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

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

Научный руководитель
Наталия Михайловна Дмитрук
канд. физ.-мат. наук, доцент

Допущена к защите

«____» _____ 2019 г.
Зав. кафедрой дискретной математики и алгоритмики
_____ В.М. Котов
доктор физ.-мат. наук, профессор

Минск 2019

ОГЛАВЛЕНИЕ

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Магистерская диссертация, 61 с., 28 рис., 7 табл., 25 источников

ОПТИМАЛЬНОЕ УПРАВЛЕНИЕ, УПРАВЛЕНИЕ С ПРОГНОЗИРУЮЩЕЙ МОДЕЛЬЮ, НЕЙРОННАЯ СЕТЬ, НЕЙРОСЕТЕВОЙ РЕГУЛЯТОР, ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ

Объектом исследования магистерской диссертации являются задачи стабилизации нелинейных динамических систем, на управляющие воздействия и траектории которых наложены жесткие ограничения, и связанные с рассматриваемой задачей схемы управления по прогнозирующей модели, реализованные на основе нейронных сетей и методов обучения с подкреплением.

Целью работы является построение нейросетевых регуляторов, гарантирующих асимптотическую устойчивость нелинейной системы, регуляторов на основе методов обучения с подкреплением.

Основной результат представлен алгоритмами управления на основе нейронных сетей, позволяющими (по результатам экспериментов) повысить производительность схемы управления с прогнозирующей моделью в 20 раз.

Работа состоит из трех глав. В первой главе определяются основные понятия и результаты в области управления с прогнозирующей моделью. Во второй главе рассматриваются методы оффлайн дизайна обратных связей и их аппроксимаций в системах с прогнозирующей моделью. В третьей главе представлены результаты построения нейросетевого регулятора для нелинейных систем с ограничениями, результаты построения регулятора с помощью методов обучения с подкреплением для систем с неявной динамикой, а также проведено сравнение предлагаемых подходов с классическими алгоритмами, основанными на решении задач оптимального управления онлайн.

Новизна представленной работы состоит в построении нейросетевых регуляторов совместно с построением областей притяжения и областей насыщенных управлений, использовании результатов теории робастного управления для гарантированной стабилизации нелинейной системы, исследовании влияния различных техник сэмплирования точек для построения обучающих выборок, что помогает сократить время обучения без потерь точности.

Областями применения разработанных алгоритмов являются прикладные задачи, решаемые в рамках теории управления с прогнозирующей моделью и возникающие в робототехнике, химической промышленности, транспортных системах, системах управления беспилотными аппаратами и др.

АГУЛЬНАЯ ХАРАКТАРЫСТЫКА РАБОТЫ

Магістарская дысертацыя, 61 с., 28 мал. , 7 табл., 25 крыніц

АПТЫМАЛЬНАЕ КІРАВАННЕ, КІРАВАННЕ ПА ПРАГНАЗУЮЧАЙ МАДЭЛІ, НЕЙРАСЕТКАВЫ РЭГУЛЯТАР, НАВУЧАННЕ З ПАДМАЦАВАННЕМ

Аб'ектам даследавання магістарскай дысертацыі з'яўляюцца заданні стабілізацыі нелінейных дынамічных сістэм, на кіроўныя ўплывы і траекторыі якіх накладзены цвёрдыя абмежаванні, і злучаныя з разгляданым заданнем схемы кіравання па прагназавальнай мадэлі, рэалізаваныя на грунце нейронавых сетак і метадаў навучання з падмацаваннем.

Мэтай працы з'яўляецца пабудова нейрасеткавых рэгулятараў, што гарантуюць асімптатычную ўстойлівасць нелінейнай сістэмы, рэгулятараў на грунце метадаў навучання з падмацаваннем.

Асноўны вынік пададзены алгарытмамі кіравання на грунце нейронавых сетак, што дазваляюць (па выніках эксперыментаў) павялічыць прадукцыйнасць схемы кіравання з прагназавальнай мадэллю ў 20 раз.

Праца складаецца з трох частак. У першай частцы вызначаюцца асноўныя паняткі і вынікі ў вобласці кіравання з прагназавальнай мадэллю. У другой частцы разглядаюцца метады афлайн дызайну зваротных сувязяў і іх апраксімацый у сістэмах з прагназавальнай мадэллю. У трэцяй частцы пададзены вынікі пабудовы нейрасеткавага рэгулятара для нелінейных сістэм з абмежаваннямі, вынікі пабудовы рэгулятара з дапамогай метадаў навучання з падмацаваннем для сістэм з невыяўнай дынамікай, а таксама праведзена параўнанне прапанаваных падыходаў з класічнымі алгарытмамі, заснаванымі на развязку заданняў аптымальнага кіравання анлайн.

Навізна пададзенай працы складаецца ў пабудове нейрасеткавых рэгулятараў супольна з пабудовай абласцей прыцягнення і абласцей насычаных кіраванняў, выкарыстанні вынікаў тэорыі робаснага кіравання для гарантаванай стабілізацыі нелінейнай сістэмы, даследаванні ўплыву розных тэхнік сэмплавання кропак для пабудовы навучальных выбарак, што дапамагае скараціць час навучання без страт дакладнасці.

Абласцямі ўжывання распрацаваных алгарытмаў з'яўляюцца ўжытковыя заданні, што развязваюцца ў рамках тэорыі кіравання з прагназавальнай мадэллю і ўзнікаюць у робататэхніцы, хімічнай прамысловасці, транспартавых сістэмах, сістэмах кіравання беспілотнымі апаратамі і інш.

GENERAL CHARACTERISTICS OF THE WORK

Master thesis, 61 p., 28 figures, 7 tables, 25 sources

OPTIMAL CONTROL, MODEL PREDICTIVE CONTROL, NEURAL NETWORK REGULATOR, REINFORCEMENT LEARNING

The object of the master's thesis research is the stabilization problems of nonlinear dynamic systems, which are subject to rigid constraints on the control actions and trajectories, and associated with the task under consideration of the control scheme according to the predictive model, implemented on the basis of neural networks and reinforcement learning methods.

The aim of the work is to build neural network regulators that guarantee the asymptotic stability of a non-linear system, regulators based on reinforced learning methods.

The main result is presented by control algorithms based on neural networks, which allow (according to the results of experiments) to increase the performance of the control circuit with the predictive model by 20 times.

The work consists of three chapters. The first chapter defines the basic concepts and results in the field of model predictive control. The second chapter discusses methods for offline design of feedbacks and their approximations in systems with a predictive model. The third chapter presents the results of building a neural network controller for nonlinear systems with constraints, the results of building a controller using training methods with reinforcement for systems with implicit dynamics, and also compared the proposed approaches with classical algorithms based on solving optimal control problems online.

The novelty of the presented work consists in building neural network regulators together with building attraction areas and saturated control areas, using the results of the robust control theory to guarantee stabilization of a nonlinear system, studying the influence of various points sampling techniques to build training samples, which helps reduce training time without loss of accuracy.

The areas of application of the developed algorithms are applied problems solved within the framework of the control theory with a predictive model and occurring in robotics, the chemical industry, transport systems, autonomous vehicle control systems, etc.

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

ОУ — оптимальное управление

MPC — Model Predictive Control, управление с прогнозирующей моделью

SVM — Support Vector Machines, метод опорных векторов

MSE — Mean square error, средняя квадратичная ошибка

RL — Reinforcement learning, метод обучения с подкреплением

MDP — Markov Decision Process, процесс принятия решений Маркова

LBMPC — Learning based model predictive control, управление с прогнозирующей моделью, полученное при применении машинного обучения

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

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

$||x||$ — 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

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

I — единичная матрица

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

1_p — вектор из единиц размерности p

ВВЕДЕНИЕ

Метод управления с прогнозирующей моделью (MPC) [1, 2] — одна из популярных современных технологий теории управления, основанная на решении в реальном времени задач оптимального управления с конечным горизонтом, аппроксимирующих решение задачи управления с бесконечным временным промежутком (например, задачи стабилизации).

Основными причинами популярности MPC при решении прикладных задач (обзор практических применений можно найти в работе [3]) 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 MPC, например, в случае быстрых процессов, для которых решение нелинейной задачи оптимального управления не может быть получено регулятором достаточно быстро, в реальном времени, т.е. в темпе обновления информации о состояниях управляемой системы.

Один из подходов к решению указанной проблемы подразумевает перенос некоторых вычислений оффлайн. В частности, в настоящей работе для реализации функции MPC-регулятора предлагается применить искусственные нейронные сети и методы обучения с подкреплением. Нейронные сети будут использоваться для аппроксимации закона управления, а обучение с подкреплением — для задач с неявным видом динамики системы.

В настоящей магистерской диссертации будут исследованы вопросы устойчивости и робастности нелинейных динамических систем, замкнутых обратной связью, построенной MPC-регулятором, и проведено сравнение с оптимальным MPC-регулятором для некоторых прикладных задач. В частности, в главе 1 изложены основные принципы MPC, рассмотрена задача стабилизации, описан базовый алгоритм MPC. Там же исследованы вопросы асимптотической устойчивости замкнутой системы вместе с алгоритмами построения терминального множества и терминальной функции для обеспечения устойчивости этой системы. В основной части работы (главы 2 и 3) разрабатываются методы построения нейросетевых регуляторов и областей притяжения для различных нелинейных систем, проводится 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 (MPC) [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 MPC на примере задачи стабилизации управляемых движений динамической системы.

1.1 Основные принципы MPC

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

Популярность MPC 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 момент времени, промежуток, для которого прогнозируется поведение системы, постоянно смещается ("скользит"), в силу чего MPC также иногда называется управлением со скользящим горизонтом — Receding Horizon Control (RHC).

1.2 MPC для решения задач стабилизации

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

Как было отмечено выше, основная идея MPC 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 MPC носят название номинальных (nominal MPC scheme).

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

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

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 $f : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^n$ 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

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

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) MPC — построить обратную связь $u(x)$, при которой замкнутая система

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

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 $X \in \mathbb{R}^n$ — 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 $\lim\limits_{t \to +\infty} ||x(t)||_A = 0 \; \forall x_0 \in X$.

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.

