МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра дискретной математики и алгоритмики

ПАВЛОВЕЦ Мария Евгеньевна

НЕЙРОСЕТЕВЫЕ РЕГУЛЯТОРЫ В СИСТЕМАХ УПРАВЛЕНИЯ С ПРОГНОЗИРУЮЩЕЙ МОДЕЛЬЮ

Отчет №3 о работе в рамках магистреской диссертации
специальность 1-31 81 09 «Алгоритмы и системы обработки больших объемов информации»

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

Минск 2018

ОГЛАВЛЕНИЕ

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

MPC – технология управления, основанная на решении в реальном времени последовательности специально сформулированных (прогнозирующих) задач оптимального управления с конечным временным горизонтом, которые аппроксимируют полубесконечную ЗОУ[1].

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

1.1 Основной принцип MPC

Основной принцип MPC состоит в следующем. В каждый момент времени измеряется текущее состояние динамического объекта управления и решается задача оптимального управления на конечном промежутке, в которой в качестве начального состояния прогнозирующей модели выбрано измеренное состояние. Вычисленное оптимальное управление (предсказанное управляющее воздействие) применяется к объекту на один момент (промежуток) времени. Затем снова измеряется состояние и оптимизация повторяется. Поскольку в каждой момент времени в задаче ОУ учитывается текущее состояние, результирующее управление представляет собой обратную связь. Популярность MPC в теоретических исследованиях и на практике обусловлена следующими свойствами:

- принимается во внимание критерий качества в ЗОУ, что позволяет учитывать экономические требования к процессу;
- учитываются жесткие ограничения на фазовые и управляющие переменные;
- метод применим к нелинейным и MIMO системам.

В западной литературе управление в реальном времени представлено теорией управления по прогнозирующей модели — Model Predictive Control (MPC), также называемая Receding Horizon Control (RHC). Основными приложениями теории являются задачи стабилизации динамических систем. Со-

временная теория нелинейного MPC предлагает основанные на решении задач оптимального управления методы построения обратных связей для нелинейных объектов.

Главная идея MPC — использование математической модели управляемого процесса в пространстве состояний для предсказания и оптимизации будущего поведения системы. Поясним на примере модели нелинейного процесса управления

$$\dot{x} = f(x, u) \tag{1.1}$$

где $x = x(t) \in \mathbb{R}^n$ — состояние модели в момент времени t; $u = u(t) \in \mathbb{R}^r$ — значение управляющего воздействия; $f : \mathbb{R}^n \to \mathbb{R}^r$ — заданная функция, обеспечивающая существование и единственность решения (1.1) при любом допустимом управляющем воздействии.

1.2 Задача стабилизации

Основным и исторически первым приложениям MPC являются задачи стабилизации и слежения. 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.

Основная идея MPC – использовать математическую модель процесса (в пространстве состояний) для предсказания и оптимизации поведения динамической системы в будущем [1]. Далее в этом разделе считаем, что используемая для предсказаний модель точно описывает процесс управления. Такие схемы MPC носят название номинальных (nominal MPC scheme). "Точная модель"означает, что на объект не действует возмущения и нет неучтенных различий между моделью и физическим объектом.

Система, которая исследуется в данном разделе, является нелинейной, дискретной, стационарной, т.е. имеет вид

$$x(t+1) = f(x(t), u(t)), x(0) = x_0. \tag{1.2}$$

Здесь $x(t) \in X \subseteq \mathbb{R}^n$ – состояние системы в момент времени t, $u(t) \in U \subseteq \mathbb{R}^n$ – управляющее воздействие в момент t, t – время, дискретное, $t \in \mathbb{I}_{\geq 0}$, где используется следующее обозначение – $t \in \mathbb{I}_{\geq a}$ множество всех целых чисел больше либо равных $a \in \mathbb{R}$.

Относительно функции $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ предполагается, что она непрерывна.

Замечание. В литературе часто встречается следующая запись для системы 1.2:

$$x^+ = f(x, u), \tag{1.3}$$

где x^+ означает "следующее"состояние (succesor state).

