МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Нижегородский государственный университет им. Н.И. Лобачевского

А.А. Федюков

Применение средств пакета MATLAB для численного решения задач стабилизации по выходу динамических систем с фазовыми ограничениями

Учебно-методическое пособие

Рекомендовано методической комиссией факультета ВМК для студентов ННГУ, обучающихся по направлению подготовки 010400 «Прикладная математика и информатика».

Нижний Новгород
2014

УДК 681.51+519.711
ББК В3В6
Ф-32

Ф-32 Федюков А.А. ПРИМЕНЕНИЕ СРЕДСТВ ПАКЕТА MATLAB ДЛЯ ЧИСЛЕННОГО РЕШЕНИЯ ЗАДАЧ СТАБИЛИЗАЦИИ ПО ВЫХОДУ ДИНАМИЧЕСКИХ СИСТЕМ С ФАЗОВЫМИ ОГРАНИЧЕНИЯМИ: учебно-методическое пособие. – [электронный ресурс]. – Нижний Новгород: Нижегородский госуниверситет, 2014. – 37 с.

Рецензенты: кандидат физ.-мат. наук **С.Ю. Городецкий**

В учебно-методическом пособии изложен теоретический материал по решению задач стабилизации динамических объектов по измеряемому выходу при фазовых ограничениях. Включен материал о способах описания и решения систем линейных матричных неравенств в пакете для инженерных расчетов MATLAB. Проведено численное решение задачи стабилизации динамическим регулятором по измеряемому выходу однозвенного перевернутого маятника на тележке при наличии фазовых ограничений.

Учебно-методическое пособие предназначено для студентов ННГУ, обучающихся по направлению подготовки 010400 «Прикладная математика и информатика».

Ответственный за выпуск: заместитель председателя методической комиссии факультета ВМК ННГУ, к.т.н., доцент **В.М. Сморкалова**

УДК 681.51+519.711
ББК В3В6

Содержание

Введение

В классической задаче стабилизации предполагают, что состояние динамической системы полностью известно и управление строят в виде обратной связи по состоянию [1-3]. Однако у реальных физических объектов и в реальных ситуациях полная информация о состоянии системы недоступна измерению, а известна лишь часть фазовых переменных или их линейная комбинация. В связи с этим возникает нетривиальная задача стабилизации динамических объектов по измеряемому выходу. Для ее решения применяют динамические регуляторы. Динамический регулятор – это объект, который описывается системой дифференциальных уравнений заданного порядка. Существуют разные способы построения динамических регуляторов, в частности, способ с применением линейных матричных неравенств.

Известно [4], что поиск параметров регулятора сводится к задаче разрешимости системы матричных неравенств. Для случая поиска регулятора полного порядка эта задача принадлежит классу задач выпуклого программирования и в настоящее время разработаны эффективные алгоритмы [4-7], которые позволяют с применением современных программ (например, программ для инженерных расчетов MATLAB) получить параметры регулятора.

Хотя существующие алгоритмы позволяют решать задачу стабилизации линейного объекта, возможна ситуация при которой полученное с помощью этих алгоритмов решение физически не может быть реализовано. Это наводит на мысль, что должно быть наложено некоторое ограничение на фазовые переменные. За необходимость введения ограничений на фазовые переменные объекта говорит и то, что синтез линейных законов управления на основе линейной математической модели управляемого объекта может быть эффективно применен только там, где линейная модель более или менее адекватно описывает реальный объект, т.е. в ограниченной области фазового пространства.

В учебно-методическом пособии рассмотрен подход к синтезу закона управления, который обеспечивает стабилизацию механического объекта с фазовыми ограничениями при неполном измерении фазового состояния системы. Подход основан на применении нового математического аппарата линейных матричных неравенств и методов выпуклой оптимизации. В качестве примера рассмотрена задача стабилизации регулятором полного порядка перевернутого маятника на тележке при наличии ограничений на угол отклонения звена и на смещение тележки. Построена математическая модель объекта. Двумя способами описаны этапы создания системы линейных матричных неравенств в пакете для инженерных расчетов MATLAB, которые необходимы для решения задачи стабилизации. Приведены результаты вычислительных экспериментов.

