

Санкт-Петербургский государственный университет  
Факультет прикладной математики - процессов управления

Ефимова Полина Александровна

Выпускная квалификационная работа аспиранта

**Билатеральное управление космическими роботами при  
больших задержках в передаче сигнала управления**

Направление подготовки —  
01.06.01 «Математика и механика»

Научный руководитель:  
д.т.н., профессор  
Кулаков Феликс Михайлович

Рецензент:  
д.т.н., профессор  
Пенский Олег Геннадьевич

Санкт-Петербург — 2019

## Оглавление

	Стр.
<b>Введение</b> . . . . .	<b>3</b>
<b>Глава 1. Билатеральное управление роботами при наличии задержек в передаче сигнала управления</b> . . . . .	<b>8</b>
1.1 История развития телеробототехники . . . . .	8
1.2 Система билатерального управления роботами . . . . .	10
1.3 Схемы управления для билатеральных систем с задержкой . . . . .	13
1.3.1 Схемы, основанные на пассивности . . . . .	13
1.3.2 Схемы, основанные на прогнозировании . . . . .	16
1.3.3 Схемы, основанные на скользящем режиме управления . . . . .	19
1.3.4 Перспективы исследований . . . . .	20
1.4 Дистанционное управление космическим роботом методом обучения . . . . .	22
1.5 Постановка задачи . . . . .	29
<b>Глава 2. Математическое моделирование исполнительной системы</b> . . . . .	<b>30</b>
2.1 Кинематические схемы . . . . .	31
2.2 Прямая позиционная задача . . . . .	33
2.3 Обратная позиционная задача . . . . .	37
2.4 Прямая задача кинематики о скорости . . . . .	44
2.5 Прямая задача динамики . . . . .	52
2.6 Метод адаптации к изменениям рабочей среды . . . . .	55
<b>Глава 3. Программная реализация</b> . . . . .	<b>59</b>
3.1 Кинематическое моделирование . . . . .	59
3.2 Динамическое моделирование . . . . .	63
<b>Заключение</b> . . . . .	<b>68</b>

## Введение

Активное освоение космического пространства подразумевает использование вспомогательных робототехнических систем. Применение космических роботов способствует повышению точности и надежности выполняемой работы, а также позволяет проводить исследования на больших дистанциях от Земли. Однако, при проведении работ в экстремальных условиях, зачастую приходится иметь дело с плохо упорядоченной и недетерминированной средой, что требует постоянного уточнения данных о среде и корректировки принятых решений. К сожалению, степень автономности современных роботов пока не способна обеспечить выполнение таких трудоемких задач, что не позволяет исключить человека-оператора из контура управления. Более того, при работе в космосе возникают значительные задержки в передаче сигналов управления и обратной связи. В связи с этим, возникает актуальная задача построения системы дистанционного управления космическим роботом, слабо зависящей от задержки в передаче управляющего сигнала.

В качестве способа дистанционного управления роботом чаще всего рассматривают систему «ведущий–ведомый» (master–slave system). В данной системе имеется два кинематически подобных робота, один из которых называется «ведомым», а другой соответственно «ведущим». С «ведущим» роботом непосредственно взаимодействует человек–оператор, задавая ему необходимое положение рабочего инструмента либо необходимую траекторию его движения. «Ведомый» робот повторяет все движения «ведущего» робота, взаимодействуя с предметами рабочей среды.

Значительное внимание исследователей привлекла система дистанционного управления роботом с силовой обратной связью, или, другими словами, система билатерального управления. Силовая обратная связь позволяет человеку–оператору чувствовать силу реакции объектов, с которыми взаимодействует «ведомый» робот, а также самому передавать необходимые силы на объекты с помощью билатеральной системы. Такой способ управления обеспечивает более комфортное управление для человека, а также позволяет более гибко манипулировать объектами в рабочей среде имеющими связи. Совершенная билатеральная система должна быть построена таким образом, чтобы человек–оператор, взаимодействуя с объектами среды посредством роботов,

мог работать таким образом, будто он взаимодействует с объектами непосредственно. Обычно, в этом случае, говорят, что такая система обладает хорошей транспарентностью.

Существенной проблемой на пути создания совершенной билатеральной системы является влияние временной задержки передачи сигналов на устойчивость и транспарентность системы. Однако следует отметить, что задержка не является единственной причиной нестабильности билатеральной системы. Также могут влиять такие факторы как шум, внешние помехи, масштабирование силы и т. п.

При дистанционном управлении космическими роботами вопрос влияния временной задержки сигнала играет ключевую роль. Исследователи из России (ЦНИИ РТК), США и Германии (DLR) активно ищут подходы к построению системы билатерального управления, слабо зависящей от задержек в передаче сигнала управления и обратной связи. Предлагаемые на данный момент подходы можно разделить на три группы: методы основанные на пассивности, методы основанные на прогнозирующих дисплеях и методы скользящего режима управления.

Метод основанный на пассивности базируется на том, что мощность, развиваемая рабочим инструментом «ведомого» манипулятора, не должна превышать мощность, развиваемой рукой человека, перемещающего инструмент «ведущего» робота. Это накладывает некоторые ограничения на функциональные возможности системы билатерального управления, однако, в соответствии с теорией рассеяния энергии, такой подход гарантирует выполнение устойчивости движения. Но, к сожалению, требование транспарентности в этом случае плохо выполняется.

Второй подход предполагает использование прогнозирующего управления. Оно базируется на использовании компьютерных моделей, с помощью которых осуществляется прогноз поведения «ведомого» робота и его внешней среды. В рамках данного метода достигается более качественное выполнение требования транспарентности.

Третий подход основан на использовании скользящего режима управления. Сложность практической реализации этого подхода обусловлена необходимостью функционирования управляющей аппаратуры и механической части робота в весьма тяжелых режимах часто изменяемого знака управлений, имеющих к тому же максимально допустимые значения. Это приводит к появлению

больших ускорений элементов конструкции, а, следовательно, появлением больших реактивных сил.

Вышеуказанные методы могут обеспечивать устойчивость и транспарентность билатеральной системы при временной задержке не более 3 секунд, что существенно ограничивает расстояние, на которое можно отправить космического робота. В связи с этим, возникает актуальный вопрос построения такой структуры билатерального управления, которая обеспечивала бы возможность дистанционного управления на больших расстояниях.

В качестве подхода к решению данного вопроса, профессор СПбГУ Кулаков Ф. М. предложил метод дистанционного управления роботом с помощью обучения будущим действиям. В данном подходе процесс управления роботом разделяется на несколько этапов. На первом этапе «ведомый» робот, с помощью датчиков, собирает информацию об объектах рабочей среды и передает данные в центр управления. На втором этапе, в центре управления, на основе полученных параметров, формируется компьютерная или натурная модель внешней среды робота. Третий этап — это этап формирования программных траекторий на основе модели внешней среды и вспомогательного наземного робота. Если используется натурная модель рабочей среды, то в качестве вспомогательного робота можно использовать промышленный наземный робот, инструмент которого будет точно повторять инструмент космического робота. Если же в наличие имеется только компьютерная модель, то в качестве вспомогательного робота можно использовать робота-перчатку, который будет помогать человеку-оператору «чувствовать» и взаимодействовать с объектами смоделированной среды. То есть, на третьем этапе человек-оператор, наблюдая объекты модельной среды через очки дополненной реальности или просто наблюдая объекты натурной модели, проводит необходимую операцию с помощью соответствующего вспомогательного робота. В результате данного этапа формируются шаблоны операции и алгоритм действий передается исполнительному механизму. На четвертом этапе робот, анализируя изменения рабочей среды, адаптирует программу действий и выполняет ее, отправляя отчет о проделанной работе в центр управления. Этапы повторяются до полного выполнения поставленной задачи.

При успешной реализации данной схемы управления, будет обеспечена возможность проведения различных типов работ на больших расстояниях от Земли. К таким работам можно отнести построение космических баз на

поверхностях различных планет, спутников и других небесных тел, ремонт функционирующих установок по добыче ресурсов с астероидов, обслуживание орбитальных космических станций, искусственных спутников и т. п. Предлагаемый подход снимает ограничения на временную задержку, так как устойчивость и транспарентность системы в таком случае в основном зависит от точности построенных моделей среды и робота.

Целью и задачей данной работы является разработка математических моделей элементов системы дистанционного управления космическими роботами методом обучения показом, функционирующими в недетерминированной внешней среде, а также подготовка соответствующих программ для верификации результатов. К ключевым задачам можно отнести:

- Построение математической модели исполнительного механизма, в качестве которой будет рассмотрен космический робот со свободнолетающим основанием.
- Построение математической модели вспомогательного наземного робота, в качестве которого будет рассматриваться промышленный манипулятор FANUC M-20iA.
- Решение и моделирование обратной задачи кинематики для наземного и космического робота.
- Динамический анализ исполнительных механизмов.
- Разработка алгоритма коррекции поведения робота при изменениях окружающей среды.
- Верификация результатов на компьютерных моделях.

При построении математических моделей элементов системы рассматриваемого способа дистанционного управления необходимо учитывать следующие особенности:

- На наземный «ведущий» робот и на космический «ведомый» робот воздействует разная гравитация.
- Если космический робот предназначен для обслуживания орбитальных станций и искусственных спутников, то обычно «ведомый» робот рассматривается как система, состоящая из свободнолетающего основания и закрепленного на нем манипулятора или системы манипуляторов. Это влечет за собой дополнительные степени подвижности и, как следствие, дополнительные уравнения движения.

- За время второго этапа управления и передачи алгоритма действий космическому роботу рабочая среда может измениться. «Ведомый» робот должен быть способен адаптировать алгоритм действий полученный с центра управления, учитывая эти изменения.

**Объем и структура работы.** Выпускная квалификационная работа состоит из введения, трёх глав, заключения и одного приложения. Полный объём работы составляет 95 страниц с 28 рисунками. Список литературы содержит 51 наименование.

Во введении обоснована актуальность темы, сформулированы цели и задачи научно-исследовательской работы. Приведено описание структуры и объема работы, а также дан краткий обзор содержания выпускной квалификационной работы по главам.

В первой главе «Билатеральное управление роботами при наличии задержек в передаче сигнала управления» приведен обзор существующих в настоящее время подходов к построению различных схем билатерального управления роботами с задержками в передаче сигналов управления и обратной связи. Рассмотрены особенности каждого метода и возможности их практической реализации. Представлено описание рассматриваемого в данной работе метода дистанционного управления. Сформулирована постановка задачи.

Во второй главе «Математическое моделирование исполнительной системы» приводится математическое описание объектов исследования в виде решения задач кинематики и динамики. Представлены кинематические схемы космического и наземного роботов. Рассматривается метод адаптации космического робота к изменениям рабочей среды.

В третьей главе «Программная реализация» представлены результаты компьютерного моделирования рассматриваемых во второй главе задач.

В заключении приведено описание результатов работы, сделаны главные выводы.

В приложении представлен программный код, результаты которого приведены в четвертой главе. Моделирование проводилось в математическом пакете Matlab.

## Глава 1. Билатеральное управление роботами при наличии задержек в передаче сигнала управления

### 1.1 История развития телеробототехники

В середине XX века начала свое развитие такая область как телеробототехника, которая занимается вопросами дистанционного управления роботами. История телеробототехники берет свое начало с атомных исследований инженера-механика Реймонда Гоерца в 1940-х и 1950-х годах [1]. Работая в Аргоннской национальной лаборатории, Гоерц разработал систему дистанционного управления роботом «ведущий-ведомый» для безопасной обработки опасных материалов из реактора плутония 100-B в Хэнфорде. Первые манипуляторы были медленными и несколько неудобными для человека-оператора, так как они были соединены механически. Данное обстоятельство существенно ограничивало расстояние между оператором и рабочей средой, а также требовало использование кинематически идентичных устройств. К тому же отсутствие тактильных ощущений при управлении существенно усложняло задачу манипулирования деликатными объектами. Тем не менее, данная разработка заложила прочный фундамент для дальнейших исследований в данной области.

В СССР в 1950-е годы, при проведении исследовательских работ с радиоактивными веществами, также использовались копирующие манипуляционные механизмы [2]. Важной особенностью используемых манипуляторов являлась возможность ощущения человеком-оператором сил реакций объектов рабочей среды. Однако близость оператора к опасной зоне и малая грузоподъемность, которая в основном зависела от физических возможностей человека-оператора, наталкивали исследователей на поиск новых методов построения системы телеуправления роботами.

Два более поздних и более усовершенствованных подхода к созданию систем дистанционного управления способных работать на больших расстояниях были разработаны Джоном Кларком из Hughes Aircraft и Ральфом Мошером из General Electric. Разработанный в 1959 году Джоном Кларком «Mobot» уже не имел механической связи «ведущего» и «ведомого» роботов, между ними осуществлялась электронная связь [3]. Более того, в связи с тем, что рассто-



яние уже было достаточно большим, человек–оператор отслеживал рабочую среду «ведомого» робота через телевизионную систему. «Handyman», разработанный Ральфом Мошером в 1958-1959 годах, также имел электрическую связь между роботами, однако «ведущий» робот был выполнен в качестве экзоскелета надеваемого на руки человека–оператора [4]. Электронное оборудование было громоздким и слишком сложным, чтобы быть практичным. Кроме того, стоимость сервомеханизмов и следящих стоек была непозволительной для чего-либо, кроме специализированных проектов.

Однако системы телеуправления основанные на механических передачах имели небольшое преимущество перед системами более позднего поколения — в них практически отсутствовала временная задержка в связи между «ведущим» и «ведомым» роботом. Поэтому, еще с начала 1960-х годов, стал актуальным вопрос влияния временной задержки на стабильность системы дистанционного управления роботами [5]. В 1966 году Уильям Феррелл показал, что задержка даже менее 50 мс может дестабилизировать систему дистанционного управления с силовой обратной связью [6]. Временная задержка может быть вызвана большим физическим расстоянием между «ведущим» и «ведомым» роботом, временем обработки контроллеров, перегрузкой сети и пропускной способностью связи. Несмотря на множество исследований в данной области, вопрос временной задержки в передаче сигналов между роботами остается актуальным до сих пор.

В 1990 году венгерский физик Антал Бэйчи представил первую систему дистанционного управления с силовой обратной связью, использующую кинематически и динамически различные манипуляторы [7]. Данная система была разработана в Jet Propulsion Laboratory для применения в космическом телеуправлении. Она продемонстрировала способность манипулирования объектами ограниченными связями, а также могла запоминать траекторию движения, задаваемую человеком–оператором.

Первым управляемым с Земли космическим роботом был разработанный Германским центром авиации и космонавтики (DLR) и запущенный на орбиту в 1993 году робот ROTEX [8]. Во время миссии «Spacelab-D2» мультисенсорный робот ROTEX выполнил ряд заранее запрограммированных задач, такие как закрытие и открытие разъемных заглушек, сборка конструкции из отдельных частей, а также захват свободноплавающего объекта. Ключевыми технологиями, послужившими успеху ROTEX, являлись: мультисенсорная технология

захвата, локальные концепции управления с сенсорной обратной связью, а также мощное прогнозирующее моделирование с компенсацией задержек.

Дальнейшие космические миссии с телеуправлением, например, такие как ROKVISS (2005, DLR), ARTEMIS (2008, DLR), KONTUR-1 (2011, DLR и ЦНИИ РТК) и KONTUR-2 (2015, DLR и ЦНИИ РТК), были нацелены на отработку методов компенсации временной задержки и проверку ее допустимых границ, а также на повышение точности и качества проводимых роботами космических операций [9].

## 1.2 Система билатерального управления роботами

Система дистанционного управления роботом имеет две стороны: сторону оператора и удаленную сторону (Рис.1). На стороне оператора, или, проще говоря, в центре управления, находятся сам человек–оператор (I), «ведущий» робот (II) и телевизионная система, по которой человек следит за удаленной стороной. На удаленной же стороне находятся «ведомый» робот (III), с установленными на нем датчиками и системой технического зрения, и рабочая среда (IV) [10].

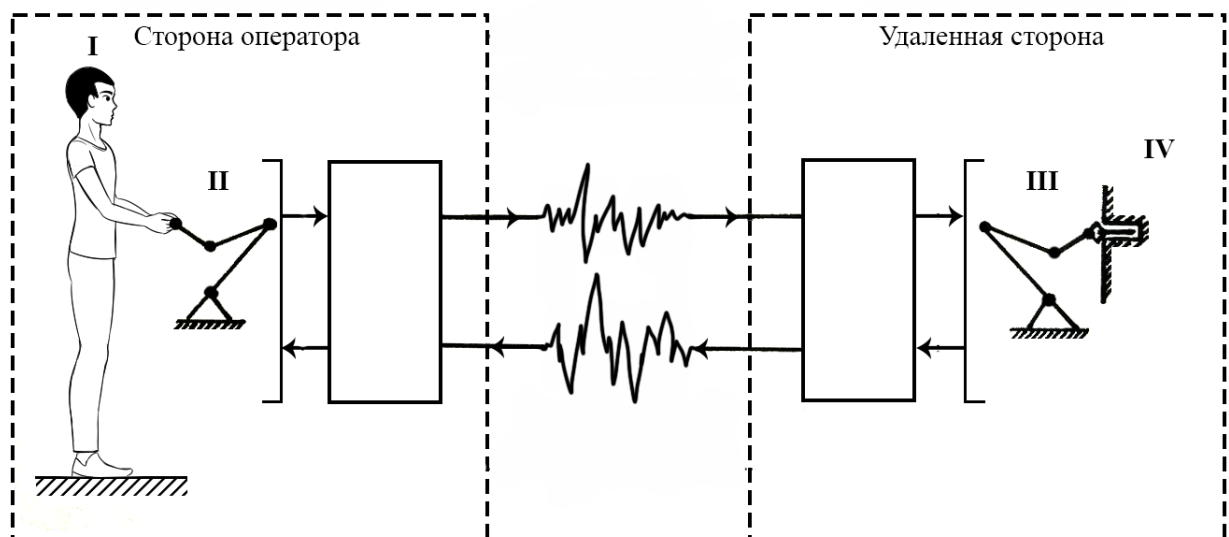


Рисунок 1.1 — Структурная схема системы дистанционного управления

В зависимости от степени принятий решения роботом, можно выделить три основные категории архитектур дистанционного управления: прямое (ручное), совместное и супервизорное [10].

Прямое управление подразумевает отсутствие интеллекта или автономности в системе робота, т.е. действия «ведомого» робота полностью определяются человеком–оператором.

Если же движение «ведомого» робота задается с помощью комбинации из команд оператора и данных от локальной сенсорной обратной связи или автономии, то такая архитектура является совместной. Совместное управление может быть также и в случае, если обратная связь с пользователем исполнена с помощью виртуальной реальности или с помощью других автоматических средств.

В супервизорном управлении команды оператора и обратная связь происходит на более высоком уровне. «Ведомому» роботу в этом случае приходится полагаться на более сильную локальную автономность для уточнения и выполнения задач, человек–оператор в свою очередь выступает в качестве наблюдателя и вмешивается в процесс лишь по необходимости.

Если возникает задача манипулирования объектами, имеющими связи, то при разработке системы дистанционного управления необходимо включить в схему обратную связь по силам и моментам. Системы дистанционного управления роботами с силовой обратной связью называются билатеральными системами управления [11].

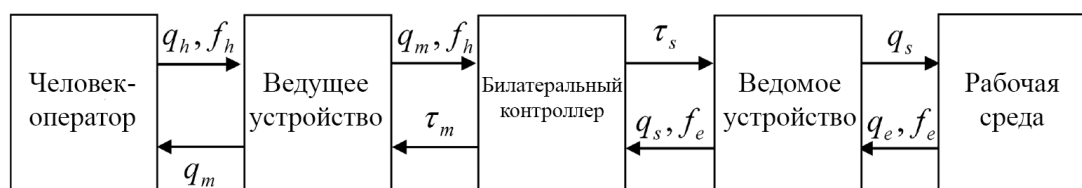


Рисунок 1.2 — Структура билатеральной системы

Схема, представленная на рисунке 1.2, описывает связь и взаимодействие между элементами системы билатерального управления роботом. Здесь  $q_h, q_m, q_s, q_e$  — обобщенные координаты руки человека (human), «ведущего» робота (master), «ведомого» робота (slave) и рабочей среды (environment) соответственно.  $f_h$  и  $f_e$  — силы, передаваемые человеком и средой соответственно.  $\tau_s$  и  $\tau_m$  — управляющие сигналы для «ведомого» и «ведущего» робота.

Человек–оператор, взаимодействуя с «ведущим» устройством, передает ему данные об изменении положения своей руки и о приложенной силе. «Ведущий» робот, в свою очередь, меняет свое положение, под воздействием человека–оператора, и передает данные о законе изменения его обобщенных координат и силе, передаваемой человеком, на билатеральный контроллер. Контроллер, используя полученную информацию, формирует закон управления для «ведомого» устройства и передает соответствующий сигнал. «Ведомый» робот, по возможности учитывая информацию с датчиков, меняет свои обобщенные координаты, в соответствии с полученным сигналом, и тем самым воздействует на объекты рабочей среды с той силой, с которой человек воздействовал на «ведущее» устройство. В процессе выполнения операции, «ведомое» устройство, собирает информацию о положении объектов рабочей среды и сил их реакции и передает данные обратно на билатеральный контроллер. Контроллер формирует соответствующий сигнал управления для «ведущего» устройства. И «ведущий» робот меняет свои координаты, передавая тем самым человеку–оператору силу реакции рабочей среды.

Основными свойствами билатеральной системы являются устойчивость, производительность и транспарентность. Транспарентность — это свойство билатеральной системы управления, характеризующая степень идеальности кинестетической связи ведущего и ведомого устройств.

Общая зависимость между положениями и силами в билатеральном управлении может быть описана следующим образом

$$\begin{bmatrix} f_h \\ q_s \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} q_m \\ f_s \end{bmatrix} = H \begin{bmatrix} q_m \\ f_s \end{bmatrix}$$

Четыре параметра  $h_{ij}$  имеют определенные физические значения:  $h_{12}$  соответствует точности отражения силы (отслеживания силы),  $h_{21}$  относится к отслеживанию положения ведомого устройства относительно ведущего устройства,  $h_{11}$  обозначает механическое сопротивление ведущей стороны и  $h_{22}$  может быть интерпретировано как механическая податливость ведомой стороны.

Идеальная транспарентность билатерального управления может быть представлена в следующей матрице  $H$

$$\begin{bmatrix} f_h \\ q_s \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} q_m \\ f_s \end{bmatrix}$$

### 1.3 Схемы управления для билатеральных систем с задержкой

Существенной проблемой, при разработке систем билатерального управления, является наличие временной задержки, которая может привести к неустойчивости и плохой транспарентности системы. Исследователи пытались решить данную проблему различными методами и в итоге были предложены и испытаны множество различных схем управления. Из предложенных методов можно выделить три основные категории: методы основанные на пассивности, методы основанные на прогнозирующих дисплеях и методы скользящего режима управления.

#### 1.3.1 Схемы, основанные на пассивности

Система называется пассивной, если она может потреблять энергию без возможности производить ее. То есть выход энергии из системы ограничен начальной накопленной энергией в системе. Пассивность, в свою очередь, является достаточным условием устойчивости.

Дистанционное управление на основе пассивности впервые было предложено Андерсеном и Спонгом в 1989 году [12]. В их подходе канал связи был определен таким образом, чтобы гарантировать пассивность независимо от временной задержки. Данная задача достигалась путем объединения теории сетей, преобразования рассеяния и пассивности.

Нимейер и Слотин продолжили эту идею и в 1991 году представили метод волновых переменных [13], который обеспечивал схему кодирования терпимую к задержке.

Рассмотрим мощность, передаваемую через систему, и разделим ее на движущуюся вперед и возвращающуюся [10]:

$$P = \dot{\mathbf{x}}^T \mathbf{F} = \frac{1}{2} \mathbf{u}^T \mathbf{u} - \frac{1}{2} \mathbf{v}^T \mathbf{v} = P_{forward} - P_{return},$$

где движущаяся вперед и возвращающаяся мощности неотрицательны. Здесь  $\dot{\mathbf{x}}(t) = [\dot{\mathbf{x}}_m(t), \dot{\mathbf{x}}_s(t)]^T$ ,  $\mathbf{F}(t) = [f_h(t), f_e(t)]^T$ .

В таком случае волновые переменные представляются в следующем виде

$$\mathbf{u} = \frac{b\dot{\mathbf{x}} + \mathbf{F}}{\sqrt{2b}}, \quad \mathbf{v} = \frac{b\dot{\mathbf{x}} - \mathbf{F}}{\sqrt{2b}},$$

где  $\mathbf{u}$  — это движущаяся вперед волна, а  $\mathbf{v}$  — возвращающаяся волна. Коэффициент волнового сопротивления  $b$  может быть настроен для изменения поведения системы, в зависимости от того будет ли на данный момент осуществляться контакт с окружающей средой или еще нет.

Проблема устойчивости решается лишь в предположении, что каждый блок системы является пассивным, включая человека–оператора, среду, контроллеры, роботов и канал связи.

Данная схема управления была протестирована на задаче жесткого контакта со средой. Результаты показали, что система остается устойчивой при постоянной задержке в диапазоне от 40 мс до 2 с. Однако прозрачность предлагаемой схемы была довольно плохой. При увеличении задержки до 2 с, отслеживание силы и положения стало сильно ухудшаться. Эти проблемы могут быть вызваны отражением волн, дискретной выборкой, числовыми ошибками и потерей данных. Для решения данной проблемы Нимейер и Слотин в 2004 году предложили два подхода [14].

Первый подход использовал волновой интеграл, где не требовалась дополнительная полоса пропускания, однако система могла передавать данные не только о скорости и силе, но и о положении. Вторым решением было добавление корректирующего слагаемого, которые не вводил дополнительную мощность, но при этом компенсировал ошибки положения. Так как метод волновых интегралов потенциально способен нарушить устойчивость системы при непостоянных задержках, было предложено использовать при проектировании системы восстанавливающий фильтр. Еще одним недостатком волновых переменных было отражение волн, которое заключалось в том, что волновые переменные, циркулирующие в системе несколько раз, ставят под угрозу прозрачность системы. Для решения данной проблемы Нимейер и Слотин предложили использовать согласование по волновому сопротивлению, где каждый блок системы спроектирован как демпфер.

Предлагаемая схема была протестирована на контакте робота с объектами рабочей среды, с применением постоянной задержки до 2 с, а затем с использованием переменной задержки от 50 мс до 1 с. Исследователи показали устойчивость системы в данном случае, и улучшенную прозрачность по

сравнению с базовой схемой волновых переменных. Однако ошибка положения между «ведущим» и «ведомым» роботами составляла порядка сантиметра.

В 2006 году Ли и Спонгом была предложена новая система управления, использующую простой пропорционально-дифференциальный регулятор (ПД-регулятор) [15]. Данная структура, в отличие от теории рассеяния, обеспечивает полную пассивность комбинации блоков связи и управления. Этот метод также предполагал постоянную временную задержку и необходимость настройки параметров ПД-регулятора. Эксперимент показал устойчивость и транспарентность системы при наличии постоянной временной задержки до 3 с. Однако, при тестировании данной схемы, произошли некоторые ошибки как в отражении силы, так и в координации положения, которые были вызваны трением, использованным в алгоритме.

Также в 2006 году был предложен еще один подход, который представлял собой комбинацию волновых переменных и предикторов [16]. Данный метод основывается на предположении, что «ведомый» робот рассеивает достаточно энергии. Устойчивость достигалась, благодаря применению передачи с переменной волной. Предлагаемая схема управления использовала переключение между двумя различными предикторами в зависимости от типа удаленной среды — полуадаптивным предиктором и полностью адаптивным предиктором. Полуадаптивный предиктор применялся при жестком контакте со средой, в то время как полностью адаптивный хорошо подходил для медленно меняющейся среды. Эксперименты показали, что транспарентность системы намного лучше при таком комбинированном методе управления, чем при применении какого-либо одного предиктора. Используемая в эксперименте переменная задержка составляла от 0,4 до 0,5 с. Еще одной ключевой особенностью проводимого исследования было моделирование протокола передачи UDP с 10% вероятностью потери данных, и система показала свою надежность в отношении этой модели передачи. В целом, эффективность отслеживания была вполне удовлетворительной, хотя и не очень точной, за исключением отслеживания положения в свободном пространстве, которое оказалось достаточно точным, несмотря на небольшие колебания, наблюдаемые в «ведомом» роботе.

Основным преимуществом методов на основе волновых переменных является то, что они гарантируют стабильность системы и полностью устойчивы к задержкам. Однако недостатком этого метода управления является предположение о том, что оператор и окружающая среда пассивны, что не всегда может



быть правдой. В случае когда имеет место переменная временная задержка, для повышения устойчивости и производительности системы необходимы некоторые дополнительные переменные, параметры управления или специальные каналы связи.

### 1.3.2 Схемы, основанные на прогнозировании

Примерно в 2000 году концепция Smith–контроллера была применена к билатеральным системам телеуправления [17]. До этого она впервые была предложена для решения проблемы задержек на химических заводах в 1957 году [18]. Основная идея метода состоит в «клонировании» «ведомого» устройства и рабочей среды на стороне оператора путем построения локальной модели. Эта прогнозирующая локальная модель является отображением объектов удаленной среды, тем самым она позволяет полностью воспроизводить реакцию удаленных объектов, устраняя отрицательные эффекты временной задержки. Этот вид управления может быть устойчивым к временной задержке более 1 с, что является очень желательным свойством, особенно для космической телеробототехники. Однако для достижения хорошей прозрачности требуется очень точная модель среды. Наряду со стабильностью системы в целом, это две основные проблемы, которые исследователи пытаются преодолеть в системах телеуправления.

