Министерство образования и науки РФ

Федеральное государственное автономное образовательное учреждение высшего образования

«Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»

В.В. ПУТОВ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО КОМПЬЮТЕРНЫМ ИССЛЕДОВАНИЯМ ПО ДИСЦИПЛИНЕ «НЕЛИНЕЙНОЕ И АДАПТИВНОЕ УПРАВЛЕНИЕ В ТЕХНИЧЕСКИХ СИСТЕМАХ»

Санкт-Петербург
2018

1. СИНТЕЗ И ИССЛЕДОВАНИЕ АДАПТИВНЫХ СИСТЕМ С ПАРАМЕТРИЧЕСКОЙ АДАПТАЦИЕЙ ДЛЯ ОБЪЕКТОВ ПЕРВОГО ПОРЯДКА

Цель работы: овладение навыками исследования адаптивной системы, исследование эффективности адаптивного управления при изменении параметров уравнений его настроек и исследование возможностей адаптивного управления по стабилизации объекта управления.

Теоретические сведения. Система с прямой адаптацией и параметрической настройкой. Рассмотрим линейный стационарный объект порядка n с одним входом и имеющий следующий вид

$$\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{b}u, \tag{1.1}$$

где $\mathbf{x} \in R^n$ – вектор состояния объекта (1.1), $u = u_a + u_0$ – скалярное управление, в том числе, u_0 – программное управление, u_a – адаптивное управление, $\mathbf{b} \in R^n$ – матрица-столбец входа, $\mathbf{A} \in R^{n\times n}$ – матрица состояния объекта.

Полагаем, что $\mathbf{A}, \mathbf{b}$ точно не известны, а известны лишь их усреднённые (номинальные) значения $\mathbf{A}_0, \mathbf{b}_0$. В этом случае можно использовать известный адаптивный закон управления следующего вида

$$u_a = \mathbf{k}_A^T \mathbf{x} + k_b u_0, \tag{1.2}$$

где вектор-стобец $\mathbf{k}_A$ и скаляр k_b – настраиваемые параметры адаптивного закона (1.2).

Желаемая динамика адаптивной системы задаётся эталонной моделью вида

$$\mathbf{x}_M = \mathbf{A}_M \mathbf{x}_M + \mathbf{b}_M u_0, \tag{1.3}$$

где $\mathbf{A}_M, \mathbf{b}_M$ – $n\times n$ и $n\times 1$ -мерные постоянные матрицы, $\mathbf{A}_M$ – гурвицева, $\mathbf{x}_M \in R^n$ – вектор состояния эталонной модели, u_0 – любая непрерывная ограниченная функция времени.

Дифференциальные уравнения алгоритмов настройки параметров $\mathbf{k}_A$, k_b (без огрубления) имеют следующий вид

$$\begin{cases} \mathbf{k}_A^T = -\mathbf{b}_M^T \mathbf{P}\mathbf{e}\mathbf{x}^T \boldsymbol{\Gamma}_A, \\ k_b = -\gamma_b \mathbf{b}_M^T \mathbf{P}\mathbf{e} u_0, \end{cases} \tag{1.4}$$

где $\mathbf{e} = \mathbf{x} - \mathbf{x}_M$ – n-мерный вектор ошибки, характеризующий отклонение состояния объекта от состояния эталонной модели, $\mathbf{\Gamma}_A$ – симметричная положительно определённая матрица, γ_b – положительный коэффициент усиления алгоритмов настройки, $\mathbf{P}$ – симметричная положительно определённая матрица, удовлетворяющая уравнению Ляпунова вида

$$\mathbf{A}_M^T \mathbf{P} + \mathbf{P}\mathbf{A}_M = -\mathbf{Q}, \tag{1.5}$$

где $\mathbf{Q}$ – любая симметричная положительно определённая (в частности, диагональная) матрица.

Пример расчёта системы с прямой адаптацией и параметрической настройкой для объекта первого порядка. Рассмотрим объект управления в виде апериодического звена первого порядка следующего вида:

$$\begin{cases} \dot{x} = a_0 x + b_0 u \\ y_0 = x \end{cases}, \tag{1.6}$$

Для примера возьмём устойчивый и неустойчивый объекты со следующими параметрами: $a_{0c} = -8, b_{0c} = 5, a_{0н} = 0.8, b_{0н} = 1$. Зададим желаемую переходную характеристику системы с помощью эталонной модели. В качестве эталонной модели также возьмём апериодическое звено первого порядка со следующими параметрами: $a_m = -15, b_m = 15$. Переходные процессы устойчивого и неустойчивого объектов, а также эталонной модели показаны на рис. 1.1.

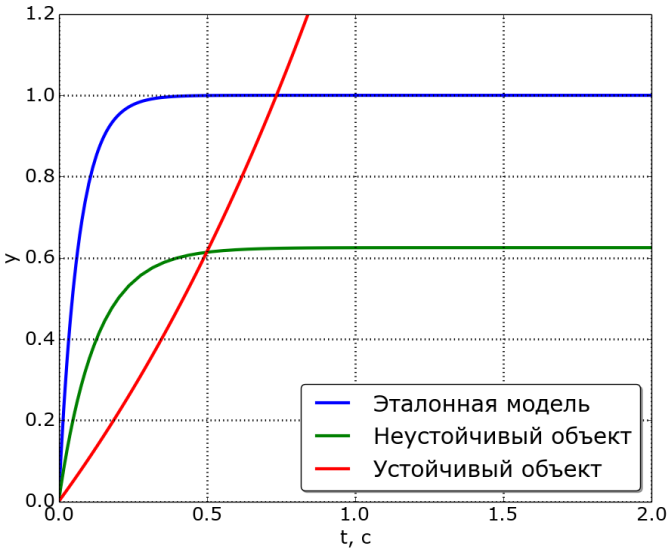

Рис. 1.1 Переходные процессы в исходных системах и эталонной модели

Рассчитаем теперь адаптивную систему с параметрической настройкой, для этого выберем постоянный положительный коэффициент p равным 10, а положительные коэффициенты γ_A, γ_B выберем из диапазона от 1 до 100. На рис. 1.2 представлена структурная схема полученной системы с прямой адаптивной системой управления и эталонной моделью.

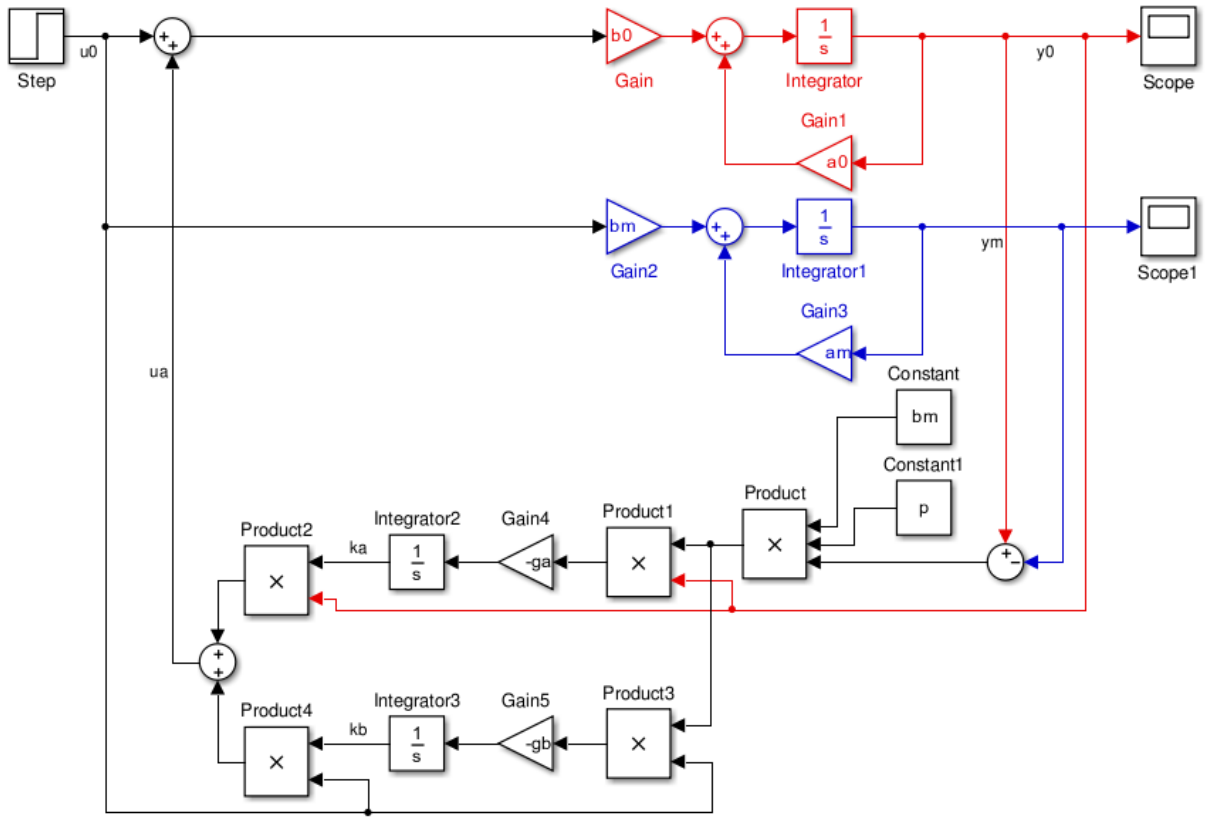

Рис. 1.2. Структурная схема системы с прямой адаптивной системой управления с параметрической настройкой и эталонной моделью

На рис. 1.3 представлены переходные процессы полученной системы с устойчивым объектом при различных значениях параметров γ_A, γ_B в сравнении с переходным процессом эталонной модели.

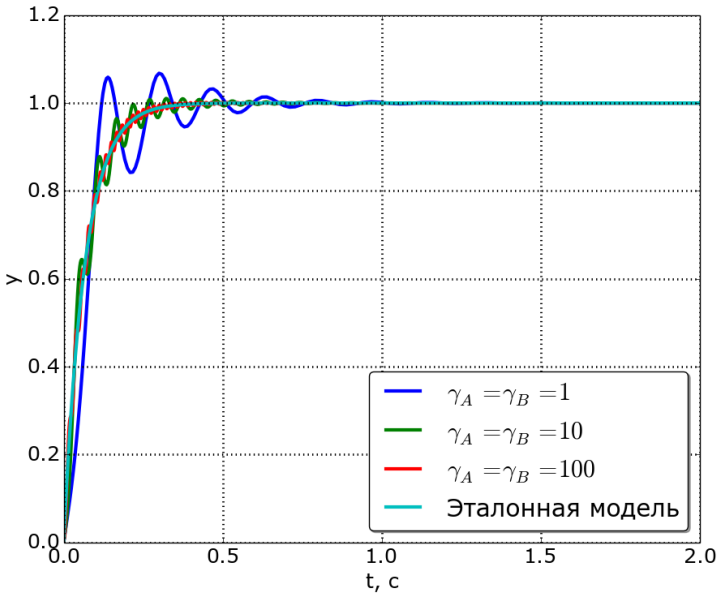

Рис. 1.3. Переходные процессы полученной системы с устойчивым объектом при различных значениях γ_A, γ_B в сравнении с эталонной моделью

На рис. 1.4 представлены переходные процессы полученной системы с неустойчивым объектом при различных значениях параметров γ_A, γ_B в сравнении с переходным процессом эталонной модели.

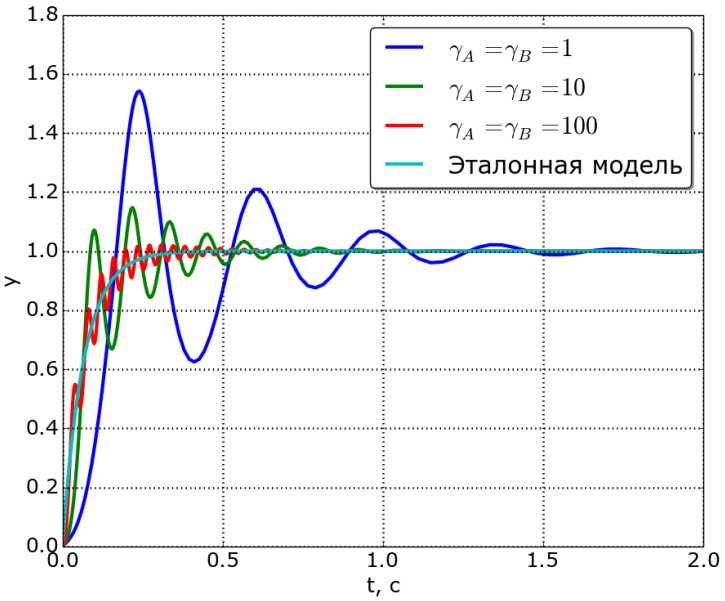

Рис. 1.4. Переходные процессы полученной системы с неустойчивым объектом при различных значениях γ_A, γ_B в сравнении с эталонной моделью

Программа лабораторной работы.

1. Построить и исследовать моделированием систему (1.1) с заданными параметрами a_0, b_0 для устойчивого случая ($a_0 < 0$).
2. Построить эталонную модель (1.3) и сравнить ее динамику с динамикой объекта управления (1.1).
3. Построить адаптивное управление (1.2) с настройками (1.4) и начальными значениями $\gamma_{\text{А}}, \gamma_{\text{В}}, p$ по своему варианту.
4. Исследовать адаптивную систему, изменяя коэффициенты γ_A, γ_B ($p = \text{const}$).
5. Записывать установившиеся значения для $k_A \to k_{\text{А}}^{\text{уст}}$ и $k_B \to k_{\text{В}}^{\text{уст}}$ и сравнивать их со значениями, полученными из формул $k_{\text{А}}^{*} = b_0^{-1}(a_{\text{М}} - a_0)$, $k_{\text{В}}^{*} = b_0^{-1}(b_{\text{М}} - b_0)$.
6. Построить и исследовать моделированием объект (1.1) с заданными параметрами a_0, b_0 для неустойчивого случая ($a_0 > 0$).
7. Повторить пп. 3-5 для случая неустойчивого объекта.

Варианты заданий.

Вариант №1

Первый порядок

Устойчивый объект $a_0 = -0.5, b_0 = 1,\ a_{\text{М}} = -5, b_{\text{М}} = 5,\ p = 2,\ \gamma_A, = \gamma_B = 1$

Неустойчивый объект $a_0 = 0.5, b_0 = 1,\ a_{\text{М}} = -5, b_{\text{М}} = 5,\ p = 2,\ \gamma_A, = \gamma_B = 1$

Вариант №2

Первый порядок

Устойчивый объект $a_0 = -3, b_0 = 1,\ a_{\text{М}} = -15, b_{\text{М}} = 10,\ p = 0.5,\ \gamma_A, = \gamma_B = 0.1$

Неустойчивый объект $a_0 = 3, b_0 = 1,\ a_{\text{М}} = -15, b_{\text{М}} = 10,\ p = 0.5,\ \gamma_A, = \gamma_B = 0.1$

Вариант №3

Первый порядок

Устойчивый объект $a_0 = -2, b_0 = 1,\ a_{\text{М}} = -15, b_{\text{М}} = 7,\ p = 0.1,\ \gamma_A, = \gamma_B = 0.01$

Неустойчивый объект $a_0 = 2, b_0 = 1,\ a_{\text{М}} = -15, b_{\text{М}} = 7,\ p = 0.1,\ \gamma_A, = \gamma_B = 0.01$

Вариант №4

Первый порядок

Устойчивый объект $a_0 = -0.1, b_0 = 1,\ a_{\text{М}} = -1, b_{\text{М}} = 1,\ p = 0.1,\ \gamma_A, = \gamma_B = 0.1$

Неустойчивый объект $a_0 = 0.1, b_0 = 1,\ a_{\text{М}} = -1, b_{\text{М}} = 1,\ p = 0.1,\ \gamma_A, = \gamma_B = 0.1$

Вариант №5

Первый порядок

Устойчивый объект $a_0 = -0.8, b_0 = 0.8,\ a_{\text{М}} = -8, b_{\text{М}} = 8,\ p = 1,\ \gamma_A, = \gamma_B = 1$

Неустойчивый объект $a_0 = 0.8, b_0 = 0.8,\ a_{\text{М}} = -8, b_{\text{М}} = 8,\ p = 1,\ \gamma_A, = \gamma_B = 1$

Вариант №6

Первый порядок

Устойчивый объект $a_0 = -0.15, b_0 = 1,\ a_{\text{М}} = -15, b_{\text{М}} = 5,\ p = 0.1,\ \gamma_A, = \gamma_B = 0.1$

Неустойчивый объект $a_0 = 0.15, b_0 = 1,\ a_{\text{М}} = -15, b_{\text{М}} = 5,\ p = 0.1,\ \gamma_A, = \gamma_B = 0.1$

Вариант №7

Первый порядок

Устойчивый объект $a_0 = -0.1, b_0 = 1,\ a_{\text{М}} = -1, b_{\text{М}} = 1,\ p = 0.1,\ \gamma_A, = \gamma_B = 0.1$

Неустойчивый объект $a_0 = 0.5, b_0 = 1,\ a_{\text{М}} = -5, b_{\text{М}} = 5,\ p = 2,\ \gamma_A, = \gamma_B = 1$

Вариант №8

Первый порядок

Устойчивый объект $a_0 = -0.3, b_0 = 1,\ a_{\text{М}} = -1, b_{\text{М}} = 10,\ p = 0.1,\ \gamma_A, = \gamma_B = 0.1$

Неустойчивый объект $a_0 = 0.35, b_0 = 1,\ a_{\text{М}} = -12, b_{\text{М}} = 1,\ p = 0.1,\ \gamma_A, = \gamma_B = 0.1$

Вариант №9

Первый порядок

Устойчивый объект $a_0 = -0.5, b_0 = 1,\ a_{\text{М}} = -5, b_{\text{М}} = 5,\ p = 2,\ \gamma_A, = \gamma_B = 1$

Неустойчивый объект $a_0 = 0.8, b_0 = 0.8,\ a_{\text{М}} = -8, b_{\text{М}} = 8,\ p = 1,\ \gamma_A, = \gamma_B = 1$

Вариант №10

Первый порядок

Устойчивый объект $a_0 = -3, b_0 = 1,\ a_{\text{М}} = -15, b_{\text{М}} = 10,\ p = 0.5,\ \gamma_A, = \gamma_B = 0.1$

Неустойчивый объект $a_0 = 0.1, b_0 = 1,\ a_{\text{м}} = -1, b_{\text{м}} = 10,\ p = 0.1,\ \gamma_A, = \gamma_B = 0.1$

Вариант №11

Первый порядок

Устойчивый объект $a_0 = -3.5, b_0 = 1,\ a_{\text{м}} = -17, b_{\text{м}} = 8,\ p = 2,\ \gamma_A, = \gamma_B = 1$

Неустойчивый объект $a_0 = 3.5, b_0 = 1,\ a_{\text{м}} = -17, b_{\text{м}} = 8,\ p = 2,\ \gamma_A, = \gamma_B = 1$

Вариант №12

Первый порядок

Устойчивый объект $a_0 = -1.5, b_0 = 1,\ a_{\text{м}} = -15, b_{\text{м}} = 15,\ p = 0.1,\ \gamma_A, = \gamma_B = 0.1$

Неустойчивый объект $a_0 = 1.5, b_0 = 1,\ a_{\text{м}} = -15, b_{\text{м}} = 15,\ p = 0.1,\ \gamma_A, = \gamma_B = 0.1$

Содержание отчёта.

- Цель работы.
- Теоретические сведения.
- Расчёт адаптивной системы управления для устойчивого объекта.
- Выводы о работе адаптивной системы с параметрической настройкой для управления устойчивым объектом.
- Расчёт адаптивной системы управления для неустойчивого объекта.
- Выводы о работе адаптивной системы с параметрической настройкой для управления неустойчивым объектом.

2. СИНТЕЗ И ИССЛЕДОВАНИЕ АДАПТИВНЫХ СИСТЕМ С ПАРАМЕТРИЧЕСКОЙ АДАПТАЦИЕЙ ДЛЯ ОБЪЕКТОВ ВТОРОГО ПОРЯДКА

Цель работы: овладение навыками исследования адаптивной системы, исследование эффективности адаптивного управления при изменении параметров уравнений его настроек и исследование возможностей адаптивного управления по стабилизации объекта управления.

Теоретические сведения. Уравнения системы с прямой адаптацией и параметрической настройкой представлены в лабораторной работе №1.

Пример расчёта системы с прямой адаптацией и параметрической настройкой для объекта второго порядка. Рассмотрим объект управления вида

$$\mathbf{x}(t) = \mathbf{A}_0\mathbf{x}(t) + \mathbf{b}_0 u(t),\ u(t) = u_{\text{A}}(t) + u^0(t), \tag{2.1}$$

где $\mathbf{A}_0 = \begin{bmatrix} 0 & 1 \\ a_1 & a_2 \end{bmatrix}$, $\mathbf{b}_0 = \begin{bmatrix} 0 \\ b \end{bmatrix}$ – неизвестные постоянные матрицы; $\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$ – вектор переменных состояния; $u(t)$ – скалярная вещественная функция; $u^0(t)$ – программное управление, а $u_a(t)$ – адаптивное управление, подлежащее определению.

Эталонная модель описывается уравнением:

$$\mathbf{x}_{\text{M}}(t) = \mathbf{A}_{\text{M}}\mathbf{x}_{\text{M}}(t) + \mathbf{b}_{\text{M}}u^0(t), \tag{2.2}$$

где $\mathbf{A}_{\text{M}} = \begin{bmatrix} 0 & 1 \\ a_{\text{M}1} & a_{\text{M}2} \end{bmatrix}$, $\mathbf{b}_{\text{M}} = \begin{bmatrix} 0 \\ b_{\text{M}} \end{bmatrix}$, $\mathbf{x}_{\text{M}}(t) = \begin{bmatrix} x_{\text{M}1}(t) \\ x_{\text{M}2}(t) \end{bmatrix}$ – переменные состояния.

Для примера возьмём следующие параметры объекта управления и эталонной модели: $a_1 = -0.8, a_2 = -3, b = 1, a_{m1} = -10, a_{m2} = -16, b_m = 16$. На рис. 2.1 представлены переходные процессы исходной системы и эталонной модели.