1. Постановка задачи стабилизации по выходу при фазовых ограничениях

Рассмотрим управляемый объект

$$\dot{x} = Ax + Bu, \qquad x(0) = x_0, \tag{1}$$
$$y = Cx, \quad z_0 = C_0 x, \quad z_i = C_i x,\ i = 1,...,m,$$

где $x \in R^n$ – состояние системы, $u \in R^S$ – управление, $y \in R^l$ – измеряемый выход системы, z_0 – управляемый выход, z_i – выходы, определяющие фазовые ограничения. Требуется построить динамический регулятор вида

$$\begin{cases} \dot{x}_r = A_r x_r + B_r y, \\ u = C_r x_r + D_r y, \end{cases} \tag{2}$$

где $x_r \in R^k$ – состояние регулятора $(k \le n)$, $x_r(0) = 0$, обеспечивающего асимптотическую устойчивость состояния равновесия $\begin{pmatrix} x \\ x_r \end{pmatrix} = 0$ замкнутой системы (1), (2) со степенью устойчивости не меньшей β ($\beta > 0$), при котором для траекторий системы достигается минимальное значение $\gamma_0 > 0$ такое, что

$$\max_{t \ge 0} \mid z_0(t) \mid \le \gamma_0,$$

а также удовлетворяются фазовые ограничения

$$\max_{t \ge 0} \mid z_i(t) \mid \le \gamma_i,\ i = 1,...,m.$$

2. Представление задачи в терминах линейных матричных неравенств

Представим уравнение замкнутой системы (1), (2) в виде

$$\dot{x}_c = A_c x_c, \quad A_c = \begin{pmatrix} A + BD_r C & BC_r \\ B_r C & A_r \end{pmatrix}, \tag{3}$$

$$z_i = \overline{C}_i x_c, \quad \overline{C}_i = (C_i \quad 0), \quad i = 0,1,...,m$$

где $x_c = \begin{pmatrix} x \\ x_r \end{pmatrix}$, а начальные условия $x_c(0) = \begin{pmatrix} x_0 \\ 0 \end{pmatrix}$.

Введем область $E = \{x_c : x_c^T X x_c \le 1\}$, ограниченную эллипсоидом $x_c^T X x_c = 1$, вписанным в область фазового пространства, заданную неравенствами $|z_i(t)| \le \gamma_i$, $i = 0,1,...,m$. Пусть $V(x_c) = x_c^T(t) X x_c(t)$ – квадратичная функция Ляпунова с матрицей $X = X^T > 0$ системы (3) такая, что по любой траектории системы выполнено неравенство

$$\dot{V}(x_c) < -2\beta V. \tag{4}$$

Тогда матрица A_C замкнутой системы (3) является асимптотически устойчивой со степенью устойчивости не меньшей β и все траектории этой системы, выходящие из множества E удовлетворяют фазовым ограничениям $\max_{t \ge 0} |z_i(t)| \le \gamma_i$, $i = 0,1,...,m$.

Введем матрицу $Y = X^{-1}$. Тогда на любой траектории системы (3) условие (4) эквивалентно следующему матричному неравенству:

$$A_C Y + Y A_C^T + 2\beta Y < 0, \; Y > 0.$$

Обобщая результаты, полученные в [8], область в фазовом пространстве, полученную объединением всех множеств E, отвечающих всевозможным функциям Ляпунова вида (4), можно выделить, в терминах линейных матричных неравенств, следующим образом.

Утверждение 1. Если начальное состояние $x_C(0)$ и матрица $Y = Y^T > 0$ удовлетворяют системе линейных матричных неравенств

$$\begin{pmatrix} Y & x_C(0) \\ x_C^T(0) & 1 \end{pmatrix} \ge 0,$$

$$A_C Y + Y A_C^T + 2\beta Y < 0, \tag{5}$$

$$trace(\overline{C}_i Y \overline{C}_i^T) \le \gamma_i^2, \; i = 0,1,...,m,$$

то все траектории системы (3) с начальными условиями $x_C(0) \in E$ удовлетворяют фазовым ограничениям $\max_{t \ge 0} |z_i(t)| \le \gamma_i$, $i = 0,1,...,m$.

