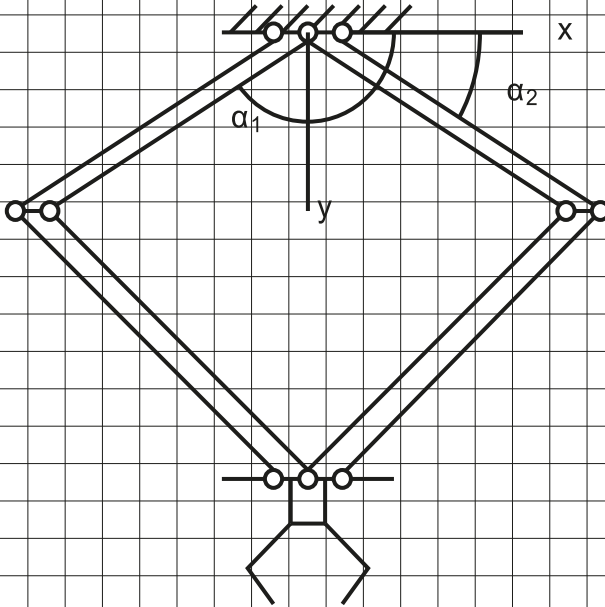


О. М. Киселев



МАТЕМАТИЧЕСКИЕ ОСНОВЫ РОБОТОТЕХНИКИ

УДК 519.7
ББК 84(2Рос=Рус)6
К 44

О.М. Киселёв

К 44 Математические основы робототехники – Орёл: Издательство
«Картуш», 2019. – 228 с.

ISBN 978-5-9708-0768-2

УДК 519.7
ББК 84(2Рос=Рус)6

Книга о приложениях математики в робототехнике. В тексте последовательно вводятся математические понятия, необходимые для формулировки математических моделей робототехнических систем с точки зрения геометрии, динамики, теории автоматического управления. Обсуждаются вопросы обработки изображений и использования искусственных нейронных сетей в роботизированных системах.

Для старших школьников, студентов младших курсов и преподавателей робототехники, информатики, прикладной математики.

ISBN 978-5-9708-0768-2

© О.М. Киселёв, 2019

Оглавление

0.1	Для кого	9
0.2	Для чего	9
0.3	О чем	9
0.4	Структура текста	10
0.5	Благодарности	10
1	Геометрия и роботы	13
1.1	Геометрия и мобильные роботы	14
1.1.1	Конфигурационное пространство мобильного робота	14
1.1.2	Длина траектории движения робота и субримановы многообразия	16
1.1.3	Уравнение движения	17
1.1.4	Преобразование координат, отображение пространства в себя при движении робота	20
1.1.5	Дифференциальное уравнение движения робота . . .	20
1.1.6	Упражнения	23
1.2	Геометрия голономного робота со всенаправленными колесами	24
1.2.1	Кинематика всенаправленного колеса	24
1.2.2	Схема робота	25
1.2.3	Преобразования координат при повороте и переме- щении. Прямая задача кинематики движения голо- номного робота	25
1.2.4	Обратная задача кинематики голономного робота . . .	30
1.2.5	Упражнения	32
1.3	Обзор литературы.	33
2	Манипуляторы. Тригонометрия и алгебра.	35
2.1	Конфигурационное пространство робота-манипулятора	35
2.1.1	Однозвенный механизм	36
2.1.2	Конфигурационное пространство двухзвенника	37

2.1.3	Плоский двухзвенник	38
2.1.4	Двухзвенник с ортогональными осями шарниров	39
2.1.5	Плоский n -звенник с избыточной степенью свободы .	42
2.1.6	Манипулятор с тремя звеньями.	43
2.1.7	Схват манипулятора и многообразия в конфигураци- онном пространстве.	45
2.1.8	Упражнения	46
2.2	Прямая и обратная задачи позиционирования схвата	47
2.2.1	Обратная задача для двухзвенника	47
2.2.2	Обратная задача для плоского трехзвенника	48
2.2.3	Обратная задача для трехзвенника в трехмерном пространстве	50
2.2.4	Обратная задача для неплоского n -звенника при $n > 3$	52
2.2.5	Оптимальное решение обратной задачи	52
2.2.6	Упражнения	53
2.3	Матричное представление положения манипулятора	55
2.3.1	Матрицы поворота	55
2.3.2	Умножение матриц	56
2.3.3	Манипулятор с n -звеньями в одной плоскости.	59
2.3.4	Упражнения	59
2.4	Манипулятор дельта-робота.	60
2.4.1	Конфигурация плоского манипулятора с параллель- ным механизмом.	60
2.4.2	Прямая и обратная задачи кинематики для плоского замкнутого четырехзвенника	62
2.4.3	Параллельный механизм в составе манипулятора дельта-робота.	63
2.4.4	Конфигурационное пространство дельта-робота	63
2.4.5	Прямая и обратная задачи для дельта-робота	65
2.4.6	Упражнения	67
2.5	Обзор литературы	68
3	Сигналы. Аппроксимация и статистика	69
3.1	Общая структура потока данных	70
3.1.1	Свойства случайной величины	71
3.1.2	Равномерное распределение случайной величины . . .	73
3.1.3	Нормальное распределение	73
3.1.4	Кластеризация данных	75
3.1.5	Упражнения	76

3.2	Виды фильтров данных.	77
3.2.1	Заранее определенный интервал	77
3.2.2	Квазистационарные данные. Повторный опрос	77
3.2.3	Скользящее среднее	77
3.2.4	Доверительный интервал	78
3.2.5	Упражнения	78
3.3	Калибровка и устранение систематической ошибки	79
3.3.1	Сдвиг	79
3.3.2	Интерполяция	80
3.3.3	Упражнения	84
3.4	Обзор литературы	85
4	Изображения, преобразования, свертки	87
4.1	Геометрия цвета	88
4.1.1	Цветовое пространство RGB	88
4.1.2	Цветовое пространство CMYK	89
4.1.3	Цветовое пространство YUV	91
4.1.4	Норма в цветовом пространстве	92
4.1.5	Перцептивные цветовые пространства HSV	93
4.1.6	Топология пространства HSV	94
4.1.7	Геометрическое представление	96
4.1.8	Упражнения	96
4.2	Поиск информации на изображении	97
4.2.1	Растровые изображения	97
4.2.2	Линейные фильтры	98
4.2.3	Нелинейные фильтры	102
4.2.4	Морфологические преобразования	104
4.2.5	Аффинные преобразования	106
4.2.6	Упражнения	108
4.3	Цифровая калибровка оптики	108
4.3.1	Дисторсия. Примеры	108
4.3.2	Цифровая калибровка объектива	109
4.3.3	Метод наименьших квадратов	110
4.3.4	МНК для калибровки объектива	113
4.3.5	Упражнения	117
4.4	Обзор литературы	117
5	ТАУ. Отображения и уравнения	119
5.1	Управление, примеры объектов управления	119
5.1.1	Терминология ТАУ	120
5.1.2	Особенность систем управления без обратной связи	121

5.1.3	Упражнения	122
5.2	Пропорциональный регулятор в системах первого порядка	122
5.2.1	Пропорциональный регулятор для стабилизации температуры	123
5.2.2	Стабилизация объекта пропорциональным регулятором	125
5.2.3	Дискретное отображение при управлении через циф- ровой контроллер	128
5.2.4	Упражнения	130
5.3	Регулятор в системах второго порядка	130
5.3.1	Остановка робота на заданном расстоянии от стены .	130
5.3.2	Неустойчивость пропорционального регулятора для остановки робота около стены	133
5.3.3	Пропорционально-дифференциальный регулятор в системах второго порядка	135
5.3.4	Дискретное отображение при управлении с помощью ПД-регулятора	137
5.3.5	Упражнения	142
5.4	ПИД-регулятор	142
5.4.1	Систематические ошибки управления	142
5.4.2	Интегральная составляющая регулятора.	143
5.4.3	Влияние интегральной составляющей в регуляторе . .	144
5.5	Классические регуляторы с обратной связью	147
5.5.1	Поплавковый регулятор	147
5.5.2	Центробежный регулятор	149
5.5.3	Математическая модель осциллятора ван дер Поля . .	153
5.6	Обзор литературы	157
6	Графы, карты, путь	159
6.1	Структура карты	159
6.1.1	Карта дорог как взвешенный граф	159
6.1.2	Специальный случай. Лабиринт	160
6.1.3	Расстояние в городском квартале	163
6.1.4	Полигональные карты, как векторные поля	165
6.1.5	Векторная карта	166
6.1.6	Упражнения	168
6.2	Расстояние на карте и кратчайший путь	168
6.2.1	Длина пути в ячейке карты	168
6.2.2	Длина пути между двумя пунктами на карте	170
6.2.3	Построение кратчайшего пути на плоскости	171

6.2.4	Построение пути по взвешенному графу	173
6.2.5	Планирование маршрута	174
6.2.6	Упражнения	175
6.3	Построение карты	176
6.3.1	Локальная карта	176
6.3.2	Атлас рабочей области	177
6.3.3	Переход в полярную систему	179
6.3.4	Упражнения	180
6.4	Локализация	181
6.4.1	Определение расстояний до препятствий	181
6.4.2	Определение положений по трем найденным ориентирам	181
6.4.3	Преобразование координат в процессе локализации робота	184
6.4.4	Использование лидара для построения локальной карты	185
6.4.5	Упражнения	186
6.5	Обзор литературы	187
7	Интеллектуальные системы и методы минимизации.	189
7.1	Цепь Маркова. Построение моделей. Примеры применения. .	190
7.1.1	Примеры марковских цепей	190
7.1.2	Формализация определения цепи Маркова	191
7.1.3	Построение стохастической матрицы	192
7.1.4	Вероятность перехода между состояниями за N шагов	193
7.1.5	Применение марковских цепей	194
7.2	Дерево решений и минимизация энтропии	194
7.2.1	Дерево решений для игры в «крестики-нолики»	194
7.2.2	Классификация и энтропия	196
7.2.3	Упражнения	200
7.3	Знакомство с нейросетями	200
7.3.1	Почему нейросеть?	200
7.3.2	Математическая модель нейрона	202
7.3.3	Персептрон	204
7.3.4	Многослойная нейросеть	205
7.3.5	Разновидности нейросетей	208
7.4	Обучение нейросетей	209
7.4.1	Методы обучения ИНС	209
7.4.2	Наборы данных для обучения	210
7.5	Методы минимизации	211
7.5.1	Метод скорейшего покоординатного спуска	212

7.5.2	Метод скорейшего градиентного спуска	215
7.5.3	Глобальная минимизация	219
7.6	Обзор литературы	222

0.1 Для кого

Этот текст написан для учеников старших классов школы и, возможно, учителей, занимающихся робототехникой. Отдельные главы могут оказаться полезны для студентов, специализирующихся в области математического моделирования, теории управления и динамических систем.

0.2 Для чего

Очевидная цель – познакомить с подходами, лежащими в основе математического моделирования сложных систем – как механических, так и систем обработки информации и управления.

Скрытая и главная цель – увлечь читателя математикой через ее приложения. Показать простоту и естественность математических определений. Привести читателя к пониманию, что настоящие математические задачи инициируются приложениями, лежат совсем рядом с приложениями и по сей день открыты для исследования.

0.3 О чем

Текст о математике и её приложениях в робототехнике.

На примере задач робототехники можно познакомиться с неевклидовой геометрией и некоммутативной алгеброй, теорией аппроксимации и методами минимизации, основами математической статистики и цепями Маркова, интегральными операторами и дифференциальными уравнениями. Причем важно, что все перечисленные разделы математики оказываются не абстрактными, а именно прикладными, которые уже используются в робототехнических устройствах.

В некотором смысле, математика – языковая надстройка над пространством естественных наук. Учить математике – учить этому языку. Учить языку без обучения словам этого языка не получится.

Для этого в текст вводятся определения и математические понятия, использующиеся скорее студентами-математиками, чем школьниками. Здесь эти определения вводятся намеренно, чтобы показать, что они просты. За математическими определениями скрывается всего лишь попытка абстрагироваться от предметной области так, чтобы эти определения оказались пригодны к использованию в как можно более широком круге задач.

0.4 Структура текста

Главы связаны последовательностью введения математических терминов и общим подходом при знакомстве с робототехникой. Но читать текст можно начинать с любой главы, если в тексте появляется понятие, введенное ранее, можно обратиться к Предметному указателю.

В тексте по мере необходимости вводятся математические термины. При первом их использовании они напечатаны курсивом. На той же странице в выключенном из основного изложения тексте приводится объяснение или строгое математическое определение использованного термина. Математические термины обычно используются не для того, чтобы изложение было наукообразным и непонятным, а для того, чтобы сократить текст и не писать лишних пояснений, дав пояснение термину лишь один раз.

Как правило, разделы сопровождаются упражнениями. Одно или два из предлагаемых упражнений подробно разобраны, остальные приведены для самостоятельного решения.

В окончании каждой главы приведен небольшой список литературы. В такой список, как правило, включены учебники, содержащие подробное описание рассмотренных в главе понятий.

0.5 Благодарности

Этот текст мне предложил написать Александр Колотов из Университета Иннополис как учебное пособие для участников олимпиады НТИ. Выражаю признательность ему за это предложение и дальнейшие организационные усилия.

Выражаю свою благодарность Павлу Козлову за подготовленный им список учебников по искусственным нейронным сетям и участие в обсуждениях на начальном этапе.

Выражаю признательность коллегам, приславшим замечания к раннему варианту текста:

Сергею Глебову,
Артуру Загидуллину,
Михаилу Иванову,
Алексею Клячину,
Олегу Лукашенко,
Валентине Любимовой,
Алексею Миклину,

Станиславу Протасову,
Инсафу Сафину,
Галине Федченко.

Коллеги оказали критически важное влияние на количество примеров, рисунков, упражнений, упрощение математических формулировок.

Олег Киселев,
ok@ufanet.ru
Иннополис-Сочи-Уфа,
ИМВЦ УФИЦ РАН
2018-2019.

Глава 1

Геометрические пространства в робототехнике

Геометрия на плоскости – наиболее наглядный и понятный раздел математики. Задачи робототехники пронизаны геометрическими построениями. Однако геометрия в робототехнике отличается от планиметрии, даже в том простом случае, когда мобильный робот движется по плоскости.

Здесь приведены две кинематических модели мобильных роботов.

- Первая модель – самая простая, с которой начинаются занятия робототехникой. Это тележка с двумя ведущими колесами. Каждое из ведущих колес подключено к своему мотору. Ведущие колеса геометрически расположены на одной оси.
- Вторая модель – робот на всенаправленных колесах. Этот робот имеет три ведущих колеса. Каждое ведущее колесо подключено к своему мотору. Оси колес расположены в одной плоскости под углом 120° друг к другу.

Для обеих моделей подробно рассмотрены

- геометрия конфигурационного пространства;
- уравнения траектории движения.

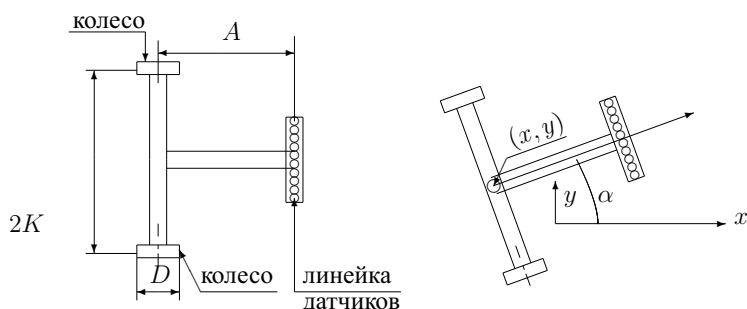


Рис. 1.1: На рисунке слева – схема робота. На схеме введены обозначения для важных геометрических характеристик робота: A – расстояние от ведущих колес до линейки датчиков; $2K$ – колея робота; D – диаметр ведущих колес. На рисунке справа указана точка, по которой определяются координаты робота – центр робота (x, y) и направление робота – угол α . Тройка координат (x, y, α) определяет положение робота на плоскости.

1.1 Геометрия и мобильные роботы

1.1.1 Конфигурационное пространство мобильного робота

Здесь под словами «мобильный робот» будет пониматься тележка с двумя ведущими колесами. Каждое колесо управляется собственным мотором. Такой робот обычно называется мобильным роботом с дифференциальным приводом.

В самой простой, и поэтому важной, ситуации мобильный робот может быть представлен как два ведущих колеса, расположенных геометрически на одной оси вращения. Однако физически оси вращения колес не связаны, так что каждое колесо может вращаться со своей скоростью. Схема робота и его размеры, необходимые в дальнейшем, приведены на рисунке 1.1.

Координатами робота на плоскости будем считать координаты середины оси (x, y) и угол α – направление движения по отношению к оси x . Коор-

динаты (x, y) обычные прямоугольные декартовы координаты, направление робота – дополнительная координата – угол между перпендикуляром к оси колес и положительным направлением оси Ox .

Робот может двигаться только в том направлении, куда указывает α . Для изменения направления движения роботу необходимо проехать некоторое расстояние, то есть робот не может сделать поворот мгновенно. Возможны разные способы поворота. В общем случае это поворот вокруг некоторой точки на плоскости. В частном случае, центр поворота может быть в точке середины оси ведущих колес робота. В этом случае при повороте координаты x, y не меняются, но каждое из колес робота совершает некоторый пробег.

- Устройство, траектория движения которого определяется связями, зависящими от вектора его скорости, называется *неголономным*.

Пример 1 *Свободное колесо, катящееся по плоскости – неголономное устройство. С другой стороны – то же колесо, ось которого длиной R шарнирно закреплена в точке $(0, 0)$, также может катиться по плоскости, но его траектория определяется уравнением:*

$$x^2 + y^2 = R^2.$$

В этом уравнении участвуют только координаты. Поэтому колесо с закреплённой осью неголономным устройством не является.

Можно считать, что робот помещен в некоторое трехмерное пространство, определяющее его конфигурацию – не только координаты центра робота, но его направление.

- *Конфигурационное пространство* для описания кинематики $(x, y, \alpha) \in \{\mathbb{R}^2 \times [-\pi, \pi)\}$. Это же конфигурационное пространство можно рассматривать как прямое произведение координатной плоскости \mathbb{R}^2 на единичную окружность \mathbb{S}^1 . Окружность определяет угол в радианах – направление робота. Кратко это можно записать в виде: $\mathbb{R}^2 \times \mathbb{S}^1$.

Такое трехмерное пространство получается, если к каждой точке плоскости формально приклеить окружность, как третье измерение. В малой окрестности любой точки такое трехмерное пространство похоже на трехмерное пространство \mathbb{R}^3 , но глобально его свойства иные. Например, если отойти на расстояние 2π по третьему измерению, то попадем в ту же точку, откуда начали движение.

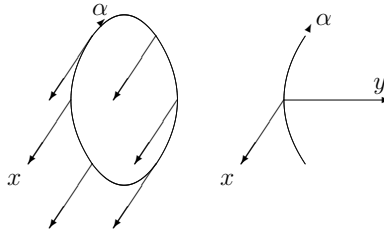


Рис. 1.2: Слева – глобальная структура пространства $\mathbb{R} \times \mathbb{S}^1$, которое получается в результате прямого произведения прямой на окружность. В результате получается обычный круговой цилиндр. Его можно изобразить вложенным в трехмерное евклидово пространство. Справа – локальная структура конфигурационного пространства $\mathbb{R}^2 \times \mathbb{S}^1$. Глобальная структура этого пространства отличается от трехмерного евклидова.

1.1.2 Длина траектории движения робота и субримановы многообразия

При движении робота координаты x, y, α меняются во времени, то есть $x(t), y(t), \alpha(t)$. Траектория движения робота – одномерная кривая, в трехмерном пространстве, заданная в параметрической форме, где параметр t – время.

При измерении длины пути робота обычно рассматривают траекторию не в трехмерном конфигурационном пространстве, а проекцию траектории на двумерную поверхность – плоскость (x, y) , по которой робот движется в физическом пространстве.

Это означает, что одна из координат робота – угол α , который определяет направление робота на поверхности, не участвует в вычислении длины пути, хотя неявно длина пути, конечно, зависит от того, как α меняется при движении.

Понятие длины в декартовой системе координат тесно связано с понятием скалярного произведения векторов. Действительно, пусть за некоторое время dt координаты робота изменились на величины $(dx, dy, d\alpha)$. Длину траектории робота на поверхности можно приближенно считать равной длине вектора $d\vec{l} = \{dx, dy\}$. По определению скалярного произведения это

$$|d\vec{l}|^2 = (dx)^2 + (dy)^2 = (d\vec{l}, d\vec{l}).$$

- Пространство, в котором скалярное произведение определено не для

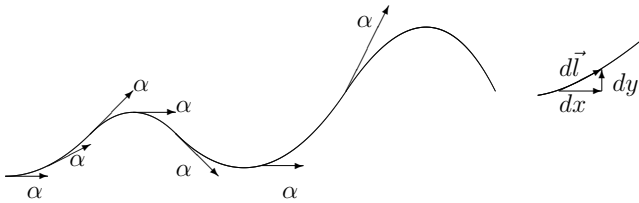


Рис. 1.3: Слева – траектория пути робота на плоскости и направление угла поворота робота α на разных участках траектории. Справа – участок траектории и метод вычисления длины этого участка по теореме Пифагора.

всех измерений, называется *субримановым пространством*.

Субриманова геометрия – популярный раздел современных исследований математиков.

1.1.3 Уравнение движения

Для вывода уравнения траектории движения робота в пространстве $\mathbb{R}^2 \times \mathbb{S}^1$ удобно принять обозначения: ω_L – угловая скорость левого колеса; ω_R – угловая скорость правого колеса.

Примем для определённости $\omega_L < \omega_R$. Тогда линейная скорость левого колеса определяется по формуле:

$$v_L = \omega_L \frac{D}{2},$$

линейная скорость правого колеса:

$$v_R = \omega_R \frac{D}{2}.$$

Если два колеса на одной оси катятся по плоскости с разными скоростями, тогда движение робота происходит по окружности.

Обозначим R – мгновенный радиус кривизны траектории. Скорость поворота робота будет обозначаться как $\dot{\alpha}$ ¹. Под скоростью поворота понима-

¹Обозначение скорости с помощью точки над соответствующей координатой введено И. Ньютоном. Фактически это производная пути по времени. В континентальной Европе примерно в то же время понятие производной функции было введено Лейбницем. Он обозначал производные как отношение $\frac{df}{dx}$. Обозначение производной с помощью штриха x' использовал Лагранж. В механике принято производные обозначать точками над функцией, а в математике – штрихами.

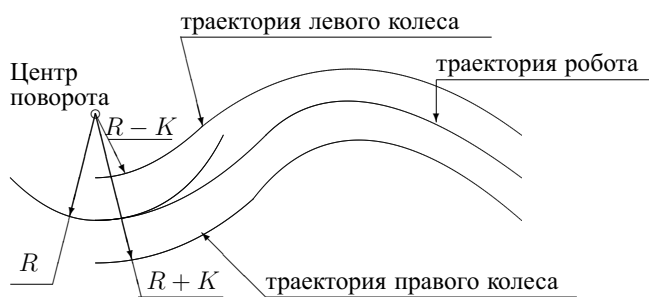


Рис. 1.4: Движение робота вокруг мгновенного центра поворота. Левое колесо поворачивается по окружности радиуса $R - K$. Правое колесо – по окружности радиуса $R + K$.

ется угловая скорость вращения робота вокруг точки середины оси ведущих колес.

Уравнение для скорости поворота оси колес $\dot{\alpha}$:

$$\omega_L \frac{D}{2} = \dot{\alpha}(R - K), \quad \omega_R \frac{D}{2} = \dot{\alpha}(R + K).$$

Тогда скорость поворота робота и радиус кривизны траектории:

$$\dot{\alpha} = \frac{D}{4K}(\omega_R - \omega_L), \quad R = \frac{\omega_R + \omega_L}{\omega_R - \omega_L} K.$$

Пусть (X, Y) – координаты центра поворота, текущие координаты робота (x_0, y_0, α_0) , тогда X, Y и R связаны формулами:

$$x_0 - X = R \sin(\alpha_0), \quad y_0 - Y = -R \cos(\alpha_0).$$

Обозначим координаты робота после поворота на угол $d\alpha$ как (x_1, y_1, α_1) . Тогда изменение координат робота:

$$\begin{aligned} \alpha_1 - \alpha_0 &= d\alpha, \\ x_1 - x_0 &= R(\sin(\alpha_1) - \sin(\alpha_0)), \\ y_1 - y_0 &= -R(\cos(\alpha_1) - \cos(\alpha_0)). \end{aligned}$$

Моторами управляет контроллер. Он посылает управляющие сигналы дискретно во времени. Обозначим δ – шаг дискретизации, тогда формулы для последовательного вычисления координат можно записать в виде рекуррентной системы:

$$\begin{aligned} \alpha_{n+1} - \alpha_n &= \frac{D}{2K} \frac{\omega_R - \omega_L}{2} \delta, \\ x_{n+1} - x_n &= \frac{\omega_R + \omega_L}{\omega_R - \omega_L} K (\sin(\alpha_{n+1}) - \sin(\alpha_n)), \\ y_{n+1} - y_n &= -\frac{\omega_R + \omega_L}{\omega_R - \omega_L} K (\cos(\alpha_{n+1}) - \cos(\alpha_n)). \end{aligned} \quad (1.1)$$

В этой системе удобно перейти к другому, растянутому виду координат $x = KX$, $y = KY$. Важно, что новые координаты X, Y теперь не имеют размерности. Действительно, координата x имеет размерность метр в системе единиц СИ. Параметр K – половина колеи робота так же измеряется в метрах. Новая координата X – безразмерный коэффициент. Также безразмерная величина – новая переменная Y . Обозначим $\omega = (\omega_R + \omega_L)/2$, $\theta = (\omega_R - \omega_L)/2$. Удобно использовать еще одну безразмерную величину, связанную с шагом дискретизации, угловой скоростью вращения колес и

геометрическими размерами: $\Delta = \delta\omega D/(2K)$. Чтобы убедиться, что Δ – безразмерный параметр, можно вычислить его размерность, исходя из правой части выражения для Δ : (сек./сек.)(м/м). Тогда получится рекуррентная система в безразмерной форме:

$$\alpha_{n+1} - \alpha_n = \frac{\theta}{\omega} \Delta, \quad (1.2)$$

$$X_{n+1} - X_n = \frac{\omega}{\theta} (\sin(\alpha_{n+1}) - \sin(\alpha_n)), \quad (1.3)$$

$$Y_{n+1} - Y_n = -\frac{\omega}{\theta} (\cos(\alpha_{n+1}) - \cos(\alpha_n)). \quad (1.4)$$

Математические формулы в безразмерной форме удобны тем, что в них переменные привязаны к характерным параметрам и потому безразмерные переменные меняются в ограниченных пределах. Например, колея робота может равняться и нескольким сантиметрам, и быть более метра, но параметры X, Y системы (1.7) в обоих случаях будут меняться в пределах нескольких безразмерных единиц.

В системе (1.7) θ – управляющий параметр, определяющий скорость поворота робота относительно центра робота, параметр ω – определяет линейную скорость движения робота, Δ – задержка в системе управления.

1.1.4 Преобразование координат, отображение пространства в себя при движении робота

Формулы (1.4) можно переписать так, чтобы рассматривать их как отображение пространства $\mathbb{R}^2 \times \mathbb{S}^1$ в себя:

$$\alpha_{n+1} = \alpha_n + \frac{1}{\kappa} \Delta, \quad (1.5)$$

$$X_{n+1} = X_n + \kappa (\sin(\alpha_{n+1}) - \sin(\alpha_n)), \quad (1.6)$$

$$Y_{n+1} = Y_n - \kappa (\cos(\alpha_{n+1}) - \cos(\alpha_n)). \quad (1.7)$$

Здесь $\kappa = \omega/\theta$ – параметр, управляющий движением робота. В зависимости от значений управляющего параметра получаются различные траектории движения.

1.1.5 Дифференциальное уравнение движения робота

Из системы разностных уравнений можно перейти к системе уравнений, связывающих координаты и скорости. Для этого используем формальную переменную T , такую, что шаг по ней соответствует шагу $T_{n+1} - T_n = \Delta$.

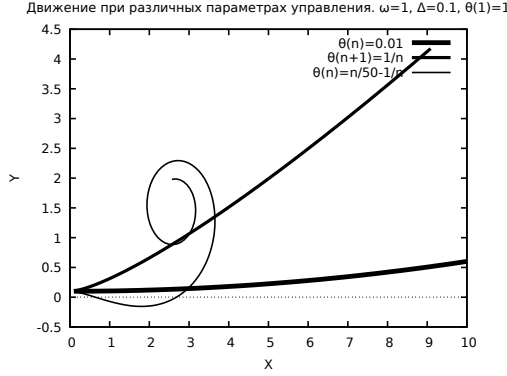


Рис. 1.5: Траектории движения робота при разных значениях управляющего параметра.

Для вывода дифференциальных уравнений из системы разностных уравнений (1.7) можно записать в разностном виде:

$$\begin{aligned}\frac{\alpha_{n+1} - \alpha_n}{\Delta} &= \frac{\theta}{\omega}, \\ \frac{X_{n+1} - X_n}{\Delta} &= \frac{\omega}{\theta} \frac{\sin(\alpha_{n+1}) - \sin(\alpha_n)}{\Delta}, \\ \frac{Y_{n+1} - Y_n}{\Delta} &= -\frac{\omega}{\theta} \frac{\cos(\alpha_{n+1}) - \cos(\alpha_n)}{\Delta}.\end{aligned}$$

В левых частях этой системы уравнений – средние скорости изменения координат. Если предположить, что параметр Δ может быть произвольно малым, тогда можно перейти к пределу $\Delta \rightarrow 0$. В этом случае вместо средних скоростей в левых частях формулы получатся мгновенные скорости. Правые части можно преобразовать, воспользовавшись тригонометрическими формулами:

$$\begin{aligned}\sin(\alpha_{n+1}) - \sin(\alpha_n) &= 2 \cos\left(\frac{\alpha_{n+1} + \alpha_n}{2}\right) \sin\left(\frac{\alpha_{n+1} - \alpha_n}{2}\right) = \\ &= 2 \cos\left(\alpha_n + \frac{\theta}{2\omega}\Delta\right) \sin\left(\frac{\theta}{2\omega}\Delta\right)\end{aligned}$$

Здесь в аргументе косинуса малым слагаемым с множителем Δ можно пренебречь, а для синуса вспомнить, что значение синуса малого угла близко к

значению самого угла, то есть:

$$\sin\left(\frac{\theta}{2\omega}\Delta\right) \sim \frac{\theta}{2\omega}\Delta.$$

Тогда можно записать:

$$\frac{\omega}{\theta} \frac{\sin(\alpha_{n+1}) - \sin(\alpha_n)}{\Delta} \sim \frac{\omega}{\theta} 2 \frac{1}{\Delta} \cos(\alpha_n) \frac{\theta}{2\omega} \Delta = \cos(\alpha_n).$$

В результате приведенных вычислений была получена производная функции $\sin(\alpha)$ по параметру α .

- По определению *производной функции* $f(x)$ является предел отношения приращения функции к приращению аргумента, когда приращение аргумента стремится к нулю или формально:

$$\frac{df}{dx} = \lim_{\Delta \rightarrow 0} \frac{f(x + \Delta) - f(x)}{\Delta}.$$

Такое же упражнение для тригонометрических формул разности косинусов даст:

$$-\frac{\omega}{\theta} \frac{\cos(\alpha_{n+1}) - \cos(\alpha_n)}{\Delta} \sim \sin(\alpha_n).$$

То есть при переходе к пределу $\Delta \rightarrow 0$ получатся уравнения:

$$\dot{\alpha} = \xi, \quad \dot{X} = \cos(\alpha), \quad \dot{Y} = \sin(\alpha). \quad (1.8)$$

Здесь $\xi = \theta/\omega$ – безразмерный управляющий параметр – отношение скорости поворота робота к скорости движения робота вдоль траектории.

Замечание 1 По существу выше были выведены формулы для производной функции синус и косинус. Действительно:

$$\begin{aligned} \frac{d \sin(\alpha)}{d\alpha} &= \lim_{\Delta \rightarrow 0} \frac{\sin(\alpha + \Delta) - \sin(\alpha)}{\Delta} = \lim_{\Delta \rightarrow 0} \frac{2 \cos\left(\frac{2\alpha + \Delta}{2}\right) \sin\left(\frac{\Delta}{2}\right)}{\Delta} = \\ &= \lim_{\Delta \rightarrow 0} \cos\left(\alpha + \frac{\Delta}{2}\right) \lim_{\Delta \rightarrow 0} \frac{\sin\left(\frac{\Delta}{2}\right)}{\frac{\Delta}{2}} = \cos(\alpha). \end{aligned}$$

Так же вычисляется и производная функции косинус:

$$\begin{aligned} \frac{d \cos(\alpha)}{d\alpha} &= \lim_{\Delta \rightarrow 0} \frac{\cos(\alpha + \Delta) - \cos(\alpha)}{\Delta} = \lim_{\Delta \rightarrow 0} \frac{-2 \sin\left(\frac{2\alpha + \Delta}{2}\right) \sin\left(\frac{\Delta}{2}\right)}{\Delta} = \\ &= - \lim_{\Delta \rightarrow 0} \sin\left(\alpha + \frac{\Delta}{2}\right) \lim_{\Delta \rightarrow 0} \frac{\sin\left(\frac{\Delta}{2}\right)}{\frac{\Delta}{2}} = - \sin(\alpha). \end{aligned}$$

1.1.6 Упражнения

Упражнение 1 Робот двигался по плоскости прямолинейными участками. Путь робота на плоскости – ломаная линия с координатами вершин $(1, 1), (2, -1), (3, 1), \dots, (n, (-1)^n)$. Вычислить длину пути робота и его угол поворота после $2N + 1$ -го участка.

Упражнение 2 Робот двигался по плоскости последовательно по нижним и верхним половинкам окружностей. Последовательные радиусы этих окружностей $1, 2, \dots, n$. Определить длину пути, который проехал робот за N участков.

Упражнение 3 Робот двигался по плоскости последовательно по дугам окружностей, проходя в $1/2$ раз меньшую угловую величину дуги на каждом следующем участке и поворачивая в одну и ту же сторону. Определить угол поворота робота после 10-го участка траектории.

Упражнение 4 Робот двигался по плоскости последовательно по дугам окружностей проходя в $1/4$ раз меньшую угловую величину дуги на каждом следующем участке. Радиус окружности каждого следующего участка уменьшался в $1/2$ раз. Определить координаты робота после 10-го участка траектории. Пусть радиус первой окружности R .

Упражнение 5 Во время движения робота левое и правое колеса вращались равномерно. Левое колесо со скоростью 2 оборота в минуту, правое – 3 оборота в минуту. Робот двигался 2 минуты. Начальное положение робота – точка с координатами $(0, 0, 0)$. Определить траекторию робота и вычислить угол поворота робота после остановки.

Упражнение 6 Робот сделал поворот на 90° по дуге радиуса 200 мм. Определить изменение координат робота на плоскости и пробег каждого из ведущих колес робота, если колея робота 100 мм.

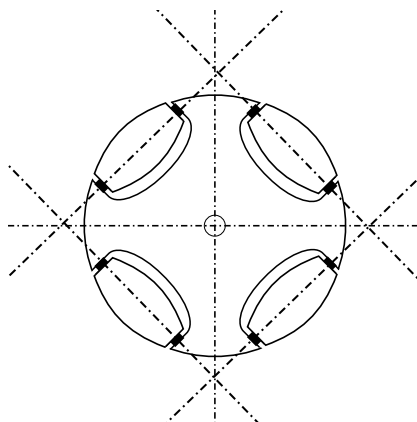


Рис. 1.6: Конструкция всенаправленного колеса

1.2 Геометрия голономного робота со всенаправленными колесами

1.2.1 Кинематика всенаправленного колеса

Конструкция всенаправленного колеса (омниколеса) приведена на рисунке (1.6)

Движение всенаправленного колеса определяется двумя осями вращения. Первая ось вращения – ось вращения колеса как целого. Вторая ось вращения – ось вращения рабочего ролика вокруг хорды колеса. Вектор, определяющий ось вращения колеса, и вектор, определяющий ось вращения рабочего ролика, в самом простом случае ортогональны, а в общем случае не коллинеарны. Здесь рассматривается кинематика омниколеса с ортогональными векторами осей вращения.

Омниколесо на оси без силового привода, например, мотора может двигаться в любом направлении на плоскости. Это позволяет вращение колеса как целого и вращение рабочего ролика в ортогональных направлениях.

Если омниколесо имеет рабочий привод, например, ось колеса является осью мотора, тогда вращение колеса как целого приводит к тому, что направление движения колеса имеет положительную проекцию на направление в котором колесо катилось бы, если бы не имело роликов по своему ободу.

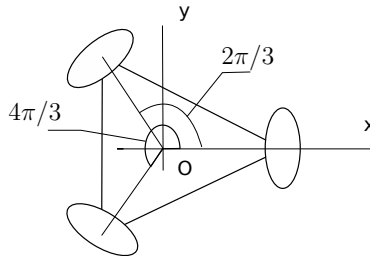


Рис. 1.7: Схема голономного робота. Радиус колес этого робота будет обозначаться R , расстояние от точки O до колес будет обозначаться L .

1.2.2 Схема робота

Голономный робот со всенаправленными колесами, который будет рассматриваться ниже, изображен на рисунке (1.7).

Этот робот может двигаться по плоскости в любом направлении. В его конструкции нет явно выбранного направления. Тем не менее ориентация осей всенаправленных колес робота важна для определения перемещений робота. Поэтому конфигурационное пространство робота на трех всенаправленных колесах также определяется с помощью двух координат на плоскости (x, y) и ориентации робота – угла α .

Для определенности пронумеруем колеса робота с первого по третье. Примем, что $\alpha = 0$, когда положительное направление оси вращения первого колеса совпадает с положительным направлением оси Ox .

1.2.3 Преобразования координат при повороте и перемещении. Прямая задача кинематики движения голономного робота

Пусть каждое из колес робота вращается со своей угловой скоростью. Выясним, как вычислить перемещение робота.

Обозначим α угол направления между осью вращения первого колеса робота и направлением оси координат Ox .

При вращении первого колеса вокруг своей оси со скоростью ω_1 вектор скорости перемещения колеса складывается из скорости качения колеса по плоскости как целого $\vec{W}_1 = (-R\omega_1 \sin(\alpha), R\omega_1 \cos(\alpha))$ и из ортогональной ей скорости качения опорного ролика $\vec{U}_1 = (u_1 \cos(\alpha), u_1 \sin(\alpha))$, рисунок 1.8. Здесь величина модуля скорости перемещения колеса за счет опорного

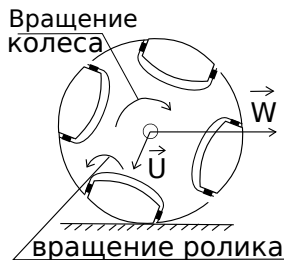


Рис. 1.8: Скорость движения всенаправленного колеса по плоскости складывается из линейной скорости оси вращения колеса за счет качения колеса, как целого \vec{W} и линейной скорости за счет качения ролика \vec{U} . На рисунке изображено колесо, которое катится направо. Вектор его угловой скорости направлен в сторону противоположную от читателя и здесь не указан.

ролика u_1 неизвестна.

В результате вектор скорости перемещения первого колеса:

$$\vec{v}_1 = (u_1 \cos(\alpha) - R\omega_1 \sin(\alpha), R\omega_1 \cos(\alpha) + u_1 \sin(\alpha)). \quad (1.9)$$

Похожие значения векторов скоростей, получающихся вращением второго \vec{v}_2 и третьего \vec{v}_3 колес:

$$\begin{aligned} \vec{v}_2 &= (u_2 \cos(\alpha + 2\pi/3) - R\omega_2 \sin(\alpha + 2\pi/3), \\ &\quad u_2 \sin(\alpha + 2\pi/3) + R\omega_2 \cos(\alpha + 2\pi/3)). \end{aligned} \quad (1.10)$$

Здесь неизвестная величина u_2 .

$$\begin{aligned} \vec{v}_3 &= (u_3 \cos(\alpha + 4\pi/3) - R\omega_3 \sin(\alpha + 4\pi/3), \\ &\quad u_3 \sin(\alpha + 4\pi/3) + R\omega_3 \cos(\alpha + 4\pi/3)). \end{aligned} \quad (1.11)$$

Здесь неизвестная величина u_3 .

В целом робот движется по плоскости xOy с поступательными составляющими скорости центра робота: $\vec{v} = (v_x, v_y)$ и с возможным вращением вокруг центра робота с угловой скоростью ω .

Поступательную и вращательную скорости движения робота теперь можно связать с составляющими скоростей каждого из трех колес.

Скорость перемещения первого колеса:

$$\vec{v}_1 = (v_x - L\omega \sin(\alpha), v_y + L\omega \cos(\alpha)). \quad (1.12)$$

Так же можно получить выражения для скоростей второго и третьего колес:

$$\vec{v}_2 = (v_x - L\omega \sin(\alpha + 2\pi/3), v_y + L\omega \cos(\alpha + 2\pi/3)); \quad (1.13)$$

$$\vec{v}_3 = (v_x - L\omega \sin(\alpha + 4\pi/3), v_y + L\omega \cos(\alpha + 4\pi/3)). \quad (1.14)$$

Теперь можно приравнять выражения для скоростей перемещения колес, полученные из условий собственного движения (1.9) – (1.11) и выражения для скоростей перемещения колес, полученные из условий движения робота как целого (1.12) – (1.14). В результате получится система из шести линейных уравнений для неизвестных величин: $u_1, u_2, u_3, v_x, v_y, \omega$.

$$\begin{aligned} u_1 \cos(\alpha) - R\omega_1 \sin(\alpha) &= v_x - L\omega \sin(\alpha), \\ R\omega_1 \cos(\alpha) + u_1 \sin(\alpha) &= v_y + L\omega \cos(\alpha), \\ u_2 \cos(\alpha + 2\pi/3) - R\omega_2 \sin(\alpha + 2\pi/3) &= v_x - L\omega \sin(\alpha + 2\pi/3), \\ u_2 \sin(\alpha + 2\pi/3) + R\omega_2 \cos(\alpha + 2\pi/3) &= v_y + L\omega \cos(\alpha + 2\pi/3), \\ u_3 \cos(\alpha + 4\pi/3) - R\omega_3 \sin(\alpha + 4\pi/3) &= v_x - L\omega \sin(\alpha + 4\pi/3), \\ u_3 \sin(\alpha + 4\pi/3) + R\omega_3 \cos(\alpha + 4\pi/3) &= v_y + L\omega \cos(\alpha + 4\pi/3). \end{aligned}$$

Процесс решения этой системы уравнений требует много простых вычислений. Лучше всего доверить такие вычисления системе компьютерной алгебры, например, *Mathima*. Ниже понадобятся формулы для ω, v_x, v_y . Именно для них и приведены формулы ниже:

$$\begin{aligned} \omega &= \frac{R}{3L}(\omega_1 + \omega_2 + \omega_3), \\ v_x &= \frac{R}{\sqrt{3}}(\omega_3 - \omega_2) \cos(\alpha) + \frac{R}{3}(\omega_3 + \omega_2 - 2\omega_1) \sin(\alpha); \\ v_y &= \frac{R}{3}(2\omega_1 - \omega_2 - \omega_3) \cos(\alpha) + \frac{R}{\sqrt{3}}(\omega_3 - \omega_2) \sin(\alpha). \end{aligned}$$

Теперь удобно переобозначить скорость собственного вращения робота: $\dot{\alpha} \equiv \omega$:

$$\dot{\alpha} = \frac{R}{3L}(\omega_1 + \omega_2 + \omega_3).$$

Если скорости вращения ω_i постоянны, тогда легко определить угол поворота робота относительно оси Ox :

$$\alpha(t) = \alpha_0 + \frac{R}{3L}(\omega_1 + \omega_2 + \omega_3)t,$$

Здесь α_0 – значение угла в начальный момент времени.

Также удобно переобозначить скорости: $\dot{x} \equiv v_x$, $\dot{y} \equiv v_y$

Скорость робота по координатам:

$$\dot{x} = \frac{R}{\sqrt{3}}(\omega_3 - \omega_2) \cos(\alpha) + \frac{R}{3}(\omega_3 + \omega_2 - 2\omega_1) \sin(\alpha),$$

$$\dot{y} = \frac{R}{3}(2\omega_1 - \omega_2 - \omega_3) \cos(\alpha) + \frac{R}{\sqrt{3}}(\omega_3 - \omega_2) \sin(\alpha).$$

При перемещении робота в пространстве из одной точки в другую вращение вокруг своей оси, как правило, приводит к дополнительному расходу энергии. Поэтому особенно интересен случай перемещений без вращения вокруг своей оси.

Условие перемещения без вращения – нулевое значение скорости $\dot{\alpha}$, то есть:

$$\omega_1 + \omega_2 + \omega_3 = 0.$$

В этом случае α – постоянная величина. Кроме того постоянными оказываются и скорости \dot{x} и \dot{y} . Это позволяет определить зависимость координат от времени:

$$x = x_0 + \left((\omega_3 - \omega_2) \cos(\alpha) + \frac{1}{\sqrt{3}}(\omega_3 + \omega_2 - 2\omega_1) \sin(\alpha) \right) \frac{Rt}{\sqrt{3}},$$

$$y = y_0 + \left(\frac{1}{\sqrt{3}}(2\omega_1 - \omega_2 - \omega_3) \cos(\alpha) + (\omega_3 - \omega_2) \sin(\alpha) \right) \frac{Rt}{\sqrt{3}}.$$

Здесь x_0 и y_0 – координаты робота на плоскости в начальный момент времени.

В ситуации общего положения

$$\omega_1 + \omega_2 + \omega_3 \neq 0.$$

Удобно обозначить:

$$\Omega = \frac{R}{3L}(\omega_1 + \omega_2 + \omega_3)$$

Тогда:

$$\alpha(t) = \alpha_0 + \Omega t.$$

Для определения зависимости координат от времени следует, зная выражения для скоростей \dot{x} и \dot{y} , найти значения координат. С формальной точки зрения это означает, что надо найти *первообразную*, то есть такую функцию, которая после дифференцирования дает исходную.

- Для первообразной используется специальное обозначение – *неопределенный интеграл*. По определению

$$F(t) = \int f(t)dt$$

– первообразная f или, что то же самое – неопределенный интеграл, если:

$$\frac{dF}{dt} = f(t).$$

Важно заметить, что первообразная определяется с точностью до произвольной постоянной. Так как постоянная величина не изменяется, то производная постоянной по любой переменной равна нулю. Отсюда следует свойство первообразной:

- Если F – первообразная f , то и $F + \text{const}$ – первообразная f .

В замечании 1 получены формулы для производных синуса и косинуса, их можно переписать:

$$\int \cos(\alpha)d\alpha = \sin(\alpha) + \text{const}, \quad \int \sin(\alpha)d\alpha = -\cos(\alpha) + \text{const}.$$

Значения постоянных подбираются, исходя из дополнительных условий. В нашем случае это дополнительное условие состоит в том, что в начальный момент времени вклад от первообразной от скорости равен нулю.

Нетрудно проверить, что таким условиям удовлетворяют первообразные вида:

$$\int \cos(\alpha_0 + \Omega\tau)d\tau = \frac{1}{\Omega} (\sin(\alpha_0 + \Omega t) - \sin(\alpha_0))$$

и

$$\int \sin(\alpha_0 + \Omega\tau)d\tau = \frac{1}{\Omega} (\cos(\alpha_0) - \cos(\alpha_0 + \Omega t)).$$

В результате получим формулы для преобразования координат робота при движении с постоянными скоростями вращения всенаправленных колес:

$$\begin{aligned} \alpha(t) &= \alpha_0 + \Omega t, \\ x &= x_0 + \frac{R}{\Omega\sqrt{3}}(\omega_3 - \omega_2) \sin(\alpha(t)) - \frac{R}{3\Omega}(\omega_3 + \omega_2 - 2\omega_1) \cos(\alpha(t)) - \\ &\quad - \frac{R}{\Omega\sqrt{3}}(\omega_3 - \omega_2) \sin(\alpha_0) + \frac{R}{3\Omega}(\omega_3 + \omega_2 - 2\omega_1) \cos(\alpha_0), \\ y &= y_0 + \frac{R}{3\Omega}(2\omega_1 - \omega_2 - \omega_3) \sin(\alpha(t)) - \frac{R}{\Omega\sqrt{3}}(\omega_3 - \omega_2) \cos(\alpha(t)) - \\ &\quad - \frac{R}{3\Omega}(2\omega_1 - \omega_2 - \omega_3) \sin(\alpha_0) + \frac{R}{\Omega\sqrt{3}}(\omega_3 - \omega_2) \cos(\alpha_0). \end{aligned}$$

Эти формулы дают решение прямой задачи кинематики для голономного робота.

Пример 2 Радиус колес робота $K = 0.025$ м, расстояние от каждого из колес до центра робота $L = 0.065$ м. Пусть в начальный момент координаты робота $(x_0, y_0, \alpha_0) = (0, 0, 0)$. Скорости вращения колес робота при $t \in (0, 5]$: $\omega_1 = 0$, $\omega_2 = 1$, $\omega_3 = -2$. Вычислить координаты робота при $t = 5$.

Решение. При $t \in (0, 5]$:

$$\Omega = \frac{0.025}{3 \cdot 0.065}(0 + 1 - 2) \sim -0.128,$$

Координаты в момент $t = 5$:

$$\alpha_1 \sim -0.128 \times 5 = -0.64;$$

$$\begin{aligned} x_1 &\sim \frac{0.025}{-0.128\sqrt{3}}(-2 - 1)\sin(-0.64) - \\ &\quad - \frac{0.025}{-3 \cdot 0.128}(-2 + 1)\cos(-0.64) + \\ &\quad + \frac{0.025}{-3 \cdot 0.128}(-2 + 1) \sim -0.19 \\ y_1 &\sim \frac{0.025}{-0.128 \cdot 3}(-1 + 2)\sin(-0.64) - \\ &\quad - \frac{0.025}{-0.128 \cdot \sqrt{3}}(-2 - 1)\cos(-0.64) + \\ &\quad + \frac{0.025}{-0.128 \cdot \sqrt{3}}(-2 - 1) \sim 0.11. \end{aligned}$$

1.2.4 Обратная задача кинематики голономного робота

Обратная задача кинематики для голономного робота – определение скоростей ω_k , $k \in \{1, 2, 3\}$ и времени T , необходимых для достижения роботом точки с координатами (x, y) . В ситуации общего положения задача определения четырех величин – три скорости и время – по заданным двум имеет неограниченное число решений.

