

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И
ИНФОРМАТИКИ**

Кафедра дискретной математики и алгоритмики

ПАВЛОВЕЦ Мария Евгеньевна

**НЕЙРОСЕТЕВЫЕ РЕГУЛЯТОРЫ В СИСТЕМАХ
УПРАВЛЕНИЯ С ПРОГНОЗИРУЮЩЕЙ МОДЕЛЬЮ**

Отчет по производственной практике по специальности:

1-31 81 09 «Алгоритмы и системы обработки больших объемов информации»

Руководитель практики от кафедры
Наталия Михайловна Дмитрук
канд. физ.-мат. наук, доцент

Руководитель практики от организации
Ярослав Игоревич Кириллов
проектный менеджер

Минск 2019

ОГЛАВЛЕНИЕ

С.

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ	3
ВВЕДЕНИЕ	4
ГЛАВА 1 ОСНОВНЫЕ ПОНЯТИЯ И ОБЗОР ЛИТЕРАТУРЫ	5
1.1 Основные принципы МРС	5
1.2 МРС для решения задач стабилизации	6
1.3 Базовый алгоритм МРС	8
1.4 Терминальные элементы и устойчивость замкнутой системы	11
1.5 Выводы	15
ГЛАВА 2 Использование методов машинного обучения в системах управления с прогнозирующей моделью	16
2.1 Основные понятия нейронных сетей	16
2.2 Методы обучения с подкреплением	20
2.3 Методы дизайна оффлайн регуляторов в системах управления с прогнозирующей моделью	22
ГЛАВА 3 Построение нейросетевого регулятора	32
3.1 Описание иллюстративного примера	32
3.2 Подход к решению	33
3.3 Базовая реализация нейросетевого регулятора	36
3.4 Построение области притяжения с помощью SVM	40
3.5 Сравнение методов сэмплирования точек для построения аппроксимирования обратной связи	42
3.6 Сравнение результатов производительности	44
3.7 Использование метода для задачи стабилизации обратного маятника	45
ЗАКЛЮЧЕНИЕ	48
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	49

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

ОУ — оптимальное управление

MPC — Model Predictive Control, управление с прогнозирующей моделью

SVM — Support Vector Machines, метод опорных векторов

$\mathbb{I}_{\geq a}$ — множество целых чисел больше либо равных $a \in \mathbb{R}$

$\mathbb{I}_{[0, N-1]} = \{0, 1, \dots, N-1\}$

$\|x\|$ — евклидова норма вектора $x \in \mathbb{R}^n$

$\|x\|_Q^2 = x^T Q x$ — взвешенная норма вектора $x \in \mathbb{R}^n$, $Q > 0$

$\|x\|_A = \inf \|x - a\|$ — расстояние от точки x до множества A

\mathcal{K}_∞ — класс функций, где функция $\alpha : [0, \delta) \rightarrow [0, \infty)$ - непрерывная, строго возрастающая, $\alpha(0) = 0$ и $\lim_{\delta \rightarrow \infty} \rightarrow \infty$

I — единичная матрица

$\lambda_{\max}(A)$, $\lambda_{\min}(A)$ — максимальное и минимальное собственное значение матрицы A

ВВЕДЕНИЕ

Метод управления с прогнозирующей моделью (МРС) - одна из популярных современных технологий теории управления, основанная на решении в реальном времени задач оптимального управления с конечным горизонтом, аппроксимирующих решение задачи с бесконечным временным промежутком (например, задачи оптимальной стабилизации). Эта модель представляет собой семейство контроллеров, которое позволяет явно использовать модель для получения управляющего сигнала.

Основными причинами популярности МРС при решении прикладных задач являются применимость схемы управления к нелинейным системам и возможность учета ограничений на управления и траектории, а также способность работать без экспертного вмешательства в течение длительного времени. С другой стороны, эти же факторы могут стать причиной нереализуемости алгоритма, например, в случае быстрых процессов, для которых решение нелинейной задачи оптимального управления не может быть получено регулятором за короткий период квантования. Один из подходов к решению указанной проблемы подразумевает перенос некоторых вычислений оффлайн. В частности, в настоящей работе для реализации функции МРС-регулятора предлагается применить искусственные нейронные сети. Будут исследованы вопросы устойчивости и робастности замкнутой системы, проведено сравнение с оптимальным МРС-регулятором для ряда прикладных задач.

В соответствии с целью диссертации определена структура работы: обзор метода управления с прогнозирующей моделью, его устойчивость и базовый алгоритм, обзор методов обработки больших данных, используемые технологии для выполнения практической части, основные результаты и оценка производительности системы. В частности, изучены основные принципы МРС, рассмотрена задача стабилизации, описан базовый алгоритм МРС. А также исследованы вопросы устойчивости замкнутой системы вместе с алгоритмами построения терминального множества и терминальной функции для обеспечения устойчивости этой системы. Будут изучаться методы построения нейросетевого регулятора и областей притяжения для различных нелинейных систем. Будет проводиться сравнение производительности и точности при использовании различных техник сэмплирования точек для аппроксимации обратной связи.

ГЛАВА 1

ОСНОВНЫЕ ПОНЯТИЯ И ОБЗОР ЛИТЕРАТУРЫ

Управление по прогнозирующей модели — Model Predictive Control (МРС) [1, 2] — современный подход к управлению линейными и нелинейными динамическими системами, основанный на решении в реальном времени последовательности задач оптимального управления (ОУ) с конечным временным горизонтом. Упомянутые задачи ОУ называются прогнозирующими, формулируются в зависимости от целей управления, учитывают текущие измерения состояний объекта управления и ограничения на траектории и управляющие воздействия, а также аппроксимируют исходную задачу управления на бесконечном полуинтервале времени.

В настоящей главе излагаются основные принципы и базовый алгоритм МРС на примере задачи стабилизации управляемых движений динамической системы.

1.1 Основные принципы МРС

МРС базируется на следующих основных принципах [2]:

- для предсказания и оптимизации будущего поведения системы используется математическая модель управляемого процесса в пространстве состояний (в отличие от описания в виде передаточной функции и других методов частотной области);
- для выбранной математической модели формулируется прогнозирующая задача ОУ (predictive optimal control problem), которая будет решаться в каждый момент времени; в этой задаче:
 - конечный промежуток управления;
 - начальное состояние математической модели совпадает с измеренным текущим состоянием физического объекта управления;
 - критерий качества отражает цели управления: если целью является стабилизация объекта управления, то критерием качества выступает отклонение траектории объекта от положения равновесия;
 - учтены ограничения на траекторию и управляющие воздействия;

- оптимальное управление прогнозирующей задачи ОУ (предсказанное управляющее воздействие) применяется к объекту в текущий момент времени и до тех пор пока не будет измерено следующее состояние объекта; затем оптимизация повторяется.

Поскольку в каждый момент времени в задаче ОУ учитывается текущее состояние, результирующее управление представляет собой обратную связь.

Популярность МРС в теоретических исследованиях [1, 2] и на практике [3] обусловлена следующими свойствами, которыми не обладают другие методы теории управления:

- критерий качества в прогнозирующей задаче ОУ позволяет учитывать экономические требования к процессу управления (например, минимизацию энергетических затрат);
- учитываются жесткие ограничения на фазовые и управляющие переменные;
- метод применим к нелинейным и многосвязным системам.

Отметим, что поскольку решение задачи ОУ повторяется для каждого текущего момента времени, промежуток, для которого прогнозируется поведение системы, постоянно смещается ("скользит"), в силу чего МРС также иногда называется управлением со скользящим горизонтом — Receding Horizon Control (RHC).

1.2 МРС для решения задач стабилизации

Основными и исторически первыми приложениями МРС являются задачи стабилизации и регулирования. Остановимся подробно на результатах, полученных в теории МРС для задачи стабилизации. В этом разделе также вводятся основные обозначения, понятия, определения, базовый алгоритм МРС, свойства МРС-регулятора.

Как было отмечено выше, основная идея МРС состоит в том, чтобы использовать математическую модель процесса в пространстве состояний для предсказания и оптимизации поведения динамической системы в будущем [2]. Далее считаем, что используемая для предсказаний модель точно описывает процесс управления: на объект не действует возмущения и нет неучтенных различий между моделью и физическим объектом. Такие схемы МРС носят название номинальных (nominal MPC scheme).

Система, которая исследуется в данном разделе, является нелинейной, дискретной, стационарной:

$$x(t+1) = f(x(t), u(t)), \quad x(0) = x_0. \quad (1.1)$$

Здесь $x(t) \in X \subseteq \mathbb{R}^n$ — состояние системы в момент времени t , $u(t) \in U \subseteq \mathbb{R}^r$ — управляющее воздействие в момент t , $t \in \mathbb{I}_{\geq 0}$ — время, дискретное. Относительно функции $f : \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^n$ предполагается, что она непрерывна.

Начальное состояние системы (1.1) задано:

$$x(0) = x_0 \in X.$$

На состояния и управляющие воздействия u накладываются ограничения вида

$$(x(t), u(t)) \in Z \subseteq X \times U, \quad t \in \mathbb{I}_{\geq 0}, \quad (1.2)$$

которые называются смешанными ограничениями. Понятно, что такая форма задания ограничений включают в себя одновременно и фазовые ограничения на состояния системы, и прямые ограничения на управляющие воздействия. Относительно множества Z предполагается, что оно компактно.

Цель (стабилизирующего) МРС — построить обратную связь $u(x)$, при которой замкнутая система

$$x(t+1) = g(x(t)) = f(x(t), u(x(t))), \quad x(0) = x_0, \quad (1.3)$$

будет устойчива в некотором заданном положении равновесия (заданном множестве), при этом переходный процесс не нарушает ограничения (1.2) при всех $t \in \mathbb{I}_{\geq 0}$.

Определение 1.1 Точка x^* является положением равновесия для системы (1.3), если выполняется $x^* = g(x^*)$.

Определение 1.2 Множество $X \subseteq \mathbb{R}^n$ называется положительно инвариантным множеством для системы (1.3), если выполняется $g(x) \in X \forall x \in X$.

Определение 1.3 Пусть $X \subseteq \mathbb{R}^n$ — положительно инвариантное множество для системы (1.3). Замкнутое, положительно инвариантное множество $A \subseteq X$ устойчиво для (1.3), если $\forall \epsilon > 0, \exists \delta > 0$, что для всех $\|x_0\|_A \leq \delta, x_0 \in X$, выполняется $\|x(t)\|_A \leq \epsilon, \forall t \in \mathbb{I}_{\geq 0}$.

Определение 1.4 Множество $A \subseteq X$, удовлетворяющее условиям определения 1.3, асимптотически устойчиво с областью притяжения X , если оно устойчиво и $\lim_{t \rightarrow +\infty} \|x(t)\|_A = 0 \forall x_0 \in X$.

Определение 1.5 Множество — глобально асимптотически устойчиво, если оно асимптотически устойчиво с $X = \mathbb{R}^n$.

При $A = \{x^*\}$ получим классические понятия устойчивости, асимптотической устойчивости и глобальной асимптотической устойчивости решения $x(t) = x^*$ по Ляпунову.

Далее будет рассматриваться случай стабилизации системы управления (1.1) для заданного положения равновесия, т.е. случай $A = \{x^*\} \in X$.

1.3 Базовый алгоритм MPC

Как было отмечено в разделе 1.1, идея алгоритма MPC состоит в том, чтобы в каждый момент $t \in \mathbb{I}_{\geq 0}$ оптимизировать будущее поведение системы (1.1) на конечном горизонте $N \geq 2$ и использовать первое значение полученного оптимального (программного) управления в качестве значения обратной связи для момента t (см. рис. 1.1). Под "оптимизацией будущего поведения" понимается решение прогнозирующей задачи ОУ.

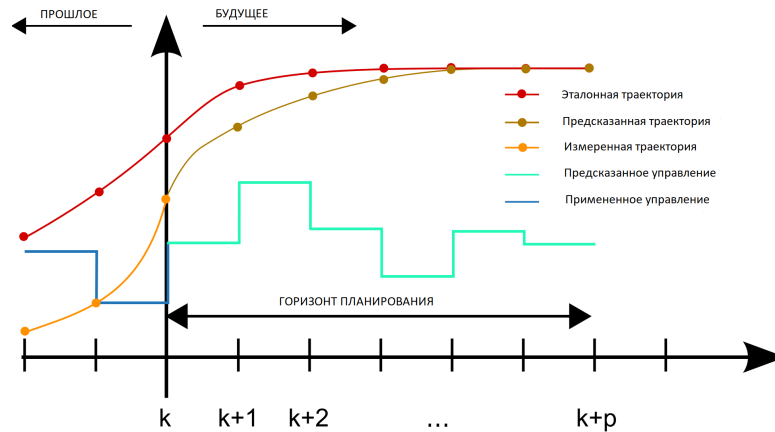


Рис. 1.1: Схема MPC

Понятно, что далее необходимо различать состояния объекта управления $x(t)$, $t \in \mathbb{I}_{\geq 0}$, которые измеряются в каждом конкретном процессе управления, и состояния математической модели, которая используется для предсказаний и формулировки прогнозирующей задачи ОУ. Поэтому состояния математической модели будем обозначать $x(k|t)$, $k \in \mathbb{I}_{[0, N-1]}$. Они изменяются

согласно уравнению

$$x(k+1|t) = f(x(k|t), u(k|t)), \quad x(0|t) = x(t), \quad k \in I_{[0, N-1]}. \quad (1.4)$$

Здесь аргумент t после черты подчеркивает зависимость от текущего момента, для которого проводится оптимизация. Начальное состояние — текущее состояние объекта управления $x(t)$.

