

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

Кафедра методов оптимального управления

**МЕТОДЫ УПРАВЛЕНИЯ ПО ПРОГНОЗИРУЮЩЕЙ
МОДЕЛИ НА ОСНОВЕ ДАННЫХ**

Курсовой проект

Кулешова Владислава Вячеславовича
студента 3 курса,
специальность «прикладная
математика»

Научный руководитель:
канд. физ.-мат. наук
доцент Н.М. Дмитрук

Минск, 2020

ОГЛАВЛЕНИЕ

	С.
ВВЕДЕНИЕ	3
ГЛАВА 1 ОСНОВНЫЕ ПОНЯТИЯ И ОБЗОР ЛИТЕРАТУРЫ	4
1.1 Теория управления по прогнозирующей модели.	4
1.2 Задачи оптимального управления	6
1.3 Представление входных и выходных сигналов дискретных стационарных линейных систем на основе матрицы Ганкеля	8
ГЛАВА 2 Управление по прогнозирующей модели на основе данных: точные данные	11
2.1 Основные предположения	11
2.2 Простейшая прогнозирующая задача на основе данных	13
2.3 Задача с терминальными ограничениями-равенствами	14
2.4 Асимптотическая устойчивость замкнутого контура	15
ГЛАВА 3 Управление по прогнозирующей модели на основе данных: неточные данные	17
3.1 Надёжная схема МРС с терминальными ограничениями	17
3.2 n-Шаговая схема управления на основе МРС	18
ГЛАВА 4 Численные эксперименты	20
4.1 Пример.	20
4.2 Программная реализация алгоритмов.	21
4.3 Результаты.	25
ЗАКЛЮЧЕНИЕ	27
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	28

ВВЕДЕНИЕ

В западной литературе управление в реальном времени представлено теорией управления по прогнозирующей модели — Model Predictive Control (МРС). Основными приложениями теории являются задачи стабилизации динамических систем. Современная теория нелинейного МРС предлагает основанные на решении задач оптимального управления методы построения обратных связей для нелинейных объектов.

В главе 1 происходит ознакомление с управлением по прогнозирующей модели, представление модели через матрицу Ганкеля, а также рассматриваются задачи оптимального управления и их классификация. В главе 2 исследуются методы решения простейшей прогнозирующей задачи на основе данных и задачи с терминальными ограничениями-равенствами. В главе 3 применена схема МРС из главы 2, рассмотрена программная реализация алгоритма и приведён полученный результат.

ГЛАВА 1

ОСНОВНЫЕ ПОНЯТИЯ И ОБЗОР ЛИТЕРАТУРЫ

В настоящей главе описываются основные понятия МРС. Приводится классификация принципов управления, используемых в современной теории управления. А также рассматривается возможность представления входных и выходных сигналов дискретных стационарных линейных систем на основе матрицы Ганкеля для МРС.

1.1 Теория управления по прогнозирующей модели

Главная идея МРС [1] – использование математической модели управляемого процесса в пространстве состояний для предсказания и оптимизации будущего поведения системы. Рассмотрим задачу стабилизации нелинейной системы

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

где

$x = x(t) \in \mathbb{R}^n$ – состояние системы в момент времени t ;

$u = u(t) \in \mathbb{R}^r$ – значение управляющего воздействия в момент времени t ;

$f : \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^n$ – заданная функция.

Пусть $f(0, 0) = 0$, следовательно точка равновесия системы находится в начале координат, и при тривиальном управлении $u \equiv 0$ система остаётся в состоянии покоя.

При заданном управлении $u(\cdot)$, траектория системы (1.1) обозначается как $x(t|0, z, u(\cdot))$, $t = 0, 1, \dots$, где начальное состояние системы в момент времени $t = 0$ задаётся условием $x(0) = z$.

Управление $u(\cdot)$ будем выбирать так, чтобы максимально приблизить траектории $x(t|0, x_0, u(\cdot))$, $t = 0, 1, \dots, N$, к началу координат.

Определение 1 Стоимость этапа – функция $l(x(t), u(t))$ вдоль траектории $x(\cdot)$ и управления $u(\cdot)$, с помощью которой для всех моментов времени $t = 0, 1, \dots$ оценивается качество выбранного управления $u(\cdot)$.

Чаще всего стоимость этапа l выбирается следующим образом:

1. Взвешенная сумма расстояний до начала координат:

$l(x, u) = \|x\|^2 + \lambda \|u\|^2$, $\lambda \geq 0$ – параметр, $\|\cdot\|$ – евклидова норма.

2. Квадратичные функции

$$l(x, u) = x'Qx + u'Ru$$

$R, Q > 0$ – положительно-определённые матрицы.

Таким образом задача оптимального управления состоит в минимизации функционала, где минимум ищем вдоль траекторий $x(t|0, x^*(\tau), u(\cdot))$, $t = 0, 1, \dots, N-1$, системы (1.1) с начальным состоянием, совпадающим с текущим состоянием объекта $x(0) = x^*(\tau)$ и при некоторых ограничениях:

$$J(x^*(\tau)) = \min_{u(\cdot)} \sum_{t=0}^{N-1} l(x(t|0, x^*(\tau), u(\cdot)), u(t)). \quad (1.2)$$

Ограничения для задачи (1.2) состоят из двух групп:

1. Физические ограничения системы (например, неотрицательность переменных, максимальное ограничение на управляющее воздействие и другие);
2. Ограничения, накладываемые алгоритмом МРС (например, терминальное ограничение вида $x(N) = 0$ или принадлежность $x(N)$ множеству X_f).

Обозначим оптимальное программное решение задачи (1.2) через $u^0(\cdot|x^*(\tau))$. Для построения обратных связей будем считать, что на объект управления подано первое значение оптимальной программы

$$\mu(x^*(\tau)) = u^0(0|x^*(\tau)).$$

Далее в момент времени $\tau + 1$ процесс повторяется для состояния $x^*(\tau + 1)$. Тогда в этот момент времени решается задача о минимизации следующего функционала:

$$J(x^*(\tau + 1)) = \min_{u(\cdot)} \sum_{t=0}^{N-1} l(x(t|0, x^*(\tau + 1), u(\cdot)), u(t)).$$

При этом будет получено очередное значение обратной связи:

$$\mu(x^*(\tau + 1)) = u^0(0|x^*(\tau + 1)).$$

После процесс повторяется при $\tau + 2, \tau + 3$ и так далее. Таким образом, алгоритм управления по прогнозирующей модели, в каждый момент времени $\tau = 0, 1, \dots$ состоит из следующих шагов:

1. Измеряется текущее состояние $x^*(\tau)$
2. Находится оптимальное программное решение $u^0(t|x^*(\tau))$ задачи (1.2).
3. Подаётся на объект системы управляющее воздействие

$$\mu^*(\tau) \equiv \mu(x^*(\tau)) = \mu^0(0|x^*(\tau)).$$

Идея управления по прогнозирующей модели заключается в оптимизации будущего поведения системы в каждый момент времени, нахождение оптимального управления и его использования в качестве значений обратной связи для следующего момента времени.

