

УДК 51-74

Составитель: П.А. Безмен

Рецензент

Кандидат технических наук, доцент кафедры механики, мехатроники
и робототехники

Е.Н. Политов

Цифровые интеллектуальные системы в робототехнике:
методические указания по выполнению практических работ для
студентов направления «Мехатроника и робототехника» / Юго-Зап.
гос. ун-т; сост.: П.А. Безмен; Курск, 2024. 61 с.

Содержат сведения по анализу и синтезу систем управления,
описанию систем в пространстве состояний, оптимальной линейной
фильтрации. Приведены методика выполнения работ, примеры,
варианты заданий.

Методические указания соответствуют требованиям
Федерального государственного образовательного стандарта.

Методические указания предназначены для студентов
направлений технического профиля для всех форм обучения.

Текст печатается в авторской редакции

Подписано в печать *26.09*. Формат 60x84 1/16.

Усл. печ. л. 3,55. Уч.-изд. л. 3,21. Тираж 100 экз. Заказ *100* Бесплатно.

Юго-Западный государственный университет.

305040, Курск, ул. 50 лет Октября, 94.

Содержание

Практическая работа № 1 Анализ системы управления	4
Практическая работа № 2 Синтез оптимального управления с полной обратной связью	16
Практическая работа № 3 Описание систем в пространстве состояний	30
Практическая работа № 4 Фильтр Калмана.....	50

Практическая работа № 1 Анализ системы управления

Цель работы

Целью работы является исследование с помощью пакета MATLAB устойчивости, управляемости, наблюдаемости и минимальности системы управления заданной структурной схемой, и построить ее модель пониженного порядка.

Краткие сведения из теории

Анализ устойчивости

Понятие устойчивости является одним из основных в теории управления. Устойчивой называют систему, которая, будучи выведена из состояния равновесия, стремится вновь вернуться в это состояние. Для неустойчивых систем характерна обратная тенденция. Примером неустойчивой системы может быть карандаш, стоящий на острие. Обычно исследование устойчивости систем сводится к анализу устойчивости соответствующих дифференциальных уравнений. У неустойчивых уравнений решение неограниченно возрастает со временем. В теории автоматического управления существуют различные методы анализа устойчивости. Для линейных систем разработаны критерии устойчивости, которые можно разделить на корневые, алгебраические и частотные.

Корневой критерий устойчивости

Для того, чтобы линейная динамическая система была устойчивой, необходимо и достаточно, чтобы все корни ее характеристического уравнения лежали в левой комплексной полуплоскости.

Например, математическая модель системы, описываемой дифференциальным уравнением

$$\ddot{x} + 2\dot{x} + 5x = 0,$$

устойчива, так как корни характеристического уравнения $p^2+2p+5=0$ имеют отрицательные вещественные части $p_{1,2} = -1 \pm 2i$, т.е. лежат слева от оси ординат.

Алгебраический критерий устойчивости

Алгебраические критерии устойчивости позволяют судить об устойчивости на основе анализа коэффициентов характеристического полинома. Наибольшую известность получил критерий устойчивости Гурвица. Из него, в частности, следует, что все коэффициенты устойчивого дифференциального уравнения должны быть положительны.

Для уравнений второго порядка это необходимое и достаточное условие.

Для уравнений третьего порядка

$$a_3\ddot{x} + a_2\dot{x} + a_1x + a_0x = 0,$$

помимо положительности коэффициентов, должно выполняться дополнительное условие, а именно, произведение средних коэффициентов должно быть больше произведения крайних:

$$a_2a_1 > a_0a_3.$$

Например, уравнение

$$\ddot{x} + 2\dot{x} + 0,5x + 3x = 0$$

неустойчиво, так как $2 \cdot 0,5 < 1 \cdot 3$.

Частотный критерий устойчивости

Частотные критерии носят графический характер. Они опираются на анализ графиков частотных характеристик - АЧХ, ФЧХ, АФХ (последняя известна также как диаграмма Найквиста). Часть из них позволяет делать заключение об устойчивости замкнутой системы управления по частотным характеристикам разомкнутой системы.

Приведем в качестве примера критерий Найквиста. Обозначим передаточную функцию разомкнутой системы $Q(p)$ и охватим ее единичной отрицательной обратной связью (рис. 1).

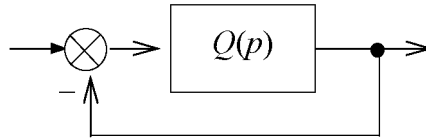


Рис. 1. Система с обратной связью

Пусть известно, что разомкнутая система устойчива. Тогда для устойчивости замкнутой системы необходимо и достаточно, чтобы годограф Найквиста разомкнутой системы не охватывал точку с координатами $(-1, 0j)$ на комплексной плоскости. В тулбоксе CONTROL пакета MATLAB диаграмма Найквиста системы sys строится командой `nyquist(sys)`, указанная точка помечена на ней красным крестиком.

В работе исследуется система управления, заданная структурной схемой, поэтому для анализа устойчивости надо сначала найти ее передаточную функцию. Структурная схема исследуемой системы приведена на рис. 2. В ее состав входят три апериодических звена, а также суммирующие и вычитающие звенья.

В схеме можно указать три пути от входа до выхода с передаточными функциями

$$Q_1(p) = \frac{k_1}{T_1 p + 1}, \quad Q_2(p) = \frac{-k_2}{T_2 p + 1} \cdot \frac{k_3}{T_3 p + 1}, \quad Q_3(p) = \frac{-k_1}{T_1 p + 1} \cdot \frac{a k_3}{T_3 p + 1}.$$

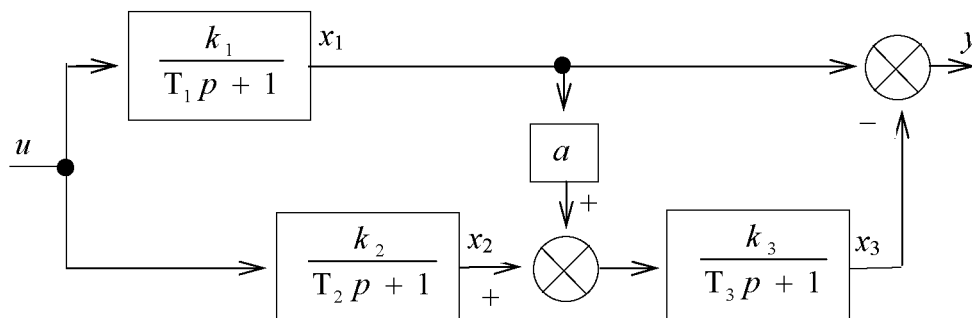


Рис. 2. Структура исследуемой системы

Общая передаточная функция получается, как их сумма

$$Q(p) = Q_1(p) + Q_2(p) + Q_3(p) = \frac{B(p)}{A(p)}. \quad (1)$$

Здесь $A(p)$ и $B(p)$ – некоторые полиномы от p (свои для каждого варианта). Характеристический полином системы равен знаменателю этой передаточной функции

$$A(p) = (T_1p + 1)(T_2p + 1)(T_3p + 1) \quad (2)$$

На основе приведенных выше критериев устойчивости можно заключить, что если коэффициенты T_1 , T_2 , T_3 положительны, то система будет устойчивой. Анализ устойчивости замкнутой системы можно осуществить с помощью критерия Найквиста.

Анализ управляемости и наблюдаемости

Понятия управляемости и наблюдаемости широко используются в современной теории автоматического управления. Линейная система называется управляемой, если с помощью входного сигнала ее можно перевести из начала координат в любую точку пространства состояний. Система называется наблюдаемой, если по измерениям входного и выходного сигналов можно однозначно определить её начальное состояние.

Анализ управляемости и наблюдаемости выполняется с помощью матриц управляемости и наблюдаемости или с помощью грамианов управляемости и наблюдаемости. Те и другие строятся по описанию в пространстве состояний

$$\dot{X} = AX + bu, \quad y = cX. \quad (3)$$

Чтобы получить такое описание, воспользуемся структурной схемой, приведенной на рис. 2. Выпишем операторные уравнения для каждого блока схемы

$$x_1 = \frac{k_1}{T_1 p + 1} u, \quad x_2 = \frac{k_2}{T_2 p + 1} u, \quad x_3 = \frac{k_3}{T_3 p + 1} (ax_1 + x_2)$$

и преобразуем их к виду:

$$\begin{aligned} T_1 p x_1 &= -x_1 + k_1 u, \\ T_2 p x_2 &= -x_2 + k_2 u, \\ T_3 p x_3 &= ak_3 x_1 + k_3 x_2 - x_3. \end{aligned}$$

Выполняя обратное преобразование Лапласа, получаем систему линейных дифференциальных уравнений первого порядка:

$$\begin{aligned} T_1 \dot{x}_1 &= -x_1 + k_1 u, \\ T_2 \dot{x}_2 &= -x_2 + k_2 u, \\ T_3 \dot{x}_3 &= ak_3 x_1 + k_3 x_2 - x_3. \end{aligned}$$

Кроме того, имеем алгебраическое уравнение для выходного сигнала $y = x_1 - x_3$.

Переписывая эти уравнения в матричной форме (3), получаем следующие выражения для матриц A , b , c :

$$A = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix},$$

$$b = \begin{bmatrix} b_1 \\ b_2 \\ 0 \end{bmatrix},$$

$$c = [c_1 \quad 0 \quad c_3],$$

где ненулевые элементы матриц имеют вид:

$$a_{11} = -1/T_1, a_{22} = -1/T_2, a_{31} = ak_3/T_3, a_{32} = k_3/T_3, a_{33} = -1/T_3, b_1 = k_1/T_1, b_2 = k_2/T_2, c_1 = 1, c_3 = -1.$$

Теперь можно перейти к анализу управляемости и наблюдаемости. Сформируем на основе матриц A , b , c две вспомогательные матрицы

$$R = [b, Ab, \dots, A^{n-1}b], \quad D = \begin{bmatrix} c \\ cA \\ \dots \\ cA^{n-1} \end{bmatrix}.$$

Матрицы R и D называются соответственно матрицей управляемости и матрицей наблюдаемости системы. В пакете MATLAB их можно построить с помощью команд **ctrb** и **obsv**.

Критерий управляемости. Для того чтобы система (3) была управляемой, необходимо и достаточно, чтобы матрица управляемости имела полный ранг $\text{rank}R = n$.

Критерий наблюдаемости. Для того чтобы система (3) была наблюдаемой, необходимо и достаточно, чтобы матрица наблюдаемости имела полный ранг $\text{rank}D = n$.

В случае систем с одним входом и одним выходом матрицы R и D квадратные, поэтому для проверки управляемости и наблюдаемости достаточно вычислить определители матриц R и D . Если они не равны нулю, то матрицы имеют полный ранг.

Другой способ проверки управляемости и наблюдаемости опирается на вычисление грамианов управляемости и наблюдаемости. Так называются симметричные квадратные матрицы W_c и W_o , определяемые равенствами

$$W_c = \int_0^{\infty} e^{At} b b^T e^{A^T t} dt, \quad W_o = \int_0^{\infty} e^{A^T t} c^T c e^{At} dt$$

В пакете MATLAB их можно найти с помощью команд типа $\text{gram}(\text{sys}, 'c')$, $\text{gram}(\text{sys}, 'o')$. Необходимые и достаточные условия управляемости и наблюдаемости имеют вид $\det W_c \neq 0$, $\det W_o \neq 0$.

Анализ минимальности моделей

Одной и той же передаточной функции $Q(p)$ можно сопоставить целый класс эквивалентных реализаций в пространстве состояний, характеризуемых различными тройками матриц (A, b, c) разных размеров. Реализация называется минимальной, если размер ее матрицы A наименьший среди всех эквивалентных реализаций. Поиск такой реализации имеет практический смысл, так как ее моделирование на ЭВМ требует меньших вычислительных затрат.

Для анализа минимальности реализации нужно проверить ее управляемость и наблюдаемость.

Критерий минимальности. Для того чтобы реализация (3) была минимальной, необходимо и достаточно, чтобы она была управляемой и наблюдаемой одновременно.

Таким образом, анализ минимальности конкретной реализации сводится к проверке пары критериев $\text{rank}R = n$, $\text{rank}D = n$.

Если хотя бы один из рангов меньше n , то реализация неминимальна. Размерность эквивалентной минимальной реализации n_0 определяется по формуле $n_0 = \text{rank}(RD)$.

Анализ минимальности с помощью грамианов управляемости и наблюдаемости проводится аналогично. Критерием минимальности служит выполнение условия $\det(W_c W_o) \neq 0$, эквивалентного паре условий $\det W_c \neq 0$, $\det W_o \neq 0$.

Если исходное описание реализации оказалось неминимальным, то возникает задача перехода к минимальной реализации. Чтобы решить ее, сначала перейдем от описания в пространстве состояний (3) к передаточной функции

$$Q(p) = c(pE - A)^{-1}b = \frac{B(p)}{A(p)}. \quad (4)$$

Далее нужно выделить общий множитель в числителе и знаменателе передаточной функции $Q(p)$ и сократить на него. Эта процедура известна, как сокращение совпадающих нулей и полюсов системы.