Начальное состояние системы 1.2 задано:

$$x(0) = x_0 \in X. \tag{1.4}$$

На состояния x и управляющие воздействия u накладываются ограничения вида $(x(t), u(t)) \in Z \subseteq X \times U, t \in \mathbb{I}_{\geq 0}$, которые являются смешанными ограничениями и включают в себя одновременно фазовые ограничения на состояния системы и прямые ограничения на управляющие воздействия.

Относительно множества $Z = X \times U$ предполагается, что оно компактно. Замечание. Вообще говоря, обычно требуется, чтобы множество U было компактным, а Z – замкнутым, и не обязательно ограничено. Но компактность Z требуется для робастного и экономического MPC, поэтому дальше будем считать Z компактным. Основная идея(стабилизирующего) MPC состоит в том, чтобы найти управляющее воздействие, при котором замкнутая система будет устойчива в некотором заданном положении равновесия (заданном множестве), при этом должны выполняться ограничения 1.2 при всех $t \in \mathbb{I}_{\geq 0}$. Дадим определения. Далее будем рассматривать систему

$$x(t+1) = g(x(t)), x(0) = x_0, \tag{1.5}$$

где $g : \mathbb{R}^n \to \mathbb{R}^n$.

Определение 1.1. $X \subseteq \mathbb{R}^n$ – положительно инвариантное множество для системы 1.5, если для $\forall x \in X$ выполняется $g(x) \in X$.

Определение 1.2 Пусть $X \in \mathbb{R}^n$ – положительно инвариантное множество для системы 1.5. Замкнутое, положительно инвариантное множество $A \subseteq X$ устойчиво для 1.5, 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 1.5, 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 про-

стоты рассмотрим только случай стабилизации системы управления 1.2 для заданного положения равновесия, т.е. случай $A = x^*$. При известной математической модели динамически управляемой системы (1.2) цель управления состоит в том, чтобы:

1. стабилизировать состояние x^*, где $\exists u^*$, так что $x^* = f(x^*, u^*), u(x^*, u^*) \in Z$;
2. обеспечить выполнение ограничений $(x(t), u(t)) \in Z$ для всех $t \in \mathbb{I}_{\geq 0}$.

1.3 Базовый алгоритм MPC

Как уже было сказано во введении к настоящему разделу, 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 (горизонт будет равен N) и использовать первое значение полученного оптимального(программного) управления в качестве значения обратной связи на интервале от t до t +1.

Под "оптимизацией будущего поведения"понимается решение ЗОУ с конечным горизонтом, здесь с $N \geq 2$ шагами.

В ЗОУ в качестве оптимизируемой системы выступает (1.1) с начальным состоянием равным текущему известному (измеряемому) состоянию.

Из вышесказанного понятно, что нужно различать состояния объекта управления x(t),$t \in \mathbb{I}_{\geq 0}$, из (1.1) и состояния модели, использующихся для предсказаний.

Поэтому будем оптимизировать состояния $x(k|t), k = 0, 1, ...N - 1 = \mathbb{I}_{[0,N-1]}$, которые изменяются, согласно

$$x(k+1|t) = f(x(k|t), u(k|t)), x(0|t) = x(t), k \in I_{[0,N-1]} \tag{1.6}$$

Здесь аргумент t после черты означает текущей момент, для которого будет проводиться оптимизация.

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 (1.6); N – горизонт планирования.

Понятно, что ЗОУ должна включать ограничения 1.2.

Кроме того, в ЗОУ также добавляются ограничения в терминальный момент времени. О "терминальных ингредиентах"ЗОУ более подробно будет рассказано ниже после формулировки ЗОУ.

Оставшийся элемент ЗОУ – критерий качества. В задачах стабилизации критерий качества выбирается исследователем, инженером, и является, скорее, параметром настройки схемы MPC. В новой теории экономического MPC критерий задан экономическими условиями.

Итак, в задаче стабилизации (см. [1]) критерий качества выбирается из соображений штрафа любого состояния $x \in X$, отклоняющегося от состояния равновесия x^*. Также часто штрафуются отклонения управления $u \in U$. Как отмечается в [1], последнее условие полезно с точки зрения вычисления решения ЗОУ, поскольку зачастую проще решить задачу, в которой в критерии качества штрафуются управляющие воздействия. С другой стороны [1], с точки зрения реализации и моделирования, это также полезно, поскольку желательно избежать значений $u \in U$, соответствующих чрезмерным энергетическим затратам.