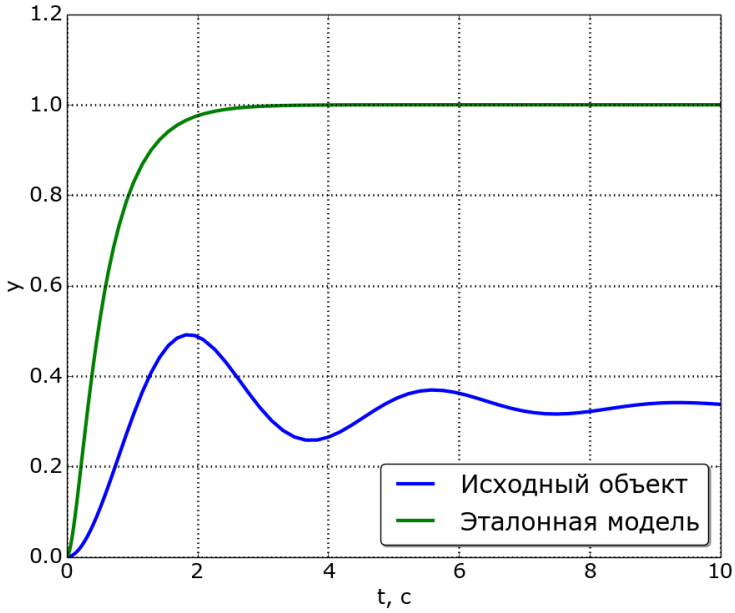

Рис. 2.1. Переходные процессы исходной системы и эталонной модели

Рассчитаем теперь адаптивную систему с параметрической настройкой, для этого выберем постоянную, симметричную, положительно определённую матрицу **P** в виде $\mathbf{P} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$, а положительные коэффициенты $\gamma_{a1}, \gamma_{a2}, \gamma_b$ выберем из диапазона от 1 до 100. На рис. 2.2 представлена структурная схема полученной системы с параметрической настройкой и эталонной моделью.

На рис. 2.3 представлены переходные процессы полученной системы при различных значениях $\gamma_{a1}, \gamma_{a2}, \gamma_b$ в сравнении с переходным процессом эталонной модели.

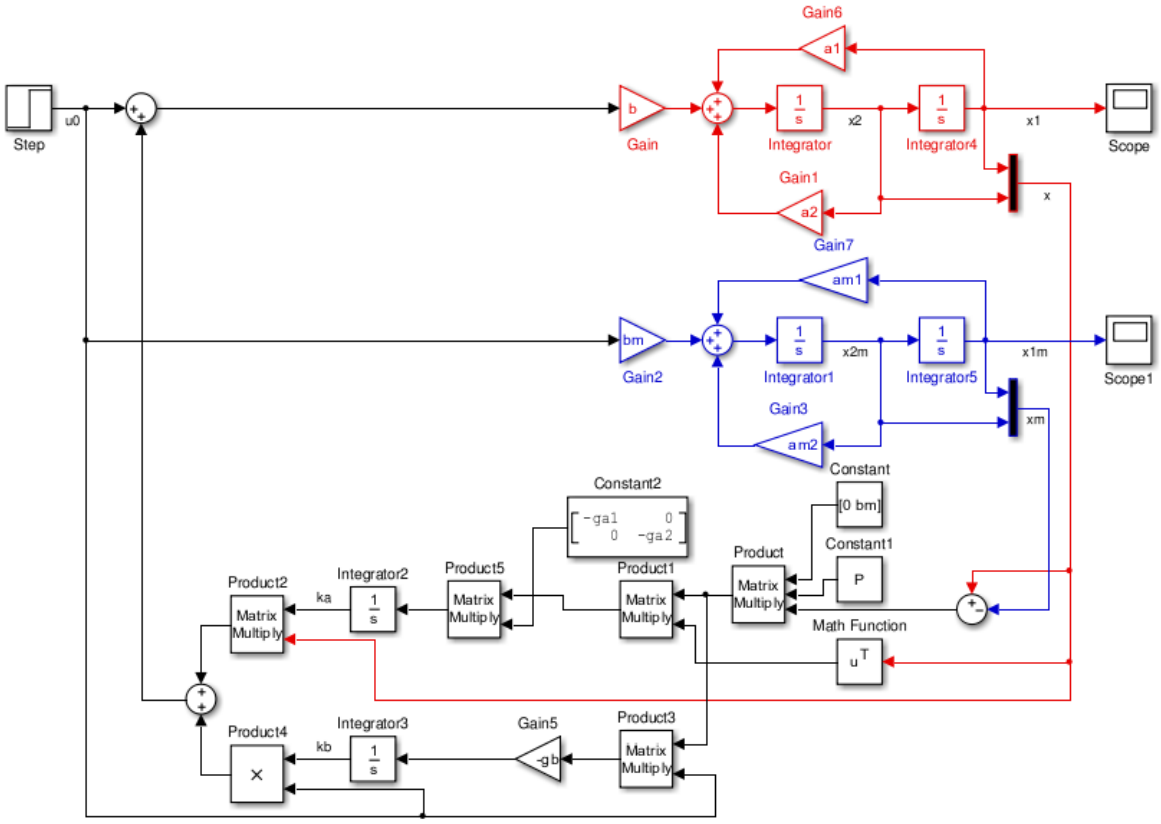

Рис. 2.2. Структурная схема системы с прямой адаптивной системой управления и эталонной моделью

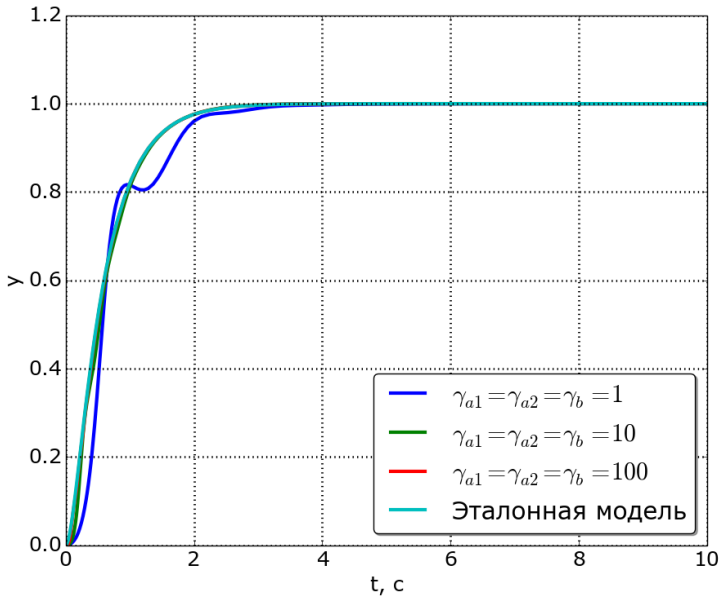

Рис. 2.3. Переходные процессы полученной системы при различных значениях $\gamma_{a1}, \gamma_{a2}, \gamma_b$ в сравнении с эталонной моделью

Программа лабораторной работы.

1. Построить и исследовать моделированием объект управления (2.1) с заданными параметрами $\mathbf{A}_0, \mathbf{b}_0$ для устойчивого случая.
2. Построить эталонную модель (2.2) и сравнить ее динамику с динамикой объекта управления (2.1).
3. Построить адаптивное управление (1.2) с настройками (1.4) и начальными значениями $\mathbf{\Gamma}_{\text{А}}, \gamma_{\text{В}}, \mathbf{P}$ по своему варианту.
4. Исследовать адаптивную систему, изменяя коэффициенты $\mathbf{\Gamma}_A, \gamma_B$ ($\mathbf{P} = \text{const}$).
5. Записывать установившиеся значения для $\mathbf{k}_A \to \mathbf{k}_{\text{А}}^{\text{уст}}$ и $k_B \to k_{\text{В}}^{\text{уст}}$ и сравнивать их со значениями, полученными из формул $\mathbf{k}_{\text{А}}^* = \mathbf{b}_0^+(\mathbf{A}_{\text{М}} - \mathbf{A}_0)$, $k_{\text{В}}^* = \mathbf{b}_0^+(\mathbf{b}_{\text{М}} - \mathbf{b}_0)$, где $\mathbf{b}_0^+ = (\mathbf{b}_0^T\mathbf{b}_0)^{-1}\mathbf{b}_0^T$ – псевдообратная матрица.
6. Построить и исследовать моделированием объект (2.1) с заданными параметрами $\mathbf{A}_0, \mathbf{b}_0$ для неустойчивого случая.
7. Повторить пп. 3-5 для случая неустойчивого объекта.

Вариант №1

Второй порядок

Устойчивый объект

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 \\ -2 & -4 \end{bmatrix}, \mathbf{b}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ \mathbf{A}_{\text{М}} = \begin{bmatrix} 0 & 1 \\ -10 & -16 \end{bmatrix}, \mathbf{b}_{\text{М}} = \begin{bmatrix} 0 \\ 16 \end{bmatrix}, \ \mathbf{P} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}, \ \mathbf{\Gamma}_A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \gamma_B = 1$$

Неустойчивый объект

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 \\ 2 & 4 \end{bmatrix}, \mathbf{b}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ \mathbf{A}_{\text{М}} = \begin{bmatrix} 0 & 1 \\ -10 & -16 \end{bmatrix}, \mathbf{b}_{\text{М}} = \begin{bmatrix} 0 \\ 16 \end{bmatrix}, \ \mathbf{P} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}, \ \mathbf{\Gamma}_A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \gamma_B = 1$$

Вариант №2

Второй порядок

Устойчивый объект

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 \\ -10 & -3 \end{bmatrix}, \mathbf{b}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ \mathbf{A}_{\text{М}} = \begin{bmatrix} 0 & 1 \\ -5 & -7 \end{bmatrix}, \mathbf{b}_{\text{М}} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \ \mathbf{P} = \begin{bmatrix} 2 & 0.1 \\ 0.1 & 0.1 \end{bmatrix}, \ \mathbf{\Gamma}_A = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \gamma_B = 2$$

Неустойчивый объект

$$\mathbf{A}_0=\begin{bmatrix}0 & 1\\ 10 & 3\end{bmatrix},\mathbf{b}_0=\begin{bmatrix}0\\ 1\end{bmatrix},\ \mathbf{A}_{\text{М}}=\begin{bmatrix}0 & 1\\ -5 & -7\end{bmatrix},\mathbf{b}_{\text{М}}=\begin{bmatrix}0\\ 2\end{bmatrix},\ \mathbf{P}=\begin{bmatrix}2 & 0.1\\ 0.1 & 0.1\end{bmatrix},\ \mathbf{\Gamma}_A=\begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix},\gamma_B=2$$

Вариант №3

Второй порядок

Устойчивый объект

$$\mathbf{A}_0=\begin{bmatrix}0 & 1\\ -2 & -0.5\end{bmatrix},\mathbf{b}_0=\begin{bmatrix}0\\ 1\end{bmatrix},\ \mathbf{A}_{\text{М}}=\begin{bmatrix}0 & 1\\ -3 & -5\end{bmatrix},\mathbf{b}_{\text{М}}=\begin{bmatrix}0\\ 5\end{bmatrix},\ \mathbf{P}=\begin{bmatrix}2 & 0.5\\ 0.5 & 0.5\end{bmatrix},\ \mathbf{\Gamma}_A=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\gamma_B=1$$

Неустойчивый объект

$$\mathbf{A}_0=\begin{bmatrix}0 & 1\\ 2 & 0.5\end{bmatrix},\mathbf{b}_0=\begin{bmatrix}0\\ 1\end{bmatrix},\ \mathbf{A}_{\text{М}}=\begin{bmatrix}0 & 1\\ -3 & -5\end{bmatrix},\mathbf{b}_{\text{М}}=\begin{bmatrix}0\\ 5\end{bmatrix},\ \mathbf{P}=\begin{bmatrix}2 & 0.5\\ 0.5 & 0.5\end{bmatrix},\ \mathbf{\Gamma}_A=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\gamma_B=1$$

Вариант №4

Второй порядок

Устойчивый объект

$$\mathbf{A}_0=\begin{bmatrix}0 & 1\\ -0.3 & -0.8\end{bmatrix},\mathbf{b}_0=\begin{bmatrix}0\\ 1\end{bmatrix},\ \mathbf{A}_{\text{М}}=\begin{bmatrix}0 & 1\\ -15 & -25\end{bmatrix},\mathbf{b}_{\text{М}}=\begin{bmatrix}0\\ 40\end{bmatrix},\ \mathbf{P}=\begin{bmatrix}20 & 10\\ 10 & 10\end{bmatrix},$$
$$\mathbf{\Gamma}_A=\begin{bmatrix}10 & 0\\ 0 & 10\end{bmatrix},\gamma_B=10$$

Неустойчивый объект

$$\mathbf{A}_0=\begin{bmatrix}0 & 1\\ 0.3 & 0.8\end{bmatrix},\mathbf{b}_0=\begin{bmatrix}0\\ 1\end{bmatrix},\ \mathbf{A}_{\text{М}}=\begin{bmatrix}0 & 1\\ -15 & -25\end{bmatrix},\mathbf{b}_{\text{М}}=\begin{bmatrix}0\\ 40\end{bmatrix},\ \mathbf{P}=\begin{bmatrix}20 & 10\\ 10 & 10\end{bmatrix},$$
$$\mathbf{\Gamma}_A=\begin{bmatrix}10 & 0\\ 0 & 10\end{bmatrix},\gamma_B=10$$

Вариант №5

Второй порядок

Устойчивый объект

$$\mathbf{A}_0=\begin{bmatrix}0 & 1\\ -3 & -1\end{bmatrix},\mathbf{b}_0=\begin{bmatrix}0\\ 1\end{bmatrix},\ \mathbf{A}_{\text{М}}=\begin{bmatrix}0 & 1\\ -8 & -16\end{bmatrix},\mathbf{b}_{\text{М}}=\begin{bmatrix}0\\ 16\end{bmatrix},\ \mathbf{P}=\begin{bmatrix}2 & 1\\ 1 & 1\end{bmatrix},\ \mathbf{\Gamma}_A=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\gamma_B=1,$$

Неустойчивый объект

$$\mathbf{A}_0=\begin{bmatrix}0 & 1\\ 3 & 1\end{bmatrix},\mathbf{b}_0=\begin{bmatrix}0\\ 1\end{bmatrix},\ \mathbf{A}_{\text{М}}=\begin{bmatrix}0 & 1\\ -8 & -16\end{bmatrix},\mathbf{b}_{\text{М}}=\begin{bmatrix}0\\ 16\end{bmatrix},\ \mathbf{P}=\begin{bmatrix}2 & 1\\ 1 & 1\end{bmatrix},\ \mathbf{\Gamma}_A=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\gamma_B=1,$$

Вариант №6

Второй порядок

Устойчивый объект

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 \\ -5 & -2 \end{bmatrix}, \mathbf{b}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \mathbf{A}_\text{М} = \begin{bmatrix} 0 & 1 \\ -8 & -32 \end{bmatrix}, \mathbf{b}_\text{М} = \begin{bmatrix} 0 \\ 16 \end{bmatrix},\ \mathbf{P} = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.1 \end{bmatrix},$$

$$\mathbf{\Gamma}_A = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \gamma_B = 0.1$$

Неустойчивый объект

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 \\ 5 & 2 \end{bmatrix}, \mathbf{b}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \mathbf{A}_\text{М} = \begin{bmatrix} 0 & 1 \\ -8 & -32 \end{bmatrix}, \mathbf{b}_\text{М} = \begin{bmatrix} 0 \\ 16 \end{bmatrix},\ \mathbf{P} = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.1 \end{bmatrix},\ \mathbf{\Gamma}_A = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \gamma_B = 0.1$$

,

Вариант №7

Второй порядок

Устойчивый объект

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 \\ -3 & -1 \end{bmatrix}, \mathbf{b}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \mathbf{A}_\text{М} = \begin{bmatrix} 0 & 1 \\ -8 & -16 \end{bmatrix}, \mathbf{b}_\text{М} = \begin{bmatrix} 0 \\ 16 \end{bmatrix},\ \mathbf{P} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix},\ \mathbf{\Gamma}_A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \gamma_B = 1,$$

Неустойчивый объект

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 \\ 2 & 0.5 \end{bmatrix}, \mathbf{b}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \mathbf{A}_\text{М} = \begin{bmatrix} 0 & 1 \\ -8 & -16 \end{bmatrix}, \mathbf{b}_\text{М} = \begin{bmatrix} 0 \\ 16 \end{bmatrix},\ \mathbf{P} = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 0.5 \end{bmatrix},\ \mathbf{\Gamma}_A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \gamma_B = 1$$

Вариант №8

Второй порядок

Устойчивый объект

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 \\ -10 & -3 \end{bmatrix}, \mathbf{b}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \mathbf{A}_\text{М} = \begin{bmatrix} 0 & 1 \\ -5 & -7 \end{bmatrix}, \mathbf{b}_\text{М} = \begin{bmatrix} 0 \\ 2 \end{bmatrix},\ \mathbf{P} = \begin{bmatrix} 2 & 0.1 \\ 0.1 & 0.1 \end{bmatrix},\ \mathbf{\Gamma}_A = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \gamma_B = 2$$

Неустойчивый объект

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 \\ 2 & 0.5 \end{bmatrix}, \mathbf{b}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \mathbf{A}_\text{М} = \begin{bmatrix} 0 & 1 \\ -5 & -7 \end{bmatrix}, \mathbf{b}_\text{М} = \begin{bmatrix} 0 \\ 2 \end{bmatrix},\ \mathbf{P} = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 0.5 \end{bmatrix},\ \mathbf{\Gamma}_A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \gamma_B = 1$$

Вариант №9

Второй порядок

Устойчивый объект

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 \\ -0.3 & -0.8 \end{bmatrix}, \mathbf{b}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \mathbf{A}_\text{М} = \begin{bmatrix} 0 & 1 \\ -15 & -25 \end{bmatrix}, \mathbf{b}_\text{М} = \begin{bmatrix} 0 \\ 40 \end{bmatrix},\ \mathbf{P} = \begin{bmatrix} 20 & 10 \\ 10 & 10 \end{bmatrix},$$
$$\mathbf{\Gamma}_A = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \gamma_B = 10$$

Неустойчивый объект

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 \\ 2 & 4 \end{bmatrix}, \mathbf{b}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \mathbf{A}_\text{М} = \begin{bmatrix} 0 & 1 \\ -10 & -16 \end{bmatrix}, \mathbf{b}_\text{М} = \begin{bmatrix} 0 \\ 16 \end{bmatrix},\ \mathbf{P} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix},\ \mathbf{\Gamma}_A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \gamma_B = 1$$

Вариант №10

Второй порядок

Устойчивый объект

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 \\ -2 & -4 \end{bmatrix}, \mathbf{b}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \mathbf{A}_\text{М} = \begin{bmatrix} 0 & 1 \\ -10 & -16 \end{bmatrix}, \mathbf{b}_\text{М} = \begin{bmatrix} 0 \\ 16 \end{bmatrix},\ \mathbf{P} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix},\ \mathbf{\Gamma}_A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \gamma_B = 1$$

Неустойчивый объект

$$\mathbf{A}_0 = \begin{bmatrix} 0 & 1 \\ 5 & 2 \end{bmatrix}, \mathbf{b}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \mathbf{A}_\text{М} = \begin{bmatrix} 0 & 1 \\ -8 & -32 \end{bmatrix}, \mathbf{b}_\text{М} = \begin{bmatrix} 0 \\ 16 \end{bmatrix},\ \mathbf{P} = \begin{bmatrix} 0.2 & 0.1 \\ 0.1 & 0.1 \end{bmatrix},\ \mathbf{\Gamma}_A = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \gamma_B = 0.1$$

Содержание отчёта.

- Цель работы.
- Теоретические сведения.
- Расчёт адаптивной системы управления для устойчивого объекта.
- Выводы о работе адаптивной системы с параметрической настройкой для управления устойчивым объектом.
- Расчёт адаптивной системы управления для неустойчивого объекта.
- Выводы о работе адаптивной системы с параметрической настройкой для управления неустойчивым объектом.

3. СИНТЕЗ И ИССЛЕДОВАНИЕ СИСТЕМ С МОДАЛЬНЫМ УПРАВЛЕНИЕМ И НАБЛЮДАТЕЛЕМ СОСТОЯНИЯ

Цель работы: овладение навыками исследования систем с модальным управлением и наблюдателем состояния, исследование эффективности модального управления и стационарного наблюдателя полного порядка при изменении параметров объекта уравнения.

Теоретические сведения. Модальное управление. Рассмотрим линейный стационарный объект порядка *n* вида (1.1). Если объект (1.1) является полностью управляемым, т.е. ранг его матрицы управляемости ($\mathbf{R}=[\mathbf{B}\ \ \mathbf{AB}\ \ \mathbf{A}^2\mathbf{B}\ ...\ \mathbf{A}^{n-1}\mathbf{B}]$) равен *n*, то в системе можно использовать модальное управление, имеющее вид полной линейной обратной связи по состоянию объекта управления:

$$u_l = \mathbf{k}^T\mathbf{x}, \tag{3.1}$$

где вещественный вектор $\mathbf{k}=\left(k_1\ k_2\ ...\ k_n\right)^{\mathrm{T}}$ коэффициентов обратных связей рассчитывается из условия обеспечения любого наперед заданного желаемого распределения всех корней характеристического уравнения замкнутой системы, с одним лишь ограничением: комплексные корни должны входить вместе со своими сопряженными значениями в силу вещественности элементов матриц $\mathbf{A}$, $\mathbf{b}$. Характеристический полином системы с модальным управлением (3.1) будет иметь следующий вид:

$$\varphi(\lambda)=\det(\mathbf{A}+\mathbf{bk}^T-\lambda\mathbf{E}), \tag{3.2}$$

где $\mathbf{E}$ – единичная матрица. Желаемый полином замкнутой системы обычно выбирается в виде стандартных полиномов Ньютона или Баттерворта, вид которых представлен в табл. 3.1.