Отметим ряд соотношений между элементами матриц A , b , c и коэффициентами передаточной функции, вытекающих из формулы (4). Знаменатель передаточной функции совпадает с характеристическим полиномом матрицы A

$$A(p) = \det(pE - A) = p^n + a_{n-1}p^{n-1} + \dots + a_1p + a_0$$

Его корни $\lambda_1, \dots, \lambda_n$ равны собственным числам матрицы A , а коэффициенты a_0 и a_{n-1} с точностью до знака равны ее определителю и следу

$$\begin{aligned} a_0 &= \det(-A) = (-1)^n \lambda_1 \lambda_2 \dots \lambda_n, \\ -a_{n-1} &= \text{tr}A = a_{11} + a_{22} + \dots + a_{nn} = \lambda_1 + \lambda_2 + \dots + \lambda_n. \end{aligned} \quad (5)$$

Старший и младший коэффициенты числителя передаточной функции

$$B(p) = b_{n-1}p^{n-1} + \dots + b_1p + b_0$$

удовлетворяют соотношениям

$$b_{n-1} = cb, \quad b_0 / a_0 = -c A^{-1} b. \quad (6)$$

Величина $k_0 = b_0 / a_0$ называется статическим коэффициентом усиления. Она равна установившемуся значению переходной функции системы. В пакете MATLAB для вычисления статического коэффициента усиления имеется команда **dcgain** (от direct current gain - коэффициент усиления по постоянному току). В ее основу положена формула $k_0 = Q(0)$. Приведенные соотношения удобно использовать для контроля вычислений.

Получение минимальной реализации в пакете MATLAB осуществляется командой **minreal**, для вычисления нулей и полюсов можно использовать функции **zero**, **pole**, **pzmap**, **zpk**. Аргументом во всех случаях служит исследуемая система **sys**, предварительно сформированная командами **ss** или **tf**.

Методика выполнения работы

1. Нарисовать схему рис. 2 для своего варианта и найти ее передаточную функцию по формуле (1).

2. Проверить устойчивость системы, используя алгебраический и корневой критерии.

3. Получить для своего варианта схемы описание в пространстве состояний вида (3) и выписать матрицы **A**, **B**, **C**.

4. Найти матрицы управляемости и наблюдаемости системы, определить их ранги и сделать вывод об управляемости, наблюдаемости и минимальности системы.

5. Найти передаточную функцию по формуле (4), и сравнить ее с полученной в п.1. Проверить выполнение соотношений (5), (6). Тремя способами (по структурной схеме, передаточной функции и описанию в пространстве состояний) найти статический коэффициент усиления системы.

6. Определить порядок минимальной реализации и найти ее передаточную функцию, выполнив сокращение нулей и полюсов. Найти реакцию минимальной реализации на единичный скачок и построить ее график.

7. Привести программы на языке MATLAB для выполнения пунктов 2 - 6 и краткое описание назначения и синтаксиса команд **rank**, **ss**, **tf**, **zpk**, **zero**, **pole**, **pzmap**, **ctrb**, **obsv**, **minreal**, **dcgain**.

Порядок выполнения работы

1. В диалоговом режиме пакета MATLAB ввести матрицы **A**, **b**, **c** и сформировать **ss**-описание системы $sys = ss(A, b, c, 0)$. Используя команды **pole**, **eig**, **pzmap**, найти полюсы системы и получить график их расположения на комплексной плоскости. Сделать вывод об устойчивости. Построить диаграмму Найквиста системы и сделать

заклучение об устойчивости системы, охваченной обратной связью (рис. 1).

2. Найти матрицы управляемости и наблюдаемости, вычислить их определители и ранги. Сделать вывод об управляемости, наблюдаемости и минимальности. Найти грамианы управляемости и наблюдаемости.

3. С помощью команд **tf** и **zpk** перейти к передаточной функции. Двумя способами получить минимальную реализацию (сокращая нули и полюса, и с помощью команды **minreal**). Сравнить переходные функции исходной и минимальной реализаций, а также их статический коэффициент усиления.

4. Набрать схемы исходной и минимальной реализаций в SIMULINK и сравнить их реакции на одинаковые входные сигналы.

Контрольные вопросы

1. Найти передаточные функции систем, заданных в пространстве состояний тройкой матриц:

$$\begin{aligned}
 \text{a) } A &= \begin{bmatrix} -1 & 0 \\ 1 & -2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad c = [1 \ 1], & \text{б) } A &= \begin{bmatrix} -3 & 1 \\ 1 & -2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad c = [0 \ 1], \\
 \text{в) } A &= \begin{bmatrix} -4 & 1 \\ 2 & -3 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad c = [1 \ 2], & \text{г) } A &= \begin{bmatrix} -4 & 2 \\ 1 & -3 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad c = [0 \ 1].
 \end{aligned}$$

2. Найти ранги матриц управляемости и наблюдаемости для систем из п.1.

3. Нарисовать структурные схемы систем, матрицы которых приведены ниже.

№	1	2	3	4	5	6	7
A	0 1	0 1	0 1	0 1	1 0	1 0	1 1
	0 0	1 0	-1 0	1 1	0 2	0 1	1 1
b^T	0 1	0 1	0 1	0 1	1 1	0 1	0 1
c	0 1	0 1	0 1	0 1	1 1	0 1	1 1

Определить, какие из них являются:

- а) устойчивыми; б) управляемыми; в) наблюдаемыми; г) минимальными.

4. Система управления задана структурной схемой, показанной на рис. 3.

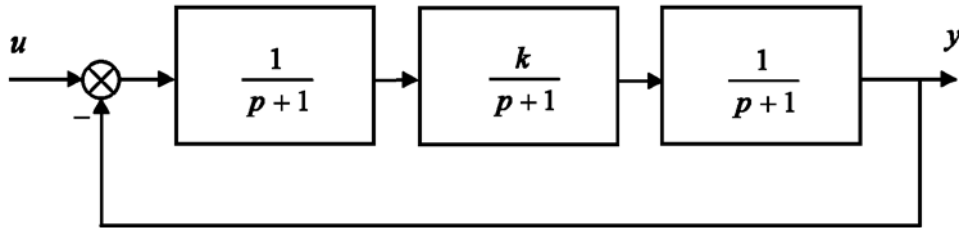


Рис. 3. Система управления третьего порядка

Требуется:

- а) найти статический коэффициент усиления схемы;
 б) найти передаточную функцию схемы и проанализировать ее устойчивость;
 в) найти описание схемы в пространстве состояний; построить матрицы управляемости и наблюдаемости, сделать вывод о минимальности;
 г) выяснить, при каких значениях коэффициента k схема будет устойчивой, управляемой, наблюдаемой.

Варианты заданий

Таблица 1

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
k_1	1	3	2	1	4	3	3	3	2	3	2	2	4	2	2	2	3	4	3	3	1
T_1	1	4	2	3	1	3	3	2	2	4	1	1	2	3	5	2	2	4	1	1	4
k_2	5	3	6	1	6	4	1	2	6	4	3	7	5	4	6	5	1	3	2	3	4
T_2	1	4	2	3	1	3	3	2	2	4	2	2	2	4	5	2	2	4	1	1	4
k_3	5	2	2	4	3	3	6	6	2	5	6	4	7	2	2	2	4	4	5	5	4
T_3	4	2	1	2	4	5	3	3	4	3	5	5	2	2	5	2	4	2	2	4	3
a	-2	-1	-1	-2	-1	-1	-1	-2	-1	-2	-2	-1	-2	-1	-1	-1	-2	-2	-2	-1	-1

Литература

1. Мироновский, Л.А. Моделирование линейных систем: Методические указания к выполнению лабораторных работ / С.-Пб.: ГУАП, 2007.
 2. Эльсгольц Л.Э. Обыкновенные дифференциальные уравнения: Учебник для вузов / С.-Пб.: Изд-во «Лань», 2002.
 3. Есипов А.А., Сазонов Л.И., Юдович В.И. Практикум по обыкновенным дифференциальным уравнениям. - М.: Вузовская книга, 2001.
 4. Романко В. К. Разностные уравнения: Учебное пособие / -М.: БИНОМ. Лаборатория знаний, 2006.
 5. Мироновский Л.А. Моделирование разностных уравнений: Учебное пособие / С.-Пб.: ГУАП, 2004.
- Мироновский Л.А., Петрова К.Ю. Введение в MATLAB: Учебное пособие / С.-Пб.: ГУАП, 2006.

Практическая работа № 2

Синтез оптимального управления с полной обратной связью

Цель работы

Целью работы является ознакомление с методикой построения линейных оптимальных систем управления с полной обратной связью методом динамического программирования Беллмана.

Постановка задачи

Математическая модель системы, описывающая поведение объекта управления, имеет вид

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k), \\ y(k) &= C(k)x(k) + D(k)u(k) \\ k &= 0, 1, \dots, N-1, \end{aligned} \quad (1)$$

с начальным условием

$$x(0) = x_0, \quad (2)$$

задан функционал качества управления –

$$J = \sum_{k=0}^{N-1} x^T(k)Q(k)x(k) + u^T(k)R(k)u(k) \quad (3)$$

где $Q(k)$ - неотрицательно определенная симметрическая матрица размера $(n \times n)$, $R(k)$ - положительно определенная симметрическая матрица $(q \times q)$.

Требуется найти управление $u^*(k, x)$ с полной обратной связью, минимизирующее функционал (3).

Краткие сведения из теории

Пусть поведение модели объекта управления описывается обыкновенным дифференциальным уравнением

$$\dot{x}(t) = f(t, x(t), u(t)), \quad (4)$$

где x - вектор состояния системы,

$x \in \mathbf{R}^n, \mathbf{R}^n$ - n -мерное евклидово пространство; u - вектор управления, и $u \in U \subset \mathbf{R}^n$,

U - некоторое заданное множество допустимых значений управления,

$t \in T = [t_0, t_1]$ - интервал времени функционирования системы, моменты начала процесса t_0 и окончания процесса t_1 заданы, $f(t, x, u)$:

$$T \times \mathbf{R}^n \times U \rightarrow \mathbf{R}^n.$$

Задан функционал качества управления:

$$J = \int_{t_0}^{t_1} f^0(t, x(t), u(t))dt + F(x(t_1)), \quad (5)$$

где $f^0(t, x, u), F(x)$ - заданные непрерывно дифференцируемые функции. Предполагается, что при управлении используется информация о текущем времени и векторе состояния x .

Применяемое в каждый момент времени $t \in T$ управление имеет вид управления с полной связью по всем переменным вектора состояния (рис. 1).

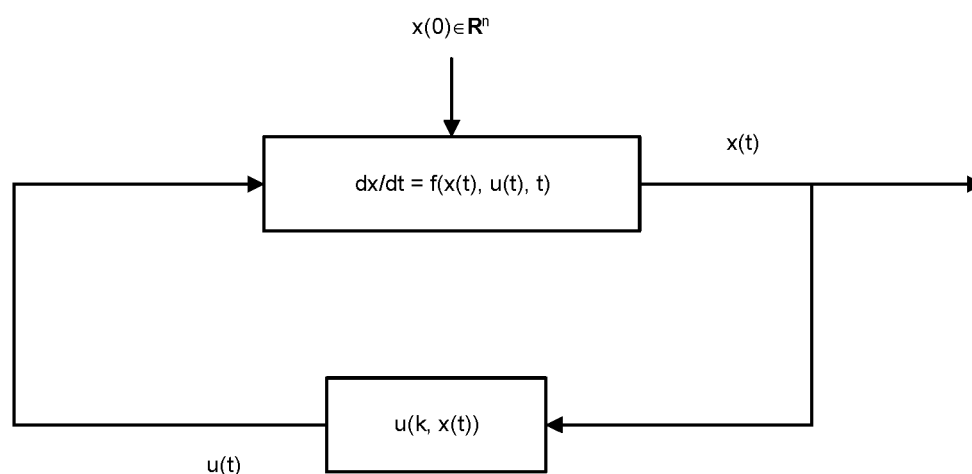


Рис. 1. Схема управления с полной обратной связью по вектору состояния

Требуется найти такую функцию

$$u^*(t, x) \in U_n$$

что:

$$J = \min_{u \in U} J, \quad (6)$$

$$\forall x_0 \in \mathbb{R}^n.$$

Функция

$$u^*(t, x) \in U_n$$

называется оптимальным управлением с полной обратной связью. Для любого начального состояния x_0 из множества \mathbb{R}^n она порождает соответствующую оптимальную пару, т.е. оптимальную траекторию x^* и оптимальное программное управление u^* .

Достаточным условием минимума функционала (5) является уравнение Беллмана для непрерывных детерминированных систем.

Если существуют функция

$$\phi(t, x) \in C^{1,1},$$

удовлетворяющая уравнению Беллмана с граничным условием:

$$\max_{u \in U} \left\{ \frac{\partial \phi(t, x)}{\partial t} + \sum_{i=1}^n \frac{\partial \phi(t, x)}{\partial x_i} f_i(t, x, u) - f^0(t, x, u) \right\} = 0, \quad \forall(t, x), \quad (7)$$

$$\phi(t_1, x) = -F(x),$$

$$\forall x \in \mathbb{R}^n,$$

и управление

$$u^*(t, x) \in U_n,$$

удовлетворяющее условию

$$u^*(t, x) = \arg \max_{u \in U} \left\{ \sum_{i=1}^n \frac{\partial \phi(t, x)}{\partial x_i} f_i(t, x, u) - f^0(t, x, u) \right\},$$

то $u^*(t, x)$ является оптимальным управлением с полной обратной связью. При этом минимальное значение функционала (5)

$$\min_u J = -\phi(t_0, x_0), \quad (8)$$

$$\forall x_0 \in \mathbb{R}^n.$$

Пусть система, описывающая поведение модели объекта управления, имеет вид

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t), \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \quad (9)$$

Пусть функционал качества управления квадратичный:

$$J = \frac{1}{2} \int_{t_0}^{t_1} [x^T(t)S(t)x(t) + u^T(t)Q(t)u(t)] dt + \frac{1}{2} x^T(t_1)\Lambda x(t_1) \quad (10)$$

где $S(t)$, Λ – неотрицательно определенные симметрические матрицы размера $(n \times n)$, а $Q(t)$ – положительно определенная симметрическая матрица $(q \times q)$.