Критерий качества будет состоять из 2-х "слагаемых т.е. это будет критерий качества типа Больца. Терминальное слагаемое будет рассмотрено позже.

Интегральное, которое для дискретных систем есть сумма стоимостей за каждый этап, будет иметь вид $\sum_{k=0}^{N-1} l(x(k|t), u(k|t)$.

В литературе, например в [1], функция l называется функций стоимости stage cost.

Относительно $l : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ предполагается, что l 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 также будут обсуждаться популярные решения при выборе l (см. стр. 46 [1]).

Сейчас мы готовы сформулировать ЗОУ (predictive optimal control problem) для момента времени t:

$$P : \min_{u(t)} \sum_{k=0}^{N-1} l(x(k|t), u(k|t)) + V_f(x(N|t)) \tag{1.7}$$

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

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 (1.7), обозначенной также P, терминальные "ингредиенты":

- функция терминального состояния $V_f(x(N|t))$ в критерии качества, которая в зарубежной литературе носит название terminal cost;
- 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.

Приведем эти условия, дополняющие условия (1 – 2) выше:

1. V_f-непрерывна
2. терминальное множество X_f замкнуто и содержит x^* как внутреннюю точку: $x^* \in intX_f$
3. существует локальная обратная связь $k_f : X_f \to U$, такая что для $\forall x \in X_f$ имеет место
   (a) $(x, k_f(x)) \in Z$;
   (b) $f(x, k_f(x)) \in X_f$;
   (c) $V_f(f(x, k_f(x))) - V_f(x) \leq -l(x, k_f(x)) + l(x^*, u^*)$.

Первое условие в 5) гарантирует допустимость локальной обратной связи $k_f(x), x \in X_f$.

Второе условие означает, что терминальное множество X_f является положительно инвариантным для системы, замкнутой локальной обратной связью: $x(t+1) = f(x(t), k_f(x(t))) = g(x(t))$.

Третье условие вместе с условиями на l, сформулированными выше означает, что терминальная стоимость может служить функцией Ляпунова на терминальном множестве X_f.

Далее используем следующие обозначения: $u^0(\cdot|t) = u^0(0|t), ..., u^0(N-1|t)$ – оптимальное управление (программное) задачи P. Если P имеет более, чем одно решение, то выбирается какое-то, из прочих соображений (например, используется другой критерий качества).

В целом считаем далее, что $u^0(\cdot|t)$ – единственное оптимальное управление $x^0(\cdot|t) = x^0(0|t), ..., x^0(N|t)$ – соответствующая траектория $V_f(x(t)) = J_n(x(t), u^0(t))$ – оптимальное значение критерия качества (value function, функция Беллмана) X_N – множество всех состояний $x \in X$, для которых решение задачи (1.7) с x(t) = x существует:

$$X_N = \{x \in R^n : \exists u = u(0), ...u(N-1) s.t. x(0) = x,$$
$$x(k+1) = f(x(k), u(k)), (x(k), u(k)) \in Z, \forall k \in I_{[0,N-1]}, x(N) \in X_f\}$$

Базовый алгоритм MPC: Для каждого $t \in \mathbb{I}_{\geq 0}$

1. измерить состояние $x(t) \in X$ системы;
2. решить задачу (1.7) с начальным условием x(0|t) = x(t), получить ее решение $u^0(\cdot|t)$;
3. подать на вход системы (1.2) управляющее воздействие

$$u(t) := u^0(0|t). \tag{1.8}$$

Замечание. В [1] имеется обозначение $u_N(t, x(t)) = u^0(t)$. Итак, в каждый момент $t \in \mathbb{I}_{\geq 0}$ на систему подается управляющее воздействие 1.8, которое неявно зависит от текущего состояния x(t). Соответственно замкнутая система имеет вид

$$x(t+1) = f(x(t), u^0(0|t)), t \in \mathbb{I}_{(\geq 0)} \tag{1.9}$$

Практически все схемы MPC зависят от выбора терминальных элементов $V_f X_f$. Например, самые первые результаты исследований по MPC были получены для состояния равновесия, совпадающего с началом координат $(x^*, u^*) = (0, 0), ..f(0, 0) = 0$, и предлагали использовать $X_f = 0$, т.е. терминальное условие принимало вид x(N|t) = 0.

