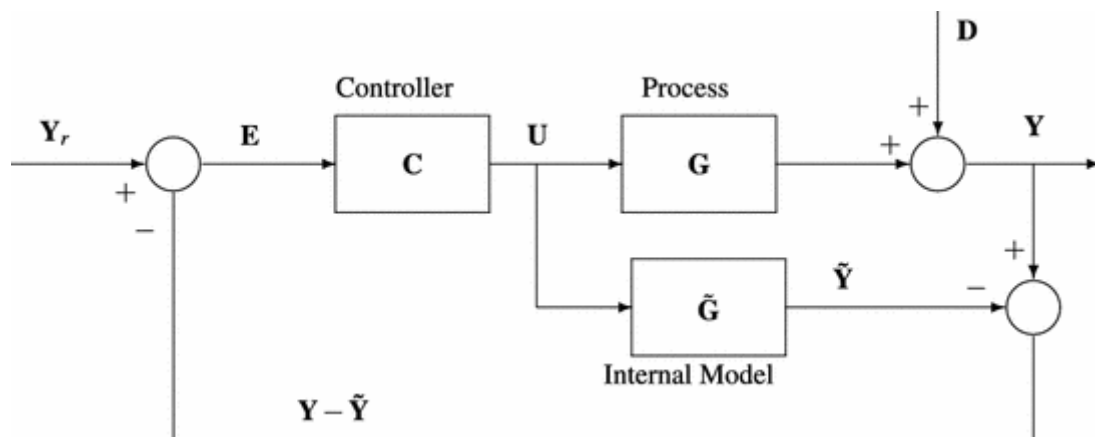


А. А. Жиленков

МОДЕЛИРОВАНИЕ СИСТЕМ И КОМПЛЕКСОВ:
Дискретные системы прогностического управления в теории, задачах
и примерах в MATLAB



Санкт-Петербург

2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
УНИВЕРСИТЕТ ИТМО

А.А. Жиленков

**МОДЕЛИРОВАНИЕ СИСТЕМ И КОМПЛЕКСОВ:
Дискретные системы прогностического управления в
теории, задачах и примерах в MATLAB**

УЧЕБНОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлению подготовки 15.04.06, 27.04.03, 27.04.04
в качестве учебного пособия для реализации основных
профессиональных образовательных программ высшего образования
магистратуры

 УНИВЕРСИТЕТ ИТМО

Санкт-Петербург
2019

Жиленков А. А., Моделирование систем и комплексов. Дискретные системы прогностического управления в теории, задачах и примерах в MATLAB. Учебное пособие. — СПб.: Университет ИТМО, 2019. — 79 с.

Рецензент:

Кремлев Артем Сергеевич, канд. техн. наук, доцент (квалификационная категория "ординарный доцент") факультета систем управления и робототехники Университета ИТМО.

В учебном пособии рассматриваются общие сведения о непрерывных и дискретных системах управления, построенных на базе прогнозирующих моделей объектов управления, подходы к их синтезу, расчёту и моделированию. Приводятся поясняющие примеры и упражнения, демонстрирующие возможности применения рассмотренных методов расчёта и моделирования.

Для студентов технических вузов, обучающихся по направлениям магистратуры 15.04.06 "Мехатроника и робототехника", 27.04.03 "Системный анализ и управление", 27.04.04 "Управление в технических системах".



Университет ИТМО — ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2019
© Жиленков А.А., 2019

ОГЛАВЛЕНИЕ

Введение	4
1 Дискретные системы управления с прогнозирующей моделью	7
1.1 Модели пространства состояний	7
1.2 Система с одним входом и одним выходом (SISO)	7
1.3 Пример проектирования расширенной модели в MATLAB	9
1.4 Задания для самостоятельной работы	11
2 Прогностическое управление в одном окне оптимизации	12
2.1 Прогнозирование переменных состояния и выхода	12
2.2 Оптимизация	14
2.3 Синтез прогностических моделей в MATLAB	21
2.4 Задания для самостоятельной работы	23
3 Управление с удаляющимся горизонтом прогнозирования	24
3.1 Система управления с замкнутым контуром	25
3.2 Моделирование систем управления с удаляющимся горизонтом в MATLAB	31
3.3 Задания для самостоятельной работы	34
4 Управление с прогнозирующей моделью в МИМО-системах	37
4.1 Общее описание модели	37
4.2 Собственные значения расширенной модели	38
4.3 Управляемость и наблюдаемость расширенной модели	38
4.4 Расчёт прогнозирующего управления для МИМО-систем	42
4.5 Наблюдатель состояния	43
4.6 Основы теории наблюдателей	44
4.7 Определение наблюдаемости	48
4.8 Фильтр Калмана	51
4.9 Настройка динамики наблюдателя	52
4.10 Прогнозирующее управление с оценкой состояния в MATLAB	53
4.11 Задания для самостоятельной работы	57
ПРИЛОЖЕНИЕ	61
Литература	70

ВВЕДЕНИЕ

Существует три основных подхода к прогностическому управлению, каждый из которых использует уникальную структуру прогнозирующей модели. Ранее, при моделировании систем прогнозирующего управления на базе внутренней модели (СУПМ), исследователи отдавали предпочтение фильтрам с конечной импульсной характеристикой (КИХ) и моделям с импульсным откликом (ИО).

Алгоритмы расчета, основанные на модели КИХ/ИО, включают динамическое матричное управление (ДМУ) Кутлера и Рамакера [1] и квадратичную формулировку ДМУ Гарсия и Моршеди [2]. КИХ-модели привлекательны для инженеров-технологов, поскольку структура модели дает прозрачное описание времени запаздывания процесса в объекте, времени отклика, усиления. Тем не менее, сфера их применения ограничена устойчивыми объектами и часто требует моделей высоких порядков. Структура подобной модели как правило требует от 30 до 60 коэффициентов импульсного отклика в зависимости от динамики процесса и длительности интервалов выборки (сэмплирования).

Модели на базе передаточных функций дают более лаконичное описание динамики процесса и применимы как к устойчивым, так и неустойчивым объектам управления (ОУ). СУПМ на базе передаточной функции как правило включают алгоритм прогностического управления Петерки [3] и алгоритм обобщенного прогностического управления (ОПУ) Кларка и его коллег [4]. Предиктивное управление на основе передаточной функции часто считается менее эффективным при работе с многопараметрическими установками.

В последние годы наблюдается растущая популярность прогностического управления с использованием методов проектирования в пространстве состояний [5 – 8]. В настоящем курсе рассматриваются модели в пространстве состояний, как непрерывного, так и дискретного времени.

Учебное пособие «Моделирование систем и комплексов: Дискретные системы прогностического управления в теории, задачах и примерах в MATLAB» является углубленным и дополненным изложением раздела, посвященного моделированию систем прогностического управления, в рамках курса «Моделирование систем и комплексов». Курс читается магистрам первого года, обучающихся по направлениям 15.04.06 "Мехатроника и робототехника", 27.04.03 "Системный анализ и управление", 27.04.04 "Управление в технических системах" в Университете ИТМО.

Овладев материалом, изложенным в учебном пособии, обучающиеся получают знания, необходимые для успешной профессиональной деятельности и выполнения самостоятельной научной работы. Работа включает в себя освоение теории, приобретение навыка использования ее аппарата, приобретение практических знаний в области современных методов построения систем прогностического управления непрерывного и дискретного времени, перехода от математических моделей к реализации компьютерных моделей, эффективных численных методов решений.

Для успешного освоения учебного пособия студентам необходимо обладать знаниями по теории автоматического управления, различным разделам высшей математики, основам программирования.

Предметом изучения являются:

- дискретные системы управления с прогнозирующей моделью;
- модели пространства состояний;
- система с одним входом и одним выходом (SISO);
- компьютерные реализации расширенной модели;
- системы прогностического управления в одном окне оптимизации;
- прогнозирование переменных состояния и выхода;
- методы оптимизации систем прогностического управления;
- методы и средства синтеза прогностических моделей;
- системы управления с удаляющимся горизонтом прогнозирования;
- системы управления с замкнутым контуром;
- методы компьютерного моделирования систем управления с удаляющимся горизонтом;
- управление с прогнозирующей моделью в МИМО-системах.

В результате изучения дисциплины слушатели должны приобрести следующие знания, умения и навыки:

- знать на уровне представлений принципы математического описания систем и комплексов;
- знать на уровне воспроизведения основные статические и динамические характеристики математических моделей преобразователей информации;
- на уровне понимания знать основные принципы построения, алгоритмы численных решений моделей систем и комплексов;
- уметь формировать постановку задачи и участвовать в разработке математических и (или) программных моделей систем и комплексов;
- уметь производить расчёты и проектирование отдельных блоков, устройств, систем и комплексов при моделировании
- получить навыки синтеза моделей и алгоритмов;
- обучиться навыкам работы с основными средствами моделирования комплексов и систем;
- получить навыки синтеза и анализа цифровых управляющих и информационных модулей.

Каждый теоретический блок учебного пособия снабжён наглядными примерами решения практических задач, использующими описанные в теории методы, и заданиями для самостоятельной работы. После изучения очередного блока теоретического материала, студенту сначала предлагается повторить по заданной пошаговой инструкции ход решения задачи, включающий подробный анализ проблемы, возможные подходы к её решению, синтез математической модели и переход к её компьютерной реализации в среде MATLAB. Затем, для закрепления пройденного материала и получения навыков, студенту следует проработать предлагаемые задания для самостоятельной работы. Успешно выполнив задания для самостоятельной работы, студент получает указанные выше навыки, знания и умения.

Примеры построения компьютерных моделей ориентированы на применение студентом инструментов среды MATLAB. Учебное пособие знакомит с особенностями данной среды, реализации в ней работы с переменными, матричными операциями, графическим интерфейсом, множеством функций, востребованных в задачах математического моделирования, и описывает применение инструментов среды для решения приводимых задач. Решение включает написание скриптов в среде MATLAB, с выводом результатов моделирования в символьной и графической формах.

1 ДИСКРЕТНЫЕ СИСТЕМЫ УПРАВЛЕНИЯ С ПРОГНОЗИРУЮЩЕЙ МОДЕЛЬЮ

1.1 Модели пространства состояний

Системы управления с прогнозирующей моделью проектируются на основе математической модели объекта [5 – 7]. Модель, используемая при проектировании системы управления, рассматривается как модель пространства состояний. При использовании модели пространства состояний текущая информация, необходимая для прогнозирования, представляется переменными состояниями.

1.2 Система с одним входом и одним выходом (SISO)

Для простоты мы начинаем наше исследование, предполагая, что ОУ представляет собой систему с одним входом и одним выходом, т.е. SISO-систему, описание которой задано уравнениями

$$x_m(k+1) = A_m x_m(k) + B_m u(k), \quad (1)$$

$$y(k) = C_m x_m(k), \quad (2)$$

где u – управляемая переменная или входная переменная; y – выход процесса; и x_m – вектор переменной состояния с размерностью n_l .

Обратим внимание, что представленная модель объекта имеет $u(k)$ в качестве входа.

Как видно, общая формулировка модели в пространстве состояний имеет слагаемое, описывающее прямую зависимость выхода $y(k)$ от входного сигнала $u(k)$:

$$y(k) = C_m x_m(k) + D_m u(k).$$

Однако, согласно принципу удаляющегося горизонта, где для прогнозирования и управления требуется текущая информация об ОУ, неявно предполагаем, что вход $u(k)$ не может влиять на выход $y(k)$. Таким образом, в модели ОУ $D_m = 0$.

Продифференцировав обе части (1), получим, что

$$x_m(k+1) - x_m(k) = A_m (x_m(k) - x_m(k-1)) + B_m (u(k) - u(k-1)).$$

Обозначим дифференциал переменной состояния как

$$\Delta x_m(k+1) = x_m(k+1) - x_m(k); \quad \Delta x_m(k) = x_m(k) - x_m(k-1),$$

а дифференциал переменной управления как

$$\Delta u(k) = u(k) - u(k-1).$$

Данные дифференциалы есть приращения переменных $x_m(k)$ и $u(k)$. Тогда дифференцирование уравнения пространства состояний примет вид

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k). \quad (3)$$

Обратим внимание, что вход модели пространства состояний есть $\Delta u(k)$. Далее необходимо связать $\Delta x_m(k)$ с выходом $y(k)$. Для этого выбирается новый вектор переменной состояния

$$x(k) = \left[\Delta x_m(k)^T \quad y(k) \right]^T,$$

где верхний индекс T означает транспонирование матрицы.

Заметим, что

$$\begin{aligned} y(k+1) - y(k) &= C_m (x_m(k+1) - x_m(k)) = C_m \Delta x_m(k+1) = \\ &= C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k). \end{aligned} \quad (4)$$

Объединение (3) и (4) приводит к следующей модели пространства состояний:

$$\begin{aligned} \overbrace{\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix}}^{x(k+1)} &= \overbrace{\begin{bmatrix} A_m & o_m^T \\ C_m A_m & 1 \end{bmatrix}}^A \overbrace{\begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}}^{x(k)} + \overbrace{\begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}}^B \Delta u(k) \\ y(k) &= \overbrace{\begin{bmatrix} o_m & 1 \end{bmatrix}}^C \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}, \end{aligned} \quad (5)$$

где $o_m = \overbrace{[0 \ 0 \ \dots \ 0]}^{n_1}$.

Тройка (A, B, C) называется моделью расширенного объекта управления, которая будет использоваться при проектировании предиктивного управления.

Пример 1.1. Рассмотрим дискретную по времени модель вида [86 – 89]

$$\begin{aligned} x_m(k+1) &= A_m x_m(k) + B_m u(k), \\ y(k) &= C_m x_m(k), \end{aligned} \quad (6)$$

где системные матрицы

$$A_m = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}; B_m = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}; C_m = [1 \ 0].$$

Необходимо найти матрицы A, B, C в расширенной модели (5) и вычислить собственные значения матрицы A расширенной модели.

Решение. Из (5) следует $n_1 = 2$, $o_m = [0 \ 0]$. Из расширенной модели объекта имеем

$$\begin{aligned} x(k+1) &= Ax(k) + B\Delta u(k), \\ y(k) &= Cx(k), \end{aligned} \quad (7)$$

где матрицы расширенной системы

$$A = \begin{bmatrix} A_m & o_m^T \\ C_m A_m & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix};$$

$$B = \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1 \\ 0.5 \end{bmatrix};$$

$$C = [o_m \ 1] = [0 \ 0 \ 1].$$

Характеристическое уравнение матрицы A задано через

$$\begin{aligned} \rho(\lambda) = \det(\lambda I - A) &= \det \begin{bmatrix} \lambda I - A_m & o_m^T \\ -C_m A_m & (\lambda - 1) \end{bmatrix} = \\ &= (\lambda - 1) \det(\lambda I - A_m) = (\lambda - 1)^3. \end{aligned} \quad (8)$$

Следовательно, расширенная модель пространства состояний имеет три собственных значения при $\lambda = 1$. Среди них два – от первоначального уравнения интегратора и одно от расширения модели ОУ.

1.3 Пример проектирования расширенной модели в MATLAB

Практическое задание 1. Цель данного практического занятия – показать, как синтезировать дискретную по времени модель пространства состояний из заданной непрерывной модели пространства состояний, и спроектировать расширенную дискретную по времени модель пространства состояний.

Рассмотрим систему непрерывного времени с моделью пространства состояний вида

$$\begin{aligned} \dot{x}_m(t) &= \begin{bmatrix} 0 & 1 & 0 \\ 3 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} x_m(t) + \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} u(t), \\ y(t) &= [0 \ 1 \ 0] x_m(t). \end{aligned} \quad (9)$$

Указания к выполнению:

1. Создайте новый файл с именем **augmodel.m** и выполните ввод текста программы согласно приведённым ниже пошаговым инструкциям.
2. Создайте непрерывную модель переменных состояния, согласно данным из (9), введя программу

```

Ac=[0 1 0; 3 0 1; 0 1 0];
Bc=[1; 1; 3];
Cc=[0 1 0];
Dc=zeros(1,1);

```

3. Введите переменную Δt интервала дискретизации со значением 1:
- ```
Delta_t=1;
```

4. Данную непрерывную модель приведите к дискретному виду с помощью функции MATLAB 'c2dm', с заданным ранее интервалом выборки  $\Delta t$ :

```
[Ad, Bd, Cd, Dd]=c2dm(Ac, Bc, Cc, Dc, Delta_t);
```

5. Размеры системных матриц определяются в соответствии с числом состояний входов и выходов. Введите следующие строки программы:

```

[m1, n1]=size(Cd);
[n1, n_in]=size(Bd);
A_e=eye(n1+m1, n1+m1);
A_e(1:n1, 1:n1)=Ad;
A_e(n1+1:n1+m1, 1:n1)=Cd*Ad;
B_e=zeros(n1+m1, n_in);
B_e(1:n1, :)=Bd;
B_e(n1+1:n1+m1, :)=Cd*Bd;
C_zeros(m1, n1+m1);
C_e(:, n1+1:n1+m1)=eye(m1, m1);

```

6. Запустите программу, после чего будет сгенерирована расширенная модель переменных состояния, используемая далее для проектирования системы предиктивного контроля.

## **1.4 Задания для самостоятельной работы**

### ***Задача 1.1.***

*Нарисуйте структуру содержания данной главы. Перечислите ключевые уравнения и понятия, которые вы встретили в главе.*

### ***Задача 1.2.***

*Представьте себя в роли водителя, который пытается управлять транспортным средством на повороте.*

*Опишите свою деятельность с точки зрения системы прогнозирующего управления.*

## 2 ПРОГНОСТИЧЕСКОЕ УПРАВЛЕНИЕ В ОДНОМ ОКНЕ ОПТИМИЗАЦИИ

Следующим шагом в синтезе математической модели при проектировании системы прогностического управления является расчет прогнозируемого выхода ОУ при будущем сигнале управления, рассматриваемого как регулируемые переменные. Данное предсказание описывается в окне оптимизации.

Рассмотрим в деталях оптимизацию, выполняемую в пределах данного окна. Примем, что  $k_i$  – текущий отсчет времени, а  $N_p$  – длина окна оптимизации, выражаемая в количестве выборок.

Для простоты сначала рассмотрим случай SISO-системы, а затем расширим результаты на системы со множеством входов и выходов (MIMO-систему).

### 2.1 Прогнозирование переменных состояния и выхода

На данном этапе примем, что для текущей выборки  $k_i$ ,  $k_i > 0$ , вектор переменной состояния  $x(k_i)$  доступен для измерения и предоставляет информацию о текущем состоянии объекта [9 – 12].

Более общий случай, когда состояние объекта не может быть измерено непосредственно, будет рассмотрен позже.

Будущую траектория управления обозначим как

$$\Delta u(k_i), \Delta u(k_i + 1), \dots, \Delta u(k_i + N_c - 1),$$

где  $N_c$  называется горизонтом управления, задающим количество параметров, используемых для определения будущей траектории управления.

На основе имеющейся в  $x(k_i)$  информации, будущие значения переменных состояния прогнозируются для  $N_p$  выборок, где  $N_p$  называется горизонтом прогнозирования, кроме того,  $N_p$  также является длиной окна оптимизации.

Обозначим будущие значения переменных состояния как

$$x(k_i + 1 | k_i), x(k_i + 2 | k_i), \dots, x(k_i + m | k_i), \dots, x(k_i + N_p | k_i),$$

где  $x(k_i + m | k_i)$  – значение переменной состояния в момент  $k_i + m$ , прогнозируемое на основании текущей информации  $x(k_i)$  об объекте. Горизонт управления  $N_c$  выбирается меньшим (или равным) горизонту прогнозирования  $N_p$ .

