x_F, y_F, 'linewidth',2); % 'построение_графиков'
xlabel('\it_x_м[]', 'fontname','CMU_Serif');
ylabel('\it_y_м[]', 'fontname','CMU_Serif');
title('Траектория_перемещения_точки_{\it_F}', 'fontname','
CMU_Serif');
grid on; % 'сетка'
axis equal; % 'масштаб'
```

График траектории перемещения точки  $F$  изображён на рисунке 2.4. Как можно видеть, построение траектории аналитическим способом является нетривиальной задачей. Альтернативным способом получения аналогичных графиков является построение имитационной модели, что представлено в следующем разделе.



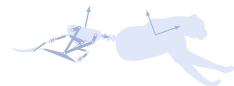
## 2.3 Имитационное моделирование

Имитационное моделирование – метод исследования, при котором изучаемая робототехническая система заменяется моделью, с достаточной точностью описывающей реальную систему. Имитационное моделирование является удобным инструментом для проведения виртуальных экспериментов с целью проверки, оценки и поддержки конструктивных решений, изучения взаимодействия между составными элементами мехатронной системы и с объектами окружающей среды, а также для анализа кинематических и динамических характеристик, настройки алгоритмов управления и получения прочей информации о проектируемой системе до построения реального устройства/опытного образца [9].

На рисунке 2.5 изображена блок-схема механизма в окне Simulink с блоками Simscape. В качестве управляющего воздействия механизма используется модель двигателя постоянного тока с поданным напряжением в 6 В — электрическая машина постоянного тока, преобразующая электрическую энергию постоянного тока в механическую энергию, подробное описание моделирования электродвигателей приведено в Главе 4.

Механизм Чебышева относится к механизмам с замкнутой кинематикой. Сборка замкнутых цепей механизма является нетривиальной задачей, так как необходимо следить за правильностью выборка систем координат соединяемых блоков. Далее представлен порядок действий для сборки подобного механизма:

1. К входному звену, моделирование которого представлено в разделе 1.4, присоединим звено  $BC$  через *Revolute Joint* кинематической пары  $B$ . Отметим, что у входного звена  $AB$  и у присоединяемого звена  $BC$  должно быть две дополнительные системы координат по бокам тела. На рисунке 2.6, *б* представлен блок звена  $BC$  с двумя портами  $F1$  и  $F2$ , чьи  $Z$  оси (отмечены синим) сонаправлены с  $Z$  осью базовой системы координат, которая представлена на картинке сбоку. В результате получим двойной маятник.
2. Аналогично присоединим к  $BC$  третье звено  $CD$  через *Revolute Joint* кинематической пары  $C$ . В результате получим тройной маятник.





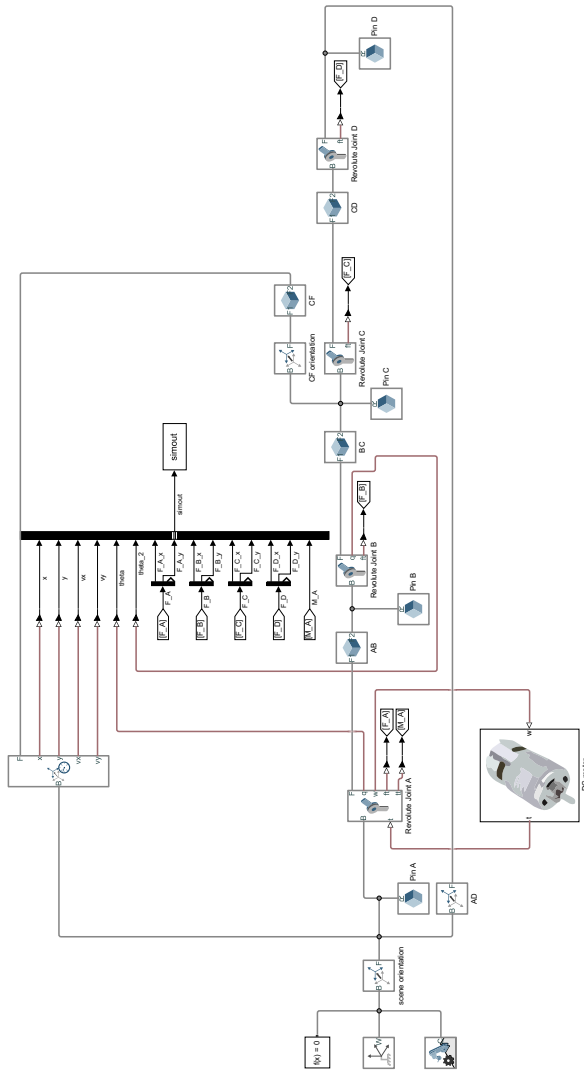


Рисунок 2.5. Блок схема имитационной модели в Simulink



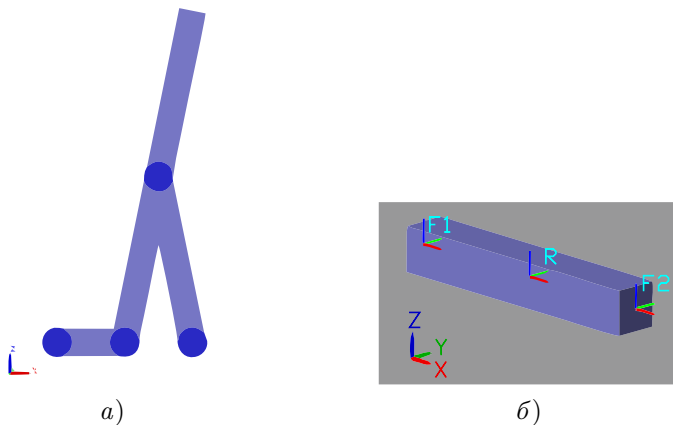
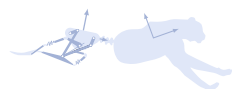


Рисунок 2.6. Визуализация имитационной модели лямбда механизма Чебышёва в окне Mechanics Explorer

3. Для завершения сборки необходимо создать стойку  $D$ , используя *Rigid Transform*. Подсоединим *Rigid Transform* (на рисунке 2.5 отмечено как  $AD$ ) портом  $B$  к *pin A*, а порт  $F$  через *Revolute Joint* кинематической пары  $D$  к свободному порту звена  $CD$ . В *Rigid Transform* настроим смещение по оси  $X$  на величину  $AD$ .
4.  $CF$  жёстко присоединим к  $BC$  без блоков *Revolute Joint*. Можно использовать *Rigid Transform* для настройки углового смещения.

Отметим, что важно осознать работу блоков *Rigid Transform*, понимать принцип проставления дополнительных систем координат в блоке *Solid* и иметь пространственное представление о сборке системы. Понимание этого концепта требует инвестиций усилий и времени. Попробуйте собрать несколько механических систем в качестве упражнений.



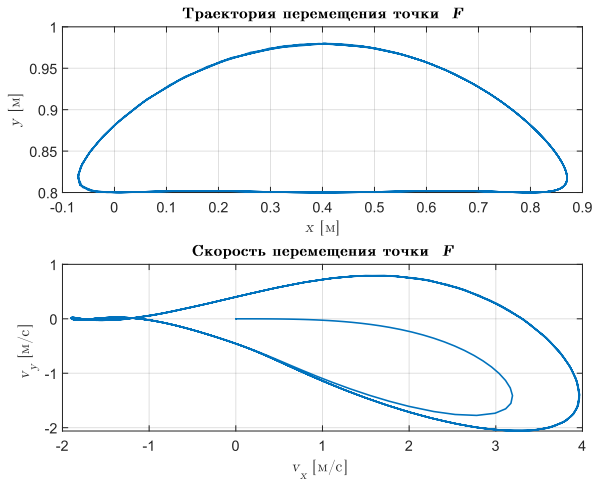


Рисунок 2.7. Графики. Кинематика

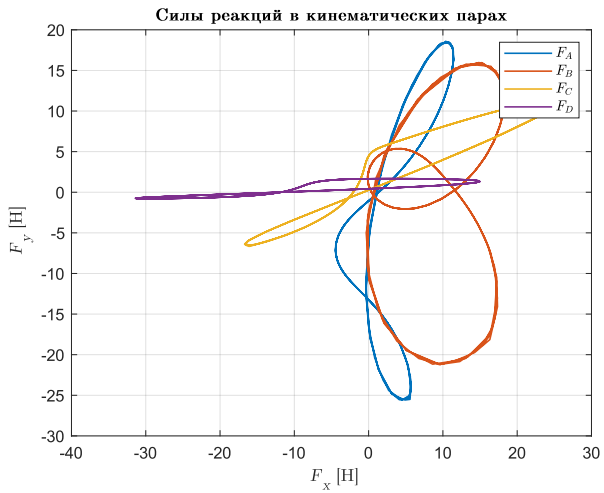


Рисунок 2.8. Графики сил реакций в кинематических парах



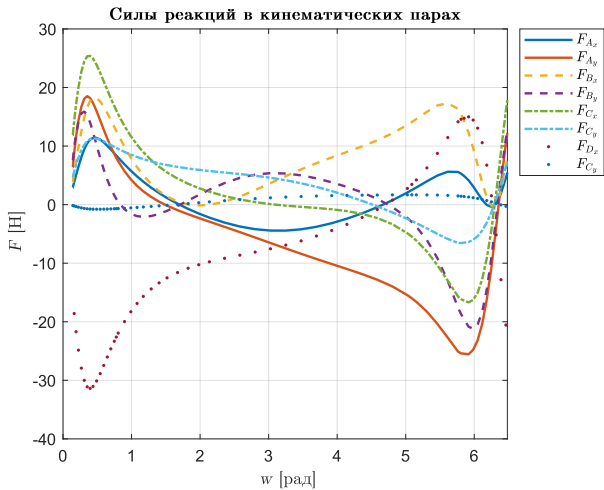


Рисунок 2.9. Графики сил реакций в кинематических парах

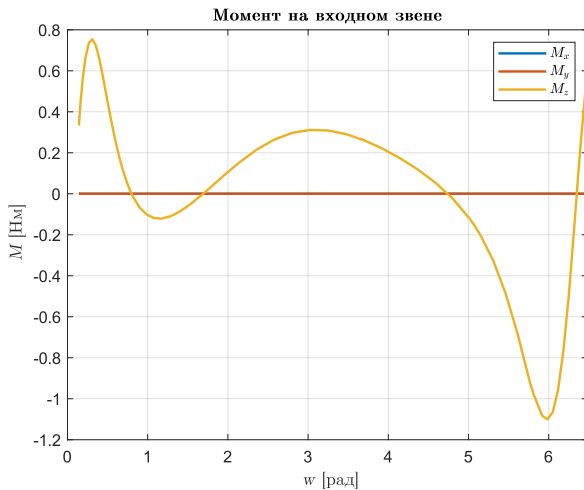
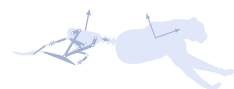


Рисунок 2.10. График момента на входном звене



Имитационная модель необходима для количественного понимания работы системы. Не следует оценивать работу системы только по анимации. Графики – более наглядный инструмент. Используя блоки *Transform Sensor*, можно снять кинематические показатели: перемещение, скорость, ускорение как прямолинейного движения, так и вращательного. Силовые показатели, а именно силы и моменты реакций, а также значения управляющих моментов и сил необходимо снимать в блоках кинематических пар в графе *Sensing* и *Composite Force/Torque Sensing*.

На рисунке 2.7 изображены графики изменения траектории и скорости перемещения точки  $F$ , построенные с помощью блока *Transform Sensor*. Рисунок 2.8 изображает зависимость сил реакций  $F_x$  от  $F_y$  во всех кинематических парах. На рисунке 2.9 представлены значения сил реакций во всех кинематических парах в зависимости от положения входного звена в радианах. Момент на входном звене в зависимости от его положения представлен на рисунке 2.10. На выход подаётся сразу три сигнала, а именно проекции на три оси. Моменты относительно осей  $x$  и  $y$  равны нулю, так как вращение осуществляется только относительно оси  $z$ .



# Глава 3

## Моделирование контактного взаимодействия

В данной главе рассмотрим моделирование контактного взаимодействия в MATLAB. Материал основан на курсе [MATLAB and Simulink Robotics Arena](#) [10].

### Зачем моделировать контакт?

Контактное взаимодействие происходит в мехатронных системах повсеместно. Несколько примеров:

- При проектировании мобильного робота на колёсной базе – постоянный контакт колес с поверхностью дороги;
- При создании шагающих и галопирующих роботов – дискретный контакт ног с подстилающей поверхностью;
- При разработке захватных устройств – взаимодействие пальцев устройства с объектами манипулирования;
- При синтезе и анализе механизмов – взаимодействие звеньев между собой.

### 3.1 Теоретические основы контактного взаимодействия

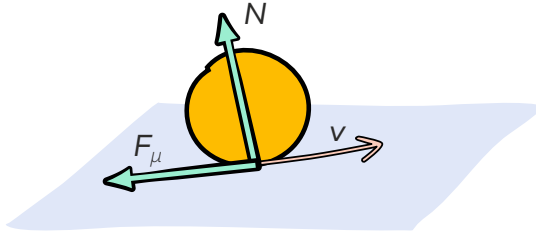
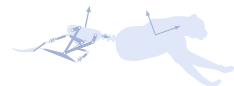


Рисунок 3.1. Диаграмма контактного взаимодействия двух тел

На рисунке 3.1 схематично обозначено контактное взаимодействие двух тел: сферическое подвижное тело сталкивается с фиксированной поверхностью пола. При соударении двух тел между ними возникает сила взаимодействия  $N$ , действующая по нормали, и сила трения  $F_\mu$ , направленная против направления движения  $v$ . Если же оба тела свободны, то дополнительно возникают моменты сил, приводящие к вращению тел.

Существуют несколько способов моделирования нормальной силы взаимодействия. Самым распространенным является метод штрафов (*penalty based approach*, [11]). Данный метод, например, используется в физических игровых движках, таких как PhysX, Havok, Cry-Physics [12]. Тела, взаимодействуя, внедряются друг в друга, тем самым имитируя упругую деформацию. Данный метод является не самым точным, но он прост и понятен для реализации. Основным недостатком является необходимость подбора правильных параметров контакта, при которых порядок силы взаимодействия принимает адекватное значение. Альтернативным способом является моделирование динамического контакта с использованием метода множителей Лагранжа [13]. Этот метод позволяет точно описать взаимодействие тел без внедрения их друг в друга, но метод гораздо сложнее<sup>1</sup>.

<sup>1</sup>Метод множителей Лагранжа будет представлен в последующих версиях методического пособия



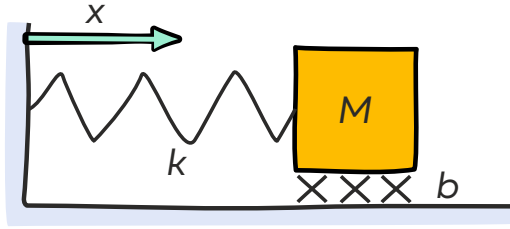


Рисунок 3.2. Масса на пружине

Рассмотрим подробнее метод штрафов на простом примере моделирования взаимодействия массы со стеной (рисунок 3.2). Масса  $M$  соединена со стеной посредством пружины с коэффициентом жёсткости  $k$  и испытывает трение скольжения о поверхность с коэффициентом трения  $b$ . У стены тело находится в равновесии. При растяжении пружины тело выводится из состояния равновесия. Запишем второй закон Ньютона для всей системы:

$$\sum F = F_S + F_v + F_n = Ma, \quad (3.1)$$

где  $F_v$  – сила трения,  $F_n$  – сила соударения со стеной,  $F_S$  – создаваемая пружиной сила, которая описывается законом Гука:

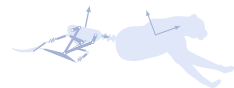
$$F_S = k\Delta x, \quad (3.2)$$

где  $\Delta x$  – деформация пружины.

Сила взаимодействия по нормали рассчитывается по линейной вязко-упругой модели:

$$F_n = k_c \delta + \beta_c \dot{x}, \quad (3.3)$$

$k_c$  и  $\beta_c$  – коэффициенты жёсткости [Н/м] и диссипации [Нс/м] контактного взаимодействия соответственно;  $\delta$  – внедрение тел [м];  $\dot{x}$  – скорость демпфирования [м/с].





