Санкт-Петербургский государственный университет

**Кафедра компьютерных технологий и систем**

**Тищук Максим Анатольевич**

**Выпускная квалификационная работа бакалавра**

**Синтез нелинейных законов управления для робота-манипулятора**

Направление 010400.62

Прикладная математика, информатика и программирования

Научный руководитель,

доктор физ.-мат. наук,
доцент
Сотникова М. В.

Санкт-Петербург

2017

# Содержание

Введение 3

Глава 1. Постановка задачи 5

1.1. Математическая модель двухзвенного робота-манипулятора 5

1.2. Проверка наблюдаемости и управляемости системы 7

1.3. Постановка задачи управления роботом-манипулятором 10

1.4. Обзор литературы 11

Глава 2. Синтез законов управления для двухзвенного
робота-манипулятора 13

2.1. Синтез асимптотического наблюдателя 13

2.2. Синтез PD-регулятора 16

2.3. Метод линеаризации обратной связью 18

2.4. Синтез закона управления с прогнозом 20

Глава 3. Практическая реализация разработанных алгоритмов управления и результаты моделирования 25

3.1. Имитационно-моделирующий комплекс 25

3.2. Результат вычислительных экспериментов для разработанных алгоритмов управления 27

Выводы 34

Список литературы 35

Приложение 36

# Введение

Задач, описываемых нелинейными моделями, во много раз больше чем задач, описываемых линейными моделями. Несмотря на это, нелинейные модели не так хорошо изучены, как линейные. Не существует универсальных алгоритмов для работы с нелинейными моделями. Различные известные методы могут быть применены лишь к некоторым особым классам нелинейных задач. Зачастую, при работе с нелинейными моделями применяются различные методы линеаризации и уже с полученной линейной системой проводят различные манипуляции. Но любая математическая модель уже не точно описывает реальный объект, а линеаризация влечет за собой дополнительные потери. Именно поэтому важно уметь работать с нелинейными математическими моделями.

Одна из сфер, где используются нелинейные математические
модели – робототехника. Сегодня сложно представить нашу жизнь без различных роботов. Область их применения обширна. Современные роботы, созданные на базе самых последних достижений науки и техники, применяются во всех сферах человеческой деятельности. Люди получили верного помощника, способного не только выполнять опасные для жизни человека работы, но и освободить человечество от однообразных рутинных операций [1]. Основным недостатком всех роботов является высокая цена. Это связано не только с дороговизной различных датчиков и вычислительных элементов, но и со сложностью моделирования этих систем, а также построения управления.

Для построения эффективных законов управления необходимо, в частности, полностью знать вектор состояния системы. Однако зачастую он бывает неизвестен. Одним из способов оценки полного вектора состояния системы является использование асимптотического наблюдателя.

В данной работе рассматриваются вопросы синтеза нелинейных законов управления для двухзвенного робота-манипулятора. Робот-манипулятор представляет собой последовательность звеньев, соединенных сочленениями различного типа, например поступательные или вращательные. На конце манипулятора обычно располагается устройство для выполнения определенного действия.

В работе используется нелинейная модель двухзвенного робота-манипулятора с вращательными сочленениями. Движение робота происходит только в вертикальной плоскости. Проверяется управляемость и наблюдаемость данной модели, строится асимптотический наблюдатель и синтезируются три закона управления: с использованием PD-регулятора, методом линеаризации обратной связью и закон управления с прогнозом. Проводится сравнительный анализ данных законов управления с помощью разработанного имитационно-моделирующего комплекса.

# Глава 1. Постановка задачи

В первой главе вводится математическая модель исследуемого объекта и необходимые теоретические сведения. Проводится анализ управляемости и наблюдаемости системы и ставится задача управления. Так же производится обзор изученной литературы.

## Математическая модель двухзвенного робота-манипулятора

Для описания динамики двухзвенного робота-манипулятора воспользуемся математической моделью из книги [2].



Рис. 1: Двухзвенный робот-манипулятор.

На рис. 1 представлена общая схема данного объекта. При этом приняты следующие обозначения: – масса первого звена, – масса второго звена, – длина первого звена, – длина второго звена, – расстояние до центра масс первого звена, – расстояние до центра масс второго звена, – момент инерции первого звена относительно его центра масс, – момент инерции второго звена относительно его центра масс, – ускорение силы тяжести, – угол между первым звеном и горизонтальной плоскостью, – угол между вторым звеном и продолжением первого звена, – крутящий момент первого двигателя, – крутящий момент второго двигателя.

Для удобства введем дополнительно пять переменных для упрощения записи уравнений динамики робота-манипулятора

; ;;
 ;.

В стандартной форме, используемой во многих источниках, уравнение динамики робота-манипулятора имеет вид

 , (1.1)
где – матрица инерции, – вектор центробежных и кориолисовых сил, – вектор гравитационных сил, – вектор моментов двигателей. Выше перечисленные величины имеют следующий вид

;

;

; .

Так как матрица не вырожденная, то из уравнения (1.1) можем найти явное выражение для :

.

Далее, с помощью замены переменных перейдем от системы двух уравнений второго порядка к системе из четырех уравнений первого порядка

 (1.2)

Здесь , .