Основываясь на модели пространства состояния  $(A, B, C)$ , будущие значения переменных состояния вычисляются последовательно, с использованием будущих значений параметров управления:

$$\begin{aligned}
x(k_i + 1 | k_i) &= Ax(k_i) + B\Delta u(k_i) \\
x(k_i + 2 | k_i) &= Ax(k_i + 1 | k_i) + B\Delta u(k_i + 1) = \\
&= A^2x(k_i) + AB\Delta u(k_i) + B\Delta u(k_i + 1) \\
&\vdots \\
x(k_i + N_p | k_i) &= A^{N_p}x(k_i) + A^{N_p-1}B\Delta u(k_i) + A^{N_p-2}B\Delta u(k_i + 1) + \dots \\
&\dots + A^{N_p-N_c}B\Delta u(k_i + N_c - 1).
\end{aligned}$$

Прогнозируемые значения переменных выхода вычисляются на основании прогнозируемых переменных состояния, путем подстановки вида

$$\begin{aligned}
y(k_i + 1 | k_i) &= CAx(k_i) + CB\Delta u(k_i) \\
y(k_i + 2 | k_i) &= CA^2x(k_i) + CAB\Delta u(k_i) + CB\Delta u(k_i + 1) \\
y(k_i + 3 | k_i) &= CA^3x(k_i) + CA^2B\Delta u(k_i) + CAB\Delta u(k_i + 1) + CB\Delta u(k_i + 2)
\end{aligned} \tag{10}$$

$$\begin{aligned}
&\vdots \\
y(k_i + N_p | k_i) &= CA^{N_p}x(k_i) + CA^{N_p-1}B\Delta u(k_i) + CA^{N_p-2}B\Delta u(k_i + 1) + \dots \\
&\dots + CA^{N_p-N_c}B\Delta u(k_i + N_c - 1).
\end{aligned} \tag{11}$$

Обратим внимание, что все прогнозируемые переменные формируются из текущей информации о переменных состояния  $x(k_i)$  и будущего изменения управления  $\Delta u(k_i + j)$ , где  $j = 0, 1, \dots, N_c - 1$ .

Далее, определим векторы

$$Y = \begin{bmatrix} y(k_i + 1 | k_i) & y(k_i + 2 | k_i) & y(k_i + 3 | k_i) & \dots & y(k_i + N_p | k_i) \end{bmatrix}^T,$$

$$\Delta U = \begin{bmatrix} \Delta u(k_i) & \Delta u(k_i + 1) & \Delta u(k_i + 2) & \dots & \Delta u(k_i + N_c - 1) \end{bmatrix}^T,$$

где, в случае SISO-системы, размерность  $Y$  равна  $N_p$ , а размерность  $U$  равна  $N_c$ . Объединим (10) и (11) в компактной матричной форме:

$$Y = Fx(k_i) + \Phi\Delta U, \tag{12}$$

где

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix};$$

$$\Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & & & & \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix}.$$

## 2.2 Оптимизация

Для значения сигнала уставки (цели управления)  $r(k_i)$ , сэмплированного в момент  $k_i$ , целью прогнозирующей системы управления в области горизонта предсказания является приведение прогнозируемого выхода как можно ближе к заданному значению сигнала уставки, причём мы предполагаем, что заданное значение сигнала уставки в окне оптимизации остается постоянным. Достижение данной цели сводится к задаче поиска "налучшего" вектора параметров управления  $\Delta U$ , обеспечивающего минимизацию функции ошибки между заданным значением уставки и прогнозируемым выходом.

Предполагая, что вектор данных, содержащий информацию о сигнале уставки, задан как

$$R_s^T = \overbrace{[1 \ 1 \ \dots \ 1]}^{N_p} r(k_i),$$

определим целевую функцию управления  $J$  как

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U, \quad (13)$$

где  $\bar{R}$  – диагональная матрица вида  $\bar{R} = r_w I_{N_c \times N_c}$ , при  $r_w \geq 0$ , где  $r_w$  используется в качестве параметра настройки для обеспечения желаемых характеристик замкнутого контура.

Первое слагаемое в (13) связано с целью минимизации ошибки между прогнозируемым и заданным выходом, в то время как второе слагаемое отражает соображения относительно размера  $\Delta U$ , когда целевая функция  $J$  становится максимально малой.

В случае, когда задано  $r_w = 0$ , целевая функция (13) интерпретируется как ситуация, в которой для нас не имеет значения, насколько велико  $\Delta U$ , а целью является максимальная минимизация  $(R_s - Y)^T (R_s - Y)$ .

В случае больших значений  $r_w$  целевая функция (13) интерпретируется как ситуация, в которой нам необходимо производить тщательную оценку величины  $\Delta U$  и уменьшение ошибки  $(R_s - Y)^T (R_s - Y)$ .

Для нахождения оптимальной  $\Delta U$ , которая позволит минимизировать  $J$ , используем (12) и выразим  $J$  как

$$J = (R_s - Fx(k_i))^T (R_s - Fx(k_i)) - 2\Delta U^T \Phi^T (R_s - Fx(k_i)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U. \quad (14)$$

Из первой производной функции  $J$

$$\frac{\partial J}{\partial \Delta U} = -2\Phi^T (R_s - Fx(k_i)) + 2(\Phi^T \Phi + \bar{R})\Delta U, \quad (15)$$

необходимое условие минимума  $J$  получим как

$$\frac{\partial J}{\partial \Delta U} = 0,$$

откуда найдём оптимальное решение для сигнала управления

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k_i)), \quad (16)$$

полагая, что  $(\Phi^T \Phi + \bar{R})^{-1}$  существует.

В литературе по оптимизации матрица  $(\Phi^T \Phi + \bar{R})^{-1}$  называется матрицей Гессе или гессианом функции.

Обратим внимание, что  $R_s$  – вектор данных, который содержит информацию об уставке и выражается как

$$R_s = \overbrace{[1 \ 1 \ 1 \ \dots \ 1]^T}^{N_p} r(k_i) = \bar{R}_s r(k_i),$$

где

$$\bar{R}_s = \overbrace{[1 \ 1 \ 1 \ \dots \ 1]^T}^{N_p}.$$

Оптимальный сигнал управления связан с  $r(k_i)$  и переменной состояния  $x(k_i)$  уравнением

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (\bar{R}_s r(k_i) - Fx(k_i)). \quad (17)$$

**Пример 2.1.** [87 – 89] Предположим, что система первого порядка описывается уравнением состояния вида

$$\begin{aligned} x_m(k+1) &= ax_m(k) + bu(k), \\ y(k) &= x_m(k), \end{aligned} \quad (18)$$

где  $a = 0,8$  и  $b = 0,1$  – скаляры.

Необходимо:

- а) найти расширенную модель объекта в пространстве состояний;
- б) предполагая, что горизонт прогнозирования задан как  $N_p = 10$ , а горизонт управления – как  $N_c = 4$ , необходимо вычислить компоненты, определяющие прогноз будущего выхода  $Y$ ;
- в) выполнить оценку  $\Phi^T \Phi$ ,  $\Phi^T F$  и  $\Phi^T \bar{R}_s$ ;
- г) полагая, что в момент  $k_i$  (для данного примера примем  $k_i = 10$ ),  $r(k_i) = 1$  и вектор состояния  $x(k_i) = [0,1 \ 0,2]^T$ , необходимо найти оптимальное решение  $\Delta U$  для случаев, соответственно, при  $r_w = 0$  и  $r_w = 10$ ;

д) сравнить результаты, полученные для указанных случаев.

**Решение.** Расширенная модель пространства состояний описывается уравнениями



$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} a & 0 \\ a & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} b \\ b \end{bmatrix} \Delta u(k), \quad (19)$$

$$y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}.$$

Учитывая (12), матрицы  $F$  и  $\Phi$  принимают вид

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ CA^4 \\ CA^5 \\ CA^6 \\ CA^7 \\ CA^8 \\ CA^9 \\ CA^{10} \end{bmatrix}$$

и

$$\Phi = \begin{bmatrix} CB & 0 & 0 & 0 \\ CAB & CB & 0 & 0 \\ CA^2B & CAB & CB & 0 \\ CA^3B & CA^2B & CAB & CB \\ CA^4B & CA^3B & CA^2B & CAB \\ CA^5B & CA^4B & CA^3B & CA^2B \\ CA^6B & CA^5B & CA^4B & CA^3B \\ CA^7B & CA^6B & CA^5B & CA^4B \\ CA^8B & CA^7B & CA^6B & CA^5B \\ CA^9B & CA^8B & CA^7B & CA^6B \end{bmatrix},$$

соответственно.

Коэффициенты в матрицах  $F$  и  $\Phi$  рассчитываются следующим образом:

$$\begin{aligned} CA &= [s_1 \quad 1], \\ CA^2 &= [s_2 \quad 1], \\ CA^3 &= [s_3 \quad 1], \\ &\vdots \\ CA^k &= [s_k \quad 1], \end{aligned} \quad (20)$$

$$\begin{aligned}
\text{где } s_1 &= a, s_2 = a^2 + s_1, \dots, s_k = a^k + s_{k-1}, \text{ и} \\
CB &= g_0 = b, \\
CAB &= g_1 = ab + g_0, \\
CA^2B &= g_2 = a^2b + g_1, \\
&\vdots \\
CA^{k-1}B &= g_{k-1} = a^{k-1}b + g_{k-2}, \\
CA^k B &= g_k = a^k b + g_{k-1}.
\end{aligned} \tag{21}$$

С заданными параметрами  $a = 0,8$  и  $b = 0,1$ ;  $N_p = 10$  и  $N_c = 4$ , получаем следующие численные значения:

$$\begin{aligned}
\Phi^T \Phi &= \begin{bmatrix} 1.1541 & 1.0407 & 0.9116 & 0.7726 \\ 1.0407 & 0.9549 & 0.8475 & 0.7259 \\ 0.9116 & 0.8475 & 0.7675 & 0.6674 \\ 0.7726 & 0.7259 & 0.6674 & 0.5943 \end{bmatrix}; \\
\Phi^T F &= \begin{bmatrix} 9.2325 & 3.2147 \\ 8.3259 & 2.7684 \\ 7.2927 & 2.3355 \\ 6.1811 & 1.9194 \end{bmatrix}; \\
\Phi^T \bar{R}_s &= \begin{bmatrix} 3.2147 \\ 2.7684 \\ 2.3355 \\ 1.9194 \end{bmatrix}.
\end{aligned}$$

Заметим, что вектор  $\Phi^T \bar{R}_s$  идентичен последнему столбцу матрицы  $\Phi^T F$ . Это является следствием того, что последний столбец матрицы  $F$  идентичен  $\bar{R}_s$ .

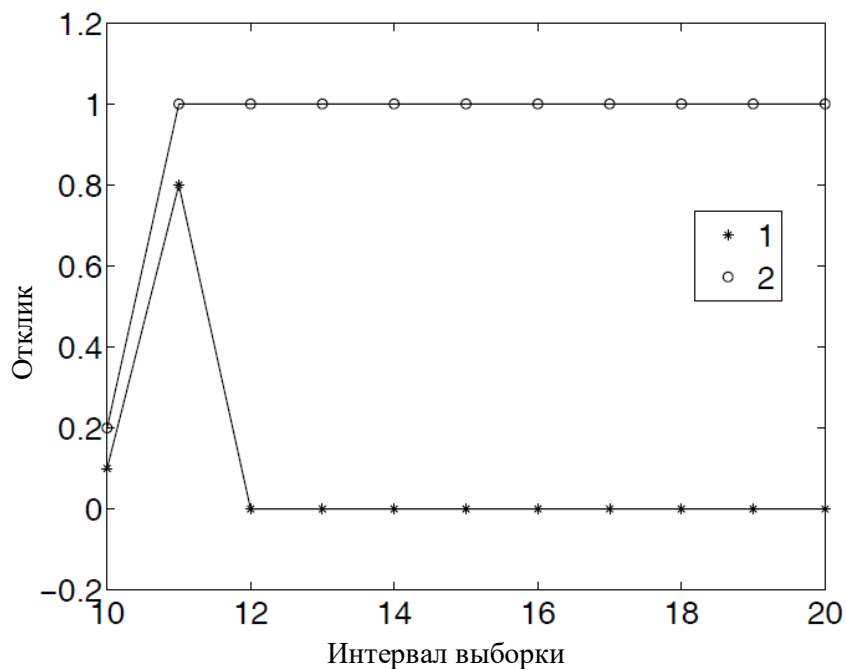
В момент времени  $k_i = 10$  вектор состояния  $x(k_i) = [0, 1, 0, 2]^T$ .

В первом случае, ошибка между прогнозируемыми  $Y$  и  $\bar{R}_s$  уменьшается без учета величины изменений сигнала управления вследствие условия  $r_w = 0$ . Тогда оптимальное  $\Delta U$  может быть найдено через выражение

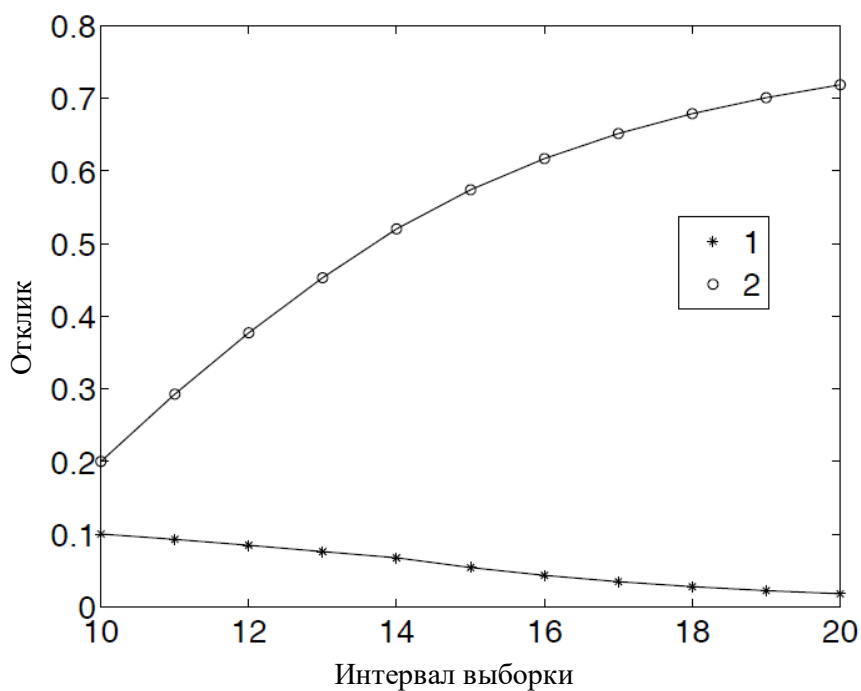
$$\Delta U = (\Phi^T \Phi)^{-1} (\Phi^T R_s - \Phi^T F x(k_i)) = [7.2 \quad -6.4 \quad 0 \quad 0]^T.$$

Заметим, что последние два элемента  $\Delta u(k_i + 2) = 0$  и  $\Delta u(k_i + 3) = 0$ , в то время как первые два элемента имеют довольно большую величину.

На рисунке 2.1а показаны изменения переменных состояния, при которых прогнозируемый выход достигает заданной уставки  $y = 1$ , а  $\Delta x_m$  затухает до нулевого значения.



а) переменные состояния без учета  $\Delta u$



б) Переменные состояния с учетом  $\Delta u$

Рисунок 2.1 – Сравнение оптимальных решений: линия 1 соответствует  $\Delta x_m$ ; линия 2 соответствует  $y$ .

Для изучения влияния веса  $r_w$  на оптимальное управление предположим, что  $r_w = 10$ . Оптимальное значение  $\Delta U$  равно

$$\begin{aligned} \Delta U &= (\Phi^T \Phi + 10 \times I)^{-1} (\Phi^T R_s - \Phi^T F x(k_i)) = \\ &= [0.1269 \quad 0.1034 \quad 0.0829 \quad 0.065]^T. \end{aligned} \quad (22)$$

где  $I$  – единичная матрица с размерностью  $4 \times 4$ . При таких значениях величина первых двух приращений управления существенно уменьшается, причем последние две составляющие больше не равны нулю.

На рисунке 2.1б показаны оптимальные переменные состояния. Видно, что выход достиг заданного значения  $y = 1$ , однако  $\Delta x_m$  приближается к нулю.

Сравнительный анализ представленных случаев приводит к следующим наблюдениям. В случае, если мы желаем более “осторожного” движения траектории управления, то сигналу управления необходимо больше времени, чтобы достичь своего устойчивого состояния (т. е. значения в  $\Delta U$  должны уменьшаться медленнее), вследствие того, что оптимальная энергия управления распределяется на более длительном периоде прогнозирования (будущего времени). Это наблюдение может быть проверено путём увеличения  $N_c$  до 9 и сохранения при этом  $r_w = 10$ . Результат показывает, что значения в  $\Delta U$  уменьшаются, но они значимы только для первых 8 элементов:

$$\Delta U^T = \begin{bmatrix} 0.1227 \\ 0.0993 \\ 0.0790 \\ 0.0614 \\ 0.0463 \\ 0.0334 \\ 0.0227 \\ 0.0139 \\ 0.0072 \end{bmatrix} .$$

В сравнении со случаем  $N_c = 4$  можно заметить, что при  $N_c = 9$ , значения первых четырёх параметров в  $\Delta U$  имеют небольшие отличия от таковых в предыдущем случае.

**Пример 2.2.** Существует альтернативный способ поиска минимума целевой функции – с помощью метода дополнения до полного квадрата. Это – интуитивный подход, при использовании которого минимум целевой функции становится его побочным продуктом [90 – 95].

Пусть необходимо найти оптимальное решение для  $\Delta U$ , дополнив квадраты целевого функционала  $J$  (14).

**Решение.** Из (14), путем добавления и вычитания слагаемого

$$(R_s - Fx(k_i))^T \Phi (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k_i))$$

к исходной функции  $J$  может быть получено

$$\begin{aligned}
J = & (R_s - Fx(k_i))^T (R_s - Fx(k_i)) - \\
& \left[ -2\Delta U^T \Phi^T (R_s - Fx(k_i)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U^T \right] + \\
& + \left[ (R_s - Fx(k_i))^T \Phi (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k_i)) \right] - \\
& - (R_s - Fx(k_i))^T \Phi (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k_i)),
\end{aligned} \tag{23}$$

где величины в квадратных скобках являются дополненными "квадратами":

$$\begin{aligned}
J_0 = & (\Delta U - (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k_i)))^T \times \\
& \times (\Phi^T \Phi + \bar{R}) (\Delta U - (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k_i))).
\end{aligned} \tag{24}$$

Поскольку первый и последний члены в (23) не зависят от переменной  $\Delta U$  (в ряде случаев будем называть её переменной (принятия) решения), а  $(\Phi^T \Phi + \bar{R})$  предполагается положительно определенным, то минимум функции  $J$  достигается в случае, когда величина  $J_0$  равна нулю, то есть,

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k_i))^T. \tag{25}$$

Данное решение соответствует оптимальному управлению. Подставляя данное оптимальное решение в функцию (23), получаем минимум:

$$\begin{aligned}
J_{\min} = & (R_s - Fx(k_i))^T (R_s - Fx(k_i)) - \\
& - (R_s - Fx(k_i))^T \Phi (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k_i)).
\end{aligned}$$

## 2.3 Синтез прогностических моделей в MATLAB

**Практическое задание 2.** Целью данного задания является создание функции для расчета значений  $\Phi^T \Phi$ ,  $\Phi^T F$  и  $\Phi^T \overline{R_s}$  в среде MATLAB. Основной задачей при этом является синтез матриц  $F$  и  $\Phi$ .

Матрица  $\Phi$  является теплицевой и создается путем определения её первого столбца, затем следующий столбец получается путем сдвига предыдущего столбца.

### Основные шаги:

1. Создайте новый файл с именем *mcsgain.m*.
2. Первым шагом является создание модели расширенного объекта для проектирования СПУМ. Входными параметрами функции являются модели пространства состояний ( $A_p, B_p, C_p$ ), горизонт прогнозирования  $N_p$  и горизонт управления  $N_c$ .

Введите в файл следующий текст программы:

```
function [Phi_Phi, Phi_F, Phi_R, A_e, B_e, C_e]
 =mpcgain (Ap, Bp, Cp, Nc, Np);

[m1, n11]=size(Cp);
[n1, n_in]=size(Bp);
A_e=eye(n1+m1, n1+m1);
A_e(1:n1, 1:n1)=Ap;
A_e(n1+1:n1+m1, 1:n1)=Cp*Ap;
B_e=zeros(n1+m1, n_in);
B_e(1:n1, :)=Bp;
B_e(n1+1:n1+m1, :)=Cp*Bp;
C_zeros(m1, n1+m1);
C_e(:, n1+1:n1+m1)=eye(m1, m1);
```

3. Обратите внимание, что матрицы  $F$  и  $\Phi$  имеют специальные формы.

