Санкт-Петербургский государственный университет

**Кафедра компьютерных технологий и систем**

**Лан Синчэн**

**Магистерская диссертация**

**Управление автомобилем на траектории с использованием нечёткой логики**

Направление 02.04.02

Фундаментальные информатика и информационные технологии

Магистерская программа Автоматизация научных исследований

Научный руководитель,  
кандидат физ.-мат. наук,  
доцент  
Жабко Н. А.

Санкт-Петербург

2016

# Содержание

Введение 3

Постановка задачи 6

Обзор литературы 9

Глава 1. Синтез законов управления автомобилем на траектории 10

1.1. Линеаризация 10

1.2. Синтез закона LQR-оптимизации 12

1.3. Автоматический выбор параметров в управлении 14

1.4. Построение TS-модель нечёткой логики 16

1.5. Синтез нечеткого управления (Fuzzy Controller) 20

Глава 2. Компьютерное и имитационное моделирование 24

2.1. Выборка точки равновесии 24

2.2. Simulink-модель непрерывной системы 24

2.3. Автоматическая настройка коэффициентов управления 27

2.4. Задание траектории 31

2.5. TS-нечеткая модель 33

2.6. Увеличение законов для нечеткого управления 36

2.7. Эксперимент для увеличения законов 39

2.8. Проверка асимптотической устойчивости 44

Выводы 48

Заключение 49

Список литературы 50

Приложение 52

# Введение

В прошлом веке с момента появления автомобиля он сразу стал важной частью повседневной жизни человека. С развитием современной промышленности в сфере машиностроения количество владельцев автомобилями во всех странах мира характеризовалось существенным ростом, и со временем проблему трафика стало невозможно игнорировать в градостроительстве. В классических моделях система «машина – водитель - дорога» является замкнутой системой с обратной связью. Увеличивающийся объем потребности в использовании возможностей современных автомобилей, растущие требования к комфорту и безопасности, делают актуальным направление исследований, ориентированное на то, как исключить человека из этой системы. На всех этапах автоматизации задача исследования динамики движения автомобиля является необходимой, при этом при исследованиях необходимо основываться на знаниях теории управления и моделирования, а также возможностях современных компьютерных и информационных технологий.

Задача управления на траектории и на заданном пути стала популярной и актуальной в настоящее время, в том числе речь идет об автоматизации такого движения для автомобилей. Люди придумают новые способы для дальнейшего увеличения удобства в жизни. Например, американская компания «Google» изобрела «Google Car» ­ автомобиль, который может сам доехать до желаемого места.

Развитие компьютерных технологий позволяет нам реализовывать синтез закона автоматического управления в устройствах системы управления, которые будут занимать место водителя или работать помощником в опасной ситуации. А методы, основанные на использовании нечеткой логики возникли как наиболее удобный и понятный способ построения систем управления различными системами, в том числе автомобилем и сложными технологическими процессами, и нашли применение в бытовой электронике, диагностических и других экспертных системах. При этом актуальной является задача автоматизации процесса разработки систем автоматического управления с нечеткой логикой, используемых на борту автомобиля, а также разработки простых вычислительных схем, легко реализуемых на борту.

Данная работа посвящена вопросам, связанным с синтезом нечеткого управления на траектории автомобилем, с использованием для его настройки средств компьютерного и имитационного моделирования. Во многих работах нечеткая логика применяется с целью реализовать интуитивные представления о поведении объекта и имеющийся опыт в нелинейном нечетком законе управления, формальный анализ свойств которого (таких как устойчивость замкнутой системы) провести, как правило, не представляется. В данном исследовании, в отличие от таких работ, нечеткая логика используется для замены исходной нелинейной модели с помощью нескольких линейных, и, аналогично, получение нечеткого регулятора TS) на основе нескольких линейных управлений. При этом такая замкнутая система может быть формально исследована на предмет глобальной асимптотически устойчивой, что делает подобный подход перспективным с точки зрения применения на практике. Для синтеза регулятора в работе также привлекаются средства классического метода LQR-оптимизации.

Для проведения исследований в работе создается компьютерная модель объекта с помощью системы Simulink. Simulink – это графическая среда имитационного моделирования, позволяющая при помощи блок-диаграмм в виде направленных графов строить динамические модели любой сложности, включая дискретные, непрерывные и гибридные, нелинейные и разрывные системы.

Построенная Simulink-модель используется для проведения численного моделирования при синтезе законов управления на траектории.

При этом возможность многократно повторять похожие эксперименты с объектом посредством имитационного моделирования существенно упрощает задачу выбора параметров в законах управления, проверки получаемых результатов, и позволяет проводить автоматизацию разработки элементов системы управления автомобилем.

# Постановка задачи

## 1. Построение математической модели

В работе рассматривается, что автомобиль двигается по плоской дороге, сухой и без возмущений, таких как ветер, снег и дождь. В качестве входного сигнала, играющего роль управления, принимается тяговое усилие или тормозное усилие *T*, и угол поворота переднего колеса . Тогда движение автомобиля можно представить с помощью следующих четырех параметров: продольная скорость, поперечная скорость, угловая скорость по курсу , и угла курса .

На основе известных законов теоретической механики, описывающих движение автомобиля, могут быть получены уравнения, которые приведены ниже ([2],[3]):

Здесь использованы различные коэффициенты, которые характеризуют параметры автомобиля: – передняя колесная база, – задняя колесная база, – высота, – масса автомобиля, – коэффициент трения, – момент инерции, – ускорение силы тяжести, – жесткость угла крена передней подвески, – жесткость угла крена задней подвески, – параметр перевозки, – параметр сопротивления. При моделировании задается ограничение .

Модель является MIMO и сильно нелинейной.

## 2. Задача стабилизации на траектории (path-following)

В данной работе изучаются вопросы управления автомобилем на траектории (заданном пути). На рисунке 1 показана целевая конфигурация представления эталонного автомобиля с теми же ограничениями, как кинематики реального.

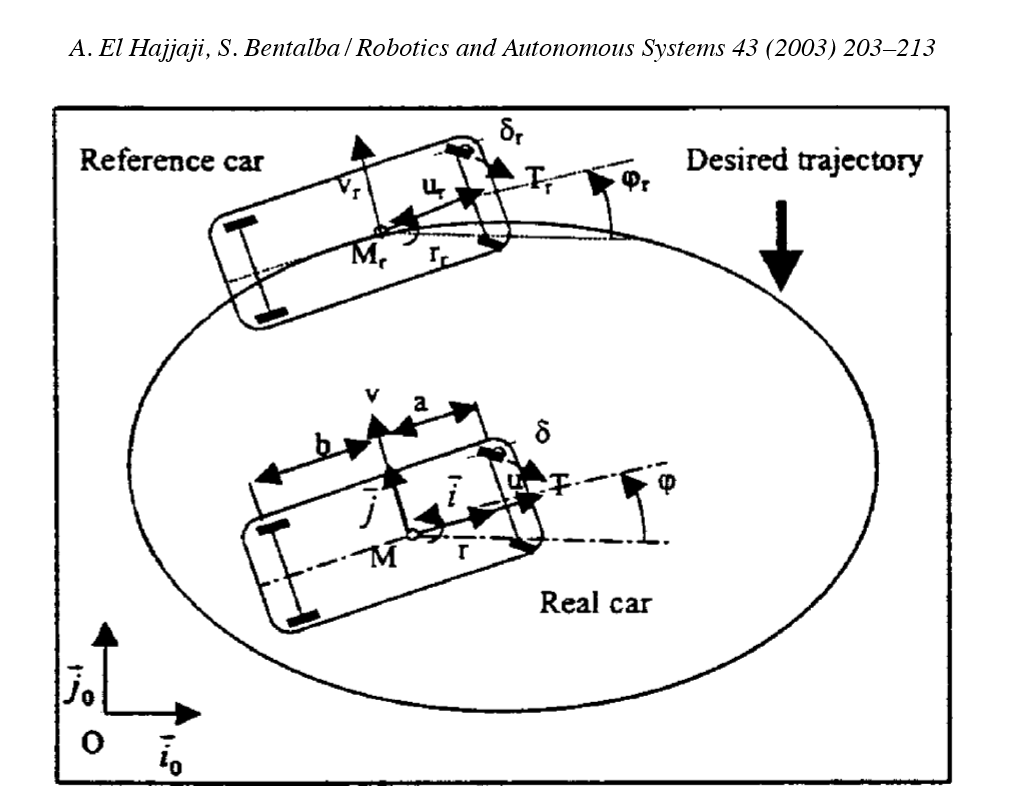


Рис. 1.

Соответственно, (, , ) и ( , , ), будут продольные, поперечные скорости и угол курса реального и желаемого автомобиля.

Наша цель состоит в определении управляющих воздействий и , которые позволяют реальному автомобилю следовать определенной траектории. Иными словами, пусть (,) являются координатами ошибки вектора положения в системе координат (M, ,), связанной с реальным автомобилем. Обозначим ошибку ориентации между обоих транспортных средств = φ − . Так что вектор ошибки положения можно описываться в системе координат таким образом:

Продифференцируем по времени, получим:

Тогда,

Кроме того, у нас есть

где

Из вышеприведенных уравнений получим следующее:

Кроме того,

Наконец, представление для задачи траектории автомобиля могут быть записаны следующим образом:

# Обзор литературы

Работа Гуо КонХуэй «Динамика управления автомобилем»(1991)[4] содержит характеристики автомобиля, позволяет составить представление о построении математической модели автомобиля на основе исходной физической модели. Постановка задачи управления автомобилем на траектории, которая рассматривается в настоящей работе, а также основная идея, на которой основано решение задачи в данной работе, приведены в работе A.El Hajjaji,S.Bentalba [3]. Эта работа совместно с работой S.Bentalba, A. El Hajjaji and A. Rachid [2] посвящена использованию нечеткой логики при управлении автомобилем, нелинейная модель, используемая при проведении вычислений, также сформирована на основе этих работ. Монография Веремея.Е.И «Линейные системы с обратной связью»(2013)[1] дает мне теоретические знания про оптимизационные методы, которые нашли применение в настоящей работе для синтеза управления движения автомобиля. Книга Ван ЧжэнЛин «Процесс управления и применение в Simulink»(2006)[5] и книга Liu Bao, Tang WanSheng «Современная теория управления»[6] помогает решить проблемы моделирования, которое применяется в среде MATLAB-Smiulink. Статья «Matlab и система нечеткой управления»[7] объясняет понятия нечеткой логики, и, совместно с работой Zhao JingHui «T-S нечеткая модель»[8], а также [13], позволяет понятие о применении T-S нечеткой модели. В работе Dao Ke «Определение LMI-системы с использованием lmiedit» [9], а также работа Dynamic «Решение линейных матричных неравенств в среде Matlab»[10] изложены вопросы, посвященные проблеме решения системы линейных матричных неравенств, в том числе в среде MATLAB. Нечеткие T-S модели успешно используются в разных приложениях и при учете некоторых дополнительных условий, примеры такого использования приведены, например, в [11]-[15], однако из указанных работ видно, что многие вопросы их использовании остаются открытыми и не имеют единственного решения.

# Глава 1. Синтез законов управления автомобилем на траектории

В данной главе опишем теоретические основы тех подходов, которые будут использоваться при проведении компьютерного и имитационного моделирования управления автомобилем на траектории.

## 1.1. Линеаризация

При выполнении синтеза регулятора в рамках применяемого в работе подхода, необходимым этапом является линеаризация ­– процесс построения приближения для нелинейной системы линейными уравнениями в окрестности некоторого решения этой системы (контролируемого движения). Как правило, в качестве такого решения выступает точка равновесия – постоянное решение. Иными словами, нелинейные дифференциальные уравнения, представляющие собой математическую модель динамики объекта

где ­ – -мерный вектор состояния системы, – -мерный вектор управлений, заменяется на линейные дифференциальные уравнения при условии непрерывной дифференцируемости компонент вектор-функции по совокупности аргументов на множестве своего определения при допустимых значениях динамических переменных, в окрестности конкретной точки заменяются системой:

где .

Для рассматриваемой математической модели (2), задаваемой исходными полными нелинейными уравнениями, указанным способом для дальнейшего исследования могут быть получены линейные модели в окрестности представляющих интерес точек .

Записывая уравнения в стандартной форме, мы получим стационарную систему линейных дифференциальных уравнений (LTI-систему) следующего вида в окрестности точки :

Здесь .

Матрицы , показаны ниже

,

,

где

для

Для того чтобы получить конкретные постоянные матрицы и , выбираем некоторые конкретные точки равновесия – наборы из постоянных значений , обращающих (2) в тождество.

При этом для полученных LTI-систем с помощью соответствующих алгоритмов можем считать обратную связь, которая будет задаваться в следующем виде:

. (3)

Следовательно, удобно считать матрицу , которая обеспечивает асимптотически устойчивость соответствующей замкнутой линейной системы. В дальнейшем, полученные матрицы , будут получены и использованы в -модели с пользованием нечёткой логики.

## 1.2. Синтез закона управления с помощью LQR-оптимизации

Матрицы из (3) могут быть построены на основе применения различных математических методов.

При синтезе управления для обеспечения движения на траектории в данной работе будем опираться на популярный в исследовательской практике метод линейной интегральной квадратичной оптимизации (LQR-оптимизации). Описание такого подхода можно найти, например, в монографии [1].

Задача LQR-оптимизация формулируется для линейных систем с математическими моделями в виде LTI-систем (linear time-invariant)

(1.1)

где – вектор состояния, – вектор управлений, причем компоненты матриц  являются постоянными, определяющими свойствами объекта. Здесь считаем, что системы (1.1) полностью управляема.