Система имеет два измеряемых параметра вектора состояния – обобщенные углы:

 , . (1.3)

## Проверка наблюдаемости и управляемости системы

Проверка наблюдаемости и управляемости нелинейной системы основана на использовании алгебры Ли векторных полей. Наиболее удачно для понимания и применения алгебра Ли векторных полей введена в книге [3].

При описании следующих математических инструментов (алгебры Ли) будем называть векторную функцию векторным полем в , чтобы соответствовать терминологии, используемой в дифференциальной геометрии. В дальнейшем нас будут интересовать только гладкие векторные поля. Под гладкостью векторного поля будем подразумевать, что функция имеет непрерывные частные производные любого требуемого порядка.

 **Производная Ли:** Пусть гладкая скалярная функция и гладкое векторное поле. Тогда производной Ли функции по направлению векторного поля будет скалярная функция

Производные более высоких порядков задаются соотношениями:

Так же, если – векторное поле, то результатом будет скалярная функция

 **Скобка Ли:** Пусть и – два векторных поля в пространстве . Тогда скобкой Ли векторных полей и будет векторное поле

При этом часто используется обозначение: . Производные более высокого порядка задаются соотношениями

Пусть имеется некоторая, вообще говоря, нелинейная система

 (1.4)

где , , ,
 и .

 **Управляемость** нелинейной системы, как и в линейном случае, определяется при помощи ранга матрицы управляемости. В нелинейном случае она состоит из скобок Ли векторных полей

.

 **Индексы наблюдаемости:** если система наблюдаема в точке и если существует такая окрестность и целые числа такие что

1. и ;
2. после некоторой перестановки в каждой точке

мерные векторы линейно независимы,

то целые числа называются индексами наблюдаемости системы.

 **Наблюдаемость** нелинейной системы определяется так же, как и в линейном случае, при помощи матрицы наблюдаемости. В нелинейном случае матрица выглядит следующим образом

где – индексы наблюдаемости системы.

 **Канонический вид.** Если система (1.4) с индексами наблюдаемости приведена к форме

,

то будем говорить, что система находится в каноническом виде.

 Заметим, что система (1.2)-(1.3) уже находится в каноническом виде. Такой вид системы называется каноническим неслучайно. Матрица наблюдаемости системы в таком виде является единичной. Действительно, вычислим матрицу системы (1.2)-(1.3). Для этого перепишем систему (1.2) в виде

где,. Тогда ,, . Отсюда имеем

.

Следовательно, система наблюдаема.

 При исследовании управляемости системы (1.2), при вычислении скобок Ли были получены громоздкие формулы, переписывать которые не имеет смысла. Основной интерес представляет ранг матрицы управляемости . Возьмем первые столбцов матрицы и посчитаем определитель. В результате получим:

Определитель не обращается в ноль, а значит

Следовательно, система (1.2) управляема.

## Постановка задачи управления роботом-манипулятором

Основной задачей для роботов-манипуляторов является перемещение объектов из одной точки в другую. Если говорить формально, то пусть ­–требуемое состояние системы, – начальное состояние системы, (*t*) – вектор управляющих моментов, – состояние системы в момент времени . Тогда потребуем выполнения условия

,

причем, будем полагать, что обобщенные скорости в начальный и конечный моменты времени, соответствующие векторам и , равны нулю.

Задача состоит в том, чтобы отыскать такое управление с обратной связью в области допустимых управлений, при котором достигается поставленная цель.

Основными критериями качества процесса управления являются: перерегулирование, колебательность и время переходного процесса. При синтезе закона управления будем требовать выполнение следующих условий:

1) устойчивость положения равновесия замкнутой системы;

2) отсутствие перерегулирования и колебательности;

3) время переходного процесса не должно превышать некоторой заданной величины .

Так же потребуем, чтобы , где – компактное множество допустимых состояний системы и, чтобы , где – множество допустимых управлений.

В данной работе рассматривается движение робота в вертикальной плоскости. Это допущение сильно упрощает синтез законов управления. Отметим, что для перемещения объектов с помощью робота в трехмерном пространстве достаточно смоделировать вращающуюся подставку.

Одним из простейших методов управления роботами-манипуляторами является использование PD-регулятора. Доказана устойчивость соответствующей замкнутой системы управления с помощью метода функций Ляпунова [3,4].

Метод линеаризации обратной связью с вычислительной точки зрения сложнее вышеизложенного, но в результате мы получаем желаемую линейную динамику замкнутой системы. В отличии от синтеза управления с использованием PD-регулятора, этот метод учитывает нелинейности системы при синтезе управляющего воздействия, что положительно сказывается на качестве управления.

Отметим, что оба указанных метода не являются оптимизационными и не учитывают ограничения, накладываемые на управление и состояние системы. В отличие от них, метод управления с прогнозом (Model Predictive Control, MPC) является оптимизационным и позволяет непосредственно учесть подобные ограничения. Этот метод широко используется в химической и нефтеперерабатывающей промышленности с 1980-го года. В последние годы он также использовался для балансировки энергосистем [12]. Главным преимуществом MPC-подхода является возможность «предвидеть» будущие события и предпринимать необходимые меры в текущий момент времени. Минусом является вычислительная сложность. Вероятность применения данного метода для динамических систем быстро меняющихся со временем крайне мала из-за ограниченности вычислительных ресурсов.