В случае концепции Smith–контроллера, наличие модели удаленной среды делает связь между прозрачностью и устойчивостью не такой очевидной. Поскольку при условии, что модель является идеальным представлением рабочей среды, наличие модели гарантирует прозрачность системы. Поэтому только устойчивость является основной проблемой.

В 2003 году Хуан и Льюис предложили схему управления, основанную на этой концепции. В [19] они представили двухканальную архитектуру управления «положение–положение» и использовали оценивающую нейронную сеть на удаленной стороне для анализа ее динамики. Оценивающая нейронная сеть — это адаптивная система, которая используется для моделирования сложных отношений между входами и выходами, которая широко применяется при идентификации и управлении нелинейными системами. Этот оценщик строит



модель на удаленной стороне путем линеаризации нелинейной динамики с использованием метода локальной нелинейной компенсации и затем отправляет эту информацию обратно в центр управления. Сторона оператора содержит в себе линейный Smith-предиктор, который принимает информацию, отправленную из модели на удаленной стороне. На данном этапе вся нелинейная динамика «ведомого» устройства аппроксимируется линейной моделью. Поскольку модель передает информацию Smith-предиктору, считается, что предиктор дает точное представление об удаленной среде. При условии, что динамика рабочей среды «ведомого» робота хорошо смоделирована, эта схема управления дает удовлетворительные результаты. Однако это может быть оправдано лишь в случае медленно меняющейся или неизменной динамики, так как модель предиктора является фиксированной.

Для данной схемы управления устойчивость замкнутой системы была доказана с помощью метода Ляпунова. Несколько важных допущений для обеспечения устойчивости включают ограничения на желаемые состояния и на выход системы. Также предполагается, что удаленная сторона хорошо представлена в своей линеаризованной модели. Если это не так, то в системе может быть нарушена устойчивость. Этот вид управления подходит для систем с постоянными временными задержками и обеспечивает удовлетворительные характеристики для отслеживания движения в свободном пространстве. Однако он обладает плохой транспарентностью при контакте с объектами удаленной среды, что является общим недостатком для схем управления «положение-положение». Результаты, основанные на такой схеме управления, получены с использованием постоянной задержки 0,5 с.

Примерно в то же время Файт, Голдфарб и Рубио предложили схему прогнозирующего управления, использующую архитектуру «положение-сила» [20]. Они предложили внедрить адаптивный контроллер на стороне оператора, чтобы скорректировать линейную оценку. В этом случае информация с удаленной среды возвращается через канал связи, а затем постоянно обновляет модель (предиктор) на «ведущей» стороне. Это дает схеме управления возможность настраиваться на изменяющиеся во времени условия. Однако, опять же, используемый здесь предиктор все еще линейный, и важно отметить, что не вся среда может быть идеально смоделирована с помощью линейной модели.

Для данной схемы управления не было строгого доказательства устойчивости. Предлагаемый метод также опирается на первоначальную идею

Smith–предиктора, утверждающего, что если модель является хорошим представлением удаленной стороны, то система является устойчивой. Так как в предлагаемом подходе используется адаптивный контроллер, то это предположение более осуществимо. В ходе экспериментов было показано, что прозрачность системы устойчива к изменению динамики среды при жесткости в диапазоне от  $75 \text{ Н} \cdot \text{м}^{-1}$  до  $1275 \text{ Н} \cdot \text{м}^{-1}$ . Предполагается, что задержка является постоянной, и в проведенных экспериментах использовалось значение 0,5 с задержки в одну сторону.

В 2006 году Смит и Хаштруди-Заад представили улучшенную модель, основанную на предикторах, в [21], используя оценивающую нейронную сеть как на «ведущей», так и на «ведомой» сторонах, что делает их обе адаптивными. Предлагаемый подход по-прежнему использует архитектуру «положение–сила». Суть метода заключается в том, что на удаленной стороне строится модель в режиме онлайн, используя накопленную информацию от «ведущего» устройства в качестве входных данных и соответствующую силовую реакцию от «ведомого» устройства в качестве выходных данных, тем самым освобождая модель от эффектов временной задержки. Эта модель затем передается обратно в центр управления и постоянно обновляется. Эта схема управления позволяет в реальном времени отображать динамическую и нелинейную среду, но для построения модели требуется небольшой период обучения. В этом случае устойчивость также не была проанализирована теоретически. Также из экспериментов, проведенных этими двумя исследователями, даже несмотря на то, что стабильность показала себя намного лучше по сравнению с не-предсказательными методами, заметная ошибка отслеживания положения все еще присутствует. Постоянная временная задержка, использованная в экспериментах, составляла 0,5 с.

В 2009 году, Йошида и Намерикава предложили другую схему прогнозирующего управления в [22] и попытались решить проблемы в предыдущих методах. Их идея основана на 3-х канальной архитектуре, в которой информация о задерживающихся положении и скорости «ведущего» устройства отправляется «ведомой» стороне, а оцененные параметры среды отправляются обратно «ведущей» стороне для прогноза. Это гарантирует устойчивость системы, а также гарантирует, что ошибка отслеживания положения сходится к нулю, в то время как ошибка отслеживания усилия остается небольшой. Для обсуждаемой схемы управления этого типа была доказана асимптотиче-

ская устойчивость с помощью метода Ляпунова, в предположении, что ошибки оценки параметров ограничены. Экспериментальные результаты показывают очень малые ошибки отслеживания после достижения устойчивого состояния. Задержка, используемая авторами, была постоянной и имела значение 1 с.

Можно заметить, что основным преимуществом схем прогнозирующего управления является возможность учета относительно большой временной задержки. Однако их сложность заключается в построении точной модели удаленной стороны. Использование адаптивных контроллеров и оценщиков, кажется, помогает моделировать нелинейную и изменяющуюся среду и показывает намного лучший результат, чем простой метод линеаризации.

### 1.3.3 Схемы, основанные на скользящем режиме управления

В отличие от метода управления на основе пассивности и метода управления на основе прогнозирования, управление в режиме скольжения является нелинейным методом управления системой. Он определяет поверхность скольжения и заставляет траектории динамической системы приближаться к ней, достигать ее и оставаться на ней. Некоторые исследователи попытались использовать методы на основе скользящего режима, чтобы усовершенствовать систему билатерального телеуправления.

В 1999 году исследователи Парк и Чо предложили модифицированный контроллер скользящего режима для билатерального телеуправления [23]. В данной работе было показано, что контроллер скользящего режима является надежным для моделирования изменений параметров, и следовательно, имеет отличные приложения для систем, работающих с изменяющейся во времени задержкой. В этом случае используется архитектура управления «положение–сила». На «ведущей» стороне использовался импеданс–контроллер, в то время как на «ведомой» стороне применялось два типа контроллеров, в зависимости от текущей задачи робота. При перемещении инструмента «ведомого» робота в свободном пространстве без контакта с объектами рабочей среды, использовался модифицированный контроллер скользящего режима первого порядка, а при контакте применялся локальный податливый контроллер. Нелинейный регулятор усиления модифицированного контроллера скользяще-

го режима был преобразован таким образом, что он не зависил от временной задержки. Это улучшило прозрачность и стабильность системы.

Для данной схемы управления устойчивость была математически доказана. Прозрачность представлялась приемлемой в соответствии с экспериментальными результатами. Схема управления имела дело с изменяющейся временной задержкой и задержка сигнала от «ведущего» к «ведомому» считалась отличной от задержки обратной связи от «ведомого» к «ведущему». Общая задержка, используемая при моделировании, имела максимальное значение до 4 секунд, что продемонстрировало большой потенциал для телеуправления с переменной задержкой передачи сигналов. При передаче данные не были потеряны и не требовалось упорядочивание пакетов, так как применялся протокол ТСР.

Несколько лет спустя было предложено управление скользящим режимом на основе «наблюдателя», где для «ведущего» робота использовалось управление по сопротивлению, а для «ведомого» робота скользящий режим управления второго порядка [24]. Несмотря на то, что это скользящий режим управления второго порядка, значение ускорения не измеряется датчиком, а определяется «наблюдателем». Используя тот же принцип, измерение скорости датчиком также не требуется. Полученная система была разработана для постоянной неизвестной временной задержки. При перемещении инструмента без контакта с объектами, выходные сигналы положения, скорости и ускорения стремятся к желаемым, однако при контакте возникает компромисс между силовым и позиционным отслеживанием. Более того, система не обеспечивает идеальной прозрачности, а дает только хорошее представление об удаленной среде.

Существуют также и другие, менее распространенные, методы построения систем билатерального телеуправления с задержкой в передаче сигналов, например, адаптивный контроллер Чжу и Салкудина [25], контроллер на основе событий [26], компенсация по расписанию усиления [27] и т. д.

### 1.3.4 Перспективы исследований

Билатеральные системы телеуправления с задержкой в передаче сигналов является быстро развивающейся областью исследований. Основываясь на его нынешнем статусе, можно сделать несколько выводов о его будущих тен-

денциях. Можно заметить, что проявляется растущий интерес от постоянной временной задержки к переменной, которая обычно использует Интернет как канал связи. Фактически, из-за его высокой доступности и низкой стоимости, он имеет большой потенциал и представляет интерес для исследований с конца 90-х годов. Однако для его реализации необходимо провести дальнейшие исследования, в которых реализована задержка по времени, а также потеря информации и упорядочение пакетов (в случае UDP). Временные задержки, которые рассматривались вышеперечисленными исследователями, в основном либо постоянны, либо малы, либо варьируются до нескольких секунд. Однако существует и другой тип задержки, которая возникает в случае проведения работ в космосе. Когда человек–оператор на Земле манипулирует роботом в космосе, временная задержка может быть намного больше, вплоть до нескольких минут. Решение данной проблемы будет очень полезно для исследовательских миссий на Луне или Марсе. Для реализации билатерального телеуправления в таких приложениях потребуются другие подходы. Билатеральная система управления с масштабированием представляет собой еще одну интересную задачу. В медицинских приложениях иногда желательно уменьшить масштаб и выполнить телеуправление в микро- и даже нано-масштабе. В других приложениях может возникнуть необходимость в увеличении, например, для умножения приложенной силы оператора.

Билатеральное телеуправление имеет весомый потенциал во многих исследовательских и промышленных областях. На данный момент задержка в передачи сигналов является одной из ключевых проблем, мешающих внедрению этого приложения. Существующий на данный момент времени ряд схем управления, направленных на решение данной проблемы, способны компенсировать влияние задержки лишь при значениях не превышающих 4 секунд, что существенно ограничивает расстояние между «ведущим» и «ведомым» роботами. В связи с этим, остается актуальным вопрос построения такой схемы управления роботом, с помощью которой можно было бы проводить работы на больших расстояниях, например, в космосе.

## 1.4 Дистанционное управление космическим роботом методом обучения

Запуск первого искусственного спутника Земли ознаменовал начало эпохи активного освоения человеком космического пространства. С тех пор исследователи ищут пути и способы построения космических станций на поверхностях и орбитах небесных тел Солнечной системы. Так как открытый космос является враждебной средой для жизнедеятельности человека, то в качестве «строителей» станций целесообразно использовать космических роботов. Однако существующие на данный момент времени методы дистанционного управления роботами не пригодны для использования их на очень больших расстояниях. Следовательно, необходимо найти принципиально новый подход к управлению роботом, который обеспечивал бы успешное выполнение операций на большой дистанции. Для решения данной задачи, профессор СПбГУ Ф.М. Кулаков предложил метод дистанционного управления роботом с помощью обучения будущим действиям. В данном подходе к управлению удаленная рабочая среда «копируется» в виртуальном или натурном виде в центре управления и человек–оператор взаимодействует необходимым образом с объектами модельной среды, после чего космический робот отрабатывает те же действия на объектах удаленной среды, при этом адаптируясь к возможным ее изменениям.

Процесс управления космическим роботом методом обучения можно разделить на 4 этапа (Рис. 1.3, 1.4).

**I этап.** На первом этапе космический робот, с помощью установленных на нем датчиков, собирает необходимую информацию о рабочей среде и передает ее в центр управления.

**II этап.** На следующем этапе, на основе полученных данных, формируется модель удаленной среды. Модель может быть выполнена в виде натуральных макетов, либо построена на компьютере.

**III этап.** Третий этап, происходящий в центре управления на Земле, может быть выполнен двумя разными способами в зависимости от типа модели удаленной среды.

Если удаленная среда представлена в виде натурной модели (Рис. 1.3.), то взаимодействие с ней происходит по следующему принципу. Человек–оператор, взаимодействуя с «ведущим» роботом, задает необходимое положение



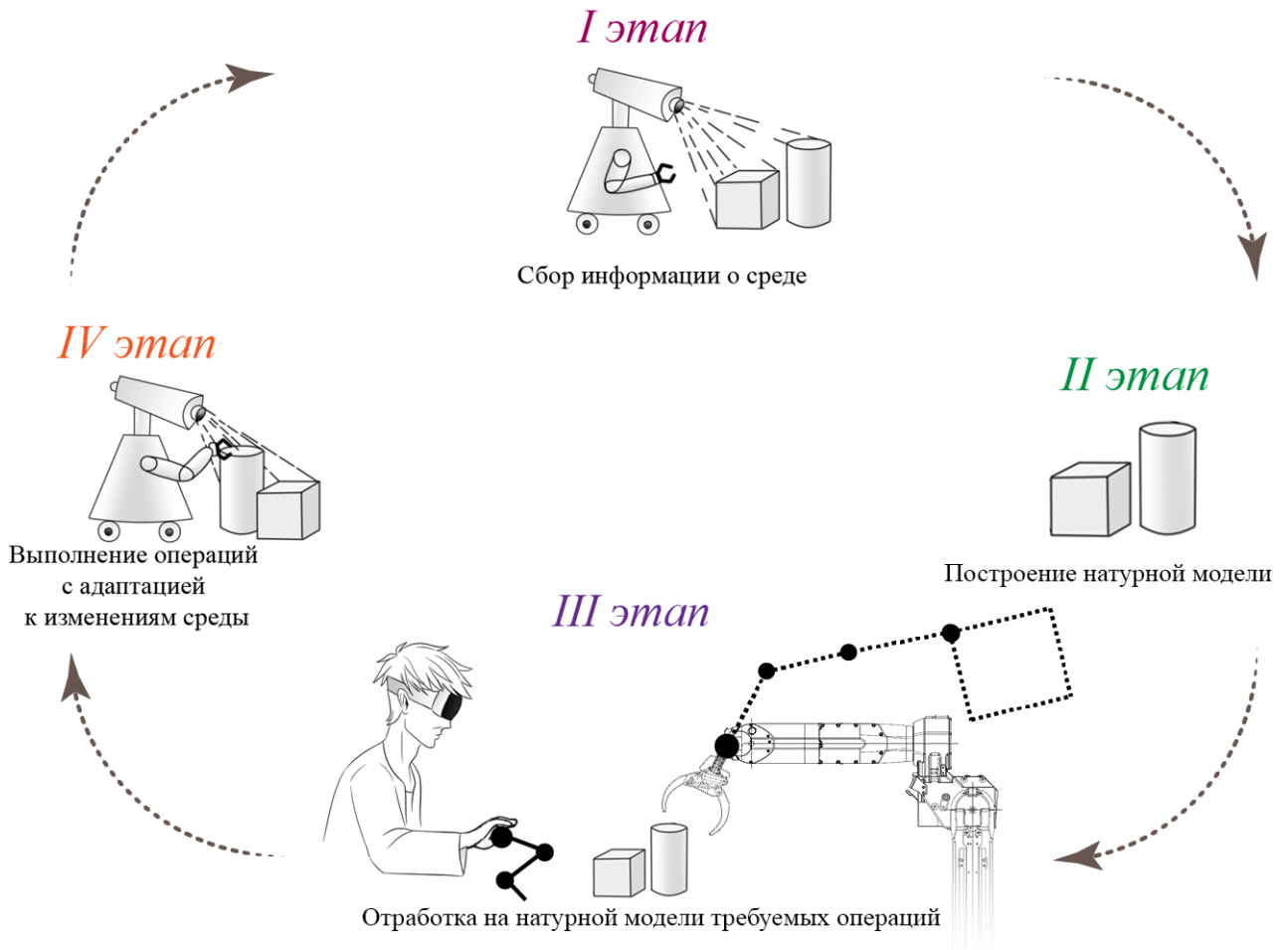


Рисунок 1.3 — Процесс управления роботом методом обучения (при использовании натурной модели рабочей среды)

и ориентацию инструменту «ведомого» наземного робота, в качестве которого может выступать промышленный манипулятор с 6-тью степенями подвижности. Инструмент данного робота должен точно интерпретировать инструмент космического робота. Таким образом, оператор взаимодействует с объектами натурной модели как при традиционном способе билатерального управления. Движения инструмента запоминаются и передаются космическому роботу. При необходимости, космический робот может отображаться для человека-оператора в очках дополненной реальности.

Если же построить натурную модель не представляется возможным, то можно использовать виртуальную модель рабочей среды (Рис. 1.4.). В таком случае, на третьем этапе человек-оператор, наблюдая за виртуальной средой через очки дополненной реальности, взаимодействует с объектами данной среды с помощью робота-перчатки, и, тем самым, отрабатывает необходимые операции.

В том и в другом случае, при необходимости, полученный алгоритм тестируется на компьютерной модели космического робота и рабочей среды, при

«искусственном» смещении ее объектов. Если тест прошел успешно, алгоритм передается «ведомому» роботу. В противном случае, в алгоритм вносятся необходимые изменения и вновь проводится тест.

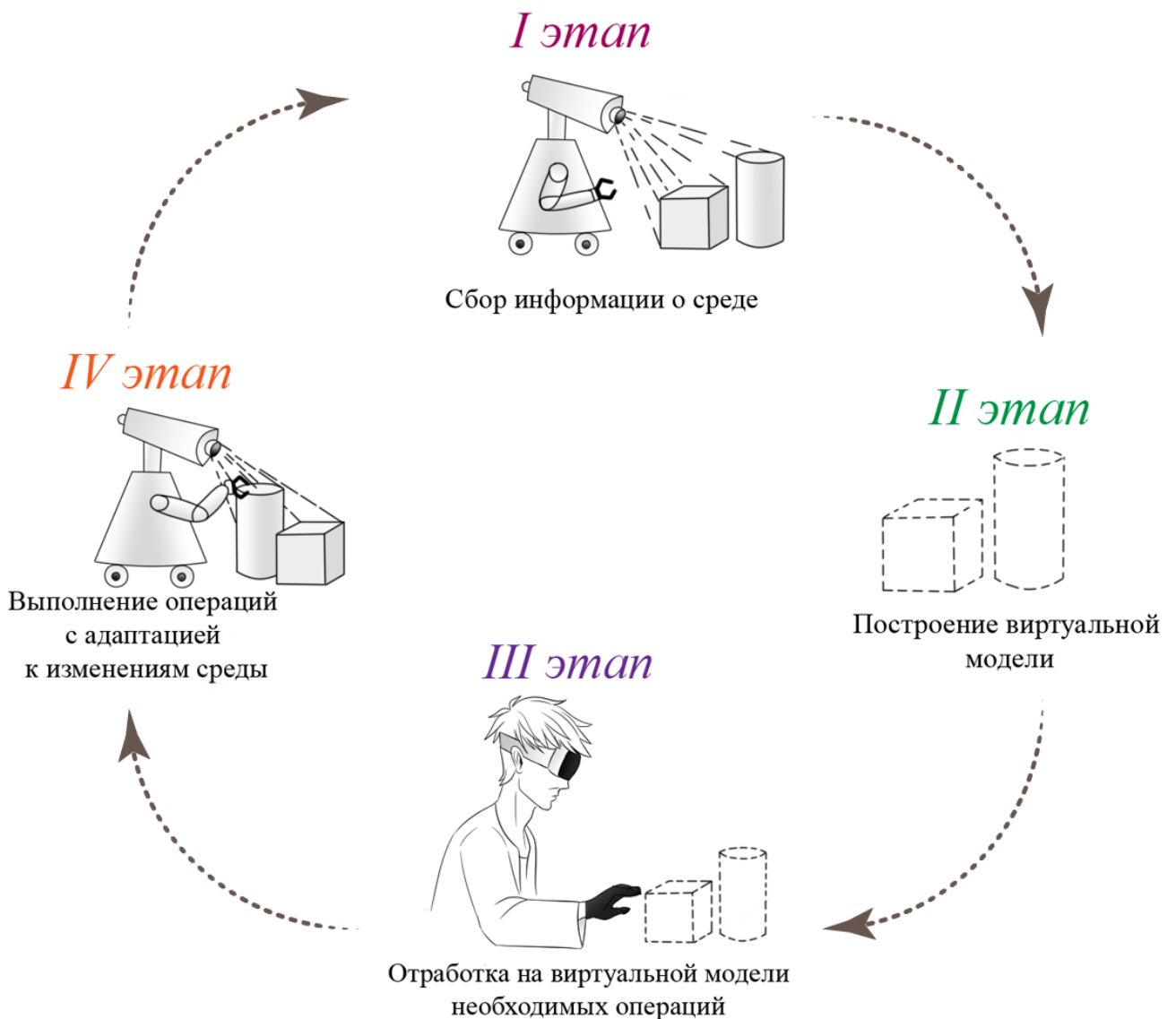


Рисунок 1.4 — Процесс управления роботом методом обучения (при использовании виртуальной модели рабочей среды)

**IV этап.** На последнем этапе космический робот отрабатывает полученный алгоритм действий, при этом адаптируется к возможным изменениям рабочей среды и передает отчет о проделанной работе в центр управления. Если центр управления подтвердил успешность и завершенность миссии, то этапы прекращаются. В противном случае цикл повторяется до полного выполнения требуемой операции.

Структурная схема системы управления методом обучения с использованием натурной модели сводится к структурной схеме классического билате-



рального управления, с тем отличием, что удаленная сторона «переносится» на сторону оператора, а космический робот лишь повторяет процесс, адаптируясь к изменениям рабочей среды. При использовании же виртуальной модели рабочей среды, структурная схема немного сложнее чем в случае традиционного билатерального управления (Рис. 1.2). Процесс передачи данных между элементами системы протекает следующим образом (Рис. 1.5., Рис. 1.6.):

### **I этап.**

1. «Ведомое» устройство, с помощью датчиков, получает информацию о координатах объектов рабочей среды  $q_e$  и передает ее на контроллер, который находится в центре управления.

### **II этап.**

2. На основе полученной информации о среде, контроллер формирует виртуальную модель рабочей среды, и передает значения ее координат  $q_{ve}$  на «ведущее» устройство (с целью определения начального положения виртуальных объектов относительно схвата «ведущего» робота) и на очки дополненной реальности, которые в свою очередь отображают объекты для человека–оператора. При построении модели, человек может внести свои корректировки, например добавить возможные связи на объекты.

### **III этап.**

3. Далее, человек–оператор, перемещая необходимым образом «ведущее» устройство, передает ему данные о положении своей руки  $q_h$  и силу взаимодействия  $f_h$ .

4. «Ведущий» робот меняет свои обобщенные координаты  $q_m$ , в соответствии с данными, полученными от человека, и тем самым взаимодействует с объектами виртуальной среды.

5. Виртуальная модель, получив информацию от «ведущего» устройства, передает на контроллер координаты робота  $q_m$ , свои текущие координаты  $q_{ve}$  и возникающие в процессе взаимодействия виртуальные силы  $f_{ve}$ .

6. Далее происходит обновление положения объектов в виртуальной среде, на основе полученных данных. Контроллер также как и на втором этапе передает обновленную информацию о среде, а также формирует управляющий сигнал для «ведущего» устройства.

7. Получив команду со стороны контроллера, «ведущий» робот меняет свои координаты, тем самым воздействуя на руку человека так, как на нее воздействовал бы реальный объект.

Шаги 3–7 повторяются непрерывно до полной отработки требуемых операций. Далее, при необходимости, происходит тест алгоритма управления на компьютерной модели. Если все прошло успешно, то далее следует шаг 8. В противном случае, шаги повторяются до выявления ошибок.

8. Контроллер формирует алгоритм взаимодействия с объектами среды и передает его «ведомому» устройству.

#### **IV этап.**

9. Космический робот анализирует изменения в рабочей среде и адаптирует полученный алгоритм. В соответствии с новым алгоритмом, робот меняет свои обобщенные координаты  $q_s$ , тем самым воздействуя на объекты среды, точно также как взаимодействовал человек–оператор с виртуальными объектами.

10. Под воздействием «ведомого» устройства, объекты меняют свои координаты  $q_e$  и взаимодействуют с роботом с силой  $f_e$ .

Далее все этапы повторяются до полного завершения миссии.

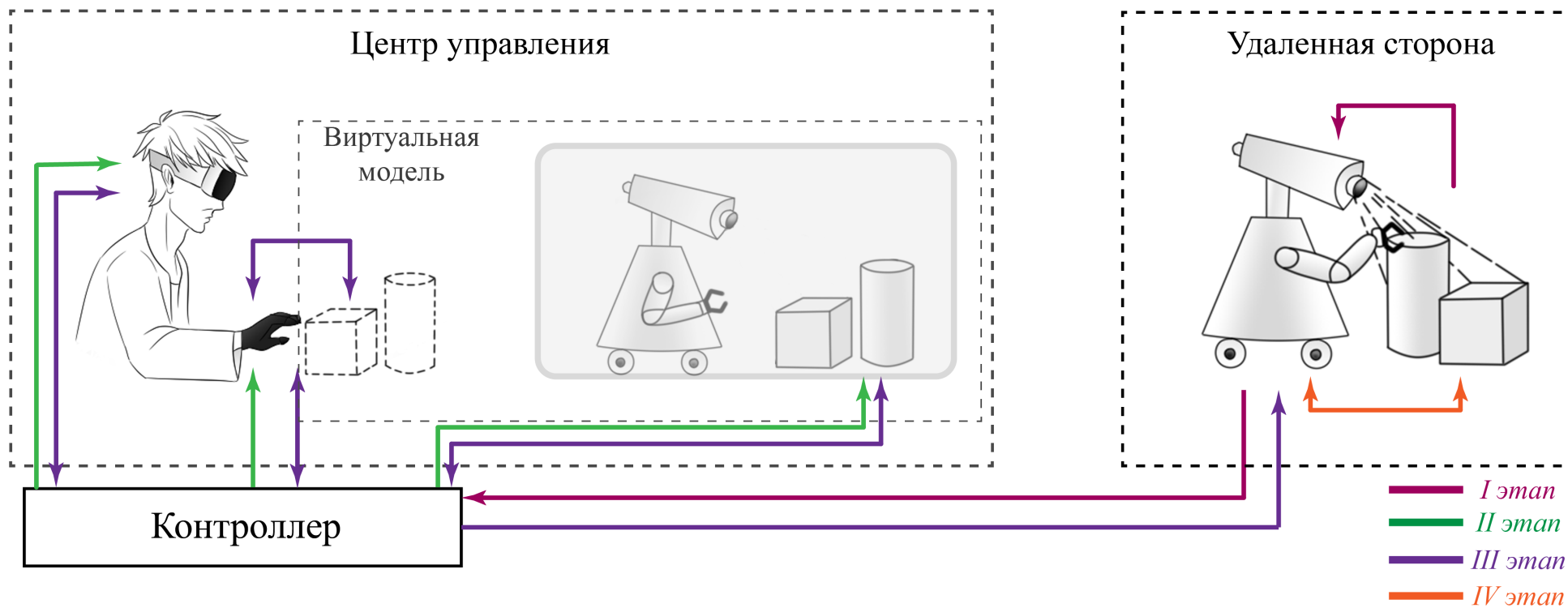


Рисунок 1.5 — Система управления методом обучения

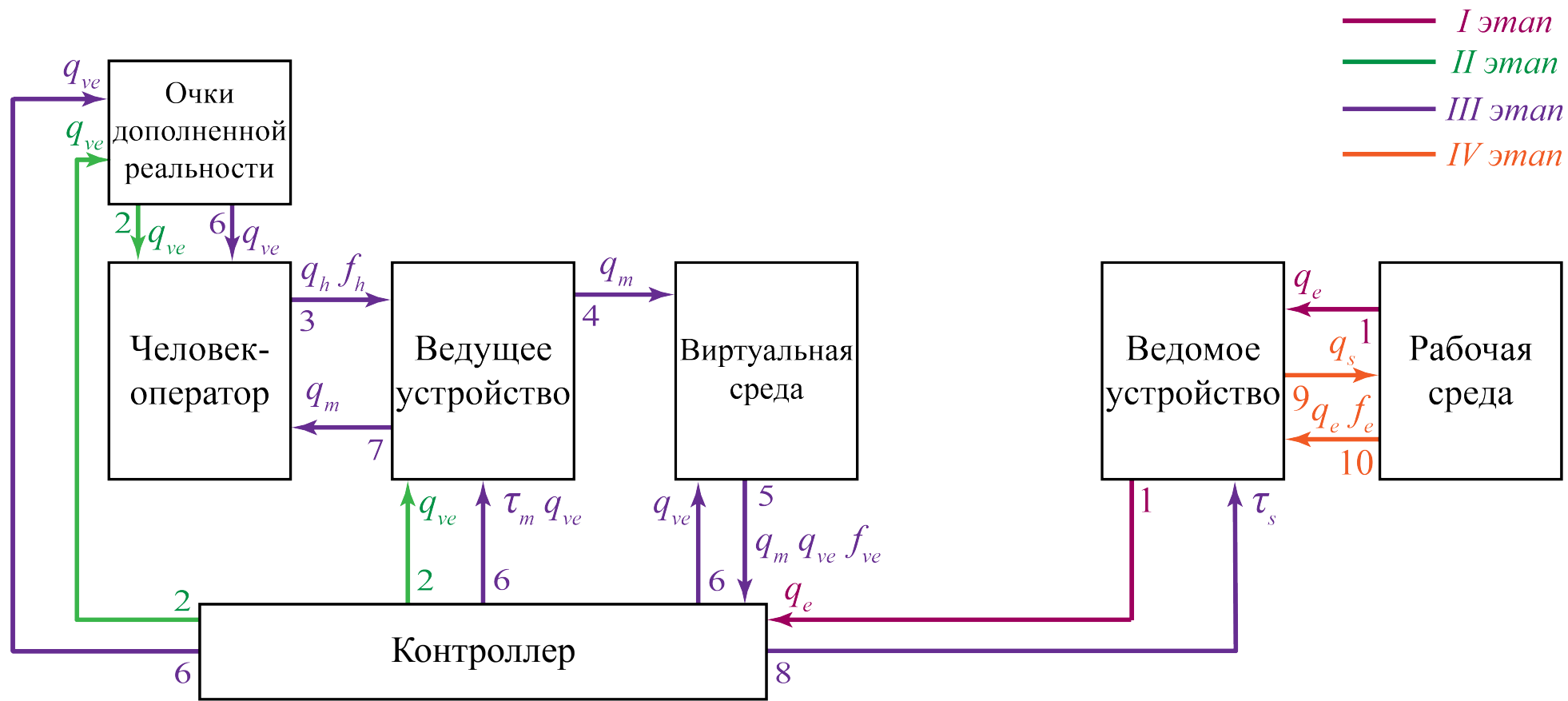


Рисунок 1.6 — Структура системы управления методом обучения

## 1.5 Постановка задачи

Ключевой задачей данной научно-исследовательской работы является построение математических моделей космического и вспомогательного наземного роботов. Для выполнения данной задачи необходимо провести решение следующих подзадач:

- **Прямая позиционная задача.** Для того, чтобы понимать как будут двигаться рассматриваемые роботы при определенных законах изменения обобщенных координат, необходимо решить прямую позиционную задачу для космического и наземного робота. Данная задача требует предварительного построения кинематических схем.
- **Обратная позиционная задача.** Решение обратной позиционной задачи, которая заключается в поиске таких значений обобщенных координат, которые обеспечивали бы заданное положение и ориентацию схвата в абсолютном пространстве, позволит построить зависимость обобщенных координат космического робота от положения и ориентации инструмента наземного робота.
- **Прямая задача кинематики о скоростях.** Прямая задача кинематики о скоростях манипулятора, которая устанавливает связь между обобщенными скоростями и линейными и угловыми скоростями звеньев, является промежуточным этапом на пути построения динамической модели.
- **Прямая задача динамики манипулятора.** Решение прямой задачи динамики манипулятора позволит найти силы, которые обеспечат необходимое движение робота.
- **Алгоритм коррекции поведения робота при изменениях окружающей среды.** Так как космический робот функционирует в недетерминированной рабочей среде, необходимо разработать алгоритм коррекции его поведения в зависимости от изменений среды.
- **Программная реализация.** Для графической визуализации, а также верификации полученных результатов, необходимо разработать программу в математическом пакете Matlab, которая будет содержать в себе кинематические модели космического и наземного робота.

