МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра методов оптимального управления

НЕЙРОСЕТЕВЫЕ РЕГУЛЯТОРЫ В СИСТЕМАХ УПРАВЛЕНИЯ С ПРОГНОЗИРУЮЩЕЙ МОДЕЛЬЮ

Отчет №3
о работе в рамках магистерской диссертации
для специальности:
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 методов оптимального управления
канд. физ.-мат. наук, доцент Н.М. Дмитрук

Минск 2019

ОГЛАВЛЕНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ

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

ВВЕДЕНИЕ

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

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

В соответствии с целью диссертации определена структура работы: обзор метода управления с прогнозирующей моделью, его устойчивость и базовый алгоритм, обзор методов обработки больших данных, используемые технологии для выполнение практической части, основные результаты и оценка производительности системы. В первом семестре изучена литература, на основании которой оформлена первая глава диссертации. В частности, изучены основные принципы MPC, рассмотрена задача стабилизации, описан базовый алгоритм MPC. А также исследованы вопросы устойчивости замкнутой системы вместе с алгоритмами построения терминального множества и терминальной функции для обеспечения устойчивости этой системы. Настоящий отчет содержит обзор изученной литературы.

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;
- 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, базовые алгоритмы MPC, его свойства.

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.

Замечание 1.1 В литературе часто встречается следующая запись для системы 1.1:

$$x^+ = f(x, u), \tag{1.2}$$

где x^+ означает "следующее" состояние.

Начальное состояние системы (1.1) задано:

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

На состояния и управляющие воздействия u накладываются ограничения вида

$$(x(t), u(t)) \in Z \subseteq X \times U,\ t \in \mathbb{I}_{\geq 0}, \tag{1.3}$$

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 $\mu(x)$, при которой замкнутая система

$$x(t+1) = g(x(t)) = f(x(t), \mu(x(t))),\ x(0) = x_0, \tag{1.4}$$

будет устойчива в некотором заданном положении равновесия (заданном множестве), при этом переходный процесс не нарушает ограничения (1.3) при всех $t \in \mathbb{I}_{\geq 0}$.

Дадим определения.

ВОЗМОЖНО ЛУЧШЕ СРАЗУ ДАТЬ ОПРЕДЕЛЕНИЯ НЕ ДЛЯ МНОЖЕСТВА, А ДЛЯ ТОЧКИ

ПОПРАВИТЬ ОФОРМЛЕНИЕ ОПРЕДЕЛЕНИЙ, КАК 1.1

Определение 1.1 Множество $X \subseteq \mathbb{R}^n$ называется положительно инвариантным множеством для системы (1.4), если выполняется $g(x) \in X$ $\forall x \in X$.

Определение 1.2 Пусть $X \in \mathbb{R}^n$ — положительно инвариантное множество для системы 1.4. Замкнутое, положительно инвариантное множество $A \subseteq X$ устойчиво для 1.4, если $\forall \epsilon > 0$, $\exists \delta > 0$, что для всех $|x_0|_A \leq \delta, x_0 \in X$, выполняется $|x(t)|_A \leq \epsilon$, $\forall t \in \mathbb{I}_{\geq 0}$. Здесь $|x|_A = inf|x - a|$ — расстояние от точки x до множества A, |.|- евклидова норма. При $A = x^*$ — точка, получим устойчивость решения x^* по Ляпунову.

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

Определение 1.4 Множество A — глобально асимптотически устойчиво, если оно асимптотически устойчиво с $X = \mathbb{R}^n$. При $A = \{x^*\}$ имеем асимптотическую устойчивость решения $x(t) = x^*$ по Ляпунову.

Далее для простоты рассмотрим только случай стабилизации системы управления 1.1 для заданного положения равновесия, т.е. случай $A = \{x^*\} \in X$. Естественно, считается, что существует значение $\exists u^* \in U$ такое что

$$x^* = f(x^*, u^*), \quad (x^*, u^*) \in Z.$$

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. Под "оптимизацией будущего поведения" понимается решение прогнозирующей задачи ОУ.

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.

Поэтому состояния математической модели будем будем обозначать $x(k|t)$, $k = 0, 1, ...N - 1 = \mathbb{I}_{[0,N-1]}$. Они изменяются согласно уравнению

$$x(k+1|t) = f(x(k|t), u(k|t)), \; x(0|t) = x(t), \; k \in I_{[0,N-1]}. \tag{1.5}$$

Здесь аргумент t после черты подчеркивает зависимость от текущего момен-

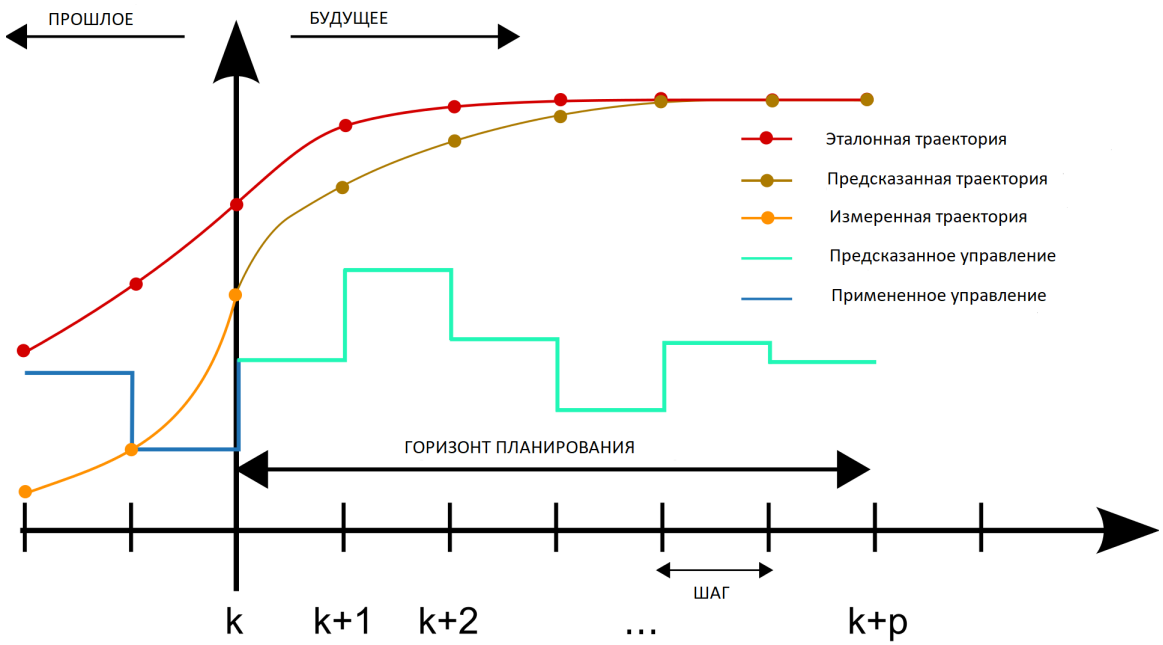

Рис. 1.1: Схема MPC.