Табл. 3.1

Порядок	Полином Ньютона	Полином Баттерворта
1	$\lambda+\omega_0$	$\lambda+\omega_0$
2	$\lambda^2+2\omega_0\lambda+\omega_0^2$	$\lambda^2+1.4\omega_0\lambda+\omega_0^2$
3	$\lambda^3+3\omega_0\lambda^2+3\omega_0^2\lambda+\omega_0^3$	$\lambda^3+2\omega_0\lambda^2+2\omega_0^2\lambda+\omega_0^3$
4	$\lambda^4+4\omega_0\lambda^3+6\omega_0^2\lambda^2+4\omega_0^3\lambda+\omega_0^4$	$\lambda^4+2.6\omega_0\lambda^3+3.4\omega_0^2\lambda^2+2.6\omega_0^3\lambda+\omega_0^4$

Все стандартные полиномы зависят от параметра ω_0. Этот параметр определяет радиус распределения корней характеристического полинома, поэтому его значение всегда положительно. Все корни полинома Ньютона располагаются в одной точке на отрицательной части вещественной оси комплексной плоскости корней, а корни полинома Баттерворта располагаются на полуокружности в левой полуплоскости комплексной плоскости корней.

Для расчёта значений вектора **k** необходимо приравнять коэффициенты при одинаковых степенях λ в характеристическом полиноме (3.2) и желаемом полиноме. Для простоты расчётов можно воспользоваться функцией пакета Matlab acker, имеющей следующую форму: `K = acker(A,B,P)`, где `K` – рассчитываемый вектор коэффициентов обратных связей, `A` и `B` – матрицы разомкнутой системы, а вектор `P` – вектор желаемых корней замкнутой системы.

При использовании модального управления, также требуется дополнительное введение нормирующего коэффициента $k_n = 1/(-\mathbf{c}/(\mathbf{A}-\mathbf{bk})\mathbf{b})$, где **c** – вектор выхода объекта (1.1) из его уравнения выхода: $\mathbf{y} = \mathbf{cx}$, а **y** – выход системы.

Стационарный наблюдатель состояния полного порядка. Для реализации модального управления (также как и для реализации системы с прямой адаптацией и параметрической настройкой) требуется полный вектор состояния системы, однако на практике не всегда имеется возможность измерения всех переменных состояния. Для получения полного вектора состояния можно использовать асимптотические оценки недостающих переменных состояния с помощью наблюдателя состояния. Если объект (1.1) является полностью наблюдаемым, т.е. ранг его матрицы наблюдаемости ($\mathbf{O} = [\mathbf{C} \;\; \mathbf{CA} \;\; \mathbf{CA}^2 \;\; ... \;\; \mathbf{CA}^{n-1}]^T$) равен *n*, то в системе можно использовать наблюдатель состояния полного порядка, имеющий следующий вид:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{b}u + \mathbf{l}(\mathbf{y} - \mathbf{c}\hat{\mathbf{x}}), \qquad (3.3)$$

где $\hat{\mathbf{x}}$ – оценка вектора состояния системы, а $\mathbf{l} = \left(l_1 \; l_2 \; ... \; l_n\right)^{\mathrm{T}}$ – некоторый вектор, обеспечивающий требуемый вид переходного процесса ошибки оценки вектора состояния системы.

Характеристический полином наблюдателя состояния будет иметь следующий вид:

$$\varphi(\lambda) = \det(\mathbf{A} + \mathbf{l}\mathbf{c}^T - \lambda\mathbf{E}). \qquad (3.4)$$

Так же как и для расчёта модального управления, выбирается желаемый полином наблюдателя состояния, и находятся коэффициенты вектора **l** путём приравнивания коэффициентов при одинаковых степенях λ в характеристическом полиноме (3.4) и желаемом полиноме.

Необходимо отметить, что исходя из независимости характеристических полиномов замкнутой системы с модальным управлением и наблюдателя состояния, мы можем назначить корни характеристических полиномов модального управления и наблюдателя состояния независимо друг от друга.

Пример расчёта модального управления и наблюдателя состояния полного порядка. Рассмотрим объект второго порядка имеющий следующий вид:

$$\begin{cases} \mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{b}\mathbf{u} \\ \mathbf{y} = \mathbf{c}\mathbf{x} \end{cases}, \qquad (3.5)$$

где $\mathbf{A} = \begin{bmatrix} 0 & a_1 \\ -a_2 & -a_2 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} 0 \\ a_2 \end{bmatrix}$ и $\mathbf{c} = \begin{bmatrix} 1 & 0 \end{bmatrix}$.

Возьмём для примера следующие значения параметров исходного объекта: $a_1 = 3,\ a_2 = 2$. Переходный процесс в исходном объекте представлен на рис. 3.1.

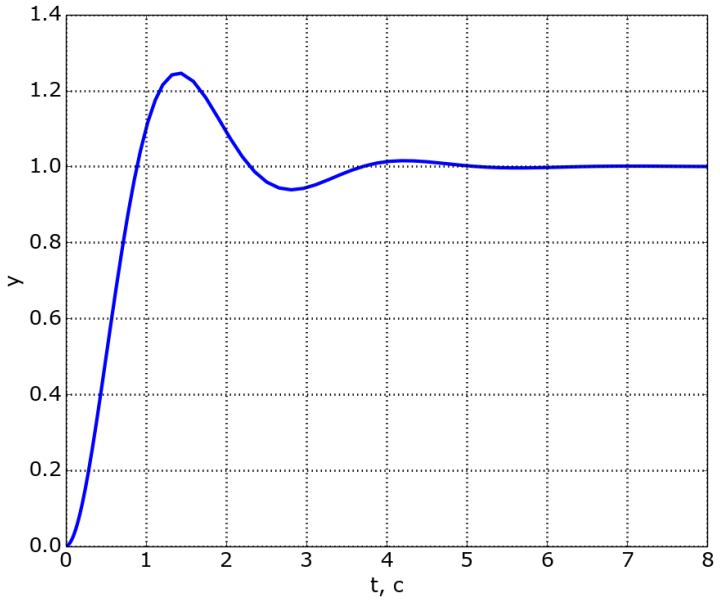

Рис. 3.1. Переходный процесс в исходной системе

Рассчитаем теперь коэффициенты модального управления, для этого возьмём в качестве желаемого характеристического полинома полином Ньютона с $\omega_0 = 4.5$. В этом случае вектор корней желаемого полинома будет иметь следующий вид $\mathbf{p} = \begin{bmatrix} -4.5 & -4.5 \end{bmatrix}$. Используя функцию acker рассчитаем коэффициенты модального управления: `k=acker(A,b,p)`. Коэффициенты модального управления имеют следующие значения: $\mathbf{k} = \begin{bmatrix} 2.375 & 3.5 \end{bmatrix}$, а нормирующий коэффициент $k_n = 3.375$.

Для расчёта наблюдателя состояния также выберем полином Ньютона. Динамика наблюдателя состояния должна быть примерно в 3-5 раз быстрее динамики системы с модальным управлением, поэтому параметр ω_0 возьмём равный 13.5. Используя функцию acker рассчитаем коэффициенты наблюдателя состояния: `l=acker(A',c',p)`. Коэффициенты наблюдателя состояния имеют следующие значения $\mathbf{l} = \begin{bmatrix} 25 & 42.0833 \end{bmatrix}$. На рисунке 3.2 представлена структурная схема полученной системы.

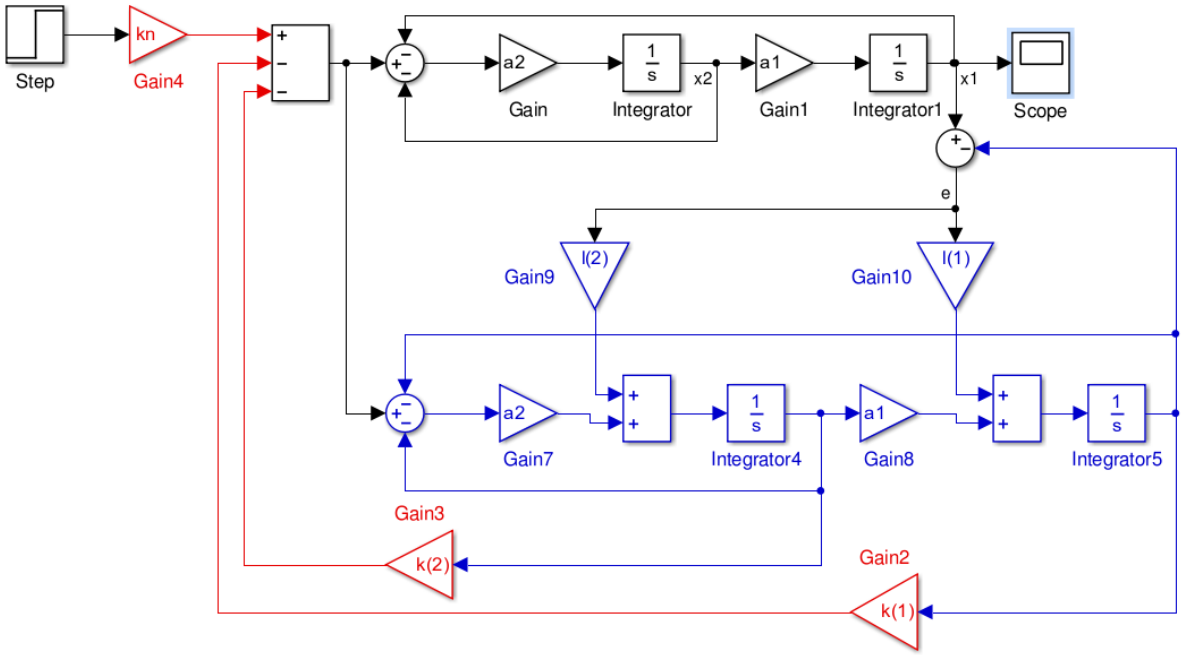

Рис. 3.2. Структурная схема системы с модальным управлением и наблюдателем состояния полного порядка

Переходный процесс в полученной системе при номинальных значениях параметров объекта управления, при увеличении параметра объекта управле-

ния a_1 в 3 раза и при уменьшении параметра объекта управления a_2 в 3 раза представлен на рис. 3.3.

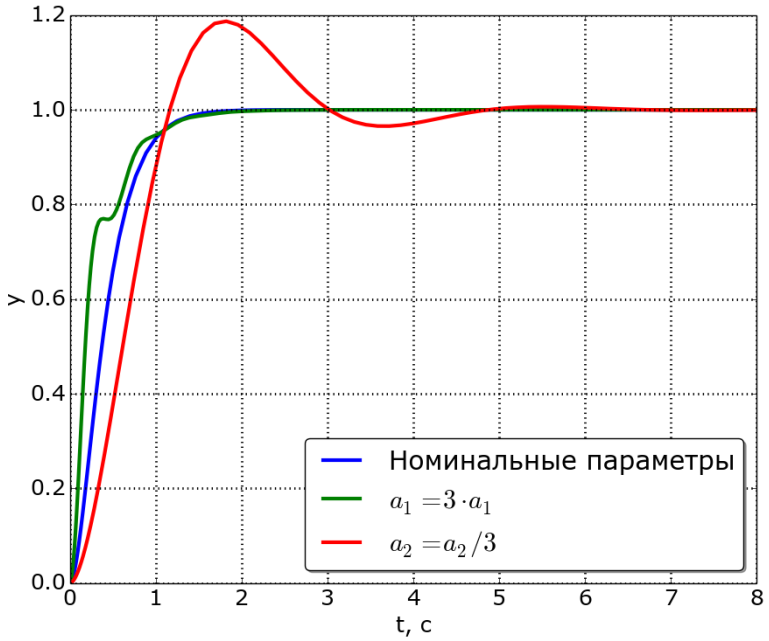

Рис. 3.3. Переходные процессы в системе с модальным управлением и наблюдателем состояния

Программа лабораторной работы:

1. Построить и исследовать моделированием объект управления (3.5) с заданными параметрами a_1 и a_2.
2. Построить модальное управление (3.1) и исследовать систему при замыкании обратных связей модального управления по переменным состояния объекта управления при номинальных параметрах и при изменении параметров a_1 и a_2 в 3 раза.
3. Построить наблюдатель состояния полного порядка (3.3) и исследовать систему с модальным управлением и наблюдателем состояния при номинальных параметрах и при изменении параметров a_1 и a_2 в 3 раза.
4. Исследовать систему с модальным управлением и наблюдателем состояния полного порядка при номинальных параметрах объекта управления и наличии отклонений начальных значений переменных состояния от нулевых значений в диапазоне $\lceil -0.5 \quad 0.5 \rceil$.

Вариант №1

$$\mathbf{A}=\begin{bmatrix}0 & 5\\ -1 & -1\end{bmatrix},\mathbf{b}=\begin{bmatrix}0\\ 1\end{bmatrix},\mathbf{c}=\begin{bmatrix}1 & 0\end{bmatrix}$$

Вариант №2

$$\mathbf{A}=\begin{bmatrix}0 & 2\\ -3 & -3\end{bmatrix},\mathbf{b}=\begin{bmatrix}0\\ 3\end{bmatrix},\mathbf{c}=\begin{bmatrix}1 & 0\end{bmatrix}$$

Вариант №3

$$\mathbf{A}=\begin{bmatrix}0 & 2\\ -1 & -1\end{bmatrix},\mathbf{b}=\begin{bmatrix}0\\ 1\end{bmatrix},\mathbf{c}=\begin{bmatrix}1 & 0\end{bmatrix}$$

Вариант №4

$$\mathbf{A}=\begin{bmatrix}0 & 2\\ -4 & -4\end{bmatrix},\mathbf{b}=\begin{bmatrix}0\\ 4\end{bmatrix},\mathbf{c}=\begin{bmatrix}1 & 0\end{bmatrix}$$

Вариант №5

$$\mathbf{A}=\begin{bmatrix}0 & 8\\ -4 & -4\end{bmatrix},\mathbf{b}=\begin{bmatrix}0\\ 4\end{bmatrix},\mathbf{c}=\begin{bmatrix}1 & 0\end{bmatrix}$$

Вариант №6

$$\mathbf{A}=\begin{bmatrix}0 & 6\\ -4 & -4\end{bmatrix},\mathbf{b}=\begin{bmatrix}0\\ 4\end{bmatrix},\mathbf{c}=\begin{bmatrix}1 & 0\end{bmatrix}$$

Вариант №7

$$\mathbf{A}=\begin{bmatrix}0 & 8\\ -1 & -1\end{bmatrix},\mathbf{b}=\begin{bmatrix}0\\ 1\end{bmatrix},\mathbf{c}=\begin{bmatrix}1 & 0\end{bmatrix}$$

Вариант №8

$$\mathbf{A}=\begin{bmatrix}0 & 6\\ -3 & -3\end{bmatrix},\mathbf{b}=\begin{bmatrix}0\\ 3\end{bmatrix},\mathbf{c}=\begin{bmatrix}1 & 0\end{bmatrix}$$

Содержание отчёта.

- Цель работы.
- Теоретические сведения.
- Расчёт системы с модальным управлением и наблюдателем состояния.
- Исследование полученной системы при изменении параметров объекта управления и его начального состояния.
- Выводы о работе системы с модальным управлением и наблюдателем состояния.

4. СИНТЕЗ И ИССЛЕДОВАНИЕ АДАПТИВНОЙ СИСТЕМЫ УПРАВЛЕНИЯ ДЛЯ ДВУХМАССОВОГО УПРУГОГО ЭЛЕКТРОМЕХАНИЧЕСКОГО ОБЪЕКТА

Цель работы: овладение навыками исследования электромеханических систем с адаптивно-модальным управлением, исследование эффективности адаптивно-модального управления при изменении параметров объекта уравнения.

Теоретические сведения. Линеаризованная математическая модель двухмассового упругого электромеханического объекта. Рассмотрим линеаризованную математическую модель двухмассового упругого электромеханического объекта с двигателем постоянного тока с постоянными магнитами без учёта сил трения и зазора в трансмиссии и с управлением по скорости вращения второй массы:

$$\begin{cases} \omega_2 = \dfrac{1}{J_2} m_y, \\ m_y = p(\omega_1 - \omega_2), \\ \omega_1 = \dfrac{k_m}{J_1} i - \dfrac{1}{J_1} m_y, \\ \dot{i} = \dfrac{1}{L}(u_\Sigma - k_e \omega_1 - Ri), \\ u_\Sigma = u_0 + u_l + u_a, \end{cases} \tag{4.1}$$

где ω_2 – скорость вращения рабочего органа (второй массы), J_2 – момент инерции рабочего органа, m_y – упругий момент, возникающий в упругой связи при её деформации, p – коэффициент упругости связи, ω_1 – скорость вращения вала двигателя постоянного тока (первая масса), k_m и k_e – постоянные коэффициенты, определяемые конструкцией электрической машины, J_1 – момент инерции двигателя постоянного тока, i – ток якоря двигателя постоянного тока, L – индуктивность обмотки якоря двигателя постоянного тока, R – активное сопротивление обмотки якоря двигателя постоянного тока, u_Σ –

скалярный сигнал управления, u_0 – программное управление, u_l – линейное (модальное) управление, u_a – адаптивное управления.

Уравнения системы с прямой адаптацией и параметрической настройкой представлены в лабораторной работе №1, а уравнения системы с модальным управлением в лабораторной работе №3.

Пример расчёта адпативно-модального управления для двухмассового упругого электромеханического объекта. Для примера возьмём следующие параметры объекта: $R = 2$ Ом, $k_e = 1$ В·с / рад, $k_m = 1$ Н·м / А, $J_1 = 0.05$ кг·м^2, $J_2 = 0.1$ кг·м^2, $L = 0.01$ Гн, $p = 0.8$ Н·м / рад. На рис. 4.1 представлен переходный процесс в исходной системе.

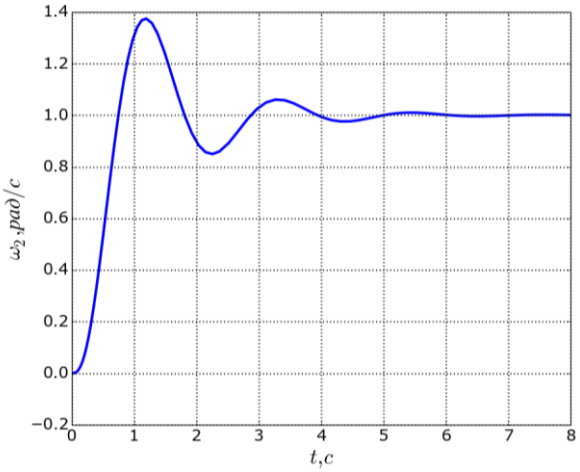

Рис. 4.1. Переходный процесс в исходной системе

Добавим в систему модальное управление. Исходный объект в пространстве состояний имеет следующую форму:

$$\dot{} \begin{bmatrix} \omega_2 \\ m_y \\ \omega_1 \\ i \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{J_2} & 0 & 0 \\ -p & 0 & p & 0 \\ 0 & -\frac{1}{J_1} & 0 & \frac{k_m}{J_1} \\ 0 & 0 & -\frac{k_e}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \omega_2 \\ m_y \\ \omega_1 \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} u_\Sigma$$

используя функцию **acker** рассчитаем вектор обратных связей модального управления. Для этого выберем в качестве желаемого полинома полином Ньютона с $\omega_0 = 7.5$. В этом случае вектор коэффициентов модального управ-

ления будет иметь следующий вид $\mathbf{k} = [0.041 \;\; 0.6047 \;\; -0.8433 \;\; -1.7]$, а нормирующий коэффициент $k_n = 0.1978$. На рис. 4.2 представлены графики переходных процессов исходной системы и системы с модальным управлением.

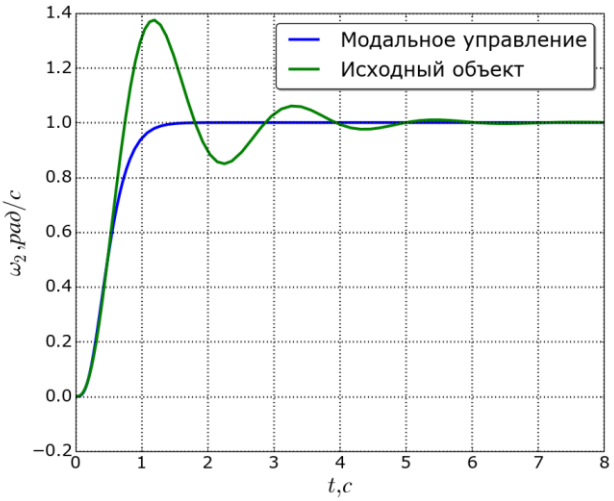

Рис. 4.2. Переходные процессы в исходной системе и системе с модальным управлением

Как видно из полученных графиков применение модального управления позволяет добиться желаемого переходного процесса в системе.

Рассмотрим теперь поведение системы при изменении параметров объекта управления. На рис. 4.3 представлены переходные процессы в системе при уменьшении момента инерции рабочего органа и коэффициента упругости связи в 3 раза.

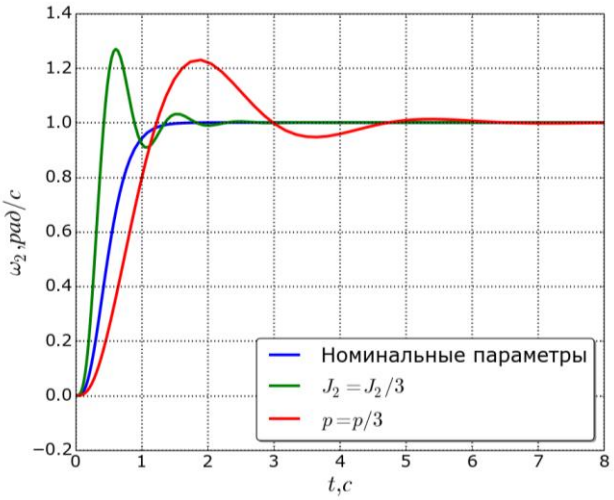

Рис. 4.3. Переходные процессы в системе с модальным управлением при изменении параметров объекта управления

Добавим теперь в систему адаптивное управление с параметрической настройкой и эталонной моделью. В качестве эталонной модели возьмём модель объекта с номинальными параметрами и модальным управлением. Для расчёта матрицы $\mathbf{P}$ воспользуемся уравнением Ляпунова (1.5), где в качестве произвольной симметричной положительно определённой матрицы $\mathbf{Q}$ возьмём следующие матрицы: $\mathbf{Q}_1 = \mathrm{diag}[1,1,1,1]$, $\mathbf{Q}_2 = \mathrm{diag}[100,10,10,100]$, $\mathbf{Q}_3 = \mathrm{diag}[1000,1,1,1]$. Для решения уравнения Ляпунова воспользуемся функцией lyap из пакета Matlab, имеющей следующий вид: `P = lyap(Am', Q)`, где `P` – искомая матрица $\mathbf{P}$, `Am'` – транспонированная матрица эталонной модели ($\mathbf{A}_\mathrm{M} = \mathbf{A} - \mathbf{bk}$). В результате применения функции lyap мы получим соответствующие квадратные, симметричные, положительно определённые матрицы $\mathbf{P}_1$, $\mathbf{P}_2$ и $\mathbf{P}_3$. Также возьмём $\mathbf{\Gamma}_\mathrm{A} = \mathrm{diag}([100,100,100,100])$, а $\gamma_\mathrm{B} = 100$. На рисунке 4.4 представлена структурная схема полученной системы с адаптивно-модальным управлением и эталонной моделью.