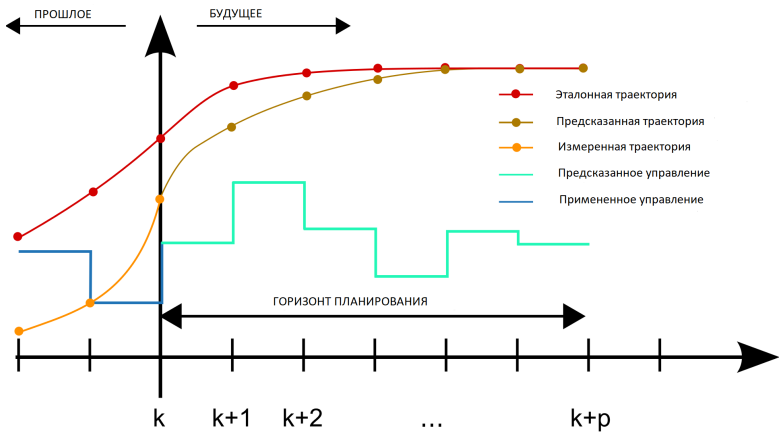


Рис. 1.1: Схема MPC

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 будем обозначать $x(k|t)$, $k \in \mathbb{I}_{[0,N-1]}$. Они изменяются

согласно уравнению

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

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 MPC. 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 $l : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}$ 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 $\mathcal{K}_\infty$, что выполняется

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

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

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

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:
$u^0(\cdot|t) = \{u^0(0|t), \ldots, u^0(N-1|t)\}$ — оптимальное (программное) управление задачи $\mathcal{P}(t)$;
$x^0(\cdot|t) = \{x^0(0|t), \ldots, x^0(N|t)\}$ — соответствующая траектория;
X_N — множество всех состояний $x \in X$, для которых существует решение задачи (1.5) с $x(t) = x$.

Базовый алгоритм MPC 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

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

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

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

После 20 лет исследований теория MPC приобрела свои нынешние черты. Согласно [5], все схемы номинального MPC 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 MPC зависят от выбора терминальных элементов V_f и X_f. Например, самые первые результаты исследований по MPC [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

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

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 $k_f : X_f \to U$, 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 MPC, 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 MPC на квази-бесконечном горизонте.

MPC 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 случит 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

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

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

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

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

терминальная стоимость — квадратичная функция вида:

$$V_f(x) = \|x\|_P^2 = x^T P x; \tag{1.10}$$

терминальное множество — эллипсоид вида:

$$X_f = \{x \in \mathbb{R}^n |\ x^T P x \le \alpha\}, \tag{1.11}$$

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

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

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

$$(A_K + kI)^T P + P(A_K + kI) = -(Q + K^T RK) \tag{1.13}$$

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

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

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 $Kx \in U\ \forall x \in X_f^{\alpha_1}$;

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

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

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 MPC 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 [19, 20] 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 могут оказаться неэффективными или слишком медленными. В связи с этим недостатком, далее в магистерской диссертации предлагается вынести некоторые вычисления из классического алгоритма MPC оффлайн, что позволит повысить производительность систем управления с прогнозирующей моделью.

ГЛАВА 2

ИСПОЛЬЗОВАНИЕ МЕТОДОВ ОФФЛАЙН ДИЗАЙНА В СИСТЕМАХ УПРАВЛЕНИЯ С ПРОГНОЗИРУЮЩЕЙ МОДЕЛЬЮ

Метод MPC опирается на решение задач ОУ в режиме реального времени, однако несмотря на то, что существуют методы эффективного решения этих задач [19, 20] и развитие вычислительной техники позволяет быстро решать достаточно сложные задачи, для многих классов нелинейных систем существующие методы могут могут оказаться неэффективными или слишком медленными. Для преодоления этих недостатков в магистерской диссертации предлагается вынести некоторые вычисления из классического алгоритма MPC оффлайн. В данной главе будут рассмотрены основные существующие подходы для задач MPC, а также теория методов машинного обучения, которая будет использоваться в экспериментах в рамках магистерской диссертации.

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 [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.

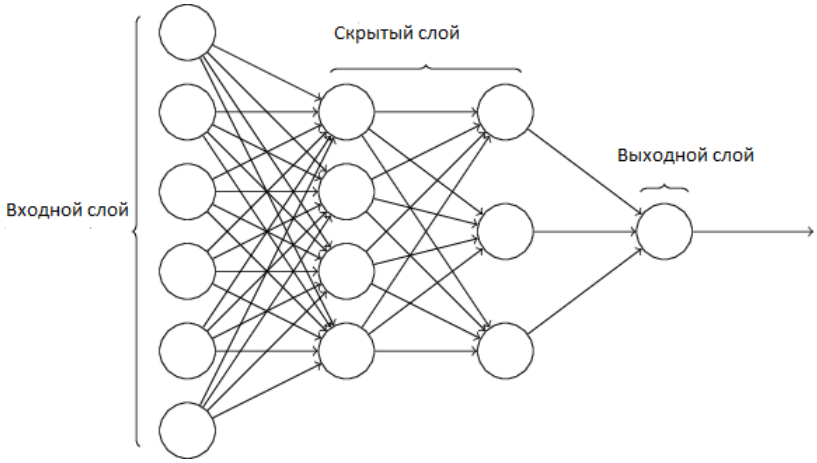


Рис. 2.1: Нейронная сеть со скрытыми слоями.

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.

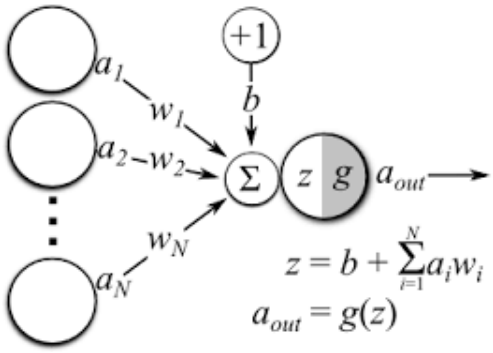


Рис. 2.2: Строение нейрона.

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. Веса — это адаптивные коэффициенты в сети, которые определяют интенсивность входного сигнала, зарегистрированного искусственным нейроном. 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 нелинейная активационная функция ReLU $g(x) = max(0, x)$,которая рекомендованна для использования с большинством нейронных сетей прямого распространения. Поскольку ReLU почти линейна, она сохраняет 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 $tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

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 весы ввода до тех пор, пока не будет достигнута приемлемая точность сети. Когда процесс тренировки заканчивается, то уже в онлайн процессах используются эти натренированные параметры и веса.

Алгоритм обучения - алгорим обратного распространения ошибки, в котором используется стохастический градиентный спуск. Для задачи регресии чаще всего в качестве функции потерь используется MSE (2.1):

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

Согласно универсальной теореме аппроксимации — нейронная сеть с одним скрытым слоем может аппроксимировать любую непрерывную функцию многих переменных с любой точностью [17]. Необходимо определить достаточное количество нейронов для достижения этой точности. 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) [18] - это область машинного обучения, связанная с тем, как агенты, взаимодейтсвующие со средой, должны предпринимать действия в этой среде, чтобы максимизировать некоторое понятие кумулятивной награды. В литературе по исследованиям и контролю операций [18] обучение с подкреплением называется приближенным динамическим программированием или нейродинамическим программированием. Интерес к обучению с подкреплением возник 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. [18]

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 [18], 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 $\pi : S \times A \to [0, 1]$:

$$\pi(a|s) = P(a_t = a|s_t = s). \tag{2.2}$$

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

Функция стоимости $V_\pi(s)$ определяется как ожидаемая прибыль, 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.

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

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

$$R = \sum_{t=0}^{\infty} \gamma^t r_t, \tag{2.4}$$

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

С помощью функции стоимости нужно найти стратегию, которая максимизирует прибыль. На основании теории MDP, стратегия называется оптимальной, если она достигает наилучшей ожидаемой прибыли из любого начального состояния.

Чтобы определить оптимальность формальным образом, определим прибыль стратегии π

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

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,

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

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

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

ляется формулой

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

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 принимаем оптимальное действие, выбирая его 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. [18]

2.3 Методы дизайна оффлайн регуляторов в системах управления с прогнозирующей моделью

Существует несколько подходов использования методов обучения в MPC системах:

- Явный MPC
- Аппроксимация закона управления
- Использование обучаемой модели для аппроксимации динамики прогнозирующей модели
- Итерационный подход для построения терминального региона и функции из предыдущих итераций

2.3.1 Явный MPC

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

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

при условиях

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

$$x(t|t) = x(t),\ t \le k \le t+N-1,$$
$$C_x x(k|t) \le d_x,\ t \le k \le t+N-1,$$
$$C_u u(k|t) \le d_u,\ t \le k \le t+N-1,$$
$$C^f x(t+N|t) \le d^f,$$

с квадратичной функцией стоимости этапа и квадратичной терминальной функцией

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

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

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

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

Перепишем функцию стоимости

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

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

Перепишем динамику системы:

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

Тогда

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

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

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

С помощью (2.13) перепишем (2.11) :

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

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

$$G = diag(C_x, ..., C_x)\Gamma,\ W = [d_x, .., d_x]^T,\ E = diag(C_x, .., C_x)\Omega.$$

Тогда, задача (2.8) примет вид

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

при условии

$$GU \leq W + Ex(t). \tag{2.15}$$

Применим замену переменных вида

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

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

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

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. [6]

Явный MPC решает задачу для всех состояний, таким образом все пространство состояний делится на области, где в каждой области для состояния есть явная функция управления.

Алгоритм нахождения явных функций управления [6]:

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 критическую область по активным ограничениям и вычислить функцию управления для этой области
5. Перейти к новому x_0

Главный недостаток этого метода состоит в том, что количество областей 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 опредилить управление для него.

2.3.2 Аппроксимация закона управления

Существует несколько подходов к получению аппроксимативного решения для оптимизации MPC. Для линейных систем в [7] алгоритм обучения представлен дополнительными ограничениями для обеспечения стабильности и наклвдывает ограничения на ошибку аппроксимации. Одним из подходов к аппроксимации MPC является выпуклое многопараметрическое нелинейное

программирование [9], где вычисляется субоптимальная аппроксимация закона управления MPC. Другой подход - аппроксимировать MPC с помощью методов машинного обучения. Это делают нейронные сети в [10], [11], [12]. Эти методы не гарантируют устойчивость или удовлетворение ограничениям для аппроксимационного MPC, что особенно важно, если рассматривать жесткие ограничения на состояния. В [8] используется метод опорных векторов (SVM) для аппроксимации MPC. Устойчивость и удовлетворение ограничений могут быть гарантированы для произвольных ошибок малого приближения, основанных на присущих свойствам устойчивости. В [13] аппроксимируется MPC с липшицевым сужающим ограничением, что обеспечивает устойчивость при неисчезающих ошибках аппроксимации. Ошибка аппроксимации, выведенная в [8], [13], обычно не достижима для практического применения.