1.2 Задачи оптимального управления

Задачи оптимального управления классифицируются [2]

- По промежутку управления:
 - задачи оптимального управления разделяются на непрерывные $T = [t_0, t_f]$ и дискретные, в которых динамический процесс рассматривается в дискретные моменты времени $k = 0, 1, \dots, N$, N – натуральное число.
 - задачи различаются по продолжительности процесса с фиксированным и нефиксированным t_f .
- По ограничениям на траекторию, которые в общем виде имеют следующий вид:

$$x \in \mathbb{X}(t), t \in [t_0, t_f].$$

Ограничения на траекторию могут накладываться:

- на правом конце траектории (терминальные ограничения), т.е. $x(t_f) \in \mathbb{X}_f$;
- на левом конце, т.е. $x(t_0) \in \mathbb{X}_0$;
- в промежуточные моменты времени, т.е. $x(t_i) \in \mathbb{X}_i, t_i \in [t_0, t_f], i = 1, \dots, l$, при этом $t_0 < t_1 < \dots < t_l < t_f$.

Также существуют смешанные ограничения на траекторию.

- По критерию качества:

- Терминальный критерий качества типа Майера

$$J(u) = \varphi(x(t_f)),$$

$$\varphi : R^n \rightarrow R.$$

- Интегральный критерий типа Лагранжа

$$J(u) = \int_{t_0}^{t_f} f_0(x(t), u(t), t) dt,$$

$$f_0 : R^n \times R^r \times R \rightarrow R.$$

- Критерий качества типа Бальса

$$J(u) = \varphi(x(t_f)) + \int_{t_0}^{t_f} f_0(x(t), u(t), t) dt.$$

- Критерий быстродействия

$$J(u) = t_f - t_0 \rightarrow \min.$$

Принцип максимума – классическое необходимое условие оптимальности для задач оптимального управления. Оно является самым сильным из известных необходимых условий оптимальности первого порядка.

Для формулировки принципа максимума введем гамильтониан:

$$H(x, \psi, u, t) = \psi' f(x, u, t) = \sum_{j=1}^n \psi_j f_j(x, u, t)$$

Рассмотрим простейшую задачу оптимального управления на промежутке времени $[t_0, t_f]$ в классе кусочно-непрерывных управлений:

$$J(u) = \phi(x(t_f)) \rightarrow \min,$$

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

$$u(t) \in \mathbb{U}, t \in [t_0, t_f].$$

Пусть $u^0(t), t \in [t_0, t_f]$, – оптимальное управление, $x^0(t), t \in [t_0, t_f]$, – оптимальная траектория, $\psi^0(t), t \in [t_0, t_f]$ – сопряжённая траектория – решение сопряженного уравнения

$$\dot{\psi} = -\frac{\partial H(x^0(t), \psi, u^0(t), t)}{\partial x},$$

$$\psi = -\frac{\partial \varphi(x^0(t_f))}{\partial x}.$$

Тогда выполняется условие максимума гамильтониана:

$$H(x^0(t), \psi^0(t), u^0(t), t) = \max_{u \in \mathbb{U}} H(x^0(t), \psi^0(t), u, t), t \in [t_0, t_f].$$

1.3 Представление входных и выходных сигналов дискретных стационарных линейных систем на основе матрицы Ганкеля

1.3.1 Построение матрицы Ганкеля

Рассмотрим систему [4] с $u(k) \in \mathbb{R}^l, y(k) \in \mathbb{R}^m, x(k) \in \mathbb{R}^n$:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{1.3}$$

где $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times l}, C \in \mathbb{R}^{m \times n}, D \in \mathbb{R}^{m \times l}$ – матрицы системы, $k \in [1, T], T \in \mathbb{N}$. Траектория сигналов $z_d = (z(1), \dots, z(k), \dots, z(T)) \in (\mathbb{R}^{n_z})^T$ линейной многомерной системы P . Система P удовлетворяет $P \subset (\mathbb{R}^{n_z})^T$ с сегментом входа-выхода $z = \begin{bmatrix} u \\ y \end{bmatrix} \in P$, где $u \in (\mathbb{R}^m)^N$ – входной сигнал, $y \in (\mathbb{R}^l)^N$ – выходной. Матрица Ганкеля $H(z_d)$ с N строками и $T - N$ столбцами составленная из ограниченного сигнала $z_d \in (\mathbb{R}^{z_n})^T$ обозначается как:

$$H(z_d) = \begin{pmatrix} z(1) & z(2) & \cdots & z(T - N + 1) \\ z(2) & z(3) & \cdots & z(T - N + 2) \\ \vdots & \vdots & \cdots & \vdots \\ z(N) & z(N + 1) & \cdots & z(T) \end{pmatrix}. \tag{1.4}$$

Построенная матрица будет использоваться в МРС вместо модели без каких-либо промежуточных шагов.

1.3.2 Прогнозирование на основе матрицы Ганкеля

Для произвольного момента времени k , принятого за текущее время, ограничение $P|_T$ поведения P для интервала $[1, T]$ определяется с использованием траектории сигнала как $z_d = (z(1), \dots, z(t), \dots, z(T))$, $z(t) \in \mathbb{R}^{n_z}$. Вектор столбец данных на горизонте управления для выхода и входа z определяется как

$$z_p = z_{k-N|k} = \begin{bmatrix} y(k-N) \\ \vdots \\ y(k) \\ u(k-N) \\ \vdots \\ u(k) \end{bmatrix} \in \mathbb{R}^{(l+m)N}, \quad (1.5)$$

$$z_f = z_{k|k+N-1} = \begin{bmatrix} z(k) \\ z(k+1) \\ \vdots \\ z(k+N-1) \end{bmatrix} \in \mathbb{R}^{(l+m)N_1}. \quad (1.6)$$