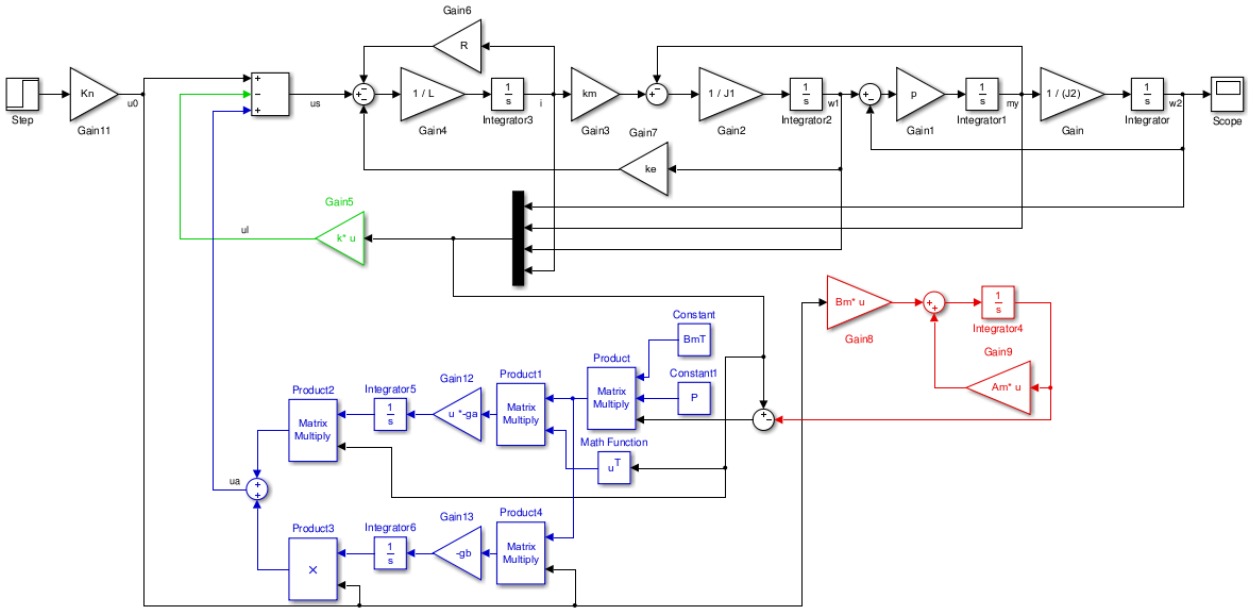

Рис. 4.4. Структурная схема системы с адаптивно-модальным управлением и эталонной моделью

Графики переходных процессов при уменьшении момента инерции рабочего органа в 3 раза в системе с адаптивно-модальным управлением представлены на рис. 4.5.

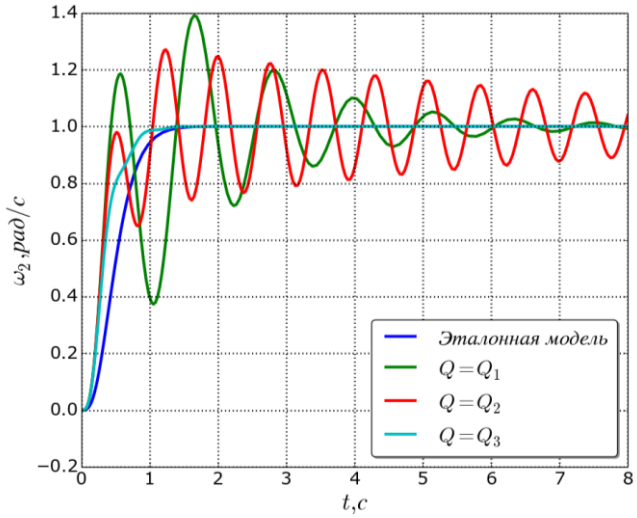

Рис. 4.5. Переходные процессы в системе с адаптивно-модальным управлением при уменьшении момента инерции рабочего органа

Графики переходных процессов при уменьшении коэффициента упругости связи в 3 раза в системе с адаптивно-модальным управлением представлены на рис. 4.6.

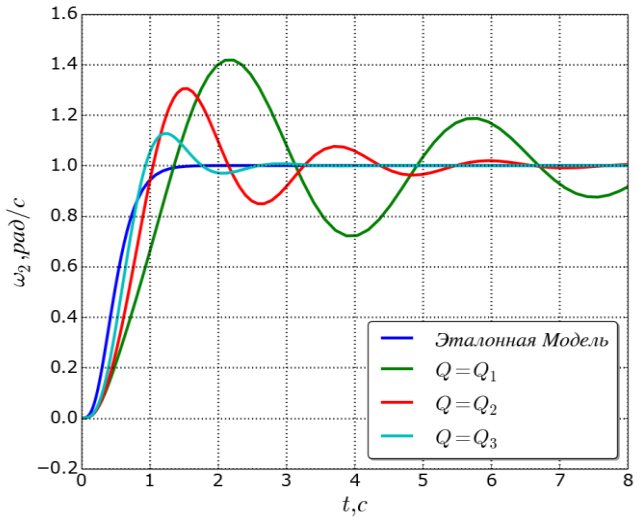

Рис. 4.6. Переходные процессы в системе с адаптивно-модальным управлением при уменьшении коэффициента упругости связи

Как видно из полученных графиков форма переходного процесса сильно зависит от выбранной матрицы $\mathbf{Q}$. Для нашего примера, при трёхкратном

изменении параметров объекта наилучшие переходные процессы позволила получить матрица $\mathbf{Q}_3$.

Программа лабораторной работы.

1. Построить и исследовать моделированием объект управления (4.1) с заданными параметрами p и J_2.
2. Построить модальное управление (3.1) и исследовать систему при замыкании обратных связей модального управления по переменным состояния объекта управления при номинальных параметрах и при изменении параметров p и J_2 в 3 раза.
3. Добавить адаптивное управление (1.2) с настройками (1.4) при использовании различных матриц $\mathbf{Q}$ и исследовать систему при номинальных параметрах и при изменении параметров p и J_2 в 3 раза.
4. Оценить установившиеся значения настраиваемых коэффициентов $\mathbf{k}_\text{A}$ и k_B при использовании адаптивного и адаптивно-модального управлений.

Вариант №1
$p = 0.5\text{Нм/рад}$, $J_2 = 0.1\text{кгм}^2$

Вариант №2
$p = 0.8\text{Нм/рад}$, $J_2 = 0.2\text{кгм}^2$

Вариант №3
$p = 0.6\text{Нм/рад}$, $J_2 = 0.15\text{кгм}^2$

Вариант №4
$p = 0.4\text{Нм/рад}$, $J_2 = 0.08\text{кгм}^2$

Вариант №5
$p = 0.7\text{Нм/рад}$, $J_2 = 0.12\text{кгм}^2$

Вариант №6
$p = 0.8\text{Нм/рад}$, $J_2 = 0.08\text{кгм}^2$

Вариант №7
$p = 1\text{Нм/рад}$, $J_2 = 0.3\text{кгм}^2$

Вариант №8
$p = 0.3\text{Нм/рад}$, $J_2 = 0.05\text{кгм}^2$

Содержание отчёта.

- Цель работы.
- Теоретические сведения.
- Расчёт системы с модальным управлением.
- Исследование полученной системы при изменении параметров объекта управления.
- Расчёт адаптивного управления.
- Исследование полученной системы с адаптивно-модальным управлением при изменении параметров объекта управления.
- Выводы о работе системы с модальным и адаптивно-модальным управлениями.

5. ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

Цель работы: изучение основ генетических алгоритмов, овладение навыками применения генетических алгоритмов в пакете Matlab.

Теоретические сведения. Генетические алгоритмы. Генетические алгоритмы относятся к стохастическим методам поиска и оптимизации. Они основаны на принципах естественного отбора и эволюции. Генетические алгоритмы работают с популяциями закодированных специальным образом потенциальных решений поставленной задачи и применяют к ним принципы выживания сильнейших особей, т.е. наиболее подходящих решений. Кодирование происходит в виде хромосом, представляющих собой цепочки генов. То, насколько подходит данное конкретное решение для поставленной задачи определяется с помощью оценочной функции (функции приспособленности), применяемой к каждой особи в популяции. Задача генетических алгоритмов состоит в том, чтобы найти решение, достигающее экстремума оценочной функции. Генетические алгоритмы относятся к итерационным алгоритмам, на каждой итерации которых создаётся новая популяция особей посредствам выбора наиболее приспособленных и применения к ним генетических операторов, заимствованных из генетики. Такой процесс ведёт к эволюции потенциальных решений и увеличению среднего значения приспособленности популяции. В сравнении с градиентными методами поиска, такими как метод градиентного спуска, метод Ньютона и алгоритм Левенберга-Марквардта, можно выделить следующие свойства генетических алгоритмов:

1. В генетических алгоритмах работа ведётся не с конкретными параметрами задачи оптимизации, а с их закодированными (чаще всего в виде двоичной последовательности) формами;
2. Поиск происходит не из единственной точки, а из некоторой популяции;
3. Используется только целевая функция, а не её производные или любая другая дополнительная информация;
4. Используются вероятностные, а не детерминированные правила выбора.

Генетические алгоритмы относятся к классу глобальных методов поиска и, ввиду того что используется только целевая функция, могут быть применены к задачам, которые не поддаются строгому и формальному анализу. Генетический алгоритм состоит из следующих этапов:

1. инициализация;
2. оценка приспособленности особей;
3. проверка условий остановки;
4. селекция особей;
5. применение генетических операторов;
6. формирование новой популяции и возврат на этап 2.

Рассмотрим подробнее работу генетического алгоритма на каждом этапе. Для этого введём необходимые понятия, заимствованные из генетики.

Популяция – конечное множество особей.

Ген – элемент хромосомы, представляющий отдельное свойство особи.

Оценочная функция – математическая функция, представляющая меру приспособленности особи в популяции.

Для использования генетического алгоритма необходимо задать способ кодирования решений проблемы в виде хромосомы. Для кодирования отдельных генов в хромосоме чаще всего используются либо двоичные числа (биты), либо вещественные числа.

На этапе инициализации происходит формирование исходной популяции в виде случайного набора особей, представляющих собой последовательности генов фиксированной длины (хромосом).

На этапе оценки приспособленности особей происходит расчёт приспособленности каждой особи в популяции с помощью оценочной функции.

На этапе проверки условий остановки проверяются критерии завершения алгоритма. Наиболее часто встречаемыми критериями являются:

- достижение функции приспособленности определённого, заранее заданного значения;
- сохранение наилучшего значения оценочной функции в популяции на одном уровне в течение определённого количества поколений;
- выполнение определённого количества итераций алгоритма.

На этапе селекции особей на основании рассчитанных значений функций приспособленности, выбираются особи для участия в создании следую-

щей популяции. Выбор производится согласно принципам естественного отбора, по которым наибольшие шансы на создание новых особей имеют особи с наилучшими значениями функции приспособленности. В результате процесса селекции образуется родительская популяция (родительский пул), из которой отбираются родительские пары. После чего из родительской популяции случайным образом формируются родительские пары.

На этапе применения генетических операторов к выбранным на предыдущем этапе родительским парам формируется новая популяция потомков. В генетических алгоритмах применяются два основных генетических оператора: оператор скрещивания и оператор мутации. При этом обычно оператор мутации играет менее важную роль по сравнению с оператором скрещивания, поэтому скрещивание происходит практически на каждой итерации алгоритма, а мутация достаточно редко. На этапе скрещивания из полученных родительских пар получаются пары потомков с помощью оператора скрещивания. Операция мутации может применяться как к родителям до скрещивания, так и к потомкам, и заключается в случайном изменении хромосом особей.

Оператор мутации используется для выведения популяции из области локального экстремума и предотвращения преждевременного схождения алгоритма. На этапе формирования новой популяции из родительского пула и потомков отбираются особи для формирования новой популяции.

В генетических алгоритмах обеспечивается возрастание среднего значения приспособленности популяции, а благодаря большому количеству особей и оператору мутации обеспечивается большая область поиска. Благодаря этим свойствам генетические алгоритмы нашли применения во многих областях науки и промышленности, включая и системы автоматического управления. В области систем управления генетические алгоритмы применяются в задачах автоматического управления, идентификации и анализе устойчивости. В некоторых случаях генетические алгоритмы применяются в качестве единственного средства разработки, в других случаях генетические алгоритмы комбинируются с аналитическими и интеллектуальными методами теории управления.

Пример использования генетических алгоритмов в пакете Matlab для поиска минимума функции. Рассмотрим в качестве примера поиск минимума нелинейной функции вида:

$$f(x,y)=\sin(y)e^{(1-\cos x)^2}+\cos(x)e^{(1-\sin y)^2}+(x-y)^2 \qquad (5.1)$$

и имеющей следующие ограничения на значения x и y:

$$\begin{aligned} -10 \le x \le 0 \\ -6.5 \le y \le 0 \end{aligned}. \qquad (5.2)$$

График поверхности рассматриваемой функции представлен на рис. 5.1.

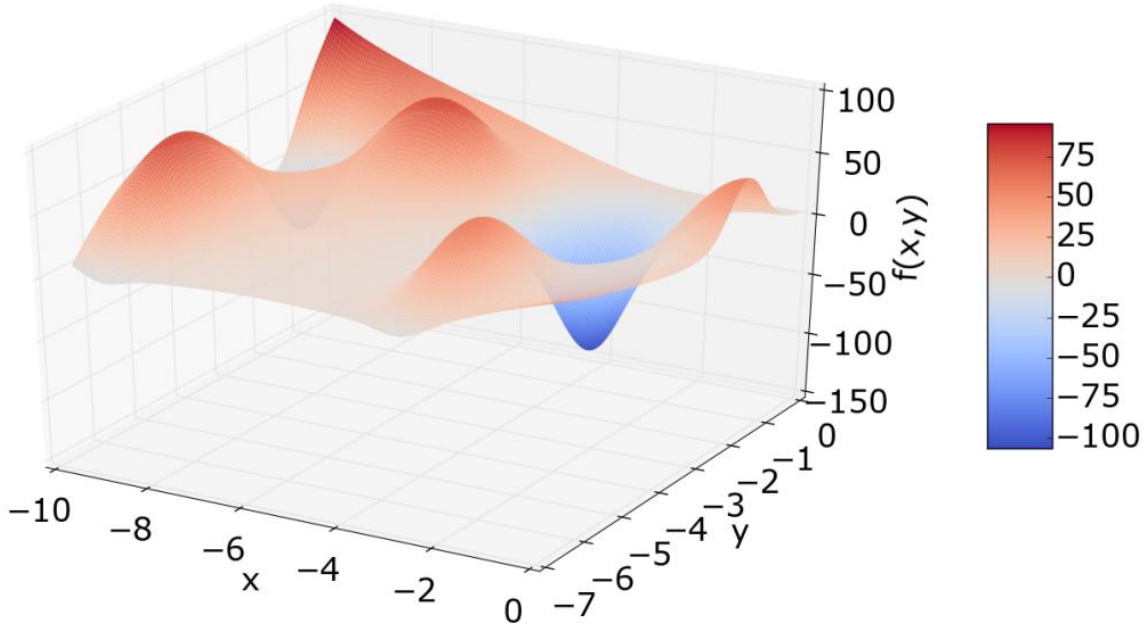

Рис. 5.1. График поверхности функции $f(x,y)$

Воспользуемся пакетом Matlab для поиска минимума функции с помощью генетических алгоритмов. Для этого сначала необходимо написать оценочную функцию генетического алгоритма, значение которой требуется минимизировать. Для этого нужно создать новый **.m** файл, содержащий следующую функцию, вычисляющую значение функции (5.1):

```
function y = test_fitness(x)
y = sin(x(2)) * exp((1 - cos(x(1))) ^ 2) + cos(x(1)) *
exp((1 - sin(x(2))) ^ 2) + (x(1) - x(2)) ^ 2;
```

Данная функция должна вычислять скалярное значение и принимать в качестве аргумента массив с параметрами функции.

Далее для наглядности воспользуемся графическим интерфейсом для работы с генетическими алгоритмами. Для его запуска в области Command Window наберём команду **`optimtool('ga')`**, после чего откроется окно для работы с оптимизационными методами, внешний вид которого представлен на рис. 5.2.

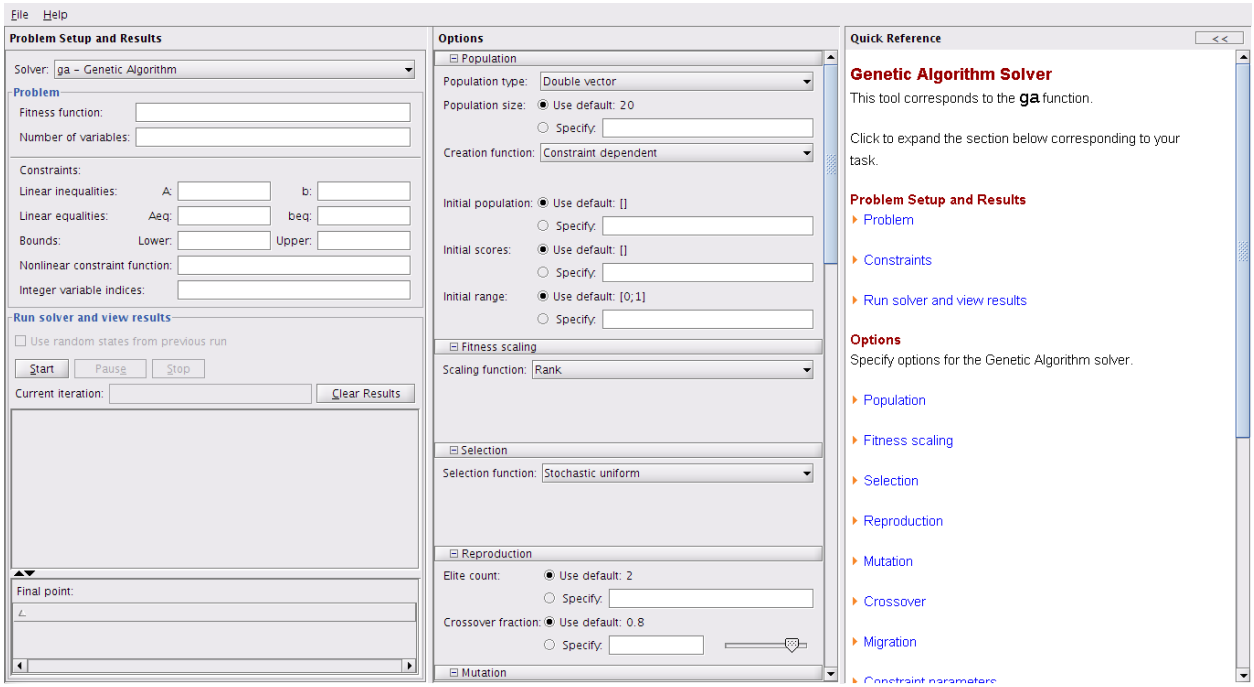

Рис. 5.2. Окно для работы с оптимизационными методами

В данном окне имеется 3 секции: 1. Описание задачи оптимизации, 2. Параметры оптимизации, 3. Краткая справка по параметрам алгоритма оптимизации. В поле “Fitness function” необходимо задать ссылку на нашу функцию для расчёта оценочной функции генетического алгоритма (в нашем примере **`@test_fitness`**), в поле “Number of variables” необходимо указать количество переменных, по которым происходит оптимизация (в нашем примере 2). В поля “Bounds” “Lower:” и “Upper:” необходимо внести ограничения на значения переменных, по которым производится оптимизация (в нашем примере **`[-10 -6.5]`** и **`[0 0]`**). Для начала не будем менять параметры генетического алгоритма, а только лишь включим отображение графика изменения лучшего и среднего значения оценочной функции генетического алгоритма (чекбокс “Best fitness” в графе параметров “Plot functions”) и попробуем запустить процесс оптимизации. Полученный график процесса оптимизации показан на рис. 5.3. (Следует помнить, что в виду стохастической при-

роды генетических алгоритмов, результат оптимизации будет отличаться от запуска к запуску, поэтому рекомендуется запускать оптимизацию несколько раз с целью выбора наилучшего результата)

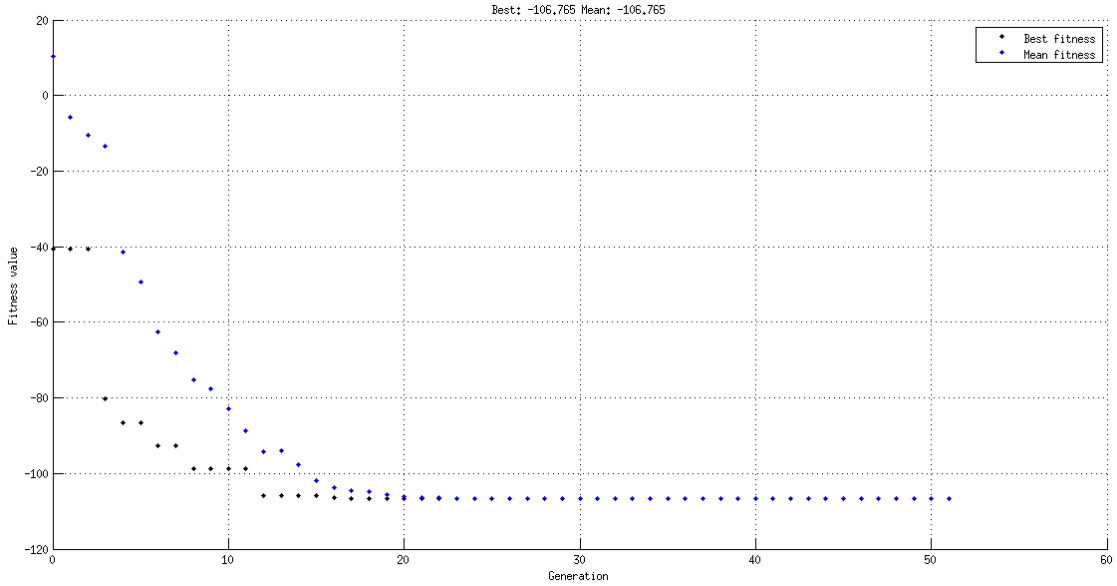

Рис. 5.3. График процесса оптимизации

Как видно из представленного графика генетический алгоритм успешно нашёл глобальный минимум представленной функции. Попробуем теперь найти минимум при наличии нелинейных ограничений вида:

$$x \cdot y - 13.8 = 0.$$

Для задания нелинейных ограничений требуется написание дополнительной функции задающей нелинейные ограничения в виде уравнений и неравенств:

$$\begin{cases} ceq = 0 \\ c \le 0 \end{cases}$$

Для нашего примера такая функция будет иметь следующий вид:

```
function [c, ceq] = test_constraint(x)
c = [];
ceq = [x(1) * x(2) - 13.8];
```

В этой функции требуется создать два массива с ограничениями в виде уравнений и неравенств. При этом если какого-то вида ограничений в системе нет, то должен вернуться пустой массив, как в примере с ограничениями в виде неравенств.