Для упрощения обратной задачи обычно добавляется условие отсутствия собственного вращения при движении, тогда обратная задача сво-

дится к решению системы уравнений:

$$\begin{aligned} 0 &= \omega_1 + \omega_2 + \omega_3, \\ x &= x_0 + \left((\omega_3 - \omega_2) \cos(\alpha) + \frac{1}{\sqrt{3}}(\omega_3 + \omega_2 - 2\omega_1) \sin(\alpha) \right) \frac{Rt}{\sqrt{3}}, \\ y &= y_0 + \left(\frac{1}{\sqrt{3}}(2\omega_1 - \omega_2 - \omega_3) \cos(\alpha) + (\omega_3 - \omega_2) \sin(\alpha) \right) \frac{Rt}{\sqrt{3}}. \end{aligned}$$

Для дальнейшего анализа эту систему удобно переписать в матричной форме:

$$\begin{pmatrix} 1 & 1 & 1 \\ -\frac{2}{3} \sin(\alpha) & \frac{1}{\sqrt{3}} \sin(\alpha) - \cos(\alpha) & \frac{1}{\sqrt{3}} \sin(\alpha) + \cos(\alpha) \\ \frac{2}{3} \cos(\alpha) & \frac{-1}{\sqrt{3}} \cos(\alpha) - \sin(\alpha) & \frac{-1}{\sqrt{3}} \cos(\alpha) + \sin(\alpha) \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{(x-x_0)\sqrt{3}}{Rt} \\ \frac{(y-y_0)\sqrt{3}}{Rt} \end{pmatrix}. \quad (1.15)$$

Определитель матрицы для этой системы уравнений не зависит от величины угла α . Это можно проверить либо прямыми вычислениями на бумаге, либо, воспользовавшись какой-либо системой компьютерной алгебры (например, свободно распространяемой системой Maxima):

$$\begin{vmatrix} 1 & 1 & 1 \\ -\frac{2}{3} \sin(\alpha) & \frac{1}{\sqrt{3}} \sin(\alpha) - \cos(\alpha) & \frac{1}{\sqrt{3}} \sin(\alpha) + \cos(\alpha) \\ \frac{2}{3} \cos(\alpha) & \frac{-1}{\sqrt{3}} \cos(\alpha) - \sin(\alpha) & \frac{-1}{\sqrt{3}} \cos(\alpha) + \sin(\alpha) \end{vmatrix} = 2\sqrt{3}.$$

Система уравнений (1.15) имеет единственное решение для любых пар (x_0, y_0) и (x, y) и любого значения $t \neq 0$.

Следовательно можно сделать утверждение.

Утверждение 1 Найдется тройка угловых скоростей $(\omega_1, \omega_2, \omega_3)$ для перемещения без собственного вращения из любой начальной точки в любую конечную точку на плоскости за любое положительное время.

Пример 3 Диаметр всенаправленных колес робота 0.05 м. Начальное положение робота $(0, 0, \pi/4)$. Найти скорости вращения осей моторов $\omega_1, \omega_2, \omega_3$, при которых робот сможет попасть в точку с координатами $x = 0.3, y = 0.2$ за время $t = 10$ сек.

Решение.

Нужно решить систему уравнений:

$$\begin{pmatrix} 1 & 1 & 1 \\ \frac{-2}{3} \sin(\pi/4) & \frac{1}{\sqrt{3}} \sin(\pi/4) - \cos(\pi/4) & \frac{1}{\sqrt{3}} \sin(\pi/4) + \cos(\pi/4) \\ \frac{2}{3} \cos(\pi/4) & \frac{-1}{\sqrt{3}} \cos(\pi/4) - \sin(\pi/4) & \frac{-1}{\sqrt{3}} \cos(\pi/4) + \sin(\pi/4) \end{pmatrix} \times \\ \times \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{0.3\sqrt{3}}{0.025 \cdot 10} \\ \frac{0.2\sqrt{3}}{0.025 \cdot 10} \end{pmatrix}.$$

Решение имеет вид: $(\omega_1, \omega_2, \omega_3) \sim (-0.28, -1.08, 1.37)$.

1.2.5 Упражнения

Упражнение 7 Пусть диаметр всенаправленных колес робота $D = 0.05$ м. Робот движется по прямой $\alpha = 0$ со скоростью 0.3 м/с. Определить скорость вращения колес робота ω_2 и ω_3 . Считая, что ролики бочкообразные с минимальным диаметром 3мм. и максимальным диаметром 5мм. вычислить максимальную скорость вращения рабочего ролика (ролика, находящегося в зацеплении с поверхностью) ведущих колес). Ведущими называются колеса, на которые подается крутящий момент от моторов.

Указание. Воспользоваться величинами скоростей движения u_1, u_2, u_3 в направлении перпендикулярном направлению качения колеса.

Упражнение 8 Пусть диаметр всенаправленных колес робота $D = 0.1$ м. Робот стартует с позиции $x = 0, y = 0$. Угловая скорость вращения колес в течении 5 секунд была равна $\omega_1 = 1$ рад./сек., $\omega_2 = 2$ рад./сек., $\omega_3 = -3$ рад./сек.. Направление робота в начальный момент $\alpha = \pi/6$. Затем в течение 10 секунд угловая скорость вращения колес была равна $\omega_1 = 1$ рад./сек., $\omega_2 = -3$ рад./сек., $\omega_3 = 2$ рад./сек.. Определить направление и координаты робота через 15 сек. после старта.

Упражнение 9 Пусть диаметр всенаправленных колес робота $D = 0.05$ м., ролики бочкообразные, минимальный диаметр 7 мм., максимальный—10 мм., расстояние от центра робота до колеса 0.15 м. Робот развернулся вокруг собственной оси. Координаты на плоскости не изменились. Получить оценки сверху и оценку снизу для пробега ролика всенаправленных колес. Считать, что на ободе колеса – четыре ролика. В каждый момент в зацеплении находится только один ролик.

Указание. Воспользоваться величинами скоростей движения u_1, u_2, u_3 в направлении перпендикулярном направлению качения колеса.

Упражнение 10 Пусть диаметр всенаправленных колес робота $D = 0.1$ м. Определить скорости вращения колес, если роботу необходимо переместиться из позиции $(0, 0, \pi/3)$ в точку с координатами $x = 10, y = 3$ за 30 секунд.

1.3 Обзор литературы.

Литература по мобильным роботам с дифференциальным управлением довольно обширна. Для справок и более детального изучения можно воспользоваться работами [18], [20]. Там же можно посмотреть вывод и использование в несколько иной форме системы уравнений (1.8).

Хотя всенаправленное колесо и было запатентовано в 1919 году, подробных работ, по кинематике голономных устройств не столь много, как для неголономных роботов. В частности, в статье [3] можно посмотреть уравнения кинематики и некоторые вопросы об управлении движением. Подробный обзор и обсуждение задач для голономного робота есть в работе [9].

Глава 2

Манипуляторы. Тригонометрия и алгебра.

Манипуляторы – наиболее распространенные и сложные промышленные роботы. Здесь рассмотрены два типа манипуляторов:

- Многозвенные манипуляторы. Это манипуляторы, с разомкнутой кинематической цепью. Звенья манипуляторов связаны плоскими шарнирами.
- Дельта-роботы – манипуляторы с замкнутой кинематической цепью, построенные так, что рабочая платформа имеет постоянную ориентацию параллельно некоторой, обычно горизонтальной, плоскости.

Рассмотрено устройство конфигурационного многообразия, объем и линейные размеры рабочего пространства. Приведены частные решения прямой и обратной задач позиционирования.

2.1 Конфигурационное пространство робота-манипулятора

В этом разделе на примере робота-манипулятора введено понятие конфигурационного пространства. Получено отображение конфигурационного пространства в физическое пространство для свободного конца двухзвенника. Показано, как можно обобщить полученные формулы для двухзвенника в формулы для n -звенного манипулятора. Рассмотрено конфигурационное пространство схвата для робота манипулятора. На примере робота манипулятора со схватом рассмотрены многосвязные многообразия в кон-

фигурационном пространстве и введено понятие субриманова многообразия.

2.1.1 Однозвенный механизм

Для понимания термина «конфигурационное пространство робота» удобно рассмотреть манипулятор. Самый простой манипулятор представляет собой одно звено, соединенное шарниром с сервомотором. Сервомотор позволяет изменять положение звена в заданной плоскости на угол 2π . Это означает, что положение манипулятора определяется одной величиной – углом поворота.

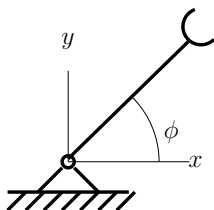


Рис. 2.1: Кинематическая схема однозвенного манипулятора

В свою очередь, угол поворота определяется как длина дуги окружности единичного радиуса, то есть в радианах. Следовательно конфигурационное пространство робота *топологически эквивалентно* окружности. Окружность обозначается символом \mathbb{S} , когда нужно подчеркнуть, что это одномерный объект: \mathbb{S}^1 .

- *Топологическая эквивалентность* одного объекта другому обычно означает, что с помощью растяжений и без разрывов можно их перевести друг в друга. Например, топологически эквивалентны круг и квадрат, шар и куб. Но кольцо и круг не являются топологически эквивалентными объектами.

Здесь удобно воспользоваться стандартной математической терминологией. Пусть ϕ – угол поворота звена, отсчитываемый от некоторой оси (назовем ее *полярной*), проходящей через шарнир, на котором закреплено звено. Пусть длина звена r .

- *Полярная система координат* определяется полярной осью – лучом, выходящим из полюса – точки начала отсчета O . Полярные координаты точки A – это расстояние до полюса r и угол между прямой (OA) и полярной осью.

Обозначим взаимно ортогональные оси системы координат на плоскости, в которой может вращаться звено через x, y . Причем направление оси x совпадает с направлением полярной оси. Начало координат совпадает с шарниром. Тогда координаты свободного конца звена могут быть определены по координате в конфигурационном пространстве ϕ :

$$x = r \cos(\phi), \quad y = r \sin(\phi). \quad (2.1)$$

Физически это означает, что при закреплении одного конца звена, второй конец звена может перемещаться по окружности с радиусом, равным длине звена r .

Подвижный конец звена обычно используется для рабочего инструмента. Рабочий инструмент может быть разным – схват, форсунка для краски, клеммы для сварки, цанга для режущего инструмента и многое другое. Ниже для простоты будем считать, что рабочий инструмент манипулятора – схват.

- *Рабочее пространство* – это область, в которой может находиться рабочий инструмент манипулятора.

Для рассмотренного однозвеного манипулятора рабочее пространство – окружность, радиус которой равен длине звена r .

2.1.2 Конфигурационное пространство двухзвенника

Следующий по сложности объект – *двухзвенник*. Пронумеруем звенья двухзвенника цифрами 1 и 2. Пусть один конец первого звена шарнирно закреплён, так, что звено может вращаться вокруг точки закрепления в плоскости. Второй конец первого звена шарнирно соединен с первым концом второго звена.



Рис. 2.2: Варианты сочленения первого и второго звеньев двухзвенника.

В шарнирном закреплении второго звена могут быть разные варианты. Пусть угол $\alpha \in [0, \pi)$ – угол между осями вращения шарнира 1 и шарнира 2. Этот угол является постоянным и определяется конструкцией сочленения

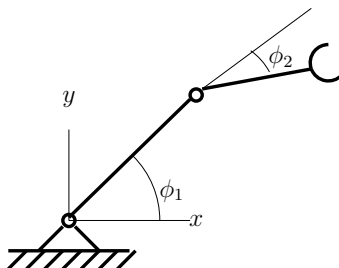


Рис. 2.3: Кинематическая схема плоского двухзвенника.

звеньев. Если $\alpha = 0$, тогда звенья 1 и 2 всегда лежат в одной и той же плоскости. Если $\alpha = \pi/2$, тогда звенья 1 и 2 вращаются в ортогональных плоскостях.

Положение каждого звена можно определить, воспользовавшись значениями углов поворота звеньев. Таких углов два, следовательно, конфигурационное пространство двухзвенника можно представить как *прямое произведение* двух окружностей $S^1 \times S^1$. Формально это означает, что к каждой точке на первой окружности приклеивается еще по одной окружности, лежащей, например, в ортогональном направлении. В результате получится *двумерный тор*. Для такой операции прямого произведения используется обозначение: $T^2 = S^1 \times S^1$.

- *Прямое произведение* одного множества на другое, означает, что образуется новое множество, состоящее из всевозможных пар элементов из первого и второго множества.

Пример 4 Например, прямое произведение двух прямых $\mathbb{R}^1 \times \mathbb{R}^1$ дает плоскость: $\mathbb{R}^1 \times \mathbb{R}^1 = \mathbb{R}^2$. Прямое произведение окружности и прямой $S^1 \times \mathbb{R}^1$ дает круговой цилиндр.

2.1.3 Плоский двухзвенник

Пусть $\alpha = 0$. Обозначим угол поворота первого звена ϕ_1 , длину первого звена — r_1 . Угол поворота второго звена относительно первого звена в шарнире, соединяющем первое и второе звенья — ϕ_2 . Длина второго звена r_2 .

Выведем формулы для положения конца второго звена в ортогональной системе координат на плоскости x, y , при заданных значениях координат ϕ_1 и ϕ_2 в конфигурационном пространстве.

Незакрепленный конец первого звена имеет координаты

$$x_1 = r_1 \cos(\phi_1), \quad y_1 = r_1 \sin(\phi_1). \quad (2.2)$$

Свободный конец второго звена:

$$x_2 = x_1 + r_2 \cos(\phi_1 + \phi_2), \quad y_2 = y_1 + r_2 \sin(\phi_1 + \phi_2).$$

В результате получены формулы для вычисления положения незакрепленного конца двухзвенника на плоскости:

$$x_2 = r_1 \cos(\phi_1) + r_2 \cos(\phi_1 + \phi_2), \quad y_2 = r_1 \sin(\phi_1) + r_2 \sin(\phi_1 + \phi_2) \quad (2.3)$$

Такие формулы обычно рассматривают как двумерную функцию, задающую отображение точки из пространства \mathbb{T}^2 в точку пространства \mathbb{R}^2 . Область определения этой функции – все пространство \mathbb{T}^2 – поверхность тора. Область значений – кольцо на плоскости x, y с внешним радиусом $R = r_1 + r_2$ и внутренним радиусом $r = |r_1 - r_2|$. То есть, если длины звеньев двухзвенника не совпадают, тогда свободный конец двухзвенника не может быть помещен в начало координат – точку $(0, 0)$ на плоскости x, y .

Утверждение 2 *Рабочее пространство плоского двухзвенника – кольцо с внешним радиусом $R = r_1 + r_2$ и внутренним радиусом $r = |r_1 - r_2|$.*

2.1.4 Двухзвенник с ортогональными осями шарниров

Теперь рассмотрим двухзвенник, в котором ось шарнира, соединяющего звенья 1 и 2, ортогональна оси шарнира, которым закреплено первое звено.

Возможны два способа положения оси шарнира в сочленении первого и второго звеньев двухзвенника. Первый – когда ось шарнира ортогональна первому звену. Второй способ – когда ось шарнира сонаправлена с первым звеном.

Здесь будет рассмотрен первый случай сочленения, который часто используется на практике. Второй случай будет предложено самостоятельно рассмотреть в упражнениях к этому разделу.

Конфигурационное пространство такого двухзвенника также состоит из прямого произведения окружностей и является тором $\mathbb{T}^2 = \mathbb{S}^1 \times \mathbb{S}^1$.

Вопрос в том, как устроено отображение этого двумерного пространства в физическое трехмерное пространство. Для этого опять удобно воспользоваться формулами, определяющими положение свободного конца двухзвенника.

Здесь плоскости вращения звеньев ортогональны, поэтому при изменении координат в конфигурационном пространстве, *вообще говоря*, положение свободного конца двухзвенника описывает пространственную кривую.

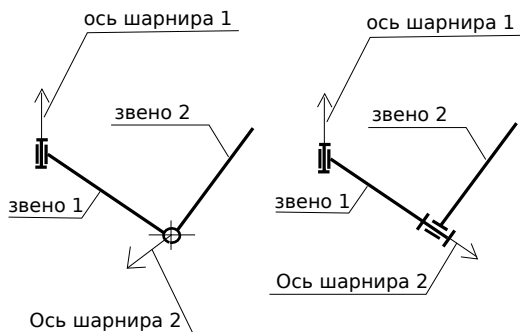


Рис. 2.4: Два способа соединения первого и второго звеньев. Слева – ось вращения шарнира ортогональна направлению первого звена. Справа – ось вращения шарнира совпадает с направлением первого звена.

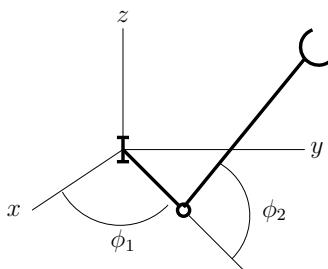


Рис. 2.5: Двухзвенник с ортогональными осями шарниров.

Поэтому придется воспользоваться координатами в трехмерной ортогональной системе координат.

- Сочетание «*вообще говоря*» обычно используется в математических текстах, чтобы обозначить наиболее общий случай. Пример такого высказывания: две прямых, вообще говоря, не лежат в одной плоскости.

Третью ось координат обозначим z . Направление выберем так, чтобы единичные векторы составляли *правую тройку координат*.

- Формальное правило, по которому строится *правая тройка координат* следующее. Пусть имеется координатная плоскость, на ней координатные оси Ox и Oy прямоугольной системы координат ориентированы так, что угол между осями $\pi/2$ и направление вращения от оси Ox к оси Oy – против вращения часовой стрелки. Построим ось Oz ортогональную плоскости xOy . Положительное направление оси Oz таково, что если смотреть с вершины оси Oz , тогда направление вращения от оси Ox к оси Oy по кратчайшему углу будет против часовой стрелки.
- Длинное объяснение можно сократить, если зажать все пальцы правой руки в кулак. Затем распрямить последовательно большой, указательный и средний. Примем направление большого пальца за ось Ox , указательного – за ось Oy и среднего – за ось Oz . В результате получится **правая тройка координат**.
- **Левую тройку координат** можно получить если поменять положительное направление оси Oz на противоположное. Либо, проделав манипуляции, описанные выше, с пальцами левой руки.

При заданных значениях ϕ_1 и ϕ_2 получим:

$$x_1 = r_1 \cos(\phi_1), \quad y_1 = r_1 \sin(\phi_1), \quad z_1 = 0.$$

То есть, плоскость x, y – плоскость, в которой вращается первое звено.

Координаты второго звена:

$$x_2 = x_1 + r_2 \cos(\phi_2) \cos(\phi_1),$$

$$y_2 = y_1 + r_2 \cos(\phi_2) \sin(\phi_1),$$

$$z_2 = z_1 + r_2 \sin(\phi_2).$$

Три координаты свободного конца двухзвенника определяются двумя координатами конфигурационного пространства. Это означает, что свободный конец двухзвенника может лежать на двумерной поверхности.

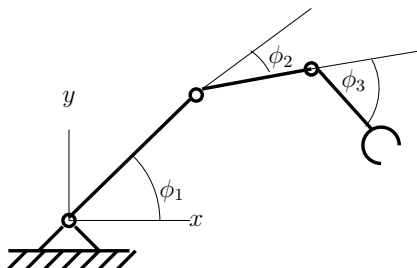


Рис. 2.6: Плоский трехзвенник

Нетрудно сообразить, что эта поверхность получается, если окружность радиуса r_2 поместить на расстоянии r_1 от начала координат и вращать вокруг оси z . При $r_1 > r_2$ получится двумерный тор в трехмерном евклидовом пространстве. При $r_1 < r_2$ – получится закрытый тор – двумерная поверхность, топологически эквивалентная сфере, при $r_1 = 0$ закрытый тор вырождается в сферу радиуса r_2 .

Утверждение 3 *Рабочее пространство двухзвенника с ортогональными осями шарниров – поверхность двумерного тора.*

2.1.5 Плоский n -звенник с избыточной степенью свободы

Если к плоскому двухзвеннику добавить третье звено, присоединив его с помощью шарнира так, что ось вращения шарнира направлена параллельно остальным двум осям трехзвенника, то свободный конец многозвенника будет лежать в плоскости перемещения остальных звеньев трехзвенника.

Формула для положения свободного конца такого трехзвенника:

$$\begin{aligned} x_1 &= r_1 \cos(\phi_1), & y_1 &= r_1 \sin(\phi_1), \\ x_2 &= x_1 + r_2 \cos(\phi_1 + \phi_2), & y_2 &= y_1 + r_2 \sin(\phi_1 + \phi_2), \\ x_3 &= x_2 + r_3 \cos(\phi_1 + \phi_2 + \phi_3), & y_3 &= y_2 + r_3 \sin(\phi_1 + \phi_2 + \phi_3), \\ z_1 &= 0, & z_2 &= 0, & z_3 &= 0. \end{aligned} \quad (2.4)$$

Область значений для координат свободного конца трехзвенника в евклидовом пространстве – двумерное кольцо, \mathcal{K}_2 с большим радиусом $R = r_1 + r_2 + r_3$ и малым радиусом $r = \min(|r_i - r_j - r_k|)$, где $i, j, k \in \{1, 2, 3\}$, $i \neq j \neq k$. Если малый радиус равен нулю, то кольцо превратится в круг радиуса R .

Согласно формулам (2.4) положение внутри кольца \mathcal{K} определяется тремя координатами конфигурационного пространства (ϕ_1, ϕ_2, ϕ_3) . Таким образом имеется отображение $\mathbb{T}^3 \rightarrow \mathcal{K}_2$. То есть трехмерное пространство отображается в двумерную область. Из общих соображений очевидно, что манипулятор имеет избыточное число степеней свободы. Формально его можно заменить на двухзвенник со специально подобранными длинами звеньев.

Нетрудно обобщить этот подход для плоского n -звенника. Конфигурационное пространство n -звенника – n -мерный тор: \mathbb{T}^n .

Область значений для координат свободного конца n -звенника в евклидовом пространстве – двумерное кольцо, \mathcal{K}_2 с большим радиусом:

$$R = \sum_{i=1}^n r_i$$

и малым радиусом

$$r = \min_{\mathcal{M}} \left(\left| \sum_{i \in \mathcal{M}} r_i - \sum_{j \notin \mathcal{M}} r_j \right| \right),$$

где минимум определяется перебором по всевозможным наборам индексов \mathcal{M} .

Формула для положения свободного конца n -звенника получается обобщением формулы для трехзвенника:

$$\begin{aligned} x_0 &= 0, & y_0 &= 0, & z_0 &= 0, \\ x_i &= x_{i-1} + r_i \cos \left(\sum_{j=1}^i \phi_j \right), \\ y_i &= y_{i-1} + r_i \sin \left(\sum_{j=1}^i \phi_j \right), \\ z_i &= 0, & i &\in \{1, \dots, n\}. \end{aligned} \tag{2.5}$$

Эти формулы определяют отображение $\mathbb{T}^n \rightarrow \mathcal{K}_2$.

Промышленные манипуляторы, как правило, имеют именно избыточное число степеней свободы. Это дает возможность использовать манипулятор в рабочих зонах с препятствиями для перемещений.

2.1.6 Манипулятор с тремя звеньями.

Выше, в разделах 2.1.3 и 2.1.4 было показано, что рабочий инструмент двухзвенника может лежать на двумерной поверхности. Для универсальных манипуляторов важно, чтобы множество положений свободного конца

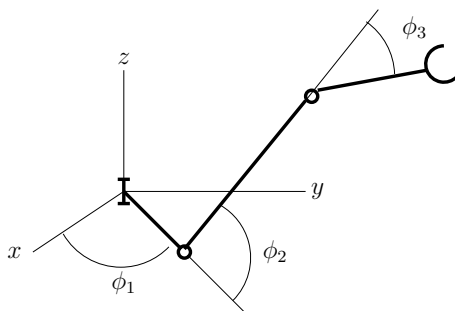


Рис. 2.7: Трехзвенный манипулятор.

манипулятора имело три измерения. Для этого достаточно к двухзвеннику с ортогональными осями шарниров добавить третье звено, например, с осью третьего шарнира, параллельной оси второго шарнира. В результате получится *трехзвенник*.

- Конфигурационное пространство такого трехзвенника *трёхмерный тор*: $\mathbb{T}^3 = \mathbb{T}^2 \times \mathbb{S}^1$.

Трёхмерный тор нельзя представить в привычном трёхмерном евклидовом пространстве, однако, его можно вложить в четырехмерное пространство. Сложность конфигурационного пространства только кажущаяся. На самом деле оно устроено очень просто: к каждой точке двумерного тора нужно приклеить окружность, так, чтоб эти приклеенные окружности не имели общих точек, то есть не пересекались.

Координаты конца второго звена вычисляются по формулам раздела 2.1.4. Координаты свободного конца такого трехзвенника можно вычислить

по уже готовым формулам раздела 2.1.3:

$$\begin{aligned}x_1 &= r_1 \cos(\phi_1), \\y_1 &= r_1 \sin(\phi_1), \\z_1 &= 0;\end{aligned}$$

$$\begin{aligned}x_2 &= x_1 + r_2 \cos(\phi_2) \cos(\phi_1), \\y_2 &= y_1 + r_2 \cos(\phi_2) \sin(\phi_1), \\z_2 &= r_2 \sin(\phi_2);\end{aligned}$$

$$\begin{aligned}x_3 &= x_2 + r_3 \cos(\phi_2 + \phi_3) \cos(\phi_1), \\y_3 &= y_2 + r_3 \cos(\phi_2 + \phi_3) \sin(\phi_1), \\z_3 &= z_2 + r_3 \sin(\phi_2 + \phi_3).\end{aligned}$$

В этих формулах координаты свободного звена трехзвенника получают через три независимых параметра – углы ϕ_1, ϕ_2, ϕ_3 . Следовательно, вообще говоря, координаты свободного конца трехзвенника определяют некоторое трехмерное многообразие в физическом пространстве \mathbb{R}^3 .

Первое звено трехзвенника вращается в плоскости (x, y) , второе и третье звенья отдельно составляют плоский двухзвенник. В разделе 2.1.3 показано, что координаты свободного конца плоского двухзвенника, вообще говоря, лежат в кольце. Теперь, если объединить вращение первого звена и кольцо двухзвенника из второго и третьего звена, получим:

Утверждение 4 *Рабочее пространство трехзвенного манипулятора – внутренность двумерного тора, образованного вращением по окружности радиуса r_1 центра кольца с внешним радиусом $R = r_2 + r_3$, внутренним радиусом $r = |r_2 - r_3|$.*

2.1.7 Схват манипулятора и многообразия в конфигурационном пространстве.

На свободный конец манипулятора обычно крепится либо схват, либо сварочный электрод, либо форсунка для распыления краски. Схват, электрод, форсунка, другие инструменты, устанавливаемые на манипуляторы, требуют правильной ориентации для их использования. Например, сварочный электрод или форсунка должны быть ориентированы ортогонально к рабочей поверхности. Схват манипулятора также требует специальную ориентацию для захвата объекта.

Изменение ориентации схвата или другого инструмента дает еще три степени свободы по трем угловым направлениям – то есть наклон схвата может меняться как $\psi_1 \in [0, \pi)$, поворот – $\psi_2 \in [0, 2\pi)$ и крен $\psi_3 \in [0, 2\pi)$.

- *Степень свободы* – возможность перемещения или вращения в некотором направлении. Например, степень свободы равна единице для объектов, перемещающихся по заданной кривой (прямой линии, окружности). Для объектов, перемещающихся на плоскости, степень свободы равна двум. Положение трехзвенного манипулятора определяется тремя независимыми величинами – углами, соответственно, он имеет три степени свободы.

Здесь углы ψ_1 и ψ_2 образуют всем привычную сферическую систему координат, которая используется для определения точки на поверхности Земли. В этой системе координат для угла ψ_1 «широты» достаточно изменения на угол π или, что то же самое, от -90° до 90° .

Дополнительно к ориентации в пространстве у схвата есть два состояния – схват замкнут или разомкнут. То есть конфигурационное пространство для схвата в общем случае представляет собой два экземпляра произведения сферы S^2 на окружность S^1 . Формально можно записать:

$$S^2 \times S^1 \times \{0, 1\}.$$

Добавление схвата с формальной точки зрения приводит к конфигурационному пространству манипулятора с n звеньями:

$$\mathbb{T}^{n+1} \times S^2 \times S^1 \times \{0, 1\}.$$

2.1.8 Упражнения

Упражнение 11 У двухзвенного плоского манипулятора длины звеньев $r_1 = 2$, $r_2 = 3$. Вычислить площадь рабочего пространства.

Упражнение 12 У двухзвенного объемного манипулятора длины звеньев $r_1 = 3$, $r_2 = 2$. Есть ограничения по углам поворота $\phi_1 \in [0, \pi)$ и $\phi_2 \in [-\pi/2, \pi/2]$. Вычислить площадь рабочего пространства.

Упражнение 13 Вычислить объем рабочего пространства трехзвенного манипулятора с длинами звеньев $r_1 = 1$, $r_2 = 2$, $r_3 = 1$. Оси вращения второго и третьего шарниров параллельны друг другу и перпендикулярны оси вращения первого шарнира.

Упражнение 14 У двухзвенного плоского манипулятора длины звеньев $r_1 = 1$, $r_2 = 2$. Есть ограничения по углам поворота $\phi_1 \in [0, \pi/2)$, $\phi_2 \in [-\pi/3, \pi/3]$. Вычислить площадь рабочего пространства.

2.2 Прямая и обратная задачи позиционирования схвата манипулятора

2.2.1 Обратная задача для двухзвенника

Есть две задачи позиционирования свата манипулятора – прямая и обратная.

- *Прямой задачей* позиционирования будем называть задачу о вычислении координат свободного конца манипулятора по заданным координатам в конфигурационном пространстве.

Формулы (2.5) дают решение прямой задачи позиционирования. Важное свойство прямой задачи – решение всегда существует и единственно. То есть, для любой точки конфигурационного пространства можно однозначно вычислить положение свободного конца манипулятора.

- *Обратной задачей* позиционирования будем называть задачу об определении координат свободного конца манипулятора в конфигурационном пространстве по заданным координатам свободного конца манипулятора в физическом пространстве.

Замечание 2 Координаты в конфигурационном пространстве определяют положение звеньев манипулятора. С этой точки зрения можно переформулировать обратную задачу, как задачу о поиске необходимого положения звеньев манипулятора.

Замечание 3 Обратная задача позиционирования не всегда разрешима.

Пример 5 Если взять плоский двухзвенный манипулятор и поставить для него задачу поместить свободный конец вне плоскости его перемещений, задача окажется неразрешимой.

Пример 6 Для плоского двухзвенного манипулятора с длинами звеньев $r_1 = 2$ и $r_2 = 1$ окажется неразрешимой задача – поместить свободный конец манипулятора в начало координат.

Ниже сформулировано очевидное утверждение.

Утверждение 5 Обратная задача позиционирования разрешима, если требуемое положение в физическом пространстве принадлежит области значений функции, определяющей множество положений свободного конца манипулятора.

Пример 7 Для плоского двухзвенника с длинами звеньев $r_1 = 1$, $r_2 = 2$ и положением свободного конца двухзвенника в координатах $(1.2, 1.5, 0)$ обратная задача имеет два решения. Положения звеньев в конфигурационном пространстве: $(\phi_1, \phi_2) \sim (-0, 72; 1, 90)$ и $(\phi_1, \phi_2) \sim (2, 06; -1, 90)$. (См. упражнение 15).

Из приведенного примера следует, что решение обратной задачи позиционирования для манипулятора, вообще говоря, неединственно.

2.2.2 Обратная задача для плоского трехзвенника

Рассмотрим трехзвенник с параллельными осями шарниров, то есть такой, что свободный конец трехзвенника лежит в плоскости. Схема такого трехзвенника изображена на рисунке 2.6.

Утверждение 6 Множество положений плоского трехзвенника, для которых обратная задача разрешима, лежат в плоском кольце. Если исключить границу кольца, тогда обратная задача для плоского трехзвенника имеет множество решений – состояний ϕ_1, ϕ_2, ϕ_3 . В этом множестве каждый из параметров ϕ_i может изменяться в некотором диапазоне.

Прежде всего важно привести общее соображение, которое привело к этому утверждению. Конфигурационное пространство трехзвенника – трехмерный тор: \mathbb{T}^3 . В физическом пространстве свободный конец трехзвенника может лежать в плоском кольце. То есть \mathbb{T}^3 из конфигурационного пространства отображается в двумерное кольцо \mathcal{K}^2 , прямая задача представляет собой отображение:

$$\mathbb{T}^3 \rightarrow \mathcal{K}^2.$$

Обратная – отображение из двумерного множества в трехмерное.

$$\mathcal{K}^2 \rightarrow \mathbb{T}^3.$$

В ситуации общего положения это отображение вырождено.

- Словосочетание *ситуация общего положения* означает общий случай. Частные, именно, исключительные случаи, не рассматриваются. Ими пренебрегают, как редкими. Например, в ситуации общего положения два вектора неколлинеарны, четыре точки не лежат в одной плоскости.

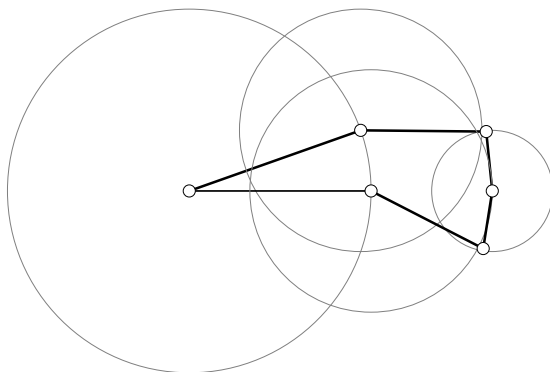


Рис. 2.8: Различные позиции плоского трехзвенного манипулятора для решения обратной задачи кинематики со вспомогательными окружностями – возможными положениями звеньев. В обеих конфигурациях начальное положение первого звена и конечное положение третьего звена имеют одни и те же координаты.

Действительно, на плоскости при малом изменении каждой из трех конфигурационных координат получатся близкие к линейным перемещения по хордам $\vec{e}_1, \vec{e}_2, \vec{e}_3$. В ситуации общего положения ни одна из пар этих перемещений не коллинеарна. То есть получился набор из трех векторов на плоскости. Известно, что любой вектор на плоскости можно выразить через два неколлинеарных вектора. Следовательно, существует набор ненулевых чисел α_i таких, что

$$\alpha_1 \vec{e}_1 + \alpha_2 \vec{e}_2 + \alpha_3 \vec{e}_3 = 0.$$

Ниже приведена схема конструктивного доказательства.

Если обратная задача разрешима, следовательно, разрешима система уравнений:

$$x = r_1 \cos(\phi_1) + r_2 \cos(\phi_1 + \phi_2) + r_3 \cos(\phi_1 + \phi_2 + \phi_3), \quad (2.6)$$

$$y = r_1 \sin(\phi_1) + r_2 \sin(\phi_1 + \phi_2) + r_3 \sin(\phi_1 + \phi_2 + \phi_3). \quad (2.7)$$

Если возвести левые и правые части уравнений в квадрат, а затем сложить результаты, получится:

$$y^2 + x^2 = 2r_1 r_3 \cos(\phi_3 + \phi_2) + 2r_2 r_3 \cos(\phi_3) + 2r_1 r_2 \cos(\phi_2) + r_3^2 + r_2^2 + r_1^2. \quad (2.8)$$

Это уравнение для ломаной длиной $\sqrt{x^2 + y^2}$. Неизвестные в этом уравнении – углы ϕ_2 и ϕ_3 . В уравнении отсутствует неизвестный угол ϕ_1 . Пример разных конфигураций приведен на рисунке 2.8.

Упорядочим длины звеньев. Для этого обозначим $R = \max\{r_1, r_2, r_3\}$ – длина самого длинного звена, $r = \min\{r_1, r_2, r_3\}$ – длина самого короткого звена и ρ – длина оставшегося звена, так что: $r \leq \rho \leq R$.

Нетрудно видеть, что для $R - (r + \rho) < \sqrt{y^2 + x^2} < r_1 + r_2 + r_3$ существуют интервалы значений углов ϕ_3 и ϕ_2 и получена ломаная, для которой расстояние от закрепленного конца трехзвенника до свободного конца составляет $\sqrt{x^2 + y^2}$. Угол ϕ_1 определит такой поворот ломаной, при котором свободный конец трехзвенника окажется в точке с координатами (x, y) .

2.2.3 Обратная задача для трехзвенника в трехмерном пространстве

Обратная задача для трехзвенника с тремя степенями свободы в трехмерном пространстве сводится к решению системы трех уравнений:

$$\begin{aligned}x_3 &= (r_1 + r_2 \cos(\phi_2) + r_3 \cos(\phi_2 + \phi_3)) \cos(\phi_1), \\y_3 &= (r_1 + r_2 \cos(\phi_2) + r_3 \cos(\phi_2 + \phi_3)) \sin(\phi_1), \\z_3 &= r_2 \sin(\phi_2) + r_3 \sin(\phi_2 + \phi_3).\end{aligned}$$

Здесь угол ϕ_1 определяет положение плоскости, в которой находится двухзвенник, состоящий из второго и третьего звеньев. Поэтому для решения обратной задачи удобно преобразовать формулы так, чтобы отдельно определить угол поворота первого звена, а затем свести задачу к обратной задаче для плоского двухзвенника в плоскости движения второго и третьего звена.

Плоскость, в которой движутся второе и третье звенья, определяется вертикальной осью Oz и вектором в горизонтальной плоскости xoy : \vec{OA} . Координаты точки $A = (x_3, y_3)$, поэтому угол ϕ_1 :

$$\phi_1 = \arctan \left(\frac{x_3}{y_3} \right).$$

После того, как определен ϕ_1 , для решения обратной задачи можно взять третье уравнение системы и любое из первых двух. Для определенности возьмем первое уравнение. Рассмотрим случай, когда $\cos(\phi_1) \neq 0$. Обозначим: $X_3 = x_3 - r_1 \cos(\phi_1)$, $Z_3 = z_3 \cos(\phi_1)$, $R_2 = r_2 \cos(\phi_1)$, $R_3 = r_3 \cos(\phi_1)$, тогда система уравнений похожа на систему уравнений обратной задачи для плоского двухзвенника:

$$X_3 = R_2 \cos(\phi_2) + R_3 \cos(\phi_2 + \phi_3), \quad (2.9)$$

$$Z_3 = R_2 \sin(\phi_2) + R_3 \sin(\phi_2 + \phi_3). \quad (2.10)$$

Сформулируем результат.

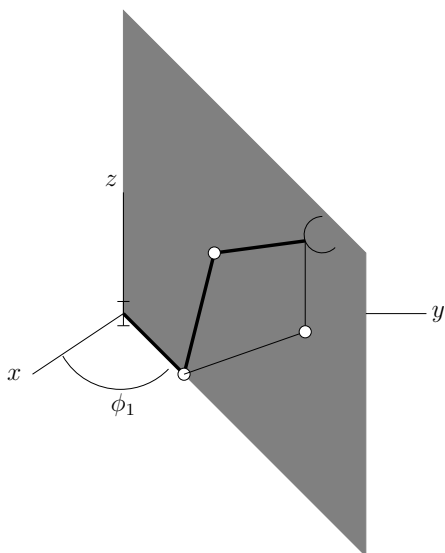


Рис. 2.9: Трехзвенный манипулятор в трехмерном пространстве. Угол ϕ_1 определяет плоскость, в которой находятся второе и третье звенья манипулятора. Обратная задача кинематики имеет два решения. Каждому из решений соответствует определенная конфигурация звеньев манипулятора. На рисунке одна возможная конфигурация изображена толстыми линиями, вторая – тонкими.

Утверждение 7 Обратная задача для трехзвенника с тремя степенями свободы имеет два решения, если координаты свободного конца трехзвенника лежат в \mathcal{T}_3 . Здесь \mathcal{T}_3 – внутренность двумерного тора. Сечение этого тора – кольцо с внешним радиусом $R = r_2 + r_3$, внутренним радиусом $r = |r_2 - r_3|$. Центр кольца лежит на окружности радиуса r_1 .

Решение обратной задачи для неплоского трехзвенника определяет отображение

$$\mathcal{T}_3 \rightarrow \mathbb{T}^3.$$

В общем случае это отображение невырождено. Пример двух конфигураций трехзвенного манипулятора приведен на рисунке 2.9.

2.2.4 Обратная задача для неплоского n -звенника при $n > 3$

Система уравнений обратной задачи для неплоского n -звенника может быть выписана с помощью комбинации тех же соображений, что в задачах для плоского n -звенника и неплоского трехзвенника.

Разрешимая обратная задача для неплоского n -звенника в ситуации общего положения представляет собой отображение из некоторой области трехмерного пространства \mathcal{T}_3^n в n -мерный тор:

$$\mathcal{T}_3^n \rightarrow \mathbb{T}^n.$$

В ситуации общего положения множество решений представляет собой область размерности $n - 3$ в \mathbb{T}^n .

2.2.5 Оптимальное решение обратной задачи

Обратная задача для манипулятора в ситуации общего положения имеет не одно, а много решений. Понятно, что нужно рассмотреть все множество решений и для практической цели выбрать оптимальное с той или иной точки зрения.

Понятие оптимальности относительное. Например, для удержания манипулятора с грузом в точке с заданными координатами (X, Y, Z) оптимальной может оказаться конфигурация, при которой необходимые моменты, развиваемые сервомоторами, будут минимальны. Для точного позиционирования в той же точке (X, Y, Z) может оказаться оптимальной другая конфигурация внутренних параметров (углов) манипулятора.

2.2.6 Упражнения

Упражнение 15 (Плоский двухзвенник) Пусть дан плоский двухзвенник. Длины звеньев $r_1 = 1$, $r_2 = 2$. Задача – вычислить координаты в конфигурационном пространстве для заданных координат свободного конца двухзвенника $(1.2, 1.5, 0)$.

Для начала необходимо определить, разрешима ли поставленная обратная задача. Область значений координат свободного конца заданного двухзвенника – кольцо на плоскости $z = 0$ с внутренним радиусом $R_0 = 1$ и внешним радиусом $R_1 = 3$. Точка с координатами $(1, 5; 1, 2; 0)$ лежит на окружности радиуса $R = \sqrt{1,5^2 + 1,2^2} = \sqrt{3.69} \sim 1,921$. Следовательно, задача разрешима.

Согласно формулам (2.3) решение задачи сводится к решению системы уравнений:

$$1,5 = \cos(\phi_1) + 2 \cos(\phi_1 + \phi_2), \quad 1,2 = \sin(\phi_1) + 2 \sin(\phi_1 + \phi_2).$$

Неизвестные в этой системе уравнений – углы ϕ_1 и ϕ_2 .

Из системы уравнений можно выделить уравнение для угла ϕ_2 . Для этого нужно возвести правые и левые части каждого из уравнений в квадрат. Полученные уравнения сложить, в результате получится уравнение:

$$3,69 = 4 \cos(\phi_2) + 5.$$

Это уравнение имеет два решения:

$$\phi_2 = \arccos(-1,31/4) \quad \text{и} \quad \phi_2 = -\arccos(-1,31/4).$$

Теперь можно в первом уравнении системы воспользоваться формулой для косинуса суммы:

$$1,5 = \cos(\phi_1) + 2 \cos(\phi_1) \cos(\phi_2) - 2 \sin(\phi_1) \sin(\phi_2).$$

Это уравнение удобно переписать в виде:

$$1,5 = (1 + 2 \cos(\phi_2)) \cos(\phi_1) - 2 \sin(\phi_2) \sin(\phi_1).$$

Теперь можно обозначить: $A = \sqrt{(1 + 2 \cos(\phi_2))^2 + (2 \sin(\phi_2))^2}$, $\tan(\psi) = 2 \sin(\phi_2)/(1 + 2 \cos(\phi_2))$. В результате уравнение примет вид:

$$1,5 = A(\cos(\psi) \cos(\phi_1) - \sin(\psi) \sin(\phi_1)),$$

или

$$1,5 = A \cos(\phi_1 + \psi).$$

Его решение: $\phi_1 = \arccos(1,5/A) - \psi$.

После прямой подстановки двух значений ϕ_2 получится два ответа для поставленной задачи: $(\phi_1, \phi_2) \sim (-0,72; 1,90)$ и $(\phi_1, \phi_2) \sim (2,06; -1,90)$

Упражнение 16 Дан плоский двухзвенник. Длины звеньев: $r_1 = 1$, $r_2 = 2$. Известны значения углов в конфигурационном пространстве $\phi_1 = \pi/3$, $\phi_2 = -\pi/6$. Определить декартовы координаты рабочего инструмента.

Упражнение 17 Дан плоский двухзвенник. Длины звеньев: $r_1 = 1$, $r_2 = 2$. Известны значения углов в конфигурационном пространстве $\phi_1 = \pi/3$, $\phi_2 = -\pi/6$. Определить уравнение прямой, на которой находится второе звено двухзвенника.

Упражнение 18 Дан двухзвенник с ортогональными осями шарниров. Длины звеньев: $r_1 = 3$, $r_2 = 1$. Известны значения углов в конфигурационном пространстве $\phi_1 = -\pi/3$, $\phi_2 = \pi/6$. Определить уравнение прямой, на которой находится второе звено двухзвенника.

Упражнение 19 Дан двухзвенник с ортогональными осями шарниров. Длины звеньев: $r_1 = 2$, $r_2 = 1$. Известны значения углов в конфигурационном пространстве $\phi_1 = -\pi/3$, $\phi_2 = \pi/6$. Определить координаты рабочего инструмента.

Упражнение 20 Дан плоский двухзвенник. Длины звеньев $r_1 = 1$, $r_2 = 2$. Определить, возможна ли конфигурация углов поворота ϕ_1 , ϕ_2 , при которой рабочий инструмент находится в точке с координатами (1.5, 2). Если такая конфигурация возможна, найти все решения.

Упражнение 21 Дан двухзвенник с ортогональными осями шарниров. Длины звеньев: $r_1 = 1$, $r_2 = 2$. Определить, возможна ли конфигурация углов поворота ϕ_1 , ϕ_2 , при которой рабочий инструмент находится в точке с координатами (1.2, 1.4, 0.8). Если такая конфигурация возможна, найти все решения.

Упражнение 22 Дан трехзвенник. Длины звеньев $r_1 = 1$, $r_2 = 2$, $r_3 = 1$. Ось шарнира первого звена сонаправлена с осью Oz . Оси шарниров второго и третьего звеньев параллельны друг другу и ортогональны оси шарнира первого звена. Определить координаты рабочего инструмента в декартовом пространстве, если углы поворота в конфигурационном пространстве: $\phi_1 = \pi/6$, $\phi_2 = \pi/3$, $\phi_3 = \pi/4$.

2.3 Матричное представление положения манипулятора

2.3.1 Матрицы поворота

Если рассматривать координаты манипулятора в разных положениях как комбинацию отображений, то удобно от геометрического подхода перейти к формальному матричному.

На плоскости свободный конец плоского n -звенника можно рассматривать как комбинацию применений преобразований поворота.

Пусть ось поворота Oz , тогда при повороте на угол ϕ вектор с координатами (x, y, z) преобразуются по правилу

$$\begin{aligned}x' &= x \cos(\phi) - y \sin(\phi), \\y' &= x \sin(\phi) + y \cos(\phi), \\z' &= z.\end{aligned}$$

В этом преобразовании видно, что вектор новых координат линейно зависит от вектора старых координат. Рассмотрим по отдельности каждое из уравнений. Правую часть первого уравнения можно представить как скалярное умножение вектора $(\cos(\phi), -\sin(\phi), 0)$ на вектора координат (x, y, z) . Правую часть второго уравнения – в виде произведения вектора $(\sin(\phi), \cos(\phi), 0)$ на тот же вектор координат (x, y, z) .

- Принято векторы $(\cos(\phi), -\sin(\phi), 0)$ и $(\sin(\phi), \cos(\phi), 0)$ располагать в виде строк матрицы, а вектор (x, y, z) в виде столбца:

$$\begin{pmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x \cos(\phi) - y \sin(\phi) \\ x \sin(\phi) + y \cos(\phi) \end{pmatrix}.$$

В результате правая часть первого уравнения получается как скалярное произведение первой строки матрицы на столбец вектора координат, правая часть второго уравнения – скалярное произведение второй строки матрицы на столбец вектора координат. Для того, чтобы по тому же правилу получить третье из уравнений, нужно к матрице добавить третью строку вида $(0, 0, 1)$.

- В результате установлено правило – для умножения матрицы на вектор-столбец необходимо каждую строку матрицы скалярно умножить на вектор-столбец, результат умножения записать в виде вектора-столбца. Для простоты нужно запомнить – умножают *строку мат-*

рицы, стоящей слева, на столбец, стоящий справа. **Порядок следования важен: матрица – слева, столбец – справа!**

- Матрица

$$A_Z(\phi) = \begin{pmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

называется матрицей поворота относительно оси Oz .

Также можно определить матрицы поворота относительно оси (Ox):

$$A_X(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{pmatrix}$$

и относительно оси Oy :

$$A_Y(\phi) = \begin{pmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{pmatrix}.$$

Примем без доказательства важное утверждение.

Утверждение 8 *Поворот на произвольный угол относительно произвольной оси можно представить как композицию поворотов относительно осей Ox, Oy и Oz .*

Матрицами поворота удобно пользоваться при вычислении положения манипулятора с заданной конфигурацией. Для использования приведенного утверждения необходимо уметь использовать умножение матриц.

2.3.2 Умножение матриц

Способ умножения матриц прямо следует из разобранного выше умножения матрицы на столбец. Пусть матрица слева имеет n строк и m столбцов. Матрица справа – m строк и k столбцов. Для умножения матрицу слева нужно представить как матрицу из n векторов-строк по m элементов, матрицу справа нужно представить, как составленную из m векторов-столбцов по k элементов. Тогда для умножения левой матрицы на правую нужно брать строки левой матрицы и умножать их на столбцы правой матрицы, результат записывать в пересечение соответствующих строки и столбца. После перемножения всех строк на столбцы получится матрица из n строк и k столбцов.

Пример 8 Например, матрица 3×3 умножается на матрицу из одного столбца: 1×3 . То есть $n = 3$, $m = 3$, $k = 1$ в результате получится 3 строки и один столбец.

Пример 9 Ниже показано пошаговое умножение матрицы 3×2 на матрицу 2×3 .

- В левой матрице две строки ($m = 2$), а в правой два столбца ($k = 2$), следовательно в результате должна получиться матрица размером 2×2 .

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ -2 & 2 \\ 3 & -3 \end{pmatrix}.$$

- Скалярное умножение вектора первой строки левой матрицы на вектор первого столбца правой, результат – первый элемент первого столбца:

$$\begin{pmatrix} 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix} = \begin{pmatrix} 6 \end{pmatrix}.$$

- Скалярное умножение первой строки левой матрицы на второй столбец правой, результат – первый элемент второго столбца:

$$\begin{pmatrix} 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ -3 \end{pmatrix} = \begin{pmatrix} -6 \end{pmatrix}.$$

- Скалярное умножение второй строки левой матрицы на первый столбец правой, результат – второй элемент первого столбца:

$$\begin{pmatrix} 3 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \\ 3 \end{pmatrix} = \begin{pmatrix} 2 \end{pmatrix}.$$

- Скалярное умножение второй строки левой матрицы на второй столбец правой, результат – второй элемент второго столбца:

$$\begin{pmatrix} 3 & 2 & 1 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ -3 \end{pmatrix} = \begin{pmatrix} -2 \end{pmatrix}.$$

- Окончательно получим:

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ -2 & 2 \\ 3 & -3 \end{pmatrix} = \begin{pmatrix} 6 & -6 \\ 2 & -2 \end{pmatrix}.$$

- Важнейшее свойство умножения матриц – *некоммутативность*:

$$A \times B \neq B \times A$$

Пример 10 *Пример некоммутативности произведения матриц:*

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix}.$$

- Произведение AB :

$$AB = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 5 & 0 \\ 4 & -3 \end{pmatrix}.$$

- Произведение BA :

$$BA = \begin{pmatrix} 1 & -2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} -3 & 0 \\ 4 & 5 \end{pmatrix}.$$

Очевидно:

$$\begin{pmatrix} 5 & 0 \\ 4 & -3 \end{pmatrix} \neq \begin{pmatrix} -3 & 0 \\ 4 & 5 \end{pmatrix}.$$

Пример 11 *Вычислим координаты свободного конца однозвенника с длиной звена r_1 при повороте на ϕ_1 вокруг оси Oz в плоскости xOy :*

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = A_Z(\phi_1) \begin{pmatrix} r_1 \\ 0 \\ 0 \end{pmatrix} \equiv \begin{pmatrix} \cos(\phi_1) & -\sin(\phi_1) & 0 \\ \sin(\phi_1) & \cos(\phi_1) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} r_1 \cos(\phi_1) \\ r_1 \sin(\phi_1) \\ 0 \end{pmatrix}.$$

Пример 12 *Приведем формулы для вычисления координат свободного конца плоского двухзвенника с длинами звеньев r_1 и r_2 при повороте первого звена на ϕ_1 вокруг оси Oz в плоскости xOy , второго звена – на угол ϕ_2 вокруг оси, параллельной оси Oz :*

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = A_Z(\phi_1) \begin{pmatrix} r_1 \\ 0 \\ 0 \end{pmatrix} + A_Z(\phi_1) A_Z(\phi_2) \begin{pmatrix} r_2 \\ 0 \\ 0 \end{pmatrix}$$

Пример 13 Пусть у двухзвенника ось первого шарнира Oz , ось второго шарнира параллельной оси Oy . Первое звено повернуто на угол ϕ_1 , второе – угол ϕ_2 . Тогда формула для координат свободного конца:

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = A_Z(\phi_1) \begin{pmatrix} r_1 \\ 0 \\ 0 \end{pmatrix} + A_Z(\phi_1)A_Y(\phi_2) \begin{pmatrix} r_2 \\ 0 \\ 0 \end{pmatrix}$$

2.3.3 Манипулятор с n -звеньями в одной плоскости.