Далее, индекс p обозначает относительное «прошлое», а f - «будущее» соответственно. Чтобы избежать промежуточного шага для идентификации представления системы, прогнозирование траектории выполняется с помощью матрицы Ганкеля данных $H(z_d)$. При построении ганкелевой матрицы $H(z_d)$ все данные можно разделить на две части. Прошлый и будущий входной блок ганкелевых матриц определяется следующим образом:

$$U_p = \begin{bmatrix} u(1) & u(2) & \dots & u(j) \\ u(2) & u(3) & \dots & u(j+1) \\ \vdots & \vdots & \ddots & \vdots \\ u(N-1) & u(N) & \dots & u(N+j-1) \end{bmatrix}, \quad (1.7)$$

$$U_f = \begin{bmatrix} u(N) & u(N+1) & \dots & u(N+j-1) \\ u(N+1) & u(N+2) & \dots & u(N+j) \\ \vdots & \vdots & \ddots & \vdots \\ u(2N-1) & u(2N) & \dots & u(2N+j-2) \end{bmatrix}. \quad (1.8)$$

Обозначения «прошлого» и «будущего» можно понять по первому столбцу. Матрица Ганкеля «будущего» блока следует за данными матрицы Ганкеля

«прошлого» блока во временной последовательности. Число N - это размерность строки блоков векторов данных, а j - номер строки матрицы Ханкеля, которая должна быть достаточно большой. Это соотношение справедливо для всех столбцов. Размерность строки U_f может отличаться от размерности U_p , что обеспечит дополнительную степень свободы для настройки предложенного алгоритма прогнозного управления. Прошлый и будущий выходной блок-ганкелевых матриц Y_p и Y_f определяются аналогично. В данной работе используются следующие сокращенные обозначения:

$$H_p = \begin{bmatrix} Y_p \\ U_p \end{bmatrix}, H_f = \begin{bmatrix} Y_f \\ U_f \end{bmatrix}. \quad (1.9)$$

Наблюдения z также можно разделить на две части:

$$z^T = \begin{bmatrix} z_p^T & z_f^T \end{bmatrix}, \quad (1.10)$$

где z_p - известные данные, а z_f - вектор будущих данных. Матрица проекции также может быть разделена на две части таким же образом:

$$z = \begin{bmatrix} z_p \\ z_f \end{bmatrix} = \begin{bmatrix} H_p & H_f \end{bmatrix} g, \quad (1.11)$$

если известная часть данных z_p используется для параметра g оценки \hat{g} , получается следующее соотношение.

$$\hat{g} = (H_p^T H_p)^{-1} H_p^T z_p, \quad (1.12)$$

где $\hat{g} = [g - 0, g_1, \dots, g_r]$ - вектор оцененного вектора изображения. Тогда оценки траекторий всех переменных для предсказания z_f :

$$\hat{z}_f = H_f \hat{g} = H_f (H_p^T H_p)^{-1} H_p^T z_p \quad (1.13)$$

Оценка $\hat{z}_f = [y_f^T \ u_f^T]^T$ с использованием матриц данных H_f , H_p и z_p эффективна для будущих траекторий. Он выполняет ту же роль, что и оценки состояния (Фильтры Калмана и т. Д.) для прогнозирования будущих траекторий в традиционных подходах МРС. Эти методы используют прошлые данные до текущего момента времени для оценки будущих данных. Для управляемых систем можно рассчитать значения будущих траекторий на горизонте управления.

ГЛАВА 2

УПРАВЛЕНИЕ ПО ПРОГНОЗИРУЮЩЕЙ МОДЕЛИ НА ОСНОВЕ ДАННЫХ: ТОЧНЫЕ ДАННЫЕ

В данной главе будут рассматриваться способы построения простейшей прогнозирующей задачи на основе данных задачи с терминальными ограничениями-равенствами.

2.1 Основные предположения

Пусть $\mathbb{I}_{[a,b]}$ – множество целых чисел на отрезке $[a, b]$. Для вектора x и положительно определённой матрицы $P = P^T > 0$ запишем $\|x\|_P = \sqrt{x^T P x}$. Далее определим максимальное и минимальное собственные значения матрицы P $\lambda_{\min}(P)$ и $\lambda_{\max}(P)$.

Для матриц $P_1 = P_1^T, P_2 = P_2^T$ запишем,

$$\lambda_{\min}(P_1, P_2) = \min\{\lambda_{\min}(P_1), \lambda_{\min}(P_2)\},$$

$$\lambda_{\max}(P_1, P_2) = \max\{\lambda_{\max}(P_1), \lambda_{\max}(P_2)\}.$$

Также $\|x\|_2, \|x\|_1, \|x\|_\infty$ – Евклидовы, ℓ_1 и ℓ_∞ нормы x , соответственно. Для $\delta > 0$, мы определим $\mathbb{B}_\delta = \{x \in \mathbb{R}^n \mid \|x\|_2 \leq \delta\}$.

Из последовательности $\{x_k\}_{k=0}^{N-1}$ составим матрицу Ганкеля

$$H_L(x) = \begin{bmatrix} x_0 & x_1 & \dots & x_{N-L} \\ x_1 & x_2 & \dots & x_{N-L+1} \\ \vdots & & \ddots & \vdots \\ x_{L-1} & x_L & \dots & x_{N-1} \end{bmatrix},$$

$$x_{[a,b]} = \begin{bmatrix} x_a \\ \vdots \\ x_b \end{bmatrix}$$

Определение 2 $\{x_k\}_{k=0}^{N-1}$ с $x_k \in \mathbb{R}^n$ постоянно возбуждающая порядка L , если $\text{rank}(H_L(x)) = nL$.

Цель паботы – управление неизвестной линейной стационарной системой G с порядком n , с m входами и p выходами.

Определение 3 Последовательность $\{u_k, y_k\}_{k=0}^{N-1}$ – траектория линейной стационарной системы G , если существуют начальное состояние $\bar{x} \in \mathbb{R}^n$, а также соответствующая траектория $\{x_k\}_{k=0}^N$ такая, что

$$x_{k+1} = Ax_{k+1} + Bu_k; \quad x_0 = \bar{x},$$

$$y_k = Cx_k + Du_k, \quad k = 0, \dots, N-1,$$

(A, B, C, D) – минимальные реализация G .

Теорема 2.1 Предположим, что $\{u_k^d, y_k^d\}_{k=0}^{N-1}$ – траектория линейной стационарной системы G , где u – постоянно возбуждающее управление порядка $L + n$.