В [11] находят аппроксимирующий закон управления

$$u^0(t) = \gamma^0(x(t)) \in U,$$

где $u^0(t)$ - первый вектор последовательности управления, которая минимизирует стоимость

$$J(x(t), u(t)) = \sum_{k=t}^{t+N-1} l(x(k), u(k)) + a\|x(t+N)\|_P^2, \ t \geq 0.$$

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 слоем с v_j нейронами на скрытом слое для $j = 1..m$ сети и линейными активационными функциями.

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

$$\min_{w_j} \max_{x_t \in X} |\gamma_j^0(x(t)) - \gamma_j^{(v_j)}(x(t), w_j)| \leq \frac{\epsilon}{\sqrt{m}}, \ j = 1, ..., m. \tag{2.17}$$

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 $\gamma_j^0(x_t)$ 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 работе [13] уже вводятся некоторые гарантии устойчивости метода

на основании предположения о непрерывности по Липшицу правой части динамической системы. Система представляется следующим образом:

$$x(t+1) = f(x(t), u(t), \xi_t),\ t \geq 0,\ x_0 = \tilde{x},$$

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

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

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 и предполагают,что существует функция K класса, такая что

$$|\hat{f}(x(t), u(t)) - \hat{f}(x(t), u'(t))| \leq \eta_u(|u(t) - u'(t)|)\ \forall x(t) \in X,\ \forall u(t) \in U,\ , u'(t) \in U.$$

Также вводится предположение об ограниченоти возмущения и то, что верно $|d_t| \leq \mu(|\xi_t|)\ t \geq 0$, а также $d_t \in D = B^m(\bar{(}d)),\ \bar{d} \in \mathbb{R}_{\geq 0} < \infty$. 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 $q_t = \xi_t - x(t)$ и управления $v_t = k^*(x_t) - k^*(\xi_t)$. Тогда система уже переписывается в таком виде:

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

И устойчивость системы доказывается, если верно следующее,что погрешности аппроксимации состояния и управления совместено с возмущением ограничены изначальным возмущением системы $\bar{d_q} + \bar{d_v} + \bar{d_w} \leq \bar{d}$. В данной статье рассматривались аппроксимации закона управления с помощью нейронной сети,которая показала довольно хорошие результаты.

В работе с использованием результатов [21] исследуются условия, при которых, несмотря на ошибки аппроксимации, гарантируется выполнение ограничений и асимптотическая устойчивость замкнутой системы. На основе этого источника будет продолжаться исследование аппроксимации закона управления с помощью нейронной сети и результаты будут продемонстрированы в главе 3.

2.3.3 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, который назвается LBMPC. LBMPC изучает динамику системы по предоставленным точкам,также имеется возможность обновления динамики системы с помощью новых измерений, обеспечивает 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:

$$x(t+1) = Ax(t) + Bu(t) + g(x(t), u(t))$$

где $g(x(t), u(t))$ 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:

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

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 MPC с обученой динамикой на известных вычисленных точках, и может адаптироваться к новым полученным точкам, чтобы улучшать точность обученной динамики системы.

Задача формулируется в таком виде:

$$V_n(x()) = \min_{c,\theta} \phi_n(\theta, \tilde{x}(t), ..., \tilde{x}(t+N), \tilde{u}(t), ..., \tilde{u}(t+N-1))$$

при условиях

$$\tilde{x}(t) = x_t,\ \bar{x}(t) = x_t,$$

$$\tilde{x}(t+i+1) = A\tilde{x}(t+i) + B\tilde{u}(t+i) + O_n(\tilde{x}(t+i), \tilde{u}(t+i)),$$

$$\bar{x}(t+i+1) = A\bar{x}(t+i) + B\tilde{u}(t+i),$$

$$\tilde{u}(t+i) = K\bar{x}(t+i) + c_{n+i},$$

$$\bar{x}(t+i+1) \in X \ominus R_i,\ \tilde{u}(t+i) \in U \ominus KR_i,$$

$$(\bar{x}(t+N) \in \Omega \ominus (R_N \times \{0\}),$$

где Ω - допустимое инвариантное робастное множество таких точек, что любая траектория системы с начальным условием, выбранным из этого множества и с управлением $u(t)$, остается в множестве для любой последовательности ограниченного возмущения, удовлетворяя ограничениям на состояние и управление.

Моделируется точка устойчивого состояния $\bar{x}_s = \Lambda\theta$ и управления $\bar{u}_s = \Psi\theta$, где $\theta \in \mathbb{R}^m$ и $\Lambda \in \mathbb{R}^{n\times m}$, $\Psi \in \mathbb{R}^{m\times m}$ - параметры моделирования, параметры будут описывать точку равновесия для системы, если $A + BK$ устойчива по Шуру при управлении

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

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

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

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

2.3.4 Итерационный подход

Существует итерационный подход для построения MPC регулятора [15]. Данный метод используется для повторяющихся задач, где эталонная траектория неизвестна. Например, системы для гоночных и раллийных машин, где среда и динамика сложны и не совсем известны. Регулятор имеет справочную информацию и способен улучшать свою эффективность, изучая предыдущие итерации. Для обеспечения рекурсивной выполнимости и неубывающей эффективности 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.

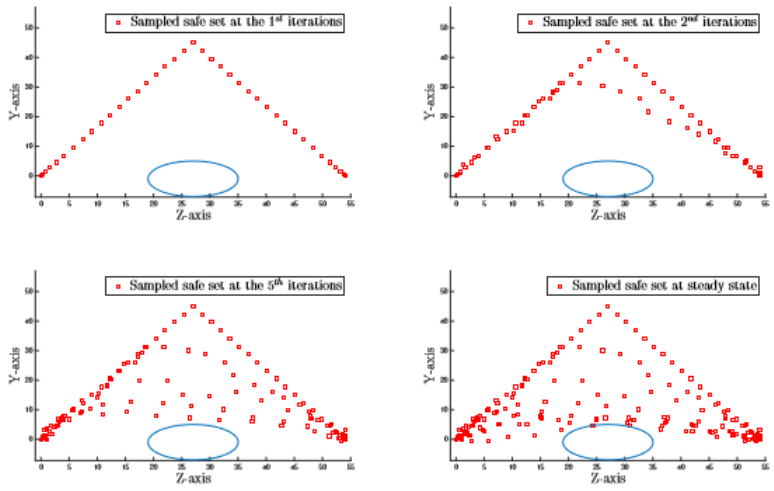


Рис. 2.3: Построение сэмплированного безопасного множества.

Управляемое на N шагов вперед множество по отношению к S

$$K_j(S) = Pre(K_{j-1}(S)) \cap X,\ K_0(S) = S, j \in \{1, ..., N\}, \tag{2.18}$$

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

Сэмплированное безопасное множество на итерации j

$$SS^j = \{\cup_{i \in M^j} \cup_{t=0}^{\infty} x^i(t)\},$$

где $M^j = \{k \in [0,j] : \lim_{t\to\infty} x^k(t) = x_F\}$.

SS^j — это множество всех траекторий на итерации i для $i \in M^j$. Как выглядит это множество показано на рисунке 2.3.

Стоимость на сэмплированном безопасном множестве вводится таким образом, чтобы мы штрафовали те состояния, которые не находятся в нашем построенном безопасном множестве на предыдущих итерациях

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

где $\forall\ x(t) \in SS_j,\ Q_j(x) = J^{i^*}_{t^*\to\infty}(x(t)) = \sum_{k=t^*}^{\infty} l(x^{i^*}(k), u^{i^*}(k))$.

$$J_{t\to t+N}(x_t^j) = \min_{u_{t|t},\ldots,u_{t+N-1|t}} [\sum_{k=t}^{t+N-1} l(x(k|t), u(k|t)) + Q^{j-1}(x(t+N|t))],$$

при условии

$$x(k+1|t) = f(x(k|t), u(k|t)) \ \forall k \in [t, .., t+N-1],$$

$$x(k|t) \in X, \ u(k|t) \in U, \ \forall k \in [t, .., t+N-1],$$

$$x(t+N|t) \in SS^{j-1},$$

$$x(t|t) = x^j(t).$$

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.

2.4 Выводы

Существует ряд подходов для вынесения вычислений оффлайн для систем с прогнозирующей моделью. Каждый метод используется для своего типа задач. Для линейных задач с небольшим количеством областей подходит явный MPC. Для задач с нелинейной динамикой будет более эффективным метод аппроксимации закона управления. Для задач с неизвестной динамикой, представленной в виде набора точек, лучше использовать аппроксимация динамики системы. Для задач с неизвестной динамикой или изменяющей средой полходят итеративные методы управления MPC.

В данной магистерской диссертации мы хотели вынести вычисления оффлайн для нелинейных систем,поэтому более эффективное направление — использование аппроксимации закона управления. Для этого будут использоваться нейронные сети, как универсальный аппроксиматор непрерывных функций. А также будет применяться обучение с подкреплением для систем с неизвестной динамикой.

ГЛАВА 3

ПОСТРОЕНИЕ НЕЙРОСЕТЕВОГО РЕГУЛЯТОРА

В настоящей главе демонстрируются потенциальные возможности вынесения вычисления оффлайн для нелинейных систем с прогнозирующей моделью. Как было показано в предыдущей главе, наиболее перспективным направлением для рассматриваемого типа задач является использование аппроксимации закона управления. Для этого будут использоваться нейронные сети, как универсальный аппроксиматор непрерывных функций.

В этой главе описаны эксперименты и результаты использования нейросетевых регуляторов, а также методы построения областей насыщения управления с помощью метода опорных векторов. Проводится сравнение производительности для стандартного MPC-регулятора, решающего онлайн прогнозирующие задачи оптимального управления, и нейросетевого регулятора, использующего нейронную сеть, которая заменит процедуру решения прогнозирующей задачи. Исследуется применение обучения с подкреплением к моделям с неизвестной динамикой.

3.1 Описание иллюстративного примера

Рассмотрим нелинейную дискретную динамическую систему

$$x(t+1) = f(x(t), u(t)), \tag{3.1}$$

где $x(t) \in \mathbb{R}^n$ — вектор состояний, $u(t) \in \mathbb{R}^m$ — вектор управления. Предполагается, что $f(0,0) = 0$.

Для системы (3.1) рассматриваются ограничения вида

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

поэтому в каждый момент времени требуется выполнение включения

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

Будем минимизировать функцию стоимости

$$\sum_{k=0}^{N-1} l(x,u) + V_f(X(N)), \tag{3.4}$$

содержащую сумму стоимостей этапов и терминальную стоимость, которая, напомним, строится специальным образом, обеспечивая вместе с терминальным ограничением асимптотическую устойчивость замкнутой системы.

Предлагаемые подходы будут демонстрироваться на примере из [10], только будут изменены ограничения на управление, чтобы лучше продемонстрировать насыщенные управления.