Здесь «*trace*» обозначает след матрицы. Доказательство Утверждения 1 приведено в [9].

Введя параметры регулятора

$$\theta = \begin{pmatrix} A_r & B_r \\ C_r & D_r \end{pmatrix}, \qquad (6)$$

представим матрицу замкнутой системы (3) в виде

$$A_C = A_0 + B_0 \theta C_0,$$

где $A_0 = \begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix}$, $B_0 = \begin{pmatrix} 0 & B \\ I & 0 \end{pmatrix}$, $C_0 = \begin{pmatrix} 0 & I \\ C & 0 \end{pmatrix}$.

Здесь символ «I» обозначает единичную матрицу размера $(k \times k)$.

Второе неравенство (5) можно переписать в виде матричного неравенства

$$YA_0^T + A_0 Y + YC_0^T \theta^T B_0^T + B_0 \theta C_0 Y + 2\beta Y < 0. \qquad (7)$$

В этом неравенстве матрицы θ и Y неизвестны, поэтому оно не является линейным относительно совокупности переменных θ и Y. В настоящее время не существует алгоритмов решения таких матричных неравенств. В то же время известны алгоритмы для численного решения линейных матричных неравенств (например, с помощью пакета для инженерных расчетов MATLAB [10]). Если мы зафиксируем θ, то получим линейное матричное неравенство относительно Y. Аналогично, фиксируя Y, получим линейное матричное неравенство относительно неизвестных параметров регулятора θ.

Запишем неравенство (7) в виде

$$\psi + P^T \theta^T Q + Q^T \theta P < 0,$$

где $\psi = YA_0^T + A_0 Y + 2\beta Y$, $P = C_0 Y$ и $Q = B_0^T$.

Известно [4], что это неравенство разрешимо относительно матрицы θ тогда и только тогда, когда разрешимы неравенства

$$\begin{cases} W_{B_0^T}^T (YA_0^T + A_0 Y + 2\beta Y) W_{B_0^T} < 0, \\ W_{C_0 Y}^T (YA_0^T + A_0 Y + 2\beta Y) W_{C_0 Y} < 0, \end{cases} \qquad (8)$$

где $W_{C_0 Y}$ и $W_{B_0^T}$ образуют базисы ядер матриц $C_0 Y$ и B_0^T соответственно.

Заметив, что $W_{C_0 Y}^T = Y^{-1} W_{C_0}$ и, подставив это выражение во второе неравенство (8), преобразуем его к виду

$$W_{C_0}^T (A_0^T X + XA_0 + 2\beta X) W_{C_0} < 0, \quad Y = X^{-1}.$$

3. Алгоритм поиска взаимно-обратных матриц

Таким образом, поиск регулятора (6), который обеспечивает асимптотическую устойчивость замкнутой системы (1), (2) со степенью устойчивости не меньшей β и, при котором, для траектории системы с начальным состоянием x_0 удовлетворяются фазовые ограничения, сводится к разрешимости относительно матриц $X = X^T > 0$ и $Y = Y^T > 0$ системы линейных матричных неравенств

$$\begin{cases} L_1(X,Y) = \begin{pmatrix} Y & x_c(0) \\ x_c^T(0) & 1 \end{pmatrix} \geq 0, \quad L_2(X,Y) = W_{B_0^T}^T (YA_0^T + A_0 Y + 2\beta Y) W_{B_0^T} < 0, \\ L_3(X,Y) = W_{C_0}^T (A_0^T X + XA_0 + 2\beta X) W_{C_0} < 0, \quad trace(\bar{C}_i Y \bar{C}_i^T) \leq \gamma_i^2, \ i = 0,1,...,m \end{cases} \tag{9}$$

в которых $Y = X^{-1}$ (т.е. таких, что $XY = I$), а столбцы матриц W_{C_0} и $W_{B_0^T}$ образуют базисы ядер матриц C_0 и B_0^T соответственно, т.е. $C_0 W_{C_0} = 0$ и $B_0^T W_{B_0^T} = 0$.