4. Продолжите ввод следующего текста программы в созданном файле:

```

n=n1+m1;
h(1,:)=C_e;
F(1,:)=C_e*A_e;
for kk=2:Np
h(kk,:)=h(kk-1,:)*A_e;
F(kk,:)=F(kk-1,:)*A_e;
end
v=h*B_e;
Phi=zeros(Np,Nc); %declare the dimension of Phi
Phi(:,1)=v; %first column of Phi
for i=2:Nc
Phi(:,i)=[zeros(i-1,1);v(1:Np-i+1,1)]; %Toeplitz matrix
end
BarRs=ones(Np,1);
Phi_Phi=Phi'*Phi;
Phi_F=Phi'*F;
Phi_R=Phi'*BarRs;

```

5. Введите в рабочем пространстве MATLAB:  $A_p = 0,8$ ;  $B_p = 0,1$ ;  $C_p = 1$ ;  $N_c = 4$  и  $N_p = 10$ . Запустите выполнение данной функции MATLAB, введя:

```
[Phi_Phi,Phi_F,Phi_R,A_e,B_e,C_e]=mpcgain(Ap,Bp,Cp,Nc,Np);
```

6. Сравните результаты с результатами, полученными в примере 2.1. Если они идентичны показанным ранее, то программа работает верно.

7. Изменяя горизонт прогнозирования и горизонт управления, наблюдайте за изменениями в данных матрицах.

8. Вычислите  $\Delta U$ , учитывая начальные условия по  $x$  и  $r$ . Обратная матрица  $M$  может быть вычислена в MATLAB с помощью функции `inv(M)`.

9. Проверьте результаты, полученные в примере 2.1.

## 2.4 Задания для самостоятельной работы

### Задача 2.1.

Предположим, что дискретная по времени система имеет вход  $u(k)$  и выход  $y(k)$ . Система имеет постоянное возмущение  $d$  входного сигнала.

**Необходимо:**

а) Найти расширенную модель пространства состояний с входом  $\Delta u(k)$  и выходом  $y(k)$  для модели системы управления вида

$$x_m(k+1) = A_m x_m(k) + B_m u(k) + B_d d,$$

$$y(k) = C_m x_m(k),$$

$$\text{где } A_m = \begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & -0.1 \\ 0 & 0 & 0.8 \end{bmatrix}; B_m = \begin{bmatrix} 0.5 \\ 1 \\ -0.6 \end{bmatrix}; C_m = [1 \ 0 \ 1]; B_d = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

б) Рассчитать передаточную функцию системы, которая связывает вход  $u(k)$  с выходом  $y(k)$ , и передаточную функцию расширенной модели пространства состояний, которая связывает вход  $\Delta u(k)$  с выходом  $y(k)$ .

в) Сравнить полюса и нули найденных передаточных функций.



### 3 УПРАВЛЕНИЕ С УДАЛЯЮЩИМСЯ ГОРИЗОНТОМ ПРОГНОЗИРОВАНИЯ

Несмотря на то, что вектор оптимальных параметров  $\Delta U$  содержит элементы управления  $\Delta u(k_i)$ ,  $\Delta u(k_i + 1)$ ,  $\Delta u(k_i + 2)$ , ...,  $\Delta u(k_i + N_c - 1)$ , при реализации принципа управления с удаляющимся горизонтом предсказания используется только первый элемент данной последовательности, то есть  $\Delta u(k_i)$ , а остальные элементы последовательности игнорируются. На интервале следующей выборки для вычисления новой последовательности сигнала управления используется результат нового измерения, формируемый вектором состояния  $x(k_i + 1)$ . Данная процедура повторяется в режиме реального времени, обеспечивая работу принципа управления удаляющимся горизонтом предсказания [99 – 103].

**Пример 3.1.** Проиллюстрируем указанную процедуру, продолжая пример 2.1, где в вычислениях используется система первого порядка с описанием пространства состояний вида

$$x_m(k+1) = 0,8x_m(k) + 0,1u(k).$$

Рассмотрим случай, когда  $r_w = 0$ .

Примем начальные условия:  $x(k_i) = [0,1 \ 0,2]^T$  и  $u(9) = 0$ .

**Решение.** На интервале выборки  $k_i = 10$  вычисление оптимального управления дало  $\Delta u(10) = 7,2$ . Предположим, что  $u(9) = 0$ , тогда сигнал управления  $u(10) = u(9) + \Delta u(10) = 7,2$ ; а  $x_m(10) = y(10) = 0,2$ . Вычислим следующую моделируемую переменную состояния:

$$x_m(11) = 0,8x_m(10) + 0,1u(10) = 0,88. \quad (26)$$

При  $k_i = 11$  новые данные  $x_m(11) = 0,88 - 0,2 = 0,68$  и  $y(11) = 0,88$ , формируют  $x(11) = [0,68 \ 0,88]^T$ . После чего получим, что

$$\Delta U = (\Phi^T \Phi)^{-1} (\Phi^T R_s - \Phi^T F x(11)) = [-4,24 \ -0,96 \ 0 \ 0]^T,$$

откуда получаем оптимальное управление  $u(11) = u(10) + \Delta u(11) = 2,96$ . Этот новый элемент управления используется для получения значения  $x_m(12)$ :

$$x_m(12) = 0,8x_m(11) + 0,1u(11) = 1. \quad (27)$$

При  $k_i = 12$  новой информацией является  $x_m(12) = 1 - 0,88 = 0,12$  и  $y(12) = 1$ , которые формируют  $x(12) = [0,12 \ 0,88]^T$  и, соответственно:

$$\Delta U = (\Phi^T \Phi)^{-1} (\Phi^T R_s - \Phi^T F x(12)) = [-0,96 \ 0 \ 0 \ 0]^T.$$

Как следствие, получаем, что управление при  $k_i = 12$  имеет вид  $u(12) = u(11) - 0,96 = 2$ . Решая данное уравнение, получим

$$x_m(13) = ax_m(12) + bu(12) = 1. \quad (28)$$

Из новых данных об объекте управления  $\Delta x_m(13) = 1 - 1 = 0$  и  $y(13) = 1$  получим:

$$\Delta U = (\Phi^T \Phi)^{-1} (\Phi^T R_s - \Phi^T F x(11)) = [0 \ 0 \ 0 \ 0]^T.$$

На рисунке 3.1 показаны траектории изменения переменных состояния  $\Delta x_m$  и  $y$ , а также управляющий сигнал, использовавшийся для регулирования выхода.

Приведённый пример также иллюстрирует различия между векторами параметров  $\Delta U$  в различных промежутках времени.

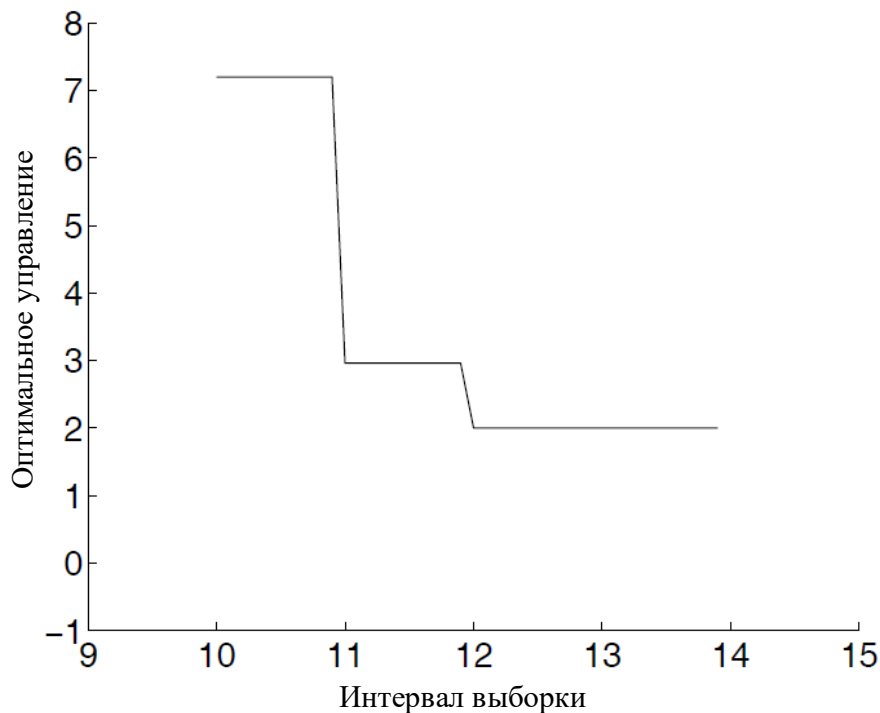
Отметим, что по мере того, как выходной отклик достигает заданной уставки сигнала, параметры в  $\Delta U$  приближаются к нулю.

### 3.1 Система управления с замкнутым контуром

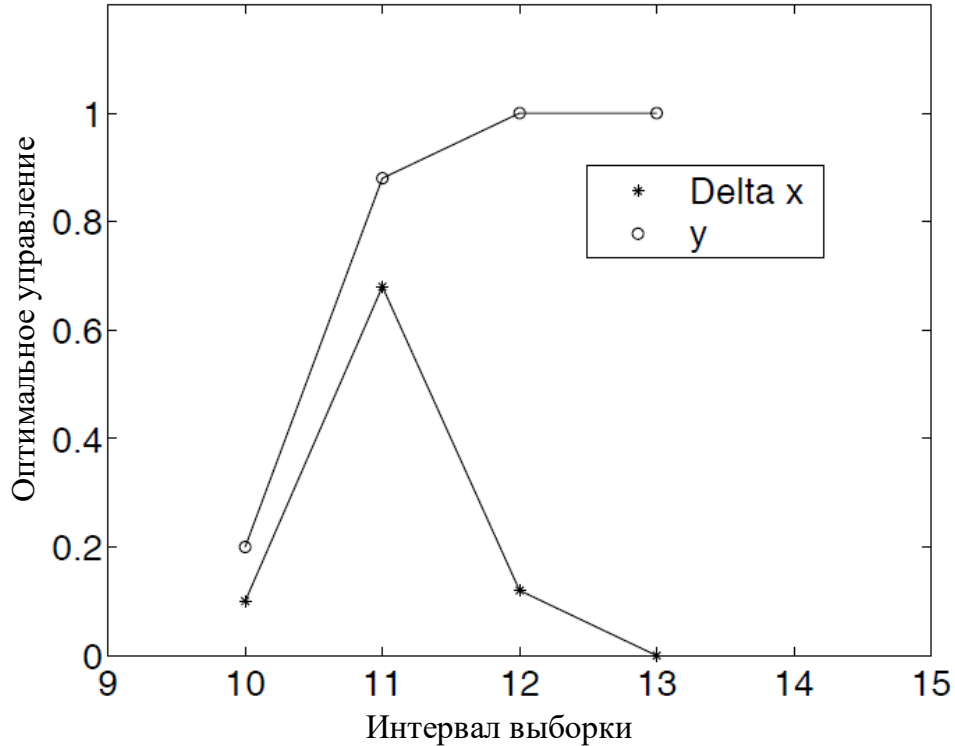
В примере 2.3 был проиллюстрирован ещё один важный аспект исследуемых систем. Если внимательно исследовать указанный пример, то можно обнаружить, что на интервале времени  $k_i$  вектор оптимальных параметров  $\Delta U$  вычисляется с помощью

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T R_s - \Phi^T F x(k_i)),$$

где  $(\Phi^T \Phi + \bar{R})^{-1} \Phi^T R_s$  соответствует изменению значения уставки, а  $(\Phi^T \Phi + \bar{R})^{-1} \Phi^T F$  соответствует управлению с обратной связью по состоянию в рамках метода прогностического управления, и оба зависят от параметров системы, а следовательно, являются постоянными матрицами для время-инвариантной системы.



а) Оптимальное управление



б) Переменная состояния

Рисунок 3.1 – Управление отступающим горизонтом

Вследствие применения принципа удаляющегося горизонта в момент времени  $k_i$  используется только первый элемент  $\Delta U$ , и, таким образом, имеем

$$\Delta u(k_i) = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^{N_c} (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T \bar{R}_s r(k_i) - \Phi^T F x(k_i)) = \quad (29)$$

$$= K_y r(k_i) - K_{mpc} x(k_i),$$

где  $K_y$  – первый элемент  $(\Phi^T \Phi + \bar{R})^{-1} \Phi^T \bar{R}_s$ , а  $K_{mpc}$  является первой строкой  $(\Phi^T \Phi + \bar{R})^{-1} \Phi^T F$ .

Уравнение (29) имеет общий вид линейного время-инвариантного управления с обратной связью по состоянию управления, где  $K_{mpc}$  – вектор усиления обратной связи по состоянию. Поэтому с расширенной моделью пространства состояний

$$x(k+1) = Ax(k) + B\Delta u(k),$$

замкнутая система может быть получена путем подстановки (29) в расширенное уравнение системы. Заменив индекс  $k_i$  на  $k$ , получаем уравнение замкнутой системы

$$x(k+1) = Ax(k) - BK_{mpc} x(k) + BK_y r(k) = \quad (30)$$

$$= (A - BK_{mpc})x(k) + BK_y r(k). \quad (31)$$

Таким образом, собственные значения замкнутого контура могут быть вычислены через его характеристическое уравнение вида

$$\det[\lambda I - (A - BK_{mpc})] = 0.$$

Вследствие особенностей структуры матриц  $C$  и  $A$  последний столбец  $F$  совпадает с  $\bar{R}_s$ , имеющим вид  $[1 \ 1 \ \dots \ 1]^T$ , и как следствие,  $K_y$  идентичен последнему элементу  $K_{mpc}$ .

Заметив, что вектор переменных состояния  $x(k_i) = [\Delta x_m(k)^T \ y(k)]^T$ , и с данным определением  $K_y$  мы можем записать  $K_{mpc} = [K_x \ K_y]$ , где  $K_x$  соответствует вектору усиления обратной связи, связанному с  $\Delta x_m(k)$ , а  $K_y$  – коэффициенту обратной связи, связанному с  $y(k)$ . После этого может быть получена структурная схема замкнутого контура СПУМ, приведённая на рисунке 3.2, где  $q^{-1}$  – оператор обратного сдвига.

На рисунке показана структура обратной связи по состоянию для дискретной модели прогнозирующего управления (ДМПУ) с интегрированием, в которой блок  $\frac{1}{1-q^{-1}}$  представляет дискретный интегратор.

**Пример 3.2.** Необходимо исследовать матрицы коэффициентов обратной связи, полученные в примере 2.1, и собственные значения замкнутой системы с весовыми коэффициентами  $r_w = 0$  и  $r_w = 10$ .

**Решение.** При  $r_w = 0$  имеем

$$K_y = [1 \ 0 \ 0 \ 0](\Phi^T \Phi + \bar{R})^{-1}(\Phi^T \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}) = 10,$$

$$K_{mpc} = [1 \ 0 \ 0 \ 0](\Phi^T \Phi + \bar{R})^{-1}(\Phi^T F) = [8 \ 10].$$

Следовательно, собственные значения замкнутой системы вычисляются путем оценки собственных значений замкнутой матрицы  $A - BK_{mpc}$ , где из примера 2.1

$$A = \begin{bmatrix} 0,8 & 0 \\ 0,8 & 1 \end{bmatrix};$$

$$B = \begin{bmatrix} 0,1 \\ 0,1 \end{bmatrix}.$$

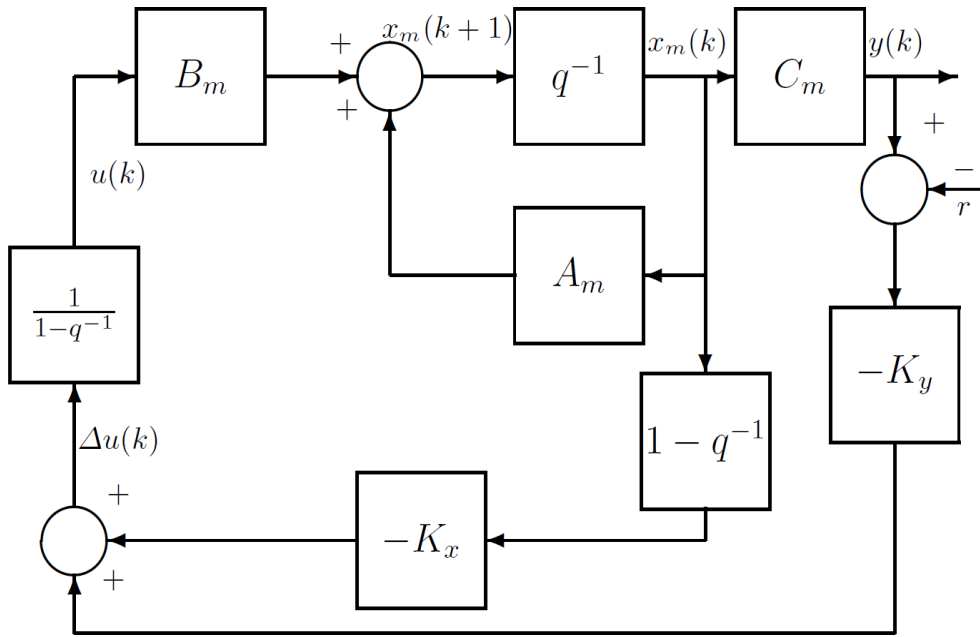


Рисунок 3.2 – Блок-схема системы прогнозирования дискретного времени

Собственные значения соответственно равны  $\lambda_1 = -6.409 \times 10^{-7}$  и  $\lambda_2 = 6.409 \times 10^{-7}$ , и расположены примерно в начале координат на комплексной плоскости.

При  $r_w = 10$  имеем

$$K_y = [1 \ 0 \ 0 \ 0] (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}) = 0,2453;$$

$$K_{mpc} = [1 \ 0 \ 0 \ 0] (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T F) = [0,6939 \ 0,2453].$$

Данному вектору усиления соответствуют собственные значения замкнутой системы  $\lambda_{1,2} = 0,8530 \pm j0,0542$ , что указывает на то, что динамика замкнутой системы изменяется гораздо медленнее, чем в случае, когда  $r_w = 0$ .

**Пример 3.3.** Предположим, что непрерывная по времени система описывается передаточной функцией Лапласа вида

$$G(s) = \frac{\omega^2}{s^2 + 0.1\omega s + \omega^2},$$

где  $\omega = 10$ .

Данная система приводится к дискретному виду с использованием интервала выборки  $\Delta t = 0,01$ .

Необходимо изучить изменение чувствительности при выборе проектных параметров для  $N_c = 3, N_p = 20$  и  $200; \bar{R} = 0,5I$ .

**Решение.** Используем приведенный ниже скрипт функции MATLAB для получения непрерывной модели пространства состояния:

```
omega=10;
numc=omega^2;
denc=[1 0.1*omega omega^2];
[Ac, Bc, Cc, Dc]=tf2ss (numc, denc);
```

Далее, следуя методике, приведённой в практическом задании 1.1, получаем дискретно-временную модель пространства состояния.

Расширенное уравнение пространства состояния с дискретным временем примет вид

$$\begin{aligned}x(k+1) &= Ax(k) + B\Delta u(k), \\ y(k) &= Cx(k),\end{aligned}$$

где

$$A = \begin{bmatrix} 0.9851 & -0.9934 & 0 \\ 0.0099 & 0.9950 & 0 \\ 0.9934 & 99.5021 & 1 \end{bmatrix};$$

$$B = \begin{bmatrix} 0.0099 \\ 0 \\ 0.0050 \end{bmatrix};$$

$$C = [0 \ 0 \ 1].$$

Во-первых, рассмотрим влияние горизонта прогнозирования на значения  $\Delta U$ .

Предположим, что в момент выборки  $k=10$  начальное условие  $x(10) = [0,1 \ 0,2 \ 0,3]^T$ .

Для  $N_p = 20$  равно  $\Delta U = [-144,9984 \ -65,4710 \ 1,2037]^T$ .

Используя принцип удаляющегося горизонта, находим коэффициенты усиления обратной связи  $K_{trc} = [45,4168 \ 705,6132 \ 0,9513]$ .

Собственные значения замкнутой системы составляют  $0,6974; 0,8959 \pm 0,1429j$ . Однако, для  $N_p = 200$ ,  $\Delta U = [-645,5885 \ -0,4664 \ 629,0276]^T$ .

Как видно, по сравнению с предыдущим случаем, где был применён более короткий удаляющийся горизонт  $N_p = 20$ , вектор параметров  $\Delta U$  изменился значительно.