Важно отметить, что полный вектор состояния рассматриваемой системы нам неизвестен, а для реализации законов управления он необходим. В связи с этим следует также построить асимптотический наблюдатель для оценки вектора состояния двухзвенного робота-манипулятора.

## Обзор литературы

Общий вид уравнений динамики робота-манипулятора с *n* звеньями и сочленениями вращательного типа описан в [3,4]. Уравнения динамики двухзвенного робота-манипулятора, полученные с помощью уравнений Лагранжа второго рода, используемые в данной работе, описаны в [2].

Методы построения наблюдателей для нелинейных динамических SISO- и MIMO-систем через преобразование координат рассмотрены в работах [5,6]. Пример построения наблюдателя для нелинейной системы гибкого однозвенного робота рассмотрен в работе [7]. Критерии управляемости и наблюдаемости нелинейных систем, основанные на алгебре Ли векторных полей, рассмотрены в лекциях [5].

В современной теории управления при синтезе законов управления нелинейными системами широко применяются работы таких авторов как Jean-Jacques E. Slotine, Eduardo F. Camacho и др., а для разработки законов управления роботами-манипуляторами – работы M. W. Spong'а. Наиболее известными и широко используемыми методами управления роботами-манипуляторами являются метод линеаризации обратной связью и PD-регулятор. Оба метода подробно рассмотрены в книгах [3,4].

Идея синтеза управления с прогнозом, теоретические сведения, различные алгоритмы синтеза управления, а также примеры приложения методов к задачам описаны в работах [8,9]. Нельзя не отметить методы понижения размерности оптимизационной задачи, рассмотренные в [9].

 Вопросы устойчивости и оптимизации играют важную роль в теории управления. При доказательстве устойчивости использовался теоретический материал, изложенный в книгах [10,11]. Так же в лекциях [10] рассмотрены вопросы оптимальности управления.

# Глава 2. Синтез законов управления для двухзвенного робота-манипулятора

В данной главе рассматриваются вопросы построения асимптотического наблюдателя для системы (1.2) и синтеза нелинейных законов управления для двухзвенного робота-манипулятора.

## Синтез асимптотического наблюдателя

Синтез нелинейного асимптотического наблюдателя является отнюдь не тривиальной задачей. Существуют различные методы синтеза, однако они применимы лишь к особым классам нелинейных задач, обладающих необходимыми свойствами. Поиск этих методов и проверка необходимых свойств – очень кропотливый труд, но полученный результат, обычно, стоит потраченного времени и сил.

Представим систему (1.2)-(1.3) в удобном для построения наблюдателя виде

 (2.1)

Здесь , , ,
. Наблюдатель для системы (2.1) будет искать в виде

 (2.2)

Здесь матрица матрица коэффициентов усиления наблюдателя. Для удобства будем предполагать следующую структуру матрицы .

Динамика вектора ошибки оценивания будет удовлетворять уравнению

 (2.3)

Здесь – вектор ошибки, .

Так как множество допустимых состояний системы – компактное, то функции можно оценить сверху, причем оценку будем искать в виде

 . (2.4)

Тогда из системы (2.3) с учетом оценок (2.4) получим систему

 , (2.5)

где . Исходя из условий устойчивости системы (2.5) будем искать коэффициенты матрицы усиления наблюдателя (2.2).

 Произведем оценку ,

Для начала оценим разность

где , – границы множества допустимых состояний (1.2). Аналогично оценим разность

Получим

Следовательно,

,

При оценке

,

не трудно видеть, что при , доставляющем максимум , функция и наоборот, при , доставляющем максимум ,
 . Таким образом, достаточно найти такие значения матрицы усиления , которые обеспечивали бы устойчивость (2.5) для и .

Отметим, что коэффициенты усиления будут достаточно большими, вероятно это связано с грубостью оценки. Однако, существует так же альтернативный метод, основанный на том, что функция – нелинейная липшицева функция, с константой Лишица то есть

 ,
где – множество допустимых состояний системы (2.1), – некоторые константы.

 Наблюдатель для системы (2.1) будем строить в следующем виде

 . (2.6)

Тогда динамика ошибки оценивания будет удовлетворять уравнению

 ,

где – вектор ошибки.

Исходя из теоремы 2 [13], будем искать матрицу в виде

где – симметричная положительно определенная матрица, являющаяся решением неравенства Риккати

 (2.7)

Здесь и – вещественные числа.

 Для удобства дальнейшего использования преобразуем (2.7) как показано ниже

 (2.8)

Здесь – некоторая константа. Для решения уравнения Риккати воспользуемся стандартной функцией

решающей уравнение вида

Тогда полученное уравнение (2.8) приведем к необходимому виду

Матрицу найдем как

Тогда выбор матрицы обеспечивает асимптотическую устойчивость наблюдателя (2.6) системы (2.1)[13]. Однако, вероятно из-за неопределенности выбора констант и найти подходящее решение не удалось, поэтому в работе реализован и используется первый метод.

## Синтез PD-регулятора

PD-регулятор состоит из двух составляющих: пропорциональной и дифференциальной. Первая составляющая вырабатывает сигнал, противодействующий отклонению регулируемой величины от заданного значения, а вторая – сигнал, противодействующий отклонениям системы, которые прогнозируются в будущем.