та, для которого проводится оптимизация. Начальное состояние — текущее состояние объекта управления $x(t)$.

НЕ НРАВЯТСЯ ОБОЗНАЧЕНИЯ НИЖЕ, СМ. ОБОЗНАЧЕНИЯ ДЛЯ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ. НУЖНЫ ЛИ ОНИ ТУТ ВООБЩЕ? ИЛИ ОНИ НУЖНЫ ВЫШЕ, ПОТОМУ ЧТО $u(k|t)$ ФИГУРИРУЕТ В (1.5)?

Далее используются обозначения:

$u(t) = \{u(0|t), u(1|t), ..., u(N-1)\}$ — предсказываемое управляющее воздействие;

$x(t) = \{x(0|t), ..., x(N|t)\}$ — соответствующая траектория системы (1.5);

N — горизонт планирования.

Обсудим ограничения прогнозирующей задачи ОУ. Понятно, что она должна включать ограничения (1.3), записанные для состояний математической модели (1.5):

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

Кроме приведенных смешанных ограничений, в задачу, как правило, добавляются ограничения в терминальный момент времени: $x(N|t) \in X_f$, где $X_f \subset \mathbb{R}^n$ — терминальное множество. "Терминальные ингредиенты" прогнозирующей задачи более подробно будет рассмотрены ниже, после ее формализации.

Оставшийся элемент прогнозирующей задачи ОУ — критерий каче-

ства. 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 ингредиентов 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).

ЗДЕСЬ НУЖНО ПРОВЕРИТЬ ПУНКТ 2 ПРЕДПОЛОЖЕНИЙ И ПРИВЕСТИ В СООТВЕТСТВИЕ НАПРИМЕР С КОНСПЕКТОМ

Относительно l предполагается, что она непрерывна, а также:

1. $l(x^*, u^*) = 0$, т.е. стоимость обращается в нуль в точке равновесия;
2. имеет 2 разных вида согласно [1], $l(x, u) > 0$ для $\forall (x, u) \in Z, x \neq x^*$ согласно [2], $\exists$ функция α_1 класса K_∞, что выполняется $l(x, u) \geq \alpha 1(|x - x^*|)$ $\forall (x, u) \in Z$.

Ниже, в разделе 1.4, будут обсуждаться популярные решения при выборе l.

А ЕСТЬ ЛИ ОБСУЖДЕНИЕ?

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

$$\mathcal{P}(t): \qquad \min_{u(t)} \sum_{k=0}^{N-1} l(x(k|t), u(k|t)) + V_f(x(N|t)), \tag{1.6}$$

при условиях

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

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

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

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

НУЖНО ЛИ ЭТО ОБОЗНАЧЕНИЕ?

Часто критерий качества обозначают через $J_N(x(t), u(t)) = \sum_{k=0}^{N-1} l(x(k|t), u(k|t)) + V_f(x(N|t))$.

В задаче (1.6) обсуждавшиеся выше "терминальные ингредиенты":

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

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

Далее используем следующие обозначения:
$u^0(\cdot|t) = \{u^0(0|t), \dots, u^0(N-1|t)\}$ — оптимальное (программное) управление задачи $\mathcal{P}(t)$;
$x^0(\cdot|t) = \{x^0(0|t), \dots, x^0(N|t)\}$ — соответствующая траектория;

В СЛЕДУЮЩЕЙ СТРОКЕ КАКАЯ-ТО ПУТАНИЦА В ОБОЗНАЧЕНИЯХ?

$V_f(x(t)) = J_N(x(t), u^0(t))$ — оптимальное значение критерия качества (value function, функция Беллмана);
X_N — множество всех состояний $x \in X$, для которых существует решение задачи (1.6) с $x(t) = x$.

Замечание 1.2 Если $\mathcal{P}(t)$ имеет не единственное решение, то выбирается какое-то одно, из прочих соображений (например, используется другой критерий качества). Считаем далее, что $u^0(\cdot|t)$ — единственное оптимальное управление.

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 (1.6) с начальным условием $x(0|t) = x(t)$, получить ее решение $u^0(\cdot|t)$;
3. подать на вход системы (1.1) управляющее воздействие

$$u_{MPC}(t) := u^0(0|t). \tag{1.7}$$

Таким образом, в каждый момент $t \in \mathbb{I}_{\geq 0}$ на систему подается управляющее воздействие (1.7), которое неявно зависит от текущего состояния $x(t)$. Соответственно, замкнутая система имеет вид

$$x(t+1) = f(x(t), u^0(0|t)), t \in \mathbb{I}_{\geq 0}. \tag{1.8}$$

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 (1.6).

1.4 Терминальные ингредиенты и устойчивость замкнутой системы

Схемы MPC зависят от выбора терминальных элементов V_f и X_f. Например, самые первые результаты исследований по MPC [] 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 $x(N|t) = 0$. Сразу отметим, что ограничения-равенства считаются "плохими" с точки зрения вычислительных алгоритмов, поэтому дальнейшее развитие теории продолжалось в направлении ослабления этих простейших условий.

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

3. V_f непрерывна;

4. терминальное множество X_f замкнуто, $\{x^*\} \in \int X_f$;

5. существует локальная обратная связь $k_f : X_f \to U$, такая что для $\forall x \in X_f$ имеет место

   а) $(x, k_f(x)) \in Z$;

   б) $f(x, k_f(x)) \in X_f$;

   в) $V_f(f(x, k_f(x))) - V_f(x) \leq -l(x, k_f(x)) + l(x^*, u^*)$.

В 5) условие а) гарантирует допустимость локальной обратной связи $k_f(x)$, $x \in X_f$. Условие б) означает, что терминальное множество X_f является положительно инвариантным для системы, замкнутой локальной обратной связью: $x(t+1) = f(x(t), k_f(x(t))) = g(x(t))$. Условие в) вместе с условиями на l означает, 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 1.8, полученная в результате применения базового алгоритма, удовлетворяет ограничениям 1.1 для всех $t \in \mathbb{I}_{\geq 0}$;

2. задача (1.6) имеет решение для всех $t \in \mathbb{I}_{\geq 0}$;

3. x^* — асимптотически устойчивое положение равновесия с областью притяжения X_N.

ПРАВИЛЬНО ЛИ Я ПОНЯЛА, ЧТО ПОЖУРИЛИ ЗА ВКЛЮЧЕНИЕ ДОКАЗАТЕЛЬСТВА? ЕСЛИ ДА, ТО УБРАТЬ. ВМЕСТО ЭТОГО ПРИВЕСТИ СООБРАЖЕНИЯ ПО ВЫБОРУ V_f, X_f, И ДАЛЕЕ НУЖЕН ОБЗОР ЛИТЕРАТУРЫ ПО ТЕМЕ РАБОТЫ – ТЕ СТАТЬИ, ЧТО ВЫ ЧИТАЛА ПО РЕКОМЕНДАЦИИ ЙОХАНЕСА. ВЕСТЬ ТЕКСТ НИЖЕ, ПОЖАЛУЙ, НЕ НУЖЕН.