Сразу отметим, что ограничения-равенства считаются среди исследователей MPC плохими с точки зрения вычислительных алгоритмов.

После 20 лет исследований по MPC теория приобрела свои нынешние черты и теперь достигнут консенсус о том, что все схемы номинального MPC описываются представленным базовым алгоритмом, а все ЗОУ, используемые для предсказаний, имеют вид (1.7).

1.4 Устойчивость замкнутой системы (основной результат)

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 1.9, полученная в результате применения базового алгоритма, удовлетворяет ограничениям 1.2 для всех $t \in \mathbb{I}_{\geq 0}$;

2. задача (1.7) имеет решение для всех $t \in \mathbb{I}_{\geq 0}$;

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

Доказательство. 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 запи-

шем 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 (1.7) 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

$$(A_K + kI)^T P + P(A_K + kI) = -Q^* \tag{1.10}$$

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.

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 1.10 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

$$\frac{d}{dt} x(t)^T p x(t) = x(t)^T (A_k^T P + P A_k) x(t) + 2x(t)^T P \phi(x(t)) \quad (1.11)$$

   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

$$X^T P\phi(x) \leq kx^T Px$$

Подставим полученное неравенство в (1.11) 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 теми 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 $\forall (t_0, x_0) \in \mathbb{R} \times X, \exists u : [t_0, t_0 + T] \rightarrow \mathbb{R}^m$ 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.

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

рые приводят к приближению функции наилучшим образом.

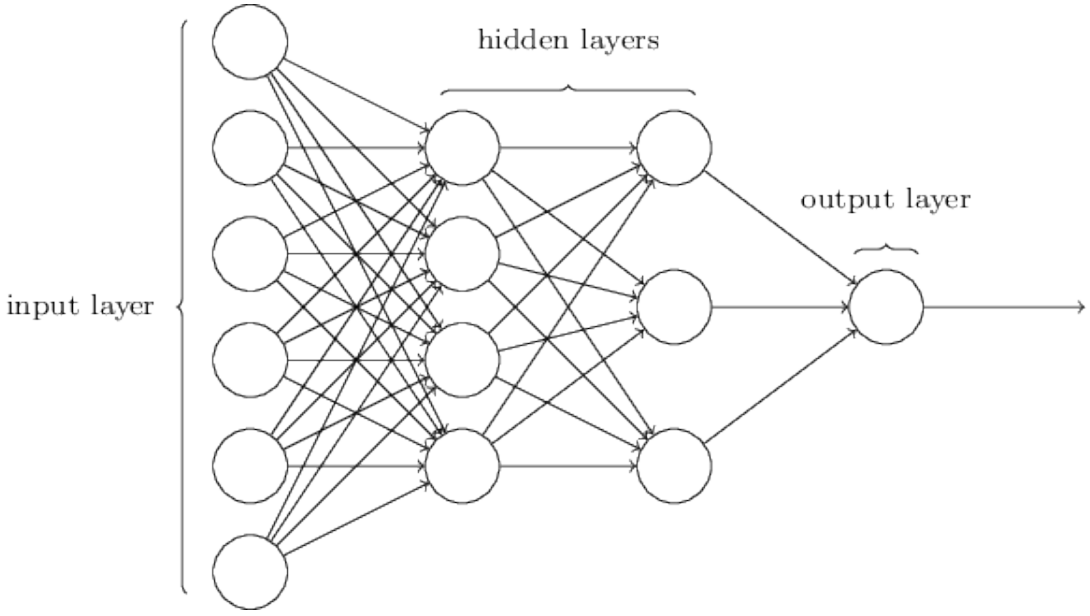

Рис. 2.1: Нейронная сеть со скрытыми слоями.