В разделе 2.1.6 сконструирован трехзвенник со свободным концом, который может принимать произвольное положение внутри двумерного тора. Вообще говоря, этого достаточно для простейшего манипулятора. Однако часто для дополнительной гибкости манипуляторы имеют более чем три звена и, соответственно, больше чем три плоских шарнира.

Конфигурационное пространство n -звенника – n -мерный тор. Методом индукции, последовательно присоединяя к $n - 1$ -му звену следующее, можно написать формулу, определяющую координату свободного конца n -звенника в физическом трехмерном пространстве.

Положим, что для свободного звена $(n - 1)$ -звенника есть формулы для положения свободного конца $(n - 1)$ -го звена в физическом пространстве x_{n-1} , y_{n-1} . Кроме того, известны координаты в конфигурационном пространстве $\phi_1, \dots, \phi_{n-1}$.

Пусть к свободному звену $(n - 1)$ -звенника присоединено n -ое звено длины r_n с плоским шарниром.

Будем считать, что все звенья лежат в плоскости xOy . Тогда все звенья поворачиваются относительно оси Oz . Формула для координат свободного конца n -го звена в матричной форме примет вид:

$$\begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} = A_Z(\phi_1) \dots A_Z(\phi_n) \begin{pmatrix} x_{n-1} \\ y_{n-1} \\ z_{n-1} \end{pmatrix}. \quad (2.11)$$

2.3.4 Упражнения

Упражнение 23 Пусть 3-х звенный манипулятор имеет звенья $r_1 = 0$, $r_2 = 1$, $r_3 = 2$. Ось вращения первого звена совпадает с осью Oz , оси вращения второго и третьего звеньев ортогональны оси вращения первого звена и параллельны друг другу. Определить координаты рабочего органа манипулятора, если углы поворота звеньев $\phi_1 = \pi/6$, $\phi_2 = \pi/3$ и $\phi_3 = -\pi/6$.

Упражнение 24 Пусть 3-х звенный манипулятор имеет звенья $r_1 = 1$, $r_2 = 2$, $r_3 = 1$. Ось вращения первого звена совпадает с осью Oz , оси вращения второго и третьего звеньев ортогональны оси вращения первого звена и параллельны друг другу. Определить координаты рабочего органа манипулятора, если углы поворота звеньев $\phi_1 = -\pi/6$, $\phi_2 = \pi/4$ и $\phi_3 = -\pi/6$.

Упражнение 25 Пусть 6-звенный манипулятор имеет звенья $r_1 = 0$, $r_k = (6 - k)$, $k = 2, 3, 4, 5, 6$. Ось вращения первого звена совпадает с осью Oz , оси вращения остальных звеньев ортогональны оси вращения первого звена и параллельны друг другу. Определить координаты рабочего органа манипулятора, если углы поворота звеньев $\phi_1 = -\pi/6$, $\phi_k = (-1)^k \pi / (3 + k)$.

Упражнение 26 Пусть шестизвенный манипулятор имеет звенья $r_1 = 0$, $r_{k+1} = (6 - k)$, $k = 1, 2, 3, 4, 5$. Ось вращения первого звена совпадает с осью Oz , оси вращения остальных звеньев ортогональны оси вращения первого звена и параллельны друг другу. Определить угол между плоскостью xOy и последним звеном манипулятора, если углы поворота звеньев $\phi_1 = -\pi/6$, $\phi_k = (-1)^k \pi / (3 + k)$.

2.4 Манипулятор дельта-робота.

Дельта-робот был придуман Реймондом Клавелом. В 1991 он защитил кандидатскую диссертацию «Концепция быстрого параллельного робота с четырьмя степенями свободы». В основе его манипулятора используются два треугольника, один как неподвижная закрепленная платформа, второй – как подвижная платформа для рабочего инструмента. Эти платформы похожи на прописную греческую букву Δ – «дельта», по-видимому отсюда и название «дельта-робот».

2.4.1 Конфигурация плоского манипулятора с параллельным механизмом.

Для начала рассмотрим геометрию и кинематические свойства плоского манипулятора на основе параллельного механизма. Для плоского движения хорошо известен механизм, позволяющий сохранить параллельную ориентацию элементов во время движения – параллельный механизм.¹ Здесь в качестве рычагов плоского четырехзвенника используются параллелограммы.

¹Такой механизм применялся еще в конструкции паровой машины Уатта.

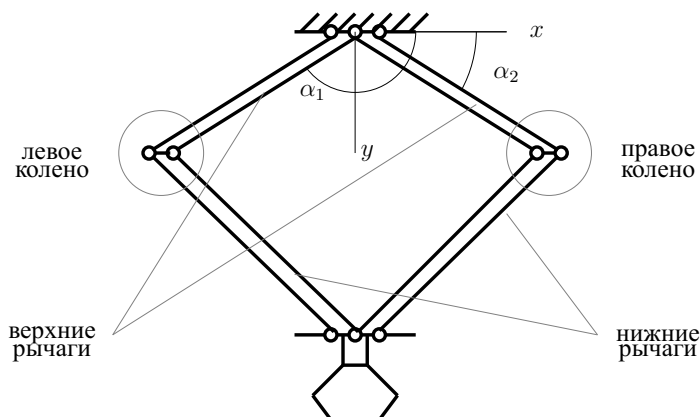


Рис. 2.10: Схема плоского замкнутого четырехзвенного механизма. Пара верхних рычагов и пара нижних рычагов механизма представляют собой параллелограммы. Поэтому нижняя подвижная платформа со схватом всегда параллельна верхней неподвижной платформе. На схеме указаны углы поворота верхних рычагов механизма α_1 и α_2 .

Использование параллелограммов приводит к тому, что подвижная платформа с рабочим инструментом всегда параллельна неподвижной платформе манипулятора.

Действительно, если короткая сторона параллелограмма закреплена, а смежные с ней длинные стороны смогут параллельно двигаться при изменении углов на закрепленной стороне, тогда сторона, противоположная закрепленной, всегда будет параллельна ей. Если все четыре рычага четырехзвенника заменить на параллелограммы, тогда рабочая платформа будет всегда параллельной неподвижной платформе.

Для упрощения расчетов будем пренебрегать линейными размерами платформ. Для простоты один из шарниров механизма, будем называть неподвижной платформой. Координаты неподвижной платформы примем за начало отсчета в декартовой системе координат. Здесь приводятся упрощенные расчеты, в которых будем пренебрегать также и горизонтальными размерами перемычек рычагов-параллелограммов. Координаты левого колена робота обозначим x_1, y_1 , координаты правого колена – x_2, y_2 , координаты схвата – x, y . Длины верхних рычагов обозначим r_1 , длины нижних – r_2 .

Из чертежа следует, что положение рабочего инструмента робота опре-

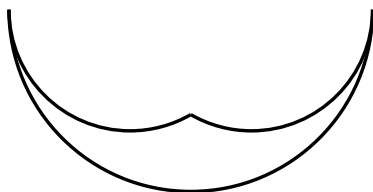


Рис. 2.11: Рабочее пространство плоского замкнутого четырехзвенного механизма. С верхними рычагами $r_1 = 1$, нижними рычагами $r_2 = 2$. Пространство представляет собой сегмент круга $R = r_1 + r_2 = 3$ с вырезанными сегментами кругов радиусов $r_2 = 2$. На дуге большой окружности точное позиционирование механизма невозможно.

деляется углами поворота верхних рычагов в начале координат. Обозначим эти углы α_1 и α_2 . Обычно из-за конструктивных ограничений верхние рычаги могут поворачиваться от $-\pi$ до 0 . Конфигурационное пространство робота – прямое произведение двух открытых интервалов $(\pi, 0) \times (-\pi, 0)$.

Для определения рабочего пространства нужно рассмотреть всевозможные крайние положения углов α_1 и α_2 . Графические решения задачи о рабочем пространстве дает чертеж 2.11.

2.4.2 Прямая и обратная задачи кинематики для плоского замкнутого четырехзвенника

Схема плоского замкнутого четырехзвенника изображена на рисунке 2.10. При повороте левого верхнего рычага на угол α_1 координаты левого колена:

$$x_1 = r_1 \cos(\alpha_1) \quad y_1 = r_1 \sin(\alpha_1).$$

Также можно выразить координаты правого колена при повороте правого верхнего рычага на угол α_2 :

$$x_2 = r_1 \cos(\alpha_2) \quad y_2 = r_1 \sin(\alpha_2).$$

Координаты рабочего инструмента удовлетворяют системе уравнений:

$$\begin{aligned} (x - x_1)^2 + (y - y_1)^2 &= r_2^2, \\ (x - x_2)^2 + (y - y_2)^2 &= r_2^2. \end{aligned} \tag{2.12}$$

Теперь в уравнения (2.12) можно подставить выражения для x_1, y_1 и x_2, y_2 . В результате получится система уравнений:

$$\begin{aligned} (x - r_1 \cos(\alpha_1))^2 + (y - r_1 \sin(\alpha_1))^2 &= r_2^2, \\ (x - r_1 \cos(\alpha_2))^2 + (y - r_1 \sin(\alpha_2))^2 &= r_2^2. \end{aligned}$$

Эти уравнения можно упростить, воспользовавшись тригонометрическим тождеством.

$$\begin{aligned} x^2 + y^2 - 2xr_1 \cos(\alpha_1) - 2yr_1 \sin(\alpha_2) &= r_2^2 - r_1^2, \\ x^2 + y^2 - 2xr_1 \cos(\alpha_2) - 2yr_1 \sin(\alpha_2) &= r_2^2 - r_1^2. \end{aligned} \quad (2.13)$$

Для определения координат рабочего инструмента необходимо решить систему двух квадратных уравнений (2.13).

Заметим, что при $\alpha_1 = \alpha_2$ оба уравнения системы (2.13) оказываются одинаковыми. В результате система уравнений вырождается в одно уравнение для двух неизвестных. Такая конфигурация соответствует границе рабочего пространства – дуге большой окружности. Нетрудно видеть, что число решений этого уравнения неограниченно. Следовательно дуга большой окружности хотя и может быть достижима, но должна быть исключена из рабочей области, так как на дуге большой окружности невозможно точное позиционирование рабочего инструмента.

В остальных случаях из чертежа следует, что решение системы уравнений (2.13) единственно.

2.4.3 Параллельный механизм в составе манипулятора дельта-робота.

В основе геометрической конструкции дельта-робота лежит шестизвенный механизм 2.12. Три нижних звена этого механизма представляют собой рычаги на основе параллелограмма. По своим кинематическим свойствам этот механизм похож на четырехзвенный механизм, разобранный в разделе 2.4.1.

Манипулятор дельта-робота сконструирован так, что его рабочая платформа при любых перемещениях параллельна неподвижной платформе.

2.4.4 Конфигурационное пространство дельта-робота

Обозначим три звена шестизвенника непосредственно присоединенных к основанию l_i^1 , $i = 1, 2, 3$. Эти звенья присоединены к основанию плоскими шарнирами.

Звенья, соединяющиеся в зоне крепления рабочего инструмента, будем обозначать l_i^2 , $i = 1, 2, 3$.

Для простоты вычислений будем считать, что три звена l_i^1 имеют одну и ту же длину: $l_i^1 = r_1$, три звена l_i^2 имеют другую длину $l_i^2 = r_2$.

Еще одно упрощение – вместо двух плоских шарниров с ортогональными осями вращения в сочленении верхних рычагов и нижних рычагов

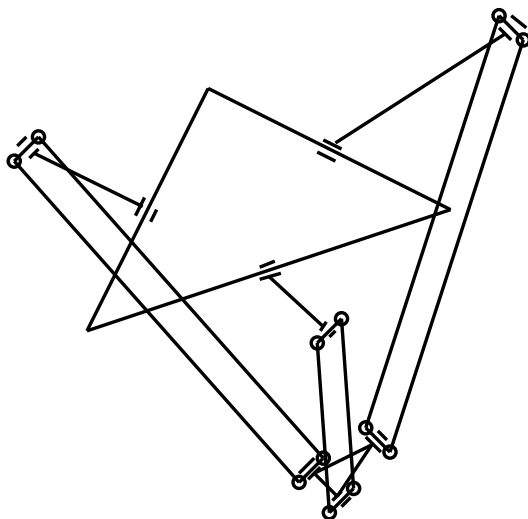


Рис. 2.12: Кинематическая схема манипулятора дельта-робота. Верхняя треугольная платформа неподвижна. Нижняя треугольная платформа подвижна. Рычаги, шарнирно прикрепленные к верхней платформе, управляются сервомоторами. Рычаги, удерживающие нижнюю платформу, собраны в виде параллелограммов. В результате плоскость нижней платформы всегда остается параллельна плоскости верхней платформы.

будем считать, что шарниры в сочленениях колена шаровые, то есть позволяют звеньям вращаться друг относительно друга в любом направлении. Это существенно упрощает расчеты и несколько ухудшает их точность.

Общее число степеней свободы рабочего инструмента – три. Это перемещение по декартовым осям координат в физическом пространстве.

Конфигурационное пространство определяют углы поворота рычагов в точке закрепления к неподвижной платформе. Каждый рычаг может поворачиваться относительно оси вращения плоского шарнира. Из-за конструктивных ограничений обычно это углы от $\phi_i \in (0, \pi)$. Конструкция позволяет независимо изменять углы поворота. Конфигурационное пространство – внутренность трехмерного куба со сторонами длиной π : $\mathcal{K}^3 = (0, \pi) \times (0, \pi) \times (0, \pi)$.

Граница рабочего пространства определяется полусферой радиуса $R = r_1 + r_2$ и тремя сегментами полусфер радиуса r_2 , рисунок 2.13.

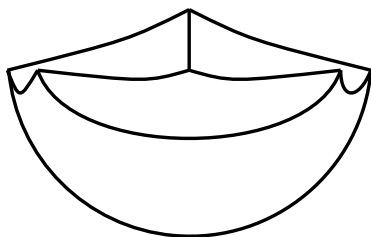


Рис. 2.13: Рабочее пространство шестизвенного механизма – основы дельта-робота. Пространство ограничено снизу сегментом сферы радиуса $r_1 + r_2$, сверху – тремя сегментами сфер радиуса r_2 с центрами, расположенными в вершинах правильного треугольника.

2.4.5 Прямая и обратная задачи для дельта-робота

Примем за начало отсчета декартовой системы координат точку крепления неподвижной платформы. Пусть все три оси плоских шарниров лежат в плоскости xOy , ось Ox совпадает с осью вращения первого рычага, ось вращения второго рычага направлена под углом $\pi/3$ к оси Ox , ось вращения третьего рычага направлена под углом $2\pi/3$ к оси Ox .

Поворот первого рычага на угол ϕ_1 приводит к тому, что координаты шарового шарнира на первом рычаге примут значения:

$$x_1 = 0, \quad y_1 = -r_1 \cos(\phi_1), \quad z_1 = -r_1 \sin(\phi_1).$$

Поворот второго рычага на угол ϕ_2 :

$$\begin{aligned} x_2 &= r_1 \cos(2\pi/3 - \pi/2) \sin(\phi_2), \\ y_2 &= -r_1 \sin(2\pi/3 - \pi/2) \cos(\phi_2), \\ z_2 &= -r_1 \sin(\phi_2). \end{aligned}$$

Координаты сферического шарнира третьего рычага:

$$\begin{aligned} x_3 &= r_1 \cos(4\pi/3 - \pi/2) \sin(\phi_3), \\ y_3 &= -r_1 \sin(4\pi/3 - \pi/2) \cos(\phi_3), \\ z_3 &= -r_1 \sin(\phi_3). \end{aligned}$$

Координаты подвижной платформы, на которой закреплен рабочий инструмент, обозначаются x, y, z . Расстояние до каждого из шарниров равно

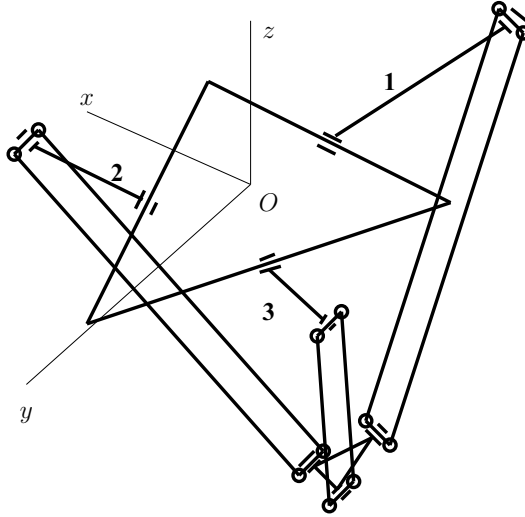


Рис. 2.14: Начало координат – верхняя треугольная платформа. Будем пренебрегать ее линейными размерами – считать, что они малы, хотя здесь на схеме она изображена довольно большой. Рычаги, шарнирно прикрепленные к верхней платформе управляются сервомоторами. Эти рычаги здесь пронумерованы цифрами 1,2,3. Углы поворота рычагов к плоскости верхней платформы (xOy) здесь не указаны.

r_2 . Формально это условие выражается в виде трех алгебраических уравнений второго порядка:

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = r_2^2, \quad i = 1, 2, 3.$$

Теперь, заменив x_i, y_i, z_i на их выражения через углы ϕ_i , получим систему алгебраических уравнений.

$$\begin{aligned} x^2 + (y + r_1 \cos(\phi_1))^2 + (z + r_1 \sin(\phi_1))^2 &= r_2^2, \\ (x - r_1 \cos(2\pi/3 - \pi/2) \sin(\phi_2))^2 + \\ (y + r_1 \sin(2\pi/3 - \pi/2) \cos(\phi_2))^2 + (z + r_1 \sin(\phi_2))^2 &= r_2^2, \\ (x - r_1 \cos(4\pi/3 - \pi/2) \sin(\phi_3))^2 + \\ (y + r_1 \sin(4\pi/3 - \pi/2) \cos(\phi_3))^2 + (z + r_1 \sin(\phi_3))^2 &= r_2^2. \end{aligned} \quad (2.14)$$

Если в системе уравнений заданы углы ϕ_i , то с помощью этой системы уравнений можно решить прямую задачу кинематики для дельта-робота –

получить координаты рабочей платформы (x, y, z) . Если заданы координаты рабочей платформы (x, y, z) , то можно получить значения углов ϕ_i и, тем самым, решить обратную задачу кинематики для дельта-робота.

2.4.6 Упражнения

Упражнение 27 Пусть плоский четырехзвенный механизм, изображенный на рисунке 2.10, имеет размеры: длина верхних рычагов $r_1 = 1$, длина нижних рычагов $r_2 = 2$. Углы поворота верхних рычагов $\alpha_1 = \pi/3$, $\alpha_2 = \pi/4$. Определить координаты рабочего инструмента.

Упражнение 28 Пусть плоский четырехзвенный механизм, изображенный на рисунке 2.10, имеет размеры: длина верхних рычагов $r_1 = 2$, длина нижних рычагов $r_2 = 3$. Определить, может ли подвижная платформа находиться в точке с координатами $(1, -4)$. Если может – определить углы поворота верхних рычагов α_1, α_2 .

Упражнение 29 Пусть плоский четырехзвенный механизм, изображенный на рисунке 2.10, имеет размеры: длина верхних рычагов $r_1 = 1$, длина нижних рычагов $r_2 = 2$. Определить площадь рабочего пространства.

Упражнение 30 Пусть плоский четырехзвенный механизм, изображенный на рисунке 2.10, имеет размеры: длина верхних рычагов $r_1 = 3$, длина нижних рычагов $r_2 = 2$. Этот механизм оснастили поворотным механизмом верхней платформы. Ось вращения платформы вертикальна и проходит через начало координат. В результате рабочее пространство робота – внутренность фигуры вращения рабочего пространства плоского четырехзвенного механизма см. рис. 2.11. Определить объем рабочего пространства такого манипулятора.

Упражнение 31 Пусть манипулятор дельта-робота 2.12 имеет размеры: длина верхних рычагов $r_1 = 1$, длина нижних рычагов $r_2 = 3$. Углы поворота верхних рычагов $\phi_1 = \pi/3$, $\phi_2 = \pi/4$, $\phi_3 = \pi/3$. Определить координаты рабочего инструмента.

Упражнение 32 Пусть манипулятор дельта-робота 2.12 имеет размеры: длина верхних рычагов $r_1 = 1$, длина нижних рычагов $r_2 = 1$. Выяснить, может ли рабочая платформа оказаться в точке с координатами $(0.5, 0.3, -1.8)$. Если может, то найти углы поворота верхних рычагов относительно неподвижной платформы.

2.5 Обзор литературы

Манипуляторы – наиболее распространенный тип робототехнических систем на производстве. Поэтому список учебников, в той или иной мере связанных с рассмотренной здесь темой, огромен. Основная книга – это классический учебник для ВТУЗов И.И. Артоболевский, Теория машин и механизмов [10]. Большое число книг по робототехнике содержит разделы о кинематике манипуляторов, например: [24], [23].

Про манипулятор дельта-робота можно почитать в кандидатской диссертации Рейдмонда Клавела [2].

Глава 3

Методы обработки сигналов. Аппроксимация и статистика.

Система управления робота, как правило, получает несколько потоков данных от сенсоров. До принятия решения о реакции на полученные значения обычно производится фильтрация полученных данных. Из общего потока выделяются данные, считающиеся достоверными. Далее они передаются для анализа системе управления.

В этом разделе приведены несколько распространенных фильтров. Основное отличие фильтров, применяющихся в робототехнике, от статистической обработки экспериментальных данных вообще – обработка данных должна производиться в режиме реального времени с минимально возможной задержкой.

- Для реакции на показания датчика важно, чтобы данные измерений были достоверными.
- Для получения достоверных данных весь поток данных с датчика можно подвергать фильтрации.
- Разработано много способов фильтрации данных, например, длительный интервал, требование дополнительного подтверждения и усреднение.
- Критически важные операции, такие как калибровка датчика, требуют повышенного внимания к достоверности данных.

Человек для ориентации в окружающей среде использует огромное количество данных от своих органов чувств и принимает решение, исходя из всей совокупности. У робота таких данных очень мало. Скорее всего у

вашего робота датчик освещенности окажется единственным источником данных об окружении. Поэтому для использования показаний датчика нужно быть уверенным в их достоверности.

Показание датчика, например, датчика освещенности, может оказаться искаженным из-за приёма сигнала от вспышки фотоаппарата, мерцающих ламп дневного света или светодиодных источников. Причиной искажений могут быть и шумы в самом датчике. Поэтому важно научиться отделять достоверные показания от недостоверных. Для этого используются различные фильтры. Ниже будет рассказано о программных фильтрах.

Важно понимать, что любой фильтр замедляет работу робота. Поэтому далеко не всегда есть необходимость в их использовании. Но если предполагается, что робот будет работать в среде с высоким уровнем шума, например, на соревнованиях или выставках, где может быть большое количество фотокамер со вспышками, стробоскопические источники света, то программный фильтр может оказаться простым и эффективным средством.

3.1 Общая структура потока данных

Рассмотрим в качестве примера последовательность измерений некоторой величины x в разные моменты времени t_i , где индекс i пробегает несколько значений. Пусть в результате последовательности измерений получен набор $\{\xi_i\}$. Понятно, что этот набор данных не является в точности значением величины x в моменты времени t_i . Чтобы применить этот набор $\{\xi_i\}$ для определения значений величины x обычно пользуются некоторой моделью.

Предположим, что измерение представляет собой сумму истинного значения измеряемой величины x_i , систематической погрешности измерения Δ_i и случайной погрешности ζ_i :

$$\xi_i = x_i + \Delta_i + \zeta_i.$$

Природа систематической и случайной погрешности существенно различна. Систематическая погрешность возникает в конечном итоге из-за неверной математической модели проводящихся измерений. Это вполне нормально, что модель учитывает только несколько основных факторов, влияющих на процесс измерения, и не учитывает огромное количество остальных.

Систематическая ошибка обычно проявляется во всех сделанных измерениях. При близких условиях систематическая ошибка мало меняется. Поэтому систематическую ошибку, как правило, выявляют на этапе калибровки датчика.

Случайная погрешность в измерениях считается в принципе неустранимой. Эта погрешность непостоянна и, что важно, при измерениях в одних

и тех же условиях случайная погрешность меняется.

3.1.1 Свойства случайной величины

В математике принято изучать случайные величины, которые удовлетворяют той или иной *статистической гипотезе* – предположению о распределении вероятностей появления значений случайной величины.

Типичный пример, в котором появляется случайная величина – процесс подбрасывания монеты. Выпадение орла будем отмечать единицей, выпадение решки – нулем. Отношение числа единиц к общему числу испытаний называется частотой выпадения орла. Считается, что орел и решка выпадают с одинаковой частотой и при неограниченном числе испытаний частота – она же вероятность выпадения орла – равна $1/2$.

- *Среднее значение*, полученное из набора данных $\{\xi_i\}$, вычисляется по формуле:

$$\mu = \frac{1}{N} \sum_{i=1}^N \xi_i.$$

- Если сделать M разных наборов по N измерений, то получится M средних значений μ_j , $j \in \{1, \dots, M\}$. Здесь важно заметить, что среднее значение случайных величин – также случайная величина.
- Если предположить, что есть вычисленное среднее неограниченной последовательности измерений или среднее по неограниченному числу наборов измерений, то такое среднее называется *математическим ожиданием*:

$$M(\xi) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \xi_i.$$

В математической статистике предполагается, что

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \xi_i = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{j=1}^M \mu_j.$$

То есть, основное предположение – среднее значение неограниченной суммы случайных величин не является случайной величиной. Если сформулировать строго, то среднее сходится к некоторой детерминированной величине с ростом числа наблюдений.

Более того, неявно предполагается, что верна так называемая *эргодическая гипотеза* – переход к пределу по числу измерений в одной последовательности и переход к пределу по числу последовательностей дают одинаковый результат. Если опустить детали, то она кратко формулируется так: среднее по времени равно среднему по пространству.

- Важной характеристикой является *дисперсия* – разброс данных около среднего значения:

$$D(\xi) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N (\mu - \xi_i)^2$$

- Наряду с дисперсией в математической статистике используется формула:

$$\sigma = \lim_{N \rightarrow \infty} \sqrt{\frac{1}{N} \sum_{i=1}^N (\mu - \xi_i)^2}$$

Величина σ , вычисленная по этой формуле, называется *среднеквадратичным отклонением*.

Утверждается, что дисперсия является оптимальным значением среднеквадратичной ошибки. Покажем, что это действительно так.

Найдем значение y , для которого среднеквадратичное отклонение минимально. Для этого рассмотрим функцию:

$$\tilde{D}(y, N) = \frac{1}{N} \sum_{i=1}^N (y - \xi_i)^2.$$

Эта функция достигает минимума в точке, где производная $\tilde{D}(y, N)$ по y обращается в нуль, то есть, в точке, определяемой уравнением:

$$\sum_{i=1}^N (y - \xi_i) = 0.$$

Если раскрыть скобки в этом уравнении:

$$Ny - \sum_{i=1}^N \xi_i = 0.$$

Решение этого уравнения – среднее значение:

$$y = \frac{1}{N} \sum_{i=1}^N \xi_i.$$

Устремив $N \rightarrow \infty$, получим $y = \mu$. Утверждение доказано.

3.1.2 Равномерное распределение случайной величины

Самый простой и распространенный закон распределения случайной величины – равномерный. Считается, что именно по такому закону распределена случайная величина – индикатор орла и решки при подбрасывании монеты. С некоторыми допущениями можно считать, что по равномерно закону распределены погрешности в результатах показаний стрелочного прибора. Когда стрелка прибора оказывается между делениями, то с равной вероятностью записываются показания как добавляющие, так и уменьшающие значение на одно деление шкалы. В сущности, если не вдаваться в подробности, то и в цифровых приборах погрешность распределяется примерно так же.

- Пусть есть дискретная последовательность из N значений, которые может принимать случайная величина. Пронумеруем все значения от 1 до N . Говорят, что *дискретная случайная величина распределена равномерно*, если в выборке из M значений частота m_i выбора любого из N значений такова, что:

$$\lim_{M \rightarrow \infty} \frac{m_i}{M} = \frac{1}{N}.$$

То есть, дискретная случайная величина распределена равномерно, если она принимает значения с равными вероятностями. На практике это означает, что относительные частоты примерно одинаковы при достаточном большом числе наблюдений.

- Пусть случайная величина может принимать значения из непрерывного множества, например, из отрезка $\xi \in [a, b]$, тогда говорят, что *непрерывная случайная величина распределена равномерно*, если при неограниченном числе испытаний частота значений $\xi_i < x$ для $x \in [a, b]$ равна $(x - a)/(b - a)$.

3.1.3 Нормальное распределение

Обычно в измерении случайная погрешность складывается из большого количества неучтенных факторов, каждый из которых слабо влияет на результат измерения. Поэтому можно считать, что сама по себе величина случайной погрешности в i -том измерении есть сумма случайных величин:

$$\xi_i = \sum_j \xi_{ij}. \quad (3.1)$$

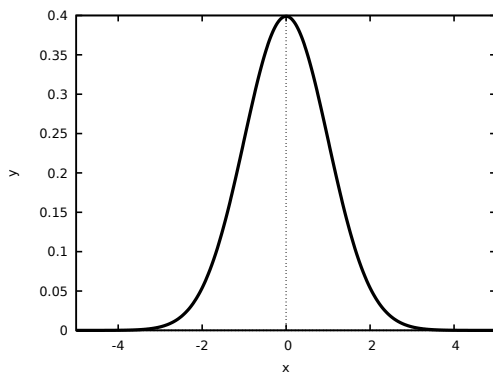


Рис. 3.1: Плотность распределения вероятностей случайной величины по нормальному закону при $\sigma = 1$ с нулевым средним.

Обычно предполагается, что каждая из этих случайных величин с равной вероятностью может принимать значения в некотором интервале около нуля.

Оказывается, что тогда для последовательности ξ_i верна так называемая *центральная предельная теорема*.

Утверждение 9 При сделанных предположениях об ξ_{ij} из формулы (3.1) плотность распределения вероятностей случайной величины ξ :

$$f(\xi) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(M(\xi)-\xi)^2}{2\sigma^2}}. \quad (3.2)$$

- Закон распределения вероятностей случайной величины по формуле (3.2) называется *нормальным распределением*.

Для того, чтобы получить вероятность того, что случайная величина окажется в диапазоне $a < \xi < b$, можно воспользоваться законом распределения:

$$P(a < \xi < b) = \int_a^b f(x)dx.$$

То есть – вычислить площадь между графиком кривой и осью O, x . Из вида графика можно предположить, что почти вся площадь такой кривой сосредоточена в окрестности нуля. Действительно, с вероятностью приблизительно 0,997 значение случайной величины ξ лежит в диапазоне $-3\sigma < \xi < 3\sigma$.

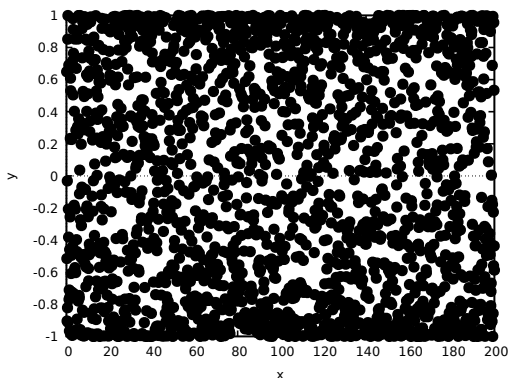


Рис. 3.2: Пример кластеризации данных в окрестностях максимумов и минимумов функции \sin при вычислении по 2000 значений, где аргумент – псевдослучайная величина на отрезке $[0, 2\pi)$.

3.1.4 Кластеризация данных

При измерении в разные моменты времени одной и той же величины и обработке результатов измерений важно знать, вызвано изменение показаний датчика случайными помехами или изменением самой измеряемой величины.

При кластеризации данные оказываются сгруппированными не около одного среднего значения, а вокруг нескольких различных значений. При этом, конечно, можно продолжать вычислять среднее значение всей выборки, однако в этом случае окажется, что среднее значение далеко от действительных значений показаний сенсора.

Пример 14 Рассмотрим типичный пример кластеризации данных.

Пусть датчик в дискретные моменты времени $i \in \{1, \dots, 2000\}$ измеряет величину $\sin(\xi_i)$, где псевдослучайная величина ξ_i распределена равномерно на отрезке $[0, 2\pi)$. В результате получится распределение, изображенное на рисунке 3.2. На графике видно, что результаты измерений группируются в окрестностях значений 1 и -1 . Эти значения соответствуют максимальным и минимальным значениям функции \sin .

Для определения количества кластеров в последовательной выборке одномерной случайной величины можно воспользоваться общим правилом.

Расположить значения ξ_i по возрастанию. Затем последовательно определять разности соседних величин $|\xi_i - \xi_{i+1}|$. Пусть максимальное значение разности достигается между некоторыми ξ_k и ξ_{k+1} и это существенно больше чем среднее значение таких разностей. Тогда можно принять, что величины до ξ_k и после ξ_{k+1} лежат в разных кластерах. Действуя таки образом, можно определить количество кластеров и их границы среди всей последовательности $\{\xi_i\}_{i=1}^N$.

3.1.5 Упражнения

Упражнение 33 По результатам 7-ми измерений получены следующие значения: 1, 2, 3, 2, 3, 1, 2. Вычислить среднее значение, определить дисперсию в этом наборе данных.

Упражнение 34 Параллельно два датчика снимают одни и те же показания. Результаты первого датчика -1, 2, -2, 3, 0, 1, 1, результаты измерений второго датчика 3, -2, 1, 0, 4, 2, 0. Погрешностью измерений назовем модуль разности между показаниями датчиков. Вычислить среднее и дисперсию погрешности измерений.

Упражнение 35 Заранее известно, что распределение случайной величины удовлетворяет нормальному закону. Дан ряд измерений

$$-2, 1, 2, 3, -7, 4, 5, 0.$$

Вычислить среднее значение, дисперсию и определить диапазон значений случайной величины с вероятностью 0,997.

Упражнение 36 Дан ряд измерений

$$0, 1, 1, 2, 3, 3, 4, 3, 3, 4, 0, 0, 3, 0, 1, 1, 1, 4, 4, 5, 4, 5.$$

Определить кластеры для случайной величины и определить среднее значение и дисперсию в кластерах и, отдельно, в предположении, что случайная величина не кластеризована, определить среднее значение и дисперсию.

3.2 Виды фильтров данных.

3.2.1 Заранее определенный интервал

Во многих случаях заранее известны диапазоны значений, которые будут приходить от сенсоров. Поэтому значения вне таких интервалов можно просто отбрасывать и считать их недостоверными.

В результате действия такого фильтра окажется, что данные, признанные достоверными, лежат внутри некоторого интервала (A, B) :

$$A < \xi_k < B$$

Примеры шумов, которые можно отфильтровать таким способом – засветка от фотовспышки датчика освещенности или шум от громкоговорящего, на который может реагировать ультразвуковой датчик.

3.2.2 Квазистационарные данные. Повторный опрос

Простой способ не реагировать на шумы – повторно опрашивать датчик и считать достоверными два последовательных измерения с близкими значениями. Какие измерения считать близкими зависит от скорости движения, от частоты измерений и вида типичных шумов. На практике можно принять, что если значения измерений отличаются на 5%-10%, то такие измерения близки.

В результате оказывается, что из последовательности измерений $\{\xi_i\}_{i=1}^N$ достоверными признаются измерения ξ_k для которых выполнено неравенство:

$$0 \leq |\xi_{k+1} - \xi_k| \leq \kappa |\xi_k|,$$

κ – коэффициент фильтра.

3.2.3 Скользящее среднее

Один из стандартных фильтров для периодических шумов – усреднение. Нужно взять несколько последовательных измерений, вычислить среднее значение, просуммировав величины измерений и разделив результат на число измерений.

$$X_k = \frac{1}{Z} \sum_{i=k}^{k+Z} \xi_i,$$

здесь Z – параметр усреднения.

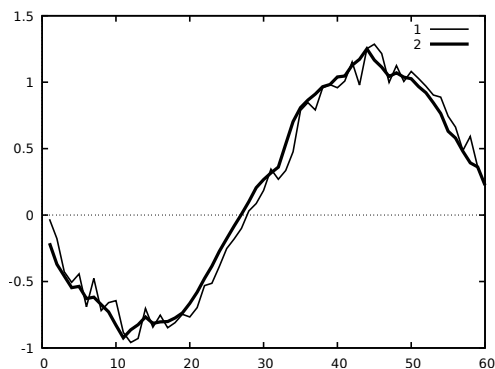


Рис. 3.3: Пример действия усреднения для трех последовательных измерений сильно зашумленных данных: сумма функции синус и псевдослучайной величины, меняющейся от 0 до 0.3. «1» — данные с шумом, «2» — результат усреднения.

3.2.4 Доверительный интервал

Основываясь на величине скользящего среднего можно вычислить скользящее значение среднеквадратичного отклонения:

$$\sigma_k = \sqrt{\frac{1}{Z} \sum_{i=k}^{k+Z} (X_k - \xi_i)^2}.$$

В предположении о нормальном распределении измеряемой величины, она с вероятностью примерно 0.997 должна лежать в интервале:

$$X_k - 3\sigma_k < \xi_i < X_k + 3\sigma_k.$$

Тогда значения вне этого интервала можно отбрасывать и считать их недопустимыми.

3.2.5 Упражнения

Упражнение 37 При измерениях получен ряд данных:

$$1, 2, 3, 2, 3, 1, 3, 4, 5, 3, 2, -1, 0.$$

Построить ряд после усреднения по правилу скользящего среднего из трех соседних значений.

Упражнение 38 При измерениях получен ряд данных:

$$1, 2, 3, 2, 3, 1, 3, 4, 5, 3, 2, -1, 0.$$

Построить ряд после фильтрации с помощью доверительного интервала, при условии вычисления параметров нормального распределения по трем предыдущим измерениям.

3.3 Калибровка и устранение систематической ошибки

Систематические ошибки, как правило, возникают при использовании слишком грубой или даже иногда ошибочной математической модели. Тем не менее в науке существует надежный способ борьбы с систематическими ошибками. Этот способ – переход от построения фундаментальной модели, основанной на моделировании механизмов явления, к построению феноменологической математической модели, которая основывается исключительно на сопоставлении наблюдений и выводе из них какой-либо формальной зависимости. В робототехнике такой переход происходит при калибровке датчиков.

3.3.1 Сдвиг

Для устранения систематической ошибки, постоянной или почти постоянной на всем интервале измерений обычно используется сдвиг системы отсчета.

Пусть заранее известно значение некоторой величины. Это значение равно X_0 . Однако после обработки измерений с помощью фильтров для устранения случайных ошибок измерений вместо ожидаемой величины X_0 получается величина X_1 . Это означает, что в ряду измерений присутствует систематическая ошибка. Такую ошибку можно устранить сдвигом начала отсчета на величину систематической ошибки ($X_1 - X_0$).

Пусть в результате измерения получена величина X , тогда, применив сдвиг на систематическую ошибку, получим скорректированное значение измеряемой величины:

$$x = X - (X_1 - X_0).$$

Такой сдвиг обычно производится именно для установки начала отсчета в шкале измерений.

3.3.2 Интерполяция

Обычно установка нуля не приводит к устранению систематической ошибки во всем диапазоне измеряемой величины. Часто бывает так, что систематическая ошибка проявляется не только в общем сдвиге, но и в изменении самой шкалы для измеряемой величины. Изменение шкалы измерений с математической точки зрения эквивалентно нормировке, то есть выяснению области значений измерений и приведению шкалы к правильному виду, используя контрольные измерения в нескольких заранее известных точках. Для определения шкалы между измеренными контрольными точками обычно пользуются интерполяцией. В самом простом случае это линейная интерполяция, в более сложных случаях – интерполяция многочленами или сплайнами.

- *Интерполяцией* называют способ получения значений некоторой величины по другой величине (аргументу) между заранее известными дискретными значениями этой величины. Например, но не обязательно, функциональной зависимости от аргумента по нескольким заданным точкам.

Для линейной интерполяции бывает достаточно двух контрольных точек i и $i+k$. Пусть в контрольных точках i и $i+k$ после нескольких измерений и фильтрации случайных погрешностей получены величины X_i и X_{i+k} . Однако известны точные значения измеряемых величин в этих контрольных точках x_i и x_{i+k} .

Если считать, что измеряемая величина линейно зависит от значения, то получим отображение из пространства измерений X в пространство значений измеряемой величины x :

$$x = \frac{x_{i+k} - x_i}{X_{i+k} - X_i}(X - X_i) + x_i.$$

Это типичная линейная интерполяция.

Если требуется провести кривую через три контрольных точки, то можно либо аппроксимировать полученные данные ломаной, либо построить параболу, проходящую через заданные точки:

$$x = a_2 X^2 + a_1 X + a_0.$$

Здесь X – значение измеряемой величины, полученное после устранения случайной ошибки. Три параметра параболы $a_i, i = 1, 2, 3$ как раз и можно определить по трем контрольным значениям X_i .

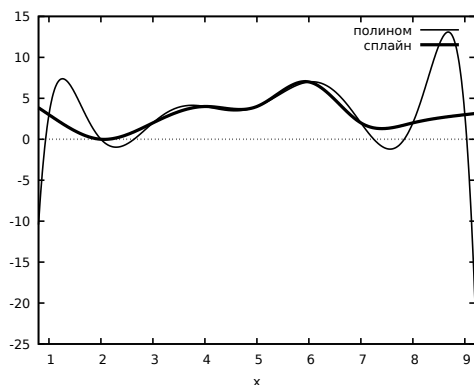


Рис. 3.4: Интерполяция по 9-ти точкам: (1, 3), (2, 0), (3, 2), (4, 4), (5, 4), (6, 7), (7, 2), (8, 2), (9, 3). Интерполяционный полином 8-й степени на краях интервала сильно отклоняется. При выходе за границы интервала интерполяции – при $x < 1$ и $9 < x$ интерполяционный полином быстро меняется. Сплайн третьей степени равномерно ведет себя на всем интервале и вблизи его границ.

Если контрольных значений N , то можно использовать интерполяцию многочленом $N - 1$ степени. Однако многочлены высоких степеней сильно осциллируют вне заданных значений. Поэтому вместо интерполяции многочленами высоких степеней применяют интерполяцию сплайнами. Пример приведен на рисунке 3.4.

В самом простом случае интерполяция сплайнами первой степени – это интерполяция ломаной линией, проходящей через все контрольные точки.

Сплайн второй степени – это сопряжение данных с помощью последовательности парабол. Каждая парабола оказывается пригодна для интерполяции между двумя контрольными точками. В результате у каждой такой параболы оказывается один свободный параметр. Обычно такой параметр используется для гладкого, без изломов, сопряжения соседних парабол. Для этого, кроме условия прохождения через две заданные точки, ставится еще одно условие – равенства производной следующей параболы в точке сопряжения с предыдущей параболой.

Широко распространены сплайны третьей степени. Сплайн третьей степени получится, если через заданные точки проложить, например, упругую линейку. В точках сопряжения таких сплайнов совпадают и первые, и вторые производные.

Пример 15 Пусть имеется ряд пар: значение датчика при калибровке и истинное значение измеряемой величины. Построить полином минимальной степени, проходящий через точки: $(1,0), (2,3), (3,4), (4,6)$.

Дано четыре пары значений. Следовательно, интерполяционный полином должен быть третьего порядка $F(\xi) \rightarrow x$ такой, что $F(1) = 0$, $F(2) = 3$, $F(3) = 4$, $F(4) = 6$. Это можно сделать, воспользовавшись интерполяционным полиномом Лагранжа:

$$F(\xi) = a_1(\xi - 2)(\xi - 3)(\xi - 4) + a_2(\xi - 1)(\xi - 3)(\xi - 4) + \\ a_3(\xi - 1)(\xi - 2)(\xi - 4) + a_4(\xi - 1)(\xi - 2)(\xi - 3);$$

Здесь i -е слагаемое подобрано так, что оно равно нулю во всех точках интерполяции, кроме точки i . Коэффициенты подбираются из значения функции в точке ξ_i :

$$F(1) = a_1(-1)(-2)(-3) = 0, \quad a_1 = 0;$$

$$F(2) = a_2(1)(-1)(-2) = 3, \quad a_2 = 3/2;$$

$$F(3) = a_3(2)(1)(-1) = 4, \quad a_3 = -2;$$

$$F(4) = a_4(3)(2)(1) = 6, \quad a_4 = 1.$$

В результате:

$$F(\xi) = \frac{3}{2}(\xi - 1)(\xi - 3)(\xi - 4) - 2(\xi - 1)(\xi - 2)(\xi - 4) + (\xi - 1)(\xi - 2)(\xi - 3);$$

Если раскрыть скобки, то получим:

$$F(\xi) = \frac{1}{2}\xi^3 - 4\xi^2 + \frac{23}{2}\xi - 8.$$

Если после применения фильтра от случайных ошибок показание датчика $\xi = 1.5$, то значение измеряемой величины с учетом систематической погрешности с помощью проведенной калибровки: $F(1.5) = 0.375$.

Пример 16 Провести сплайн второго порядка через те же четыре точки с заданными координатами по калибровочной последовательности $(1,0), (2,3), (3,4), (4,6)$.

Сплайн второго порядка состоит из суммы парабол, которые используются на ограниченных промежутках. Параболу второго порядка можно провести через три точки. Таким образом, одна точка – последняя останется не задействована. Через нее и через предыдущую точку проведем второй многочлен сплайна, производная которого в точке ξ_3 должна совпадать с производной первой функции сплайна в этой же точке.

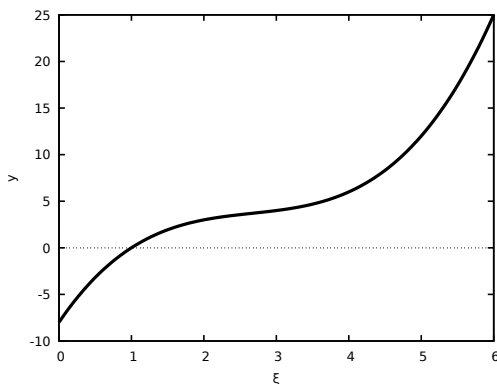


Рис. 3.5: Интерполяционный полином по калибровочной последовательности $(1, 0)$, $(2, 3)$, $(3, 4)$, $(4, 6)$.

Через первые три точки проводим интерполяционный многочлен Лагранжа второй степени:

$$f_1(\xi) = -3(\xi - 1)(\xi - 3) + 2(\xi - 1)(\xi - 2);$$

Если раскрыть скобки:

$$f_1(\xi) = -\xi^2 + 6\xi - 5;$$

Вторая функция сплайна:

$$f_2(\xi) = a_2\xi^2 + a_1\xi + a_0.$$

Производные по ξ :

$$f'_1 = -2\xi + 6;$$

$$f'_2 = 2a_2\xi + a_1.$$

Условия для определения коэффициентов второй функции сплайна:

$$f_2(4) = a_2 \cdot 16 + a_1 \cdot 4 + a_0 = 6,$$

$$f_2(3) = a_2 \cdot 9 + a_1 \cdot 3 + a_0 = 4,$$

$$f'_2(3) = f'_1(3), \quad \text{или} \quad 2a_2 \cdot 3 + a_1 = -2 \cdot 3 + 6.$$

Тогда:

$$f_2(\xi) = 2\xi^2 - 12\xi + 22;$$

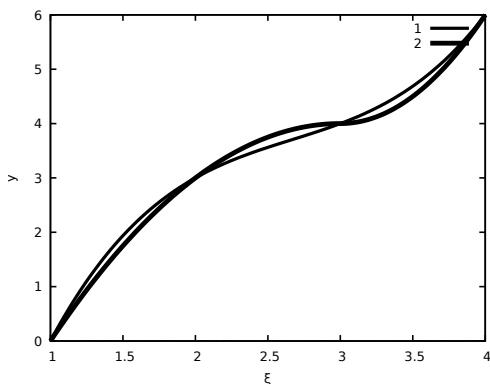


Рис. 3.6: Результаты интерполяции калибровочных данных «1» –полиномом третьей степени, «2» – сплайном второй степени.

В результате интерполяционный сплайн:

$$f(\xi) = \begin{cases} -\xi^2 + 6\xi - 5, & x \in [0, 3); \\ 2\xi^2 - 12\xi + 22, & x \in [3, 6]. \end{cases}$$

3.3.3 Упражнения

Упражнение 39 При калибровке получены пары данных (измеренное значение, истинное значение): (1, 2), (2, 3), (4, 5), (7, 6). Построить интерполяционный многочлен третьей степени.

Упражнение 40 При калибровке получены пары данных (измеренное значение, истинное значение): (1, 2), (2, 3), (4, 5), (7, 6). Интерполировать данные сплайнами первой степени.

Упражнение 41 При калибровке получены пары данных (измеренное значение, истинное значение): (1, 2), (2, 3), (4, 5), (7, 6). Интерполировать данные сплайнами второй степени.

3.4 Обзор литературы

С основами теории теории вероятностей и математической статистики можно познакомиться, например, по учебникам [12], [13], [11].

Основы теории интерполяции изложены, например, в учебнике [14].

Глава 4

Обработка изображений. Аффинные преобразования и свертки.

Передача цвета основана на векторном представлении. Цвет отдельного пикселя определяется любым из следующих способов:

- трехмерным вектором в пространстве RGB;
- трехмерным вектором в пространстве HLS;
- четырехмерным вектором в пространстве CMYK.

Отображения для перехода из одного представления в другое являются преобразованиями между системами координат.

Растровое изображение можно представить как двумерную матрицу.

- Каждый элемент этой матрицы – вектор, определяющий цвет пикселя матрицы.
- Изменение изображения – это преобразование над векторами – элементами матрицы.
- Линейные преобразования над элементами матрицы изображения – это операторы свертки в дискретном конечномерном пространстве.
- Повороты и растяжения изображения можно рассматривать как аффинные преобразования.

4.1 Геометрия цветовых пространств

4.1.1 Цветовое пространство RGB

Пространство RGB формируется излучением красного, зеленого и синего цветов. Здесь не обсуждается как физически формируется это излучение. Будем считать, что красный цвет это излучение близкое к длинному концу спектра видимого света, синий цвет близок к короткому концу видимого спектра, зеленый – к середине.

Часть цветов из набора, воспринимаемого человеком, можно напрямую сопоставить электромагнитному излучению определенной длины волны. Цвета из такого набора принято называть чистыми цветами. Однако набор чистых цветов не составляет все воспринимаемые цвета. Причина в том, что пространство наполнено электромагнитными волнами совершенно разных частот и часть из полного набора цветов является составной. Самый простой и известный пример – белый цвет. Белый состоит из всех частот видимого спектра. Обратно, черный цвет совсем не содержит электромагнитных колебаний из видимого спектра.