Нелинейная система в дискретном виде представляет собой

$$\dot{x}_1(t) = -x_2(t) + 0.5(1 + x_1(t))u(t), \tag{3.5}$$

$$\dot{x}_2(t) = x_1(t) + 0.5(1 - 4x_2(t))u(t),$$

с ограничениями на фазовые и управляющие переменные

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

$$u \in U = \{u \in \mathbb{R} |\ |u| \leq 1\}, \tag{3.7}$$

Шаг дискретизации $\delta = 0.1$. Данная система имеет стабилизируемое первое приближение в начале координат, которое имеет вид

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

Для системы (3.5) выбрана квадратичная функция стоимости этапа:

$$l(x,u) = ||u||^2 + 0.1||x||^2,$$

а терминальная стоимость и терминальное множество найдены методом квази-бесконечного MPC из главы 1:

$$V_f(x(t)) = x(t)^T P x(t),$$

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

где $\alpha = 0.2129$,

$$P = \begin{bmatrix} 48.3043 & -2.53 \\ -2.53 & 70.8191 \end{bmatrix}$$

Линейная локальная обратная связь для терминального множества имеет вид

$$k_f(x(t)) = Kx(t) = 0.3799x_1(t) - 0.4201x_2(t).$$

Тогда прогнозирующая задача оптимального управления

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

при условиях

$$x_1(k+1|t) = x_1(k|t) + \delta(-x_2(k|t) + 0.5(1 + x_1(k|t))u(k|t)),$$

$$x_2(k+1|t) = x_2(k|t) + \delta(0.5(1 - 4x_2(k|t))u(k|t)),$$

$$-2 \le x_1(k|t) \le 0.5,\ -1 \le x_2(k|t) \le 2,$$

$$|u(k|t)| \le 1,$$

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

Таким образом, для системы (3.5) определены все параметры MPC-регулятора.

Задача (3.8) будет решаться онлайн при стандартной реализации алгоритма MPC, или использоваться для подготовки подготовки датасета для нейрорегулятора.

3.2 Подход к решению

Алгоритм MPC для стабилизации положения равновесия $x = 0$ системы (3.5) состоит в следующем: Для каждого $t = 0, 1, \ldots$
1. измерить состояние $x(t) \in X$ системы (3.5);
2. решить задачу (3.8) с начальным условием $x(0|t) = x(t)$, получить $u^0(\cdot|t)$;
3. подать на вход системы (3.5) управляющее воздействие $u_{MPC}(x(t)) := u^0(0|t)$.

Поскольку для построенных функций l, V_f и множества X_f выполнены требования теоремы 1.1, замкнутая система

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

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 (3.8) с $x(0|0) = x_0$ имеет решение [2].

Как отмечено выше, шаг 2 приведенного алгоритма может оказаться достаточно трудоемким. В связи с этим в настоящей работе предлагается вместо онлайн решения прогнозирующей задачи (3.8) 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 $x(t)$.

Далее будем рассматривать систему (3.5) в контексте робастного MPC [**?**], поскольку ошибка аппроксимации $\bar{u}_{MPC} - u_{MPC}$ может рассматриваться как возмущение специального вида в динамике замкнутой системы, которая теперь имеет вид:

$$x(t+1) = f(x(t), u_{MPC}(x(t)) + d(t)),$$

где $d(t) \in D$, $D = \{d \in \mathbb{R}^m | \ ||d||_\infty \leq \eta\}$ — множество допустимых ошибок реализации управления, $\eta > 0$.

В работе с использованием результатов [21] исследуются условия, при которых, несмотря на ошибки аппроксимации, гарантируется выполнение ограничений и асимптотическая устойчивость замкнутой системы. При необходимые для выполнения этих свойств предположения следующие:

1. Локальная инкрементируемая стабилизуемость: Существует такая обратная связь $k : X \times X \times U \to \mathbb{R}^m$, δ-функция Ляпунова $V_\delta : X \times X \times U \to \mathbb{R}_{\geq 0}$, непрерывная по первому аргументу и удовлетворяющая $V_\delta(x, x, v) = 0 \ \forall x \in X, \ \forall v \in U$, 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 всех $(x, z, v) \in X \times X \times U$, $(z+, v+) \in X \times U$ с $V_\delta(x, z, v) \leq \delta_{loc}$:

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

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

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

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

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

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

3. Радиус множества D удовлетворяет неравенству $\eta \leq \frac{1}{\lambda}\sqrt{\frac{\delta_{loc}}{c_{\delta,u}}}$.

Для робастного выполнения ограничений используются сужение ограничений в виде увеличивающейся трубки: 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

$$\epsilon_k = \epsilon \left(\frac{1 - \sqrt{\rho}^k}{1 - \sqrt{\rho}} \right),\ k = 0, 1, ..., N;$$

$$\epsilon = \eta\lambda\sqrt{\frac{c_{\delta,u}}{c_{\delta,l}}} \max\{||H||_\infty, ||L||_\infty k_{\max}\}.$$

В настоящей работе для построения аппроксимации $\bar{u}_{MPC}(x(t))$ применяются метод опорных векторов (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.

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

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

Далее в численных экспериментах 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

Для численных экспериментов в рассматриваемом примере стабилизации системы (3.5) был выбран горизонт планирования $N = 30$. В качестве начального состояния выбрана точка $x_0 = (0.4; 1.5)$.

После реализации стандартной процедуры MPC для системы (3.5), описанной в главе 1, получена траектория замкнутой системы, представленная на рисунке 3.1. Соответствующая реализация MPC управления $u_{MPC}(t)$ Изображена на рисунке 3.2.

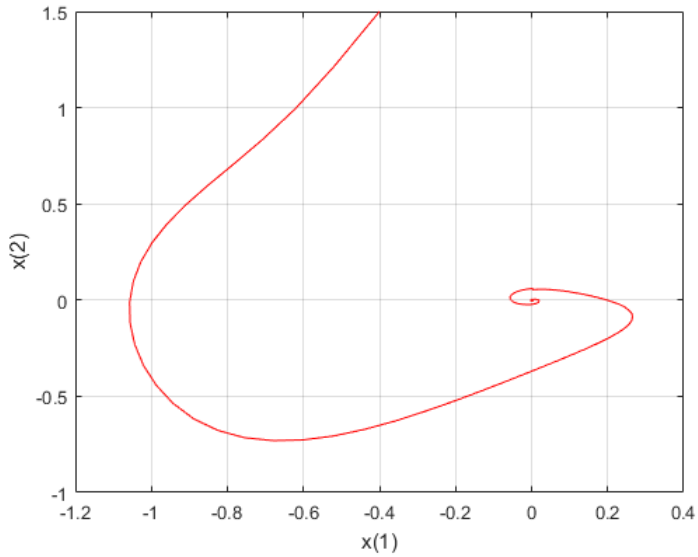


Рис. 3.1: Траектория замкнутой системы в примере (3.5) при применении стандартной процедуры MPC

Базовая реализация нейрорегулятора включает создание нейронной сети для аппроксимации обратной связи $\bar{u}_{MPC}(x)$ для системы (3.5).

Для построения аппроксимации обратной связи необходимо создать обучающую выборку для нейронной сети. Для этого в первом варианте реализации алгоритма нейросетевого управления построим равномерную сетку на множестве (3.6) допустимых состояний. Шаг равнометрой сетки выбран равным 0.05.

В узлах равномерной сетки x^i, $i \in I$, вычислим решение прогнозирующей задачи (3.8) с начальным условием $x(0|0) = x^i$. Запомним значение $u^i :=$

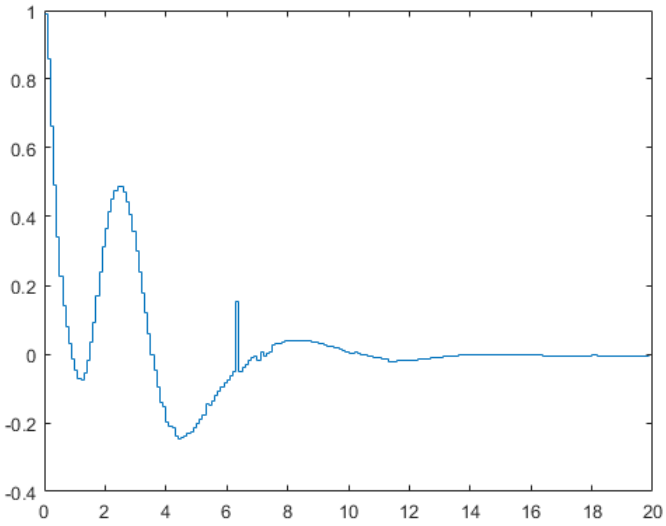


Рис. 3.2: Реализация обратной связи для системы (3.5) при применении стандартной процедуры MPC

$u^0(0|0, x^i)$.

На рисунке 3.3 показано векторное поле сетки с шагом 0.1, а не 0.05, для более хорошей демоснтрации, в котором видно как происходит переход из состояния $x(0|0) = x^i$ в состояние $x(1|0) = f(x^i, u^i)$, используя полученные значения управления (обратной связи). На рисунке видно, что для большинства точек следующее состояние находится в пределах допустимых значений. Однако, существуют также точки, где следующее состояние выходит за пределы области обучения, и в этом случаем можно получить неустойчивое решение.

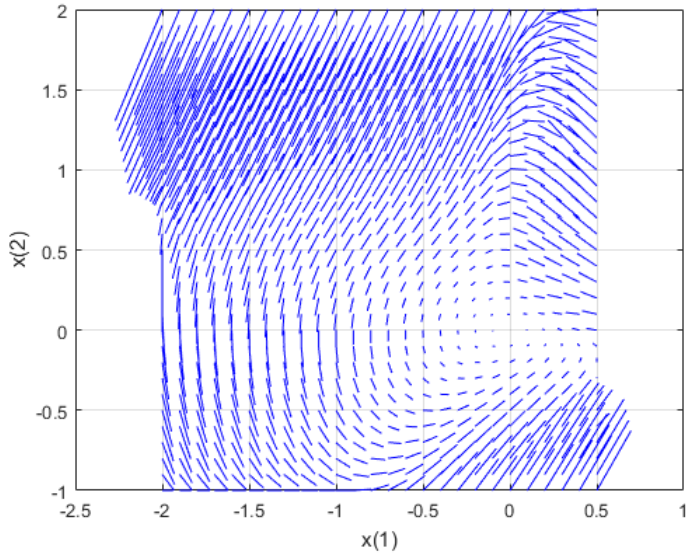


Рис. 3.3: Векторное поле для сетки с шагом 0.1.