В случае нелинейных ограничений не на всех итерациях выполнения алгоритма получаются результаты удовлетворяющие ограничениям, но если алгоритм сходится к решению, то конечное решение будет удовлетворять поставленным ограничениям.

Ссылку на функцию с нелинейными ограничениями требуется написать в поле “Nonlinear constraint function”. Для данного примера повысим размер популяции с 20 до 60 организмов, а также выберем в качестве оператора мутации “Adaptive feasible” и запустим процесс оптимизации снова. График процесса оптимизации представлен на рис. 5.4.

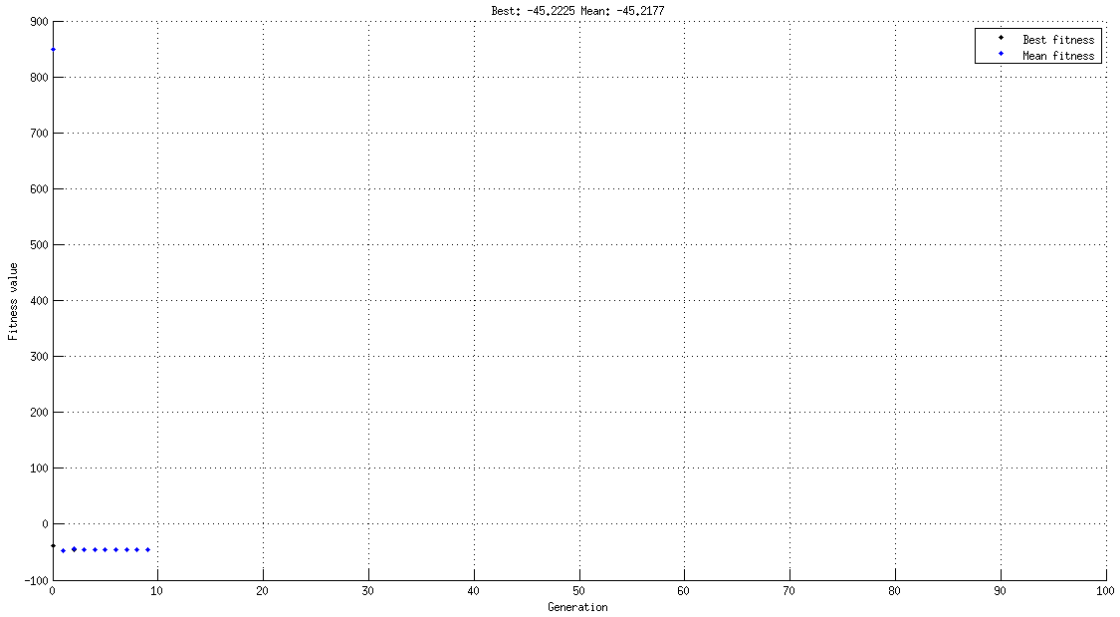

Рис. 5.4. График процесса оптимизации

Как видно из полученного графика генетический алгоритм справился с поиском минимума функции с нелинейными ограничениями.

Рассмотрим ещё один пример с более сложной для поиска функцией:

$$f(x,y) = -\cos(x)\cdot\cos(y)\cdot e^{-((x-\pi)^2+(y-\pi)^2)} \qquad (5.3)$$

Поверхность данной функции представлена на рис. 5.5.

График процесса оптимизации с функцией (5.3) представлен на рис.5.6.

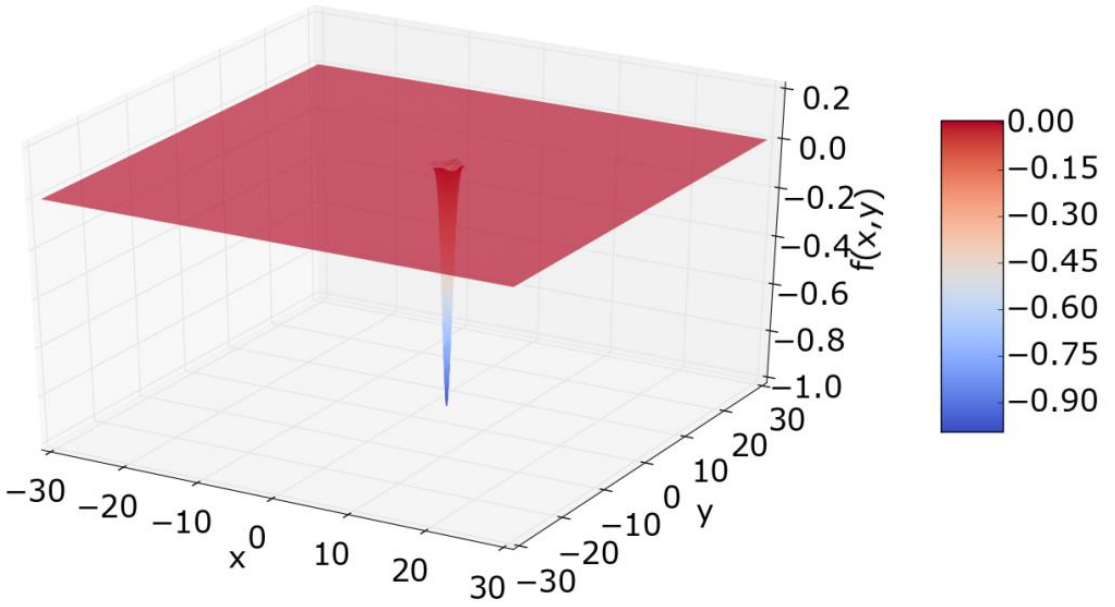

Рис. 5.5. График поверхности функции (5.3)

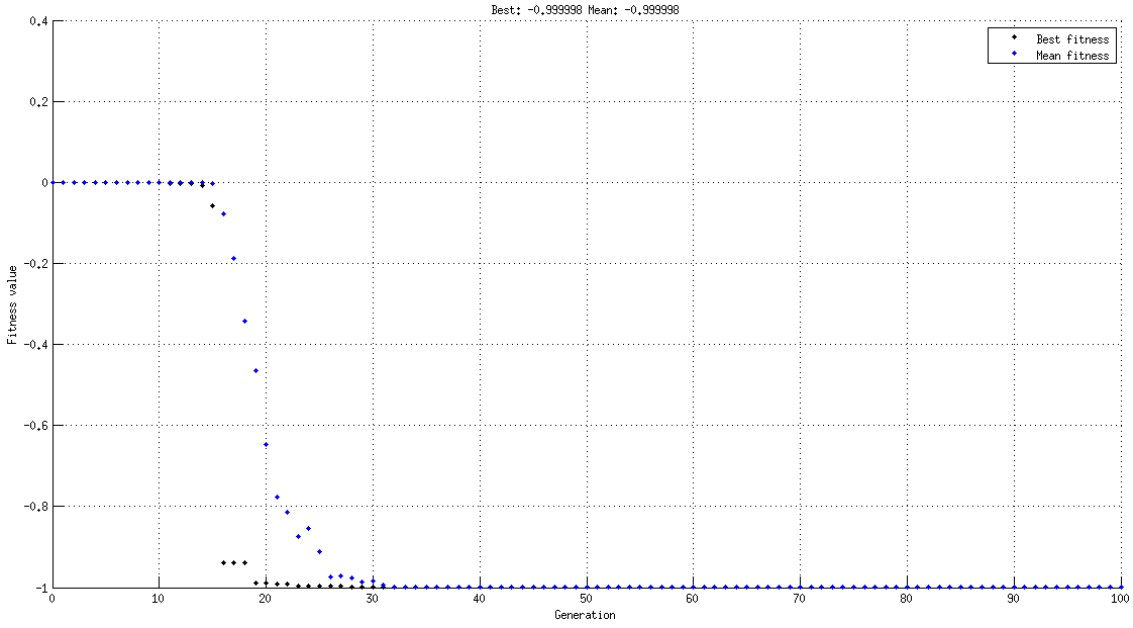

Рис. 5.6. График процесса оптимизации

Как видно из полученного графика генетический алгоритм справился с задачей поиска глобального минимума функции (5.3).

Программа лабораторной работы.

1. Используя генетические алгоритмы и пакет программ Matlab найти минимум функции одной переменной:

$$f(x) = 8x - 16 - 12\sqrt[3]{(x+4)^2}, x \geq -4$$

2. Используя генетические алгоритмы и пакет программ Matlab найти максимум функции двух переменных $z(x,y)=\exp(-x^2-y^2)+\sin(x+y)$ (для поиска максимума нужно преобразовать задачу в поиск минимума функции $-z(x,y)$) Поверхность функции $z(x,y)$ представлена на рис.5.7
3. Используя генетические алгоритмы и пакет программ Matlab найти минимум функции $f(x,y)=100\cdot(x^2-y)^2+(1-x)^2$ при наличии следующих ограничений:

$$\begin{cases} x\cdot y+x-y+1.5\le 0, \\ 10-x\cdot y\le 0, \\ 0\le x\le 1, \\ 0\le y\le 13. \end{cases}$$

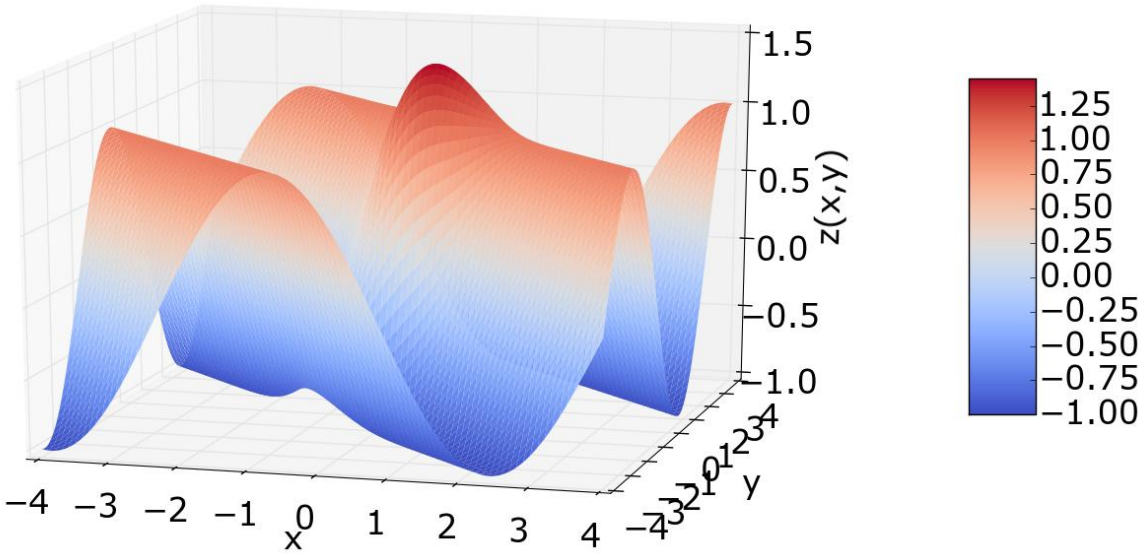

Рис. 5.7. Поверхность функции $z(x,y)$

Содержание отчёта.

- Цель работы.
- Теоретические сведения.
- Найденные значения минимумов и максимумов заданных функций вместе с графиками соответствующих процессов оптимизации.
- Выводы по работе генетических алгоритмов.

6. НАСТРОЙКА АДАПТИВНОЙ СИСТЕМЫ УПРАВЛЕНИЯ С ПОМОЩЬЮ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

Цель работы: изучение основ генетических алгоритмов, овладение навыками применения генетических алгоритмов для настройки систем автоматического управления.

Теоретические сведения. Генетические алгоритмы. Теоретические основы генетических алгоритмов представлены в работе №5.

Рассмотрим подробнее наиболее важные параметры генетического алгоритма, которые можно задать в Matlab Optimization Tool.

Секция 1: описание задачи оптимизации.

`Fitness function` – ссылка на Matlab функцию расчёта оценочной функции алгоритма, значение которой требуется минимизировать.

`Number of variables` – количество переменных, по которым производится поиск.

Подсекция ограничения (**`Constraints`**):

Поля **`Linear inequalities A`** и **`b`** задают ограничения в виде системы неравенств вида $\mathbf{Ax} \le \mathbf{b}$.

Поля **`Linear equalities Aeq`** и **`beq`** задают ограничения в виде системы уравнений вида $\mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq}$.

Поля **`Bounds Lower`** и **`Upper`** задают ограничения на значение переменных в виде $\mathbf{L} \le \mathbf{x} \le \mathbf{U}$.

`Nonlinear constraint function` – ссылка на Matlab функцию задания массивов нелинейных ограничений (подробнее рассматривалась в работе №5).

`Integer variable indices` – массив индексов переменных вектора **`x`**, значение которых должны быть целочисленными (Использование целочисленных переменных накладывает ряд ограничений на использование Matlab Optimization Tool, например, в системе должны отсутствовать ограничения в виде уравнений как линейных, так и нелинейных).

Секция 2: параметры оптимизации.

`Population type` (**`Double vector`**, **`Bit string`**, **`Custom`**) – указывает тип переменных вектора входных значений оценочной функции, **`Double vector`** – вектор вещественных чисел, **`Bit string`** – строка двоичных кодов, **`Custom`** – пользовательское представление.

`Population size` – размер популяции генетического алгоритма, т.е. количество особей в каждом поколении (если указать вектор, то алгоритм создаст подпопуляции, размеры которых соответствуют значениям вектора).

Creation function (**Uniform**, **Feasible population**, **Constraint dependent**, **Custom**) – метод создания исходной популяции организмов. **Uniform** создаёт исходную популяцию случайным образом с равномерным распределением, **feasible population** создаёт случайную популяцию удовлетворяющую ограничениям на значения переменных и линейным ограничениям. **Constraint dependent** используется по умолчанию и выбирает **Uniform** в случае отсутствия ограничений в системе и **feasible population** в обратном случае. **Custom** позволяет задать пользовательскую функцию для создания популяции.

Initial population позволяет задать начальную популяцию организмов, если размер задаваемой популяции будет меньше значения **Population size**, то остальные особи будут созданы с помощью **Creation function.**

Initial scores позволяет задать значения оценочной функции для исходной популяции, в случае если они не заданы, то они будут расчитанны с помощью функции заданной в **Fitness function.**

Initial range позволяет задать границы для значений генов особей исходной популяции.

Fitness scaling (**Rank**, **Proportional**, **Top**, **Shift linear**, **Custom**) задаёт функцию для преобразования результата оценочной функции в очки, используемые на этапе селекции особей.

Метод **Rank** масштабирует результат оценочной функции в зависимости от ранга особи в популяции. Ранг особи является её позицией в отсортированном по значению оценочной функции списке особей. Метод **Proportional** даёт очки пропорционально значению оценочной функции. Метод **Top** даёт одинаковые очки для всех особей, которые попадают в определённый процент лучших особей, остальные особи получают нулевые очки. Метод **Shift linear** масштабирует очки особей таким образом, чтобы значение очков самой приспособленной особи равнялось произведению **Max survival rate** и среднего значения оценочной функции среди особей данного поколения. **Custom** позволяет задать собственную функцию для преобразования результата оценочной функции.

Selection function (**Stochastic uniform**, **Remainder**, **Uniform**, **Roulette**, **Tournament**, **Custom**) задаёт функцию для выбора родителей для создания следующего поколения. При выборе **Stochastic uniform** для выбора родителей создаётся прямая, длина которой соответствует сумме всех очков особей данного поколения, при этом на прямой расположены отрезки, соответствующие количеству очков каждой особи. Случайным образом выбирается шаг, с которым алгоритм обходит данную прямую и выбирает в качестве ро-

дителей те особи, на чьи отрезки попадают шаги алгоритма. При выборе **Remainder** часть родителей выбираются строго в соответствии с целой частью их очков, затем дробные части участвуют в методе рулетки для выбора оставшихся родителей. При выборе **Uniform** родители выбираются случайным образом, что приводит к ненаправленному поиску. При выборе **Roulette** имитируется игровая рулетка, где каждой особи соответствует сектор на рулетке, размер которого пропорционален очкам особи. При выборе **Tournament** выбирается турнирный метод, в котором случайным образом выбираются несколько особей (их количество задано в **Tournament size**), после чего из них выбирается особь с наивысшими очками. При выборе **Custom** можно задать собственную функцию для выбора родителей и формирования родительского пула.

Подсекция **Reproduction** определяет некоторые параметры создания потомства из родительского пула. **Elite count** включает элитарную стратегию, при которой определённое количество лучших особей гарантированно попадают в следующее поколение. **Crossover fraction** задаёт какая часть следующей популяции будет получена с помощью кроссинговера. Оставшаяся часть особей будет получена с помощью мутации.

Mutation function (**Constraint dependent**, **Gaussian**, **Uniform**, **Adaptive feasible**, **Custom**) задаёт каким образом будет происходить мутация особей. При выборе **Gaussian** при мутации к каждому гену хромосомы будет прибавлено случайное число с нормальным распределением вероятности и нулевым математическим ожиданием. Параметр **Scale** задаёт величину дисперсии в первом поколении, а параметр **Shrink** задаёт скорость уменьшения величины дисперсии с каждым новым поколением. При выборе **Adaptive feasible** случайным образом выбираются направления для изменения генов хромосомы с учётом предыдущего поколения. Величина шага изменения хромосомы в выбранном направлении выбирается такой, чтобы удовлетворять линейным ограничениям и ограничениям на величину компонентов искомого вектора. При выборе **Constraint dependent** в случае отсутствия ограничений выбирается Гауссова мутация (**Gaussian**), иначе выбирается **Adaptive feasible**. При выборе **Uniform** сначала случайным образом выбирается часть генов в хромосоме, при этом вероятность мутации каждого из выбранных генов равна параметру **Rate**. Если мутация гена происходит, то он заменяется на случайное число из разрешённого диапазона. При выборе **Custom** можно задать свою функцию мутации.

Crossover function (**Scattered**, **Single point**, **Two point**, **Intermediate**, **Heuristic**, **Arithmetic**, **Custom**) задаёт метод создания потомства из двух

особей. При выборе **Scattered** случайным образом создаётся двоичный вектор, после чего новая особь получает гены первого родителя в тех локусах, где в случайном векторе расположены единицы, и гены второго родителя в остальных локусах. При выборе **Single point** (одноточечный кроссинговер) случайным образом выбирается позиция в хромосоме, после чего потомок получает гены от первого до выбранной позиции и гены от второго родителя после этой позиции. При выборе **Two point** (Двухточечный кроссинговер) случайным образом выбираются две позиции в хромосоме, после чего потомок получает гены от первого родителя в локусах расположенных до первой выбранной позиции и после второй выбранной позиции, остальные гены он получает от второго родителя. При выборе **Intermediate** гены потомка рассчитываются по следующей формуле $child_1 = parent_1 + rand \cdot Ratio \cdot (parent_2 - parent_1)$, где *rand* – случайная величина, а параметр *Ratio* задаётся с помощью поля **Ratio**. При выборе Heuristic хромосомы потомков находятся на прямой, соединяющей хромосомы родителей в n-мерном пространстве хромосом, где n – количество генов в хромосоме. При этом хромосомы потомков находятся вблизи родителя с большими очками, в стороне от родителя с меньшими очками. При выборе **Arithmetic** хромосомы потомков также находятся на прямой, соединяющей хромосомы родителей, но в случайной позиции между ними. При выборе **Custom** можно задать свою функцию кроссинговера.

Подсекция **Migration** задаёт параметры копирования лучших особей из одной подпопуляции в другую с заменой в ней худших особей.

Direction (**Forward**, **Both**) задаёт напраление перехода особей, либо только в одном направлении между подпопуляциями (**Forward**), либо в двух (**Both**). **Fraction** задаёт количество особей подпопуляции, которое будет копироваться, а параметр **Interval** задаёт количество поколений, между котороми происходит миграция.

Пример использования генетических алгоритмов в пакете Matlab для настройки прямой адаптивной системы управления с параметрической настройкой. Рассмотрим адаптивную систему для двухмассового упругого объекта из лабораторной работы №4. В предыдущей работе расчёт весовой матрицы $\mathbf{P}$ осуществлялся с использованием уравнения Ляпунова (1.5) и произвольной симметричной положительно оперделённой матрицы $\mathbf{Q}$. Также было показано, что выбор матрицы $\mathbf{Q}$ оказывает большое влияние на качество переходного процесса в системе. Воспользуемся теперь генетическими алгоритмами для поиска коэффициентов матрицы $\mathbf{P}$. Ввиду того что рассматриваемая система имеет только один вход и только один не нулевой ко-

эффициент в матрице $\mathbf{b}$, то в алгоритмах настройки адаптивного закона управления (1.4) будет участвовать только одна строка из искомой матрицы $\mathbf{P}$. Поэтому в нашем примере поиску подлежат только 4 коэффициента из матрицы $\mathbf{P}$. Построим оценочную функцию в этом случае следующим образом:

1. Промоделируем переходный процесс эталонной модели с постоянным шагом интегрирования Δt для получения переходных процессов $\mathbf{x}_m(t)$.