Ограничения (1.2), записанные для состояний математической модели (1.4), имеют вид:

$$(x(k|t), u(k|t)) \in Z, \quad k \in \mathbb{I}_{[0, N-1]}.$$

Кроме приведенных смешанных ограничений, в задачу, как правило, добавляются ограничения в терминальный момент времени. "Терминальные элементы" прогнозирующей задачи более подробно будут рассмотрены ниже после ее формулировки.

Оставшийся элемент прогнозирующей задачи ОУ — критерий качества. В задачах стабилизации критерий качества выбирается исследователем, практиком, и является, скорее, параметром настройки схемы МРС. Например, в задаче стабилизации (см. [2]) критерий качества выбирается из соображений штрафа любого состояния $x \in X$, отклоняющегося от состояния равновесия x^* . Также часто штрафуются отклонения управления $u \in U$ от значения u^* . Как отмечается в [2], последнее условие полезно с вычислительной точки зрения, поскольку для численных методов зачастую проще решить задачу, в которой в критерии качества штрафуются управляющие воздействия. С другой стороны [2], с точки зрения реализации управления также желательно избежать значений $u \in U$, соответствующих чрезмерным энергетическим затратам.

Критерий качества будет состоять из терминальной стоимости $V_f(x(N|t))$ и суммарной стоимости переходного процесса, т.е. это будет критерий качества типа Больца. Терминальная стоимость будет рассмотрена ниже, при обсуждении терминальных элементов задачи ОУ. Стоимость переходного процесса для дискретных систем задается суммой стоимостей за каждый этап (для каждого $k \in \mathbb{I}_{[0, N-1]}$):

$$\sum_{k=0}^{N-1} l(x(k|t), u(k|t)).$$

В литературе функция $l : \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}$ называется стоимостью этапа (stage cost). Предполагается [2], что она непрерывна, а также:

1. $l(x^*, u^*) = 0$, т.е. стоимость обращается в нуль в точке равновесия;

2. существует функция α_1 класса \mathcal{K}_∞ , что выполняется

$$l(x, u) \geq \alpha_1(|x - x^*|) \quad \forall (x, u) \in Z.$$

Таким образом, прогнозирующая задача ОУ для момента времени t имеет вид:

$$\mathcal{P}(t) : \quad V(x(t)) = \min_{u(\cdot|t)} \sum_{k=0}^{N-1} l(x(k|t), u(k|t)) + V_f(x(N|t)), \quad (1.5)$$

при условиях

$$x(k+1|t) = f(x(k|t), u(k|t)), \quad k \in \mathbb{I}_{[0, N-1]},$$

$$x(0|t) = x(t),$$

$$(x(k|t), u(k|t)) \in Z, \quad k \in \mathbb{I}_{[0, N-1]},$$

$$x(N|t) \in X_f.$$

В задаче (1.5) обсуждавшиеся выше "терминальные элементы":

- терминальная стоимость $V_f(x(N|t))$ в критерии качества;
- терминальное ограничение $x(N|t) \in X_f$, где X_f — терминальное множество.

Именно условия на эти элементы обеспечивают устойчивость замкнутой системы несмотря на то, что решается задача с конечным горизонтом (см. раздел 1.4).

Далее используем следующие обозначения:

$u^0(\cdot|t) = \{u^0(0|t), \dots, u^0(N-1|t)\}$ — оптимальное (программное) управление задачи $\mathcal{P}(t)$;

$x^0(\cdot|t) = \{x^0(0|t), \dots, x^0(N|t)\}$ — соответствующая траектория;

X_N — множество всех состояний $x \in X$, для которых существует решение задачи (1.5) с $x(t) = x$.

Базовый алгоритм МРС состоит в следующем:

Для каждого $t \in \mathbb{I}_{\geq 0}$

1. измерить состояние $x(t) \in X$ системы (1.1);
2. решить задачу (1.5) с начальным условием $x(0|t) = x(t)$, получить ее решение $u^0(\cdot|t)$;

3. подать на вход системы (1.1) управляющее воздействие

$$u_{MPC}(t) := u^0(0|t). \quad (1.6)$$

Таким образом, в каждый момент $t \in \mathbb{I}_{\geq 0}$ на систему подается управляющее воздействие (1.6), которое неявно зависит от текущего состояния $x(t)$. Соответственно, замкнутая система имеет вид

$$x(t+1) = f(x(t), u^0(0|t)), \quad t \in \mathbb{I}_{\geq 0}. \quad (1.7)$$

После 20 лет исследований теория МРС приобрела свои нынешние черты. Согласно [5], все схемы номинального МРС описываются представленным базовым алгоритмом, а все прогнозирующие задачи ОУ в общем виде формулируются как (1.5).

1.4 Терминальные элементы и устойчивость замкнутой системы

Схемы МРС зависят от выбора терминальных элементов V_f и X_f . Например, самые первые результаты исследований по МРС [2] были получены для состояния равновесия, совпадающего с началом координат $(x^*, u^*) = (0, 0)$ (т.е. $f(0, 0) = 0$) и предлагали использовать $X_f = \{0\}$, т.е. терминальное условие принимало вид $x(N|t) = 0$. Понятно, что включение терминальной стоимости в критерий качества в таком подходе не имеет смысла. Прогнозирующая задача ОУ $\mathcal{P}(t)$ принимает вид

$$\mathcal{P}(t) : \quad V(x(t)) = \min_{u(\cdot|t)} \sum_{k=0}^{N-1} l(x(k|t), u(k|t)),$$

при условиях

$$x(k+1|t) = f(x(k|t), u(k|t)), \quad k \in \mathbb{I}_{[0, N-1]},$$

$$x(0|t) = x(t),$$

$$(x(k|t), u(k|t)) \in Z, \quad k \in \mathbb{I}_{[0, N-1]},$$

$$x(N|t) = 0.$$

Сразу отметим, что ограничения-равенства считаются "плохими" с точ-

ки зрения вычислительных методов оптимизации, поэтому дальнейшее развитие теории продолжалось в направлении ослабления этих простейших условий.

В частности, в работе [5] приведены следующие общие требования, которым должны удовлетворять терминальное множество X_f и терминальная стоимость V_f . Эти условия дополняют условия 1–2 на функцию стоимости этапа l и имеют вид:

3. терминальная стоимость V_f непрерывна на X ;
4. терминальное множество X_f замкнуто, $\{x^*\} \in X_f$;
5. существует локальная обратная связь $k_f : X_f \rightarrow U$, такая что для $\forall x \in X_f$ имеет место
 - а) $(x, k_f(x)) \in Z$, т.е. локальная обратная связь допустима;
 - б) $f(x, k_f(x)) \in X_f$, т.е. терминальное множество X_f является положительно инвариантным для системы $x(t+1) = f(x(t), k_f(x(t)))$;
 - в) $V_f(f(x, k_f(x))) - V_f(x) \leq -l(x, k_f(x)) + l(x^*, u^*)$, откуда следует, что терминальная стоимость V_f может служить функцией Ляпунова на терминальном множестве X_f .

В работах [1, 2] для дискретных систем можно найти следующий основной результат:

Теорема 1.1 Пусть $x_0 \in X_N$ и выполнены все предположения 1 – 5 относительно функций l , V_f и терминального множества X_f . Тогда

1. замкнутая система (1.7), полученная в результате применения базового алгоритма МРС, удовлетворяет ограничениям (1.2) для всех $t \in \mathbb{I}_{\geq 0}$;
2. задача (1.5) имеет решение для всех $t \in \mathbb{I}_{\geq 0}$;
3. x^* — асимптотически устойчивое положение равновесия системы (1.7) с областью притяжения X_N .

Выбор терминальных элементов V_f и X_f зачастую зависит от конкретной задачи, однако существует общий подход к их нахождению и этот способ называется МРС на квази-бесконечном горизонте.

МРС на квази-бесконечном горизонте имеет две отличительные черты:

1. локальная обратная связь является линейной, строится по первому приближению нелинейной системы; она является фиктивной, т.е. для фактического управления системой не используется, однако служит для построения терминальных элементов и при доказательстве основных результатов по асимптотической устойчивости;
2. терминальное ограничение и терминальная стоимость являются квадратичными, их параметры находятся по решению уравнения Ляпунова.

Впервые идея квази-бесконечного MPC появилась в работе [4] для непрерывных нелинейных систем

$$\dot{x} = f(x, u), \quad x(t_0) = x_0, \quad (1.8)$$

с дважды непрерывно дифференцируемой функцией $f : \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^n$, для которых считается, что положение равновесия находится в начале координат, т.е. $f(0, 0) = 0$, при этом $\{0\} \in \text{int } U$.

В квази-бесконечном MPC все элементы прогнозирующей задачи выбираются квадратичными: функция стоимости этапа задается в виде

$$L(x, u) = \|x\|_Q^2 + \|u\|_R^2 = x^T Q x + u^T R u; \quad (1.9)$$

терминальная стоимость — квадратичная функция вида:

$$V_f(x) = \|x\|_P^2 = x^T P x; \quad (1.10)$$

терминальное множество — эллипсоид вида:

$$X_f = \{x \in \mathbb{R}^n \mid x^T P x \leq \alpha\}, \quad (1.11)$$

где $Q \in \mathbb{R}^{n \times n}$, $R \in \mathbb{R}^{r \times r}$, $P \in \mathbb{R}^{n \times n}$ — положительно определенные матрицы. Матрицы Q, R выбираются исходя из целей процесса управления (скорость сходимости, минимизация затрат и др.) и являются параметрами настройки MPC-регулятора. Матрица P и константа $\alpha > 0$ не зависят от выбора Q, R , а лишь от свойств системы управления. Их выбор описывается ниже.

Будем считать, что первое приближение системы (1.8) в начале координат

$$\dot{x} = Ax + Bu, \quad A = \partial f(0, 0)/\partial x, \quad B = \partial f(0, 0)/\partial u, \quad (1.12)$$

стабилизируемо. Это означает, что существует линейная обратная связь по

состоянию $k_f(x) = Kx$, что замкнутая система

$$\dot{x} = Ax + Bk_f(x) = (A + BK)x$$

асимптотически устойчива, т.е. матрица $A_K = A + BK$ является гурвицевой.

Известно [4], что если первое приближение (1.12) системы (1.8) в начале координат стабилизируемо, то

1. уравнение Ляпунова

$$(A_K + kI)^T P + P(A_K + kI) = -(Q + K^T R K) \quad (1.13)$$

имеет единственное положительно определенное решение P , где параметр $k \in [0, \infty)$ удовлетворяет неравенству $k < -\lambda_{\max}(A_K)$;

2. $\exists \alpha \in (0, \infty)$, определяющая окрестность начала координат

$$X_f^\alpha = \{x \in \mathbb{R}^n \mid x^T P x \leq \alpha\}$$

такую что

- а) $Kx \in U \quad \forall x \in X_f^\alpha$, т.е. линейная обратная связь в области X_f^α не нарушает ограничений на управление;
- б) множество X_f^α инвариантно для нелинейной системы (1.8), замкнутой локальной линейной обратной связью $k_f(x) = Kx$;
- в) для любого $x_1 \in X_f^\alpha$ терминальная стоимость $V_f(x_1)$ ограничивает сверху "хвост" критерия качества для бесконечного горизонта:

$$\int_{t_1}^{\infty} (\|x(t; t_1, x_1, u)\|_Q^2 + \|u(t)\|_R^2) dt \leq x_1^T P x_1.$$

Тогда алгоритм нахождения терминальной функции и терминального множества, согласно [4] состоит в следующем:

1. найти линейную обратную связь $k_f(x) = Kx$;
2. выбрать константу $k \in [0, \infty)$, удовлетворяющую неравенству

$$k < -\lambda_{\max}(A_K),$$

и решить уравнение Ляпунова (1.13) для нахождения P ;

3. найти наибольшее α_1 , для которого имеет место $Kx \in U \quad \forall x \in X_f^{\alpha_1}$;

4. найти наибольшее $\alpha \in (0, \alpha_1)$, при котором выполнено неравенство

$$\sup \left(\frac{\|f(x, Kx) - A_K x\|}{\|x\|} \mid x \in X_f^\alpha, x \neq 0 \right) \leq \frac{k\lambda_{\min}(P)}{\|P\|}.$$

В итоге получим матрицу P и значение α , которые определяют терминальную функцию (1.10) и терминальное множество (1.11). Эти элементы строятся оффлайн, до начала процесса управления.

Теорема 1.2 [4] Если первое приближение (1.12) системы (1.8) в начале координат стабилизируемо и прогнозирующая задача оптимального управления $\mathcal{P}(t)$ имеет решение в $t = 0$, то в отсутствие возмущений замкнутая система асимптотически устойчива. Кроме того, если $X \subseteq \mathbb{R}^n$ — множество состояний x_0 , для которых $\mathcal{P}(0)$ имеет решение, то X — область притяжения замкнутой системы.