Опять же, для коэффициентов усиления  $K_{\text{трс}} = [80,6 \quad 3190 \quad 0,79]$ , собственные значения замкнутой системы равны  $0,9749; 0,5207 \pm j0,2919$ .

Это сравнительное исследование показало существование чувствительности к выбору горизонта прогнозирования.

При более тщательном рассмотрении может быть замечено, что матрица Гессе

$$\Phi^T \Phi + \bar{R}$$

является функцией горизонта прогнозирования.

Например, для  $N_p = 20$

$$\Phi^T \Phi = \begin{bmatrix} 9,8796 & 8,9387 & 8,0099 \\ 8,9387 & 8,1020 & 7,2737 \\ 8,0099 & 7,2737 & 6,5425 \end{bmatrix},$$

с номером условия  $\kappa(\Phi^T \Phi + 0,5I) = 49,98$ .

Однако, при  $N_p = 200$ ,

$$\Phi^T \Phi = \begin{bmatrix} 236,0557 & 235,5010 & 234,5466 \\ 235,5010 & 235,3753 & 234,8473 \\ 234,5466 & 234,8473 & 234,7473 \end{bmatrix},$$

при  $\kappa(\Phi^T \Phi + 0,5I) = 1410$ .

Как видно, значение  $\kappa(\Phi^T \Phi + 0,5I)$  значительно увеличивается по мере увеличения горизонта прогнозирования  $N_p$  до 200. Это большое число состояний матрицы Гессе для длинного горизонта прогнозирования приводит к количественной чувствительности, которая вызывает значительное различие в результатах, полученных для случая короткого и длинного горизонта прогнозирования.

С короткими горизонтами прогнозирования и управления замкнутая система прогнозирующего управления не обязательно устойчива, и традиционно данные параметры используются для обеспечения устойчивости и заданных параметров качества в замкнутой системе.

Далее будет предложен подход, который использует большие горизонты прогнозирования и управления для обеспечения устойчивости в замкнутой системе.

## 3.2 Моделирование систем управления с удаляющимся горизонтом в MATLAB

**Практическое задание 3.** Целью задания является получение навыков проектирования системы прогнозирования с использованием управления с удаляющимся горизонтом.

Модель пространства состояния задана в виде

$$\begin{aligned}x_m(k+1) &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_m(k) + \begin{bmatrix} 0,5 \\ 1 \end{bmatrix} u(k), \\ y(k) &= [1 \quad 0] x_m(k).\end{aligned}\tag{32}$$

**Основные шаги:**

1. Создайте новый файл программы с именем *gcd.m*
2. Прежде всего необходимо определить объект управления, ввести значения параметров горизонта прогнозирования и горизонта управления. Объект представляет собой систему с двойным интегратором (32). Горизонт управления выбирается равным  $N_c = 4$ , а горизонт прогноза  $N_p = 20$ .

Введите в файл следующий текст программы:

```
Ap=[1 1;0 1];
Bp=[0.5;1];
Cp=[1 0];
Dp=0;
Np=20;
Nc=4;
```

3. Программа вызывает функцию *tcsgain.m* для генерации необходимых матриц усиления и определяет начальные условия для реализации управления с удаляющимся горизонтом.

Начальное значение переменной состояния для объекта управления равна  $x_m=0$ ; а переменная начальное значение переменной состояния обратной связи равно  $x_f=0$ ; далее задаётся уставка сигнала и указывается количество точек моделирования, равное 100.

4. Продолжите ввод следующего текста программы:



```

[Phi_Phi,Phi_F,Phi_R,A_e, B_e,C_e]
 = mcsgain (Ap,Bp,Cp,Nc,Np) ;

[n,n_in]=size(B_e);
xm=[0;0];
Xf=zeros(n,1);
N_sim=100;
r=ones(N_sim,1);
u=0; % u(k-1) =0
y=0;

```

5. Согласно методу управления с удаляющимся горизонтом в момент времени  $kk$  вектор  $\Delta U$  рассчитывается с использованием заданного сигнала  $r(kk)$  и вектора состояния  $Xf$ . Затем  $\Delta u(kk)$  берётся как первый элемент  $\Delta U$ ; и  $u(kk) = u(kk - 1) + \Delta u(k)$ .

Весовой коэффициент выбирается равным 0,1.

6. Продолжайте вводить следующий текст программы:

```

for kk=1:N_sim;
DeltaU=inv (Phi_Phi+0.1*eye (Nc,Nc)) * (Phi_R*r(kk) -
 Phi_F*Xf);
deltau=DeltaU(1,1);
u=u+deltau;
u1(kk)=u;
y1(kk)=y;

```

7. Состояние и мощность установки моделируются с использованием генерируемого управляющего сигнала; переменная состояния, используемая в механизме обратной связи, обновляется как  $Xf$ .

8. Продолжайте вводить в файл следующую программу:

```

xm_old=xm;
xm=Ap*xm+Bp*u;
y=Cp*xm;
Xf=[xm-xm_old;y];
end

```

10. Введите в файл следующий текст программы:

```
k=0:(N_sim-1);
figure
subplot(211)
plot(k, y1)
xlabel('Sampling Instant')
legend('Output')
subplot(212)
plot(k, u1)
xlabel('Sampling Instant')
legend('Control')
```

*Входные и выходные сигналы строятся относительно выборок.*

*11. Сохраните программу в той же папке, которая содержит функцию. Запустите программу.*

*12. Измените вес  $r_w$  на 2 и обратите внимание, что при замкнутом контуре скорость отклика медленнее.*

*13. Смоделируйте свой собственный объект, используя различные  $A_p$ ,  $B_p$  и  $C_p$ , и проведите эксперименты с различными величинами горизонтов прогнозирования и управления.*

### 3.3 Задания для самостоятельной работы

#### Задача 3.1.

Пусть расширенная математическая модель дискретной системы определена как

$$x(k+1) = Ax(k) + B\Delta u(k)$$

$$y(k) = Cx(k),$$

$$\text{где } A = \begin{bmatrix} 0.6 & 0 \\ 0.6 & 1 \end{bmatrix}; B = \begin{bmatrix} 0.3 \\ 0.3 \end{bmatrix}; C = [0 \quad 1].$$

**Необходимо:**

- а) Для момента времени  $k_i = 0$ , предполагая, что горизонт управления  $N_c = 4$ ,  $N_p = 10$  и начальное значение переменной состояния  $x(0) = [0,1 \quad 0,2]^T$ , записать решение прогнозируемого выхода

$$y(k_i+1 | k_i), y(k_i+2 | k_i), \dots, y(k_i+N_p | k_i)$$

через  $\Delta U$ , где

$$\Delta U = [\Delta u(k_i) \quad \Delta u(k_i+1) \quad \Delta u(k_i+2) \quad \Delta u(k_i+3)]^T.$$

- б) Для заданной уставки  $r(k_i) = 0$ , найти оптимальное управление  $\Delta U$ , минимизирующее целевую функцию

$$J = Y^T Y + \Delta U^T \bar{R} \Delta U,$$

где  $\bar{R}$  – диагональная матрица ( $\bar{R} = r_\omega I$ ,  $r_\omega = 3$  и  $I$  – тождественная матрица  $4 \times 4$ ). Каков минимум функции стоимости,  $J_{\min}$ ?

- в) Уменьшайте  $r_\omega$  до 0, наблюдая за изменениями оптимального управления  $\Delta U$ .
- г) Сравните значение  $J_{\min}$  для случая  $r_\omega = 0$ , со значением из предыдущего случая  $r_\omega = 3$ .

#### Задача 3.2.

Модель первого порядка часто используется для описания динамики резервуара с жидкостью, где вход системы – это скорость потока, а выход – уровень жидкости (рисунок 3.3).

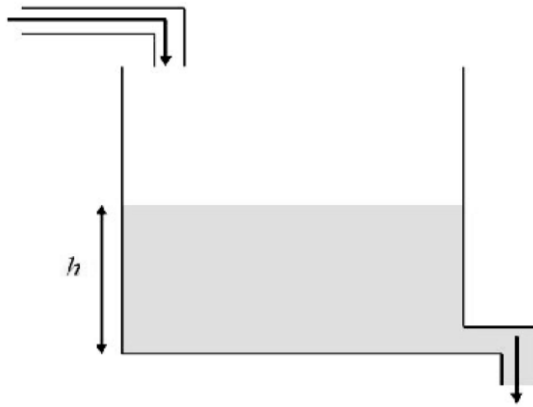


Рисунок 3.3 – Контроль уровня жидкости в резервуаре

Предположим, что для данной системы получена дискретная передаточная функция первого порядка, описывающая связь между входом и выходом, вида

$$Y(z) = \frac{0,01}{z - 0,6} U(z)$$

**Необходимо:**

- а) Выбрав  $x_m(k) = h(k) = y(k)$ , преобразовать модель передаточной функции в модель пространства состояний и спроектировать модель прогнозирующей системы управления, которая будет поддерживать уровень жидкости на заданном уровне  $r(k) = 0,5$ . Расчетные параметры системы прогнозирующего управления определяются как  $N_c = 4$ ,  $N_p = 16$ ,  $\bar{R} = I$ .
- б) При начальном условии  $x(0) = [0 \ 0]^T$  смоделируйте замкнутую систему прогнозирующего управления, используя управление с удаляющимся горизонтом. (Подсказка: переменные состояния  $x_1(k) = y(k) - y(k - 1)$  и  $x_2(k) = y(k)$ . Обе – измеримы)

### Задача 3.3.

Временная задержка в дискретной системе появляется как часть знаменателя передаточной функции. Следовательно, число задержек увеличит размерность модели пространства состояний. Предположим, что дискретная система описывается передаточной функцией  $z$

$$G(z) = \frac{0,1z^{-5}}{1 - 1,4z^{-1} + 0,48z^{-2}}.$$

Это система с временной задержкой. Альтернативное представление этой передаточной функции

$$G(z) = \frac{0,1}{z^3(z^2 - 1,4z + 0,48)},$$

указывает на то, что система имеет пятый порядок (число полюсов определяет порядок модели). Модель пространства состояний для этой передаточной функции

$$x_m(k+1) = A_m x_m(k) + B_m u(k)$$

$$y(k) = C_m x_m(k)$$

$$\text{где } A_m = \begin{bmatrix} 1.4 & -0.48 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}; B_m = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}; C_m = [0 \ 0 \ 0 \ 0 \ 0.1].$$

#### Необходимо:

1. Создать прогнозирующую систему управления, которая будет отслеживать опорный сигнал шага. Для достижения ступенчатой реакции этой системы требуется около 20 выборков, чтобы достичь устойчивого состояния. Предполагая, что отклик замкнутой системы будет быстрее, чем отклик разомкнутой, выберите горизонт прогнозирования  $N_p = 16$  и горизонт управления  $N_c = 4$ . Выбор весовой матрицы  $\bar{R}$  может быть связан с выбором  $N_p$  и  $N_c$ . Здесь выберите  $\bar{R} = r_\omega I$  и  $r_\omega = 0,01$ . Для большего диагонального элемента  $r_\omega$  в  $\bar{R}$ , горизонт прогнозирования может быть больше.
2. При выборе параметров проектирования вычислите коэффициент усиления обратной связи системы прогнозирующего управления и собственные числа замкнутой системы.
3. С нулевым начальным условием при  $x(0)$  и входным сигналом единичного шага в момент  $k = 0$  смоделируйте реакцию замкнутой системы, как было показано в примере главы.

## 4 УПРАВЛЕНИЕ С ПРОГНОЗИРУЮЩЕЙ МОДЕЛЬЮ В МИМО-СИСТЕМАХ

В предыдущем разделе для простоты иллюстрации СУПМ была разработана на основе системы с одним входом и одним выходом. Данная Представленная методология проектирования может быть легко расширена на МИМО-системы. Такая простота обобщения связана с применением описания пространства состояния.

### 4.1 Общее описание модели

Предположим, что объект управления имеет  $m$  входов,  $q$  выходов и  $n_l$  параметров состояния.

Предположим также, что количество выходов меньше или равно количеству входов (т. е.  $q \leq m$ ).

Если количество выходов будет превышать количество входов, будет утрачена возможность независимого управления каждым из измеренных выходов с нулевыми статическими ошибками.

В общей формулировке проблемы прогнозирующего управления примем также во внимание шум и помехи:

$$x_m(k+1) = A_m x_m(k) + B_m u(k) + B_d \omega(k), \quad (33)$$

$$y(k) = C_m x_m(k), \quad (34)$$

где  $\omega(k)$  – возмущение входа, относительно которого предполагается, что оно является последовательностью интегрированного белого шума. Это подразумевает, что входное возмущение  $\omega(k)$  связано с нулевым средним последовательности белого шума  $\epsilon(k)$  через разностное уравнение

$$\omega(k) - \omega(k-1) = \epsilon(k). \quad (35)$$

Заметим, что из (33) справедливо также следующее разностное уравнение:

$$x_m(k) = A_m x_m(k-1) + B_m u(k-1) + B_d \omega(k-1). \quad (36)$$

Определив  $\Delta x_m = x_m(k) - x_m(k-1)$  и  $\Delta u(k) = u(k) - u(k-1)$ , вычитание (36) из (33) приводит к

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) + B_d \epsilon(k). \quad (37)$$

Чтобы связать выход  $y(k)$  с переменной состояния  $\Delta x_m(k)$ , выведем, что

$$\Delta y(k+1) = C_m \Delta x_m(k+1) = C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k) + C_m B_d \epsilon(k),$$

где  $\Delta y(k+1) = y(k+1) - y(k)$ .

Выбирая новый вектор переменных состояния  $x(k) = [\Delta x_m(k)^T \ y(k)^T]^T$ , имеем

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A_m & o_m^T \\ C_m A_m & I_{q \times q} \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k) + \begin{bmatrix} B_d \\ C_m B_d \end{bmatrix} \in(k), \quad (38)$$

$$y(k) = \begin{bmatrix} o_m & I_{q \times q} \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix},$$

где  $I_{q \times q}$  – единичная матрица размерности  $q \times q$ , то есть число выходов;  $o_m$  – нулевая матрица  $q \times n_1$ .

В (38)  $A_m$ ,  $B_m$  и  $C_m$  имеют размерность  $n_1 \times n_1$ ,  $n_1 \times m$  и  $q \times n_1$  соответственно.

Для простоты изложения запишем (38) в виде

$$\begin{aligned} x(k+1) &= Ax(k) + B\Delta u(k) + B_\epsilon \in(k), \\ y(k) &= Cx(k), \end{aligned} \quad (39)$$

где  $A$ ,  $B$  и  $C$  – матрицы, соответствующие формам, приведенным в (38).

В дальнейшем размерность расширенного уравнения пространства состояния принимается равной  $n = n_1 + q$ .

Здесь присутствуют два важных момента, которым необходимо уделить отдельное внимание. Первый связан с собственными значениями расширенной модели. Второй момент связан с реализацией модели пространства состояния. Оба этих момента помогут в понимании описываемой модели.

#### 4.2 Собственные значения расширенной модели

Заметим, что характеристическое полиномиальное уравнение расширенной модели имеет вид

$$p(\lambda) = \det \begin{bmatrix} \lambda I - A_m & o_m^T \\ -C_m A_m & (\lambda - 1)I_{q \times q} \end{bmatrix} = (\lambda - 1)^q \det(\lambda I - A_m) = 0, \quad (40)$$

где было использовано свойство равенства определителя блока нижней треугольной матрицы и произведения определителей матриц на диагонали. Следовательно, собственные значения расширенной модели являются объединением собственных значений модели объекта и  $q$  собственных значений  $\lambda = 1$ . Это означает, что существует  $q$  интеграторов, встроенных в синтезируемую расширенную модель.

Данный подход используется нами для получения интегральных свойств СУПМ.

#### 4.3 Управляемость и наблюдаемость расширенной модели

Поскольку исходная модель объекта дополнена интеграторами и СУПМ проектируется на основе расширенной модели пространства состояния, с точки зрения проектирования системы управления является

важным, что расширенная модель не становится неуправляемой или ненаблюдаемой, особенно в случае неустойчивых динамик системы.

Управляемость является показателем того, что система прогнозирующего управления способна достичь требуемых характеристик управления в замкнутой системе.

Наблюдаемость является свидетельством правильного проектирования наблюдателя.

Однако в случаях, когда основным требованием является устойчивость замкнутой системы, условия могут быть ослаблены до требования стабилизируемости и детектируемости.

Система **стабилизуема**, если ее неуправляемые моды – если таковые имеются – являются устойчивыми. Её управляемые моды могут быть устойчивыми или неустойчивыми.

Система **детектируема**, если ее ненаблюдаемые моды – если таковые имеются, – устойчивы. Её наблюдаемые моды могут быть стабильными или нестабильными.

Под устойчивыми модами здесь понимается, что соответствующие собственные значения находятся строго внутри единичного круга.

Далее (если не специально не указано иного), для достижения желаемых характеристик в замкнутой системе, по умолчанию требуется, чтобы модель была как управляемой, так и наблюдаемой.

В разделе 4.5 приведён пример, иллюстрирующий важность наблюдаемости при разработке наблюдателя.

Поскольку расширенная модель вводит дополнительные интегральные моды, необходимо исследовать, при каких условиях данные дополнительные моды становятся управляемыми. Самый простой способ исследования основан на предположении минимальной реализации модели объекта.

**Определение.** Реализация передаточной функции  $G(z)$  является любым триплетным пространством состояний  $(A, B, C)$  таким, что  $G(z) = C(zI - A)^{-1}B$ . Если такое множество  $(A, B, C)$  существует, то  $G(z)$  называется реализуемой. Реализация  $(A, B, C)$  называется минимальной реализацией передаточной функции, если никакая другая реализация меньшего размера триплета не существует.

Минимальная реализация имеет отличительную особенность, обобщенную в теореме ниже.

**Теорема 4.1.** Минимальная реализация обладает свойствами управляемости и наблюдаемости [86 – 88].

С помощью данной теоремы покажем условия, при которых через факт минимальной реализации расширенная модель является управляемой и наблюдаемой.



**Теорема 4.2.** Предположим, что модель объекта  $(A_m, B_m, C_m)$  управляема и наблюдаема с передаточной функцией  $G_m(z)$  с минимальной реализацией, где

$$G_m(z) = C_m(zI - A_m)^{-1}B_m.$$

Тогда передаточная функция проектируемой расширенной модели (39) имеет вид

$$G(z) = \frac{z}{z-1}G_m(z), \quad (41)$$

и одновременно управляема и наблюдаема тогда и только тогда, когда модель объекта  $G_m(z)$  не имеет нуля при  $z=1$ .

Нулями передаточной функции ММО-системы являются те значения  $z$ , которые обеспечивают потерю ранга матрицы  $G_m(z)$ .

**Доказательство.** Чтобы доказать, что расширенная модель управляема и наблюдаема, необходимо показать, что (41) верно. После этого результаты следуют из минимальной структуры расширенной модели без исключения полюса. Заметим, что для заданной квадратной матрицы  $M$  с блочной структурой

$$M = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix},$$

если  $A_{11}^{-1}$  и  $A_{22}^{-1}$  существуют, то

$$M^{-1} = \begin{bmatrix} A_{11}^{-1} & 0 \\ -A_{22}^{-1}A_{21}A_{11}^{-1} & A_{22}^{-1} \end{bmatrix}. \quad (42)$$

Используя (42), получим

$$G(z) = C(zI - A)^{-1}B, \quad (43)$$

где

$$(zI - A)^{-1} = \begin{bmatrix} (zI_m - A_m)^{-1} & 0 \\ (1 - z^{-1})C_m A_m (zI_m - A_m)^{-1} & (1 - z^{-1})I_q \end{bmatrix}.$$

Подставляя в (39) матрицы  $B$  и  $C$ , из (41) получаем передаточную функцию расширенной модели. В предположении, что модель объекта  $G_m(z)$  не имеет нуля при  $z=1$  и имеет минимальную реализацию, передаточная функция расширенной модели имеет минимальную структуру из (41), поэтому она управляема и наблюдаема.

Для системы с одним входом и одним выходом, в случае если один из нулей передаточной функции находится в точке  $z=1$ , расширенная модель является неуправляемой. Например, если

$$G_m(z) = \frac{(z-1)}{(z-0,6)(z-0,8)},$$

то будет отмена полюса в  $G_m(z)$ , что даёт

$$G_m(z) = \frac{z}{z-1} \frac{(z-1)}{(z-0,6)(z-0,8)}.$$

В случае с одним входом и одним выходом установившийся коэффициент усиления модели объекта равен нулю и не допускает интегрального управления.