Тогда $\{\bar{u}_k, \bar{y}_k\}_{k=0}^{L-1}$ – траектория G , тогда и только тогда, когда $\exists \alpha \in \mathbb{R}^{N-L+1}$ такое, что

$$\begin{bmatrix} H_L(u^d) \\ H_L(y^d) \end{bmatrix} \alpha = \begin{bmatrix} \bar{u} \\ \bar{y} \end{bmatrix}. \quad (2.1)$$

Определение 4 Пара $(u^s, y^s) \in \mathbb{R}^{m+p}$ – положение равновесия линейной стационарной системой G , если последовательность $\{\bar{u}_k, \bar{y}_k\}_{k=0}^{n-1}$ с $(\bar{u}_k, \bar{y}_k) = (u^s, y^s) \forall k \in \mathbb{I}[0, n-1]$ – траектория G .

Для равновесия (u^s, y^s) мы определим u_n^s, y_n^s как столбец векторов содержащий n раз u^s и y^s , соответственно. Предположим, что система подчиняется ограничениям, $u_t \in \mathbb{U} \subseteq \mathbb{R}^m, y_t \in \mathbb{Y} \subseteq \mathbb{R}^p \forall t \geq 0$, и предположим, что $(u^s, y^s) \in \text{int}(\mathbb{U} \times \mathbb{Y})$. $\{u_k^d, y_k^d\}_{k=0}^{N-1}$ – априори измеряемые траектории длиной N , использующиеся в (2.1). Прогнозируемые входные и выходные траектории в момент времени t в течение некоторого горизонта прогнозирования L записываются как $\{\bar{u}_k(t), \bar{y}_k(t)\}_{k=-n}^{L-1}$. Обратим внимание, что индексы времени начинаются с $k = -n$, так как последние n входов и выходов будут использоваться для вызова уникального начального состояния в момент времени t . Кроме того, вход с обратной связью, состояние в некоторой минимальной реализации и выход в момент времени t обозначаются как u_t, x_t, y_t , соответственно.

2.2 Простейшая прогнозирующая задача на основе данных

Теорема 2.1 обеспечивает привлекательную альтернативу математической модели в пространстве состояний, так как (2.1) достаточно, чтобы охватить все траектории системы. Таким образом, для реализации схемы МРС, основанной на данных, можно просто заменить динамику системы ограничением, которому удовлетворяют прогнозируемые траектории ввода-вывода (2.1). Чтобы быть более точным, предложенная управляемая данными схема МРС решает задачу

$$J_L(u_{[t-n,t-1]}, y_{[t-n,t-1]}, \alpha(t)) = \sum_{k=0}^{L-1} \ell(\bar{u}_k(t), \bar{y}_k(t)), \quad (2.2)$$

$$\begin{bmatrix} \bar{u}_{[-n,-L-1]}(t) \\ \bar{y}_{[-n,-L-1]}(t) \end{bmatrix} = \begin{bmatrix} H_{L+n}(u^d) \\ H_{L+n}(y^d) \end{bmatrix}, \quad (2.3)$$

$$\begin{bmatrix} \bar{u}_{[-n,-1]}(t) \\ \bar{y}_{[-n,-1]}(t) \end{bmatrix} = \begin{bmatrix} u_{[t-n,t-1]} \\ y_{[t-n,t-1]} \end{bmatrix}, \quad (2.4)$$

$$\ell(\bar{u}, \bar{y}) = \|\bar{u} - u^s\|_R^2 + \|\bar{y} - y^s\|_Q^2, \quad Q, R \succ 0. \quad (2.5)$$

Ограничение (2.3) заменяет динамику системы по сравнению с классическими схемами МРС на основе моделей. Кроме того, (2.4) гарантирует, что внутреннее состояние истинной траектории совпадает с внутренним состоянием прогнозируемой траектории в момент времени t . Начальные траектории указываются до временного шага $t - 1$, поскольку входной сигнал в момент времени t может уже влиять на выходной сигнал в момент времени t в случае проходного элемента установки. Стоимость (2.2) зависит только от переменной оптимизации $\alpha(t)$, поскольку $\bar{u}(t)$ и $\bar{y}(t)$ неявно фиксируются через динамическое ограничение (2.3).

2.3 Задача с терминальными ограничениями-равенствами

В этом пункте рассмотрим простое терминальное ограничение, которое может быть включено непосредственно в структуру управления данными МРС.

$$J_L^*(u_{[t-n,t-1]}, y_{[t-n,t-1]}) = \min_{\alpha(t)} \sum_{k=0}^{L-1} \ell(\bar{u}_k(t), \bar{y}_k(t)) \quad (2.6)$$

$$\begin{bmatrix} \bar{u}_{[-n,-L-1]}(t) \\ \bar{y}_{[-n,-L-1]}(t) \end{bmatrix} = \begin{bmatrix} H_{L+n}(u^d) \\ H_{L+n}(y^d) \end{bmatrix} \alpha(t), \quad (2.7)$$

$$\begin{bmatrix} \bar{u}_{[-n,-1]}(t) \\ \bar{y}_{[-n,-1]}(t) \end{bmatrix} = \begin{bmatrix} u_{[t-n,t-1]} \\ y_{[t-n,t-1]} \end{bmatrix}, \quad (2.8)$$

$$\begin{bmatrix} \bar{u}_{[L-n,L-1]}(t) \\ \bar{y}_{[L-n,L-1]}(t) \end{bmatrix} = \begin{bmatrix} u_n^s \\ y_n^s \end{bmatrix}, \quad (2.9)$$

$$\bar{u}_k(t) \in \mathbb{U}, \bar{y}_k(t) \in \mathbb{Y}, k \in \mathbb{I}_{[0,L-1]}. \quad (2.10)$$

Терминальное ограничение-равенство (2.9) подразумевает, что $\bar{x}_L(t)$, который является внутренним состоянием, предсказанным на L шагов вперед, соответствующей предсказанной траекторией, выравнивается с постоянным состоянием x^s , соответствующим (u^s, y^s) , то есть $\bar{x}_L(t) = x^s$ в любой минимальной реализации. В то время как задача требует, чтобы (u^s, y^s) было равновесием неизвестной системы по определению (4), это требование может быть отброшено, когда (u^s, y^s) заменено искусственным равновесием, которое также оптимизируется онлайн. Расширение представленной схемы МРС до такой настройки является предметом будущей работы. Как и в стандартном МРС, задача решена в виде отступающего горизонта, который обобщен в алгоритме (1).