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

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

Нейрон обычно получает много одновременных входов. Каждый вход имеет свой собственный относительный вес, который дает входное воздействие, которое ему необходимо для функции суммирования элемента обработки. Эти веса выполняют тот же тип функции, что и различные синаптические силы биологических нейронов. В обоих случаях некоторые входы становятся более важными, чем другие, так что они оказывают большее влияние на об-

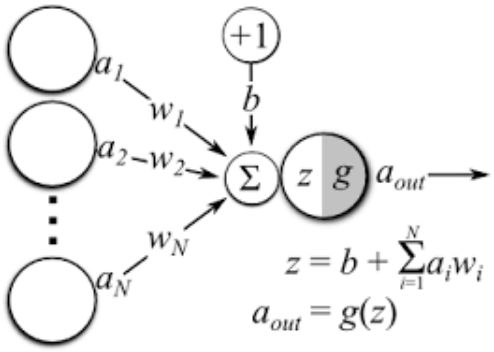

Рис. 2.2: Строение нейрона.

рабатывающий 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 использования функции активации заключается в том, чтобы позволить суммируемому результату меняться в зависимости от времени.Функцией по умолчанию является выпрямленная линейная активационная функция ReLU $g(x) = max(0, x)$,которая 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 любой точностью. Главное чтобы в этой сети было достаточное количество нейронов. И еще важно удачно подобрать начальные значения весов нейронов. Чем удачнее будут подобраны веса, тем быстрее

нейронная сеть будет сходиться к исходной функции. Это означает, что нелинейная характеристика нейрона может быть произвольной: от сигмоидальной до произвольного волнового пакета или вейвлета, синуса или многочлена. От выбора нелинейной функции может зависеть сложность конкретной сети, но с любой нелинейностью сеть остаётся универсальным аппроксиматором и при правильном выборе структуры может достаточно точно аппроксимировать функционирование любой непрерывной функции.

2.2 Методы обучения с подкреплением

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

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

Базовое RL моделируется как процесс принятия марковских решений:

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

- $R_a(s, s')$ - непосредственная награда после перехода от s к s' с действием a.

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

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

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

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

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 машинного обучения 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[2]. Таким образом, получается явный закон управления. Расширение [2] до нелинейных систем не является прямым, и существуют также проблемы сложности вычислений, касающиеся метода из [2].

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 явном виде, обзначим

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

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

- Перепишем функцию стоимости

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

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

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

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

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

Подставим (2.11) в (2.10) : $J(x(t), U) = \frac{1}{2} x^T(t) Y x(t) + \frac{1}{2} U^T H U + x^T(t) F U$

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

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

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

Ограничения можно переписать таким же образом и получим

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

$$\min_U \frac{1}{2} U^T H U + x^T F U + \frac{1}{2} x^T Y x \tag{2.12}$$

при условии $GU \leq W + Ex$

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

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

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

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

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

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 [10] алгоритм обучения представлен дополнительными ограничениями для обеспечения стабильности и ограничения удовлетворения аппроксимационного MPC. Одним из подхо-

дов к аппроксимации MPC является выпуклое многопараметрическое нелинейное программирование [11, 12], где вычисляется субоптимальная аппроксимация закона управления MPC. Другой подход - аппроксимировать MPC с помощью методов машинного обучения. Это делают нейронные сети в [13, 14, 15]. Эти методы не гарантируют устойчивость или удовлетворение ограничениям для аппроксимационного MPC, что особенно важно, если рассматривать жесткие ограничения на состояния. В [5] 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 [16] аппроксимируется MPC с липшицевым сужающим ограничением, что обеспечивает устойчивость при неисчезающих ошибках аппроксимации. Ошибка аппроксимации, выведенная в [5, 16], обычно не достижима для практического применения.

2.3.2 Аппроксимация динамики системы с прогнозирующей моделью

2.3.3 Итерационный подход

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 $\pi_{feas} : X_{feas} \to U | u = \pi_{approx}(x)$. 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.

TODO

Попробовать уменьшить кол-во генерируемых точек для гарантирования аппроксимации

Поппробовать более точно вычислить X_{feas}

Попробовать расспаралеллить получение точек и валидацию аппроксимированной функции нейронной сети

Хотелось бы еще рассмотреть возможность использования unsupervised learning и reinforcement learning

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 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.

Следующими этапами будут исследования по решению указанной проблемы путем переноса некоторых вычислений оффлайн. В частности, для реализации функции MPC-регулятора будут применяться искусственные нейронные сети. Будут рассмотрены алгоритмы обработки больших данных и построения искусственных нейронных сетей для решения данной проблемы.

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 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.

4 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.