Продолжая анализ, на рисунке 3.4 представлен график u^i в зависимости

от x^i. Можно увидеть множество одинаковых управлений для состояний.

Важно отметить, что сетку значений состояния в окрестности начала координат нужно сделать достаточно детальной, чтобы не получалось зацикливание регулятора, т.е. не происходил выбор одного и того же управления около нуля,которое не способствует приближению регулятора к началу координат.

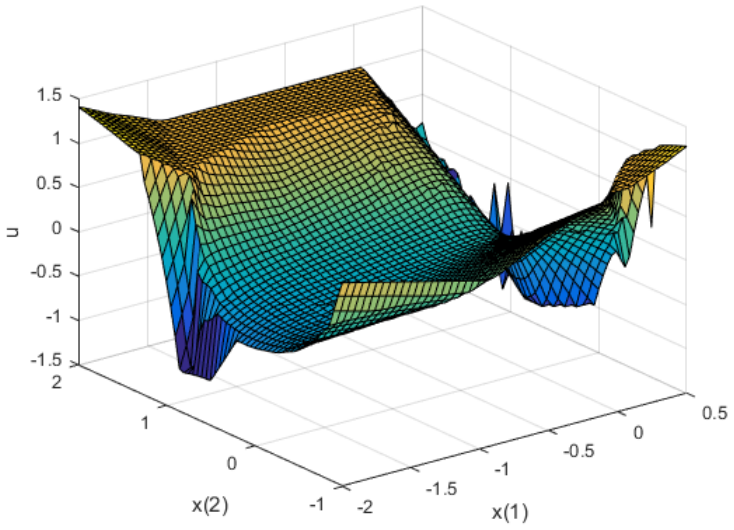


Рис. 3.4: Управление для равномерной сетки значений состояний для обучения нейронной сети

Перейдем к построению нейронной сети. Для выбора количества нейронов на скрытом слое примем подход работы [11], где использовался итеративный процесс для поиска оптимального количества нейронов. Алгоритм выбирает некоторое начальное значение количества нейронов, строит сеть, которая аппроксимирует функцию, а затем для каждого узла сетки вычисляет значение приближенной обратной связи и обратной связи, полученной из стандартной процедуры MPC, пока не будет достигнута необходимая точность

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

где, напомним, m — размерность вектора управления.

В таблице 3.1 представлена зависимость точности от количества нейронов на скрытом слое. Для достижения точности $5 \cdot 10^{-3}$ понадобилось 25 нейронов.

На рисунке 3.5 представлены траектории замкнутой системы (3.5), полученные при помощи стандартной процедуры MPC (красная кривая) и при помощи построенного нейросетевого регулятора (синяя кривая). Как видно

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

из рисунка, получились достаточно близкие трактории, асимптотически стремящиеся к началу координат.

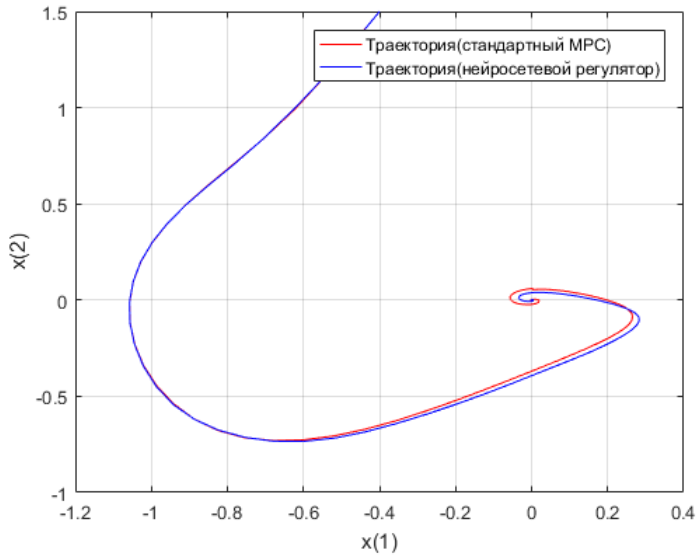


Рис. 3.5: Траектории, полученные при помощи стандартной процедуры MPC и нейросетевого регулятора

Подведем итоги представленной базовой реализации нейрорегулятора. В представленном подходе необходимо сначала вычислить значения управления в каждой точке равномерной сетки на множестве допустимых значений состояний. Затем на полученной выборке, состоящей из пар состояние-значение обратной связи, обучается нейронная сеть, как задача обучения с учителем.

Отметим основные недостатки данного подхода:

- Для хорошей аппроксимации необходима выборка, содержащая достаточно много точек, хотя некоторые области имеют одинаковые значения управления и для них его не надо вычислять заново.

- Обучение и валидация нейронной сети проходят достаточно долго.

Предполагаемые улучшения для этого подхода:

- Уменьшить кол-во генерируемых точек для построения аппроксимации, рассмотреть генерирование точек с помощью стохастического подхода и стохастического с увеличением плотности точек у начала координат.
- Построить X_0 — область притяжения, множество состояний, из которых система достигает терминального множества за N шагов.
- Для ускорения поиска управления, выделить множества точек, для которых применяется насыщенное управление.
- Рассмотреть возможность использования обучения без учителя и обучение с подкреплением для построения множества допустимых значений и закона управления.

3.4 Построение области притяжения и областей насыщения управления с помощью SVM

При построении базового регулятора в разд. 3.3 было замечено, что в выбранной области допустимых значений состояний можно выделить подмножество точек, из которых можно достичь начало координат — область притяжения, и подмножества, в которых замкнутая система неустойчива. Предлагается аппроксимировать область притяжения и только по точкам из нее 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 [22]. Для всех этих точек определяем возможность достижения из данного состояния точки начала координат. Помечаем состояние, из которого достижимо начало координат, с помощью метки 1, а неустойчивые состояния как -1. Для классификации применяется метод опорных векторов (SVM) [23] 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 устойчивых состояний.

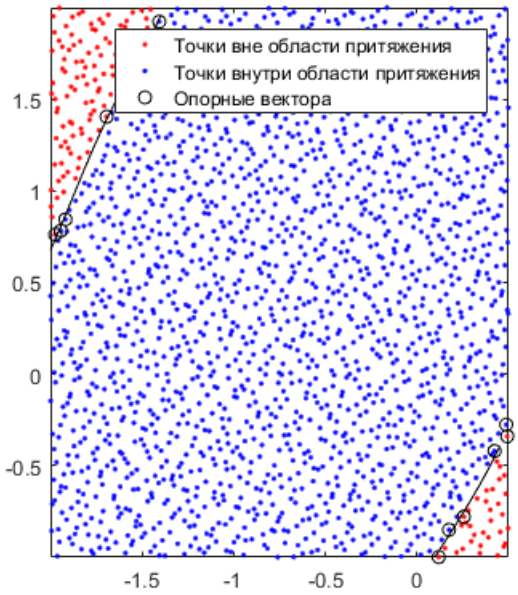


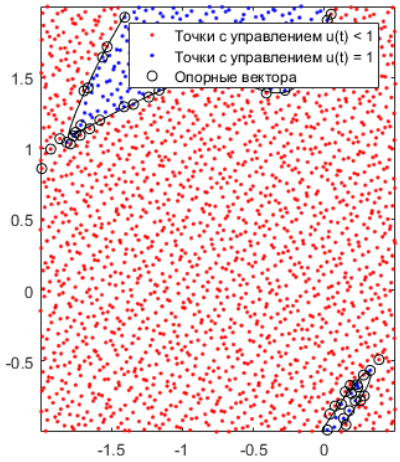
Рис. 3.6: Аппроксимация области притяжения для задачи (3.5)

Для определения областей с насыщенными управлениями также воспользуемся методом SVM.

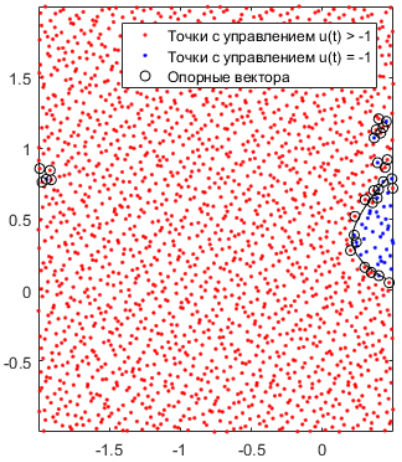
Метод SVM хорошая работает для разграничения двух классов, а в рассматриваемой задачи у будет 3 класса: 1) насыщение на верхней границе, $u = 1$, 2) насыщение на верхней границе, $u = -1$, 3) промежуточные значения управления, $|u| < 1$. В связи с этим создадим две отдельные модели для аппроксимации областей: первую модель для определения области с управлением $u = 1$ и вторую модель для определения области с управлением $u = -1$. Для каждой модели помечаем те состояние, которые соответствует насыщенному управлению, как 1, остальные состояния как -1.

Здесь нужно принять во внимание, что в робастной версии используются сужения ограничений, поэтому насыщение управления будем определять с зазором в ϵ, т.е. пометим все состояния из которых будет управление $u(t) >= 1-\epsilon$ либо $u(t) <= -1+\epsilon$. 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 приближенной обратной связи

В базовой реализации используется равномерная сетка для построения множества точек для приближения 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 в окрестности точки начала координат. Последний подход объясняется тем, что в окрестности точки начала координат значения управления малы, и хорошее приближение крайне важно для сохранения свойства асимптотической.

На равномерной сетке было взято 3000 точек, поэтому будем 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.

На рисунке 3.9 представлены сэмплированные точки при стохастическом выборе 2600 точек для области (3.6) и 400 точек сэмплированных из окрестности 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 в окрестности начала координат, увеличивается точ-

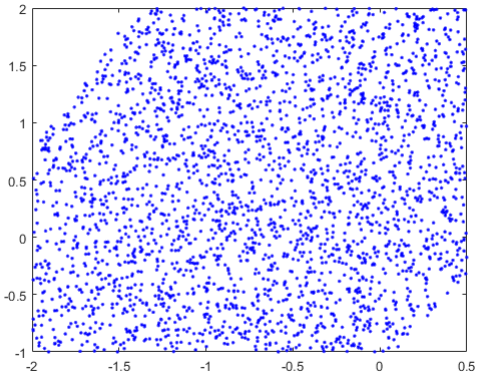


Рис. 3.8: Сэмплированные точки стохастически

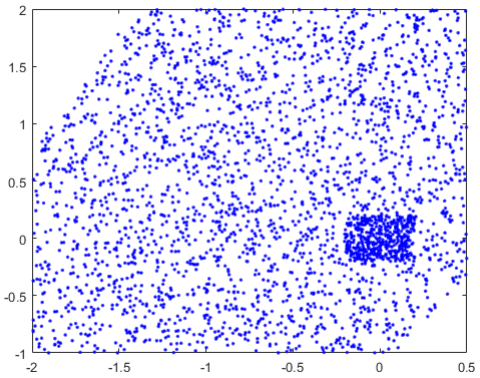


Рис. 3.9: Сэмплированные точки стохастически с увеличенной плотностью около нуля

Таблица 3.2: Зависимость точности аппроксимации от типа сэмплирования

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

ность аппроксимации по сравнению с равномерной сеткой и простом стохастическом способе.