Алгоритм 1 (Схема управления на основе МРС)

1. В момент времени t взять прошлые n измерений $u_{[t-n,t-1]}, y_{[t-n,t-1]}$ и решить задачу.

2. Взять за управление $u_t = \bar{u}_0^*(t)$
3. Установить $t = t + 1$ и вернуться к пункту 1).

2.4 Асимптотическая устойчивость замкнутого контура

Для анализа предположим, что $u^s = 0, y^s = 0$ и, следовательно, $x^s = 0$. Далее определим множество начальных состояний, для которых выполнимо (2.6 – 2.9), $x_L = \{x \in \mathbb{R}^n | J_L^*(x) < \infty\}$. Для доказательства экспоненциальной устойчивости предложенной схемы предположим, что функция оптимальных значения (2.6 – 2.9) квадратично ограничена сверху.

Предположение 1 Функция оптимальных значения $J *_{L}(x)$ - квадратично ограничена сверху на \mathbb{X}_L , т.е. $\exists c_u > 0$ такое, что $J_L^*(x) \leq c_u \|x\|_2^2 \forall x \in \mathbb{X}_L$

Теорема 2.2 Пусть предположение (1) верно, $L \geq, \{u_k^d, y_k^d\}_{k=0}^{N-1}$ – траектория линейной стационарной системы G , где u^d – постоянно возбуждающий порядка $L + 2n$. Если задача МРС (2.6 – 2.9) возможно в начальный момент времени $t = 0$:

1. это возможно $\forall t$,
2. замкнутый цикл удовлетворяет ограничениям $u_t \in \mathbb{U}, y_t \in \mathbb{Y} \forall t \in \mathbb{N}$,
3. равновесие $x^s = 0$ экспоненциально устойчиво для полученного замкнутого цикла.

Доказательство (Экспоненциальная устойчивость)
Цена в момент времени $t+1$

$$J_L(x_{t+1}, \alpha'(t+1)) = \sum_{k=0}^{L-1} \ell(\bar{u}'_k(t+1), \bar{y}'_k(t+1)) = \sum_{k=0}^{L-1} \ell(\bar{u}^*_k(t), \bar{y}^*_k(t)) = J_L^*(x_t) - \ell(\bar{u}_0^*(t), \bar{y}_0^*(t)).$$

Откуда следует, что

$$J_L^*(x_{t+1}) \leq J_L^*(x_t) - \ell(\bar{u}_0^*(t), \bar{y}_0^*(t)) \quad (2.11)$$

Поскольку \bar{u}_0^* – состояние наблюдаемой (и, следовательно, обнаруживаемой) минимальной реализации, существует матрица $P \succ 0$ такая, что $W(x) = \|x\|_P^2$ – функция стабильности Ляпунова и удовлетворяет

$$W(Ax + Bu) - W(x) \geq -\frac{1}{2}\|x\|_2^2 + c_1\|u\|_2^2 + c_2\|y\|_2^2, \quad c_1, c_2 > 0. \quad (2.12)$$

где $\forall x \in \mathbb{R}^n, u \in \mathbb{R}^m, y = Cx + Du$. Определим функцию Ляпунова $V(x) = \gamma W(x) + J_L^*(x)$ при некоторой $\gamma > 0$. Т.к. V квадратично ограничена снизу, $V(x) \geq \gamma W(x) \geq \gamma \lambda_{\min}(P)\|x\|_2^2 \quad \forall x \in \mathbb{X}_L$. По предположению (1) J_L^* квадратично ограничена сверху, т.е. $J_L^*(x) \leq c_u\|x\|_2^2 \quad \forall x \in \mathbb{X}_L$.

$$V(x) = J *_{\mathbb{X}_L}(x) + \gamma W(x) \leq (c_u + \gamma \lambda_{\max}(P))\|x\|_2^2 \quad \forall x \in \mathbb{X}_L,$$

т.е. V – квадратично ограничена сверху.

Вдоль траекторий с замкнутым контуром, используя (2.11) и (2.12), получим

$$V(x_{t+1}) - V(x_t) \leq \gamma(-\frac{1}{2}\|x_t\|_2^2 + c_1\|u_t\|_2^2 + c_2\|y_t\|_2^2) - \|u_t\|_R^2 - \|y_t\|_Q^2 \leq -\frac{\gamma}{2}\|x_t\|_2^2$$

Таким образом, V затухает экспоненциально вдоль траекторий замкнутого контура, он квадратично ограничен сверху и снизу и удовлетворяет условию $V(0) = 0$. Следовательно, из стандартных аргументов Ляпунова следует, что равновесие $x^s = 0$ экспоненциально устойчиво.

Было показано, что замкнутый контур в n -ступенчатой схеме управления по прогнозирующей модели рекурсивно выполним и практически экспоненциально стабилен, следовательно, он является и асимптотически устойчивым.

ГЛАВА 3

УПРАВЛЕНИЕ ПО ПРОГНОЗИРУЮЩЕЙ МОДЕЛИ НА ОСНОВЕ ДАННЫХ: НЕТОЧНЫЕ ДАННЫЕ

Прямое применение схемы МРС из Главы 2 может привести к проблемам с осуществимостью или может сделать замкнутый контур нестабильным. В этой главе мы решаем проблему зашумленных измерений с помощью надежной схемы МРС с терминальными ограничениями.

3.1 Надёжная схема МРС с терминальными ограничениями

Ключевой идеей для учета зашумленных измерений является ослабление ограничения равенства (2.7). При зашумленной начальной траектории ввода-вывода $(u_{[t-n,t-1]}, \tilde{y}_{[t-n,t-1]})$ длины n , предлагаем следующую надежную модификацию (2.6-2.10).

$$J_L^*(u_{[t-n,t-1]}, \tilde{y}_{[t-n,t-1]}) = \min_{\alpha(t), \bar{u}(t), \bar{y}(t), \sigma(t)} \sum_{k=0}^{L-1} \ell(\bar{u}_k(t), \bar{y}_k(t)) + \lambda_\alpha \bar{\varepsilon} \|\alpha(t)\|_2^2 + \lambda_\sigma \|\sigma(t)\|_2^2 \quad (3.1)$$

$$\begin{bmatrix} \bar{u}(t) \\ \bar{y}(t) + \sigma(t) \end{bmatrix} = \begin{bmatrix} H_{L+n}(u^d) \\ H_{L+n}(\tilde{y}^d) \end{bmatrix} \alpha(t), \quad (3.2)$$

$$\begin{bmatrix} \bar{u}_{[-n,-1]}(t) \\ \bar{y}_{[-n,-1]}(t) \end{bmatrix} = \begin{bmatrix} u_{[t-n,t-1]} \\ y_{[t-n,t-1]} \end{bmatrix}, \quad (3.3)$$

$$\begin{bmatrix} \bar{u}_{[L-n, L-1]}(t) \\ \bar{y}_{[L-n, L-1]}(t) \end{bmatrix} = \begin{bmatrix} u_n^s \\ y_n^s \end{bmatrix}, \bar{u}_k \in \mathbb{U} \quad (3.4)$$

$$\|\sigma_k(t)\|_\infty \leq \bar{\varepsilon}(1 + \|\alpha(t)\|_1), k \in \mathbb{I}[0, L-1] \quad (3.5)$$