1.5 Выводы

Основная идея алгоритма МРС состоит в том, чтобы в каждый момент времени оптимизировать будущее поведение системы на конечном горизонте и использовать первое значение полученного оптимального (программного) управления в качестве значения обратной связи для этого момента времени. В связи с тем, что эта идея интуитивно понятна практикам и достаточно проста в реализации, она получила широкое распространение в промышленных приложениях [3].

Метод опирается на решение задач ОУ в режиме реального времени, которое в подавляющем числе случаев может быть получено только численно. Несмотря на то, что с привлечением методов алгоритмического дифференцирования численные методы решения задач ОУ достигли значительной эффективности [17, 18] и развитие вычислительной техники позволяет быстро решать достаточно сложные задачи, для существенно нелинейных систем, систем большой размерности, систем с быстро меняющейся динамикой, существующие методы могут оказаться неэффективными или слишком медленными. В связи с этим недостатком, далее в магистерской диссертации предлагается вынести некоторые вычисления из классического алгоритма МРС оффлайн, что позволит повысить производительность систем управления с прогнозирующей моделью.

ГЛАВА 2

ИСПОЛЬЗОВАНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В СИСТЕМАХ УПРАВЛЕНИЯ С ПРОГНОЗИРУЮЩЕЙ МОДЕЛЬЮ

2.1 Основные понятия нейронных сетей

Нейронная сеть представляет собой серию алгоритмов, которые стремятся распознать базовые отношения в наборе данных посредством процесса, который имитирует работу человеческого мозга.

Нейронная сеть основана на наборе связанных единиц или узлов, называемых искусственными нейронами, которые свободно моделируют нейроны в биологическом мозге. Каждое соединение, подобно синапсам в биологическом мозге, может передавать сигнал от одного искусственного нейрона к другому. Искусственный нейрон, который получает сигнал, может обрабатывать его, а затем сигнализировать дополнительные искусственные нейроны, связанные с ним.

В общих реализациях нейронных сетей сигнал при соединении между искусственными нейронами является действительным числом, а выход каждого искусственного нейрона вычисляется некоторой нелинейной функцией от суммы его входов. Связи между искусственными нейронами называются ребрами. Искусственные нейроны и ребра обычно имеют вес, который регулируется по мере продолжения обучения. Вес увеличивает или уменьшает силу сигнала при соединении. Искусственные нейроны могут иметь такой порог, что сигнал посылается только тогда, когда совокупный сигнал пересекает этот порог. Как правило, искусственные нейроны агрегируются в слои. Различные слои могут выполнять различные виды преобразований на своих входах. Сигналы перемещаются от первого уровня (входного уровня) к последнему слою (выходному слою), возможно, после пересечения слоев несколько раз. [16]

Ключевой моделью глубокого обучения являются нейронные сети с прямым распространением (многослойные персептроны). Целью данного вида нейронных сетей является аппроксимация некоторой функции f^* . Например, для классификатора $y = f^*(x)$ сеть отображает вход x в категорию y . Сеть определяет отображение $y = f(x; \theta)$ и изучает значение параметров θ , кото-

рые приводят к приближению функции наилучшим образом.

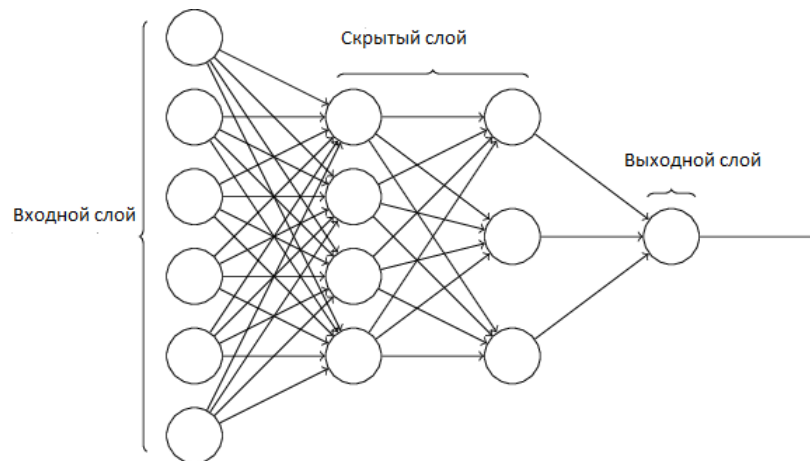


Рис. 2.1: Нейронная сеть со скрытыми слоями.

Нейронные сети называются сетями, потому что они обычно представляются объединением многих различных функций. Модель связана с ориентированным ациклическим графом (Рис. 2.1), описывающим, как функции состоят вместе. Например, мы могли бы иметь три функции $f^{(1)}$, $f^{(2)}$ и $f^{(3)}$, связанные в цепочке, с образованием $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$. Эти цепные структуры являются наиболее часто используемыми структурами нейронных сетей. В этом случае $f^{(1)}$ называется первым слоем сети, $f^{(2)}$ называется вторым слоем и т. д. Длина цепочки слоев называется глубиной сети.

Каждый скрытый уровень сети обычно является векторным. Размерность этих скрытых слоев определяет ширину модели. Каждый элемент вектора может быть интерпретирован как играющий роль, аналогичную нейрону. Вместо того, чтобы думать о том, что слой представляет собой единую вектор-векторную функцию, мы также можем думать о том, что этот слой состоит из множества единиц, которые действуют параллельно, каждый из которых представляет собой вектор-скалярную функцию. Каждый блок напоминает нейрон в том смысле, что он получает вход от многих других единиц и вычисляет его собственное значение активации.

Нейрон обычно получает много одновременных входов. Каждый вход имеет свой собственный относительный вес, который дает входное воздействие, которое ему необходимо для функции суммирования элемента обработки. Эти веса выполняют тот же тип функции, что и различные синаптические силы биологических нейронов. В обоих случаях некоторые входы становятся более важными, чем другие, так что они оказывают большее влияние на обрабатывающий элемент, поскольку они объединяются для создания нейронного ответа. Веса - это адаптивные коэффициенты в сети, которые определяют интенсивность входного сигнала, зарегистрированного искусственным

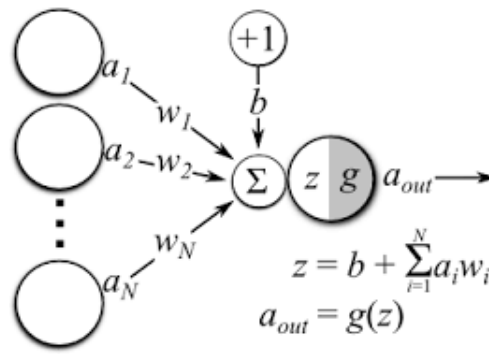


Рис. 2.2: Строение нейрона.

нейроном. Они являются мерой прочности соединения входа. Эти сильные стороны могут быть изменены в ответ на различные обучающие наборы и в соответствии с конкретной топологией сети или с помощью ее правил обучения. На рисунке (2.2) веса обозначены w_i , значения нейронов предыдущего слоя - a_i . b параметр представляет собой смещение для линейного преобразования входных нейронов. Таким образом, мы получаем значение функции суммирования в виде линейного преобразования $z = b + \sum_{i=1}^N a_i w_i$.

Функция g на рисунке (2.2) - это функция активации нейрона. Цель использования функции активации заключается в том, чтобы позволить суммируемому результату меняться в зависимости от времени. Функцией по умолчанию является выпрямленная линейная активационная функция ReLU $g(x) = \max(0, x)$, которая рекомендована для использования с большинством нейронных сетей прямого распространения. Применение этой функции к выходу линейного преобразования приводит к нелинейному преобразованию. Однако функция остается очень близкой к линейной, в том смысле, что это кусочно-линейная функция с двумя линейными частями. Поскольку выпрямленные линейные единицы почти линейны, они сохраняют многие свойства, которые упрощают оптимизацию линейных моделей с помощью методов, основанных на градиенте. Другими популярными видами функции активации являются сигмоидальная (логистическая) функция $\sigma(x) = \frac{1}{1+e^{-x}}$, гиперболический тангенс $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

Существует несколько видов обучения нейронных сетей:

- Обучение с учителем
- Обучение без учителя

Подавляющее большинство искусственных нейронных сетевых решений проходят обучение с учителем. В этом режиме фактический выход нейронной сети сравнивается с желаемым выходом. Веса, которые обычно начинаются с произвольного начала, затем корректируются сетью, так что следующая

итерация или цикл приведут к более близкому совпадению между желаемым и фактическим выходом. Метод обучения пытается минимизировать текущие ошибки всех элементов обработки. Это глобальное сокращение ошибок создается со временем, постоянно изменяя веса ввода до тех пор, пока не будет достигнута приемлемая точность сети. При контролируемом обучении искусственная нейронная сеть должна быть обучена, прежде чем она станет полезна. Обучение состоит в представлении входных и выходных данных в сеть. Эти данные часто упоминаются как набор тренировок. То есть для каждого набора входных данных, предоставляемого системе, также предусмотрен соответствующий желаемый выходной набор. В большинстве приложений должны использоваться фактические данные. Эта стадия обучения может потреблять много времени. Затем используя метрики для расчета точности и качества модели, происходит процесс тренировки сети. Когда процесс тренировки заканчивается, то уже в online процессах используются эти натренированные параметры и веса. Некоторые типы сетей позволяют проводить непрерывную тренировку с гораздо меньшей скоростью во время работы. Это помогает сети адаптироваться к постепенно меняющимся условиям.

Сети без учителя не используют внешние воздействия для корректировки своих весов. Вместо этого они внутренне контролируют свою работу. Эти сети ищут закономерности или тенденции во входных сигналах и делают адаптацию в соответствии с функцией сети. Хотя и сеть обучается сама, необходимо специализировать, как сети организовать себя. Эта информация встроена в сетевую топологию и правила обучения.

Алгоритм обучения - алгоритм обратного распространения ошибки, в котором используется стохастический градиентный спуск. Для задачи регрессии чаще всего в качестве функции потерь используется средняя квадратичная ошибка(MSE)(2.1):

$$L(y_{out}, y_{true}) = \frac{1}{N} \sum_{i=1}^N (y_{out}(i) - y_{true}(i))^2 \quad (2.1)$$

Согласно универсальной теореме аппроксимации — нейронная сеть с одним скрытым слоем может аппроксимировать любую непрерывную функцию многих переменных с любой точностью. Главное чтобы в этой сети было достаточное количество нейронов. И еще важно удачно подобрать начальные значения весов нейронов. Чем удачнее будут подобраны веса, тем быстрее нейронная сеть будет сходиться к исходной функции. Это означает, что нелинейная характеристика нейрона может быть произвольной: от сигмоидальной до произвольного волнового пакета или вейвлета, синуса или многочлена. От

выбора нелинейной функции может зависеть сложность конкретной сети, но с любой нелинейностью сеть остаётся универсальным аппроксиматором и при правильном выборе структуры может достаточно точно аппроксимировать функционирование любой непрерывной функции.

2.2 Методы обучения с подкреплением

Обучение с подкреплением (RL) - это область машинного обучения, связанная с тем, как агенты программного обеспечения должны предпринимать действия в среде, чтобы максимизировать некоторое понятие кумулятивной награды. В литературе по исследованиям и контролю операций обучение с подкреплением называется приближенным динамическим программированием или нейродинамическим программированием. Проблемы интереса к обучению подкреплению изучались также в теории оптимального управления, которая в основном связана с существованием и характеристикой оптимальных решений, алгоритмами их точного вычисления или аппроксимацией, особенно в отсутствие математической модели среды.

В машинном обучении среда обычно формулируется как процесс принятия решений Маркова (MDP), так как многие алгоритмы обучения с подкреплением для этого контекста используют методы динамического программирования. Основное различие между классическими методами динамического программирования и алгоритмами обучения с подкреплением заключается в том, что последние не предполагают знания точной математической модели MDP и нацеливаются на большие MDP, где точные методы становятся неосуществимыми. Компромисс между использованием наилучшей вычисленной стратегии и исследованием новых стратегий наиболее тщательно изучается в рамках многорукой бандитской проблемы и в конечных MDP.

Базовое RL моделируется как процесс принятия марковских решений:

- набор состояний среды и агента S
- набор действий агента A
- $P_a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$ - вероятность перехода из состояния s в состояние s' под действием агента a .
- $R_a(s, s')$ - непосредственная награда после перехода от s к s' с действием a .

Обучение с подкреплением требует механизмов исследования. Случайный выбор действий без ссылки на вероятное распределение вероятности показывает низкую производительность. Простые методы исследования являются наиболее практичными.

Одним из таких методов является ϵ -greedy, когда агент выбирает действие, которое, по его мнению, имеет лучший долгосрочный эффект с вероятностью $1 - \epsilon$. Если никакое действие, удовлетворяющее этому условию, не найдено, агент выбирает действие равномерно случайным образом. Здесь $\epsilon < 1$ является параметром настройки, который иногда изменяется либо по фиксированному расписанию, либо адаптивно основываясь на эвристике.