В этом параграфе вместо обозначения вектора состояния системы как
 удобней использовать два вектора
 и . А систему представить в виде

 . (2.9)

Синтезируем закон управления для системы (2.9), состоящий из PD-регулятора и компенсации силы тяжести

 , (2.10)

где , – постоянные положительно-определенные диагональные матрицы, – вектор ошибки, – желаемое состояние системы.

Для доказательство устойчивости системы (2.9) с законом управления (2.10) и сходимости системы к желаемому положению введем квадратичную форму – кандидата на функцию Ляпунова

Здесь первое слагаемое – это кинетическая энергия робота. Второе слагаемое учитывает пропорциональную обратную связь . Заметим, что значение
 – это полная кинетическая энергия системы, в которой вращательные сочленения были заменены на пружины с жесткостью, представленной матрицей , и положением равновесия в точке .

 Вычислим полную производную функции

 (2.11)

Теперь выразим из (2.9) и подставим в (2.11):

Здесь использовался тот факт, что матрица является кососимметричной матрицей [2]. Теперь подставим закон управления с
PD-регулятором (2.10) вместо . В результате получим

 (2.12)

Данный результат показывает, что функция убывает, если вектор . Этого недостаточно для доказательства желаемого результата, поскольку вполне вероятно, что манипулятор может достичь положения, где , но . Предположим, что . Тогда из (2.12) следует, что , отсюда следует, что . Тогда из уравнения движения робота-манипулятора (2.9) с законом управления (2.10) получаем . Из того, что матрица положительно определенная и невырожденная следует, что А из теоремы Барбашина–Красовского [11] следует, что полученное тривиальное решение , является асимптотически устойчивым.

 Как уже упоминалось ранее, данный метод не является оптимизационным и не учитывает ограничения ни на управляющий сигнал, ни на компоненты вектора состояния. Для получения желаемого управления производится настройка PD-регулятора подбором коэффициентов матриц и .

## Метод линеаризации обратной связью

Метод линеаризации обратной связью, то есть устранение нелинейности и получение желаемой линейной динамки, может быть применен к классу нелинейных систем, которые находятся в канонической форме. В рассматриваемой задаче математическая модель объекта управления имеет каноническую форму, поэтому мы можем использовать данный метод. Для пояснения его основной идеи сначала рассмотрим пример системы со скалярным входом и состоянием, а затем обобщим на многомерный случай.

Пусть имеется система

,

которая представима в виде

Тогда, при условии, что , примем закон управления в форме

В результате приходим к уравнению . Теперь сформируем регулятор вида

,

где коэффициенты выбираются таким образом, чтобы корни полинома

находились в левой полуплоскости, что обеспечивает экспоненциальную устойчивость нулевого положения равновесия для уравнения

а это значит, что . Если в задаче требуется достичь желаемого состояния , то закон управления зададим следующим образом:

где – вектор отклонения от заданного положения. При этом новое положение равновесия будет также экспоненциально устойчивым.

Полученные результаты справедливы и для векторного случая.
Для удобства введем новые векторы , . Тогда система (1.2) может быть записана в виде

где

, , .

Заметим, что . Будем искать управление в виде

 (2.13)

Сформируем функцию следующим образом

,

где – задают желаемое состояние системы, , – векторы ошибок, и – некоторые диагональные матрицы, обеспечивающие нахождение вещественных частей корней системы

 (2.14)

в левой полуплоскости. В [3,4] предлагается искать матрицы в виде

 , , (2.15)

потому что корнями системы (2.14) в этом случае будут числа .

 Метод линеаризации обратной связью также не является оптимизационным и не учитывает ограничения ни на управляющий вектор, ни на компоненты вектора состояния. Настройка производится подбором коэффициентов матриц , .

## Синтез регулятора с прогнозом

Данный закон управления синтезируется для дискретной модели. Связано это с тем, что управление с прогнозирующими моделями практически реализуются только на базе цифровых элементов – иные варианты в настоящее время не представляются актуальными [9].

Горизонтом прогноза будем называть некоторую величину , обозначающую количество прогнозируемых тактов. Горизонтом управления будем называть величину , , такую что В рамках данной работы будем предполагать, что

Пусть задана математическая модель динамики объекта в дискретном времени

 (2.16)

Здесь – текущий момент дискретного времени, задающий соответствующий момент непрерывного времени, где – период дискретности, – текущее состояние, – текущее управление и – текущее измерение. Здесь и – множества допустимых состояний и управлений системы соответственно.

 Будем считать, что целью управления объектом (2.16) является обеспечение выполнения равенств

 (2.17)

где – заданные последовательности векторов, определяющие желаемое движения объекта.

 На движениях объекта (2.16) зададим некоторый функционал

 (2.18)

определяющий качество процессов управления.

 Будем рассматривать задачу о синтезе управления для объекта (2.16), которое обеспечивает достижение цели (2.17) и доставляет минимум функционалу (2.18).

 Наряду с математической моделью (2.16) введем систему разностных уравнений

 (2.19)

Здесь – текущее состояние, – текущее управление и – текущее измерение, .; и – множества допустимых состояний и управлений системы соответственно.

 Будем полагать, что функция системы (2.19) задана таким образом, что для любой последовательности управлений соответствующие последовательности и векторов состояний систем (2.19) и (2.16) близки между собой. Систему (2.19) будем называть прогнозирующей для системы (2.16).