2. Промоделируем переходные процессы адаптивной системы с тем же постоянным шагом интегрирования Δt при уменьшении и увеличении в четыре раза во всех комбинациях всех неизвестных параметров матриц $\mathbf{A}$ и $\mathbf{b}$ для получения переходных процессов $\mathbf{x}(t)$.

3. Рассчитаем сумму модулей разностей между полученными переходными процессами в адаптивной системе и переходным процессом в эталонной модели для каждой точки интегрирования $s_j = \sum_{i=1}^{l} \left| x_{j_i} - x_{m_i} \right|$, где l – количество точек интегрирования.

4. В качестве результата оценочной функции выберем наибольшее из рассчитанных на этапе 3 значений $f = \max s_j$.

Для нашего примера примем в качестве неизвестных параметров матриц $\mathbf{A}$ и $\mathbf{b}$ момент инерции второй массы и коэффициент упругости связи, в этом случае описываемая оценочная функция может иметь следующий вид:

```
function fit = test_fitness(x)
P(4,1) = 20.7754524965707;
P(4,2) = 56.5826276830420;
P(4,3) = 4.53378207133062;
P(4,4) = 3.03918804755375;
assignin('base', 'P', P);
J2 = 0.1;
p = 0.8;
J2_nom = J2;
p_nom = p;
assignin('base', 'J2', J2);
assignin('base', 'p', p);
sim('elasticSystem');
youtm = yout;
P = zeros(4);
P(4,1) = x(1) / 10;
P(4,2) = x(2) / 10;
P(4,3) = x(3) / 10;
P(4,4) = x(4) / 10;
```

```
assignin('base', 'P', P);
result = [];
fit = 0;

params = [[J2_nom / 4, p_nom    ];
          [J2_nom * 4, p_nom    ];
          [J2_nom,     p_nom / 4];
          [J2_nom * 4, p_nom / 4];
          [J2_nom / 4, p_nom / 4]];

try
    for i = 1:length(params)
        J2 = params(i, 1);
        p  = params(i, 2);
        assignin('base', 'J2', J2);
        assignin('base', 'p', p);
        sum = 0;
        sim('elasticSystem');
        for j = 1:5:length(yout)
            sum = sum + abs(yout(j,1) - youtm(j,1));
        end
        result = [result (sum / 1000)];

        display(sum / 1000);
    end
    m = max(result);
    fit = m;
catch
    fit = 10000000000000;
end
display(P)
display(x)
display(fit)
J2 = J2_nom;
p = p_nom;
assignin('base', 'J2', J2);
assignin('base', 'p', p);
end
```

где для расчёта переходного процесса эталонной модели используются значения матрицы $\mathbf{P}$ рассчитанные с помощью уравнения Ляпунова. Для получения вектора состояния системы в переменную yout добавим выходные порты в блок схему системы. Блок схема полученной системы представлена на рис. 6.1. При этом номера выходных портов должны совпадать с позиционными номерами соответствующих переменных состояния в векторе состояния. В качестве метода интегрирования выберем метод с фиксированным ша-

гом Дорманда-Принца 8-го прядка точности и размер шага равный 0.002 с. В качестве параметров генетического алгоритма возьмём значения по умолчанию и увеличим размер популяции до 50 особей. График процесса оптимизации представлен на рис. 6.2. В результате процесса оптимизации была получена особь со следующей хромосомой: $\mathbf{x} = \begin{bmatrix} 3.773 & 4.22 & 0.106 & 0.014 \end{bmatrix}$.

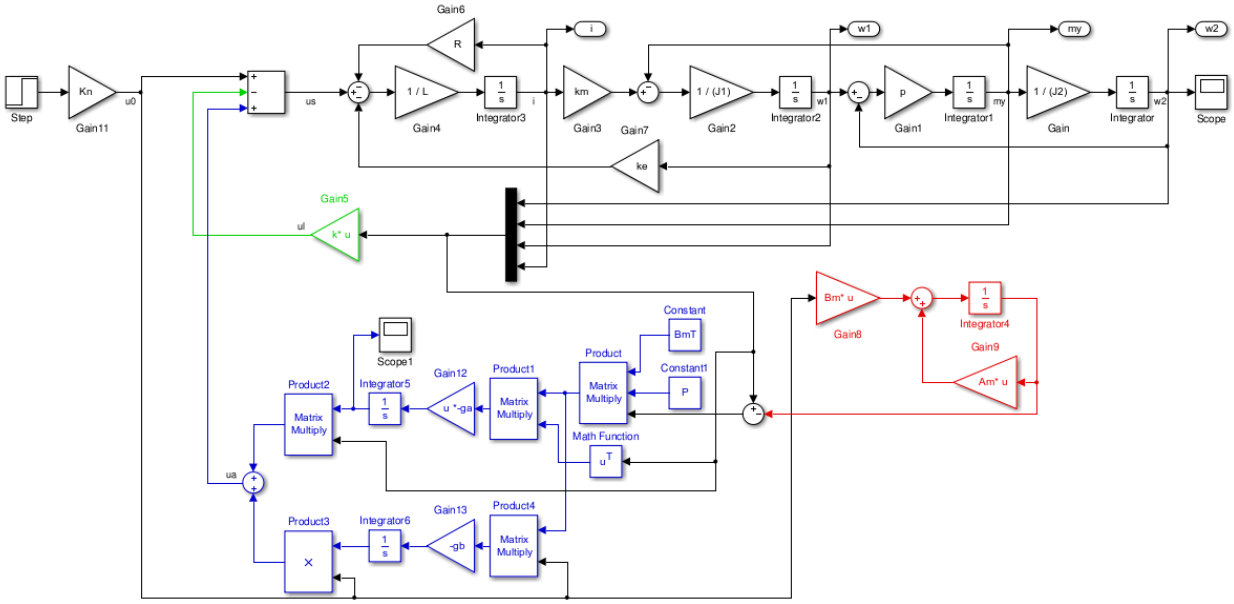

Рис. 6.1. Блок схема системы

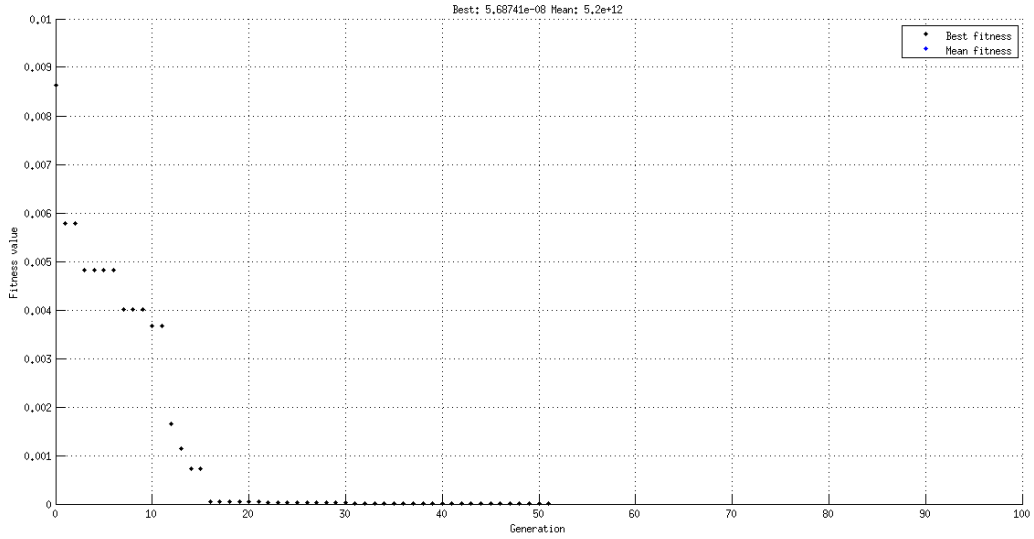

Рис. 6.2. График процесса оптимизации.

Рассмотрим теперь работу адаптивной системы управления с параметрической настройкой, рассчитанной с помощью генетических алгоритмов в сравнении с адаптивной системой, рассчитанной с помощью уравнения Ляпунова и матрицы $\mathbf{Q}_3$ показавшей наилучшие результаты в лабораторной ра-

боте №4. На рис. 6.3 представлены графики переходных процессов в рассматриваемых системах при уменьшении момента инерции второй массы в 4 раза. На рис. 6.4 представлены графики переходных процессов в рассматриваемых системах при увеличении момента инерции второй массы в 4 раза.

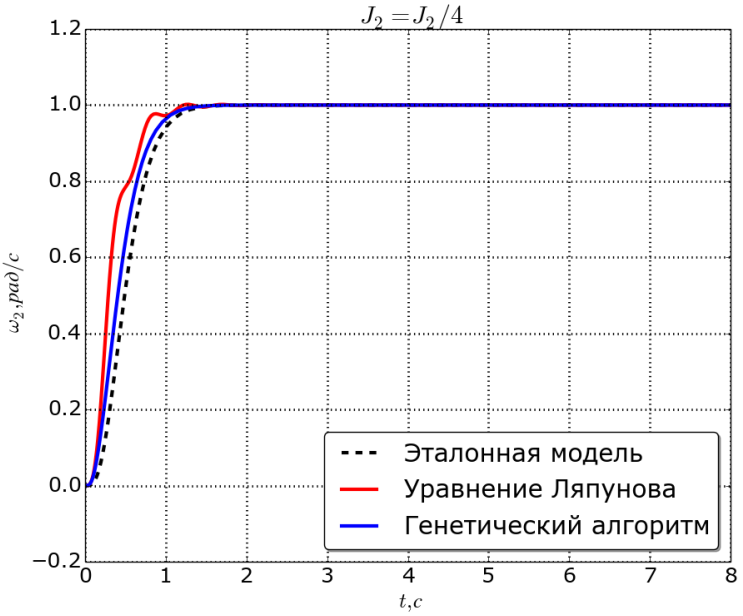

Рис. 6.3. Графики переходных процессов при уменьшении момента инерции второй массы в 4 раза

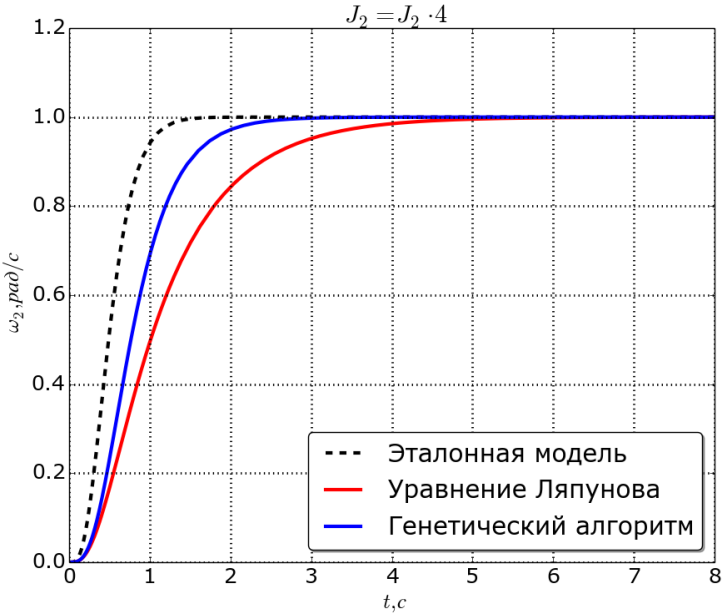

Рис. 6.4. Графики переходных процессов при увеличении момента инерции второй массы в 4 раза

На рис. 6.5. представлены графики переходных процессов в рассматриваемых системах при уменьшении коэффициента упругости связи в 4 раза. На рис. 6.6 представлены графики переходных процессов в рассматриваемых

системах при уменьшении коэффициента упругости связи и момента инерции второй массы в 4 раза.

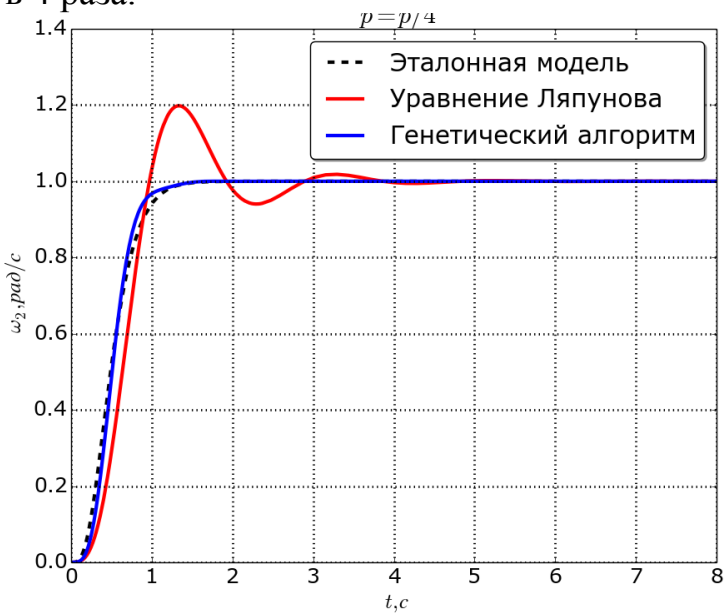

Рис. 6.5. Графики переходных процессов при уменьшении коэффициента упругости связи в 4 раза

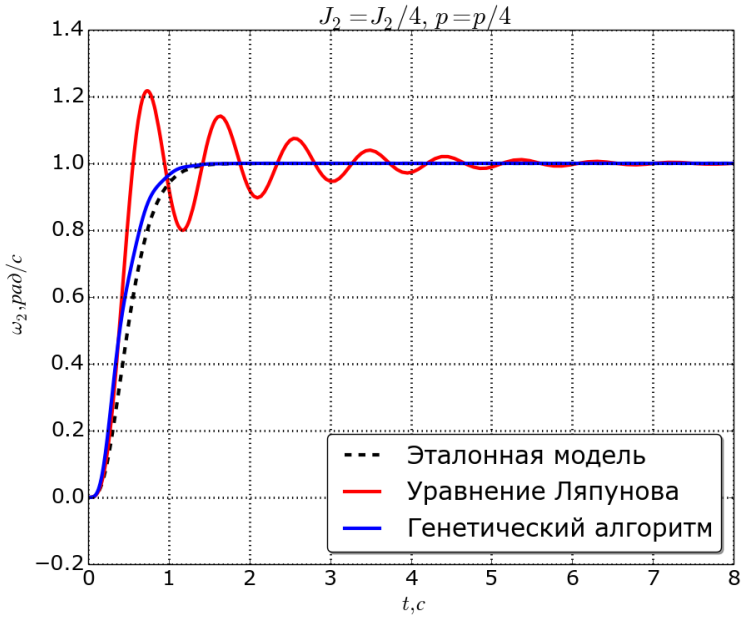

Рис. 6.6. Графики переходных процессов при уменьшении коэффициента упругости связи и момента инерции второй массы в 4 раза

На рис. 6.7. представлены графики переходных процессов в рассматриваемых системах при уменьшении коэффициента упругости связи и увеличении момента инерции второй массы в 4 раза.

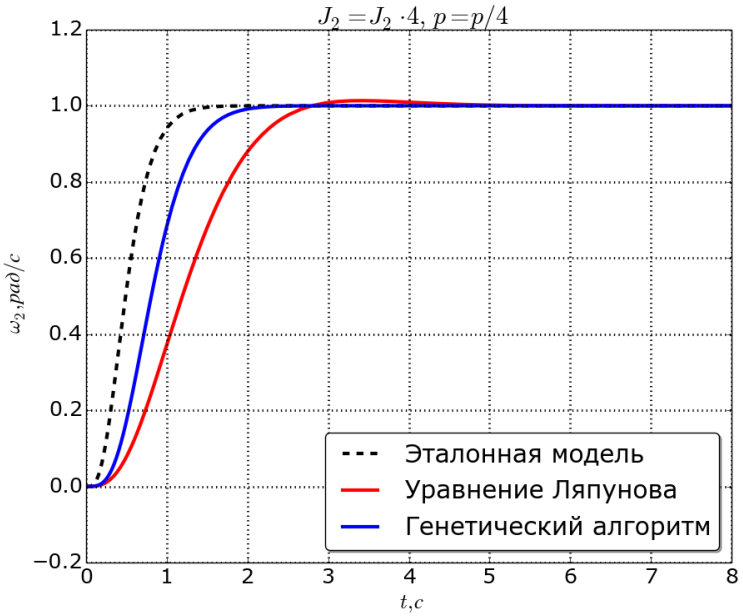

Рис. 6.7. Графики переходных процессов в рассматриваемых системах при уменьшении коэффициента упругости связи и увеличении момента инерции второй массы в 4 раза

Как видно из полученных графиков, применение генетических алгоритмов при расчёте адаптивной системы управления позволяет повысить качество переходных процессов в системе.

Программа лабораторной работы.

1. Используя генетические алгоритмы и Matlab Optimization Tool рассчитать параметры прямой адаптивной системы управления с параметрической настройкой для управления скоростью вращения второй массы в двухмассовом упругом электромеханическом объекте.
2. Произвести сравнительный анализ влияния 3-х различных параметров генетических алгоритмов на оптимизационный процесс.

Варианты параметров двухмассового объекта такие же как и в лабораторной работе №4

Содержание отчёта.

- Цель работы.
- Теоретические сведения.
- Расчёт адаптивной системы управления с помощью генетических алгоритмов.
- Сравнение результатов с системой, полученной в лабораторной работе №4.
- Сравнительный анализ влияния параметров генетического алгоритма.
- Выводы по эффективности использования генетических алгоритмов для расчёта адаптивной системы управления.

7. АДАПТИВНЫЙ НАБЛЮДАТЕЛЬ И ИДЕНТИФИКАТОР ЛЮДЕРСА-НАРЕНДРЫ

Цель работы: изучение основ адаптивных наблюдателей и идентификаторов, овладение навыками применения адаптивных наблюдателей для восстановления вектора состояния системы.

Теоретические сведения. Адаптивный наблюдатель и идентификатор Людерса-Нарендры. Для применения многих алгоритмов управления, например, модального или адаптивного управления требуется измерение полного вектора состояния системы. Однако на практике не всегда имеется возможность измерить весь вектор состояния системы, а наличие шума измерения делает невозможным восстановление недостающих переменных состояния с помощью дифференцирования измеряемых переменных. Стационарный наблюдатель состояния может быть применен, если параметры объекта управления известны, но в случае параметрической неопределённости его применение может привести к ухудшению качества работы системы или даже к её полной неработоспособности. Возможным решением проблемы восстановления вектора состояния при неизвестных параметрах объекта может служить адаптивный наблюдатель и идентификатор Людерса-Нарендры. Рассмотрим линейную стационарную систему *n*-го порядка с одним входом и одним выходом вида:

$$\begin{cases} \mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{b}u(t) \\ y = \mathbf{c}\mathbf{x} \end{cases}, \tag{7.1}$$

данная система может быть представлена в виде передаточной функции *n*-го порядка, поэтому для её идентификации необходимо оценить только $2n$ параметров. Если система (7.1) является полностью наблюдаемой, то такую систему можно записать в следующем виде:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_1 & 1 & & \\ a_2 & & \ddots & \\ \vdots & & & 1 \\ a_n & & \mathbf{\Lambda} & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} u, \tag{7.2}$$

$$y = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \mathbf{x}$$

где $\mathbf{\Lambda}$ – квадратная $(n-1)\times(n-1)$ диагональная матрица с известными отрицательными диагональными элементами $-\lambda_i (i = 2 \dots n)$, $a = (a_1, a_2, \dots a_n)^T$ и $b = (b_1, b_2, \dots, b_n)^T$ – неизвестные $2n$ параметров, подлежащие определению.

Передаточная функция системы (7.2) может быть записана следующим образом:

$$T(s)=\frac{b_1+b_2\dfrac{1}{s+\lambda_2}+\ldots+b_n\dfrac{1}{s+\lambda_n}}{s-a_1-a_2\dfrac{1}{s+\lambda_2}-\ldots-a_n\dfrac{1}{s+\lambda_n}}. \qquad (7.3)$$

Введём в систему дополнительные сигналы вида

$$\begin{matrix} \dot{v}_i+\lambda_i v_i=x_1, i=2,\ldots n \\ \dot{s}_i+\lambda_i s_i=u, i=2,\ldots n \end{matrix}, \qquad (7.4)$$

при этом алгоритмы оценки неизвестных $2n$ параметров будут иметь следующий вид:

$$\begin{matrix} \dot{\alpha}_1=-c_1x_1(\dot{\hat{x}}_1-x_1) & \dot{\beta}_1=-d_1u(\dot{\hat{x}}_1-x_1) \\ \dot{\alpha}_2=-c_2v_2(\dot{\hat{x}}_1-x_1) & \dot{\beta}_2=-d_2s_2(\dot{\hat{x}}_1-x_1) \\ \vdots & \vdots \\ \dot{\alpha}_n=-c_nv_n(\dot{\hat{x}}_1-x_1) & \dot{\beta}_n=-d_ns_n(\dot{\hat{x}}_1-x_1) \end{matrix}, \qquad (7.5)$$

где c_i и d_i – произвольные положительные коэффициенты. В этом случае уравнения адаптивного наблюдателя будут иметь следующий вид:

$$\begin{matrix} \dot{\hat{x}}_1=\alpha_1x_1+\hat{x}_2+\ldots+\hat{x}_n+\beta_1u-\lambda_1(\hat{x}_1-x_1) \\ \dot{\hat{x}}_2=\alpha_2x_1-\lambda_2\hat{x}_2+\beta_2u+(c_2v_2^2+d_2s_2^2)(\hat{x}_1-x_1) \\ \vdots \\ \dot{\hat{x}}_n=\alpha_nx_1-\lambda_n\hat{x}_n+\beta_nu+(c_nv_n^2+d_ns_n^2)(\hat{x}_1-x_1) \end{matrix}, \qquad (7.6)$$

где λ_1 – произвольный положительный коэффициент.