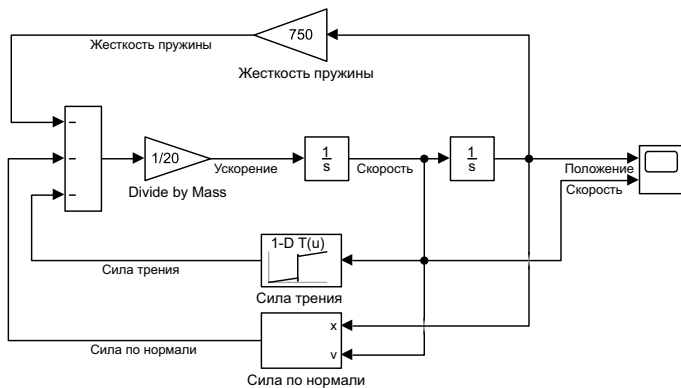


Рисунок 3.3. Блок-схема массы на пружине в Simulink

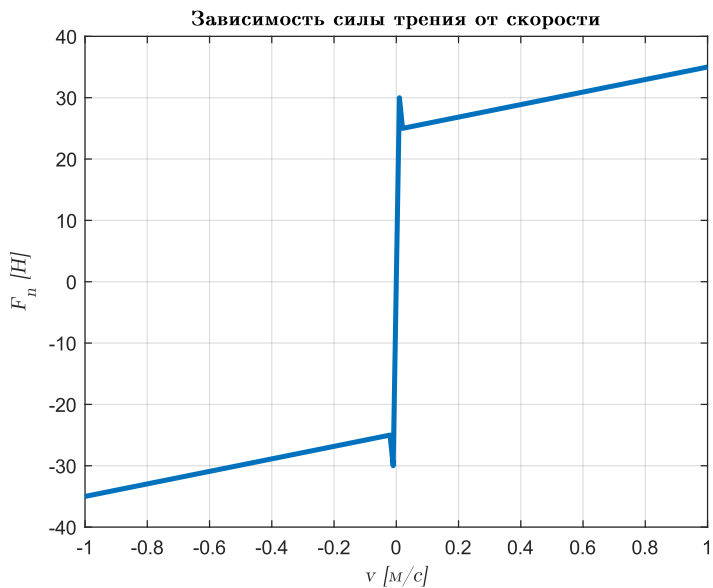
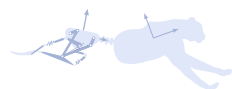


Рисунок 3.4. Зависимость силы трения от скорости



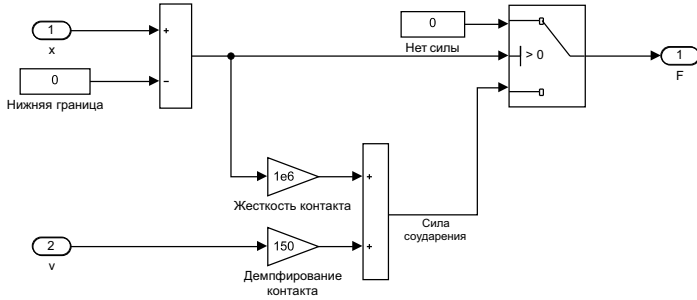


Рисунок 3.5. Моделирование силы по нормали в Simulink

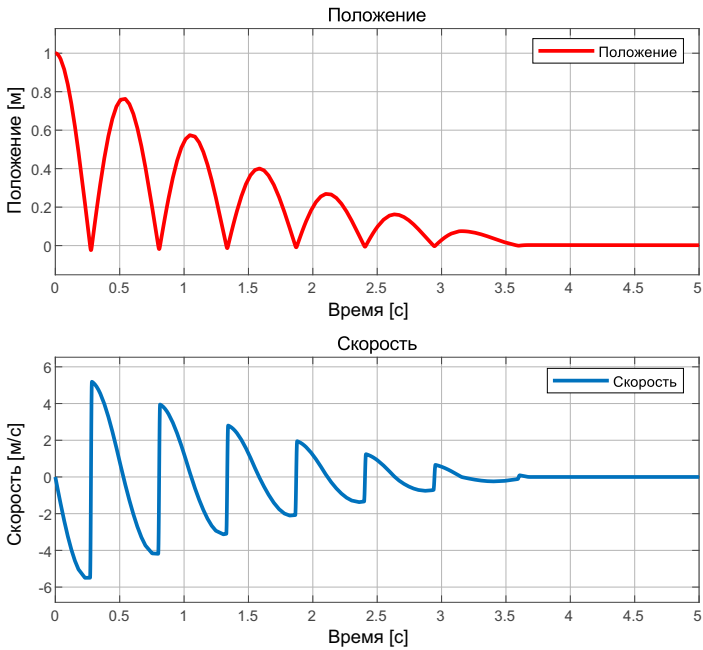
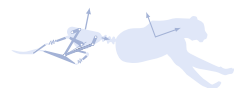


Рисунок 3.6. Результаты моделирования соударения тела о стенку



Рассмотрим имитационное моделирование данной системы, выполненной с помощью блоков Simulink, Simscape 1D и Simscape 3D. Модели доступны для скачивания на MATLAB File Exchange<sup>2</sup>.

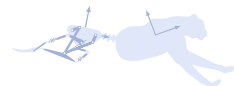
На рисунке 3.3 представлена блок-схема модели. В сумматоре суммируются все три составляющие из уравнения (3.1). Согласно закону Ньютона, сумма сил равна произведению массы и ускорение. Выход сумматора делим на массу, в результате получаем ускорение. Интегрируя ускорение, получим скорость. Интегрируя ещё раз получим изменение перемещения. Перемножив изменение перемещения на коэффициент жёсткости пружины  $k$ , получим создаваемую в пружине силу согласно формуле (3.2).

Сила трения зависит от нормальной силы и скорости взаимодействия. Зная нормальную силу и скорость взаимодействия, а также характеристики материалов обоих тел, можно восстановить силу трения. Зная скорость перемещения, можно восстановить силу трения с помощью блока *Look-up Table*. На рисунке 3.4 представлена зависимость силы трения скольжения от скорости взаимодействия. Отметим, когда скорость перемещения равна нулю, то сила трения покоя выше, чем при движении. Линейный рост силы трения при увеличении скорости перемещения обусловлен вязкостью взаимодействия.

На рисунке 3.5 представлен блок расчёта силы взаимодействия по нормали. Основу блока составляет блокочек *Switch*, на выходе которого рассчитывается нормальная реакция взаимодействия. Если второй сигнал, входящий в блок, удовлетворяет условию  $> 0$ , то выход равен значению сигнала, входящего в первый вход, в противном случае – в третий. Другими словами, пока положение  $\Delta x > 0$ , сила соударения со стеной равна  $F_n = 0$ , иначе  $F_n$  рассчитывается по формуле (3.3). Так как метод основан на внедрении одного тела в другое на небольшую величину  $\delta$ , то результаты моделирования неточные. На графике 3.6 видно, что положение  $x$  в момент соударения ниже 0, это значит что одно тело пересекает другое. Коэффициенты жёсткости  $k_c$  и диссипации  $\beta_c$  контактного взаимодействия подбираются эмпирически для получения адекватных и прогнозируемых результатов моделирования.

---

<sup>2</sup><https://www.mathworks.com/matlabcentral/fileexchange/64001-matlab-and-simulink-robotics-arena-introduction-to-contact-modeling>



## 3.2 Моделирование контакта в Simscape

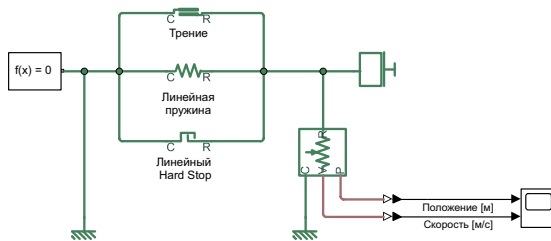


Рисунок 3.7. Моделирование системы тело-пружина в Simscape

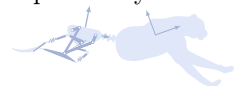
Аналогично представленной модели, в Simulink можно собрать ту же схему, используя блоки Simscape. На рисунке 3.7 представлена блок-схема рассмотренной системы, выполненная с помощью блоков Simscape:

- *Mass* – имитирует идеальную массу для прямолинейного движения, описывается следующей формулой:

$$F = m \frac{dv}{dt},$$

где  $F$  – сила инерции,  $m$  – масса,  $v$  – скорость,  $t$  – время. Блок оснащён одним механическим портом для прямолинейного движения. Положительное направление блока – от порта до опорной точки, т.е. сила инерции положительна, если масса ускоряется в положительном направлении.

- *Translational Friction* – моделирует трение скольжения между движущимися телами как функцию относительной скорости и принимается равной сумме компонентов Стрибека, Кулона и вязкости. Трение Стрибека  $F_S$  – это отрицательно наклонная кривая, имеющая место при низких скоростях. Кулоновское трение  $F_C$  приводит к постоянной силе при любой скорости. Вязкое трение  $F_V$  противостоит движению с силой, прямо пропорциональной относительной скорости. Сумма сил



трений Кулона и Стрибека в окрестности нулевой скорости часто называется трением покоя или *breakaway force*  $F_{brk}$ .

Трение аппроксимируется следующими уравнениями:

$$F = \sqrt{2\epsilon}(F_{brk} - F_C) \cdot \exp\left(-\left(\frac{v}{v_{brk}\sqrt{2}}\right)^2\right) \cdot \frac{v}{v_{brk}\sqrt{2}} + F_C \cdot \tanh\left(\frac{10v}{v_{brk}}\right) + bv, \quad (3.4)$$

где  $v = v_R - v_C$ ,  $v_{brk}$  – скорость выхода на значение кулоновского трения,  $v_R$ ,  $v_C$  – абсолютные значения скоростей для портов  $R$  и  $C$  соответственно,  $v$  – относительная скорость,  $b$  – коэффициент трения.

- *Translational Spring* – поступательный пружинный блок представляет собой идеальную механическую линейную пружину, описываемую следующими уравнениями:

$$F = Kx,$$

$$x = x_0 + x_R - x_C,$$

$$v = dx/dt,$$

где  $F$  – сила, генерируемая пружиной,  $K$  – коэффициент жёсткости,  $x$  – деформация пружины,  $x_0$  – начальная деформация,  $x_R$  и  $x_C$  – абсолютное перемещение портов  $R$  и  $C$  соответственно,  $v$  – относительная скорость,  $t$  – время.

- *Translational Hard Stop* – блок представляет собой двусторонний механический поступательный жёсткий упор, ограничивающий движение тела между верхним и нижним пределами. Предполагается, что ударное взаимодействие между ползуном и упорами является упругим. Упор представлен в виде пружины, которая входит в контакт с ползуном при уменьшении зазора и противостоит проникновению ползуна в упор с силой, линейно пропорциональной этому проникновению. Для учёта диссипации энергии и неупругих эффектов в качестве параметра блока вводится демпфирование, что позволяет учитывать потери энергии. Описывается формулой (3.3).



### 3.3 Моделирование контактного взаимодействия плоского робота с объектами окружающей среды

Для роботов, механизмы которых представляют собой совокупность абсолютно твёрдых звеньев и сочленений, в большинстве случаев достаточно сложно добиться стабильного характера силового взаимодействия с обеспечением требуемой силы и характера контакта, используя только регуляторы по положению и скорости.

В задачах, где необходимо силовое физическое взаимодействие между роботом и объектом производства, применяются методы управления по силе, в том числе с использованием дорогостоящих силомоментных датчиков.

Для динамического силового воздействия применяют методы управления по импедансу, впервые представленные в статьях Нэвила Хогана [14] как подход управления роботами-манипуляторами.

Механический импеданс – это отношение силы на выходе системы к движению на входе, аналогично электрическому импедансу, равному отношению напряжения на выходе к силе тока на входе.

При управлении по импедансу робот может быть представлен в виде пружины. Соответственно, сила взаимодействия робота с окружающей средой описывается как

$$F = -K(x - x_0) - bv, \quad (3.5)$$

где коэффициент жёсткости  $K$  определяет значение силы на выходе при отклонении от опорной траектории, коэффициент демпфирования  $b$  определяет значение силы на выходе для значения скорости на входе. При управлении импедансом робота мы контролируем силу сопротивления внешним возмущениям, т.е. силу взаимодействия с окружающей средой.



Рассмотрим пример моделирования динамического физического взаимодействия для робота с открытой кинематической цепью и двумя вращательными сочленениями, который изображён на рисунке 3.8, а.

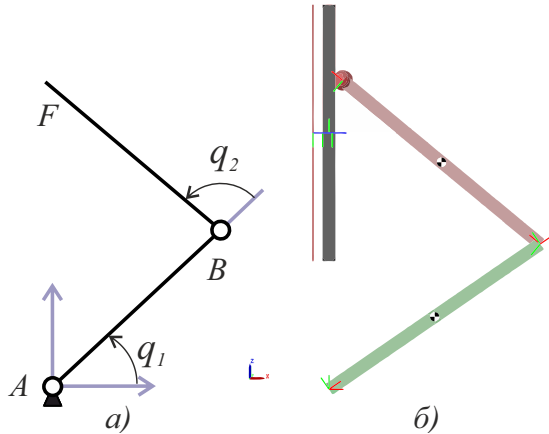


Рисунок 3.8. а) Схематичное изображение плоского 2R робота, б) имитационная модель

Решим прямую задачу кинематики и определим положение точки  $F$  в зависимости от положения звеньев:

$$x_F = AB \cos q_1 + BF \cos(q_1 + q_2),$$

$$y_F = AB \sin q_1 + BF \sin(q_1 + q_2)$$

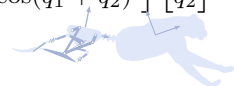
Продифференцируем данные выражения для получения линейной скорости:

$$\dot{x}_F = -AB\dot{q}_1 \sin q_1 - BF(\dot{q}_1 + \dot{q}_2) \sin(q_1 + q_2),$$

$$\dot{y}_F = AB\dot{q}_1 \cos q_1 + BF(\dot{q}_1 + \dot{q}_2) \cos(q_1 + q_2)$$

Перепишем в матричном виде:

$$\begin{bmatrix} \dot{x}_F \\ \dot{y}_F \end{bmatrix} = \begin{bmatrix} -AB \sin q_1 - BF \sin(q_1 + q_2) & -BF \sin(q_1 + q_2) \\ AB \cos q_1 + BF \cos(q_1 + q_2) & BF \cos(q_1 + q_2) \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$



В результате получим зависимость между вектором линейной скорости  $\mathbf{v}$  точки  $F$  и вектором угловых скоростей звеньев  $\dot{\mathbf{q}}$

$$\mathbf{v} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (3.6)$$

где Якобиан  $\mathbf{J}(\mathbf{q})$  равен

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} -AB \sin q_1 - BF \sin(q_1 + q_2) & -BF \sin(q_1 + q_2) \\ AB \cos q_1 + BF \cos(q_1 + q_2) & BF \cos(q_1 + q_2) \end{bmatrix}$$

Представим, что на точку  $F$ , движущуюся со скоростью  $\mathbf{v}$ , действует сила  $\mathbf{F}$ . Тогда в кинематических парах должны возникнуть противодействующие моменты сил с вектором  $\tau$ , приложенным к звеньям, которые перемещаются с вектором угловой скорости  $\dot{\mathbf{q}}$ . Рассчитаем мощность приложенной силы и приравняем её к развиваемой мощности электродвигателей в кинематических парах:

$$P = \mathbf{v}\mathbf{F} = \dot{\mathbf{q}}\tau \quad (3.7)$$

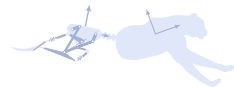
Принимая во внимание формулы (3.6) и (3.7), получим выражение для управляющих моментов:

$$\tau = \mathbf{J}^T(\mathbf{q})\mathbf{F}, \quad (3.8)$$

где силу  $\mathbf{F}$  рассчитаем по формуле (3.5). Подробнее управление по импедансу представлено в [15].

Имитационная модель изображена на рисунке 3.8, б. Блок-схема приведена на рисунке 3.9. Блоки *link\_1* и *link\_2* обозначают звенья  $AB$  и  $BF$  соответственно. Блок *Plane* служит внешним препятствием для робота. Точка  $F$  (*end effector*) соединяется блоком *Sphere to Plane* из библиотеки *Simscape Multibody Contact Forces Library*<sup>3</sup>. Данная библиотека содержит блоки моделирования контактного взаимодействия для моделирования многозвенных систем в *Simscape Multibody*. Представлены блоки для моделирования контакта в дву- и трёхмерном пространствах, как с трением, так и без него. Представлено более 45 примеров, включая моделирование соударения мяча о поверхность и более сложные системы, например, кулачковый механизм, робот с дифференциальным приводом и рука робота с захватом.