Д о к а з а т е л ь с т в о Доказательство взято из источника [2].Стандартный подход для доказательства устойчивости - использовать функцию Ляпунова как функцию для вычисления оптимального значения задачи оптимального управления на бесконечном горизонте. Это соответствует использованию $V_N^0(\cdot)$ для задачи оптимального управления на конечном горизонте. Функция $V_\infty(\cdot)$ удовлетворяет равенству $V_\infty^0(f(x,k_\infty(x))) = V_\infty^0(x) - l(x,k_\infty(x))$, тем самым удовлетворяя предположение 5. Оптимальность системы не всегда обеспечивает устойчивость этой системы, когда горизонт конечный, поэтому от правильного выбора терминальных ингредиентов будет зависеть устойчивость системы. Если мы докажем для функции на конечном горизонте верно, что $V_N^0(f(x,k_N(x))) \leq V_N^0(x) - l(x,k_N(x))$ для $\forall x \in X_N$,то это будет гарантировать асимптотическую устойчивость.

Пусть у нас не будут ограничения на состояние, т.е. $X = X_f = X_N = \mathbb{R}^n$ и x - любое состояние $\in X_N$ в момент времени 0. Тогда

$$V_N^0(x) = V_N(x, u^0(x))$$

в которой $u^0(\cdot|x) = \{u^0(0|x), u^0(1|x), ..., u^0(N-1|x)\}$ - минимизирующая управляющая последовательность, и соотвествующая ей оптимальная последовательность состояний $x^0(\cdot|x) = \{x^0(0|x), x^0(1|x), ..., x^0(N|x)\}$, где $x^0(0|x) = x, x^0(1|x) = x^+$. Следующее состояние по отношение к x в момент времени 0 - это $x^+ = f(x, k_N(x)) = x^0(1|x)$ в момент времени 1, где $K_N(x) = u^0(0|x)$ и

$$V_N^0(x^+) = V_N(x^+, u^0(x^+))$$

в которой $u^0(\cdot|x^+) = \{u^0(0|x^+), u^0(1|x^+), ..., u^0(N-1|x^+)\}$. Сложно сравнивать $V_N^0(x)$ и $V_N^0(x^+)$ непосредственно, но

$$V_N^0(x^+) = V_N(x^+, u^0(x^+)) \leq V_N(x^+, \bar{u})$$

где $\bar{u}$ некоторая допустимая управляющая последовательность. Для облегчения сравнения, мы выберем $\bar{u} = \{u^0(1|x), ..., u^0(N-1|x), u\}$ и соотвествующая ей последовательность состояний $\bar{x} = \{x^0(1|x), ..., x^0(N|x), f(x^0(N|x), u)\}$. Так как u и $\bar{u}$ совпадают для $i = 1, .., N-1$, но не для $i = N$. То запишем в таком виде

$$V_N^0(x) = V_N(x, u^0(x)) = l(x, k_N(x)) + \sum_{j=1}^{N-1} l(x^0(j|x), u^0(j|x)) + V_f(x^0(N|x))$$

а значит мы можем выразить $V_N(x^+, \bar{u})$:

$$V_N(x^+, \bar{u}) = V_N^0(x) - l(x, k_N(x)) - V_f(x^0(N|x)) + l(x^0(N|x), u) + V_f(f(x^0(N|x), u))$$

А так как $V_N^0(x^+) \leq V_N(x^+, \bar{u})$, то подставим сюда полученные выше равенства и получим, что

$$V_N^0(f(x, k_N(x))) - V_N^0(x) \leq -l(x, k_N(x))$$

если верно следующее

$$V_f(f(x,u)) - V_f(x) + l(x,u) \leq 0$$

Что и требовалось доказать.

1.5 Терминальное множество и терминальная функция

В задаче (??) терминальные "ингредиенты":

- функция терминального состояния $V_f(x(N|t))$ 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. Существует несколько подходов для нахождения этих терминальных "ингредиентов". В рамках диссертации нас будут интересовать следующие два подхода:

- MPC на квази-бесконечном горизонте
- Обобщенный фреймворк для MPC

1.5.1 MPC на квази-бесконечном горизонте

MPC на квази-бесконечном горизонте оптимизирует on-line функционал, состоящий из стоимости на конечном горизонте и терминальной стоимости, при условиях динамичности системы, входных ограничений и дополнительному ограничению на терминальное состояние. Выполнимость неравества для терминального ограничения подразумевает под собой, что состояния в конце конечного горизонта в предписанном терминальном множестве. Терминальные состояния штрафуются таким образом, что терминальная стоимость ограничивает стоимость на бесконечном горизонте для нелинейной системы, управляемой фиктивной локальной линейной обратной связью. Если первое приближение системы стабилизируемо, то существует единственное решение уравнения Ляпунова, и тогда можно определить терминальную функцию и множество off-line.

Рассмотрим первое приближение системы в начале координат

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

и получим линейную систему

$$\dot{x} = Ax + Bu$$

где $A = (\frac{\partial f}{\partial x})(0,0)$ и $B = (\frac{\partial f}{\partial u})(0,0)$ Если уравнение можно стабилизировать, то линейная обратная связь для состояния

$$u = Kx$$
$$A_K = A + BK$$

асимптотически устойчива.

Лемма 1.1 Предположим, что первое приближение системы в начале координат стабилизируемое, тогда

1. уравнение Ляпунова

$$(A_K + kI)^T P + P(A_K + kI) = -Q^* \tag{1.9}$$

допускает единственную положительно определенную и симметричную матрицу P, где $Q^* = Q + K^T RK$ - положительно определенная и симметричная; $k \in [0, \infty)$ удовлетворяет $k < -\lambda_{max}(A_K)$.

2. $\exists \alpha \in (0, \infty)$ определяющая окрестность Ω_α начала координат в форме $\Omega_\alpha = \{x \in \mathbb{R}^n | x^T P x \leq \alpha\}$ такая что

   (a) $Kx \in U$, для $\forall x \in \Omega_\alpha$, т.е. линейный контроллер с обратной связью не нарушает входных ограничений в Ω_α

   (b) Ω_α инвариантно для нелинейных систем, контролируемых локальной линейной обратной связью $u = Kx$

   (c) для любого $x_1 \in \Omega_\alpha$, критерий качества для бесконечного горизонта $J^\infty(x_1, u) = \int_{t_1}^{\infty} (\|x(t)\|_Q^2 + \|u(t)\|_R^2) dt$ ограничен таким образом:

$$J^\infty(x_1, u) \leq x_1^T P x_1.$$

Доказательство Доказательство взято из источника [4].