Используем известные правила и обозначения:

1. $\frac{\partial(Ax)}{\partial x} = A^T$;
2. $\frac{\partial(x^T Ax)}{\partial x} = Ax + A^T x$;
3. $(AB)^T = B^T A^T$;
4. $x^T Ax \equiv 0 \Leftrightarrow A + A^T = 0$;
5. $\text{tr } A = \sum_i a_{ii}$.

Уравнение Беллмана для данной задачи имеет вид:

$$\max_{u \in R^q} \left\{ \frac{\partial \varphi(t, x)}{\partial t} + \left(\frac{\partial \varphi(t, x)}{\partial t} \right)^T [A(t)x + B(t)u] - \frac{1}{2} [x^T(t)S(t)x(t) + u^T(t)Q(t)u(t)] \right\} = 0,$$

$$\varphi(t_1, x) = -\frac{1}{2} x^T \Lambda x \quad (11)$$

Отсюда

$$u^*(t, x) = \arg \max_{u \in U} \left\{ \left(\frac{\partial \varphi(t, x)}{\partial t} \right)^T B(t)u - \frac{1}{2} u^T(t)Q(t)u(t) \right\}.$$

Найдем максимум в последнем выражении по управлению с использованием необходимых условий экстремума и правила 1-3. Дифференцируя выражение в квадратных скобках по u и приравнявая результат нулю, получаем структуру оптимального управления:

$$u^*(t, x) = Q^{-1}(t)B^T(t) \frac{\partial \varphi(t, x)}{\partial t}. \quad (12)$$

Решение уравнения (11) ищется в виде:

$$\varphi(t, x) = \frac{1}{2} x^T K_2 x, \quad (13)$$

где $K_2(t)$ - неизвестная симметрическая матрица ($n \times n$).

Подставляя (13) в уравнение (11), приравнявая нулю квадратичные формы, получаем:

$$\dot{K}_2(t) = -A^T(t)K_2(t) - K_2(t)A(t) - K_2(t)B(t)Q^{-1}(t)B^T(t)K_2(t) + S(t), \quad (14)$$

$$K_2(t_1) = -\Lambda$$

Решая уравнение Риккати (14), можно получить явный вид оптимального управления (12) с полной обратной связью

$$u^*(t, x) = Q^{-1}(t)B^T(t)K_2(t). \quad (15)$$

Минимальная величина функционала вычисляется по формуле:

$$\min J = -\varphi(t_0, x_0) = -\frac{1}{2} x_0^T K_2(t_0) x_0.$$

Рассмотрим дискретный случай

$$\begin{aligned} x(k+1) &= A(k) x(k) + B(k) u(k), \\ y(k) &= C(k) x(k) + D(k) u(k) \\ k &= 0, 1, \dots, N-1, \end{aligned} \quad (16)$$

с начальным условием

$$x(0) = x_0, \quad (17)$$

и функционалом качества

$$J = \sum_{k=0}^{N-1} (x^T(k) Q(k) x(k) + u^T(k) R(k) u(k)) + (x^T(N) \Lambda x(N)) \quad (18)$$

где $Q(k)$, Λ - неотрицательно определенные симметрические матрицы размера $(n \times n)$, $R(k)$ - положительно определенная симметрическая матрица $(q \times q)$.

Требуется найти управление $u^*(k, x)$ с полной обратной связью, минимизирующее функционал.

Уравнение Беллмана принимает вид:

$$B(k, x) = \min_u [x^T Q(k) x + u^T R(k) u + B(k+1, A(k)x + B(k)u)] \quad (19)$$

Функция Беллмана $B(k, x)$ ищется в форме:

$$B(k, x) = x^T P(k) x, \quad (20)$$

где $P(k)$ - неизвестная неотрицательно определенная симметрическая матрица размера $(n \times n)$.

Подставляя (20) в (19) получаем, что в задаче (16)-(18) оптимальное управление определяется соотношением

$$u^*(k, x) = -\overline{K(k)} x, \quad k = 0, N-1, \quad (21)$$

где $K(k)$ - матрица коэффициентов усиления регулятора размера $(q \times n)$

$$K(k) = [R(k) + B^T P(k+1) B(k)]^{-1} B^T(k) P(k+1) A(k), \quad k = \overline{0, N-1}, \quad (22)$$

а матрица $P(k)$ размера $(n \times n)$ удовлетворяет уравнению

$$P(k) = Q(k) + K^T(k) R(k) K(k) + [A(k) - B(k) K(k)]^T P(k+1) [A(k) - B(k) K(k)],$$

$$k = \overline{N-1, 0}, \quad (23)$$

$$P(N) = \Lambda.$$

Минимальная величина функционала определяется по формуле

$$\min J = x_0^T P(0) x_0 \quad (24)$$

Структурная схема регулятора системы управления с обратной связью по всем переменным состояния изображена на рис 2.

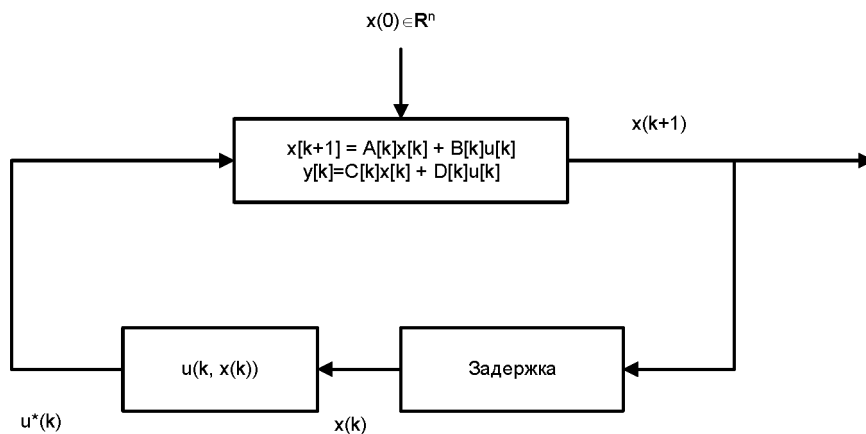


Рис. 2. Структурная схема регулятора системы управления

Для каждого начального состояния x_0 оптимальный линейный регулятор порождает оптимальное программное управление $u^*(x, k)$ и оптимальную траекторию $x^*(k)$.

Методика выполнения работы

Для синтеза оптимальных регуляторов линейных стационарных систем в Control System Toolbox имеются функции решений уравнений Беллмана (таблица 1).

Таблица 1 - Функции Control System Toolbox

Синтаксис	Описание
[K P e] = lqr(A, B, Q, S)	Синтез непрерывного регулятора
[K P e] = lqr(A, B, Q, S, N)	Синтез непрерывного регулятора
[K P e] = dlqr(A, B, Q, R)	Синтез дискретного регулятора
[K P e] = dlqr(A, B, Q, R, N)	Синтез дискретного регулятора
[K P e] = lqrd(A, B, Q, R, Ts)	Синтез дискретного регулятора
[K P e] = lqrd(A, B, Q, R, N, Ts)	Синтез дискретного регулятора

Функция lqr вычисляет матрицу коэффициентов регулирования K со среднеквадратичным функционалом качества без терминального члена:

$$J = \int_{t_0}^{t_1} [x^T Q x + u^T S u + 2x^T N u] dt,$$

при этом вычисляются матрица P , являющаяся решением уравнения Риккати, и собственные значения e матрицы $(A - BK)$.

Функция dlqr вычисляет матрицу коэффициентов регулирования по всем переменным состояния K для дискретной системы со среднеквадратичным функционалом качества без терминального члена:

$$J = \sum_{k=0}^{N-1} (x^T(k) Q x(k) + u^T(k) R u(k) + x^T(k) N u(k)),$$

при этом вычисляются матрица P , являющаяся решением уравнения Риккати, и собственные значения e матрицы $(A - BK)$.

Функция `lqrd` предназначена для синтеза оптимального дискретного регулятора непрерывной системы со среднеквадратичным функционалом качества:

$$J = \int_{t_0}^{t_1} [x^T Q x + u^T S u + 2x^T N u] dt.$$

В качестве параметра в функцию передается шаг дискретизации T_s , возвращаются значения матрицы K дискретного управления, матрица P , являющаяся решением уравнения Риккати и собственные значения e матрицы системы управления, полученные в результате дискретизации.

При использовании всех команд синтеза оптимального линейного регулятора по всем переменным состояния на исходные данные накладываются следующие ограничения:

- система, определяемая матрицами (A, B) должна быть стабилизируема;
- должны выполняться неравенства $S > 0, Q - NR^{-1}N^T > 0$,
- пара матриц $(Q - NR^{-1}N^T, A - BR^{-1}B^t)$ не должна иметь наблюдаемые моды с собственными значениями на действительной оси.

Для выполнения практической работы необходимо выполнить следующие действия:

1. Изучить теоретические сведения.
2. Запустить систему MATLAB.
3. Создать ss-объект, в соответствии с заданным вариантом.
4. Определить матрицы $P(k), K(k)$.
5. Построить оптимальный регулятор $u^*(k, x) = -K(k)x$.
6. Определить значение функционала на оптимальном управлении.
7. Построить графики динамики системы при ненулевых начальных условиях.
8. Ответить на контрольные вопросы.
9. Оформить отчет и защитить работу.

Методический пример

Ниже приведен пример script-файла, моделирующего систему управления и синтез оптимального регулятора.

```
% Параметры системы
A=[1 0; -2 1];
B=[1 0; 1 0]';

% Параметров критерия качества управления
Q=[1/2 0; 0 1/2];

R=[1/2 0; 0 1/2];

% Время регулирования
T=10;
% Величина шага
SS=0.5;
% Количество шагов
N=T/SS

% Вычисление параметров регулятора
[k p e]= dlqr(A, B, Q, R)

x = zeros(2, N);
u= zeros(2, N-1);

% Начальные условия
x(1,1)=2;
x(2,1)=1;

% Построение графиков динамики системы
for i=1:N-1,
    u(:, i)= - k*x(:, i);,
    x(:, i+1)=A*x(:, i)+B*u(:, i);
end

x1= x(1,:);
x2= x(2,:);
t = 0:SS:T-SS;

subplot(4, 1, 1);
plot(t, x1, 'b');
subplot(4, 1, 2);
plot(t, x2, 'g');

subplot(4, 1, 3);
plot(SS:SS:T-SS, u(1, :), 'y');

subplot(4, 1, 4);
plot(SS:SS:T-SS, u(2, :), 'r');
```

Результаты вычисления следующие: значения параметров оптимального регулятора:

$$\begin{aligned}
 k &= \\
 & \quad 0.8229 \quad -0.1771 \\
 & \quad 0.8229 \quad -0.1771 \\
 p &= \\
 & \quad 3.7343 \quad -1.4114 \\
 & \quad -1.4114 \quad 1.1614 \\
 e &= \\
 & \quad 0.1771 + 0.1771i \\
 & \quad 0.1771 - 0.1771i
 \end{aligned}$$

Зависимости величин $x_1(t)$, $x_2(t)$, $u_1(t)$, $u_2(t)$ представлены на рис.

3.