Для объекта вида (1.1) управление строим в виде линейной обратной связи:

. (1.2)

На решениях замкнутой системы (1.1), (1.2) зададим интегральный квадратичный функционал:

(1.3)

который характеризует поведение замкнутой системы . Здесь неотрицательно определенная симметрическая матрица, а положительно определенная симметрическая матрица. Матрицы и являются весовыми матричными множителями.

Тогда задачу о поиске LQR-оптимального управления можно сформулировать следующим образом:

 (1.4)

где множество постоянных матриц таких, что корни характеристического полинома замкнутой системы расположены в открытой левой комплексной полуплоскости, то есть.

Для поиска матрицы в оптимальном управлении (1.2) необходимо выполнить следующие действия:

1. Сформировать матричное алгебраическое уравнение Риккати

его решение симметрическая положительно определенная матрица: полином является гурвицевым, который имеет все корни в левой полуплоскости.

2. Построить матрицу коэффициентовоптимального регулятора по вектору состояния.

Выполнение указанных действий приводит к стабилизирующему управлению, которое является оптимальным по отношению к функционалу (1.3), для заданных матриц и . Как правило, в исходной постановке задачи такие матрицы не заданы, соответственно, применение метода LQR-оптимизации предполагает выбор элементов этих матриц как дополнительный этап решения задачи. Выбирать указанные матрицы следует таким образом, чтобы они отражали требования к динамическим процессам в замкнутой системе. В общем случае, подобная процедура не является формализованной, и это позволяет ставить задачу о выборе матриц и таких, чтобы замкнутая система удовлетворяла заданным требованиям.

## **1.3. Автоматический выбор параметров в управлении**

Рассмотрим теперь другой оптимизационный подход, его описание можно найти в монографии [1], в котором требования к качеству динамических процессов задается с помощью учета ограничений, непосредственно накладываемых на контролируемые переменные с помощью задания области, которой они должны принадлежать. Если контролируемые переменные принадлежат заданным областям, это означает, что требования к динамики процессов выполнены. Для задания такой области необходимо задать две функции времени так, чтобы выполнялось.

Пусть в правую часть модели замкнутой системы входит ряд параметров, объединенных в общий вектор , от которых зависит ее поведение. Пусть – некоторая динамическая контролируемая переменная (представляющая для нас интерес), тогда можно считать допустимым динамический процесс, если вектор настраиваемых параметров выбран так, чтобы выполнялись неравенства:

(1.5)

Графически это проиллюстрировано на следующем рисунке 2.

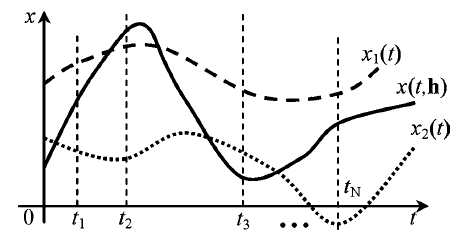


Рис.2 Допустимая область для характеристики динамического процесса [1]

Видно, что допустимая область, заданная парой [,], может ограничить динамический процесс. Для получения допустимого динамического процесса необходимо подобрать вектор так**,** чтобы моделируемый процесс по переменной лежал внутри области, то есть обеспечить выполнение ограничений (1.5).

Эффективным способом достижения такого результата является постановка и решения оптимизационной задачи, рассмотрим её. Зафиксируем некоторый набор моментов времени из промежутка, на котором выполняем имитационное моделирование . Для каждой точки зададим функцию, которая будет служить мерой выхода переменной за пределы допустимой области в момент (штрафом), с помощью формулы

(1.6)

Тогда общий штраф за выход из коридора на всем отрезке будет определяться суммой:

.

Исходя из желания минимизировать общий штраф, можно перейти к оптимизационной задаче

(1.7)

Такая задача решается численно, и это означает, что важной особенностью и преимуществом такого подхода является то, что при моделировании и получении значения минимизируемой функции может использоваться нелинейная модель замкнутой системы, содержащая любые необходимые для моделирования элементы.

Для вычислений в работе объединяются возможности обоих описанных оптимизационных подходов (в настоящем и предыдущем пунктах).

Для этого требования к динамическим процессам в замкнутой системе характеризуются с помощью задания допустимой области на плоскости, в которую должна попасть контролируемая динамическая переменная, а вектор параметров составляется из параметров в матрицах и в функционале (1.3). Для упрощения ненулевыми элементами указанных матриц будут считаться только диагональные. В этом случае эти элементы будут задавать весовые коэффициенты перед соответствующими переменными состояния и управления в функционале (1.3), определяя вклад каждой такой переменной в функционал. Диагональные значения матрицы  будем считать положительными фиксированными вещественными числами (матрица должны быть положительно определена). Варьируемые диагональные элементы матрицы (часть может быть фиксирована изначально) задаются как квадраты некоторых вещественных чисел (параметров), которые исходно не заданы. Все такие параметры объединяются в вектор в функционале (1.7), и выбираются путем решения оптимизационной задачи (1.7).

## 1.4. Построение нечёткой TS-модели

Нечеткая логика представляет собой многозначной логикой, которая позволяет определить промежуточные значения для таких общепринятых оценок, как да или нет*,* истинно или ложно*,* черное или белое и т.п. Подобные выражения, которые определяются с помощью нечеткой логики, будет представляться, например, следующим образом: чуть-чуть жарко или достаточно холодно. Такой способ описания может быть формализован математически.

Одним из основных понятий нечеткой логики является понятие нечеткого множества. Известно, что из классической математики задание четких множеств означает использование вполне определенных значений, а нечетких – неопределенных. Допустим, что множество A всех чисел от 0 до 5, а подмножество B множества A от 2 до 4. Для характеристики множества B может быть задана функция  (функция принадлежности), ставящая в соответствие число от 0 до 1 каждому элементу множества B, и характеризующая степень принадлежности конкретного элемента множества A нечеткому множеству B. Значение 1 означает, что элемент определенно принадлежит нечеткому множеству B, соответственно, 0 – определенно не принадлежит нечеткому множеству B. Функции принадлежности могут задаваться разными способами. Например, популярным вариантом являются треугольные функции принадлежности, которые задаются как:

Если задаем то получается следующий график:

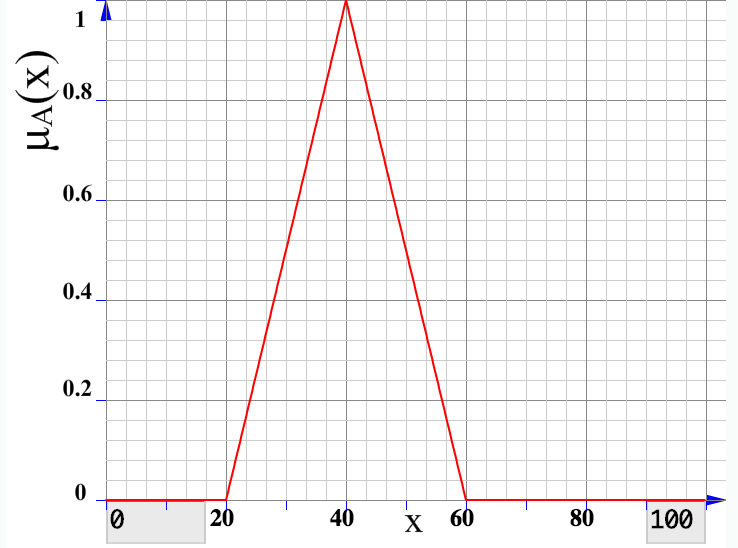


Рис.3. Пример функции принадлежности

Ещё пример, если жарко в комнате, тогда включим кондиционер. Но как определим, что в комнате жарко и когда включать кондиционер? Тогда получается с помощью нечеткой логики значения: если немножко жарко, тогда открываем кондиционер несильно, то есть степень принадлежности промежутку [0,1] может быть получена, например, 0.3. Если жарко, тогда открываем кондиционер побольше, что значит, что степень может быть уже 0.5. А если достаточно жарко, тогда открываем кондиционер сильно. Это значит уже, что уровень принадлежности максимален, то есть значение функции принадлежности 1. Общая процедура использования нечетких множеств предполагает наличие следующих этапов: сначала следует выполнить фаззификацию или переход к нечеткости, задание правил и функций принадлежности для конкретных переменных, и, наконец, дефаззификацию или устранение нечеткости чтобы получить конкретной определенное значение переменной.

Известно, что в нечёткой логике необходимо задавать несколько правил такого типа, как “IF…THEN…”. Это соответствует, например, методу Мамдани (Mamdani), то есть после THEN… как сдаваем значение 1 или 0 (true или false). Поставим как условие просто равенство состояний некоторым желаемым значениям.

А идея метода Takagi-Sugeno предполагает, что в правиле “IF…THEN…” в условии “…THEN…” задается явно какое-нибудь математическое выражение. Это означает, что этап дефаззификации в данном случае отсутствует.

Данный метод может быть применен для построения нечеткой TS-модели, которая заменяет собой исходную нелинейную модель.

Эти наборы “IF…THEN…” представляют собой замену нелинейной системы так, что каждое правило соответствовало замене нелинейной системы одной локальной линейной подсистеме, а вся нечёткая система, заменяющая исходную нелинейную систему, представлена как комбинация всех локальных подсистем:

: IF is and … and is , THEN

где – нечёткие множества, ( , ) – пара матриц соответствующих -тому набору , ­­– динамические переменные, задающие конкретную равновесную точку в правилах.

Теперь зададим функции принадлежности, которые в нечёткой логике представляет степень принадлежности рассматриваемой переменной к данному нечёткому множеству. Пусть обозначает, что переменной соответствует функция принадлежности в -м правиле.

Берём произведение, получаем:

представляет собой вес -ого правила.

Средневзвешенный метод широко используется в промышленном управлении, используем его и получим уравнения общей системы (TS-модель), которую будем рассматривать вместо исходной нелинейной системы (2) при синтезе управления:

Поскольку TS-модель (1.8) построена на основе нескольких линейных моделей, ее использование вместо исходной нелинейной модели при построении регулятора, означает, что можно попробовать свести эту задачу к применению стандартного математического аппарата для работы с отдельными линейными системами, которые задают модель (1.8).

## 1.5. Синтез нечеткого управления (Fuzzy Controller)

В предыдущем пункте была построена -модель для того, чтобы на ее основе построить управление нелинейной модели, гарантирующее асимптотически устойчивость -модели.

Правила, которые использованы для построения-модели для рассматриваемого объекта с исходной моделью (2), будут иметь вид

: IF is , THEN

Идея, на которой основывается синтез регулятора, обеспечивающего асимптотическую устойчивость замкнутой системы для динамической модели (1.8), задающей объект, состоит в том, что управляющее воздействие определяется правилами с теми же предпосылками, что и при задании -модели:

: if is , then .

Таким образом, схема нечеткого регулятора для нелинейной системы при управлении автомобилем на траектории, предполагает задание управляющего воздействия как

Основной результат, который используется в работе, может быть представлен следующей теоремой [3], [13]:

**Теорема**:

Нечеткая система (1.8), (1.9) асимптотически устойчива, если существует одна общая положительная определенная матрица , такая что

где

где

Таким образом, управление строится для нечеткой системы (1.8), а используется для исходной системы (2).

Достоинством такого управления является то, что основным этапом при его построении является синтез линейных обратных связей для локальных LTI-систем, и для такой задачи существуют известные методы, гарантирующие построение требуемого управления. При этом, в соответствии с теоремой имеет место глобальная асимптотическая устойчивость замкнутой системы, если выполнены условия, указанные в теореме. Для синтеза линейных обратных связей для локальных LTI-систем на основе рассматриваемого метода и выбора неизвестных матриц в разных ситуациях применялись различные подходы, которые предполагали, в том числе, адаптивную настройку параметров всех матриц одновременно [11], обеспечение учета каких-то дополнительных ограничений, как, например, в [15].

Однако в любых ситуациях применение такого подхода требует решение некоторых практических вопросов.

Во-первых, необходимо выбрать равновесные точки для линеаризации. Общей рекомендации по их выбору нет, однако от него зависит, насколько близка будет динамика TS-модели нелинейной модели, а это влияет на то, насколько хорошо построенное управление будет работать для исходного объекта.

Во-вторых, точки и матрицы обратной связи должны быть выбраны так, чтобы выполнялись условия теоремы, то есть система линейных матричных неравенств, заданная в теореме. Задача поиска матрицы для обеспечения выполнения неравенств может быть решена с помощью ППП Robust Control Toolbox системы MATLAB, содержащего инструментальные средства для работы с линейными матричными неравенствами.

При этом, стремление получить более точное соответствии TS-модели исходной модели приводит к использованию большего количества точек, а чем больше количество использованных равновесных точек, тем больше количество линейных матричных неравенств, которые должны выполняться в соответствии с теоремой, то есть тем большим оказывается общий объем вычислений и тем сложнее может быть поиск матрицы **.**

Кроме того, построение нечеткого TS-регулятора так же, как и построение регулятора иными способами, как правило, должно проводиться с учетом дополнительных требований и ограничений, которые накладываются на динамику замкнутой системы.