## Глава 2. Математическое моделирование исполнительной системы

Любая роботизированная система нуждается в предварительном математическом моделировании, как здание в предварительном чертеже. Поэтому, для проверки работоспособности системы дистанционного управления космическим роботом, предлагаемой в данной работе, необходимо смоделировать ее элементы и их взаимодействие. Рассматриваемая система состоит из пяти ключевых элементов: человек–оператор, «ведущий» робот, модель космического робота, реальный космический робот и рабочая среда. Так как «ведущий» робот представляет собой инструмент, являющейся передатчиком данных о движении и силе между моделью космического робота и человеком–оператором, то математическое моделирование такого объекта не представляет интереса. В связи с этим, систему «человек–оператор — ведущий робот» будем представлять как блок с выходными и входными данными. Рабочая среда также будет представляться аналогичным блоком, так как она будет натурной и данные о ней будут считываться при непосредственной отработке необходимых операций. Поэтому необходимо построить лишь модель космического робота и модель наземного робота, схват которого будет отображать инструмент модели космического робота. На Рис. 2.1. синим цветом отображены элементы моделирование которых необходимо произвести.

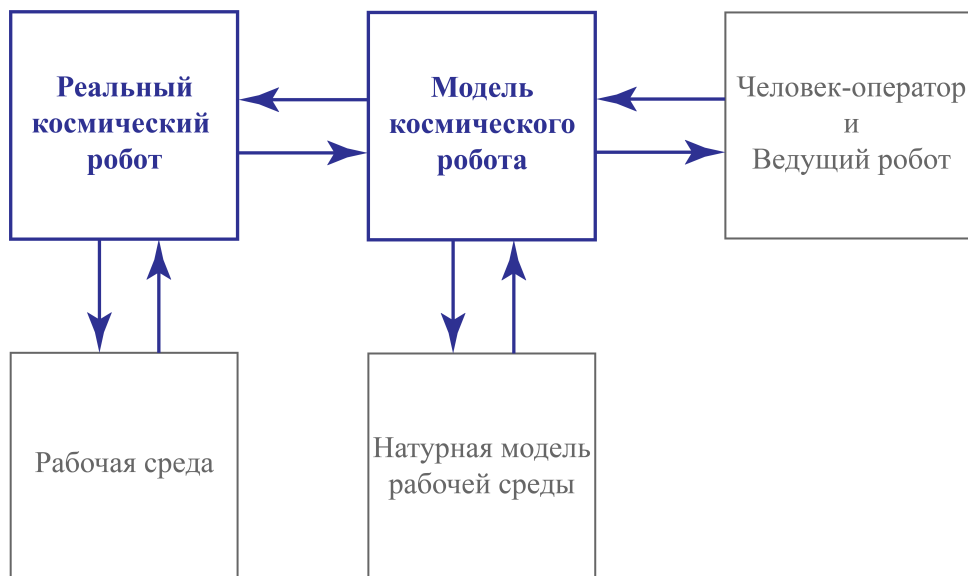


Рисунок 2.1 — Элементы системы дистанционного управления космическим роботом методом обучения

Кинематическое моделирование роботов необходимо для определения его кинематических характеристик с течением времени. Информация о положении характеристических точек робота будет полезна при графическом отображении его движения. Решение обратной задачи кинематики позволит определить значения обобщенных координат, которые обеспечивают заданное положение и ориентацию инструмента в абсолютном пространстве. А информация о скоростях необходима при построении динамической модели.

## 2.1 Кинематические схемы

Построим кинематическую схему космического робота, который в общем виде будет представлять собой корпус с установленным на нем  $n$  – звеном манипулятором (Рис. 2.2). Таким образом робот будет состоять из  $n + 1$  твердых тел. Свяжем с каждым элементом робота соответствующую систему координат.

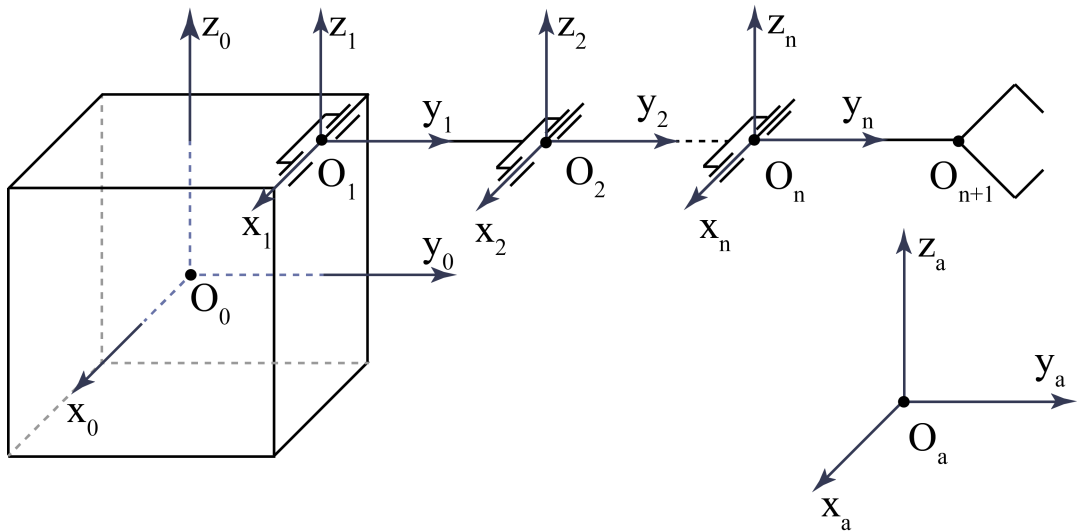


Рисунок 2.2 — Обобщенная кинематическая схема космического робота

Будем предполагать, что каждое звено может линейно перемещаться и вращаться относительно соответствующей оси по отношению к предыдущему телу, то есть имеет как минимум две обобщенные координаты — линейную и вращательную. В таком случае робот будет обладать  $2n + 6$  - ю обобщенными координатами. Первые шесть координат соответствуют положению и ориентации корпуса робота, следующие  $2n$  координат отвечают за положение и ориентацию каждого звена манипулятора. Обобщенные координаты  $q_1, q_2, q_3$

определяют вращение корпуса относительно осей  $O_0x_0, O_0y_0$  и  $O_0z_0$  соответственно, а  $q_4, q_5, q_6$  отвечают за перемещение центра масс корпуса вдоль тех же осей. Координата  $q_{6+2i-1}$  соответствуют вращению  $i$ -го звена, а  $q_{6+2i}$  – его линейному перемещению. Схематичное представление обобщенных координат космического робота изображено на Рис. 2.3. Вращение и перемещение  $i$ -го звена может происходить вдоль любой из трех осей его системы координат ( $O_ix_i, O_iy_i$  или  $O_iz_i$ ).

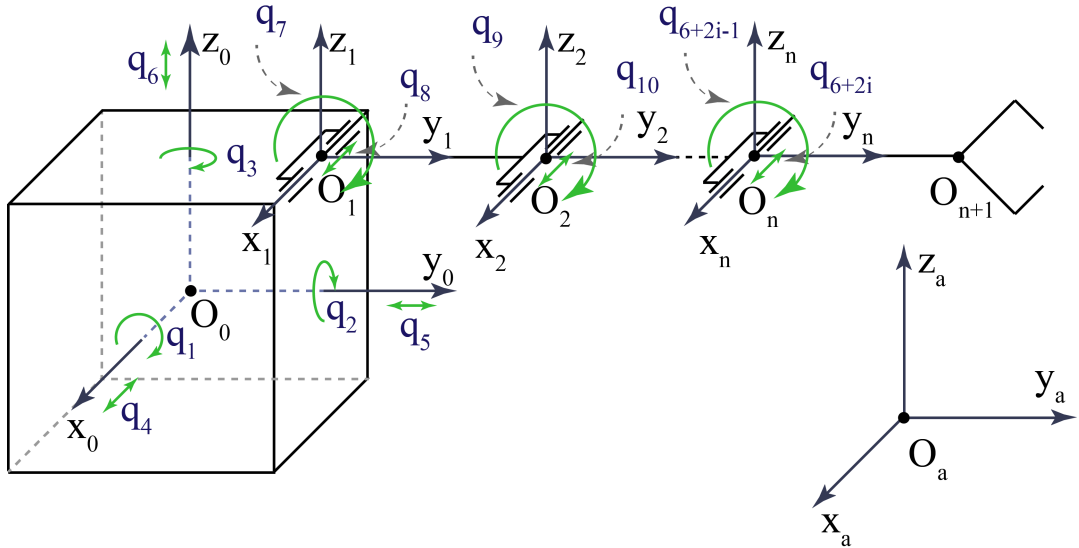


Рисунок 2.3 — Обобщенные координаты космического робота

Если робот предназначен для проведения работ на орбите небесного тела, то корпус робота будет свободным и будут иметь место все его обобщенные координаты. Если же робот будет функционировать на поверхности небесного тела, то обобщенные координаты  $q_1, q_2$  и  $q_6$  будут постоянными.

Перейдем к построению кинематической схемы наземного робота, инструмент которого будет отображать инструмент космического робота. В качестве такого робота рассмотрим промышленный манипулятор FANUC M20-i, который активно используется на факультете Прикладной математики - процессов управления Санкт-Петербургского Государственного Университета. Схематичное представление данной системы представлено на Рис. 2.4.

Свяжем с каждым телом манипулятора соответствующую систему координат и обозначим обобщенные координаты (Рис. 2.5).

На основе построенных схем можно решить прямую и обратную позиционные задачи, прямую и обратную задачи о скорости, представляя роботов как совокупность взаимосвязанных координатных систем.



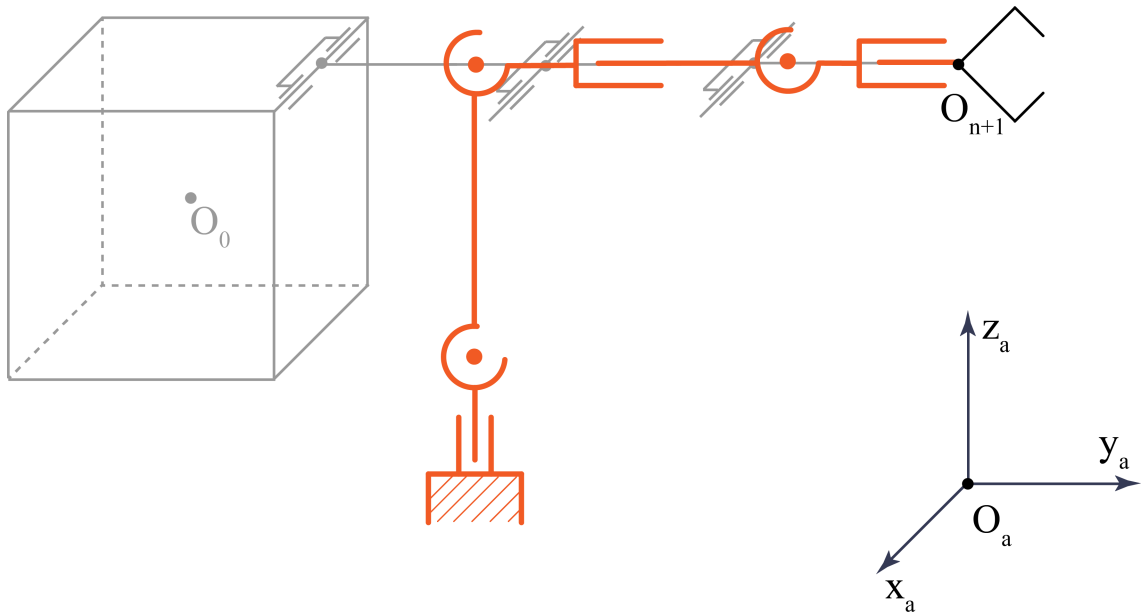


Рисунок 2.4 — Модель космического робота в виде наземного манипулятора, отображающего движение инструмента космического робота

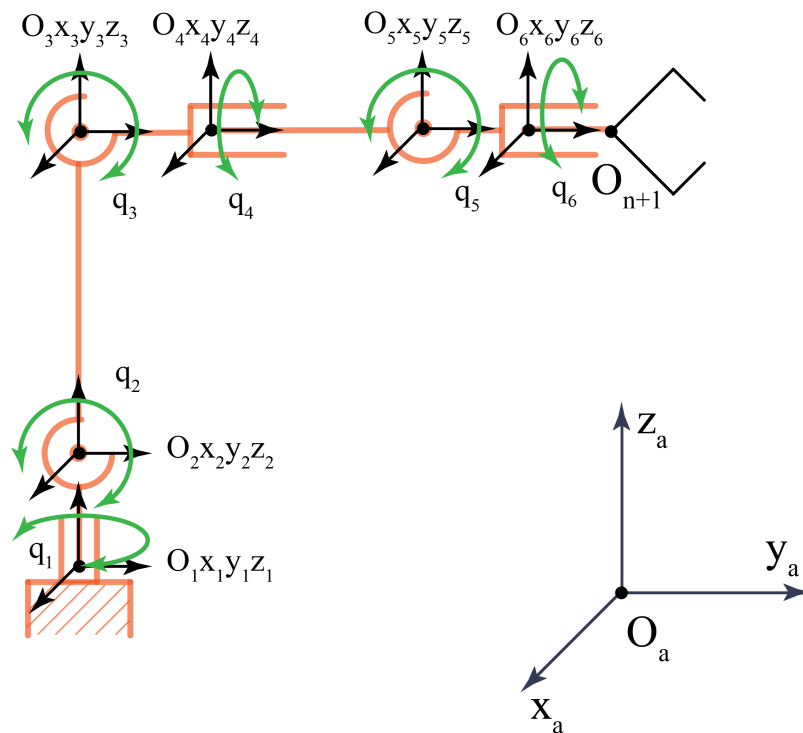


Рисунок 2.5 — Кинематическая схема FANUC и его обобщенные координаты

## 2.2 Прямая позиционная задача

Прямая позиционная задача состоит в определении положения и ориентации инструмента робота в абсолютном пространстве при заданных обобщенных координатах. Решение прямой задачи также позволяет определить зависимость

положения характерных точек робота (вершины корпуса, центр масс корпуса, соединения звеньев и т. п.) от изменения обобщенных координат, что полезно при моделировании движения робота на компьютере.

Решим прямую позиционную задачу для космического робота. Представим его как совокупность координатных систем  $O_0x_0y_0z_0, O_1x_1y_1z_1, \dots, O_nx_ny_nz_n$ , где система  $O_0x_0y_0z_0$  жестко связана с корпусом робота, а система  $O_ix_iy_iz_i$  связана с  $i$  – м звеном (Рис. 2.1). Каждая система координат меняет свое положение в зависимости от изменения обобщенных координат соответствующего ей твердого тела, а также в зависимости от изменения положения предыдущих координатных систем. Вся механическая система функционирует в абсолютной системе координат  $O_ax_ay_az_a$ , связанной либо с поверхностью небесного тела, либо с космической станцией, в зависимости от предназначения космического робота.

За матрицу преобразований между системой координат  $i$  и  $(i-1)$  примем однородную матрицу  $(4 \times 4)$  вида

$${}^{i-1}T_i = \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}\mathbf{p}_i \\ \mathbf{0}^T & 1 \end{bmatrix},$$

здесь  ${}^{i-1}R_i$  — матрица поворота  $(3 \times 3)$   $i$  – ой системы координат по отношению к  $(i-1)$  – ой,  ${}^{i-1}\mathbf{p}_i$  — радиус–вектор положения точки  $O_i$   $i$  – ой координатной системы в системе координат  $O_{i-1}x_{i-1}y_{i-1}z_{i-1}$ ,  $\mathbf{0}^T$  — нулевой вектор  $1 \times 3$ .

В таком случае, если за  ${}^i\mathbf{r}$  обозначить вектор  $(4 \times 1)$ , первые три компоненты которого определяют положение характеристической точки робота в  $i$  – ой системе координат, а четвертая компонента является единичной, то положение характеристической точки инструмента в абсолютной системе определяется по формуле

$${}^a\mathbf{r} = {}^aT_n \cdot {}^n\mathbf{r},$$

где  ${}^aT_n$  — матрица преобразования между  $n$ -ой и абсолютной системой координат, которая вычисляется следующим образом

$${}^aT_n = {}^aT_0 \cdot {}^0T_1 \times \dots \times {}^{n-1}T_n.$$

Проведем решение прямой позиционной задачи для конкретной кинематической схемы космического робота, приведенной на Рис. 2.6.

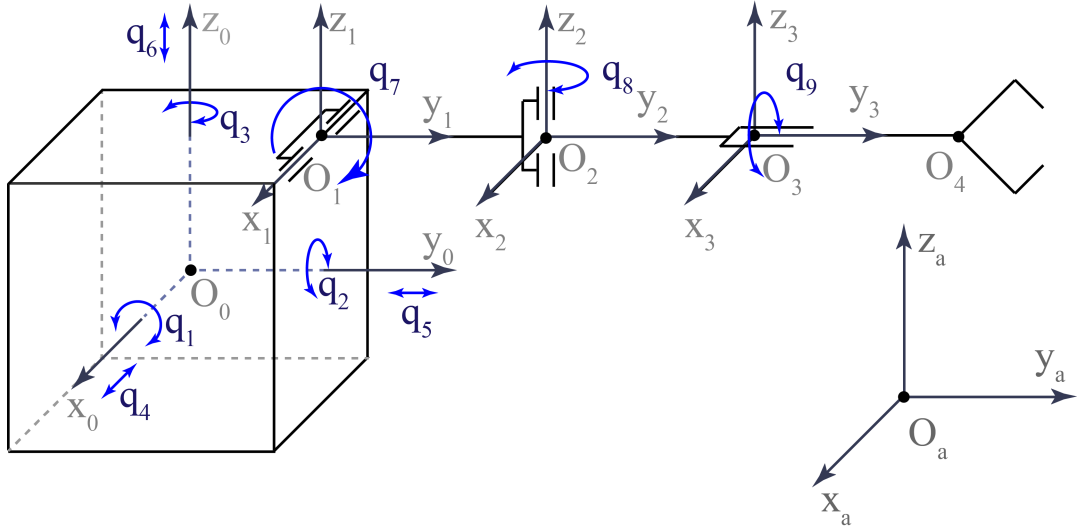


Рисунок 2.6 — Кинематическая схема 9-ти степенного космического робота

Данный робот имеет 4 тела и 9 степеней подвижности. Положение характеристической точки инструмента  $O_4$  в системе координат  $O_a x_a y_a z_a$  будет определяться следующим образом ( $Cq_i = \cos q_i$ ,  $Sq_i = \sin q_i$ )

$${}^a \mathbf{r} = {}^a T_0 \cdot {}^0 T_1 \cdot {}^1 T_2 \cdot {}^2 T_3 \cdot {}^3 \mathbf{r},$$

$${}^a T_0 = \begin{bmatrix} Cq_2 Cq_3 - Sq_2 Cq_1 Sq_3 & Sq_3 Cq_2 + Cq_3 Cq_1 Sq_2 & Sq_1 Sq_2 & q_4 \\ -Cq_3 Sq_2 - Sq_3 Cq_1 Cq_2 & -Sq_2 Sq_3 + Cq_2 Cq_1 Cq_3 & Sq_1 Cq_2 & q_5 \\ Sq_3 Sq_1 & -Cq_3 Sq_1 & Cq_1 & q_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0 T_1 = \begin{bmatrix} 1 & 0 & 0 & {}^0 x_1 \\ 0 & Cq_7 & -Sq_7 & {}^0 y_1 \\ 0 & Sq_7 & Cq_7 & {}^0 z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^1 T_2 = \begin{bmatrix} Cq_8 & -Sq_8 & 0 & 0 \\ Sq_8 & Cq_8 & 0 & l_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^2 T_3 = \begin{bmatrix} Cq_9 & 0 & Sq_9 & 0 \\ 0 & 1 & 0 & l_2 \\ -Sq_8 & 0 & Cq_8 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^a \mathbf{r} = \begin{bmatrix} {}^a x \\ {}^a y \\ {}^a z \\ 1 \end{bmatrix}, \quad {}^3 \mathbf{r} = \begin{bmatrix} 0 \\ l_3 \\ 0 \\ 1 \end{bmatrix},$$

здесь  $l_i$  — длина  $i$ -го звена, а  ${}^0 x_1, {}^0 y_1, {}^0 z_1$  — положение точки  $O_1$  в системе координат корпуса.

Решением прямой позиционной задачи будет определение координат  ${}^a x, {}^a y$  и  ${}^a z$  через обобщенные координаты, а также определение ориентации инструмента, которая будет представлять собой вектор-столбцы результирующей матрицы поворота (ориентация  $O_3 x_3 y_3 z_3$  в системе  $O_a x_a y_a z_a$ ).

Решение прямой позиционной задачи для наземного вспомогательного робота (Рис. 2.5) представим в следующем виде (здесь мы учитываем особенность робота FANUC, которая заключается в том, что при изменении координаты  $q_2$  меняется координата  $q_3$  на то же значение, но в обратную сторону)

$${}^a \mathbf{r} = {}^a T_1 \cdot {}^1 T_2 \cdot {}^2 T_3 \cdot {}^3 T_4 \cdot {}^4 T_5 \cdot {}^5 T_6 \cdot {}^6 \mathbf{r},$$

$${}^a T_1 = \begin{bmatrix} Cq_1 & -Sq_1 & 0 & {}^a x_1 \\ Sq_1 & Cq_1 & 0 & {}^a y_1 \\ 0 & 0 & 1 & {}^a z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^1 T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & Cq_2 & -Sq_2 & 0 \\ 0 & Sq_2 & Cq_2 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(-q_2) & -S(-q_2) & 0 \\ 0 & S(-q_2) & C(-q_2) & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^2 T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & Cq_3 & -Sq_3 & 0 \\ 0 & Sq_3 & Cq_3 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^3 T_4 = \begin{bmatrix} Cq_4 & 0 & Sq_4 & 0 \\ 0 & 1 & 0 & l_3 \\ -Sq_4 & 0 & Cq_4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^4 T_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & Cq_5 & -Sq_5 & l_4 \\ 0 & Sq_5 & Cq_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^5 T_6 = \begin{bmatrix} Cq_6 & 0 & Sq_6 & 0 \\ 0 & 1 & 0 & l_5 \\ -Sq_6 & 0 & Cq_6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^a \mathbf{r} = \begin{bmatrix} {}^a x \\ {}^a y \\ {}^a z \\ 1 \end{bmatrix}, \quad {}^6 \mathbf{r} = \begin{bmatrix} 0 \\ l_6 \\ 0 \\ 1 \end{bmatrix}.$$

Использование однородных преобразований весьма удобно для записи матричного представления прямой позиционной задачи, а также удобно для восприятия. Однако при программировании уравнений в таком виде будет много ненужных операций типа умножения на ноль. Поэтому для программирования целесообразно пользоваться преобразованием координат в другом виде

$${}^j \mathbf{r} = {}^j R_i \cdot {}^i \mathbf{r} + {}^j \mathbf{p}_i,$$

где  ${}^j \mathbf{r}, {}^i \mathbf{r}$  —  $(3 \times 1)$  — векторы положения искомой точки в  $j$  — ой и в  $i$  — ой системе координат соответственно,  ${}^j R_i$  —  $(3 \times 3)$  — матрица ориентации  $i$  — ой системы координат по отношению к  $j$  — ой,  ${}^i \mathbf{p}_i$  — радиус-вектор положения начала координат  $i$  — ой системы координат в системе координат  $j$ .

### 2.3 Обратная позиционная задача

Обратная позиционная задача заключается в определении обобщенных координат робота по заданным декартовым координатам характеристической точки инструмента и заданной его ориентации в абсолютном пространстве. В отличие от прямой позиционной задачи, обратная может иметь несколько решений, либо не иметь решений совсем.

Проведем решение обратной задачи для кинематической схемы космического робота, приведенной на Рис. 2.6. Для достижения заданного положения и ориентации инструмента робота достаточно использовать 6 обобщенных координат. Пусть за достижение необходимой ориентации инструмента будут отвечать обобщенные координаты  $q_7, q_8, q_9$ , характеризующие ориентацию звеньев, а за перемещение характеристической точки инструмента в заданные координаты будут отвечать компоненты  $q_4, q_5, q_6$ .

Пусть необходимое положение характеристической точки инструмента манипулятора в абсолютной системе координат задано через вектор  $\mathbf{r}_d = (x_d, y_d, z_d)$ , а ориентация — тремя углами поворота  $\varphi_d, \theta_d, \psi_d$  вокруг осей  $Ox, Oy, Oz$  соответственно. Таким образом, матрица поворота системы координат, связанной с инструментом, по отношению к абсолютной системе координат будет выглядеть следующим образом

$$R_d = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\varphi_d & -S\varphi_d \\ 0 & S\varphi_d & C\varphi_d \end{bmatrix} \cdot \begin{bmatrix} C\theta_d & 0 & S\theta_d \\ 0 & 1 & 0 \\ -S\theta_d & 0 & C\theta_d \end{bmatrix} \cdot \begin{bmatrix} C\psi_d & -S\psi_d & 0 \\ S\psi_d & C\psi_d & 0 \\ 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} C\theta_d C\psi_d - S\theta_d C\varphi_d S\psi_d & S\psi_d C\theta_d + C\psi_d C\varphi_d S\theta_d & S\varphi_d S\theta_d \\ -C\psi_d S\theta_d - S\psi_d C\varphi_d C\theta_d & -S\theta_d S\psi_d + C\theta_d C\varphi_d C\psi_d & S\varphi_d C\theta_d \\ S\psi_d S\varphi_d & -C\psi_d S\varphi_d & C\varphi_d \end{bmatrix}$$

Посчитав значения элементов данной матрицы, можно представить ее в следующем виде

$$R_d = \begin{bmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{bmatrix},$$

столбцы данной матрицы представляют собой проекции ортов системы координат, связанной с инструментом, в желаемой ее ориентации, на оси вспомогательной системы координат, начало которой совпадает с началом системы координат инструмента, а оси параллельны осям абсолютной системы.