По сравнению с задачей (2.6-2.10) траектория выходных данных \tilde{y}^d и начальный выходной сигнал $\tilde{y}_{[t-n, t-1]}$, полученные из онлайн измерений, были заменены их зашумленными аналогами. Также были добавлены следующие элементы:

1. Переменная слабину σ для учёта онлайн измерений $\tilde{y}_{[t-n, t-1]}$ и для зашумленных данных \tilde{y}^d , используемых для предсказаний.
2. Квадратичная норма для σ и α с весами $\lambda_\alpha \bar{\varepsilon}, \lambda_\sigma > 0$, норма α зависит от уровня шума.

3.2 n-Шаговая схема управления на основе MPC

В этой главе мы рассматриваем замкнутый цикл, полученный в результате применения (3.1-3.5) в n-шаговой схеме MPC. Чтобы быть более точным, мы рассматриваем сценарий, в котором сразу после решения (3.1-3.5) первые n вычисленных входных сигналов применяются к системе. После этого горизонт сдвигается на n шагов, прежде чем вся схема повторяется.

Алгоритм 2 (n-Шаговая схема управления на основе MPC)

1. В момент времени t взять прошлые n измерений $u_{[t-n, t-1]}, \tilde{y}_{[t-n, t-1]}$ и решить задачу (3.1-3.5).
2. Взять за управление $u_{[t, t+n-1]} = \bar{u}_{[t, n-1]}^*(t)$
3. Установить $t = t + n$ и вернуться к пункту 1).

(3.1-3.5) - это строго выпуклая квадратичная задача и она может быть эффективно решена. Однако ограничение на σ в (3.5) невыпукло из-за зависимости правой части от $\|\alpha(t)\|_1$, что затрудняет эффективную реализацию (3.1-3.5). (3.5) требуется для доказательства рекурсивной выполнимости и практической экспоненциальной устойчивости. Однако его можно заменить (выпуклым) ограничением $\|\sigma_k(t)\|_\infty \leq c \cdot \bar{\varepsilon}$ для достаточно большой постоянной $c > 0$ с сохранением тех же теоретических гарантий. Как правило, больший выбор c увеличивает область притяжения, но также увеличивает размер экспоненциально устойчивого множества, к которому сходится замкнутый контур. Кроме того, ограничение (3.5) может быть реализовано неявно, выбирая λ_σ достаточно большим. В примерах моделирования было замечено, что ограничение (3.5) обычно выполняется (для достаточно большого выбора λ_σ), не применяя его явно в задаче оптимизации, и, таким образом, в большинстве случаев им можно пренебречь при онлайн-оптимизации.

ГЛАВА 4

ЧИСЛИННЫЕ ЭКСПЕРЕМЕНТЫ

В этой главе применим схемы МРС из глав 2 и 3 и приведём результаты, полученные программными реализациями алгоритмов 1 и 2.

4.1 Пример

Рассмотрим пример, который описывает систему из 4 сообщающихся резервуаров [3]. Это хорошо изученная в теории управления и в частности в теории МРС система. Она описывает достаточно медленный и устойчивый динамический процесс. Однако, известно также, что при выборе короткого горизонта управления система может стать неустойчивой. Линеаризованная динамика системы в дискретном времени описывается уравнениями

$$x_{k+1} = \begin{bmatrix} 0.921 & 0 & 0.041 & 0 \\ 0 & 0.918 & 0 & 0.033 \\ 0 & 0 & 0.924 & 0 \\ 0 & 0 & 0 & 0.937 \end{bmatrix} x_k + \begin{bmatrix} 0.017 & 0.001 \\ 0.001 & 0.023 \\ 0 & 0.061 \\ 0.072 & 0 \end{bmatrix} u_k,$$
$$y_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_k.$$

Целью управления будет являться отслеживание заданного значения си-

СТЕМЫ

$$(u^s, y^s) = \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.65 \\ 0.77 \end{bmatrix} \right).$$

Предположим, что системные матрицы неизвестны, но доступна одна траектория входа-выхода $\{u_k^d, y_k^d\}_{k=0}^{N-1}$ длины $N = 400$, которая генерируется путем равномерной выборки u_k^d из $[-1, 1]^2$. Горизонт прогнозирования установим в $L = 30$, и следующие параметры $Q = 3 \cdot E_2$, $R = 10^{-4} \cdot E_2$, $\lambda_\sigma = 1000$, $\lambda_\alpha \bar{\varepsilon} = 0.1$.

4.2 Программная реализация алгоритмов

Рассмотрим способ реализации алгоритмов (1-2) на языке Matlab. Сперва зададим матрицу и вектор ограничений-равенств для quadprog.

Для алгоритма (1):

```
for i = 1:4:(L + n) * 4
    Aeq(i, i) = 1;
    Aeq(i + 1, i + 1) = 1;
    Aeq(i + 2, i + 2) = 1;
    Aeq(i + 3, i + 3) = 1;
    hIndex = fix(i / 4) + 1;
    Aeq(i, alphaRange) = -uHankel(hIndex, :, 1);
    Aeq(i + 1, alphaRange) = -uHankel(hIndex, :, 2);
    Aeq(i + 2, alphaRange) = -yHankel(hIndex, :, 1);
    Aeq(i + 3, alphaRange) = -yHankel(hIndex, :, 2);
end
j = 0;
for i = (L + n) * 4 + 1: numel(Aeq(:, 1))
    Aeq(i, steadyFirstIndex + j) = 1;
    j = j + 1;
end
beq(1, (L + n) * 4 + 1:(L + n) * 4 + (L * 4)) = 1;
```

Для алгоритма (2):

```
for i = 1:4:(L + n) * 4
    Aeq(i, i) = 1;
    Aeq(i + 1, i + 1) = 1;
```

```

Aeq(i + 2, i + 2) = 1;
Aeq(i + 3, i + 3) = 1;
hIndex = fix(i / 4) + 1;
Aeq(i, alphaRange) = -uHankel(hIndex, :, 1);
Aeq(i + 1, alphaRange) = -uHankel(hIndex, :, 2);
Aeq(i + 2, alphaRange) = -yHankel(hIndex, :, 1);
Aeq(i + 3, alphaRange) = -yHankel(hIndex, :, 2);

Aeq(i + 2, sigmaShift + sigmaIndex) = 1;
Aeq(i + 3, sigmaShift + sigmaIndex + 1) = 1;

sigmaIndex = sigmaIndex + 2;
end

```

Сведём задачу к минимизации функции

$$\min_x \frac{1}{2} x^T H x + f^T,$$

где H и f задаются следующим образом.