<sup>3</sup><https://www.mathworks.com/matlabcentral/fileexchange/47417-simscape-multibody-contact-forces-library>





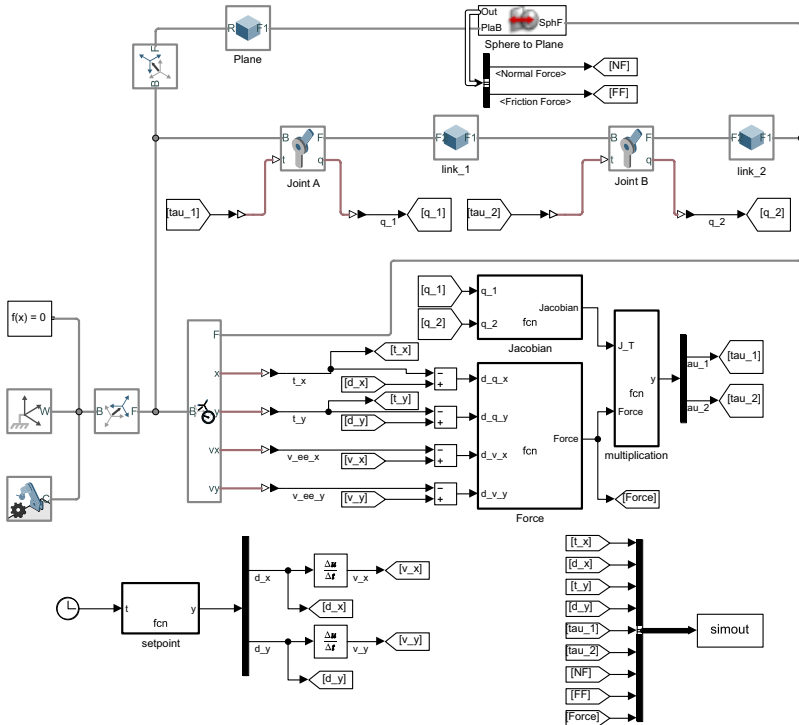
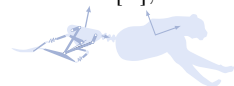



Рисунок 3.9. Блок схема модели 2R робота, испытывающего динамическое взаимодействие со стеной

Блок *Sphere to Plane* прикрепляет к порту  $F1$  блока  $link\_1$  виртуальную красную сферу (рисунок 3.8, б), которая при взаимодействии пересекает виртуальную плоскость, ассоциированную с портом  $F1$  блока *Plane*. В результате внедрение одного тела в другое происходит на уровне виртуальных объектов, ассоциированных с взаимодействующими телами, в то время как сами тела друг друга не пересекают. *Sphere to Plane* необходимо указать:

- *Sphere Radius* – радиус сферы в [м];
- *Plane Length x* – длина ответной плоскости по  $x$  в [м];

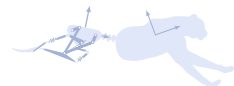


- *Plane Length y* – длина ответной плоскости по  $y$  в [м];
- *Plane Depth to Reference Frame* – расстояние между верхним и нижним пределами ответной плоскости по  $z$ ;
- *Force Law* – вид закона, описывающего силу по нормали;
- *Contact Stiffness* – коэффициент жёсткости контактного взаимодействия в [Н/м];
- *Contact Damping* – коэффициент демпфирования контактного взаимодействия в [Н/(м/с)];
- *Friction Law* – вид закона, описывающего силу трения, включая коэффициенты трения;
- *Visualize Contact Surfaces* – включение/выключение виртуальных объектов взаимодействия.

Для работы блока *Sphere to Plane* необходимо выключить отображение ошибки топологии. Для этого необходимо зайти в настройки  → *Simscape Multibody* → *Diagnostics* → *Topology* → *Rigidly constrained block* → и выбрать *none*.

Блок *MATLAB function* под названием *setpoint* задаёт требуемую траекторию перемещения точки  $F$ . Функция зависит только от времени, поэтому на вход *setpoint* подаются «часики». Блок отдельно рассчитывает траекторию по  $x$  и по  $y$ . Код представлен в пузыре ниже. Дифференцирование  $x$  и  $y$  даёт оценку скорости по соответствующим осям.

```
% 'функция_требуемая_траектория'
function y = fcn(t)
y=[0.8*sin(pi/2*t)+ 0.5;    0.3*sin(pi*t) + 1.2];
```



Блок *MATLAB function* под названием *Jacobian* рассчитывает транспортированный якобиан, согласно выражению (3.3), которое зависит только от положения звеньев *AB* и *BF* –  $q_1$  и  $q_2$  соответственно. Код представлен в пузыре ниже.

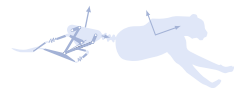
```
% 'функция_расчета_матрица_Якоби'
function Jacobian = fcn(q_1,q_2)
L=1;
J_T=[-L*sin(q_1)-L*sin(q_1+q_1)...
      L*cos(q_1)+L*cos(q_1+q_1);...
      -L*sin(q_1+q_2) L*cos(q_1+q_2)];
Jacobian=J_T;
```

Блок *MATLAB function* под названием *Force* рассчитывает значение силы, согласно выражению (3.5), которое зависит от разниц положения и скорости по осям  $x$  и  $y$ . Код представлен в пузыре ниже.

```
% 'функция_расчета_значения_силы'
function Force = fcn(d_q_x, d_q_y, d_v_x, d_v_y)
K=50; D=5;
Force=(K*[d_q_x; d_q_y]+D*[d_v_x; d_v_y]);
```

Блок *MATLAB function* под названием *multiplication* рассчитывает значение моментов, согласно выражению (3.8), которое зависит от ранее рассчитанных якобиана и силы. Код представлен в пузыре ниже.

```
% 'функция_расчета_момента'
function y = fcn(J_T, Force)
y=J_T*Force;
```



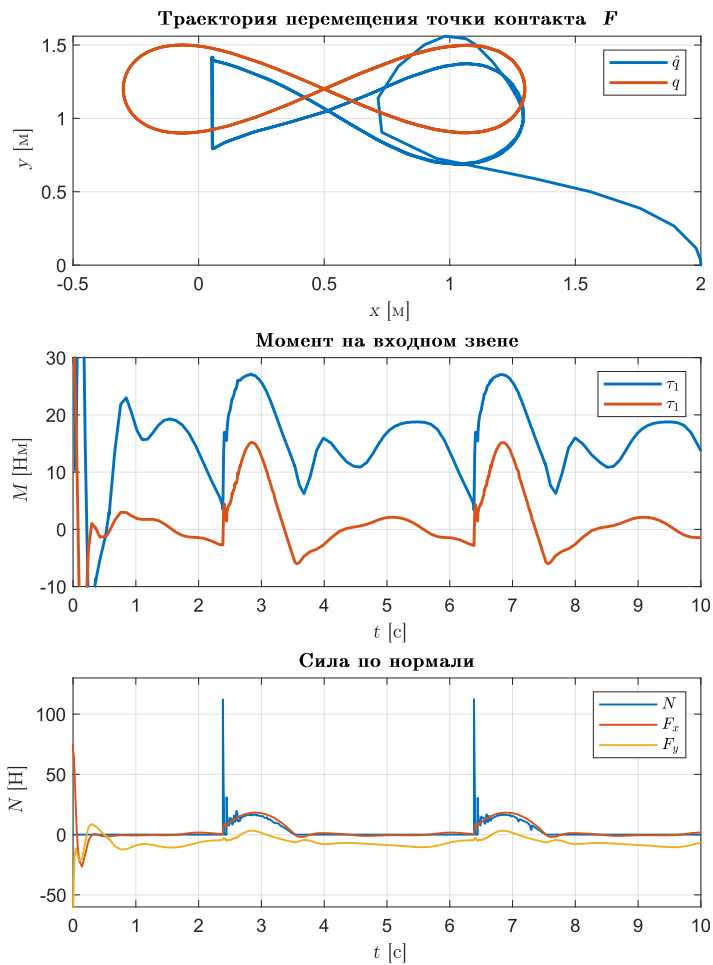
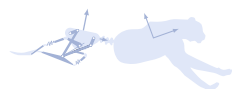


Рисунок 3.10. Результаты имитационного моделирования 2R робота



Результаты моделирования представлены на рисунке 3.10. На первом графике  $q$  отображает требуемую траекторию точки  $F$ , а  $\hat{q}$  – реальную. Так как в данном примере не была учтена динамика звеньев робота – есть постоянное отклонение от требуемой траектории. Кроме того, так как в сцене есть внешний объект, робот не может отработать всю траекторию.

Второй график отображает изменение моментов на обоих сочленениях за первые 10 секунд моделирования.

Третий график представляет сравнение силы реакции взаимодействия с сенсора блока *Sphere to Plane N*, а также расчётные значения сил реакции по оси  $x$  и  $y$ . Отметим, что значения  $N$  и  $F_x$  близки, однако в первый момент соударения  $N$  скачкообразно увеличивается до большого значения. Таким образом, метод штрафов имитирует диссипацию энергию соударения.



# Глава 4

## Моделирование составных элементов

В данном разделе представлено краткое описание разнообразных дополнительных элементов, позволяющих подробнее описать физику робототехнической системы. Кроме того, представлены примеры их исполнения.

### 4.1 Двигатель постоянного тока

#### 4.1.1 Аналитическая модель

Рассмотрим моделирование электропривода как составную часть задачи управления вращательным звеном робота, приводимым в движение двигателем постоянного тока (рисунок 4.1). Динамическая модель электромеханического привода на базе ДПТ может быть описана дифференциальными уравнениями вида

$$J\ddot{\theta} + b\dot{\theta} = \tau - \frac{1}{j}\tau_L, \quad (4.1)$$

$$L\frac{di}{dt} + Ri = v - k_e\dot{\theta}, \quad (4.2)$$

где  $\theta$  — угол поворота вала [рад],  $\dot{\theta}$  — угловая скорость вала [рад/с],  $J$  — приведенный момент инерции на валу двигателя [кг·м<sup>2</sup>],  $b$  — коэффициент вязкого трения [Н·с/(м)],  $\tau$  — момент, развиваемый двигателем, [Нм],  $\tau_L = \text{const}$  — постоянный момент нагрузки,  $j$  —

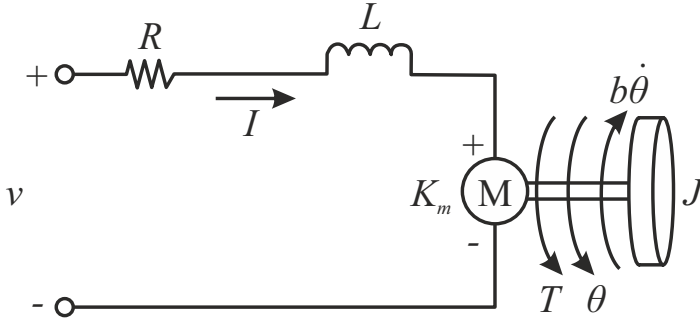


Рисунок 4.1. Принципиальная схема двигателя постоянного тока

передаточное число редуктора,  $i$  — сила тока [А],  $L$  — индуктивность [Гн],  $R$  — электрическое сопротивление [Ом],  $v$  — напряжение [В].

Обратим внимание, что развиваемый момент силы  $\tau$  пропорционален силе тока якоря ДПТ:

$$\tau = k_{\tau} i, \quad (4.3)$$

где  $k_{\tau}$  — механическая постоянная [Нм/А].

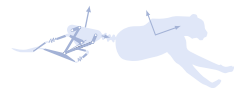
С учётом (4.3) перепишем модель (4.1), (4.2) в виде

$$\frac{J}{b} \ddot{\theta} + \dot{\theta} = \frac{1}{b} \left[ k_{\tau} i - \frac{1}{j} \tau_L \right], \quad (4.4)$$

$$\frac{L}{R} \frac{di}{dt} + i = \frac{1}{R} [v - k_e \dot{\theta}], \quad (4.5)$$

где обратим внимание, что  $T_1 = \frac{J}{b}$  и  $T_2 = \frac{L}{R}$  являются постоянными времени «механической» и «электрической» частей модели двигателя соответственно. Примем допущение о том, что постоянная времени электрической цепи пренебрежимо мала относительно второй,

$$\frac{J}{b} \gg \frac{L}{R} \approx 0$$



в таком случае выражение (4.5) примет вид

$$i = \frac{1}{R}[v - k_e \dot{\theta}]. \quad (4.6)$$

Подставим уравнение (4.6) в (4.4), получим

$$\frac{J}{b} \ddot{\theta} + \dot{\theta} = \frac{1}{b} \left[ \frac{k_\tau}{R} [v - k_e \dot{\theta}] - \frac{1}{j} \tau_L \right]$$

и преобразуем к виду

$$J \ddot{\theta} + B \dot{\theta} = \tau - \frac{1}{j} \tau_L, \quad (4.7)$$

где  $B = b + \frac{k_\tau k_e}{R}$ ,  $\tau = \frac{k_\tau}{R} v$ .





## 4.1.2 Имитационная модель

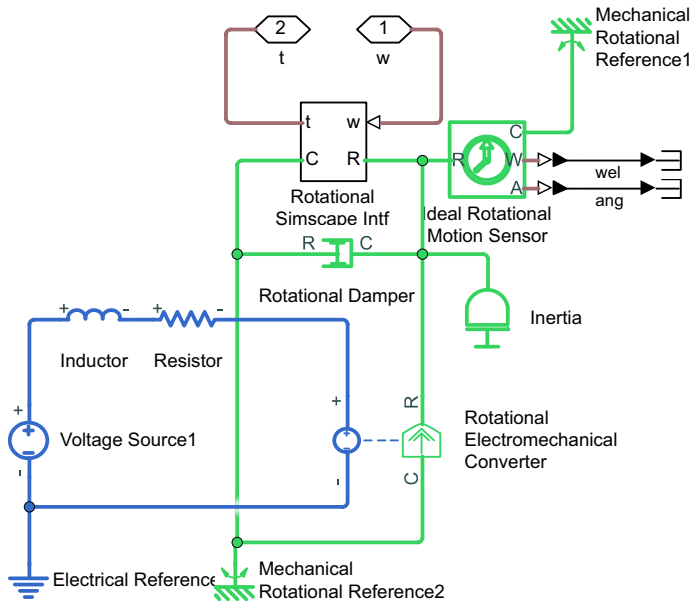
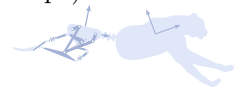


Рисунок 4.2. Имитационная модель двигателя постоянного тока

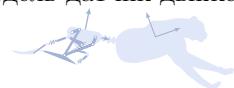
Рассмотрим, как с помощью Simscape можно создать модель двигателя постоянного тока. Моделируемая система выглядит следующим образом: двигатель, который имеет два электрических контакта и два механических, включая вращательную кинематическую пару. Основная задача – создать такую модель, которая включает в себя и электрическую, и механическую составляющую. Проектируемая имитационная модель должна выглядеть, как изображено на рисунке 4.2.

Построим данную модель двигателя с нуля:

1. В командной строке Matlab введем `ssc_new`, чтобы создать новую модель Simulink (в поле модели уже будут присутствовать основные блоки, используемые в Simscape).



2. Добавим *DC voltage source*, начав набирать на клавиатуре его название и выбрав в появившемся списке *DC voltage source*, после чего указать в параметрах напряжения 6 В.
3. Добавим индуктор и резистор, также вписав их названия в поисковую строку – *Inductor* и *Resistor* соответственно, соединив их последовательно с положительным контактом двигателя. В индукторе записать следующие параметры: *Inductance* = 121e-6 Гн; в резисторе записать *Resistance* = 0,346 Ом.
4. Нужен блок, который преобразует электрическую энергию в механическую. Добавим *Rotational electromechanical converter* и соединим отрицательный контакт резистора и положительный конвертора, отрицательный контакт двигателя с отрицательным конвертора. В параметрах укажем, что механическая составляющая двигателя *Constant of proportionality K* равна 1.
5. Добавим заземление *Electrical reference*, соединив его с отрицательным контактом электродвигателя.  
Теперь займёмся механической частью модели. Оставшиеся два порта (зелёная часть блока) *Rotational electromechanical converter* представляют механические контакты двигателя.
6. Прикрепим корпус двигателя к точке в пространстве, добавив блок *Mechanical rotational reference* и соединив его с портом *C* конвертора.
7. Это соединение представляет собой вал. Добавим блок *Inertia*, соединим с портом *R* конвертора для моделирования инерции вала. В параметрах *Inertia* установим, что инерция составляет 98,6e-7 кг·м<sup>2</sup>
8. Если необходимо смоделировать вязкое трение в подшипнике или опоре двигателя, то необходимо добавить демпфирование – блок *Damper*, соединим его согласно рисунку 4.2. В параметре *Damping coefficient* укажем 3e-5 Н·м/(рад/с).