Таким образом, можно предложить следующий подход для синтеза TS-регулятора для обеспечения стабилизации нелинейной системы, в данной работе – для обеспечения движения автомобиля на заданном пути:

для построения управлений таких, чтобы в замкнутой системе обеспечивалось требуемое динамическое качество, и выполнялись указанные неравенства, матрицы в управлении для локальных подсистем могут быть выбраны на основе решения задачи параметрической оптимизации с подбором варьируемых параметров. В частности, для этого разумно формировать минимизируемую функцию с помощью задания «динамического» коридора, с введением варьируемых параметров в функционал метода LQR- оптимизации так, как это было описано в п.1.3. После выбора матриц локальных обратных связей необходимо в соответствии с теоремой осуществить поиск матрицы , обеспечивающей выполнение неравенств из теоремы. Если оказалось, что такой матрицы не существует, необходимо изменить равновесные точки, их количество, минимизирумые функции в задаче параметрической оптимизации, в результате заново выбрать матрицы линейной обратной связи в управлении и опять выполнить проверку, действуя так до тех пор, пока матрица не будет найдена. При этом, в отдельных ситуациях может оказаться успешным следующая последовательность действий: сначала выбирается малое количество равновесных точек, для них ищется матрица , далее набор используемых точек и, соответственно, матриц локальных обратных связей, расширяется для улучшения управляющего воздействия так, чтобы та же матрица обеспечивала выполнение и новой группы неравенств **.** Следует отметить, что выбор способа задания функций принадлежности также может оказать влияние на итоговый результат.

# Глава 2. Компьютерное и имитационное моделирование

## 2.1. Выборка точки равновесия

Для отслеживания траектории (пути), мы выбираем некоторые точки, которые удовлетворяют следующим уравнениям:

Например, выберем три операционные точки:

## 2.2. Simulink-модель непрерывной системы

Для исследования динамики объекта построим компьютерную модель, которая формируется на основе известной математической модели (2). В работе компьютерная модель автомобиля построена в среде MATLAB-Simulink, в которой реализуется компьютерное и имитационное моделирование рассматриваемых процессов. На рисунке 4 представлена схема, соответствующая построенной Simulink-модели.

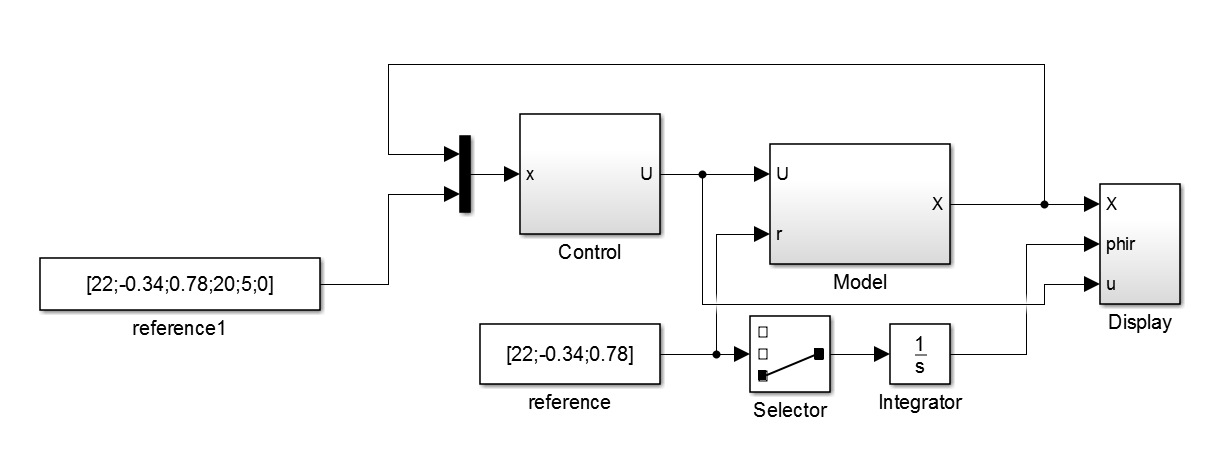


Рис.4. Simulink-модель динамики автомобиля

Рассмотрим назначение подсистем и использованных в модели блоков. Здесь подсистемой «Model» (приведена на рис.5) является сама модель автомобиля, которая соответствует уравнениям (2), подсистема «Control» – подсистема, отвечающая за выработку управляющего сигнала, которое будет в виде.

В блоках «reference» и «reference1» задаются значения желаемой траектории (пути).

Блок «Selector» выбирает нужные параметры, выходящей из блока «reference».

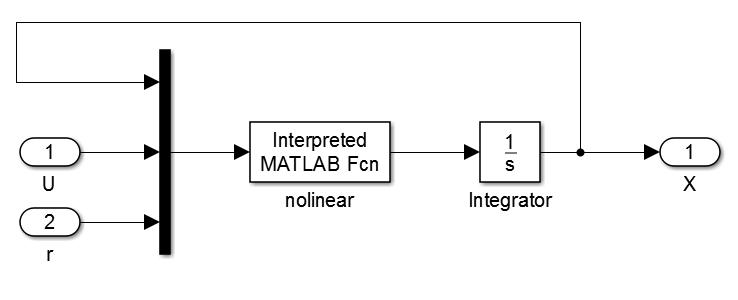


Рис.5. Подсистема «Model»

В подсистеме «Model» на вход поступают сигналы: вектор состояния, управление и желаемые значения . Блок «Interpreted MATLAB Function» обеспечивает применение заданной m-функции (функции, написанной на языке MATLAB) к входному сигналу и выдачу результата на выходе блока. В данной подсистеме блок «nolinear» используется для задания функции, которая описывает правую часть системы (**Листинг 1.**) нелинейных уравнений движения автомобиля. Блок «Integrator» интегрирует входной сигнал, чтобы получить вектор состояния.

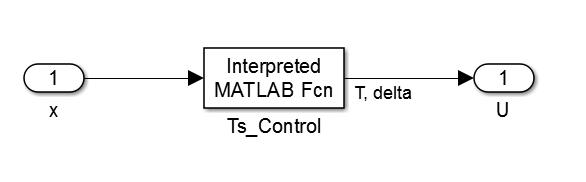


Рис.6. Подсистема «Control»

В подсистеме «Control» (рис.6) есть два входных сигнала: внутри – вектор и внешне – желаемое значение переменных состояния из блока «reference1». С помощью блока функции «Ts\_Control» обеспечивается выработка управляющего сигнала управления динамикой движения автомобиля с учетом изменения скорости.

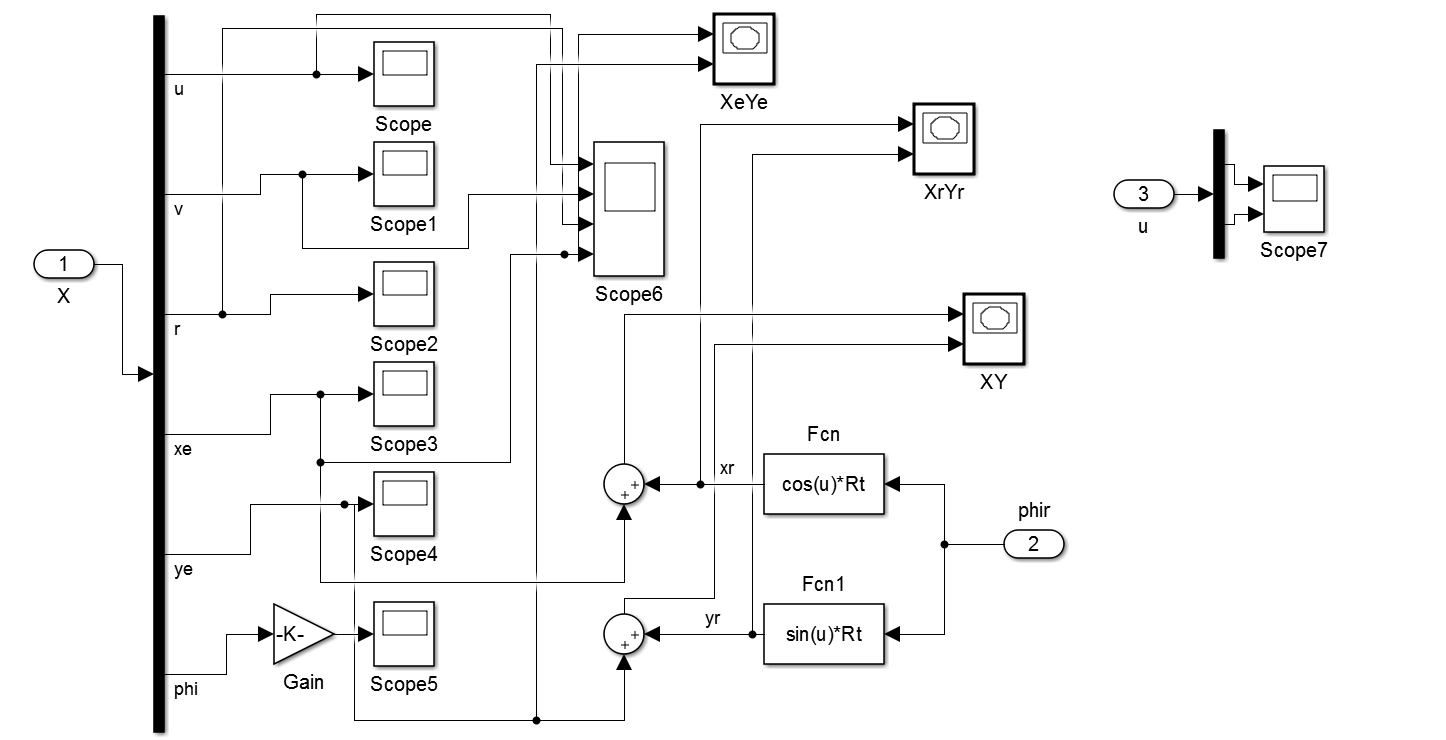


Рис.7. Подсистема «Display»

Подсистема «Display» содержит блоки, обеспечивающие визуализацию результатов численного моделирования. Основную роль играет блок «Scope», который можем увидеть визуальный результат моделирования, и позволяет анализировать данные результаты.

## 2.3. Автоматическая настройка коэффициентов управления

В данном разделе описан процесс реализации подхода, описанного в разделе 1.2, в среде MATLAB-Simulink.

Решение задачи LQR-оптимизация в MATLAB может основываться на применении специальной функции:

где– строка с искомыми коэффициентами управления,– решение матричного уравнения Риккати, используемое при построении закона управления,– вектор с корнями характеристического полинома замкнутой системы. Для использования метода LQR-оптимального синтеза необходимо задать матрицы и в качестве входных аргументов функции, а также матрицы самого LTI-объекта.

Задаем.

Для выбора элементов матрицыможем использовать подход параметрической оптимизации, в рамках которого будем изменять настраиваемые параметры на основе (1.7).Составим матрицу соответствующий вектор варьируемых параметров

Теперь необходимо организовать связь между конкретным набором параметров, используемых при решении LQR-оптимизации, и функционал в (1.7):

Схема компьютерной модели, предназначенной для выполнения параметрической оптимизации, представлена на рисунке 7.

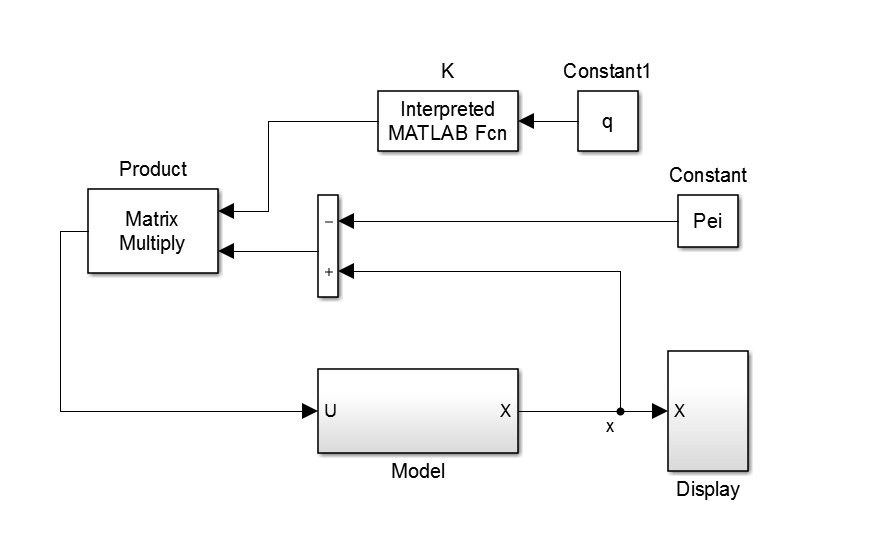


Рис.8. Компьютерная модель для проведения оптимизации

Здесь в блоке «Constant1» задаются варьируемые параметры матрицы:

,

в блоке «Interpreted MATLAB Function» ставится в соответствие конкретному набору параметров вычисленная матрица в управлении с помощью LQR, входом которого является блок «Constant1». В блоке «Matrix Multiply» считается умножение матрицы , чтобы получить управление .