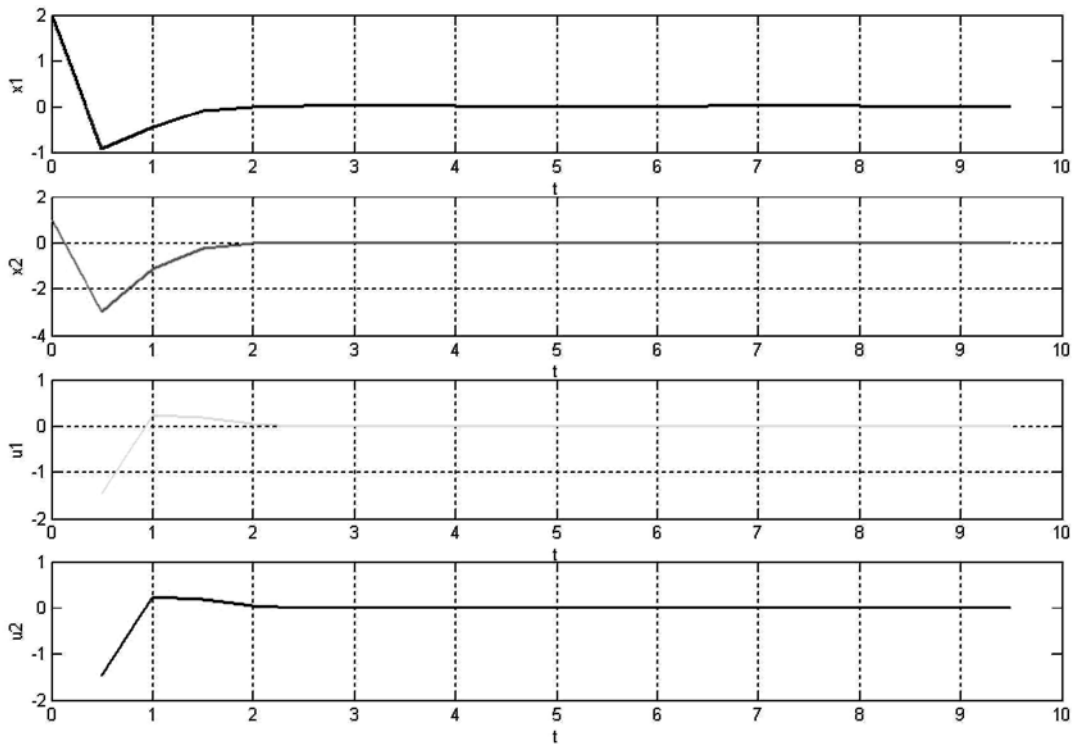


Рис. 3. Динамика состояний и управлений: зависимости $x_1(t)$, $x_2(t)$, $u_1(t)$, $u_2(t)$

Отчет о работе

Отчет оформляется в соответствии с требованиями, предъявляемыми к оформлению работ в вузе, и должен содержать:

1. Титульный лист.
2. Наименование и цель работы.
3. Постановка задачи в соответствии с вариантом.
4. Порядок и результаты определения вычисления матриц P и K .
5. Уравнение Беллмана для решаемой задачи.
6. Значение минимальной величины функционала качества управления.
7. Результаты моделирования динамики системы в числовом и графическом виде.
8. Анализ результатов и выводы.

Контрольные вопросы

1. Сформулировать основную задачу оптимального управления.
2. Дать определение критерия качества. Привести примеры критериев и дать их физическую интерпретацию.
3. Вывести необходимое условие оптимальности.
4. Показать, что для применения метода необходимо, чтобы система была стабилизируема.
5. Разработать в среде MATLAB интерфейс для интерактивного построения регулятора с полной обратной связью.
6. Выяснить влияние задержки при синтезе дискретного регулятора непрерывной системы.

Варианты заданий

Таблица 2 - Варианты заданий

Модель системы	Функционал качества управления
1. $x_1(k+1) = 3x_1(k) - u_1(k) + 3u_2(k)$ $x_2(k+1) = x_1(k) - x_2(k) + u_1(k)$ $x_1(0) = 1, x_2(0) = 3$	1. $J = \frac{1}{2} \sum_{k=0}^1 3u_1^2(k) + 4u_2^2(k) + x_2^2(k)$
	2. $J = \frac{1}{2} \sum_{k=0}^1 2u_1^2(k) + x_2^1(k) + 4x_2^1(k)$
	3. $J = \frac{1}{2} \sum_{k=0}^1 2u_1^2(k) + 4u_2^2(k) + 8x_2^1(k) + 12x_2^1(k)$
2. $x_1(k+1) = x_1(k) + 2x_2(k) - 2u_1(k) - 3u_2(k)$ $x_2(k+1) = 2x_1(k) + x_2(k) - 4u_1(k) - u_2(k)$ $x_1(0) = 1, x_2(0) = 2$	1. $J = \frac{1}{2} \sum_{k=0}^1 7u_1^2(k) + u_2^2(k) + 5x_2^2(k)$
	2. $J = \frac{1}{2} \sum_{k=0}^1 2u_1^2(k) + 9x_2^1(k) + x_2^1(k)$
	3. $J = \frac{1}{2} \sum_{k=0}^1 u_1^2(k) + 4u_2^2(k) + 6x_2^1(k) + 8x_2^1(k)$
3. $x_1(k+1) = 2x_2(k) + x_1(k) - u_1(k)$ $x_2(k+1) = x_1(k) + 4x_2(k) - 2u_2(k)$ $x_1(0) = 1, x_2(0) = 1$	1. $J = \frac{1}{2} \sum_{k=0}^1 2u_1^2(k) + 4u_2^2(k) + x_2^2(k)$
	2. $J = \frac{1}{2} \sum_{k=0}^1 2u_1^2(k) + 3x_2^1(k) + 4x_2^1(k)$
	3. $J = \frac{1}{2} \sum_{k=0}^1 u_1^2(k) + 2u_2^2(k) + 3x_2^1(k) + 7x_2^1(k)$
4. $x_1(k+1) = x_1(k) - x_2(k) - u(k)$ $x_2(k+1) = 2x_1(k) + x_2(k) - 2u(k)$ $x_1(0) = 1, x_2(0) = 1$	1. $J = \frac{1}{2} \sum_{k=0}^1 4u^2(k) + 8x_2^2(k)$
	2. $J = \frac{1}{2} \sum_{k=0}^1 u^2(k) + x_2^1(k) + 4x_2^1(k)$
	3. $J = \frac{1}{2} \sum_{k=0}^1 3u^2(k) + 3x_2^1(k) + x_2^1(k)$

Литература

1. Никульчев Е.В. Практикум по теории управления в среде MATLAB: Учебное пособие. – М.: МГАПИ, 2002.
2. Андриевский Б.Р., Фрадков А.Л. Избранные главы теории автоматического управления с примерами в системе MATLAB. - СПб.: Наука, 1999.
3. Медведев В.С., Потемкин В.Г. Control System Toolbox. MATLAB 5 для студентов /Под общ. ред. Потемкина В.Г. - М.: Диалог-МИФИ, 1999.
4. Семенов В.В., Пантелеев А.В., Бортаковский А.С. Математическая теория управления в примерах и задачах. - М.: МАИ, 1997.
5. Сю Д., Мейер А. Современная теория автоматического управления и ее применение. - М.: Машиностроение, 1972.
6. Дьяконов В., Круглов В. MATLAB. Анализ, идентификация и моделирование систем. Специальный справочник. - СПб.: Питер, 2002
7. Лазарев Ю.Ф. MATLAB 5.x. Библиотека студента. - К.: Издательская группа BHV, 2000.
8. Малышев С. А. Метод корневого годографа: Методического указания по выполнению лабораторных работ по теории управления. - М.: МИП, 1992.
9. Методы классической и современной теории автоматического управления: Учебник в 3-х т. / Под общ. ред. Н.Д. Егупова - М.: Изд-во МГТУ им. Н.Э.Баумана, 2000.
10. Мороз А.И. Методического указания по выполнению лабораторных работ по теории управления. - М.: МИП, 1989.
11. Using the Control System Toolbox with MATLAB 6: Computation. Visualization. Programming - The MathWorks, Inc., 2000.

Практическая работа № 3 Описание систем в пространстве состояний

Цель работы

Целью работы является ознакомление с описанием и исследованием динамических систем управления в пространстве состояний.

Постановка задачи

Даны математические модели трех систем и структурная схема, представляющая собой соединение этих систем. Необходимо:

- получить модель результирующей системы в пространстве состояний,
- исследовать наблюдаемость и управляемость трех подсистем в отдельности и их соединения в соответствии со схемой.

Краткие сведения из теории

Многомерные системы, в отличие от одномерных имеют несколько входов и несколько выходов.

Для описания таких систем используются три набора параметров (три вектора), см. рис.1:

1. вектор входных воздействий (управлений);
2. вектор переменных состояний;
3. вектор выходных параметров;

и два преобразования:

1. преобразование «входы-состояния»;
2. преобразование «состояния-выходы».

Широкое распространение, обусловленное разработанным математическим аппаратом, получили линейные модели многомерных систем в пространстве состояний, которые имеют вид:

$$\begin{aligned}\dot{x}(t) &= A(t)x(t) + B(t)u(t); \\ y(t) &= C(t)x(t);\end{aligned}\tag{1}$$

Первое соотношение называется уравнением состояния системы, второе - уравнением выхода системы. Здесь

$$x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$$

- вектор переменных состояний;

$$u = (u_1, u_2, \dots, u_r)^T \in U \subseteq \mathbb{R}^r$$

- вектор управлений;

$$y = (y_1, y_2, \dots, y_m)^T \in \mathbb{R}^m$$

- вектор измеряемых параметров; t - время;

$A(t)$, $B(t)$, $C(t)$ - матрицы размерности $(n \times n)$, $(n \times r)$, $(m \times n)$ соответственно.

Предполагается, что известны начальные состояния $x(t_0) = x_0$, где t_0 - начальный момент времени.

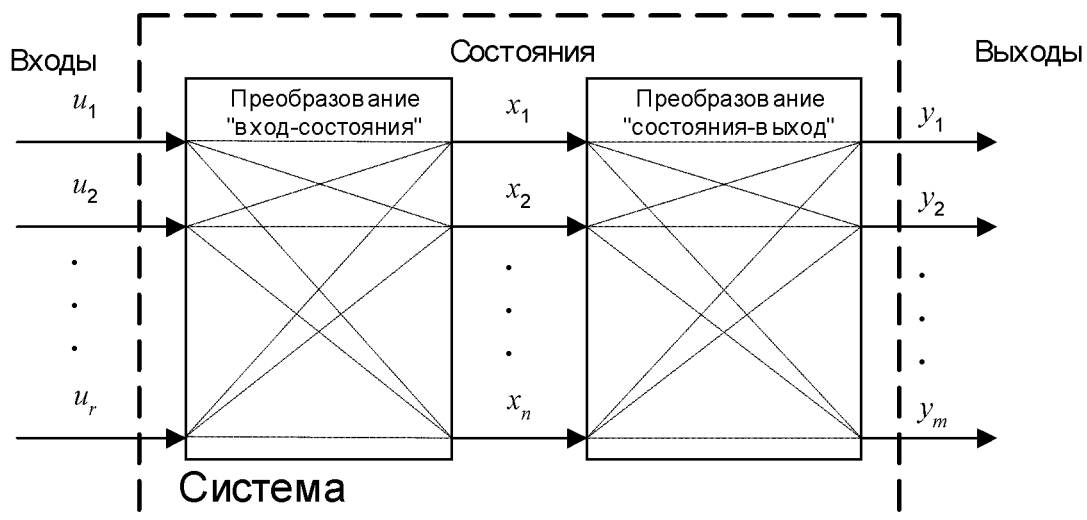


Рис. 1. Многомерные системы

Если матрицы $A(t)$, $B(t)$, $C(t)$ не зависят от времени t , то система называется стационарной. Далее предполагается, что системы стационарны.

Рассмотрим задачи соединения двух подсистем в систему. При соединении возможны три варианта (рис. 2): параллельное (а), последовательное (б) и в обратной связи (в). Предполагается, что обе системы описываются в пространстве состояний соотношениями:

$$\begin{aligned} \dot{x}^1 &= A_1 x^1 + B_1 u^1; & y^1 &= C x^1; \\ \dot{x}^2 &= A_2 x^2 + B_2 u^2; & y^2 &= C x^2; \end{aligned}$$

где: x^1, u^1, y^1 - векторы состояний, управлений, выходов первой системы, x^2, u^2, y^2 - второй. Необходимо по известным матрицам $A_1, B_1, C_1, A_2, B_2, C_2$ получить матрицы A, B, C (рис. 2-з).

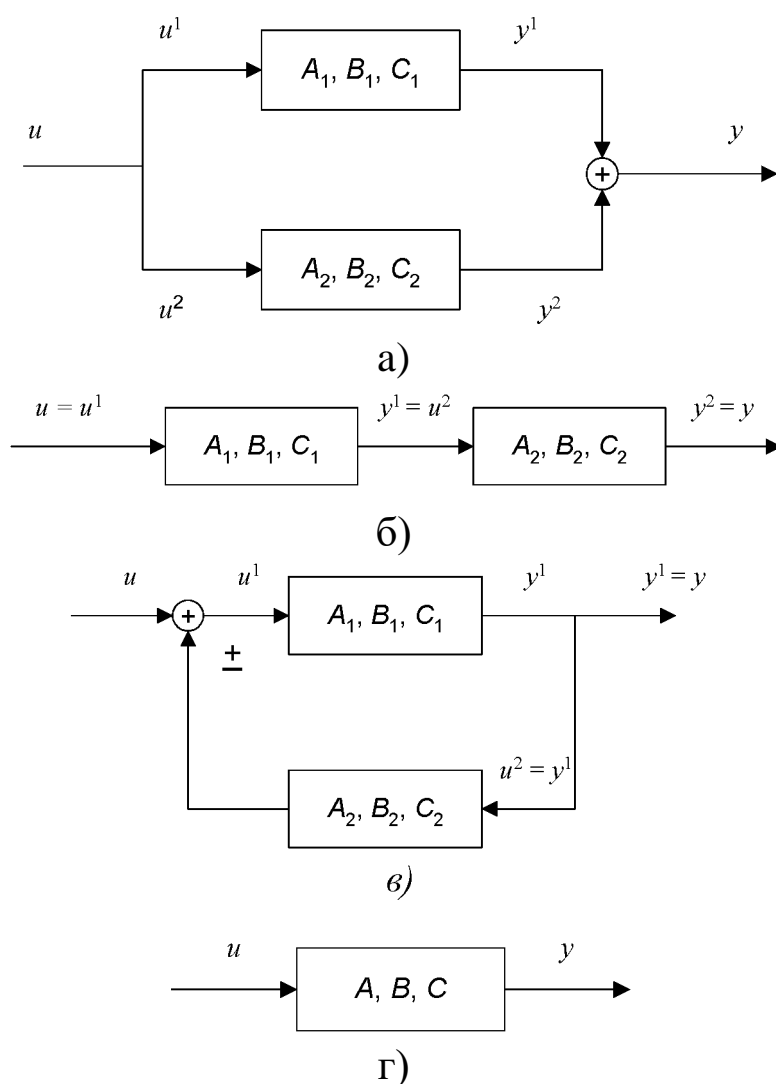


Рис. 2. Соединение двух систем

1. Параллельное соединение.