9. Блоки датчиков необходимы для снятия интересующих данных имитационной модели. Добавим в модель датчик движе-



ния *Ideal Rotational Motion Sensor*, присоединим его согласно рисунку 4.2, выход соединим с *Mechanical rotational reference*.

10. Блоки датчиков выдают физические сигналы, которые должны быть преобразованы в сигналы Simulink для «оцифровки», т.е. получения численных характеристик физических сигналов. Для этого добавим блок *PS-Simulink Converter* и соединим с выходом *W*, в параметрах которого необходимо указать единицы измерения **rad/s**, так как это угловая скорость. С другой стороны соединим его, например, с блоком *Terminator* или *To Workspace*. Аналогично поступим с выходом *A*, в параметрах которого укажем **rad**, так как это угловое положение звена.
11. Для того чтобы вывести предварительные графики, нужно выделить связи между конвертером и терминатором, в появившемся меню выбрать *Enable Data Logging* (значок Wi-Fi), в верхней панели программы выбрать *Data Inspector*, в появившемся окне поставить галочки у тех параметров, которые необходимо посмотреть. Если необходимо построить *аккуратные* графики, необходимо использовать блок *To Workspace* и использовать команду **plot**.

Для подключения описанного выше механизма (в главе 2.3) к данному двигателю необходимо их соединить через блок *Rotational Simscape Intf*, к двум портам которого подводятся значения момента и угловой скорости с механизма робота, а два другие подключены к двигателю.



## 4.2 Ремни, тросы, шкивы и катушки

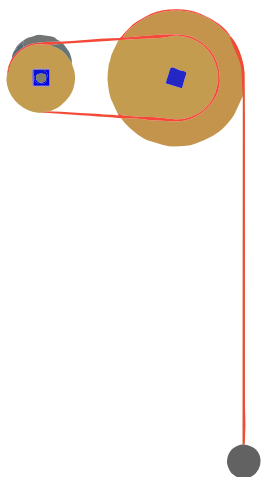


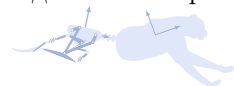
Рисунок 4.3. Визуализация имитационной модели со шкивами, катушкой, гибкими ремнём и тросом

В данном разделе рассмотрим, как можно спроектировать модель системы, которая включает в себя гибкие элементы, например, ремни и тросы, а также элементы, участвующие в передаче движения – шкивы и катушки.

На рисунке 4.3 изображена визуализация имитационной модели, состоящей из двух шкивов и катушки. Меньший левый шкив, присоединённый к валу двигателя, задаёт движение, которое передаётся посредством ремня на второй шкив, который жёстко соединён с катушкой большего радиуса. Свободный конец троса катушки присоединён к грузу.

Рассмотрим этапы проектирования данной имитационной модели. Далее приведены подробные комментарии к элементам, которые ещё не были описаны ранее в данном учебном пособии. Блок-схема проектируемой системы представлена на рисунке 4.4.

1. Создадим новый файл модели с помощью команды `smnew`.
2. К *World Frame* присоединим блок *Solid* с визуализацией двигателя, используя STEP файл. Переименуем блок в *Motor*. Блок *Rigit Transform* перед двигателем необходим для обеспечения вращения в плоскости  $xy$ .
3. Для обеспечения вращательного движения первого шкива относительно оси  $z$  добавим в модель блок *Revolute Joint*, присоединив порт *B* с портом *F1* блока *Motor*.
4. В блоке *Revolute Joint* выберем, что движение задаётся с помощью управляющего момента, которое подаётся на порт *t*.



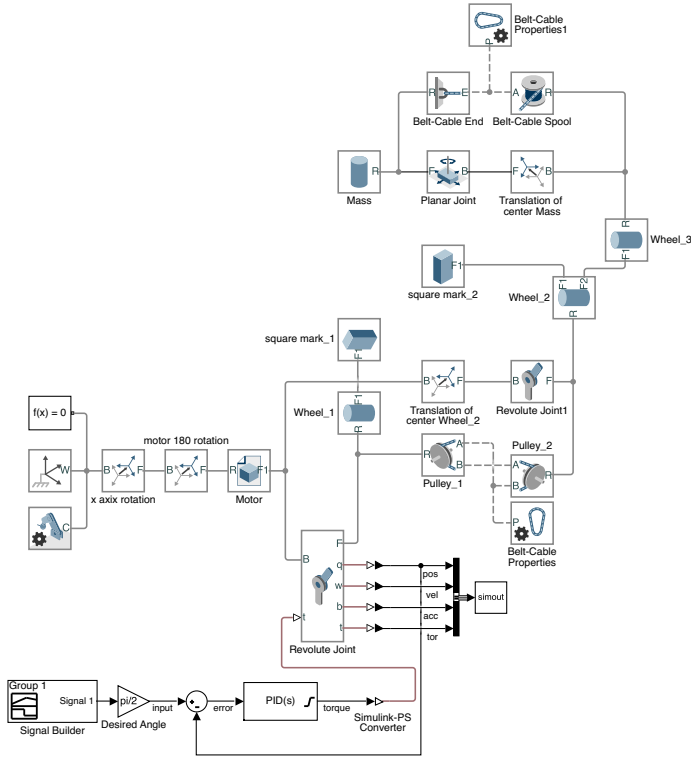
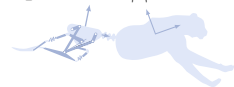


Рисунок 4.4. Блок-схема имитационной модели со шкивами, катушкой, гибкими ремнём и тросом

Управляющий момент рассчитывается посредством ПИД регулятора, который управляется по ошибке позиционирования. Требуемое положение задаётся с помощью блоков *Signal Builder* и *Gain*. *Signal Builder* задаёт плавный ступенчатый сигнал с 0 до 1. Коэффициент блока *Gain* равен требуемому углу поворота. Блок необходим для удобства настройки коэффициента. Момент с регулятора подаётся на порт *t* блока *Revolute Joint* через *Simulink-PS Converter*, в котором указывается система измерения.



5. Добавим блок *Solid* с первым шкивом *Wheel\_1*, со следующими параметрами: *Radius* =  $r_p$  м, *Length* = 0.5 см. К центру *Wheel\_1* добавим прямоугольник *square mark\_1* для наглядности вращения шкива на визуализации.
6. Добавим второй шкив *Wheel\_2* со следующими параметрами: *Radius* =  $r_{p\_2}$  м, *Length* = 0.5 см и с прямоугольником в центре *square mark\_2*. С помощью блока *Rigid transform* под названием *Translation of center Wheel\_2* переместим центр шкива правее на значение *offset*. Соединим этот блок с первым *Revolute Joint*, а также создадим второй *Revolute Joint* и соединим его с *Rigid transform* и со вторым шкивом.
7. Чтобы передать вращательное движение через ременную передачу с одного шкива на второй, добавим два блока *Pulley* из библиотеки *sm\_lib* → *Belts and Cables*. Этот блок имитирует катушку, которая вращается вокруг оси *Z*, а ремень находится в плоскости *X – Y*.
8. Соединим порты *R* блоков *Pulley\_1* (параметры: *Pitch radius* =  $r_p$  м) и *Wheel\_1*. Аналогично соединим *Pulley\_2* (параметры: *Pitch radius* =  $r_{p\_2}$  м) и *Wheel\_2*. Таким образом, ремень был установлен на оба шкива.
9. Соединим порты двух *Pulley*. Если соединить порт *A* с *A*, а *B* с *B*, то будет перекрестная передача, если соединить *A* с *B*, то прямая. Построим прямую ременную передачу.
10. Необходимо добавить блок *Belt – CableProperties* с параметрами передачи.
11. Добавим катушку *Wheel\_3* с тросом, который крепится к грузу. Катушка соединена через порт *F1* с портом *F2* второго шкива на одной оси. Благодаря этому вращение с *Wheel\_2* передаётся на *Wheel\_3*.
12. Добавим блок *Solid* под названием *Mass*, имитирующий массу на конце троса. С помощью блока *Rigid transform* под названием *Translation of center Mass* переместим центр массы ниже шкива на значение [ $r_{p*2}$  -*offset\_mass* 0] м и соединим их

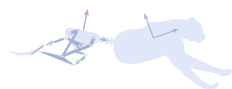


с помощью блока *PlanarJoint*, плоского соединения с 3 степенями свободы: две поступательные и одна вращательная. Таким образом было описано движение груза.

13. Смоделируем накручивание троса с массой на шкив. Добавим блок *Belt-Cable Spool* из библиотеки *sm\_lib* → *Belts and Cables*, который представляет катушку с бесконечной намоткой и размоткой. Катушка вращается вокруг оси *Z* против часовой стрелки. Соединим порт *R* блока *Cable Spool* с портом *R* *Wheel\_3*. Чтобы соединить катушку с массой, нужно присоединить ее с помощью блока *Belt-Cable End*, точка крепления на конце троса. Соединить их порты *R*, а порт *E* соединить с портом *A* *Belt-Cable Spool*. Добавить блок параметров троса *Belt-Cable Properties*

Таким образом, была собрана имитационная модель системы со шкивами, катушкой, гибкими ремнём и тросом. Ниже представлен код *m*-файла с параметрами системы.

```
M=1; %'Масса'  
r_p=0.02;%'радиусшкива_1'  
r_p_2=0.025;%'радиусшкива_2'  
offset=0.08;%'м'  
K_P=2.5E+01;  
K_D=0.1;  
K_I=10;
```



# Глава 5

## Комплексное моделирование мехатронных систем

Данная глава посвящена комплексному имитационному моделированию мехатронных систем на примере разработанных авторами захватного устройства «Ухват» [16] и галопирующего робота «Гепард» [17].

На примере моделирования конструкции захватного устройства в первом разделе показан способ верификации спроектированной твердотельной конструкции в SolidWorks на соответствие требуемым характеристикам работы.

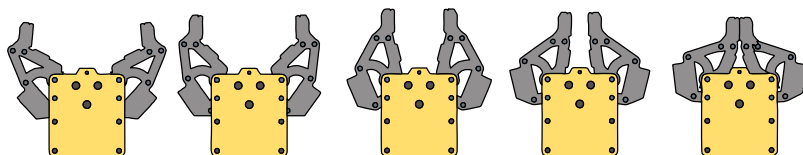


Рисунок 5.1. Последовательность сгибания пальцев захватного устройства «Ухват» при имитационном моделировании щипкового захвата тел коробчатой формы

Во втором разделе главы на примере имитационного моделирования галопирующего робота показан пример комплексного проведения виртуального эксперимента работы робота с абстрактной геометрией. Приведено решение задачи структурной и параметрической оптимизации конструкции ноги.



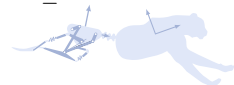
## 5.1 Установка Simscape Multibody Link и экспорт моделей САПР

При построении имитационных моделей сложных мехатронных систем, конструкция которых предварительно уже была проработана посредством твердотельного моделирования, востребованной является функция экспорта данных из САПР в среду MATLAB. Далее рассмотрим, каким образом это может быть реализовано на примере интеграции Simscape Multibody Link[18], плагина для экспорта сборок из САПР, таких как SolidWorks, Autodesk Inventor и PTC Creo. Плагин генерирует XML файл, содержащий информацию о структуре, свойствах и геометрии деталей экспортируемой из САПР сборки. Сгенерированный XML файл может быть импортирован в Simscape Multibody для автоматической генерации эквивалентной имитационной модели.

1. Скачайте установочные файлы:
  - Перейдите на страницу загрузки [Simscape Multibody Link](#)<sup>1</sup>;
  - Следуйте инструкциям на странице загрузки;
  - Сохраните zip архив и файл MATLAB в удобной папке. Выберите файл, соответствующий вашей версии MATLAB, например, версия R2018b для архитектуры x64. Не извлекайте zip архив.
2. Запустите установку:
  - Запустите MATLAB от имени администратора;
  - Добавьте сохранённые установочные файлы в активную папку MATLAB. Вы можете сделать это, введя команду `addpath('имя папки')` в командной строке MATLAB. Замените *имя папки* именем папки, в которой вы сохранили установочные файлы, например, `C:\Temp`.
  - В командной строке MATLAB введите `install_addon('название архива')`. Замените *название архива*

---

<sup>1</sup>[https://www.mathworks.com/campaigns/offers/download\\_smlink.html](https://www.mathworks.com/campaigns/offers/download_smlink.html)

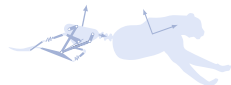


именем скаченного zip архива, например, `smlink.r2018b.win64.zip`.

3. Зарегистрируйте MATLAB в качестве сервера. Каждый раз, когда вы экспортируете модель сборки САПР, модуль *Simscape Multibody Link plug-in* пытается подключиться к MATLAB. Для установления соединения необходимо зарегистрировать MATLAB в качестве сервера. Вы можете сделать это двумя способами:
  - В MATLAB, работающем в режиме администратора, в командной строке введите `regmatlabserver`.
  - В окне MS-DOS, работающем в режиме администратора, в командной строке введите `matlab-regserver`.
4. Включите модуль *Simscape Multibody Link Plug-In* в SolidWorks:
  - В командной строке MATLAB введите `smlink_linksw`;
  - Запустите SolidWorks;
  - В меню *Tools* (Инструменты) выберите пункт *Add-Ins* (Дополнения).
  - В диалоговом окне *Add-Ins* поставьте галочку у *Simscape Multibody Link*. При запуске или открытии сборки САПР в строке меню SolidWorks появляется меню *Simscape Multibody Link*.

Если на вашем компьютере установлено более одной версии SolidWorks, команда `smlink_linksw` добавляет модуль *Simscape Multibody Link plug-in* ко всем версиям одновременно. Однако вы должны поставить галочку у *Simscape Multibody Link* в диалоговом окне *Add-Ins* отдельно для каждой версии SolidWorks, из которой вы хотите экспортировать сборки.

5. Откройте в SolidWorks сборку для экспорта. Обратите внимание, что модели деталей не поддерживаются, поэтому меню экспорта *Simscape Multibody Link* для деталей недоступно. Если необходимо экспортировать только одну деталь, просто сохраните её в STEP формате и используйте блок *Solid* для MATLAB R2017-R2018 или *From File* для MATLAB R2019.



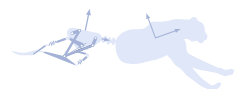
6. Экспортируйте модель из SolidWorks, для чего пройдите по следующему пути: *Tools* → *Simscape Multibody Link* → *Export* → *Simscape Multibody*.
7. Укажите имя файла и папку для сохранения XML. Модуль Simscape Multibody Link plug-in сгенерирует XML файл и STEP файлы геометрии деталей, необходимые для визуализации модели после импорта. Обратите внимание, что названия деталей должны содержать только латинские буквы и цифры, это же касается информации о конфигурации деталей. Если вы используете русскую версию SolidWorks, убедитесь, что конфигурация деталей переименована с изначального «По умолчанию».
8. Используйте функцию `smimport('имя_файла.xml')` с именем файла XML для импорта сборки САПР в среду моделирования Simscape Multibody. Функция анализирует файл и использует содержащиеся в нем данные для создания многозвенной структурной схемы Simscape и сопутствующего файла данных MATLAB.

Дополнительная информация о работе Simscape Multibody Link приведена в статьях [CAD Translation](#) [19] и [Mates and Joints](#) [20].

## 5.2 Моделирование захватных устройств

### 5.2.1 Описание механизма

На рисунке 5.2 представлены разработанные опытные образцы захватных устройств (ЗУ): промышленных трёхпалого и двухпалого ЗУ, а также протеза кисти. Особенностью данных захватных устройств является способность обеспечивать различные виды захватов. Механизм пальца ЗУ с помощью звена переменной длины с активным регулированием способен переключаться между полноприводным и неполноприводным режимами работы для обеспечения либо точного щипкового захвата, либо обхватывающего силового соответственно.



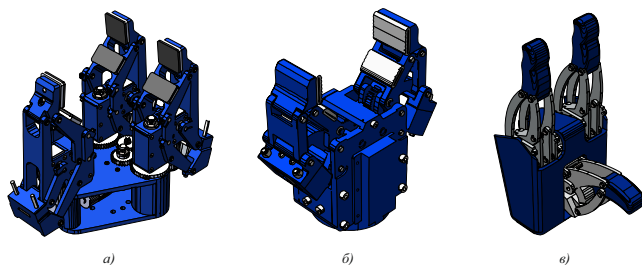


Рисунок 5.2. Визуализация прототипов захватных устройств: а) трёхпальный и б) двухпальный промышленный захват, в) протез кисти

На рисунке 5.3, а-в изображена последовательность сгибания пальца в полноприводном режиме. Отметим, что рабочая поверхность дистальной фаланги  $FD$  на всем диапазоне сгибания пальца сохраняет свою ориентацию в пространстве.