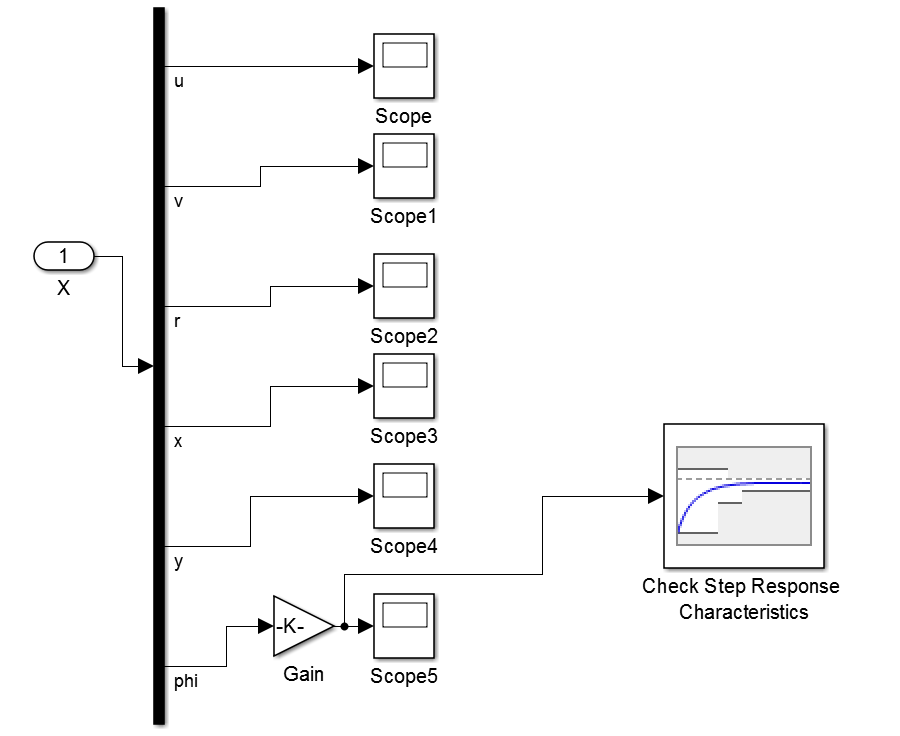
**

Рис.9. Блок в подсистеме «Display»

Идея с заданием динамической области, в которой должна находиться некоторая динамическая переменная, при условии, что эта переменная представляет собой сигнал, который формируется в Simulink-модели, реализована посредством специального инструмента Simulink Design Optimization. Функция формируется с помощью средства Simulink Design Optimization, позволяющего задать коридор такой, чтобы кривая функции не выходила за переделы заданной области. Коридор формируется вручную путем перетаскивания отрезков, из которых он собирается, либо по точкам, которые задаются в специальном поле. Вычисление точек, в которых накладывается штраф, и вычисление итогового значения (1.7) выполняется автоматически без участия пользователя.

В подсистеме «Display» подсоединим блок «Check Step Response Characteristics» к линии, которая соответствует интересующему нас сигналу -курсу, в нем нажимаем «Response Optimization». Появилось новое окно, в котором выбираем «Design Variables Set», затем нажимаем «New» для добавления новых параметров. Выберем из списка переменную «», которую мы поставили в блок «Constant1», тем самым определив параметры, которые будут изменяться в процессе численной оптимизации. Далее выбираем «Plot Current Response», показывающий нам переходный процесс по курсу.

Коридор зададим, исходя из пожелания не иметь перерегулирования по курсу и уменьшить время регулирования

На рис.10, показан пример результатов оптимизационной процедуры. В окне, в которое выводятся результаты, видно, что на плоскости есть пять черных линий, передвигая которые мы можем сами регулировать высоту верхней и нижней границы, разделяющие на три части. Первая часть дает задать желаемое время для перехода кривой от начального условия к желаемой цели. Вторая часть задает цель обеспечить стремление кривой к желаемому значению. Третья часть фактически служит для контроля асимптотической устойчивости. Обычно задаем верхнюю и нижнюю границу желаемой цель, то есть. В нашем случае, поставим верхнюю и нижнюю границу, получим график, который показан на рис.9.

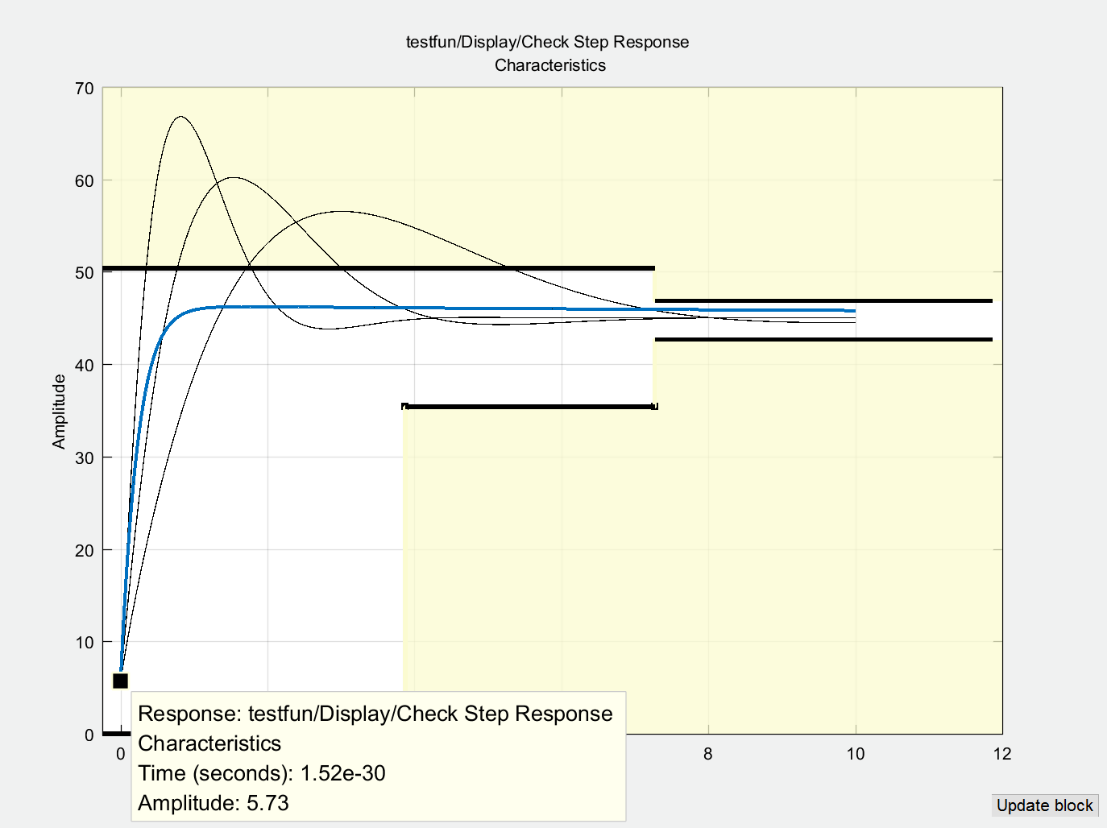


Рис.10. Результат параметрической оптимизации

После выполнения всех настроек запускаем процесс численной оптимизации, предварительно можно выбрать алгоритм, который будет использоваться, а также задать параметры численной минимизации. Если в этом имеется потребность.

После завершения оптимизации в средстве Simulink Design Optimization получили аргументы матрицы:

## 2.4. Задание траектории на плоскости

В данной пункте опишем желаемую траекторию (путь), для которой в работе представлены численные результаты. Способ задания траектории автомобиля на плоскости:

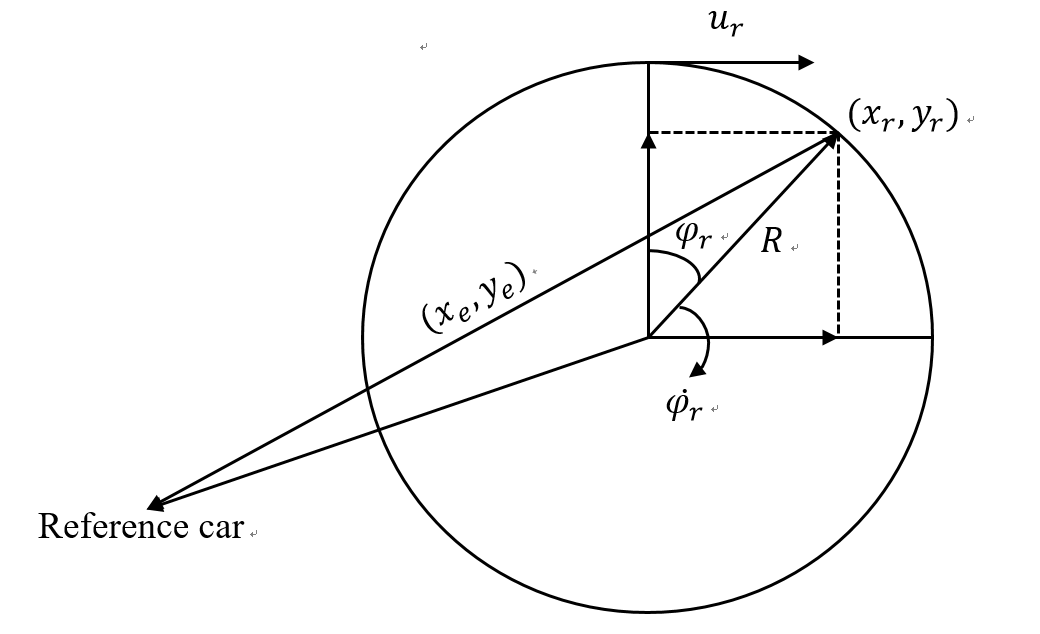


Рис.11. Задание траектории

Пусть автомобиль двигается по кругу, соответственно, скорость будет продольной скорости и радиус обозначается , тогда угловая скорость , и по закону треугольника помещения желаемого автомобиля будет в следующем виде:

где .

Соответственно, по закону вектора следует получить перемещение реального автомобиля:

где – отклонение между желаемым и реальным путями автомобиля.

Реализация задания траектории в Simulink-модели приведено на рисунке 6. С помощью блока «Fcn» задаем перемещение , соответственно, блок «Fcn1» задаем перемещение . Считаем сумму перемещение желаемого и отклонения автомобиля с использованием блока «Sum». Получим результат перемещение, то есть траектории реального автомобиля выведено в блоке «XY».

## 2.5. TS-нечеткая модель

Для того чтобы построить TS-модель с помощью нечеткой логики, необходимо задавать функции принадлежности для всех переменных.

В качестве примера функции принадлежности (**Листинг 2.**) для одной из переменных представлен на графике ниже, функции принадлежности для остальных переменных могут быть выбраны аналогично.

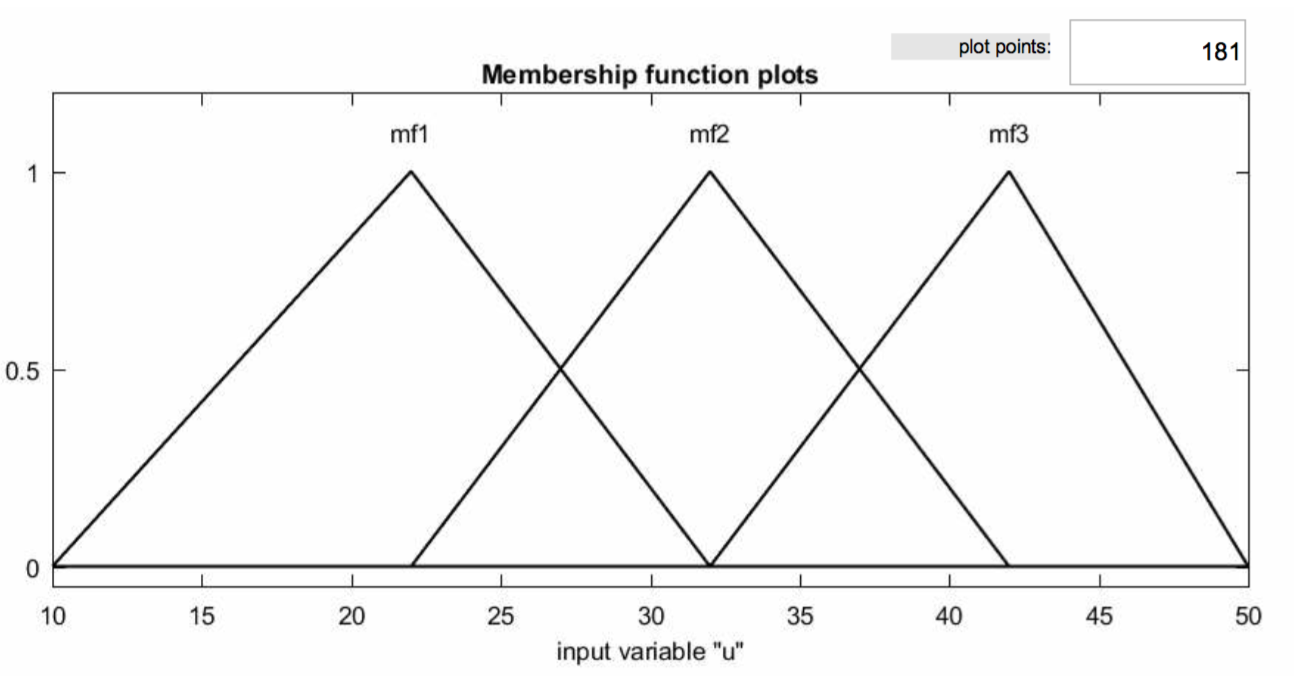


Рис.12. Функция принадлежности

Для простого исследования, сначала создаем только три правила:

Соответственно, управление для исходной нелинейной модели получается три правила:

где – наши желаемые значения переменных, задающие траекторию.

Построенная Simulink-модель проведена на Рис.3., рассмотрим некоторые испытания.

Пусть задаем несколько начальных состояний и желаемых траекторий для проверки нечеткого управления.