Поскольку для исследуемой задачи достаточно получить точность $5 \cdot 10^{-3}$, то можно уменьшить количество генерируемых точек. В таблице 3.3 представлены результаты аппроксимации для разного количества точек в обучающей выборке. Понятно, что для получения заданной точности, достаточно 2500 точек, а это значит можно сократить количество точек на 17%.

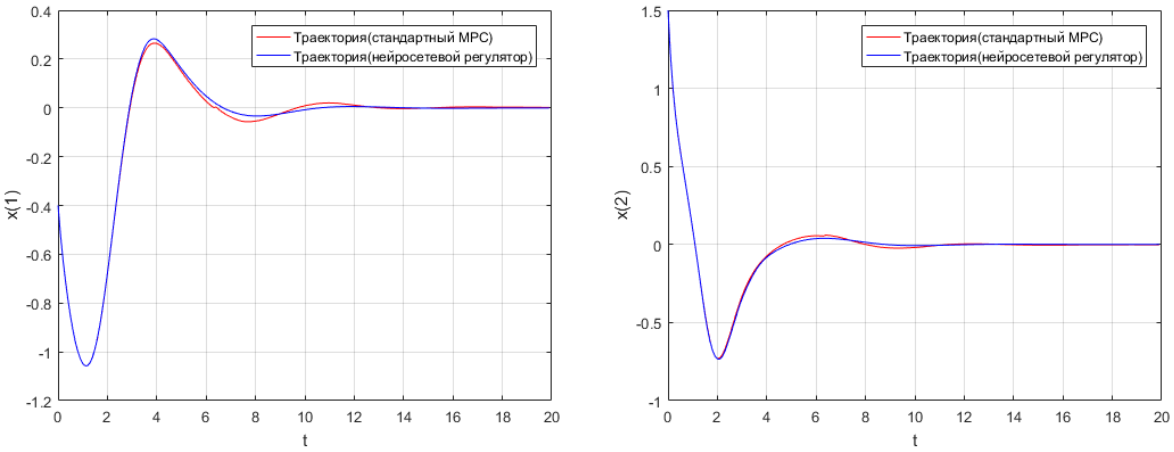
Таблица 3.3: Зависимость точности аппроксимации от количества точек при сэмплировании стохастически и увеличенной плотности точек вокруг начала координат

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

3.6 Сравнение результатов производительности

Сравним результаты производительности при реализации классического алгоритма MPC и при реализации нейросетевого регулятора с обучением оффлайн для решения задачи стабилизации нелинейной системы (3.5).

Траектории отдельно для x_1 и x_2 представлены на рисунках 3.10.



(a) Траектория для x_1 (b) Траектория для x_2

Рис. 3.10: Траектории для x_1 и x_2 для системы (3.8)

Управления, построенные используя стандартный MPC и при помощи нейросетевого регулятора, показаны на рисунке 3.11

В таблице 3.4 представлены значения времени в среднем для проведения одного шага алгоритма MPC онлайн. Для вычисления среднего значения времени была проведена 1000 вычислений первого шага процедуры MPC для различных начальных точек при помощи представленных выше методов.

Как видно из таблицы, последний метод, использующий дополнительно SVM для выделения областей насыщения управления, незначительно выиг-

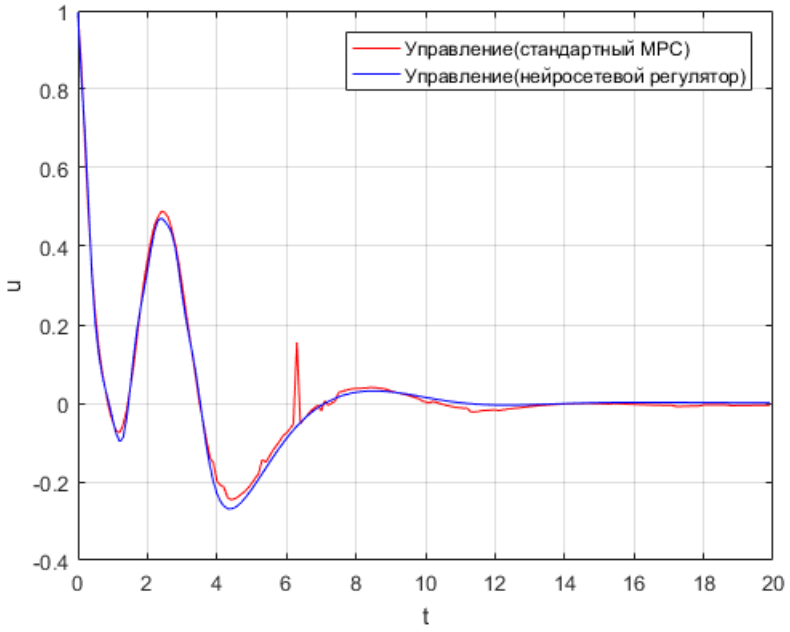


Рис. 3.11: Функция управления для задачи (3.8)

Таблица 3.4: Зависимость времени проведения одного шага MPC от использованного метода

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

рал у нейросетевого регулятора, не различающего граничные и промежуточные значения управления, однако по сравнению со стандартной процедурой MPC время решения уменьшено в 23 раза.

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

Необходимо отметить, что применение нейросетевых регуляторов (также в комбинации с SVM) приводит к траекториям замкнутой системы, которые могут существенно отличаться от "эталонной", где под "эталонной" в данном случае понимается траектория, полученная при помощи стандартной процедуры MPC. Было подсчитано, что расхождения траекторий для 1000 запусков из различных начальных состояний относительно эталонных траекторий

составили не более 16%.

Однако приведенные потери можно считать несущественными в связи с целью управления системой, которая состоит в стабилизации системы и гарантированном выполнении ограничений.

В то же время, существенный выигрыш по времени вычисления $\bar{u}_{MPC}(x(t))$ в сравнении с $u_{MPC}(x(t))$ позволяет применять нейросетевые регуляторы в тех случаях, когда дискретизация непрерывной системы осуществлена с маленьким шагом h. Например, в рассматриваемом примере стандартная процедура MPC вычисляет управление за 1240 мс, а нейросетевой регулятор за 55 мс.

Таким образом, нейросетевые регуляторы предпочтительнее в использовании в ряде производственных задач для систем с быстро меняющейся динамикой.

3.7 Использование нейросетевого регулятора для задачи стабилизации обратного маятника

Рассмотрим задачу стабилизации обратного маятника на подвижном основании (тележке) [24]. Линеаризация уравнений в окрестности верхнего устойчивого положения маятника имеет вид

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

$$+ \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Ml^2} \\ 0 \\ \frac{ml}{I(M+m)+Ml^2} \end{bmatrix} u,$$

где

x — положение тележки на платформе, а ϕ — угол наклона тележки, и параметры выбраны следующим образом:

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

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

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

Определены терминальные элементы для формулировки задачи в рамках квазибесконечного MPC. Линейная обратная связь в терминальном множестве имеет вид:

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

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

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

Точность аппроксимации, как и в предыдущей задаче, выбрана равной $\eta = 5 \cdot 10^{-3}$.

Для построения обучающей выборки и дальнейшего анализа будем рассматривать следующую сетку значений состояний:

$$\left| \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} \right| \leq \begin{bmatrix} 2 \\ 2 \\ \frac{\pi}{3} \\ \frac{\pi}{3} \end{bmatrix} \tag{3.10}$$

Используя равномерную сетку значений с шагом 0.1 получим 705600 точек. Используя стохастическое сэмплирование, создадим выборку, состоящую из 585000 точек, при этом 50000 точек будем сэмплировать из множества

$$\left| \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} \right| \leq \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \end{bmatrix}$$

т.е. окрестности начала координат, в которой необходимо увеличить плотность точек для более точной аппроксимации. Оставшиеся точки будем сэмплировать из 3.10.

На этих двух датасетах обучается нейронная сеть для аппроксимации закона управления с точностью $\eta = 5 \cdot 10^{-3}$. Для достижения указанной точности 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 (рамномерная 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 3.12. Насыщенных управлений в данном датасете нет, поэтому области насыщенных управлений для этой задачи не строятся.

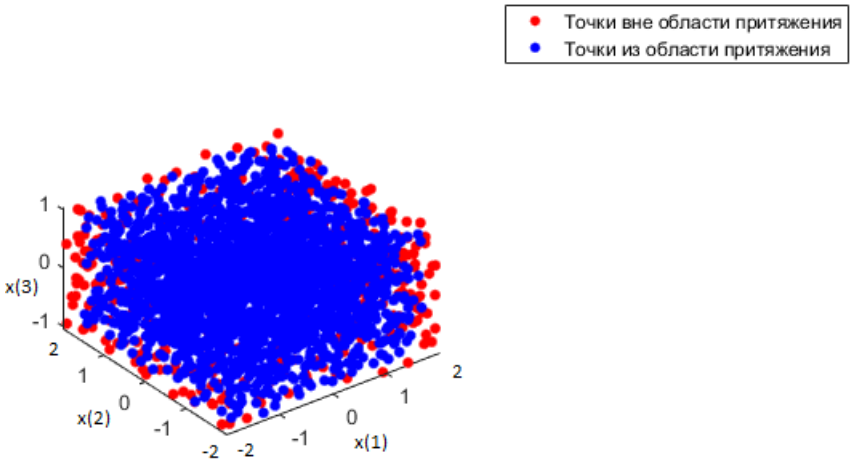
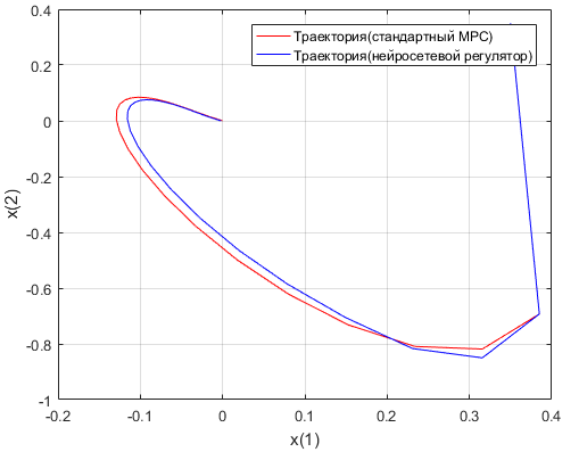


Рис. 3.12: Область притяжения для задачи (3.9)

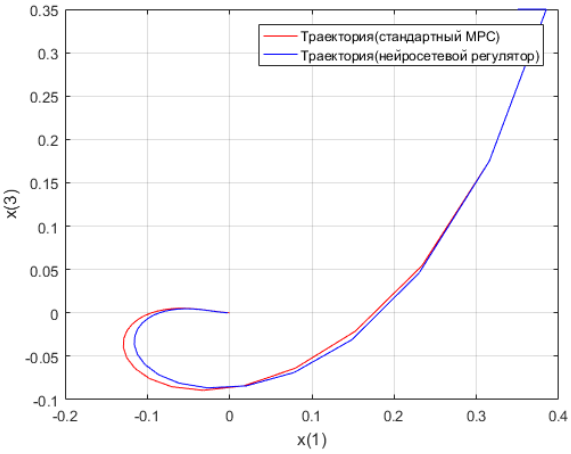
Результаты представлены на рисунке 3.13. На рисунке приведены траектории стандартного MPC-регулятора и нейросетевого регулятора, начиная из точки $x(t) = [0.35, 0.35, 0.35, 0.35]$.