1. Так как $Q^* > 0$, для разрешимости уравнения Ляпунова необходимо,чтобы действительные части всех собственных значений $A_k + kI$ были отрицательными, если это выполняется, то уравнение Ляпунова 1.9 разрешимо и решение единственно, положительно определенное и симметричное.Так как A_k - асимптотически устойчива, то любая константа $k \in [0, -\lambda_{max}(A_K)]$ гарантирует отрицательность действительной части собственных значений $(A_K + kI)$

2. (a) Так как $P > 0$ и $0 \in \mathbb{R}^m$ в $\text{int}U$, тогда можно найти $\alpha_1 \in (0, \infty)$, такое, что $Kx \in U$ для $\forall x \in \Omega_{\alpha_1}$, а значит линейная управляющая обратная связь удовлетворяет входным ограничениям на Ω_{α_1}. Допустим $\alpha \in (0, \alpha_1]$ определяет область в форме $\Omega_\alpha = \{x \in \mathbb{R}^n | x^T P x \leq \alpha\}$, тогда это множество тоже будет удовлетворять входным ограничениям, так как $\alpha \leq \alpha_1$.

   (b) Продифференцируем $x^T P x$ вдоль траектории $\dot{x} = f(x, Kx)$ и получим

$$\frac{d}{dt} x(t)^T p x(t) = x(t)^T (A_k^T P + P A_k) x(t) + 2x(t)^T P \phi(x(t)) \tag{1.10}$$

   где $\phi(x) = f(x, Kx) - A_k x$. Так как последнее слагаемое ограничено таким образом

$$x^T P \phi(x) \leq \| x^T \| \cdot \| \phi(x) \| \leq \| P \| \cdot L_\phi \cdot \| x \|^2 \leq \frac{\| P \| L_\phi}{\lambda_{min}(P)} \| x \|_P^2$$

   где $L_\phi = sup\{\| \phi(x) \| / \| x \| | x \in \Omega_\alpha, x \neq 0\}$. Теперь мы выбираем $\alpha \in (0, \alpha_1]$, такой что в Ω_α

$$L_\phi \leq \frac{k \lambda_{min}(P)}{\| P \|}$$

   Тогда неравенство ведет к

$$X^T P \phi(x) \leq k x^T P x$$

Подставим полученное неравенство в (1.10) и получим следующее

$$\frac{d}{dt}x(t)^T Px(t) \leq x(t)^T((A_k + kI)^T P + P(A_k + kI))x(t)$$

в свою очередь это ведет к $\frac{d}{dt}x(t)^T Px(t) \leq -x(t)^T Q^* x(t)$. Так как $P > 0$ и $Q^* > 0$, то это неравенство предполагает, что область Ω_α инвариантна для системы, и любая траектория начинающаяся из этой области сходится к началу координат.

(c) Если проинтегрируем $\frac{d}{dt}x(t)^T Px(t) \leq -x(t)^T Q^* x(t)$ от t_1 до ∞ с начальным условием $x(t_1) = x_1$, то получим необходимый результат $J^\infty(x_1, u) \leq x_1^T Px$.

Алгоритм нахождения терминальной функции и терминального множества

1. Найти линейную обратную связь Kx.
2. Выбрать константу $k \in [0, \infty)$, удовлетворяющую неравенству $k < -\lambda_{max}(A_K)$ и решить уравнение Ляпунова для нахождения P.
3. Найти наибольшую из возможных α_1, такую что $Kx \in U$, для $\forall x \in \Omega_{\alpha_1}$.
4. Найти наибольшую из возможных $\alpha \in (0, \alpha_1)$, такую что неравенство

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

удовлетворяется для Ω_α.

Теорема 1.2 Допустим

1. $f : \mathbb{R}^n \times \mathbb{R}^m -> \mathbb{R}^n$ дважды дифференцируема, $f(0, 0) = 0 => 0$ - точка равновесия системы с $u = 0$
2. $U \subset \mathbb{R}^m$ компактно, выпукло и $0 \in intU$
3. $\exists$ решение задачи для любого начального $x_0 \in \mathbb{R}^n$, задача имеет кусочно непрервную $u(\cdot) : [0, \infty) -> U$

Если 1-3 предположения выполняются и

- первое приближение системы стабилизируемо
- программируемое управление разрешимо в $t = 0$

То замкнутая система асимптотически устойчива.

1.5.2 Обобщенный фреймворк для MPC

Данный фреймворк был придуман для расширения класса систем, для которых применим MPC. Системы должны удовлятворять достаточно нестрогим условиям и терминальные игредиенты будут выбираться таким образом, чтобы удовлетворить некоторые условия устойчивости, и таким образом замкнутая система будет иметь гарантированную асимптотическую устойчивость. В MPC на квази-бесконечном горизонте мы работали с

темами системами, для которых первое приближение стабилизируемо, в данном же фреймворке такие требования не предъявляются и класс систем расширяется и включает в себя неголономные системы тоже.

Нелинейная система

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

Гипотезы, которые предполагаем для систем, которыми будем управлять:

1. $U(t)$ содержит начальную точку и $f(t, 0, 0) = 0$
2. Функция f непрерывная и $x \mapsto f(t, x, u)$ локально непрерывна по Липшицу для $\forall(t, u)$
3. $U(t)$ компактно и для любой пары $(t, x)f(t, x, U(t))$ выпукло
4. Функция f компактна на компакте множеств x, т.е. $\{\| f(t, x, u) \|: t \in \mathbb{R}, x \in X, u \in U(t)\}$ компактно

Как видно из гипотез, в отличие от MPC на квази-бесконечном горизонте, мы не требуем от системы, чтобы существовало ее решение, поэтому данные условия достаточно не строгие, но они вместе с новыми условиями устойчивости будут гарантировать асимптотическую устойчивостьт системы. Условия устойчивости:

1. Множество X_f замкнуто и содержит начало координат
2. l непрерывна, $l(\cdot, 0, 0) = 0$ и радиально неограниченная функция $\exists M : \mathbb{R}^n -> \mathbb{R}_+$, такая что $l(t, x, u) \geq M(x)\forall(t, u) \in \mathbb{R} \times \mathbb{R}^m$. Более того расширенное множество скорости $\{(v, l) \in \mathbb{R}^n \times \mathbb{R}_+ : v = f(t, x, u), l \geq l(t, x, u), u \in U(t)\}$ выпукло для любых (t, x)
3. V_f положительно полуопределенная и непрерывно дифференцируема
4. Горизонт планирования T такой что, множество X_f достижимо для любого начального состояния и что существует $\forall(t_0, x_0) \in \mathbb{R} \times X, \exists u : [t_0, t_0 + T] \to \mathbb{R}^m$ удовлетворяющее $x(t_0 + T; t_0, x_0, u) \in X_f$ и $x(t; t_0, x_0, u) \in X, \forall t \in [t_0, t_0 + T]$
5. Тогда существует $\exists \epsilon > 0, \forall t \in [T; \infty), x_t \in X_f$ мы можем выбрать управляющее воздействие $\bar{u} : [t, t + \epsilon] -> \mathbb{R}^m, \bar{u}(s) \in U(s)$ удовлетворяющее

$$\forall s \in [t, t + \epsilon], \frac{dV_f(t, x_t)}{dt} + \frac{dV_f(t, x_t)}{dx} f(t, x_t, \bar{u}(t)) \leq -l(t, x_t, \bar{u}(t))$$