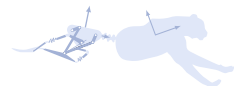
На рисунке 5.3, г-д изображён процесс сгибания пальца в режиме адаптивного захвата при разомкнутом звене  $BK$ . Видим, что, начиная своё движение (рисунок 5.3, г), палец сгибается аналогично полноприводному режиму. Войдя в контакт с объектом (рисунок 5.3, д), проксимальная фаланга  $ODE$  фиксируется, но благодаря тому, что звено  $BK$  разомкнуто, входное звено  $AB$  продолжает движение, сгибая дистальную фалангу  $CDF$  вокруг кинематической пары  $D$  до тех пор, пока поверхность дистальной фаланги  $FD$  не соприкоснётся с захватываемым объектом (рисунок 5.3, е).

Подробнее о механизме пальца, конструкции захватных устройств, звене переменной длины и способе управлении рассказано в работах [21, 16, 22, 23, 24, 25].

## 5.2.2 Описание задач моделирования

Задачами имитационного моделирования являются:

- Верифицировать способность захватного устройства обеспечить заявленные типы захватов: полноприводный и неполноприводный режимы работы для обеспечения точного щипкового захвата и обхватывающего силового соответственно.



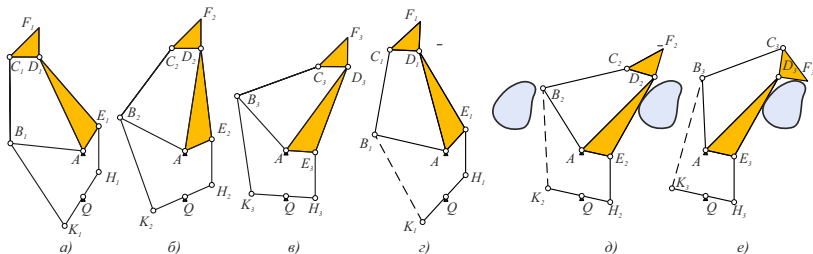


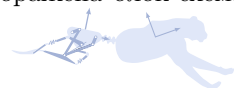
Рисунок 5.3. Полноприводный захват: а) начальное, б) промежуточное и в) конечное положения пальца; неполноприводный захват: г) начальное положение, д) контакт проксимальной и е) контакт дистальной фаланг

- Измерить отклонение рабочей поверхности дистальной фаланги от вертикальности при выполнении точного захвата. Параллельность рабочих поверхностей фаланг является важным параметром, который влияет на надёжность захвата.
- Построить графики изменения положения входного звена, приложенного к нему момента, значения силы реакции на рабочей поверхности дистальной фаланги, изменения длины звена переменной длины.

### 5.2.3 Построение имитационной модели

Экспортируем сборку спроектированного двупалого хватного устройства из SolidWorks в Simulink, воспользовавшись *Simscape Multibody Link Plug-In*. Simulink создаст большее количество кинематических пар, чем есть на самом деле, вследствие неправильной интерпретации части сопряжений SolidWorks. По этой причине после импорта сборки необходимо перепроверить схему, привести её в порядок, сгруппировать элементы по блокам.

На рисунке 5.4, а показана блок-схема имитационной модели хватного устройства, взаимодействующего с блоками тела коробчатой формы *Brick* и с блоком сферы *Sphere*, которые лежат на плоскости *World Plane*. На рисунке 5.4, б изображена блок-схема



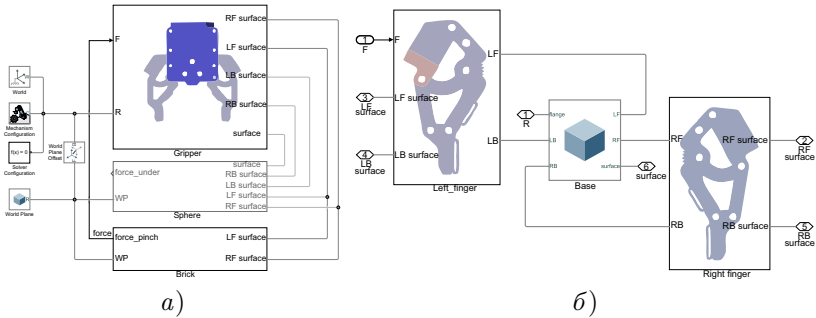


Рисунок 5.4. Блок-схема имитационной модели захватного устройства *a)* и блок-схема строения захвата *б)*

строения захвата, представляющего собой корпус захвата *Base* и два зеркальных пальца *Left finger* и *Right finger*.

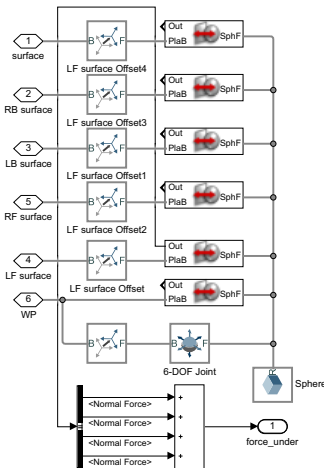
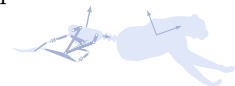


Рисунок 5.5. Блок-схема сферы

На рисунках 5.5 и 5.6 изображены блоки сферы и тела коробчатой формы с блоками моделирования контакта соответственно. Предполагается оставлять активным только тот блок, моделирование тела которого необходимо произвести, в то время как второй блок должен быть закомментирован.

При моделировании захвата тела сферической формы необходимо описать 6 точек контакта сферы: 1 с плоскостью, на которой лежит тело, 2 с дистальными фалангами, 2 с проксимальными и 1 с корпусом. Сама сфера не имеет кинематических ограничений, поэтому описывается через блок кинематической пары со шестью степе-



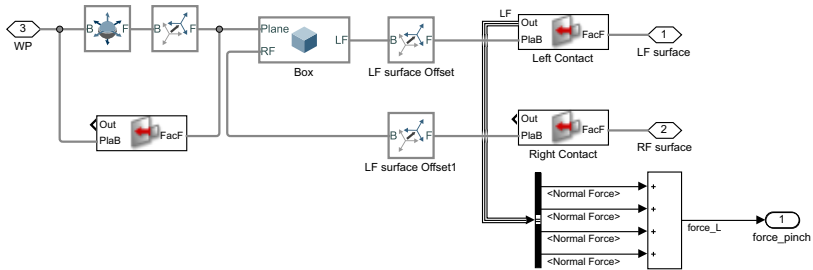


Рисунок 5.6. Блок-схема тела коробчатой формы

нями свободы  $6-DOF$ . Для моделирования точки контакта должен использоваться блок *Sphere to Plane Force*.

При моделировании захвата тела коробчатой формы необходимо описать 3 плоскости контакта: 1 с плоскостью, на которой лежит тело и 2 с дистальными фалангами. Аналогично, тело не имеет кинематических ограничений, поэтому описывается через блок кинематической пары со шестью степенями свободы  $6-DOF$ . Для моделирования плоскости контакта должен использоваться блок *Face to Plane Force*.

На рисунке 5.7 представлена блок-схема левого пальца захватного устройства. Позиционное управление реализовано с помощью ПИД регулятора. Для моделирования звена переменной длины необходимо оставить активным модуль *Rigid Transform* под названием *clutch*, в то время как блок *Face to Plane Force* должен быть закомментирован. В таком случае механизм пальца будет иметь только одну степень свободы. В противном случае, когда *clutch* закомментирован, а *Face to Plane Force* активен, длина звена переменной длины *ВК* будет переменной и палец будет иметь 2 степени свободы.

Остаётся определить коэффициенты регуляторов двигателей, приводящих в движение пальцы, и настроить блоки моделирования контакта.



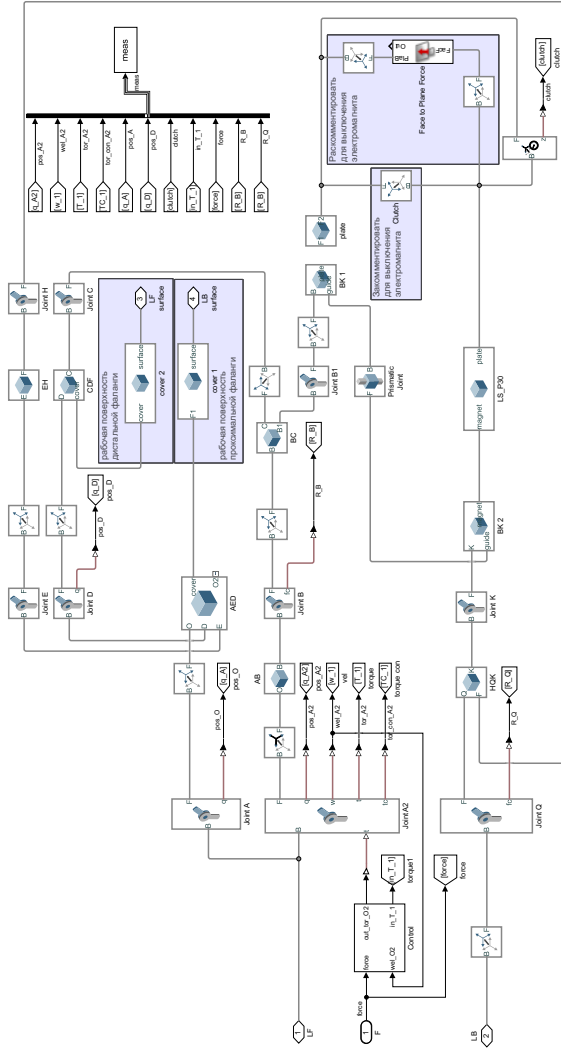
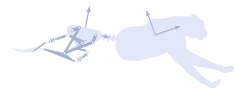


Рисунок 5.7. Блок-схема имитационной модели пальца захвата





## 5.2.4 Результаты моделирования

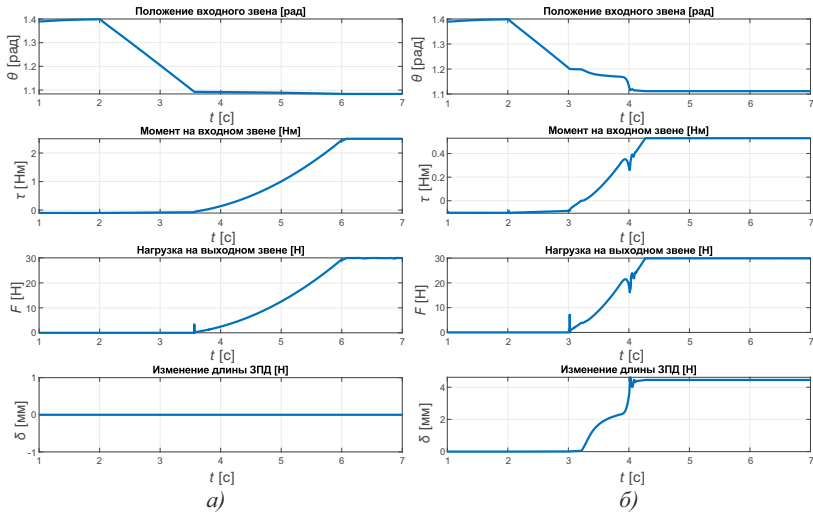
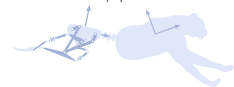


Рисунок 5.8. *а)* Переходные процессы системы в полноприводном режиме и *б)* в неполноприводном режиме

На рисунке 5.8, *а* изображены результаты имитационного моделирования для переходных процессов при захвате объекта в полноприводном режиме и на рисунке 5.8, *б* – в неполноприводном. Приведены графики изменения положения входного звена, управляющего момента, силы контактного взаимодействия, а также сравнение изменения длины звена переменной длины  $\delta$  при выполнении щипкового и обхватывающего захватов. В полноприводном режиме длина звена постоянна, движение механизма полностью детерминировано. При неполноприводном режиме длина звена начинает изменяться после первого контакта проксимальной фаланги с объектом манипулирования.

На рисунке 5.9 показано отклонение поверхности дистальной фаланги от параллельности на всем диапазоне перемещения входного звена по модулю. Минимальное отклонение от геометрических размеров звеньев приводит к заметному отклонению дистальной



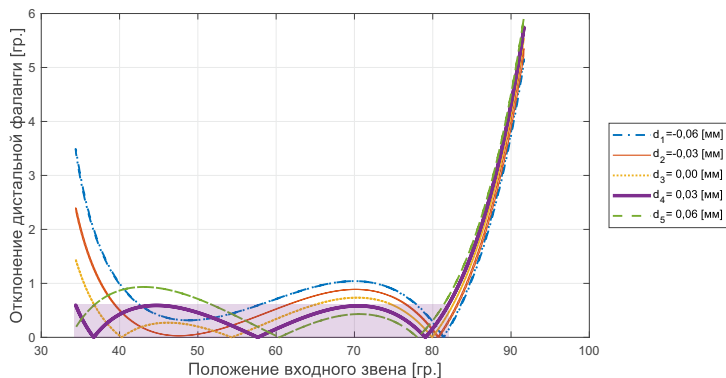
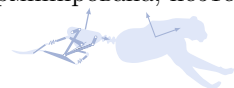


Рисунок 5.9. Отклонение дистальной фаланги от параллельности на всем диапазоне перемещения входного звена. Бледно-фиолетовым прямоугольником указан рабочий интервал

фаланги. На графике отображены размеры зазоров между соединяемыми элементами звена переменной длины.

Жирная сплошная фиолетовая линия соответствует зазору  $d_4 = 0,03$  мм, при котором отклонение от параллельности минимально, причём оба экстремума находятся на одной горизонтальной прямой. Сильное отклонение от параллельности наблюдается только в самом крайнем раскрытом положении. На остальном промежутке сгибания отклонение от параллельности составляет менее  $0,6^\circ$ . Фиолетовая кривая пересекает отметку нулевого отклонения три раза - в трёх положениях синтеза. Моделирование показывает, насколько точно должны быть выполнены детали захватного устройства для обеспечения требуемого захвата. Геометрия конструкции была скорректирована с учётом вычисленного параметра, на который требуется увеличить длину звена.

На рисунке 5.10 изображены конфигурационные пространства для механизма пальца в обоих режимах работы, представляющие зависимость между углами входного звена, проксимальной и дистальной фаланг относительно сочленения  $O$ . В полноприводном режиме траектория движения полностью детерминирована, поэто-



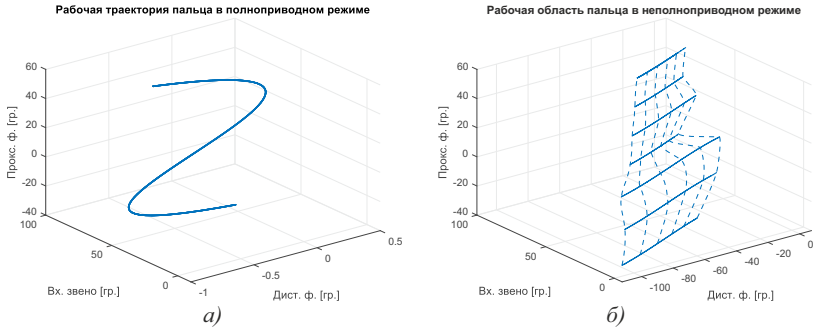


Рисунок 5.10. Зависимость положений проксимальной и дистальных фаланг от положения входного звена. *а)* Траектория пальца в полноприводном и *б)* рабочая зона в неполноприводном режимах при дискретных положениях проксимальной фаланги

му конфигурация имеет вид пространственной кривой. В неполноприводном режиме конфигурация представляется в виде рабочей зоны. Сплошные линии отображают зависимость положения дистальной фаланги от входного звена при дискретных положениях проксимальной фаланги. Пунктирные линии являются аппроксимацией для остальных положений проксимальной фаланги.



## 5.3 Моделирование галопирующих роботов

### 5.3.1 Описание механизмов

Предлагается осуществлять динамическую локомоцию робота с помощью звена переменной длины (ЗПД) с гибким элементом, для возбуждения резонанса, к которому должно быть приложено гармоническое воздействие. В качестве периодической силы может быть использована сила инерции колебания тела робота.

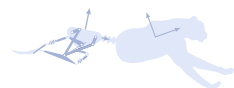
Необходим механизм, способный:

- Обеспечить колебательное перемещение тела робота по вертикальной оси на небольшое расстояние, но с высокой частотой;
- Обеспечивать перемещение по горизонтальной оси с широким диапазоном скоростей, в том числе и с нулевой горизонтальной скоростью, т.е прыгать на месте.