Выбор действия агента моделируется как таблица, называемая стратегией $\pi : S \times A \rightarrow [0, 1]$:

$$\pi(a|s) = P(a_t = a | s_t = s) \quad (2.2)$$

Таблица стратегии дает вероятность принятия действий a в состоянии s . Существуют также невероятностные стратегии.

Функция стоимости $V_{\pi}(s)$ определяется как ожидаемый возврат, начиная с состояния s , то есть $s_0 = s$ и последовательно следуют политике π . Следовательно, грубо говоря, функция значений оценивает «насколько хорошо» она должна находиться в определенном состоянии.

$$V_{\pi}(s) = E[R] = E\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s\right] \quad (2.3)$$

где случайная величина R обозначает прибыль, и определяется как сумма будущих дисконтированных вознаграждений.

$$R = \sum_{t=0}^{\infty} \gamma^t r_t \quad (2.4)$$

где r_t - вознаграждение на этапе t , $\gamma \in [0, 1]$ - ставка дисконтирования.

Функция стоимости пытается найти стратегию, которая максимизирует прибыль, поддерживая набор оценок ожидаемых результатов для некоторой стратегии (обычно либо «текущая» [внутри стратегии], либо оптимальная [вне стратегии]).

Эти методы основаны на теории MDP, где оптимальность определяется как: стратегия называется оптимальной, если она достигает наилучшей ожидаемой прибыли из любого начального состояния.

Чтобы определить оптимальность формальным образом, определим при-

быль от стратегии π

$$V^\pi(s) = E[R|s, \pi] \quad (2.5)$$

где R обозначает прибыль, связанную со следующей π из исходного состояния s . Определим $V^*(s)$ как максимально возможное значение $V^\pi(s)$, где π разрешено изменять,

$$V^*(s) = \max_{\pi} V^\pi(s) \quad (2.6)$$

Стратегия, которая достигает этих оптимальных значений в каждом состоянии, называется оптимальной.

Хотя значения состояний достаточно для определения оптимальности, полезно определить функцию прибыли от действия агента. Учитывая состояние s , действие a и политику π , значение действия пары (s, a) от π определяется формулой

$$Q^\pi(s, a) = E[R|s, a, \pi] \quad (2.7)$$

где R теперь обозначает случайную прибыль, связанную с первым действием a в состоянии s и последующим π .

Теория MDP утверждает, что если π^* является оптимальной стратегией, мы действуем оптимально (принимая оптимальное действие), выбирая действие из $Q^{\pi^*}(s, \cdot)$ с наивысшим значением в каждом состоянии, s . Функция прибыли от действия такой оптимальной стратегией Q^{π^*} называется оптимальной функцией прибыли от действия и является обычно обозначаемый Q^* . Таким образом, знание оптимальной функции стоимости действия достаточно, чтобы знать, как действовать оптимально.

Существует несколько алгоритмов для нахождения оптимальных стратегий: метод Монте-Карло, метод конечных разностей, метод прямого поиска стратегии. Каждый из которых имеет свои достоинства и недостатки, однако чаще всего используется стохастическая оптимизация и методы градиентного подъема.

2.3 Методы дизайна оффлайн регуляторов в системах управления с прогнозирующей моделью

Существует несколько подходов использования методов обучения в МРС системах:

- Аппроксимация закона управления
- Использование обучаемой модели для аппроксимации динамики про-

гнозирующей модели

- Итерационный подход для построения терминального региона и функции из предыдущих итераций

2.3.1 Аппроксимация закона управления

При некоторых слабых предположениях для линейных систем задача оптимизации может быть решена оффлайн, т.е. до начальной реальной процедуры управления [6]. В результате, получается явный закон управления $u(x)$. Развитие подхода на нелинейные системы не тривиально, и кроме того даже в линейных случаях существуют проблемы с эффективностью вычислений для метода из [6].

В момент времени t , вычисляем состояние $x(t)$, решаем линейную задачу с линейными ограничениями

$$V(x) = \min_{u(\cdot|t)} \sum_{k=t}^{t+N-1} L(x(k|t), u(k|t)) + F(x(t+N|t)) \quad (2.8)$$

при условиях

$$x(k+1|t) = Ax(k|t) + Bu(k|t), \quad t \leq k \leq t+N-1$$

$$x(t|t) = x(t), \quad t \leq k \leq t+N-1$$

$$C_x x(k|t) \leq d_x, \quad t \leq k \leq t+N-1$$

$$C_u u(k|t) \leq d_u, \quad t \leq k \leq t+N-1$$

$$C^f x(t+N|t) \leq d^f,$$

с квадратичной функцией стоимости этапа и квадратичной терминальной функцией

$$L(x(t), u(t)) = x(t)^T Q x(t) + u(t)^T R u(t), \quad Q, R > 0, \quad F(x(t)) = x(t)^T P x(t)$$

Можно переписать задачу 2.8 в виде задачи квадратичного программирования. Для этого обозначим

$$X := [x^T(t+1|t), \dots, x^T(t+N|t)]^T, \quad (2.9)$$

$$U := [u^T(t+1|t), \dots, u^T(t+N-1|t)]^T, \quad (2.10)$$

Перепишем функцию стоимости

$$F(x(t), U) = x^T(t)Qx(t) + X^T\tilde{Q}X + U^T\tilde{R}U, \quad (2.11)$$

где $\tilde{Q} = \text{diag}(Q, \dots, Q, P) \in \mathbb{R}^{n \times (N+1)}$, $\tilde{R} = \text{diag}(R, \dots, R) \in \mathbb{R}^{m \times N}$

Перепишем динамику системы:

$$x(t+k|t) = A^k x(t) + \sum_{j=0}^{k-1} A^j B u(t+k-j-1|t), \quad k = 1, \dots, N, \quad (2.12)$$

Тогда

$$X = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} x(t) + \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ A^{N-1}B & A^{N-2}B & \dots & \dots & B \end{bmatrix} U \quad (2.13)$$

или, введя обозначения $\Omega = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}$ и $\Gamma = \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ A^{N-1}B & A^{N-2}B & \dots & \dots & B \end{bmatrix}$

сократим запись $X = \Omega x(t) + \Gamma U$.

С помощью (2.13) перепишем (2.11) :

$$J(x(t), U) = \frac{1}{2}x^T(t)Yx(t) + \frac{1}{2}U^T H U + x^T(t)F U$$

где

$$Y = 2(Q + \Omega^T \tilde{Q} \Omega),$$

$$H = 2(\Gamma^T \tilde{Q} \Gamma + \tilde{R}),$$

$$F = 2\Omega^T \tilde{Q} \Gamma.$$

Аналогично, для всех ограничений можно провести такие же преобразования

$$GU \leq W + Ex(t)$$

где

$$G = \text{diag}(C_x, \dots, C_x)\Gamma, \quad W = [d_x, \dots, d_x]^T, \quad E = \text{diag}(C_x, \dots, C_x)\Omega.$$

Тогда, задача (2.8) примет вид

$$\min_U \frac{1}{2}U^T H U + x^T F U + \frac{1}{2}x^T Y x, \quad (2.14)$$

при условии

$$GU \leq W + Ex(t) \quad (2.15)$$

Применим замену переменных вида

$$z := U + H^{-1}F^T x,$$

Нетрудно заметить, что H^{-1} — положительно определенная матрица. Тогда (2.14) примет вид

$$\min_z \frac{1}{2} z^T H z + \frac{1}{2} x^T \tilde{Y} x, \quad (2.16)$$

при условиях

$$Gz \leq W + Sx,$$

$$\tilde{Y} := Y - FH^{-1}F^T,$$

$$S := E + GH^{-1}F^T.$$

Далее эта задача решается с помощью теории выпуклой оптимизации через условия Каруша-Куна-Такера(ККТ). Так как задача (строго)выпуклая с допустимым множеством с непустой внутренней частью(по предположениям), то условия Слейтера выполняются. Оптимальное решение единственное и характеризуется условиями ККТ.

Явный МРС решает задачу для всех состояний, таким образом все пространство состояний делится на регионы, где в каждом регионе для состояния есть явная функция управления.

Алгоритм нахождения явных функций управления:

1. Взять любой $x_0 \in \mathbb{X}$
2. Решить задачу (2.16) с начальным условием $x = x_0$
3. Определить активные ограничения для оптимизационной задачи (2.16)
4. Вычислить критический регион по активным ограничениям и вычислить функцию управления для этого региона
5. Перейти к новому x_0

Главный недостаток этого метода состоит в том, что количество регионов может быть достаточно большим, что в онлайн процедуре может плохо сказываться на производительности. Так как в каждый момент времени нужно будет искать к какому из регионов относится текущее состояние, чтобы определить управление для него.

Существует несколько подходов к получению аппроксимативного решения для оптимизации МРС. Для линейных систем в [7] алгоритм обучения представлен дополнительными ограничениями для обеспечения стабильности и ограничения удовлетворения аппроксимационного МРС. Одним из подходов к аппроксимации МРС является выпуклое многопараметрическое нелинейное программирование [9], где вычисляется субоптимальная аппроксимация закона управления МРС. Другой подход - аппроксимировать МРС с помощью методов машинного обучения. Это делают нейронные сети в [10], [11], [12]. Эти методы не гарантируют устойчивость или удовлетворение ограничениям для аппроксимационного МРС, что особенно важно, если рассматривать жесткие ограничения на состояния. В [8] используется метод опорных векторов (SVM) для аппроксимации МРС. Устойчивость и удовлетворение ограничений могут быть гарантированы для произвольных ошибок малого приближения, основанных на присущих свойствам устойчивости. В [13] аппроксимируется МРС с липшицевым сужающим ограничением, что обеспечивает устойчивость при исчезающих ошибках аппроксимации. Ошибка аппроксимации, выведенная в [8], [13], обычно не достижима для практического применения.

В [11] хотят найти аппроксимирующий закон управления $u_t^{RH^0} = \gamma_{RH}^0(x_t) \in U$, где $u_t^{RH^0}$ - первый вектор последовательности управления, которая минимизирует стоимость

$$J_{FH}(x_t, u_{t,t+N-1}, N, a, P) = \sum_{i=t}^{t+N-1} l(x_i, u_i) + a\|x_{t+N}\|_P^2, \quad t \geq 0$$

Стоимость формируется из стоимости переходов на горизонте планирования длины N и терминальной функции. Аппроксимация закона управления происходит с помощью нейронной сети: m параллельных сетей с одним выходным параметром, состоящий из одного скрытого слоя с v_j нейронами на скрытом слое для сети $j = 1..m$ и линейными активационными функциями.

Для каждой функции $\hat{\gamma}_{RH_j}^{(v_j)}$ нужно найти количество нейронов v_1^*, \dots, v_m^* , такое что

$$\min_{w_j} \max_{x_t \in X} |\gamma_{RH_j}^0(x_t) - \hat{\gamma}_{RH_j}^{(v_j)}(x_t, w_j)| \leq \frac{\epsilon}{\sqrt{m}}, \quad j = 1, \dots, m \quad (2.17)$$

Процедура нахождения количества нейронов представляется таким образом: для каждого j увеличиваем v_j пока (2.17) не станет верным. Также приводится в статье теорема, в которой утверждается, что для любой функции управления $\gamma_{RH_j}^0(x_t)$ число параметров, необходимых для достижения

погрешности приближения L_2 или L_∞ порядка $O(\frac{1}{v_j})$, равно $O(v_j n)$, которое растет линейно с размерностью n вектора состояния.

В [12] рассматривалась система нелинейного МРС с управляемым выходом:

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ y(k) = h(x(k)) \end{cases}$$

И для этой системы минимизировалась стоимость вида:

$$J_N(U_N(k); k) = \sum_{i=k}^{k+N-1} [(\hat{y}(i+1|k) - y_r(i+1))^T Q (\hat{y}(i+1|k) - y_r(i+1)) + \Delta u(i)^T R \Delta u(i)] + q_N(\hat{x}(k+N|k)) \quad (2.18)$$

где y_r - эталонная траектория, $\Delta u(k) = u(k) - u(k-1)$ - инкрементное управление. Для этой задачи создают нейронный регулятор вида $\Delta u_{opt}(k) = g(I_{MPC}(k))$, где $I_{MPC}(k) = \{\hat{x}(k|k), y_r(k+1), \dots, y_r(k+N)\}$. Данный нейронный регулятор обучается с функцией потерь вида (2.18). Для решения оптимизационной задачи на этапе обучения нейронной сети используется градиентный спуск. Достаточно хорошие результаты на вычислительных этапах были получены, однако опять же не гарантируется устойчивость данного метода.