и $x(t + r; t, x_t, \bar{u}) \in X_f \forall r \in [0, \epsilon]$

Теорема 1.3 Предположим гипотезы 1-4. Выберем параметры для системы,удовлетворяющей условиям 1-5. Тогда замкнутая система будет асимптотически устойчивой, $\| x^*(t) \| - > 0$ при $t - > \infty$

ГЛАВА 2

ИСПОЛЬЗОВАНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В СИСТЕМАХ УПРАВЛЕНИЯ С ПРОГНОЗИРУЮЩЕЙ МОДЕЛЬЮ

2.1 Основные понятия нейронных сетей

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

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

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

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

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

Каждый скрытый уровень сети обычно является векторным. Размерность этих скрытых слоев определяет ширину модели. Каждый элемент вектора может быть интерпре-

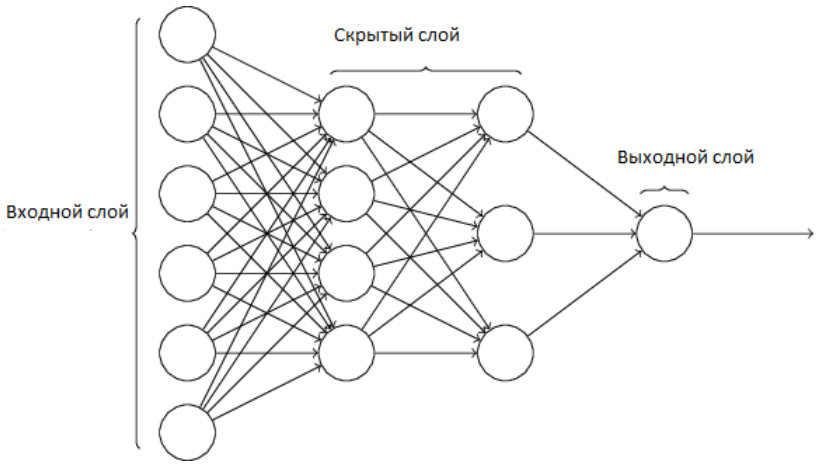

Рис. 2.1: Нейронная сеть со скрытыми слоями.

тирован как играющий роль, аналогичную нейрону. Вместо того, чтобы думать о том, что слой представляет собой единую вектор-векторную функцию, мы также можем думать о том, что этот слой состоит из множества единиц, которые действуют параллельно, каждый из которых представляет собой вектор-скалярную функцию. Каждый блок напоминает нейрон в том смысле, что он получает вход от многих других единиц и вычисляет его собственное значение активации.

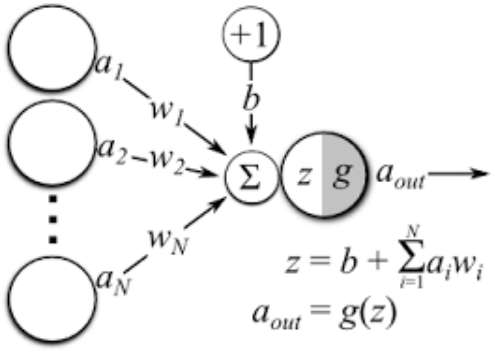

Рис. 2.2: Строение нейрона.

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. В обоих случаях некоторые входы становятся более важными, чем другие, так что они оказывают большее влияние на обрабатывающий элемент, поскольку они объединяются для создания нейронного ответа. Веса - это адаптивные коэффициенты в сети, которые определяют интенсивность входного сигнала, зарегистрированного искусственным нейроном. Они являются мерой прочности соединения входа. Эти сильные стороны могут быть изменены в ответ на различные обучающие наборы и в соответствии с конкретной топологией сети или с помощью ее правил обучения.На рисунке (2.2) веса обозначены w_i, значения нейронов предыдущего слоя - a_i. b параметр представляет собой смещение для линейного преобразования входных нейронов. Таким образом, мы получаем значение функции суммирования в виде линейного преобразования $z = b + \sum_{i=1}^{N} a_i w_i$.

Функция g на рисунке (2.2) - это функция активации нейрона. Цель использова-

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

Существует несколько видов обучения нейронных сетей:

- Обучение с учителем
- Обучение без учителя

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

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

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

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

Согласно универсальной теореме аппроксимации — нейронная сеть с одним скрытым слоем может аппроксимировать любую непрерывную функцию многих переменных с лю-

бой точностью. Главное чтобы в этой сети было достаточное количество нейронов. И еще важно удачно подобрать начальные значения весов нейронов. Чем удачнее будут подобраны веса, тем быстрее нейронная сеть будет сходиться к исходной функции. Это означает, что нелинейная характеристика нейрона может быть произвольной: от сигмоидальной до произвольного волнового пакета или вейвлета, синуса или многочлена. От выбора нелинейной функции может зависеть сложность конкретной сети, но с любой нелинейностью сеть остаётся универсальным аппроксиматором и при правильном выборе структуры может достаточно точно аппроксимировать функционирование любой непрерывной функции.

2.2 Методы обучения с подкреплением

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

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

Базовое RL моделируется как процесс принятия марковских решений:

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

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

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

изменяется либо по фиксированному расписанию, либо адаптивно основываясь на эвристике.

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

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

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

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

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

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,$\gamma \in [0, 1]$ - ставка дисконтирования.

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

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

Чтобы определить оптимальность формальным образом, определим прибыль от 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 определяется 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 действуем оптимально (принимаем оптимальное действие), выбирая действие из $Q^{\pi^*}(s, \cdot)$ с

наивысшим значением в каждом состоянии, s. Функция прибыли от действия такой оптимальной стратегией Q^{π^*} называется оптимальной функцией прибыли от действия и является обычно обозначаемый Q^*. Таким образом, знание оптимальной функции стоимости действия достаточно, чтобы знать, как действовать оптимально.

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

2.3 Методы дизайна оффлайн регуляторов в системах управления с прогнозирующей моделью

Существует несколько подходов использования методов обучения в MPC системах:

- 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 Аппроксимация закона управления

Для линейных систем задача оптимизации может быть решена оффлайн, т.е. до онлайн запуска системы при некоторых слабых предположениях [6]. Таким образом, получается явный закон управления. Расширение [6] до нелинейных систем не является прямым, и существуют также проблемы сложности вычислений, касающиеся метода из [6].