Красный, зеленый и синий оказались не только удобными, но, скажем так, базисными цветами, видимо еще и потому, что цветовое зрение человека основано на трех видах колбочек, максимумы чувствительности которых приходятся как раз на красный, зеленый и синий цвета.

По этой причине принятое цветовое пространство трехмерно. С формальной стороны цвет можно воспринимать как трехмерный вектор в трехмерном пространстве. С помощью вектора из трех величин – составляющей красного (R), зеленого (G) и синего – (B) можно кодировать большую часть цветов, воспринимаемых человеком. В компонентах вектора в пространстве *RGB* нет отрицательных значений. Кроме того, так как это пространство используется в цифровой технике, принята дискретная градация для его составляющих, например, от 0 до 255. Примеры векторов в пространстве *RGB*: Базисными в RGB считаются векторы

- $(1, 0, 0)$ – красный,
- $(0, 1, 0)$ – зеленый,
- $(0, 0, 1)$ – синий.

Пространство RGB можно воспринимать как первый октант из пространства \mathbb{R}^3 – в котором все компоненты вектора могут быть только положительные. Ограничение на положительное значение компонент вектора приводит к тому, что с формальной точки зрения RGB не является линейным пространством.

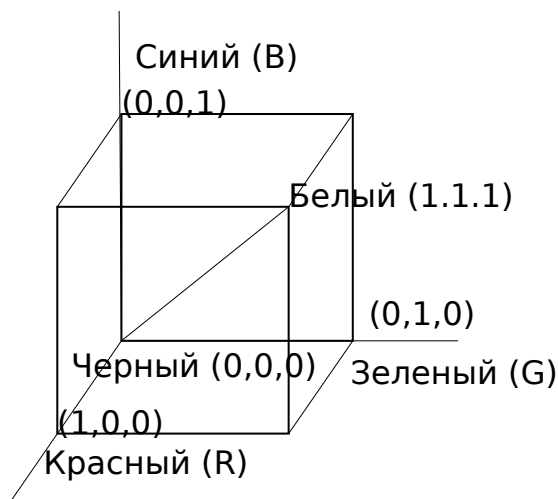


Рис. 4.1: Условное изображение векторного пространства RGB и цветовых векторов в нем.

В пространстве RGB определена обычная для векторных пространств операция умножения на число и операция сложения. Сумма компонент двух векторов есть опять вектор из пространства RGB. Это пространство называется аддитивным. Все цвета из этого пространства можно получить умножением базисных векторов на число и сложением.

Белый цвет в пространстве RGB имеет координаты $(1, 1, 1)$, черный – $(0, 0, 0)$.

4.1.2 Цветовое пространство СМΥΚ

Другой способ получения цвета используется в полиграфии. Изначально бумага белая. Вектор белого цвета в RGB $(1, 1, 1)$. Если из белого цвета извлечь компоненту синего, то получится $(1, 1, 0)$ – это желтый цвет в представлении RGB. Если из представления белого цвета извлечь компоненту зеленого цвета, то получится $(1, 0, 1)$ – пурпурный, а если из белого извлечь красный – получится цвет $(0, 1, 1)$ – бирюзовый. Эти цвета составляют базис в пространстве СМΥΚ:

- $(1, 1, 0)$ – желтый,
- $(1, 0, 1)$ – пурпурный,

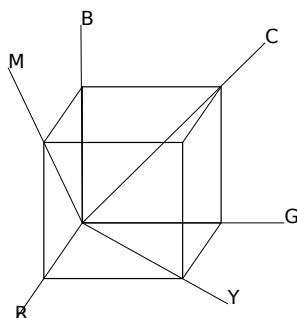


Рис. 4.2: Куб RGB и координаты векторов пространства CMY. Бирюзовый (C (0,1,1) –Cyan), желтый (Y (1,1,0) – Yellow), пурпурный (M (1,1,0)–Magenta).

- (0, 1, 1) – бирюзовый.

В таком подходе черный цвет получается, если смешать базисные цвета из CMY. Цвета из пространства CMY используются в полиграфии для определения спектра поглощения. Например, желтый излучает желтый и поглощает все остальные цвета. Пурпурный – излучает только пурпурный. Бирюзовый пигмент излучает только бирюзовый.

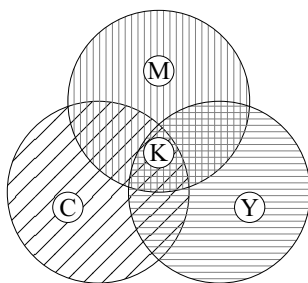


Рис. 4.3: На рисунке обозначены условно: С – бирюзовый, М – пурпурный, Y – желтый, К– черный. Смешение базовых цветов не дает истинно черный цвет. Поэтому к базовым цветам пространства CMY добавляют черный цвет – К.

В результате смешения всех трех пигментов краски получится черный цвет, рисунок 4.3 (в RGB наличие всех цветов даст белый). Однако в поли-

графии прямым смешением трудно получить истинно черный цвет. Поэтому к полному, с математической точки зрения, базису добавляется черный, он обозначается символом К. Такое расширенное пространство называется СМУК.

4.1.3 Цветовое пространство YUV

В представлении цвета важная величина – яркость. Технически ее важно выделять, например, для определения подсветки. Яркость определяется суммой всех трех компонент.

Модель YUV выделяет отдельно яркость (Y) и две цветовые компоненты (U,V), опирающиеся на крайние в видимом спектре красную и синюю составляющие.

Человеческим глазом наиболее яркими воспринимаются цвета из середины спектра. Поэтому величина Y вычисляется по формуле, содержащей специально подобранные веса при каждой из компонент вектора RGB:

$$Y = K_R r + (1 - K_R - K_B)g + K_B b.$$

Коэффициенты K_R, K_B могут иметь разное значение в разных технических вариантах реализации этой модели. Как правило коэффициенты лежат в интервалах: $0.2 < K_R < 0.3$, $0.05 < K_B < 0.073$.

Для цветовых составляющих U и V используются синяя и красная составляющие, но также включается и зеленая составляющая спектра, в том виде, в котором она есть в величине Y – яркости:

$$U = b - Y, \quad V = r - Y.$$

Переход от RGB к YUV является линейным преобразованием:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} K_R & (1 - K_R - K_B) & K_B \\ -K_R & (K_R + K_B - 1) & (1 - K_B) \\ (1 - K_R) & (K_R + K_B - 1) & -K_B \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix} \quad (4.1)$$

Обратное преобразование имеет вид:

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & \frac{K_B}{K_R + K_B - 1} & \frac{K_R}{K_R + K_B - 1} \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} Y \\ U \\ V \end{pmatrix}$$

4.1.4 Норма в цветовом пространстве

В геометрии важно понятие расстояния. В цветовых пространствах также бывает важно сортировать объекты по цвету. Для этого нужно ввести понятие расстояния – нормы. Есть разные способы введения понятия нормы в векторных пространствах. Известно, что в конечномерных пространствах все нормы эквивалентны. Для пространства RGB нормой вектора $\vec{\alpha} = (r, g, b)$ будем называть сумму его компонент:

$$||\vec{\alpha}|| = r + g + b.$$

Расстояние между двумя точками пространства можно рассматривать как норму вектора разности $\vec{\alpha}_0 = (r_0, g_0, b_0)$ и $\vec{\alpha}_1 = (r_1, g_1, b_1)$:

$$||\vec{\alpha}_0 - \vec{\alpha}_1|| = |r_0 - r_1| + |g_0 - g_1| + |b_0 - b_1|.$$

Упражнение 42 Показать, что введенное понятие нормы удовлетворяет неравенству треугольника:

$$||\vec{\alpha}_0 - \vec{\alpha}_1|| + ||\vec{\alpha}_1 - \vec{\alpha}_2|| \geq ||\vec{\alpha}_0 - \vec{\alpha}_2||.$$

Если пользоваться этим определением нормы для классификации объектов по цвету, то могут возникнуть трудности.

Пример 17 Векторы, определяющие один и тот же красный цвет, $\vec{\alpha}_0 = (1, 0, 0)$ и $\vec{\alpha}_1 = (5, 0, 0)$ отстоят друг от друга на 4 единицы. Хотя на самом деле это объекты красного цвета, но разной интенсивности.

Пример 18 Если к вектору, определяющему красный цвет $\vec{\alpha}_0 = (1, 0, 0)$ добавить вектор, определяющий белый цвет: $\vec{\alpha}_1 = (3, 3, 3)$, тогда получится вектор с координатами:

$$\vec{\alpha}_2 = \vec{\alpha}_0 + \vec{\alpha}_1 = (4, 3, 3)$$

Этот вектор опять будет определять красный цвет, добавление вектора $\vec{\alpha}_1$ соответствует изменению освещения – дополнительной засветке. Однако, расстояние, определенное выше покажет, что векторы не совпадают.

Приведенные примеры показывают, что различие координат объектов в пространстве RGB не позволяет классифицировать их как объекты разного цвета. Для классификации объектов по цвету необходима дополнительная обработка координат в пространстве RGB. То же самое справедливо и для координат объекта в дополнительном пространстве СМΥΚ.

4.1.5 Перцептивные цветовые пространства HSV

Рассмотренные примеры 17 и 18 показывают, что для определения истинного цвета необходимо удалить засветку – белую составляющую в векторе цвета и привести полученный вектор к эталонной, обычно единичной, длине. После чего можно определять оттенок цвета этого нового вектора.

Представление цвета, в котором в качестве координат используются яркость (Lightness или Value), насыщенность (Saturation) и оттенок цвета (Hue) обозначается HSV.

Компонента яркости определяется просто – это максимальное из значений компонент вектора RGB:

$$V = \max\{r, g, b\}.$$

Под насыщенностью понимается максимальное значение компоненты вектора RGB после удаления засветки. Засветка – белая составляющая в векторе RGB. Величина засветки – это минимальное значение компоненты в векторе RGB:

$$m = \min\{r, g, b\}.$$

Насыщенность принято измерять в пределах $[0, 1]$, она определяется как отношение разности максимального значения и минимального значения компонент RGB к максимальному значению.

$$S = \frac{M - m}{M}.$$

Третий параметр вектора в пространстве HSV – оттенок цвета. Он определяется после удаления засветки из компонент вектора в пространстве RGB:

$$\alpha' = (r - m, g - m, b - m).$$

В результате одна, две или все три компоненты вектора α' равны нулю. Рассмотрим все эти случаи отдельно.

- 1 Все компоненты α' нули. Значит в исходном векторе все компоненты были равны друг другу. Цвет исходного вектора:

- если $m = 0$ – черный;
- если $m > 0$ – белый.

- 2 Два компонента вектора α' равны нулю. Следовательно цвет исходного вектора:

- если $r - m > 0$ – красный;

- если $g - m > 0$ – зеленый;
- если $b - m > 0$ – синий.

3 Один компонент вектора α' равен нулю. Следовательно цвет исходного вектора:

- если $b = 0$ – цвет между красным и зеленым;
- если $r = 0$ – цвет между зеленым и синим;
- если $g = 0$ – цвет между синим и красным.

Здесь нужно подробно рассмотреть случай 3. Формально можно заключить, что оттенки цвета меняются по кругу. То есть параметр H лежит на окружности. Координаты на окружности принято измерять либо в градусах, либо в радианах.

Величину H принято измерять в градусах. Угловая величина окружности 360° . За нуль градусов принято направление вектора с координатами $(1, 0, 0)$. направление $(0, 1, 0)$ соответствует углу 120° и направление $(0, 0, 1)$ будем считать 240° . Формула для вычисления H кусочно линейна:

$$r = M \text{ тогда } H = 60^\circ \frac{g-b}{M-m}, \quad \text{mod } 360^\circ;$$

$$g = M \text{ тогда } H = 60^\circ \left(\frac{b-r}{M-m} + 2 \right), \quad \text{mod } 360^\circ;$$

$$b = M \text{ тогда } H = 60^\circ \left(\frac{r-g}{M-m} + 4 \right), \quad \text{mod } 360^\circ.$$

Так компоненты вектора RGB можно перевести в компоненты вектора HSV.

4.1.6 Топология пространства HSV

Пространство HSV представляет собой прямое произведение окружности и двух отрезков $\mathbb{S}^1 \times [0, 1] \times [0, A]$. Здесь $A = 3D$ – максимальное значение параметра V – число D градаций в шкале RGB умноженное на три. Число D градаций в шкале обычно называется глубиной цвета. Стандартные значения глубины цвета: 2, 4, 8, 16.

Параметры пространства HSV удобно использовать в распознавании цветов, градации серого и интенсивности освещения.

Если рассматривать расстояние как различие в тоне цвета, то достаточно взять минимальную по угловой величине из двух дуг окружности между цветовыми оттенками объектов.

Если необходимо определить градации серого, то нужно вычислить разность между значениями насыщенности.

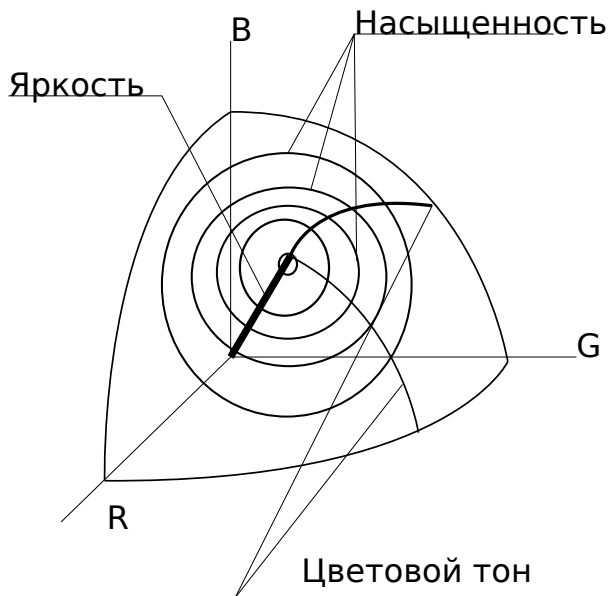


Рис. 4.4: Представление координат HSV в пространстве RGB. Радиус сегмента сферы соответствует яркости (Value). Точка пересечения сегмента сферы с вектором белого цвета – направление белого цвета (1,1,1) – точка нулевой насыщенности (Saturation). Концентрические окружности соответствуют разной насыщенности цвета. Кривые на сфере ортогональные окружностям – разным цветовым оттенкам (Hue).

Для определения различия в яркости нужно вычислить разность параметров V и разделить ее на 3.

Таким образом каждую из компонент вектора в пространстве HSV можно использовать для определения метрики.

4.1.7 Геометрическое представление

Рассмотрим RGB как куб в трехмерном пространстве со сторонами, длина которых равна максимальному значению глубины цвета D . Начало координат отвечает черному цвету. Точки на диагонали куба, выходящей из точки $(0, 0, 0)$, соответствуют белому цвету разной интенсивности.

Переход к модели HSV соответствует, в некотором смысле, переходу к сферической системе координат.

При выделении яркости происходит проектирование всех точек куба на сегмент сферы – восьмую часть сферы, лежащую в первом октанте трехмерного пространства. На этом сегменте сферы, в центре, лежит точка, соответствующая белому цвету.

На следующем шаге сегмент сферы растягивается в круг и в этом круге устраивается подобие полярной системы координат. Насыщенность S соответствует расстоянию от центра круга, а цветовой тон – полярному углу.

4.1.8 Упражнения

Упражнение 43 Координаты вектора в пространстве RGB $(3, 2, 1)$. Выделить засветку и получить цветовой вектор без засветки.

Упражнение 44 Координаты вектора в пространстве RGB $(2, 4, 2)$. Преобразовать вектор в координаты CMY.

Упражнение 45 Координаты вектора в пространстве RGB $(2, 4, 3)$. Преобразовать координаты вектора координаты пространства HSV.

Упражнение 46 Координаты вектора в пространстве YUV $(2, 4, 2)$. Преобразовать вектор в координаты RGB.

4.2 Поиск информации на изображении

4.2.1 Растровые изображения

Простой и распространенный способ представления графической информации, картинок, – растровый. Изображение в таком формате есть матрица в N строк и M столбцов. Элементами матрицы – пикселями обычно являются векторы в одном из рассмотренных выше форматов RGB, YUV или CMYK.

При получении изображения возможно как проявление шумов матрицы камеры, так и других случайных помех на изображении. Для удаления шумов с изображения можно пользоваться методами, рассмотренными в разделе 3, при этом важно учитывать двумерный характер обрабатываемой величины.

Основное отличие при обработке изображений состоит в том, что обрабатываемые объекты, как правило, имеют некоторую форму или контуры. Более того, часто задача состоит в том, чтобы выделить на изображении некоторый объект, например, штрих-код или QR-код. Штрих-код и QR-код – удобные примеры для обсуждения основ обработки изображений. Прежде всего это черно-белые объекты. Для черно-белых объектов обычно используется двоичное представление цвета. Например, если значение элемента матрицы 0, тогда цвет пикселя черный, если значение 1, тогда цвет этого пикселя белый.

Для простоты здесь разбирается обработка двухцветного изображения, для определенности любое двухцветное изображение будет называться черно-белым.

Пример 19 *Рассмотрим в качестве примера обработку картинку, содержащей штрих-код. Будем полагать, что картинка представляет собой растровое изображение в формате RGB. Обычно это результат фотографии смартфоном в условиях плохого освещения и нетвердой рукой. То есть изображение размыто и сильно зашумлено. Рассмотрим, какие шаги можно сделать для определения штрих-кода.*

1. *Привести картинку к черно-белому виду. Самый простой шаг здесь – пороговое преобразование. Однако, из-за неравномерности освещения простое пороговое преобразование может ухудшить изображение. Пороговое преобразование должно быть адаптивным – величина порога должна меняться в зависимости от яркости обрабатываемой области.*
2. *Выделить угловые точки четырехугольника, в котором располагается штрих-код и определить их координаты в изображении.*

3. Преобразовать изображение так, чтобы четырехугольник штрих-кода был преобразован в прямоугольник.
4. Убрать шумы в области штрих-кода, так чтобы четко прорисовались вертикальные линии.
5. По набору линий определить код объекта.

На этом примере можно разобрать основные виды обработки изображения.

Для простоты ниже рассматривается картинка с L градациями серого. Будем считать, что L – яркость пикселя.

4.2.2 Линейные фильтры

С математической точки зрения фильтры – это отображения одного множества в другое. Как правило, отображение фильтрации необратимо. То есть из результата не удастся получить исходное изображение.

Один из наиболее широко применяемых линейных фильтров – *фильтр скользящего среднего*.

Пусть матрица изображения имеет размер $N \times M$ пикселей, то есть N строк и M столбцов. Для вычисления среднего значения в окрестности элемента матрицы выберем квадрат со стороной $2n + 1 < \min\{M, N\}$, так что в центре такого квадрата всегда будет находиться пиксель, для которого вычисляется значение с помощью фильтра.

Формула для среднего значения:

$$L_{i,j} = \frac{1}{(2n+1)^2} \sum_{k=i-n}^{i+n} \sum_{l=j-n}^{j+n} L_{k,l},$$

где $L_{k,l} = (r + g + b)/3$ – яркость пикселя.

- Это выражение можно записать в виде *оператора свертки*:

$$L_{i+n,j+n} = \sum_{k,l} h_{k,l} L_{i-(n-k),j-(n-l)}.$$

- Здесь *ядро оператора свертки* h – матрица размером $(2n+1) \times (2n+1)$ с одинаковыми элементами

$$h_{k,l} \equiv \frac{1}{(2n+1)^2}.$$

- Термин «свертка» в математике появляется вполне естественно. Например, если рассмотреть произведение двух полиномов

$$P_n(x) = p_n x^n + p_{n-1} x^{n-1} + \dots + p_1 x + p_0$$

$$Q_m(x) = q_m x^m + p_{m-1} x^{m-1} + \dots + q_1 x + q_0$$

Получится длинное выражение. Если его опять переписать (свернуть) в виде полинома, теперь уже степени $l = n + m$,

$$R_l = r_l x^l + r_{l-1} x^{l-1} + \dots + r_1 x + r_0,$$

то коэффициенты этого нового полинома можно получить по формуле:

$$r_k = \sum_{i=0}^k p_i q_{k-i}.$$

Такая сумма и называется сверткой.

- Если теперь зафиксировать, например, полином $Q_m(x)$ и умножать на него другие полиномы, то окажется, что во всех формулах для коэффициентов получающихся полиномов присутствует сумма с коэффициентами q_{k-i} . Теперь можно пойти дальше и вместо полиномов рассматривать векторы, составленные из их коэффициентов, например, для полинома $P_n(x)$ это вектор $\vec{p} = (p_n, p_{n-1}, \dots, p_1, p_0)$. Тогда умножение полиномов на полином $Q_m(x)$ можно рассматривать как действие оператора свертки \mathcal{Q} на пространстве векторов:

$$\mathcal{Q}(\vec{p}) = \vec{r},$$

где компоненты вектора r получаются по формуле:

$$r_k = \sum_{i=0}^k p_i q_{k-i}.$$

В этом случае q_{k-i} является ядром оператора свертки.

- В этом примере мономы x^k для разных значений k используются как базисные векторы для представления функций – полиномов. В математике рассматривают представление функций, используя и другие наборы базисных векторов, например, – тригонометрические функции. Общий подход при сворачивании произведений таких представлений оказывается похожим – в них участвуют операторы свертки.



Рис. 4.5: Пример действия фильтра скользящего среднего. Слева исходный файл, справа – после фильтра. Для примера использовалась программа GIMP.

Фильтр скользящего среднего хорошо убирает шумы, но такая фильтрация приводит к тому, что размываются контуры объекта.

Для матриц h , используемых при фильтрации, обычны значения $n = 1, 2$, то есть, это матрицы размером 3×3 или 5×5 .

- Важное свойство ядра оператора свертки – сумма его компонент равна единице. Такой оператор не увеличивает общую яркость после обработки изображения.

Пример 20 *Пример действия фильтра скользящего среднего на фотографию штрих-кода. Построим матрицу фильтра, или другими словами – ядро оператора свертки для скользящего среднего. Для простоты выберем размер ядра оператора усреднения небольшой, например, 3×3 . Тогда усреднение будет проводиться по 9-ти соседним элементам, а результат усреднения будет записан в новую матрицу, в элемент, соответствующий среднему элементу фильтра. Значения элементов ядра при усреднении по 9-ти элементам $h = 1/9 \sim 0.1111$. Матрица фильтра:*

$$h = \begin{pmatrix} 0.111 & 0.111 & 0.111 \\ 0.111 & 0.111 & 0.111 \\ 0.111 & 0.111 & 0.111 \end{pmatrix}.$$

Яркость картинке почти не увеличилась, так как сумма компонент ядра оператора 0.999 близка к единице. Сам штрих-код размазлся, как и полается при усреднении.

С помощью подобного рода фильтров можно не только размывать контуры, но и увеличивать градиенты изменения яркости, то есть выявлять контуры объектов.



Рис. 4.6: Пример действия сверточного фильтра для усиления вертикальных градиентов. Слева исходный файл, справа – после фильтра. Для примера использовалась программа GIMP.



Рис. 4.7: Пример действия сверточного фильтра для усиления горизонтальных градиентов. Слева исходный файл, справа – после фильтра. Для примера использовалась программа GIMP.

Пример 21 Например, если в качестве ядра оператора свертки взять матрицу вида

$$f = \begin{pmatrix} -1 & 2 & -1 \\ -1 & 3 & -1 \\ -1 & 2 & -1 \end{pmatrix}$$

тогда в результате свертки

$$L_{i+n,j+n} = \sum_{k,l} f_{k,l} L_{i-(n-k),j-(n-l)}.$$

произойдет усиление вертикальных градиентов.

Пример 22 Для усиления горизонтальных градиентов можно использовать ядро свертки:

$$g = \begin{pmatrix} -1 & -1 & -1 \\ 2 & 3 & 2 \\ -1 & -1 & -1 \end{pmatrix}$$

Можно воспользоваться оператором свертки для того, чтобы находить контуры объектов.

При переходе от одного объекта на изображении к другому меняется яркость. Следовательно, для поиска контуров можно искать линии, где меняется значение градиента. *Градиент двумерной функции* $L(x, y)$ – вектор из частных производных $\vec{l} = \nabla(L) = (\partial_x f, \partial_y f)$. Для дискретных функций производные нужно заменить конечными разностями:

$$\vec{l}_{i,j} = (L_{i+1,j} - L_{i-1,j}, L_{i,j+1} - L_{i,j-1}).$$

Абсолютную величину градиента можно вычислить как корень квадратный из скалярного произведения вектора на себя:

$$|\vec{l}| = \sqrt{(\vec{l}, \vec{l})}.$$

Для получения векторного поля, состоящего из градиента яркости L , можно воспользоваться операторами свертки с ядрами вида:

$$\nabla_x = \frac{1}{3} \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad \nabla_y = \frac{1}{3} \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

4.2.3 Нелинейные фильтры

В математической статистике медианой называется число, которое делит выборку чисел так, что половина чисел из оставшейся выборки больше этого числа, остальная половина – чисел выборки меньше этого числа.

Пример 23 Пусть есть выборка чисел 1, 2, 1, 1, 2, 2, 1, 2, 1, 2, 100, 3. Сумма этих 12-ти чисел равна 118-ти, соответственно среднее значение $118/12 \sim 9.93$, а медианное значение равно 2, так как есть ровно половина чисел в наборе не больше 2 и ровно половина не меньше 2.

Для выборок, числа в которых можно расположить по возрастанию с одинаковым шагом, среднее значение выборки совпадает с медианой. В остальных случаях медиана отлична от среднего значения.

Преобразуем растровое изображение следующим способом. Над каждой клеткой изображения разместим окно размером $(2m + 1) \times (2m + 1)$ так, что эта клетка будет находится в центре окна. Найдем медианное значение яркости в этом окне и запишем это значение яркости в новую матрицу для растрового изображения. Такое преобразование изображения – *Медианный фильтр*. Про новую матрицу растрового изображения говорят, что

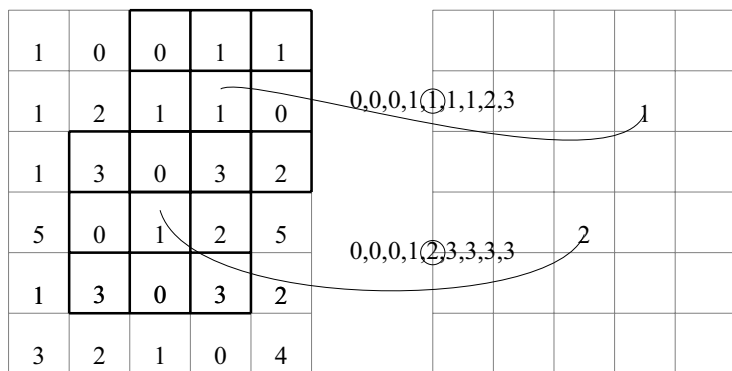


Рис. 4.8: Действие медианного фильтра. Слева матрица растрового изображения с занесенными в нее значениями яркости. Справа результат действия медианного фильтра. Кружками обведены медианные значения множеств элементов матрицы в окнах размером 3×3 около центрального элемента, для которого вычисляется значение фильтра.

это изображение обработанное медианным фильтром. Способ вычисления медианного значения для фильтра размером 3×3 показан на рисунке 4.8.

Медианный фильтр отбрасывает максимальные и минимальные значения из окрестности текущего пикселя. Таким образом с помощью медианного фильтра можно избавиться от отдельных ярких или наоборот темных пикселей в изображении, которые могут возникать из-за шумов матрицы либо дефектов светочувствительного датчика (матрицы) с помощью которого получено изображение.

Медианный фильтр нельзя представить в виде линейного оператора. Поэтому он является разновидностью нелинейного фильтра.

Часто бывает удобно переходить к двухцветному изображению. Для этого можно использовать *пороговый фильтр*. Для этого нужно посчитать среднее значение яркости по всему кадру. Новое растровое изображение сформировать по простому правилу. Если яркость превышает некоторое пороговое значение – например среднее значение, то считать пиксель белым, в противном случае – черным.

Если изображение неравномерно по яркости – например, часть изображения сильно засвечена или затемнена, тогда пороговый фильтр с постоянным значением порога для всего изображения может исказить настоящую картину. В таких случаях удобным бывает *пороговый фильтр с адаптивным порогом*. В таком фильтре пороговое значение выставляется как сред-

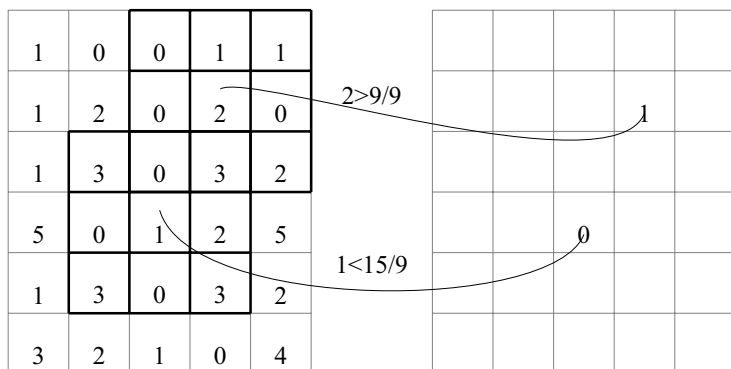


Рис. 4.9: Действие адаптивного порогового фильтра 3×3 . Слева матрица растрового изображения с занесенными в нее значениями яркости. Над вспомогательными линиями указано отношение числа в центре окна фильтра и среднего значения в окне. Среднее значение указано в виде дроби.

нее значение в некотором окне размером $(2m + 1) \times (2m + 1)$ с центром в рассматриваемой точке. Новое изображение, сформированное после пробега по всем точкам старого называется изображением после преобразования фильтра с адаптивным порогом.

4.2.4 Морфологические преобразования

Часто приходится обрабатывать двухцветные черно-белые изображения. Это могут быть задачи распознавания текста, распознавание штрих-кода или QR-кода. При работе с такими изображениями важную роль играет форма объекта.

Для удаления шумов и восстановления очертания объекта используются два преобразования – эрозия и наращивание.

Для этих преобразований используются структурные элементы. Типичный структурный элемент – квадратная матрица со сторонами 3×3 пикселя или крест из пяти пикселей, в котором четыре пикселя, прилегают сторонами к центральному пикселю.

Эрозия – центральный пиксель в структурном элементе принимает минимальное значение яркости из всех пикселей структурного элемента.

Действие эрозии на матрицу растрового изображения приведено на рисунке 4.10.

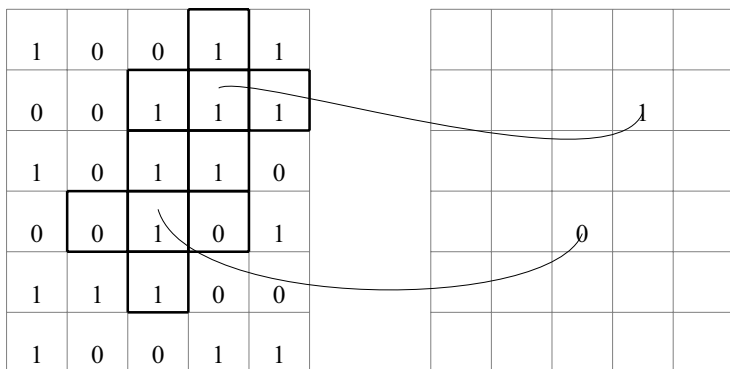


Рис. 4.10: Действие эрозии для структурного элемента «крест». Слева матрица растрового изображения с занесенными в нее значениями после эрозии.

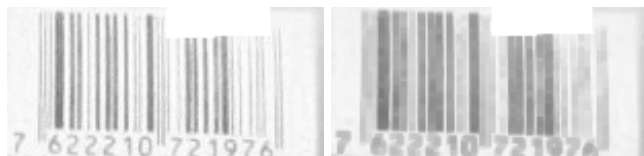


Рис. 4.11: Пример действия эрозии (справа). Для примера использовалась программа GIMP.

Пример 24 Пример действия эрозии на фотографию штрих-кода можно увидеть на рисунке 4.11.

В результате этого действия все объекты меньше структурного элемента удаляются из изображения.

Наращивание – центральный пиксель в структурном элементе принимает максимальное значение яркости из всех пикселей структурного элемента. В результате наращивания увеличивается размер яркого объекта. То есть наращивание – обратное действие эрозии.

Последовательное применение эрозии и наращивания называется *размыкающим фильтром*. В результате этой операции убираются элементы вне выделенной области.

Применение наращивания, а затем эрозии называется *замыкающим фильтром* – в результате удаляются помехи внутри объекта.

4.2.5 Аффинные преобразования

При обработке изображений бывает важно сделать геометрические преобразования над изображением, чтобы привести его к стандартному виду. Например, при распознавании штрих-кода было бы удобным правильно ориентировать прямоугольник штрих-кода. Это же бывает важным при распознавании текста и во многих других задачах.

Взаимно однозначное преобразование плоскости, сохраняющее прямые называется *аффинным преобразованием*.

Так как аффинное преобразование сохраняет прямые, то есть линейные уравнения, тогда и само преобразование должно быть линейным. Общий вид линейного преобразования:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} S_1 \\ S_2 \end{pmatrix}.$$

Здесь $A_{i,j}$ – элементы матрицы A , $\vec{S} = (S_1, S_2)$ – вектор сдвига.

Так как аффинное преобразование плоскости взаимно однозначно, то есть обратимо, то матрица A также должна быть обратимой, то есть ее определитель не равен нулю.

$$\det(A) \neq 0.$$

С геометрической точки зрения аффинные преобразования сводятся к параллельному переносу, повороту и растяжению вдоль прямой. Параллельный перенос на вектор \vec{S} ниже не рассматривается, так как это преобразование обычно вполне очевидно. Поэтому будем считать, что $\vec{S} \equiv 0$. При

работе с растровыми изображениями удобно, чтобы преобразованное изображение имело тот же размер. С точки зрения геометрии это означает, что преобразованное изображение должно иметь то же количество пикселей, что и исходное. Это требование эквивалентно условию

$$\det(A) = \pm 1. \quad (4.2)$$

Таким образом ниже обсуждаются аффинные преобразования, в которых может быть три произвольных параметра – при трех заданных коэффициентах матрицы A четвертый определяется из уравнения для определителя (4.2). С геометрической точки зрения это поворот и сжатие-растяжение с условием сохранения площади.

Пример 25 Пусть в изображении нужно преобразовать параллелограмм с координатами углов $(0, 0)$, (x_1, k_1x_1) , $(x_1 + x_3, k_1x_1 + k_3x_3)$, (x_3, k_3x_3) в прямоугольник с вершиной в начале координат и сторонами параллельными осям координат. То есть с вершинами в точках $(0, 0)$, $(x'_1, 0)$, (x'_1, y'_2) , $(0, y'_2)$. Для этого нужно сделать аффинное преобразование, при котором получим:

$$\begin{pmatrix} x'_1 \\ 0 \\ x'_1 \\ y'_2 \\ 0 \\ y'_2 \end{pmatrix} = \begin{pmatrix} A_{11}x_1 + A_{12}k_1x_1 \\ A_{21}x_1 + A_{22}k_1x_1 \\ A_{11}(x_1 + x_3) + A_{12}(k_1x_1 + k_3x_3) \\ A_{21}(x_1 + x_3) + A_{22}(k_1x_1 + k_3x_3) \\ A_{11}x_3 + A_{12}k_3x_3 \\ A_{21}x_3 + A_{22}k_3x_3 \end{pmatrix}.$$

Линейно независимые уравнения в этой системе второе и пятое. В результате получится система уравнений:

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} A_{21}x_1 + A_{22}k_1x_1 \\ A_{11}x_3 + A_{12}k_3x_3 \end{pmatrix}.$$

Совместно с уравнением (4.2) это дает выражения для трех коэффициентов матрицы A через четвертый:

$$A_{21} = -k_1A_{22}, \quad A_{12} = \frac{1}{(k_1 - k_3)A_{22}}, \quad A_{11} = \frac{k_3}{(k_3 - k_1)A_{22}}.$$

Параметр A_{22} – произвольный не равный нулю. Это означает, что существует неограниченное количество прямоугольников с площадью, равной площади исходного параллелограмма. Задать значение параметра A_{22} можно, например, если заранее задать длину одной из сторон прямоугольника – либо x'_1 , либо y'_2 .

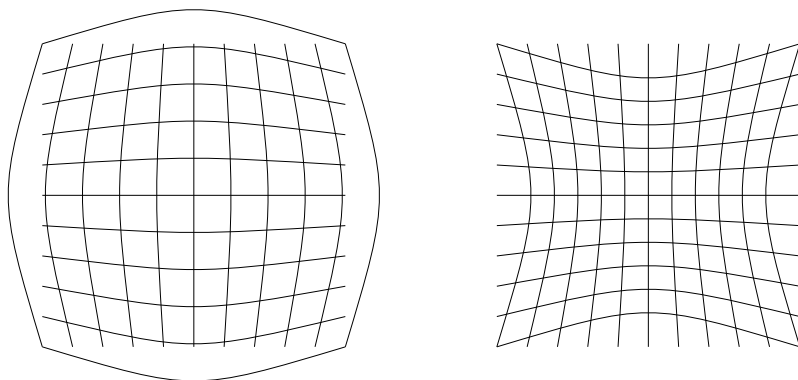


Рис. 4.12: Примеры дисторсии. Слева – «бочка», справа – «подушка». Бочкообразная дисторсия свойственна широкоугольным объективам, а подушкообразная – телеобъективам или снимкам с большим фокусным расстоянием для объективов с варьируемым фокусным расстоянием.

4.2.6 Упражнения

Упражнение 47 Для упражнений с фильтрами удобно воспользоваться, например, свободно распространяемой программой *GIMP*. В пункте меню «Фильтры->Общие->Матрица свертки» можно задавать произвольное ядро оператора свертки и видеть результат преобразования изображения.

4.3 Цифровая калибровка оптики

4.3.1 Дисторсия. Примеры

Изображение проектируется на матрицу фото или видеокамеры с искажениями. Чем сложнее оптическая система, тем разнообразнее могут быть вносимые искажения. Обычно искажения разбиваются на три большие группы.

- Виньетирование– затенение кадра по краям.
- Хроматические аберрации – разложение белого света на составляющие в силу разности длин световых волн и разных углов преломления для различных цветовых составляющих.

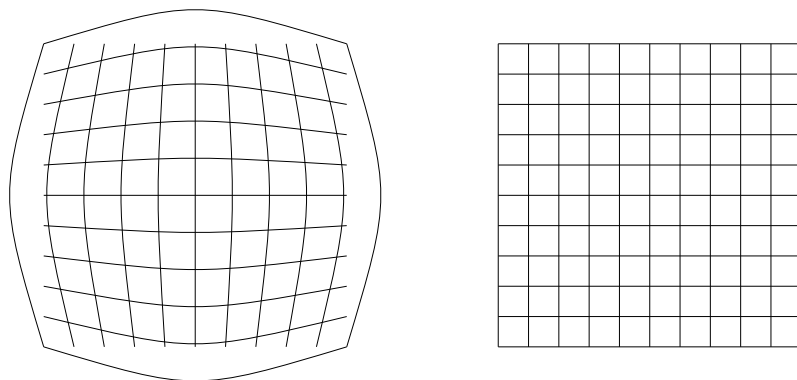


Рис. 4.13: Для устранения дисторсии необходимо найти отображение плоскости в себя, при котором параллельные прямые оригинала оказались бы параллельными на обработанном изображении (справа).

- Дисторсия – геометрические искажения формы объектов из-за преобразований изображения в объективе.

4.3.2 Цифровая калибровка объектива

В этом разделе будет рассмотрен способ цифровой калибровки объектива для компенсации дисторсии.

Для компенсации искажений решетки удобно воспользоваться методом калибровки.

Для того, чтобы искаженную проекцию решетки из отрезков параллельных прямых преобразовать в параллельные прямые, будем искать преобразование, которое переводит двумерную область на себя, так, что искаженные прямые переводятся в неискаженные.

Искажениям формы подвергаются прямые, не проходящие через центр изображения. Но и на прямых, проходящих через центр, также присутствуют искажения – искажения размера, равные по длине отрезки на изображении прямой соответствуют отрезкам разной длины на оригинале. На примерах рис. 4.12 это две прямые – центральная вертикальная и центральная горизонтальная.

Будем предполагать, что в объективе используется система радиально симметричных линз. Поэтому не будем рассматривать сложные случаи, когда в изображении присутствуют искажения как типа бочка, так и типа

подушка одновременно для разных участков изображения.¹

Пусть имеется решетка из N горизонтальных и M вертикальных прямых, расположенных в кадре. На изображении она принимает вид 4.12. Построим отображение плоскости на себя такое, чтобы после обработки получилась квадратная решетка.

Для определенности примем, что точки этой сетки на неискаженном изображении должны иметь координаты $\{(i, j)\}$, где $i = 1, \dots, N$, $j = 1, \dots, M$

Пересечение этих прямых дает сетку пикселей изображения $\{(x_{ij}, y_{ij})\}$. Будем искать такое преобразование плоскости на себя:

$$X = X(x, y), \quad Y = Y(x, y),$$

чтобы на преобразованном изображении получилась решетка как можно более близкая к неискаженной сетке $\{(i, j)\}$, где $i = 1, \dots, N$, $j = 1, \dots, M$.

Обозначим точки сетки после преобразования (X_{ij}, Y_{ij}) . Необходимо подобрать это преобразование так, чтобы (X_{ij}, Y_{ij}) расположить как можно ближе к неискаженной решетке.

Задачу построения отображения можно сформулировать так:

$$S = \sum_{i=0}^N \sum_{j=0}^M (X(x_{ij}, y_{ij}) - i)^2 + (Y(x_{ij}, y_{ij}) - j)^2 \rightarrow \min. \quad (4.3)$$

То есть сумма квадратов расстояний от точки изображения до ее позиции в неискаженной форме должна быть минимально возможной.

Построение отображения, удовлетворяющего условия минимума, и будем называть цифровой калибровкой объектива.

4.3.3 Метод наименьших квадратов

В предыдущем разделе была поставлена задача построения отображения, которое минимизирует сумму квадратов расстояний значений такого отображения от заданных точек. Такой способ построения отображений называется *методом наименьших квадратов*, сокращенно – МНК.

Обычно метод наименьших квадратов используется для построения *аппроксимации* – приближения некоторой неизвестной функции с помощью подбора параметров выбранной функции, которая в каком-либо смысле больше всего подходит для приближения.

¹Сложные комбинации дисторсии могут встречаться на изображениях, полученных с помощью адаптивной оптики, которая имеется, например, в больших телескопах.

Искомое отображение, которое минимизирует сумму квадратов расстояний до заданных точек, называется *регрессией*.

Основные шаги метода наименьших квадратов следующие.

- Выбрать множество точек, для которого необходимо построить приближенную зависимость. Например, это пары $\{(x_i, y_i)\}_{i=1}^N$, где x_i – значение аргумента, y_i – значение неизвестной функции в этой точке.
- Выбрать общую формулу регрессии. Эта общая формула должна содержать параметры, которые будут подбираться с помощью метода наименьших квадратов. Такой общей формулой регрессии может быть, например, линейная функция $y = ax + b$, в которой неизвестные параметры – это коэффициент наклона прямой a и сдвиг b , или полином какой-нибудь степени, у которого заранее неизвестны коэффициенты при разных степенях независимой переменной.
- Решить задачу минимизации для суммы квадратов расстояний от точек неизвестной функции до точек регрессии с теми же значениями аргумента (функции невязки). То есть определить параметры регрессии.

Пример 26 Написать минимизируемую функцию для метода наименьших квадратов, для линейной регрессии, если задано множество точек $(1, 2)$, $(2, 1)$, $(3, 3)$.

Линейную функцию $y = a_1x + a_2$ нельзя провести через заданные три точки. Значение значения функции регрессии в заданных точках:

$$\begin{aligned} y|_{x=1} &= a_1 + a_2; \\ y|_{x=2} &= 2a_1 + a_2; \\ y|_{x=3} &= 3a_1 + a_2. \end{aligned}$$

Построим целевую функцию:

$$S(a_1, a_2) = (a_1 + a_2 - 2)^2 + (2a_1 + a_2 - 1)^2 + (3a_1 + a_2 - 3)^2.$$

Известно, что в точке экстремума функции ее производная по аргументу обращается в нуль. Функция невязки S зависит от нескольких переменных. Кроме того, важно заметить, что функция S имеет только один экстремум, этот экстремум – точка минимума.

Функция S зависит от нескольких аргументов. Для функций нескольких аргументов используется понятие частной производной.

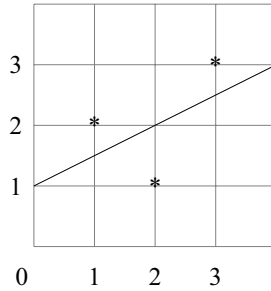


Рис. 4.14: Линейная регрессия $y = x/2 + 1$, построенная методом наименьших квадратов, для набора точек, отмеченных знаком «*»: (1,2),(2,1),(3,3).

- *Частная производная* функции $S(\vec{a})$ по переменной a_i – это производная функции по переменной a_i при условии, что все остальные переменные, от которых зависит функция S , считаются неизменными параметрами. То есть формальное определение:

$$\frac{\partial S}{\partial a_i} = \lim_{\Delta \rightarrow 0} \frac{S(a_1, \dots, a_i + \Delta, \dots, a_N) - S(a_1, \dots, a_i, \dots, a_N)}{\Delta}.$$

Пример 27 Вычислим частную производную по a_3 функции:

$$S(a_1, a_2, a_3) = (a_1 - 2)^2 + a_2 a_1 a_3 + (a_2 - 1)^3 + a_3^4.$$

$$\frac{\partial S}{\partial a_3} = a_2 a_1 + 4a_3^3.$$

По аналогии с функциями одной переменной, функция нескольких переменных имеет экстремум в некоторой точке, если в этой точке равны нулю все ее частные производные.

Пример 28 Построить линейную регрессию для примера (26).

Найдем точку экстремума для функции из примера (26). Для этого вычислим частные производные:

$$\begin{aligned} \frac{\partial S}{\partial a_1} &= 2(a_1 + a_2 - 2) + 4(2a_1 + a_2 - 1) + 6(3a_1 + a_2 - 3), \\ \frac{\partial S}{\partial a_2} &= 2(a_1 + a_2 - 2) + 2(2a_1 + a_2 - 1) + 2(3a_1 + a_2 - 3). \end{aligned}$$

Приведем подобные слагаемые:

$$\begin{aligned}\frac{\partial S}{\partial a_1} &= 28a_1 + 12a_2 - 26, \\ \frac{\partial S}{\partial a_2} &= 12a_1 + 6a_2 - 12.\end{aligned}$$

Тогда условия экстремума по обоим переменным:

$$\frac{\partial S}{\partial a_1} = 0, \quad \frac{\partial S}{\partial a_2} = 0$$

приводят к системе линейных уравнений:

$$28a_1 + 12a_2 = 26, \quad 12a_1 + 6a_2 = 12.$$

Решение этой системы дает $a_1 = 1/2$, $a_2 = 1$.

В результате формула линейной регрессии имеет вид

$$y = \frac{1}{2}x + 1.$$

Построенная регрессия и точки, по которым она построена, изображены на рисунке (4.14).

4.3.4 МНК для калибровки объектива

Ниже разобран способ вычисления параметров регрессии для калибровки объектива..

Глобальный минимум в формуле (4.3) равен нулю. Этот минимум достигается на преобразовании, которое преобразует искаженное объективом изображение решетки в точности в прямоугольную решетку. Такое преобразование теоретически существует, и его можно построить, однако, это связано со значительными вычислительными затратами.

Рассмотрим линейное преобразование, как способ коррекции изображения.

$$\begin{aligned}X(x, y) &= ax + by + c, \\ Y(x, y) &= ux + vy + w.\end{aligned}$$

Задача состоит в том, чтобы подобрать коэффициенты отображения a , b , c , u , v , w минимизирующие невязку (4.3).

Для заданной функции S в точке минимума в нуль обращаются производные по всем ее аргументам. Это условие дает систему линейных уравнений для определения коэффициентов:

$$\begin{aligned}\frac{\partial S}{\partial a} &= 0, & \frac{\partial S}{\partial b} &= 0, & \frac{\partial S}{\partial c} &= 0, \\ \frac{\partial S}{\partial u} &= 0, & \frac{\partial S}{\partial v} &= 0, & \frac{\partial S}{\partial w} &= 0.\end{aligned}$$

Эти производные вычисляются явно, так, что получаются линейные уравнения относительно переменных a, b, c, u, v, w :

$$\begin{aligned}2a \sum_{i,j} x_{ij}^2 + 2b \sum_{i,j} x_{ij} y_{ij} + 2c \sum_{i,j} x_{ij} - 2 \sum_{i,j} i x_{ij} &= 0, \\ 2a \sum_{i,j} x_{ij} y_{ij} + 2b \sum_{i,j} y_{ij}^2 + 2c \sum_{i,j} y_{ij} - 2 \sum_{i,j} i y_{ij} &= 0, \\ 2a \sum_{i,j} x_{ij} + 2b \sum_{i,j} y_{ij} + 2cMN - M(N-1)N &= 0, \\ 2u \sum_{i,j} x_{ij}^2 + 2v \sum_{i,j} x_{ij} y_{ij} + 2w \sum_{i,j} x_{ij} - 2 \sum_{i,j} j x_{ij} &= 0, \\ 2u \sum_{i,j} x_{ij} y_{ij} + 2v \sum_{i,j} y_{ij}^2 + 2w \sum_{i,j} y_{ij} - 2 \sum_{i,j} j y_{ij} &= 0, \\ 2u \sum_{i,j} x_{ij} + 2v \sum_{i,j} y_{ij} + 2wMN - NM(M-1) &= 0.\end{aligned}$$

Выглядит эта система громоздко, однако, ее вид можно существенно упростить, используя новые обозначения:

$$\begin{aligned}\tilde{X}_3 &= \sum_{i,j} x_{ij}^2, & \tilde{X}_2 &= \sum_{i,j} i x_{ij}, & \tilde{X}_1 &= \sum_{i,j} j x_{ij}, & \tilde{X}_0 &= \sum_{i,j} x_{ij} \\ \tilde{Z} &= \sum_{i,j} x_{ij} y_{ij}, & \tilde{Y}_3 &= \sum_{i,j} y_{ij}^2, & \tilde{Y}_2 &= \sum_{i,j} i y_{ij}, \\ \tilde{Y}_1 &= \sum_{i,j} j y_{ij}, & \tilde{Y}_0 &= \sum_{i,j} y_{ij}.\end{aligned}$$

Обозначения с тильдой – это числа, которые получаются суммированием соответствующих выражений известных значений x_{ij} и y_{ij} .

Исходные уравнения теперь можно переписать в виде двух систем уравнений. Это система уравнений для определения a, b, c

$$\begin{aligned}2a\tilde{X}_3 + 2b\tilde{Z} + 2c\tilde{X} - 2\tilde{X}_2 &= 0, \\ 2a\tilde{Z} + 2b\tilde{Y}_3 + 2c\tilde{Y} - 2K\tilde{Y}_2 &= 0, \\ 2a\tilde{X} + 2b\tilde{Y} + 2cMN - MN(N-1) &= 0\end{aligned}$$

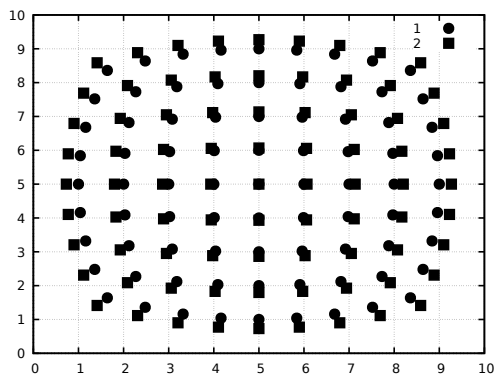


Рис. 4.15: Здесь 1 – искаженная сеть точек, 2 – наложенная на нее сеть точек, преобразованная после линейной калибровки методом наименьших квадратов.