,

и в результате имеем графический результат:

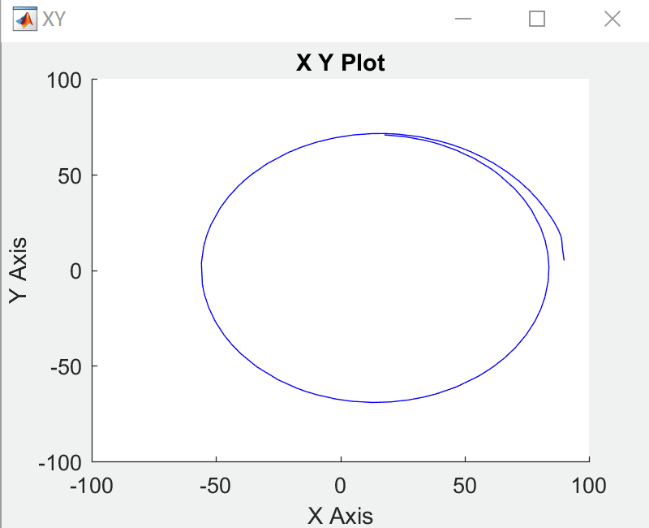
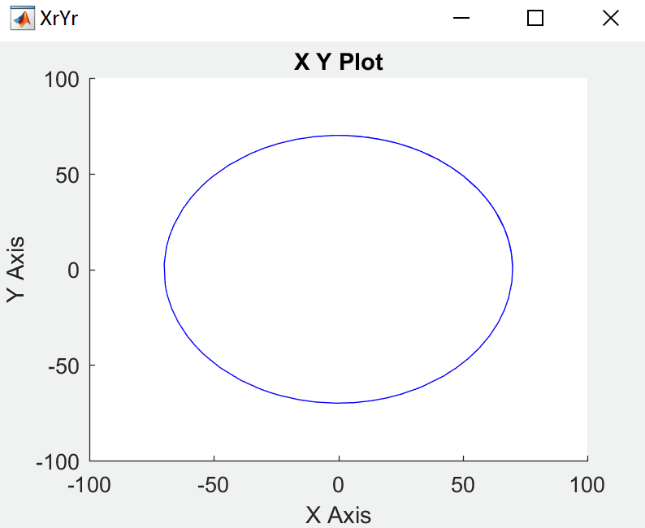


Рис.13. Желаемая траектория Рис.14. Реальная путь

Вышеприведенный результат показал, что автомобиль хорошо следует желаемой траектории.

А если поменяем условия, то будет:

,

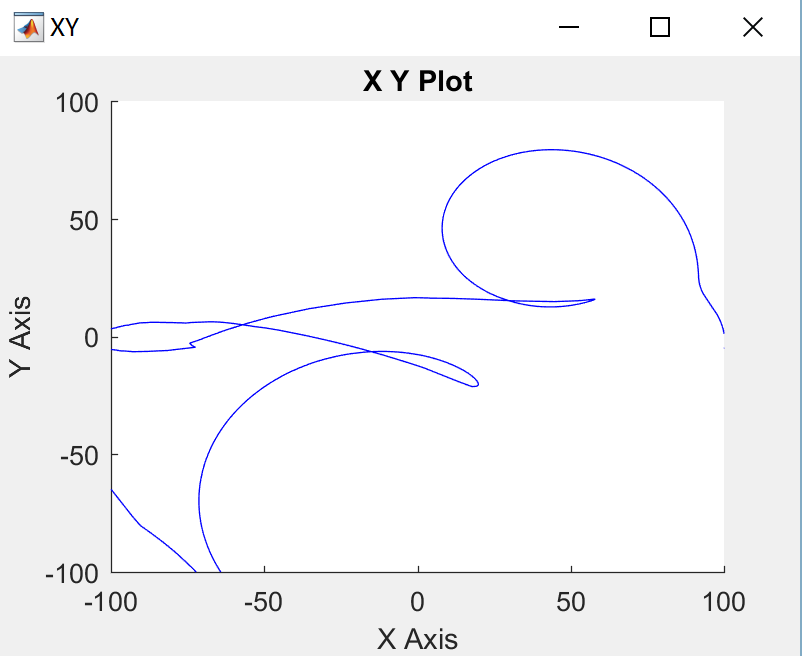
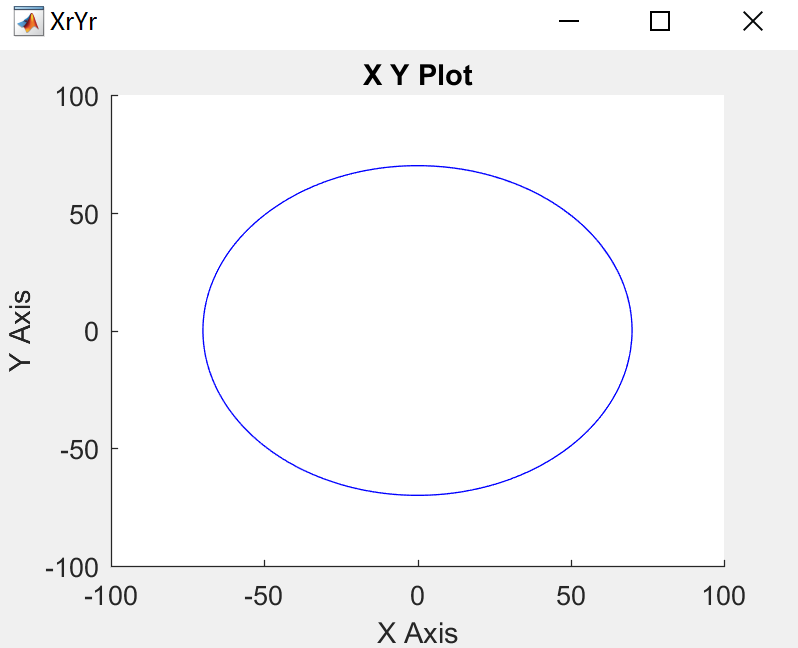


Рис.15. Желаемая траектория Рис.16. Реальная путь

Если поставим оба отклонения перемещения []=[0,0], тогда графический результат тоже может стать идеальным:

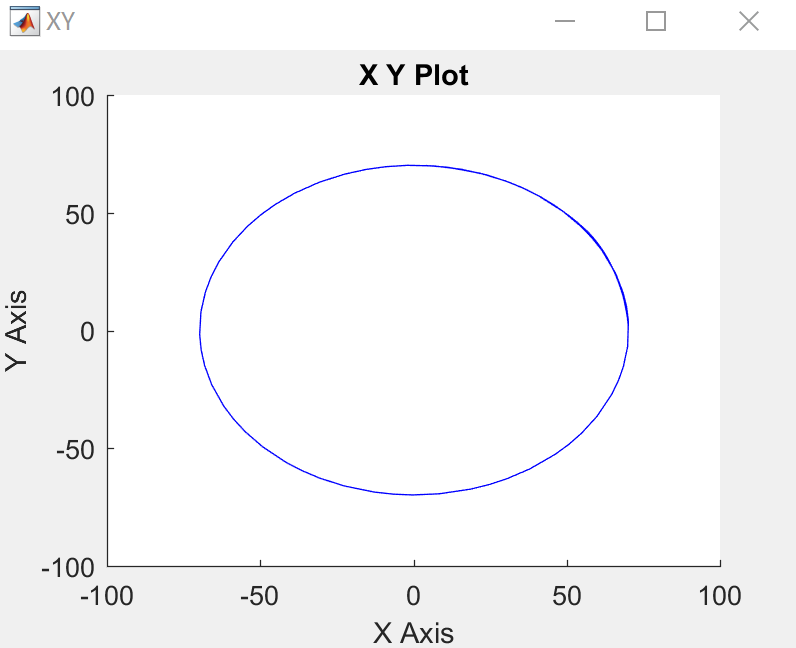
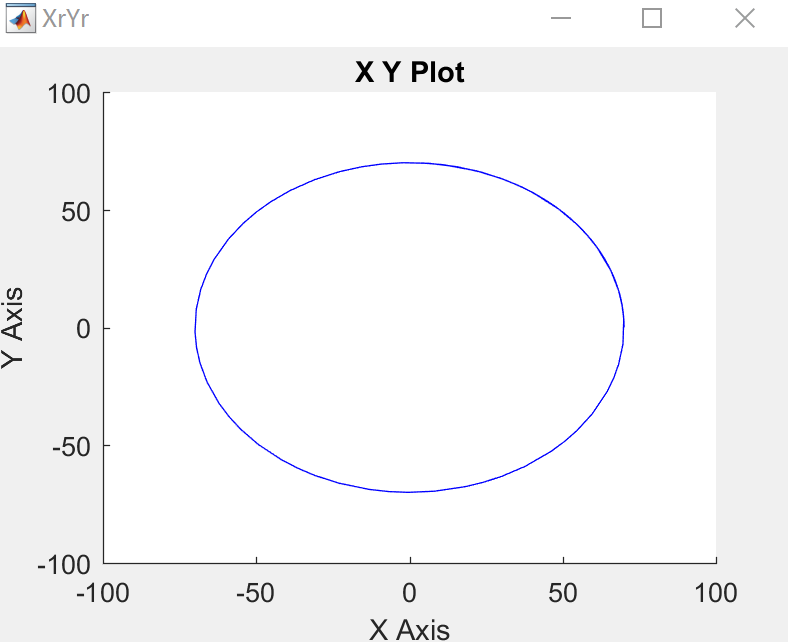


Рис.17. Желаемая траектория Рис.18. Реальная путь

Проблема возникла в недостаточности управления, так что автомобиль не может следовать траекторию.

## 2.6. Улучшение свойств TS-регулятора

Из-за того, что регулятор, который построен для нелинейной системы, обладал не очень хорошими свойствами, для улучшения теперь добавляем ещё немного правил. В нашем случае, больше правил – значит больше точек.

Добавляем ещё две точки

,

.

Тогда, наконец, у нас пять правил получается. Иллюстрирующий программный код приведен в приложении (**Листинг 3.**):

Проверяем те же условия, которые в предыдущем варианте не сработали в нелинейной системе:, . Результат показан внизу:

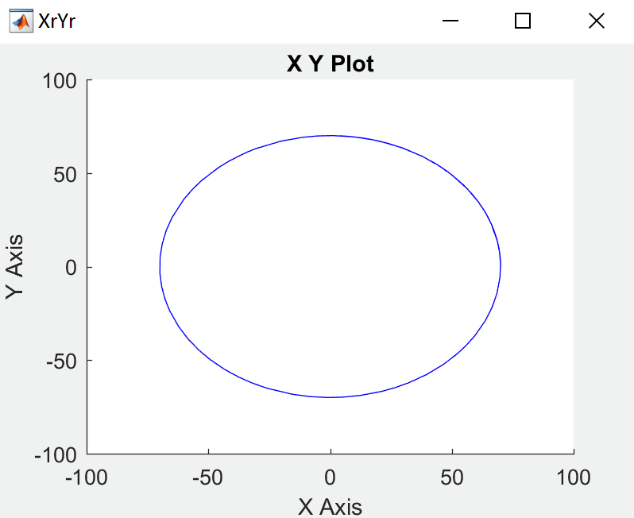
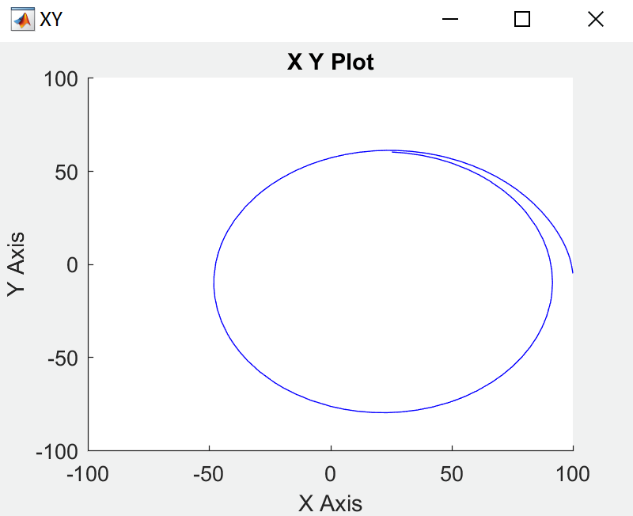
****

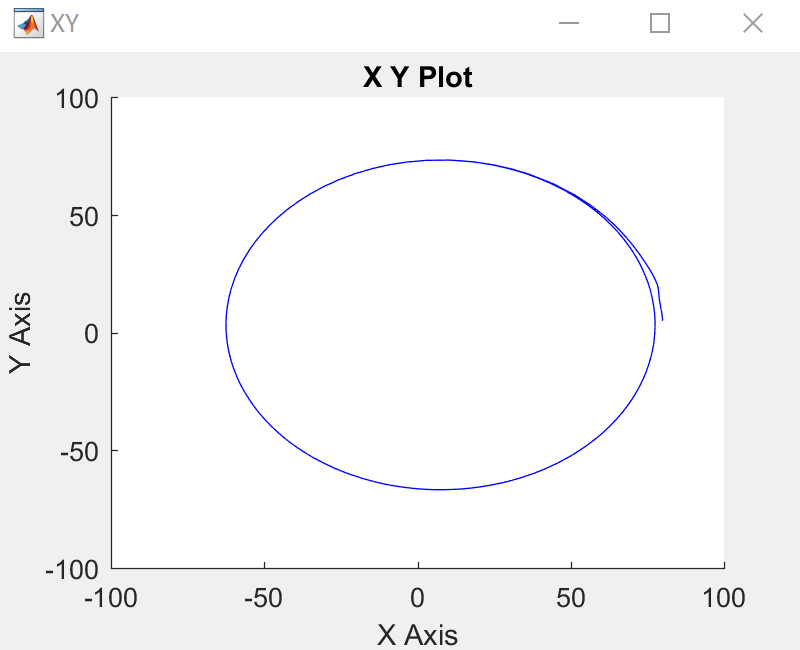
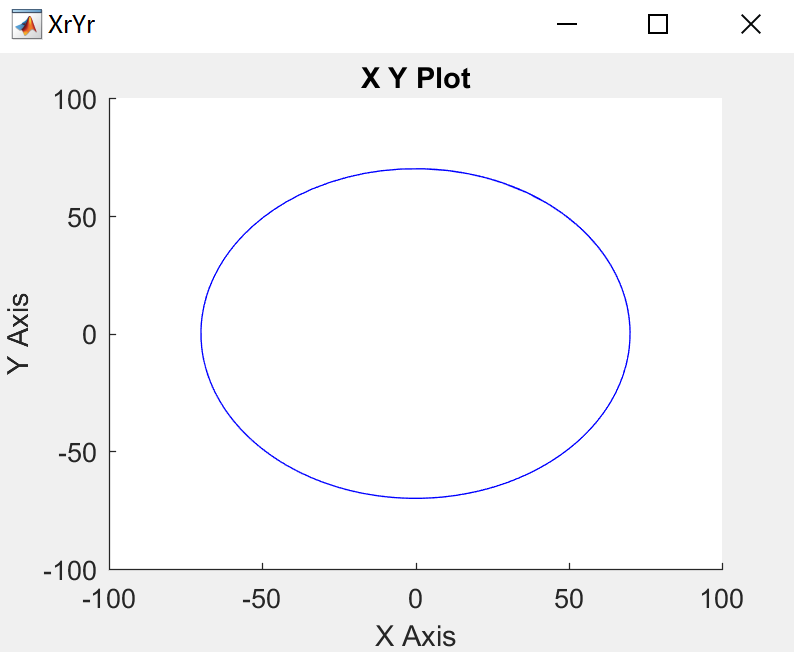
Рис.19. Желаемая траектория Рис.20. Реальная путь

Результат улучшается, значит, наш подход увеличения количества правил работает.