Система вида $x^+ = Ax + Bu$ с линейными ограничениями вида $C_x x \leq d_x$, $C_u u \leq d_u$, где оптимизационная задача записывается в таком виде:

В момент времени t, вычисляем состояние $x(t)$, решаем задачу

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

с условиями

$$x(k+1|t) = Ax(k|t) + Bu(k|t)$$
$$x(t|t) = x(t)$$
$$C_x x(k|t) \leq d_x$$
$$C_u u(k|t) \leq d_u$$

for $t \leq k \leq t+N-1$ и терминальным ораничением

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

с квадратичной функцией стоимости перехода и квадратичной терминальной функцией $L(x,u) = x^T Q x + u^T R u,\ Q, R > 0,\ F(x) = x^T P x$

Можно переписать задачу в виде задачи квадратичного программирования, обозначим

$$X := [x^T(t+1|t), \dots, x^T(t+N|t)]^T \tag{2.8}$$

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

- Перепишем функцию стоимости

$$F(x(t), U) = x^T(t)Qx(t) + X^T\tilde{Q}X + U^T\tilde{R}U \tag{2.10}$$

with $\tilde{Q} = \begin{bmatrix} Q & & & \\ & \ddots & & \\ & & Q & \\ & & & P \end{bmatrix}$, $\tilde{R} = \begin{bmatrix} R & & \\ & \ddots & \\ & & R \end{bmatrix}$

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

$$\Rightarrow X = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} x(t) + \begin{bmatrix} B & 0 & \dots & & 0 \\ AB & B & 0 & \dots & 0 \\ \dots & & & & \\ A^{N-1}B & A^{N-2}B & \dots & \dots & B \end{bmatrix} U \tag{2.11}$$

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

Подставим (2.11) в (2.10) : $J(x(t), U) = \frac{1}{2}x^T(t)Yx(t) + \frac{1}{2}U^THU + x^T(t)FU$

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

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

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

Ограничения можно переписать таким же образом и получим

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

$$\min_U \frac{1}{2}U^THU + x^TFU + \frac{1}{2}x^TYx \tag{2.12}$$

при условии $GU \leq W + Ex$

Далее еще применим замену переменных вида $z := U + H^{-1}F^Tx$,где H^{-1} - положительно определенная матрица.

$$\min_z \frac{1}{2}z^THz + \frac{1}{2}x^T\tilde{Y}x \tag{2.13}$$

s.t. $Gz \leq W + Sx$

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

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

Далее эта задача решается посредством выпуклой оптимизации через условия Каруша-Куна-Такера и тогда так как задача (строго)выпуклая с разрешим множеством с непустой внутренней частью(по предположениям), значит условия Слейтера выполняются. Оптимальное решение единственное и характеризуется условиями ККТ.

Оптимальное множество $z^*(x)$ - оптимальное решение задачи (2.13) для данного x

$$A(x) := \{i \in \{1, \dots q\} | G^i z^*(x) = W^i + S^i x\}$$

G^A, W^A, $S^A \dots$ матрицы, содержашие ряды G, U, S, которые ассоциируются с множеством A.

Тогда мы имеем решение λ^A и $z^*(x)$

$$\lambda^A = -(G^A H^{-1} (G^A)^T)^{-1} (W^A + S^A x) \tag{2.14}$$

$$z^*(x) = H^{-1} (G^A)^T (G^A H^{-1} (G^A)^T)^{-1} (W^A + S^A x) \tag{2.15}$$

Множество состояний, где A - оптимальное активное множество(критический регион CR^A). Критический регион задается посредством следующих неравенств:

$$\begin{cases} GH^{-1}(G^A)^T (G^A H^{-1} (G^A)^T)^{-1} (W^A + S^A x) \leq W + Sx \\ -(G^A H^{-1} (G^A)^T)^{-1} (W^A + S^A x) \geq 0 \end{cases}$$

Теорема 2.1 Для линейных MPC(линейной системы, линейных ограничений, квадритичной стоимости перехода), результирующий MPC положительно определенный регулятор $u_{MPC}(x)$ непрерывный и кусочно-афинный на регионах в виде многогранника.Оптимальное значение функции стоимости для задачи (2.12), $F^*(x)$, непрерывное, выпуклое и кусочно-квадратичное.

Явный MPC решает задачу для всех состояний, таким образом все пространство состояний делится на регионы, где в каждом регионе для состояния есть явная функция управления. Главный недостаток этого метода состоит в том, что количество регионов может быть достаточно большим, что в онлайн процедуре может плохо сказываться на производительности, т.к. нужно будет искать к какому из регионов относится текущее состояние,чтобы опредилить управление для него.

Существует несколько подходов к получению аппроксимативного решения для оптимизации MPC. Для линейных систем в [7] алгоритм обучения представлен дополнительными ограничениями для обеспечения стабильности и ограничения удовлетворения аппроксимационного MPC. 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 неисчезащих ошибках аппроксимации. Ошибка аппроксимации, выведенная в [8], [13], обычно не достижима для практического применения.

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

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

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 сети $j = 1..m$ и линейными активационными функциями.

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

$$\min_{w_j} \max_{x_t \in X} |\gamma_{RH_j}^0(x_t) - \gamma_{RH_j}^{(v_j)}(x_t, w_j)| \leq \frac{\epsilon}{\sqrt{m}},\ j = 1, ..., m \tag{2.16}$$

Процедура нахождения количества нейронов достаточно наивная, но работающая: для каждого j увеличиваем v_j пока (2.16) не станет верным. Также придовится в статье теорема, в которой утверждается, что для любой функции управления $\gamma_{RH_j}^0(x_t)$ число параметров, необходимых для достижения погрешности приближения L_2 или L_∞ порядка $O(\frac{1}{v_j})$, равно $O(v_j n)$, которое растет линейно с размерностью n вектора состояния.

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

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

И для этой системы минимизировалась стоимость вида:

$$\begin{aligned} J_N(U_N(k); k) = \sum_{i=k}^{k+N-1} [(\hat{y}(i+1|k) - y_r(i+1))^T Q(\hat{y}(i+1|k) - y_r(+1)) \\ + \triangle u(i)^T R \triangle u(i)] + q_N(\hat{x}(k+N|k)) \end{aligned} \tag{2.17}$$

где y_r - эталонная траектория, $\triangle u(k) = u(k) - u(k-1)$ - инкрементное управление. Для этой задачи создают нейронный регулятор вида $\triangle u_{opt}(k) = g(I_{MPC}(k))$, где $I_{MPC}(k) = \{\hat{x}(k|k), y_r(k+1), ..., y_r(k+N)\}$. Данный нейронный регулятор обучается с функцией потерь вида (2.17). Для решения оптимизационной задачи на этапе обучения нейронной сети используется градиентный спуск. Достаточно хорошие результаты на вычислительных этапах были получены, однако опять же не гарантируется устойчивость данного метода.