Подчеркнем, что количество входов больше или равно количеству выходов ( $m \geq q$ ). Когда количество входов меньше, чем количество выходов, расширенные интегральные моды могут стать неуправляемыми.

При использовании MATLAB минимальная реализация модели пространства состояний достигается за счет сокращения порядка модели. Например, для дискретной передаточной функции

$$G_m(z) = \frac{(z-0,1)}{(z-0,1)(z-0,9)},$$

наблюдается отмена полюса при  $z=0,1$ . Одна из реализаций модели пространства состояний на основе  $G_m(z)$  с использованием функции MATLAB (`tf2ss.m`) имеет две переменных состояния:

$$A_m = \begin{bmatrix} 1 & -0,09 \\ 1 & 0 \end{bmatrix};$$

$$B_m = \begin{bmatrix} 0 \\ 1 \end{bmatrix};$$

$$C_m = [1 \quad -0,1].$$

Это не является минимальной реализацией, так как соответствующая передаточная функция имеет исключение полюсов. Для получения минимальной реализации пространства состояний, в качестве иллюстративного примера может быть использована следующая программа MATLAB:

```
numd=[1 -0.1];
dend=conv([1 -0.1],[1 -0.9]);
sys1=tf(numd,dend);
sys=ss(sys1,'min');
[Am,Bm,Cm,Dm]=ssdata(sys);
```

Минимальная реализация, полученная путём уменьшения порядка модели, имеет вид

$$A_m = 0,9;$$

$$B_m = -0,9806;$$

$$C_m = -1,0198,$$

и имеет только одну переменную состояния, как и ожидалось в рассматриваемом примере для минимальной реализации.

#### 4.4 Расчёт прогнозирующего управления для ММО-систем

Расширение решения прогнозирующего управления довольно простое, и мы должны обратить внимание на размеры состояния, управления и вывода векторов в среде с множеством входных и множеством выходных данных. Определим векторы  $Y$  и  $\Delta U$  как

$$\Delta U = \begin{bmatrix} \Delta u(k_i)^T & \Delta u(k_i + 1)^T & \dots & \Delta u(k_i + N_c - 1)^T \end{bmatrix}^T,$$

$$Y = \begin{bmatrix} y(k_i + 1 | k_i)^T & y(k_i + 2 | k_i)^T & y(k_i + 3 | k_i)^T & \dots & y(k_i + N_p | k_i)^T \end{bmatrix}^T.$$

На основе модели пространства состояния  $(A, B, C)$  вычисляются будущие переменные состояния, последовательно используя набор будущих параметров управления

$$\begin{aligned} x(k_i + 1 | k_i) &= Ax(k_i) + B\Delta u(k_i) + B_d \varepsilon(k_i), \\ x(k_i + 2 | k_i) &= Ax(k_i + 1 | k_i) + B\Delta u(k_i + 1) + B_d \varepsilon(k_i + 1 | k_i) = \\ &= A^2 x(k_i) + AB\Delta u(k_i) + B\Delta u(k_i + 1) + AB_d \varepsilon(k_i) + B_d \varepsilon(k_i + 1 | k_i), \\ &\dots \\ x(k_i + N_p | k_i) &= A^{N_p} x(k_i) + A^{N_p-1} B\Delta u(k_i) + A^{N_p-2} B\Delta u(k_i + 1) + \\ &+ A^{N_p-N_c} B\Delta u(k_i + N_c - 1) + A^{N_p-1} B_d \varepsilon(k_i) + \\ &+ A^{N_p-2} B_d \varepsilon(k_i + 1 | k_i) + \dots + B_d \varepsilon(k_i + N_p - 1 | k_i). \end{aligned}$$

Предполагая, что  $\varepsilon(k)$  представляет собой нулевую среднюю последовательность белого шума, предсказанное значение  $\varepsilon(k_i + i | k_i)$  в будущей выборке  $i$  считаем нулевым.

Прогнозирование переменной состояния и переменной выхода рассчитываются как ожидаемые значения соответствующих переменных. Следовательно, эффект шума для прогнозируемых значений равен нулю.

Фактически, имеем

$$Y = Fx(k_i) + \Phi \Delta U, \quad (44)$$

где

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \dots \\ CA^{N_p} \end{bmatrix};$$

$$\Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix}.$$

Инкрементное оптимальное управление в одном окне оптимизации задается формулой

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T \bar{R}_s r(k_i) - \Phi^T F x(k_i)), \quad (45)$$

где матрица  $\Phi^T \Phi$  имеет размерность  $mN_c \times mN_c$ ;  $\Phi^T F$  имеет размерность  $mN_c \times n$ , а  $\Phi^T \bar{R}_s$  состоит из последних  $q$  столбцов  $\Phi^T F$ . Весовая матрица  $\bar{R}$  является блочной с  $t$  блоками и имеет размерность, равную размерности  $\Phi^T F$ . Сигналы уставки заданы как  $r(k_i) = [r_1(k_i) \ r_2(k_i) \ \dots \ r_q(k_i)]^T$  в виде  $q$  сигналов в системе с множеством выходов.

Исходя из принципа управления с удаляющимся горизонтом, первые  $t$  элементов в  $\Delta U$  используются для формирования инкрементного оптимального управления вида

$$\begin{aligned} \Delta u(k_i) &= \overbrace{[I_m \ o_m \ \dots \ o_m]}^{N_c} (\Phi^T \Phi + \bar{R})^{-1} (\Phi^T \bar{R}_s r(k_i) - \Phi^T F x(k_i)) = \\ &= K_y r(k_i) - K_{mpc} x(k_i), \end{aligned} \quad (46)$$

где  $I_m$  и  $o_m$  соответственно являются единичной и нулевой матрицей с размерностью  $t \times t$ .

#### 4.5 Наблюдатель состояния

При разработке регуляторов с прогнозирующей моделью нами было принято допущение, что информация  $x(k_i)$  доступна в момент  $k_i$ . Это предполагает, что все переменные состояния измеримы. В действительности, в большинстве приложений не все переменные состояния измеримы (или доступны). Некоторые из них невозможно измерить.

Один из подходов в таком случае состоит в выборе переменных состояния, соответствующих входу и выходу, используя специальную

реализацию пространства состояний, и альтернативой является оценка переменной состояния  $x(k)$  из измерения процесса.

«Мягкий» инструмент, используемый для оценки неизвестных переменных состояния на основе процесса измерения, в контексте управления называется наблюдателем.

Концепция наблюдателя широко используется в различных областях науки и техники. А в условиях шумной внешней среды наблюдатель состояния может выполнять функции фильтра, уменьшая влияние шума на измерения.

Нашей целью далее является демонстрация возможности применения наблюдателя при проектировании систем прогнозирующего управления.

#### 4.6 Основы теории наблюдателей

Наблюдатель построен на основе математической модели объекта. Например, предположим, что модель пространства состояния объекта имеет вид

$$x_m(k+1) = A_m x_m(k) + B_m u(k). \quad (47)$$

Далее данная модель может быть использована для вычисления переменной состояния  $\hat{x}_m(k)$ ,  $k=1,2,\dots$ , с начальным условием состояния  $\hat{x}_m(0)$  и входным сигналом  $u(k)$  как

$$\hat{x}_m(k+1) = A_m \hat{x}_m(k) + B_m u(k). \quad (48)$$

Данный подход, по факту, начинает работать после некоторого переходного времени, если модель объекта устойчива и предположение, сделанное выше о начальном состоянии, почти верно.

Какие проблемы могут возникнуть при данном типе подхода? В своей основе он представляет прогнозирование в разомкнутой системе. Ошибка  $\tilde{x}_m(k) = x_m(k) - \hat{x}_m(k)$  удовлетворяет разностному уравнению

$$\tilde{x}_m(k+1) = A_m (x_m(k) - \hat{x}_m(k)) = A_m \tilde{x}_m(k). \quad (49)$$

Для заданной начальной ошибки  $\tilde{x}_m(0) \neq 0$  имеем

$$\tilde{x}_m(k) = A_m^k \tilde{x}_m(0). \quad (50)$$

Здесь необходимо рассмотреть два аспекта:

- 1) Если  $A_m$  имеет все собственные значения внутри единичного круга, то система ошибок (50) устойчива и  $\|\tilde{x}_m(k)\| \rightarrow 0$  при  $k \rightarrow \infty$ , что означает, что наблюдаемая переменная состояния  $\hat{x}_m(k)$  сходится к  $x_m(k)$ .

Однако, если  $A_m$  имеет одно или несколько собственных значений вне единичного круга, то ошибка системы (50) неустойчива и  $\|\tilde{x}_m(k)\| \rightarrow \infty$  при  $k \rightarrow \infty$ , что означает, что прогноз  $\hat{x}_m(k)$  не сходится к  $x_m(k)$ .

Если  $A_m$  имеет одно или несколько собственных значений на единичной окружности, состояние ошибок  $\|\tilde{x}_m(k)\|$  не сходится к нулю.

2) В случае устойчивой модели объекта  $A_m$  мы не имеем «контроль» над скоростью сходимости ошибки  $\|\tilde{x}_m(k)\| \rightarrow 0$ , зависящей от расположения полюсов модели объекта. А именно, если полюса объекта близки к началу координат комплексной плоскости, то ошибка сходится с высокой скоростью к нулю; в противном случае скорость сходимости может быть медленной.

Вопрос заключается в том, как улучшить наблюдение  $x_m(k)$ . Решение состоит в применении обратной связи, где повышение точности наблюдения обеспечивается с помощью сигнала ошибки наблюдения.

Пусть наблюдатель построен с использованием уравнения

$$\hat{x}_m(k+1) = \overbrace{A_m \hat{x}_m(k) + B_m u(k)}^{\text{model}} + \overbrace{K_{ob}(y(k) - C_m \hat{x}_m(k))}^{\text{correction term}}, \quad (51)$$

где  $K_{ob}$  – матрица усиления наблюдателя.

В модели наблюдателя наблюдение переменной состояния  $\hat{x}_m(k+1)$  состоит из двух слагаемых. Первым является исходная модель, а второе слагаемое – поправочный член, зависящий от ошибки между измеренным и прогнозируемым выходом, рассчитанной с помощью наблюдения  $\hat{x}_m(k)$ .

Для выбора коэффициента усиления наблюдателя  $K_{ob}$  рассмотрим уравнение ошибки замкнутого контура. Подставляя  $y(k) = C_m x_m(k)$  в (51), с ошибкой состояния  $\tilde{x}_m(k) = x_m(k) - \hat{x}_m(k)$ , получаем, что

$$\tilde{x}_m(k+1) = A_m \tilde{x}_m(k) - K_{ob} C_m \tilde{x}_m(k) = (A_m - K_{ob} C_m) \tilde{x}_m(k). \quad (52)$$

Теперь, с заданной начальной ошибкой  $\tilde{x}_m(0)$ , имеем

$$\tilde{x}_m(k) = (A_m - K_{ob} C_m)^k \tilde{x}_m(0). \quad (53)$$

Сравнивая отклик ошибки наблюдателя, заданный (53), с прогнозом в разомкнутом контуре (50), нетрудно заметить, что усиление наблюдателя  $K_{ob}$  может быть использовано для управления скоростью сходимости ошибки.

Если объект имеет только один выход, то применяется широко используемый подход, который состоит в том, чтобы поместить собственные значения ошибки матрицы  $A_m - K_{ob} C_m$  разомкнутой системы в нужных координатах комплексной плоскости.

В следующем примере показано, как может быть выбран коэффициент усиления наблюдателя  $K_{ob}$ .

**Пример 4.1.** Рассмотрим линеаризованное уравнение движения простого маятника

$$\frac{d^2\theta}{dt^2} + \omega^2\theta = u, \quad (54)$$

где  $\theta$  – угол отклонения маятника.

Необходимо спроектировать наблюдатель, который будет выполнять оценку угла отклонения маятника для заданных измерений  $\frac{d\theta}{dt}$ .

Предположим, что  $\omega = 2$  рад/сек и интервал выборки  $\Delta t = 0,1$  сек. Выбранные полюса наблюдателя равны 0,1 и 0,2.

Необходимо сравнить оценку в разомкнутом контуре с оценкой, полученной с помощью наблюдателя.

**Решение.** Пусть  $x_1(t) = \theta$  и  $x_2(t) = \dot{\theta}$ . Учитывая уравнение движения (54), соответствующая модель пространства состояния имеет вид

$$\begin{aligned} \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \\ y(t) &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}. \end{aligned} \quad (55)$$

При  $\omega = 2$  рад/сек и интервале выборки равном  $\Delta t = 0,1$  сек, дискретная модель пространства состояния имеет вид

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0,9801 & 0,0993 \\ -0,3973 & 0,9801 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0,0050 \\ 0,09930 \end{bmatrix} u(k), \quad (56)$$

$$y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}. \quad (57)$$

Прежде всего, выясним, что даст реализация наблюдения за углом отклонения  $\theta(x_1)$  только на базе колебательной модели.

Предположим, что входной сигнал  $u(k) = 0$  и начальные условия переменных состояния равны соответственно  $\theta(0) = x_1(0) = 1$  и  $\dot{\theta}(0) = x_2(0) = 0$ . Траектории движения как для  $\theta$ , так и для  $\dot{\theta}$  показаны на рисунке 4.1а. Оба сигнала синусоидальны.

Предположим теперь, что начальные условия переменных состояния заданы как  $\hat{x}_1(0) = 0,3$  и  $\hat{x}_2(0) = 0$ . Рассчитанные с помощью модели пространства состояний (48) оценки  $\theta$  и  $\dot{\theta}$  показаны на рисунке 4.1а, на фоне действительной траектории.

Видно, что оценка  $\theta$ , обозначенная как  $\hat{x}_1$ , не близка к действительной  $\theta$  (верхний график рисунка 4.1а).

Данное исследование показало, что использование только модели объекта недостаточно для прогнозирования угла отклонения маятника.

Теперь спроектируем и реализуем наблюдатель угла маятника. Предположим, что коэффициент усиления наблюдателя  $K_{ob} = [j_1 \ j_2]^T$ .

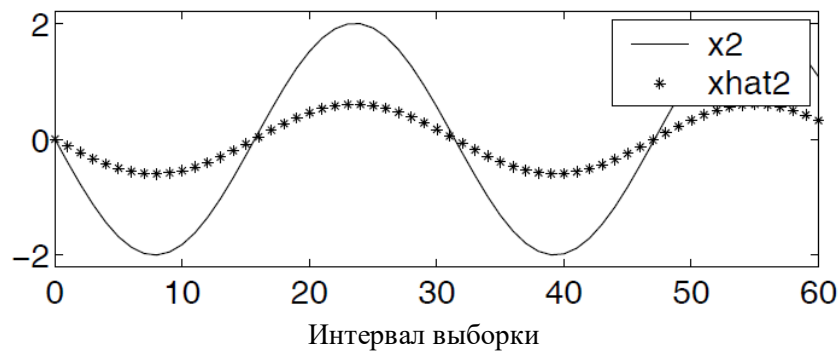
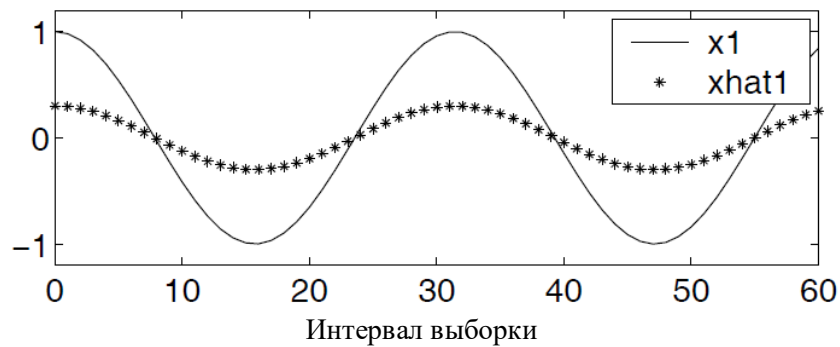
Характеристический полином замкнутой системы для наблюдателя имеет вид

$$\det \left( \lambda I - \begin{bmatrix} 0,9801 & 0,0993 - j_1 \\ -0,3973 & 0,9801 - j_2 \end{bmatrix} \right) =$$

$$= (\lambda - 0,9801)(\lambda + j_2 - 0,9801) - 0,3973 \times (j_1 - 0,0993),$$

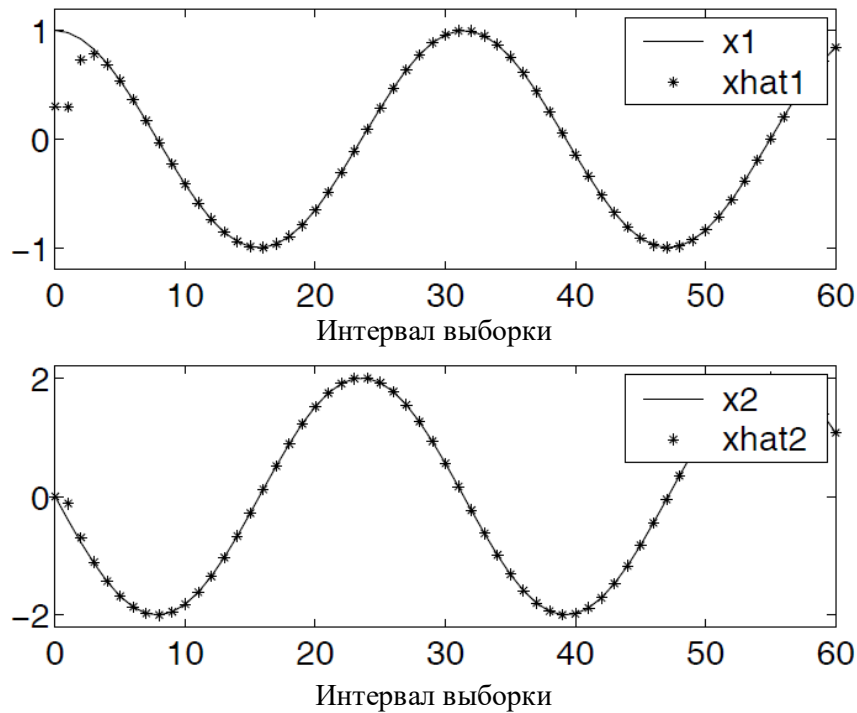
и принят равным желаемому характеристическому полиному замкнутой системы  $(\lambda - 0,1)(\lambda - 0,2)$ , то есть,

$$(\lambda - 0,9801)(\lambda + j_2 - 0,9801) - 0,3973 \times (j_1 - 0,0993) = (\lambda - 0,1)(\lambda - 0,2).$$



а) Оценка без наблюдателя





б) Оценка с наблюдателем

Рисунок 4.1 – Результаты наблюдения за траекторией маятника [102]

Решение полиномиального уравнения дает коэффициент усиления наблюдателя  $j_1 = -1,6284$  и  $j_2 = 1,6601$ .

Оценка угла выполняется с использованием уравнения наблюдателя вида

$$\begin{bmatrix} \hat{x}_1(k+1) \\ \hat{x}_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.9801 & 0.0993 \\ -0.3973 & 0.9801 \end{bmatrix} \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \end{bmatrix} + K_{ob}(x_2(k) - \hat{x}_2(k)), \quad (58)$$

с начальным условием  $\hat{x}_1(0) = 0,3$  и  $\hat{x}_2(0) = 0$ .

На рисунке 4.1б показано, что наблюдаемый угол сходится к истинному углу примерно за три шага.

#### 4.7 Определение наблюдаемости

Модель переменной состояния динамической системы называется полностью наблюдаемой, если для любого времени выборки  $k_0$  существует время выборки  $k_1 > k_0$  такое, что знание выхода  $y(k)$  и входа  $u(k)$  во временном интервале  $k_0 \leq k \leq k_1$  достаточно для определения начального состояния  $x_m(k_0)$  и, как следствие,  $x_m(k)$ , для всех  $k$  между  $k_0$  и  $k_1$ .

Необходимое и достаточное условия полной наблюдаемости линейной дискретной по времени системы выполняются, если матрица наблюдаемости

$$L_0 = \begin{bmatrix} C_m \\ C_m A_m \\ C_m A_m^2 \\ \vdots \\ C_m A_m^{n-1} \end{bmatrix}$$

имеет ранг  $n$ , где  $n$  – размерность модели переменных состояния.