Теперь тогда рассмотрим другие условия и проверяем, например,

,

:

Рис.21. Желаемая траектория Рис.22. Реальная путь

Пусть начальная скорость ещё немножко увеличивается, разница между скоростями будет , и посмотрим графический результат:

,

,

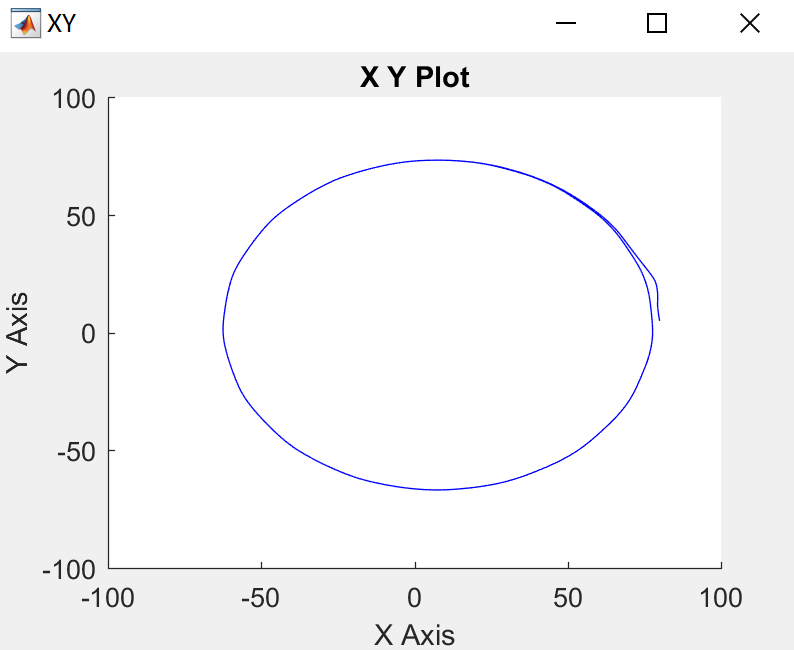
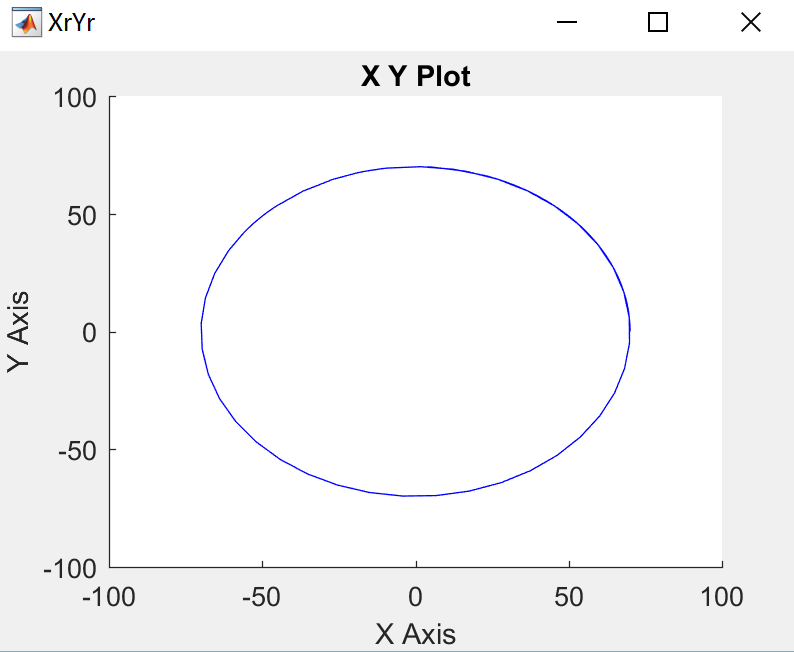


Рис.23. Желаемая траектория Рис.24. Реальная путь

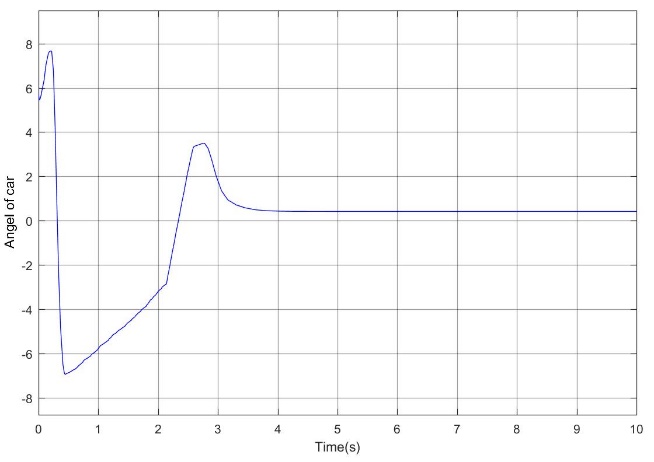
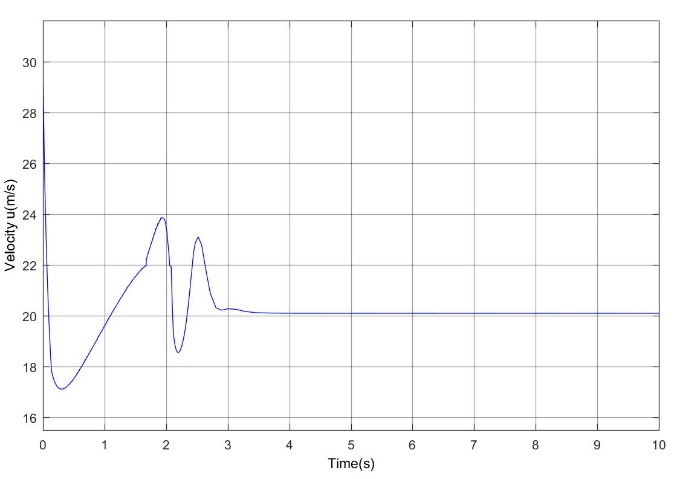


Рис.25. Скорость u Рис.26. Угол курса

Если ещё увеличим отклонение между ними, используем скорости такие же, как предыдущие, а разница перемещения побольше будет:

,

,

Результаты показаны внизу:

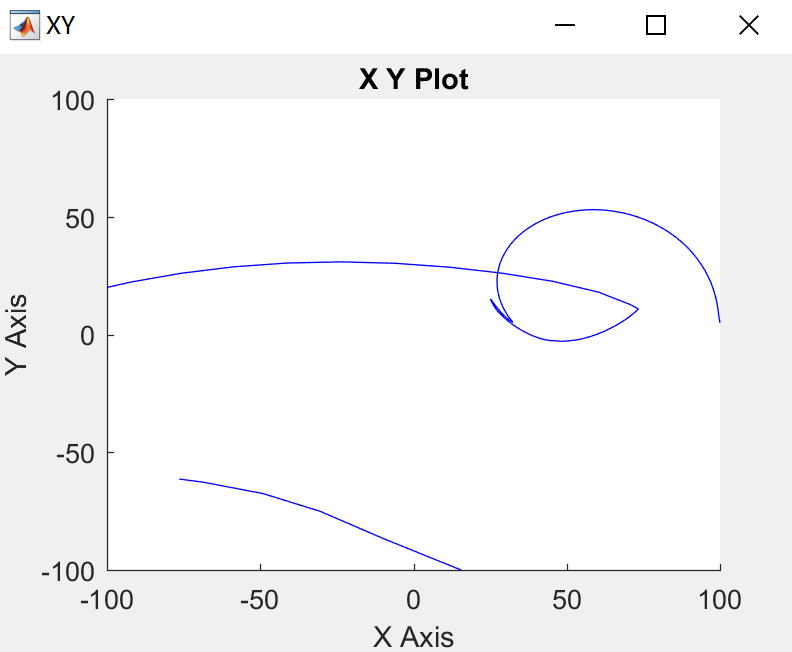
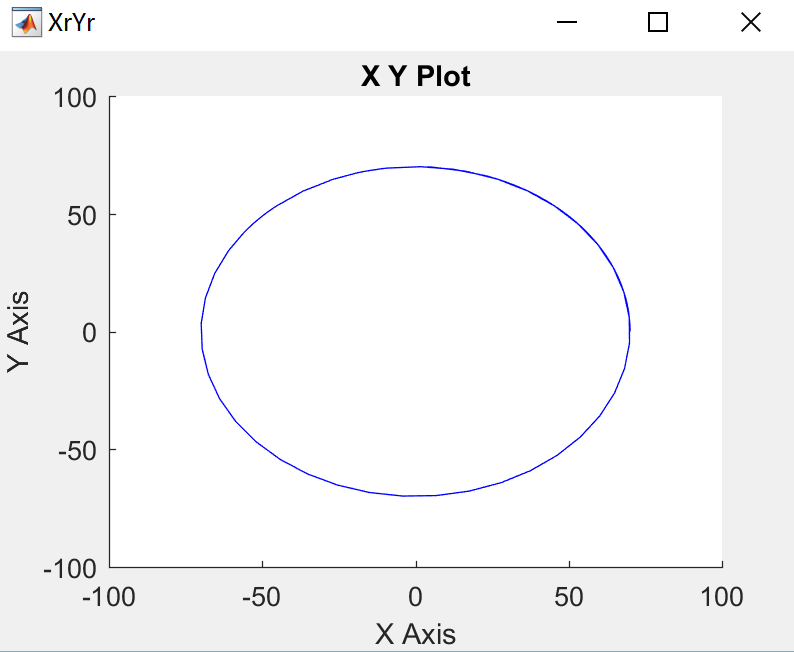


Рис.27. Желаемая траектория Рис.28. Реальная путь

Проблема в том, что если разница между желаемым и реальным перемещениями будет большой, тогда отслеживание не будет получаться. Для большего улучшения свойств регулятора, проведем следующий эксперимент.

## 2.7. Эксперимент для улучшения свойств управления

У нас были три операционные точки, так что каждая переменная может описываться в трех вариантах. Перебираем все эти значения, получая точки, соответствующие всем возможным наборам так, что каждая переменная принимает одно из трех значений. Тогда можем получить больше правил , то есть больше точек и больше матриц .

Таким образом, количество правил составляет 27, которые проведены внизу:

Теперь будем проверять новые законы для построения нечеткого регулятора. Поставим такие же условия, в которых не сработал предыдущий регулятор:

,

*.*

Даже можем составить побольше разницу перемещений для желаемого и реального автомобилей .