 Прогнозирующая модель инициируется на начальном такте текущим состоянием системы (2.16) и, при заданной последовательности управлений {*u*[*k*]}, позволяет приближенно спрогнозировать поведение реального объекта на горизонте , путем нахождения решения системы (2.19) для моментов . Таким образом, получаем конечную последовательность управлений , и соответствующую ей конечную последовательность векторов состояния , . Будем говорить, что при этом сформирован прогноз движения реального объекта с горизонтом .

 Рассмотрим функционал, характеризующий качество управления прогнозирующей моделью на горизонте прогноза

 (2.20)

где

,

.

Функционал представляет собой исходный функционал , заданный на конечном -м горизонте прогноза.

 Последовательностью допустимых управлений будем называть такую последовательность , каждый элемент которой принадлежит множеству допустимых управлений . Последовательность допустимых управлений будем называть оптимальной, если она доставляет минимум функционалу (2.18), то есть

Здесь – последовательность состояний системы, соответствующих последовательности управлений .

 По аналогии, конечную последовательность допустимых управлений , будем называть оптимальной, если она доставляет минимум функционалу (2.20), то есть

Здесь

,

.

Конечная последовательность состояний , соответствует конечной последовательности допустимых управлений , .

 Так как движения системы однозначно определяются последовательностью управлений , то . В результате получаем оптимизационную задачу размерности

 (2.21)

Несмотря на то, что обычно прогнозирующая система выбирается с вычислительной точки зрения достаточно простой, задача минимизации функции переменных сложна, поэтому в некоторых задачах необходимо применять методы понижения размерности [9].

 В данной работе для понижения размерности оптимизационной задачи (2.21) выберем период дискретности управления кратным периоду функционирования системы . То есть будет верно соотношение , где . Таким образом получим размерность оптимизационной задачи, где .

 В итоге, управление будет осуществляться по следующему алгоритму:

1. Оценивается текущее состояние объекта (2.16) .
2. Выбором управления на горизонте прогноза оптимизируется движение прогнозирующей модели (2.19) по отношению к функционалу (2.20).
3. Из полученной оптимальной программной последовательности векторов управления в качестве управляющего воздействия на текущем такте для системы (2.16) выбирается только первый вектор.
4. На следующем такте переходим к пункту 1.

 В рассматриваемой задаче в качестве объекта управления и прогнозирующей модели будет использована система (1.2), дискретизированная с помощью метода Рунге-Кутты с высоким и низким порядком точности соответственно. Множество допустимых управлений определим некоторыми линейными и нелинейными ограничениями на компоненты контролируемых и управляющих переменных. В качестве функционала (2.20) примем квадратичный функционал

 , (2.22)

где и – диагональные матрицы необходимых размерностей.

# Глава 3. Практический результат

Процесс моделирования динамки движения робота-манипулятора на основе модели (1.2) с выходом (1.3), наблюдателем (2.2) и законами управления, описанными в главе 2, осуществлялся в среде Matlab (Simulink). В первом параграфе данной главы представлены блок-схемы программ и описаны некоторые скрипты, использованные в работе. Во второй главе продемонстрированы результаты моделирования и произведено сравнение результатов

## Имитационно-моделирующий комплекс

Для удобства моделирования были реализованы несколько скриптов, представленных в [14]. Результатом работы RobotArmModelSym.m является модель динамики робота-манипулятора (1.2) в символьных переменных. Полученная модель используется при проверке наблюдаемости и управляемости системы с помощью скриптов ObsAbil.m и CntrlAbil.m соответственно. Элементы алгебры Ли, а именно скобки Ли и производная Ли реализованы в скриптах LieBrackets.m и LieDerivative.m.



Рис. 2. Simulink-модель робота-манипулятора с наблюдателем и управлением.

При задании всех необходимых параметров можно воспользоваться скриптом RobotArmModel.m, результатом работы которого является файл RobotSystem.m, содержащий правые части уравнения динамики.

Моделирование динамики системы (1.2) с законами управления (2.10) и (2.13) осуществляется с помощью блок-схемы, представленной на рис. 2.



Рис. 3. Simulink-модель подсистемы system.

Подсистема «system» (рис. 3) описывает динамику робота-манипулятора (1.2). На вход, данный блок, принимает вектор управляющих воздействий τ, выходом является вектор измеримых параметров y, согласно (1.3).



Рис. 4. Simulink-модель подсистемы observer.

Подсистема «observer» (рис. 4) – наблюдатель (2.2). На вход принимается вектор y измеримых параметров системы (1.2). Выходом является оцененный полный вектор состояния системы (1.2).

В подсистеме «control» (рис. 5) осуществляется синтез закона управления (2.10), если используется PD-регулятор, или (2.13) если применяется линеаризация обратной связью. На вход блок принимает вектор оцененного полного состояния системы. Выходом является вектор управляющих воздействий.



Рис. 5. Simulink-модель подсистемы control.

Моделирование динамики робота-манипулятора (1.2) с управлением с прогнозирующей моделью осуществляется с помощью скрипта startscript.m (листинг 1).