и система уравнений для определения u, v, w .

$$\begin{aligned} 2u\tilde{X}_3 + 2v\tilde{Z} + 2w\tilde{X} - 2\tilde{X}_1 &= 0, \\ 2u\tilde{Z} + 2v\tilde{Y}_2 + 2w\tilde{Y} - 2\tilde{Y}_1 &= 0, \\ 2u\tilde{X} + 2v\tilde{Y} + 2wMN - NM(M-1) &= 0. \end{aligned}$$

Решения этих систем уравнений определяют линейные отображения $X(x, y)$ и $Y(x, y)$, которые оптимальны с точки зрения метода наименьших квадратов.

Пример использования линейного преобразования к калибровке объекта представлен на рисунке 4.15. Суммарная невязка – функция S из (4.3) для искаженной сетки $S \sim 8.5$, для сетки после линейной коррекции $S \sim 4.23$.

Ниже будет построено преобразование в виде двух полиномов третьей степени, которое окажется наилучшим из полиномов третьей степени для (4.3).

Будем искать преобразование в виде:

$$\begin{aligned} X(x, y) &= a_0 + a_1x + a_2y + b_{20}x^2 + b_{11}xy + b_{02}y^2 + \\ &\quad c_{30}x^3 + c_{21}x^2y + c_{12}xy^2 + c_{03}y^3, \\ Y(x, y) &= u_0 + u_1x + u_2y + v_{20}x^2 + v_{11}xy + v_{02}y^2 + \\ &\quad w_{30}x^3 + w_{21}x^2y + w_{12}xy^2 + w_{03}y^3. \end{aligned}$$

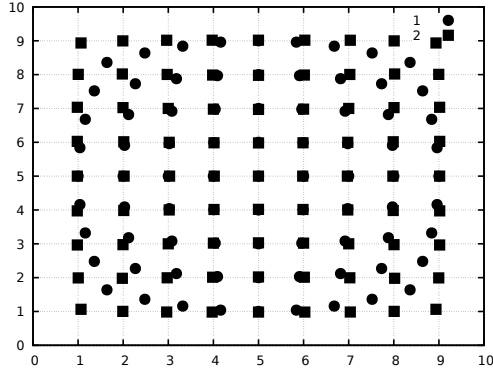


Рис. 4.16: Здесь 1 – искаженная сеть точек, 2 – наложенная на нее сеть точек, преобразованная после кубической калибровки методом наименьших квадратов.

Задача – подобрать коэффициенты полиномов так, чтобы после преобразования решетка была близка к прямоугольной, то есть чтобы на найденном наборе коэффициентов достигался минимум (4.3).

Так как в экстремальных точках производные равны нулю, тогда частные производные по каждой из переменных должны быть равны нулю. Это условие дает систему линейных уравнений для определения коэффициентов:

$$\begin{aligned} \frac{\partial S}{\partial a_i} &= 0, \quad i = 0, 1, 2; \\ \frac{\partial S}{\partial u_i} &= 0, \quad i = 0, 1, 2; \\ \frac{\partial S}{\partial b_{ij}} &= 0 \quad 0 \leq i, j, \quad i + j = 2; \\ \frac{\partial S}{\partial v_{ij}} &= 0 \quad 0 \leq i, j, \quad i + j = 2; \\ \frac{\partial S}{\partial c_{ij}} &= 0 \quad 0 \leq i, j, \quad i + j = 3; \\ \frac{\partial S}{\partial b_{ij}} &= 0 \quad 0 \leq i, j, \quad i + j = 3. \end{aligned}$$

Это линейная система, но более громоздкая, чем для линейного отображения. Решение этой системы уравнений даст коэффициенты искомым

полиномов.

Пример использования кубического отображения для цифровой коррекции приведен на рисунке 4.16. Суммарная невязка – функция S из (4.3) для искаженной сетки $S \sim 8.5$, для сетки после кубической коррекции $S \sim 0.08$. То есть коррекция с помощью кубического отображения существенно эффективнее линейной.

4.3.5 Упражнения

Упражнение 48 Построить методом наименьших квадратов прямую $ax + by + c = 0$ проходящую ближе всего к точкам $(0, 0)$, $(1, 3)$, $(4, 4)$.

Упражнение 49 Построить методом наименьших квадратов параболу $ax^2 + by^2 + cxy + dx + ey + f = 0$, проходящую ближе всего к точкам $(0, 0)$, $(1, 3)$, $(4, 4)$, $(1, 2)$, $(0, 3)$, $(2, 2)$, $(3, 1)$.

4.4 Обзор литературы

В качестве краткого учебного пособия по обработке изображений можно использовать учебное пособие университета ИТМО [22] толщиной в 192 страницы. Для более подробного изучения подойдет книга [16], в которой тема изложена на 1072 страницах. Еще одно издание [30], в котором подробно излагаются рассмотренные в главе темы, содержит 927 страниц.

Глава 5

Теория автоматического управления. Дискретные отображения и дифференциальные уравнения.

Теория автоматического управления применима для объектов различной природы. Одинаковые принципы управления используются для поддержания заданной температуры, скорости вращения вала двигателя и, например, уровня жидкости в резервуаре.

Как правило, разделы про теорию автоматического управления используют математические модели и методы дифференциальных уравнений. В этом разделе кроме дифференциальных уравнений рассматриваются разностные уравнения. Такие уравнения проще для восприятия читателем. Кроме того именно разностные уравнения наиболее близки природе рассматриваемых ниже систем управления – дискретных цифровых систем обработки данных. Такой системой может быть как микроконтроллер, так и компьютер.

5.1 Управление, примеры объектов управления

Теория автоматического управления (ТАУ) – довольно абстрактная и математизированная дисциплина. Замечательно, что общие принципы теории позволяют ее использовать для объектов разной природы: от простых механизмов до сложных электронных устройств и, даже, социальных групп в обществе.

Здесь ТАУ будет рассматриваться как общий подход к управлению элек-

тромеханическими системами, используемыми в робототехнике. Как правило, это управление приводами – моторами. Однако примеры приводятся и из других областей техники – от управления бытовыми приборами, до регуляторов с обратной связью паровых двигателей.

5.1.1 Терминология ТАУ

В каждой развитой научной дисциплине используется своя терминология. Терминология придумывается не для того, чтобы сразу отличить специалиста, а для того, чтобы при обсуждении технических деталей с помощью небольшого количества слов избавиться от пространных пояснений, необходимых для точного определения темы.

Здесь, чтобы выбраться за пределы рассмотрения отдельных важных и показательных примеров применения ТАУ, удобно ввести несколько терминов.

- Объектом управления будем называть предмет, которым предполагаем управлять. Таким предметом может быть электромотор, светодиод, летательный аппарат или утюг. Предполагается, что по контексту изложения можно будет разобраться, какие объекты пригодны для того или иного типа управления.
- *Под управлением* обычно понимается какое либо контролируемое воздействие на объект. Это подача электрического напряжения на контакты нагревающего элемента, в результате которого объект, например, утюг будет нагреваться, или изменение электрического напряжения на клеммах электромотора, в результате чего изменится скорость вращения вала этого мотора.

Говорят, что объект управляем, если управляющее воздействие позволяет менять характеристику объекта.

Системы управления предназначены для изменения характеристик объекта. Системы управления бывают разомкнутые или с обратной связью.

- *Разомкнутые системы управления* работают независимо от состояния системы, по некоторой заранее заданной программе. Например, робот пылесос, может не проверять насколько хорошо был почищен пол в помещении, а работать по заданной программе, предполагающей многократный проезд по всей площади пола, что заведомо должно привести к сбору мусора и пыли со всей поверхности. Еще один пример – режим отжима в стиральной машине предполагает вращение барабана с бельем с максимальной скоростью в течение заранее заданного времени. При этом машина не проверяет, хорошо ли отжато белье.

- Система управления с обратной связью реагирует на состояние объекта управления. Пример системы с обратной связью – электрический чайник, который отключается при закипании в нем воды. Обратная связь в таком чайнике устроена с помощью датчика пара – биметаллической пластины. Еще один пример – поддержание заданной температуры в кастрюле-мультиварке. Здесь необходимо постоянное поддержание определенной температуры. Для этого подогрев включается при достижении некоторой температуры T_0 и отключается при достижении температуры T_1 . При этом для нормальной работы важно, чтобы $T_1 > T_0$. Так работает система управления с обратной связью импульсного воздействия.

Существуют системы управления, воздействующие непрерывно на объект регулирования, практически без задержек. Один из самых распространенных приборов с таким воздействием – часовой механизм. Довольно простой пример механической системы, регулирующей скорость вращения – центробежный регулятор. В электротехнике одна из первых систем с обратной связью – осциллятор ван-дер-Поля.

5.1.2 Особенность систем управления без обратной связи

Управление без обратной связи несмотря на ограниченность возможностей довольно широко распространено и вполне пригодно, если свойства управляемого объекта хорошо известны и прогнозируемы.

Пример 29 Мобильному роботу-погрузчику необходимо передвинуться на расстояние $L = 5$ метров вперед. Если робот не снабжен датчиками, тогда придется запускать систему управления без обратной связи. Простой расчет может быть основан на знании скорости вращения мотора, примем $\omega = 0.5$ рад./с., и диаметра ведущих колес робота, примем $D = 0.5$ м. В результате окажется, что мотор должен отработать в течение $t = 2L/(\omega D) = 2 \times 5/(0.5 \times 0.5) = 40$ сек.

На самом деле при таком простом расчете не учитывался дополнительный груз на погрузчике, напряжение его аккумуляторной батареи, возможный наклон или неровности пола. В результате следует ожидать, что погрузчик либо не доедет до требуемой позиции, либо переедет ее.

Отсюда напрашивается вывод:

- Для использования незамкнутой системы управления необходимы сложные математические модели объектов.

5.1.3 Упражнения

Упражнение 50 Мобильный робот с дифференциальным управлением имеет геометрические размер: ширина колеи $-0,2\text{м.}$, диаметр ведущих колес $-0,04\text{м.}$, номинальная угловая скорость вращения моторов $0,5\text{ рад/с.}$ Указать время работы моторов для разворота робота на месте.

Упражнение 51 Мобильный робот с дифференциальным управлением имеет геометрические размеры: ширина колеи $-0,2\text{м.}$, диаметр ведущих колес $-0,04\text{м.}$, номинальная угловая скорость вращения моторов $0,5\text{ рад/с.}$ Указать время работы мотора, приводящего в движение левое колесо, для поворота робота на угол 90° при неподвижном правом колесе.

Упражнение 52 Мобильный робот со всенаправленными колесами 1.7 имеет геометрические размеры: расстояние от центра робота до колес $0,1\text{ м.}$, $-$ диаметр всенаправленных колес $-0,04\text{м.}$, номинальная угловая скорость вращения моторов $0,5\text{ рад/с.}$ Указать время работы моторов для поворота робота вокруг своей оси на угол 90° .

5.2 Пропорциональный регулятор в системах первого порядка

Здесь рассмотрен регулятор, использующий обратную связь. Он работает по отклонению от требуемого значения и выдает управляющее воздействие пропорционально этому отклонению. Воздействует регулятор не на саму характеристику, а на скорость ее изменения, то есть на первую производную. В результате получается уравнение, которое содержит только текущее значение характеристики и скорость изменения этой характеристики.

- Уравнения, которые содержат только неизвестную функцию и ее производную, называются *дифференциальными уравнениями первого порядка*.

Здесь рассматриваются процессы, математические модели которых являются дифференциальными уравнениями первого порядка. В основном это процессы, связанные с рассеянием тепла¹.

¹Для мобильных роботов и манипуляторов используются уравнения ньютоновской динамики. Такие уравнения содержат ускорения. Ускорение является второй производной координаты, поэтому, в ситуации общего положения это дифференциальные уравнения второго порядка.

5.2.1 Пропорциональный регулятор для стабилизации температуры

Рассмотрим задачу о поддержании заданной температуры в автоклаве. Гипотетический контур управления:

- пусть в автоклав встроен термодатчик, который выдает значение температуры;
- цифровой контроллер сравнивает текущее значение температуры. Если текущая температура ниже требуемой, то увеличивает напряжение на нагревательном элементе пропорционально разности между требуемой температурой и текущей.

Рассмотрим математическую модель для такого регулятора.

- В автоклав поступает тепло пропорционально квадрату напряжения электрического напряжения u на нагревательном элементе, согласно закону Джоуля-Ленца: $(u^2/R)\Delta t$, где R – электрическое сопротивление нагревательного элемента, Δt – временной интервал.
- Тепло рассеивается в окружающую среду пропорционально разности температур, согласно закону Ньютона: $K(T - T_0)\Delta t$, где K – некоторый коэффициент, который, будем считать, получен опытным путем; T – температура в автоклаве; T_0 – температура окружающей среды.

В результате получается уравнение теплового баланса для изменения количества тепла ΔQ за время Δt :

$$\Delta Q = \left(\frac{u^2}{R} - K(T - T_0)\right)\Delta t.$$

Пусть требуемая температура $T = T_1$. Тогда управление должно быть таким, чтобы при этой температуре изменение количества тепла равнялось нулю. Это условие дает возможность определить значение напряжения на нагревательном элементе:

$$u_0 = \sqrt{RK(T_1 - T_0)}.$$

В результате полученное значение u_0 без управления должно обеспечить стабилизацию температуры при T_1 .

Однако температура окружающего пространства T_0 , используемая в формулировке закона Ньютона, не постоянна. Так же в этой формулировке есть коэффициент K , величина которого, вообще говоря, зависит от внешних условий, например, потока воздуха около автоклава. Кроме того при

нагреве сопротивление проводника в нагревательном элементе зависит от его температуры. Очевидно, есть и другие неучтенные факторы.

Для того, чтобы автоклав не перегревался и не остывал при изменениях внешних условий, к значению расчетного напряжения u_0 добавляют управляемое регулятором слагаемое, которое зависит от разности температур:

$$u = u_0 + k_p(T_1 - T).$$

Здесь k_p – коэффициент пропорционального регулятора. В результате формула теплового баланса примет вид:

$$\Delta Q = \left(\frac{(u_0 + k_p(T_1 - T))^2}{R} - K(T - T_0) \right) \Delta t.$$

Если подставить формулу для u_0 и произвести очевидные преобразования, то получится:

$$R\Delta Q = \left((2\sqrt{RK(T_1 - T_0)}k_p + RK)(T_1 - T) + k_p^2(T_1 - T)^2 \right) \Delta t.$$

Заметим, что согласно физическому закону о постоянстве теплоемкости (закон Дюлонга-Пти), изменение тепла в автоклаве пропорционально изменению температуры:

$$\Delta Q = C_p \Delta T.$$

Тогда скорость изменения температуры в автоклаве за время Δt зависит от отклонения текущей температуры от требуемого значения $T - T_1$:

$$RC_p \frac{\Delta T}{\Delta t} = (2\sqrt{RK(T_1 - T_0)}k_p + RK)(T_1 - T) + k_p^2(T_1 - T)^2.$$

Очевидно, при $T = T_1$ изменение температуры равно нулю. То есть, $T = T_1$ стационарная точка.

Рассмотрим знак правой части формулы для $(\Delta T)/(\Delta t)$ при малых отклонениях от стационарного значения.

- $T < T_1$. При этом температура должна расти, так как регулятор должен поднимать температуру до требуемого значения. Следовательно, правая часть формулы должна быть больше нуля. В правой части есть два слагаемых:

- слагаемое $(2\sqrt{RK(T_1 - T_0)}k_p + RK)(T_1 - T)$ линейно по отклонению и положительно при условии

$$k_p > -\frac{1}{2}\sqrt{\frac{RK}{T_1 - T_0}}; \quad (5.1)$$

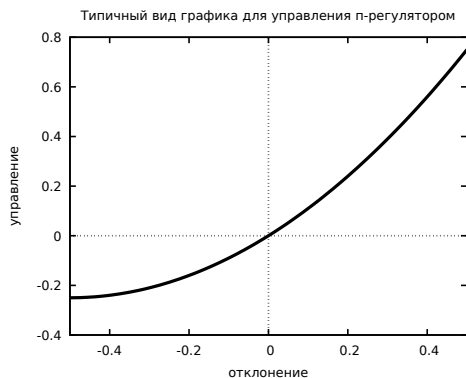


Рис. 5.1: Вид графика управления электрическим напряжением нагревательного элемента при пропорциональном регулировании.

– слагаемое $k_p^2(T_1 - T)^2$ положительно при любом k_p .

Следовательно, при условии (5.1) температура в термостате растет.

- $T > T_1$. При этом правая часть выражения должна быть меньше нуля, так как температура должна убывать при превышении требуемого значения. Для этого должно быть выполнено условие:

$$(2\sqrt{RK(T_1 - T_0)}k_p + RK) + k_p^2(T_1 - T) > 0,$$

или:

$$\frac{1}{k_p^2}(2\sqrt{RK(T_1 - T_0)}k_p + RK) > -(T_1 - T).$$

То есть, существует окрестность $(T_1 - T) < 0$, в которой температура убывает под действием пропорционального регулятора.

В результате доказано утверждение:

Утверждение 10 При условии (5.1) пропорциональный регулятор стабилизирует температуру в окрестности точки $T = T_1$.

5.2.2 Стабилизация объекта пропорциональным регулятором

В этом разделе попробуем формализовать рассмотренный выше способ управления, абстрагировавшись от того, какой объект подвергается управле-

нию.

Пусть объект имеет характеристику x . Будем полагать, что есть способ воздействия на объект так, что в результате управления характеристика x изменяется. То есть объект управляем. Задача регулятора – поддерживать значение этой характеристики равное x_0 .

Пусть со временем без воздействия управления характеристика x меняется по некоторому закону $x = x_0 + vt$. В результате ее отклонение от требуемого значения $x - x_0 = vt$. Для того, чтобы вернуть систему в требуемое состояние, подействуем на объект управлением.

В самом простом случае управление $u(t)$ может определять характеристику непосредственно:

$$x - x_0 = vt + u(t).$$

Для того, чтобы характеристика объекта приняла значение x_0 необходимо управление $u = -vt$. Однако если вместо точного значения v известно только его приближение \tilde{v} , тогда:

$$x - x_0 = vt - \tilde{v}t.$$

Погрешность в определении v приведет к тому, что со временем характеристика x достаточно далеко отклонится от требуемого значения x_0 .

Обозначим погрешность в определении скорости изменения характеристики $v - \tilde{v} = \delta$. Тогда:

$$x - x_0 = \delta t,$$

то есть разомкнутое управление в этом случае оказывается непригодным на больших временах. Следующий шаг состоит в том, чтоб влиять на погрешность в определении управления. Для этого можно воспользоваться отклонением характеристики от требуемого значения.

Пусть в управление будет включена составляющая, пропорциональная отклонению характеристики от требуемой. Тогда в момент времени t

$$V = \tilde{v} + k(x - x_0).$$

Здесь k – коэффициент пропорциональности. Величина V непостоянна во времени, так как со временем меняется разность $x - x_0$. Однако можно получить формулу для скорости изменения характеристики x . Обозначим скорость изменения характеристики \dot{x} . Тогда уравнение для \dot{x} :

$$\dot{x} = \delta - (x - x_0)k.$$

Рассмотрим, как поведет себя величина x в этом случае.

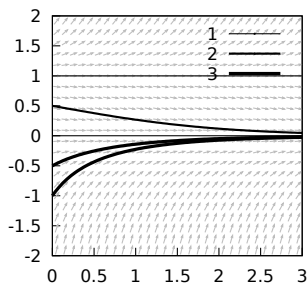


Рис. 5.2: Поле направлений и графики для зависимости температуры от времени. 1 – состояние равновесия, 2 – кривая с начальным условием с большей температурой, 3 – кривые с меньшей начальной температурой. Поле направлений показывает устойчивость нулевого решения.

Прежде всего можно заметить, что если:

$$x - x_0 = \delta/k,$$

то скорость изменения характеристики x равна нулю. Такое значение $x_s = x_0 + \delta/k$ называется стационарным.

Из формулы для стационарного решения следует, что чем меньше δ и чем больше коэффициент k , тем ближе стационарное решение к требуемому x_0 .

Выясним, как ведут себя решения, близкие к стационарному. Для этого выведем уравнение для скорости изменения разности $X = (x - x_s)$.

Из формулы для скорости:

$$\dot{x} = \delta - (x - x_s + \delta/k)k \quad (5.2)$$

или

$$\dot{x} = -k(x - x_s).$$

Так как x_s – постоянна, то скорость изменения разности $(x - x_s)$ пропорциональна \dot{x} . То есть:

$$\dot{X} = -kX.$$

Из полученной формулы следует, что при положительном значении разности X скорость ее изменения отрицательна, то есть x замедляется; если $X < 0$, то скорость x растет. Отсюда следует, что управление приводит X к нулю, а саму величину x – к стационарному решению.

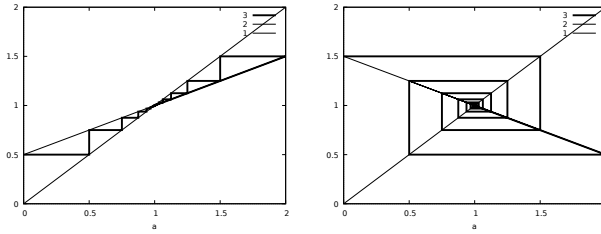


Рис. 5.3: Дискретные отображения для (5.3), где $x_s = 1$, $k = 0.5$ при $\Delta = 1$ – слева и при $\Delta = 3$ – справа. Соответственно, на графике слева – прямая 1: $y = x$, прямая 2: $y = x - k(x - x_s)\Delta t \equiv 0.5x + 0.5$; на графике справа – прямая 1: $y = x$, прямая 2: $y = x - k(x - x_s)\Delta t \equiv -0.5x + 1.5$. Стационарная точка на обоих рисунках $(1, 1)$ – точка пересечения прямых, отмеченных индексами 1 и 2. Индексом 3 на графиках изображены сходящиеся итерации с двумя разными начальными условиями: $x_0 = 0$ и $x_0 = 2$.

5.2.3 Дискретное отображение при управлении через цифровой контроллер

Обычно эволюция объекта управления изучается с точки зрения дифференциальных уравнений. Теория дифференциальных уравнений, как большая часть математического анализа, основана на понятии о бесконечно малой величине и теории пределов. Такой подход вполне себя оправдывает в случае классических регуляторов, рассмотренных в разделе 5.5.

Использование цифровой техники подразумевает дискретность управления. В случае, когда шаг дискретизации мал по сравнению с типичным временем реакции объекта на управление, использование непрерывных моделей, таких как дифференциальные уравнения вполне оправдано. Однако если шаг дискретизации в системе управления не мал, тогда приходится переходить к изучению дискретных отображений. В нашем случае дискретные отображения представляются еще и интуитивно понятнее, чем дифференциальные уравнения.

Рассмотрим уравнение, связывающее мгновенную скорость и управление (5.2). Для записи этого уравнения в дискретном виде будем считать, что управление меняется в моменты времени t_i , где $i \in \mathbb{Z}$. Тогда скорость можно представить как среднюю скорость между моментами времени t_{i+1} и t_i . Правая часть уравнения включает управление. Управление вычисляется в момент времени t_i и до следующего момента t_{i+1} управление не меняется. Поэтому в правой части величины x соответствуют моменту времени t_i , их

будем обозначать x_i . В результате дискретная форма уравнения (5.2) примет вид:

$$\frac{x_{i+1} - x_i}{t_{i+1} - t_i} = \delta - (x_i - x_s + \delta/k)k,$$

Эту формулу можно использовать для определения последующей координаты по предыдущей:

$$x_{i+1} = x_i - k(x_i - x_s)(t_{i+1} - t_i).$$

В этом выражении используется разность $t_{i+1} - t_i$ – текущий шаг дискретизации управления. Обычно для каждого объекта управления величина этого шага постоянна. Удобно обозначить $\Delta t = t_{i+1} - t_i$. В результате получим:

$$x_{i+1} = x_i - k(x_i - x_s)\Delta t. \quad (5.3)$$

Это дискретное отображение числовой прямой в себя $\mathbb{R} \rightarrow \mathbb{R}$.

Отображение (5.3) имеет стационарную точку $x = x_s$. Чтобы выяснить устойчивость стационарной точки, рассмотрим, как отображение меняет разность $x_{i+1} - x_i$. Для этого можно записать отображение (5.3) для точек x_{i+1} и x_i :

$$\begin{aligned} x_{i+1} &= x_i - k(x_i - x_s)\Delta t, \\ x_i &= x_{i-1} - k(x_{i-1} - x_s)\Delta t, \end{aligned}$$

затем вычесть из первой формулы вторую. В результате получится:

$$(x_{i+1} - x_i) = (x_i - x_{i-1}) - k((x_i - x_{i-1})\Delta t).$$

Теперь удобно обозначить разность $y_{i+1} = (x_i - x_{i-1})$. Тогда для y_i получится отображение:

$$y_{i+1} = (1 - k\Delta t)y_i.$$

Тогда отношение y_{i+1} к y_i :

$$\frac{y_{i+1}}{y_i} = 1 - k\Delta t.$$

Очевидно, что y_i – геометрическая последовательность. Если показатель последовательности – выражение $1 - k\Delta t$ удовлетворяет условию $|1 - k\Delta t| < 1$, то $y_i \rightarrow 0$ при $i \rightarrow \infty$. Так как при этом разность между x_{i+1} и x_i стремится к нулю, следовательно, последовательность x_i сходится.

Утверждение 11 Дискретное отображение для пропорционального регулятора сходится к стационарной точке x_s , если выполнено условие $|1 - k\Delta t| < 1$

5.2.4 Упражнения

Упражнение 53 Отображение $x_{n+1} = x_n - \sin(x_n)\Delta$. Определить, при каких значениях параметра Δ решение $x = 0$ устойчиво.

Упражнение 54 Найти стационарные точки отображения $x_{n+1} = x_n/2 + x_n^2 \Delta$. Определить, при каких значениях параметра Δ они устойчивы.

Упражнение 55 Отображение $x_{n+1} = x_n(1 - x_n)\Delta$. Определить, при каких значениях параметра Δ существуют стационарные точки отображения, отличные от $x = 0$.

5.3 Регулятор в системах второго порядка

Здесь рассмотрим работу регулятора в случае, когда регулятор может влиять на регулируемую характеристику объекта, меняя вторую производную этой характеристики, то есть ускорение. Например, именно ускорение меняется при приложении дополнительных сил – управляющих воздействий в механических системах.

5.3.1 Остановка робота на заданном расстоянии от стены

Здесь рассматривается математическая модель, в которой предполагается, что время за которое проходит цикл управления очень мало, так что можно считать, что система управления работает непрерывно. В этом случае можно использовать мгновенные значения координат, скоростей и ускорений робота, именно так, как эти мгновенные значения понимаются в механике:

- координаты – непрерывные функции времени;
- скорости – первые производные координат;
- ускорения – вторые производные координат.

Рассмотрим мобильного робота с датчиком расстояния. Пусть робот движется по прямой. Задача – останавливать робота на заданном расстоянии X_* от стены. Для простоты формул примем систему отсчета в которой $X_* = 0$.

САУ робота устроена следующим образом. Контроллер регулярно опрашивает датчик. Датчик выдает величину расстояния до стены перед роботом. Контроллер может менять напряжение на контактах электромоторов, приводящих в движение робота.

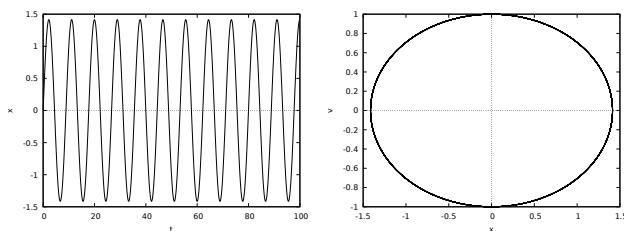


Рис. 5.4: Слева – зависимость расстояния от времени в математической модели робота под действием пропорционального регулятора без учета механического трения. Справа – траектория движения робота на фазовой плоскости в математической модели с пропорциональным регулятором без учета трения.

Для остановки на заданном расстоянии можно запрограммировать контроллер так, чтобы величина крутящего момента мотора линейно зависела от разности текущего расстояния до X_* .

Пусть масса робота m . Моментом инерции колес и ротора самого мотора будем пренебрегать. Ускорение робота будем обозначать \ddot{X} . Будем считать, что колеса робота не проскальзывают. Тогда сила, действующая на робота со стороны колес, пропорциональна крутящему моменту мотора. В результате уравнение динамики для робота примет вид:

$$m\ddot{X} + kX = 0. \quad (5.4)$$

Это хорошо известное уравнение грузика на пружинке – линейного осциллятора. Его решения – колебания с постоянной амплитудой. То есть, при принятых предположениях в нашей, во многом идеальной модели, робот не сможет остановиться на заданном расстоянии, а будет двигаться вперед-назад, пока не закончится заряд его батареи.

Замечательно, у этой идеальной модели есть закон сохранения энергии:

$$E = m \frac{\dot{X}^2}{2} + k \frac{X^2}{2}.$$

Величина E не меняется со временем. Если обозначить $v = \dot{X}$ и посмотреть на формулу закона сохранения энергии с точки зрения геометрии, то окажется, что скорость робота v и координата X во время движения лежат на эллипсе:

$$\frac{m}{2E} v^2 + \frac{k}{2E} X^2 = 1.$$

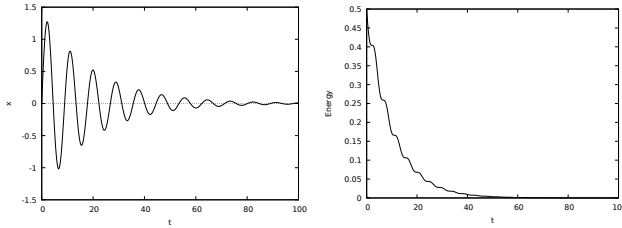


Рис. 5.5: Слева – амплитуда затухающих колебаний в окрестности нуля в математической модели робота с пропорциональным регулятором с учетом механического трения. Справа – график зависимости механической энергии от времени в этой же математической модели.

Этот эллипс располагается на плоскости, где в качестве ортогональных координат используются X, v .

- Плоскость, на которой в качестве координат используются координата X и скорость $v = \dot{X}$, называется *фазовой плоскостью* для уравнения (5.4).

Для уравнения (5.4) все решения лежат на концентрических эллипсах в фазовой плоскости.

На самом деле обычно робот не двигается без остановок вперед-назад около точки 0. В нашей модели не учтено механическое трение. В самых простых моделях механическое трение – дополнительное слагаемое, пропорциональное скорости $\mu\dot{X}$ в уравнении динамики, здесь $\mu > 0$ – коэффициент трения. Это механическое трение, возможно и малое, ведет в конечном итоге к затуханию колебаний в рамках нашей новой модели с трением:

$$m\ddot{X} + kX + \mu\dot{X} = 0. \quad (5.5)$$

Для этой модели закон сохранения механической энергии уже не действует. Если продифференцировать по времени формулу для энергии E , считая энергию неизвестной функцией времени, получим:

$$\frac{dE}{dt} = m\dot{X}\ddot{X} + k\dot{X}X = (m\ddot{X} + kX)\dot{X}.$$

Теперь, если считать, что X – решение уравнения (5.5), тогда

$$m\ddot{X} + kX = -\mu\dot{X}.$$

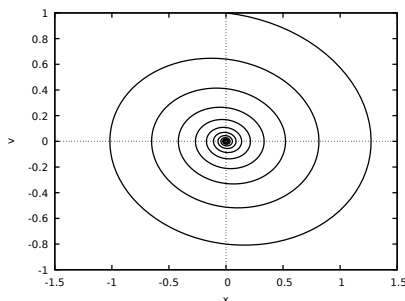


Рис. 5.6: Траектория на фазовой плоскости для уравнения с трением – спираль, сходящаяся к началу координат.

Следовательно, производная энергии:

$$\frac{dE}{dt} = -\mu \dot{X}^2.$$

В правой части этой формулы неположительная величина, следовательно, механическая энергия убывает со временем.

5.3.2 Неустойчивость пропорционального регулятора для остановки робота около стены

Если время цикла управления не мало, то вместо математической модели для непрерывной системы управления приходится использовать дискретную модель. В дискретной модели вместо мгновенных значений скоростей и ускорений используются средние значения.

Для построения дискретной модели нужно вычислять ускорение между тактами системы управления. Для простоты будем считать, что состояние системы меняется в моменты времени t_i , такт управления – промежуток времени между моментами t_{i+1} и t_i одинаков для всего процесса и равен Δ .

Среднюю скорость робота между моментами t_{i+1} и t_i можно вычислить по стандартной формуле:

$$v_i = \frac{X_{i+1} - X_i}{\Delta}.$$

Среднее ускорение:

$$a_i = \frac{v_{i+1} - v_i}{\Delta}.$$

Теперь, если подставить выражения для средних скоростей в формулу для среднего ускорения, получим:

$$a_i = \frac{1}{\Delta} \left(\frac{X_{i+1} - X_i}{\Delta} - \frac{X_i - X_{i-1}}{\Delta} \right).$$

После очевидных преобразований получится:

$$a_i = \frac{X_{i+1} - 2X_i + X_{i-1}}{\Delta^2}.$$

Тогда дискретная система для управляемой величины X_{i+1} примет вид:

$$m \frac{X_{i+1} - 2X_i + X_{i-1}}{\Delta^2} + kX_i = 0.$$

Эту дискретную систему можно переписать в виде отображения:

$$mX_{i+1} = m2X_i - mX_{i-1} - kX_i\Delta^2.$$

Здесь X_{i+1} зависит от двух предыдущих значений X_i и X_{i-1} .

По двум начальным значениям X_0 и X_1 можно построить последовательность X_i .

Вместо одной формулы удобнее пользоваться двумя – в одной вычислять скорость на i -м участке, а во второй – координату.

Обозначим V_i среднюю скорость, тогда можно получить рекуррентную систему для последовательных отображений, эквивалентную дискретному уравнению:

$$X_{i+1} = X_i + V_i\Delta, \quad (5.6)$$

$$mV_{i+1} = mV_i - kX_i\Delta. \quad (5.7)$$

Для того, чтобы понять, как ведет себя эта последовательность, удобно воспользоваться тем же приемом – посмотреть поведение суммы:

$$2\mathcal{E} = kX_{i+1}^2 + mV_{i+1}^2. \quad (5.8)$$

Здесь \mathcal{E} – дискретный аналог полной механической энергии – сумма «потенциальной энергии» – это слагаемое $kX_{i+1}^2/2$ и «кинетической энергии» – слагаемое $mV_{i+1}^2/2$. Для непрерывного управления под действием пропорционального регулятора эта сумма не меняется со временем.

Выясним, меняется ли \mathcal{E} при изменении от i к $i+1$. Для этого подставим в правую часть (5.8) выражения для X_{i+1} и V_{i+1} из (5.6) и (5.7). В результате получим:

$$2\mathcal{E} = kX_i^2 + 2k\Delta X_i V_i + k\Delta^2 V_i^2 + mV_i^2 - 2k\Delta V_i X_i + \frac{k^2}{m} X_i^2 \Delta^2.$$

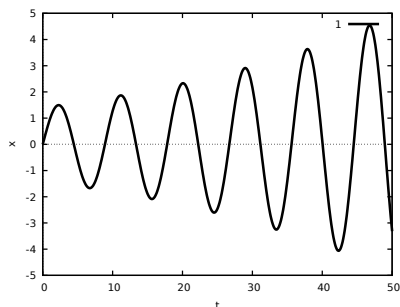


Рис. 5.7: Колебания с возрастающей амплитудой в дискретной модели управления роботом массы $m = 1$ с пропорциональным регулятором, коэффициентом регулятора $k = 0.5$ и временем дискретизации управления $\Delta = 0.1$.

После приведения подобных слагаемых получим:

$$kX_{i+1}^2 + mV_{i+1}^2 = (k + \frac{k^2}{m}\Delta^2)X_i^2 + (m + k\Delta^2)V_i^2.$$

Параметры k и m - положительные. Поэтому:

$$kX_{i+1}^2 + mV_{i+1}^2 > kX_i^2 + mV_i^2.$$

То есть последовательные отображения расходятся. Нулевое решение $X = 0$, $V = 0$ – неустойчивая точка дискретного отображения.

- Следствие – пропорциональный регулятор для остановки робота около стены в идеальных условиях – очень малом трении и достаточно большом временном шаге управления не стабилизирует робота.

5.3.3 Пропорционально-дифференциальный регулятор в системах второго порядка

В механических системах трение, как правило, приводит к стабилизации. В электрических системах потеря энергии также стабилизирует ситуацию. Однако если такие потери малы, то стабилизация движения проходит медленно. Поэтому для быстрой остановки малых колебаний схвата робота или колебаний мобильного робота при повороте в системы управления с пропорциональным регулятором вводят дополнительное слагаемое – дифференциальную часть регулятора.

В системах управления она является аналогом потери энергии. Дифференциальная часть регулятора работает по отклонению скорости изменения регулируемой характеристики.

Рассмотрим модель управления роботом для остановки на заданном расстоянии от стены из предыдущего раздела. Было показано, что трение играет важную роль в стабилизации стационарной точки. Однако трение в механической системе, как правило, является нежелательным элементом. Бывает, что величина трения не поддается контролю.

Надежный способ стабилизировать стационарную точку за счет управления – добавить в регулятор зависимость от скорости – полный аналог трения.

Если в качестве системы управления для остановки робота взять закон управления в виде:

$$u = kx + \mu\dot{x},$$

то уравнение движения робота под действием такого управления примет вид:

$$m\ddot{x} + kx + \mu\dot{x} = 0. \quad (5.9)$$

Здесь k – коэффициент пропорциональной составляющей регулятора, μ – коэффициент дифференциальной составляющей регулятора.

Нетрудно сообразить, что это уравнение имеет нулевое решение $x(t) \equiv 0$. Важно знать свойства решений в окрестности нуля.

Для этого удобно воспользоваться свойством *асимптотической устойчивости*.

Нулевое решение *асимптотически устойчиво*, всякое решение, лежащее в малой окрестности нуля, стремится к нулю при $t \rightarrow \infty$.

Покажем, что нулевое решение асимптотически устойчиво. Для этого рассмотрим всегда положительное выражение:

$$E(x, \dot{x}) = m\frac{\dot{x}^2}{2} + k\frac{x^2}{2}.$$

Постоянное значение E определяет эллипс с центром в начале координат.

Если продифференцировать это выражение, то получим:

$$\frac{dE}{dt} = m\dot{x}\ddot{x} + kx\dot{x}.$$

Это выражение можно записать, выделив \dot{x} как множитель:

$$\frac{dE}{dt} = (m\ddot{x} + kx)\dot{x}.$$

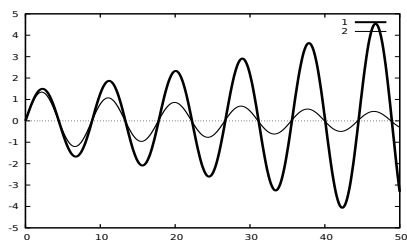


Рис. 5.8: Зависимость расстояния от времени для робота под управлением пропорционального регулятора "1" и пропорционально-дифференциального регулятора "2". Формальные параметры $m = 1$, коэффициент пропорционального регулятора $k = 0.5$, дифференциальный коэффициент пропорционально-дифференциального регулятора $\mu = 0.1$, шаг дискретизации по времени $\Delta = 0.1$. Робот под управлением пропорционально-дифференциального регулятора стабилизируется в окрестности $x = 0$.

Теперь можно воспользоваться уравнением для управления робота под действием пропорционально-дифференциального регулятора (5.9). В этом уравнении удобно перенести слагаемое, соответствующее дифференциальной части регулятора, в правую часть уравнения:

$$m\ddot{x} + kx = -\mu\dot{x}.$$

Тогда производную E можно записать в виде:

$$\frac{dE}{dt} = -\mu\dot{x}^2.$$

Выражение в правой части неположительно. В частности, из него следует, что E убывает со временем. Траектория на фазовой плоскости x, \dot{x} – спираль. Решение со временем все ближе подходит к началу координат. Это говорит, что нулевое решение устойчиво.

5.3.4 Дискретное отображение при управлении с помощью ПД-регулятора

Построим модель для дискретного отображения при управлении ПД-регулятором для остановки робота на заданном расстоянии от препятствия. Будем считать, что шаг по времени одинаков для всех циклов управления и равен Δ . Координаты робота после i -го цикла управления будем обозначать X_i .

Средняя скорость робота на $i + 1$ цикле управления:

$$V_i = \frac{X_{i+1} - X_i}{\Delta}.$$

Среднее ускорение на $i + 1$ цикле управления:

$$A_i = \frac{V_{i+1} - V_i}{\Delta}.$$

Второй закон Ньютона на i -м цикле:

$$mA_i + kX_i + \mu V_i = 0.$$

Если воспользоваться принятыми обозначениями для средней скорости и среднего ускорения, то получим:

$$\begin{aligned} V_i &= \frac{X_{i+1} - X_i}{\Delta}, \\ m \frac{V_{i+1} - V_i}{\Delta} + kX_i + \mu V_i &= 0. \end{aligned}$$

Теперь можно переписать эти уравнения так, чтобы оставить слагаемые с индексом $i + 1$ в левых частях уравнений, а остальные слагаемые перенести в правые части:

$$\begin{aligned} X_{i+1} &= X_i + \Delta V_i, \\ V_{i+1} &= -\frac{k}{m} \Delta X_i + V_i - \frac{\mu}{m} \Delta V_i. \end{aligned}$$

Эту систему удобно преобразовать к матричному виду:

$$\begin{pmatrix} X_{i+1} \\ V_{i+1} \end{pmatrix} = \begin{pmatrix} 1 & \Delta \\ -\frac{k}{m} \Delta & 1 - \frac{\mu}{m} \Delta \end{pmatrix} \begin{pmatrix} X_i \\ V_i \end{pmatrix}. \quad (5.10)$$

Эту систему можно рассматривать как линейное отображение $\mathbb{R}^2 \rightarrow \mathbb{R}^2$.

Для краткости обозначим:

$$A = \begin{pmatrix} 1 & \Delta \\ -\frac{k}{m} \Delta & 1 - \frac{\mu}{m} \Delta \end{pmatrix}.$$

Определитель этого отображения

$$\det(A) = \begin{vmatrix} 1 & \Delta \\ -\frac{k}{m} \Delta & 1 - \frac{\mu}{m} \Delta \end{vmatrix} = 1 - \frac{\mu}{m} \Delta + \frac{k}{m} \Delta^2.$$

Если этот определитель не равен нулю:

$$1 - \frac{\mu}{m}\Delta + \frac{k}{m}\Delta^2 \neq 0,$$

то существует и обратное отображение.

Для линейного отображения существует способ построить все решения. Для этого придется использовать теорию собственных значений и собственных векторов матриц.

Собственным значением матрицы называется число λ такое, что

$$\det(A - \lambda E) = 0.$$

Здесь E – единичная матрица – матрица с единицами на главной диагонали и нулями на всех других позициях.

Пример 30 Возьмем матрицу A вида:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}.$$

Уравнение для собственных значений матрицы размером 2×2 – квадратное уравнение относительно λ . Действительно:

$$\begin{aligned} \det(A - \lambda E) &= \det \left(\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) = \\ &= \begin{vmatrix} A_{11} - \lambda & A_{12} \\ A_{21} & A_{22} - \lambda \end{vmatrix} = (A_{11} - \lambda)(A_{22} - \lambda) - A_{12}A_{21}. \end{aligned}$$

Это уравнение удобно переписать в виде:

$$\lambda^2 - \text{Tr}(A)\lambda + \det(A) = 0. \quad (5.11)$$

Здесь $\text{Tr}(A)$ – след матрицы – сумма диагональных элементов. В нашем случае: $\text{Tr}(A) = A_{11} + A_{22}$.

Вектор \vec{Z}_i называется *собственным вектором* для собственного числа λ_i , если справедливо равенство:

$$A\vec{Z}_i = \lambda_i\vec{Z}_i.$$

Здесь и ниже ограничимся только квадратными матрицами размером 2×2 . В примере 30 показано, что для таких матриц собственные числа – решения квадратного уравнения (5.11). Квадратное уравнение для λ имеет два различных решения, если выполнено неравенство $\text{Tr}(A)^2 - 4\det(A) \neq 0$.

Каждому из этих значений λ соответствует свой собственный вектор \vec{Z}_i . При этом эти векторы линейно независимы.

Любой начальный вектор \vec{Y} можно представить в виде суммы линейно независимых векторов \vec{Z}_i :

$$\vec{Y} = c_1 \vec{Z}_1 + c_2 \vec{Z}_2.$$

Тогда отображение

$$\vec{Y}_1 = A\vec{Y}_0$$

можно представить в виде:

$$\vec{Y}_1 = A(c_1 \vec{Z}_1 + c_2 \vec{Z}_2) = c_1 A\vec{Z}_1 + c_2 A\vec{Z}_2 = c_1 \lambda_1 \vec{Z}_1 + c_2 \lambda_2 \vec{Z}_2.$$

Дальнейшие итерации по индукции:

$$\vec{Y}_{n+1} = A(c_1 \lambda_1^n \vec{Z}_1 + c_2 \lambda_2^n \vec{Z}_2) = c_1 \lambda_1^n A\vec{Z}_1 + c_2 \lambda_2^n A\vec{Z}_2 = c_1 \lambda_1^{n+1} \vec{Z}_1 + c_2 \lambda_2^{n+1} \vec{Z}_2.$$

Окончательно получается формула:

$$\vec{Y}_{n+1} = c_1 \lambda_1^{n+1} \vec{Z}_1 + c_2 \lambda_2^{n+1} \vec{Z}_2.$$

Если оба собственных числа меньше единицы по абсолютной величине: $|\lambda_{1,2}| < 1$, то последовательность $Y_n \rightarrow 0$ при $n \rightarrow \infty$. Если абсолютная величина хотя бы одного из собственных чисел больше единицы, то $Y_n \rightarrow \infty$ при $n \rightarrow \infty$.

Для матрицы

$$A = \begin{pmatrix} 1 & \Delta \\ -\frac{k}{m}\Delta & (1 - \frac{\mu}{m}\Delta) \end{pmatrix}.$$

собственные значения – решения уравнения:

$$\lambda^2 - \left(2 - \frac{\mu}{m}\Delta\right)\lambda + 1 - \frac{\mu}{m}\Delta + \frac{k}{m}\Delta^2 = 0.$$

Дискриминант этого квадратного уравнения:

$$D = \left(2 - \frac{\mu}{m}\Delta\right)^2 - 4\left(1 - \frac{\mu}{m}\Delta + \frac{k}{m}\Delta^2\right).$$

Если раскрыть скобки и сократить подобные слагаемые, то получим:

$$D = \left(\frac{\mu^2}{m^2} - 4\frac{k}{m}\right)\Delta^2.$$

Теперь можно выписать решение:

$$\lambda_{1,2} = 1 - \frac{\mu\Delta}{2m} \pm \frac{\Delta}{2m} \sqrt{\mu^2 - 4km}.$$

Здесь рассмотрим только ответ на вопрос – насколько малым может быть коэффициент дифференциальной составляющей μ , чтобы рекуррентная последовательность из (5.10) сходилась к точке $X = 0$. Поэтому нас будет интересовать случай не слишком больших μ , таких, что $\mu^2 - 4km < 0$.

Выражение $\sqrt{\mu^2 - 4km}$ удобно переписать в виде:

$$\sqrt{\mu^2 - 4km} = \sqrt{-1} \sqrt{-\mu^2 + 4km}$$

или

$$\sqrt{\mu^2 - 4km} = i \sqrt{-\mu^2 + 4km}.$$

В этом выражении под корнем – положительное число, и функция корень понимается обычным образом. Здесь $i = \sqrt{-1}$ – мнимая единица – такое число, что $i^2 = -1$.

Тогда λ_1 и λ_2 – комплексные числа. Их можно представить в виде:

$$\lambda_1 = 1 - \frac{\mu\Delta}{2m} + i \frac{\Delta}{2m} \sqrt{-\mu^2 + 4km}, \quad \lambda_2 = 1 - \frac{\mu\Delta}{2m} - i \frac{\Delta}{2m} \sqrt{-\mu^2 + 4km}$$

Числа λ_1 и λ_2 имеют одинаковые вещественные части:

$$\operatorname{Re}(\lambda_1) = \operatorname{Re}(\lambda_2) = 1 - \frac{\mu\Delta}{2m}$$

и разные по знаку мнимые части:

$$\operatorname{Im}(\lambda_1) = -\operatorname{Im}(\lambda_2) = \frac{\Delta}{2m} \sqrt{-\mu^2 + 4km}.$$

Такие числа называются комплексносопряженными числами.

Модуль комплексного числа – корень квадратный из суммы квадратов вещественной и мнимой частей:

$$|\lambda_1| = |\lambda_2| = \sqrt{\left(1 - \frac{\mu\Delta}{2m}\right)^2 + \left(\frac{\Delta}{2m}\right)^2 (-\mu^2 + 4km)} = \sqrt{1 - \frac{\Delta}{m}(\mu - k\Delta)}.$$

Из этой формулы следует, что $|\lambda_{1,2}| < 1$, если $\mu > k\Delta$.

В результате получаем, что при $\mu^2 < 4km$ и $\mu > k\Delta$ последовательные отображения (5.10) сходятся к нулю. То есть пропорционально-дифференциальный регулятор стабилизирует работа в окрестности точки $X = 0$.

5.3.5 Упражнения

Упражнение 56 Пусть масса робота $m = 0.300$ кг, шаг дискретизации управления $\Delta = 0.5$, коэффициент пропорционального регулятора $k = 1$. Указать, при каких значениях коэффициента дифференциального регулятора расстояние до стены $x = 5$ окажется устойчивой точкой.

Упражнение 57 Найти собственные числа матрицы

$$\begin{pmatrix} 1 & 3 \\ 2 & 1 \end{pmatrix}.$$

5.4 Модель с пропорционально-интегрально-дифференциальным регулятором

5.4.1 Систематические ошибки управления

Есть множество причин появления систематических ошибок систем управления.

Прежде всего, это несовершенство математической модели. Для построения правильной системы управления объектом, как правило, строится его математическая модель. С помощью математической модели определяется управление нужной характеристикой. Очевидно, что математическая модель не учитывает всех факторов, влияющих на характеристику объекта, то есть математическая модель описывает объект приближенно. Из-за несоответствия математической модели и объекта даже при правильно спроектированной системе управления возникают систематические ошибки.

Приведем пример ошибки системы управления мобильным роботом при остановке на заданном расстоянии от препятствия из-за несоответствия математической модели. В движущихся системах, как правило, присутствует нелинейная составляющая в силе трения. В частности, для того, чтобы сдвинуть объект с места, нужно приложить силу, большую некоторого порога. Для робота под действием пропорционально -дифференциального регулятора движение возникает при условии:

$$|kx| > F.$$

Здесь F – пороговое значение силы трения покоя. Это означает, что при учете трения покоя стационарные координаты робота могут быть в пределах $-F/k \leq x_s \leq F/k$, то есть не обязательно в точке $x_s = 0$. Если остановка возникнет при $x_s \neq 0$, тогда появится систематическая ошибка системы управления.

Еще один пример – это использование объекта управления в нерасчетных условиях. Как правило, система управления проектируется для определенных условий работы объекта. При работе на нерасчетных режимах появляется систематическая ошибка управления.

В задачах о движении вдоль линии коэффициент пропорционального регулятора обычно имеет небольшое значение для линий с небольшой кривизной. Однако если появляется участок с большей кривизной, то пропорциональный регулятор не в состоянии вернуть робота на линию. Возникает систематическая ошибка в положении робота относительно линии.