Если условия (9) выполнены и такие матрицы найдены, то параметры θ искомого регулятора (6) находятся как решения линейного матричного неравенства (7) относительно переменной θ.

Неравенства (9) являются линейными матричными неравенствами относительно матриц X и Y. Но есть еще одно условие на матрицы, а именно, $Y = X^{-1}$, которое не является линейным и, соответственно, не позволяет легко решать эту задачу специальными программными средствами. Обозначим эту задачу поиска взаимно-обратных матриц X и Y, удовлетворяющих условию (9), как **Задача 1.**

Для ее решения сначала рассмотрим другую задачу.

Задача 2: найти

$$\lambda_{\min} = \min\{\lambda : X - Y^{-1} \leq \lambda I, \ X = X^T > 0, \ Y = Y^T > 0, \ L_1(X,Y) \geq 0,$$
$$L_i(X,Y) < 0, \ i = 2,3, \ L_4(X,Y) \leq 0, \ trace(\bar{C}_i Y \bar{C}_i^T) \leq \gamma_i^2, \ i = 0,1,...,m\},$$

где

$$L_4(X,Y) = \begin{pmatrix} -X & I \\ I & -Y \end{pmatrix}.$$

Дополнительное линейное матричное неравенство $L_4(X,Y) \leq 0$ в силу леммы Шура эквивалентно неравенствам $X \geq 0$ и $X \geq Y^{-1}$. Поэтому в случае, когда в **Задаче 2** $\lambda_{\min} = 0$**,** соответствующие матрицы $X > 0$ и $Y > 0$ являются также решением **Задачи 1**.

Для решения **Задачи 2** требуется минимизировать линейную функцию при ограничениях, одно из которых

$$X - Y^{-1} \le \lambda I \qquad (10)$$

не является выпуклым и, следовательно, не может быть представлено в виде линейного матричного неравенства. Это обстоятельство вновь не позволяет решать **Задачу 2** методами выпуклой оптимизации.

В связи с этим рассмотрим еще одну вспомогательную задачу.

Задача 3: найти

$$\lambda_{\min} = \min\{\lambda : F(X,Y,G_1,G_2) \le \lambda I,\ X = X^T > 0,\ Y = Y^T > 0,\ L_1(X,Y) \ge 0,$$
$$L_i(X,Y) < 0,\ i = 2,3,\ L_4(X,Y) \le 0,\ trace\, C_i Y C_i^T \le \gamma_i^2,\ i = 0,1,...,m\},$$

где

$$F(X,Y,G_1,G_2) = X + Y + 2G_1 + 2G_2 + G_1 Y G_1 + G_2 X G_2,$$

$G_i = G_i^T$, $i = 1,2$, – некоторые заданные матрицы.

В **Задаче 3** по сравнению с **Задачей 2** вместо неравенства (10) стоит линейное матричное неравенство $F(X,Y,G_1,G_2) \le \lambda I$. Представим функцию $F(X,Y,G_1,G_2)$ в виде

$$F(X,Y,G_1,G_2) = (G_1 + Y^{-1})Y(G_1 + Y^{-1}) + (G_2 + X^{-1})X(G_2 + X^{-1}) +$$
$$+ (X - Y^{-1}) + (Y - X^{-1}).$$

Нетрудно видеть, что в силу неравенства $X \ge Y^{-1}$ функция $F(X,Y,G_1,G_2) \ge 0$, и когда матрицы $G_1 = -Y^{-1}$, $G_2 = -X^{-1}$ и величина $\lambda_{\min} = 0$, то соответствующие решения X и Y **Задачи 3** являются и решением **Задачи 2**.

Тогда алгоритм поиска взаимно-обратных матриц можно представить следующим образом:

Алгоритм.

Шаг 1) Полагаем $j = 0$.