Запишем уравнения системы, с учетом особенностей соединения, указанных на рис. 2-а:

$$\begin{aligned}\dot{x}^1 &= A_1 x^1 + B_1 u; \\ \dot{x}^2 &= A_2 x^2 + B_2 u; \\ y &= C_1 x^1 + C_2 x^2;\end{aligned}$$

отсюда

$$\begin{aligned}\frac{d}{dt} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} &= \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} u; \\ y &= (C_1 \quad C_2) \begin{pmatrix} x^1 \\ x^2 \end{pmatrix}.\end{aligned}$$

Окончательно матрицы соединения имеют вид:

$$A = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}; \quad B = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}; \quad C = (C_1 \quad C_2).$$

2. Последовательное соединение:

$$\begin{aligned}\dot{x}^1 &= A_1 x^1 + B_1 u; \\ \dot{x}^2 &= A_2 x^2 + B_2 C_1 x^1; \\ y &= C_2 x^2;\end{aligned}$$

в матричном виде:

$$\begin{aligned}\frac{d}{dt} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} &= \begin{pmatrix} A_1 & 0 \\ B_2 C_1 & A_2 \end{pmatrix} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} + \begin{pmatrix} B_1 \\ 0 \end{pmatrix} u; \\ y &= (0 \quad C_2) \begin{pmatrix} x^1 \\ x^2 \end{pmatrix};\end{aligned}$$

окончательно, имеем

$$A = \begin{pmatrix} A_1 & 0 \\ B_2 C_1 & A_2 \end{pmatrix}; \quad B = \begin{pmatrix} B_1 \\ 0 \end{pmatrix}; \quad C = (0 \quad C_2).$$

3. Обратная связь:

$$\dot{x}^1 = A_1 x^1 + B_1 u \pm B_1 C_2 x^2;$$

$$\dot{x}^2 = A_2 x^2 + B_2 C_1 x^1;$$

$$y = C_1 x^1;$$

в матричном виде:

$$\frac{d}{dt} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} A_1 & \pm B_1 C_2 \\ B_2 C_1 & A_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} B_1 \\ 0 \end{pmatrix} u;$$

$$y = (C_1 \quad 0) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

Следовательно,

$$A = \begin{pmatrix} A_1 & \pm B_1 C_2 \\ B_2 C_1 & A_2 \end{pmatrix}; \quad B = \begin{pmatrix} B_1 \\ 0 \end{pmatrix}; \quad C = (C_1 \quad 0).$$

Для линейных систем легко показать справедливость следующего результата, называемого принципом суперпозиции: эффект, вызываемый суммой нескольких воздействий, равен сумме нескольких воздействий, равен сумме эффектов от нескольких воздействий в отдельности. Закон изменения вектора состояний линейной системы представляется в виде суммы свободного и вынужденного колебания

$$x(t) = x_c(t) + x_e(t).$$

Свободное движение $x_c(t)$ происходит при отсутствии внешнего воздействия в ненулевых начальных условиях. Оно определяется решением однородной системы уравнений, соответствующей исходному уравнению состояний:

$$\dot{x}(t) = A(t)x(t)$$

с начальными условиями $x(t_0) = x_0$.

Вынужденное движение $x_e(t)$ - это реакция системы на внешнее воздействие $u(t)$ при нулевых начальных условиях. Оно определяется решением неоднородного уравнения при нулевых начальных условиях.

Для многомерных нестационарных систем, описываемых соотношениями, поведение векторов состояния и выхода определяется по формулам:

$$x(t) = \Phi(t, t_0)x(t_0) + \int_{t_0}^t \Phi(t, \tau)B(\tau)u(\tau)d\tau \quad (2)$$

$$y(t) = C(t)\Phi(t, t_0)x(t_0) + \int_{t_0}^t C(t)\Phi(t, \tau)B(\tau)u(\tau)d\tau \quad (3)$$

где $\Phi(t, \tau)$ - переходная матрица, или матрица Коши, являющаяся решением уравнения

$$\frac{\partial \Phi(t, \tau)}{\partial t} = A(t)\Phi(t, \tau), \quad (4)$$

с начальным условием $\Phi(\tau, \tau) = E$.

Первые слагаемые в (2), (3) описывают свободное движение, а вторые - вынужденное.

Для многомерных стационарных систем, описываемых уравнениями (1), законы изменения вектора состояния и вектора выхода находятся по формулам

$$x(t) = \Phi(t)x(0) + \int_{t_0}^t \Phi(t - \tau)Bu(\tau)d\tau$$

$$y(t) = C\Phi(t)x(0) + \int_{t_0}^t C\Phi(t - \tau)Bu(\tau)d\tau$$

где

$$\Phi(t - \tau)$$

- переходная матрица стационарной системы, зависящая от разности $t - \tau$.

В данном случае решение уравнения (4) имеет вид

$$\Phi(t, \tau) = \Phi(t - \tau) = \exp[A(t - \tau)].$$

Одними из важнейших задач теории управления является исследование управляемости и наблюдаемости динамических систем. Приведем соответствующие определения и критерии для стационарных линейных систем, полученные Калманом.

Система называется вполне управляемой, если выбором управляющего воздействия $u(t)$ на интервале времени $[t_0, t_1]$ можно перевести систему из любого начального состояния $x(t_0)$ в произвольное заранее заданное конечное состояние $x(t_1)$.

Система называется вполне наблюдаемой, если по реакции $y(t_1)$ на выходе системы на интервале времени $[t_0, t_1]$ при заданном управляющем воздействии $u(t)$ можно определить начальное состояние $x(t_0)$.

Критерий управляемости линейных систем. Для того чтобы система была вполне управляемой, необходимо и достаточно, чтобы ранг матрицы управляемости

$$M_U = (B \mid AB \mid A^2B \mid \dots \mid A^{n-1}B)$$

равнялся размерности вектора состояния:

$$\text{rank } M_U = n.$$

Критерий наблюдаемости линейных систем. Для того чтобы система была вполне наблюдаемой, необходимо и достаточно, чтобы ранг матрицы наблюдаемости

$$M_Y = (C^T \mid A^T C^T \mid (A^T)^2 C^T \mid \dots \mid (A^T)^{n-1} C^T)$$

равнялся размерности вектора состояния:

$$\text{rank } M_Y = n.$$

Знак $Z = (X \mid Y)$ означает присоединение матриц, т.е. для получения i -ой строки матрицы Z берется сначала i -ая строка матрицы X , затем следуют элементы i -ой строки матрицы Y . Предполагается, что количество строк у матриц одинаково.

Напомним, что под рангом матрицы подразумевается наивысший из порядков отличных от нуля миноров этой матрицы. Ранг матрицы равен наибольшему числу линейно независимых строк.

Методика выполнения работы

В Control System Toolbox имеется тип данных, определяющих динамическую систему в пространстве состояний. Синтаксис команды, создающий непрерывную LTI (Linear Time Invariant)-систему в виде ss-объекта с одним входом и одним выходом

$$SS(A, B, C, D)$$

В эту функцию в качестве параметров передаются матрицы уравнений состояний и выходов вида

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t); \\ y(t) &= Cx(t) + Du(t);\end{aligned}$$

В связи с тем, что рассматривается модель вида (1), то матрица динамики D будет нулевой.

Для выполнения работы могут применяться команды, приведенные в таблице 1.

Таблица 1 – Некоторые команды Control System Toolbox

Синтаксис команды	Описание
<code>ctrb(LTI-объект)</code> <code>ctrb(A, B)</code>	Формирование матрицы управляемости
<code>obsv(<LTI-объект>)</code> <code>obsv(A, C)</code>	Формирование матрицы наблюдаемости
<code>parallel(<LTI1>,<LTI2>)</code>	Параллельное соединение
<code>series(<LTI1>,<LTI2>)</code>	Последовательное соединение
<code>feedback(<LTI1>,<LTI2>)</code>	Соединение обратной связью
<code>append(<LTI1>, ..., <LTIN>)</code>	Объединение систем
<code>connect(<sys>,<Con>,<in>,<out>)</code>	Установление связей в соединении

Для получения результатов вычисления матриц, результирующей системы, по структурной схеме, воспользуемся последними двумя командами.

Функция *append* создает объект *sys*, представляющий собой объединение всех подсистем. При этом первый входной сигнал первой системы становится входом номер 1, второй входной сигнал первой системы - номер 2, и т.д. далее идут входы второй системы, и т.д.; аналогично определяются и выходы.

В функции *connect* - параметр *<Con>* определяет матрицу связей по структурной схеме. Матрица формируется по следующему правилу: каждая строка представляет собой один вход системы *sys*, первый элемент - номер входа (в соответствии с порядком в команде *append*), затем идут номера выходов, которые суммируются и подаются на рассматриваемый вход. Параметры *<in>*, *<out>* - строки из номеров входов и выходов соединения, являющиеся внешними. Например, для последовательного соединения двух систем (рис. 2-б):

```
sys1= ss(A1, B1, C1, D1)
sys2= ss(A2, B2, C2, D2)
sys=append (sys1, sys2)
sysc=connect(sys, [2 1], [1], [2])
```

В этом случае на вход второй системы (общий вход номер 2), поступает выход первой (общий выход номер 1); вход первой системы (номер один) и выход второй системы (номер два) являются внешними.

Последовательность выполнения работы следующая:

1. Ознакомиться с основными элементами теории.
2. Привести все системы в варианте в форму (1).
3. Запустить систему MATLAB.
4. Создать три ss-объекта, в соответствии с заданным вариантом.
5. Определить управляемость и наблюдаемость каждой системы.
6. В соответствии со структурной схемой получить матрицы А, В, С соединения.
7. Определить управляемость и наблюдаемость соединения.
8. Оформить отчет.

Методический пример

Даны три линейные стационарные системы:

$$\begin{aligned}
 1. \quad & \begin{cases} \dot{x}^1 = \begin{pmatrix} 7 & 3 \\ 2 & 1 \end{pmatrix} x^1 + \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} u^1 \\ y^1 = \begin{pmatrix} 3 & -2 \\ 2 & 1 \end{pmatrix} x^1 \end{cases} ; \\
 2. \quad & \begin{cases} \dot{x}^2 = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix} x^2 + \begin{pmatrix} 1 & 5 \\ 2 & 1 \end{pmatrix} u^2 \\ y^2 = \begin{pmatrix} 4 & 3 \end{pmatrix} x^2 \end{cases} ; \\
 3. \quad & \begin{cases} \dot{x}^3 - 3\dot{x}^3 - 2x^3 = 4u \\ y^3 = x^3 \end{cases} ;
 \end{aligned}$$

и имеется структурная схема соединения систем:

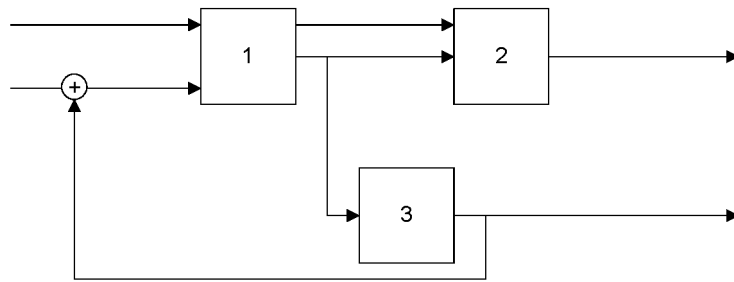


Рис. 3. Вариант задания

1. Приведем систему 3 к виду (1), для этого введем переменные

$$\begin{aligned} x_1^3 &= x^3 \\ x_2^3 &= \dot{x}_1^3 = \dot{x}^3; \end{aligned}$$

и, подставляя их в исходные уравнения, получим

$$\begin{cases} \dot{x}_1^3 = x_2^3 \\ \dot{x}_2^3 - 3x_2^3 - 2x_1^3 = 4u^3 \\ y^3 = x_1^3 \end{cases}; \quad \begin{cases} \dot{x}_1^3 = x_2^3 \\ \dot{x}_2^3 = 2x_1^3 + 3x_2^3 + 4u^3 \\ y^3 = x_1^3 \end{cases}; \quad \begin{cases} \dot{x}^3 = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix} x^3 + \begin{pmatrix} 0 \\ 4 \end{pmatrix} u^3 \\ y^3 = (1 \ 0) x^3 \end{cases}.$$

2. Создадим матрицы первой системы

```
>> A1=[7 3;2 1]
A1 =
     7     3
     2     1
>> B1=[1 0; 0 2]
B1 =
     1     0
     0     2
>> C1=[3 -2; 2 1]
C1 =
     3    -2
     2     1
```

Создавая, аналогично, матрицы двух других систем создадим ss-объекты:


```
>> s1=ss(A1, B1, C1,0)
```

```
a =
      x1 x2
x1  7  3
x2  2  1
```

```
b =
      u1 u2
x1  1  0
x2  0  2
```

```
c =
      x1 x2
y1  3 -2
y2  2  1
```

```
d =
      u1 u2
y1  0  0
y2  0  0
```

Continuous-time model.

```
>> s2=ss(A2, B2, C2,0)
```

```
a =
      x1 x2
x1  1  2
x2  3  2
```

```
b =
      u1 u2
x1  1  5
x2  2  1
```

```
c =
      x1 x2
y1  4  3
```

```
d =
      u1 u2
y1  0  0
```

Continuous-time model.

```
>> s3=ss(A3, B3, C3,0)
```

```
a =
      x1 x2
x1  0  1
x2  2  3
```

```
b =
      u1
x1  0
x2  4
```

```
c =
      x1 x2
y1  1  0
```

```
d =
      u1
y1  0
```

Continuous-time model.