Для устранения систематических ошибок управления используется интегральная составляющая.

5.4.2 Интегральная составляющая регулятора.

Если ошибка управления в системе с пропорциональным регулятором существует в течении длительного промежутка времени, это означает, что имеющаяся пропорциональная часть регулятора устранить эту ошибку не в состоянии. Для увеличения регулирующего воздействия в таком случае используется сумма величин ошибок, накопленных за время работы.

Пусть целевое значение характеристики объекта x_* , циклы управления происходят с периодом Δ . Обозначим x_i – значение характеристики во время i -го цикла управления. Обозначим:

$$I_{i+1} = I_i + (x_i - x_*)\Delta. \quad (5.12)$$

Эта величина пропорциональна времени существования систематической ошибки.

Величину I можно считать по-другому:

$$I = \sum_{j=0}^i (x_j - x_*)\Delta.$$

Если обозначить t_0 – время в момент $j = 0$, T – время в момент $j = i$, тогда сумму можно переписать в несколько ином виде:

$$I = \sum_{t=t_0}^T (x(t) - x_*)\Delta.$$

Теперь, если уменьшить шаг цикла управления: $\Delta \rightarrow 0$, то получится *определённый интеграл*:

$$\mathcal{I} = \int_{t_0}^T (x(t) - x_*)dt.$$

Именно из-за прямой аналогии между суммой и интегралом составляющая (5.12) называется интегральной.

Таким образом, есть механизм исправления систематической ошибки с помощью интегральной составляющей регулятора.

5.4.3 Влияние интегральной составляющей в регуляторе

Рассмотрим, как повлияет добавка интегральной составляющей на стабилизацию робота в окрестности $x = 0$. Динамика робота под действием управления для остановки на заданном расстоянии до препятствия с учетом интегральной составляющей и трения покоя принимает вид:

$$\begin{aligned} \text{при } |kx + \mu\dot{x} + \eta \int_0^t x(\tau)d\tau| > F \quad \text{или} \quad \dot{x} \neq 0 : \\ m\ddot{x} + kx + \mu\dot{x} + \eta \int_0^t x(\tau)d\tau + \text{sgn}(\dot{x})F = 0, \\ \text{иначе:} \quad \dot{x} = 0. \end{aligned}$$

Здесь $\eta > 0$ – коэффициент при интегральной составляющей пропорционально-интегрально-дифференциального регулятора.

Вдали от точки равновесия можно рассмотреть, как меняется полная механическая энергия робота:

$$E = m\frac{\dot{x}^2}{2} + k\frac{x^2}{2}.$$

Продифференцируем E по времени, в результате получим:

$$\dot{E} = m\dot{x}\ddot{x} + k\dot{x}x = \dot{x}(m\ddot{x} + kx).$$

Теперь заменим выражение в скобках в силу уравнения динамики:

$$m\ddot{x} + kx = -\mu\dot{x} - \eta \int_0^t x(\tau)d\tau - \text{sgn}(\dot{x})F$$

Тогда формула для производной полной энергии по времени примет вид:

$$\dot{E} = -\mu\dot{x}^2 - \text{sgn}(\dot{x})F\dot{x} - \dot{x}\eta \int_0^t x(\tau)d\tau.$$

Первые два слагаемых в правой части формулы неположительны. Они ведут к уменьшению полной энергии. Однако интегральное слагаемое может быть как положительным, так и отрицательным, то есть оно может вести

как к потере энергии, так к ее увеличению – накачке. Обычно для предотвращения накачки энергии за счет интегральной составляющей в системе управления уменьшают коэффициент интегрального регулятора.

С качественной точки зрения понятно, что при некотором достаточно большом значении энергии E робот может совершать колебания вокруг состояния равновесия. Если считать коэффициенты μ, η, F малыми, то можно сказать, что колебания близки к колебаниям груза на пружинке без трения. Такие колебания определяются формулой

$$\tilde{x} = \sqrt{\frac{2E}{k}} \cos \left(\sqrt{\frac{k}{m}} t + \phi \right).$$

Здесь

$$E = \frac{m\dot{\tilde{x}}^2}{2} + \frac{k\tilde{x}^2}{2}$$

– механическая энергия. Она определяет амплитуду колебаний, ϕ – начальное положение – определяет фазу колебаний. Частота колебаний равна $\sqrt{k/m}$. Период таких колебаний $T = 2\pi\sqrt{m/k}$.

Рассмотрим изменение энергии за один период, считая, что решение близко к $\tilde{x}(t)$.

По определению интеграла:

$$E(t+T) - E(t) = \int_t^{t+T} \dot{E}(\tau) d\tau.$$

Следовательно, чтобы вычислить изменение энергии за период, достаточно проинтегрировать по периоду правую часть формулы для \dot{E} . Прежде, чем вычислять интегралы, выясним явный вид правой части формулы для \dot{E} .

$$\begin{aligned} S_1 &= -\mu\dot{\tilde{x}}^2 = -\mu\frac{2E}{m} \sin^2 \left(\sqrt{\frac{k}{m}} t + \phi \right); \\ S_2 &= -\operatorname{sgn}(\dot{\tilde{x}}) F \dot{\tilde{x}} = -F \sqrt{\frac{2E}{m}} \left| \sin \left(\sqrt{\frac{k}{m}} t + \phi \right) \right|; \\ S_3 &= -\dot{\tilde{x}} \eta \int_0^t \tilde{x}(\tau) d\tau = \eta \frac{2E}{k} \sin^2 \left(\sqrt{\frac{k}{m}} t + \phi \right) - \\ &\quad - \eta \frac{2E}{k} \sin(\phi) \sin \left(\sqrt{\frac{k}{m}} t + \phi \right). \end{aligned}$$

Последнее слагаемое в этой формуле содержит неотрицательное слагаемое и слагаемое от периодической функции $\sin\left(\sqrt{\frac{k}{m}}t + \phi\right)$ со средним по периоду равным нулю. То есть последним слагаемым в этой формуле при вычислении изменения E за период можно пренебречь.

Рассмотрим по отдельности интегралы по периоду от каждого из слагаемых. Так как интегралы по периоду, то постоянный сдвиг на ϕ в аргументах тригонометрических функций не важен. В вычислениях для удобства его можно принять равным нулю.

$$\begin{aligned}\int_0^T S_1 d\tau &= -\mu \frac{k}{m} E \int_0^T \sin^2\left(\sqrt{\frac{k}{m}}t\right) d\tau; \\ \int_0^T S_2 d\tau &= -F \sqrt{\frac{2E}{m}} \int_0^T \left| \sin\left(\sqrt{\frac{k}{m}}\tau\right) \right| d\tau = \\ &= -2F \sqrt{\frac{2E}{m}} \int_0^{T/2} \sin\left(\sqrt{\frac{k}{m}}\tau\right) d\tau = \\ &= -2F \sqrt{\frac{2E}{k}} (-\cos(\pi) + \cos(0)) = -4F \sqrt{\frac{2E}{k}}; \\ \int_0^T S_3 d\tau &= \eta E \int_0^T \sin^2\left(\sqrt{\frac{k}{m}}(\tau + \phi)\right) d\tau.\end{aligned}$$

Выражения S_1 и S_2 отрицательны на всем периоде. Поэтому интегралы от них также отрицательны. Интеграл от S_2 здесь вычислен явно.

Интегралы от S_1 и от S_3 содержат интеграл по периоду от $\sin^2(\sqrt{k/m}\tau + \phi)$. Здесь мы не будем его вычислять². Заметим, что для того, чтобы энергия за период не возрастала, в предположении о малости μ, F, η достаточно, чтобы интеграл от третьего слагаемого был не больше, чем сумма двух других. То есть, например, если пренебречь интегралом от S_2 , достаточно, чтобы $\eta < \mu k/m$.

Таким образом, при достаточно малых значениях μ, F, η механическая энергия колебаний убывает, если $\eta < \mu k/m$.

Теперь рассмотрим поведение решения этой системы на качественном уровне в малой окрестности $x = 0$. Пусть $x = x_s$ – робот находится в

²Для вычисления этого интеграла можно воспользоваться тригонометрическими формулами для выражения косинуса двойного угла через квадрат синуса или прямо воспользоваться программой для аналитических вычислений, например, свободно распространяемой программой Maxima

состоянии покоя, начиная с некоторого момента $t = t_n$, но $x_s \neq 0$. В это время нарастет интегральное слагаемое:

$$I = I(t_n) + x_s(t - t_n).$$

Тогда в момент $t = t_m$ окажется, что $|kx_s + \eta(I(t_n) + x_s(t_m - t_n))| > F$. В результате робот начнет движение в направлении точки $x = 0$,

Сформулируем полученный результат.

Утверждение 12 *Чтобы пропорционально-интегрально-дифференциальный регулятор стабилизировал робота на заданном расстоянии от препятствия при малых значениях μ, F, η достаточно, чтобы выполнялось условие $\eta < \mu k / m$.*

5.5 Классические регуляторы с обратной связью

Здесь разобраны математические модели классических регуляторов с обратной связью, не использующие микропроцессоры. Такие системы обычно называются аналоговыми. Аналоговые системы управления, как правило, проще и потому надежнее цифровых. Кроме того, они не имеют задержек на обработку данных.

5.5.1 Поплавковый регулятор

Поплавковый регулятор используется в устройствах для регулирования уровня жидкости, например, воды в котлах паровых двигателей. Рассмотрим принцип работы самого простого поплавкового регулятора, приведенного на рисунке 5.9.

Пусть вода поступает в емкость по верхней трубе и вытекает по нижней. Задача регулятора поддерживать целевой уровень воды в сосуде. Сечение в трубе для подачи меняется пропорционально разности текущего и целевого уровня воды. Примем высоту уровня воды, при котором входная труба перекрывается полностью – h . Будем считать, что масса воды, проходящей через входную трубу за время dt , пропорциональна разности между уровнем воды в данный момент и требуемым уровнем воды: $k_1(h - x)dt$. Здесь k_1 – коэффициент пропорциональности. Пусть за время dt вытекает mdt воды. При условии $x < h$, баланс воды в емкости определяется уравнением:

$$dM = k_1(h - x)dt - mdt, \quad x < h.$$

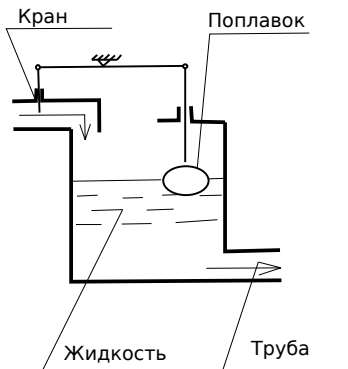


Рис. 5.9: Схема поплавкового регулятора.

Для простоты будем считать, что уровень воды в котле изменяется пропорционально изменению количества воды, например, стенки сосуда вертикальные:

$$dM = k_2 dx.$$

Здесь k_2 – произведение площади дна сосуда на плотность воды.

Тогда получится уравнение баланса для уровня воды в сосуде:

$$k_2 dx = k_1(h - x)dt - mdt.$$

Если обе части уравнения баланса разделить на dt , то получится уравнение:

$$\frac{dx}{dt} = -kx + X, \quad k = \frac{k_1}{k_2}, \quad X = kh - \frac{m}{k_2}.$$

В этом уравнении слева – скорость изменения уровня воды, справа – текущий уровень воды x и известные коэффициенты.

Стационарный уровень воды – это значение $x = x_s$ при котором $(dx)/(dt) = 0$:

$$x_s = h - \frac{m}{k_1}.$$

Стационарный уровень устойчив. Действительно, если $x < x_s$, то скорость изменения уровня:

$$\frac{dx}{dt} > 0,$$

то есть уровень воды повышается. Если же уровень выше стационарного, $x > x_s$, тогда:

$$\frac{dx}{dt} < 0,$$

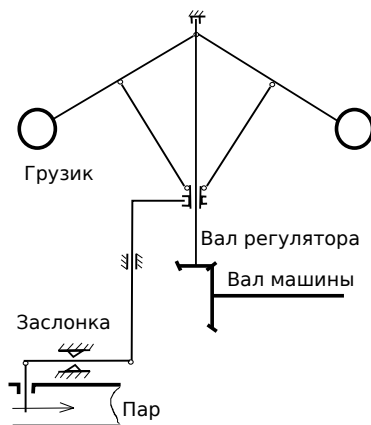


Рис. 5.10: Схема центробежного регулятора. Вращение от вала паровой машины передается на вал регулятора. Под действием центробежной силы грузики регулятора расходятся. Это приводит к закрытию заслонки в трубе с паром, идущей к паровой турбине. Если же скорость вращения вала машины замедляется, то грузики опускаются, в результате заслонка открывается, пар поступает на турбину, и она раскручивает вал паровой машины.

то есть уровень воды понижается.

В результате получено:

Утверждение 13 Поплавковый регулятор поддерживает стационарный уровень жидкости.

5.5.2 Центробежный регулятор

Центробежный регулятор, изображенный на рисунке 5.10, использовался, например, для регулирования скорости вращения мельничных жерновов, валов паровых машин и больших телескопов.

В центробежном регуляторе связь между рабочим валом турбины и регулятором обеспечивается зубчатой парой. В свою очередь регулятор влияет на скорость вращения паровой машины, регулируя количество подаваемого пара через заслонку.

Одна из проблем таких регуляторов – возникновение колебаний в системе «регулятор – паровая машина» при неудачном подборе параметров весов грузиков регулятора и длин плеч. Процесс выглядит следующим образом.

- 1 Регулятор при медленном вращении открывает заслонку и пар подается к паровой машине.
- 2 Вал машины разгоняется и раскручивает вал регулятора.
- 3 Грузики регулятора поднимаются под действием центробежной силы, и регулятор перекрывает заслонку.
- 4 Вал машины затормаживается, вращение вала регулятора замедляется.
- 5 Далее цикл повторяется с большей амплитудой колебаний.

Проблема устойчивости регуляторов интересовала многих математиков 19 века ³.

Первый шаг для ее решения, видимо, был сделан знаменитым Дж. К. Максвеллом. Для формализации задачи выведем уравнения, связывающие угол раскрытия регулятора и скорость вращения вала машины. Для этого примем обозначения:

- ω – угловая скорость вращения вала;
- k – передаточный коэффициент зубчатой пары;
- θ – угловая скорость вращения регулятора, $\omega = k\theta$ – связь между скоростью вращения рабочего вала и регулятора;
- ϕ – угол раскрытия регулятора, $\dot{\phi}$ – скорость изменения угла раскрытия регулятора, $\ddot{\phi}$ – ускорение угла раскрытия регулятора;
- f – угол раскрытия регулятора для нормального хода. Будем считать, что ускорение угловой скорости вращения вала пропорционально – обозначим его $\dot{\omega}$ – разности между текущим углом раскрытия регулятора ϕ и углом раскрытия регулятора для нормального хода f :

$$\dot{\omega} = \kappa(\phi - f).$$

Здесь κ – коэффициент пропорциональности.

³Вопросы устойчивости регуляторов исследовали математики 19 века Д.Б. Эйри [1], Д.К.Максвелл [5], И.А.Вышнеградский [8]. Перевод этих статей с историческим комментарием можно найти в издании Академии наук СССР [19].

Выведем уравнение для регулятора с грузиками массы m и плечом регулятора длины l , вращающегося со скоростью θ . Для этого просуммируем моменты сил, действующих на левый грузик, относительно центра шарнира в верхней точке регулятора. Моменты сил, действующих против часовой стрелки, будем считать положительными.

- Момент инерции грузика, возникающий из-за поворота грузика относительно шарнира в верхней части регулятора $\dot{\phi}ml^2$, действует в положительном направлении исчисления угла ϕ , поэтому будет включен в уравнение с положительным знаком.
- Момент силы тяжести. Сила тяжести действует вертикально вниз. Ее величина mg . Плечо силы $l \sin(\phi)$. В результате момент силы тяжести $mgl \sin(\phi)$ направлен против часовой стрелки – он входит в уравнение моментов с положительным знаком.
- Момент инерции грузика, возникающий из-за вращения вокруг вертикальной оси регулятора со скоростью θ . Это центробежная сила, она лежит в горизонтальной плоскости и направлена от оси вращения. Этот момент равен произведению центробежной силы $\theta^2 ml \sin(\phi)$ на плечо этой силы относительно точки крепления $l \cos(\phi)$. В результате момент равен $\theta^2 ml^2 \sin(\phi) \cos(\phi)$. Этот момент стремится поднять грузик и, следовательно, действует по часовой стрелке. В уравнение его следует включать со знаком минус.
- Момент силы трения в шарнире в верхней части осциллятора $k\dot{\phi}$ действует по направлению производной и будет учтен со знаком плюс.

$$\ddot{\phi}ml^2 + mgl \sin(\phi) - \theta^2 ml^2 \sin(\phi) \cos(\phi) + k\dot{\phi} = 0.$$

Поделив на ml^2 и обозначив $\mu = k/(ml)$, получим:

$$\ddot{\phi} + \frac{g}{l} \sin(\phi) - \theta^2 \sin(\phi) \cos(\phi) + \frac{k}{ml^2} \dot{\phi} = 0.$$

В этом уравнении все слагаемые имеют размерность углового ускорения, то есть сек^{-2} . Для того, чтобы уравнение было записано в безразмерном виде, примем типичное время изменения положения грузика T сек. Здесь T – некоторая постоянная. Ее значение будет определено ниже. Тогда безразмерная независимая переменная: $\tau = t/T$. Такая замена приводит к изменениям в формулах для первой и второй производных:

$$\frac{d\phi}{dt} = \frac{d\phi}{d(T\tau)} = \frac{d\phi}{T d\tau},$$

$$\frac{d^2\phi}{dt^2} = \frac{d\phi}{d(T\tau)^2} = \frac{d\phi}{T^2 d\tau^2}.$$

В результате подстановки $t = T\tau$ уравнение примет вид:

$$\frac{d^2\phi}{T^2 d\tau^2} + \frac{g}{l} \sin(\phi) - \theta^2 \sin(\phi) \cos(\phi) + \frac{k}{ml^2} \frac{d\phi}{T d\tau} = 0.$$

Теперь умножим уравнение на T^2 , получим:

$$\frac{d^2\phi}{d\tau^2} + \frac{gT^2}{l} \sin(\phi) - T^2\theta^2 \sin(\phi) \cos(\phi) + \frac{Tk}{ml^2} \frac{d\phi}{d\tau} = 0.$$

Скорость вращения вала регулятора θ изначально измеряется в радианах в секунду. Чтобы обезразмерить ее примем: $\theta = \vartheta/T$. Здесь введена новая безразмерная скорость вращения ϑ . Для краткости обозначим: $\mu = Tk/(ml^2)$. Чтобы сократить формулу, примем: $gT^2/l = 1$. То есть определим постоянную $T = \sqrt{l/g}$. В результате уравнение примет вид:

$$\frac{d^2\phi}{d\tau^2} + \sin(\phi) - \vartheta^2 \sin(\phi) \cos(\phi) + \mu \frac{d\phi}{d\tau} = 0.$$

Второе уравнение для управления также следует преобразовать к безразмерному виду:

$$\frac{d\omega}{Td\tau} = \kappa(\phi - f).$$

Чтобы замкнуть систему уравнений, заметим, что $k\theta = \omega$, тогда $k\vartheta/T = \omega$. В результате уравнение для ϑ :

$$\frac{k}{T^2} \frac{d\vartheta}{d\tau} = \kappa(\phi - f).$$

Окончательно система уравнений для регулятора имеет вид:

$$\begin{aligned} \frac{d^2\phi}{d\tau^2} + \sin(\phi) - \vartheta^2 \sin(\phi) \cos(\phi) + \mu \frac{d\phi}{d\tau} &= 0. \\ \frac{d\vartheta}{d\tau} &= \nu(\phi - f), \quad \nu = \frac{T^2\kappa}{k} \end{aligned}$$

В этой системе уравнений неизвестными являются ϕ – угол раскрытия регулятора и θ – скорость вращения регулятора вокруг своей оси.

Механическая система находится в равновесии, когда скорости и ускорения ее компонентов равны нулю. В случае системы уравнений для регулятора условие равновесия дает уравнения:

$$\phi = f, \quad \sin(f) - \vartheta^2 \sin(f) \cos(f) = 0.$$

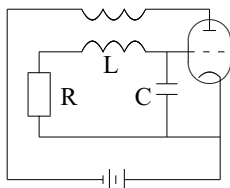


Рис. 5.11: Принципиальная схема осциллятора ван дер Поля на основе лампы-триода. В результате работы возникают циклы зарядки и разрядки конденсатора. Разность напряжений на сетке и катode лампы регулируется конденсатором. Анодный ток наводит ЭДС на вторичной катушке колебательного контура. Этот ток заряжает конденсатор, напряжение на сетке триода увеличивается и сила катодного тока уменьшается. Конденсатор начинает разряжаться. Сила катодного тока увеличивается, в катушке колебательного контура наводится ЭДС, конденсатор заряжается и цикл повторяется.

Откуда

$$v_*^2 = \frac{1}{\cos(f)}.$$

Регулятор обеспечивает устойчивую работу паровой машины, если точка равновесия устойчива. Вопрос об устойчивости сводится к исследованию зависимости корней кубического (алгебраического) уравнения от параметров от коэффициентов этого уравнения. Здесь этот анализ не приводится. Подробности можно посмотреть в учебнике [28]

5.5.3 Математическая модель осциллятора ван дер Поля

Осциллятор ван дер Поля генерирует колебания, частота и амплитуда которых устойчивы по отношению к возмущениям. По существу генератор ван дер Поля можно рассматривать как регулятор, поддерживающий колебания постоянной частоты и амплитуды.

- С математической точки зрения осциллятор ван дер Поля – один из самых простых примеров, в котором появляются особые решения – *предельные циклы* – периодические решения, в окрестности которых других периодических решений нет.

Для вывода уравнения осциллятора ван дер Поля будем пользоваться обозначениями:

M – коэффициент взаимоиנדукции;

I – анодный ток;

E – сеточное напряжение;

E_s – характерное напряжение;

L – индуктивность;

C – ёмкость конденсатора;

R – сопротивление контура;

σ – проводимость лампы.

Разность напряжения на сетке триода и на катоде определяется зарядом на конденсаторе Q :

$$E = \frac{Q}{C}$$

Электрический ток от катода к аноду зависит от напряжения на сетке E . Для этой зависимости используется приближенная формула, в которой есть два слагаемых – одно линейное, второе – кубичное. Поэтому анодный ток нелинейно зависит от напряжения на сетке:

$$I = \sigma \left(1 - \frac{1}{3} \frac{E^2}{E_s^2} \right) E.$$

Уравнение для падения напряжения в колебательном контуре, в который входят катушка индуктивности, сопротивление и конденсатор:

$$L \frac{dI}{dt} + RI + \frac{Q}{C} = M \frac{dI}{dt}.$$

В этом уравнении неизвестные величины – сила тока I и заряд на обкладках конденсатора Q . Изменение заряда равно силе тока:

$$\frac{dQ}{dt} = I,$$

кроме того, напряжение пропорционально заряду на обкладках конденсатора:

$$Q = CE.$$

В результате уравнение можно переписать в виде:

$$LC \frac{d^2 E}{dt^2} + RC \frac{dE}{dt} + E = M\sigma \frac{d}{dt} \left[\left(1 - \frac{1}{3} \frac{E^2}{E_s^2} \right) E \right]$$

В правой части в квадратных скобках E_s не зависит от времени, поэтому продифференцировать нужно только E . В результате:

$$LC \frac{d^2 E}{dt^2} + RC \frac{dE}{dt} + E = M\sigma \frac{dE}{dt} - M\sigma \frac{E^2}{E_s^2} \frac{dE}{dt}.$$

В этом уравнении все слагаемые имеют размерность напряжения (Вольт).

Теперь перейдем к безразмерным переменным. Для этого примем: $E = E_s U$, где E_s – постоянная, имеющая размерность напряжения, U – безразмерная функция времени. Также перейдем к новой независимой переменной величине вместо времени: $t = T\tau$, где T – пока некоторая неопределенная постоянная, имеющая размерность времени, τ – новая безразмерная переменная.

При таком переходе:

$$\frac{dE}{dt} = \frac{E_s}{T} \frac{dU}{d\tau}, \quad \frac{d^2 E}{dt^2} = \frac{E_s}{T^2} \frac{d^2 U}{d\tau^2}.$$

Тогда уравнение примет вид:

$$\frac{LCE_s}{T^2} \frac{d^2 U}{d\tau^2} + \frac{RCE_s}{T} \frac{dU}{d\tau} + E_s U = M\sigma \frac{E_s}{T} \frac{dU}{d\tau} - M\sigma \frac{E_s}{T} U^2 \frac{dU}{d\tau}.$$

Теперь удобно разделить уравнение на LCE_s/T^2 . В результате получим:

$$\frac{d^2 U}{d\tau^2} + \frac{RT}{L} \frac{dU}{d\tau} + \frac{T^2}{LC} U = M\sigma \frac{T}{LC} \frac{dU}{d\tau} - M\sigma \frac{T}{LC} U^2 \frac{dU}{d\tau}.$$

Примем $T = \sqrt{LC}$. Тогда получим:

$$\frac{d^2 U}{d\tau^2} + R\sqrt{\frac{C}{L}} \frac{dU}{d\tau} + U = \frac{M\sigma}{\sqrt{LC}} \frac{dU}{d\tau} - \frac{M\sigma}{\sqrt{LC}} U^2 \frac{dU}{d\tau}.$$

Перенесем все слагаемые в левую часть уравнения:

$$\frac{d^2 U}{d\tau^2} - \left(\left(\frac{M\sigma}{\sqrt{LC}} - R\sqrt{\frac{C}{L}} \right) - \frac{M\sigma}{\sqrt{LC}} U^2 \right) \frac{dU}{d\tau} + U = 0.$$

Выражение в скобках удобно преобразовать к виду:

$$\left(\left(\frac{M\sigma}{\sqrt{LC}} - R\sqrt{\frac{C}{L}} \right) - \frac{M\sigma}{\sqrt{LC}} U^2 \right) = \frac{M\sigma - RC}{\sqrt{LC}} \left(1 - \frac{M\sigma}{M\sigma - RC} U^2 \right)$$



Рис. 5.12: Траектории решений ван дер Поля в плоскости переменных (y, \dot{y}) – координата и скорость. Видно, что как внутренняя, так и наружная траектории накручиваются на некоторую предельную кривую – предельный цикл.

Уравнение можно еще упростить, если обозначить $U = Ay$. Здесь $y = y(\tau)$ – новая неизвестная функция, A – параметр, который подберем так, чтобы упростить уравнение:

$$A \frac{d^2 y}{d\tau^2} - \frac{M\sigma - RC}{\sqrt{LC}} \left(1 - \frac{M\sigma}{M\sigma - RC} A^2 y^2 \right) A \frac{dy}{d\tau} + Ay = 0.$$

Это уравнение поделим на A , обозначим:

$$\nu = \frac{M\sigma - RC}{\sqrt{LC}}, \quad A = \sqrt{1 - \frac{RC}{M\sigma}}.$$

В результате для y получим:

$$\frac{d^2 y}{d\tau^2} - \nu(1 - y^2) \frac{dy}{d\tau} + y = 0.$$

Это уравнение называется уравнением ван дер Поля. Решения этого уравнения обладают важным свойством – существует единственное периодическое решение, к которому стремятся все остальные решения – (кроме решения $y \equiv 0$). Это периодическое решение называется предельным циклом.

Некоторое указание на существование предельного цикла можно усмотреть из следующих рассуждений.

Обозначим $x = \frac{dy}{d\tau}$ и $R^2 = x^2 + y^2$. Продифференцируем выражение для R по τ :

$$\frac{dR^2}{d\tau} = 2R \frac{dR}{d\tau} = 2x \frac{dx}{d\tau} + 2y \frac{dy}{d\tau}.$$

Так как

$$\frac{dx}{d\tau} = \frac{d^2y}{d\tau^2},$$

то вторую производную y можно заменить в силу уравнения, а

$$\frac{dy}{d\tau} = x,$$

тогда:

$$2R \frac{dR}{d\tau} = 2x(-\nu(1 - y^2)x) + 2yx$$

или

$$R \frac{dR}{d\tau} = -\nu x^2(1 - y^2).$$

То есть, при малых R траектории выходят из окрестности точки $(0, 0)$ и величина R растет, при $|y| > 1$ величина R убывает.

5.6 Обзор литературы

Список литературы по теории автоматического регулирования огромен. По большей части это вузовские учебники для инженерных специальностей: инженеров механиков, инженеров электриков и собственно робототехников [26], [27], [21], [17].

С точки зрения математики теория автоматического управления во многом рассматривается как часть теории дифференциальных уравнений и в более общей форме – теории динамических систем. Вместе с тем есть фундаментальные математические издания, посвященные именно теории управления [25].

Глава 6

Графы, карты, потенциальные поля и планирование пути

В этой главе рассматриваются методы

- формирования полигональных и векторных карт;
- вычисления кратчайшего пути с использованием взвешенного графа или метрического тензора;
- построения карты по данным датчиков и преобразования в системах координат;
- локализации на карте с помощью метода триангуляции.

6.1 Структура карты

6.1.1 Карта дорог как взвешенный граф

Дороги можно рассматривать как одномерные объекты – отрезки, связывающие перекрёстки – узлы. Если узлы пронумеровать, а отрезки снабдить весовым коэффициентом – пропорциональным длине дороги между перекрестками, тогда карту дорог можно рассматривать как *взвешенный граф*.

Взвешенный граф, приведенный на рисунке 6.1, удобен для выбора маршрута, но непригоден для навигации робота. Для возможности навигации необходимо добавить значения углов, под которыми пересекаются дороги в узловых точках. В этом случае весовые коэффициенты в ребрах графа оказываются трехмерными векторами. Такой граф изображен на рисунке 6.2.

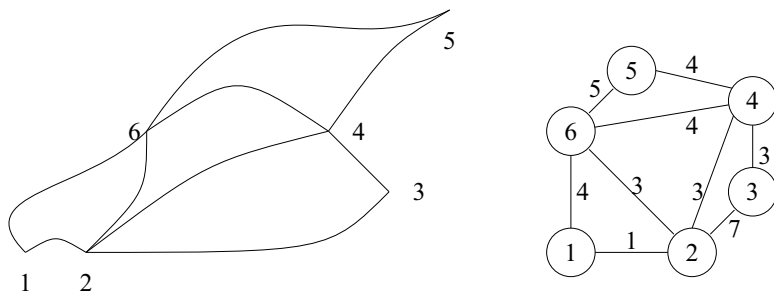


Рис. 6.1: Карта дорог между пунктами 1-6 и эквивалентный ей взвешенный граф. Веса на ребрах соответствуют округленному расстоянию между пунктами 1-6.

Формат вектора весов ребра:

- первый компонент вектора – расстояние между узлами,
- второй компонент вектора – угол, под которым выходит дорога из узла с меньшим номером в узел с большим номером
- третий компонент вектора – угол, под которым выходит дорога из узла с большим номером в узел с меньшим номером.

В результате добавления навигационной информации взвешенный граф для дорожной сети из 6.1 примет вид 6.2.

6.1.2 Специальный случай. Лабиринт

Один из самых простых объектов для составления карты – лабиринт. Пример лабиринта – набор одинаковых квадратных ячеек со стенами. Такой лабиринт можно рассматривать как набор ячеек. Для каждой ячейки лабиринта можно составить карту.

Ячейку лабиринта можно рассматривать как узел графа. Расстояния между ячейками в составных лабиринтах, как правило одинаковы, поэтому карту лабиринта можно представить в виде графа.

Для лабиринта можно использовать другой подход. Каждую ячейку лабиринта рассматривать как элемент матрицы. Для каждой ячейки отдельно построить карту – вектор из четырех элементов, который указывает есть ли

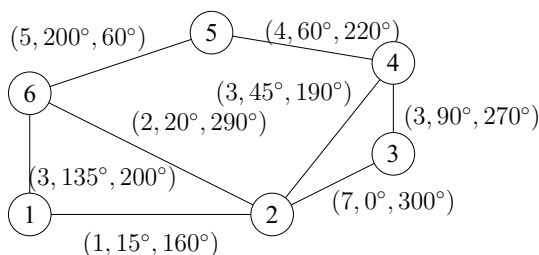


Рис. 6.2: Карта дорог между пунктами 1-6 и эквивалентный ей взвешенный граф. Веса на ребрах соответствуют округленному расстоянию между пунктами 1-6 приблизительным значениям углов поворота относительно направления слева направо.

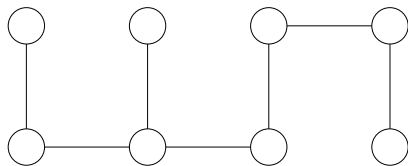


Рис. 6.3: Лабиринт как связанный граф.

у ячейки стенки в четырех направлениях. Набор карт всех ячеек лабиринта – атлас лабиринта.

Атлас типичного прямоугольного лабиринта с квадратными ячейками может быть сформирован в виде двумерного массива. Каждый элемент массива – это ячейка лабиринта. Для него важно показать, в каком направлении стоит стена лабиринта. Ячейка имеет 4 стороны. Каждая из сторон может быть либо свободна от стены, либо закрыта стеной.

Состояние каждой стороны ячейки может быть нулем, если стены нет, и единицей, если стена есть. На первой позиции двоичного 4-х-значного вектора – состояние стены поперек оси x , на второй позиции – стена поперек оси y , на третьей – поперек оси x в обратном направлении, на четвертом, – поперек оси y в обратном направлении.

В результате в левой нижней ячейке находится вектор с координатами $(0, 0, 1, 1)$, то есть из этой ячейки путь открыт в направлении оси y и в направлении оси x . Этот вектор, в свою очередь, можно считать картой ячейки. Так можно составить карту для каждой ячейки лабиринта и затем объединить набор карт в атлас.

В атласе содержится излишняя информация – соседние ячейки имеют одинаковые стенки. Поэтому информация об одной и той же стенке повторяется, если эта стенка не наружная.

Основываясь на атласе лабиринта, можно построить его полную карту. Для этого нужно взять прямоугольную сеть с шагом ячейки лабиринта и в каждый узел сети внести 0, если в этом узле стены нет и единицу, если стена есть. В результате получится двумерный массив нулей и единиц – карта лабиринта. Карта лабиринта по сути – карта препятствий.

Левая нижняя ячейка может быть закодирована как $(0, 0, 1, 1)$, если закодировать остальные ячейки, получится атлас лабиринта в виде матрицы:

$$\begin{array}{cccc}
 (1, 1, 1, 0) & (1, 1, 1, 0) & (0, 1, 1, 0) & (1, 1, 0, 0) \\
 (0, 0, 1, 1) & (0, 0, 0, 1) & (1, 0, 0, 1) & (1, 0, 1, 1)
 \end{array} \rightarrow \begin{array}{cccc}
 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 0 & 1 \\
 0 & 0 & 0 & 0 & \\
 1 & 0 & 0 & 1 & 1 \\
 1 & 1 & 1 & 1 &
 \end{array}$$

Если вместо атласа лабиринта сделать карту в виде матрицы, тогда у матрицы будет 5 строк и в каждой строке либо четыре, либо пять элементов. Первая строка – горизонтальные стенки сверху, вторая строка – вертикальные стенки ячеек верхнего ряда, третья строка – горизонтальные стенки между ячейками первого и второго ряда, четвертая строка – вертикальные стенки ячеек второго ряда и пятая строка – горизонтальные стенки ячеек нижнего ряда.

Карту лабиринта можно составлять разными способами. Один из самых наглядных и простых способов – устроить координатную сеть из половины длины ячейки. На такой карте вертикальные стенки лабиринта – ориентированные вдоль оси ординат (оси y) – отмечать единицами, горизонтальные – ориентированные вдоль оси абсцисс (оси x) – двойками, отсутствие стен

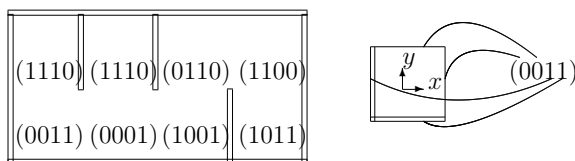


Рис. 6.4: Лабиринт. В каждой ячейке лабиринта приведена карта этой ячейки в виде вектора из четырех чисел. Справа – карта левой нижней ячейки.

– нулями. В результате получится:

2	2	2	2	2	2	2	2	2
1	0	1	0	1	0	0	0	1
1	0	0	0	0	0	0	0	1
1	0	0	0	0	0	1	0	1
2	2	2	2	2	2	2	2	2

6.1.3 Расстояние в городском квартале

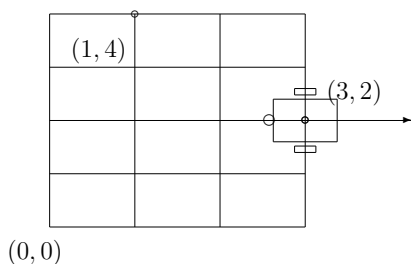


Рис. 6.5: Полигон, разделенный черными линиями, удобен для задания координат, используя число перекрестков.

Рассмотрим пример карты на рисунке 6.5. На такой карте удобно координаты задавать не через расстояние, а используя объекты на самой карте – черные линии. Так для достижения точки $(3, 2)$ нужно пересечь четыре вертикальных линии (начиная с нулевой) и две горизонтальные линии – начиная с нулевой.

Так же можно передвигаться по такой карте, используя известные расстояния между линиями, считая углы поворота колес и повороты робота.

Если двигаться только по черным линиям, то и расстояние естественно измерять по ним же.

Такой метод измерения расстояний по линиям под прямым углом часто используется в математике. Называется он *метрикой в пространстве* l_1 .¹ Иногда в научно-популярной литературе метод измерения расстояний с помощью отрезков под прямым углом называют *манхеттенское расстояние* из-за прямоугольной планировки кварталов на острове Манхеттен в Нью-Йорке.

¹Метрика l_1 была введена математиком Германом Минковским, который придумал и метрику (метод измерения расстояния), использующуюся в специальной теории относительности.

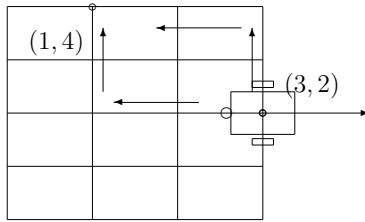


Рис. 6.6: Выбор пути из точки (3,2) в точку (1,4).

Пример, манхеттенское расстояние между точкой O с координатами $(0, 0)$ и точкой A с координатами $(3, 2)$ равно сумме координат точки A : $|OA|_{l_1} = 3 + 2 = 5$. Расстояние между точкой A и точкой $B = (1, 4)$

$$|AB|_{l_1} = |3 - 1| + |4 - 2| = 2 + 2 = 4.$$

Пример. Пусть робот находится в точке $(3, 2)$ и направлен в сторону возрастания первого индекса. Следующий пункт, который робот должен посетить, – точка $(1, 4)$. Один вариант – развернуться и двигаться до первой линии, а затем повернуться на первой линии и ехать по вертикали до четвертой линии. На этом пути робот проедет:

разворот $\rightarrow (3, 2) \rightarrow (2, 2) \rightarrow (1, 2) \rightarrow$ поворот $\rightarrow (1, 3) \rightarrow (1, 4)$.

Разворот можно рассматривать как два поворота. Хотя на самом деле, разворот занимает немного меньше времени, чем два поворота, так как позиционироваться нужно только один раз – в конце разворота, а при двух поворотах – дважды, в конце каждого из поворотов.

поворот \rightarrow поворот $\rightarrow (3, 2) \rightarrow (2, 2) \rightarrow (1, 2) \rightarrow$ поворот $\rightarrow (1, 3) \rightarrow (1, 4)$.

Другой вариант пути:

поворот $\rightarrow (3, 2) \rightarrow (3, 3) \rightarrow (3, 4) \rightarrow$ поворот $\rightarrow (2, 4) \rightarrow (1, 4)$.

Каждый из вариантов пути представляет собой *граф*. Второй граф короче, чем первый. Линейная длина пути будет та же, однако, на первом пути придется развернуться и повернуться, что по времени близко к трем поворотам, а на втором только два раза повернуться. В результате второй путь – оптимальный.

Конечно, путь из точки с координатами $(3, 2)$ в точку $(1, 4)$ можно проложить не по черным линиям, а напрямую. Однако, в этом случае важно точное позиционирование и движение, например, по энкодерам. Путь по линиям с поворотами под прямым углом иногда оказывается короче по времени из-за сравнительной легкости позиционирования.

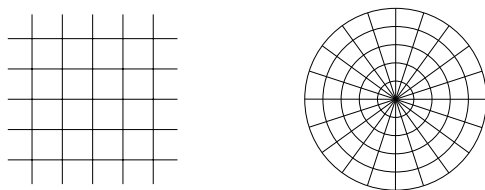


Рис. 6.7: Разбиение на элементарные ячейки – прямоугольники или кольцевые секторы.

6.1.4 Полигональные карты, как векторные поля

Полигональная карта представляет обобщение карты в виде матрицы, построенной для лабиринта. Теперь всю рабочую поверхность можно рассматривать в виде ячеек. Существуют разные подходы к тому, что считать типовой ячейкой.

- В компьютерной графике и в математических задачах моделирования оболочек часто в качестве элементарной ячейки поверхности рассматривается треугольник, так как для определения плоскости, в которой находится треугольник, достаточно только трех точек.
- В задачах робототехники и физического моделирования часто используются прямоугольные (кубические) элементы. Это оказывается естественным из-за того, что сетка с максимальной чувствительностью датчиков или же максимальной точностью передвижения оказывается равномерной прямоугольной.

Шаг детализации карты разумно определять, исходя из двух параметров. Такими параметрами являются:

- чувствительность, разрешаемая способность датчиков, используемых для движения;
- точность позиционирования робота, определяемая приводами – моторами, энкодерами, сервоприводами.

Для выбранного шага всё рабочее пространство, доступное роботу, делится на элементарные ячейки, например, квадраты со сторонами равными шагу детализации или секторы колец, рисунок 6.7.

Карта задает соответствие элемента матрицы с индексами (i, j) и ячейки полигона с координатами $x_i \leq x < x_i + \Delta_x$ и $y_j \leq y < y_j + \Delta_y$. При этом x, y

могут быть координатами в заранее выбранной системе координат. Типичные системы координат – декартова система координат и полярная система координат. Иногда бывают удобны системы координат, определяемые семействами пересекающихся кривых.

В самом простом случае в ячейку матрицы (i, j) можно записывать 0 в случае доступности этой ячейки для перемещения или 1, если в ячейке расположено непреодолимое для робота препятствие.

Вообще говоря, в ячейке матрицы (i, j) можно записывать вектор – набор числовых значений, характеризующих тот или иной параметр рабочей области. В качестве параметров, определяющих ячейку, могут выступать:

- высота над ячейкой с координатой $(0, 0)$, если полигон неплоский;
- угол наклона к горизонту;
- магнитное склонение, если магнитное поле на полигоне неоднородно;
- освещенность, если важно использовать видеокамеру или датчик света;
- уровень отражательной способности поверхности;
- другие параметры, важные для решаемой задачи.

С формальной точки зрения, карта со значениями параметров полигона, записанными в элементах матрицы, задает некоторое поле над двумерной плоскостью \mathbb{R}^2 :

- если элементы матрицы – числа, то есть скаляры, тогда карта определяет отображение $\mathbb{R}^2 \rightarrow \mathbb{R}$, такое отображение называется *скалярным полем*;
- если элементы матрицы сами являются n -мерными векторами – наборами из n чисел, такое отображение $\mathbb{R}^2 \rightarrow \mathbb{R}^n$ называется *векторным полем*.

6.1.5 Векторная карта

Полигональная карта, как правило, занимает довольно много памяти, кроме того, она плохо масштабируется. При увеличении масштаба – разбиении на более мелкие ячейки, появляются ячейки с неопределенным содержимым. При уменьшении масштаба – укрупнении ячеек, появляется излишняя информация, которую необходимо обрабатывать для формирования вектора

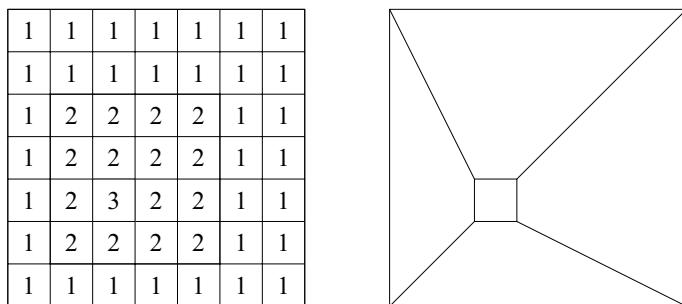


Рис. 6.8: Представление усеченной пирамиды слева на полигональной карте с указанием высот ячеек. Справа – та же усеченная пирамида в векторном формате. Вид сверху.

данных в новой, более крупной ячейке. Чтобы избежать таких дополнительных трудностей, используют векторные карты.

В векторных картах информация хранится не по ячейкам, а в виде объектов. Различные виды объектов имеют разные характеристики и соответственно форматы для их описания.

Пусть на поле установлена горка в виде четырехгранной усеченной пирамиды (рисунок 6.8), на которую может или должен въехать робот. Но горка может иметь разные углы наклона с разных сторон, так, что одни направления для робота могут оказаться непроходимыми, другие – вполне преодолимыми.

На полигональной карте такая горка скорее всего займет несколько ячеек, и, при этом, каждая ячейка будет иметь свой уровень высоты над нулевой точкой полигона. То есть, получится набор ступенек. Если робот не преодолевает ступеньки, то вполне возможно, что он не сможет воспользоваться такой картой для заезда на пирамиду.

На векторной карте горка будет определена как геометрический объект с заданными углами наклона граней и геометрическими размерами основания. Зная геометрические размеры робота, можно определить, какие углы подъема для него являются проходимыми, и выбрать заведомо преодолимый маршрут.

Формально и полигональную карту можно считать векторной, в том смысле, что объектом такой карты является ячейка. Чем больше параметров задано для такой ячейки, например, кроме высоты можно задать еще и вектор градиента наклона поверхности, тем более карта будет похожа на настоящую векторную карту и тем удобнее она оказывается для ориентации

и планирования маршрута.

6.1.6 Упражнения

Упражнение 58 Пусть заданы координаты точек $(1, 2)$, $(5, 8)$, $(-3, 6)$, $(4, 9)$, написать для них векторный взвешенный граф с указанием направлений.

Упражнение 59 Пусть заданы координаты точек $(1, 2)$, $(5, 8)$, $(-3, 6)$, $(4, 9)$, вычислить попарные расстояния между ними, в метрике l_1 .

Упражнение 60 Пусть точка $(1, 2)$ – центр цилиндрического препятствия, радиус цилиндра 3, точки $(5, 8)$ и $(4, 9)$ – вершины диагонали прямоугольного препятствия. Составить карту в прямоугольной системе координат с шагом 1 для области $0, 10$ по горизонтали и $0, 10$ по вертикали.

Упражнение 61 Пусть точка $(1, 2)$ – центр цилиндрического препятствия, радиус цилиндра 3, точки $(7, 8)$ и $(8, 9)$ – вершины диагонали прямоугольного препятствия. Составить карту в полярной системе координат с центром в точке $(5, 5)$, максимальным радиусом 5, с шагом 1 по радиусу и $\pi/6$ по угловой координате.

6.2 Расстояние на карте и кратчайший путь

6.2.1 Длина пути в ячейке карты

Длина пути складывается из суммы длин отрезков, проходящих по ячейкам полигональной карты. Ячейка карты определяется по координатам $(x, y) \in \mathbb{R}^2$. Длина пути по ячейке вычисляется, исходя из проекции отрезка пути на оси координат, dx – проекция пути по ячейке на ось x , и dy – проекция пути на ось y .

Здесь наша задача понять, как вывести формулы для вычисления длины пути в ячейке по заданным проекциям на оси координат.

- Карта представляет собой векторное поле $\vec{F}(x, y)$ над пространством переменных (x, y) . То есть в каждой точке (x, y) определен вектор \vec{F} .
- Размеры ячейки предполагаются бесконечно малыми.

Возможны разные случаи.

- В самом простом случае – путь представляет собой отрезок на горизонтальной плоскости. Тогда квадрат длины пути по ячейке:

$$dl^2 = dx^2 + dy^2.$$

Теперь представим перемещение по ячейке, как вектор перемещения, компоненты которого разложены по координатам:

$$\vec{dl} = (dx, dy).$$

В таком представлении длина вектора $|\vec{dl}|$ – это корень из скалярного произведения вектора на себя:

$$|\vec{dl}| = \sqrt{dx^2 + dy^2}.$$

- Если ячейка лежит на наклонной плоскости, то для вычисления пути необходимо учесть, что при движении по отрезку на ячейке происходит еще и подъем (или спуск) на величину

$$dz = \sqrt{\left(\frac{\partial F}{\partial x} dx\right)^2 + \left(\frac{\partial F}{\partial y} dy\right)^2}.$$

То есть, длина пути увеличивается. Формула для квадрата длины пути примет вид:

$$dl^2 = (dx)^2 + (dy)^2 + \left(\frac{\partial F}{\partial x} dx\right)^2 + \left(\frac{\partial F}{\partial y} dy\right)^2,$$

или

$$dl^2 = \left(1 + \left(\frac{\partial F}{\partial x}\right)^2\right) (dx)^2 + \left(1 + \left(\frac{\partial F}{\partial y}\right)^2\right) (dy)^2.$$

Здесь также можно представить формулу длины как скалярное произведение:

$$|\vec{dl}|^2 = (\vec{dl}, \vec{dl}).$$

Но в этом случае придется обобщить формулу для скалярного произведения векторов $\vec{a} = (a_x, a_y)$ и $\vec{b} = (b_x, b_y)$:

$$(\vec{a}, \vec{b}) = g_{11}a_xb_x + g_{22}a_yb_y,$$

где

$$g_{11} = \left(1 + \left(\frac{\partial F}{\partial x}\right)^2\right),$$

$$g_{22} = \left(1 + \left(\frac{\partial F}{\partial y}\right)^2\right).$$

- Пусть ячейка представляет собой сектор кольца. Координата x направлена по радиусу кольца, координата y – угловая координата. Тогда квадрат длины пути является суммой квадратов проекции вдоль радиуса и тангенциальной проекции (проекции ортогональной к радиальной).

Проекция пути по тангенциальному направлению, ортогональному к радиальному $x dy$, где x – радиус текущей ячейки, dy – изменение угловой величины в радианах. Тогда:

$$|d\vec{l}|^2 = (dx)^2 + (x dy)^2 = (dx)^2 + x^2(dy)^2.$$

Следовательно, в этом случае скалярное произведение в полярных координатах:

$$(\vec{a}, \vec{b}) = g_{11}a_x b_x + g_{22}a_y b_y,$$

где

$$g_{11} = 1, \quad g_{22} = x^2.$$

Здесь диагональные элементы g_{11} и g_{22} – элементы *метрического тензора* в двумерном пространстве. Метрический тензор

$$g = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix}$$

определяет свойства *скалярного произведения* и, соответственно, длины в римановом пространстве:

$$(\vec{a}, \vec{b}) = g_{11}a_1 b_1 + g_{12}a_1 b_2 + g_{21}a_2 b_1 + g_{22}a_2 b_2.$$

Здесь рассматриваются такие метрические тензоры для которых $g_{12} = g_{21} \equiv 0$.²

6.2.2 Длина пути между двумя пунктами на карте

Путь между двумя точками A, B на карте является суммой отрезков пути по ячейкам карты. Обозначим путь между точками A, B как S_{AB} . Тогда длина такого пути может быть представлена в виде суммы путей по ячейкам карты \mathcal{M} :

$$S_{AB} = \sum_{\mathcal{M}} \sqrt{g_{11}(dx)^2 + g_{22}(dy)^2}.$$

²В каждой точке риманова пространства существует локальная система координат, в которой метрический тензор имеет диагональный вид. Доказательство этого утверждения для двумерного риманова пространства можно рассматривать как небольшое упражнение по линейной алгебре. Оно сводится к решению системы четырех линейных уравнений.