В работе [13] уже вводятся некоторые гарантии устойчивости метода на основании предположения о непрерывности по Липшицу функции динамики. Система представляется следующим образом:

$$x_{t+1} = f(x_t, u_t, \xi_t), \quad t \geq 0, \quad x_0 = \tilde{x}$$

где $\xi_t \in \mathbb{R}^r$ - возмущение системы. Номинальная система вводится для дизайна управления таким образом:

$$x_{t+1} = \hat{f}(x_t, u_t) + d_t, \quad t \geq 0, \quad x_0 = \tilde{x}$$

где $d_t = f(x_t, u_t, \xi_t) - \hat{f}(x_t, u_t)$. Предполагают непрерывность по Липшицу для функции \hat{f} относительно x с константой L_{f_x} , а также относительно управления u предполагают, что существует функция K класса, такая что $|\hat{f}(x, u) - \hat{f}(x, u')| \leq \eta_u(|u - u'|) \forall x \in X$ и $\forall (u, u') \in U^2$. Также вводится предположение об ограниченности возмущения и то что верно $|d_t| \leq \mu(|\xi_t|) \quad t \geq 0$, а также $d_t \in D = B^m(\bar{d}) \quad \bar{d} \in \mathbb{R}_{\geq 0} < \infty$. Предполагается, что для системы существует управление $k(x_t) \in U$, которое является стабилизирующим относительно состояния.

Накладываются дополнительные ограничения на погрешности относительно состояния $q_t \in Q = B^m(\bar{q})$ и управления $v_t \in V = B^m(\bar{v})$, где погрешность аппроксимации состояния $q_t = \xi_{x_t} - x_t$ и управления $v_t = k^*(x_t) - k(\xi_{x_t})$. Тогда система уже переписывается в таком виде:

$$x_{t+1} = \hat{f}(x_t, k(x_t + q_t) + v_t) + w_t, \quad x_0 = \tilde{x}, \quad t \geq 0$$

И устойчивость системы доказывается, если верно следующее, что погрешности аппроксимации состояния и управления совместно с возмущением должны быть ограничены изначальным возмущением системы $\bar{d}_q + \bar{d}_v + \bar{d}_w \leq \bar{d}$. В данной статье рассматривались два метода аппроксимации закона управления с помощью метода ближайшей точки и с помощью нейронной сети. Метод ближайшей точки обладает рядом недостатков, так как хранит целую сетку значений управления для состояний и еще требует в онлайн части решать задачу по нахождению ближайшей точки для текущего состояния. Нейронная сеть показала довольно хорошие результаты, так как она не обладает такими недостатками.

2.3.2 Аппроксимация динамики системы с прогнозирующей моделью

Существует методы для аппроксимации динамики системы, в работе [14] был предложен доказуемо безопасный и робастный метод управления основанный на обучении для систем с прогнозирующей моделью. Введена форма надежного, адаптивного модельного прогнозирующего управления, основанный на обучении метод, на который ссылаются, как LBMPC. LBMPC повышает производительность за счет выбора входных данных, которые минимизируют затраты, связанные с динамикой изученной модели, которая обновляется с использованием статистики, обеспечивая при этом безопасность и устойчивость, используя теорию из робастного MPC, чтобы проверить, примененное управление сохраняет ли номинальную модель устойчивой, когда она подвержена неопределенности.

Динамика системы представляется в таком виде:

$$x_{n+1} = Ax_n + Bu_n + g(x_n, u_n)$$

где $g(x, u)$ описывает несмоделированную динамику, которая по предположению ограничена и лежит в политопе W .

Данный метод вводит дополнительную систему, которая обучается на

данных и имеет такой вид:

$$\tilde{x}_{n+1} = A\tilde{x}_n + B\tilde{u}_n + O_n(\tilde{x}_n, \tilde{u}_n)$$

где O_n - зависящая от времени функция, которая обучается с помощью любого из статистических методов.

Вся теория устойчивости строится на том, что наша система представляется в виде робастного МРС с обученной динамикой на известных вычисленных точках, и может адаптироваться к новым полученным точкам, чтобы улучшать точность обученной динамики системы.

Задача уже формулируется в таком виде:

$$V_n(x_n) = \min_{c, \theta} \phi_n(\theta, \tilde{x}_n, \dots, x_{n+N}, \tilde{u}_n, \dots, \tilde{u}_{n+N-1})$$

при условиях

$$\begin{aligned} \tilde{x}_n &= x_n, \quad \bar{x}_n = x_n \\ \tilde{x}_{n+i+1} &= A\tilde{x}_{n+i} + B\tilde{u}_{n+i} + O_n(\tilde{x}_{n+i}, \tilde{u}_{n+i}) \\ x_{n+i+1} &= Ax_{n+i} + Bu_{n+i} \\ u_{n+i} &= Kx_{n+i} + c_{n+i} \\ x_{n+i+1} &\in X \ominus R_i, \quad u_{n+i} \in U \ominus KR_i \\ (x_{n+N}, \theta) &\in \Omega \ominus (R_N \times \{0\}) \end{aligned}$$

где Ω - допустимое инвариантное робастное множество таких точек, что любая траектория системы с начальным условием, выбранным из этого множества и с управлением u_n , остается в множестве для любой последовательности ограниченного возмущения, удовлетворяя ограничениям на состояние и управление. И тогда с помощью данного неравенства формулируется удовлетворение ограничений:

$$\Omega \subseteq \{(\bar{x}, \theta) : \bar{x} \in X; \Lambda\theta \in X; K\bar{x} + (\Psi - K\Lambda)\theta \in U; \Psi\theta \in U\}$$

А с помощью данного неравенства инвариантность возмущения:

$$\begin{bmatrix} A + BK & B(\Psi - K\Lambda) \\ 0 & \mathbb{I} \end{bmatrix} \Omega \oplus (W \times \{0\}) \subseteq \Omega$$

Так как $\bar{x}_s = \Lambda\theta$ и $\bar{u}_s = \Psi\theta$, где $\theta \in \mathbb{R}^m$ и $\Lambda \in \mathbb{R}^{n \times m}$, $\Psi \in \mathbb{R}^{m \times m}$ - точки устойчивого состояния, то их можно достичь для системы если $A + BK$

устойчива по Шуру при управлении

$$\bar{u}_n = K(\bar{x}_n - \bar{x}_s) + \bar{u}_s = K\bar{x}_n + (\Psi - K\Lambda)\theta.$$

Множества $R_0 = \{0\}$ и $R_i = \oplus_{j=0}^{i-1} (A + BK)^j W$ необходимы для робастного МРС, сужающиеся ограничения.

Устойчивость данного метода основана на робастном МРС. Были получены хорошие результаты относительно предсказанных траекторий и быстрой сходимости к точке устойчивого состояния, однако по производительности уступало нелинейному МРС с нейросетевыми регуляторами.

2.3.3 Итерационный подход

Существует итерационный подход для построения МРС регулятора [15]. Регулятор имеет справочную информацию и способен улучшать свою производительность, изучая предыдущие итерации. Для обеспечения рекурсивной выполнимости и неубывающей производительности на каждой итерации используются безопасное терминальное множество и терминальная функция стоимости. Построение данного регулятора обеспечивает тот факт, что функция стоимости убывает с каждой итерацией, также из удовлетворения ограничений на $j - 1$ итерации следует удовлетворение ограничений на j итерации и точка равновесия замкнутой системы асимптотически устойчива. Оптимальность траекторий построенных этим регулятором доказывается для выпуклых задач.

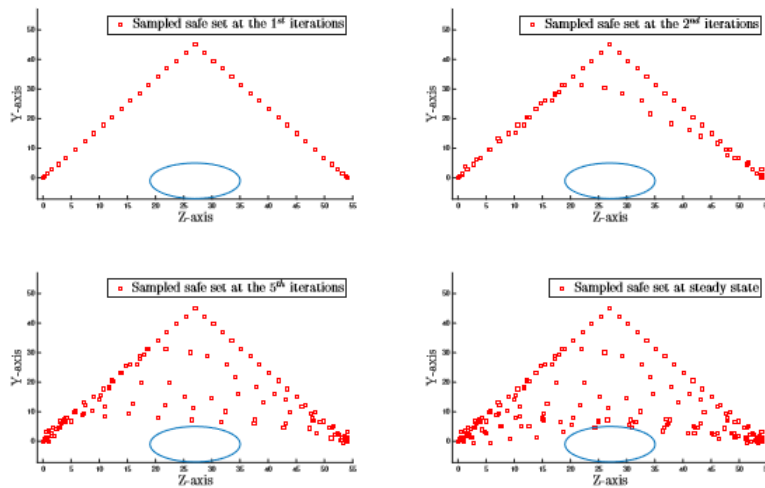


Рис. 2.3: Построение сэмплированного безопасного множества.

Управляемое на N шагов вперед множество по отношению к S форму-

лируется рекурсивно как:

$$K_j(S) = Pre(K_{j-1}(S)) \cap X, \quad K_0(S) = S, j \in \{1, \dots, N\} \quad (2.19)$$

где $Pre(S) = \{x \in \mathbb{R}^n : \exists u \in U \text{ s.t. } f(x, u) \in S\}$

Сэмплированное безопасное множество на итерации j

$$SS^j = \{\cup_{i \in M^j} \cup_{t=0}^{\infty} x_t^i\}$$

где $M^j = \{k \in [0, j] : \lim_{t \rightarrow \infty} x_t^k = x_F\}$. SS^j это множество всех траекторий на итерации i для $i \in M^j$. Как выглядит это множество показано на рисунке 2.3.

Вводится также стоимость на этом сэмплированном безопасном множестве:

$$Q^j(x) = \begin{cases} \min_{(i,t) \in F^j(x)} J_{t \rightarrow \infty}^i, & x \in SS^j \\ +\infty, & x \notin SS^j \end{cases}$$

Тогда формулировка MPC функции стоимости, которую необходимо минимизировать будет звучать так:

$$J_{t \rightarrow t+N}^{LMPC,j}(x_t^j) = \min_{u_{t|t}, \dots, u_{t+N-1|t}} \left[\sum_{k=t}^{t+N-1} h(x_{k|t}, u_{k|t}) + Q^{j-1}(x_{t+N|t}) \right]$$

при условии

$$x_{k+1|t} = f(x_{k|t}, u_{k|t}) \quad \forall k \in [t, \dots, t+N-1]$$

$$x_{k|t} \in X, \quad u_{k|t} \in U, \quad \forall k \in [t, \dots, t+N-1]$$

$$x_{t+N|t} \in SS^{j-1}$$

$$x_{t|t} = x_t^j$$

Данный метод обладает рекурсивной выполнимостью, устойчивостью и сходимостью. Но предлагаемый подход является дорогостоящим с точки зрения вычисления даже для линейной системы, поскольку регулятор должен решать задачу смешанного целочисленного программирования в каждый момент времени. Есть улучшения данного подхода с использованием параллельных вычислений, а также попытки сделать терминальные ограничения более выпуклыми.

ГЛАВА 3

ПОСТРОЕНИЕ НЕЙРОСЕТЕВОГО РЕГУЛЯТОРА

3.1 Описание иллюстративного примера

Рассмотрим нелинейную дискретную динамическую систему

$$x(t+1) = f(x(t), u(t)), \quad (3.1)$$

где $x(t) \in \mathbb{R}^n$ - вектор состояний, $u(t) \in \mathbb{R}^m$ - вектор управления и $f(0, 0) = 0$. Мы рассматриваем ограничения вида

$$X = \{x \in \mathbb{R}^n | Hx \leq 1_p\}, \quad U = \{u \in \mathbb{R}^m | Lu \leq 1_q\} \quad (3.2)$$

Также необходимо, чтобы выполнялось ограничение

$$x(t), u(t) \in X \times U \quad \forall t \geq 0. \quad (3.3)$$

Будем оптимизировать функцию стоимости

$$\min_{u(\cdot|t)} \sum_{k=0}^{N-1} l(x, u) + V_f(X(N)). \quad (3.4)$$

Исследования будут проводиться на примере из [10].

Нелинейная система в дискретном виде представляет собой

$$\begin{aligned} \dot{x}_1(t+1) &= -x_2(t) + 0.5(1 + x_1(t))u(t) \\ \dot{x}_2(t+1) &= x_1(t) + 0.5(1 - 4x_2(t))u(t) \end{aligned} \quad (3.5)$$

с ограничениями

$$x \in X = \{-2 \leq x_1 \leq 0.5, -1 \leq x_2 \leq 2\} \quad (3.6)$$

$$u \in U = \{-1 \leq u \leq 1\} \quad (3.7)$$

Шаг дискретизации $\delta = 0.1$. Данная система имеет стабилизируемое пер-

вое приближение в начале координат, которое имеет вид

$$x(t+1) = Ax(t) + Bu, \quad A = \begin{bmatrix} 1 & -0.05 \\ 0.05 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.025 \\ 0.025 \end{bmatrix}.$$

Линейный локальный закон управления для терминального множества представляет собой $k_f(x(t)) = Kx(t) = [0.3799, -0.4201] x(t)$.

Параметры для терминального множества и функции посчитаны с помощью алгоритмов квази-бесконечного МРС из первой главы. Для данной системы функция качества имеет вид

$$V(x(t)) = \min_{u(\cdot|t)} \sum_{k=0}^{N-1} (||u(k|t)||^2 + 0.1||x(k|t)||^2) + V_f(x(t)) \quad (3.8)$$

где

$$V_f(x(t)) = x(t)^T P x(t) = x(t)^T \begin{bmatrix} 48.3043 & -2.53 \\ -2.53 & 70.8191 \end{bmatrix} x(t).$$

Терминальное множество задается в виде эллипсоида

$$X_f = \{x \in \mathbb{R}^n | x(t)^T P x \leq \alpha\},$$

где $\alpha = 0.2129$. Таким образом, для системы определены все дизайн параметры для проведения процедуры МРС алгоритма.