## Результат вычислительных экспериментов для разработанных алгоритмов

При моделировании будем использовать следующие параметры для (1.2)

|  |  |
| --- | --- |
| Параметр: | Значение: |
|  | 1 |
|  | 1 |
|  | 0.5 |
|  | 0.5 |
|  | 0.12 |
|  | 0.12 |
|  | 1 |
|  | 1 |
|  | 10 |

Таблица 1. Параметры компьютерной модели

Сначала продемонстрируем работу наблюдателя (2.2) системы (1.2). Множество допустимых состояний системы (1.2) зададим следующим образом

 (3.1)

Следовательно M = 5. Проведем три эксперимента с разными начальными состояниями системы (1.2) и наблюдателя (2.2). Результат демонстрируется на графиках (рис.6, рис. 7, рис. 8). Здесь – вектор ошибок оценки, где , – компонента состояния системы, – компонента вектора оценки,



Рис. 6. График ошибки оценки с начальными условиями.

Как видно из графиков, ошибки по компонентам обобщенных скоростей, , даже при достаточно большом начальном отклонении быстро сходятся к нулю. А ошибки по компонентам обобщённых углов, , хотя и сходятся к нулю, но влекут за собой сильное отклонение оценки обобщённых скоростей. Хотя в обоих случаях оценка сходится за доли секунды. Скорее всего, такая динамика ошибки связана с большими значениями коэффициентов усиления.

 Рис. 7. График ошибки оценки с начальными условиями.

Рис. 8. График ошибки оценки с начальными условиями.

Перед реализацией законов управления введем необходимые параметры. Максимальное время переходного процесса , множество допустимых состояний зададим как (3.1), множество допустимых управлений , сначала введем неограниченным .

Приступим к настройке управлений. Начнем с выбора коэффициентов усиления PD-регулятора (2.10). Пусть , здесь – единичная матрица. Для настройки управления линеаризацией обратной связью (2.13) нужно подобрать параметр (2.15). Пусть К сожалению, реализация MPC подхода позволяет выбирать управление лишь на заданном ограниченном промежутке, но можно выбрать достаточно большой промежуток, например пусть . Настройка данного закона управления осуществляется заданием коэффициентов матриц и (2.22), определением величины горизонта прогноза , а также наложением различных ограничений, в том числе и нелинейных, на вектор состояния объекта, вектор управляющих воздействий, или на движение системы. Пусть , , , а так же зададим нелинейное ограничение на движение (листинг 5).

**

Рис. 9. Движение системы с разными законами управления, множество .

Как видно из графиков движения системы (1.2) на рис. 9, все переходные процессы завершаются до момента , однако движения системы под управлением PD-регулятора и MPC имеют перерегулирования. Это связано с тем, что PD-регулятор не учитывает влияние динамики первого звена на второе, а MPC подход считается численно, и, вероятно, при решении оптимизационной задачи был найден некий локальный минимум, который вызвал данную нежелательную динамику. Это также отразилось на управляющем сигнале. На рис. 10 видно, что управляющий сигнал MPC делает резкие скачки в момент времени близкий к начальному.

**

Рис. 10. Управляющий сигнал разных законов управления, множество .



Рис. 11. Движение системы с разными законами управления, . Без перенастройки законов управления, не учитывающих ограничения.

Теперь ограничим множество допустимых управлений

и повторим эксперименты. Ожидалось, что оба метода синтеза законов управления, не учитывающие ограничения, буду вести себя неадекватно, если их не перенастроить, однако на рис. 11 видно, что PD-регулятор показывает хороший результат без перенастройки с небольшими перерегулированиями как и раньше. Чего нельзя сказать о методе линеаризации обратной связью,



Рис. 12. Управляющий сигнал разных законов управления, .
Без перенастройки законов управления, не учитывающих ограничения.

появились неприемлемые перерегулирования. Управляющий сигнал этих методов продемонстрирован на рис. 12.

 Перенастроим метод линеаризации обратной связью с помощью уменьшения параметра . Остальные методы оcтавим без изменений. Пусть . При дальнейшем уменьшении переходный процесс затягивается, но перерегулирование убывает.



Рис. 13. Движение системы с разными законами управления, , .

Рис. 12. Управляющий сигнал разных законов управления, , .

Из экспериментов видно, что MPC подход показывает себя лучше, чем другие методы управления, реализованные в данной работе. Однако, стоит заметить, что время моделирования системы с управлением с прогнозирующей моделью в сотни раз превышает время моделирования систем с управлением с помощью PD-регулятора и линеаризации обратной связью.

# Выводы

Как показал сравнительный анализ, каждый из методов имеет ряд преимуществ и недостатков.

1. PD-регулятор прост с вычислительной точки зрения, универсален и удивительно хорошо справился с ограничениями на управление. Но не учитывает особенности системы, настройка осуществляется подбором коэффициентов.
2. Метод линеаризации обратной связью использует информацию о системе, относительно простой с вычислительной точки зрения. Но удивительно плохо реагирует на ограничения на управляющий сигнал, настраивается вручную подбором параметров.
3. Закон управления с прогнозом является оптимизационным, учитывает ограничения на вектор состояния и управления, имеет достаточное количество параметров для настройки, что позволяет добиться требуемых качеств управления. Но очень сложен с вычислительно точки зрения, что является весомым недостатком при управлении быстро протекающими процессами.

В будущем планируется рассмотреть другие методы нелинейного управления, например backstepping и sliding mode control.

 При выполнении работы были получены следующие результаты.