**Пример 4.2.** Двигатель постоянного тока может быть описан моделью второго порядка с интегратором и одной постоянной времени (рисунок 4.2).

Входным сигналом является напряжение питания двигателя, а выходным – положение вала.

Постоянная времени соответствует механической части системы.

Динамикой, обусловленной электрическими контурами двигателя, пренебрегается, поскольку они имеют малые постоянные времени.

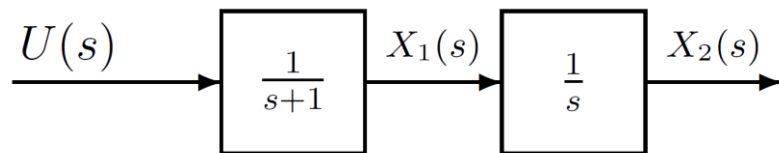


Рисунок 4.2 – Модель двигателя постоянного тока

Приняв за переменную  $x_1$  угловую скорость, а за  $x_2$  – угловое положение вала двигателя, получим уравнение пространства состояний для непрерывного времени

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t). \quad (59)$$

Предполагая, что скорость вращения вала доступна для измерения, получим

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}.$$

Модель дискретизируется с интервалом выборки  $\Delta t = 0,1$ , в результате чего получаем

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.9048 & 0 \\ 0.0952 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0.0952 \\ 0.0048 \end{bmatrix} u(k), \quad (60)$$

$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}.$$

Убедимся, что модель дискретного времени (60) не является наблюдаемой, и, как следствие, система наблюдателей с замкнутым контуром имеет полюс в 1.

**Решение.** Система с разомкнутым контуром имеет собственные значения в 0,9048 и 1. Матрица наблюдаемости

$$L_o = \begin{bmatrix} C_m \\ C_m A_m \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0.9048 & 0 \end{bmatrix},$$

где  $\det(L_o) = 0$ .

Таким образом, пара  $(C_m, A_m)$  не является наблюдаемой.

Исследуем данный пример, чтобы узнать последствия ненаблюдаемости системы.

Предположим, что  $K_{ob} = [j_1 \ j_2]^T$ . Тогда ошибка наблюдателя замкнутой системы имеет вид

$$\tilde{x}(k+1) = \left[ \begin{bmatrix} 0.9048 & 0 \\ 0.0952 & 1 \end{bmatrix} - \begin{bmatrix} j_1 \\ j_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \right] \tilde{x}(k). \quad (61)$$

Предположим, что мы принимаем искомые собственные значения системы наблюдателей равными 0,1 и 0,2 (мы также называем собственные значения системы наблюдателей замкнутыми полюсами наблюдателя).

Тогда характеристический полином замкнутой системы равен

$$\det \begin{bmatrix} \lambda - 0,9048 + j_1 & 0 \\ -0,0952 & \lambda - 1 \end{bmatrix} = (\lambda - 0,9048 + j_1)(\lambda - 1). \quad (62)$$

При синтезе наблюдателя было принято, что фактический характеристический полином замкнутой системы равен желаемому характеристическому полиному замкнутой системы, откуда был вычислен вектор коэффициентов усиления наблюдателя. В данном случае мы получили бы

$$(\lambda - 0,9048 + j_1)(\lambda - 1) = (\lambda - 0,1)(\lambda - 0,2).$$

Заметим, что второй полюс при  $\lambda = 1$  в (62) не может быть перемещен независимо от того, какой выбор мы сделаем для  $j_2$ , просто потому, что полюс замкнутой системы не зависит от коэффициента усиления наблюдателя.

Это является следствием отсутствия наблюдаемости пары  $(C_m, A_m)$ . Если полюс, который нельзя изменить с помощью наблюдателя, асимптотически устойчив, то система не наблюдаема, однако она детектируема.

В данном конкретном примере ненаблюдаемый полюс находится на единичном круге, поэтому система даже не детектируема.

Таким образом, если мы измеряем угловую скорость вала двигателя, то по данному измерению нельзя точно оценить угловое положение вала двигателя.

#### 4.8 Фильтр Калмана

Если пара  $(A_m, C_m)$  наблюдаема, то в рассмотренном случае системы с одним выходом для определения  $K_{ob}$  может быть использована стратегия расположения полюсов таким образом, что собственные значения наблюдателя (то есть матрицы  $A_m - K_{ob}C_m$ ) имеют желаемое расположение.

Для системы с множеством выходов  $K_{ob}$  может быть вычислен рекурсивно с использованием фильтра Калмана.

Фильтры Калмана рассматриваются далее в стохастической постановке. Положим, что

$$\begin{aligned} x_m(k+1) &= A_m x_m(k) + B_m u(k) + d(k), \\ y(k) &= C_m x_m(k) + \xi(k), \end{aligned} \quad (63)$$

где ковариационные матрицы  $d$  и  $\xi$  соответственно определяются как

$$\begin{aligned} E\{d(k)d(\tau)^T\} &= \Theta \delta(k - \tau), \\ E\{\xi(k)\xi(\tau)^T\} &= \Gamma \delta(k - \tau), \end{aligned}$$

где  $\delta(k - \tau) = 1$ , если  $k = \tau$  и  $\delta(k - \tau) = 0$ , если  $k \neq \tau$ .

Оптимальный коэффициент усиления  $K_{ob}$  наблюдателя определяется рекурсивно для  $i = 0, 1, \dots$  через

$$K_{ob}(i) = A_m P(i) C_m^T (\Gamma + C_m P(i) C_m^T)^{-1} \quad (64)$$

и

$$P(i+1) = A_m \{P(i) - P(i) C_m^T (\Gamma + C_m P(i) C_m^T)^{-1} C_m P(i)\} A_m^T + \Theta. \quad (65)$$

Точнее,  $P(0)$  удовлетворяет условию

$$E\{[x(0) - \hat{x}(0)][x(0) - \hat{x}(0)]^T\} = P(0).$$

Предполагая, что система  $(A_m, C_m)$  детектируема по выходу  $y(k)$  (то есть нет неустойчивых состояний, отклик которых не может быть «замечен» на выходе) и  $(A_m, \theta^{1/2})$  стабилизируемо, то, при  $k \rightarrow \infty$ , установившиеся решения (64) и (65) удовлетворяют алгебраическому дискретному уравнению Риккати

$$P(\infty) = A_m \{P(\infty) - P(\infty) C_m^T (\Gamma + C_m P(\infty) C_m^T)^{-1} C_m P(\infty)\} A_m^T + \Theta, \quad (66)$$

и

$$K_{ob}(\infty) = A_m P(\infty) C_m^T (\Gamma + C_m P(\infty) C_m^T)^{-1}. \quad (67)$$

Кроме того, собственные значения  $A_m - K_{ob}(\infty)C_m$  гарантированно находятся внутри единичного круга (то есть устойчивы).

Во избежание путаницы подчеркнём, что в реальном времени итерационное решение уравнения Риккати (65) искать не требуется. В приложениях СУПМ коэффициент усиления наблюдателя рассчитывается автономно (не в режиме реального времени).

#### 4.9 Настройка динамики наблюдателя

Во многих случаях ковариационные матрицы  $\theta$  и  $\Gamma$ , соответствующие характеристикам возмущений, неизвестны. Таким образом, на практике  $\theta$ ,  $\Gamma$  и начальное  $P(0)$  выбираются для вычисления коэффициента усиления наблюдателя с помощью итерационного решения уравнения Риккати, выполняемого до тех пор, пока решение не сойдется к постоянной матрице. Затем полученная замкнутая система анализируется относительно: расположения собственных значений, содержащихся в  $A_m - K_{ob}C_m$ ; переходной характеристики наблюдателя; робастности и влияния шума на характеристику.

Элементы ковариационной матрицы изменяются до тех пор, пока не будет получен желаемый результат. Такая процедура проб и ошибок может занимать длительное время, неудобна и, таким образом, является одной из проблем, с которыми мы сталкиваемся при использовании многомерной системы на базе фильтра Калмана. Однако в некоторых случаях можно указать область, в которой должны находиться полюса замкнутой системы ошибок наблюдателя, и применить это в решении.

Рассмотрим простой подход, имеющий аналогию с классическим подходом Андерсона и Мура [112 – 113], в котором полюсы наблюдателей с замкнутым контуром назначаются внутри окружности с заданным радиусом  $\alpha$  ( $0 < \alpha < 1$ ). Приведем ниже краткое изложение данной процедуры.

Пусть ошибка наблюдаемого состояния  $\tilde{x}(k) = x(k) - \hat{x}(k)$ . Тогда система ошибок наблюдателя имеет вид

$$\tilde{x}(k+1) = (A_m - K_{ob}C_m)\tilde{x}(k). \quad (68)$$

Выполним преобразование  $\hat{A}_m = \frac{A_m}{\alpha}$  и  $\hat{C}_m = \frac{C_m}{\alpha}$ , где  $0 < \alpha < 1$ , что приводит к преобразованной системе вида

$$\tilde{x}_t(k+1) = \frac{1}{\alpha}(A_m - \hat{K}_{ob}C_m)\tilde{x}_t(k) = (\hat{A}_m - \hat{K}_{ob}\hat{C}_m)\tilde{x}_t(k). \quad (69)$$

Решение итерационных уравнений (64) и (65) или стационарного уравнения Риккати (66), с использованием  $\hat{A}_m$  и  $\hat{C}_m$  для замены матриц  $A_m$  и  $C_m$ , обеспечивает нахождение собственных чисел  $\hat{A}_m - \hat{K}_{ob}(\infty)\hat{C}_m$  внутри единичного круга (то есть устойчиво).

Полученный коэффициент усиления наблюдателя  $\hat{K}_{ob}$  затем применяется к исходной системе наблюдателей (68), что приводит к характеристическому уравнению замкнутой системы вида

$$\det(zI - (A_m - \hat{K}_{ob}C_m)) = \det(zI - (\hat{A}_m - \hat{K}_{ob}\hat{C}_m)) \cdot \alpha = 0. \quad (70)$$

Таким образом, собственные значения  $(A_m - \hat{K}_{ob}C_m)$  равны собственным значениям  $\hat{A}_m - \hat{K}_{ob}\hat{C}_m$ , умноженным на коэффициент  $\alpha$ , что

обеспечивает нахождение собственных значений системы ошибок наблюдателя с  $\hat{K}_{ob}$  внутри круга радиуса  $\alpha$ .

Данная процедура через выбор  $\alpha$  обеспечивает прямую связь с динамикой наблюдателя. Процедура проб и ошибок может быть сокращена до выбора подходящих  $\alpha$ ,  $\theta$  и  $\Gamma$ , для достижения желаемых характеристик в данной замкнутой системе.

#### 4.10 Прогнозирующее управление с оценкой состояния в MATLAB

При реализации прогнозирующего управления, наблюдатель используется в случаях, когда переменная состояния  $x(k_i)$  в момент времени  $k_i$  недоступна для измерения. В общем, переменная состояния  $x(k_i)$  оценивается с помощью наблюдателя вида

$$\hat{x}(k_i + 1) = A\hat{x}(k_i) + B\Delta u(k_i) + K_{ob}(y(k_i) - C\hat{x}(k_i)). \quad (71)$$

Обратим внимание, что при реализации прогнозирующего управления с использованием наблюдателя сигнал управления равен  $\Delta u(k_i)$ , а матрицы  $(A, B, C)$  следуют из расширенной модели, используемой при проектировании СУПМ.

Полученное наблюдение  $\hat{x}(k_i)$ , заменяющее  $x(k_i)$ , используется в модифицированном законе прогнозирующего управления для вычисления  $\Delta U$ , путем минимизации

$$J = (R_s - F\hat{x}(k_i))^T (\bar{R}_s r(k_i) - F\hat{x}(k_i)) - 2\Delta U^T \Phi^T (R_s - F\hat{x}(k_i)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U, \quad (72)$$

где  $\bar{R}_s$ ,  $F$ ,  $\Phi$ ,  $\bar{R}$  и  $\Delta U$  были определены в (44) и (45).

Оптимальное решение  $\Delta U$  имеем в виде

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - F\hat{x}(k_i)). \quad (73)$$

Кроме того, применение принципа управления с удаляющимся горизонтом приводит к оптимальному решению  $\Delta u(k_i)$  в момент времени  $k_i$ , вида

$$\Delta u(k_i) = K_y r(k_i) - K_{mpc} \hat{x}(k_i), \quad (74)$$

который является стандартным законом управления с обратной связью с наблюдением  $x(k_i)$ .

Структура замкнутой системы с обратной связью по состоянию показана на рисунке 4.3.

Остаются вопросы касательно характеристического уравнения замкнутой системы и, следовательно, собственных значений замкнутой системы. Чтобы исследовать данные вопросы, запишем уравнение пространства состояний замкнутой системы вида

$$x(k+1) = Ax(k) + B\Delta u(k) = Ax(k) + BK_y r(k) - BK_{mpc} \hat{x}(k), \quad (75)$$

где  $\Delta u(k)$  заменено на (74).

Обратим внимание, что уравнение ошибки наблюдателя в замкнутой системе

$$\tilde{x}(k+1) = (A - K_{ob}C)\tilde{x}(k), \quad (76)$$

где  $\tilde{x}(k) = x(k) - \hat{x}(k)$ .

После замены  $\hat{x}(k)$  на  $x(k) - \tilde{x}(k)$  уравнение (75) может быть записано как

$$x(k+1) = (A - BK_{mpc})x(k) - BK_{mpc}\tilde{x}(k) + BK_y r(k). \quad (77)$$

Для (76) и (77) получаем

$$\begin{bmatrix} \tilde{x}(k+1) \\ x(k+1) \end{bmatrix} = \begin{bmatrix} A - K_{ob}C & o_{n \times n} \\ -BK_{mpc} & A - BK_{mpc} \end{bmatrix} \begin{bmatrix} \tilde{x}(k) \\ x(k) \end{bmatrix} + \begin{bmatrix} o_{n \times m} \\ BK_y \end{bmatrix} r(k), \quad (78)$$

где  $o_{n \times n}$  – нулевая матрица  $n \times n$ , а  $o_{n \times m}$  – нулевая матрица  $n \times m$ .

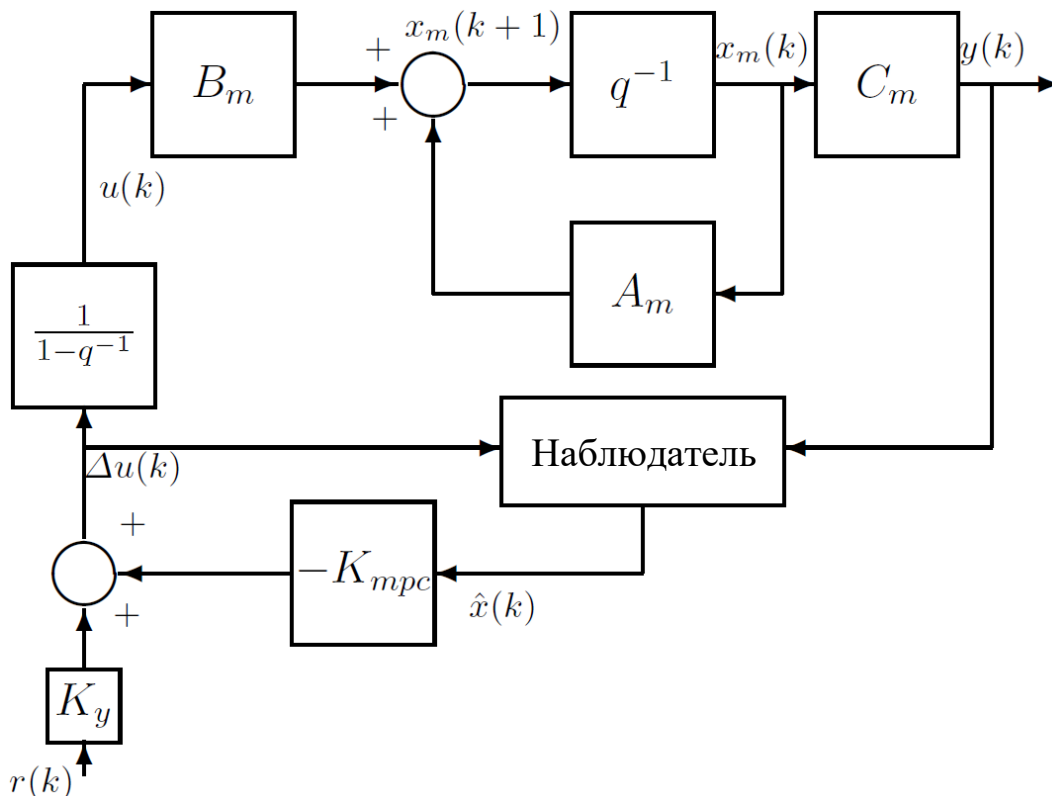


Рисунок 4.3 – Структурная схема прогнозирующей модели дискретной системы управления с наблюдателем

Характеристическое уравнение пространства состояний замкнутой системы определяется как

$$\det \left[ \lambda I - \begin{bmatrix} A - K_{ob}C & o_{n \times n} \\ -BK_{mpc} & A - BK_{mpc} \end{bmatrix} \right] = 0,$$

что эквивалентно

$$\det(\lambda I - (A - K_{ob}C)) \det(\lambda I - (A - BK_{mpc})) = 0,$$

так как системная матрица в (78) имеет более низкую блочную треугольную структуру. Фактически это означает, что замкнутая система прогнозирующего управления с оценкой состояния имеет два независимых характеристических уравнения:

$$\det(\lambda I - (A - K_{ob}C)) = 0, \quad (79)$$

$$\det(\lambda I - (A - BK_{mpc})) = 0. \quad (80)$$

Так как собственные числа замкнутой системы являются решениями характеристических уравнений, (79) и (80) показывают, что множество собственных чисел комбинированной замкнутой системы состоит из собственных чисел прогнозирующей системы управления и собственных чисел наблюдателя. Это означает, что законы прогностического управления и наблюдателя могут работать независимо (или отдельно друг от друга), и, в то же время, если они используются совместно, то собственные значения остаются неизменными.

**Пример 4.3.** Пусть, расширенная модель системы с двумя интеграторами (см. пример 1.1) имеет вид [102 – 103]

$$\begin{aligned} x(k+1) &= Ax(k) + B\Delta u(k), \\ y(k) &= Cx(k), \end{aligned} \quad (81)$$

$$\text{где } A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}; B = \begin{bmatrix} 0,5 \\ 1 \\ 0,5 \end{bmatrix}; C = [0 \quad 0 \quad 1].$$

Необходимо разработать систему прогнозирующего управления с наблюдателем состояния и смоделировать реакцию замкнутой системы на изменение уставки.

Примем исходные данные:  $N_c = 5$ ,  $N_p = 30$ , весовой коэффициент сигнала управления  $r_w = 10$ .

Наблюдатель синтезируется с использованием метода назначения полюсов, где полюсы наблюдателя замкнутой системы равны 0,01; 0,0105; 0,011, что соответствует быстрому динамическому отклику наблюдателя.

**Решение.** Разомкнутая система имеет три собственных значения, равных 1, два из которых определяются системой с двумя интеграторами и один – структурой прогнозирующего регулятора.

Используем команду MATLAB 'place' и введём текст программы MATLAB, синтезирующей вектор коэффициента усиления наблюдателя  $K_{ob}$ .

```
Pole=[0.01 0.0105 0.011];
K_ob=place(A',C',Pole)';
```



где  $A'$ ,  $C'$  – транспонированные матрицы  $A^T$  и  $C^T$ .

Транспонирование необходимо вследствие того, что программа MATLAB 'place' была написана для проектирования регулятора. Используя данную программу, мы используем двойственную связь между регулятором и наблюдателем.

Результурующий коэффициент усиления наблюдателя

$$K_{ob} = [1,9685 \quad 0,9688 \quad 2,9685]^T.$$

При указанном наборе параметров качества коэффициент усиления обратной связи по состоянию составляет  $K_{прс} = [0,8984 \quad 1,3521 \quad 0,4039]$ , что фактически дает набор собственных значений замкнутой системы, при  $0,3172 \pm j0,4089$  и  $0,3624$ .