Пример расчёта адаптивного наблюдателя и идентификатора Людерса-Нарендры. Рассмотрим линейную систему второго порядка из лабораторной работы №3, имеющую следующие параметры:

$$\mathbf{A}=\begin{bmatrix} 0 & a_1 \\ -a_2 & -a_2 \end{bmatrix},\ \mathbf{b}=\begin{bmatrix} 0 \\ a_2 \end{bmatrix} \text{ и } \mathbf{c}=\begin{bmatrix} 1 & 0 \end{bmatrix}, \text{ где } a_1=3,\ a_2=2.$$

Представление данной системы в пространстве состояний не соответствует представлению (7.2), поэтому для применения в ней адаптивного наблюдателя необходимо произвести преобразование. Получим передаточную функцию исходной системы с помощью функции из пакета Matlab **`ss2tf`** имеющей следующую форму **`[NUM, DEN] = ss2tf(A, B, C, D)`**, где **`NUM`** – это коэффициенты числителя передаточной функции, **`DEN`** – коэффициенты знамена-

теля передаточной функции, а **D** – матрица прямой связи (в нашем примере она равна 0). Для нашего примера коэффициенты передаточной функции имеют следующие значения: $\mathrm{NUM}=[0,0,6]$, $\mathrm{DEN}=[1,2,6]$ что соответствует следующей передаточной функции:

$$T(s)=\frac{6}{s^2+2s+6}, \tag{7.7}$$

Передаточная функция системы второго порядка в форме (7.2) согласно (7.3) может быть записана с помощью следующей передаточной функции:

$$T_o(s)=\frac{b_1+\dfrac{b_2}{s+\lambda_2}}{s-a_1-\dfrac{a_2}{s+\lambda_2}}=\frac{b_1 s+b_1\lambda_2+b_2}{s^2+(\lambda_2-a_1)s-a_1\lambda_2-a_2}. \tag{7.8}$$

Приравнивая коэффициенты при одинаковых степенях s, мы можем получить следующие уравнения для параметров системы в форме (7.2) в зависимости от значения коэффициента λ_2:

$$\mathbf{A}_o=\begin{bmatrix}\lambda_2-2 & 1\\ -\lambda_2^2+2\lambda_2-6 & -\lambda_2\end{bmatrix},\ \mathbf{b}_o=\begin{bmatrix}0\\6\end{bmatrix}. \tag{7.9}$$

Возьмём для нашего примера следующие значения для параметров адаптивного наблюдателя: $\lambda_1=100$, $\lambda_2=20$, $c_1=100$, $c_2=100$, $d_1=100$ и $d_2=100$.

Блок схема полученной системы с адаптивным наблюдателем представлена на рис. 7.1. Блок схема подсистемы Adaptive Observer представлена на рис. 7.2.

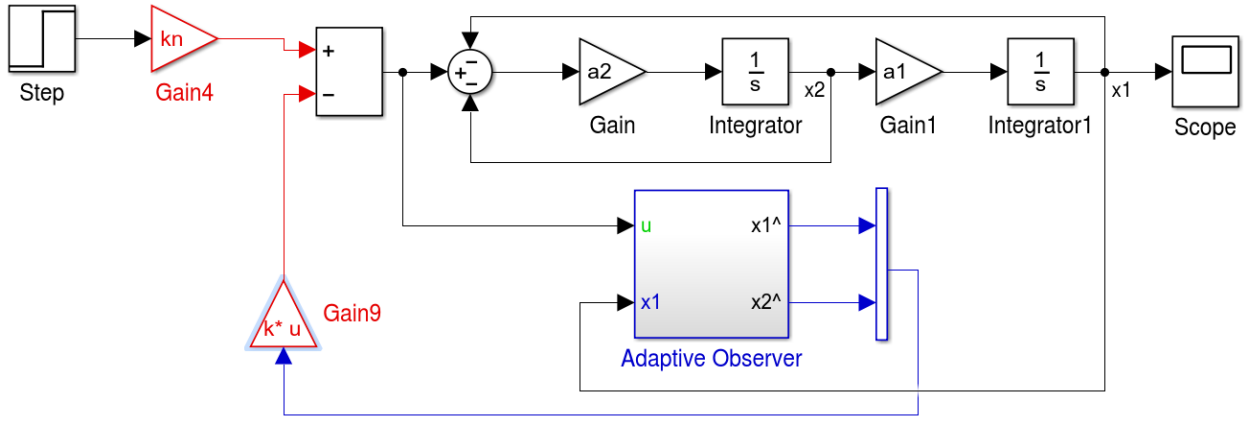

Рис. 7.1. Блок схема системы

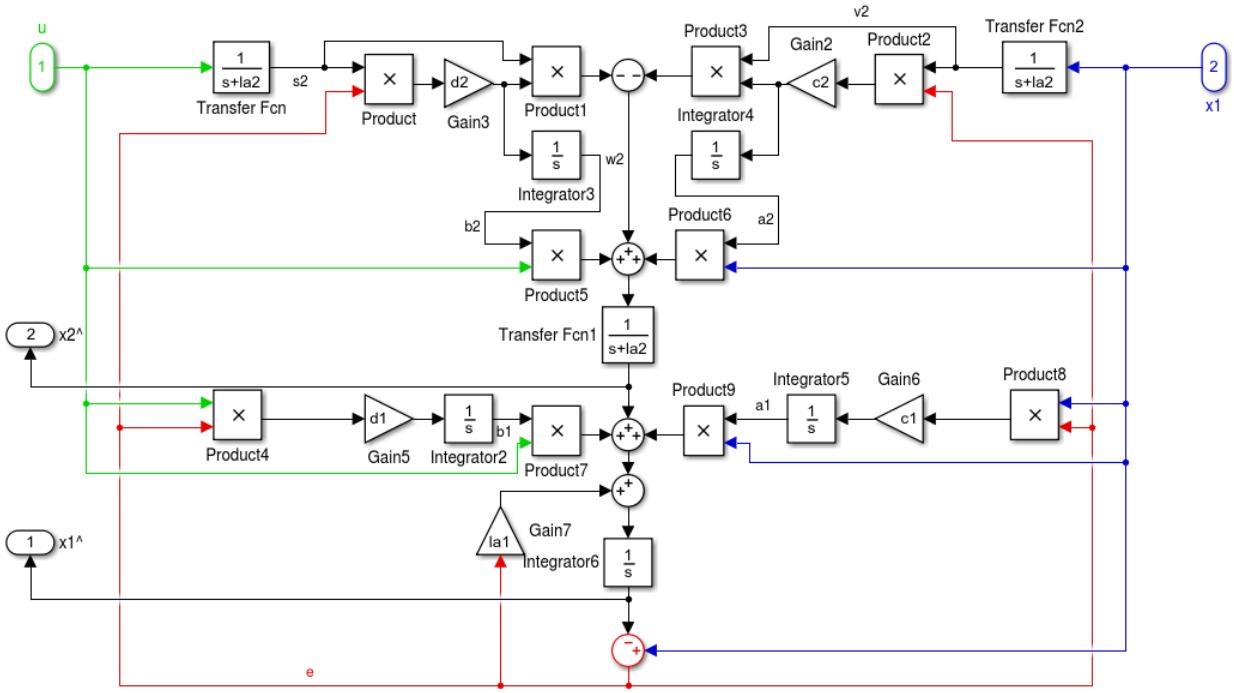

Рис. 7.2. Блок схема адаптивного наблюдателя

В качестве начальных значений идентифицированных параметров объекта выберем значения матрицы $\mathbf{A}_o$ рассчитанные при номинальных параметрах объекта управления, установив их в качестве начальных значений соответствующих интеграторов на схеме. Параметры модального управления для системы (7.9) имеют следующие значения: $\mathbf{k} = [23.375 \quad 1.1667]$, $k_n = 3.375$. На рис. 7.3. представлены переходные процессы в системе с адаптивным наблюдателем и модальным управлением при изменении параметров a_1 и a_2 исходного объекта в 4 раза.

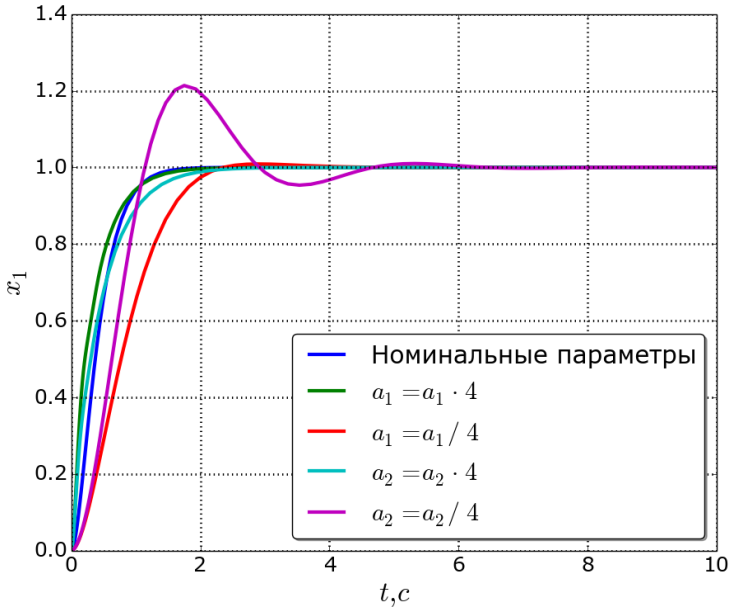

Рис. 7.3. Переходные процессы в системе с адаптивным наблюдателем и модальным управлением

Как видно из полученных графиков применение адаптивного наблюдателя позволяет восстановить полный вектор состояния объекта управления.

Программа лабораторной работы.

1. Построить и исследовать моделированием объект управления (3.5) с заданными параметрами a_1 и a_2.

2. Построить модальное управление (3.1) и исследовать систему при замыкании обратных связей модального управления по переменным состояния объекта управления при номинальных параметрах и при изменении параметров a_1 и a_2 в 4 раза.

3. Построить адаптивный наблюдатель состояния (7.4-7.6) и исследовать систему с модальным управлением и наблюдателем состояния при номинальных параметрах и при изменении параметров a_1 и a_2 в 4 раза.

4. Произвести сравнение рассчитанных значений матрицы $\mathbf{A}_o$ при изменении параметров объекта управления и идентифицированных значений с помощью адаптивного наблюдателя состояния.

Варианты параметров объекта управления такие же как и в лабораторной работе №3

Содержание отчёта.

- Цель работы.
- Теоретические сведения.
- Расчёт параметров системы в форме (7.2).
- Расчёт модального управления для системы в форме (7.2).
- Расчёт адаптивного наблюдателя и идентификатора Людерса-Нарендры.
- Исследование системы с модальным управлением и адаптивным наблюдателем при изменении параметров объекта управления.
- Сравнение рассчитанных и идентифицированных значений параметров объекта.
- Выводы по эффективности использования адаптивного наблюдателя и идентификатора Людерса-Нарендры для восстановления вектора состояния системы и идентификации параметров объекта.

8. НАБЛЮДАТЕЛЬ СОСТОЯНИЯ СО СКОЛЬЗЯЩИМИ РЕЖИМАМИ

Цель работы: изучение основ наблюдателей со скользящими режимами, овладение навыками применения наблюдателей со скользящими режимами для восстановления вектора состояния системы.

Теоретические сведения. Наблюдатель состояния со скользящими режимами. Для восстановления полного вектора состояния при неизвестных параметрах объекта управления может также использоваться наблюдатель состояния со скользящими режимами. Рассмотрим следующую линейную стационарную систему *n*-го порядка

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{b}u \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned}, \tag{8.1}$$

где $\mathbf{x} \in R^n$, $\mathbf{y} \in R^m$. Если система (8.1) полностью наблюдаема и матрица $\mathbf{C}$ имеет полный ранг, то наблюдатель состояния со скользящими режимами будет иметь следующий вид

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{b}u + \mathbf{L}_s \operatorname{sgn}(\mathbf{y} - \hat{\mathbf{y}}), \tag{8.2}$$

где функция sgn определена следующим образом

$$\operatorname{sgn}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \operatorname{sgn}(x_1) \\ \operatorname{sgn}(x_2) \\ \vdots \\ \operatorname{sgn}(x_n) \end{bmatrix}, \operatorname{sgn}(x) = \begin{cases} 1, x > 0 \\ -1, x < 0 \end{cases},$$

а матрица $\mathbf{L}_s \in R^{n \times m}$ матрица наблюдателя состояния, параметры которой выбираются исходя из условия сходимости $\hat{\mathbf{x}}$ к $\mathbf{x}$.

Применим статическое преобразование к системе (8.1) для прямого получения всех *m* выходов системы как части её *n*-мерного вектора состояния с помощью следующей невырожденной матрицы преобразования:

$$\mathbf{T} = \begin{bmatrix} \mathbf{C} \\ \mathbf{R} \end{bmatrix}, \tag{8.3}$$

где $\mathbf{R} \in R^{(n-m) \times n}$ матрица преобразования, определяющая следующее преобразование $\mathbf{x}_1 = \mathbf{R}\mathbf{x}$, где $\mathbf{x}_1 \in R^{n-m}$, и новую форму записи исходной системы:

$$\begin{aligned} \dot{\mathbf{y}} &= \mathbf{A}_{11}\mathbf{y} + \mathbf{A}_{12}\mathbf{x}_1 + \mathbf{b}_1 u \\ \dot{\mathbf{x}}_1 &= \mathbf{A}_{21}\mathbf{y} + \mathbf{A}_{22}\mathbf{x}_1 + \mathbf{b}_2 u \end{aligned}, \tag{8.4}$$

где $\mathbf{TAT}^{-1} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$ и $\mathbf{Tb} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$.

Уравнение наблюдателя состояния со скользящими режимами в таком случае можно записать следующим образом:

$$\begin{aligned} \dot{\hat{\mathbf{y}}} &= \mathbf{A}_{11}\hat{\mathbf{y}} + \mathbf{A}_{12}\hat{\mathbf{x}}_1 + \mathbf{b}_1 u + \mathbf{L}_1 \operatorname{sgn}\mathbf{e}_y \\ \dot{\hat{\mathbf{x}}} &= \mathbf{A}_{21}\hat{\mathbf{y}} + \mathbf{A}_{22}\hat{\mathbf{x}}_1 + \mathbf{b}_2 u + \mathbf{L}_2 \operatorname{sgn}\mathbf{e}_y \end{aligned}, \qquad (8.5)$$

где $\mathbf{e}_y = \mathbf{y} - \hat{\mathbf{y}}$ – ошибка восстановления, $\mathbf{L}_1 \in R^{m\times m}$, $\mathbf{L}_2 \in R^{(n-m)\times m}$ – параметры наблюдателя.

Наблюдатель состояния со скользящими режимами (8.2) асимптотически восстанавливает полный вектор состояния системы (8.1) в случае, если

$$\mathbf{L}_s = \mathbf{T}^{-1}\begin{bmatrix} \mathbf{L}_1 \\ \mathbf{L}_2\mathbf{L}_1 \end{bmatrix},$$

где $\mathbf{L}_1 \in R^{m\times m}$ выбрана в виде положительно определённой диагональной матрицы, каждый элемент которой больше чем $\left\|\mathbf{A}_{11}\mathbf{e}_y + \mathbf{A}_{12}\mathbf{e}_{x1}\right\|_\infty$, $\mathbf{L}_2 \in R^{(n-m)\times m}$ матрица располагающая желаемым образом собственные числа матрицы $\left(\mathbf{A}_{22} - \mathbf{L}_2\mathbf{A}_{12}\right)$, $\mathbf{e}_{x1} = \mathbf{x}_1 - \hat{\mathbf{x}}_1$.

Пример расчёта наблюдателя состояния со скользящими режимами. Адаптивная система управления, построенная в лабораторной работе №6, использует полный вектор состояния системы, однако на практике не всегда есть возможность измерять весь вектор состояния системы. Примем, что в системе из лабораторной работы №6 имеется датчик скорости вращения второй массы, а также датчик тока двигателя постоянного тока. Для восстановления полного вектора состояния применим стационарный наблюдатель состояния Люенбергера из лабораторной работы №3, а также наблюдатель состояния со скользящими режимами. Так как быстродействие наблюдателя состояния должно быть выше быстродействия адаптивной системы управления, то выберем желаемое расположение корней для наблюдателя состояния Люенбергера в виде $\mathbf{p}_o = \begin{bmatrix} -450 & -450.0001 & -449.9999 & -450 \end{bmatrix}$. Так как в системе присутствуют 2 датчика, то матрица выхода в данном примере будет иметь следующий вид $\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$. Так как матрица выхода имеет 2 строки, то мы не можем использовать функцию acker из пакета Matlab для расчёта матрицы усиления наблюдателя, поэтому воспользуемся похожей функцией place и следующей командой в системе Matlab `Lo = place(A',`

`C', po)'`, где `po` желаемый вектор корней наблюдателя Люенбергера. Функция place может иметь трудности с расположением корней в одной точке, поэтому значения вектора $\mathbf{p}_o$ выбраны не одинаковыми, но близкими. Таким образом, матрица параметров стационарного наблюдателя будет иметь следующие значения:

$$\mathbf{L}_o = \begin{bmatrix} 899.999 & -0.08 \\ 20248 & -7.2018 \\ -1800 & -2004.8 \\ 199.999 & 700 \end{bmatrix}.$$

Рассчитаем также параметры наблюдателя состояния со скользящими режимами. Так как измерению подлежат только первая и четвёртая переменные состояния, то матрица преобразования неизмеряемых переменных состояния будет иметь следующий вид $\mathbf{R} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$. В таком случае, матрица преобразования системы, согласно (8.3) будет иметь следующие значения:

$$\mathbf{T} = \begin{bmatrix} \mathbf{C} \\ \mathbf{R} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

В таком случае, согласно (8.4) мы можем получить следующие матрицы системы (8.5):

$$\mathbf{A}_{11} = \begin{bmatrix} 0 & 0 \\ 0 & -\dfrac{R}{L} \end{bmatrix}, \mathbf{A}_{12} = \begin{bmatrix} \dfrac{1}{J_2} & 0 \\ 0 & -\dfrac{k_e}{L} \end{bmatrix}, \mathbf{A}_{21} = \begin{bmatrix} -p & 0 \\ 0 & \dfrac{k_m}{J_1} \end{bmatrix}, \mathbf{A}_{22} = \begin{bmatrix} 0 & p \\ -\dfrac{1}{J_1} & 0 \end{bmatrix}, \mathbf{b}_1 = \begin{bmatrix} 0 \\ \dfrac{1}{L} \end{bmatrix},$$

$$\mathbf{b}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Выберем в качестве матрицы параметров наблюдателя следующую положительно определённую диагональную матрицу $\mathbf{L}_1 = \mathrm{diag}[60 \quad 60]$, а в качестве желаемого вектора корней наблюдателя состояния со скользящими режимами выберем $\mathbf{p}_s = \begin{bmatrix} -450 & -450 \end{bmatrix}$. Рассматриваемые матрицы $\mathbf{A}_{12}$ и $\mathbf{A}_{22}$ нельзя использовать с функцией пакета Matlab acker, поэтому мы воспользу-

емся функцией пакета Matlab place. Для расчёта второй матрицы параметров наблюдателя состояния со скользящими режимами воспользуемся следующей командой пакета Matlab **`L2 = place(A22, A12, ps)`**, где **`A22`** и **`A11`** соответствующие матрицы из (8.5), а **`ps`** – вектор желаемых корней. В таком случае искомая матрица наблюдателя состояния со скользящими режимами (8.2) будет иметь следующие значения:

$$\mathbf{L}_s = \begin{bmatrix} 100 & 0 \\ 4500 & 8 \\ 20 & -450 \\ 0 & 100 \end{bmatrix}.$$

Структурная схема системы из лабораторной работы №6 с двумя подключаемыми наблюдателями состояния представлена на рис. 8.1. Структурная схема подсистемы наблюдателя Люенбергера представлена на рис. 8.2. Структурная схема подсистемы наблюдателя со скользящими режимами представлена на рис. 8.3. Для ускорения процесса моделирования в системе Matlab/Simulink функция sgn была реализована в виде блока насыщения с границами $\begin{bmatrix} 1 & -1 \end{bmatrix}$, а также высокого коэффициента усиления равного $1 \cdot 10^{10}$. Переходные процессы в системе с наблюдателем состояния с уменьшенным моментом инерции второй массы в 4 раза представлены на рис. 8.4.