3. Исследуем наблюдаемость и управляемость каждой системы, для чего построим соответствующие матрицы и посчитаем их ранги

```
>> rank(ctrb(A1,B1))

ans =

     2

>> rank(observ(A1,C1))

ans =

     2

>> rank(ctrb(A2,B2))

ans =

     2

>> rank(observ(A2,C2))

ans =

     2

>> rank(ctrb(A3,B3))

ans =

     2

>> rank(observ(A3,C3))

ans =

     2
```

Видно, что во всех случаях ранги матриц управляемости и наблюдаемости совпадают с размерностями пространства состояний.

4. Получим систему, определяемую соединением.

Для корректного использования функции `connect` введем дополнительную систему, передаточная функция которой равна 1 (рис. 4).

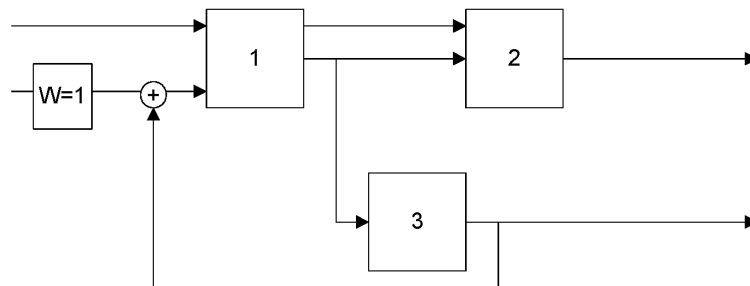


Рис. 4. Эквивалентная схема

```
>> s4 = tf(1)
Transfer function:
1
>> sys=append(s1,s2,s3,s4);
>> Q=[2 -4 5; 3 1 0; 4 2 0; 5 2 0];
>> in=[1 5];
>> out=[3 4];
>> s_com=connect(sys,Q, in,out);
```

Обращаясь к данным объекта, можно получить матрицы A, B, C:

```
>> A=s_com.A;
>> B=s_com.B;
>> C=s_com.C;
```

5. Вычислим ранги матриц наблюдаемости и управляемости итоговой системы:

```
>> rank(ctrb(A,B))
ans =
    6
>> rank(observ(A,C))
ans =
    6
```

Результаты показывают, что система управляема и наблюдаема.

Отчет о работе

Отчет оформляется в соответствии с требованиями, предъявляемыми к оформлению работ в вузе, и должен содержать:

1. Титульный лист
2. Наименование и цель работы.
3. Результаты выполнения работы.
4. Анализ результатов и выводы.

Контрольные вопросы

1. Дать определение и примеры состояний управляемой системы.
2. Показать на примере справедливость принципа суперпозиции.
3. Вывести уравнения в пространстве состояний для заданной схемы соединения трех систем.
4. Провести анализ влияния размерности векторов управления и выходов на управляемость и наблюдаемость схемы.

Таблица 3 – Варианты заданий 11-18

11	$1. \begin{cases} \dot{x}^1 = \begin{pmatrix} 5 & 3 \\ 2 & 1 \end{pmatrix} x^1 + \begin{pmatrix} -1 \\ 3 \end{pmatrix} u^1 \\ y^1 = \begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix} x^1 \end{cases}$ $2. \begin{cases} -\dot{x}^3 + 3\dot{x}^3 - 2x^3 = 2u \\ y^3 = x^3 \end{cases}$ $3. \begin{cases} \dot{x}^3 = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix} x^3 + \begin{pmatrix} 1 \\ 2 \end{pmatrix} u^3 \\ y^3 = \begin{pmatrix} -1 & 2 \end{pmatrix} x^2 \end{cases}$	5
12	$1. \begin{cases} -\dot{x}^3 + 2\dot{x}^3 - x^3 = 4u \\ y^3 = x^3 \end{cases}$ $2. \begin{cases} \dot{x}^2 = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix} x^2 + \begin{pmatrix} 1 & 5 \\ -2 & 1 \end{pmatrix} u^2 \\ y^2 = \begin{pmatrix} 4 & 3 \end{pmatrix} x^2 \end{cases}$ $3. \begin{cases} \dot{x}^3 = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix} x^3 + \begin{pmatrix} 14 \\ 1 \end{pmatrix} u^3 \\ y^3 = \begin{pmatrix} 5 & 2 \end{pmatrix} x^2 \end{cases}$	7
13	$1. \begin{cases} \dot{x}^3 + 2\dot{x}^3 - x^3 = -2u \\ y^3 = x^3 \end{cases}$ $2. \begin{cases} \dot{x}^2 = \begin{pmatrix} -1 & 2 \\ 3 & 2 \end{pmatrix} x^2 + \begin{pmatrix} 1 & -1 \\ 2 & 1 \end{pmatrix} u^2 \\ y^2 = \begin{pmatrix} -2 & 3 \end{pmatrix} x^2 \end{cases}$ $3. \begin{cases} \dot{x}^3 = \begin{pmatrix} 1 & 2 \\ 3 & -2 \end{pmatrix} x^3 + \begin{pmatrix} -4 \\ 1 \end{pmatrix} u^3 \\ y^3 = \begin{pmatrix} 5 & -1 \\ 3 & 1 \end{pmatrix} x^2 \end{cases}$	6
14	$1. \begin{cases} \dot{x}_1^1 = 2x_2^1 + u \\ \dot{x}_2^1 = -x_1^1 + 3x_2^1 - u \end{cases}$ $\begin{cases} y_1^1 = x_1^1 \\ y_2^1 = x_1^2 - 2x_2^1 \end{cases}$ $2. \begin{cases} \dot{x}^2 = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix} x^2 + \begin{pmatrix} 1 & 5 \\ 2 & 1 \end{pmatrix} u^2 \\ y^2 = \begin{pmatrix} 4 & 3 \end{pmatrix} x^2 \end{cases}$ $3. \begin{cases} \dot{x}^3 = \begin{pmatrix} 1 & 2 \\ 3 & -2 \end{pmatrix} x^3 + \begin{pmatrix} -4 \\ 1 \end{pmatrix} u^3 \\ y^3 = \begin{pmatrix} 5 & -1 \\ 3 & 1 \end{pmatrix} x^2 \end{cases}$	8
15	$1. \begin{cases} \dot{x}_1^1 = -2x_2^1 + 2u \\ \dot{x}_2^1 = -x_1^1 + 3x_2^1 - u \end{cases}$ $\begin{cases} y_1^1 = -x_1^1 \\ y_2^1 = x_1^2 - 2x_2^1 \end{cases}$ $2. \begin{cases} \dot{x}^2 = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix} x^2 + \begin{pmatrix} 1 & -1 \\ 2 & 1 \end{pmatrix} u^2 \\ y^2 = \begin{pmatrix} -4 & 3 \end{pmatrix} x^2 \end{cases}$ $3. \begin{cases} \dot{x}^3 = \begin{pmatrix} 1 & 2 \\ 3 & -2 \end{pmatrix} x^3 + \begin{pmatrix} -4 \\ 1 \end{pmatrix} u^3 \\ y^3 = \begin{pmatrix} 5 & -1 \\ 3 & 1 \end{pmatrix} x^2 \end{cases}$	8
16	$1. \begin{cases} \dot{x}^1 = \begin{pmatrix} 5 & 3 \\ 2 & 1 \end{pmatrix} x^1 + \begin{pmatrix} -1 \\ 3 \end{pmatrix} u^1 \\ y^1 = \begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix} x^1 \end{cases}$ $2. \begin{cases} \dot{x}_1^2 = 3x_1^2 - 2x_2^2 + 3u \\ \dot{x}_2^2 = -x_1^2 + 3x_2^2 - u \end{cases}$ $\begin{cases} y_1^1 = -x_1^1 \\ y_2^1 = x_1^2 - x_2^1 \end{cases}$ $3. \begin{cases} \dot{x}^3 = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix} x^3 + \begin{pmatrix} 1 \\ 2 \end{pmatrix} u^3 \\ y^3 = \begin{pmatrix} -1 & 2 \end{pmatrix} x^2 \end{cases}$	9
17	$1. \begin{cases} \dot{x}_1^1 = x_1^1 + 4x_2^1 + 3u \\ \dot{x}_2^1 = -x_1^1 + 3x_2^1 - 2u \end{cases}$ $\begin{cases} y_1^1 = -x_1^1 + 2x_2^1 \\ y_2^1 = x_1^2 - x_2^1 \end{cases}$ $2. \begin{cases} \dot{x}^2 = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix} x^2 + \begin{pmatrix} 1 & 5 \\ -2 & 1 \end{pmatrix} u^2 \\ y^2 = \begin{pmatrix} 4 & 3 \end{pmatrix} x^2 \end{cases}$ $3. \begin{cases} \dot{x}^3 = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix} x^3 + \begin{pmatrix} 14 \\ 1 \end{pmatrix} u^3 \\ y^3 = \begin{pmatrix} 5 & 2 \end{pmatrix} x^2 \end{cases}$	10
18	$1. \begin{cases} \dot{x}_1^1 = x_1^1 - 4x_2^1 + 3u \\ \dot{x}_2^1 = -x_1^1 + 3x_2^1 - 4u \end{cases}$ $\begin{cases} y_1^1 = -x_1^1 + 5x_2^1 \\ y_2^1 = x_1^2 - x_2^1 \end{cases}$ $2. \begin{cases} \dot{x}^2 = \begin{pmatrix} -1 & 2 \\ 3 & 2 \end{pmatrix} x^2 + \begin{pmatrix} 1 & -1 \\ 2 & 1 \end{pmatrix} u^2 \\ y^2 = \begin{pmatrix} -2 & 3 \end{pmatrix} x^2 \end{cases}$ $3. \begin{cases} \dot{x}^3 = \begin{pmatrix} 1 & 2 \\ 3 & -2 \end{pmatrix} x^3 + \begin{pmatrix} -4 \\ 1 \end{pmatrix} u^3 \\ y^3 = \begin{pmatrix} 5 & -1 \\ 3 & 1 \end{pmatrix} x^2 \end{cases}$	8