Рассмотрим отдельно задачу приведения системы координат инструмента в заданную ориентацию, не рассматривая вопрос о конечном положении ее характеристической точки. Ориентация системы координат, связанной с инструментом, в нашем предположении, зависит только от координат  $q_7, q_8, q_9$  и определяется через них как произведение соответствующих матриц поворота:

$${}^0R_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & Cq_7 & -Sq_7 \\ 0 & Sq_7 & Cq_7 \end{bmatrix} \cdot \begin{bmatrix} Cq_8 & -Sq_8 & 0 \\ Sq_8 & Cq_8 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} Cq_9 & 0 & Sq_9 \\ 0 & 1 & 0 \\ -Sq_9 & 0 & Cq_9 \end{bmatrix}.$$

Следовательно, для того чтобы найти зависимость обобщенных координат от известных компонент желаемой матрицы ориентации, необходимо выразить координаты  $q_7, q_8, q_9$  из системы

$$\begin{bmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & Cq_7 & -Sq_7 \\ 0 & Sq_7 & Cq_7 \end{bmatrix} \cdot \begin{bmatrix} Cq_8 & -Sq_8 & 0 \\ Sq_8 & Cq_8 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} Cq_9 & 0 & Sq_9 \\ 0 & 1 & 0 \\ -Sq_9 & 0 & Cq_9 \end{bmatrix}.$$

Воспользуемся методом обратных преобразований. Домножим обе части уравнения на обратную матрицу поворота на угол  $q_9$ :

$$\begin{bmatrix} Cq_9 & 0 & -Sq_9 \\ 0 & 1 & 0 \\ Sq_9 & 0 & Cq_9 \end{bmatrix} \cdot \begin{bmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & Cq_7 & -Sq_7 \\ 0 & Sq_7 & Cq_7 \end{bmatrix} \cdot \begin{bmatrix} Cq_8 & -Sq_8 & 0 \\ Sq_8 & Cq_8 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

В результате получим

$$\begin{bmatrix} x_x Cq_9 - x_z Sq_9 & y_x Cq_9 - y_z Sq_9 & z_x Cq_9 - z_z Sq_9 \\ x_y & y_y & z_y \\ x_x Sq_9 + x_z Cq_9 & y_x Sq_9 + y_z Cq_9 & z_x Sq_9 + z_z Cq_9 \end{bmatrix} = \begin{bmatrix} Cq_8 & -Sq_8 & 0 \\ Cq_7 Sq_8 & Cq_7 Cq_8 & -Sq_7 \\ Sq_7 Sq_8 & Sq_7 Cq_8 & Cq_7 \end{bmatrix}.$$

Приравнивая компоненты в первой строке и третьем столбце, находим  $q_9$

$$z_x Cq_9 - z_z Sq_9 = 0,$$

откуда

$$q_9 = \text{atan2}(z_x, z_z),$$

$$\text{atan2}(a,b) = \begin{cases} \arctan \frac{a}{b}, & b > 0, \\ \arctan \frac{a}{b} + \pi, & b < 0, a > 0, \\ \arctan \frac{a}{b} - \pi, & b < 0, a < 0. \end{cases}$$

Координату  $q_8$  можно найти, используя уже известное значение  $q_9$  и уравнения, полученные путем приравнивания компонент первой строки и первого столбца, а также первой строки и второго столбца:

$$Cq_8 = x_x Cq_9 - x_z Sq_9,$$

$$Sq_8 = y_z Sq_9 - y_x Cq_9,$$

следовательно

$$q_8 = \text{atan2}(y_z Sq_9 - y_x Cq_9, x_x Cq_9 - x_z Sq_9).$$

Аналогичным образом находим  $q_7$ , используя компоненты на пересечении второй строки и третьего столбца, а также третьей строки и третьего столбца:

$$Cq_7 = z_x Sq_9 + z_z Cq_9,$$

$$Sq_7 = -z_y,$$

отсюда следует, что

$$q_7 = \text{atan2}(-z_y, z_x Sq_9 + z_z Cq_9).$$

Теперь определим зависимость обобщенных координат  $q_4, q_5, q_6$ , отвечающих за линейное перемещение робота.

Пусть повороты звеньев уже произошли, тогда, для того чтобы перевести характеристическую точку в необходимое положение, достаточно сместить центр масс корпуса на вектор разности между желаемым положением характеристической точки и текущим (Рис. 2.7).

Так как значения координат желаемого положения точки нам известны, то осталось только определить текущее положение характеристической точки в абсолютной системе. Для этого воспользуемся прямой позиционной задачей (в предположении, что  $q_1, q_2, q_3 = 0$ ).

$${}^a \mathbf{r} = {}^0 R_1 \cdot \{ {}^1 R_2 \cdot \{ {}^2 R_3 \cdot {}^3 \mathbf{r} + {}^2 \mathbf{p}_3 \} + {}^1 \mathbf{p}_2 \} + {}^0 \mathbf{p}_1 + {}^a \mathbf{p}_0,$$

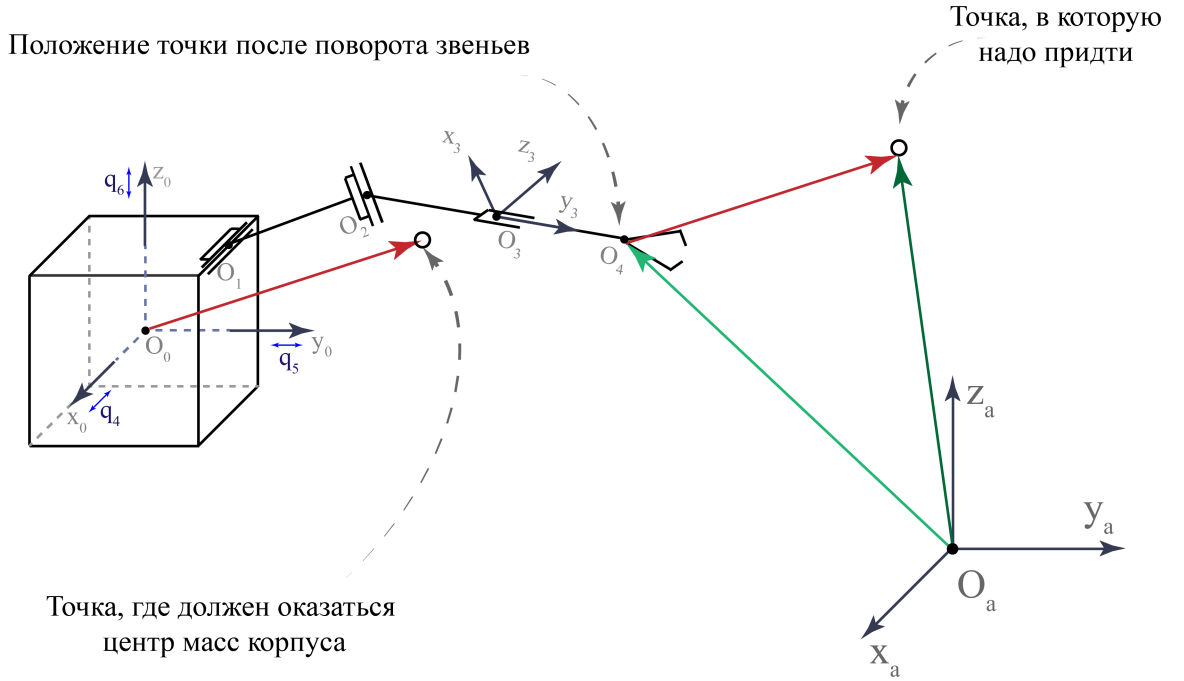


Рисунок 2.7 — Изменение линейных обобщенных координат корпуса робота таким образом,

$$\begin{aligned}
 \begin{bmatrix} {}^a x_3 \\ {}^a y_3 \\ {}^a z_3 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & Cq_7 & -Sq_7 \\ 0 & Sq_7 & Cq_7 \end{bmatrix} \times \\
 &\times \left\{ \begin{bmatrix} Cq_8 & -Sq_8 & 0 \\ Sq_8 & Cq_8 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \left\{ \begin{bmatrix} Cq_9 & 0 & Sq_9 \\ 0 & 1 & 0 \\ -Sq_9 & 0 & Cq_9 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ l_3 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ l_2 \\ 0 \end{bmatrix} \right\} + \begin{bmatrix} 0 \\ l_1 \\ 0 \end{bmatrix} \right\} + \\
 &\quad + \begin{bmatrix} {}^0 x_1 \\ {}^0 y_1 \\ {}^0 z_1 \end{bmatrix} + \begin{bmatrix} q_4(0) \\ q_5(0) \\ q_6(0) \end{bmatrix}
 \end{aligned}$$

Вычислив значения  ${}^a x_3$ ,  ${}^a y_3$ ,  ${}^a z_3$ , можно найти  $q_4$ ,  $q_5$ ,  $q_6$  в виде

$$q_4(t) = q_4(0) + x_d(t) - {}^a x_3,$$

$$q_5(t) = q_5(0) + y_d(t) - {}^a y_3,$$

$$q_6(t) = q_6(0) + z_d(t) - {}^a z_3.$$

Таким образом, найденные значения  $q_4$ ,  $q_5$ ,  $q_6$ ,  $q_7$ ,  $q_8$ ,  $q_9$  через заданные значения  $x_d$ ,  $y_d$ ,  $z_d$ ,  $\varphi_d$ ,  $\theta_d$ ,  $\psi_d$  и будут решением обратной позиционной задачи для свободнолетающего космического робота.



Приведем решение обратной позиционной задачи для наземного робота, кинематическая схема которого представлена на Рис. 2.5.

Пусть необходимое положение и ориентация инструмента робота также задана шестью параметрами:  $x_d, y_d, z_d, \varphi_d, \theta_d, \psi_d$ . Как было показано выше, желаемую ориентацию инструмента можно задать в виде матрицы поворота  $R_d$

$$R_d = \begin{bmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{bmatrix}.$$

Тогда, требуемое положение системы координат, связанной с инструментом робота, можно записать через однородные преобразования следующим образом

$$T_d = \begin{bmatrix} x_x & y_x & z_x & x_d \\ x_y & y_y & z_y & y_d \\ x_z & y_z & z_z & z_d \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Следовательно, чтобы решить обратную позиционную задачу, необходимо выразить обобщенные координаты робота  $q_1, \dots, q_6$  через известные компоненты матрицы  $T_d$ . Проведем решение «вручную».

Инструмент робота должен прийти в заданную точку с заданной ориентацией. Последнее звено поворачивает инструмент вокруг собственной оси. Таким образом, нам необходимо привести с помощью обобщенных координат  $q_1, \dots, q_5$  систему координат, связанную с шестым звеном, так, чтобы ее начало находилось на расстоянии  $l_6$  от заданной точки и чтобы положительное направление оси  $O_6y_6$  пересекало данную точку и было сонаправлено с вектором  $[y_x; y_y; y_z]^T$  из матрицы желаемой ориентации инструмента. Обозначим желаемое положение точки  $O_6$  за  $O_{6d}$ . Исходя из имеющихся данных, ее можно вычислить следующим образом:

$$O_{6d} = \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} - l_6 \cdot \begin{bmatrix} y_x \\ y_y \\ y_z \end{bmatrix}.$$

Можно заметить, что оси  $O_5y_5$  и  $O_6y_6$  всегда остаются сонаправленными. В связи с этим, можно найти желаемое положение точки  $O_5$ :

$$O_{5d} = \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} - (l_5 + l_6) \cdot \begin{bmatrix} y_x \\ y_y \\ y_z \end{bmatrix}.$$

Таким образом, нам необходимо найти такие обобщенные координаты  $q_1, \dots, q_4$ , которые привели бы точку  $O_5$  в данное положение.

Координата  $q_4$  осуществляет поворот вокруг оси звена, то есть она не меняет координаты точки  $O_5$ . Поэтому, рассмотрим задачу приведения характеристической точки ( $O_5$ ) трехзвенного робота (звено 3 и 4 примем за одно звено) в заданную точку.

Чтобы найти  $q_1$  применим метод обратных преобразований.

$$\begin{bmatrix} O_{5dx} \\ O_{5dy} \\ O_{5dz} \\ 1 \end{bmatrix} = {}^a T_1 \cdot {}^1 T_2 \cdot {}^2 T_3 \cdot {}^3 \mathbf{r},$$

$${}^a T_1 = \begin{bmatrix} Cq_1 & -Sq_1 & 0 & {}^a x_1 \\ Sq_1 & Cq_1 & 0 & {}^a y_1 \\ 0 & 0 & 1 & {}^a z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^1 T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & Cq_2 & -Sq_2 & 0 \\ 0 & Sq_2 & Cq_2 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(-q_2) & -S(-q_2) & 0 \\ 0 & S(-q_2) & C(-q_2) & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^2 T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & Cq_3 & -Sq_3 & 0 \\ 0 & Sq_3 & Cq_3 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^3 \mathbf{r} = \begin{bmatrix} 0 \\ l_3 + l_4 \\ 0 \\ 1 \end{bmatrix},$$

В предположении, что начало абсолютной системы координат совпадает с началом системы координат, связанной с первым звеном, получаем:

$$q_1 = \text{atan2}(-O_{5dx}, O_{5dy}). \quad (2.1)$$

Пусть  $q_1$  уже находится в своем конечном положении. Точка  $O_{5d}$  будет находиться в плоскости трехзвенника. Звено 2 и 3 перемещаются в этой плоскости. Таким образом, если мы выразим точку  $O_{5d}$  в системе координат 2-го звена, то мы сведем задачу к обратной задаче для двухзвенника, решение которой известно [28].

Положение точки  $O_{5d}$  в системе координат 2-го звена находится следующим образом:

$${}^2O_{5d} = \begin{bmatrix} C(-q_1) & -S(-q_1) & 0 \\ S(-q_1) & C(-q_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} O_{5dx} \\ O_{5dy} \\ O_{5dz} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ l_1 \end{bmatrix}$$

Тогда, используя решение из [28], и учитывая особенности рассматриваемого робота, получаем выражения для координат  $q_2$  и  $q_3$ :

$$q_2 = -atan2({}^2O_{5dy}, {}^2O_{5dz}) + \arccos((r^2 + l_2^2 - (l_3 + l_4)^2)/(2 \cdot l_2 \cdot r)), \quad (2.2)$$

$$q_3 = \arccos((l_2^2 + (l_3 + l_4)^2 - r^2)/(2 \cdot l_2 \cdot (l_3 + l_4))) - \pi/2 + q_2, \quad (2.3)$$

где  $r = \sqrt{{}^2O_{5dy}^2 + {}^2O_{5dz}^2}$ .

Теперь, когда мы привели точку  $O_5$  в нужное положение, необходимо найти углы  $q_4$ ,  $q_5$  и  $q_6$ , которые приведут инструмент в заданное положение с заданной ориентацией.

Обобщенная координата  $q_5$  отвечает за поворот вокруг оси  $O_5x_5$  и ось 5-го звена всегда сонаправлена с осью 6-го. Следовательно, чтобы привести 6-е звено в заданную ориентацию, необходимо чтобы ось  $O_5x_5$  была перпендикулярна направлению оси  $y$  желаемой ориентации. За направление оси  $O_5x_5$  может отвечать координата  $q_4$ . Получается, что 4-е звено должно повернуться на угол между текущим положением оси  $O_5x_5$  и вектором  $\mathbf{i}_s$ , который будет являться результатом векторного произведения единичного вектора  $\mathbf{j}_{yd} = [y_x, y_y, y_z]^T$  и единичного вектора оси  $O_4y_4 - \mathbf{j}_{y4}$ . В свою очередь, 5-е звено, должно повернуться на угол между полученным в итоге направлением оси  $O_5y_5$  и вектором  $\mathbf{j}_{yd}$ . После чего, координата  $q_6$  приведет инструмент в требуемую ориентацию, изменив свое значение на угол между осью  $O_6x_6$  (вектор  $\mathbf{i}_{x6}$ ) и  $\mathbf{i}_{xd} = [x_x, x_y, x_z]^T$ . В таком случае, координаты  $q_4$ ,  $q_5$ ,  $q_6$  можно вычислить следующим образом:

$$q_4 = (-1)^{k+1} \cdot \arccos \left( \frac{(\mathbf{i}_{x5}, \mathbf{i}_s)}{|\mathbf{i}_{x5}| \cdot |\mathbf{i}_s|} \right), \quad (2.4)$$

$$q_5 = -\arccos \left( \frac{(\mathbf{j}_{yd}, \mathbf{j}_{y5})}{|\mathbf{j}_{yd}| \cdot |\mathbf{j}_{y5}|} \right), \quad (2.5)$$

$$q_6 = (-1)^k \arccos \left( \frac{(\mathbf{i}_{xd}, \mathbf{i}_{x6})}{|\mathbf{i}_{xd}| \cdot |\mathbf{i}_{x6}|} \right), \quad (2.6)$$

где

$$k = \begin{cases} 2, & x_d \geq 0, \\ 1, & x_d < 0, \end{cases}$$

$$\mathbf{i}_{x5} = [Cq_1, Sq_1, 0]^T,$$

$$\mathbf{i}_s = \mathbf{j}_{yd} \times \mathbf{j}_{y4},$$

$$\mathbf{j}_{y4} = [-Cq_3Sq_1, Cq_1Cq_3, Sq_3]^T,$$

$$\mathbf{i}_{x6} = \begin{bmatrix} Cq_1Cq_4 + Sq_1Sq_3Sq_4 \\ Cq_4Sq_1 - Cq_1Sq_3Sq_4 \\ Cq_3Sq_4 \end{bmatrix}.$$

Формулы (2.1)–(2.6) и есть решение обратной позиционной задачи для рассматриваемого наземного робота–манипулятора.

## 2.4 Прямая задача кинематики о скорости

Для построения динамической модели робота необходимо прежде всего найти зависимость линейных и угловых скоростей звеньев от обобщенных скоростей. Поиск будем осуществлять по рекуррентным формулам, представленных в [29]. Проведем решение прямой задачи кинематики о скоростях для космического свободнолетающего робота.

**Угловые скорости.** Угловая скорость твердого тела направлена вдоль оси вращения. Направление вектора угловой скорости зависит от направления вращения, следовательно можно отобразить векторы угловых скоростей на кинематической схеме космического робота следующим образом (Рис. 2.8).

Если использовать теорему о сложении угловых скоростей в сложном движении, то связь между вектором угловой скорости  $i$ -го твердого тела и вектором обобщенных скоростей, выраженной в абсолютной системе координат, можно найти по формуле [29]:

$$\begin{bmatrix} {}^a\omega_x \\ {}^a\omega_y \\ {}^a\omega_z \end{bmatrix}_i = {}^i\Omega \cdot [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_9]^T,$$

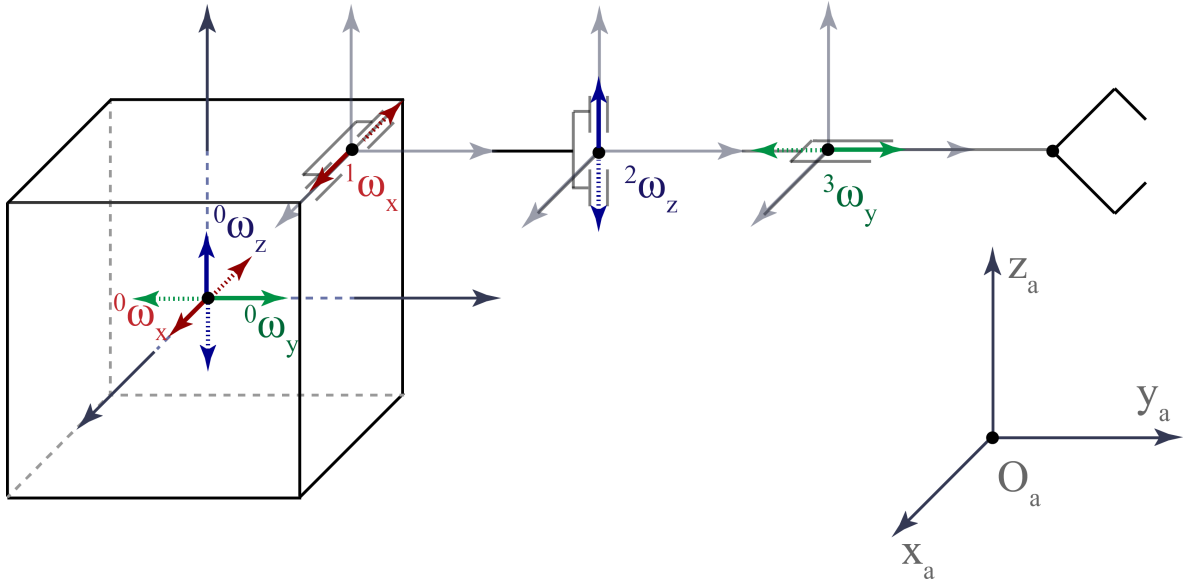


Рисунок 2.8 — Векторы угловых скоростей свободнолетающего космического робота

где

$${}^i\Omega = {}^{i-1}R_i \cdot {}^{i-1}\Omega + {}^iI_\omega,$$

здесь матрица  ${}^iI_\omega$  определяет направление оси вращения  $i$ -го тела.

Рассматриваемый космический робот состоит из четырех тел. Проекция угловой скорости корпуса робота не зависит от угловых скоростей его звеньев. Следовательно в абсолютной системе координат угловая скорость корпуса будет выражаться через вектор обобщенных скоростей следующим образом:

$$\begin{bmatrix} {}^a\omega_x \\ {}^a\omega_y \\ {}^a\omega_z \end{bmatrix}_0 = {}^aR_0 \cdot \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix},$$

$${}^aR_0 = \begin{bmatrix} Cq_2Cq_3 - Sq_2Cq_1Sq_3 & Sq_3Cq_2 + Cq_3Cq_1Sq_2 & Sq_1Sq_2 \\ -Cq_3Sq_2 - Sq_3Cq_1Cq_2 & -Sq_2Sq_3 + Cq_2Cq_1Cq_3 & Sq_1Cq_2 \\ Sq_3Sq_1 & -Cq_3Sq_1 & Cq_1 \end{bmatrix},$$

в таком случае

$${}^0\Omega = \begin{bmatrix} Cq_2Cq_3 - Sq_2Cq_1Sq_3 & Sq_3Cq_2 + Cq_3Cq_1Sq_2 & Sq_1Sq_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -Cq_3Sq_2 - Sq_3Cq_1Cq_2 & -Sq_2Sq_3 + Cq_2Cq_1Cq_3 & Sq_1Cq_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ Sq_3Sq_1 & -Cq_3Sq_1 & Cq_1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Тогда для 1-го звена угловая скорость будет равна

$$\begin{bmatrix} {}^a\omega_x \\ {}^a\omega_y \\ {}^a\omega_z \end{bmatrix}_1 = [{}^0R_1 \cdot {}^0\Omega + {}^1I_\omega] \cdot [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_9]^T,$$

здесь

$${}^0R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & Cq_7 & -Sq_7 \\ 0 & Sq_7 & Cq_7 \end{bmatrix}, \quad {}^1I_\omega = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Тогда

$${}^1\Omega = \begin{bmatrix} {}^1\omega_{11} & {}^1\omega_{12} & {}^1\omega_{13} & 0 & 0 & 0 & 1 & 0 & 0 \\ {}^1\omega_{21} & {}^1\omega_{22} & {}^1\omega_{23} & 0 & 0 & 0 & 0 & 0 & 0 \\ {}^1\omega_{31} & {}^1\omega_{32} & {}^1\omega_{33} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$${}^1\omega_{11} = Cq_2Cq_3, \quad {}^1\omega_{12} = -Cq_1Sq_3 - Cq_3Sq_1Sq_2, \quad {}^1\omega_{13} = Sq_1Sq_3 - Cq_1Cq_3Sq_2,$$

$${}^1\omega_{21} = Cq_2Cq_7Sq_3 - Sq_2Sq_7, \quad {}^1\omega_{22} = Cq_7(Cq_1Cq_3 - Sq_1Sq_2Sq_3) - Cq_2Sq_1Sq_7,$$

$${}^1\omega_{23} = -Cq_7(Cq_3Sq_1 + Cq_1Sq_2Sq_3) - Cq_1Cq_2Sq_7,$$

$${}^1\omega_{31} = Cq_7Sq_2 + Cq_2Sq_3Sq_7, \quad {}^1\omega_{32} = Sq_7(Cq_1Cq_3 - Sq_1Sq_2Sq_3) + Cq_2Cq_7Sq_1,$$

$${}^1\omega_{33} = Cq_1Cq_2Cq_7 - Sq_7(Cq_3Sq_1 + Cq_1Sq_2Sq_3).$$

Аналогично для 2-го и 3-го звена

$${}^2\Omega = \begin{bmatrix} {}^2\omega_{11} & {}^2\omega_{12} & {}^2\omega_{13} & 0 & 0 & 0 & Cq_8 & 0 & 0 \\ {}^2\omega_{21} & {}^2\omega_{22} & {}^2\omega_{23} & 0 & 0 & 0 & Sq_8 & 0 & 0 \\ {}^2\omega_{31} & {}^2\omega_{32} & {}^2\omega_{33} & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$${}^3\Omega = \begin{bmatrix} {}^3\omega_{11} & {}^3\omega_{12} & {}^3\omega_{13} & 0 & 0 & 0 & Cq_8Cq_9 & -Sq_9 & 0 \\ {}^3\omega_{21} & {}^3\omega_{22} & {}^3\omega_{23} & 0 & 0 & 0 & Sq_8 & 0 & 1 \\ {}^3\omega_{31} & {}^3\omega_{32} & {}^3\omega_{33} & 0 & 0 & 0 & Cq_8Sq_9 & Cq_9 & 0 \end{bmatrix}.$$

**Линейные скорости.** Найдем зависимость линейных скоростей характеристически точек робота от вектора обобщенных скоростей. За линейные перемещения робота отвечают обобщенные координаты  $q_4$ ,  $q_5$  и  $q_6$  его корпуса. К тому же, при вращении звеньев и основания робота, характеристические точки будут обладать линейными скоростями, перпендикулярные направлению вектора угловой скорости и высчитываемые по формуле Эйлера. Представим в общем случае эти скорости на кинематической схеме космического робота (Рис. 2.9):

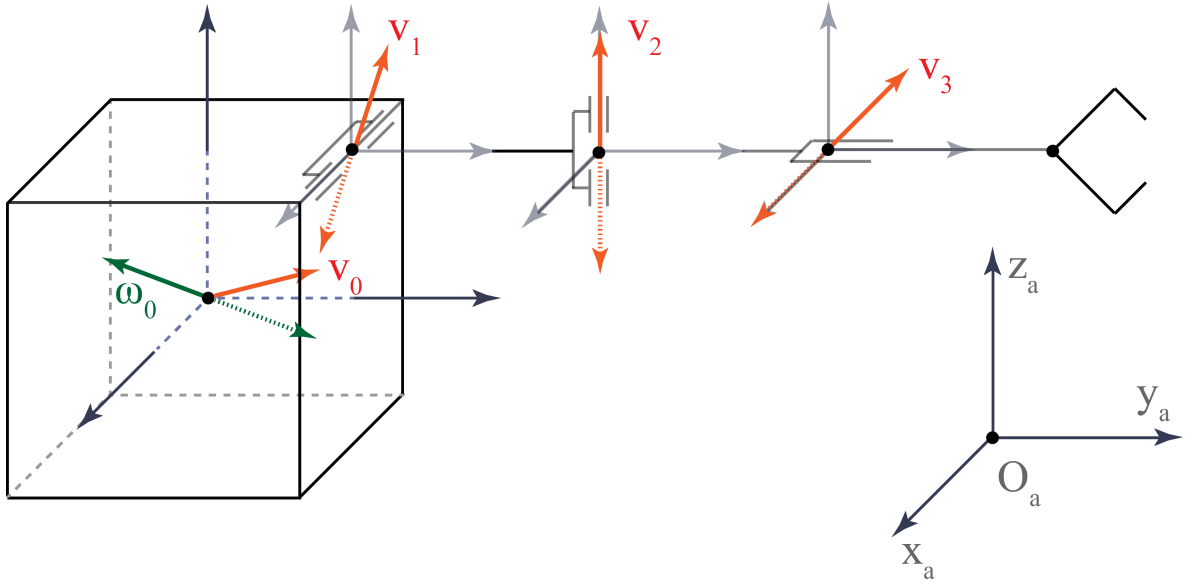


Рисунок 2.9 — Векторы линейных скоростей характеристических точек космического робота (по отношению к предыдущему звену)

Связь линейных и обобщенных скоростей, по отношению к  $i$ -ой характеристической точке («0» – центр масс корпуса, «1, 2, 3» – начало 1, 2, 3 звена, «4» – характеристическая точка инструмента), определяется в следующем виде (используя теорему о сложении скоростей в сложном движении и формулу распределения скоростей в твердом теле):

$$\begin{bmatrix} {}^a v_x \\ {}^a v_y \\ {}^a v_z \end{bmatrix}_i = {}^i V \cdot [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_9]^T,$$

где

$${}^i V = {}^{i-1} R_i \cdot {}^{i-1} V + {}^{i-1} R_i \cdot L_{i-1} \cdot {}^{i-1} \Omega,$$

здесь кососимметрическая матрица  $L_{i-1}$  определяет расстояние от  $i$ -ой точки до оси вращения предыдущего тела.