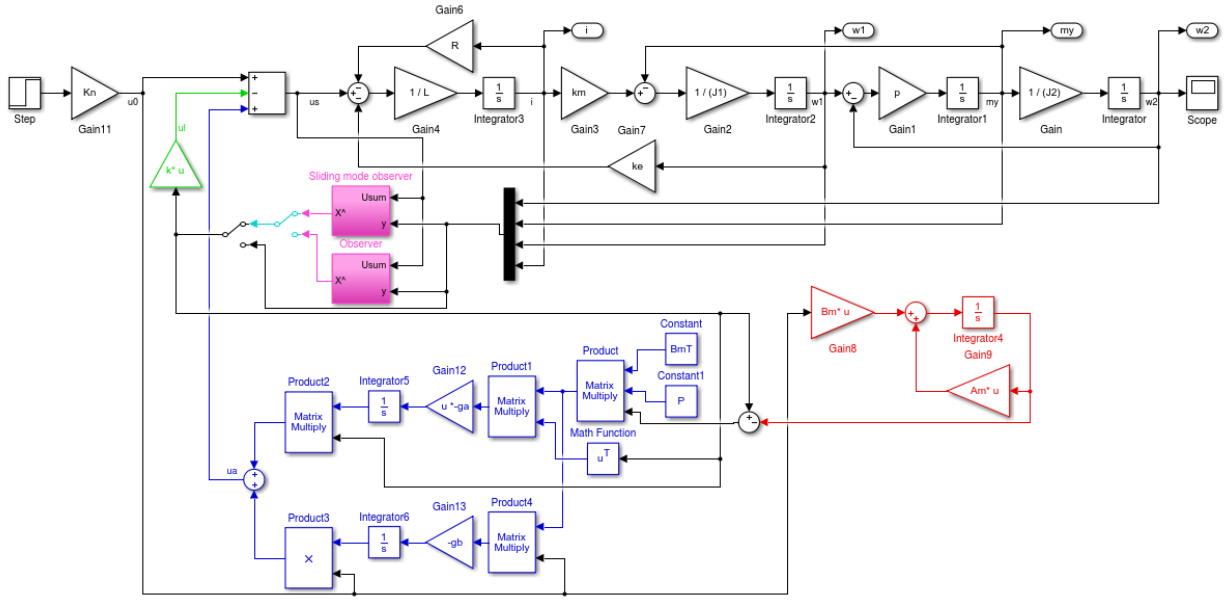

Рис. 8.1. Структурная схема системы

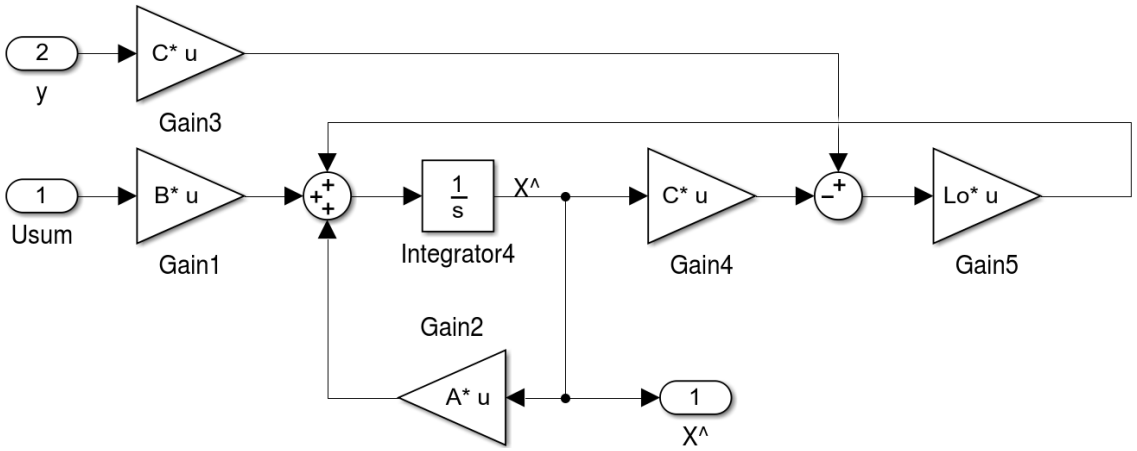

Рис. 8.2. Структурная схема наблюдателя состояния Люенбергера

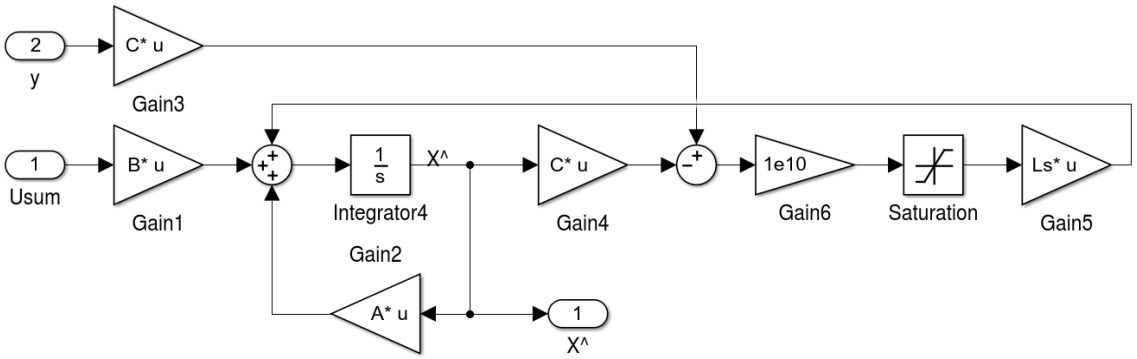

Рис. 8.3. Структурная схема наблюдателя состояния со скользящими режимами.

На рис. 8.4-8.11 представлены переходные процессы в системе при четырёх-кратном изменении параметров объекта управления.

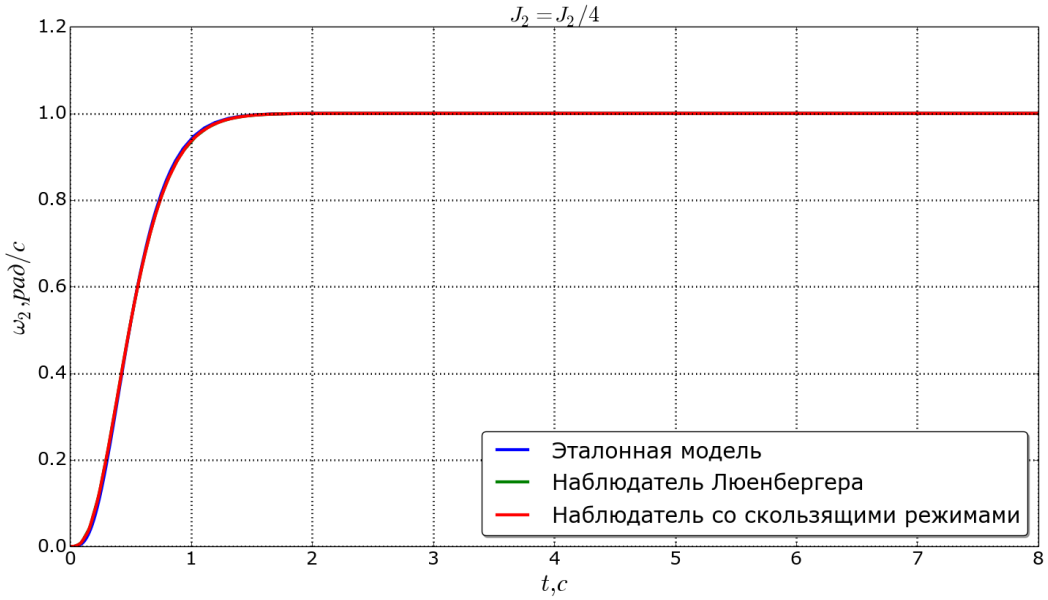

Рис. 8.4. Переходные процессы в системе при уменьшении момента инерции второй массы

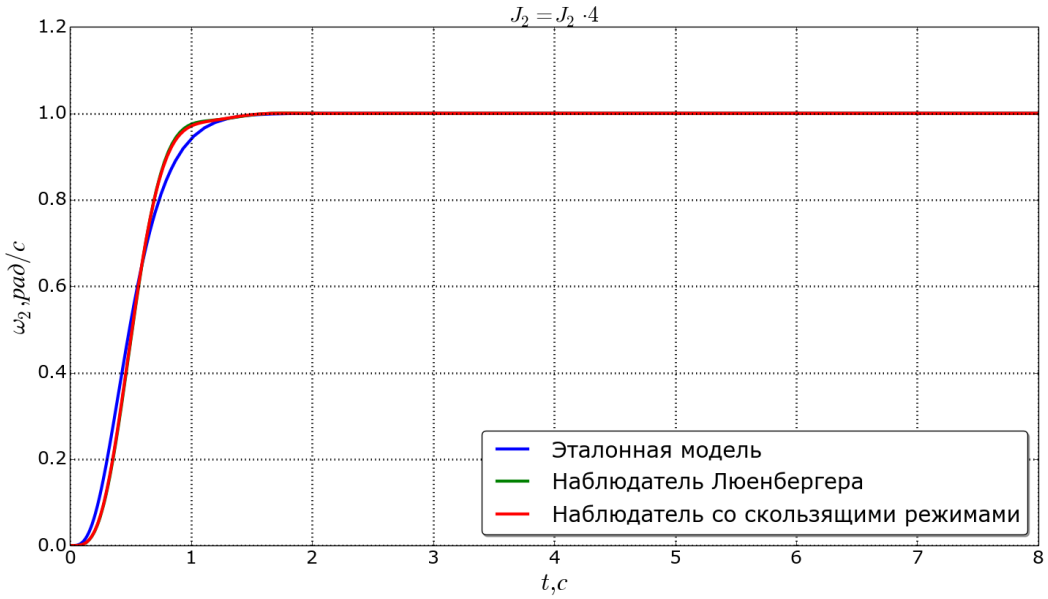

Рис. 8.5. Переходные процессы в системе при увеличении момента инерции второй массы

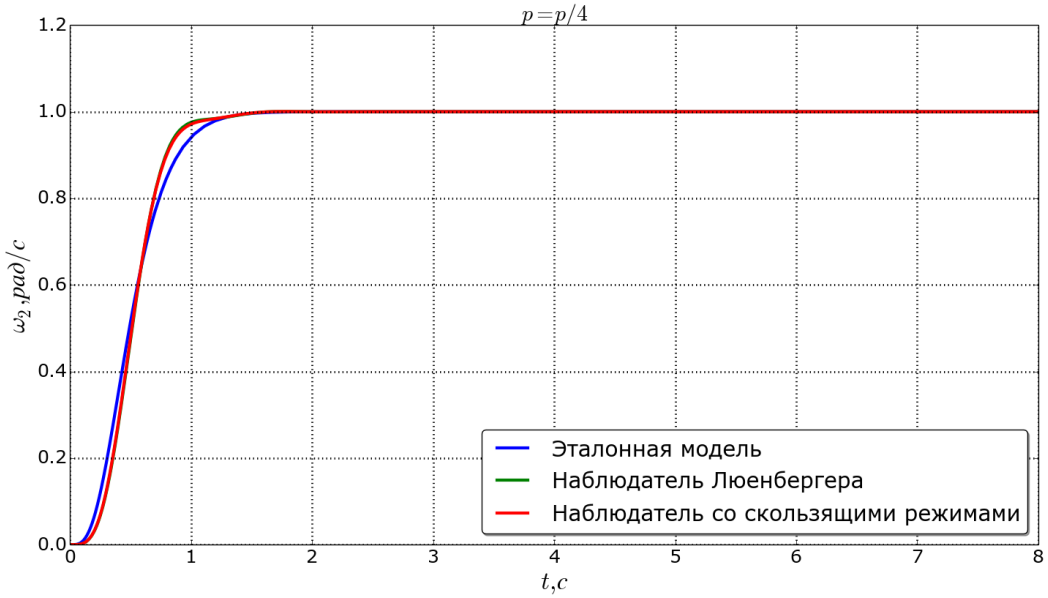

Рис. 8.6. Переходные процессы в системе при уменьшении коэффициента упругости связи

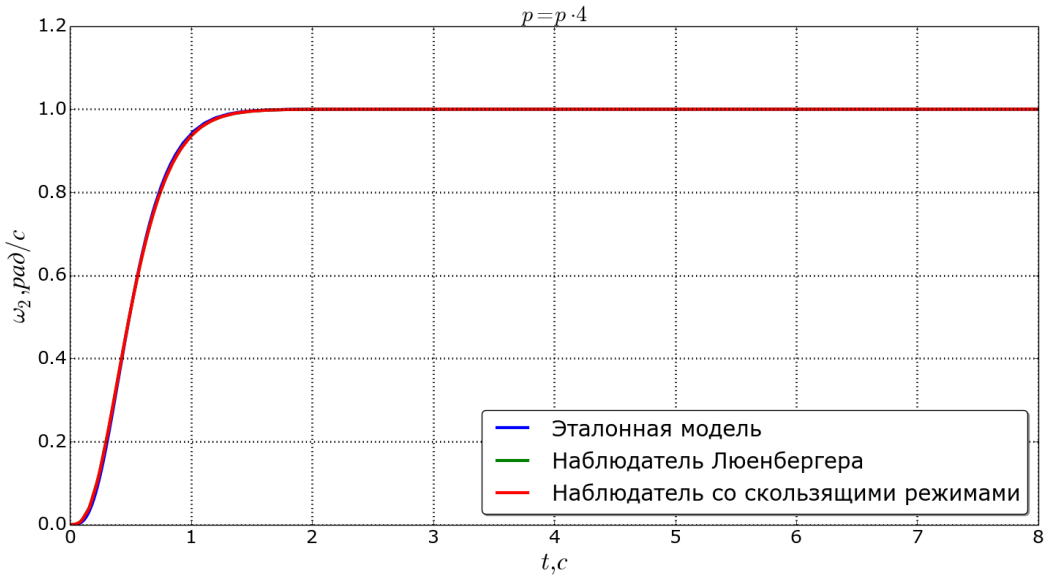

Рис. 8.7. Переходные процессы в системе при увеличении коэффициента упругости связи

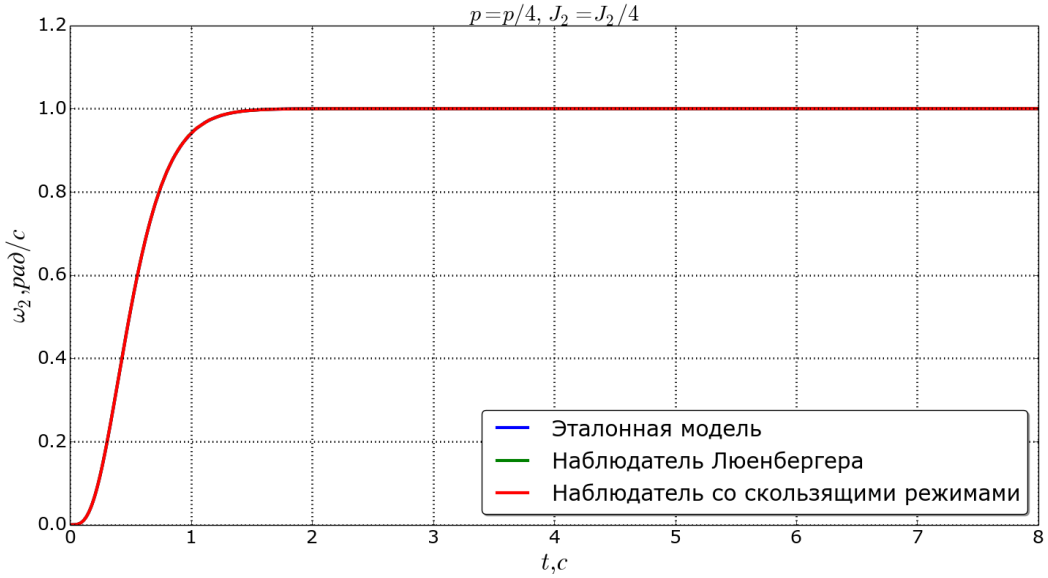

Рис. 8.8. Переходные процессы в системе при уменьшении коэффициента упругости связи и уменьшении момента инерции второй массы

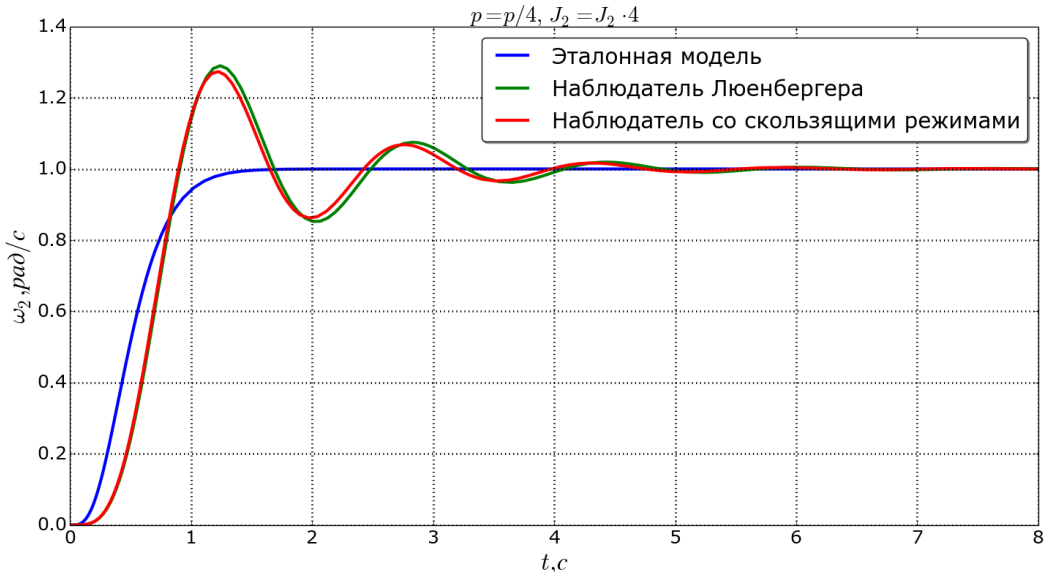

Рис. 8.9. Переходные процессы в системе при уменьшении коэффициента упругости связи и увеличении момента инерции второй массы

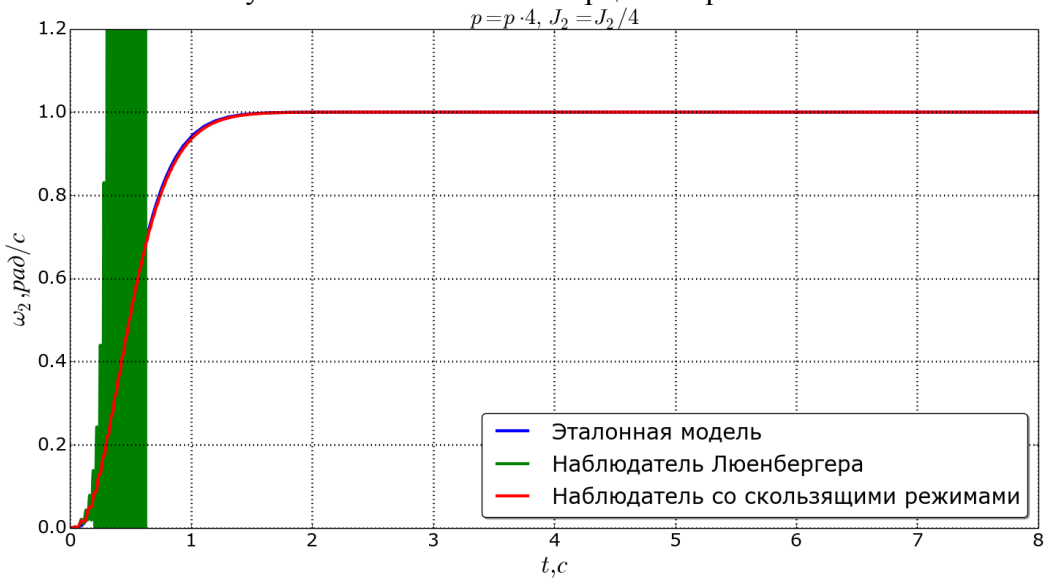

Рис. 8.10. Переходные процессы в системе при увеличении коэффициента упругости связи и уменьшении момента инерции второй массы

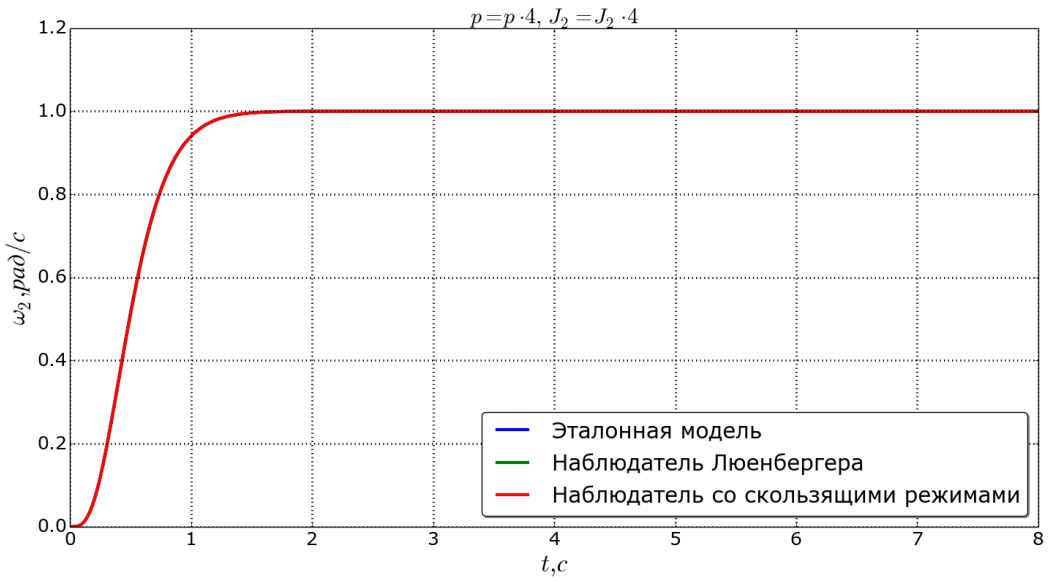

Рис. 8.11. Переходные процессы в системе при увеличении коэффициента упругости связи и увеличении момента инерции второй массы

Как видно из полученных переходных процессов, применение наблюдателя состояния Люенбергера в исследуемой системе даёт близкие результаты к результатам системы использующей наблюдатель состояния со скользящими режимами, однако при увеличении коэффициента упругости связи и уменьшении момента инерции второй массы система, замкнутая по выходу наблюдателя состояния Люенбергера оказывается не работоспособной, в то время как система использующая наблюдатель состояния со скользящими режимами имеет переходный процесс близкий к переходному процессу эталонной модели. Применение наблюдателя состояния со скользящими режимами позволяет сохранить работоспособность адаптивной системы в широком диапазоне изменения параметров объекта управления.

Программа лабораторной работы.

1. Построить наблюдатель состояния Люенбергера полного порядка для двухмассовой системы из лабораторной работы №6, имеющей датчики угловой скорости второй массы и тока двигателя.
2. Исследовать полученную систему при четырёхкратном изменении параметров объекта управления.
3. Построить наблюдатель состояния со скользящими режимами для двухмассовой системы из лабораторной работы №6, имеющей датчики угловой скорости второй массы и тока двигателя.
4. Исследовать полученную систему при четырёхкратном изменении параметров объекта управления.

5. Построить наблюдатель состояния Люенбергера полного порядка для двухмассовой системы из лабораторной работы №6, имеющей датчики угловых скоростей двигателя и рабочего органа (второй массы).
6. Исследовать полученную систему при четырёхкратном изменении параметров объекта управления.
7. Построить наблюдатель состояния со скользящими режимами для двухмассовой системы из лабораторной работы №6, имеющей датчики угловых скоростей двигателя и рабочего органа (второй массы).
8. Исследовать полученную систему при четырёхкратном изменении параметров объекта управления.

Варианты параметров двухмассового объекта такие же как и в лабораторной работе №4

Содержание отчёта.

- Цель работы.
- Теоретические сведения.
- Расчёт наблюдателей состояния Люенбергера полного порядка для двухмассовой системы.
- Исследование двухмассовых систем с наблюдателями состояния Люенбергера.
- Расчёт наблюдателей состояния со скользящими режимами для двухмассовой системы.
- Исследование двухмассовых систем с наблюдателями состояния со скользящими режимами.
- Выводы об эффективности использования наблюдателей состояния Люенбергера и наблюдателей состояния со скользящими режимами в адаптивном управлении двухмассовыми упругими электромеханическими объектами.