В качестве примера рассмотрим моделирование реконфигурируемого механизма с возможностью изменения траектории движения ноги робота с развязанными каналами управления. Первый канал должен отвечать за перемещение по вертикали, второй канал – за перемещение по горизонтали: высота прыжка и скорость бега соответственно.

Для достижения стабильной работы при прыжке на месте центр масс должен располагаться точно над точкой контакта с полом, а переход в режим бега должен осуществляться отклонением центра масс от точки контакта.

На рисунке 5.11, *a* изображён механизм «минитаур», который был использован впервые при разработке робота *Ghost Robotics Minitaur* [26]. В [27] механизм «минитаур» был использован в качестве конструкции бедра ноги робота-гепарда, чьё основное отличие от *Ghost Robotics Minitaur* состоит в том, что входными звеньями являются постоянно вращающиеся кривошипы, в отличие от коромысел, испытывающих импульсное движение, а также принцип локомоции основан на резонансе гибкого элемента, обеспечивающего автоколебательный характер поведения.



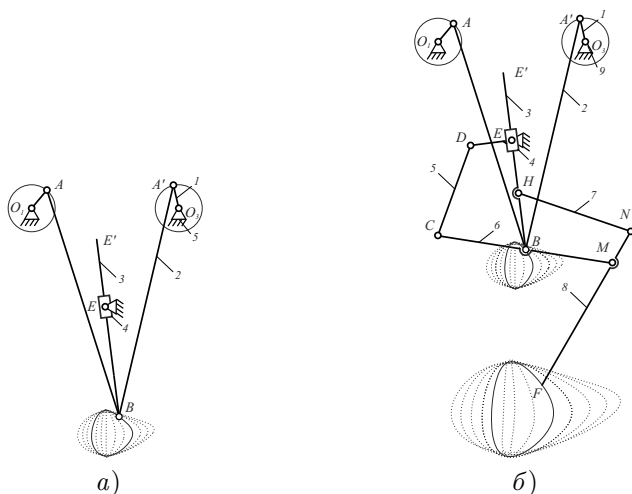
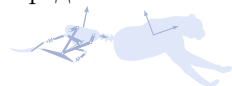


Рисунок 5.11. Механизмы ног галопирующих роботов. а) Механизм бедра и б) механизм целой ноги вместе с траекториями выходных точек.

В рассмотренном примере было использовано телескопическое звено с пружиной сжатия для поглощения ударных сил, пассивного накопления и рекуперации энергии перемещения. В данном разделе речь идёт о синтезе целой ноги, инспирированной строением ноги реального гепарда, способной достичь более высоких показателей ускорения, скорости и меньшего коэффициента затрат энергии на перемещение КЗП. Добавленный четырехзвенный механизм к механизму «минитаур» необходим для абсорбирования удара с полом и его неровностей, для чего необходимо интегрировать в него гибкий/-кие элемент/-ы. Кроме того, механизм должен обеспечивать плавное толкающее движение ноги в момент толчка и группироваться в фазе полёта для уменьшения инерции.

Различие между траекториями точек  $B$  и  $F$  можно увидеть на рисунке 5.11. Заметим, что механизм плюсны действует как преобразователь движения, так как соотношение между обеими траекториями равно 2. В результате траектория бега в два раза шире и выше, что приводит к более высокой скорости передвижения.



### 5.3.2 Описание задач моделирования

Задачами имитационного моделирования являются:

- Верифицировать способность галопирующего робота осуществлять динамическую локомоцию;
- Провести оптимизацию расположения и характеристик гибких элементов для обеспечения энергоэффективного перемещения;
- Построить графики зависимостей преодоленной дистанции, высоты прыжка и коэффициента затрат энергии на перемещение от значений фазы между кривошипами при различных параметрах гибких элементов.

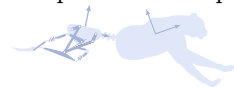
### 5.3.3 Имитационное моделирование

Расположение и характеристики звена переменной длины с гибким элементом оптимизируются методом направленного поиска для обеспечения наиболее энергоэффективного передвижения. Целевой функцией, требующей минимизации, является коэффициент затрат энергии на перемещение (КЗП, cost of transport), равный отношению потраченной энергии на производство веса робота на преодоленное расстояние:

$$CoT = \frac{E}{Wd}, \quad (5.1)$$

где  $E$  - энергия [Дж],  $W$  - вес [Н],  $d$  - перемещение [м]. Регулируемыми параметрами являются параметры ЗПД: коэффициент жёсткости, коэффициент демпфирования и начальная длина при использовании пружины сжатия/растяжения, либо значения модуля Юнга и коэффициента Пуассона изгибаемого эластичного элемента, которые ограничены небольшим интервалом значений, при которых достижимо стабильное галопирующее поведение.

На рисунке 5.12 изображена блок-схема подсчёта затрачиваемой энергии на перемещение. Вначале задаётся желаемая скорость вращения кривошипов и значение фазы между ними. Определяются значения положений, скоростей, ускорений кривошипов при



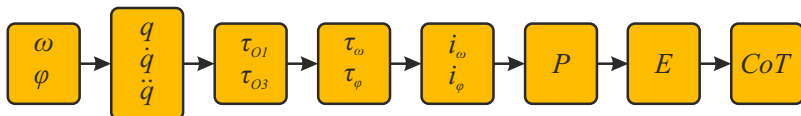


Рисунок 5.12. Блок схема подсчёта затрачиваемой энергии

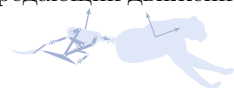
моделировании бега. С помощью имитационной модели определяются значения моментов на кривошипах, значения моментов на двигателях, потребляемые токи. Далее рассчитывается затраченная мощность электродвигателей, после чего определяется затраченная энергия.

Критерием стабильности является регулярность траектории точки контакта с полом, критерием энергоэффективности является стремление целевой функции к значению КЗП реального гепарда, равного 0,4 [28] или меньше. Оптимизация параметров была проведена направленным поиском с использованием имитационного моделирования.

При проектировании галопирующего робота имитационное моделирование необходимо для верификации синтезированных кинематических схем, проверки работоспособности конструкции устройства с учётом распределения масс звеньев и их параметров инерции, выбора местоположения и характеристик гибкого элемента, настройки системы управления, а также определения динамических параметров: скорость горизонтального перемещения, усреднённая высота прыжка и показатели энергозатрат.

### 5.3.4 Моделирование привода ноги

Так как оба механизма, редуцированный и полный механизмы ноги робота-гепарда, основаны на структуре минитаур, привод и система управления в целом едина. Оба механизма приводятся в движение вращением двух кривошипов  $AO_1$  и  $A'O_3$  (рисунок 5.11, б). Для их управления можно использовать два двигателя, но в таком случае необходимо очень точно контролировать синхронность их вращения. Вместо этого был реализован редуктор, передающий движение



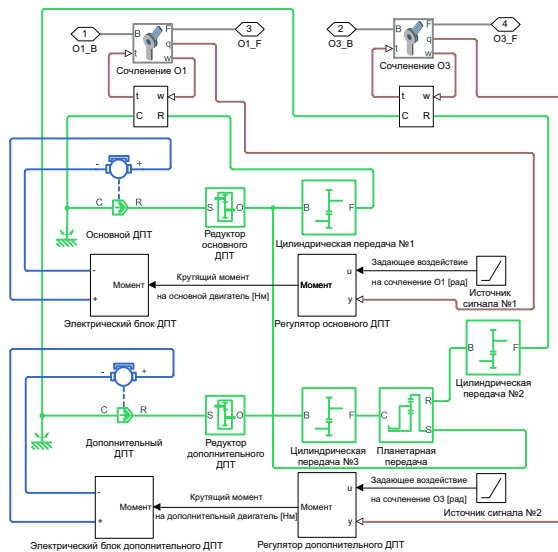
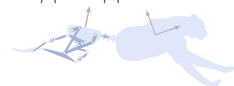


Рисунок 5.13. Блок двигателя. Привод состоит из двух вращательных сочленений, блоков источников задающих воздействий, регуляторов, электрических блоков и редукторов

с одного двигателя на оба кривошипа, при этом фаза между кривошипами управляется с помощью дополнительного серводвигателя и планетарной передачи.

На рисунке 5.13 изображена структурная схема привода робота, реализованная с помощью блоков Simscap. Скорость и значение фазы на кривошипах задаётся с помощью источников сигнала №1 и №2 для кинематических пар  $O_1$  и  $O_3$  соответственно. Задающие воздействия вместе со значениями текущих положений подаются в соответствующие блоки регуляторов для основного и дополнительного двигателей, по разнице сигналов рассчитываются значения управляющих моментов. Положение обоих кривошипов управляется с помощью блоков ПИД-регулятора. Моменты подаются в электронные блоки, моделирующие ШИМ-трансформатор и H-мост, на выходе получаем значения токов, подаваемых в модели двигателей.





Далее движение с двигателями передаётся с помощью блоков редуктора через интерфейсы на кинематические пары  $O_1$  и  $O_3$ . Основной двигатель передаёт движение на сочленение  $O_1$  с помощью цилиндрической передачи №1, кроме того, вал двигателя сопряжён с солнечным центральным зубчатым колесом  $S$  планетарной передачи. Эпицикл  $R$  планетарной передачи сопряжён с цилиндрической передачей №2, которая приводит в движение сочленение  $O_3$ . Планетарные зубчатые колеса через цилиндрическую передачу №3 соединяются с валом дополнительного двигателя. Благодаря такой конструкции сигналы управления скорости вращения кривошипов и значение фазы, задаваемое дополнительным серво двигателем, декомпозированы механически

Блоки, учитывающие электромагнитные эффекты, важны по двум причинам. Во-первых, они необходимы для оценки потребляемой электроэнергии, которая влияет на ёмкость батарей; во-вторых, они позволяют непосредственно вычислять напряжения и токи, которые являются усилиями и потоками в рамках порт-Гамильтонова подхода, который планируется использовать в дальнейшем при разработке регуляторов, учитывающих объем энергии в системе.

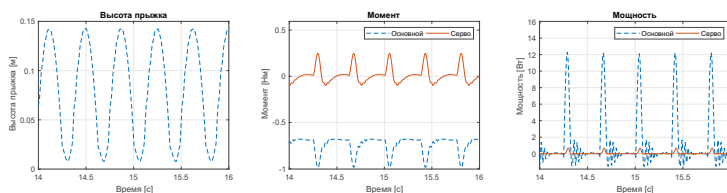
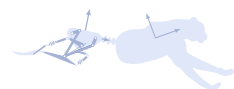


Рисунок 5.14. Результаты моделирования прыжка на месте для редуцированного механизма ноги. Параметры моделирования  $K_1 = 1,8 \text{ Н/мм}$ ,  $b=1 \text{ Н/(м/с)}$ ,  $N=5 \text{ см}$



### 5.3.5 Моделирование редуцированного механизма ноги

Начнём имитационное моделирование с построения модели для прыгающего робота, который представляет собой типичный перевернутый маятник с пружиной. Для достижения галопирующего поведения используется ЗПД с пружиной. При нулевой фазе между кривошипами робот прыгает на месте, при изменении фазы робот начинает перемещаться с горизонтальной скоростью.

Результаты бега прыгающего робота представлены на рисунке 5.16. Симуляция проводилась для широкого диапазона фаз: от  $-2,9$  рад до  $2,9$  рад, с различными коэффициентами жёсткости пружины и демпфирования. Размер шага  $0,26$  рад. Анимация бега для прыгающего робота отображена на рисунке 5.15, а. Было обнаружено, что стабильное поведение возможно в узком диапазоне коэффициентов жёсткости. На графике отображены результаты моделирования для коэффициентом жёсткости  $K_1 = 1,8$  Н/мм,  $K_2 = 1,6$  Н/мм,  $K_3 = 1,4$  Н/мм и демпфирования  $\beta = 1$  Н/(м/с). Длина пружины в состоянии покоя  $5$  см. Основная масса сосредоточена в теле робота, общая масса робота равна  $0,875$  кг.

На рисунке 5.16 изображена зависимость КЗП от фазы между кривошипами. С увеличением фазы существенно увеличивается скорость бега и, соответственно, пройденное расстояние за  $20$  секунд моделирования. На первом графике показана зависимость высоты прыжка от значения фазы. Видно, что высота стабильна и практически не зависит от фазы и составляет почти  $14$  см. Второй график показывает расстояние, пройденное за  $20$  секунд моделирования. Максимальная горизонтальная скорость составляет примерно  $1$  м/с. Третий график - это соотношение между коэффициентом затрат на перемещение КЗП и фазой. КЗП значительно уменьшается с увеличением фазы из-за большего пройденного расстояния за время моделирования. Если расстояние равно нулю (прыжок на месте), то КЗП стремится к бесконечности.

Механизм бедра способен самостоятельно прыгать и бегать, однако траектория точки контакта с полом может быть увеличена для осуществления более высокого и длинного прыжка, расширив механизм до целой ноги с имитацией большой, малой берцовых костей и плюсны.



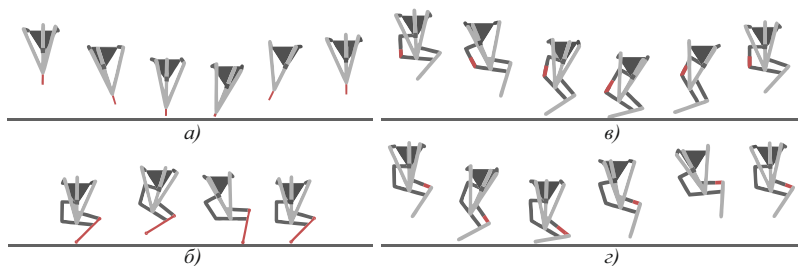
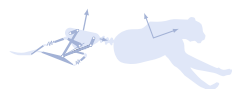


Рисунок 5.15. Последовательность анимации бега прыгающего робота и ноги робота-гепарда. *а)* Бег прыгающего робота, *б)* целой ноги с гибкой плюсовой, *в)* с гибким элементом в звене, имитирующем портняжную мышцу, и *г)* гибким элементом в звене, имитирующем малую берцовую кость

Результаты моделирования прыжка на месте для редуцированного механизма галопирующего робота представлены на рисунке 5.14. Датчик вертикального положения крепится к центру контактной сферы, которая соединена с точкой контакта, по этой причине он показывает значения выше нуля в самом низком положении удара на графике высоты прыжка. В контексте данной работы наиболее интересной характеристикой является энергопотребление. Для выбранного примера пиковое значение потребляемой мощности составляет ориентировочно 12 Вт за удар. Как видно из графика, мощность сервопривода колеблется около нуля, так как он отвечает только за изменение фазы.

### 5.3.6 Моделирование полного механизма ноги

При разработке полного механизма ноги робота-гепарда необходимо произвести выбор расположения и характеристик гибкого элемента, основываясь на динамических параметрах: усреднённой высоте прыжка, скорости горизонтального бега и показателях энергозатрат.

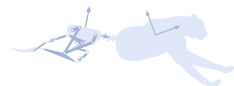


При проектировании механизма целой ноги к структуре мини-таур были добавлены только четыре звена. Поскольку звено  $6\text{ CM}$  выполняет роль большой берцовой кости, оно должно быть жёстким, чтобы удерживать внешние нагрузки. Следовательно, только три звена остаются кандидатами на замену гибкими элементами – либо на звено переменной длины с пружиной, либо на гибкий элемент, работающий на изгиб. Проводя параллели с анатомией гепарда, рассмотрим следующие места расположения ЗПД (рисунок 5.11):

1. Звено  $MF$ , имитирующее плюсну (metatarsal).
2. Звено  $DC$ , имитирующее работу портняжной мышцы (sartorius).
3. Звено  $HN$ , имитирующее работу малой берцовой кости (fibula).

**Гибкий элемент в metatarsal.** Симуляция проводилась для широкого диапазона фаз: от  $-2,9$  рад до  $2,9$  рад, с различными коэффициентами жёсткости пружины и демпфирования. Размер шага  $0,26$  рад. Основная масса сосредоточена в теле робота, общая масса робота равна  $0,875$  кг. Проанализируем результаты моделирования целой ноги робота-гепарда с изгибаемым элементом, расположенном в плюсне (рисунок 5.15, б). Симуляция проводилась для различных материалов, таких как нейлон (модуль Юнга  $E = 2$  ГПа, коэффициент Пуассона  $\nu = 0,39$ ), резина ( $E = 0,5$  ГПа,  $\nu = 0,48$ ) и стекловолокно ( $E = 72$  ГПа,  $\nu = 0,21$ ). Соответствующие результаты моделирования показаны на рисунке 5.16. Результаты аналогичны прыгающему роботу: узкая область прыжков посередине, есть области бега вперёд и назад (но они инвертированы), высота прыжка практически не связана с фазой. Однако, поскольку в системе недостаточно гибкости, высота прыжка очень мала, а КПЗ для гибкой плюсны слишком велик.