3.2 Подход к решению

Алгоритм МРС для стабилизации положения равновесия $x = 0$ системы (3.1) состоит в следующем: Для каждого $t = 0, 1, \dots$

1. измерить состояние $x(t) \in X$ системы (1);
2. решить задачу (3) с начальным условием $x(0|t) = x(t)$, получить $u^0(\cdot|t)$;
3. подать на вход системы (1) управляющее воздействие $u_{MPC}(x(t)) := u^0(0|t)$.

При выполнении ряда требований к функциям l , V_f и множеству X_f замкнутая система

$$x(t+1) = f(x(t), u_{MPC}(x(t))), \quad t = 0, 1, \dots,$$

асимптотически устойчива с областью притяжения

$$X_0 = \{x_0 : V(x_0) < +\infty\} \subseteq X,$$

состоящей из всех точек $x_0 \in X$, для которых задача $\mathcal{P}(x_0)$ имеет решение [1, 2]. В первой главе приводится простой способ построения квадратичных функций l , V_f и эллипсоида X_f , удовлетворяющих упомянутым условиям.

Мы будем рассматривать систему (3.5) в контексте робастного МРС, так как аппроксимационная ошибка управления представляет собой отклонение для функции управления, а это и есть частный случай робастного МРС.

В работе с использованием результатов [19] исследуются условия, при которых, несмотря на ошибки аппроксимации, гарантируется выполнение ограничений и асимптотическая устойчивость замкнутой системы. При выполнении следующих предположений для динамики $x(t+1) = f(x(t), u(t)) + d$, где d — допустимое отклонение для управления, можно гарантировать выше перечисленное:

1. Локальная инкрементируемая стабилизуемость. Существует такая обратная связь $k : X \times X \times U \rightarrow \mathbb{R}^m$, δ -функция Ляпунова $V_\delta : X \times X \times U \rightarrow \mathbb{R}_{\geq 0}$, которая непрерывная по первому аргументу и удовлетворяет $V_\delta(x, x, v) = 0 \ \forall x \in X, \ \forall v \in U$ и параметры $c_{\delta, l}$, $c_{\delta, u}$, δ_{loc} , $k_{max} \in \mathbb{R} > 0$, $\rho \in (0, 1)$, такие что следующие неравенства выполняются для $\forall (x, z, v) \in X \times X \times U$, $(z^+, v^+) \in X \times U$ с $V_\delta(x, z, v) \leq \delta_{loc}$:

$$c_{\delta, l} \|x - z\|^2 \leq V_\delta(x, z, v) \leq c_{\delta, u},$$

$$\|k(x, z, v) - v\| \geq k_{max} \|x - z\|,$$

$$V_\delta(x^+, z^+, v^+) \geq \rho V_\delta(x, z, v)$$

для динамики $x^+ = f(x, k(x, z, v))$, $z^+ = f(z, v)$.

2. Локальная непрерывность по Липшицу. Существует $\lambda \in \mathbb{R}$, такая что при $\forall x \in X, \ \forall u \in U, \ \forall u + d \in U$ будет

$$\|f(x, u + d) - f(x, u)\| \leq \lambda \|d\|_\infty.$$

3. Ограничение на отклонение функции управления должен удовлетворять $\eta \leq \frac{1}{\lambda} \sqrt{\frac{\delta_{loc}}{c_{\delta, u}}}$.

Для выполнения ограничений используются сужение ограничение в виде увеличивающейся трубки ограничений: чем ближе к началу координат, тем

более увеличивающееся трубка для ограничений. Поэтому мы заменяем ограничения (3.2) на суженные вида:

$$\bar{X}_k = (1 - \epsilon_k)X = \{x \in \mathbb{R}^n : Hx \leq (1 - \epsilon_k)1_p\},$$

$$\bar{U}_k = (1 - \epsilon_k)U = \{u \in \mathbb{R}^m : Lu \leq (1 - \epsilon_k)1_q\}.$$

где

$$\epsilon_k = \epsilon \left(\frac{1 - \sqrt{\rho}^k}{1 - \sqrt{\rho}} \right), \quad k \in \{0, 1, \dots, N\}, \quad \epsilon = \eta \lambda \sqrt{\frac{c_{\delta,u}}{c_{\delta,l}}} \max\{\|H\|_\infty, \|L\|_\infty k_{\max}\}.$$

Как отмечено выше, шаг 2 приведенного алгоритма может оказаться достаточно трудоемким. В связи с этим в настоящей работе предлагается вместо онлайн решения прогнозирующей задачи численными методами оптимального управления использовать до начала процесса управления ряд методов машинного обучения, которые на основе обучающей выборки $(x, u_{MPC}(x)) \in X \times U$ будут строить приближенные значения $\bar{u}_{MPC}(x(t))$ обратной связи $u_{MPC}(x)$, $x \in X$, для текущих состояний $x(t)$.

В настоящей работе применяются метод опорных векторов (Support Vector Machine, SVM) и нейронные сети. Метод опорных векторов с радиальной базисной функции Гаусса [8] используется, во-первых, для выделения и аппроксимации области притяжения X_0 системы (3.1). Это позволяет эффективно обрабатывать текущие состояния динамической системы и не допускать выход за пределы области притяжения при управлении с помощью приближенных обратных связей. Во-вторых, метод опорных векторов применяется для многоклассовой классификации с целью выделения областей насыщения управления. Наконец, в областях, в которых обратная связь u_{MPC} принимает промежуточные значения, она аппроксимируется с использованием нейронной сети.

Обучение нейронной сети и классификация на основе SVM дает приближенное управления типа обратной связи $\bar{u}_{MPC}(x)$, $x \in X_0$. Система, замкнутая обратной связью $\bar{u}_{MPC}(x)$, $x \in X_0$, имеет вид

$$x(t+1) = f(x(t), \bar{u}_{MPC}(x(t))), \quad t = 0, 1, \dots \quad (4)$$

Будет проводится сравнение приближенных законов управления, обученных на равномерной сетке, на сетке, полученной на основе равномерно распределенных последовательностей, и на случайной сетке с увеличением объема обучающей выборки в окрестности положения равновесия.

3.3 Базовая реализация нейросетевого регулятора

После проведения стандартной процедуры МРС для системы (3.5), описанной в главе 1 получилась следующая траектория для системы, представленная на рисунке 3.1, полученная обратная связь $\bar{u}(t)$ представлена на рисунке 3.2. Горизонт планирования был взят $N = 30$, т.е. мы планируем и высчитываем управления и состояния на 30 шагов вперед.

Базовый вариант включает себя создание нейронной сети для аппроксимации обратной связи $\bar{u}(t)$ для данной системы.

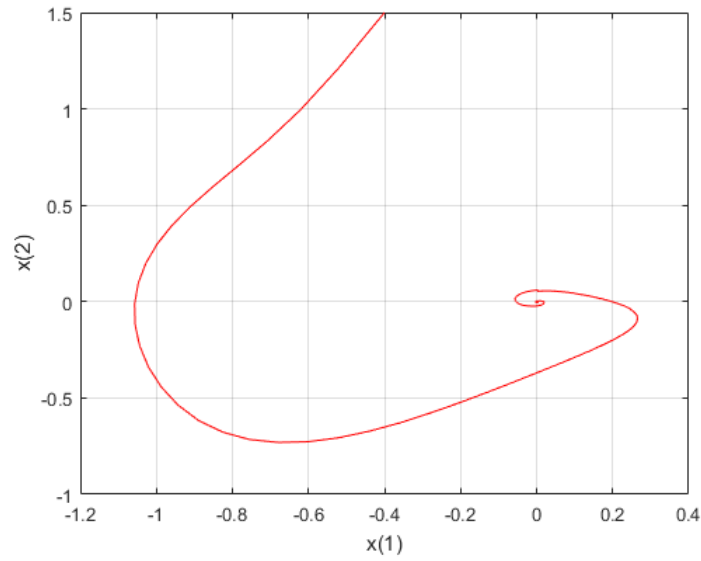


Рис. 3.1: Полученная траектория для системы (3.5) по стандартной процедуре МРС.

Для построения аппроксимации обратной связи будет вычисляться обратная связь для состояний из равномерной сетки всех допустимых состояний из множества (3.6) с шагом 0.05. Для этих точек будет построено управление с помощью стандартной процедуры МРС и будет запоминаться только первое управление для этого состояния. На рисунке 3.3 показано векторное поле сетки с шагом 0.1, в котором видно как происходит переход из одного состояния в другое, используя полученные обратные связи. На рисунке видно, что для большинства точек следующее состояние находится в пределах допустимых значений, однако если взять некоторые точки где следующее состояние выходит за рамки нашего обучения, то мы можем получить нестабильное решение.

Далее если посмотреть на рисунок 3.4, то можно увидеть множество одинаковых управлений для состояний. Однако важно отметить, что сетка значений около начала координат должна быть достаточно детальной, чтобы не

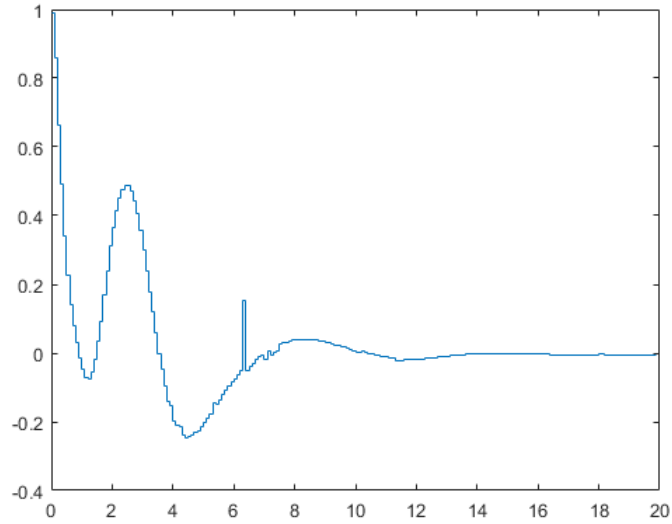


Рис. 3.2: Полученная обратная связь для системы (3.5) по стандартной процедуре МРС.

Таблица 3.1: Таблица зависимости точности аппроксимации от количества нейронов на скрытом слое

Количество нейронов	Ошибка аппроксимации
5	0.021
8	0.0092
10	0.008
15	0.0075
20	0.0063
25	0.0049

попасть в нестабильное положение управления.

В работе [11] использовался итеративный процесс для поиска оптимального количества нейронов для скрытого слоя: выбирается начальное значение количества нейронов, затем строится сеть, которая аппроксимирует функцию и далее для каждого состояния сетки вычисляется значение аппроксимированной обратной связи и обратной связи, полученной из стандартной процедуры МРС, если

$$\|\bar{u}(t) - u^*(t)\| < \frac{\epsilon}{\sqrt{m}},$$

где m — размерности вектора управления. Для достижения точности $5 \cdot 10^{-3}$ понадобилось 25 нейронов, в таблице 3.1 представлена зависимость точности от количества нейронов на скрытом слое.

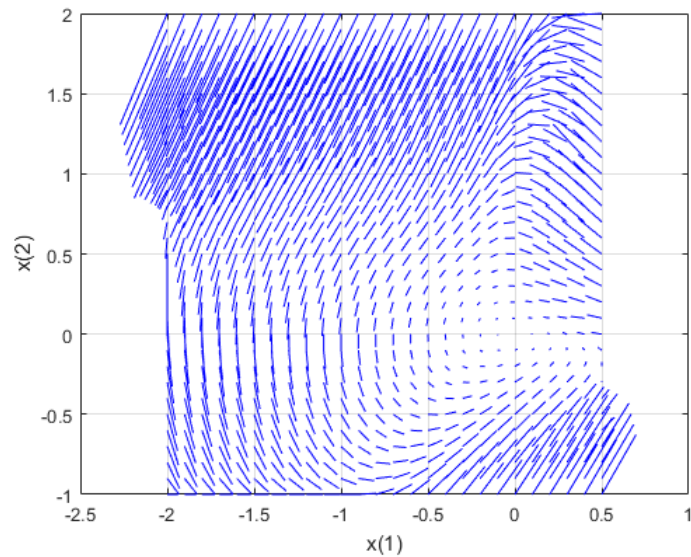


Рис. 3.3: Векторное поле для сетки с шагом 0.1.