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

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

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 предполагают,что существует функция K класса, такая что $|\hat{f}(x,u) - \hat{f}(x,u')| \leq \eta_u(|u-u'|)\ \forall x \in X$ и $\forall(u,u') \in U^2$. Также вводится предположение об ограниченности возмущения и то что верно $|d_t| \leq \mu(|\xi_t|)\ t \geq 0$, 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 $q_t = \xi_{x_t} - x_t$ и управления $v_t = k^*(x_t) - k^(\xi_{x_t})$. Тогда система уже переписывается в таком виде:

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

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

2.3.2 Аппроксимация динамики системы с прогнозирующей моделью

Существует методы для аппроксимации динамики системы, в работе [14] был предложен доказуемо безопасный и робастный метод управления основанный на обучении для систем с прогнозирующей моделью. Мотивация этой статьи состоит в том, чтобы разработать схему управления, которая может

1. обрабатывать ограничения состояния и управления
2. оптимизировать производительность системы в отношении функции стоимости
3. использовать инструменты статистической идентификации для изучения неопределенностей модели
4. доказуемо сходиться

Введена форма надежного, адаптивного модельного прогнозирующего управления, основанный на обучении метод, на который ссылаются, как LBMPC. Основная идея LBMPC заключается в том, что производительность и безопасность могут быть разделены в рамках MPC с использованием инструментов доступности. В частности, LBMPC повышает производительность за счет выбора входных данных, которые минимизируют затраты, связанные с динамикой изученной модели, которая обновляется с использованием статистики, обеспечивая при этом безопасность и стабильность, используя теорию из робастного MPC, чтобы проверить, примененное управление сохраняет ли номинальную модель устойчивой, когда она подвержена неопределенности.

Динамика системы представляется в таком виде:

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

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

Данный метод вводит дополнительную систему, которая обучается на данных и имеет такой вид:

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

где O_n - зависящая от времени функция, которая обучается с помощью любого из статистических метод, параметрических или нет.

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 уже формулируется в таком виде:

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

при условиях

$$\tilde{x_n} = x_n,\ \bar{x_n} = x_n$$

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

$$\bar{x_{n+i+1}} = A\bar{x_{n+i}} + B\tilde{u_{n+i}}$$

$$\tilde{u_{n+i}} = K\bar{x_{n+i}} + c_{n+i}$$

$$\bar{x_{n+i+1}} \in X \ominus R_i,\ \tilde{u_{n+i}} \in U \ominus KR_i$$

$$(\bar{x_{n+N,\theta}}) \in \Omega \ominus (R_N \times \{0\})$$

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

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

А с помощью данного неравенства инвариантность возмущения:

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

Так как $\bar{x_s} = \Lambda\theta$ и $\bar{u_s} = \Psi\theta$, где $\theta \in \mathbb{R}^m$ и $\Lambda \in \mathbb{R}^{n\times m},\ \Psi \in \mathbb{R}^{m\times m}$ - точки устойчивого состояния, то их можно достичь для системы если $A+BK$ устойчива по Шуру при управлении $\bar{u_n} = K(\bar{x_n} - \bar{x_s}) + \bar{u_s} = K\bar{x_n} + (\Psi - K\Lambda)\theta$.Множества $R_0 = \{0\}$ и $R_i = \oplus_{j=0}^{i-1}(A+BK)^j W$ необходимы для робастного MPC, сужающиеся ограничения.

Устойчивость данного метода основана на робастном MPC.Результаты экспериментов показали хорошие результаты, были получены хорошие результаты относительно предсказанных траекторий и быстрой сходимости к точке устойчивого состояния, однако по производительности уступало нелинейному MPC.

2.3.3 Итерационный подход

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

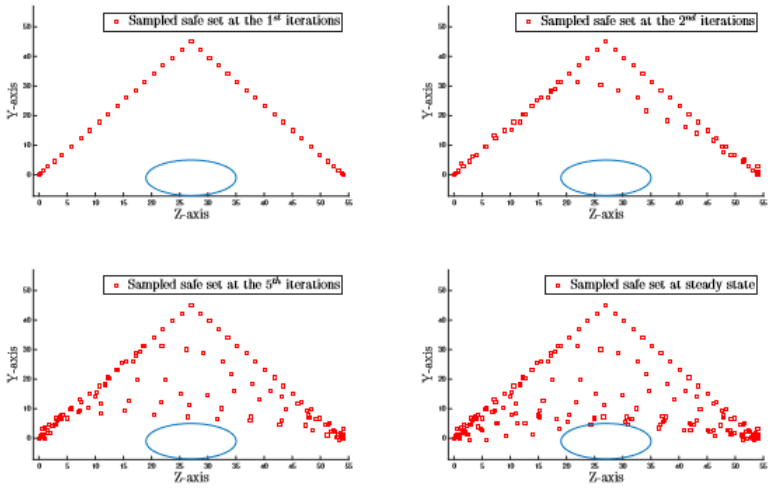

Рис. 2.3: Построение сэмплированного безопасного множества.

Управляемое на N шагов вперед множество по отношению к S формулируется рекурсивно как:

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

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

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

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

Вводится также стоимость на этом сэмплированном безопасном множестве:

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

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

$$J^{LMPC,j}_{t\to t+N}(x_t^j) = \min_{u_{t|t},\ldots,u_{t+N-1|t}} [\sum_{k=t}^{t+N-1} h(x_{k|t}, u_{k|t}) + Q^{j-1}(x_{t+N|t})]$$

при условии

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

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

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

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

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

ГЛАВА 3

ПОСТРОЕНИЕ НЕЙРОСЕТЕВОГО РЕГУЛЯТОРА

3.1 Постановка задачи

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 и $f(0,0) = 0$. Мы рассматриваем ограничения вида

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

Также необходимо, чтобы выполнялось ограничение $(x(t), u(t)) \in X \times U \; \forall t \geq 0$. Будем оптимизировать функцию стоимости $min_{u(\cdot|t)} \sum_{k=0}^{N-1} l(x,u) + V_f(X(N))$

3.2 Подход к решению

MPC регулятор, построенный для задачи (3.1), робастный по отношению к неточному управлению u в пределах выбранных границ.RMPC дискретизируется по множеству допустимых состояний X_{feas} и аппроксимируется с использованием нейронной сети, основанной на этих точках. Обучение дает апроксимированный MPC $\pi_{feas} : X_{feas} \rightarrow U | u = \pi_{approx}(x)$. С этим регулятором система замкнутого контура задается как $x(t+1) = f(x(t), \pi_{approx}(x(t)))$. Стабильность замкнутого контура гарантируется, если погрешность аппроксимации ниже допустимой границы входного возмущения от робастного MPC. Используется метод проверки, основанный на неравенстве Хоэффинга, чтобы гарантировать эту оценку.