Шаг 2) Фиксируем матрицы $G_1 = G_1^{(j)}$ и $G_2 = G_2^{(j)}$.

Шаг 3) Решаем **Задачу 3**, например, с помощью команды mincx пакета MATLAB и находим λ_{j+1}, X_j, Y_j.

Шаг 4) Задаем $G_1^{(j+1)} = -Y_j^{-1}$ и $G_2^{(j+1)} = -X_j^{-1}$.

Шаг 5) Если разность между двумя итерациями $|\lambda_j - \lambda_{j-1}| < \varepsilon$, где ε – некоторое заданное значение, то взаимно-обратные матрицы $X = X^T > 0$ и $Y = Y^T > 0$ считаются найденными с достаточной точностью и **Алгоритм** останавливается. Иначе полагаем $j = j + 1$ и переходим на **Шаг** 2).

Сходимость алгоритма доказана в [9].

При остановке алгоритма возможны две ситуации. В случае, когда $\lambda_* = 0$, $X_* Y_* = I$ и матрицы X_*, Y_* являются решениями **Задачи 1**. Во втором случае, когда $\lambda_* > 0$, нельзя сделать определенного вывода о разрешимости **Задачи 1**. В этом случае целесообразно повторить алгоритм поиска взаимно-обратных

матриц при других начальных условиях $G_1 = G_1^{(0)}$ и $G_2 = G_2^{(0)}$, как это обычно делают в задачах глобальной оптимизации.

В случае поиска регулятора полного порядка, т.е. при $n = k$, поиск стабилизирующего регулятора сводится к решению задачи выпуклого программирования. Он может быть осуществлен следующим образом.

Матрицы X и Y, входящие в линейные матричные неравенства (9), имеют размерность $2n$, где n – размерность матрицы A. В соответствии с блочной структурой матрицы A_0 представим матрицы X и Y в блочном виде

$$X = \begin{pmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{pmatrix}, \quad Y = \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{pmatrix},$$

а матрицы $W_{B_0^T} = \begin{pmatrix} W_{B^T} \\ 0 \end{pmatrix}$, $W_{C_0} = \begin{pmatrix} W_C \\ 0 \end{pmatrix}$.

Тогда неравенства (8) примут вид

$$W_{B^T}^T (Y_{11}A^T + AY_{11} + 2\beta Y_{11}) W_{B^T} < 0, \quad W_C^T (A^T X_{11} + X_{11}A + 2\beta X_{11}) W_C < 0. \qquad (11)$$

Здесь матрицы X_{11} и Y_{11} размера $(n \times n)$ не являются взаимно-обратными, но являются блоками взаимно-обратных матриц.

Известна форма Фробениуса для обращения блочной матрицы. Из нее следует, что при выполнении неравенства $Y_{11} - X_{11}^{-1} = V > 0$, которое по лемме Шура эквивалентно условию

$$\begin{pmatrix} X_{11} & 1 \\ 1 & Y_{11} \end{pmatrix} > 0, \qquad (12)$$

выбор соответствующих Y_{12} и Y_{22} может быть осуществлен следующим образом: $Y_{12} = Y_{22} = V$.

Первое неравенство (9) т.к. $x_c(0) = \begin{pmatrix} x_0 \\ 0 \end{pmatrix}$ будет эквивалентно неравенству

$$\bar{x}_0^T Y^{-1} \bar{x}_0 = \bar{x}_0^T X \bar{x}_0 = x_0^T X_{11} x_0 \le 1. \qquad (13)$$

С учетом структуры матриц $\overline{C_i}$ последние $m+1$ неравенства (9) преобразуются в неравенства

$$trace\left(C_i Y_{11} C_i^T \right) \le \gamma_i^2, \quad i = 0,1,...,m. \qquad (14)$$