По отношению к 0-ой характеристической точке:

$$\begin{bmatrix} {}^a v_x \\ {}^a v_y \\ {}^a v_z \end{bmatrix}_0 = {}^a R_0 \cdot \begin{bmatrix} \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix}.$$

Следовательно

$${}^0 V = \begin{bmatrix} 0 & 0 & 0 & Cq_2 Cq_3 - Sq_2 Cq_1 Sq_3 & Sq_3 Cq_2 + Cq_3 Cq_1 Sq_2 & Sq_1 Sq_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -Cq_3 Sq_2 - Sq_3 Cq_1 Cq_2 & -Sq_2 Sq_3 + Cq_2 Cq_1 Cq_3 & Sq_1 Cq_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & Sq_3 Sq_1 & -Cq_3 Sq_1 & Cq_1 & 0 & 0 & 0 \end{bmatrix}.$$

Для 1-ой характеристической точки:

$${}^1V = {}^0R_1 \cdot {}^0V + {}^0R_1 \cdot L_0 \cdot {}^0\Omega,$$

$${}^0R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & Cq_7 & -Sq_7 \\ 0 & Sq_7 & Cq_7 \end{bmatrix}, \quad L_0 = \begin{bmatrix} 0 & -{}^a z_1 & {}^a y_1 \\ {}^a z_1 & 0 & 0 \\ -{}^a y_1 & 0 & 0 \end{bmatrix}.$$

Аналогично для 2, 3 и 4-ой характеристической точки:

$${}^2V = {}^1R_2 \cdot {}^1V + {}^1R_2 \cdot L_1 \cdot {}^1\Omega,$$

$${}^1R_2 = \begin{bmatrix} Cq_8 & -Sq_8 & 0 \\ Sq_8 & Cq_8 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad L_1 = \begin{bmatrix} 0 & 0 & l_1 \\ 0 & 0 & 0 \\ -l_1 & 0 & 0 \end{bmatrix}$$

$${}^3V = {}^2R_3 \cdot {}^2V + {}^2R_3 \cdot L_2 \cdot {}^2\Omega,$$

$${}^2R_3 = \begin{bmatrix} Cq_9 & 0 & Sq_9 \\ 0 & 1 & 0 \\ -Sq_9 & 0 & Cq_9 \end{bmatrix}, \quad L_2 = \begin{bmatrix} 0 & 0 & l_2 \\ 0 & 0 & 0 \\ -l_2 & 0 & 0 \end{bmatrix},$$

$${}^4V = {}^3V + {}^2R_3 \cdot L_3 \cdot {}^3\Omega,$$

$$L_3 = \begin{bmatrix} 0 & 0 & l_3 \\ 0 & 0 & 0 \\ -l_3 & 0 & 0 \end{bmatrix}.$$

Теперь найдем зависимость между векторами угловых и линейных скоростей с вектором обобщенных скоростей для наземного манипулятора. Расчеты будут проводиться аналогичным образом, поэтому просто приведем распределение скоростей на кинематической схеме (Рис. 2.10, 2.11) и матрицы  $\Omega$  и  $V$ .

### Угловые скорости.

$$\begin{bmatrix} {}^a\omega_x \\ {}^a\omega_y \\ {}^a\omega_z \end{bmatrix}_i = {}^i\Omega \cdot [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_6]^T,$$

$${}^1\Omega = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$



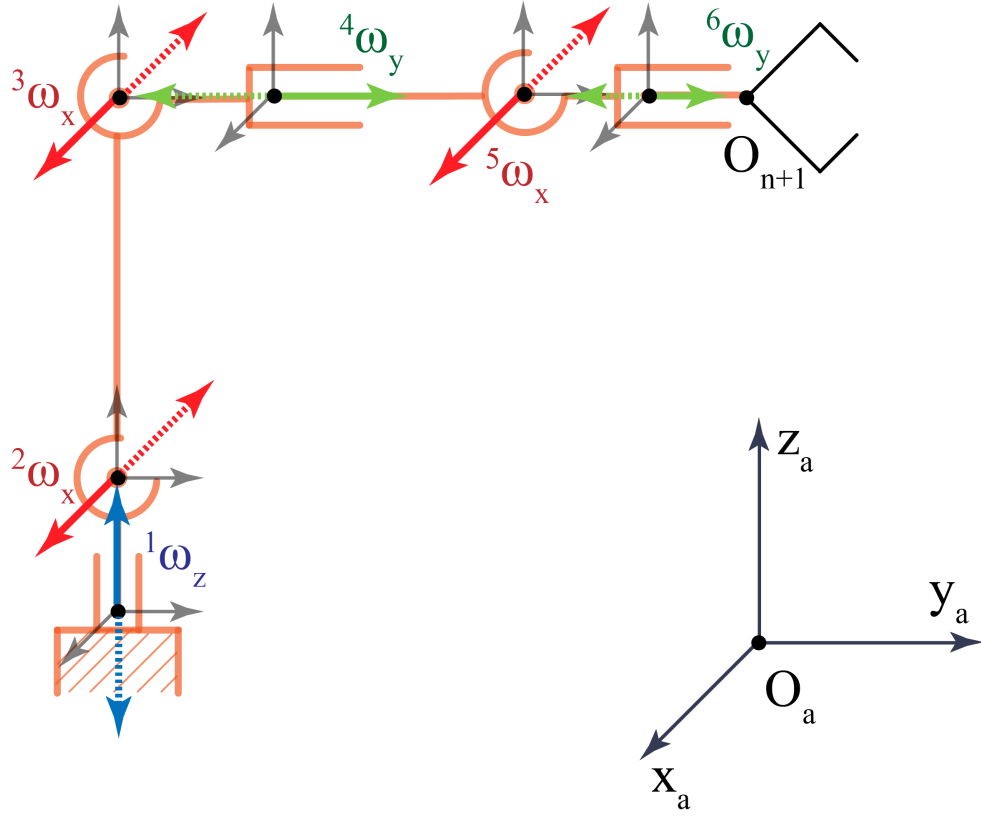


Рисунок 2.10 — Векторы угловых скоростей звеньев наземного робота

$${}^2\Omega = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -Sq_2 & 0 & 0 & 0 & 0 & 0 \\ Cq_2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$${}^3\Omega = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ -Cq_2Sq_3 - Cq_3Sq_2 & 0 & 0 & 0 & 0 & 0 \\ Cq_2Cq_3 - Sq_2Sq_3 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$${}^4\Omega = \begin{bmatrix} -Sq_4(Cq_2Cq_3 - Sq_2Sq_3) & Cq_4 & Cq_4 & 0 & 0 & 0 \\ -Cq_2Sq_3 - Cq_3Sq_2 & 0 & 0 & 1 & 0 & 0 \\ Cq_4(Cq_2Cq_3 - Sq_2Sq_3) & Sq_4 & Sq_4 & 0 & 0 & 0 \end{bmatrix},$$

$${}^5\Omega = \begin{bmatrix} {}^5\omega_{11} & Cq_4 & Cq_4 & 0 & 1 & 0 \\ {}^5\omega_{21} & -Sq_4Sq_5 & Sq_4Sq_5 & Cq_5 & 0 & 0 \\ {}^5\omega_{31} & Cq_5Sq_4 & Cq_5Sq_4 & Sq_5 & 0 & 0 \end{bmatrix},$$

$${}^5\omega_{11} = -Sq_4(Cq_2Cq_3 - Sq_2Sq_3),$$

$${}^5\omega_{21} = -Cq_5(Cq_2Sq_3 + Cq_3Sq_2) - Cq_4Sq_5(Cq_2Cq_3 - Sq_2Sq_3),$$

$${}^5\omega_{31} = Cq_4Cq_5(Cq_2Cq_3 - Sq_2Sq_3) - Sq_5(Cq_2Sq_3 + Cq_3Sq_2),$$

$${}^6\Omega = \begin{bmatrix} {}^6\omega_{11} & Cq_4Cq_6 - Cq_5Sq_4Sq_6 & Cq_4Cq_6 - Cq_5Sq_4Sq_6 & -Sq_5Sq_6 & Cq_6 & 0 \\ {}^6\omega_{21} & -Sq_4Sq_5 & Sq_4Sq_5 & Cq_5 & 0 & 1 \\ {}^6\omega_{31} & Cq_4Sq_6 + Cq_5Cq_6Sq_4 & Cq_4Sq_6 + Cq_5Cq_6Sq_4 & Cq_6Sq_5 & Sq_6 & 0 \end{bmatrix},$$

$${}^6\omega_{11} = Sq_6(Sq_5(Cq_2Sq_3 + Cq_3Sq_2) - Cq_4Cq_5(Cq_2Cq_3 - Sq_2Sq_3)) - Cq_6Sq_4(Cq_2Cq_3 - Sq_2Sq_3),$$

$${}^6\omega_{21} = -Cq_5(Cq_2Sq_3 + Cq_3Sq_2) - Cq_4Sq_5(Cq_2Cq_3 - Sq_2Sq_3),$$

$${}^6\omega_{31} = -Cq_6(Sq_5(Cq_2Sq_3 + Cq_3Sq_2) - Cq_4Cq_5(Cq_2Cq_3 - Sq_2Sq_3)) - Sq_4Sq_6(Cq_2Cq_3 - Sq_2Sq_3),$$

**Линейные скорости.**

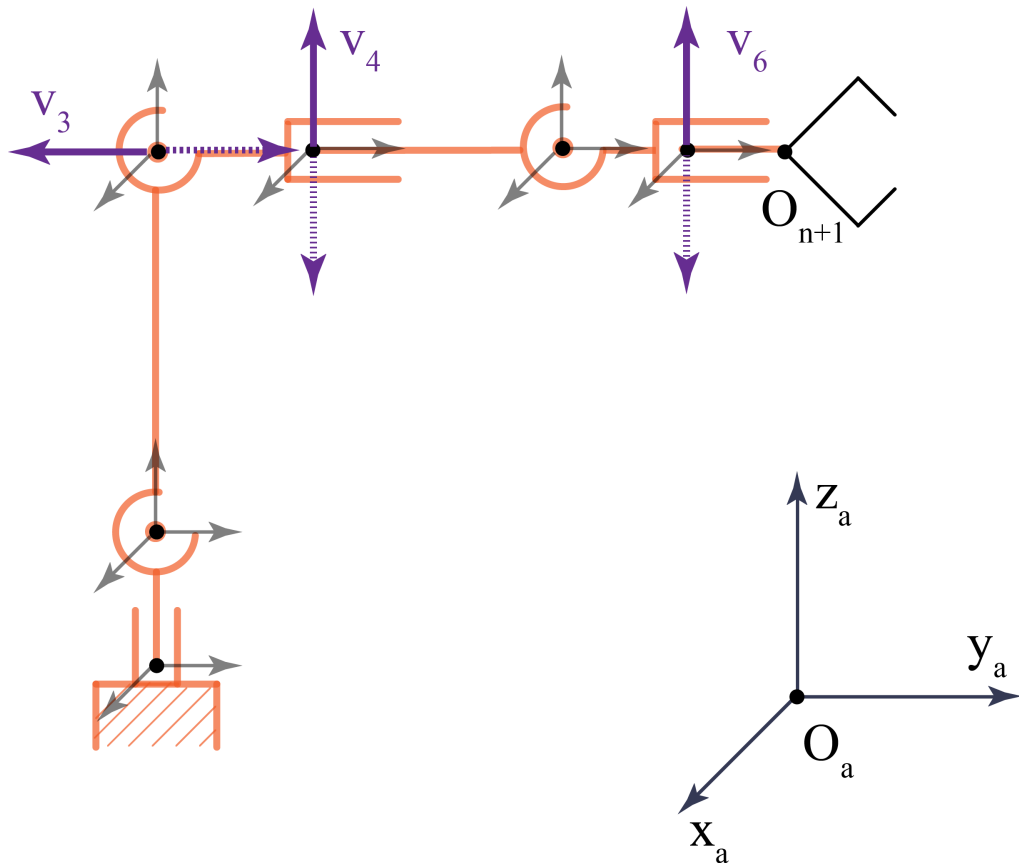


Рисунок 2.11 — Векторы линейных скоростей характеристических точек наземного робота (по отношению к предыдущему звену)

$$\begin{bmatrix} {}^a v_x \\ {}^a v_y \\ {}^a v_z \end{bmatrix}_i = {}^i V \cdot [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_6]^T,$$

$$\begin{aligned}
\begin{bmatrix} {}^a v_x \\ {}^a v_y \\ {}^a v_z \end{bmatrix}_1 &= \begin{bmatrix} {}^a v_x \\ {}^a v_y \\ {}^a v_z \end{bmatrix}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \\
{}^3 V &= {}^2 R_3 \cdot L_2 \cdot {}^2 \Omega, \\
{}^2 R_3 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & Cq_3 & -Sq_3 \\ 0 & Sq_3 & Cq_3 \end{bmatrix}, \quad L_2 = \begin{bmatrix} 0 & -l_2 & 0 \\ l_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\
{}^4 V &= {}^3 R_4 \cdot {}^3 V + {}^3 R_4 \cdot L_3 \cdot {}^3 \Omega, \\
{}^3 R_4 &= \begin{bmatrix} Cq_4 & 0 & Sq_4 \\ 0 & 1 & 0 \\ -Sq_4 & 0 & Cq_4 \end{bmatrix}, \quad L_3 = \begin{bmatrix} 0 & 0 & l_3 \\ 0 & 0 & 0 \\ -l_3 & 0 & 0 \end{bmatrix}, \\
{}^5 V &= {}^4 R_5 \cdot {}^4 V + {}^4 R_5 \cdot L_4 \cdot {}^4 \Omega, \\
{}^4 R_5 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & Cq_5 & -Sq_5 \\ 0 & Sq_5 & Cq_5 \end{bmatrix}, \quad L_4 = \begin{bmatrix} 0 & 0 & l_4 \\ 0 & 0 & 0 \\ -l_4 & 0 & 0 \end{bmatrix}, \\
{}^6 V &= {}^5 R_6 \cdot {}^5 V + {}^5 R_6 \cdot L_5 \cdot {}^5 \Omega, \\
{}^5 R_6 &= \begin{bmatrix} Cq_6 & 0 & Sq_6 \\ 0 & 1 & 0 \\ -Sq_6 & 0 & Cq_6 \end{bmatrix}, \quad L_5 = \begin{bmatrix} 0 & 0 & l_5 \\ 0 & 0 & 0 \\ -l_5 & 0 & 0 \end{bmatrix}, \\
{}^7 V &= {}^6 V + {}^5 R_6 \cdot L_6 \cdot {}^6 \Omega, \\
L_6 &= \begin{bmatrix} 0 & 0 & l_6 \\ 0 & 0 & 0 \\ -l_6 & 0 & 0 \end{bmatrix}.
\end{aligned}$$

**Вектор квазискоростей.** Если обозначить за  $\dot{\mathbf{x}}_i$  вектор угловых и линейных скоростей  $i$ -го звена ( $\dot{\mathbf{x}}_i = [{}^a \omega_x, {}^a \omega_y, {}^a \omega_z, {}^a v_x, {}^a v_y, {}^a v_z]^T$ ), то связь между вектором квазискоростей и вектором обобщенных скоростей можно представить в виде

$$\dot{\mathbf{x}} = C \cdot \dot{\mathbf{q}}, \quad (2.7)$$

где

$$\dot{\mathbf{x}} = [\dot{\mathbf{x}}_0, \dots, \dot{\mathbf{x}}_n]^T, \quad C = [{}^0 \Omega, {}^0 V, \dots, {}^n \Omega, {}^n V]^T.$$

## 2.5 Прямая задача динамики

Для дальнейшего исследования работоспособности системы билатерального управления методом обучения, необходимо построить динамическую модель космического свободнолетающего робота и наземного манипулятора FANUC.

Прямая задача динамики состоит в поиске сил и моментов по заданному движению системы и масс–инерционных характеристик ее элементов. Проведем решение прямой задачи динамики для космического и наземного роботов.

Уравнения движения роботов будем искать в виде уравнения Лагранже 2-го рода:

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{\mathbf{q}}} - \frac{\partial T}{\partial \mathbf{q}} = \mathbf{Q}.$$

Кинетическая энергия системы равна сумме кинетических энергий ее элементов. Найдем кинетическую энергию для космического свободнолетающего робота (кинематическая схема – Рис. 2.6). Робот состоит из четырех тел – корпуса и трех звеньев, следовательно, посчитав значение кинетической энергии для каждого тела, мы найдем кинетическую энергию всей системы:

$$T = \sum_{i=0}^3 T_i.$$

Кинетическая энергия твердого тела в общем случае движения равна:

$$T_i = \frac{1}{2}(m_i \mathbf{v}_i^2 + I_i \boldsymbol{\omega}_i^2),$$

что в матричном представлении имеет вид

$$T_i = \dot{\mathbf{x}}_i^T B_i \dot{\mathbf{x}}_i,$$

где  $\dot{\mathbf{x}}$  — вектор квазискоростей тела (2.7), а матрица  $B$  характеризует масс–инерционные его характеристики. Матрицы  $B$  для твердых тел космического робота будут иметь следующий вид (моментами инерций по отношению

к осям звеньев мы пренебрегаем)

$$B_0 = \begin{bmatrix} J_0^1 & 0 & 0 & 0 & 0 & 0 \\ 0 & J_0^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & J_0^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_0 & 0 \\ 0 & 0 & 0 & 0 & 0 & m_0 \end{bmatrix},$$

$$B_i = \begin{bmatrix} I_i^1 & 0 & 0 & 0 & 0 & m_i y_i^c \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_i^3 & -m_i y_i^c & 0 & 0 \\ 0 & 0 & -m_i y_i^c & m_i & 0 & 0 \\ 0 & 0 & 0 & 0 & m_i & 0 \\ m_i y_i^c & 0 & 0 & 0 & 0 & m_i \end{bmatrix}, i = 1, 2, 3,$$

здесь  $y_i^c$  — координата центра масс звена по оси  $Oy$  (в системе координат  $i$ -го звена).

В таком случае кинетическая энергия всего космического робота будет находиться по формуле

$$T_i = \dot{\mathbf{x}}^T B \dot{\mathbf{x}},$$

где

$$B = \begin{bmatrix} B_0 & 0 & 0 & 0 \\ 0 & B_1 & 0 & 0 \\ 0 & 0 & B_2 & 0 \\ 0 & 0 & 0 & B_3 \end{bmatrix}.$$

Для наземного робота матрицы  $B$  будут иметь следующий вид

$$B_i = \begin{bmatrix} I_i^1 & 0 & 0 & 0 & -m_i z_i^c & 0 \\ 0 & I_i^2 & 0 & m_i z_i^c & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_i z_i^c & 0 & m_i & 0 & 0 \\ -m_i z_i^c & 0 & 0 & 0 & m_i & 0 \\ 0 & 0 & 0 & 0 & 0 & m_i \end{bmatrix}, i = 1, 2,$$

$$B_i = \begin{bmatrix} I_i^1 & 0 & 0 & 0 & 0 & m_i y_i^c \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_i^3 & -m_i y_i^c & 0 & 0 \\ 0 & 0 & -m_i y_i^c & m_i & 0 & 0 \\ 0 & 0 & 0 & 0 & m_i & 0 \\ m_i y_i^c & 0 & 0 & 0 & 0 & m_i \end{bmatrix}, i = 3, 4, 5, 6,$$

здесь  $z_i^c$  — координата центра масс звена по оси  $Oz$  (в системе координат  $i$ -го звена), а  $y_i^c$  — координата центра масс звена по оси  $Oy$ .

Если мы подставим выражение квазискоростей через обобщенные скорости, то можем представить кинетическую энергию системы в виде

$$T = \frac{1}{2} \dot{\mathbf{q}}^T A(\mathbf{q}) \dot{\mathbf{q}},$$

где

$$A = C^T B C.$$

Теперь подставим полученное выражение кинетической энергии через обобщенные скорости в уравнение Лагранжа 2-го рода:

$$\frac{d}{dt} \frac{\partial [\frac{1}{2} \dot{\mathbf{q}}^T A \dot{\mathbf{q}}]}{\partial \dot{\mathbf{q}}} - \frac{\partial [\frac{1}{2} \dot{\mathbf{q}}^T A \dot{\mathbf{q}}]}{\partial \mathbf{q}} = \mathbf{Q},$$

$$\frac{\partial [\frac{1}{2} \dot{\mathbf{q}}^T A \dot{\mathbf{q}}]}{\partial \dot{\mathbf{q}}} = A(\mathbf{q}) \dot{\mathbf{q}},$$

$$\frac{d}{dt} (A(\mathbf{q}) \dot{\mathbf{q}}) = A(\mathbf{q}) \ddot{\mathbf{q}} + \dot{A}(\mathbf{q}) \dot{\mathbf{q}},$$

В случае 9-ти степенного космического робота (для наземного будет 6 компонент)

$$\dot{A}(\mathbf{q}) = \begin{bmatrix} \sum_{j=1}^9 \frac{\partial a_{11}(q)}{\partial q_j} \dot{q}_j & \dots & \sum_{j=1}^9 \frac{\partial a_{19}(q)}{\partial q_j} \dot{q}_j \\ \dots & \dots & \dots \\ \sum_{j=1}^9 \frac{\partial a_{91}(q)}{\partial q_j} \dot{q}_j & \dots & \sum_{j=1}^9 \frac{\partial a_{99}(q)}{\partial q_j} \dot{q}_j \end{bmatrix},$$

$$\frac{\partial [\frac{1}{2} \dot{\mathbf{q}}^T A \dot{\mathbf{q}}]}{\partial \mathbf{q}} = \frac{1}{2} \dot{\mathbf{q}}^T \left[ \frac{\partial A(\mathbf{q})}{\partial \mathbf{q}} \right] \dot{\mathbf{q}},$$

$$\frac{\partial A(\mathbf{q})}{\partial \mathbf{q}} = \begin{bmatrix} \sum_{j=1}^9 \frac{\partial a_{11}(q)}{\partial q_j} & \cdots & \sum_{j=1}^9 \frac{\partial a_{19}(q)}{\partial q_j} \\ \cdots & \cdots & \cdots \\ \sum_{j=1}^9 \frac{\partial a_{91}(q)}{\partial q_j} & \cdots & \sum_{j=1}^9 \frac{\partial a_{99}(q)}{\partial q_j} \end{bmatrix},$$

Тогда уравнения движения в общем виде можно записать следующим образом (для наземного робота вместо 9-ти будет 6)

$$A(\mathbf{q})\ddot{\mathbf{q}} + \sum_{s=1}^9 [\dot{\mathbf{q}}^T D_s(\mathbf{q}) \dot{\mathbf{q}}] \mathbf{e}_s = \mathbf{Q}, \quad (2.8)$$

где

$$D_s(\mathbf{q}) = (d_{it}^s(\mathbf{q})), i, t, s = 1, \dots, n,$$

$$d_{it}^s(\mathbf{q}) = \frac{1}{2} \left( \frac{\partial a_{is}}{\partial q_t} + \frac{\partial a_{ts}}{\partial q_i} - \frac{\partial a_{it}}{\partial q_s} \right),$$

где  $\mathbf{e}_s$  — единичный вектор–столбец, у которого единица стоит в  $s$  – ой строке.

Подставив в (3.1) желаемый закон изменения обобщенных координат, найдем обобщенные силы, которые обеспечивают желаемое движение робота. Следовательно формула (3.1) и есть решение прямой задачи динамики.

## 2.6 Метод адаптации к изменениям рабочей среды

Для успешного выполнения рабочим инструментом робота операции с предметом, необходимо, чтобы в процессе реализации операции положение предмета относительно рабочего инструмента, однозначно определяемое положением его характерных точек (не менее трех), а также величина силы взаимодействия инструмента с предметом были идентичны силам и положению их моделей в процессе обучения. То есть, необходимо чтобы паспорт операции, полученный с помощью моделей в процессе обучения, совпадал с паспортом реализуемой операции. Таким образом, для осуществления предлагаемого подхода необходимо оснастить модель робота набором сенсоров, который обеспечивает возможность формирования на этапе обучения «паспортных» траекторий, используемых после передачи их космическому роботу в качестве программных.

Для отслеживания этих траекторий система управления реального космического робота должна использовать информацию аналогичную той, которая была использована для построения «паспортных» траекторий. Это требует оснащения реального космического робота тем же сенсорным обеспечением, которое было использовано на его модели и установленного аналогичным образом.

В случае достаточно точного отслеживания вышеупомянутых программных траекторий рабочим инструментом реального робота, требуемая операция с предметом в недетерминированной внешней среде оказывается выполненной независимо от возможной неточности модели внешней среды, в которой осуществлялся процесс обучения, обусловленной в основном отличием реальных позиций объектов в базовой системе координат корпуса робота от их модельных позиций. Для повышения точности отслеживания программных траекторий, вместо традиционного регулирования «по отклонению» целесообразно использовать более совершенные методы управления, например, комбинированное управление, когда сигнал рассогласования формируется как разность между программными текущими значениями и дополняется сигналом пропорциональным величине возмущения порождающее это отклонение.

Рассмотрим метод адаптации, основанный на поиске характерных точек объектов с помощью лазерных дальномеров, которые необходимы для вычисления величины отличия реального положения и поворота тела от его модельного положения. Местом установки дальномеров будет конец последнего звена манипулятора, жестко связанный с корпусом запястного силомоментного сенсора (Рис. 3.1). Подвижная, смещаемая часть конструкции сенсора обычно скрепляется с рабочим инструментом манипулятора, что позволяет схвату незначительно сместиться относительно последнего звена из-за деформации упругой конструкции датчика под действием силы, приложенной к схвату, при взаимодействии его с предметами внешней среды. Таким образом, положения характерных точек предметов внешней среды, представленных в системе координат последнего звена (корпуса сенсора) незначительно отличается от их положения в системе координат схвата и совпадает с ним при отсутствии упругой деформации датчика, то есть при отсутствии взаимодействия схвата с объектами внешней среды.

При использовании лазерных дальномеров, характерными считаются точки отражения луча лазера от поверхности моделей предметов внешней среды, полученных в процессе обучения, и точки отражения луча от реальных



предметов, полученные при реализации требуемого действия. Каждой точке соответствуют три компоненты вектора положения в системе координат последнего звена манипулятора. Представление вектора положения точки возможно с использованием как декартовых координат, так и сферических координат. В последнем случае это два угла, определяющие направление луча лазера, третьей координатой является длина луча до точки ее отражения. Соответствующими друг другу считается пара характерных точек, полученных при обучении и при реализации действия, которые имеют две одинаковые угловые координаты.

Наиболее часто встречающееся отличие реальной внешней среды от ее модели состоит не в разнице пространственных конфигураций предметов, а в основном только в относительном смещении и повороте друг относительно друга. Поэтому, с целью достижения положения рабочего инструмента относительно поверхности внешней среды идентичного их относительному модельному положению, необходимо соответствующим образом повернуть и сдвинуть схват. Рисунок 3.1. поясняет это. На нем, для удобства восприятия представлен не трехмерный, а двухмерный случай внешней среды. Для того чтобы положение схвата относительно реальной поверхности было идентичным положению его модели относительно модельной поверхности, необходимо чтобы векторы положения по крайней мере двух точек (в трехмерном случае трех точек, не лежащих на одной прямой) были равными, то есть  $\mathbf{X}_d^i = \mathbf{X}^i, i = 1, 2, 3, \dots$

Для практической реализации эффекта совмещения соответствующих друг другу характерных точек в процессе управления, целесообразно использовать в законе управления для локальной системы управления космического робота член, являющийся функцией величины рассогласования между желаемым  $\mathbf{X}_d^i$  и текущим  $\mathbf{X}^i$  векторами положения характерных точек внешней среды. А для поддержания вектора силы взаимодействия рабочего инструмента с предметами внешней среды  $\mathbf{Q}$  близкого к желаемому  $\mathbf{Q}^d$ , полученному при обучении, нужно использовать в законе управления, дополнительно к вышеупомянутому, член, являющийся функцией величины рассогласования между векторами  $\mathbf{Q}^d$  и  $\mathbf{Q}$ . Закон имеет нижеследующий вид

$$U = \frac{1}{n} \sum_{i=1}^n \left( \frac{\partial \mathbf{X}^i}{\partial \mathbf{q}} \right)^T K_p (\mathbf{X}_d^i - \mathbf{X}^i) + K_Q (\mathbf{Q}^d - \mathbf{Q}),$$

где  $\mathbf{Q}^d$  и  $\mathbf{Q}$  — векторы желаемого и текущего значений сил взаимодействия схвата с объектом внешней среды.

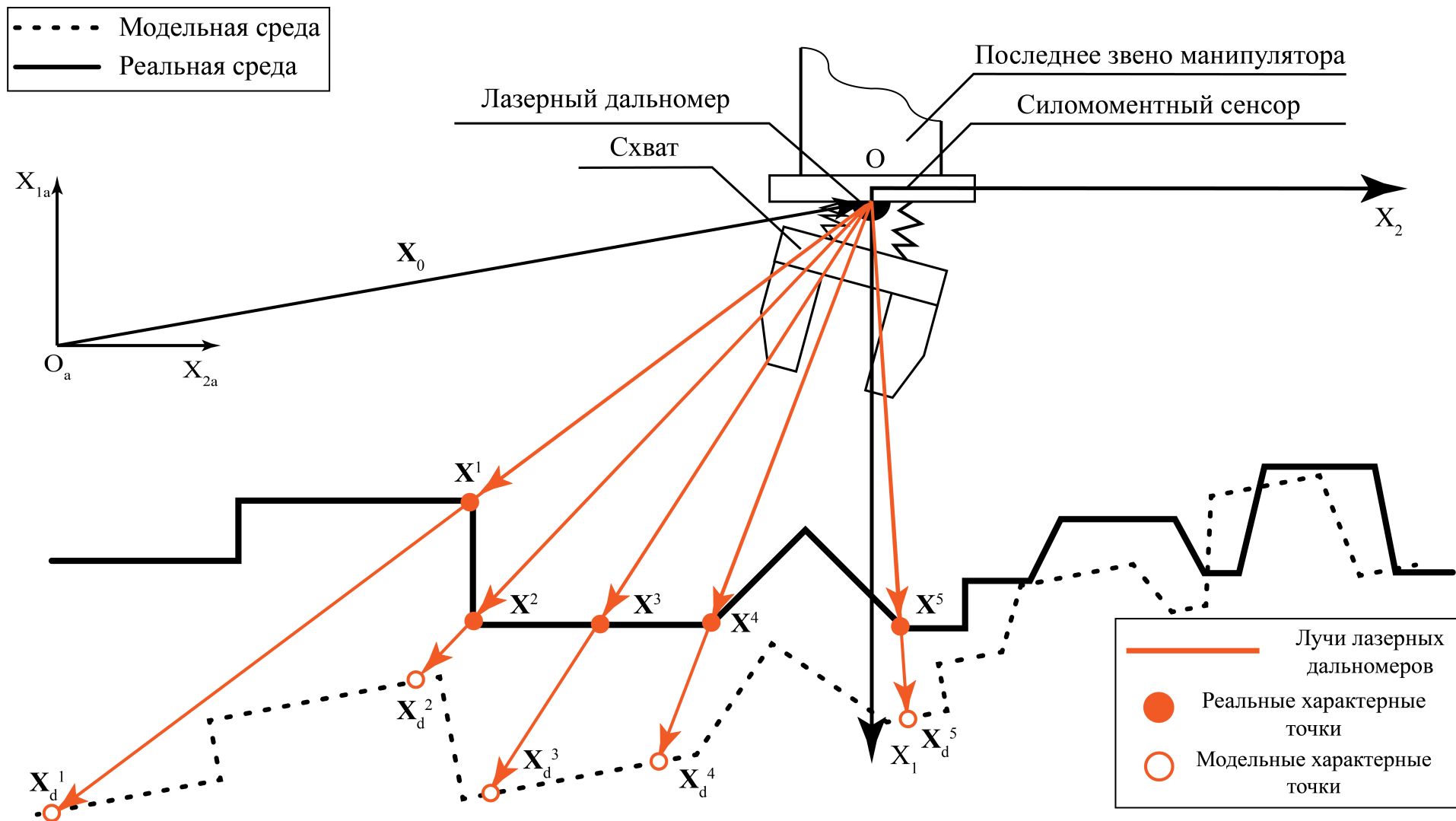


Рисунок 2.12 — Поиск характерных точек среды с помощью лазерных дальномеров

## Глава 3. Программная реализация

Для верификации результатов, полученных во второй и третьей главах, проведем компьютерное моделирование в математическом пакете Matlab.