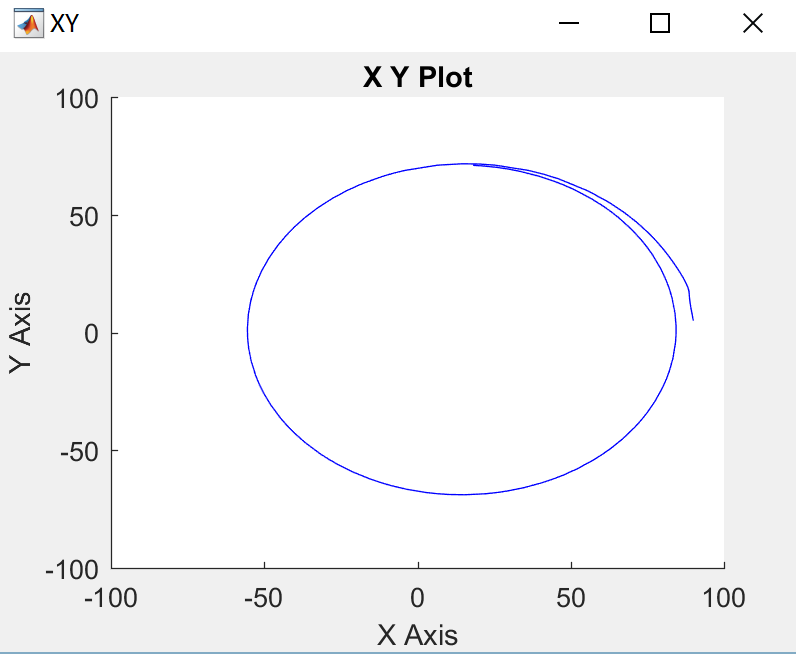
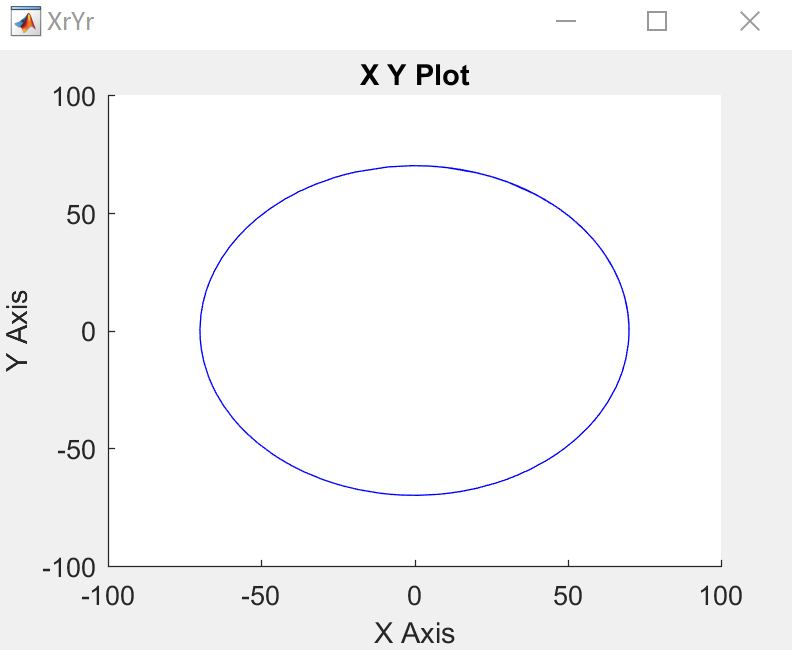


Рис.29. Желаемая траектория Рис.30. Реальная путь

Графический результат показывает, что улучшилось поведение системы с полученным TS-регулятором.

Теперь покажем некоторые результаты для данного закона с 27 правилами,

,

.

Для нового регулятора результат имеет следующий вид:

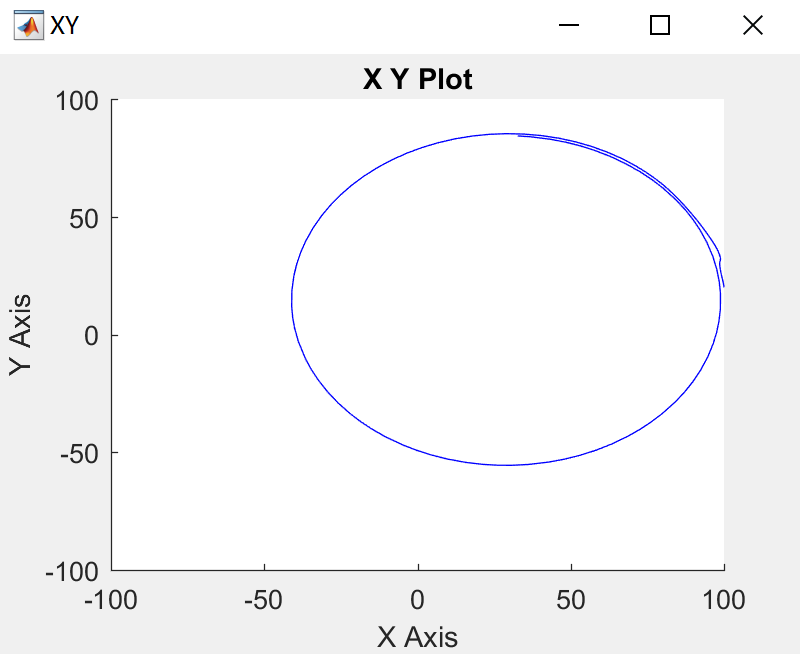
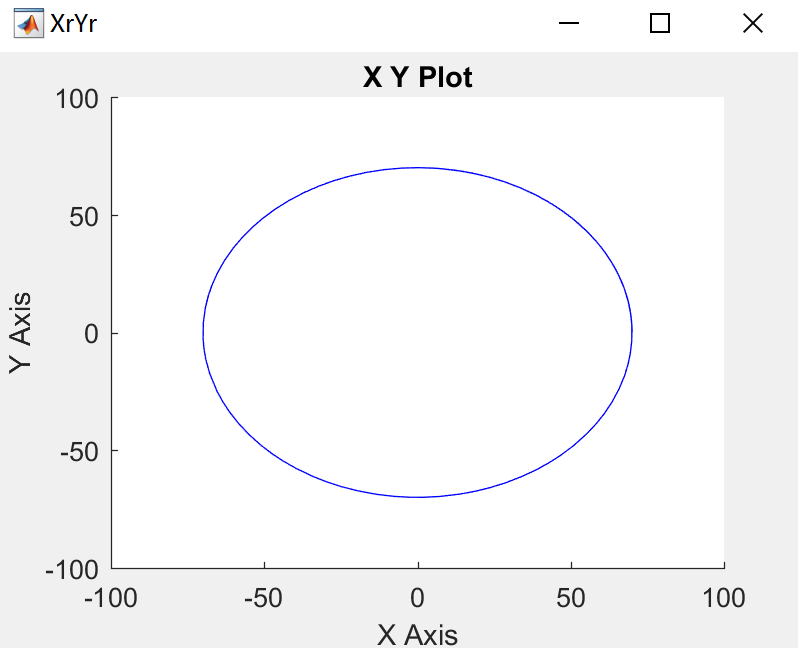


Рис.31. Желаемая траектория Рис.32. Реальная путь

Для значений

, для нового закона регулятора результат имеет следующий вид:

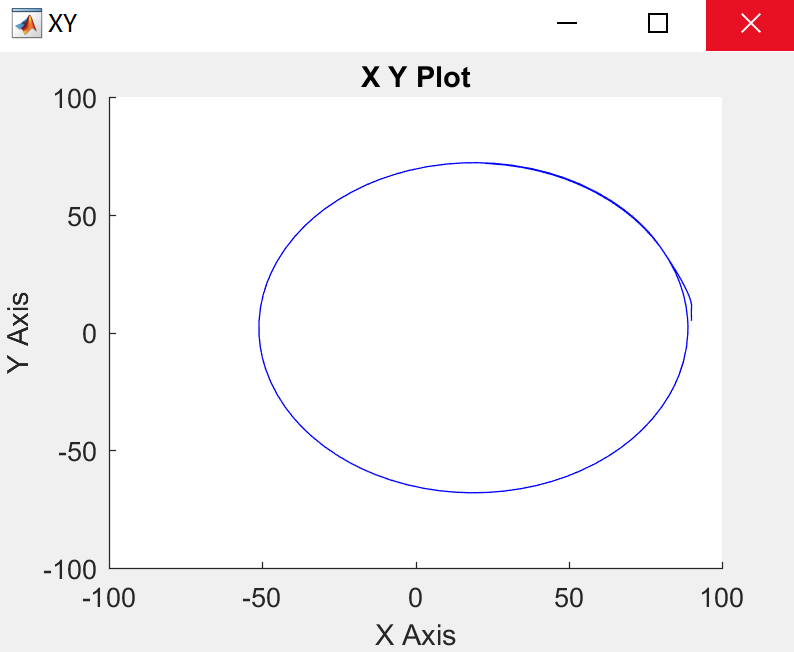
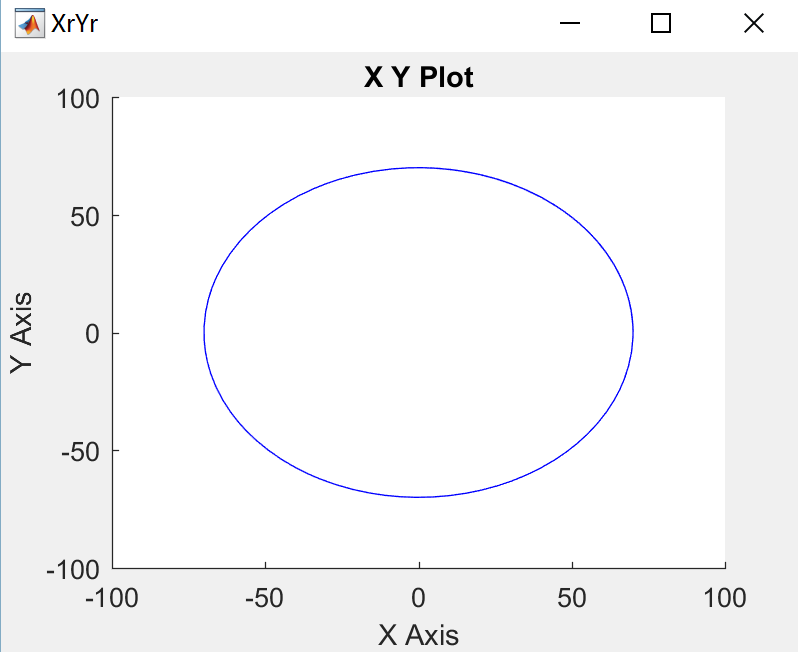


Рис.33. Желаемая траектория Рис.34. Реальная путь

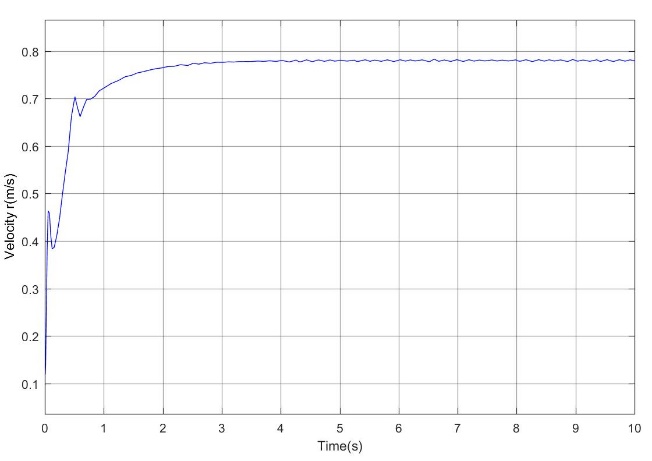
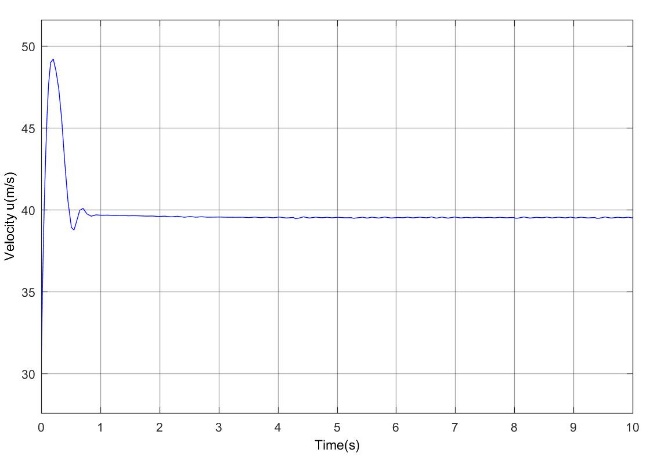


Рис.35. Скорость Рис.36. Скорость

## 2.8. Проверка асимптотической устойчивости

В данном параграфе будем рассматривать вопрос асимптотической устойчивости замкнутой TS-модели, заменяющей исходную нелинейную систему (2), с использованием **теоремы**,котораяпроведена в параграфе [1.5]:

Напомним, что в соответствии указанной теоремой нечеткая система асимптотически устойчива, если существует одна общая положительная определенная матрица P, такая что

где

На данном этапе требуется решить систему линейных матричных неравенств. Решение такой системы может быть найдено с помощью ППП Robust Control Toolbox системы MATLAB. Функция позволяет определить переменные в системе линейных неравенств. В нашем случае матрица задается таким образом: , она будет с размером . Функция позволяет создавать вонкретное неравенство. В нашем случае, например, представляет собой неравенство . Программный код, где находится такое решение, приведен в приложении (**Листинг 4.**).

Для первого варианта с тремя правилами нашлась такая матрица , чтобы обеспечивала всю систему линейных матричных неравенств:

Так что, получается, вся TS-модель система асимптотически устойчива при нечетком регуляторе с тремя правилам.

Теперь проверяем асимптотическую устойчивость TS-модели с пятью правилами. При помощи метода LMI функций в MATLAB также находится одна общая положительная определенная матрица P, такая что:

С помощью подхода LMI находится решение системы линейных матричных неравенств, и матрица P описается в следующем виде:

где,

,,,

,,,

,,,

,,,

,,,

,,,

,,,

,,,

,,,

,,,

,,,

,,.

Эта же матрица P соответствует решению и для исходных трех правил. Соответственно, можем считать, что найдется матрица P, которая должна будет обеспечить асимптотически устойчивость TS-модели с 27 правилами.

# Вывод

В данной работе рассмотрена задача управления движением автомобиля на желаемой траектории, для решения использована идея синтеза нечеткого регулятора на основе нечеткой TS-модели в соответствии с (1), (2). Такая идея может быть достаточно успешно реализована, при этом для синтеза локальных управлений может применяться LQR-подход, который позволит изменять параметры и влиять на результаты управления.

Соответственно, простой вариант регулятора был построен и найдены недостатки. Далее из простого варианта сложились правила, так что удалось улучшить результат на траектории. Так же на эксперименте тоже получилось улучшить результаты для базы правил, без формальной проверки глобальной асимптотической устойчивости.