Таким образом, для поиска регулятора полного порядка вида (6), который обеспечивает асимптотическую устойчивость замкнутой системы (1), (2) со степенью устойчивости не меньшей β и, при котором, для траектории системы с начальным состоянием x_0 выполнено условие $\max\limits_{t \ge 0} \left| z_0(t) \right| \le \gamma_0$ и удовлетворяются фазовые ограничения $\max\limits_{t \ge 0} \left| z_i(t) \right| \le \gamma_i$, $i = 1,...,m$, необходимо:

1. Решить систему линейных матричных неравенств (11)-(14) относительно матриц $X_{11} = X_{11}^{\ T} > 0$ и $Y_{11} = Y_{11}^{\ T} > 0$ при заданном начальном состоянии x_0;

2. Восстановить матрицу $Y = \begin{pmatrix} Y_{11} & Y_{11} - X_{11}^{\ -1} \\ Y_{11} - X_{11}^{\ -1} & Y_{11} - X_{11}^{\ -1} \end{pmatrix}$;

3. Подставить эту матрицу в неравенство (7) и, решив линейное матричное неравенство относительно неизвестной матрицы θ, найти параметры регулятора (6).

Из **Утверждения 1**. следует, что если для заданного начального состояния x_0 найдены матрицы X и Y, удовлетворяющие неравенствам (9) и построен регулятор, то для всех значений x_0, принадлежащих множеству $E = \{x: \ x^T Xx \le 1\}$, этот же регулятор обеспечивает выполнение для заданного γ_0 условия

$$\max_{t \ge 0} \left| \ z_0(t) \ \right| \le \gamma_0, \qquad (15)$$

а также фазовых ограничений $\max_{t \ge 0} \left| \ z_i(t) \ \right| \le \gamma_i$, $i = 1,..., m$.

Решая задачу минимизации значения γ_0 условия (15) получим минимальное значение γ_0^* при котором матричные неравенства (9) выполнены.

4. Математическая модель перевернутого маятника на тележке

Рассмотрим плоский перевернутый маятник, изображенный на рис. 1. Звено маятника имеет длину l, а масса на конце звена равна m. Угол отклонения звена маятника от вертикали обозначим φ. Тележка имеет массу M и смещается в плоскости качания маятника относительно начала координат на величину x под воздействием внешней силы $U(t)$.

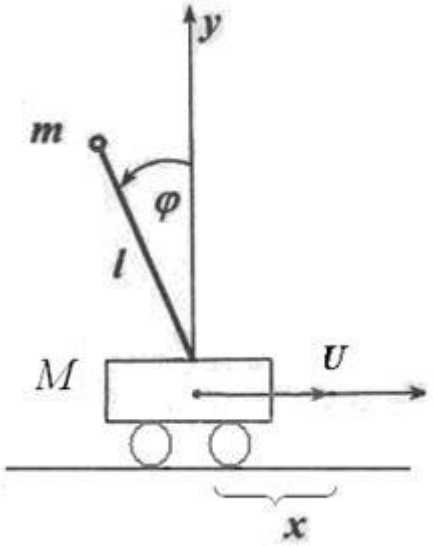

Рис. 1. Перевернутый маятник на тележке

Составим математическую модель плоского однозвенного перевернутого маятника на тележке. Обозначим x_1, y_1 координату массы маятника. Непосредственно из рис. 1 находим, что

$$x_1 = x - l\sin\varphi, \qquad y_1 = l\cos\varphi.$$

Тогда кинетическая T и потенциальная V энергии маятника будут иметь следующий вид:

$$T = \frac{m}{2}[(\dot{x} - l\dot{\varphi}\cos\varphi)^2 + (l\dot{\varphi}\sin\varphi)^2] + \frac{M}{2}\dot{x}^2,$$
$$V = mgl\cos\varphi.$$

Введем функцию Лагранжа $L = T - V$ и напишем уравнения Лагранжа $\frac{d}{dt}\frac{\partial L}{\partial \dot{\varphi}} - \frac{\partial L}{\partial \varphi} = 0$ и $\frac{d}{dt}\frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = U$ для нашей системы:

$$-\ddot{x}\cos\varphi + l\ddot{\varphi} - g\sin\varphi = 0, \qquad (16)$$
$$(m+M)\ddot{x} - ml\ddot{\varphi}\cos\varphi + ml\dot{\varphi}^2\sin\varphi = U.$$