### 3.1 Кинематическое моделирование

#### Прямая позиционная задача

Пусть заданы следующие начальные и конечные значения обобщенных координат свободнолетающего космического робота (кинематическая схема – Рис. 2.6):

$$\mathbf{q}_0 = \left[ 0 \ 0 \ -\pi/2 \ 0 \ 2 \ 1 \ 0 \ 0 \ 0 \right]^T,$$

$$\mathbf{q}_k = \left[ -\pi/6 \ 0 \ \pi/4 \ 0 \ -2 \ 5 \ 2\pi/3 \ -\pi/2 \ 0 \right]^T$$

Введем следующую зависимость изменения обобщенных координат от времени:

$$\dot{q}_i(t) = \dot{\lambda}(t)(q_i^1 - q_i^0), \quad i = 1, \dots, 9, \quad (3.1)$$

где

$$\int_{t^0}^{t^1} \dot{\lambda}(t) dt = 1,$$

Пусть

$$\dot{\lambda}(t) = \frac{1}{T}$$

Тогда получим следующее движение свободнолетающего космического робота (Рис. 3.1).

Код программы, решающий прямую позиционную задачу для свободнолетающего космического робота представлен в Приложении 1.

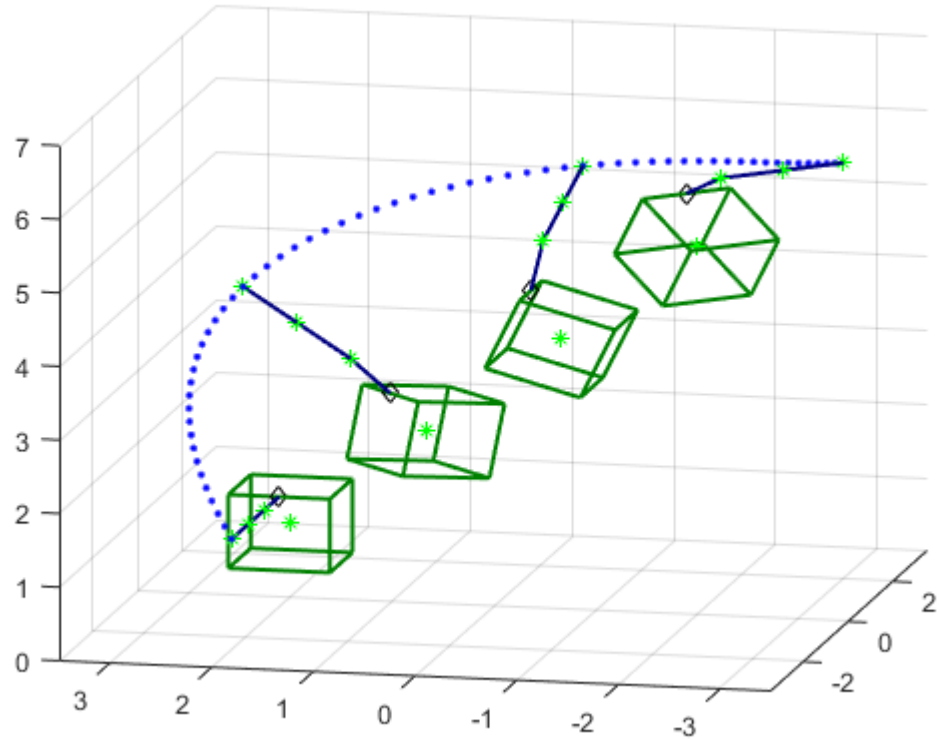


Рисунок 3.1 — Решение прямой позиционной задачи для космического робота

Представим компьютерное моделирование прямой позиционной задачи для наземного манипулятора FANUC. В качестве начальных и конечных значений возьмем следующие координаты:

$$\mathbf{q}_0 = \mathbf{0},$$

$$\mathbf{q}_k = \left[ 3\pi/2 \quad -\pi/8 \quad \pi/3 \quad \pi/6 \quad -\pi/2 \quad \pi/4 \right]^T$$

Зависимость обобщенных координат от времени возьмем аналогичную (4.1). В таком случае движение манипулятора FANUC (кинематическая схема – Рис. 2.5) представлено на Рис. 3.2.

Код программы, решающий прямую позиционную задачу для наземного манипулятора FANUC, представлен в Приложении 2.

### Обратная позиционная задача

Решение обратной позиционной задачи произведем одновременно как для наземного, так и для космического робота, так как их инструменты при

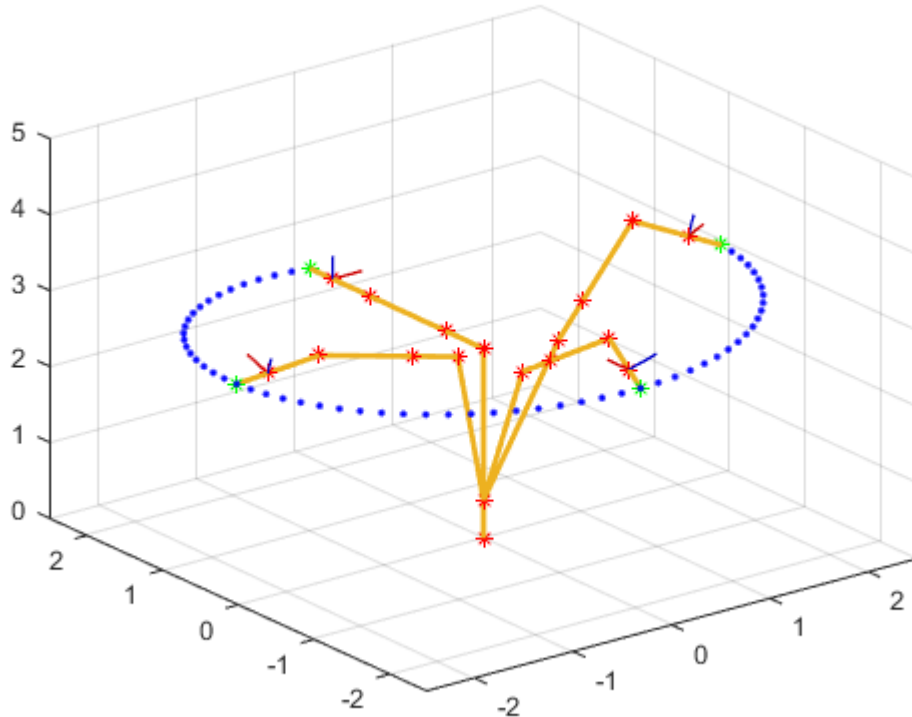


Рисунок 3.2 — Решение прямой позиционной задачи для наземного робота

билатеральном управлении должны совпадать. Космический робот будет виртуальным, поэтому он будет изображен пунктирной линией. В качестве входных данных возьмем два положения инструмента, в которые должны прийти роботы из исходного их состояния:

$$\mathbf{x}_{d1} = \left[ 1.5 \quad 2 \quad 3 \quad \pi/3 \quad 0 \quad \pi/12 \right]^T,$$

$$\mathbf{x}_{d2} = \left[ -1 \quad -1 \quad 4 \quad \pi/3 \quad 0 \quad -\pi/12 \right]^T,$$

здесь первые три компоненты вектора отвечают за координаты характеристической точки инструмента в абсолютном пространстве, а последние три за ориентацию инструмента.

Движение роботов, как и в случае прямой позиционной задачи, также будет равномерным. Рисунок 3.3 иллюстрирует положения роботов в начальный момент времени, а также в заданных положениях инструмента. Рисунок 3.4 отвечает за отображение решения обратной позиционной задачи в проекции на плоскость  $Oxy$  абсолютного пространства.

Код программы, решающий обратную позиционную задачу для наземного манипулятора FANUC и свободнолетающего космического робота, представлен в Приложении 3.

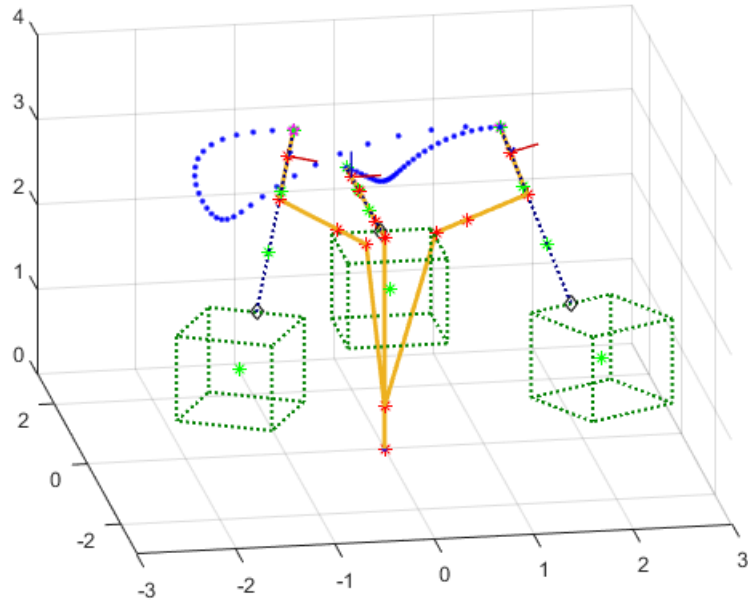


Рисунок 3.3 — Решение обратной позиционной задачи для наземного и космического робота

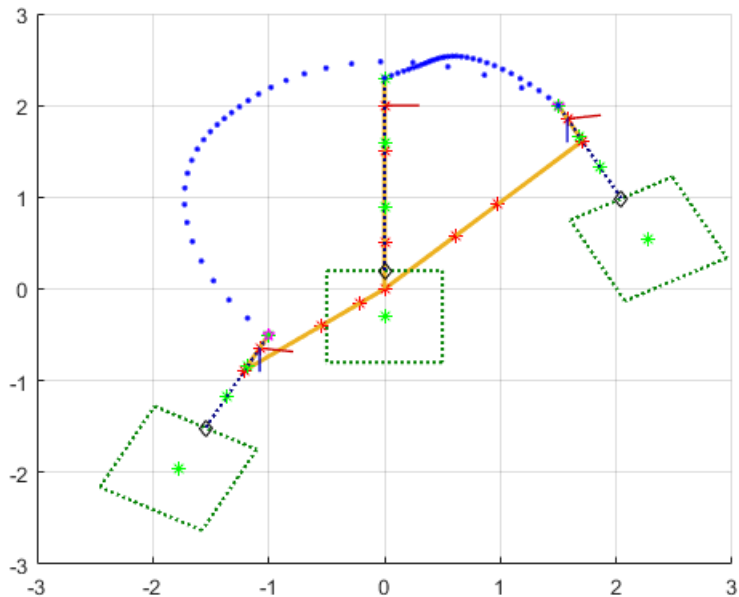


Рисунок 3.4 — Решение обратной позиционной задачи для наземного и космического робота (в проекции на плоскость  $Oxy$ )

### 3.2 Динамическое моделирование

Представим решение прямой задачи динамики, описание которого проводилось во второй главе. Задача заключается в поиске обобщенных сил, которые обеспечивают заданный закон изменения обобщенных координат. Решением данной задачи будет уравнение (2.8). Особую сложность вызывает подсчет матриц  $A(\mathbf{q})$  и  $D_s(\mathbf{q})$ , которые содержат в себе матрицы перехода от вектора квазискоростей к обобщенным скоростям и матрицы масс-инерционных характеристик робота. Алгоритм нахождения матриц  $A(\mathbf{q})$  и  $D_s(\mathbf{q})$  представлен на Рис. 3.10. Приведем решение задачи для свободнолетающего космического робота. Зададим следующий закон изменения обобщенных координат:

$$\mathbf{q}_0 = \mathbf{0},$$

$$\mathbf{q}_k = \left[ 0 \ 0 \ \pi/2 \ 6 \ -6 \ 4 \ \pi/4 \ \pi/3 \ -\pi/2 \right]^T,$$

$$\dot{q}_i(t) = \dot{\lambda}(t)(q_i^1 - q_i^0), \quad i = 1, \dots, 9,$$

где

$$\int_{t^0}^{t^1} \dot{\lambda}(t) dt = 1,$$

$$\dot{\lambda}(t) = \frac{2}{T} \sin^2 \frac{\pi t}{T}, \quad T = t_1 - t_0.$$

При таком законе изменения обобщенных координат космический робот будет совершать следующее движение (Рис. 3.5). Соответствующие данному движению обобщенные силы представлены на Рис. 3.6 и Рис. 3.7.

Код программы, решающий прямую задачу динамики для свободнолетающего космического робота, представлен в Приложении 4.

Приведем решение задачи для наземного робота FANUC. Смоделируем следующий закон изменения обобщенных координат.

$$\mathbf{q}_0 = \mathbf{0},$$

$$\mathbf{q}_k = \left[ 3\pi/2 \ -\pi/8 \ \pi/3 \ \pi/6 \ -\pi/2 \ \pi/4 \right]^T$$

$$\dot{q}_i(t) = \dot{\lambda}(t)(q_i^1 - q_i^0), \quad i = 1, \dots, 9,$$

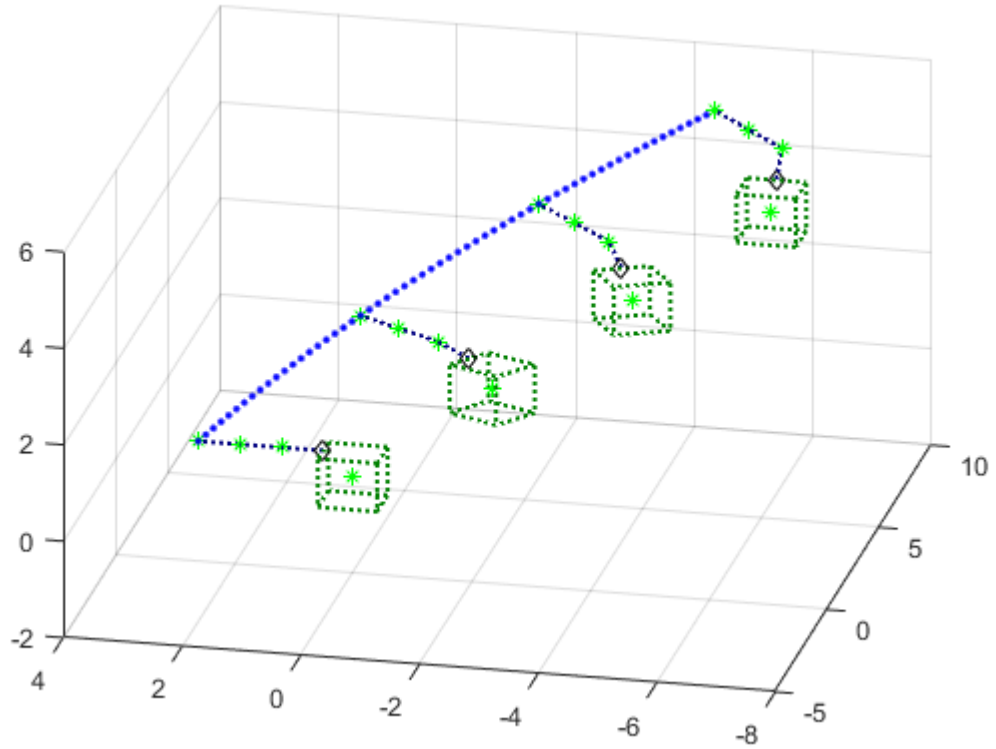


Рисунок 3.5 — Движение космического робота при заданном законе изменения обобщенных координат

где

$$\int_{t^0}^{t^1} \dot{\lambda}(t) dt = 1,$$

$$\dot{\lambda}(t) = \frac{2}{T} \sin^2 \frac{\pi t}{T}, T = t_1 - t_0.$$

Движение робота будет совершать аналогичное движению на Рис. 3.2, но более «плавное» с нулевыми начальными и конечными скоростями. Обобщенные силы для данного случая изображены на Рис. 3.8. и 3.9.

Код программы, решающий прямую задачу динамики для наземного робота, представлен в Приложении 5.



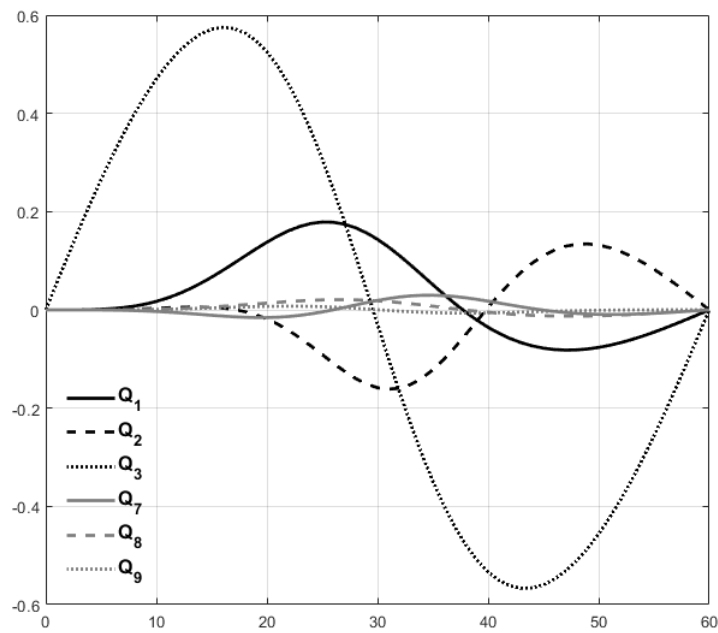


Рисунок 3.6 — Закон изменения обобщенных сил  $Q_1, i = 1, 2, 3, 7, 8, 9$ , для космического робота

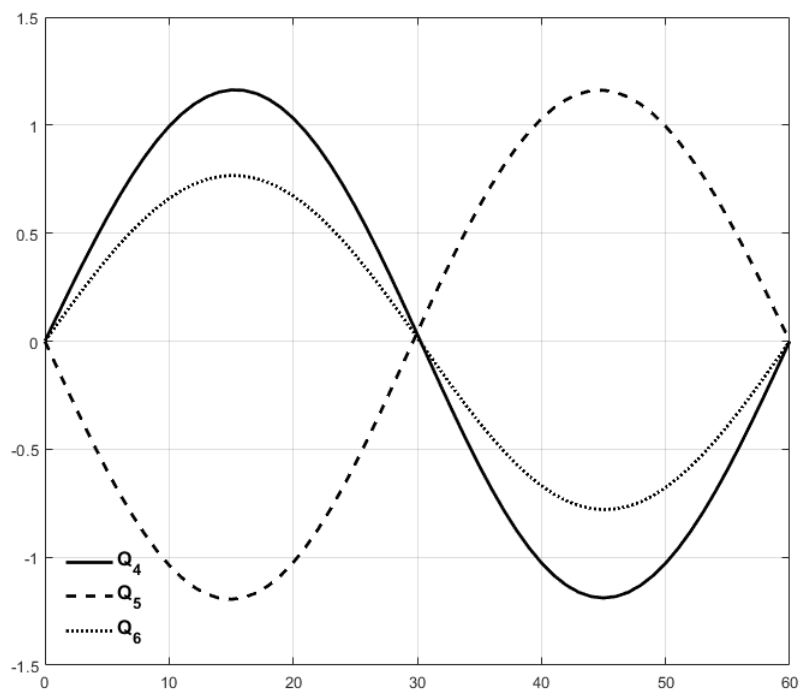


Рисунок 3.7 — Закон изменения обобщенных сил  $Q_1, i = 4, 5, 6$ , для космического робота

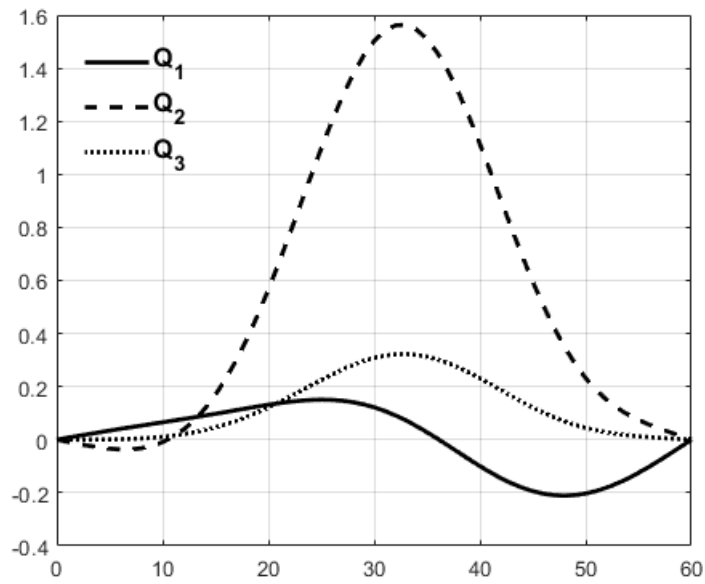


Рисунок 3.8 — Обобщенные силы для 1-го, 2-го и 3-го звена манипулятора

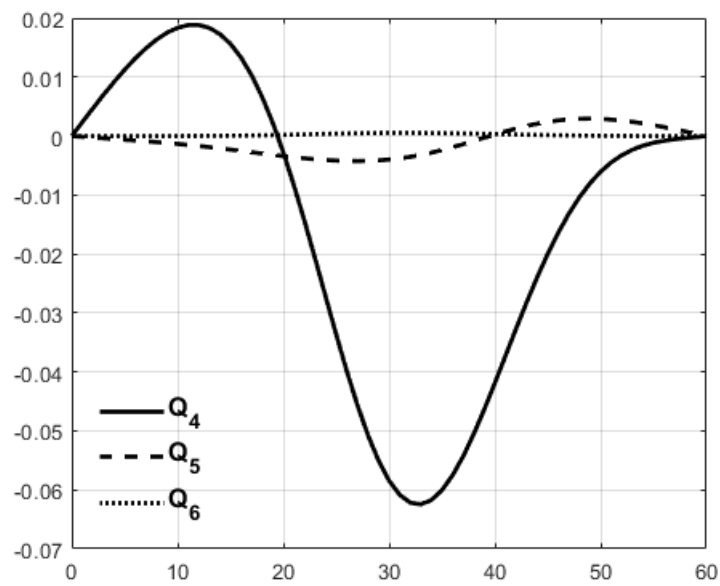
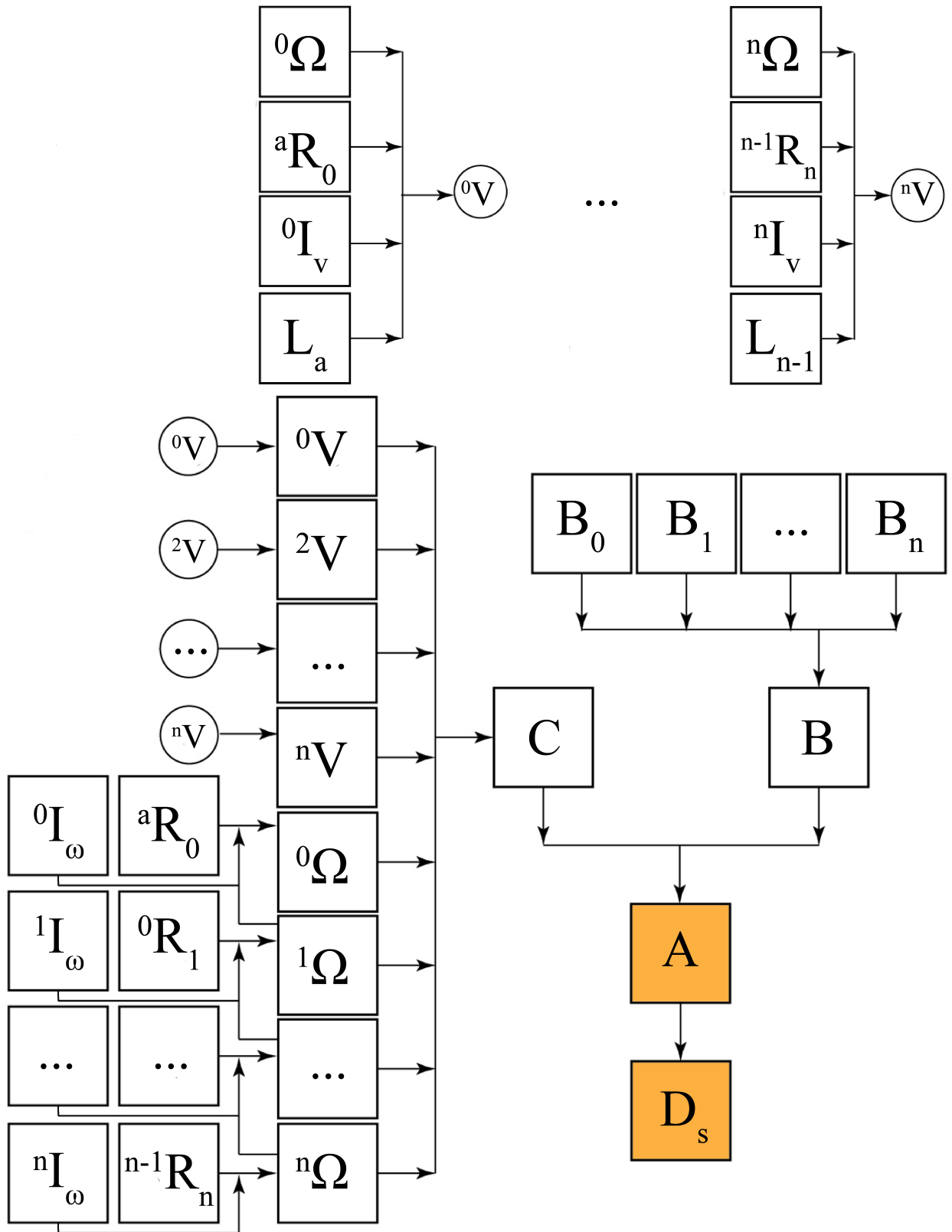


Рисунок 3.9 — Обобщенные силы для 4-го, 5-го и 6-го звена манипулятора

Рисунок 3.10 — Алгоритм вычисления матриц  $A(q)$  и  $D_s(q)$

## Заключение

Результатом выполненного исследования является построение кинематической и динамической модели космического и наземного роботов, которые являются фундаментом для будущих исследований системы билатерального управления методом показа будущих действий. В дальнейшем планируется оптимизировать алгоритм решения прямой задачи динамики, выявить из уравнений динамики вектор управления, провести решение обратной задачи динамики при заданном векторе управления, проверить работоспособность закона управления для адаптации робота к изменениям окружающей среды, а также исследовать динамику поведения космического и наземного робота при взаимодействии инструмента робота с объектами рабочей среды.

В ходе данной научно–исследовательской работы были опубликованы следующие статьи [30–43].

Некоторые результаты данной работы были представлены на следующих конференциях:

- Международная научная конференция по механике «VII Поляховские чтения»
- Международная научная конференция по механике «VIII Поляховские чтения»
- III Международная конференция «Устойчивость и процессы управления»
- 44–я Международная научная конференция «Процессы управления и устойчивость»
- 45–я Международная научная конференция «Процессы управления и устойчивость»
- 47–я Международная научная конференция «Процессы управления и устойчивость»
- 49–я Международная научная конференция «Процессы управления и устойчивость»

Исследование проводилось в рамках проекта РФФИ № 18-08-00419 «Билатеральное дистанционное управление манипуляционными роботами с использованием обучения показом будущих действий».

Другие работы с участием автора [44–51].