Структурные схемы к вариантам

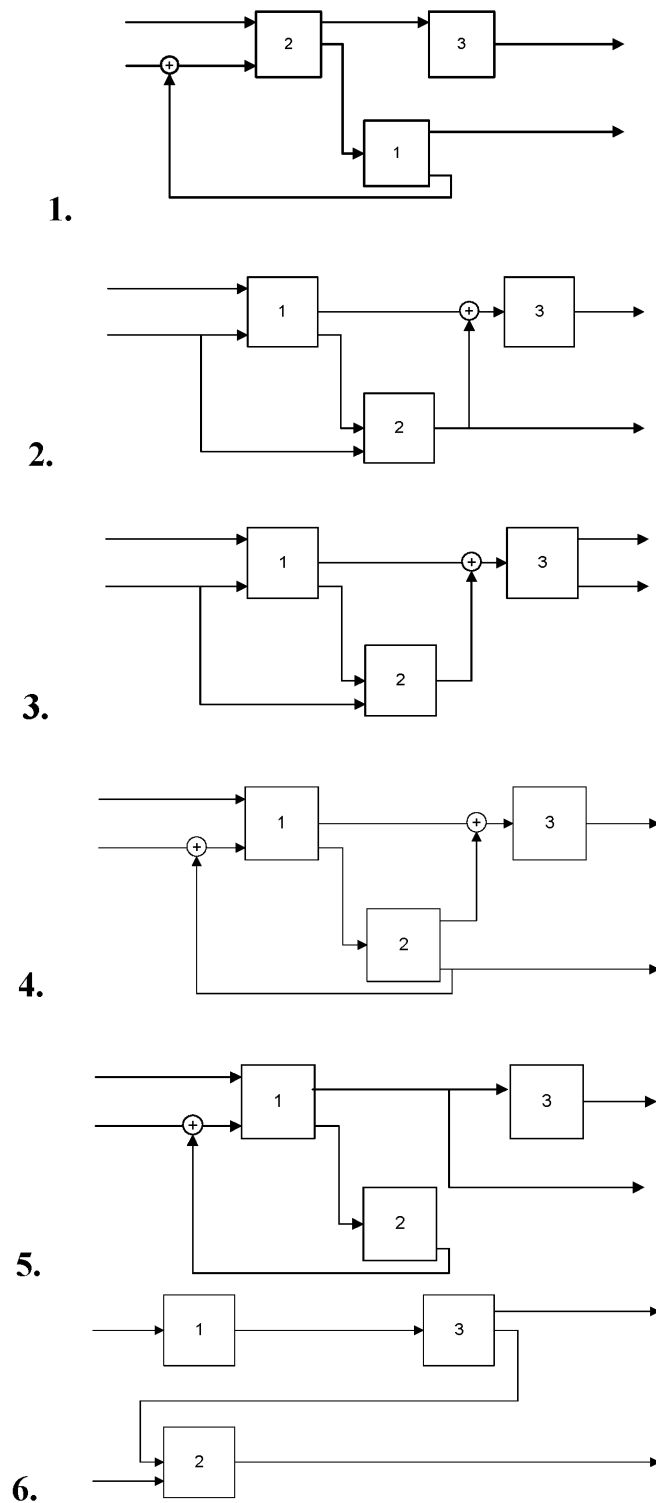


Рис. 5. Структурные схемы 1 - 6 к вариантам заданий

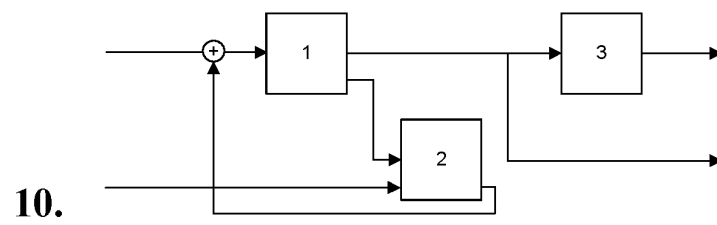
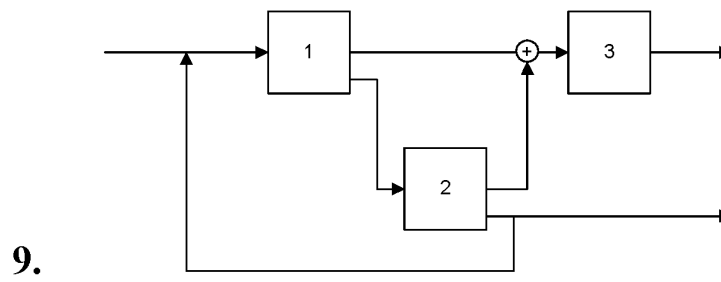
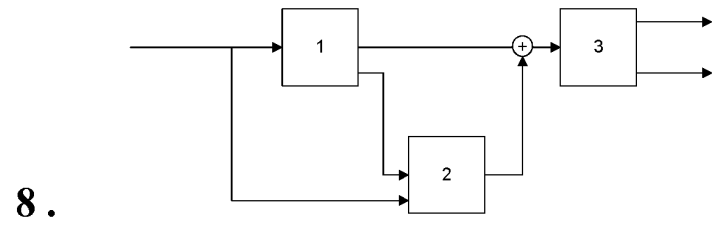
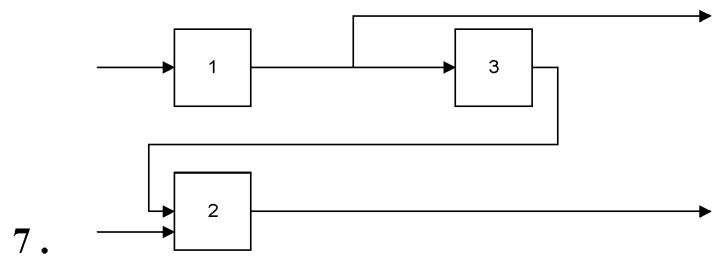


Рис. 6. Структурные схемы 7 - 10 к вариантам заданий

Литература

1. Никульчев Е.В. Практикум по теории управления в среде MATLAB: Учебное пособие. – М.: МГАПИ, 2002.
2. Андриевский Б.Р., Фрадков А.Л. Избранные главы теории автоматического управления с примерами в системе MATLAB. - СПб.: Наука, 1999.
3. Медведев В.С., Потемкин В.Г. Control System Toolbox. MATLAB 5 для студентов /Под общ. ред. Потемкина В.Г. - М.: Диалог-МИФИ, 1999.
4. Семенов В.В., Пантелеев А.В., Бортакровский А.С. Математическая теория управления в примерах и задачах. - М.: МАИ, 1997.
5. Сю Д., Мейер А. Современная теория автоматического управления и ее применение. - М.: Машиностроение, 1972.
6. Дьяконов В., Круглов В. MATLAB. Анализ, идентификация и моделирование систем. Специальный справочник. - СПб.: Питер, 2002
7. Лазарев Ю.Ф. MATLAB 5.x. Библиотека студента. - К.: Издательская группа BHV, 2000.
8. Малышев С. А. Метод корневого годографа: Методического указания по выполнению лабораторных работ по теории управления. - М.: МИП, 1992.
9. Методы классической и современной теории автоматического управления: Учебник в 3-х т. / Под общ. ред. Н.Д. Егупова - М.: Изд-во МГТУ им. Н.Э.Баумана, 2000.
10. Мороз А.И. Методического указания по выполнению лабораторных работ по теории управления. - М.: МИП, 1989.
11. Using the Control System Toolbox with MATLAB 6: Computation. Visualization. Programming - The MathWorks, Inc., 2000.

Практическая работа № 4 Фильтр Калмана

Цель работы

Целью работы является фильтрация дискретных сигналов используя один из методов оптимальной линейной фильтрации – алгоритм фильтра Калмана в среде ПО MATLAB.

Пояснения к работе

Фильтр Калмана – последовательный рекуррентный алгоритм, использующий принятую модель динамической системы для получения оценки, которая может быть существенно скорректирована в результате анализа каждой новой выборки измерений во временной последовательности.

Фильтр Калмана предназначен для рекуррентного дооценивания вектора состояния априорно известной динамической системы. Для расчёта текущего состояния системы необходимо знать текущее измерение (наблюдение), а также предыдущее состояние самого фильтра. Таким образом, фильтр Калмана реализован во временной области (домене), а не в частотной области.

Алгоритм фильтра Калмана работает в два этапа. На этапе прогнозирования фильтр Калмана экстраполирует значения переменных состояния, а также их неопределённости. На втором этапе по данным измерения (полученного с некоторой погрешностью) результат экстраполяции уточняется. Благодаря этому алгоритм может в реальном времени отслеживать состояние некоторого объекта/процесса.

Данный алгоритм находит применение в процессе управления многими сложными динамическими системами, например, непрерывными производственными процессами, самолетами, кораблями и космическими аппаратами. При управлении динамической системой, прежде всего, необходимо полностью знать её состояние в каждый момент времени. Но измерение всех переменных, которыми необходимо управлять, не всегда возможно, и в этих случаях фильтр Калмана является тем средством, которое

позволяет восстановить недостающую информацию посредством имеющихся неточных (зашумленных) измерений.

Фильтр Калмана также называют фильтром Стратоновича-Калмана-Бьюси, т.к. он является частным случаем более общего, нелинейного фильтра, разработанного советским математиком Р.Л. Стратоновичем [1]. Алгоритм фильтра Калмана назван в честь Р.Э. Калмана, являющегося одним из основных разработчиков теории данного фильтра [2, 3]. Под алгоритмом дискретного фильтра Калмана понимают последовательный рекуррентный алгоритм фильтрации информации дискретных динамических систем, заданных в пространстве состояний.

Для описания работы алгоритма фильтра Калмана введем следующие обозначения [4].

Текущее общее состояние некоторой системы (под состоянием системы будем подразумевать подлежащие оценке значения физических величин) представим в виде значений элементов $x_1 \dots x_n$ вектора состояния x , обладающего размерностью n :

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}.$$

Измерения (наблюдения) величин на выходе системы, обозначим в виде вектора z размерностью m :

$$z = \begin{bmatrix} z_1 \\ z_2 \\ \dots \\ z_m \end{bmatrix}.$$

Элементы $z_1 \dots z_m$ вектора z являются входными данными для алгоритма фильтра Калмана – данными, полученными в ходе измерений датчиками физических величин.

Модель системы для дискретного времени подразумевает, что состояние системы x в момент времени $(k + 1)$ определяется уравнением состояния:

$$x(k + 1) = F(k)x(k) + B(k)u(k) + w(k), \quad (1)$$

где:

$F(k)$ – матрица эволюции (процесса изменения состояния) системы в момент времени k – размерность матрицы $n \times n$;

$x(k)$ – состояние системы в момент времени k ;

$B(k)$ – матрица управления – размерность матрицы $n \times e$;

$u(k)$ – вектор управляющих воздействий размерностью e в момент времени k ;

$w(k)$ – вектор шума процесса (случайного процесса) размерностью n , описывающий случайный характер эволюции системы в момент времени k , характеризуется ковариационной матрицей $Q(k)$ – матрицей ковариаций шума процесса размерностью $n \times n$.

Произведение матрицы управления B на вектор управляющих воздействий u в (1) может отсутствовать, если в фильтре Калмана не учитываются управляющие воздействия на систему. В задаче фильтрации данных матрица B является нулевой матрицей.

В момент времени k выполняется наблюдение системы с целью сформировать вектор $z(k)$. Данный вектор выразим в виде уравнения наблюдений:

$$z(k) = H(k)x(k) + v(k), \quad (2)$$

где:

$H(k)$ – матрица наблюдений системы – размерность матрицы $m \times n$;

$x(k)$ – состояние системы в момент времени k ;

$v(k)$ – вектор шума измерения размерностью m , характеризующегося ковариационной матрицей $R(k)$ – матрицей ковариаций ошибок измерения размерностью $m \times m$.

Вектор $\hat{x}(k | k-1)$ априорной (прогнозируемой) оценки состояния системы выразим следующим образом:

$$\hat{x}(k | k-1) = F(k)\hat{x}(k-1 | k-1) + B(k)u(k), \quad (3)$$

где:

$\hat{x}(k-1 | k-1)$ – вектор оценки состояния системы в момент времени $(k-1)$ с учетом наблюдений по момент времени $(k-1)$ включительно.

Ковариационную матрицу $P(k | k-1)$ априорной (прогнозируемой) ошибки (ошибки оценки) состояния системы выразим как

$$P(k | k-1) = F(k)P(k-1 | k-1)F^T(k) + Q(k), \quad (4)$$

где:

$\mathbf{P}(k-1 | k-1)$ – матрица ковариаций ошибки (ошибки оценки) состояния системы в момент времени $(k-1)$ – размерность матрицы $n \times n$,

$\mathbf{Q}(k)$ – матрица ковариаций шума процесса.

Вектор $\hat{\mathbf{x}}(k | k)$ апостериорной (обновленной) оценки состояния системы выразим следующим образом:

$$\begin{aligned} \hat{\mathbf{x}}(k | k) &= [\mathbf{1} - \mathbf{G}(k)\mathbf{H}(k)]\hat{\mathbf{x}}(k | k-1) + \mathbf{G}(k)\mathbf{z}(k) = \\ &= \hat{\mathbf{x}}(k | k-1) + \mathbf{G}(k)[\mathbf{z}(k) - \mathbf{H}(k) \cdot \hat{\mathbf{x}}(k | k-1)] \end{aligned} \quad (5)$$

где:

$\mathbf{1}$ – единичная матрица размерностью $n \times n$,

$\mathbf{G}(k)$ – матрица коэффициентов наблюдателя, оптимальных по Калману – размерность матрицы $n \times m$,

$\mathbf{H}(k)$ – матрица наблюдений системы.

Ковариационную матрицу $\mathbf{P}(k | k)$ апостериорной (обновленной) ошибки (ошибки оценки) состояния системы выразим как

$$\mathbf{P}(k | k) = \mathbf{P}(k | k-1) - \mathbf{G}(k)\mathbf{S}(k)\mathbf{G}^T(k), \quad (6)$$

где $\mathbf{S}(k)$ – матрица ковариаций отклонений (инноваций) размерностью $m \times m$:

$$\mathbf{S}(k) = \mathbf{H}(k)\mathbf{P}(k | k-1)\mathbf{H}^T(k) + \mathbf{R}(k). \quad (7)$$

Вектор некоторого отклонения (инновации) $\tilde{\mathbf{y}}(k)$ полученного на шаге k наблюдения системы от наблюдения системы, ожидаемого при произведенном прогнозировании, определим как

$$\tilde{\mathbf{y}}(k) = \mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k | k-1), \quad (8)$$

где $\tilde{\mathbf{y}}(k)$ – вектор отклонения (инновации) размерностью m .

Таким образом, вектор апостериорной (обновленной) оценки состояния системы можно выразить как

$$\hat{\mathbf{x}}(k | k) = \hat{\mathbf{x}}(k | k-1) + \mathbf{G}(k)[\mathbf{z}(k) - \mathbf{H}(k) \cdot \hat{\mathbf{x}}(k | k-1)] = \hat{\mathbf{x}}(k | k-1) + \mathbf{G}(k) \cdot \tilde{\mathbf{y}}(k). \quad (9)$$

Вектор $\hat{x}(k|k)$ является результатом работы алгоритма фильтра Калмана.

Вектор оценки состояния \hat{x} и вектор измерений (наблюдений) z в данной работе имеют размерность 2, при этом для решения задачи фильтрации дискретного сигнала в данных векторах используются только элементы \hat{x}_1 и z_1 соответственно.

Для решения задачи фильтрации данных примем значения матриц F и H постоянными:

$$F(k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad H(k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Далее приведен пример программы для среды MATLAB, выполняющей генерацию зашумленного сигнала и его последующую фильтрацию.