Таким образом, указанный подход в отдельных ситуациях может быть успешно применен, однако улучшение поведения объекта за счет увеличения количества используемых локальных подсистем влечет существенное увеличение объема вычислительной работы. Гарантия применимости рассматриваемого регулятора к любым допустимым условиям, также отсутствует.

# Заключение

1. Рассмотрены вопросы моделирования и управления движением автомобиля на заданной траектории (пути).

2. Сформирован моделирующий комплекс для проведения компьютерного и имитационного моделирования в среде MATLAB-Simulink в задаче управления движением автомобиля на заданном пути.

3. Организован процесс выполнения имитационного моделирования для автоматической настройки коэффициентов в законе управления с учетом заданных ограничений и требований к качеству динамических процессов с использованием специального инструмента Simulink Design Optimization.

4. В среде MATLAB реализован подход и проведены вычисления для построения нечеткого управления с использованием ППП Robust Toolbox.

5. Проведен анализ результатов на траектории с различными наборами правил, в том числе состоящим из трех и пяти правил, а также проведен эксперимент с 27 правилами.

# Список литературы

1. Веремей Е.И.Линейные системы с обратной связью. СПб.: Издательство «Лань», 2013. 448 с.
2. Bentalba S, El Hajjaji A, Rachid A. Fuzzy parking and point stabilization: application car dynamics model[C]//5th IEEE Mediterranean Conference on Control and Systems. 1997.
3. El Hajjaji A, Bentalba S. Fuzzy path tracking control for automatic steering of vehicles[J]. Robotics and Autonomous Systems, 2003, 43(4): 203-213.
4. Гуо КонХуэй.Динамика управления автомобилем. Издательство «Научные технологии Цзилинь»: 1991. 700 с. На кит.яз.
5. Ван ЧжэнЛин.Процесс управления и использования Simulink. Издательство «Электронная промышленность»: 2006. 278 с. На кит.яз.
6. Liu Bao, Tang WanSheng. Современная теория управления. Издательство «Машиностроение»: 1983. 166 с. На кит.яз.
7. Qing Hua. Matlab и система нечеткого управления. На кит.яз.
8. T-S нечеткая модель.

http://wenku.baidu.com/link?url=AxjATkZdZg4NOER0\_7dWz18OdacwtEWFcr5kZgmrBexxmJzS9M5D\_fqlBsFIpBNllqx9pKrIWB3biIto\_4zOHngyUrkSySeEpAA0-b9wljO

1. Задание LMI-системы с использованием lmiedit.

http://wenku.baidu.com/link?url=AxjATkZdZg4NOER0\_7dWz18OdacwtEWFcr5kZgmrBexxmJzS9M5D\_fqlBsFIpBNllqx9pKrIWB3biIto\_4zOHngyUrkSySeEpAA0-b9wljO

1. Решение линейных матричных неравенств в среде Matlab. http://wenku.baidu.com/link?url=r1QMgyWl4c79ib4ce3Xj\_rJ1up10SBAG8dhULntphvVz5lVK8\_ed\_CBQtPYqJQl3b4hpEZOSBS4NHeJmRzM8WueWsGJNiekW054J5\_kVOJa
2. Kratmüller M. The Adaptive Control of Nonlinear Systems Using the TSK Fuzzy Logic[J]. Acta Polytechnica Hungarica, 2009, 6(2): 5-16.
3. Kashyapa Narenathreyas, Fuzzy Logic Control for Aircraft Longitudinal Motion.
4. Chadli M, Maquin D, Ragot J. Observer-based controller for Takagi-Sugeno models[C]//Systems, Man and Cybernetics, 2002 IEEE International Conference on. IEEE, 2002, 5: 6 pp. vol. 5.
5. Abdelkrim A, Ghorbel C, Benrejeb M, et al. Lmi-based tracking control for takagi-sugeno fuzzy model[J]. International Journal of Control & Automation, 2010, 3(2).
6. Xingquan G, Shuang G. LMI-based H2 Control for TS Fuzzy System with Hard Constraints[J]. International Journal of Control and Automation, 2015, 8(3): 21-30.

# Приложение

**Листинг 1.**

function Xn = nonlinearcar(U)

u = U(1);

v = U(2);

r = U(3);

xe = U(4);

ye = U(5);

phie = U(6);

T = U(7);

delta = U(8);

ur = U(9);

vr = U(10);

dphir = U(11);

a = 1.05;

b = 1.63;

h = 0.53;

M = 1480;

f = 0.02;

Iz = 2350;

g = 8.81;

Cf = 135000;

Cr = 95000;

k1 = 0.005;

k2 = 0.41;

du = v\*r-f\*g+u^2\*(f\*k1-k2)/M+Cf\*delta\*(v/u+a\*r/u)/M+T/M;

dv = -u\*r-(Cf+Cr)\*v/(M\*u)+(b\*Cr-a\*Cf)\*r/(M\*u)+Cf\*delta/M+T\*delta/M;

dr = -f\*M\*h\*u\*r/Iz+(b\*Cr-a\*Cf)\*v/(Iz\*u)-(b^2\*Cr+a^2\*Cf)\*r/(Iz\*u)+a\*Cf\*delta/Iz+a\*T\*delta/Iz;

dphi = r;

dxe = ur\*cos(phie)+vr\*sin(phie)-u+ye\*dphi;

dye = -ur\*sin(phie)+vr\*cos(phie)-v-xe\*dphi;

dphie = dphi - dphir;

Xn = [du;dv;dr;dxe;dye;dphie];

end

**Листинг 2.**

function mff = mff11( x )

% by uei

% mf11

if x >= 10 && x <= 22

mff(1) = ((x - 10)/(22 - 10))\*(1 - 0);

elseif x >= 22 && x <= 32

mff(1) = ((x - 22)/(30 - 22))\*(0 - 1) + 1;

elseif x < 10

mff(1) = 1;

else mff(1) = 0;

end

% mf12

if x >= 22 && x <= 32

mff(2) = ((x - 22)/(32 - 22))\*(1 - 0);

elseif x >= 32 && x <= 42

mff(2) = ((x - 32)/(42 - 32))\*(0 - 1) + 1;

else mff(2) = 0;

end

% mf13

if x >= 32 && x <= 42

mff(3) = ((x - 32)/(42 - 32))\*(1 - 0);

elseif x >= 42 && x <= 50

mff(3) = ((x - 42)/(50 - 42))\*(50 - 42) + 1;

else mff(3) = 0;

end

end

function mff = mff21( x )

% by vei

% mf21

if x <= 0 && x >= -3.2

% y = k\*x + b

mff(1) = 1/3.2\*x + 1;

elseif x > 0

mff(1) = 1;

else mff(1) = 0;

end

% mf22

if x <= 0 && x >= -3.2

mff(2) = -1/3.2\*x;

elseif x >= -8 && x <= -3.2

% y = ((x - x0) / (x1 - x0)) \* (y1 - y0) + y0

mff(2) = ((x + 3.2)/(-8 + 3.2))\*(0 - 1) + 1;

else mff(2) = 0;

end

% mf23

if x >= -8 && x <= -3.2

mff(3) = ((x + 3.2)/(-8 + 3.2))\*(1 - 0);

elseif x >= -10 && x <= -8

mff(3) = ((x + 8)/(-10 + 8))\*(0 - 1) + 1;

else mff(3) = 0;

end

end

function mff = mff31( x )

% by rei

% mf31

if x >= 0 && x <= 0.8

mff(1) = -1/0.8\*x + 1;

elseif x < 0

mff(1) = 1;

else mff(1) = 0;

end

% mf32

if x >= 0 && x <= 0.8

mff(2) = 1/0.8\*x;

elseif x >= 0.8 && x <= 1

mff(2) = ((x - 0.8)/(1 - 0.8))\*(0 - 1) + 1;

else mff(2) = 0;

end

% mf33

if x >= 0.8 && x <= 1

mff(3) = ((x - 0.8)/(1 - 0.8))\*(1 - 0);

% elseif x >= 1 && x <= 1.2

% mff(3) = ((x - 1)/(1.2 - 1))\*(0 - 1) + 1;

elseif x > 1

mff(3) = 1;

else mff(3) = 0;

end

end

**Листинг 3.**

function [Y, W] = r5Rule\_Model( inp , w )

x = inp(1:6);

xr = inp(7:12);

Y = 0; W = 0;

K1 = [2.8466\*10^4, -9.5333\*10^3, 5.9163\*10^4, -2.4935\*10^5, 3.2292\*10^5, 1.0677\*10^5;

-3.0888\*10^-3, 0.13649, 0.087483, -0.087662, -0.88671, 6.8637];

K2 = [3.6311\*10^4, 3.0077\*10^4, 5.8368\*10^5, -3.1659\*10^5, 2.9973\*10^2, 3.2963\*10^6 ;

-1.9610\*10^-2, 1.3031\*10^-1, -2.0189\*10^-1, 4.4963\*10^-2, -8.6899\*10^-1, 5.3208];

K3 = [3.1642\*10^4, -2.2318\*10^5, 2.4846\*10^5, -1.9236\*10^5, 1.5170\*10^6, -8.6682\*10^6;

-1.3971\*10^-2, 2.8546\*10^-1, -1.5647\*10^-2, -1.6701\*10^-1, -1.8193, 1.2974\*10];

K4 = [3.6311\*10^4, -9.5333\*10^3, 5.9163\*10^4, -2.4935\*10^5, 3.2292\*10^5, 1.0677\*10^5;

-1.9610\*10^-2, 0.13649, 0.087483, -0.087662, -0.88671, 6.8637];

K5 = [3.1642\*10^4, -9.5333\*10^3, 5.9163\*10^4, -2.4935\*10^5, 3.2292\*10^5, 1.0677\*10^5;

-1.3971\*10^-2, 0.13649, 0.087483, -0.087662, -0.88671, 6.8637];

% Rule 1

y = K1 \* (x - xr) \* w(1) \* w(4) \* w(7);

Y = Y + y;

W = W + w(1) \* w(4) \* w(7);

% Rule 2

y = K2 \* (x - xr) \* w(2) \* w(5) \* w(8);

Y = Y + y;

W = W + w(2) \* w(5) \* w(8);

% Rule 3

y = K3 \* (x - xr) \* w(3) \* w(6) \* w(9);

Y = Y + y;

W = W + w(3) \* w(6) \* w(9);

% Rule 4

y = K4 \* (x - xr) \* w(2) \* w(4) \* w(7);

Y = Y + y;

W = W + w(2) \* w(4) \* w(7);

% Rule 5

y = K5 \* (x - xr) \* w(3) \* w(4) \* w(7);

Y = Y + y;

W = W + w(3) \* w(4) \* w(7);

end

function [ u\_ts ] = r5Ts\_Control(x)

global w;

%count weight

w=zeros(1,9);

w(1:3)=mff11(x(1));

w(4:6)=mff21(x(2));

w(7:9)=mff31(x(3));

[Y, W] = r5Rule\_Model(x,w);

if W == 0

u\_ts = [0;0];

else u\_ts = - Y./W;

end

end

**Листинг 4.**

%seting LMI

setlmis([]);

P=lmivar(1,[6 1]);

% N

lmiterm([1 1 1 P],N1',1,'s'); % LMI #1: N1'\*P+P\*N1

lmiterm([2 1 1 P],N2',1,'s'); % LMI #2: N2'\*P+P\*N2

lmiterm([3 1 1 P],N3',1,'s'); % LMI #3: N3'\*P+P\*N3

lmiterm([4 1 1 P],N4',1,'s'); % LMI #4: N4'\*P+P\*N4

lmiterm([5 1 1 P],N5',1,'s'); % LMI #5: N5'\*P+P\*N5

% G

lmiterm([6 1 1 P],G11',1,'s'); % LMI #6: G11'\*P+P\*G11

lmiterm([7 1 1 P],G12',1,'s'); % LMI #7: G12'\*P+P\*G12

lmiterm([8 1 1 P],G13',1,'s'); % LMI #8: G13'\*P+P\*G13

lmiterm([8 1 1 P],G14',1,'s'); % LMI #8: G14'\*P+P\*G14

lmiterm([9 1 1 P],G15',1,'s'); % LMI #9: G15'\*P+P\*G15

lmiterm([10 1 1 P],G22',1,'s'); % LMI #10: G22'\*P+P\*G22

lmiterm([11 1 1 P],G23',1,'s'); % LMI #11: G23'\*P+P\*G23

lmiterm([12 1 1 P],G24',1,'s'); % LMI #12: G24'\*P+P\*G24

lmiterm([13 1 1 P],G25',1,'s'); % LMI #13: G25'\*P+P\*G25

lmiterm([14 1 1 P],G33',1,'s'); % LMI #14: G33'\*P+P\*G33

lmiterm([15 1 1 P],G34',1,'s'); % LMI #15: G34'\*P+P\*G34

lmiterm([16 1 1 P],G35',1,'s'); % LMI #16: G35'\*P+P\*G35

lmiterm([17 1 1 P],G44',1,'s'); % LMI #17: G44'\*P+P\*G44

lmiterm([18 1 1 P],G45',1,'s'); % LMI #18: G45'\*P+P\*G45

lmiterm([19 1 1 P],G55',1,'s'); % LMI #19: G55'\*P+P\*G55

lmiterm([-20 1 1 P],1,1); % LMI #20: P > 0

lmisys=getlmis;

[tmin,xfeas]=feasp(lmisys); % решить lmi

P=dec2mat(lmisys,xfeas,P); % count P

eig(P),