## Список литературы

1. *Goertz, R. C.* FUNDAMENTALS OF GENERAL-PURPOSE REMOTE MANIPULATORS / R. C. Goertz // *Nucleonics (U.S.) Ceased publication.* — 1952. — Т. 10, № 11.
2. *Игнатъев, М. Б.* Алгоритмы управления роботами–манипуляторами / М. Б. Игнатъев, Ф. М. Кулаков, А. М. Покровский. — Л.: Машиностроение, 1972. — 248 с.
3. *Clark, J. W.* The Mobot: A Fully Remote Mobile Handling Device / J. W. Clark ; Nuclear Electronics Lab., Hughes Aircraft Co., Culver City, Calif., Tech. Memo. — 1959.
4. *Mosher, R. S.* Operator/Machine Relationships in Manipulators / R. S. Mosher ; Ordnance Dept., General Electric Co., — 1962.
5. *Sheridan, T. B.* Remote Manipulative Control with Transmission Delay / T. B. Sheridan, W. R. Ferrell // *IEEE Transactions on Human Factors in Electronics.* — 1963. — Сент. — Т. HFE—4, № 1. — С. 25—29.
6. *Ferrell, W. R.* Delayed Force Feedback / W. R. Ferrell // *Human Factors: The Journal of the Human Factors and Ergonomics Society.* — 1966. — Окт. — Т. 8, № 5. — С. 449—455.
7. *Bejczy, A. K.* Toward advanced teleoperation in space / A. K. Bejczy // *Progress in Astronautics and Aeronautics.* — 1994. — Т. 161. — С. 107—107.
8. Sensor-based space robotics-ROTEX and its telerobotic features / G. Hirzinger [и др.] // *IEEE Transactions on Robotics and Automation.* — 1993. — Т. 9, № 5. — С. 649—663.
9. A Brief History of DLR's Space Telerobotics and Force-Feedback Teleoperation // *Acta Polytechnica Hungarica.* — 2016. — Янв. — Т. 13, № 1.
10. *Niemeyer, G.* Telerobotics / G. Niemeyer, C. Preusche, G. Hirzinger // *Springer Handbook of Robotics* / под ред. В. Siciliano, О. Khatib. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2008. — С. 741—757. — URL: [https://doi.org/10.1007/978-3-540-30301-5\\_32](https://doi.org/10.1007/978-3-540-30301-5_32).

11. Bilateral Control Systems // Motion Control Systems. — John Wiley & Sons (Asia) Pte Ltd, 03.2011. — С. 233—275.
12. *Anderson, R. J.* Bilateral control of teleoperators with time delay / R. J. Anderson, M. W. Spong // IEEE Transactions on Automatic Control. — 1989. — Май. — Т. 34, № 5. — С. 494—501.
13. *Niemeyer, G.* Stable adaptive teleoperation / G. Niemeyer, J.-J. E. Slotine // IEEE Journal of Oceanic Engineering. — 1991. — ЯНВ. — Т. 16, № 1. — С. 152—162.
14. *Niemeyer, G.* Telemanipulation with Time Delays / G. Niemeyer, J.-J. E. Slotine // The International Journal of Robotics Research. — 2004. — СЕНТ. — Т. 23, № 9. — С. 873—890.
15. *Lee, D.* Passive Bilateral Teleoperation With Constant Time Delay / D. Lee, M. W. Spong // IEEE Transactions on Robotics. — 2006. — АПР. — Т. 22, № 2. — С. 269—281.
16. *Ching, H.* Internet-Based Bilateral Teleoperation Based on Wave Variable With Semi-Adaptive Predictor and Direct Drift Control / H. Ching, W. J. Book // Dynamic Systems and Control, Parts A and B. — ASME, 2005.
17. *Rivals, I.* Nonlinear internal model control using neural networks: application to processes with delay and design issues / I. Rivals, L. Personnaz // IEEE transactions on neural networks. — 2000. — Т. 11, № 1. — С. 80—90.
18. *Smith, O. J. M.* Closer Control of Loops with Dead Time, Chem. Engng. Progr / O. J. M. Smith // 53, S. 217. — 1957. — Т. 219.
19. *Huang, J.-Q.* Neural-network predictive control for nonlinear dynamic systems with time-delay / J.-Q. Huang, F. L. Lewis // IEEE Transactions on Neural Networks. — 2003. — МАРТ. — Т. 14, № 2. — С. 377—389.
20. *Fite, K. B.* Transparent telemanipulation in the presence of time delay / K. B. Fite, M. Goldfarb, A. Rubio // Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003). — IEEE.
21. *Smith, A. C.* Smith Predictor Type Control Architectures for Time Delayed Teleoperation / A. C. Smith, K. Hashtrudi-Zaad // The International Journal of Robotics Research. — 2006. — АВГ. — Т. 25, № 8. — С. 797—818.

22. *Yoshida, K.* Stability and tracking properties in predictive control with adaptation for bilateral teleoperation / K. Yoshida, T. Namerikawa // 2009 American Control Conference. — IEEE, 2009.
23. *Park, J. H.* Sliding mode control of bilateral teleoperation systems with force-reflection on the Internet / J. H. Park, H. C. Cho // Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113). — IEEE.
24. *Garcia-Valdovinos, L.* Observer-based Higher-Order Sliding Mode Impedance Control of Bilateral Teleoperation under Constant Unknown Time Delay / L. Garcia-Valdovinos, V. Parra-Vega, M. Arteaga // 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. — IEEE, 10.2006.
25. *Salcudean, S. E.* Stability guaranteed teleoperation: an adaptive motion/force control approach / S. E. Salcudean, W.-H. Zhu // IEEE Transactions on Automatic Control. — 2000. — Т. 45, № 11. — С. 1951—1969.
26. *Sano, A.* Gain-scheduled compensation for time delay of bilateral teleoperation systems / A. Sano, H. Fujimoto, M. Tanaka // Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146). — IEEE.
27. *Xi, N.* Stability analysis of non-time referenced Internet-based telerobotic systems / N. Xi, T. J. Tarn // Robotics and Autonomous Systems. — 2000. — Август. — Т. 32, № 2/3. — С. 173—178.
28. *Murray, R. M.* A Mathematical Introduction to Robotic Manipulation / R. M. Murray, Z. Li, S. S. Sastry. — CRC Press, 12.1994.
29. *Кулаков, Ф. М.* Супервизорное управление манипуляционными роботами / Ф. М. Кулаков ; под ред. М. Наука. — 1980.
30. Remote Control of Space Robots Change-Adaptive in its External Environment / P. A. Efimova [и др.] // International Journal of Online and Biomedical Engineering. — 2019. — Т. 15, № 7. — С. 84—98.
31. *Горовенко, П. А.* Системы билатерального управления роботами / П. А. Горовенко // Процессы управления и устойчивость. 5 (21). — 2018. — С. 117—121.

32. Дистанционное управление космическими роботами с адаптацией к изменениям его внешней среды. / П. А. Ефимова [и др.] // Вестник Пермского Университета. Серия: Математика. Механика. Информатика. — 2018. — 4(43). — С. 16—26.
33. Dynamic Analysis of Space Robot Remote Control System / P. Gorovenko [и др.] // 8th Polyakhov's Reading. Т. 1959 / под ред. Е. Kustova [и др.]. — American Institute of Physics, 2018. — (AIP Conference Proceedings).
34. Управление многозвенными манипуляционными роботами при наличии связей у перемещаемых ими объектов / П. А. Ефимова [и др.] // . — Российская Федерация : Издательский Дом Федоровой Г.В., 2015. — С. 121—122.
35. *Ефимова, П. А.* Методы дистанционного управления космическими роботами / П. А. Ефимова, Г. В. Алферов, Ф. М. Кулаков // . — 2015. — С. 210. — URL: <http://pol2015.math.spbu.ru/>.
36. *Ефимова, П. А.* Кинематическая модель космического манипуляционного робота / П. А. Ефимова // Молодой ученый. — 2015. — № 6. — С. 1—7.
37. *Ефимова, П. А.* Динамическая модель космического манипуляционного робота / П. А. Ефимова // Процессы управления и устойчивость: Труды 46-й научной конференции факультета ПМ-ПУ СПбГУ. — 2015. — С. 173—179.
38. Modeling and control of robot manipulators with the constraints at the moving objects / P. Efimova [и др.] // Stability and Control Processes in Memory of V.I. Zubov (SCP), 2015 International Conference. — 2015.
39. *Efimova, P.* Methods of remote control over space robots / P. Efimova, F. Kulakov, G. Alferov // 2015 International Conference on Mechanics - Seventh Polyakhov's Reading (Saint Petersburg) / под ред. А. А. Tikhonov. — IEEE Canada, 2015. — С. 113—118.
40. *Efimova, P. A.* Dynamic model of the space manipulator robot / P. A. Efimova // . — 2015. — С. 32. — URL: <http://www.apmath.spbu.ru/ru/research/conference/pm/archive/2015.html> ; The XLVI annual international conference on Control Processes and Stability (CPS'15) ;
41. *Efimova, P.* Dynamic Model of Space Robot Manipulator / P. Efimova, D. Shymanchuk // Applied Mathematical Sciences. — 2015. — Т. 9, № 94. — С. 4653—4659.



42. *Ефимова, П. А.* Моделирование движения космического манипуляционного робота / П. А. Ефимова, Д. В. Шиманчук // Проблемы механики и управления: нелинейные динамические системы. — 2014. — № 46. — С. 20—30.
43. *Ефимова, П. А.* Об особенностях управления космическими роботами-манипуляторами / П. А. Ефимова // Процессы управления и устойчивость: Труды 44-й международной научной конференции аспирантов и студентов. — 2013. — С. 211—214.
44. *Иванов, Г. Г.* Условия устойчивости линейных однородных систем с переключениями / Г. Г. Иванов, Г. В. Алферов, П. А. Ефимова // Вестник Пермского Университета. Серия: Математика. Механика. Информатика. — 2016.
45. *Alferov, G. V.* The Structural Study of Limited Invariant Sets of Relay Stabilized Systems / G. V. Alferov, G. G. Ivanov, P. A. Efimova // Mechanical Systems: Research, Applications and Technology. — Nova Science Publishers. — 09.2017.
46. Study on the Structure of Limit Invariant Sets of Stationary Control Systems with Nonlinearity of Hysteresis Type / G. Alferov [и др.] // International Conference of Numerical Analysis and Applied Mathematics, ICNAAM 2016. Т. 1863 / под ред. Т. Simos, С. Tsitouras. — United States : American Institute of Physics, 07.2017. — (AIP Conference Proceedings).
47. Conditions of Asymptotic Stability for Linear Homogeneous Switched Systems / G. Ivanov [и др.] // Proceedings of The International Conference on Numerical Analysis and Applied Mathematics 2016 (ICNAAM-2016). Т. 1863 / под ред. Т. Simos, С. Tsitouras. — Соединенные Штаты Америки : American Institute of Physics, 2017. — (AIP Conference Proceedings).
48. *Ivanov, G.* Integrability of Nonsmooth One-Variable Functions / G. Ivanov, G. Alferov, P. Efimova // 2017 Constructive Nonsmooth Analysis and Related Topics (dedicated To The Memory Of V.F. Demyanov) (CNSA) / под ред. L. Polyakova. — Соединенные Штаты Америки : Institute of Electrical, Electronics Engineers Inc., 2017. — С. 134—137.

49. Stability of Linear Systems with Multitask Right-hand Member / G. V. Alferov [и др.] // Stochastic Methods for Estimation and Problem-Solving in Engineering. — Nova Science Publishers. — United States : IGI Global, 03.2018. — С. 74—112.
50. Estimation of Periodic Solutions Number of First-Order Differential Equations / G. Ivanov [и др.] // 8th Polyakhov's Reading. Т. 1959 / под ред. Е. Kustova [и др.]. — Соединенные Штаты Америки : American Institute of Physics, 2018. — (AIP Conference Proceedings).
51. *Алферов, Г. В.* Производные числа функций одной переменной / Г. В. Алферов, Г. Г. Иванов, П. А. Горовенко // Вестник Пермского Университета. Серия: Математика. Механика. Информатика. — 2018. — 3(42). — С. 5—19.

## Приложение 1

```
%Постоянные величины
```

```
T=60;
```

```
R=0.5; ls1=0.7; ls2=0.7; ls3=0.3;
```

```
%Начальные данные
```

```
qs0=[0 0 -pi/2 0 2 1 0 0 0];
```

```
Ps0=[qs0(1,4) -R -R R R -R -R R R 0 0 0 0;...
```

```
qs0(1,5) -R R R -R -R R R -R R ls1 ls2 ls3;...
```

```
qs0(1,6) R R R R -R -R -R -R R 0 0 0];
```

```
qs=[-pi/6 0 pi/4 0 -2 5 2*pi/3 -pi/2 0];
```

```
result=zeros(3,T);
```

```
%Начальное положение
```

```
Psk(:,1)=Ps0(:,1);
```

```
A=(alpha1(qs0(1,1))*alpha2(qs0(1,2))*alpha3(qs0(1,3))).';
```

```
for i=2:1:10
```

```
Psk(:,i)=Psk(:,1)+A*Ps0(:,i);
```

```
end
```

```
Psk(:,11)=Psk(:,10)+A*alpha1(qs0(1,7))*Ps0(:,11);
```

```
Psk(:,12)=Psk(:,11)+A*alpha1(qs0(1,7))*alpha3(qs0(1,8))*Ps0(:,12);
```

```
Psk(:,13)=Psk(:,12)+A*alpha1(qs0(1,7))*alpha3(qs0(1,8))*...
```

```
alpha2(qs0(1,9))*Ps0(:,12);
```

```
syms tt
```

```
fun=int(1/T,0,tt);
```

```
tt=0:1:T;
```

```
lambda=zeros(1,T+1);
```

```
lambda=eval(fun);
```

```
for t=1:1:(T+1)
```

```
qst(1,:)=qs0(1,:)+lambda(1,t)*(qs(1,:)-qs0(1,:));
```

```
Psk(:,1)=qst(1,4:6).';
```

```
A=(alpha1(qst(1,1))*alpha2(qst(1,2))*alpha3(qst(1,3))).';
```

```
for i=2:1:10
```

```
Psk(:,i)=Psk(:,1)+A*Ps0(:,i);
```

```
end
```

```
Psk(:,11)=Psk(:,10)+A*alpha1(qst(1,7))*Ps0(:,11);
```

```

Psk(:,12)=Psk(:,11)+A*alpha1(qst(1,7))*alpha3(qst(1,8))*Ps0(:,12);
Psk(:,13)=Psk(:,12)+A*alpha1(qst(1,7))*alpha3(qst(1,8))*...
alpha2(qst(1,9))*Ps0(:,12);
result(:,t)=Psk(:,13);
end

for t=1:20:(T+1)

qst(1,:)=qs0(1,:)+lambda(1,t)*(qs(1,:)-qs0(1,:));

Psk(:,1)=qst(1,4:6).';
A=(alpha1(qst(1,1))*alpha2(qst(1,2))*alpha3(qst(1,3))).';
for i=2:1:10
Psk(:,i)=Psk(:,1)+A*Ps0(:,i);
end
Psk(:,11)=Psk(:,10)+A*alpha1(qst(1,7))*Ps0(:,11);
Psk(:,12)=Psk(:,11)+A*alpha1(qst(1,7))*alpha3(qst(1,8))*Ps0(:,12);
Psk(:,13)=Psk(:,12)+A*alpha1(qst(1,7))*alpha3(qst(1,8))*...
alpha2(qst(1,9))*Ps0(:,12);
axis([-3.5 3.5 -3.5 3.5 0 7]);
hold on
grid on
draw_robot
for s=1:1:T
plot3(result(1,s), result(2,s), result(3,s),'b.')
end
pause(0.0001)

end

%%%%%program draw_robot%%%%%

plot3(Psk(1,1),Psk(2,1),Psk(3,1),'g*')
plot3([Psk(1,2) Psk(1,3)], [Psk(2,2) Psk(2,3)],...
[Psk(3,2) Psk(3,3)], 'LineWidth',1.5, 'Color', [0 0.5 0]) %1
plot3([Psk(1,3) Psk(1,4)], [Psk(2,3) Psk(2,4)],...
[Psk(3,3) Psk(3,4)], 'LineWidth',1.5, 'Color', [0 0.5 0])%2
plot3([Psk(1,4) Psk(1,5)], [Psk(2,4) Psk(2,5)],...
[Psk(3,4) Psk(3,5)], 'LineWidth',1.5, 'Color', [0 0.5 0])%3
plot3([Psk(1,5) Psk(1,2)], [Psk(2,5) Psk(2,2)],...
[Psk(3,5) Psk(3,2)], 'LineWidth',1.5, 'Color', [0 0.5 0])%4

```

```

plot3([Psk(1,2) Psk(1,6)], [Psk(2,2) Psk(2,6)], ...
[Psk(3,2) Psk(3,6)], 'LineWidth', 1.5, 'Color', [0 0.5 0])%5
plot3([Psk(1,3) Psk(1,7)], [Psk(2,3) Psk(2,7)], ...
[Psk(3,3) Psk(3,7)], 'LineWidth', 1.5, 'Color', [0 0.5 0])%6
plot3([Psk(1,4) Psk(1,8)], [Psk(2,4) Psk(2,8)], ...
[Psk(3,4) Psk(3,8)], 'LineWidth', 1.5, 'Color', [0 0.5 0])%7
plot3([Psk(1,5) Psk(1,9)], [Psk(2,5) Psk(2,9)], ...
[Psk(3,5) Psk(3,9)], 'LineWidth', 1.5, 'Color', [0 0.5 0])%8
plot3([Psk(1,6) Psk(1,7)], [Psk(2,6) Psk(2,7)], ...
[Psk(3,6) Psk(3,7)], 'LineWidth', 1.5, 'Color', [0 0.5 0])%9
plot3([Psk(1,7) Psk(1,8)], [Psk(2,7) Psk(2,8)], ...
[Psk(3,7) Psk(3,8)], 'LineWidth', 1.5, 'Color', [0 0.5 0])%10
plot3([Psk(1,8) Psk(1,9)], [Psk(2,8) Psk(2,9)], ...
[Psk(3,8) Psk(3,9)], 'LineWidth', 1.5, 'Color', [0 0.5 0])%11
plot3([Psk(1,9) Psk(1,6)], [Psk(2,9) Psk(2,6)], ...
[Psk(3,9) Psk(3,6)], 'LineWidth', 1.5, 'Color', [0 0.5 0])%12
plot3([Psk(1,10) Psk(1,11)], [Psk(2,10) Psk(2,11)], ...
[Psk(3,10) Psk(3,11)], 'LineWidth', 1.5, 'Color', [0 0 0.5])%13
plot3([Psk(1,11) Psk(1,12)], [Psk(2,11) Psk(2,12)], ...
[Psk(3,11) Psk(3,12)], 'LineWidth', 1.5, 'Color', [0 0 0.5])%14
plot3([Psk(1,12) Psk(1,13)], [Psk(2,12) Psk(2,13)], ...
[Psk(3,12) Psk(3,13)], 'LineWidth', 1.5, 'Color', [0 0 0.5])%14

plot3(Psk(1,10), Psk(2,10), Psk(3,10), 'kD')
plot3(Psk(1,11), Psk(2,11), Psk(3,11), 'g*')
plot3(Psk(1,12), Psk(2,12), Psk(3,12), 'g*')
plot3(Psk(1,13), Psk(2,13), Psk(3,13), 'g*')

%%%%%program alpha1%%%%%%%%
function alpha1=alpha1(x)
alpha1=[1 0 0;0 cos(x) -sin(x);0 sin(x) cos(x)];
end

%%%%%program alpha2%%%%%%%%
function alpha2=alpha2(x)
alpha2=[cos(x) 0 -sin(x); 0 1 0;sin(x) 0 cos(x)];
end

%%%%%program alpha3%%%%%%%%
function alpha3=alpha3(x)
alpha3=[cos(x) -sin(x) 0;sin(x) cos(x) 0;0 0 1];
end

```

## Приложение 2

```

T=60; l1=0.5; l2=2; l3=0.5; l4=1; l5=0.5; l6=0.3;

%Начальные данные
q0=zeros(1,6);
q=[3*pi/2 -pi/8 pi/3 pi/6 -pi/2 pi/4];

P0=[0 0 0 0 0 0 0 0.4 0.3 0 0 0.3 0;...
0 0 0 13 14 15 16 0 0 0.3 0 0 0;...
0 11 12 0 0 0 0 0 0 0.3 0 0.3];

Pk(:,1)=P0(:,1);
Pk(:,2)=Pk(:,1)+alpha3(q0(1,1))*P0(:,2);
Pk(:,3)=Pk(:,2)+alpha3(q0(1,1))*alpha1(q0(1,2))*P0(:,3);
Pk(:,4)=Pk(:,3)+alpha3(q0(1,1))*alpha1(q0(1,2))*...
alpha1(-q0(1,2))*alpha1(q0(1,3))*P0(:,4);
Pk(:,5)=Pk(:,4)+alpha3(q0(1,1))*alpha1(q0(1,2))*...
alpha1(-q0(1,2))*alpha1(q0(1,3))*alpha2(q0(1,4))*P0(:,5);
Pk(:,8)=Pk(:,5)+alpha3(q0(1,1))*alpha1(q0(1,2))*...
alpha1(-q0(1,2))*alpha1(q0(1,3))*alpha2(q0(1,4))*P0(:,8);
Pk(:,6)=Pk(:,5)+alpha3(q0(1,1))*alpha1(q0(1,2))*...
alpha1(-q0(1,2))*alpha1(q0(1,3))*alpha2(q0(1,4))*...
alpha1(q0(1,5))*P0(:,6);
Pk(:,7)=Pk(:,6)+alpha3(q0(1,1))*alpha1(q0(1,2))*...
alpha1(-q0(1,2))*alpha1(q0(1,3))*alpha2(q0(1,4))*...
alpha1(q0(1,5))*alpha2(q0(1,6))*P0(:,7);
Pk(:,12)=Pk(:,6)+alpha3(q0(1,1))*alpha1(q0(1,2))*...
alpha1(-q0(1,2))*alpha1(q0(1,3))*alpha2(q0(1,4))*...
alpha1(q0(1,5))*alpha2(q0(1,6))*P0(:,12);
Pk(:,13)=Pk(:,6)+alpha3(q0(1,1))*alpha1(q0(1,2))*...
alpha1(-q0(1,2))*alpha1(q0(1,3))*alpha2(q0(1,4))*...
alpha1(q0(1,5))*alpha2(q0(1,6))*P0(:,13);
Pk7(:,1)=Pk(:,7);

syms tt
fun=int(1/T,0,tt);
tt=0:1:T;
lambda=zeros(1,T+1);
lambda=eval(fun);

```

```

for t=1:1:(T+1)
qt(1,:)=q0(1,:)+lambda(1,t)*(q(1,:)-q0(1,:));
Pk(:,1)=P0(:,1);
Pk(:,2)=Pk(:,1)+alpha3(qt(1,1))*P0(:,2);
Pk(:,3)=Pk(:,2)+alpha3(qt(1,1))*alpha1(qt(1,2))*P0(:,3);
Pk(:,4)=Pk(:,3)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*P0(:,4);
Pk(:,5)=Pk(:,4)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*P0(:,5);
Pk(:,8)=Pk(:,5)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*P0(:,8);
Pk(:,6)=Pk(:,5)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*...
alpha1(qt(1,5))*P0(:,6);
Pk(:,7)=Pk(:,6)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*...
alpha1(qt(1,5))*alpha2(qt(1,6))*P0(:,7);
Pk7(:,t)=Pk(:,7);
end
for t=1:20:(T+1)
% cla
qt(1,:)=q0(1,:)+lambda(1,t)*(q(1,:)-q0(1,:));
Pk(:,1)=P0(:,1);
Pk(:,2)=Pk(:,1)+alpha3(qt(1,1))*P0(:,2);
Pk(:,3)=Pk(:,2)+alpha3(qt(1,1))*alpha1(qt(1,2))*P0(:,3);
Pk(:,4)=Pk(:,3)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*P0(:,4);
Pk(:,5)=Pk(:,4)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*P0(:,5);
Pk(:,8)=Pk(:,5)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*P0(:,8);
Pk(:,6)=Pk(:,5)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*...
alpha1(qt(1,5))*P0(:,6);
Pk(:,7)=Pk(:,6)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*...
alpha1(qt(1,5))*alpha2(qt(1,6))*P0(:,7);
Pk(:,12)=Pk(:,6)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*...
alpha1(qt(1,5))*alpha2(qt(1,6))*P0(:,12);
Pk(:,13)=Pk(:,6)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*...

```

```

alpha1(qt(1,5))*alpha2(qt(1,6))*P0(:,13);
XXX=X+Pk(:,5);
XXX2=XX+Pk(:,5);
%Pk7(:,t)=Pk(:,7);
axis([-2.5 2.5 -2.5 2.5 0 5]);
hold on
grid on
for s=1:1:T
plot3(Pk7(1,s), Pk7(2,s), Pk7(3,s), 'b.')
end
draw_fanuc
pause(0.0001)
end

%%%%%%%%draw_fanuc%%%%%%%%

plot3(Pk(1,1),Pk(2,1),Pk(3,1), 'r*')
plot3([Pk(1,1) Pk(1,2)], [Pk(2,1) Pk(2,2)], ...
[Pk(3,1) Pk(3,2)], 'LineWidth', 2, 'Color', [0.9290 0.6940 0.1250]) %2
plot3(Pk(1,2),Pk(2,2),Pk(3,2), 'r*')
plot3([Pk(1,2) Pk(1,3)], [Pk(2,2) Pk(2,3)], ...
[Pk(3,2) Pk(3,3)], 'LineWidth', 2, 'Color', [0.9290 0.6940 0.1250]) %3
plot3(Pk(1,3),Pk(2,3),Pk(3,3), 'r*')
plot3([Pk(1,3) Pk(1,4)], [Pk(2,3) Pk(2,4)], ...
[Pk(3,3) Pk(3,4)], 'LineWidth', 2, 'Color', [0.9290 0.6940 0.1250])%4
plot3(Pk(1,4),Pk(2,4),Pk(3,4), 'r*')
plot3([Pk(1,4) Pk(1,5)], [Pk(2,4) Pk(2,5)], ...
[Pk(3,4) Pk(3,5)], 'LineWidth', 2, 'Color', [0.9290 0.6940 0.1250])%5
plot3(Pk(1,5),Pk(2,5),Pk(3,5), 'r*')
plot3([Pk(1,5) Pk(1,6)], [Pk(2,5) Pk(2,6)], ...
[Pk(3,5) Pk(3,6)], 'LineWidth', 2, 'Color', [0.9290 0.6940 0.1250])%6
plot3(Pk(1,6),Pk(2,6),Pk(3,6), 'r*')
plot3([Pk(1,6) Pk(1,7)], [Pk(2,6) Pk(2,7)], ...
[Pk(3,6) Pk(3,7)], 'LineWidth', 2, 'Color', [0.9290 0.6940 0.1250])%7
plot3(Pk(1,7),Pk(2,7),Pk(3,7), 'g*')
%plot3([Pk(1,5) Pk(1,8)], [Pk(2,5) Pk(2,8)], ...
[Pk(3,5) Pk(3,8)], 'LineWidth', 2, 'Color', [0.5 0 0]) % ось 05x
plot3([Pk(1,6) Pk(1,12)], [Pk(2,6) Pk(2,12)], ...
[Pk(3,6) Pk(3,12)], 'LineWidth', 1, 'Color', [0.8 0 0])
plot3([Pk(1,6) Pk(1,13)], [Pk(2,6) Pk(2,13)], ...
[Pk(3,6) Pk(3,13)], 'LineWidth', 1, 'Color', [0 0 0.8])

```



## Приложение 3

```

%Постоянные величины
T=30;
%FANUC
l1=0.5; l2=2; l3=0.5; l4=1; l5=0.5; l6=0.3;
%Space
R=0.5; ls1=0.7; ls2=0.7; ls3=0.3;

%Начальные данные
q0=zeros(1,6);
qs0=zeros(1,9);

Pk7=zeros(3,2*T);

q=zeros(1,6);
qs=zeros(1,9);

%Начальные координаты точек робота:
% %1 - основание; 2-7 - конечные точки звеньев
P0=[0 0 0 0 0 0 0 0.4 0.3 0 0 0.3 0;...
0 0 0 13 14 15 16 0 0 0.3 0 0 0;...
0 11 12 0 0 0 0 0 0 0.3 0 0.3];

Ps0=[qs0(1,4) -R -R R R -R -R R R 0 0 0 0;...
qs0(1,5) -R R R -R -R R R -R R ls1 ls2 ls3;...
qs0(1,6) R R R R -R -R -R -R R 0 0 0];

%Заданное положение и ориентация схвата the best Xd=[1 -1 4 pi/4 0 0];
Xd=[1.5 2 3 pi/3 0 pi/12;...
-1 -1 4 pi/3 0 -pi/12];

%Начальное положение робота

k=1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%FANUC
Pk(:,1)=P0(:,1);
Pk(:,2)=Pk(:,1)+alpha3(q(1,1))*P0(:,2);
Pk(:,3)=Pk(:,2)+alpha3(q(1,1))*alpha1(q(1,2))*P0(:,3);

```

```

Pk(:,4)=Pk(:,3)+alpha3(q(1,1))*alpha1(q(1,2))*...
alpha1(-q(1,2))*alpha1(q(1,3))*P0(:,4);
Pk(:,5)=Pk(:,4)+alpha3(q(1,1))*alpha1(q(1,2))*...
alpha1(-q(1,2))*alpha1(q(1,3))*alpha2(q(1,4))*P0(:,5);
Pk(:,8)=Pk(:,5)+alpha3(q(1,1))*alpha1(q(1,2))*...
alpha1(-q(1,2))*alpha1(q(1,3))*alpha2(q(1,4))*P0(:,8);
Pk(:,6)=Pk(:,5)+alpha3(q(1,1))*alpha1(q(1,2))*...
alpha1(-q(1,2))*alpha1(q(1,3))*alpha2(q(1,4))*alpha1(q(1,5))*P0(:,6);
Pk(:,7)=Pk(:,6)+alpha3(q(1,1))*alpha1(q(1,2))*...
alpha1(-q(1,2))*alpha1(q(1,3))*alpha2(q(1,4))*...
alpha1(q(1,5))*alpha2(q(1,6))*P0(:,7);
Pk(:,9)=06.'+Rd*P0(:,9);
Pk(:,10)=06.'+Rd*P0(:,10);
Pk(:,11)=06.'+Rd*P0(:,11);
Pk(:,12)=Pk(:,6)+alpha3(q(1,1))*alpha1(q(1,2))*...
alpha1(-q(1,2))*alpha1(q(1,3))*alpha2(q(1,4))*...
alpha1(q(1,5))*alpha2(q(1,6))*P0(:,12);
Pk(:,13)=Pk(:,6)+alpha3(q(1,1))*alpha1(q(1,2))*...
alpha1(-q(1,2))*alpha1(q(1,3))*alpha2(q(1,4))*...
alpha1(q(1,5))*alpha2(q(1,6))*P0(:,13);