Пример программы в среде MATLAB, выполняющей фильтрацию данных

```
%-----
close all;
clear all;
%-----

Amp      = 100;
MAX_NOISE = 50;
Hz       = 1 / 100;

OPEN = 1; %установка значения флага работы программы

sig = 0; %текущее значение сигнала в формате числа с плавающей запятой

%-----

Qx = 0.001 %значения дисперсий случайных величин на
           %главной диагонали матрицы ковариаций шума процесса
Rx = 0.01; %значения дисперсий случайных величин на
           %главной диагонали матрицы ковариаций шума измерений

clear S; %инициализация (очистка) массива структур для
         %хранения переменных данных фильтра

Q = [ Qx 0.0; 0.0 Qx ]; %матрица ковариаций шума процесса
R = [ Rx 0.0; 0.0 Rx ]; %матрица ковариаций шума измерений

F = [ 1.0 0.0; 0.0 1.0 ]; %матрица эволюции (процесса изменения состояния)
                             %системы
H = [ 1.0 0.0; 0.0 1.0 ]; %матрица наблюдений (коэффициентов выхода) системы

S.x = [ 0.0; 0.0 ]; %инициализация вектора оценки состояния системы
```

```

S.z = [ 0.0; 0.0 ]; %инициализация вектора наблюдений (выхода системы)

S.P = [ 0.0 0.0; 0.0 0.0 ]; %матрица ковариаций ошибки оценки состояния системы

%-----

%создание окна интерфейса пользователя
FIG = figure('Name', 'Монитор', 'NumberTitle', 'off', ...
            'units', 'normalized', ...
            'position', [0.25 0.25 0.5 0.5]);

%установка параметров шрифта текста, отображаемого в окне
set(gcf, 'FontName', 'Arial Cyr', 'FontSize', 10);

%создание кнопки для остановки процесса фильтрации
OFF_BUTTON = uicontrol('style', 'pushbutton', 'string', 'Остановить', ...
                      'units', 'normalized', ...
                      'position', [0.01 0.94 0.4 0.05], 'fontsize', 9, ...
                      'callback', 'OPEN = 0;');

%установка возможности закрытия окна только при остановке
set(gcf, 'CloseRequestFcn', 'if (OPEN == 0) close(FIG); end;');

%-----

j = 1; %инициализация переменной-счетчика

while (OPEN) %цикл программы
    %создание случайного числа
    %с помощью функции rand - диапазон значений: -1 ... 1
    noise = (2 * rand(length(1), 1) - 1);

    %создание сигнала в дискретной форме
    sig = Amp * cos(2 * pi * j * Hz) + (MAX_NOISE * noise);

    S(end).z = [ sig; 0.0 ]; %формирование вектора наблюдений

    %априорная (прогнозируемая) оценка состояния системы
    S(end).x = F * S(end).x;
    S(end).P = F * S(end).P * F' + Q;
    %матрица коэффициентов наблюдателя, оптимальных по Калману
    S(end).G = S(end).P * H' * ...
    inv(H * S(end).P * H' + R);
    %апостериорная (обновленная) оценка состояния системы
    S(end).x = S(end).x + S(end).G * (S(end).z - H * S(end).x);
    S(end).P = S(end).P - S(end).G * H * S(end).P;

    S(end + 1) = S(end); %добавление новой записи к массиву структур
                        %переменных данных фильтра

    %применение метода простого скользящего среднего к
    %результатам работы фильтра (оценке состояния системы)
    if (length(S) >= 5)
        aver = mean([ S(end-4).x(1) ...
                     S(end-3).x(1) ...
                     S(end-2).x(1) ...
                     S(end-1).x(1) ...
                     S(end).x(1)   ]);
    else
        aver = S(end).x(1);
    end;
end;

```

```

if (OPEN == 1)
    %построение графика данных наблюдений
    plot(j, sig, 'r-o', 'LineWidth', 3); hold on; grid on;
    %построение графика оценки состояния системы
    %(результата фильтрации)
    plot(j, S(end).x(1), 'k:x', 'LineWidth', 3); hold on; grid on;
    %построение графика усредненной оценки состояния системы
    %(усредненного результата фильтрации)
    plot(j, aver, 'b:d', 'LineWidth', 2); hold on; grid on;
    %отображение легенды
    legend('Данные наблюдений', ...
          'Оценка состояния', ...
          'Усредненная оценка', -1);
    %обновление (перерисовка) окна интерфейса пользователя
    drawnow;
end;

j = j + 1;
end;

%-----
clear S;
%-----

```

На рис. 1 представлено изображение окна интерфейса пользователя, появляющегося на экране монитора персонального компьютера при выполнении в среде MATLAB текста программы фильтрации данных.

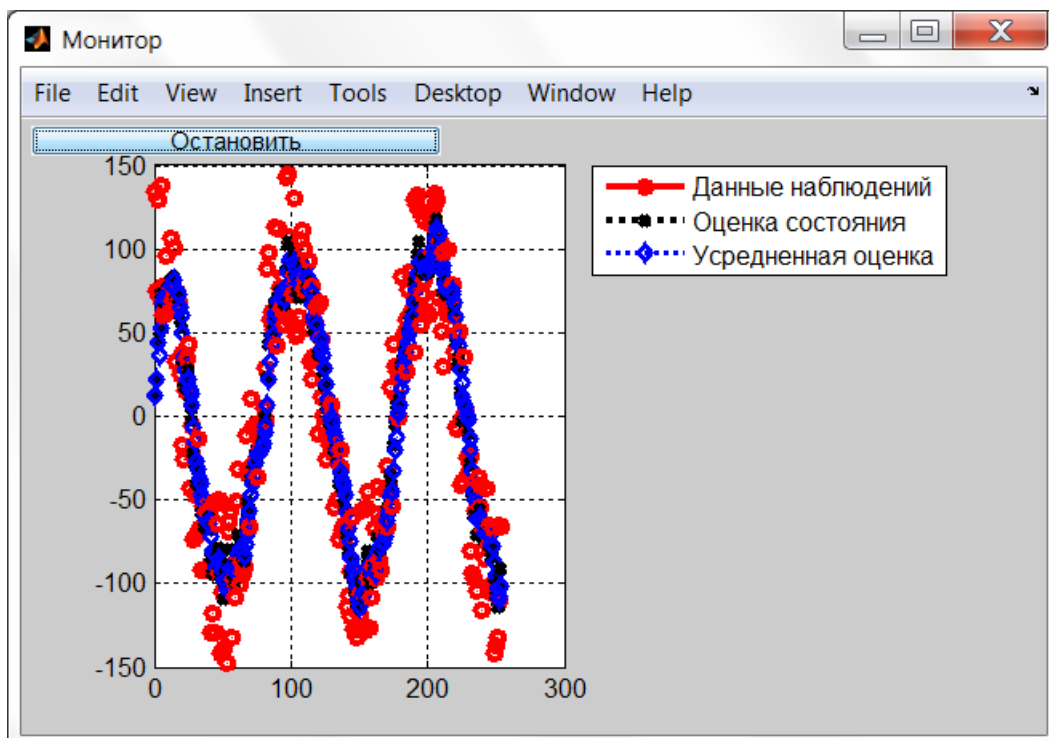


Рис. 1 Вид окна интерфейса пользователя

Порядок выполнения работы

1. По заданию преподавателя, пользуясь данными табл. 1 и примером программы для среды MATLAB, составить программу для формирования зашумленного сигнала и его фильтрации, выбрав значения параметров формируемого контроллером зашумленного сигнала:

Amp - амплитуда синусоидального сигнала
 Hz - частота синусоидального сигнала, Гц
 MAX_NOISE - максимальная "амплитуда" шума, налагаемого на синусоидальный сигнал

Для выполнения фильтрации зашумленного сигнала подобрать значения дисперсий случайных величин в диапазоне от 0,01 до 1:

Qx - значения дисперсий случайных величин на главной диагонали матрицы ковариаций шума процесса
 Rx - значения дисперсий случайных величин на главной диагонали матрицы ковариаций шума измерений

2. Запустить работу программы в среде MATLAB. Применяя четыре различных пары значений дисперсий Qx и Rx, сделать для каждой пары значений Qx и Rx снимок окна интерфейса программы (см. рис. 1). Сделать выводы о разных полученных результатах фильтрации сигнала.

Содержание отчета по работе

1. Титульный лист с указанием названия работы и номера варианта задания.
2. Лист с вариантом задания.
3. Краткое описание хода работы.
4. Текст программы фильтрации сигнала на языке MATLAB.
5. Снимки окна интерфейса среды MATLAB, выводимого на экран монитора при выполнении программы фильтрации сигнала.
6. График зависимости значения выходного сигнала датчика от значения измеряемой датчиком физической величины.
7. Выводы по проделанной работе и анализ полученных результатов.

После оформления отчет проверяется преподавателем. Если в отчете содержатся ошибки, преподаватель возвращает отчет студенту для исправления и/или доработки. Отчет по работе, в котором все

ошибки и замечания устранены, принимается преподавателем для последующей защиты студентом. На защите студент должен ответить на вопросы по выполненной работе.

Контрольные вопросы

1. При каких предположениях относительно шума измерений и шума системы строится теория фильтра Калмана?
2. Чем отличается априорная оценка вектора состояния и апостериорная оценка вектора состояния? Каков принцип построения этих оценок?
3. Записать уравнения фильтра Калмана для векторных наблюдений и процессов.
4. В каком случае фильтр Калмана применим для нелинейных систем? Как в этом случае определяется матрица F и матрица H ?
5. Опишите один цикл работы фильтра Калмана.
6. Какую роль играют априорные сведения в задаче фильтрации?
7. Каково основное назначение математического аппарата оптимальной линейной фильтрации с точки зрения обработки сигналов?

Варианты заданий

Таблица 1 – Варианты заданий

Вариант	Задание
1	Amp = 5; Hz = 0.5; MAX_NOISE = 2;
2	Amp = 1; Hz = 1.5; MAX_NOISE = 0.2;
3	Amp = 5; Hz = 1; MAX_NOISE = 1.5;
4	Amp = 10; Hz = 2; MAX_NOISE = 2;
5	Amp = 50; Hz = 0.2; MAX_NOISE = 10;
6	Amp = 6; Hz = 2; MAX_NOISE = 2;
7	Amp = 1; Hz = 0.8; MAX_NOISE = 0.2;
8	Amp = 4; Hz = 1; MAX_NOISE = 1.5;
9	Amp = 9; Hz = 0.5; MAX_NOISE = 4;
10	Amp = 10; Hz = 2; MAX_NOISE = 3;
11	Amp = 3; Hz = 0.1; MAX_NOISE = 1;
12	Amp = 100; Hz = 0.7; MAX_NOISE = 35;
13	Amp = 1; Hz = 0.3; MAX_NOISE = 0.4;
14	Amp = 1.5; Hz = 0.1; MAX_NOISE = 0.5;
15	Amp = 12; Hz = 1; MAX_NOISE = 4;
16	Amp = 4; Hz = 0.5; MAX_NOISE = 1;

Вариант	Задание
17	Amp = 50; Hz = 2; MAX_NOISE = 10;
18	Amp = 0.5; Hz = 1; MAX_NOISE = 0.2;
19	Amp = 5; Hz = 1.5; MAX_NOISE = 2;
20	Amp = 2.5; Hz = 0.1; MAX_NOISE = 1;
21	Amp = 40; Hz = 1.1; MAX_NOISE = 15;
22	Amp = 20; Hz = 1.5; MAX_NOISE = 8;
23	Amp = 80; Hz = 1; MAX_NOISE = 35;
24	Amp = 3; Hz = 0.3; MAX_NOISE = 1.2;
25	Amp = 25; Hz = 0.5; MAX_NOISE = 10;
26	Amp = 3.5; Hz = 0.9; MAX_NOISE = 1;
27	Amp = 2.7; Hz = 0.4; MAX_NOISE = 1;
28	Amp = 8; Hz = 2; MAX_NOISE = 2;
29	Amp = 7; Hz = 0.5; MAX_NOISE = 2.9;
30	Amp = 30; Hz = 0.5; MAX_NOISE = 10;

Литература

1. Стратонович, Р.Л. К теории оптимальной нелинейной фильтрации случайных функций // Теория вероятностей и ее применения. 1959. Т. 4, № 2. С. 239-242.
2. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems / Transactions of the ASME (American Society of Mechanical Engineers), Journal of Basic Engineering, 1960, vol. 82 (1), P. 35-45.
3. Kalman, R.E., Busy, R.S. New Results in Linear Filtering and Prediction Theory / Transactions of the ASME (American Society of Mechanical Engineers), Journal of Basic Engineering, 1961, vol. 83 (1), P. 95-108.
4. Безмен, П.А. Комплексование данных системы управления мобильным роботом с использованием расширенного фильтра Калмана / П.А. Безмен // Известия Юго-Западного государственного университета. – Курск: Юго-Западный государственный университет, 2019. – №2 – С. 53-64.
5. Яцун, С.Ф. Информационные устройства и системы в мехатронике: учебное пособие / С.Ф. Яцун, П.А. Безмен; Юго-Зап. гос. ун-т. Курск, 2013. – 240 с.
6. Яцун, С.Ф. Датчики и обработка сигналов в мехатронике: учебное пособие / С.Ф. Яцун, П.А. Безмен; Юго-Зап. гос. ун-т. Курск, 2014. – 238 с.