Траектории отдельно для x_1 и x_3 представлены на рисунках 3.14.

Управления, построенные используя стандартный MPC и при помощи нейросетевого регулятора, показаны на рисунке 3.15.

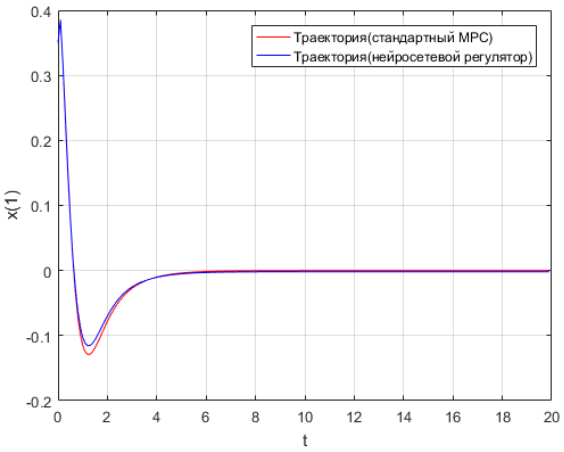


(a) Траектория для обратного маятника на срезе для координат x_1 и x_2

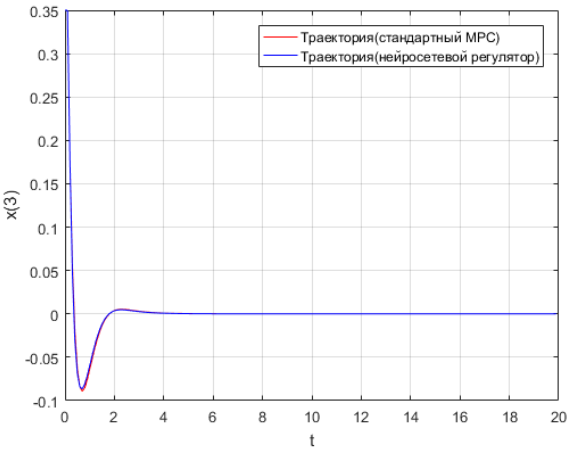


(b) Траектория для обратного маятника на срезе для координат x_1 и x_3

Рис. 3.13: Срезы траектории для стабилизации обратного маятника (3.9)



(a) Траектория x_1



(b) Траектория x_3

Рис. 3.14: Траектории для стабилизации обратного маятника (3.9)

Для вычисления среднего значения времени было проведено 1000 процедур вычисления первого шага процедуры MPC из различных начальных точек при помощи вышепредставленных методов. При использовании стандартной процедуры MPC среднее значение времени вычисления одного равняется 1633 мс, в то время как применение нейросетевого регулятора сокращает время выполнения одного шага до 78 мс.

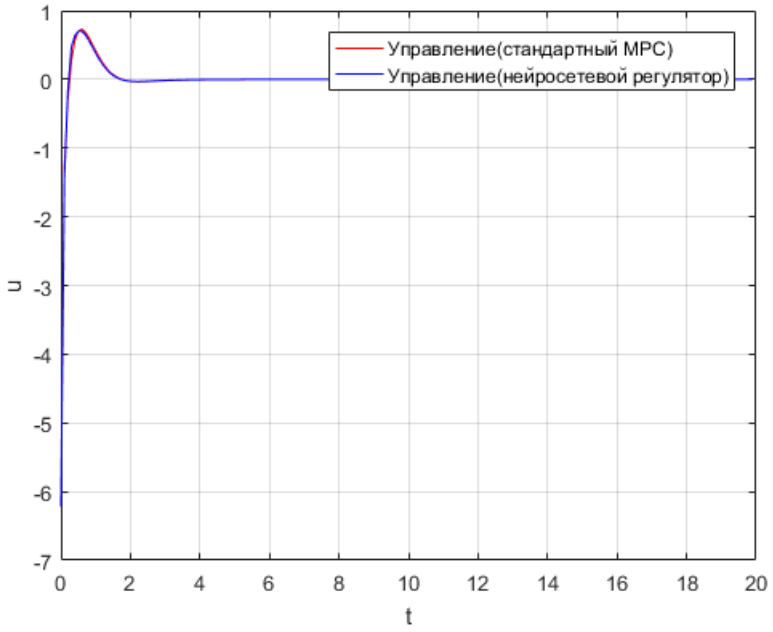


Рис. 3.15: Функция управления для задачи (3.9)

3.8 Применение обучения с подкреплением для систем с неявной динамикой

Рассмотрим задачу перевернутого маятника, расположенного на подвижной платформе, данная конструкция представлена на рис. 3.16.

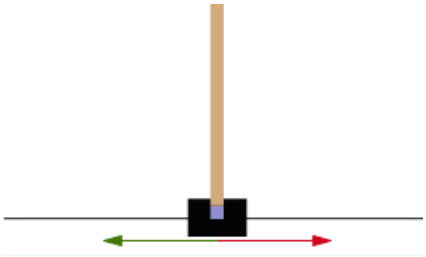


Рис. 3.16: Перевернутый маятник на платформе

Маятник поддерживается в равновесии за счет изменения скорости тележки, движение маятника происходит по подвижной платформе и есть возможность двигаться только влево и вправо. Управление маятником может быть описано марковским процессом. В таких процессах управляющее действие определяется только текущим состоянием объекта управления и не зависит от предшествующих состояний. Данный марковский процесс будет использоваться для обучения с подкреплением. Для эмулирования системы будет использоваться объект CartPole-v0 библиотеки gym.

Для решения задачи необходимо в каждом состоянии s объекта управле-

ния правильно определять действие a, которое применяется к тележке с маятником. Множество доступных действий над маятников: движение влево и движение вправо. Как описывалось в главе 2, задача обучения с подкреплением решается с помощью агента, взаимодействующего с средой, фиксирующей в том числе и состояние объекта, а также среда передает агенту состояние объекта управления и награду за действие агента, переведшее объект в текущее состояние. Агент, получив от среды информацию о состоянии объекта и свою награду, определяет следующее действие, которое он считает наиболее правильным в данный момент времени.

Множество состояний для данной задачи задается следующим образом:

$$X = \{|x_1| \leq 2.4,\ x_2 \in \mathbb{R},\ |x_3| \leq 0.4,\ x_4 \in \mathbb{R}\}$$

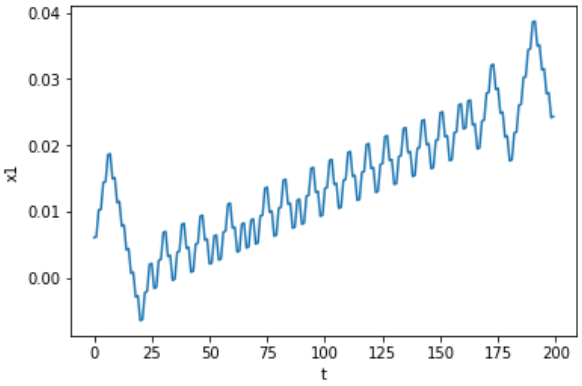
Стартовым состоянием маятника будет равномерное случайное значение в пределах $\pm 0,05$. Награда будет назначаться в размере 1.0 за то, что маятник не будет перевернут и не уйдет за границы множества X. Также ограничим эксперименты количеством эпизодов стабилизации 200 тиков времени. Награда для всего эпизода будет не просто сумма всех полученных наград, а дисконтированная сумма с константой дисконтирования $\gamma = 0.99$. Количество эпизодов для обучение 10000. Количество эпизодов для валидации 1000.

Выбор оптимального управления будет осуществляться с помощью нейронной сети с одним скрытым слоем. На вход будет подаваться вектор состояний, в скрытом слое будет 12 нейронов, а на выходе вероятность выбора действия движения влево и движения вправо. Выбор количества нейронов на скрытом слое основывался на талице (3.6). Во время обучения будет использоваться ϵ-жадная стратегия выбора оптимального действия в конкретный момент времени, поэтому после получения вероятности событий в данный момент времени, будет выбираться действие случайным образом, где дейтсвие будет выбрано случайно с вероятностью, которую выдала нейронная сеть.

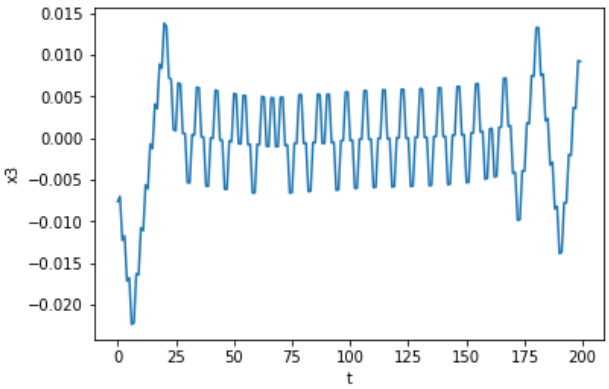
Таблица 3.6: Таблица зависимости процента удачных эпизодов от количества нейронов на скрытом слое

Количество нейронов на скрытом слое	Удачные эпизоды(%)
5	76
8	95
10	98
12	100

Результат обученного регулятора представлены на рисунке 3.17.