Путь удобно представить в параметрической форме:

$$x = x(t), \quad y = y(t).$$

В этом представлении координаты точки A : $x_A = x(t_0)$, $y_A = y(t_0)$, соответственно координаты точки B : $x_B = x(t_1)$, $y_B = y(t_1)$,

$$dx = \dot{x}dt, \quad dy = \dot{y}dt.$$

Теперь удобно представить сумму в виде интеграла:

$$S_{AB} = \int_{t_0}^{t_1} \sqrt{g_{11}(x, y)(\dot{x})^2 + g_{22}(x, y)(\dot{y})^2} dt.$$

Задача о построении кратчайшего пути между двумя точками A и B сводится к задаче о нахождении нижней границы (инфимума) интеграла:

$$\int_{t_0}^{t_1} \sqrt{g_{11}(x, y)(\dot{x})^2 + g_{22}(x, y)(\dot{y})^2} dt \rightarrow \inf.$$

Инфимум – нижняя граница множества. Понятие нижней границы множества обобщает понятие минимума. В случае открытого интервала $(0, 1)$ нижняя граница – 0 не принадлежит множеству. То есть, минимума этот открытый интервал не имеет, но имеет нижнюю границу – инфимум.

Задача о построении кратчайшего пути называется задачей о построении геодезической. По определению *геодезическая* – кратчайшая линия, соединяющая две точки.

- Геодезическая на плоскости – отрезок прямой.
- Геодезическая на сфере – дуга большой окружности.
- Геодезическая на круговом цилиндре – дуга спирали.

Вывод дифференциального уравнения для геодезической основывается на математическом аппарате вариационного исчисления и выходит за рамки изложения в этом тексте. Впрочем, интересующийся читатель может обратиться к фундаментальному труду [29].

6.2.3 Построение кратчайшего пути на плоскости

Для упрощения задачи о построении кратчайшего пути можно рассмотреть задачу о пути на плоскости с препятствием. Препятствие удобно моделировать большими значениями метрического тензора.

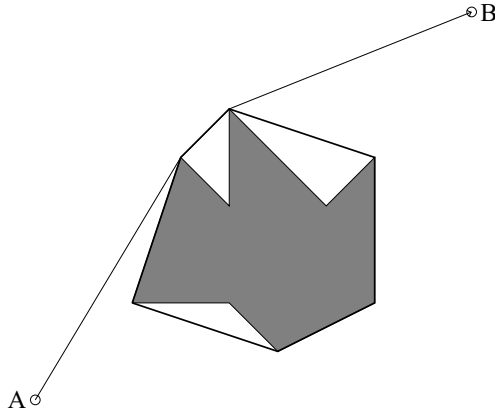


Рис. 6.9: Кратчайший путь на плоскости из точки А в точку В. Препятствие – обозначено серым, обходится по части его выпуклой оболочки.

Например, пусть непреодолимое препятствие для робота находится в точках с координатами

$$x^2 + y^2 < 1.$$

Тогда можно принять $g_{12} = g_{21} \equiv 0$,

$$\begin{aligned} g_{11}(x, y) = g_{22}(x, y) &= 1, & x^2 + y^2 &\geq 1, \\ g_{11}(x, y) = g_{22}(x, y) &= 100, & x^2 + y^2 &< 1. \end{aligned}$$

Очевидно, большое значение компонент метрического тензора приведет к тому, что при построении задачи о кратчайшем пути интеграл, проходящий через препятствие, окажется значительно больше, чем интеграл, обходящий это препятствие.

Пусть имеется препятствие с основанием – множеством M в плоскости движения робота.

Назовем \mathcal{M} выпуклой оболочкой множества M , если $M \in \mathcal{M}$, множество \mathcal{M} выпукло и мера множества \mathcal{M} минимальна для всех выпуклых множеств \mathcal{N} таких, что $M \in \mathcal{N}$ (рисунок 6.9).

Назовем диаметром D препятствия наибольший линейный размер выпуклой оболочки этого препятствия.

Периметр проекции этой выпуклой оболочки препятствия на плоскость движения обозначим P .

Справедливо утверждение:

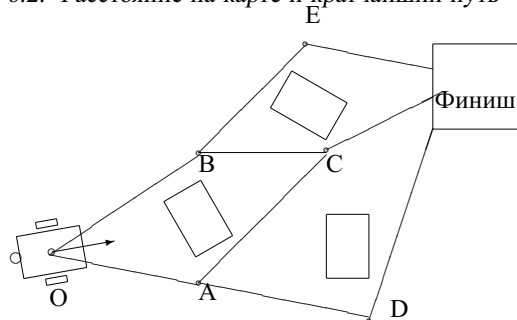


Рис. 6.10: Графы для выбора пути на финиш в обход препятствий.

Утверждение 14 Если $(g_{11} + g_{22})D > P$, тогда условие минимального пути приводит к обходу препятствия извне.

Пусть необходимо построить кратчайший путь на плоскости из точки A в точку B так, что отрезок AB проходит через некоторое препятствие M . Справедливо утверждение:

Утверждение 15 Кратчайший путь из A в B проходит по отрезку касательной из точки A до выпуклой оболочки препятствия M , отрезку касательной из точки B до M и кратчайшей дуги на M от одной точки касания до другой.

6.2.4 Построение пути по взвешенному графу

Построение оптимального пути в случае, если карта представляет собой взвешенный граф, решается алгоритмически.

Оптимальные пути на графах обычно рассматриваются в разделах дискретной математики. Как правило, это различные алгоритмы перебора, такие как волновой алгоритм или алгоритм Дейкстры.

Маршрут робота можно представить в виде ломаной линии с поворотами в вершинах. Для примера рассмотрим на рисунке 6.10 различные маршруты до финиша с объездом препятствий.

Маршруты можно представить в виде нескольких графов:

- (1) $O \rightarrow A \rightarrow D \rightarrow \text{Финиш}$.
- (2) $O \rightarrow A \rightarrow C \rightarrow \text{Финиш}$.
- (3) $O \rightarrow B \rightarrow C \rightarrow \text{Финиш}$.
- (4) $O \rightarrow B \rightarrow E \rightarrow \text{Финиш}$.

В вершинах графа – точках A, B, C, D – робот должен сделать поворот. Ребра графа изображают прямолинейное движение от одной точки маршрута до другой. Для выбора оптимального графа нужно сложить все абсолютные величины поворотов, которые сделает робот при движении по каждому из маршрутов, и сложить длины расстояний, соответствующих каждому ребру графа. В результате для каждого из маршрутов получим пару чисел – суммарный угол поворота на маршруте и суммарное расстояние.

Пусть для примера, для маршрута (1) эта пара $(110, 200)$, для (2) – $(100, 180)$, для (3) – $(60, 170)$, для (4) – $(80, 190)$.

Теперь нужно экспериментально определить, какое время роботу нужно для поворота, например, 10 градусов и на проезд 10 единиц расстояния. Пусть на поворот робот тратит t_1 секунд, а на проезд t_2 секунд. Нетрудно посчитать время, затрачиваемое для каждого маршрута:

$$|(1)| = 11t_1 + 20t_2,$$

$$|(2)| = 10t_1 + 18t_2,$$

$$|(3)| = 6t_1 + 17t_2,$$

$$|(4)| = 8t_1 + 19t_2.$$

Для выбранного примера очевидно, что маршрут (3) окажется самым быстрым, так как и суммарный угол, и суммарное расстояние на нем меньше остальных.

6.2.5 Планирование маршрута

Путь робота по карте можно представить в виде графа – перемещение из точки A в точку B . Действие в точке B , движение к точке C и далее.

Способ выполнения задания может быть различным, поэтому при выполнении задания программа управления роботом должна уметь выбирать оптимальный путь.

Бывает так, что робот должен выполнять некоторые манипуляции и передвижения в окрестности какого либо ориентира. В этом случае удобно в окрестности этого ориентира использовать отдельный кусок карты и, например, координаты на карте записывать в терминах перемещения робота.

Выбор маршрута как правило сводится к задаче оптимизации, да еще и при нечетких критериях.

- Маршрут должен быть кратчайшим:

– по расстоянию;

- по времени прохождения.
- Движение по маршруту должно уверенно контролироваться:
 - сход с маршрута должен легко определяться;
 - положение робота должно быть легко определяемым по ориентирам, если робот сойдет с маршрута.

Бывает, что оптимальный путь построить сложно из-за нечетких правил или изменяющихся условий на самом полигоне. Например, если на полигоне есть объекты, движущиеся по неизвестным программисту, и соответственно роботу, маршрутам. Тогда выбирают максимально удобный и простой путь.

6.2.6 Упражнения

Упражнение 62 На карте с прямоугольной системой координат системой координат с шагом 1 найти длину кратчайшего пути из точки с координатами $(1, 1)$ в точку $(9, 0)$, если имеется препятствие в виде кругового цилиндра с центром основания в точке $(4, 0)$, радиус основания 3.

Упражнение 63 На карте с полярной системой координат с шагом 1 по радиусу и $\pi/18$ по углу найти длину кратчайшего пути из точки с координатами $(1, 1)$ в точку $(9, 0)$. Замечание. В записи координат первая координата – соответствует радиальному направлению, вторая – угловому.

Упражнение 64 На карте с полярной системой координат с шагом 1 по радиусу и $\pi/18$ по углу обозначено препятствие в ячейках с координатами $(5, 0)$, $(6, 1)$, $(6, 2)$, $(7, 3)$. Найти длину кратчайшего пути из точки с координатами $(1, 1)$ в точку $(9, 0)$. Замечание. В записи координат первая координата – соответствует радиальному направлению, вторая – угловому.

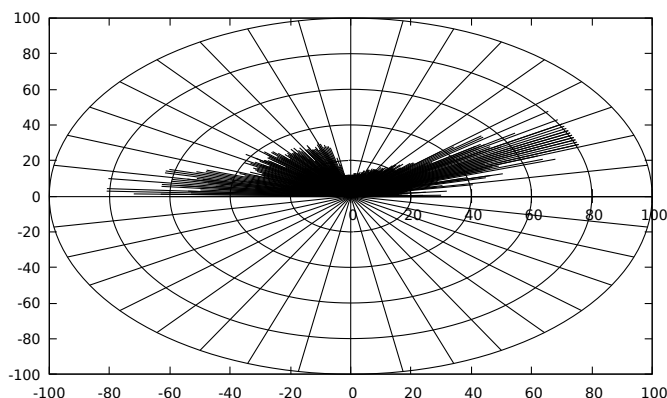


Рис. 6.11: Результат сканирования окрестности робота в диапазоне углов $(0, \pi)$ инфракрасным датчиком расстояния с максимальной дальностью 80 см. Видно, что слева препятствий нет, прямо перед датчиком находится угол препятствия на расстоянии около 15 см. с прямой стенкой, длиной около 20 см, правее еще одно препятствие на расстоянии около 30 см и длиной около 20 см, слева от датчика препятствий нет.

6.3 Построение карты

6.3.1 Локальная карта

Для определенности будем считать, что робот сканирует рабочее пространство с помощью датчика расстояния. В результате сканирования получает расстояние до препятствия r и угол поворота датчика ϕ . Максимальная дальность сканера будет обозначаться R .

Будем называть систему координат, связанную с роботом, подвижной.

После сканирования получится набор отрезков лучей, исходящих из точки полюса, в которой находится датчик во время сканирования. Концы отрезков упрутся в препятствие, либо имеют длину равную максимальной дальности датчика.

Для получения карты возьмем круг, радиус которого равен максимальной дальности определения препятствия датчиком расстояния. Разобьем его на ячейки в виде секторов круговых колец, где шаг по радиусу равен чувствительности датчика расстояния, а шаг по угловой координате равен чувствительности датчика поворота по углу.

Первоначально все ячейки такой карты заполним значением «Nan» –

неопределено.

Набор пар $(r_i, \phi_i)_0^N$ задает множество препятствий в полярной системе координат с центром в точке положения датчика. В ячейки с препятствиями запишем, например, значение «True»

Ячейки карты между сканером и препятствием обозначим как свободные для перемещения, например, поместим в них значение «False».

Для преобразования этой карты в прямоугольную систему координат можно сделать преобразование:

$$\xi_i = r_i \cos(\phi_i), \quad \eta_i = r_i \sin(\phi_i).$$

Начало координат в этой прямоугольной системе совпадает с полюсом в полярной системе координат.

Для формирования карты в прямоугольной системе координат выберем квадрат, вписанный в круг радиуса R . Сторона такого квадрата $a = R\sqrt{2}$.

Пр выборе шага разбиения прямоугольной карты приходится решать задачу оптимизации. Шаг карты удобно выбирать достаточно крупным, чтобы между роботом и препятствиями не образовывались ячейки, через которые не проходит отрезок луча сканера.

Все ячейки, через которые проходит луч сканера, будем считать доступными для робота.

Ячейки карты, в которых заканчиваются отрезки лучей сканера, будем считать препятствиями, ячейки карты за препятствиями будем считать неопределёнными, если они ранее не имели других пометок.

6.3.2 Атлас рабочей области

Полученная с помощью такого сканирования карта обычно неполна. То есть она не охватывает все возможные ячейки рабочего пространства по двум причинам:

- дальность обнаружения меньше, чем размеры рабочего пространства;
- препятствия закрывают рабочее пространство, находящееся за ними.

Для того, чтобы дополнить имеющуюся информацию о рабочем пространстве, можно переместить робота в новую позицию. Позиция должна отвечать следующим условиям:

- сканеру должна быть доступна как можно большая часть ячеек, помеченных «NaN» в декартовой системе координат;

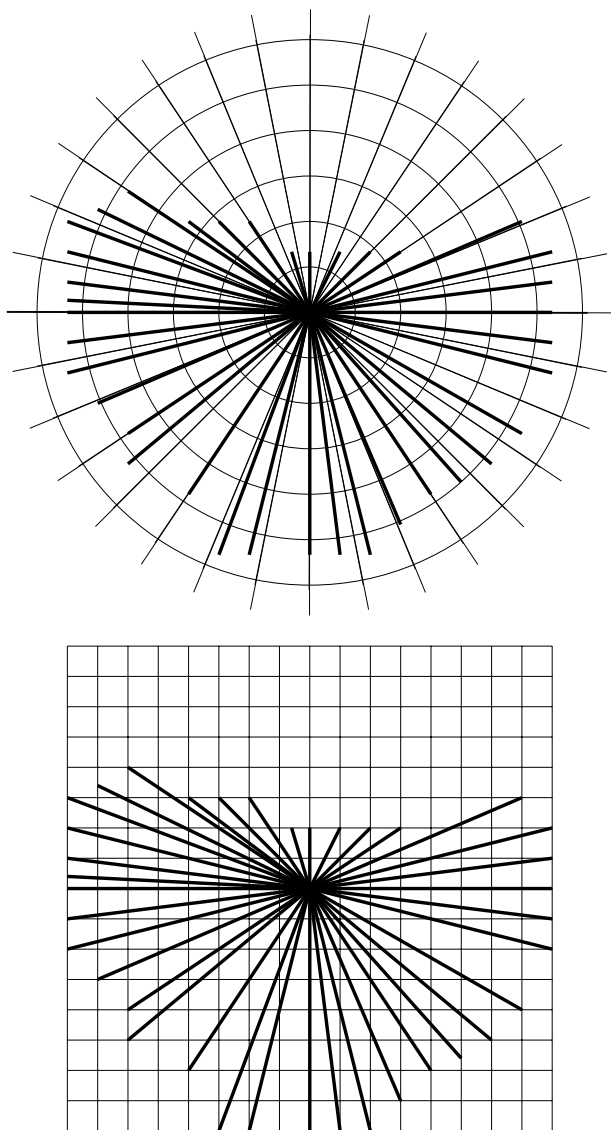


Рис. 6.12: Разбиение полигона на элементарные ячейки – кольцевые секторы или прямоугольники внизу. Ячейки, в которых заканчивается отрезок луча сканирования, заняты препятствием. Ячейки, через которые проходит отрезок луча сканирования, – доступные. Состояние остальных ячеек карты – «не определено».

- дальность сканера должна позволять построить карту в новой позиции так, что ячейки новой карты будут пересекаться с ячейками карты, построенной на предыдущей позиции робота.

После перемещения в новую позицию необходимо опять провести сканирование и построить локальную карту в полярной системе координат. Затем отобразить эту карту в глобальную декартову систему координат:

$$\xi_i^{(1)} = X^{(1)} + r_i \cos(\phi_i), \quad \eta_i^{(1)} = Y^{(1)} + r_i \sin(\phi_i).$$

Здесь $(X^{(1)}, Y^{(1)})$ – новая позиция робота в декартовой системе координат.

Так, действуя последовательно, можно построить набор карт для всей рабочей области.

В результате будет составлен атлас рабочей области из набора M локальных карт в полярной системе координат с набором центральных точек $\{(X^{(k)}, Y^{(k)})\}_{k=0}^M$ в декартовой системе координат.

6.3.3 Преобразование карты в прямоугольной проекции в карту в полярной системе координат

Рассмотрим, как можно получить информацию об окрестности точки с декартовыми координатами (x, y) по атласу рабочей поверхности.

Точка с координатами (x, y) имеется на всех картах атласа, для которых выполнено условие:

$$(X^{(k)} - x)^2 + (Y^{(k)} - y)^2 \leq R^2.$$

Чтобы выяснить, есть ли препятствие в окрестности точки (x, y) , бывает удобно перейти из декартовой в полярную систему координат.

Значение угла в полярной системе координат обычно вычисляется как обратная функция тригонометрического тангенса при вершине угла в точке $(X^{(k)}, Y^{(k)})$ для треугольника $(X^{(k)}, Y^{(k)})$, $(x - X^{(k)}, y - Y^{(k)})$, $(x - X^{(k)}, 0)$.

Обозначим $\xi^{(k)} = x - X^{(k)}$ и $\eta^{(k)} = y - Y^{(k)}$ – локальные декартовы координаты. Для вычисления соответствующих ϕ и r нужно рассмотреть несколько специальных случаев.

- $\xi^{(k)} > 0, \quad \eta^{(k)} \geq 0 \quad \phi = \arctan\left(\frac{\eta^{(k)}}{\xi^{(k)}}\right);$
- $\xi^{(k)} = 0, \quad \eta^{(k)} > 0, \quad \phi = \frac{\pi}{2};$
- $\xi^{(k)} < 0, \quad \eta^{(k)} \geq 0 \quad \phi = \pi + \arctan\left(\frac{\eta^{(k)}}{\xi^{(k)}}\right);$

- $\xi^{(k)} < 0, \quad \eta^{(k)} = 0 \quad \phi = \pi;$
- $\xi^{(k)} < 0, \quad \eta^{(k)} < 0 \quad \phi = \pi + \arctan\left(\frac{\eta^{(k)}}{\xi^{(k)}}\right);$
- $\xi^{(k)} = 0, \quad \eta^{(k)} < 0 \quad \phi = 3\pi/2;$
- $\xi^{(k)} > 0, \quad \eta^{(k)} < 0 \quad \phi = 2\pi + \arctan\left(\frac{\eta^{(k)}}{\xi^{(k)}}\right).$

$$r = \sqrt{(\xi^{(k)})^2 + (\eta^{(k)})^2},$$

Таким образом от карты в прямоугольной проекции можно перейти к карте в полярной системе координат.

6.3.4 Упражнения

Упражнение 65 На векторной карте с прямоугольной системой координат с шагом 1 имеется объект в виде кругового цилиндра с центром основания в точке $(4, 0)$, радиус основания 3. Привести эту карту в полигональную карту в прямоугольной системе координат с шагом 2.

Упражнение 66 На векторной карте с прямоугольной системой координат с шагом 1 имеется объект в виде кругового цилиндра с центром основания в точке $(4, 0)$, радиус основания 3. Перевести эту карту в полигональную карту в полярной системе координат с шагом по радиусу 1 и шагом по угловой переменной $\pi/18$.

Упражнение 67 Имеется две локальные карты в полярной системе координат. Карта 1 с шагом по радиусу 1 и максимальным радиусом 5, с шагом по угловой переменной $\pi/18$. Карта получена круговым сканированием. При сканировании найдено препятствие. Координаты препятствия $(3, 0), (3, 1), (3, 2), (4, 3)$. Карта 2 получена круговым сканированием после перемещения робота в направлении $\pi/5$ на расстояние 4 в координатах карты 1. При сканировании на карте 2 отмечено препятствие с координатами $(5, 32), (4, 33), (4, 34), (5, 35)$. На основании карт 1 и 2 построить карту в прямоугольной системе координат и нанести на ней обнаруженные препятствия.

6.4 Локализация

6.4.1 Определение расстояний до препятствий

Пусть робот имеет возможность определять расстояние и угловое направление до препятствия. Область сканирования ограничена либо размерами рабочей области доступной роботу, либо дальностью действия сканера. Детализация сканирования определяется погрешностями сканера по расстоянию и погрешностью позиционирования сканера по угловой координате.

Для простоты будем считать, что препятствия на карте робот определяет как точки.

Если препятствие нельзя представить как точку, то есть сканер робота может это препятствие фиксировать как протяженное, тогда крайние по угловой переменной точки препятствия можно рассматривать, как две отдельных точки.

В этом случае препятствие будет определяться как набор точек

$$\{(x_i, y_i)\}_{i=1}^n,$$

где x_i – расстояние от робота до препятствия y_i – угловое направление на препятствие.

6.4.2 Определение положений по трем найденным ориентирам

Верно следующее утверждение.

Утверждение 16 *Для определения координат объекта на плоскости необходимо и достаточно определения расстояний от объекта до трех точек плоскости с заранее известными координатами.*

Пусть расстояние от робота до точки с координатами (x_0, y_0) равно R_0 . Это означает, что координаты робота лежат на окружности

$$(x - x_0)^2 + (y - y_0)^2 = R_0^2.$$

Если расстояние до точки с координатами (x_1, y_1) равно R_1 . Это означает, что координаты робота лежат на окружности

$$(x - x_1)^2 + (y - y_1)^2 = R_1^2.$$

Эти две окружности пересекаются в двух точках. Поэтому для однозначного определения координат робота необходимо расстояние до третьего ориентира $(x_2, y_2) - R_2$. Это означает, что координаты робота лежат на

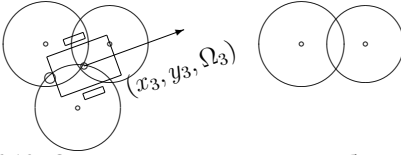


Рис. 6.13: Определение координат робота по трем расстояниям до трех ориентиров. Справа показано, что по расстояниям до трех точек можно определить координаты робота. Рисунок слева показывает, что если две окружности пересекаются, то, как правило, в двух точках. Поэтому по известным расстояниям до двух ориентиров и без дополнительных условий нельзя однозначно определить положение робота.

окружности

$$(x - x_2)^2 + (y - y_2)^2 = R_2^2.$$

Из этих формул легко получить систему линейных уравнений для определения координат робота. Действительно, вычтем из первого уравнения второе. В результате получим линейное уравнение с двумя неизвестными x, y :

$$2x(x_1 - x_0) + x_0^2 - x_1^2 + 2y(y_1 - y_0) + y_0^2 - y_1^2 = R_0^2 - R_1^2.$$

Второе линейное уравнение можно получить, если вычесть из второго уравнения третье:

$$2x(x_2 - x_1) + x_1^2 - x_2^2 + 2y(y_2 - y_1) + y_1^2 - y_2^2 = R_1^2 - R_2^2.$$

Тогда система линейных уравнений:

$$\begin{aligned} 2x(x_1 - x_0) + 2y(y_1 - y_0) &= F_1, \\ 2x(x_2 - x_1) + 2y(y_2 - y_1) &= F_2, \\ F_1 &= R_0^2 - R_1^2 + x_1^2 - x_0^2 + y_1^2 - y_0^2, \\ F_2 &= R_1^2 - R_2^2 + x_2^2 - x_1^2 + y_2^2 - y_1^2. \end{aligned}$$

Здесь в F_1 и F_2 объединены несколько слагаемых для краткости записи формул. Решение этой системы уравнений существует, если определитель матрицы не равен нулю:

$$W = 4(x_1 - x_0)(y_2 - y_1) - 4(x_2 - x_1)(y_1 - y_0) \neq 0.$$

Если определитель близок к нулю, например, например если величина $|W|$ близка к погрешности определения расстояния датчиком, то матрица считается *плохо обусловленной*. В таком случае вычисления координат робота

по выбранным ориентирам доверять не стоит. Плохо обусловленная матрица может получиться, если две из окружностей на рисунке 6.13 касаются друг друга. В этом случае можно:

- Второе уравнение получить, вычитая из уравнения первой окружности уравнение для третьей окружности.

Если и опять получается плохо обусловленная матрица, тогда:

- либо поменять хотя бы один ориентир из выбранной тройки,
- либо роботу переехать в другую позицию.

Если $|W|$ существенно отличается от нуля, тогда координаты можно вычислить по формулам:

$$x = \frac{2F_1(y_2 - y_1) - 2F_2(y_1 - y_0)}{4(x_1 - x_0)(y_2 - y_1) - 4(x_2 - x_1)(y_1 - y_0)},$$

$$y = -\frac{2F_2(x_1 - x_0) - 2F_1(x_2 - x_1)}{4(x_1 - x_0)(y_2 - y_1) - 4(x_2 - x_1)(y_1 - y_0)}.$$

Таким образом, воспользовавшись датчиком расстояния и тремя ориентирами с известными координатами в пределах прямой видимости, можно определить координаты робота. Именно на этом принципе – *триангуляции*³ основано определение положения на карте с помощью сигналов систем ГЛОНАСС и GPS – достаточно трех видимых спутников для определения положения на поверхности Земли.

Важное замечание. Здесь из трех уравнений окружностей выведена система двух линейных уравнений. На самом деле таких уравнений три. Из-за того, что расстояния до ориентиров определяются с некоторой погрешностью и из-за ограничения в точности измерений датчика, уравнения нам известны неточно. Как правило, оказывается так, что полная система из трех уравнений точного решения не имеет. Здесь использован самый простой способ получения значений координат x, y , которые минимизируют невязку этой системы уравнений. Есть и другие способы. При их применении получим другие значения координат (x, y) . Если матрица не является *плохо обусловленной*, то эти координаты оказываются близки друг к другу.

Не всегда можно определить все координаты робота однозначно.

Пример 31 Рассмотрим элементарный пример. Пусть робот провел круговое сканирование и определил расстояние до препятствий. Результат сканирования приведен на графике 6.14.

³Метод триангуляции использовался голландцем Снеллиусом (1617 год). Позднее (1820-22 годы) системный математический подход к триангуляции земной поверхности разрабатывался немцем Карлом Гауссом. В частности, он отмечал, что сумма углов треугольника на земной поверхности больше 180 градусов.

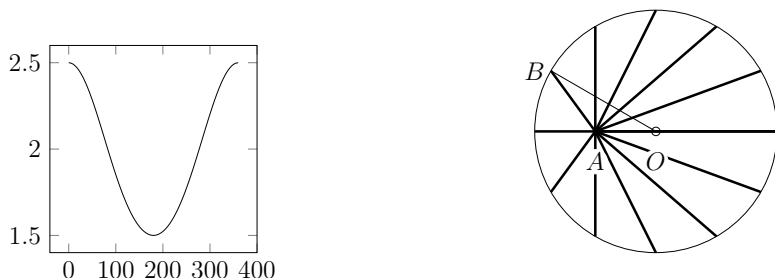


Рис. 6.14: Слева – развертка данных, полученных при круговом сканировании. По оси абсцисс откладывается угловая величина в градусах. На оси ординат откладывается условная единица длины – расстояние от датчика до стены. Справа – представление на плоскости. Точка А – точка, в которой находится сканирующий датчик. Точка О – центр окружности.

Зная карту, легко определить координаты. При повороте на угол y_i во время кругового робота, центр круга (карты) и препятствие на которое падал луч сканера, образовывали треугольник. Угол между кратчайшим направлением на препятствие и направлением к центру круга равнялся $\pi - y_i$.

Обозначим $a = |OA|$ – расстояние от робота до центра круга, тогда вследствие теоремы косинусов получим:

$$a^2 + x^2 + 2ax \cos(y) = R^2.$$

В этом уравнении известны R – радиус круга – это вся доступная область для робота по карте; x – расстояние до препятствия (стены); y – угол поворота робота при сканировании. Величину a – расстояние от робота до центра круга легко вычислить, решив квадратное уравнение. Это вычисление даст положение робота на карте – расстояние до центра круга. Более точная локализация – определение угла поворота внутри круга невозможна из-за симметрии карты.

Нетрудно рассмотреть и другие случаи симметричного расположения препятствий, для которых определить все координаты робота не удастся.

6.4.3 Преобразование координат в процессе локализации робота

Для ориентации и локализации в пространстве робот может построить локальную карту. Существует много способов строить локальную карту – от

совместного использования датчиков касания и счисления пути, до использования лидаров и стереозрения. Каждый из способов требует своего математического аппарата. Скорее всего самые простые математические преобразования требуются при построении карты с помощью лидара.

6.4.4 Использование лидара для построения локальной карты

В этом разделе будем считать, что робот сканирует пространство в плоскости, поворачиваясь на определенный угол. Результатом измерений являются расстояние до препятствия x_i и текущее значение угла y_i . Набор пар $(x_i, y_i)_0^N$ задает множество препятствий в полярной системе координат с центром в точке положения датчика. Будем называть эту систему координат подвижной, как она перемещается вместе с датчиком.

Предположим, что имеется карта в некоторой неподвижной системе координат. Кроме того, на этой карте имеется область, в которой нанесены те же препятствия, что обнаружены роботом при построении карты в подвижной системе координат.

Пусть карта в неподвижной системе координат – прямоугольная ортогональная. Тогда для локализации робота координаты неподвижной карты нужно перевести в полярные координаты с центром в произвольной ячейке (x_i, y_i) .

Значение угла в полярной системе координат обычно вычисляется как обратная функция тригонометрического тангенса при вершине угла в точке $(0, 0)$ для треугольника $(0, 0)$, (x, y) , $(x, 0)$.

Здесь могут быть несколько специальных случаев.

- $x > 0, \quad y \geq 0 \quad \phi = \arctan\left(\frac{y}{x}\right);$
- $x = 0, \quad y > 0, \quad \phi = \frac{\pi}{2};$
- $x < 0, \quad y \geq 0 \quad \phi = \pi + \arctan\left(\frac{y}{x}\right);$
- $x < 0, \quad y = 0 \quad \phi = \pi;$
- $x < 0, \quad y < 0 \quad \phi = \pi + \arctan\left(\frac{y}{x}\right);$
- $x = 0, \quad y < 0 \quad \phi = 3\pi/2;$
- $x > 0, \quad y < 0 \quad \phi = 2\pi + \arctan\left(\frac{y}{x}\right).$

$$x' = \sqrt{(x - x_i)^2 + (y - y_i)^2}, \quad y' = \phi,$$

Затем, последовательно поворачивая карту на угол разрешения датчика, следует сравнивать полученную после преобразования неподвижную карту с картой, полученной роботом после сканирования окружения.

Задача локализации состоит в том, чтобы совместить карту в подвижной системе координат с картой в неподвижной системе и определить на карте в неподвижной системе координат препятствия, обнаруженные при составлении карты в подвижной системе.

Если совпадений, то есть вариантов текущего положения робота, окажется несколько, тогда необходимо запомнить все позиции робота на неподвижной карте. Затем нужно определить перемещение робота, после которого локализация даст наименьшее количество из возможных вариантов локализации робота.

Далее, переместить робота на новую позицию, просчитать это перемещение для всех пригодных вариантов на подвижной карте и затем опять провести процедуру сканирования окружения и локализации, считая, что новая позиция робота соответствует перемещению из одной из позиций, выбранных ранее. Выполняя такие перемещения последовательно, можно определить положение робота.

6.4.5 Упражнения

Упражнение 68 Пусть известны координаты маяков $(0, 0)$, $(10, 0)$, $(0, 10)$. Робот определил расстояния до маяков как 4.5; 2.5 и 3.5 соответственно. Вычислить координаты робота.

Упражнение 69 Пусть известны координаты маяков $(0, 0)$, $(10, 0)$, $(0, 10)$. Робот определил расстояния до маяков как 4.5; 2.5 и 3.5. Вычислить координаты робота.

Упражнение 70 Пусть известны координаты маяков $(0, 0)$, $(10, 0)$, $(0, 10)$. Указать области рабочего пространства, в которых локализация невозможна. Предполагается, что робот не может передвигаться в пространстве, но может производить круговое сканирование.

Указание. Маяки могут перекрывать друг друга.

Упражнение 71 *В трехмерном рабочем пространстве манипулятора известны координаты маяков в прямоугольной системе: $(0, 0, 0)$, $(10, 0, 0)$, $(0, 10, 0)$, $(0, 0, 10)$. Робот определил расстояния от рабочего инструмента до маяков как 7, 7, 9.5 и 8.4 соответственно. Вычислить координаты рабочего инструмента робота.*

6.5 Обзор литературы

Задачи построения карты и локализации мобильных роботов активно разрабатываются. В сети интернет можно найти учебные курсы различного уровня сложности.

Список доступных работ по этой теме можно найти на сервере препринтов arxiv.org в разделе `robotics`.

В частности, в препринте [34] сделан большой обзор различных подходов к построению карты и локализации. Есть обсуждение открытых задач.

Глава 7

Интеллектуальные системы и методы минимизации.

В этой главе рассмотрены несколько типичных подходов к построению программ, использующихся при анализе текущей информации и принятии решений.

- Цепи Маркова используются для статистического анализа и формирования последовательностей, в той или иной мере похожих на часто встречающиеся в проанализированных данных. Их можно использовать, например,
 - при конструировании чат-ботов;
 - автоматическом генерировании подсказок при наборе текста;
 - генерировании музыки в заданном стиле.
- Деревья решений используются при построении графов последовательных действий. Подобные деревья могут быть полезны, например,
 - для ботов технической поддержки;
 - при выборе последовательности шагов в ситуациях с конечным набором вариантов действий.
- Искусственные нейронные сети (ИНС) оказываются полезны при распознавании образов, при решении задач классификации. ИНС часто используют
 - при распознавании лиц;
 - определении местоположения;

- в задачах о поиске информации;
- при обработке изображений в беспилотных автомобилях.

Создание подобных программ требует обработки больших объемов информации и решения задач минимизации. Для деревьев решений это задачи минимизации информационной энтропии, для задач о построении ИНС – задачи обучения. Поэтому в главе большую часть занимает обсуждение методов минимизации:

- методы поиска локальных минимумов:
 - метод наискорейшего координатного спуска;
 - метод наискорейшего градиентного спуска.
- кратко обсуждаются методы поиска глобальных минимумов:
 - метод Монте-Карло;
 - полет Леви;
 - генетические алгоритмы.

7.1 Цепь Маркова. Построение моделей. Примеры применения.

7.1.1 Примеры марковских цепей

Один из часто рассматриваемых процессов в робототехнике – движение робота по линии. Положение робота относительно черной линии определяется по показаниям датчика линии. Датчик возвращает освещенность плоскости, по которой движется робот. Линия широкая, и датчик обычно находится в малой окрестности границы черной линии. Чем больше робот заезжает датчиком на линию, тем меньше показания освещенности, которые определяются датчиком и, наоборот, чем дальше робот отворачивает от линии, тем выше освещенность. Датчик возвращает контроллеру показания освещенности, и по этим показаниям система управления регулирует скорость вращения моторов робота.

Показания датчика освещенности бывают сильно зашумлены. Получаемую информацию об освещенности можно считать случайной величиной с некоторой функцией распределения вероятности.

В результате оказывается, что система управления формирует команды моторам в соответствии со случайно распределенной величиной, получаемой от датчика. Таким образом, движение робота можно рассматривать как функцию случайной величины.

Теперь можно формализовать полученную модель движения. В момент времени t_i робот находится в положении x_i . Имеется случайная величина a_i – показание датчика. Её использует система управления для перевода робота из положения x_i в положение x_{i+1} . Таким образом, имеется отображение состояния x_i в состояние x_{i+1} , причем вероятность перехода от i -го состояния в $i + 1$ определяется по некоторому закону распределения случайной величины, так что вероятность перехода $P(x_i|x_{i+1})$ зависит от состояния робота x_i в текущий момент.

Другой пример похожего процесса. Пусть состояние объекта принадлежит множеству слов русского языка и знаков препинания. Известна вероятность следования элементов множества из этих слов и знаков препинания. Можно последовательно реализовать цепочку элементов этого множества (или состояний), то есть последовательность из слов и знаков препинания. Здесь опять наблюдается последовательность состояний, в которой каждое последующее с некоторой вероятностью определяется из предыдущего.

Последовательность состояний, в которой каждое следующее состояние с некоторой вероятностью определяется только текущим состоянием называется *цепью Маркова*.

7.1.2 Формализация определения цепи Маркова

Для получения математической модели цепи Маркова множество состояний процесса можно пронумеровать натуральными числами, например, от 1 до N . Обозначим вероятность перехода из состояния i в состояние j как P_{ij} . В результате получится квадратная матрица. В этой матрице элементы i -й строки – вероятности перехода из i -го состояния в j . Так как элементы матрицы перехода определяют вероятность, то все они должны быть не меньше нуля. Кроме того, сумма вероятностей перехода из состояния i во все другие состояния равна единице. Поэтому и сумма всех элементов строки матрицы равна единице. Матрица с такими свойствами называется *стохастической*.

По определению в цепи Маркова следующее состояние зависит только от текущего, однако это ограничение легко снимается, если текущее состояние характеризовать не одним числом, а набором чисел – вектором. При этом в состав вектора текущего состояния можно включить часть элементов вектора предыдущего состояния. В результате получится марковская цепь с памятью. Если в вектор текущего состояния включать только один

из элементов вектора предыдущего состояния, то такая марковская цепью называется цепью первого порядка. Если включать элементы из более ранних состояний, то получится марковская цепь высшего порядка – второго, третьего или выше.

7.1.3 Построение стохастической матрицы

Для построения стохастической матрицы переходов в марковской цепи обычно используется какая-либо последовательность элементов множества. Например, для построения матрицы переходов можно воспользоваться каким либо текстом. В этом случае марковская цепь будет определять вероятность следования одного слова за другим. В результате получится матрица, которая определяет, какова вероятность следования за текущим словом других слов из анализируемого текста.

Также можно построить матрицу переходов одной ноты за другой в каком-либо музыкальном произведении. Последовательная реализация переходов от одного состояния к другому даст последовательность нот. Так марковская цепь, основанная на матрице переходов между нотами, будет генерировать нотную последовательность. Чем больший порядок такой марковской цепи, тем более узнаваемыми окажутся фрагменты цепи, и тем ближе сгенерированная последовательность нот окажется к исходному музыкальному произведению.

Пример 32 *Пример – первая строка песенки «Маленькой ёлочке холодно зимой». Ноты: соль, ми, ми, соль, ми, ми, соль, фа, ми ре, до. Число нот – 5, значит матрица 5×5 . Пронумеруем ноты до-1, ре-2, ми-3, фа-4, соль-5. матрица переходов:*

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1/5 & 2/5 & 0 & 2/5 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2/3 & 1/3 & 0 \end{pmatrix}.$$

Нота «до» последняя в рассмотренном отрезке и после нее нет продолжения ни внутри отрезка, ни после. Однако, если оставить в первой строке нули, то матрица формально не будет стохастической. Для того, чтобы соответствовать формальному признаку стохастичности можно замкнуть, например, ноту «до» на себя – поставить 1 в первой строке первого столбца.

7.1.4 Вероятность перехода между состояниями за N шагов

Пусть есть состояние i . С какой вероятностью можно перейти из этого состояния в состояние j за один шаг? Ответ на этот вопрос дает элемент матрицы P_{ij} . С какой вероятностью можно перейти из состояния i в состояние j за два шага через состояние k ? Ответ на этот вопрос дает формула перемножения вероятностей:

$$p = P_{ik}P_{kj}.$$

Методом индукции эту формулу легко обобщить на любые N шагов, для заранее заданной последовательности.

Если нужно посчитать вероятность перехода из i в j за два шага и при этом не важны переходы внутри этих шагов, то вероятность перехода определяется суммой вероятностей всех возможных последовательностей 2 шагов, с начальным состоянием i , конечным – j . Эта вероятность:

$$p_{i,j}(2) = \sum_{k=1}^N P_{i,k}P_{k,j}.$$

Согласно определению умножения матриц это элемент i -й строки и j -го столбца матрицы, полученной умножением P на себя, то есть матрицы P^2 . Тогда по математической индукции можно получить, что переход от состояния i в состояние j за N шагов дает элемент ij матрицы P^N .

Реализацией марковской цепи называется последовательность состояний марковской цепи, полученная при последовательной реализации состояний цепи в соответствии вероятностями переходов.

Пример 33 Вычислим вероятность того, что марковская цепь из примера 32 «сыграет» (реализует) первые две строки из песенки «Маленькой елочке холодно зимой».

$$P(5, 3, 3, 5, 3, 3, 5, 4, 3, 2, 1) = P_{5,3}P_{3,3}P_{3,5}P_{5,3}P_{3,3}P_{3,5}P_{5,4}P_{4,3}P_{3,2}P_{2,1} = \\ \frac{2}{3} \frac{2}{5} \frac{2}{5} \frac{2}{3} \frac{2}{5} \frac{2}{5} \frac{1}{3} \frac{1}{5} 1 = \frac{2^6}{3^3 5^4} \sim 0,00379.$$

Если бы «играла» не марковская цепь, а мячик на клавишах пианино, который с равной вероятностью мог бы попадать на соответствующие клавиши, тогда вероятность извлечь звуки первых двух строк песенки равнялась бы

$$(1/5)^{11} \sim 2,048 \cdot 10^{-8}.$$

7.1.5 Применение марковских цепей

Марковские цепи применяются в физике, например, при изучении броуновского движения, в биологии – при исследовании эволюции ДНК, в лингвистике – при исследовании построения предложений, в распознавании речи.

Кроме того, на основе марковских цепей можно стоять чат-боты, генераторы музыкального сопровождения, например, для компьютерных игр, подсказки при наборе текстов, основанные на статистически обработанном набранном куске текста.

При реализации марковской цепью длинной последовательности может возникнуть цикличность – могут образовываться повторяющиеся последовательности состояний, из которых нет переходов в состояния вне этой последовательности. Существование таких последовательностей является следствием так называемой теоремы о предельных вероятностях [13], [15]. Это существенно снижает область применения марковских цепей при построении систем с элементами искусственного интеллекта.

7.2 Дерево решений и минимизация энтропии

Самый простой способ сделать что-либо разумное – заполучить инструкцию и прямо ей следовать. Именно такому поведению учат в школе. Особенно жестко придерживаться инструкции требуют учителя математики и языков, не важно, каких – русского, английского или языка си.

Инструкции для языка си – простые, для математики – сложнее, для естественных языков – очень сложные. Для таких языков приходится вводить дополнительный предмет – изучать литературу. Потому что часть инструкций даже лингвисты до сих пор не научились формализовывать. Приходится изучать примеры – литературные произведения.

По такому же пути идут программисты при разработке программ. Формирование дерева решений – один из эффективных подходов для алгоритмов, используемых в различных роботизированных устройствах. Также они могут использоваться в программах – автоответчиках, например, в технической поддержке.

7.2.1 Дерево решений для игры в «крестики-нолики»

Один из примеров, где может использоваться дерево решений, – различные логические игры, которые могут быть просчитаны полностью. К таким довольно простым играм относится игра «крестики-нолики» в ее классическом варианте – на поле размером 3×3 .

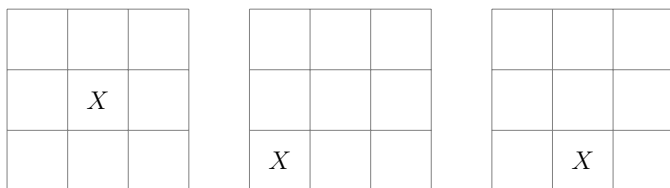


Рис. 7.1: Варианты первого хода в игре крестики-нолики.

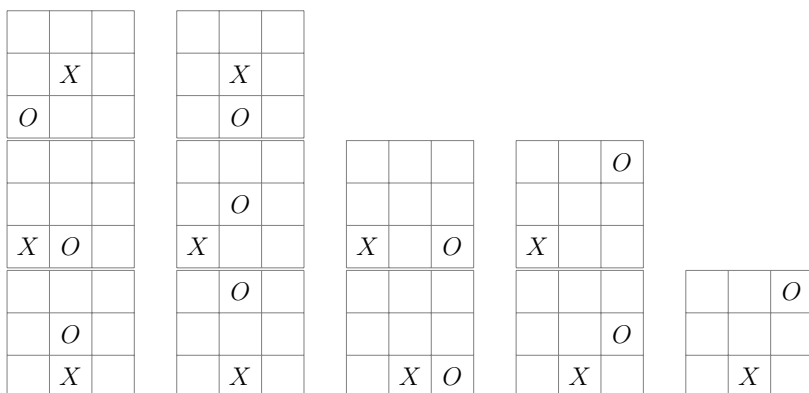


Рис. 7.2: Возможные варианты расстановки на поле перед третьим ходом в игре крестики-нолики с учетом симметрии.

Пусть первый шаг делает робот. Сколько вариантов ходов для него имеется?

Всего клеток для игры 9. Однако, если учесть симметрию, оказывается вариантов всего три, рисунок 7.1.

- 1 Постановка крестика в центр.
- 2 Постановка крестика в угол.
- 3 Постановка крестика в середину боковой стороны.

Предположим, что один из трех вариантов выбран и сделан.

После ответного хода соперника ситуация на поле изменилась и опять нужно рассмотреть все возможные варианты, рисунок 7.2.

Пока неизвестно, какой ход приведет к успеху, нужно опять рассмотреть все возможные варианты.

Если действовать последовательно, то получится граф – *дерево возможностей*, которое состоит из всех возможных позиций в игре «крестики-нолики».

Известно, что при оптимальной стратегии обоих игроков игра сведется вничью [31]. Поэтому после построения дерева возможностей следующая задача – задача классификации. Необходимо отбросить ветви дерева, которые ведут к поражению.

Затем в каждой позиции выбирать ход, продолжение которого содержит наибольшее количество выигрышных ветвей.

Один из способов реализовать дерево решений с помощью спичечных коробков был предложен Дональдом Митчи [32]. Краткий обзор этой статьи и обсуждение можно посмотреть в книге М.Гарднера [33].

Идея Митчи заключается в том, что можно довольно простым способом «обучить» набор спичечных коробков «играть» в крестики-нолики.

Для этого нужно каждой уникальной позиции сопоставить один коробок. Утверждается, что всего существует 300 позиций, соответственно понадобится 300 коробков.

В каждой позиции существует несколько вариантов продолжения. Каждому варианту продолжения игры (ходу) нужно сопоставить бусину или пуговку определенного цвета.

Здесь нет смысла пересказывать игру, замечательно описанную Гарднером. В конце-концов, добавляя бусины «сыгравших» цветов в случае победы коробков и, убавляя в случае поражений, можно «обучить» коробки выигрывать, либо сводить вничью все матчи. То есть получить обученное дерево решений для игры в крестики-нолики.

7.2.2 Классификация и энтропия

Пусть имеется множество из N объектов – бусинок, вариантов ходов или просто набор чисел. Часть из них может совпадать. В этом множестве есть M типов: n_i объектов a_i -го типа, так, что:

$$N = \sum_{i=1}^M n_i.$$

Пусть последовательно извлекаются объекты из множества. В результате получается последовательность $\{a_{k_1}, \dots, a_{k_N}\}$ – перестановка содержащая все элементы множества. Возможное число таких последовательностей равно числу перестановок:

$$K = \frac{N!}{n_1! \cdot n_M!}.$$

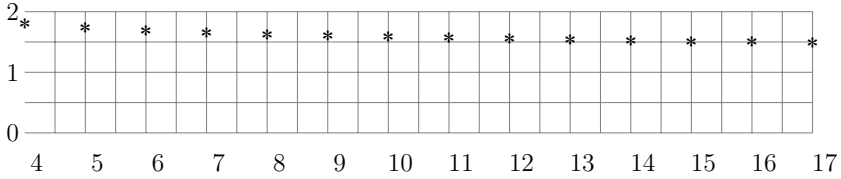


Рис. 7.3: На графике «*» отмечены значения отношения $n \log_2(n) / \log_2(n!)$. На оси абсцисс указаны значения переменной n . Можно заметить, что при возрастании числа n точки, отмеченные «*», медленно приближаются к единице.

На один элемент множества приходится

$$S = \frac{1}{N} \log_2(K) = \frac{1}{N} \left(\log_2(N!) - \sum_{i=1}^M \log_2(n_i!) \right)$$

бит информации.

Величина S называется *энтропией*.

Теперь рассмотрим случай когда $N \rightarrow \infty$. Будем считать, что при этом все $n_i \rightarrow \infty$, так что частота появления a_i в выборке остается той же n_i/N .

Здесь придется воспользоваться следствием из формулы Муавра-Стирлинга (рисунок 7.3):

$$\log_2(z!) \sim z \log_2(z), \quad z \rightarrow \infty.$$

Здесь « \sim » означает *асимптотическую эквивалентность* левой и правой частей при $z \rightarrow \infty$

- Функции $f(x)$ и $g(x)$ *асимптотически эквивалентны* при $x \rightarrow x_*$, если:

$$\lim_{x \rightarrow x_*} \frac{f(x)}{g(x)} = 1.$$

Так что для S получится:

$$\begin{aligned} S &\sim \frac{1}{N} \left(N \log_2(N) - \sum_{i=1}^M n_i \log_2(n_i) \right) = \\ &= - \sum_{i=1}^M \frac{n_i}{N} \log_2 \left(\frac{n_i}{N} \right), \quad N \rightarrow \infty. \end{aligned}$$

При $N \rightarrow \infty$ частота случайного извлечения из множества объекта a_i по определению равна вероятности извлечения такого объекта:

$$p_i = \frac{n_i}{N}.$$

Теперь при переходе к пределу $n \rightarrow \infty$ получится предельная формула для энтропии:

$$S = - \sum_{i=1}^M p_i \log_2(p_i).$$

Величина, посчитанная по этой формуле, называется *энтропией Шенона*.

Задача построения дерева решений состоит в том, чтобы по мере продвижения по дереву уменьшать энтропию имеющегося множества.

Исходя из формулы для энтропии наиболее эффективным с точки зрения уменьшения оказывается выделение элементов a_i из множества для которых:

$$n_i = \max_{i=j} (n_j), \quad j \in \{1, \dots, M\}.$$

Действуя последовательно, можно наиболее эффективным способом уменьшать энтропию множества на каждом узле дерева решений.

Для неограниченного n дерево решений необходимо строить так, чтобы отделять наиболее вероятные возможности в результате шагов по дереву решений.

Пример 34 *Предположим, есть заказ от интернет-провайдера на работа – текстового автоответчика на сайте компании для поддержки пользователей.*

Известно, что пользователь готов ответить примерно на 4-5 вопросов в режиме вопрос-ответ, дальше он теряет терпение в общении с чат-ботом. Это означает, что есть возможность сформировать дерево решений глубиной в пять запросов для ответов «да» или «нет».

Статистика запросов к провайдеру говорит о том, что в 70% случаев связь с провайдером теряется из-за отключения абонента при окончании оплаченного трафика. Ещё 15% из-за зависания маршрутизатора у клиента, 10% из-за потери контакта сетевого кабеля между маршрутизатором и компьютером клиента, и только оставшиеся 5% требуют вмешательства специалиста компании.