```

```

Pk7(:,1)=Pk(:,7);
axis([-2.5 2.5 -2.5 2.5 0 5]);
draw_fanuc

```

```

% %%%%%%%%%%%Space

```

```

Psk(:,1)=Ps0(:,1);
A=(alpha1(qs(1,1))*alpha2(qs(1,2))*alpha3(qs(1,3))).';
for i=2:1:10
Psk(:,i)=Psk(:,1)+A*Ps0(:,i);
end
Psk(:,11)=Psk(:,10)+A*alpha1(qs(1,7))*Ps0(:,11);
Psk(:,12)=Psk(:,11)+A*alpha1(qs(1,7))*alpha3(qs(1,8))*Ps0(:,12);
Psk(:,13)=Psk(:,12)+A*alpha1(qs(1,7))*alpha3(qs(1,8))*...
alpha2(qs(1,9))*Ps0(:,12);

```

```

qs(1,4:6)=-Psk(:,13)+Pk(:,7);

```

```

Psk(:,1)=qs(1,4:6).';
A=(alpha1(qs(1,1))*alpha2(qs(1,2))*alpha3(qs(1,3))).';

```

```

for i=2:1:10
Psk(:,i)=Psk(:,1)+A*Ps0(:,i);
end
Psk(:,11)=Psk(:,10)+A*alpha1(qs(1,7))*Ps0(:,11);
Psk(:,12)=Psk(:,11)+A*alpha1(qs(1,7))*alpha3(qs(1,8))*Ps0(:,12);
Psk(:,13)=Psk(:,12)+A*alpha1(qs(1,7))*alpha3(qs(1,8))*...
alpha2(qs(1,9))*Ps0(:,12);
draw_robot

for k=1:1:2
%Матрица желаемой ориентации и желаемые точки 05 и 06
Rd=alpha1(Xd(k,4))*alpha2(Xd(k,5))*alpha3(Xd(k,6));
06=Xd(k,1:3)-16*Rd(:,2).';
05=Xd(k,1:3)-(16+15)*Rd(:,2).';
01d=Xd(k,1:3)-(1s1+1s2+1s3+R)*Rd(:,2).';

%Обнуление переменных
q=zeros(1,6);
qs=zeros(1,9);
Pk=zeros(3,13);
qt=zeros(1,6);
qst=zeros(1,9);
Psk=zeros(3,13);
result=zeros(3,T);

%Конечное положение робота

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%FANUC
%Координата q1
q(1,1)=atan2(-05(1,1), 05(1,2));

%Координаты q2 и q3
052=alpha3(-q(1,1))*05.'-[0;0;11];
r=sqrt((052(2,1))^2+(052(3,1))^2);
if (r<=(12+13+14))
q(1,2)=-atan2(052(2,1),052(3,1))+acos((r^2+12^2...
-(13+14)^2)/(2*12*r));
q(1,3)=acos((12^2+(13+14)^2-r^2)/(2*12*(13+14)))...
-pi/2+q(1,2);
end

%Координаты q4-q6

```



```

fun=int(1/T,0,tt);
tt=0:1:T;
lambda=zeros(1,T+1);
lambda=eval(fun);

for t=1:1:(T+1)
qt(1,:)=q0(1,:)+lambda(1,t)*(q(1,:)-q0(1,:));
qst(1,:)=qs0(1,:)+lambda(1,t)*(qs(1,:)-qs0(1,:));
Pk(:,1)=P0(:,1);
Pk(:,2)=Pk(:,1)+alpha3(qt(1,1))*P0(:,2);
Pk(:,3)=Pk(:,2)+alpha3(qt(1,1))*alpha1(qt(1,2))*P0(:,3);
Pk(:,4)=Pk(:,3)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*P0(:,4);
Pk(:,5)=Pk(:,4)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*P0(:,5);
Pk(:,8)=Pk(:,5)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*P0(:,8);
Pk(:,6)=Pk(:,5)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*...
alpha1(qt(1,5))*P0(:,6);
Pk(:,7)=Pk(:,6)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*...
alpha1(qt(1,5))*alpha2(qt(1,6))*P0(:,7);
Pk7(:,(k-1)*T+t)=Pk(:,7);
end

for t=1:30:(T+1)
% cla
qt(1,:)=q0(1,:)+lambda(1,t)*(q(1,:)-q0(1,:));
qst(1,:)=qs0(1,:)+lambda(1,t)*(qs(1,:)-qs0(1,:));

%%%%%%%%%%FANUC
Pk(:,1)=P0(:,1);
Pk(:,2)=Pk(:,1)+alpha3(qt(1,1))*P0(:,2);
Pk(:,3)=Pk(:,2)+alpha3(qt(1,1))*alpha1(qt(1,2))*P0(:,3);
Pk(:,4)=Pk(:,3)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*P0(:,4);
Pk(:,5)=Pk(:,4)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*P0(:,5);
Pk(:,8)=Pk(:,5)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*P0(:,8);
Pk(:,6)=Pk(:,5)+alpha3(qt(1,1))*alpha1(qt(1,2))*...

```

```

alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*...
alpha1(qt(1,5))*P0(:,6);
Pk(:,7)=Pk(:,6)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*...
alpha1(qt(1,5))*alpha2(qt(1,6))*P0(:,7);
Pk(:,9)=06.'+Rd*P0(:,9);
Pk(:,10)=06.'+Rd*P0(:,10);
Pk(:,11)=06.'+Rd*P0(:,11);
Pk(:,12)=Pk(:,6)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*...
alpha1(qt(1,5))*alpha2(qt(1,6))*P0(:,12);
Pk(:,13)=Pk(:,6)+alpha3(qt(1,1))*alpha1(qt(1,2))*...
alpha1(-qt(1,2))*alpha1(qt(1,3))*alpha2(qt(1,4))*...
alpha1(qt(1,5))*alpha2(qt(1,6))*P0(:,13);
XXX=X+Pk(:,5);
XXX2=XX+Pk(:,5);
%Pk7(:,(k-1)*T+t)=Pk(:,7);
axis([-3.5 3.5 -3.5 3.5 0 7]);
hold on
grid on
for s=1:1:2*T
plot3(Pk7(1,s), Pk7(2,s), Pk7(3,s), 'b.')
end
draw_fanuc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Space

Psk(:,1)=qst(1,4:6).';
A=(alpha1(qst(1,1))*alpha2(qst(1,2))*alpha3(qst(1,3))).';
for i=2:1:10
Psk(:,i)=Psk(:,1)+A*Ps0(:,i);
end
Psk(:,11)=Psk(:,10)+A*alpha1(qst(1,7))*Ps0(:,11);
Psk(:,12)=Psk(:,11)+A*alpha1(qst(1,7))*alpha3(qst(1,8))*Ps0(:,12);
Psk(:,13)=Psk(:,12)+A*alpha1(qst(1,7))*alpha3(qst(1,8))*...
alpha2(qst(1,9))*Ps0(:,12);
qst(1,4:6)=qst(1,4:6)-Psk(:,13).'+Pk(:,7).';
Psk(:,1)=qst(1,4:6).';
A=(alpha1(qst(1,1))*alpha2(qst(1,2))*alpha3(qst(1,3))).';
for i=2:1:10
Psk(:,i)=Psk(:,1)+A*Ps0(:,i);

```

```

end
Psk(:,11)=Psk(:,10)+A*alpha1(qst(1,7))*Ps0(:,11);
Psk(:,12)=Psk(:,11)+A*alpha1(qst(1,7))*alpha3(qst(1,8))*Ps0(:,12);
Psk(:,13)=Psk(:,12)+A*alpha1(qst(1,7))*alpha3(qst(1,8))*...
alpha2(qst(1,9))*Ps0(:,12);
draw_robot
% %%%%%%%%%%%
plot3(Xd(k,1),Xd(k,2),Xd(k,3),'mP')
pause(0.0001)
end
q0=q;
qs0=qs;
end

```

## Приложение 4

```

m=100; a=2; m1=3; m2=3; m3=3; l1=1; l2=1; l3=0.5;
T=60;
q0=zeros(1,9);
q=[0 0 pi/2 6 -6 4 pi/4 pi/3 -pi/2];

MassIn0=diag([m*(a^2)/6 m*(a^2)/6 m*(a^2)/6 m m m]);
MassIn1=[(m1*l1^2)/3 0 0 0 -m1*(l1/2) 0;...
0 (m1*l1^2)/3 0 m1*(l1/2) 0 0;...
0 0 0 0 0 0;...
0 m1*(l1/2) 0 m1 0 0;...
-m1*(l1/2) 0 0 0 m1 0;...
0 0 0 0 0 m1];
MassIn2=[(m2*l2^2)/3 0 0 0 -m2*(l2/2) 0;...
0 (m2*l2^2)/3 0 m2*(l2/2) 0 0;...
0 0 0 0 0 0;...
0 m2*(l2/2) 0 m2 0 0;...
-m2*(l2/2) 0 0 0 m2 0;...
0 0 0 0 0 m2];
MassIn3=[(m3*l3^2)/3 0 0 0 -m3*(l3/2) 0;...
0 (m3*l3^2)/3 0 m3*(l3/2) 0 0;...
0 0 0 0 0 0;...
0 m3*(l3/2) 0 m3 0 0;...
-m3*(l3/2) 0 0 0 m3 0;...

```

```

0 0 0 0 0 m3];
B=[MassIn0 zeros(6,18);zeros(6,6) MassIn1 zeros(6,12);zeros(6,12) MassIn2 zeros(6,6);
zeros(6,18) MassIn3 ];

syms q1 q2 q3 q4 q5 q6 q7 q8 q9;
syms qt1 qt2 qt3 qt4 qt5 qt6 qt7 qt8 qt9;
syms qtt1 qtt2 qtt3 qtt4 qtt5 qtt6 qtt7 qtt8 qtt9;
qq=[q1;q2;q3;q4;q5;q6;q7;q8;q9];
qt=[qt1;qt2;qt3;qt4;qt5;qt6;qt7;qt8;qt9];
qtt=[qtt1;qtt2;qtt3;qtt4;qtt5;qtt6;qtt7;qtt8;qtt9];

W0=[0 sin(q1)*sin(q3) cos(q1) 0 0 0 0 0 0;...
0 cos(q1)*sin(q3) -sin(q1) 0 0 0 0 0 0;...
1 cos(q3) 0 0 0 0 0 0 0];
I=zeros(3,9);
I(3,7)=1;
W1=alpha1(q7)*W0+I;
I=zeros(3,9);
I(3,8)=1;
W2=alpha3(q8)*W1+I;
I=zeros(3,9);
I(3,9)=1;
W3=alpha2(q9)*W2+I;
a0=[cos(q2)*cos(q3)-sin(q2)*cos(q1)*sin(q3)...
sin(q3)*cos(q2)+cos(q3)*cos(q1)*sin(q2) sin(q1)*sin(q2);...
-cos(q3)*sin(q2)-sin(q3)*cos(q1)*cos(q2)...
-sin(q2)*sin(q3)+cos(q2)*cos(q1)*cos(q3) sin(q1)*cos(q2);...
sin(q3)*sin(q1) -cos(q3)*sin(q1) cos(q1)];
V0=[zeros(3,3) a0 zeros(3,3)];
V1=alpha1(q7)*V0-alpha1(q7)*[0 -11 0; 11 0 0;0 0 0]*W0;
V2=alpha3(q8)*V1-alpha3(q8)*[0 -12 0; 12 0 0;0 0 0]*W1;
V3=alpha2(q9)*V2-alpha2(q9)*[0 -13 0; 13 0 0;0 0 0]*W2;
C=[W0;V0;W1;V1;W2;V2;W3;V3];
A=C.'*B*C;
D=sym(zeros(9,81));

for s=1:1:9
for i=1:1:9
for j=1:1:9
p=((s-1)*9)+j;
D(i,p)=(1/2)*(diff(A(i,s),qq(j,1),1)+diff(A(j,s),qq(i,1),1)...
-diff(A(i,j),qq(s,1),1));

```



```

end
end
end
Q=(qt.'*D(1:9,1:9)*qt)*[1;0;0;0;0;0;0;0;0];
for i=2:1:9
e_s=zeros(9,1);
e_s(i,1)=1;
Q=Q+(qt.'*D(1:9,((i-1)*9+1):(9*i))*qt)*e_s;
end
Q=A*qtt+Q;

Q1=zeros(1,T+1);
Q2=zeros(1,T+1);
Q3=zeros(1,T+1);
Q4=zeros(1,T+1);
Q5=zeros(1,T+1);
Q6=zeros(1,T+1);
Q7=zeros(1,T+1);
Q8=zeros(1,T+1);
Q9=zeros(1,T+1);

q1=zeros(1,T+1);
q2=zeros(1,T+1);
q3=zeros(1,T+1);
q4=zeros(1,T+1);
q5=zeros(1,T+1);
q6=zeros(1,T+1);
q7=zeros(1,T+1);
q8=zeros(1,T+1);
q9=zeros(1,T+1);

qt1=zeros(1,T+1);
qt2=zeros(1,T+1);
qt3=zeros(1,T+1);
qt4=zeros(1,T+1);
qt5=zeros(1,T+1);
qt6=zeros(1,T+1);
qt7=zeros(1,T+1);
qt8=zeros(1,T+1);
qt9=zeros(1,T+1);

qtt1=zeros(1,T+1);

```

```

qtt2=zeros(1,T+1);
qtt3=zeros(1,T+1);
qtt4=zeros(1,T+1);
qtt5=zeros(1,T+1);
qtt6=zeros(1,T+1);
qtt7=zeros(1,T+1);
qtt8=zeros(1,T+1);
qtt9=zeros(1,T+1);

```

```
t=0:1:T;
```

```

q1(1,1:T+1)=q0(1,1)+(t/T - sin((pi*t)/(T/2))/(2*pi))*(q(1,1)-q0(1,1));
q2(1,1:T+1)=q0(1,2)+(t/T - sin((pi*t)/(T/2))/(2*pi))*(q(1,2)-q0(1,2));
q3(1,1:T+1)=q0(1,3)+(t/T - sin((pi*t)/(T/2))/(2*pi))*(q(1,3)-q0(1,3));
q4(1,1:T+1)=q0(1,4)+(t/T - sin((pi*t)/(T/2))/(2*pi))*(q(1,4)-q0(1,4));
q5(1,1:T+1)=q0(1,5)+(t/T - sin((pi*t)/(T/2))/(2*pi))*(q(1,5)-q0(1,5));
q6(1,1:T+1)=q0(1,6)+(t/T - sin((pi*t)/(T/2))/(2*pi))*(q(1,6)-q0(1,6));
q7(1,1:T+1)=q0(1,7)+(t/T - sin((pi*t)/(T/2))/(2*pi))*(q(1,7)-q0(1,7));
q8(1,1:T+1)=q0(1,8)+(t/T - sin((pi*t)/(T/2))/(2*pi))*(q(1,8)-q0(1,8));
q9(1,1:T+1)=q0(1,9)+(t/T - sin((pi*t)/(T/2))/(2*pi))*(q(1,9)-q0(1,9));

```

```

qt1(1,1:T+1)=((2/T)*(sin((pi*t)/T)).^2)*(q(1,1)-q0(1,1));
qt2(1,1:T+1)=((2/T)*(sin((pi*t)/T)).^2)*(q(1,2)-q0(1,2));
qt3(1,1:T+1)=((2/T)*(sin((pi*t)/T)).^2)*(q(1,3)-q0(1,3));
qt4(1,1:T+1)=((2/T)*(sin((pi*t)/T)).^2)*(q(1,4)-q0(1,4));
qt5(1,1:T+1)=((2/T)*(sin((pi*t)/T)).^2)*(q(1,5)-q0(1,5));
qt6(1,1:T+1)=((2/T)*(sin((pi*t)/T)).^2)*(q(1,6)-q0(1,6));
qt7(1,1:T+1)=((2/T)*(sin((pi*t)/T)).^2)*(q(1,7)-q0(1,7));
qt8(1,1:T+1)=((2/T)*(sin((pi*t)/T)).^2)*(q(1,8)-q0(1,8));
qt9(1,1:T+1)=((2/T)*(sin((pi*t)/T)).^2)*(q(1,9)-q0(1,9));

```

```

qtt1(1,1:T+1)=((pi*cos((pi*t)/T).*sin((pi*t)/T))/900*(q(1,1)-q0(1,1));
qtt2(1,1:T+1)=((pi*cos((pi*t)/T).*sin((pi*t)/T))/900*(q(1,2)-q0(1,2));
qtt3(1,1:T+1)=((pi*cos((pi*t)/T).*sin((pi*t)/T))/900*(q(1,3)-q0(1,3));
qtt4(1,1:T+1)=((pi*cos((pi*t)/T).*sin((pi*t)/T))/900*(q(1,4)-q0(1,4));
qtt5(1,1:T+1)=((pi*cos((pi*t)/T).*sin((pi*t)/T))/900*(q(1,5)-q0(1,5));
qtt6(1,1:T+1)=((pi*cos((pi*t)/T).*sin((pi*t)/T))/900*(q(1,6)-q0(1,6));
qtt7(1,1:T+1)=((pi*cos((pi*t)/T).*sin((pi*t)/T))/900*(q(1,7)-q0(1,7));
qtt8(1,1:T+1)=((pi*cos((pi*t)/T).*sin((pi*t)/T))/900*(q(1,8)-q0(1,8));
qtt9(1,1:T+1)=((pi*cos((pi*t)/T).*sin((pi*t)/T))/900*(q(1,9)-q0(1,9));

```

```

Q1(1,:)=eval(Q(1,1));
Q2(1,:)=eval(Q(2,1));
Q3(1,:)=eval(Q(3,1));
Q4(1,:)=eval(Q(4,1));
Q5(1,:)=eval(Q(5,1));
Q6(1,:)=eval(Q(6,1));
Q7(1,:)=eval(Q(7,1));
Q8(1,:)=eval(Q(8,1));
Q9(1,:)=eval(Q(9,1));

t=0:1:T;
figure
H1=plot(t,Q1,'k','LineWidth',2);
hold on
H2=plot(t,Q2,'k--','LineWidth',2);
H3=plot(t,Q3,'k:','LineWidth',2);
H4=plot(t,Q7,'k','LineWidth',2,'Color',[0.5 0.5 0.5]);
H5=plot(t,Q8,'k--','LineWidth',2,'Color',[0.5 0.5 0.5]);
H6=plot(t,Q9,'k:','LineWidth',2,'Color',[0.5 0.5 0.5]);
legend([H1,H2,H3,H4,H5,H6],{'Q_1','Q_2','Q_3','Q_7','Q_8','Q_9'},...
'Location','southwest','Box','off','FontSize',12,'FontWeight','bold')
grid on

figure
H1=plot(t,Q4,'k','LineWidth',2);
hold on
H2=plot(t,Q5,'k--','LineWidth',2);
H3=plot(t,Q6,'k:','LineWidth',2);
legend([H1,H2,H3],{'Q_4','Q_5','Q_6'},'Location','southwest','Box','off',...
'FontSize',12,'FontWeight','bold')
grid on

clear all
m1=10; m2=20; m3=10; m4=20; m5=5; m6=2; l1=0.5; l2=2; l3=0.5; ...
l4=1; l5=0.5; l6=0.3;
T=60;
q0=zeros(1,6);

```

## Приложение 5

```
q=[3*pi/2 -pi/8 pi/3 pi/6 -pi/2 pi/4];
```

```
MassIn1=[(m1*l1^2)/3 0 0 0 -m1*(l1/2) 0;...
0 (m1*l1^2)/3 0 m1*(l1/2) 0 0;...
0 0 0 0 0 0;...
0 m1*(l1/2) 0 m1 0 0;...
-m1*(l1/2) 0 0 0 m1 0;...
0 0 0 0 0 m1];
```

```
MassIn2=[(m2*l2^2)/3 0 0 0 -m2*(l2/2) 0;...
0 (m2*l2^2)/3 0 m2*(l2/2) 0 0;...
0 0 0 0 0 0;...
0 m2*(l2/2) 0 m2 0 0;...
-m2*(l2/2) 0 0 0 m2 0;...
0 0 0 0 0 m2];
```

```
MassIn3=[(m3*l3^2)/3 0 0 0 0 m3*(l3/2);...
0 0 0 0 0 0;...
0 0 (m3*l3^2)/3 -m3*(l3/2) 0 0;...
0 0 -m3*(l3/2) m3 0 0;...
0 0 0 0 m3 0;...
m3*(l3/2) 0 0 0 0 m3];
```

```
MassIn4=[(m4*l4^2)/3 0 0 0 0 m4*(l4/2);...
0 0 0 0 0 0;...
0 0 (m4*l4^2)/3 -m4*(l4/2) 0 0;...
0 0 -m4*(l4/2) m4 0 0;...
0 0 0 0 m4 0;...
m4*(l4/2) 0 0 0 0 m4];
```

```
MassIn5=[(m5*l5^2)/3 0 0 0 0 m5*(l5/2);...
0 0 0 0 0 0;...
0 0 (m5*l5^2)/3 -m5*(l5/2) 0 0;...
0 0 -m5*(l5/2) m5 0 0;...
0 0 0 0 m5 0;...
m5*(l5/2) 0 0 0 0 m5];
```

```
MassIn6=[(m6*l6^2)/3 0 0 0 0 m6*(l6/2);...
0 0 0 0 0 0;...
0 0 (m6*l6^2)/3 -m6*(l3/2) 0 0;...
0 0 -m6*(l6/2) m6 0 0;...
0 0 0 0 m6 0;...
m6*(l6/2) 0 0 0 0 m6];
```

```
B=[MassIn1 zeros(6,30);zeros(6,6) MassIn2 zeros(6,24);zeros(6,12) MassIn3 zeros(6,18)
zeros(6,18) MassIn4 zeros(6,12);...
zeros(6,24) MassIn5 zeros(6,6);zeros(6,30) MassIn6];
```

```

syms q1 q2 q3 q4 q5 q6;
syms qt1 qt2 qt3 qt4 qt5 qt6;
syms qtt1 qtt2 qtt3 qtt4 qtt5 qtt6;
qq=[q1;q2;q3;q4;q5;q6];
qt=[qt1;qt2;qt3;qt4;qt5;qt6];
qtt=[qtt1;qtt2;qtt3;qtt4;qtt5;qtt6];

W1=[0 0 0 0 0 0;...
0 0 0 0 0 0;...
1 0 0 0 0 0];
Omega_1=zeros(3,6);
Omega_1(3,1)=1;
I_2=zeros(3,6);
I_2(1,2)=1;
Omega_2=alpha1(q2)*Omega_1+I_2;
I_3=zeros(3,6);
I_3(1,3)=1;
Omega_3=alpha1(q3)*Omega_2+I_3;
I_4=zeros(3,6);
I_4(2,4)=1;
Omega_4=alpha2(q4)*Omega_3+I_4;
I_5=zeros(3,6);
I_5(1,5)=1;
Omega_5=alpha1(q5)*Omega_4+I_5;
I_6=zeros(3,6);
I_6(2,6)=1;
Omega_6=alpha2(q6)*Omega_5+I_6;

V1=zeros(3,6);
V2=zeros(3,6);
V3=alpha1(q3)*[0 -12 0; 12 0 0;0 0 0]*Omega_2;
V4=alpha2(q4)*V3+alpha2(q4)*[0 -13 0; 13 0 0;0 0 0]*Omega_3;
V5=alpha1(q5)*V4+alpha1(q5)*[0 -14 0; 14 0 0;0 0 0]*Omega_4;
V6=alpha2(q6)*V5+alpha2(q6)*[0 -15 0; 15 0 0;0 0 0]*Omega_5;
C=[Omega_1;V1;Omega_2;V2;Omega_3;V3;Omega_4;V4;Omega_5;V5;Omega_6;V6];
A=C.'*B*C;
D=sym(zeros(9,81));

for s=1:1:6
for i=1:1:6
for j=1:1:6
p=((s-1)*6)+j;

```

```

D(i,p)=(1/2)*(diff(A(i,s),qq(j,1),1)+diff(A(j,s),qq(i,1),1)...
-diff(A(i,j),qq(s,1),1));
end
end
end
Q=(qt.'*D(1:6,1:6)*qt)*[1;0;0;0;0;0];
for i=2:1:6
e_s=zeros(6,1);
e_s(i,1)=1;
Q=Q+(qt.'*D(1:6,((i-1)*6+1):(6*i))*qt)*e_s;
end
Q=A*qtt+Q;
Q1=zeros(1,T+1);
Q2=zeros(1,T+1);
Q3=zeros(1,T+1);
Q4=zeros(1,T+1);
Q5=zeros(1,T+1);
Q6=zeros(1,T+1);

q1=zeros(1,T+1);
q2=zeros(1,T+1);
q3=zeros(1,T+1);
q4=zeros(1,T+1);
q5=zeros(1,T+1);
q6=zeros(1,T+1);

qt1=zeros(1,T+1);
qt2=zeros(1,T+1);
qt3=zeros(1,T+1);
qt4=zeros(1,T+1);
qt5=zeros(1,T+1);
qt6=zeros(1,T+1);

qtt1=zeros(1,T+1);
qtt2=zeros(1,T+1);
qtt3=zeros(1,T+1);
qtt4=zeros(1,T+1);
qtt5=zeros(1,T+1);
qtt6=zeros(1,T+1);

t=0:1:T;
q1(1,1:T+1)=q0(1,1)+(t/T - sin((pi*t)/(T/2)))/(2*pi))*(q(1,1)-q0(1,1));

```

```

q2(1,1:T+1)=q0(1,2)+(t/T - sin((pi*t)/(T/2))/(2*pi))*(q(1,2)-q0(1,2));
q3(1,1:T+1)=q0(1,3)+(t/T - sin((pi*t)/(T/2))/(2*pi))*(q(1,3)-q0(1,3));
q4(1,1:T+1)=q0(1,4)+(t/T - sin((pi*t)/(T/2))/(2*pi))*(q(1,4)-q0(1,4));
q5(1,1:T+1)=q0(1,5)+(t/T - sin((pi*t)/(T/2))/(2*pi))*(q(1,5)-q0(1,5));
q6(1,1:T+1)=q0(1,6)+(t/T - sin((pi*t)/(T/2))/(2*pi))*(q(1,6)-q0(1,6));
qt1(1,1:T+1)=((2/T)*(sin((pi*t)/T)).^2)*(q(1,1)-q0(1,1));
qt2(1,1:T+1)=((2/T)*(sin((pi*t)/T)).^2)*(q(1,2)-q0(1,2));
qt3(1,1:T+1)=((2/T)*(sin((pi*t)/T)).^2)*(q(1,3)-q0(1,3));
qt4(1,1:T+1)=((2/T)*(sin((pi*t)/T)).^2)*(q(1,4)-q0(1,4));
qt5(1,1:T+1)=((2/T)*(sin((pi*t)/T)).^2)*(q(1,5)-q0(1,5));
qt6(1,1:T+1)=((2/T)*(sin((pi*t)/T)).^2)*(q(1,6)-q0(1,6));
qtt1(1,1:T+1)=((pi*cos((pi*t)/T).*sin((pi*t)/T))/900*(q(1,1)-q0(1,1));
qtt2(1,1:T+1)=((pi*cos((pi*t)/T).*sin((pi*t)/T))/900*(q(1,2)-q0(1,2));
qtt3(1,1:T+1)=((pi*cos((pi*t)/T).*sin((pi*t)/T))/900*(q(1,3)-q0(1,3));
qtt4(1,1:T+1)=((pi*cos((pi*t)/T).*sin((pi*t)/T))/900*(q(1,4)-q0(1,4));
qtt5(1,1:T+1)=((pi*cos((pi*t)/T).*sin((pi*t)/T))/900*(q(1,5)-q0(1,5));
qtt6(1,1:T+1)=((pi*cos((pi*t)/T).*sin((pi*t)/T))/900*(q(1,6)-q0(1,6));
Q1(1,:)=eval(Q(1,1));
Q2(1,:)=eval(Q(2,1));
Q3(1,:)=eval(Q(3,1));
Q4(1,:)=eval(Q(4,1));
Q5(1,:)=eval(Q(5,1));
Q6(1,:)=eval(Q(6,1));
t=0:1:T;
figure
H1=plot(t,Q1,'k','LineWidth',2);
hold on
H2=plot(t,Q2,'k--','LineWidth',2);
H3=plot(t,Q3,'k:','LineWidth',2);
legend([H1,H2,H3],{'Q_1','Q_2','Q_3'},'Location','northwest',...
'Box','off','FontSize',12,'FontWeight','bold')
grid on
figure
H1=plot(t,Q4,'k','LineWidth',2);
hold on
H2=plot(t,Q5,'k--','LineWidth',2);
H3=plot(t,Q6,'k:','LineWidth',2);
legend([H1,H2,H3],{'Q_4','Q_5','Q_6'},'Location','southwest',...
'Box','off','FontSize',12,'FontWeight','bold')
grid on

```