(a) Траектория по x_1 (положение тележки)



(b) Траектория по x_3 (угол наклона маятника)

Рис. 3.17: Траектории для стабилизации обратного маятника

Как видно из рисунка, траектория по первой координате смещается с каждым тиком вправо,а значит, что скоро маятник перейдет за установленные границы, но можно отметить, что угол достаточно хорошо стабилизирован.

Обучение с подкреплением обучается за счет наград и пытается максимизировать получаемые награды, так как в данной конфигурации, мы получаем константное значение награды за позицию на подвижной платформе, то управление недостаточно хорошо научилось определять оптимальную политику в данном случае. Нужно добавить изменяющееся количество награды за позицию тележки на платформе.

Исследуем функции награды вида

$$r(\alpha) = \frac{1}{\alpha + |x_1|}, \tag{3.11}$$

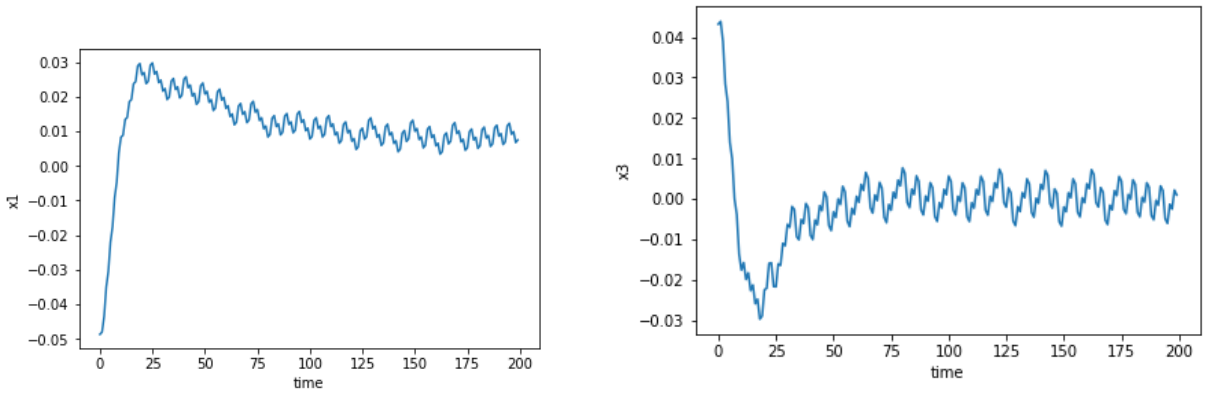
будем проверять следующие значения $\alpha = \{0.01, 0.05, 0.1, 0.5, 1.0, 2.0\}$. Так как мы хотим стабилизировать маятник в пределах этих положений тележки на платформе, то будем проверять удачен ли эпизод, если удалось оставаться в пределах $|x_1| \leq (\alpha + \min(\alpha, 0.1))$. Валидировать эпизоды стабизиации на временном промежутке 400 тиков.

Как видно из результатов таблицы 3.7 стабилизировать с точностью $0.01 - 0.05$ не получилось, слишком малый процент удачных стабилизаций, возможно это связано с тем, что уже происходят слишком малые изменения маятника и поэтому сложно обучиться выбирать правильную стратегию. Начиная с позиции $0.1 - 1.0$ получилось стабилизировать достаточно качественно, но чем ближе стабилизациия к границам множества состояний, тем

Таблица 3.7: Таблица зависимости процента удачных эпизодов от позиции вокруг которой стабилизируем маятник

Позиция	Удачные эпизоды(%)
0.01	36
0.05	45
0.1	90
0.5	96
1.0	99
2.0	85

хуже получается обучить на эту позицию маятник, поэтому на позиции 2.0 получилось количество удачных позиций 81%.



(a) Траектория по координате x_1 (b) Траектория по координате x_3

Рис. 3.18: Траектории для стабилизации обратного маятника с наградой по формуле (3.11) с $\alpha = 0.1$

Результат обучения по формуле (3.11) с $\alpha = 0.1$ представлены на рисунке 3.18. Здесь можно увидеть, что стабилизация прошла не только по углу наклона тележки, но и за счет дифференцированной награды за положение тележки, тележка стабилизировалась около 0.1.

Можно отметить, что обучение с подкреплением может научиться оптимальному управления за счет взаимодействия со средой и не требует знания динамики объекта. Если добавлять необходимые изменения в функции награды, то можно регулировать оптимальное управление. Однако обучение с подкреплением по представленным результатам не может гарантировать высокую точность вычисления и требует дальнейших исследований в этой области.

3.9 Выводы

В данной магистерской диссертации было исследовано применение нейронных сетей для аппроксимации закона управления для нелинейных систем, которые повысили производительность онлайн вычислений в 20 раз. Также были исследованы разные техники сэмплирования, и было показано, что можно сократить количество точек тренировочного набора на 17% при использовании стохастического сэмплирования и повышения плотности точек у начала координат. Были построены области притяжения с помощью SVM для более качественного сэмплирования.

Для систем с неявной динамикой было использовано обучение с подкреплением. Было показана возможность регулировать обучение за счет изменения функций наград. Данный метод не требует знаний о динамике системы, однако не может гарантировать высокую точность вычислений, поэтому нужны дальнейшие исследования в этой области.

ЗАКЛЮЧЕНИЕ

MPC представляет собой семейство регуляторов, которое позволяет явно использовать модель для получения управляющего сигнала, однако данный метод не подходит для 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. MPC предполагает наличие модели динамики лдя построении своей процедуры оптимального управления, поэтому в условиях неизвестной динамики MPC не сможет находить эффективное управления, поэтому необходимо использовать другие методы.

Для построения аппроксимации управления в MPC регуляторе использовались нейронные сети. Для модификации задачи обратного маятника в виде задачи с неизвестной динамики был использован метод обучения с подкреплением, который позволяет определить управление на основе взаимодействия со средой и получения награды за действие в этой среде.

Были проведены исследования по решению указанной проблемы путем переноса некоторых вычислений оффлайн. Предоставлены результаты базовой реализации алгоритма. В частности, для реализации функции MPC-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. Для исследования применения нейросетевого регулятора использовались задачи с двумерными и четырехмерными множествами состояний. Техника стохастического сэмплирования с увеличением плотности вокруг начала координат уменьшила количество точек, необходимых для построения тренировочного набора на 17%, а значит и сократила время на обучения. Среднее время выполнения одного шага MPC для стандартной процедуры составляло 1240 мс, а при использовании нейростевого регулятора составлят 52 мс. Производительность системы с нейросетевым регулятором уменьшила время выполнения одного шага MPC в 20 раз.

В дальнейшем, будут проводиться исследования применения различных архитектур нейронных сетей для обеспечения необходимых параметров MPC регулятора, а также исследования построения регуляторов с неизвестной динамикой.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1 Grune, L. Nonlinear model predictive control / L. Grune, J. Pannek // Springer London. – 2011.

2 Rawlings, J.B. Model Predictive Control: Theory and Design / J.B. Rawlings, D.Q. Mayne // Madison: Nob Hill Publishing. – 2009. – P. 576.

3 Badgwell, T.A. Model-Predictive Control in Practice /T.A. Badgwell, S.J. Qin // Encyclopedia of Systems and Control. Springer, London. – 2015.

4 Chen, H. 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 Fontes, Fernando A.C.C. 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. The explicit linear quadratic regulator for constrained systems / A. Bemporad et al. // Automatica. – 2002. – Vo. 38, No. 1. – P. 3-20.

7 Domahidi, A. Learning a feasible and stabilizing explicit model predictive control law by robust optimization / A. Domahidi et al. // Proceedings of the IEEE Conference on Decision & Control. – 2011. – No. EPFL-CONF-169723.

8 Chakrabarty, A. Support Vector Machine Informed Explicit Nonlinear Model Predictive Control Using Low-Discrepancy Sequences /A. Chakrabarty et al. // IEEE Transactions on Automatic Control. – 2017. – Vol. 62, No. 1. – P. 135-148.

9 Johansen, T. A. Approximate explicit receding horizon control of constrained nonlinear systems // Automatica. – 2004. – Vol.40, No. 2. – P. 293-300.

10 Parisini, T. A receding-horizon regulator for nonlinear systems and a neural approximation / T. Parisini, R. Zoppoli // Automatica. – 1995. – Vol. 31, No.10. – P. 1443-1451.

11 Parisini, T. Nonlinear stabilization by receding-horizon neural regulators /T. Parisini, M. Sanquineti, R. Zoppoli // International Journal of Control. – 1998. – Vol. 70,No. 3. – P. 341-362.

12 Akesson, B. M. A neural network model predictive controller /B.M.

Akesson, H.T. Toivonen // Journal of Process Control. – 2006. – Vol.16, No. 9. – P. 937-946.

13 Pin, G. Approximate model predictive control laws for constrained nonlinear discrete-time systems: analysis and offline design / G. Pin et al. // International Journal of Control. – 2013. – Vol.86, No.5. – P. 804-820.

14 Aswani, A. Provably safe and robust learning-based model predictive control / A. Aswani et al. // Automatica. – 2013. – Vol. 49, No. 5. – P. 1216-1226.

15 Rosolia, U. Learning model predictive control for iterative tasks. a data-driven control framework /U.Rosolia, F. Borrelli // IEEE Transactions on Automatic Control. – 2018. – Vol. 63, No. 7.

16 Goodfellow, I. Deep learning / I.Goodfellow et al. // Cambridge : MIT press. – 2016. – Vol. 1.

17 Csaji, B. C. Approximation with artificial neural networks // Faculty of Sciences, Etvs Lornd University, Hungary. – 2001. – Vol. 24. – P. 48.

18 Саттон, Р. С. Обучение с подкреплением: пер. с англ / Р. С. Саттон, Э.Г. Барто //М.: БИНОМ, Лаборатория знаний. – 2012.

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

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

21 Hertneck, M. Learning an approximate model predictive controller with guarantees / M.Hertneck et al. // IEEE Control Systems Letters. – 2018. – Vol. 2. No. 3. – P. 543-548.

22 Kocis, L. Computational investigations of low-discrepancy sequences / L. Kocis, W.J. Whiten // ACM Transactions on Mathematical Software (TOMS). – 1997. – No. 23. Vol. 2. – P. 266-294.

23 Cristianini, N. An introduction to support vector machines and other kernel-based learning methods.// Cambridge university press. – 2000.

24 Liberzon, D. Switching in systems and control. // Springer Science and Business Media. – 2003.

25 Павловец, М.Е. Применение методов машинного обучения в системах управления по прогнозирующей модели /М.Е. Павловец, Н.М. Дмитрук // XIX Международная научная конференция по дифференциальным уравнениям "Еругинские чтения-2019": материалы международной научной конференции. Могилев, 14-17 мая 2019 г. – С. 127-129