На рисунке 4.4 показана реакция замкнутой системы на ступенчатое изменение уставки. Как можно видеть, выходной сигнал замкнутой системы следует за изменением заданного значения, а сигнал управления стремится к нулю, так как в системе присутствуют интеграторы.

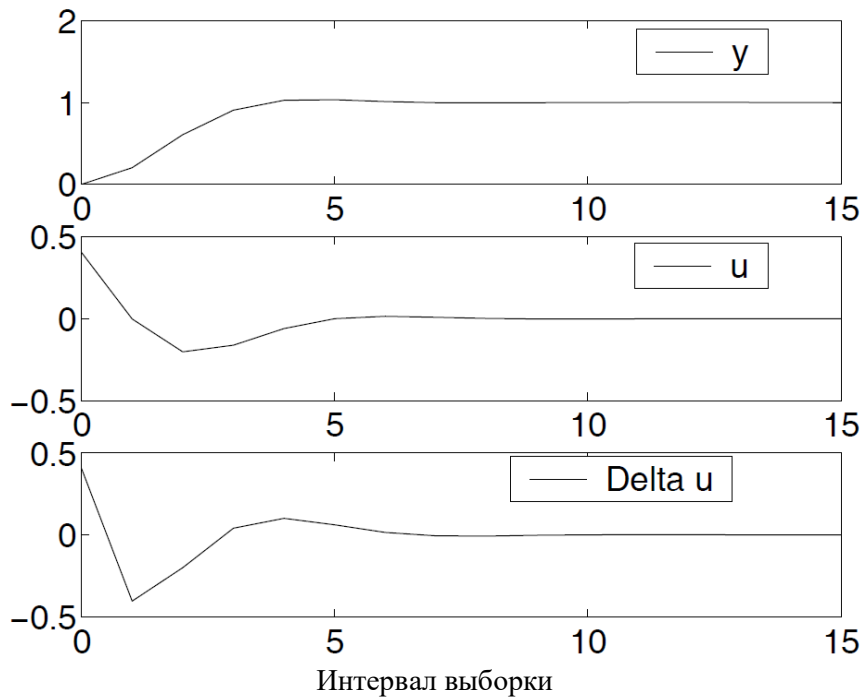


Рисунок 4.3 – Прогнозирующая система управления с двойным интегратором

## 4.11 Задания для самостоятельной работы

### Задача 4.1.

Сигнал дискретного времени  $f_0(k) = \beta(k - 10)^2$  искажается белым шумом  $\epsilon(k)$ , где  $\beta$  неизвестно.

**Необходимо:**

Синтезировать наблюдатель, оценивающий неизвестный коэффициент  $\beta$  из измерения шума  $f(k) = f_0(k) + \epsilon(k)$ .

Полюсы замкнутой системы наблюдателя располагаются на  $0,1$ ;  $0,15$ ;  $0,2$ .

Подсказка: примите

$$x_1(k) = \beta(k - 10)^2; x_2(k) = \beta(k - 10); x_3(k) = \beta.$$

### **Задача 4.2.**

Слабо демпфированная механическая система описана передаточной функцией  $G(s)$  модели непрерывного времени, где

$$G(s) = \frac{0,1}{s^2 + 2\xi\omega_0 s + \omega_0^2},$$

с коэффициентом затухания  $\xi = 0,001$  и  $\omega_0 = 1$ .

**Необходимо:**

- а) Получить дискретную передаточную функцию с интервалом дискретизации  $\Delta t = 0,5$ .
- б) Создать прогнозирующую систему управления с наблюдателем, обеспечивающую компенсацию постоянного возмущения на входе и обработку ступенчатого изменения уставки с нулевой статической ошибкой.  
Поскольку система с разомкнутым контуром недостаточно демпфирована, то есть имеет слабозатухающие колебания, то для достижения желаемого качества требуются большие горизонты прогнозирования и управления.  
Выберите горизонт прогнозирования  $N_p = 60$ , а горизонт управления  $N_c = 20$ .  
Весовая матрица  $\bar{R} = r_\omega I$ , при  $r_\omega = 0.1$ .  
Полюсы наблюдателя располагаются на  $0,1; 0,2; 0,3$ .
- в) Смоделировать работу замкнутой системы управления при единичном ступенчатом изменении уставки на входе, при  $k = 0$ , и ступенчатом единичном входном возмущении, при  $k = 100$ .

### **Задача 4.3.**

*Важным вопросом проектирования является взаимное влияние робастности системы прогнозирующего управления и неучтенной динамики системы.*

#### **Необходимо:**

- а) Без изменения системы прогнозирующего управления ввести время запаздывания трёх выборок в передаточную функцию системы  $G(z)$  в задаче 4.2.*
- б) Смоделировать и исследовать качество замкнутой системы с условиями, идентичными указанным в задании а).*
- в) Выяснить, устойчива ли замкнутая система прогнозирующего управления? Если нет, необходимо вернуться к исходной задаче 4.2 и увеличивать  $r_w$  весовой матрицы до тех пор, пока замкнутая система не станет устойчивой. Проанализируйте свои наблюдения.*

#### **Задача 4.4.**

Убедитесь, что модель пространства состояний двигателя постоянного тока является наблюдаемой, когда положение вала двигателя принимается за измерение и уравнение пространства состояний имеет вид

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.9048 & 0 \\ 0.0952 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0.0952 \\ 0.0048 \end{bmatrix} u(k),$$
$$y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix},$$

где  $x_1$  – скорость, а  $x_2$  – положение вала двигателя.

**Необходимо:**

- а) Расширить дискретную модель системы интегратором и проверить управляемость и наблюдаемость расширенной модели.
- б) Приняв  $N_p = 10$ ,  $N_c = 4$ , и  $\bar{R} = 0,1I$ , вычислить матрицы коэффициентов усиления  $K_y$  и  $K_{mpc}$  для прогнозирующей системы управления.  
Где собственные значения замкнутой системы?
- в) Спроектировать наблюдатель, располагая полюсы наблюдателя замкнутой системы в  $0,1$ ;  $0,2$  и  $0,3$ .
- г) Смоделировать прогнозирующую систему управления оценкой состояния, используя заданное значение сигнала  $r(k) = 2$  для всех  $k$ . После этого создайте входное падение напряжения путем добавления константы  $d = -0,5$  управляющего сигнала при  $k = 20$ , и смоделируйте, как система прогнозирующего управления отклоняет возмущения и поддерживает на выходе заданное  $r = 2$ .

## ПРИЛОЖЕНИЕ

### Решение линейных матричных неравенств (ЛМН)

В данном разделе приводится краткое введение по линейным матричным неравенствам, широко использовавшиеся для решения проблем, описанных в учебнике.

**П.1 Введение в ЛМН.** ЛМН – матричные неравенства, линейные или аффинные в наборе матричных переменных. Они являются, по существу, выпуклыми ограничениями, и поэтому многие проблемы оптимизации с выпуклыми целевыми функциями и ограничениями ЛМН можно легко решить эффективно, используя много существующих программ. Этот метод был очень популярен среди специалистов по системам управления в последние годы. Это связано с тем, что множество проблем управления можно сформулировать как проблемы ЛМН.

ЛМН имеет следующую форму:

$$F(x) = F_0 + x_1 F_1 + \dots + x_n F_n = F_0 + \sum_i^n x_i F_i > 0, \quad (\text{П1})$$

где  $x \in R^m$  – вектор переменных решения и  $F_0, F_1, \dots, F_n$  заданы постоянными симметричными вещественными матрицами, т. е.  $F_i = F_i^T, i = 0, \dots, n$ .

Символ неравенства в уравнении  $F(x)$  положительно определен, т. е.  $u^T F(x) u > 0$  для всех ненулевых  $u \in R^n$ . Это матричное неравенство линейно по переменным  $x_i$ .

В качестве примера рассмотрим неравенство Ляпунова

$$A^T P + P A < 0, \quad (\text{П2})$$

где  $A \in R^{n \times n}$  задано, а  $X = X^T$  – переменная решения, которая может быть выражена в виде ЛМН (П1) следующим образом: пусть  $P_1, P_2, \dots, P_m$  – базис для симметричных  $n \times n$  матриц ( $m = n(n+1)/2$ ), то возьмем  $F_0 = 0$  и  $F_i = -A^T P_i - P_i A$ .

### П.1.1 Приемы, используемые в матричных неравенствах

Хотя многие проблемы управления могут быть сформулированы как проблемы ЛМН, некоторые из этих проблем приводят к нелинейным матричным неравенствам. Есть определенные способы, которые могут быть использованы для преобразования этих нелинейных неравенств в подходящие формы ЛМН. Некоторые из способов, которые часто используются в управлении, описываются здесь подходящими примерами.

**1. Изменение переменных.** Определяя новые переменные, иногда можно линеаризовать нелинейные матричные неравенства.

*Пример А.1.* Синтез контроллера обратной связи состояния. Задача состоит в том, чтобы определить матрицу  $F \in R^{m \times n}$  такую, что все собственные значения матрицы  $A + BF \in R^{n \times n}$  лежат в открытой левой половине комплексной плоскости. С помощью теоремы Ляпунова можно показать, что это эквивалентно найти матрицу  $F$  и положительно определенную матрицу  $P \in R^{n \times n}$  такую, что имеет место неравенство:

$$A + (BF)^T P + P(A + BF) < 0 \quad (\text{П3})$$

или

$$A^T P + PA + F^T B^T P + PBF < 0. \quad (\text{П4})$$

Заметим, что члены с произведениями  $F$  и  $P$  являются нелинейными или билинейными. Умножим каждую сторону этого уравнения на  $Q = P^{-1}$ . Это дает

$$QA^T + AQ + QF^T B^T + BFQ < 0. \quad (\text{П5})$$

Это новое матричное неравенство в переменных  $Q > 0$  и  $F$ . Но оно все еще нелинейно. Определим вторую новую переменную  $L = FQ$ . Это дает

$$QA^T + QA + L^T B^T + BL < 0. \quad (\text{П6})$$

Это дает возможность ЛМН решать проблему с новыми переменными  $Q > 0$  и  $L \in R^{m \times n}$ . После решения этой ЛМН матрица обратной связи  $F$  и переменная Ляпунова  $P$  могут быть получены из  $F = LQ^{-1}$  и  $P = Q^{-1}$ . Это показывает, что, изменив переменные, мы можем получить ЛМН из нелинейного матричного неравенства.

**2. Дополнения Шура** Формула Шура используется для преобразования нелинейных неравенств выпуклого типа в ЛМН. Это говорит о том, что ЛМН

$$\begin{bmatrix} Q(x) & S(x) \\ S(x)^T & R(x) \end{bmatrix} < 0, \quad (\text{П7})$$

где  $Q(x) = Q(x)^T$ ,  $R(x) = R(x)^T$  и  $S(x)$  аффинно зависит от  $x$ , что эквивалентно

$$R(x) < 0, Q(x) - S(x)R(x)^{-1}S(x)^T < 0. \quad (\text{П8})$$

Другими словами, множество нелинейных неравенств (П8) можно преобразовать в ЛМН (П7).

*Пример 2.* Рассмотрим следующее матричное неравенство:

$$A^T P + PA + PBR^{-1}B^T P + Q < 0, \quad (\text{П9})$$

где  $P = P^T > 0$  и  $R > 0$ , что эквивалентно

$$\begin{bmatrix} A^T P + PA + Q & PB \\ B^T P & -R \end{bmatrix} < 0. \quad (\text{П10})$$

**3. S-процедура.** Эта процедура принимается, когда мы хотим объединить несколько квадратичных неравенств в одно единственное неравенство. Во многих проблемах проектирования управления мы хотели бы удостовериться, что одна квадратичная функция от  $x \in R^m$  такова, что

$$F_0(x) \leq 0, F_0(x) := x^T A_0 x + 2b_0 x + c_0 \quad (\text{П11})$$

всякий раз, когда некоторые другие квадратичные функции являются положительными полуопределенными, т.е.

$$F_i(x) \geq 0, F_i(x) := x^T A_i x + 2b_i x + c_i \quad i \in (1, 2, \dots, q). \quad (\text{П12})$$

*Пример.* Чтобы проиллюстрировать S-процедуру, рассмотрим простой случай  $i = 1$ . Нам нужно гарантировать, что  $F_0(x) \leq 0$  для всех  $x$  таких, что  $F_1(x) \geq 0$ .

Если существует положительный (или нулевой) скаляр  $\tau$  такой, что

$$F_{aug}(x) := F_0(x) + \tau F_1(x) \leq 0 \quad \forall x, \text{ s.t. } F_1(x) \geq 0, \quad (\text{П13})$$

то наша цель достигнута. Это связано с тем, что  $F_{aug}(x) \leq 0$  означает, что  $F_0(x) \leq 0$ , если  $\tau F_1(x) \geq 0$  так как  $F_0(x) \leq F_{aug}(x)$ , при  $F_1(x) \geq 0$ .

Расширяя данный результат  $q$  ограничений неравенства получим следующее:

условие  $F_0(x) \leq 0$  (при  $F_i(x) \geq 0$ ) выполняется, если

$$F_0(x) + \sum_{i=1}^q \tau_i F_i(x) \leq 0, \tau_i \geq 0. \quad (\text{П14})$$



## П.1.2 Решение ЛМН с помощью инструментальной панели MATLAB

Набор инструментов ЛМН MATLAB предоставляет набор полезных функций для решения ЛМН. Некоторые из этих функций обсуждаются здесь с примерами кодов.

*Шаг 1. Инициализация.* Вначале инициализируйте описание ЛМН с помощью команды `setlmis ([ ])`. Обратите внимание, что эта функция не принимает никаких параметров.

*Шаг 2. Определение переменных принятия решений.* Затем необходимо определить переменные решения, то есть неизвестные переменные задачи ЛМН. В качестве примера рассмотрим ЛМН  $C^T X C < 0$ , где  $C$  – постоянная матрица, а  $X$  – матрица переменных решений. Матрица переменных решений определяется с помощью функции `lmivar`, которая имеет следующий синтаксис:

$$X = \text{lmivar}(\text{type}, \text{structure}).$$

Эта команда позволяет нам определить несколько форм матриц решений, таких как симметричные матрицы, прямоугольные матрицы или матрицы другого типа. В зависимости от выбранного типа матрицы структура содержит различную информацию. Таким образом, сначала мы определяем тип, а затем определяем структуру, которая зависит от типа.

- Если `type = 1`, это означает, что матрица  $X$  является квадратной и симметричной. Элемент структуры  $(i, 1)$  определяет размер  $i$ -го блока, а структурный элемент  $(i, 2)$  задает тип  $i$ -го блока (1 для полного, 0 для скаляра и -1 для нулевого блока).

- Если `type = 2`, матрица  $X$  является прямоугольной размером  $m \times n$ , как указано в структуре `= [m, n]`.

- Если `тип = 3`, матрица  $X$  имеет другой тип.

*Шаг 3: Определите ЛМН один за другим.* Это делается с помощью команды `lmiterm`.

Синтаксис команды

$$\text{lmiterm}(\text{termID}, A, B, \text{flag}).$$

`lmiterm` принимает 3 или 4 аргумента. Первый аргумент `termID` является вектором  $1 \times 4$ .

Первый элемент этого вектора указывает, какой ЛМН определен. Вторая и третья записи в этом векторе определяют положение определяемого термина. И четвертая запись указывает, какая переменная решения ЛМН задействована. Четвёртый параметр может быть равен 0 или  $X$  в зависимости от того, является ли этот терм постоянным, типа  $AXB$  или  $AX^T B$ . Второй и третий аргументы функции `lmiterm` являются левым и правым мультипликатором матрицы решений. Если флаг установлен в 's', он

позволяет указать с помощью одной команды, что данный термин и его симметричное значение отображаются в ЛМН.

*Пример 3.* Рассмотрим следующий набор ЛМН:

$$\begin{bmatrix} CX^T C^T + B^T YA & XF \\ F^T X & Y \end{bmatrix}, \quad (15)$$

$$DXD^T > 0. \quad (16)$$

Здесь мы имеем две переменные решения X и Y и два ЛМН. Пусть Y – полная симметричная матрица размерности 5, X имеет 5 блоков, а размеры различных блочных матриц равны 5, 4, 3, 1 и 2.

Сначала мы определяем переменные X и Y, используя `lmivar`-функцию следующим образом:

```
structureX=[5,1;4,1;3,1;1,0;2,1];
X=lmivar(1,structureX);

structureY=[5,1];
Y=lmivar(1,structureY);
```

Затем мы определяем ЛМН с использованием функции `lmiterm` следующим образом:

```
% LMI (15)
lmiterm([1 1 1 X],C,C');
lmiterm([1 1 1 Y],B',A);
 YA
lmiterm([1 1 2 X],1,F);
lmiterm([1 2 1 X],F',1);
 X
lmiterm([1 2 2 Y],1,1);

% LMI (16)
lmiterm([-1, 1, 1, X],D,D');
```

Наконец, мы создаем объект ЛМН, используя следующую команду.

```
myLMIsystem=getlmis;
```

## 2 Примеры применения ЛМН для решения различных проблем управления

*Пример 4.* Определить устойчивость линейных инвариантных по времени систем

Рассмотрим линейную инвариантную по времени систему

$$\dot{x} = Ax$$

Система стабильна при соблюдении следующих неравенств:

$$P > 0, A^T P + PA < 0$$

Эти два неравенства могут быть объединены в один ЛМН как

$$\begin{bmatrix} A^T P + PA & 0 \\ 0 & -P \end{bmatrix} < 0.$$

Код MATLAB для решения проблемы устойчивости для  $A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$

следующим образом.

### Программа

```
A=[0 1;-2 -3];
```

```
% определим неизвестную матрицу, которая должна
определяться ЛМН setlmis ([])
```

```
P=lmivar(1,[size(A,1) 1]);
```

```
% определяем ЛМН
```

```
lmiterm([1 1 1 P],1,A,'s'); % A' P+PA
```

```
lmiterm([1 1 2 0],1); % 0
```

```
lmiterm([1 2 2 P],[-1,1]); % P>0
```

```
LMISYS = getlmis;
```

```
[tmin,Psol]=feaspl(LMISYS);
```

```
P=dec2mat(LMISYS,Psol,P)
```

После запуска этой программы получаем  $t_{\min} = -2.615451$  и

$$P = \begin{bmatrix} 65.9992 & 12.8946 \\ 12.8946 & 15.1836 \end{bmatrix}$$

*Пример 5.* Задача LQR: решение уравнения Риккати.

Рассмотрим систему, представленную следующими линейными уравнениями состояния непрерывного времени:

$$\dot{x} = Ax + Bu, x(0) = x_0,$$

где  $x \in R^n$  – вектор состояния,  $u \in R^m$  – входной вектор,  $A$  и  $B$  – известные матрицы соответствующих размерностей. Цель состоит в том, чтобы определить управляющий вход  $u$ , который минимизирует следующий индекс производительности:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt,$$

где  $Q \in R^{n \times n}$  – вещественная симметричная положительная полуопределенная матрица, а  $R \in R^{p \times p}$  – вещественная положительно определенная матрица.