Выражение (16) можно переписать в виде

$$\begin{pmatrix}\ddot{\varphi}\\ \ddot{x}\end{pmatrix} = \frac{1}{\Delta_1}\begin{pmatrix} g(m+M) - ml\dot{\varphi}^2\cos\varphi \\ gml\cos\varphi - ml^2\dot{\varphi}^2\end{pmatrix}\sin\varphi + \frac{1}{\Delta_1}\begin{pmatrix}\cos\varphi \\ l\end{pmatrix}U \qquad (17)$$

где $\Delta_1 = l(m+M) - ml\cos^2\varphi$.

Обозначим $\mu = \frac{m}{M}$. Введем безразмерное время $\bar{t} = \sqrt{\frac{g}{l}}t$ и новые безразмерные величины $x_H = \frac{1}{l}x$, $u = \frac{1}{gM}U$. Тогда уравнения (17), описывающие нелинейную модель движения плоского однозвенного перевернутого маятника на тележке, примут вид

$$\begin{pmatrix} \ddot{\varphi} \\ \ddot{x}_H \end{pmatrix} = \frac{1}{\Delta}\begin{pmatrix} (1+\mu)\sin\varphi - \mu\dot{\varphi}^2\cos\varphi \\ \mu\cos\varphi - \mu\dot{\varphi}^2 \end{pmatrix}\sin\varphi + \frac{1}{\Delta}\begin{pmatrix} \cos\varphi \\ 1 \end{pmatrix}u \qquad (18)$$

где $\Delta = (1+\mu) - \mu\cos^2\varphi$.

Построим линеаризованную модель движения плоского однозвенного перевернутого маятника на тележке. Ограничимся в (18) малым углом φ отклонения звена маятника от вертикали и малой скоростью отклонения звена $\dot{\varphi}$. Получим уравнения:

$$\ddot{\varphi} = (1+\mu)\varphi + u,$$
$$\ddot{x}_H = \mu\varphi + u.$$

Запишем их в каноническом виде (1) управляемой динамической системы

$$\dot{x} = Ax + Bu \qquad (19)$$

где матрицы $A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1+\mu & 0 & 0 & 0 \\ \mu & 0 & 0 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$, состояние системы $x = \begin{pmatrix} \varphi \\ x_H \\ \dot{\varphi} \\ \dot{x}_H \end{pmatrix}$.

Будем считать, что можем измерить только угол φ отклонения звена маятника от вертикали и величину x_H, т.е.

$$y_1 = \varphi,$$
$$y_2 = x_H.$$

Таким образом, измеряемый выход системы $y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$ имеет вид

$$y = Cx \qquad (20)$$

где матрица $C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$.

Из рис. 1. следует, что угол отклонения звена маятника от вертикали φ находится в диапазоне $-\frac{\pi}{2} \le \varphi \le \frac{\pi}{2}$.

Следовательно, выход, определяющий фазовые ограничения $z_1 = C_1 x$ имеет матрицу

$$C_1 = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}.$$

5. Решение линейных матричных неравенств в MATLAB

Линейным матричным неравенством называется неравенство вида:

$$F(x) = F_0 + x_1 F_1 + ... + x_m F_m > 0,$$

в котором $x \in R^m$, $x = (x_1,..., x_m)$ – неизвестная переменная, $F_i = F_i^T \in R^{n \times n}$, $i = 0,..., m$ – заданные действительные симметрические матрицы. Неравенство $F(x) > 0$ обозначает, что матрица в левой части неравенства является положительно определенной, то есть $u^T F(x) u > 0$ для любого ненулевого $u \in R^n$.

В основу численных методов решения линейных матричных неравенств положены методы выпуклой оптимизации. В пакете MATLAB используются так называемые методы внутренней точки, разработанные в [11].

Из множества задач, в которых возникают линейные матричные неравенства можно выделить три основных типа задач, которые в настоящее время эффективно решаются с помощью LMI Control Toolbox пакета MATLAB [10,12].