Это приводит к выводу, что использование одного гибкого звена без дополнительных пружин является недостаточным.



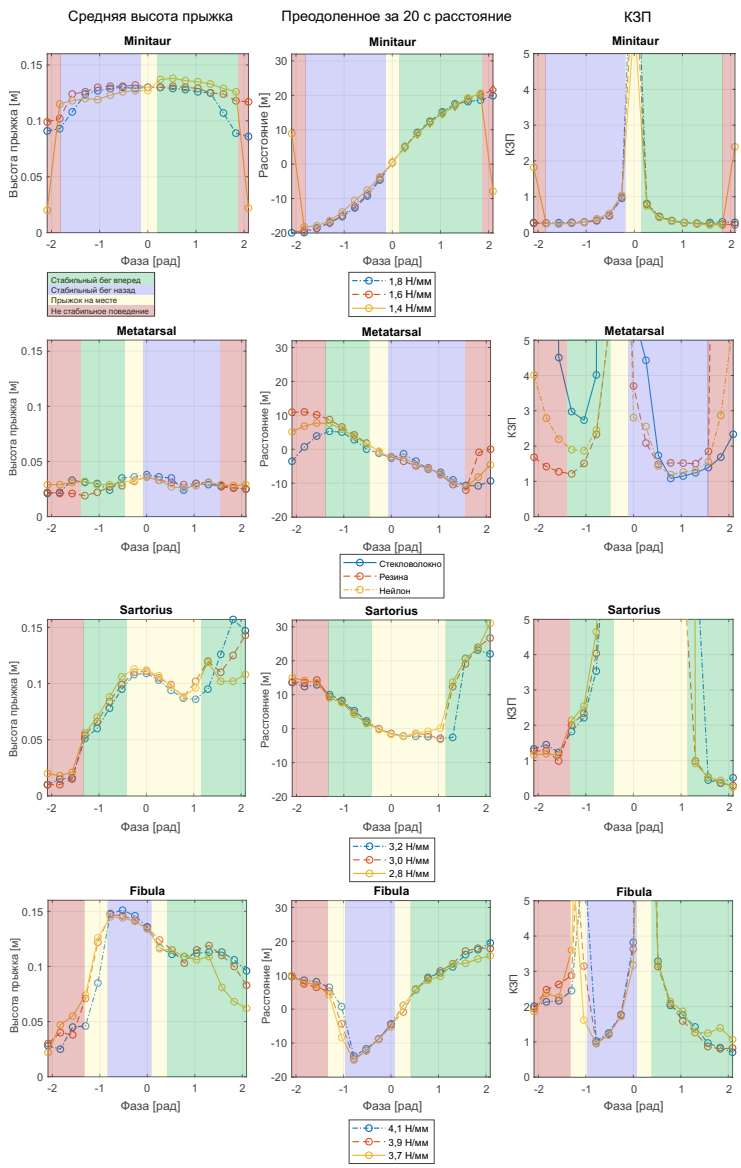
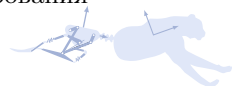


Рисунок 5.16. Результаты моделирования



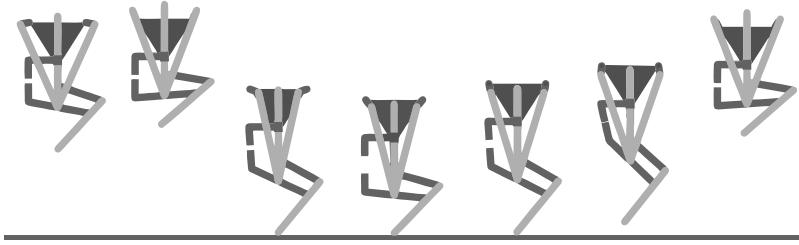
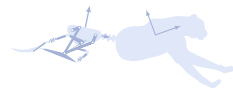


Рисунок 5.17. Моделирование прыжка на месте полной ноги робота-гепарда

**Гибкий элемент в *sartorius*.** Теперь рассмотрим работу ноги со звеном переменной длины  $DC$ , которое имитирует портняжную мышцу (рисунок 5.15, *в*). Симуляция аналогично проводилась в диапазоне коэффициентов жёсткости пружины от  $K_1 = 3,2$  Н/мм до  $K_3 = 2,8$  Н/мм, в том числе  $K_2 = 3,0$  Н/мм, с тем же коэффициентом демпфирования  $\beta = 1$  Н/(м/с). Длина пружины в нормальном состоянии 3,5 см. Соответствующие результаты симуляции показаны на рисунке 5.16. Поведение значительно отличается по сравнению с прыгающим роботом и целой ноги с гибким элементом, расположенном в *metatarsal*. Имеется широкая область прыжка на месте с резким переходом в режим бега. Средняя величина высоты прыжка составляет приблизительно 20 см, в то время как в целом она зависит от жёсткости пружины: чем жёстче пружина, тем больше величина. При фазе 1,04 рад осуществляется прыжок на месте, однако при увеличении фазы до 1,3 рад робот резко начинает бежать со скоростью 0,7 м/с. Максимальная горизонтальная скорость составляет 1,5 м/с при фазе 2,09 рад, что намного быстрее по сравнению с изначальным прыгающим роботом.

Основным недостатком является отсутствие управляемости, свойственной ранее рассмотренному прыгающему роботу. Если необходимо передвигаться с меньшей скоростью, то должен быть изменён знак фазы, так как вместо движения назад при такой конфигурации механизм ноги движется вперёд с другим видом бега.



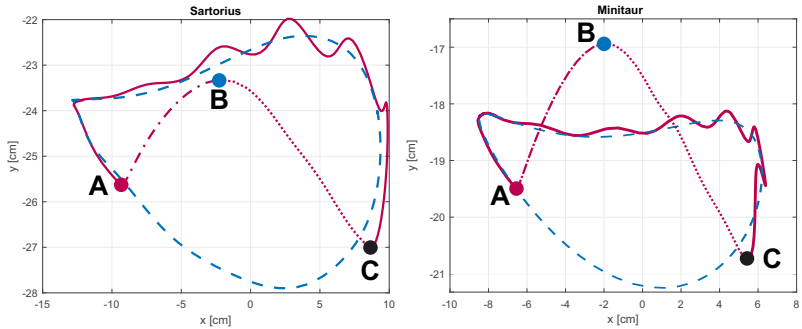


Рисунок 5.18. Траектория точки контакта с полом при беге. Траектории точки контакта с землей, при расположении ЗПД в *sartorius* и редуцированного механизма ноги в системе координат относительно тел роботов

На рисунке 5.18 показаны траектории точки контакта с землёй для целой ноги с гибким элементом в *sartorius* и механизма прыгающего робота в системе координат относительно тела роботов. Синие пунктирные линии показывают траектории без моделирования контакта аналогично на рисунку 5.11. Красные линии - траектории точек контакта при моделировании соударения. Точка *A* обозначает начальный момент контакта, точка *B* обозначает отрыв от земли. Штрих-пунктирная кривая *AB* описывает сжатие пружины, пунктирная кривая *BC* означает разжатие пружины. Чтобы получить устойчивое поведение при беге, траектория должна быть аналогична изображённым траекториям.



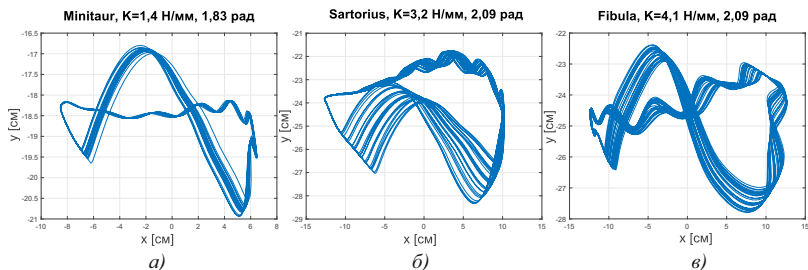


Рисунок 5.19. Траектория точки контакта с полом при беге. а) Прыгающий робот, б) целая нога робота-гепарда с ЗПД в *sartorius* и в) целая нога робота-гепарда с ЗПД в *fibula*

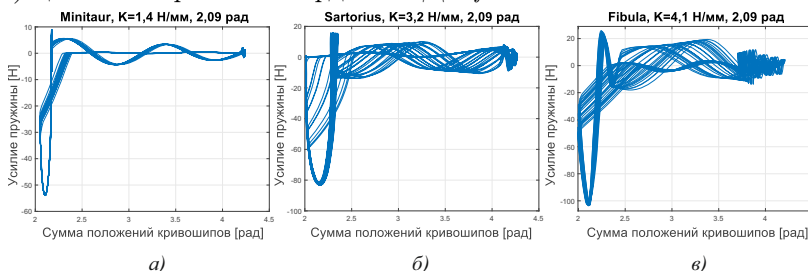
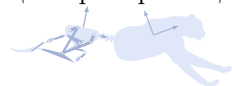


Рисунок 5.20. Зависимость силы пружины от суммы положений кривошипов. а) Прыгающий робот, б) целая нога робота-гепарда с ЗПД в *sartorius* и в) целая нога робота-гепарда с ЗПД в *fibula*

**Гибкий элемент в *fibula*.** Моделирование ноги со звеном переменной длины  $HN$ , имитирующее малую берцовую кость, представлено на рисунке 5.15, г. Моделирование было проведено в диапазоне коэффициентов жёсткости от  $K_1 = 4,1$  Н/мм до  $K_3 = 3,7$  Н/мм, в том числе  $K_2 = 3,9$  Н/мм, и коэффициентом демпфирования  $\beta = 1$  Н/(м/с). Длина пружины в покое 3,5 см. Соответствующие результаты симуляции приведены на рисунке 5.16. Характеристики бега лучше, чем в случае гибкой плюсны, но хуже, чем у изначального прыгающего робота и целой ноги при ЗПД в портняжной мышце. Управляемость лучше по сравнению с предыдущим примером. Од-



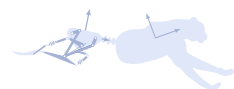


нако нет никаких преимуществ по сравнению с изначальным прыгающим механизмом. Из результатов моделирования можно сделать вывод, что звено переменной длины должно быть установлено на место звена  $DC$  (*sartorius*), а не на  $HN$  (*fibula*).

На рисунке 5.19 изображены траектории точки контакта при беге для прыгающего робота и целой ноги со ЗПД в звене, имитирующей портняжную мышцу *sartorius*, и в звене, имитирующей малую берцовую кость *fibula*. Прыгающий робот является образцом. Все три графика демонстрируют высокую повторяемость выходной траектории.

При установке ЗПД на место  $DC$  при соударении ноги с землёй пружина, вытягиваясь, будет накапливать потенциальную энергию при падении робота, а после выплёскивать ее в кинетическую энергию робота. Путём настройки коэффициента жёсткости пружины и скорости вращения кривошипов было получено стабильное прыгающее состояние на месте (рисунок 5.17).

Проведённый динамический анализ с исследованием расположения гибких элементов, их влияния на скорость и на коэффициент затрат энергии на перемещение необходим для создания энергоэффективного четырехногого галопирующего робота-гепарда, который использует гибкие элементы для рекуперации энергии при контакте с землёй, что позволяет снизить энергопотребление для обеспечения большего запаса хода.



# Приложение А.

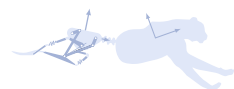
## Задания для курсового проекта

**Целью** курсового проекта является проведение упрощенного процесса проектирования робототехнического устройства, начиная от структурного анализа, заканчивая подготовкой эскизной документации.

Для выполнения поставленной цели необходимо выполнить следующие **задачи**:

1. Структурный анализ механизма
  - Разбить на группы Ассура и представить класс механизма;
  - Представить расчёт степени свободы.
2. Кинематический анализ механизма.
  - Вывести в матричном виде выражение координат выходной точки механизма;
  - Аналогично вывести для неё выражение скорости.
3. Имитационное моделирование абстрактной геометрии
  - Решение задачи стабилизации скорости вращения кривошипов. Управление по ошибке скорости;

- Построение графиков положения и скоростей выходных точек механизма с помощью блоков *Transform Sensor*. Построение аналогичных графиков с использованием полученных выражений при выполнении кинематического анализа. Сравнение графиков;
  - Моделирование контакта с поверхностью пола. Снятие значений сил нормальной реакции взаимодействия и сил трения в точках контакта. Построение сил реакций во всех кинематических парах механизма одной ноги, а также построение графика момента на входном звене.
4. Конструирование деталей робота. По значениям сил реакций в кинематических парах оптимизировать форму и размер деталей, подобрать подшипники на сайте SKF (или аналог). Проработать подшипниковые узлы в кинематических парах.
  5. Имитационное моделирование проработанной конструкции с моделированием контакта с поверхностью пола для верификации проработанной конструкции.
  6. Разработка эскизного чертежа всего робота и чертежа выходного звена.
  7. Написание пояснительной записки по ГОСТ 7.32.



## Варианты проектов

### А.1 Шагающий робот на основе механизма Кланна

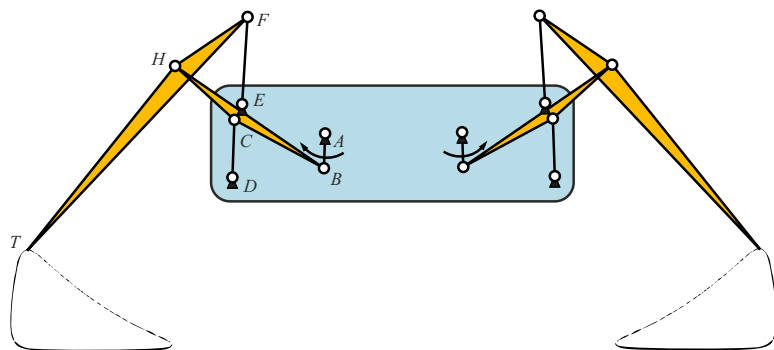


Рисунок 5.21. Шагающий робот на основе механизма Кланна

На Рис. 5.21 изображён плоский шагающий робот, чьи ноги реализованы на основе механизма Кланна. Механизм обладает одной степенью свободы. Передняя и задняя ноги зеркальны. Входным звеном является  $AB$ , выходным  $FHT$ . Кривая линия отображает траекторию перемещения точки  $T$  – контакта с полом в системе координат тела робота. Масса тела робота  $M=10$  кг, длина звена  $TF=100$  мм.



## А.2 Прыгающий робот Минитаур

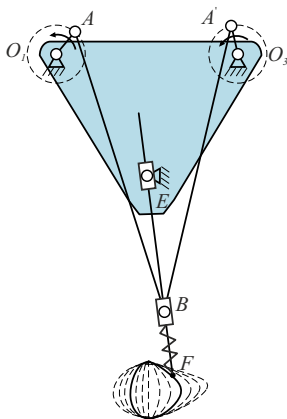


Рисунок 5.22. Прыгающий робот Минитаур

На Рис. 5.22 изображён плоский прыгающий робот, реализованный на основе механизма «минитаур». Данный механизм был использован впервые при разработке робота Ghost Robotics Minitaur<sup>2</sup>. Принципиальным отличием робота, изображённого на Рис. 5.22, является принцип приведения в движение входных звеньев  $O_1A$  и  $O_3A'$ , которые испытывают постоянное вращение, а не импульсное движение, как в Ghost Robotics Minitaur. При совпадении частоты вращения кривошипов с частотой колебаний пружины робот способен войти в состояние резонанса для обеспечения динамической локомоции.

Механизм обладает тремя степенями свободы: 2 активных и 1 пассивная. Механизм зеркален относительно вертикальной оси. Тело  $O_1O_3E$  способно перемещаться по вертикальной и горизонтальной оси относительно мировой системы координат. Кривые линии показывают перемещение точки взаимодействия с полом  $F$ . Масса тела робота  $M=1$  кг, длина звена  $AB = 20$  см.

<sup>2</sup>Kenneally G., De A., Koditschek D. E. Design principles for a family of direct-drive legged robots //IEEE Robotics and Automation Letters. – 2016. – Т. 1. – №. 2. – С. 900-907.