Оптимальный управляющий вход, который минимизирует  $J$ , задается формулой

$$u(t) = R^{-1} B^T P x(t) = K x(t), K = R^{-1} B^T P, \quad (\text{П17})$$

где матрица  $P$  получается путем решения следующего уравнения Риккати:

$$A^T P + P A + P B R^{-1} B^T P + Q < 0, P > 0, R > 0.$$

Заметим, что уравнение Риккати, в отличие от уравнений Ляпунова, является нелинейным уравнением в  $P$ . Это связано с тем, что квадратичный член  $P B R^{-1} B^T P$  появляется в неравенстве. Используя дополнение Шура, мы можем представить это неравенство как ЛМН следующим образом:

$$P > 0, Q > 0, R > 0 \text{ и } \begin{bmatrix} A^T P + P A + Q & P B \\ B^T & -R \end{bmatrix} < 0. \quad (\text{П18})$$

Код MATLAB для вычисления коэффициента усиления  $K$  задается следующим образом:

### Программа

```
A=[-2 1;1 -1];
```

```
B=[1;1];
```

```
setlmis([])
```

```
% указываем структуру и размер P,Q и R
```

```
P=lmivar(1,[2,1]);
```

```
Q=lmivar(1,[2,1]);
```

```
R=lmivar(1,[1,1]);
```

```
% определяем ЛМН
```

```
lmiterm([-1 1 1 P],1,1); %P>0
```

```
lmiterm([-2 1 1 Q],1,1); %Q>0
```

```
lmiterm([-3 1 1 R],1,1); %R>0
```

```
%ЛМН (18)
```

```
lmiterm([4 1 1 P],A',1,'s');
```

```
lmiterm([4 1 1 Q],1,1);
```

```
lmiterm([4 1 2 P],1,B);
```

```
lmiterm([4 2 2 R],-1,1);
```

```
LMIs = getlmis;
```

```
[TMIN,XFEAS] = feasp(LMIs);
```

```
Q=dec2mat(LMIs,XFEAS,Q);
```

```
P=dec2mat(LMIs,XFEAS,P);
```

```
R=dec2mat(LMIs,XFEAS,R);
```

```
K=inv(R)*B'*P
```

```
% производим вычисления
```

После решения ЛМН управляющее усиление  $K = [0.2454 \quad 0.2210]$

*Пример 6.* Контроллер  $H_\infty$  с использованием полной обратной связи для системы

$$\dot{x} = Ax + B_1 w + B_2 u,$$

$$z = C_1 x + D_{11} w + D_{12} u.$$

Параметры контроллера можно получить, разрешив следующие ЛМН:

$$Y > 0$$

$$\begin{bmatrix} YA^T + AY + Z^T B_2^T + B_2 Z & B_1 & YC_1^T + Z^T D_{12}^T \\ B_1^T & -\gamma^I & D_{11}^T \\ C_1 Y + D_{12} Z & D_{11} & -\gamma^I \end{bmatrix} < 0. \quad (\text{П19})$$

Усиление контроллера можно вычислить из  $F = ZY^{-1}$ . Для предопределенного  $\gamma = 1$  мы имеем следующий код:

### Программа

```
% известные параметры
```

```
A=[0.8 -.25;1 0];
```

```
B1=[0 ; 0.1];
```

```
B2=[.1; 0.03];
```

```
C1=[0 .1];
```

```
D11=0.05;
```

```
D12=0.1;
```

```
gamma=1;
```

```
setlmis([])
```

```
% указываем структуру и размер Y и Z
```

```
Y=lmivar(1,[2,1]);
```

```
% симметричная матрица 2
```

```
на 2
```

```
Z=lmivar(2,[1,2]);
```

```
% вектор 1 на 2
```

```

lmiterm([-1 1 1 Y],1,1); %Y>0

% JIMH (19)
lmiterm([2 1 1 Y],A,1,'s');
lmiterm([2 1 1 Z],B2,1,'s');
lmiterm([2 1 2 0],B1);
lmiterm([2 1 3 Y],1,C1');
lmiterm([2 1 3 -Z],1,D12');
lmiterm([2 2 2 0],-gama);
lmiterm([2 2 3 0],D11');
lmiterm([2 3 3 0],-gama);

LMIs = getlmis;
options = [1e-5,0,0,0,0];
[TMIN,XFEAS]=feasp(LMIs,options,0);

Y=dec2mat(LMIs,XFEAS,Y);
Z=dec2mat(LMIs,XFEAS,Z);
F=Z*inv(Y)

```

После решения в MATLAB, мы имеем  $F = [-20.3758 \quad -6.1154]$ . Данный результат может быть проверен тем, что  $A + B_2 F \begin{bmatrix} -1.2376 & -0.86115 \\ 0.3887 & -0.1835 \end{bmatrix}$  и его собственные значения  $-0.7105 + 0.2390i$  и  $-0.7105 - 0.2390i$ , что показывает, что система устойчива.

## ЛИТЕРАТУРА

1. C. R. Cutler and B. L. Ramaker. Dynamic matrix control-a computer control algorithm. *Presented at the Meeting of the American Institute of Chemical Engineers, Houston, Texas, 1979.*
2. C. E. Garcia and A. M. Morshedi. Quadratic programming solution to Dynamic Matrix Control (QDMC). *Chemical Engineering Communication*, 46:73–87, 1986.
3. V. Peterka. Predictor-based self-tuning control. *Automatica*, 20:39–50, 1984.
4. D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized predictive control. part 1: The basic algorithm. part 2: Extensions and interpretations. *Automatica*, 23:137–160, 1987.
5. P. J. Gawthrop. Linear predictive pole-placement control: practical issues. *39<sup>th</sup> IEEE Conference on Decision and Control, Sydney, Australia*, 160–165, 2000.
6. P. J. Gawthrop and E. Ronco. Predictive pole-placement control with linear models. *Automatica*, 38:421–432, 2002.
7. G. C. Goodwin, S. Graebe, and M. Salgado. *Control System Design*. Prentice-Hall, Englewood Cliffs, NJ, 2000.
8. V. Gopal and L. T. Biegler. Large scale inequality constrained optimization and control. *IEEE Control System Mag.*, 18:59–68, 1998.
9. G.L. Baker, J.A. Blackburn, *The Pendulum* (Oxford University Press, Oxford, 2005).
10. H.T. Banks, H.T. Tran, *Mathematical and Experimental Modeling of Physical and Biological Processes*. Textbooks in Mathematics (CRC Press, Boca Raton, 2009).
11. Дорф Р., Бишоп Р. *Современные системы управления*. – М.: Лаборатория Базовых Знаний, 2002.
12. Самарский А.А., Гулин А.В. *Численные методы: Учеб. пособие для вузов*. – М.: Наука, 1989.
13. Yan XG, Edwards C (2008) Robust sliding mode observer-based actuator fault detection and isolation for a class of nonlinear systems. *Int J Syst Sci* 39(4):349–359.
14. Wang Z, Shen Y, Zhang X (2014) Actuator fault estimation for a class of nonlinear descriptor systems. *Int J Syst Sci* 45:487–496.
15. Zhang K, Jiang B, Shi P (2009) Fast fault estimation and accommodation for dynamical systems. *IET Control Theory Appl* 3:189–199.

16. Chen W, Chowdhury FN (2010) A synthesized design of sliding-mode and Luenberger observers for early detection of incipient faults. *Int J Adapt Control Signal Process* 24:1021–1035.
17. Alwi H, Edwards C, Tan CP (2011) Fault detection and fault-tolerant control using sliding modes. Springer, Heidelberg.
18. Каханер Д., Моулер К., Нэш С. *Численные методы и программное обеспечение* / Пер. с англ. – М.: Мир, 2002.
19. Zhang J, Swain AK, Nguang SK (2011) Reconstruction of actuator fault for a class of nonlinear systems using sliding mode observer. In: *Proceedings of American control conference*, pp.1370–1375.
20. Chae S, NguangSK(2013)SOSbased robust fuzzy dynamic output feedback control of nonlinear networked control systems. *IEEE Trans Cybern* 44:1204–1213.
21. Saat S, Nguang S.K. (2015) Nonlinear  $H_\infty$  output feedback control with integrator for polynomial discrete time systems. *Int J Robust Nonlinear Control* 25:1051–1065.
22. R. Illner, C.S. Bohun, S. McCollum, T. van Roode, *Mathematical Modelling: A Case Studies Approach*, vol. 27 (American Mathematical Society, Providence, 2005).
23. Острейковский В.А. *Теория систем: Учеб. пособие.* – М.: Высш. школа, 1997.
24. Вавилов А.А., Имаев Д.Х. *Машинные методы расчёта систем управления: Учеб. пособие.* – Л.: Изд-во Ленингр. ун-та, 1981.
25. Gao C, Zhao Q, Duan G (2013) Robust actuator fault diagnosis scheme for satellite attitude control systems. *J Frankl Inst* 350:2560–2580 <http://dx.doi.org/10.1016/j.jfranklin.2013.02.021>
26. Edwards C, Tan CP (2006) Sensor fault tolerant control using sliding-mode observers. *Control Eng Pract* 14(8):897–908
27. L.G. Leal, *Advanced Transport Phenomena* (Cambridge University Press, Cambridge, 2007).
28. M. Levi, *Classical Mechanics with Calculus of Variations and Optimal Control*. Student. Mathematical Library, vol. 69 (American Mathematical Society, Providence, 2014).
29. J.D. Logan, *Applied Mathematics*, 3rd edn. (Wiley-Interscience, Hoboken, 2006).
30. J.D. Logan, *An Introduction to Nonlinear Partial Differential Equations*, 2nd edn. (Wiley-Interscience, Hoboken, 2008).
31. Введение в математическое моделирование: Учеб. пособие / Под ред.



- П.В. Трусова. – М.: Логос, 2010.
32. C.R. MacCluer, *Calculus of Variations* (Pearson Prentice Hall, Upper Saddle River, 2005).
  33. J.E. Marsden, A.J. Tromba, *Vector Calculus*, 5th edn. (W.H. Freeman, New York, 2003).
  34. R.M.M. Mattheij, S.W.Rienstra, J.H.M. tenThije Boonkkamp, *Partial Differential Equations: Modeling, Analysis, Computation* (SIAM, Philadelphia, 2005).
  35. Dang B., Zhilenkov A.A. Methods of Adaptive Signal Processing in the Space-Time Coding System//Proceedings of the 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2019, IET – 2019, pp. 8656904
  36. J.D. Meiss, *Differential Dynamical Systems* (Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2007).
  37. M. Mesterton-Gibbons, *A Primer on the Calculus of Variations and Optimal Control Theory* (American Mathematical Society, Providence, 2009).
  38. Мышкис А.Д. *Элементы теории математических моделей*. – М: Физматлит, 2004.
  39. A.J. Roberts, *Model Emergent Dynamics in Complex Systems* (SIAM, Philadelphia, 2014).
  40. R.A. Serway, *Physics for Scientists and Engineers* (Brooks/Cole, Boston, 2014).
  41. Самарский А.А., Михайлов А.П. *Математическое моделирование: Идеи. Методы. Примеры*. – М.: Физматлит, 2005.
  42. D.R. Smith, *Singular-Perturbation Theory: An Introduction with Applications* (Cambridge University Press, Cambridge, 1985).
  43. Edwards C, Spurgeon SK, Patton RJ (2000) Sliding mode observers for fault detection and isolation. *Automatica* 36:541–553.
  44. Tan CP, Edwards C (2003) Sliding mode observers for robust detection and reconstruction of actuator and sensor faults. *Int J Robust Nonlinear Control* 13:443–463.
  45. Zhilenkov A.A. High productivity numerical computations for gas dynamics modelling based on DFT and approximation//IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2018, IET – 2018, pp. 400 – 403.
  46. Арнольд В.И. *Обыкновенные дифференциальные уравнения*. – М.: Наука, 1984.
  47. Tan CP, Edwards C (2002) Sliding mode observers for detection and

- reconstruction of sensor faults. *Automatica* 38:1815–1821.
48. Yan XG, Edwards C (2007) Sensor fault detection and isolation for nonlinear systems based on a sliding mode observer. *Int J Adapt Control Signal Process* 21:657–673.
  49. Жиленков А.А. Разработка метода решения уравнений теплопроводности с неравномерной дискретизацией для моделирования процессов в реакторах газофазной эпитаксии // Системы управления и информационные технологии – 2017. – Т. 69. – № 3. – С. 11 – 15
  50. Alwi H, Edwards C, Tan CP (2009) Sliding mode estimation schemes for incipient sensor faults. *Automatica* 45:1679–1685.
  51. Yan XG, Edwards C (2007) Nonlinear robust fault reconstruction and estimation using a sliding mode observer. *Automatica* 43:1605–1614.
  52. Yan XG, Edwards C (2008) Robust sliding mode observer-based actuator fault detection and isolation for a class of nonlinear systems. *Int J Syst Sci* 39(4):349–359.
  53. Raoufi R, Marquez HJ, Zinober ASI (2010)  $H_\infty$  sliding mode observer for uncertain nonlinear Lipschitz systems with fault estimation synthesis. *Int J Robust Nonlinear Control* 20:1785–1801.
  54. F. Verhulst, *Methods and Applications of Singular Perturbations* (Springer, New York, 2005).
  55. R.B. White, *Asymptotic Analysis of Differential Equations*, Revised edn. Imperial College Press, London, 2010).
  56. L. Wang. Use of orthonormal basis functions in continuous-time mpc design. *UKACC*, 2000.
  57. L. Wang. Discrete model predictive control design using Laguerre functions. *Proceedings of American Control Conference*, 2001a.
  58. L.Wang. Continuous time model predictive control using orthonormal functions. *International Journal of Control*, 74:1588–1600, 2001b.
  59. L. Wang. Use of exponential data weighting in model predictive control design. *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001c.
  60. Liu M, Cao X, Shi P (2013) Fuzzy-model-based fault-tolerant design for nonlinear stochastic systems against simultaneous sensor and actuator faults. *IEEE Trans Fuzzy Syst* 21:789–799.
  61. Bejarano FJ, Figueroa M, Pacheco J, Rubio J (2012) Robust fault diagnosis of disturbed linear systems via a sliding mode high order differentiator. *Int J Control* 85:648–659.
  62. Lee DJ, Park YJ, Park YS (2012) Robust  $H_\infty$  sliding mode descriptor observer

- for fault and output disturbance estimation of uncertain systems. *IEEE Trans Autom Control* 57:2928–2934.
63. Zhang J, Swain AK, Nguang SK (2012) Detection and isolation of incipient sensor faults for a class of uncertain nonlinear systems. *IET Control Theory Appl* 6:1870–1880.
  64. Оран Э., Борис Дж. *Численное моделирование реагирующих потоков.* – М.: Мир, 2006.
  65. Zhang J, Swain AK, Nguang SK (2013) Robust sensor fault estimation scheme for satellite attitude control systems. *J Frankl Inst* 350:2581–2604.
  66. Floquet T, Edwards C, Spurgeon SK (2007) On sliding mode observers for systems with unknown inputs. *Int J Adapt Control Signal Process* 21:638–656.
  67. Kalsi K, Lian J, Hui S, Zak SH, (2010) Sliding-mode observers for systems with unknown inputs: a high-gain approach. *Automatica* 46:347–353.
  68. Zhu F (2012) State estimation and unknown input reconstruction via both reduced-order and high-order sliding mode observers. *J Process Control* 22:296–302.
  69. Tan CP, Edwards C (2001) An LMI approach for design sliding mode observers. *Int J Control* 74:1559–1568.
  70. Chen W, Saif M (2007) Observer-based strategies for actuator fault detection, isolation and estimation for certain class of uncertain nonlinear systems. *IET Control Theory Appl* 1:1672–1680.
  71. Zhang K, Jiang B, Shi P (2009) Fast fault estimation and accommodation for dynamical systems. *IET Control Theory Appl* 3:189–199.
  72. L. Wang. Discrete model predictive control design using Laguerre functions: Numerical sensitivity analysis. *Proceedings of American Control Conference*, 2003.
  73. L. Wang. Discrete model predictive control design using Laguerre functions. *Journal of Process Control*, 14:131–142, 2004.
  74. L. Wang, C. Chessari, and E. Karpriel. Inferential control of product quality attributes: application to food cooking extrusion process. *Journal of Process Control*, 11:621–636, 2001.
  75. L. Wang and W. R. Cluett. *From Plant Data to Process Control: Ideas for Process Identification and PID Design.* Taylor and Francis, London, 2000.
  76. L. Wang, P. Gawthrop, C. Chessari, and T. Podsiadley. Continuous time system identification of food extruder: experiment design and data analysis. *Proceedings of IFAC Symposium on System Identification, Holland*, 2003.
  77. L. Wang, P. Gawthrop, C. Chessari, and T. Podsiadley. Indirect approach to continuous time system identification of food extruder. *Journal of Process*

*Control*, 14:603–615, 2004.

78. Жиленков А.А. Гибридное решение уравнений Навье-Стокса в пространствах аналитических функций с применением билинейных форм и функции Грина // Системы управления и информационные технологии – 2018. – Т. 71. – № 1. – С. 4 – 7.
79. Жиленков А.А. Численное моделирование процессов переноса в реакторах газофазной эпитаксии с аппроксимацией в пространствах аналитических функций // Двойные технологии – 2018. – № 2(83). – С. 66–68.
80. Yan XG, Edwards C (2007) Nonlinear robust fault reconstruction and estimation using a sliding mode observer. *Automatica* 43:1605–1614.
81. Gao ZW, Ding SX (2008) Fault reconstruction for Lipschitz nonlinear descriptor systems via linear matrix inequality approach. *Circuits Syst Signal Process* 27:295–308.
82. Жиленков А.А., Яковлева Д.Н. Синтез модели синхронной машины, оперирующей естественными фазными сигналами обмоток статора // Системы управления и информационные технологии – 2017. – Т. 68. – № 2. – С. 49–52.
83. Raoufi R, Marquez HJ, Zinober ASI (2010)  $H_\infty$  sliding mode observer for uncertain nonlinear Lipschitz systems with fault estimation synthesis. *Int J Robust Nonlinear Control* 20:1785–1801.
84. Jiang B, Staroswiecki M, Cocquempot V (2004) Fault estimation in nonlinear uncertain systems using robust/sliding-mode observers. *IEE Proc Control Theory Appl* 151(1):29–37.
85. Zhilenkov A.A. A Solution to the Inverse Problem of a Uniform Thermal Field on the Susceptor Surface in a Vapor-Phase Epitaxy Reactor with Induction Heating // Proceedings of the 2017 International Conference on “Physics, Mechanics of New Materials and Their Applications”, IET – 2018, pp. 60.
86. Edgar Thomas F, Ivn Castillo, Fernnndez Benito R (2012) Robust model-based fault detection and isolation for nonlinear processes using sliding modes. *Int J Robust Nonlinear Control* 22:89–104.
87. L. Wang and P. C. Young. An improved structure for model predictive control using non-minimal state space realisation. *Journal of Process Control*, 16:355–371, 2006.
88. L. Wang, P. C. Young, P.J. Gawthrop, and C.J. Taylor. Non-minimal state space model-based continuous-time model predictive control with constraints. *To appear in International Journal of Control*, 2009.
89. C.C. Lee, Fuzzy logic in control systems: Fuzzy logic controller - part I and

II, IEEE Trans. SMC 20 (2) (1990) 404-435.

90. T. Yamakawa, A fuzzy inference engine in non-linear analog mode and its application to a fuzzy logic control, IEEE Trans. Neural Networks 4 (3) (1993) 496-522.
91. H. Watanabe, W.D. Detloff, K.E. Yount, A VLSI fuzzy logic controller with reconfigurable, cascaded architecture, IEEE J. Sol. State circuits. 25 (2) (1990) 376-382.
92. N.M. Botros, M. Abdul-Aziz, Hardware implementation of an artificial neural network, Proc. IEEE Int. Conf. on Neural Networks, 1993.
93. Zhilenkov A.A., Epifantsev I.R. Problems of a trajectory planning in autonomous navigation systems based on technical vision and AI//IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, EIConRus 2018, IET – 2018, pp. 1032 – 1035.
94. Dudarenko N.A., Kapitonov A.A., Shavetov S.V., Zhilenkov A.A., Pyrkin A.A., Gerasimov D.N., Vedyakov A.A., Borisov O.I., Gromov V.S., Zimenko K.A., Kolyubin S.A., Krasnov A.Y. International summer school of Control Systems and Robotics. Learning book, IET – 2018.
95. Zhilenkov A.A., Kotlyarevskaya M.V. Synthesis of model of hardware realization of izhikevich model of biological neuron on the basis of FPGA//Proceedings of the 2018 IEEE Russia Section Young Researchers in Electrical and Electronic Engineering Conference, EIConRus 2018, IET – 2018, pp. 1040 – 1043.
96. Zhilenkov A.A., Chernyi S.G., Nyrkov A.P., Sokolov S.S. Optimization Problem of Thermal Field on Surface of Revolving Susceptor in Vapor-Phase Epitaxy Reactor//IOP Conference Series: Earth and Environmental Science, IET – 2017, Vol. 87, No. 8, pp. 082060.

Жиленков Антон Александрович

**Моделирование систем и комплексов:**  
**Дискретные системы прогностического управления в теории,**  
**задачах и примерах в MATLAB**

**Учебное пособие**

В авторской редакции  
Редакционно-издательский отдел Университета ИТМО  
Зав. РИО Н.Ф. Гусарова  
**Подписано к печати**  
Заказ №  
Тираж  
Отпечатано на ризографе

**Редакционно-издательский отдел**  
**Университета ИТМО**  
197101, Санкт-Петербург, Кронверкский пр., 49