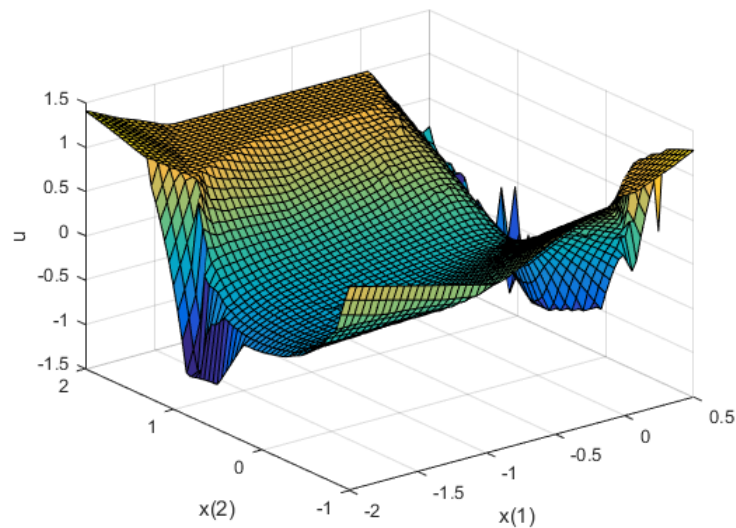


Рис. 3.4: Вычисленное управление для сетки значений состояний для обучения нейронной сети

На рисунке 3.5 представлены траектории для задачи (3.5), полученные при помощи стандартной процедуры МРС и при помощи построенного нейросетевого регулятора. Как видно из рисунка, получились достаточно близкие траектории и они сошли к началу координат.

Для этого подхода необходимо вычислить значения управления в каждой точке сетки состояний для допустимых значений состояний. Далее обучается сеть на этих значениях, как задача обучения с учителем. Основные недостатки данного подхода:

- Необходимо брать достаточно много точек для хорошей аппроксимации,

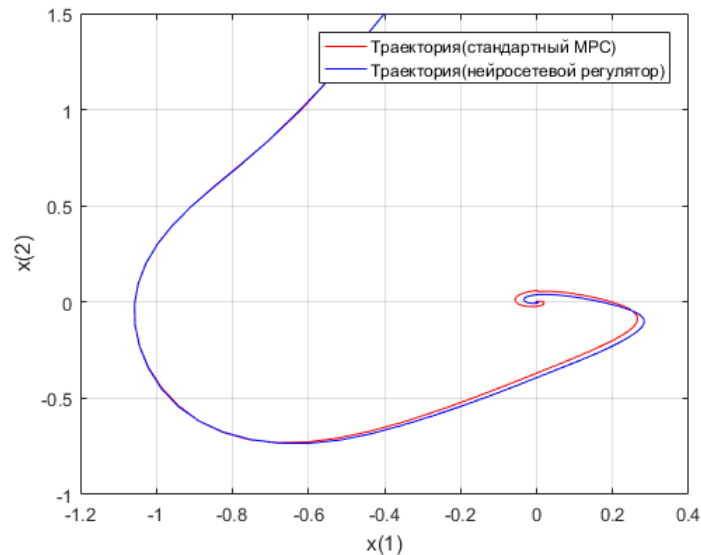


Рис. 3.5: Траектории, полученные при помощи стандартной процедуры MPC и нейросетевого регулятора

хотя иногда некоторые области имеют одинаковые значения и для них не надо вычислять заново управление,

- Обучение и валидация проходят достаточно долго

Предполагаемые улучшения для этого подхода:

- Уменьшить кол-во генерируемых точек для гарантирования аппроксимации, рассмотреть генерирование точек с помощью стохастического подхода и стохастического с увеличением плотности точек у начала координат,
- Вычислить X_0 — область притяжения, множество состояний из которых, можно достигнуть начало координат,
- Вычислить множество точек, для которых применяется насыщенное управление, для ускорения поиска управления в этих областях,
- Рассмотреть возможность использования обучение без учителя и обучение с подкреплением для построения множества допустимых значений и закона управления

3.4 Построение области притяжения с помощью SVM

При построении базового регулятора было замечено, что внутри области допустимых значений состояний есть область притяжения, множество точек, из которых можно достигнуть начало координат, и области, в которых могут быть нестабильные решения. Предлагается аппроксимировать область притяжения и только по точкам из области притяжения строить приближение для обратной связи.

Было сгенерировано 2000 точек внутри области (3.6) допустимых значений состояний с использованием квази-случайного метода построения с помощью множеств Хальтона. Для всех этих точек определяем возможность достижения из данного состояния точки начала координат. Помечаем состояние, из которого достижимо начало координат, с помощью пометки 1, а нестабильное состояние как -1 . Далее применяем метод SVM для аппроксимации области притяжения. В качестве функции ядра для метода SVM применяет радиальная функция Гаусса.

После применения SVM для аппроксимации области притяжения, предсказанные пометки для точек из области достижимых значений представлены на рисунке 3.6. Также на рисунке отмечены опорные вектора, которые определяют область стабильных состояний.

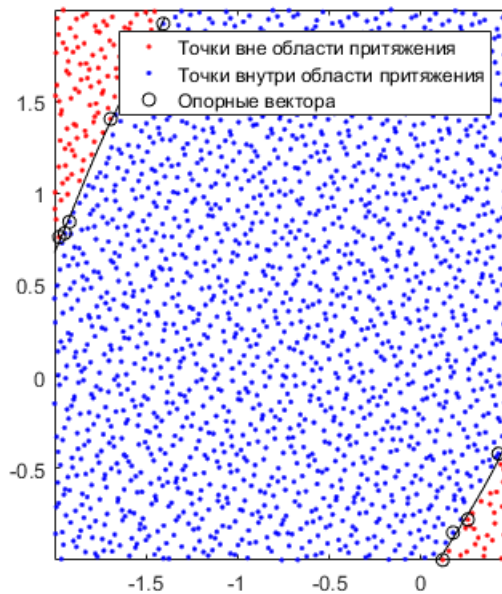
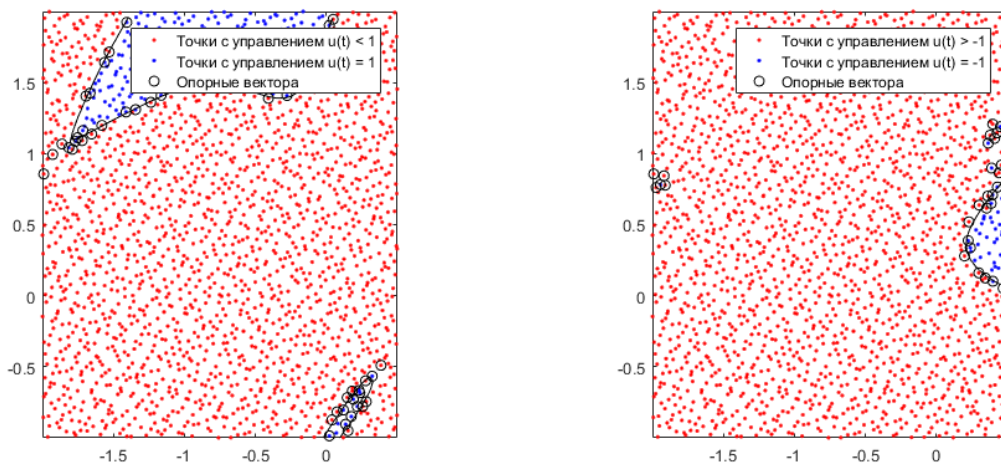


Рис. 3.6: Аппроксимированная область притяжения для задачи (3.5).

Для определения областей с насыщенными управлениями также вос-

пользуемся методом SVM. Метод SVM хорошо работает для разграничения двух классов, а в нашей задаче у нас будет 3 класса, поэтому будем создавать две отдельные модели для аппроксимации областей: первую модель для определения области с управлением $u(t) = 1$ и вторую модель для определения области с управлением $u(t) = -1$. Для каждой модели мы помечаем те состояния, которые соответствуют насыщенному управлению, как 1, а остальные состояния как -1 . Однако в нашем случае мы будем использовать, как и описывалось ранее суживающиеся ограничения, поэтому вместо насыщенных равных граничным значениям, будем определять с зазором в ϵ , т.е. пометим все состояния из которых будет управление $u(t) \geq 1 - \epsilon$ либо $u(t) \leq -1 + \epsilon$. Далее обучаем модели с помощью SVM с радиальной ядерной функцией Гаусса.

На рисунке 3.7 представлены области состояний, для которых необходимо применять насыщенные управления. Данные области помогут ускорить онлайн процедуру, так как теперь не нужно будет вычислять даже с помощью нейросетевого регулятора обратную связь для данных состояний, нужно будет только применять соответствующее насыщенное управление и не контролировать выход аппроксимированного управления за границы допустимых значений.



(a) Аппроксимированная область применения $u(t) = 1$ для задачи (3.5).

(b) Аппроксимированная область применения $u(t) = -1$ для задачи (3.5).

Рис. 3.7: Аппроксимированные области с насыщенным управлением.

3.5 Сравнение методов сэмплирования точек для построения аппроксимирования обратной связи

В базовой реализации используется равномерная сетка для построения множества точек для аппроксимирования обратной связи. Однако для равномерной сетки нужно вычислить много точек, но не все они одинаково важны для построения обратной связи. Поэтому предлагается проверить сэмплирование точки стохастически и стохастически с увеличением плотности точек вокруг точки начала координат, так как вокруг точки начала координат, достаточно маленькие управления и аппроксимационный закон в этой зоне может вести не совсем хорошо. На равномерной сетке было взято 3000 точек, поэтому исследовать другие способы сэмплирования на этом же количестве точек.

На рисунке 3.8 представлены сэмплированные точки при стохастическом выборе 3000 точек для области (3.6) внутри области притяжения.

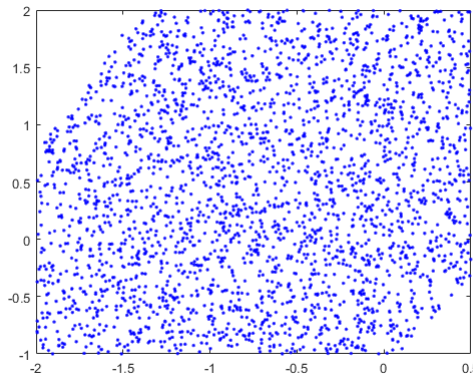


Рис. 3.8: Сэмплированные точки стохастически

На рисунке 3.9 представлены сэмплированные точки при стохастическом выборе 2600 точек для области (3.6) и 400 точек сэмплированных из области округ начала координат $X_{zero} = \{(x_1, x_2) : \|x_1\| \leq 0.2, \|x_2\| \leq 0.2\}$ внутри области притяжения.

Ошибка аппроксимации для этих двух способов для выбранного количества нейронов на скрытом слое $n = 25$ представлена в таблице 3.2.

Как видно из таблицы 3.2 при способе сэмплирования стохастически с увеличенной плотностью вокруг точки начала координат мы получаем выше точность, чем при равномерной сетке и просто стохастическом способе.

Так как нам необходимо получить точность $5 \cdot 10^{-3}$, то можно уменьшить количество генерируемых точек. В таблице 3.3 представлены результаты аппроксимации для разного количества точек. Для получения заданной

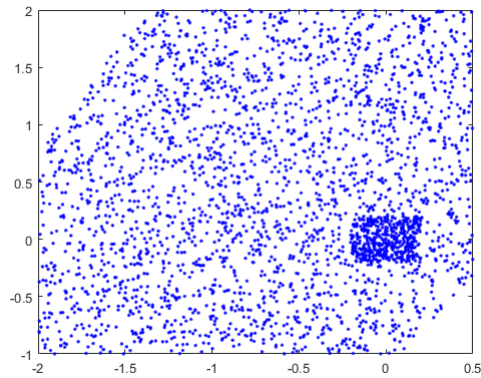


Рис. 3.9: Сэмплированные точки стохастически с увеличенной плотностью около нуля

Таблица 3.2: Таблица зависимости точности аппроксимации от типа сэмплирования

Тип сэмплирования	Ошибка аппроксимации
Равномерная сетка	0.0051
Стохастически сэмплированные точки	0.0062
Стохастически сэмплированные точки и увеличенная плотность точек около начала координат	0.0039

точности, мы можем генерировать 2500 точек, а это значит можно сократить количество точек на 17%.

Таблица 3.3: Таблица зависимости точности аппроксимации от количества точек при сэмплировании стохастически и увеличенной плотности точек вокруг начала координат

Количество точек	Ошибка аппроксимации
3000	0.0039
2800	0.0044
2600	0.0047
2500	0.0049

3.6 Сравнение результатов производительности

Результаты производительности будут сравниваться для онлайн процедуры и построения и обучения оффлайн частей для решения задачи (3.5).

В таблице 3.4 представлены значения времени в среднем для проведения одного шага алгоритма МРС онлайн. Для вычисления среднего значения времени было проведено 1000 процедур вычисления первого шага процедуры МРС из различных начальных точек при помощи вышепредставленных методов.

Таблица 3.4: Таблица зависимости времени проведения одного шага МРС от использованного метода

Использованный метод	Время работы(мс)
Стандартная процедура МРС	1240
Нейросетевой регулятор	55
Нейросетевой регулятор с областями насыщенных управлений	52

Как видно из таблицы последний метод незначительно выиграл у просто нейросетевого регулятора, однако по сравнению со стандартной процедурой было уменьшено время на один шаг МРС очень значительно, и составило уменьшение времени в 23 раза.

В предыдущем пункте было рассмотрено проведение обучения на меньшем количестве точек за счет сэмплирования стохастически и увеличения плотности сэмплированных точек вокруг точки начала координат. Данный метод уменьшил время для обучения на 17%.