Для алгоритма (1):

```

quadH = zeros(((L + n) * 4) + (N - (L + n) + 1) + (L * 4));
quadF = zeros(1, ((L + n) * 4) + (N - (L + n) + 1) + (L * 4));
for i = n * 4 + 1:4:(L + n) * 4
    quadH(i, i) = coeffR;
    quadH(i + 1, i + 1) = coeffR;
    quadH(i + 2, i + 2) = coeffQ;
    quadH(i + 3, i + 3) = coeffQ;
    for j = steadyFirstIndex:4:numel(quadH(1, :))
        quadH(i, j) = -2 * uSteady(1) * coeffR;
        quadH(i + 1, j + 1) = -2 * uSteady(2) * coeffR;
        quadH(i + 2, j + 2) = -2 * ySteady(1) * coeffQ;
        quadH(i + 3, j + 3) = -2 * ySteady(2) * coeffQ;
    end
end
squares = L * 4 * ...
    (coeffR * uSteady(1)^2 + coeffR * uSteady(2)^2 ...
    + coeffQ * ySteady(1)^2 + coeffQ * ySteady(2)^2);
quadF(1, steadyFirstIndex) = squares;
quadH = quadH * 2;

```

Для алгоритма (2):

```

quadH = zeros(((L + n) * 4) + (N - (L + n) + 1) + (L * 4) + (L + n) * 2);
quadF = zeros(1, ((L + n) * 4) + (N - (L + n) + 1) + (L * 4) + (L + n) * 2);

```

```

for i = alphaShift+1:alphaRange(end)
    quadH(i,i)=lambdaAlphaEps;
end
for i = sigmaShift+1:sigmaRange(end)
    quadH(i,i)=lambdaSigma;
end
for i = n * 4 + 1:4:(L + n) * 4
    quadH(i, i) = coeffR;
    quadH(i + 1, i + 1) = coeffR;
    quadH(i + 2, i + 2) = coeffQ;
    quadH(i + 3, i + 3) = coeffQ;
    for j = steadyFirstIndex:4:numel(quadH(1, :))
        quadH(i, j) = -2 * uSteady(1) * coeffR;
        quadH(i + 1, j + 1) = -2 * uSteady(2) * coeffR;
        quadH(i + 2, j + 2) = -2 * ySteady(1) * coeffQ;
        quadH(i + 3, j + 3) = -2 * ySteady(2) * coeffQ;
    end
end
squares = L * 4 * ...
    (coeffR * uSteady(1)^2 + coeffR * uSteady(2)^2 ...
    + coeffQ * ySteady(1)^2 + coeffQ * ySteady(2)^2);
quadF(1, steadyFirstIndex) = squares;
quadH = quadH * 2;

```

После начнём итерационный процесс от 0 до N . В начале которого будем вычислять вектор ограничений.

Для алгоритма (1):

```

for i = 1:4:L * 4
    beq(i) = uRes(j, 1);
    beq(i + 1) = uRes(j, 2);
    beq(i + 2) = yRes(j, 1);
    beq(i + 3) = yRes(j, 2);
    j = j + 1;
end

```

Для алгоритма (2):

```

Aeq(i + 2, sigmaRange) = 0;
Aeq(i + 3, sigmaRange) = 0;

beq(i) = uRes(j, 1);
beq(i + 1) = uRes(j, 2);
beq(i + 2) = yRes(j, 1);
beq(i + 3) = yRes(j, 2);
j = j + 1;

```

Зададим терминальные ограничения.

Для алгоритма (1):

```
for i = L * 4 + 1:4:(L + n) * 4
    beq(i) = uSteady(1, 1);
    beq(i + 1) = uSteady(2, 1);
    beq(i + 2) = ySteady(1, 1);
    beq(i + 3) = ySteady(2, 1);
end
```

Для алгоритма (2):

```
for i = L * 4 + 1:4:(L + n) * 4
    Aeq(i + 2, sigmaRange) = 0;
    Aeq(i + 3, sigmaRange) = 0;

    beq(i) = uSteady(1, 1);
    beq(i + 1) = uSteady(2, 1);
    beq(i + 2) = ySteady(1, 1);
    beq(i + 3) = ySteady(2, 1);
end
```

Решим задачи (2.6 - 2.9) и (3.1 - 3.5) с помощью quadprog.

```
options = optimoptions('quadprog', ...
    'MaxIter', 10000, ...
    'TolFun', 1e-15, ...
    'TolX', 1e-15);
[res, value] = quadprog(quadH, quadF, [], [], ...
    Aeq, beq, [], [], [], options);
```