Тогда при построении дерева выгодно классифицировать случаи с большой вероятностью в начале опроса. Дерево изображено на рисунке 7.4.

Рассмотрим, что происходит с энтропией при таком подходе. При этом вероятности каждой из возможных неполадок: $p_1 = 0.7$, $p_2 = 0.15$,

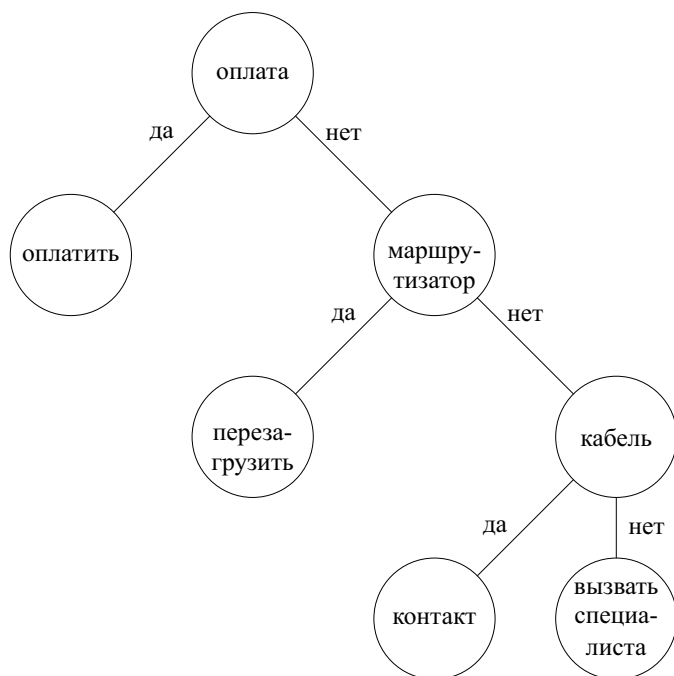


Рис. 7.4: Дерево для классификации проблем у клиента интернет-провайдера.

$p_3 = 0.1$, $p_4 = 0.05$. При поступлении запроса подразумевается, что может быть любой из случаев. То есть:

$$S_0 = -0.7 \log_2(0.7) - 0.15 \log_2(0.15) - 0.1 \log_2(0.1) + 0.05 \log_2(0.05) \sim 1.32.$$

После первого узла дерева остается три варианта неполадок и их энтропия:

$$S_1 = -0.15 \log_2(0.15) - 0.1 \log_2(0.1) + 0.05 \log_2(0.05) \sim 0.96.$$

После второго узла дерева остается два варианта неполадок и их энтропия:

$$S_2 = -0.15 \log_2(0.15) - 0.1 \log_2(0.1) - 0.05 \log_2(0.05) \sim 0.55.$$

После третьего узла дерева остается один вариант, который неустраним с помощью этого древа решений:

$$S_3 = -0.05 \log_2(0.05) \sim 0.22.$$

7.2.3 Упражнения

Упражнение 72 За 10 испытаний произошло 2 события «А», 3 события «В» и 5 событий «С». Считая, что частота появления событий близка к их вероятности, вычислить энтропию Шенона для 10 испытаний.

Упражнение 73 За 30 испытаний произошло 7 событий «А», 3 события «В» и 11 событий «С». Остальные события неклассифицированы. Определить минимальную энтропию Шенона для 100 испытаний после классификации обученным деревом решений, считая, что полученная ранее в 30 испытаниях частота появления событий близка к их вероятности.

7.3 Знакомство с нейросетями

7.3.1 Почему нейросеть?

Попытки построить искусственного человека известны со средних веков. Тогда это были различные механические куклы. Сейчас делаются попытки построить математическую модель, которая бы имитировала нечто похожее на мышление. Здесь важно заметить, что с математической точки зрения нет даже определения, что такое мышление. При всем том, что математики с легкостью дают определения непонятным явлениям.

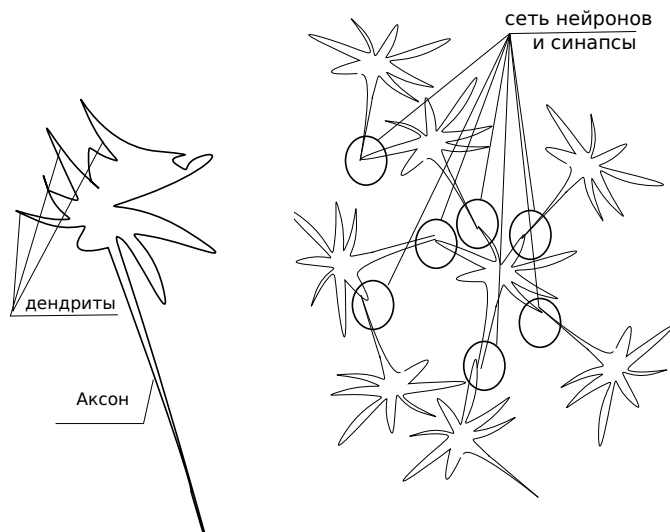


Рис. 7.5: Слева – схематическое изображение нейрона. Справа – изображение сети нейронов. Связь нейронов осуществляется через синапсы. Здесь синапс – место «контакта» аксона одного нейрона с дендритом другого.

Для моделирования сначала нужно определиться, что моделировать. В частности моделируется поведение нервных клеток, которые называются нейронами. Впервые выделить нейрон удалось с помощью метода окрашивания отдельных нейронов (Гольджи 1873г.).

Нейрон состоит из тела диаметром от 3 до 130 мкм, дендритов и аксона. Нейрон имеет развитый скелет, микротрубочки для транспорта веществ тянутся вдоль аксона.

Аксон – длинный отросток, по которому проходит возбуждение от нейрона к другому нейрону или исполнительному органу. Дендриты – короткие, сильно разветвленные отростки. Синапс – место контакта между нейроном и другим нейроном или клеткой.

Аксоны у жирафов могут быть до нескольких метров – проходят через шею. Самый длинный аксон у человека проходит через седалищный нерв до большого пальца ноги. Типичный объект, на котором изучают свойства аксонов – это гигантский аксон кальмара. Толщина этого аксона около 0.5 мм.

Известные свойства нейрона, которые моделируются при построении искусственных нейронных сетей:

- синапсы – точки связи нейронов;
- дендриты воспринимают внешние изменения, как через рецепторы, например колбочки в глазу, так и через синапсы – связи с другими нейронами;
- реакция на внешние раздражители не мгновенна, то есть гораздо медленнее, чем, например, скорость химических реакций;
- реакция на внешние раздражения проходит через аксон;
- существует пороговая величина раздражения. На раздражения меньшие, чем этот порог, нейрон не реагирует, то есть не формирует импульс.
- Обычно математические модели отдельных нейронов связывают между собой. Получающиеся таким образом объекты называют искусственными нейронными сетями (ИНС).

Считается, что сеть связанных нейронов определяет поведение животных и человека. Поэтому ожидается, что компьютерные модели нейронных сетей могут демонстрировать интеллектуальное поведение. Для справки, число нейронов у червей – несколько сотен, у мухи дрозофилы – примерно 10 тысяч, у человека – чуть меньше 100 миллиардов.

7.3.2 Математическая модель нейрона

Математическая модель нейрона имеет аналоги элементов биологического нейрона.

- Вектор входных параметров

$$\vec{x} = (x_1, \dots, x_n)$$

– аналог дендритов. Входные параметры могут быть различными – от значений цветов пикселей на картинке, до параметров данных на финансовых биржах, это зависит от области применения ИНС, в которой участвует нейрон.

- Вектор характеристик нейрона

$$\vec{w} = (w_1, \dots, w_n)$$

– аналог синаптических связей (синапсов). Это весовые коэффициенты, которые умножаются на вектор входных данных. От того, насколько правильно подобраны эти коэффициенты, во многом зависят свойства ИНС.

- Скалярное умножение вектора входных параметров на вектор характеристик нейрона

$$S = (\vec{x}, \vec{w}) = \sum_{i=1}^n x_i w_i,$$

– тело нейрона.

- Выходное значение – функция активации. Это монотонная или ступенчатая функция от результата скалярного произведения – аналог аксона:

$$y = f(S).$$

Обычно предполагается, что вывод – значение аксона – величина в пределах от $[0, 1]$, но это правило не всегда соблюдается. В качестве типичных входных функций используются:

– *ступенчатая функция*:

$$\theta(S) = \frac{1}{2}(1 + \operatorname{sgn}(S - z)),$$

здесь z – сдвиг от нулевого значения – порог реакции на возбуждение;

– так называемая «*сигмоида*» – сленговое название функции

$$\sigma(S) = \frac{1}{1 + e^{-S}},$$

эта функция хоть и не имеет порога реагирования, но ее значение близко к единице только при достаточно больших значениях аргумента S ;

– так называемая «*ReLU*» (от англ. Rectified Linear Unit) – сленговое название функции

$$R(S) = \max\{0, S\},$$

эта функция, так же как и «сигмоида», не имеет положительного порогового значения, но при значениях аргумента S близких к нулю функция также близка к нулю. Кроме того в отличие от «сигмоиды» функция «ReLU» быстро вычисляется. Однако отклик может оказаться больше, чем единица.

7.3.3 Персептрон

Персептрон представляет собой искусственную нейронную сеть (ИНС), состоящую из одного слоя нейронов. Первоначально концепция персептрона с точки зрения ИНС была предложена Френком Розенблаттом.

Ниже рассмотрен персептрон, состоящий из одного нейрона. Персептрон преобразует вектор \vec{x} в скаляр после скалярного произведения на вектор весов \vec{w} :

$$(\vec{x}, \vec{w}) = S.$$

Затем в зависимости от значения этого скаляра S получается ответ – нуль или число близкое к нулю для скаляров меньших чем пороговое значение, необходимое для формирования импульса; либо, если скаляр достаточно большой, персептрон выдает единицу или число близкое к единице.

Таким образом, основное свойство персептрона – разделение входных векторов на два класса. Это разделение происходит из-за линейной зависимости S от входного вектора и монотонной или ступенчатой функции активации. В результате входные векторы делятся *гиперплоскостью* на два класса. Грубо говоря – векторы для которых аргумент функции активации таков, что её значение равно или близко к нулю, и векторы, для которых аргумент функции активации равен или близок к единице.

- *Гиперплоскостью* называется $(n - 1)$ -мерная плоскость в n -мерном векторном пространстве. Формально она задается уравнением:

$$\sum_{i=1}^n a_i x_i = c,$$

где c – постоянная, $a = (a_1, \dots, a_n)$ – заданный вектор, x_1, \dots, x_n – точки гиперплоскости. Пример гиперплоскости в трехмерном пространстве – обычная плоскость. Гиперплоскость в двумерном пространстве – прямая.

Персептроны используются, например, в системах распознавания печатных знаков, знаков дорожного движения и других, довольно простых приложениях.

Пример 35 Рассмотрим устройство персептрона, обученного логической функции «или», рисунок 7.6. Такой персептрон можно представить в виде единственного нейрона с двумя дендритами. На дендриты подается либо 0 либо 1, персептрон выдает ответ:

$$R(0, 0) = 0, \quad R(0, 1) = 1, \quad R(1, 0) = 1, \quad R(1, 1) = 1.$$

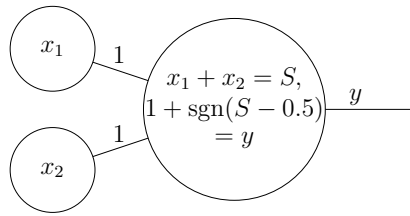


Рис. 7.6: Модель персептрона, который обучен функции «или». На вход подаются значения x_1 и x_2 . Вектор синаптических связей $\vec{w} = (1, 1)$. Вычисляется скалярное произведение (\vec{w}, \vec{x}) и к нему применяется функция активации.

Параметры персептрона: входной вектор $\vec{x} = (x_1, x_2)$, вектор весов $\vec{W} = (1, 1)$, ступенчатая функция: $y(S) = 1 + \text{sgn}(S - 1/2)$. В результате получается функция

$$R(x_1, x_2) = 1 + \text{sgn}(x_1 + x_2 - 1/2).$$

Существует множество ограничений для применения персептронов. Для примера здесь приведем одно, по-видимому, само простое. Стандартный однослойный персептрон не может реализовать логическую функцию «XOR»:

$$X(0, 0) = 0, \quad X(0, 1) = 1, \quad X(1, 0) = 1, \quad X(1, 1) = 0.$$

Проблема заключается в том, что суммирование входных параметров и монотонное возрастание функции активации персептрона не позволяют получить значение $X(1, 1) = 0$, если они уже выдают ответы $X(0, 1) = 1$, $X(1, 0) = 1$. То есть для построения функции «XOR» разграничения пространства входных данных одной гиперплоскостью недостаточно.

7.3.4 Многослойная нейросеть

Для того, чтобы реализовать на ИНС логическую функцию «XOR» рассмотрим двухслойную нейросеть. В такой сети нейроны первого слоя передают данные не на выход нейросети, а на вход второго слоя, такого же по структуре слоя нейронов. Каждый нейрон первого слоя передает значение со своего выхода на входы всех нейронов второго слоя. При этой передаче выходное значение каждого нейрона первого слоя умножается на некоторый коэффициент связи каждой пары нейронов первого и второго слоев.

Параметры нейронов i -го слоя будем обозначать верхним индексом (i) , например, $N^{(i)}$. Пусть внешний слой ИНС состоит из $N^{(1)}$ нейронов, внутренних – скрытый слой $N^{(2)}$ – нейронов. Тогда на выходе из первого слоя получится вектор из $N^{(1)}$ элементов: $\vec{y}^{(1)} = (y_1^{(1)}, \dots, y_{N^{(1)}}^{(1)})$. В теле j -го нейрона второго слоя есть вектор $\vec{w}_j^{(2)}$ из $N^{(1)}$ элементов, который скалярно умножается на $\vec{y}^{(1)}$ для получения аргумента функции активации $S_j^{(2)}$ для j -го нейрона.

Векторы \vec{w}_j можно разместить в виде строк. В результате получится матрица $W^{(2)}$, размером в $N^{(2)}$ строк и $N^{(1)}$ столбцов. Эта матрица называется весовой матрицей скрытого слоя ИНС. В результате умножения этой матрицы на вектор $\vec{y}^{(1)}$ – выходной вектор первого слоя получают аргументы весовых функций $S_j^{(2)}$ для каждого из нейронов второго слоя:

$$\begin{pmatrix} W_{11}^{(2)} & \dots & W_{1N^{(1)}}^{(2)} \\ \dots & \dots & \dots \\ W_{N^{(2)}1}^{(2)} & \dots & W_{N^{(2)},N^{(1)}}^{(2)} \end{pmatrix} \begin{pmatrix} y_1^{(1)} \\ \dots \\ y_{N^{(1)}}^{(1)} \end{pmatrix} = \begin{pmatrix} S_1^{(2)} \\ \dots \\ S_{N^{(2)}}^{(2)} \end{pmatrix}.$$

Тогда на выходе из слоя для функции активации $\vec{f}(\vec{S})$ получим:

$$\vec{f}(S^{(2)}) = \vec{y}^{(2)}.$$

Очевидно, что таким образом можно добавлять скрытые слои нейросети. При этом выходные данные $k-1$ -го слоя $\vec{y}^{(k-1)}$ окажутся набором входных данных $\vec{X}^{(k)}$

Рассмотрим устройство двухслойной ИНС, которая реализует логическую функцию «XOR», рисунок 7.7. В первом слое ИНС два нейрона. Весовая матрица для первого слоя $W^{(1)}$:

$$W^{(1)} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

преобразует входной вектор $\vec{x} = (x_1, x_2)$ первого слоя в аргументы функций активаций для нейронов первого слоя:

$$\begin{aligned} x_1 - x_2 &= S_1^{(1)}, \\ -x_1 + x_2 &= S_2^{(1)}. \end{aligned}$$

Функции активации нейронов первого слоя выдают значения на свои аксоны:

$$\begin{aligned} 1 + \operatorname{sgn}(S_1^{(1)} - 1/2) &= y_1^{(1)}, \\ 1 + \operatorname{sgn}(S_2^{(1)} - 1/2) &= y_2^{(1)}. \end{aligned}$$

Пример 36

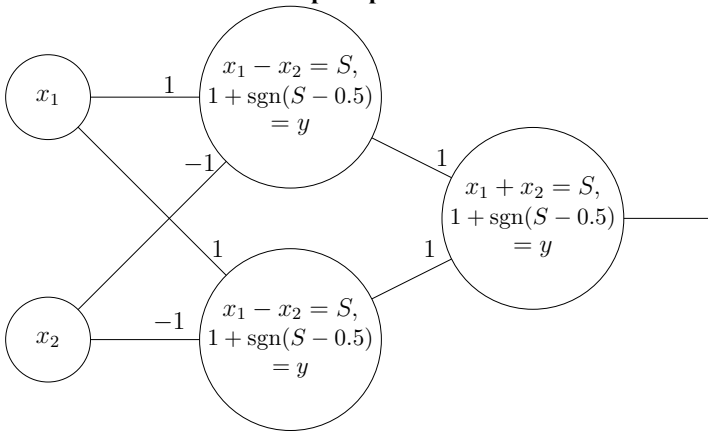


Рис. 7.7: Схема двухслойной ИНС, обученной функции XOR. На ребрах графов указаны коэффициенты весовых матриц связей. На выходе каждого нейрона – значение y .

Во втором слое – один нейрон. Его весовая матрица – строка:

$$W^{(2)} = (1, 1).$$

В результате аргумент функции активации единственного нейрона второго слоя:

$$y_1^{(1)} + y_2^{(1)} = S_1^{(2)}.$$

Результат работы нейрона второго слоя и, соответственно, всей ИНС:

$$1 + \text{sgn}(S_1^{(2)} - 1/2) = y_1^{(2)}.$$

- В многослойной нейросети вектор выходных данных каждого предыдущего слоя (назовем его первым) умножается на матрицу связи этого слоя с последующим (вторым) слоем. В результате образуется входной вектор данных для нейронов этого (второго) слоя. Этот вектор входных данных каждый нейрон (второго) слоя преобразует с помощью своей функции активации. В результате формируется новый вектор, который может быть входным для нейронов последующего (уже третьего) слоя.

7.3.5 Разновидности нейросетей

Со времен первого персептрона появилось большое разнообразие математических моделей и вычислительных алгоритмов, базирующихся на идеях моделирования нейросетей. Как правило они расширяют рассмотренный выше многослойный персептрон. Во многих случаях их применение для частных задач оказывается гораздо более успешным. Ниже приведены несколько примеров таких сетей. Список примеров не претендует на полноту и сколь-нибудь внятную классификацию.

При обработке изображений важную роль играют фильтры различных типов. Часть из таких фильтров рассмотрена в разделе 4. Для ИНС, специализирующихся на обработке изображений, естественно включить слои, работающие похожим образом. Они, подобно оператору свертки, содержат небольшие ядра – скользящие окна. Такие окна пробегает по матрице изображения и формируют ответ. ИНС, использующие этот подход к обработке изображений, называют *сверточными нейросетями*.

Выше обсуждались ИНС прямого распространения. В них данные распространяются от слоя к слою и предыдущий слой не имеет доступа к данным, обработанным последующим слоем. Естественное обобщение, при котором выходы (аксоны) последующих слоев соединены со входами (дендритами) предыдущих слоев, приводит к ИНС, которые называются *рекуррентными*. По существу в ИНС такой топологии понятие «слой нейронов» теряет свой первоначальный смысл. Рекуррентные нейросети имеют свойство, которое можно рассматривать как память. Такие ИНС могут обрабатывать события, растянутые во времени, когда информация подается на входные нейроны сети последовательно такт за тактом.

Нейроны мозга обмениваются импульсами, длительность и частота импульсов различны. Это наблюдение привело к созданию *импульсных нейросетей*. В таких нейросетях нейрон в нормальном состоянии имеет некоторый заряд. Если акцепторы (дендриты) нейрона получают достаточно высокое возмущение, то нейрон активируется и отправляет импульс связанным с ним акцепторам других нейронов. После активации нейрону требуется некоторое время для накопления заряда и перехода в нормальное состояние.

7.4 Обучение нейросетей

7.4.1 Методы обучения ИНС

Важнейший этап подготовки ИНС к использованию – обучение. Существует несколько подходов к обучению ИНС.

- Обучение с учителем. С формальной точки зрения обучение с учителем – подбор весовых коэффициентов для связей нейронов ИНС таким образом, чтобы ответ ИНС на контрольные вопросы (наборы данных) соответствовал ожиданиям. Для обучения с учителем необходим тестовый набор данных, для которых заранее известен желательный ответ ИНС. Если принять, что ИНС выдает ответ в виде вектора m -мерного пространства, то удобно ввести какую-либо естественную метрику в этом пространстве и подбирать весовые коэффициенты ИНС так, чтобы сумма расстояний между ответами ИНС и соответствующими им желательными ответами была наименьшей. Поиск оптимальных весовых коэффициентов обычно проводится с помощью стандартных алгоритмов минимизации целевой функции в многомерном пространстве.
- Обучение без учителя. Вообще говоря, этот метод обучения предполагает, что заранее неизвестны желаемые ответы ИНС на контрольные вопросы (наборы данных). Однако, что важно, задана метрика – способ вычисления расстояния между выходными данными ИНС. В такой ситуации можно обучить ИНС разбивать выходные данные на кластеры. Например, с помощью алгоритма кластеризации из раздела 3.1.4 выделить кластеры в выходных данных и использовать алгоритм подбора весовых коэффициентов в нейросети для минимизации расстояний внутри кластеров выходных данных.
- Обучение с помощью автокодирования. Для использования этого алгоритма ИНС должна иметь узкое горло – слой, в котором число нейронов меньше, чем число нейронов во входном слое. Алгоритм автокодирования состоит в том, чтобы к узкому слою ИНС дописать несколько расширяющихся слоев так, чтобы в выходном слое новой сети число выходных аксонов равнялось числу входных акцепторов этой сети. Алгоритм подбора весовых коэффициентов – минимизация различий между входными данными и выходными.

После обучения ИНС-автоэнкодера, обычно разрывают сеть в узком слое. В результате получаются две ИНС. Одна из них имеет широкий

слой как входной. Она в качестве вектора ответов выдает определяющие признаки набора входных данных.

Вторая ИНС – с узким входным слоем. При подаче в узкий слой признаков, в качестве ответов в широком слое ИНС будет выдавать реконструированные данные по проведенной ИНС классификации.

7.4.2 Наборы данных для обучения

После того, как определен способ обучения ИНС, подбираются наборы данных для обучения. Данные для обучения должны обладать специальными свойствами.

- Выборка данных должна быть репрезентативной – представлять все многообразие типов данных, с которыми ИНС будет работать после обучения.
- Данные должны быть нормированными. То есть нужно выделить нормировочные коэффициенты для каждого из элементов вектора данных и нормировать (поделить) все компоненты всех входных данных на соответствующие коэффициенты. Этот шаг важен для стандартизации подходов. Типичные функции активации предназначены для использования в небольшом диапазоне около нуля.
- Для обучения с учителем данные должны быть разделены на обучающую выборку и тестовую выборку. Тестовая выборка позволяет независимо оценить результаты обучения, поскольку примеры из тестовой выборки «не показывают» ИНС во время обучения.

В частности, если при обучении от одной итерации обучения к другой итерации ошибка уменьшается, но при проверке результатов обучения на тестовой выборке ошибка растет, это значит, что сеть запоминает тестовую выборку, но не обучается. Это явление – аналог так называемого явления Гиббса в теории интерполяции – интерполяционные полиномы высоких порядков сильно осциллируют между точками интерполяции.

В теории интерполяции это означает, что нужно понизить степень интерполяционного полинома, а в теории ИНС – что сеть перестала обучаться и только запоминает требуемые результаты – перешла к «зубрежке». Для ИНС это явление называется «переобучение».

7.5 Методы минимизации

Этап обучения ИНС практически сводится к минимизации некоторого положительно определенного функционала¹ – оператора, который ставит в соответствие вектору из многомерного пространства неотрицательное число. Переменные, по которым проводится минимизация – все весовые коэффициенты всех нейронов ИНС. Ниже все эти весовые коэффициенты формально будут обозначаться вектором $w = (w_1, \dots, w_n)$.

Пример 37 *Пример положительно определенного функционала. Для обучения с учителем.*

Пусть ИНС имеет N выходных параметров. Обозначим их $y = (y_1, \dots, y_N)$. Пусть имеется обучающая выборка из M элементов. Для каждого элемента известно желаемое выходное значение для всех параметров ИНС: $\{Y^{(j)}\}_{j=1}^M$.

В качестве положительно определенного функционала для этой выборки можно использовать:

$$F(y) = \sum_{j=1}^M \|y - Y^{(j)}\|,$$

где $\|y\|$ – норма в пространстве выходных данных ИНС.

- *Нормой* в некотором пространстве \mathcal{M} называется функционал, который ставит в соответствие элементу пространства $y \in \mathcal{M}$ неотрицательное число: $\|y\| \rightarrow \mathbb{R}^+$.

Функционал нормы должен удовлетворять трем аксиомам.

1. $\|y\| \geq 0$ – неотрицательность нормы;
2. $\|x + y\| \leq \|x\| + \|y\|, \quad \forall x, y \in \mathcal{M}$ – правило треугольника;
3. $\|\lambda y\| = |\lambda| \cdot \|y\|, \quad \forall \lambda \in \mathbb{R}, \quad y \in \mathcal{M}$ – правило умножения на число.

Нормой можно считать расстояние между точками в пространстве. Примеры:

- евклидово расстояние между точками;
- метрика l_1 ;

¹ Функционалом называют отображение множества в пространство чисел. Область определения функционала – некоторое множество объектов, как-то: чисел, векторов, функций. Множество значений – интервал на числовой оси (возможно и вся числовая ось).

- норма в цветовом пространстве RGB.

С точки зрения геометрии задача минимизации сводится к задаче о минимальном значении некоторого скалярного поля, определенного на элементах многомерного пространства. Многомерное пространство здесь состоит из весовых коэффициентов связей нейронов, а функционал определяет скалярное поле.

В самом простом случае, если пространство весовых коэффициентов двумерно (коэффициентов только два), то скалярное поле можно представлять как поверхность над плоскостью. Задача минимизации – найти точку с минимальной высотой на такой двумерной поверхности.

7.5.1 Метод скорейшего покоординатного спуска

Самый простой способ поиска такого минимума – алгоритм покоординатного спуска. Характерный параметр метода – шаг спуска по координате. Пусть есть заданный заранее массив шагов по каждой из координат $\Delta_1, \dots, \Delta_n$. В простом случае все шаги могут быть одинаковой длины Δ .

- Текущая точка имеет координаты w_1, \dots, w_n .
- Цикл по всем координатам от w_1 до w_N :

- Шаг по координате $W_i = w_i + \Delta_i$.
- Проверка условия

$$F(w_1, \dots, W_i, \dots, w_N) < F(w_1, \dots, w_i, \dots, w_N).$$

- Если условие выполнено, переход в текущую точку с новой координатой $w_i = W_i$.
- Если условие не выполнено, шаг по координате в противоположную сторону: $W_i = w_i - \Delta_i$.
- Проверка условия:

$$F(w_1, \dots, W_i, \dots, w_N) < F(w_1, \dots, w_i, \dots, w_N).$$

- Если условие выполнено, переход в текущую точку с новой координатой $w_i = W_i$.
- Если условие не выполнено, переход к следующей координате $i + 1$

- Если в результате цикла координата изменилась – процесс повторяется на следующем шаге с новой начальной точкой.
- Если в результате цикла координата не изменилась – достигнут локальный минимум.

Такой наивный координатный спуск работает медленно, так как шаг делается на заранее заданную для каждой координаты величину Δ_i . Очевидно, необходима его модификация. По сути при покоординатном спуске на каждом этапе функция многих переменных $F(w_1, \dots, w_i, \dots, w_N)$ заменяется на функцию одной переменной w_i . Остальные переменные рассматриваются как постоянные параметры:

$$\Phi(w_i) = F(w_1, \dots, w_i, \dots, w_N), \quad w_j = \text{const}, \quad \forall j \neq i.$$

Поиск локального минимума для функции одной переменной можно проводить разными способами. Если, например, на первом этапе с помощью нескольких итераций определить границы отрезка $x \in [w_i, b]$, в котором $\Phi(x) \leq \Phi(w_i)$, то одним из эффективных методов является поиск минимума с помощью процедуры деления пополам. В результате получится модификация метода покоординатного спуска, которая называется наискорейший покоординатный спуск.

Пример 38 Рассмотрим наискорейший покоординатный спуск на примере минимизации функции

$$f(x_1, x_2) = (x_1 + x_2)^2 + 3(x_1 - x_2)^2.$$

Линии уровня этой функции – эллипсы с большой полуосью, ориентированной вдоль прямой $x_1 = x_2$. Точка минимума $(0, 0)$.

Пусть начальная точка $(2, 3)$.

Значение функции в этой точке:

$$f(2, 3) = 28.$$

Для спуска по первой координате зафиксируем вторую координату: $x_2 = 3$, тогда получим функцию:

$$f(x_1, 3) = (x_1 + 3)^2 + 3(x_1 - 3)^2.$$

Раскроем скобки и приведем подобные слагаемые. В результате:

$$f(x_1, 3) = 4x_1^2 - 12x_1 + 36 = 4(x_1 - 1.5)^2 + 27.$$

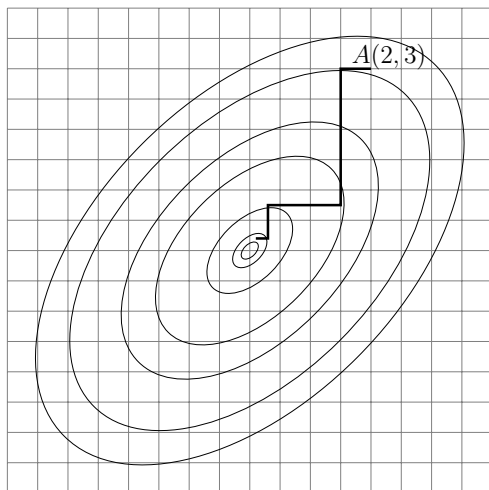


Рис. 7.8: Поиск минимума функции $f = (x_1 + x_2)^2 + 3(x_1 - x_2)^2$ методом наискорейшего покоординатного спуска из точки $A(2,3)$. Замкнутые кривые на графике – линии уровня функции f .

Тогда минимум по переменной x_1 окажется в точке $x_1 = 1.5$. Значение функции:

$$f(1.5, 3) = 27.$$

Теперь зафиксируем $x_1 = 1.5$ будем искать минимум по переменной x_2 :

$$f(1.5, x_2) = 4x_2^2 - 6x_2 + 9 = 4(x_2 - 0.75)^2 + 6.75;$$

Минимум этой функции по переменной $x_2 = 0.75$. Значение функции:

$$f(1.5, 0.75) = 6.75.$$

Таким образом, после скорейшего спуска по обоим переменным новая точка имеет координаты $(1.5, 0.75)$.

Делая последовательно шаги метода скорейшего покоординатного спуска, можно прийти в сколь угодно малую окрестность минимума – точку $(0, 0)$.

- Достоинством метода покоординатного спуска является простота реализации и возможность использования для негладких функций.
- Недостаток метода – медленный спуск в глубокий овраг минимизируемой функции, если овраг не ориентирован по одной из координат.

7.5.2 Метод скорейшего градиентного спуска

Поиск локального минимума можно рассматривать как спуск в самую низкую точку в окрестности. С локальной точки зрения, не зная всю окружающую картину, спускаться быстрее по направлению наибольшего наклона. Это направление противоположно направлению градиента функции, вычисленной в точке текущего положения.

- *Градиентом функции* называется вектор частных производных функции по всем аргументам:

$$\text{grad}(F(\vec{w})) = \vec{\nabla} F = \left(\frac{\partial F}{\partial w_1}, \dots, \frac{\partial F}{\partial w_N} \right).$$

Вектор градиента дает направление наискорейшего роста функции, соответственно противоположное направление – направление наискорейшего убывания.

Пример 39 Вычислим вектор градиента функции

$$f(\vec{x}) = x_1^2 - 3x_1x_2 + x_2^4,$$

В точке $(1, 2)$.

Чтобы определить градиент функции, вычислим частные производные:

$$\frac{\partial f}{\partial x_1} = 2x_1 - 3x_2, \quad \frac{\partial f}{\partial x_2} = -3x_1 + 3x_2^3.$$

В вычисленные выражения для производных подставим координаты точки $(1, 2)$. В результате получится:

$$\vec{\nabla} f|_{(1,2)} = (-5, 18).$$

Рассмотрим типичные шаги алгоритма градиентного спуска. Характерный параметр метода – длина шага в направлении градиента. Обозначим этот параметр Δ .

- Текущая точка имеет координаты w_1, \dots, w_n .
- По формулам приближенных вычислений вычислить градиент функции F .
- Шаг метода в направлении противоположном градиенту $\vec{W} = \vec{w} - \Delta \text{grad}(F)$.

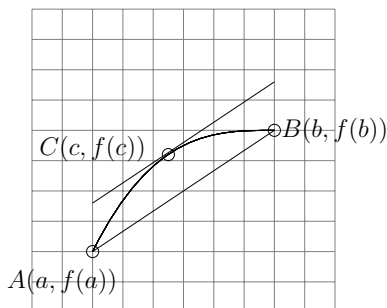


Рис. 7.9: Иллюстрация теоремы Лагранжа. Касательная в точке C параллельна секущей AB . То есть тангенс угла наклона секущей равен тангенсу угла наклона касательной в точке C .

- Проверка условия $F(\vec{W}) < F(\vec{w})$.
- Если условие выполнено, то переход в текущую точку с новой координатой $\vec{w} = \vec{W}$.
- Если условие не выполнено, то локальный минимум лежит на расстоянии меньше длины шага Δ . Можно считать, что минимум достигнут.

Для приближенного вычисления частных производных функции удобно воспользоваться следствием из теоремы Лагранжа.

- Теорему Лагранжа для дифференцируемой (имеющей производную) на интервале (a, b) функции $f(x)$ можно записать в виде формулы:

$$\frac{f(b) - f(a)}{a - b} = f'(c), \quad c \in (a, b),$$

или сформулировать геометрически: тангенс угла наклона секущей равен производной функции в некоторой точке между концами секущей.

- Для приближенного вычисления частной производной можно воспользоваться следствием из теоремы Лагранжа. Значение производной внутри малого интервала близко к тангенсу угла наклона секущей на этом интервале. В частности:

$$\frac{\partial F}{\partial w_i} \sim \frac{F(\dots, w_i + \delta, \dots) - F(\dots, w_i - \delta, \dots)}{2\delta}.$$

здесь δ – малая величина.

Существует множество модификаций градиентного спуска. Наиболее популярным методом из множества методов, использующих градиент функции, является метод наискорейшего градиентного спуска.

В методе наискорейшего градиентного спуска функция многих переменных $F(w_1, \dots, w_i, \dots, w_N)$ на каждом шаге после вычисления градиента в точке $(w_1, \dots, w_i, \dots, w_N)$ заменяется на функцию одной переменной Δ :

$$\Phi(\Delta) = F(\vec{w} - \vec{\nabla} F(\vec{w}) \cdot \Delta), \quad \{w_1, \dots, w_N\} = \text{const}.$$

При этом поиск локального минимума функции многих переменных подменяется на более простую задачу – поиск минимума функции одной переменной – $\Phi(\Delta)$.

Поиск минимума функции одной переменной состоит из двух этапов. На первом этапе с помощью нескольких итераций определяются границы отрезка $\Delta \in [0, b]$, на котором $\Phi(\Delta) \leq \Phi(0)$. Затем для поиска минимума функции $\Phi(\Delta)$ на этом отрезке применяется, например, метод деления пополам. В результате будет найдено значение Δ_* , при котором функция $\Phi(\Delta)$ принимает минимальное значение на отрезке $\Delta \in [0, b]$.

Найденная величина Δ_* используется для перехода к новой точке в методе наискорейшего спуска:

$$\vec{w} - \vec{\nabla} F \cdot \Delta_* \rightarrow \vec{w}.$$

Затем процесс поиска повторяется до тех пор, пока близость текущей точки к точке минимума не будет признана достаточной.

Пример 40 Рассмотрим наискорейший градиентный спуск на примере минимизации функции

$$f(x_1, x_2) = (x_1 + x_2)^2 + 3(x_1 - x_2)^2.$$

Линии уровня этой функции – эллипсы с большой полуосью, ориентированной вдоль прямой $x_1 = x_2$. Точка минимума $(0, 0)$.

Пусть начальная точка $(2, 3)$. Значение функции в этой точке:

$$f(2, 3) = 28.$$

Вычислим частные производные функции:

$$\frac{\partial f}{\partial x_1} = 8x_1 - 4x_2, \quad \frac{\partial f}{\partial x_2} = -4x_1 + 8x_2.$$

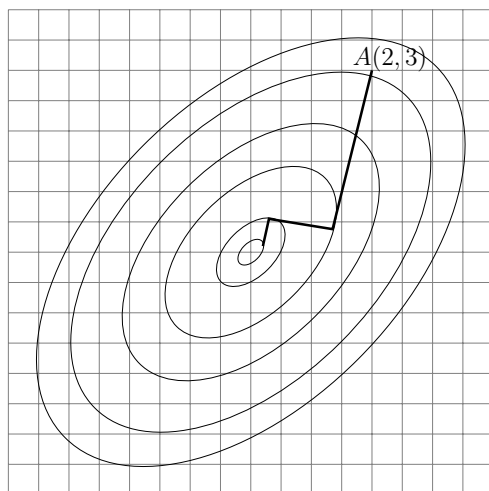


Рис. 7.10: Поиск минимума функции $f = (x_1 + x_2)^2 + 3(x_1 - x_2)^2$ методом наискорейшего градиентного спуска из точки $A(2,3)$. Замкнутые кривые на графике – линии уровня функции f . В каждой точке поворота линия спуска ортогональна касательной линии уровня функции.

Тогда градиент функции в точке $(2, 3)$ функции:

$$\vec{\nabla} f|_{(2,3)} = (4, 16);$$

Тогда движение в противоположную сторону из точки $(2, 3)$ – это движение по лучу:

$$x_1 = 2 - 4t, \quad x_2 = 3 - 16t, \quad t > 0.$$

Подставим выражения для x_1, x_2 на луче в формулу для минимизируемой функции:

$$f(x_1(t), x_2(t)) = 832t^2 - 272t + 28 = 832 \left(t - \frac{271}{2 \cdot 832} \right)^2 + \frac{75}{13}.$$

Следовательно, минимум на луче находится в точке

$$t_1 = \frac{271}{2 \cdot 832} \sim 0.16346.$$

Координаты точки

$$x_1 = 2 - 4t_1 \sim 2 - 4 \cdot 0.16346 \sim 1.35, \quad x_2 = 3 - 16t_1 \sim 3 - 16 \cdot 0.16346 \sim 0.38.$$

Значение минимизируемой функции в этой точке:

$$f = \frac{75}{13} \sim 5.77.$$

Заметим, что в результате одного шага методом наискорейшего спуска достигнута точка, в которой минимизируемая функция меньше, чем та, что достигнута в примере 38 за два шага.

- По сравнению с покоординатным спуском этот метод лучше ведет себя при спуске в извилистые овраги.
- В извилистых оврагах спуск по этому методу происходит медленно.

7.5.3 Глобальная минимизация

Локальных минимумов может быть несколько. Алгоритмы минимизации такие минимумы находят. Например, градиентный спуск из разных точек показан на рисунке 7.11. Однако для поиска глобального минимума методы спуска, вообще говоря, непригодны по своей природе.

Универсальный способ поиска глобального минимума известен – это полный перебор. Но полный перебор требует очень много ресурсов – либо времени, либо вычислительных ядер.

Глобальный и локальный минимумы

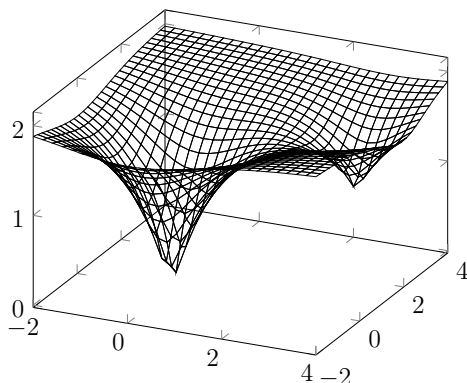


Рис. 7.11: Пример поверхности с двумя минимумами. Слева глобальный минимум, справа – локальный. Если начальная точка для поиска методом градиентного спуска окажется в окрестности локального минимума, то глобальный минимум для такого поиска будет недостижим.

Есть различные приемы, которые позволяют находить несколько локальных минимумов и выбирать наименьший из них, либо модифицировать алгоритмы поиска локального минимума так, чтобы модифицированный алгоритм не застревал в неглубоких локальных минимумах.

Метод Монте-Карло или случайного перебора получил свое название в связи с теорией вероятностей и названием казино. Смысл метода в том, чтобы случайным образом многократно инициализировать начальную точку \vec{w} и вычислять значения $F(\vec{w})$, выбирая из всего инициализированного набора точек ту, которая дает наименьшее значение целевой функции $F(\vec{w})$. Сам по себе этот метод выглядит не очень обнадеживающе, как и всякая игра в казино. Однако при использовании параллельных вычислений можно проверять большое количество вариантов. Кроме того, если из каждой случайно инициализированной точки запускать один из методов спуска, то возможно построение вполне успешного алгоритма.

Полет Леви. Наблюдение за поведением хищников подсказывает одну из возможных модификаций алгоритмов поиска минимума. Хищники во время охоты могут долгое время находиться в некоторой небольшой окрестности при поиске добычи. Но затем они не расширяют область поиска, а производят длинный переход и опять ищут добычу уже в другой области,

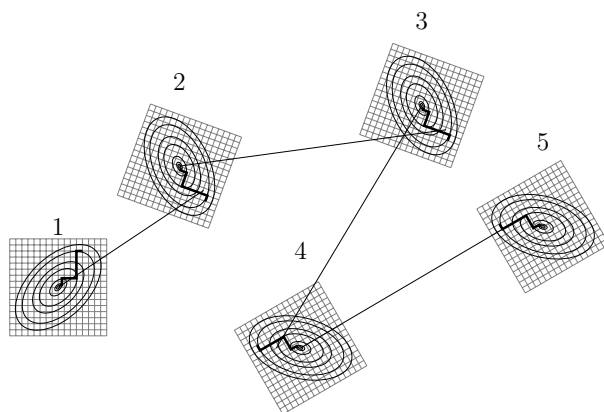


Рис. 7.12: Схема использования алгоритма полета Леви при поиске глобального минимума функции f . Замкнутые кривые на графике – линии уровня функции f . Начинается поиск в окрестности локального минимума в области 1, затем, переход из локального минимума области 1 в область 2, затем из локального минимума области 2 – в область 3 и так далее.

достаточно удаленной от первоначальной. Такое поведение связано с тем, что если при неудачном поиске добычи отойти на небольшое расстояние от исходной точки поиска, то есть большая вероятность при локальном движении вернуться опять в ту область, в которой уже был поиск добычи ранее. Чтобы почти наверняка проводить локальный поиск добычи в другой области нужно совершить большой переход. Такой алгоритм поиска иногда называется полетом Леви. Его можно применить и для поиска глобального минимума. Так как направление дальнего перехода к новой области локального поиска случайно, то можно считать этот алгоритм комбинацией метода случайного перебора и методов спуска для поиска локального минимума.

Генетические алгоритмы очень разнообразны, но имеют несколько общих черт. Прежде всего, генетический алгоритм подразумевает возможность скрещивания или размножения объектов из одного или нескольких родительских объектов. Дочерние объекты при этом наследуют структуру родительских объектов не полностью. В новых объектах возможны случайные изменения в структуре. Таким образом из одного или нескольких родительских объектов получается новое поколение объектов со случайными изменениями в структуре. На следующем шаге запускается алгоритм отбора – оценки полученных новых объектов с точки зрения их успешности. Часть

случайно отобранных из неудачных объектов в дальнейшем алгоритме не участвует. Затем опять переходят к шагу скрещивания. В результате поколение за поколением отбираются более успешные объекты. Такой алгоритм оптимизации применим как к ИНС, так и к самим алгоритмам оптимизации.

7.6 Обзор литературы

Марковские цепи обсуждаются как в учебниках по теории вероятностей [13], [12], так и в журнальной литературе [6].

Список работ по основам искусственных нейронных сетей необозрим. Для знакомства с основными подходами к построению нейросетей и их использованию можно порекомендовать книги [7],[4].

Литература

- [1] G. Airy. On the regulator of the clock-work for effecting uniform movement of equatoreals. *Memories of the Royal astronomical society*, XI:249, 1840.
- [2] R. Clavel. *Conception d'un robot parallele rapide a 4 degres de liberte*. PhD thesis, Ecole polytechnique federale de Lausanne, 1991.
- [3] G. Indiveri. Swedish wheeled omnidirectional mobile robots: Kinematics analysis and control. *IEEE Transactions on Robotics*, 25(1):164–171, Feb 2009.
- [4] D. Kreisel. *A Brief Introduction to Neural Networks*. http://www.dkriesel.com/en/science/neural_networks.
- [5] J.C. Maxwell. On governors. *The proceeding of the Royal society*, 100:105–120, 1868.
- [6] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [7] R. Rojas. *Neural networks. A systematic introduction*. Springer, 1996.
- [8] Wischnegradskii. Sur la theorie generale des regulateurs. *Comptes rendus hebdomadaires des seances de l'Academie des sciences*, 83:318–321, 1876.
- [9] И.С. Мамаев А.В. Борисов, А.А. Килин. Тележка с омниколесами на плоскости и сфере. *Нелинейная динамика*, 7(4):785–801, 2011.
- [10] И.И. Артоболевский. *Теория машин и механизмов*. Физ.мат. лит., 1988.
- [11] Е. С. Вентцель. *Теория вероятностей*. Наука, 1969.
- [12] В.Е. Гмурман. *Теория вероятностей и математическая статистика*. Высшая школа, 2003.

- [13] Б.В. Гнеденко. *Курс теории вероятностей*. Наука, 1988.
- [14] Н.Н. Калиткин. *Численные методы*. Наука, 1978.
- [15] В.Феллер. *Введение в теорию вероятностей и её приложения*. Том 1, Мир, 1963.
- [16] Р. Гонсалес and p. Вудс. *Цифровая обработка изображений*. Техносфера, 2005.
- [17] В.А. Иванов and А.С. Ющенко. *Теория дискретных систем автоматического управления*. МГТУ им. Баумана, 2015.
- [18] Г. Кампион, Ж. Бастен, and Б. д'Андреа-Новель. Структурные свойства и классификация кинематических и динамических моделей колесных мобильных роботов. *Нелинейная динамика*, 7(4):733–769, 2011.
- [19] Д.К. Максвелл, И.А. Вышнеградский, and А. Стодола. *Теория автоматического регулирования*. Классики науки. Академия наук СССР, 1949.
- [20] Ю. Г. Мартыненко. Управление движением мобильных колёсных роботов. *Фундаментальная и прикладная математика*, 11(8):29–80, 2005.
- [21] А.А. Первозванский. *Курс теории автоматического управления*. Наука, 1986.
- [22] В.Т. Фисенко and Т.Ю. Фисенко. *Компьютерная обработка и распознавание изображений*. Университет ИТМО, 2008.
- [23] К. Фу, Р. Гонсалес, and К.Ли. *Робототехника*. Мир, 1989.
- [24] Е.И. Юревич. *Основы робототехники*. ВНУ, 2018.
- [25] А.А. Красовский (ред.). *Справочник по теории автоматического управления*. Наука, 1987.
- [26] В.В. Солодовников (ред). *Устройства и элементы систем автоматического регулирования и управления, книга 1*. Машиностроение, 1973.
- [27] В.В. Солодовников (ред). *Устройства и элементы систем автоматического регулирования и управления, книга 2*. Машиностроение, 1975.
- [28] Л.С. Понтрягин. *Обыкновенные дифференциальные уравнения*. Наука, 1974.

- [29] Б.А. Дубровин, С.П. Новиков, А.Т. Фоменко. *Современная геометрия. Методы и приложения*. Наука, 1986.
- [30] Д. Форсайт, Ж. Понс. *Компьютерное зрение. Современный подход*. Вильямс, 2004.
- [31] М. Гарднер. *Крестики-нолики*. Мир, 1988.
- [32] D. Mitchie. *Trial and Error* Science Survey, 1961, Harmondsworth: Penguin, Part 2, pp. 129-145.
- [33] М. Гарднер. *Математические досуги*. Мир, 1972
- [34] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, J. Leonard. *Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age*. <https://arxiv.org/pdf/1606.05830.pdf>

Предметный указатель

- Метрический тензор., 170
адаптивный пороговый фильтр, 103
аффинное преобразование, 106
аппроксимация, 110
асимптотическая эквивалентность, 197
асимптотическая устойчивость, 136
центральная предельная теорема, 74
цепь Маркова, 191
частная производная, 112
дифференциальное уравнение первого порядка, 122
дисперсия, 72
двухзвенник, 37
двумерный тор, 38
энтропия, 197
энтропия Шенона, 198
эргодическая гипотеза, 72
эрозия, 104
фазовая плоскость, 132
фильтр скользящего среднего, 98
функционал, 211
генетический алгоритм, 221
геодезическая, 171
гиперплоскость, 204
градиент функции, 102, 215
импульсные нейросети, 208
инфимум, 171
интерполяция, 80
комплексносопряженные числа, 141
комплексное число, 141
конфигурационное пространство, 15
математическое ожидание, 71
медианный фильтр, 102
метод Монте-Карло., 220
метрика l_1 , 163
мнимая единица, 141
модуль комплексного числа, 141
наращивание, 106
неголономное устройство, 15
некоммутативность умножения матриц, 58
нелинейный фильтр, 103
неопределенный интеграл, 29
норма, 211
нормальное распределение, 74
обратная задача, 47
общее положение, 48
оператор свертки, 98
определенный интеграл, 143
первообразная, 28
полет Леви, 220
полярная система координат, 36
пороговый фильтр, 103
правая тройка координат, 41
предельный цикл, 153
производная функции, 22
прямая задача, 47
прямое произведение, 38
рабочее пространство, 37
равномерное распределение дискретной случайной величины, 73
равномерное распределение непрерывной случайной величины, 73
размыкающий фильтр, 106
разомкнутая система, 120
реализация марковской цепи, 193

регрессия, 111
рекуррентные нейросети, 208
сигмоида, 203
система с обратной связью, 121
скалярное поле, 166
собственный вектор, 139
собственное значение, 139
среднее значение, 71
среднеквадратичное отклонение, 72
статистическая гипотеза, 71
степень свободы, 46
стохастическая матрица, 191
ступенчатая функция, 203
субриманово пространство, 17
теорема Лагранжа, 216
топологическая эквивалентность, 36
трехзвенник, 44
трёхмерный тор, 44
умножение матрицы на вектор, 55
управление, 120
векторное поле, 166
вообще говоря, 41
взвешенный граф, 159
ядро оператора свертки, 98
замыкающий фильтр, 106

сверточные нейросети, 208

ReLU, 203

Научное издание

О.М. Киселёв

МАТЕМАТИЧЕСКИЕ ОСНОВЫ
РОБОТЕХНИКИ

Подписано в печать 06.09.2019 г. Формат 60×84 1/16
Печать ризография. Бумага офсетная. Гарнитура Computer
Modern Объём 14,25 усл. печ. л. Тираж 100 экз. Заказ № 249РФ

Лицензия ПД № 8-0023 от 25.09.2000 г.
Отпечатано с готового оригинал-макета в авторской редакции
в ООО Полиграфическая фирма «Картуш»
г. Орел, ул. 2-я Посадская, 26. Тел.: (4862) 44-51-46.
E-mail: kartushizdat@yandex.ru www.издатькнигу.рф