Однако стоит отметить, что несмотря на улучшения в уменьшении времени проведения одного шага теряется оптимальность траектории. Оптимальной траекторией считается в данном случае, траектория вычисленная при помощи стандартной процедуры МРС. Было подсчитано, что расхождения траекторий для 1000 запусков из различных точек относительно оптимальных траекторий составили не более 16%. Однако эта потеря несущественна, так как главное, чтобы с помощью этого контроллера мы могли стабилизировать нашу систему и не выходили за рамки ограничений. Выигрыш по времени выполнения помогает быстро управлять системой и может быть использован в ряде производственных задач.

3.7 Использование метода для задачи стабилизации обратного маятника

Рассмотрим задачу стабилизации обратного маятника. Система линейна и представляется в таком виде:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u, \quad (3.9)$$

где параметры системы задаются таким образом:

$$M = 0.5, \quad m = 0.2, \quad b = 0.1, \quad I = 0.006, \quad g = 9.8, \quad l = 0.3,$$

$$q = (M + m) \cdot (I + m \cdot l^2) - (m \cdot l)^2.$$

Ограничения накладываются только на угол $|\phi| \leq \frac{\pi}{3}$ и на управление $|u| \leq 200$.

Необходимо определить терминальные элементы для формулирования задачи в рамках квазибесконечного МРС. Линейная обратная связь в терминальной области представляет собой:

$$u_{loc}(t) = Kx(t) = [10.0000, 15.8365, -79.1980, -18.5048]x(t).$$

Терминальная функция — квадратичная функция вида $x(t)^T Px(t)$, где

$$P = \begin{bmatrix} 1.5979 & 0.7505 & -1.8801 & -0.3226 \\ 0.7505 & 0.9811 & -2.6173 & -0.4276 \\ -1.8801 & -2.6173 & 9.2988 & 1.2221 \\ -0.3226 & -0.4276 & 1.2221 & 0.2119 \end{bmatrix}.$$

Будем использовать также как и в предыдущей задаче точность аппроксимации $\eta = 5 \cdot 10^{-3}$.

Для построения датасета и дальнейшего анализа будем рассматривать

следующую сетку значений состояний:

$$\begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} \leq \begin{bmatrix} 2 \\ 2 \\ \frac{\pi}{3} \\ \frac{\pi}{3} \end{bmatrix} \quad (3.10)$$

Используя равномерную сетку значений с шагом 0.1 мы получим 705600 точек. Используя стохастическое сэмплирование, создадим датасет, состоящий из 585000 точек. 50000 точек будем сэмплировать из множества

$$\{i \in 0, 1, 2, 3 \mid |x(t, i)| \leq 0.2\}$$

, это множество будет окрестностью нуля в которой мы хотим увеличить плотность точек для более точной аппроксимации. Оставшиеся точки будем сэмплировать из 3.10.

На этих двух датасетах мы должны обучить нейронную сеть для аппроксимации закона управления с точностью $\eta = 5 \cdot 10^{-3}$. Для достижения этой точности нам понадобилось 40 нейронов на скрытом слое как для равномерной сетке, так и для стохастического сэмплирования, в таблице (3.5) показаны значения ошибки в зависимости от количества нейронов на скрытом слое для двух сэмплированных датасетов.

Таблица 3.5: Таблица зависимости точности аппроксимации от количества нейронов на скрытом слое

Количество нейронов	Ошибка аппроксимации (равномерная сетка)	Ошибка аппроксимации (стохастическое сэмплирование)
5	1.5362	1.2145
10	0.2873	0.2712
15	0.1267	0.0934
20	0.0544	0.0511
25	0.0219	0.013
30	0.0081	0.0078
40	0.0049	0.0047

Область притяжения имеет вид, представленный на рисунке 3.10. Насыщенных управлений в данном датасете нет, поэтому области насыщенных управлений для этой задачи строить не будем.

Результат построенного нейросетевого регулятора представлен на рисун-

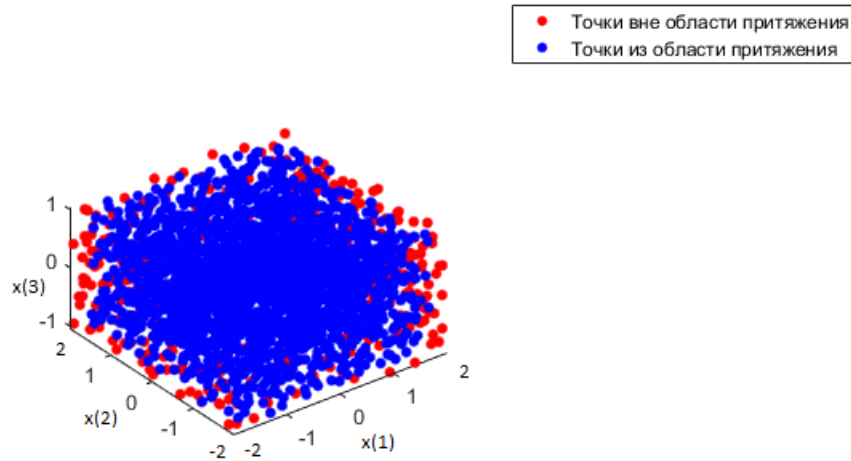
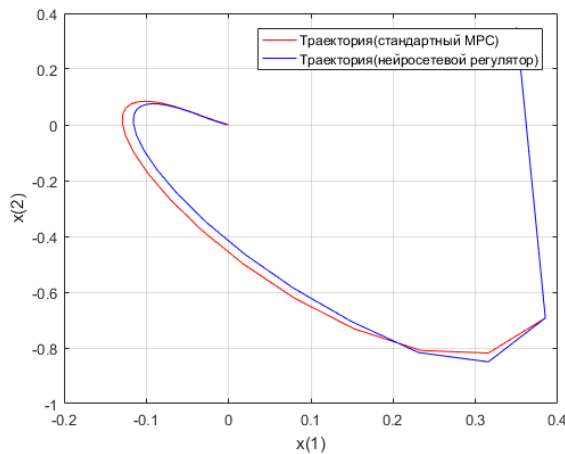
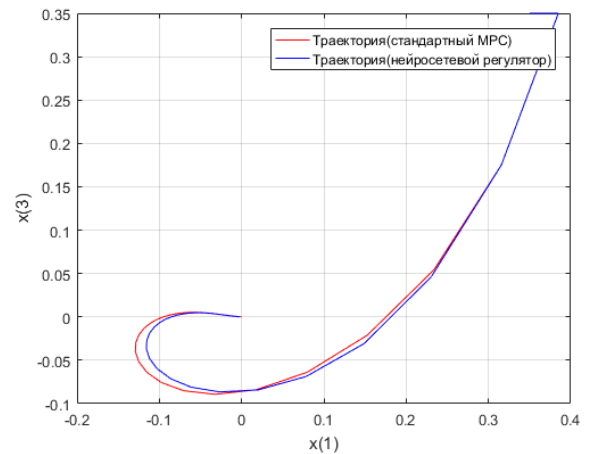


Рис. 3.10: Область притяжения для задачи (3.9)

ке 3.11. На рисунке показывается траектория МРС и нейросетевого регулятора, начиная из точки $x(t) = [0.35, 0.35, 0.35, 0.35]$.



(а) Траектория для обратного маятника на срезе для координат x_1 и x_2



(б) Траектория для обратного маятника на срезе для координат x_1 и x_3

Рис. 3.11: Срезы траектория для стабилизации обратного маятника (3.9)

В таблице 3.4 представлены значения времени в среднем для проведения одного шага алгоритма МРС онлайн. Для вычисления среднего значения времени было проведено 1000 процедур вычисления первого шага процедуры МРС из различных начальных точек при помощи вышепредставленных методов. При использовании стандартной процедуры МРС среднее значение времени вычисления одного равняется 1633 мс, в то время как применение нейросетевого регулятора сокращает время выполнения одного шага до 78 мс.

ЗАКЛЮЧЕНИЕ

На данный момент были изучены материалы по теме методов управления с прогнозирующей моделью. Были исследованы вопросы устойчивости и робастности замкнутой системы, описаны основные подходы для обеспечения устойчивости замкнутых систем, определены алгоритмы для нахождения терминальных параметров и базовый алгоритм МРС.

Метод управления с прогнозирующей моделью (МРС) - одна из популярных современных технологий теории управления, основанная на решении в реальном времени задач оптимального управления с конечным горизонтом, аппроксимирующих решение задачи с бесконечным временным промежутком (например, задачи оптимальной стабилизации). Эта модель представляет собой семейство регуляторов, которое позволяет явно использовать модель для получения управляющего сигнала. Основными причинами популярности МРС при решении прикладных задач являются применимость схемы управления к нелинейным системам и возможность учета ограничений на управления и траектории, а также способность работать без экспертного вмешательства в течение длительного времени.

С другой стороны, эти же факторы могут стать причиной нереализуемости алгоритма, например, в случае быстрых процессов, для которых решение нелинейной задачи оптимального управления не может быть получено регулятором за короткий период квантования. Один из подходов к решению указанной проблемы подразумевает перенос некоторых вычислений оффлайн. В частности, в настоящей работе для реализации функции МРС-регулятора предлагается применить искусственные нейронные сети.

Были проведены исследования по решению указанной проблемы путем переноса некоторых вычислений оффлайн. Предоставлены результаты базовой реализации алгоритма. В частности, для реализации функции МРС-регулятора были применены искусственные нейронные сети. Были проведены эксперименты для построения области притяжения с помощью SVM. Проанализированы различные техники сэмплирования точек для аппроксимации обратной связи. В дальнейшем будут добавлены результаты экспериментов для явного МРС и построения области притяжения с помощью обучения с подкреплением.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Grune L., Pannek J. Nonlinear model predictive control. – Springer London, 2011.
- 2 Rawlings, J.B. Model Predictive Control: Theory and Design / J.B. Rawlings, D.Q. Mayne. – Madison: Nob Hill Publishing, 2009. – 576 p.
- 3 Badgwell T.A., Qin S.J. (2015) Model-Predictive Control in Practice. In: Baillieul J., Samad T. (eds) Encyclopedia of Systems and Control. Springer, London
- 4 H.Chen, F.Allgower A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability / H. Chen, F. Allgower // Automatica. – 1998. – Vol. 34, no. 10. – P. 1205-1217.
- 5 Fernando A.C.C. Fontes A General Framework to Design Stabilizing Nonlinear Model Predictive Controllers / F. A.C.C. Fontes // Systems & Control letters. – 2000. P. 1-13.
- 6 Bemporad A. et al. The explicit linear quadratic regulator for constrained systems //Automatica. - 2002. - T. 38. - N^o . 1. - C. 3-20.
- 7 Domahidi A. et al. Learning a feasible and stabilizing explicit model predictive control law by robust optimization //Proceedings of the IEEE Conference on Decision & Control. - 2011. - N^o. EPFL-CONF-169723.
- 8 Chakrabarty A. et al. Support Vector Machine Informed Explicit Nonlinear Model Predictive Control Using Low-Discrepancy Sequences //IEEE Trans. Automat. Contr. - 2017. - T. 62. - N^o. 1. - C. 135-148.
- 9 Johansen T. A. Approximate explicit receding horizon control of constrained nonlinear systems //Automatica. - 2004. - T.40. - N^o. 2. - C. 293-300.
- 10 Parisini T., Zoppoli R. A receding-horizon regulator for nonlinear systems and a neural approximation //Automatica. - 1995. -T. 31. - N^o.10, - C. 1443-1451.
- 11 Parisini T., Sanguineti M., Zoppoli R. Nonlinear stabilization by receding-horizon neural regulators //International Journal of Control. - 1998. - T. 70. - N^o 3. - C. 341-362.
- 12 Akesson B. M., Toivonen H. T. A neural network model predictive controller //Journal of Process Control. - 2006. - T.16. - N^o. 9. - C. 937-946.
- 13 Pin G. et al. Approximate model predictive control laws for constrained

nonlinear discrete-time systems: analysis and offline design //International Journal of Control. - 2013. - T.86. - N^o.5. - C. 804-820.

14 Aswani A. et al. Provably safe and robust learning-based model predictive control //Automatica. - 2013. - T. 49. - N^o.5. - C. 1216-1226.

15 Rosolia U., Borrelli F. Learning model predictive control for iterative tasks. a data-driven control framework //IEEE Transactions on Automatic Control. - 2018. - T. 63. -N^o.7.

16 Goodfellow I. et al. Deep learning.. - Cambridge : MIT press, 2016. - T.1.

17 CasADi: a software framework for nonlinear optimization and optimal control / J.A.E. Andersson et al. // Mathematical Programming Computation. - 2018. - P. 1-36.

18 Diehl M., Ferreau H. J., Haverbeke N. Efficient numerical methods for nonlinear MPC and moving horizon estimation // Nonlinear model predictive control. - Springer, Berlin, Heidelberg, 2009. - P. 391-417.

19 Hertneck M. et al. Learning an approximate model predictive controller with guarantees //IEEE Control Systems Letters. - 2018. - T. 2. - N^o. 3. - C. 543-548.