1. Построена имитационная модели двухзвенного-робота манипулятора.
2. Для этой модели синтезированы: асимптотический наблюдатель, закон управления с использованием PD-регулятора, закон управления методом линеаризации обратной связью и закон управления с прогнозом.
3. Проведено сравнение результатов имитационного моделирования.

# Список литературы

1. Макаров И. М., Топчеев Ю. И. Робототехника. История и перспективы. М. : Наука, 2003. 349 с.
2. FantoniI., LozanoR. Nonlinear Control forUnderactuated Mechanical Systems. London: Springer-VerlagLondonLtd., 2002. P. 301.
3. Slotine J. E., LiW. Appliednonlinear control. Prentice-Hall, Inc, 1991. P. 476.
4. Spong M. W., Hutchinson S., Vidyasagar M. Robot Dynamics and Control. 2004. P. 303.
5. Наблюдатели часть 1 // Московский Государственный Технический Университет имени Н.Э.Баумана URL: http://www.bmstu.ru/ps/\~s\\_tkachev/fileman/ls/Математические методы теории управления/Наблюдатели часть 1.pdf (дата обращения: 01.01.2017).
6. Krener A.J., Respondek W. Nonlinear observers with linearizableerror dynamics // SIAM J. Control and Optimization. 1985. V. 23, №2. P. 197 - 216.
7. Голубев А. Е., Крищенко А. П., Ткачев С. Б. Принцип разделения для аффинных систем // Дифференциальные уравнения. 2001. Т. 37. №. 11. С. 1468-1475.
8. Camacho E. F., Bordons C. Model Predictive Control. London: Springer-Verlag London Ltd., 2007. P. 422.
9. Веремей Е. И., Сотникова М. В. Управление с прогнозирующими моделями. Воронеж: Издательство «Научная книга», 2016. 214 с.
10. Зубов В. И. Лекции по теории управления. Главная редакция физико-математической литературы. 1975. 495 с.
11. Яковенко Г. Н. Лекции по теоретической механике. 2003. 187 с.
12. Arnold M., Andersson G. Model predictive control of energy storage including uncertain forecasts // Power Systems Computation Conference (PSCC), Stockholm, Sweden. 2011. Т. 23. С. 24-29.
13. Rajamani R. Observers for Lipschitz nonlinear systems // IEEE Trans. Automat.Contr. 1998. V. 43. No3. P. 397–401.
14. GitHub URL: https://github.com/Au6ojlut/forrobotarm (дата обращения: 15.05.2017).

# Приложение

В этом разделе приведены исходные коды программ на языке MATLAB, использовавшихся в ходе работы. В листинге 1 приведен код скрипта, реализующего метод MPC.

**Листинг 1**

%startscript.m

clear all;

t = 3; delta = 0.01;%t – время функционирования; delta – шаг дискретизации системы

Tp = 1; %Горизонт прогноза

xk(:,1)=[0;0;0;0]; %начальное состояние

xd = [pi/2;0;0;0]; %желаемое состояние

global iter xr xrr Tc

Tc = 0.05; %шаг дискретизации прогнозирующей модели

iter = double(int32(Tp/Tc));

xr = repmat(xd(1:2:end),iter,1);

xrr = repmat(xd,iter,1);

first = (xd(1) - xk(1,1))\*180/pi;

second = xd(3) - xk(3,1) + 0.3 \*( xd(1) - xk(1,1))\*180/pi;

u0 = [sign(first) \* max(abs(first),50), sign(second) \* max(abs(second),50)];

uk = u0;

for i= 2:iter

 uk = [uk; u0\*(1-Tc\*0.1)^double(i)];

end

for i = 1 : t/delta

 uopt= RobotOptControl(xk(:,i),Tp,Tc,uk);

 un(:,i) = uopt(1,:)';

 [tp, yp] = ode45(@(t,x) RobotSystem(t,x,un(:,i)),[0 delta],xk(:,i));

 xk(:,i+1) = yp(end,:);

 uk = [uopt(2:end,:);uopt(end,:)\*(1-Tc\*0.1)];

 fprintf('Step %d\n', i);

 fprintf('%d\n', xk(1:2:end,i+1)\*180/pi);

 fprintf('%d\n', un(:,i));

end

В листингах 2-5 представлены коды программ, используемых в
листинге 1.

**Листинг 2**

function u = RobotOptControl(x,Tp,Tc,u0)

umax = [50 50];

umin = [-50 -50];

global iter xrr xr

ust = u0;

options = optimoptions('fmincon', 'Algorithm','sqp', 'MaxIter', 2000,'MaxFunEvals',15000, 'UseParallel', 'always');

[y,fval] = fmincon(@(u) J(x,xrr,u,Tp,Tc,iter),ust,[],[],[],[],repmat(umin,iter,1), repmat(umax,iter,1), @(u) noncon(u,Tc,iter,x,xr),options);

u = y;

end

**Листинг 3**

function A = J(xk,xrr,u,Tp,Tc,iter)

n = length(xk);

for i = 1 : iter

 uk = u(i,:);

 [tp, yp] = ode23(@(t,x) RobotSystem(t,x,uk),[0 Tc],gather(xk));

 xk = yp(end,:);

 x\_pred(:,i) = xk;

end

x\_pred = reshape(x\_pred,double(iter) \* n,1);

u = reshape(u',iter\*n/2,1);

x\_err = (x\_pred - xrr)\*180/pi;

x\_err(2:2:end) = x\_err(2:2:end) \* 0.005;