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

- Необходимо брать достаточно много точек для хорошей аппроксимации, хотя иногда некоторые области имеют одинаковые значения и для них не надо вычислять заново управление
- Обучение и валидация проходят достаточно долго

Предполагаемые улучшения для этого подхода:

- Уменьшить кол-во генерируемых точек для гарантирования аппроксимации
- Попробовать более точно вычислить X_{feas}

- Расспараллелить получение точек и валидацию аппроксимированной функции нейронной сети
- Рассмотреть возможность использования обучение без учителя и обучение с подкреплением

3.3 Результаты обучения нейронной сети для данной задачи. Базовая реализация

Базовый вариант включает себя создание нейронной сети, которая обучается на состояниях $X = \{-1 \leq x_1 \leq 0.5,\ -1 \leq x_2 1.5\}$ и берутся точки этого множества с шагом 0.1. Далее вычисляем для этих точек управление и следующее состояние из этой точки. На рисунке 3.1 показано векторное поле этой сетки. Из каждой точки сетки можно увидеть куда эта точка перейдет. Горизонт планирования был взят $N = 20$б т.е. мы планируем и высчитываем управления и состояния на 20 шагов вперед. Относительно рисунка видно, что мы смогли из точки $(-0.7, -0.8)$ попасть достаточно близко к началу координат, однако если взять некоторые точки где следующее состояние выходит за рамки нашего обучения, то мы можем получить нестабильное решение.

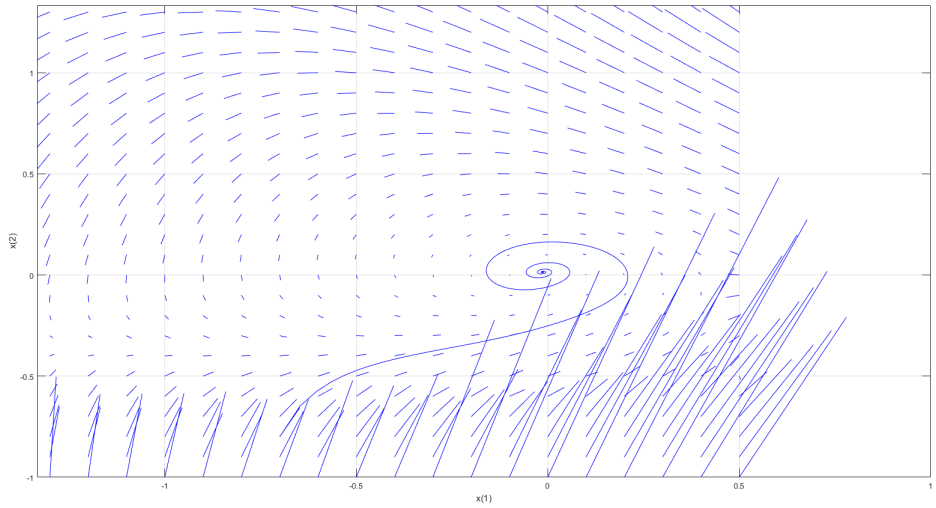

Рис. 3.1: Результаты базовой реализации.Векторное поле.

Далее если посмотреть на рисунок 3.2, то можно увидеть множество одинаковых управлений для состояний. Однако важно отметить, что сетка значений около начала координат должна быть достаточно детальной, чтобы не попасть в нестабильное положение управления.

Обучаемая нейронная сеть представляет сеть с одним скрытым слоем с 20 нейронами. Впоследствии будет описана процедура поиска оптимальнаго количества нейронов, а также других параметров нейронной сети.

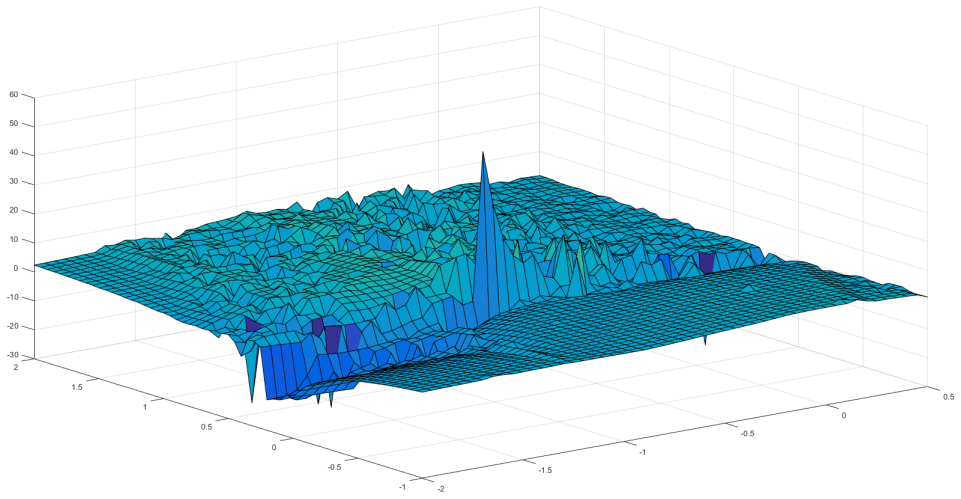

Рис. 3.2: Вычисленное управление для сетки значений состояний для обучения нейронной сети

ЗАКЛЮЧЕНИЕ

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

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

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

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. Будут рассмотрены алгоритмы обработки больших данных и построения искусственных нейронных сетей для решения данной проблемы. А также возможно использование обучение с подкреплением.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

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

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

3 Badgwell T.A., Qin S.J. (2015) Model-Predictive Control in Practice. In: Baillieul J., Samad T. (eds) Encyclopedia of Systems and Control. Springer, London

4 H.Chen, F.Allgower A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability / H. Chen, F. Allgower // Automatica. – 1998. – Vol. 34, no. 10. – P. 1205-1217.

5 Fernando A.C.C. Fontes A General Framework to Design Stabilizing Nonlinear Model Predictive Controllers / F. A.C.C. Fontes // Systems & Control letters. – 2000. P. 1-13.

6 Bemporad A. et al. The explicit linear quadratic regulator for constrained systems //Automatica. - 2002. - T. 38. - №. 1. - C. 3-20.

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

8 Chakrabarty A. et al. Support Vector Machine Informed Explicit Nonlinear Model Predictive Control Using Low-Discrepancy Sequences //IEEE Trans. Automat. Contr. - 2017. - T. 62. - №. 1. - C. 135-148.

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

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

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

12 Akesson B. M., Toivonen H. T. A neural network model predictive controller //Journal of Process Control. - 2006. - T.16. - №. 9. - C. 937-946.

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

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

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

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