### А.3 Шагающий робот на основе механизма Янсена

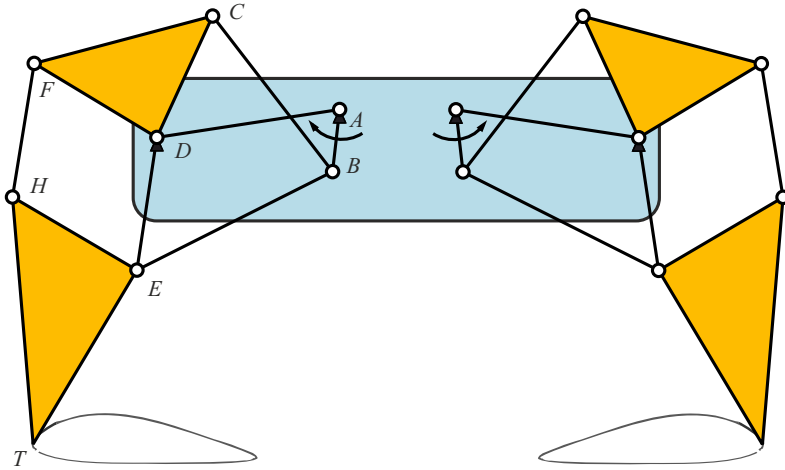


Рисунок 5.23. Шагающий робот на основе механизма Янсена

На Рис. 5.23 изображён плоский шагающий робот, чьи ноги реализованы на основе механизма Янсена. Механизм обладает одной степенью свободы. Передняя и задняя ноги зеркальны. Входным звеном является  $AB$ , выходным  $ETH$ . Кривая линия отображает траекторию перемещения точки  $T$  контакта с полом в системе координат тела робота. Масса тела робота  $M=10$  кг.



## А.4 Неполноприводное захватное устройство

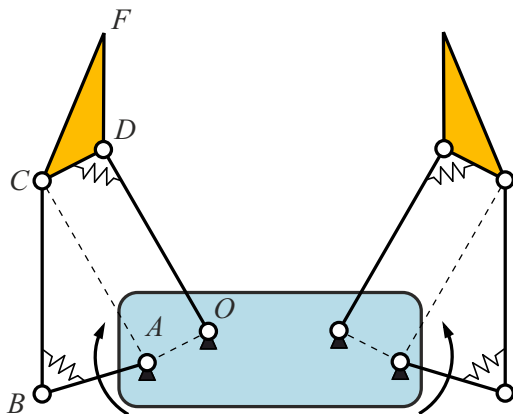


Рисунок 5.24. Неполноприводное захватное устройство

На Рис. 5.24 изображён плоский захват с неполноприводными механизмами пальцев.  $OD \parallel AC$ ,  $AO \parallel CD$ . Гибкие элементы в сочленениях  $B$  и  $D$  необходимы для стабилизации работы механизма при выполнении адаптивного захвата.



## А.5 Стопоходящая машина Чебышева

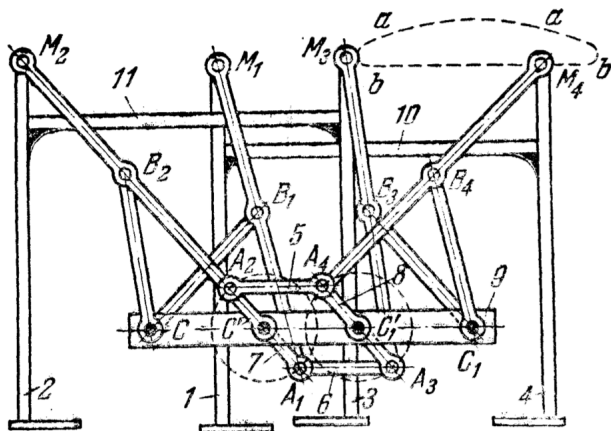


Рисунок 5.25. Стопоходящая машина Чебышева

На Рис. 5.25 изображён плоский шагающий робот, чьи ноги реализованы на основе лямбда-механизма Чебышева. Механизм обладает одной степенью свободы. Входными звеньями являются  $A_1A_2$  и  $A_3A_4$ , выходным – звенья 1, 2, 3 и 4. Масса тела робота  $9 M=10$  кг. Длины звеньев механизма удовлетворяют условиям:

$$\begin{aligned}
 A_1B_1 = B_1C = B_1M_1 = A_2B_2 = B_2C = B_2M_2 = A_3B_3 = \\
 = B_3C_1 = B_3M_3 = A_4B_4 = B_4C_1 = B_4M_4 = 1; \\
 A_1C' = A_2C' = A_3C_1' = A_4C_1' = 0.355; \\
 CC' = C_1C_1' = -0.785; A_2A_4 = A_1A_3 = C'C_1' = 0.634
 \end{aligned}$$

Изображение взято из справочника механизмов И.И. Артоблевского [29].





## А.6 3R робот

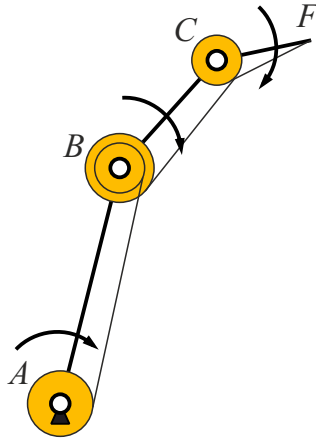


Рисунок 5.26. Робот с тремя вращательными сочленениями

На Рис. 5.26 изображён плоский робот-манипулятор с тремя вращательными сочленениями. Все три двигателя расположены в основании робота. Робот должен двигаться по сложной траектории:

$$x = A \cdot \sin(f \cdot t + \phi) + b$$

$$y = A \cdot \sin(f \cdot t + \phi) + b$$

где  $A$  – амплитуда,  $f$  – частота,  $\phi$  – фаза,  $b$  – сдвиг.



## А.7 Механизм Хайна с гибким звеном

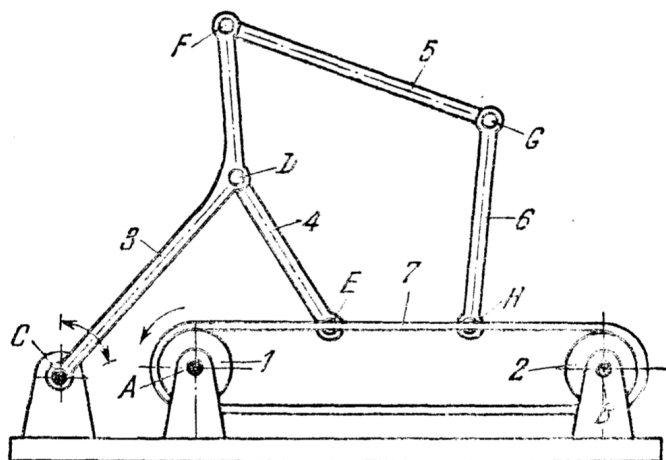
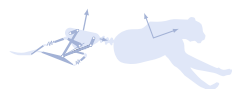


Рисунок 5.27. Механизм Хайна с гибким звеном

Радиусы шкивов 1 и 2 равны. Шкив 1, вращающийся вокруг неподвижной оси  $A$ , гибким звеном 7 приводит во вращение вокруг неподвижной оси  $B$  шкив 2. Звено 3, имеющее форму коленчатого рычага, вращается вокруг неподвижной оси  $C$ , входя в вращательные пары  $F$  и  $D$  со звеном 5 и звеном 4, входящим в вращательную пару  $E$  с гибким элементом 7. Звено 6 входит во вращательную пару  $H$  с гибким звеном 7 и вращательную пару  $G$  со звеном 5. При повороте шкива 1 вокруг оси  $A$  звенья 4, 5 и 6 совершают сложные движения, а звено 3 качается вокруг оси  $C$ .

Изображение взято из справочника механизмов И.И. Артоблевского [29].



## А.8 Кулисно-рычажный механизм убирающего шасси самолёта

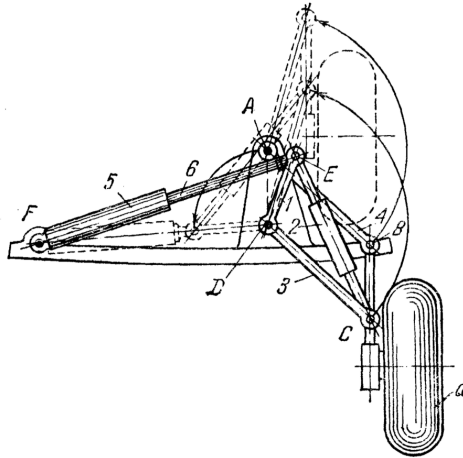
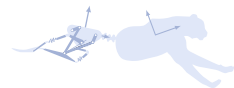


Рисунок 5.28. Кулисно-рычажный механизм убирающего шасси самолёта

Длины звеньев механизма удовлетворяют условиям:  $AB = DC$  и  $AD = BC$ . Таким образом, фигура  $ABCD$  является механизмом параллелограмма, с шатуном  $BC$  которого жёстко связано колесо  $a$ . Звено 1 вращается вместе со звеном 3 вокруг неподвижной оси  $D$  – рамы самолёта. Между точками  $E$  и  $C$  установлено масляное амортизационное устройство 2. Таким образом, система  $EDC$  вращается вокруг общей оси  $D$ . В точке  $E$  с этой системой входит шток 6 цилиндра подъёма 5, вращающегося вокруг неподвижной оси  $F$  рамы самолёта. При движении штока 6 внутрь цилиндра подъёма 5 звенья 1, 3 и 4 повернутся в направлениях, указанных стрелками, и механизм займёт показанное штриховой линией положение, обеспечивающее уборку шасси самолёта.

Изображение взято из справочника механизмов И.И. Артоблевского [29].



## А.9 Энергоэффективный галопирующий робот с гибкой спиной

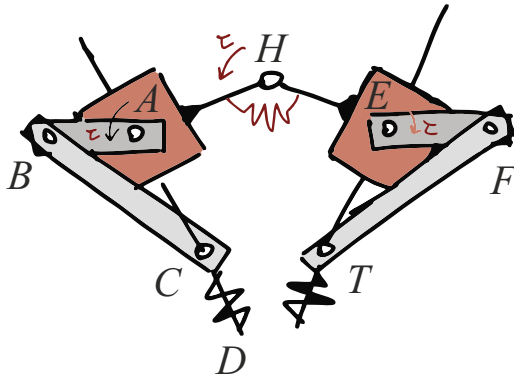


Рисунок 5.29. Энергоэффективный галопирующий робот с гибкой спиной

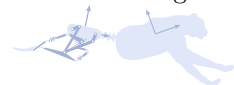
Данный проект направлен на проектирование и моделирование двуногого галопирующего робота с гибким позвоночником. Ноги должны быть способны только прыгать на месте, в то время как горизонтальное перемещение обеспечивается только гибким позвоночником. Ваша цель состоит в том, чтобы найти стабильное и энергоэффективное галопирующее поведение. Цель модели-понять, как привести в действие гибкий позвоночник и какие характеристики упругих элементов являются наилучшими для того, чтобы создать наиболее энергоэффективное и производительное устройство.



# Литература

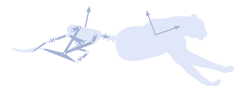
- [1] Г. А. Тимофеев. *Теория механизмов и механика машин: учебник для вузов*. Москва: Издательство МГТУ им. Н.Э. Баумана, 2017.
- [2] Georg Pelz and Rachel Waddington. *Mechatronic systems*. Wiley Online Library, 2004.
- [3] MathWorks Steve Miller. What is simscape? <https://www.mathworks.com/videos/simscape-overview-61215.html>howpublished, 2020. Дата обращения: 10.03.2020.
- [4] MathWorks Steve Miller. What is simscape electrical? <https://www.mathworks.com/products/simscape-electrical.html>, 2020. Дата обращения: 10.03.2020.
- [5] MathWorks Steve Miller. What is simscape driveline? <https://www.mathworks.com/videos/simscape-driveline-overview-117981.html>, 2020. Дата обращения: 10.03.2020.
- [6] MathWorks Steve Miller. What is simscape multibody? <https://www.mathworks.com/videos/simscape-multibody-overview-117986.html>, 2020. Дата обращения: 10.03.2020.
- [7] MathWorks Steve Miller. What is simscape fluids? <https://www.mathworks.com/videos/simscape-fluids-overview-117985.html>, 2020. Дата обращения: 10.03.2020.

- [8] Юрий Алексеевич Семенов and Надежда Сергеевна Семенова. Структурный анализ механизмов. *Теория механизмов и машин*, 1(2):3–14, 2003.
- [9] RA Hyde and J Wendlandt. Tool-supported mechatronic system design. In *2008 34th Annual Conference of IEEE Industrial Electronics*, pages 1674–1679. IEEE, 2008.
- [10] Sebastian Castro and Connell Dsouza. Matlab and simulink robotics arena. <https://www.mathworks.com/academia/student-competitions/roboticsarena.html>, 2020. Дата обращения: 5.03.2020.
- [11] Carlos Rengifo, Yannick Aoustin, Christine Chevallereau, and Franck Plestan. A penalty-based approach for contact forces computation in bipedal robots. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 121–127. IEEE, 2009.
- [12] Evangelos Kapros and Maria Koutsombogera. *Designing for the User Experience in Learning Systems*. Springer, 2018.
- [13] Stephan Brunßen, Stefan Hieber, and Barbara Wohlmuth. Contact dynamics with lagrange multipliers. In *IUTAM Symposium on Computational Methods in Contact Mechanics*, pages 17–32. Springer, 2007.
- [14] Neville Hogan. Impedance control: An approach to manipulation. In *1984 American control conference*, pages 304–313. IEEE, 1984.
- [15] Kevin M Lynch and Frank C Park. *Modern Robotics*. Cambridge University Press, 2017.
- [16] Ivan I Borisov, Sergey A Kolyubin, and Alexei A Bobtsov. Static force analysis of a finger mechanism for a versatile gripper. In *International Conference Cyber-Physical Systems and Control (CPSC 2019)*, pages 308–324, 2019.
- [17] Ivan I Borisov, Ivan Kulagin, Anastasiya Larkina, Artem Egorov, Sergey A Kolyubin, and Stefano Stramigioli. Study on elastic elements allocation for energy-efficient robotic cheetah leg. In



2019 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1696 – 1701. IEEE, 2019.

- [18] Help Center. Simscape multibody link. <https://www.mathworks.com/help/physmod/smlink/index.html>, 2020. Дата обращения: 15.03.2020.
- [19] Help Center. Cad translation. <https://www.mathworks.com/help/physmod/smlink/ug/cad-translation.html>, 2020. Дата обращения: 15.03.2020.
- [20] Help Center. Mates and joints. <https://www.mathworks.com/help/physmod/smlink/ref/mates-and-joints.html>, 2020. Дата обращения: 15.03.2020.
- [21] И. И. Борисов. *Разработка адаптивных манипуляционных и энергоэффективных локомоционных роботов*. Университет ИТМО, 2019.
- [22] Ivan I Borisov, Oleg I Borisov, Daniil S Monich, Taniel A Dodashvili, and Sergey A Kolyubin. Novel optimization approach to development of digit mechanism for bio-inspired prosthetic hand. In *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, pages 726–731. IEEE, 2018.
- [23] Ivan I Borisov, Oleg I Borisov, Vladislav S Gromov, Sergey M Vlasov, Dmitrii Dobriborsci, and Sergey A Kolyubin. Design of versatile gripper with robust control. *IFAC-PapersOnLine*, 51(22):56–61, 2018.
- [24] Ivan I Borisov, Oleg I Borisov, Vladislav S Gromov, Sergey M Vlasov, and Sergey A Kolyubin. Versatile gripper as key part for smart factory. In *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, pages 476–481. IEEE, 2018.
- [25] И. И. Борисов and С. А. Колюбин. Структурный и геометрический синтез механизма пальца универсального захватного устройства, осуществляющего точный щипковый и силовой обхватывающий захваты. *Вестник машиностроения*, (11):14–19, 2019.



- [26] Gavin Kenneally, Avik De, and Daniel E Koditschek. Design principles for a family of direct-drive legged robots. *IEEE Robotics and Automation Letters*, 1(2):900–907, 2016.
- [27] Martijn Snippe. Cheetah robot leg mechanism: analysis, design and cost of transport. 2017.
- [28] Sangok Seok, Albert Wang, Meng Yee Chuah, David Otten, Jeffrey Lang, and Sangbae Kim. Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot. In *2013 IEEE International Conference on Robotics and Automation*, pages 3307–3312. IEEE, 2013.
- [29] И.И. Артоболевский. *Механизмы в современной технике. В 7 томах. Том 1. Элементы механизмов. Простейшие рычажные и шарнирно-рычажные механизмы.* М.: Наука, 1979.





Борисов Иван Игоревич  
Колюбин Сергей Алексеевич

## **Имитационное моделирование мехатронных систем**

**Учебно-методическое пособие**

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Подписано к печати

Заказ №

Тираж

Отпечатано на ризографе

**Редакционно-издательский отдел**  
**Университета ИТМО**  
197101, Санкт-Петербург, Кронверкский пр., 49