A =(x\_err' \* x\_err)+u' \* u;

end

**Листинг 4**

function X = RobotSystem(t,x,u)

q1 = x(1); dotq1 = x(2); q2 = x(3); dotq2 = x(4); tau1 = u(1); tau2 = u(2);

X(1) = dotq1;

X(2) = (3700.\*(5.\*cos(q1 + q2) + 15.\*cos(q1) - dotq2.\*((dotq1.\*sin(q2))./2 + (dotq2.\*sin(q2))./2) - (dotq1.\*dotq2.\*sin(q2))./2))./(2500.\*cos(q2).^2 - 5069) - (3700.\*tau1)./(2500.\*cos(q2).^2 - 5069) - (100.\*(5.\*cos(q1 + q2) + (dotq1.^2.\*sin(q2))./2).\*(50.\*cos(q2) + 37))./(2500.\*cos(q2).^2 - 5069) + (100.\*tau2.\*(50.\*cos(q2) + 37))./(2500.\*cos(q2).^2 - 5069);

X(3) = dotq2;

X(4) = (200.\*(5.\*cos(q1 + q2) + (dotq1.^2.\*sin(q2))./2).\*(50.\*cos(q2) + 87))./(2500.\*cos(q2).^2 - 5069) - (100.\*(50.\*cos(q2) + 37).\*(5.\*cos(q1 + q2) + 15.\*cos(q1) - dotq2.\*((dotq1.\*sin(q2))./2 + (dotq2.\*sin(q2))./2) - (dotq1.\*dotq2.\*sin(q2))./2))./(2500.\*cos(q2).^2 - 5069) + (100.\*tau1.\*(50.\*cos(q2) + 37))./(2500.\*cos(q2).^2 - 5069) - (200.\*tau2.\*(50.\*cos(q2) + 87))./(2500.\*cos(q2).^2 - 5069);

X = X';

**Листинг 5**

function [c, ceq] = noncon(u,Tc, iter,xk,xr)

n=length(xk);

%eps = 0.0001;

x0 = xk;

for i = 1 : iter

 uk = u(i,:);

 [tp, yp] = ode23(@(t,x) RobotSystem(t,x,uk),[0 Tc],xk);

 xk = yp(end,:);

 x\_pred(:,i) = xk;

end

xd\_pred = reshape(x\_pred(1:2:end,:),iter \* n/2,1);

x\_pred = reshape(x\_pred(1:2:end,:),iter \* n/2,1);

err\_d = abs(xr - x\_pred);

err\_k = abs(xr - repmat(x0(1:2:end),iter,1));

c = max([err\_d - err\_k, xd\_pred-5]’);

ceq = [];

В листинге 6 представлен код программы для получения файла RobotSystem.m, содержащего правые части системы (1.2)

**Листинг 6**

clear all

m1 = 1; m2 = 1;

lc1 = 0.5; lc2 = 0.5;

I1 = 0.12; I2 = 0.12;

l1 = 1; l2 = 1;

g=10;

q = [q1;q2]; dotq = [dotq1; dotq2]; tau = [tau1; tau2];

m = [m1;m2]; lc = [lc1;lc2]; I = [I1;I2]; l = [l1;l2];

theta = [m(1)\*lc(1)^2 + m(2)\*l(1)^2 + I(1); m(2) \* lc(2)^2 + I(2); m(2)\*l(1)\*lc(2); m(1)\*lc(1) + m(2)\*l(1); m(2)\*lc(2)];

H = [theta(1) + theta(2) + 2 \* theta(3)\*cos(q(2)), theta(2) + theta(3) \* cos(q(2));

 theta(2) + theta(3) \* cos(q(2)), theta(2)];

C = [-theta(3) \* sin(q(2)) \* dotq(2), -theta(3) \* sin(q(2)) \* dotq(2) - theta(3) \* sin(q(2)) \* dotq(1);

 theta(3) \* sin(q(2)) \* dotq(1), 0];

G = [ theta(4) \* g \* cos(q(1)) + theta(5) \* g \* cos(q(1) + q(2)); theta(5) \* g \* cos(q(1) + q(2))];

InvH = H^(-1);

tmpsys = InvH \* (-C \* dotq - G +tau);

sys = [dotq(1);tmpsys(1);dotq(2);tmpsys(2)];

x = [q(1);dotq(1);q(2) ;dotq(2)];

Cx = [1 0 0 0; 0 0 1 0];

 obs = Cx \* x;

%creating file

string{1} = 'function X = RobotSystem(t,x,u)';

string{2} = 'q1 = x(1); dotq1 = x(2); q2 = x(3); dotq2 = x(4); tau1 = u(1); tau2 = u(2);'; %define parametrs

string{3} = ['X(1) = ' vectorize(sys(1)) ';']; %define function

string{4} = ['X(2) = ' vectorize(sys(2)) ';'];

string{5} = ['X(3) = ' vectorize(sys(3)) ';'];

string{6} = ['X(4) = ' vectorize(sys(4)) ';'];

string{7} = ['X = X'';'];

openedfile = fullfile(pwd, 'Simulink models/MPC/RobotSystem.m');%change way if needed

fid = fopen(openedfile, 'w');

fprintf(fid, '%s\n', string{:});

fclose(fid);