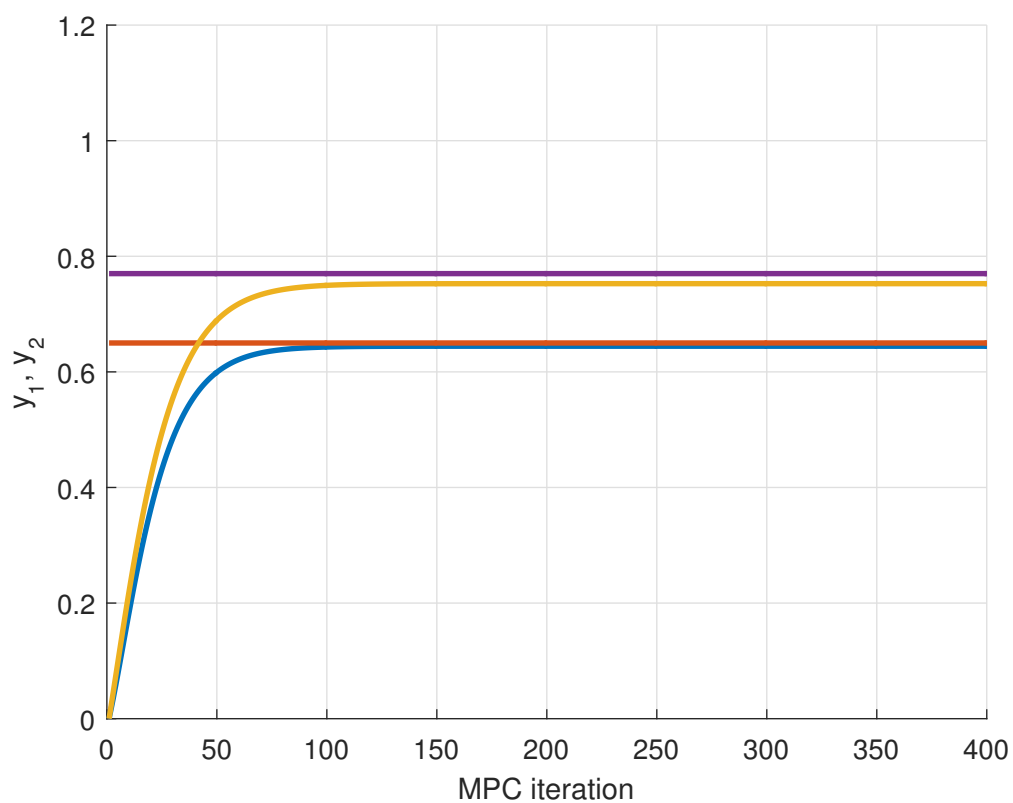
И запишем результат.

```
j = timeIndex - n;
for i = 1:4:(L + n) * 4
    uRes(j, 1) = res(i);
    uRes(j, 2) = res(i + 1);
    [dsResY, dsResX] = dynamicSystemFunc(uRes(j,:) ', xRes(j,:) ', A, B, C);
    yRes(j,:) = dsResY;
    j = j + 1;
    xRes(j,:) = dsResX';
end
```

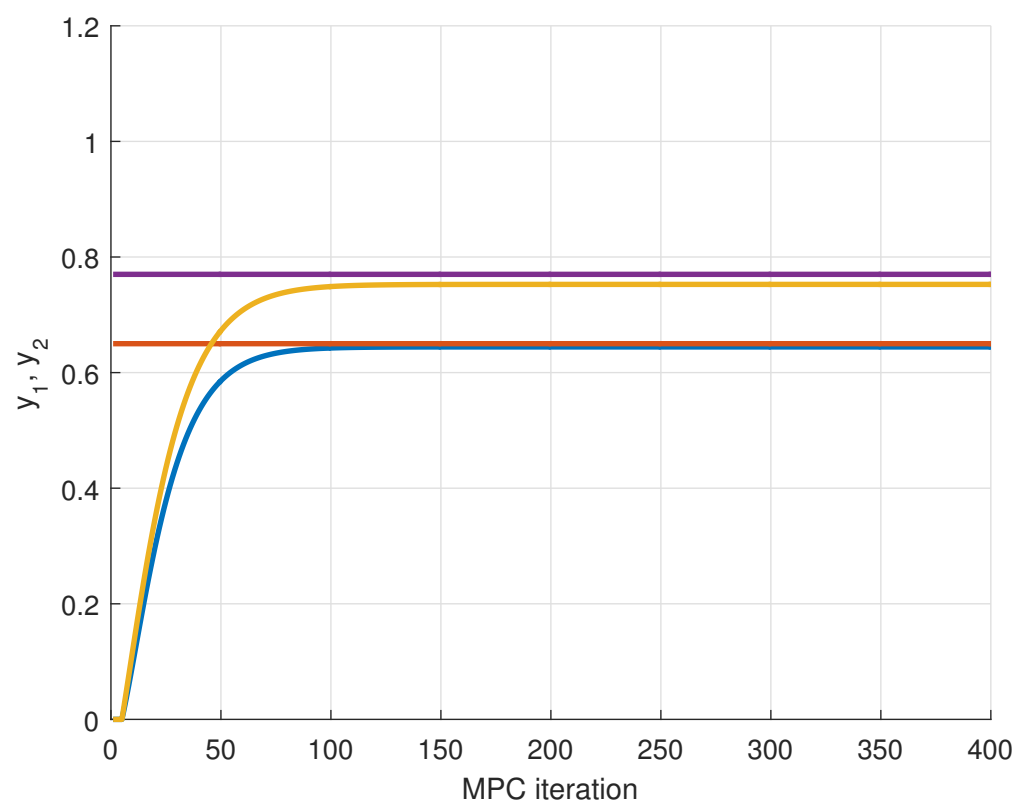

4.3 Результаты

Ниже приведены графики результатов программ, описанных в прошлом пункте. На которых прямые – соответствующие значения y_1^s , y_2^s .

Результаты программной реализации алгоритма (1):



Результаты программной реализации алгоритма (2):



ЗАКЛЮЧЕНИЕ

В данной работе описаны основные определения теории управления по прогнозирующей модели, проанализированы схемы управления по прогнозирующей модели и показаны способы решений простейшей прогнозирующей задачи и задачи с терминальными ограничениями-равенствами, в которых для прогнозирования используются только прошлые измеренные данные без какого-либо предварительного шага идентификации системы. Также показано, что замкнутый контур в схеме управления по прогнозирующей модели рекурсивно допустим и практически экспоненциально устойчив.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Rawlings, J.B. Model Predictive Control: Theory and Design / J.B. Rawlings, D.Q. Mayne. – Madison: Nob Hill Publishing, 2009. – 576 p.
- 2 Методы оптимизации. Учебное пособие / В.В. Альсевич [и др.] – Мн.: «Четыре четверти», 2011.
- 3 Berberich, J. A trajectory-based framework for data-driven system analysis and control / J. Berberich, F. Allgöwer // arXiv, 2019 – 6p. (preprint arXiv:1903.10723).
- 4 Yang, H. A data-driven predictive controller design based on reduced hankel matrix / H. Yang, S. Li // Asian Control Conference: proc. of the 10th ACC – 2015. – 7p.
- 5 Coulson, J. Data-enabled predictive control: in the shallows of the DeePC / J.Coulson, J. Lygeros, F. Dörfler // European Control Conference: proc of the 18thECC. – 2019. – P. 307-312.