Первая из них называется *задачей разрешимости*: найти какое-либо решение x для системы заданной в виде линейного матричного неравенства

$$L(x) < R(x). \qquad (21)$$

Если такого решения нет, то задача называется неразрешимой. В пакете MATLAB эту задачу решает функция f*easp* . Попутно функция проверяет, имеет ли система неравенств вообще какие-либо решения. По исходному неравенству (21) составляется вспомогательная задача выпуклого программирования:

$$t \to \min \quad \text{при} \quad L(x) < R(x) + tI .$$

где t – скалярный параметр. С помощью итерационного алгоритма функция f*easp* формирует строго убывающую последовательность приближений t_1, t_2,…. Для того чтобы неравенство (21) было разрешимо относительно x, необходимо и достаточно, чтобы решение t оптимизационной задачи было строго отрицательным.

Вызов функции f*easp* имеет вид:

$$[\text{tmin}, xfeas] = \text{f}easp(lmisys, options, \text{target}).$$

Ее выходные значения: tmin – минимальное найденное значение параметра t и *xfeas* – оптимальное значение вектора x. Чтобы получить из него матричные переменные, следует применить описанную ниже функцию *dec2mat* . Параметр *lmisys* содержит описание системы матричных неравенств (внутреннее представление) в виде одномерного массива. Параметр *options* представляет собой описание параметров алгоритма оптимизации. Параметр target определяет условие окончания итерационного процесса. Вычисления прекращаются, когда $t < \text{target}$. По умолчанию $\text{target} = 0$.

Вторая важная задача, которая эффективно решается с использованием линейных матричных неравенств – это *задача оптимизации с линейными*

матричными ограничениями: минимизировать функцию $c^T x$ при ограничениях $L(x) < R(x)$.

В пакете MATLAB для этого используется команда

$$[copt, xopt] = \text{mincx}(lmisys, c, options, xinit, \text{target}).$$

Здесь *lmisys* – внутреннее описание системы матричных неравенств, задающей ограничение $L(x) < R(x)$, c – вектор-столбец того же размера, что и x. Как и для функции f*easp* параметр *options* – описание параметров алгоритма оптимизации. Параметр *xinit* – задает начальное приближение оптимальной точки x. Если начальное приближение не задано, его можно положить пустым. Если этот параметр не удовлетворяет системе ограничений, то он игнорируется. Параметр target – необязательный параметр, который определяет правило остановки вычислений. Итерации прекращаются, если найдено значение x, которое удовлетворяет ограничениям $L(x) < R(x)$ и такое, что $c^T x < \text{target}$. По умолчанию $\text{target} = -10^{20}$.

Функция mincx возвращает два значения: *copt* – глобальный минимум целевой функции $c^T x$ и *xopt* – оптимальное значение вектора скалярных переменных x. Для того, чтобы извлечь из него матричные переменные требуется применить функцию *dec2mat*. В случае, если система ограничений не имеет решения (множество допустимых переменных пусто), оба выходных параметра принимают пустое значение [ ].

Третья задача – это *задача на обобщенное собственное значение*, т.е. задача найти минимальное $\lambda \in R$, для которого $\begin{cases} A(x) < \lambda B(x), \\ B(x) > 0, \\ C(x) < 0, \end{cases}$

где $A(x)$, $B(x)$, $C(x)$ – заданные симметричные матрицы зависящие от вектора скалярных параметров x. В частном случае, когда $B(x) = I$, неравенство $A(x) < \lambda I$ дает оценку максимального собственного значения симметричной матрицы $A(x)$.

В пакете MATLAB вызов функции для оценки обобщенных собственных чисел выглядит следующим образом:

$$[\lambda\min, xopt] = gevp(lmisys, nlfc, options, \lambda_0, x_0, \text{target}).$$

Здесь *lmisys* содержит внутреннее описание системы неравенств, задающих ограничения, параметр *nlfc* – число ограничений, задаваемых линейными матричными неравенствами; *options* – описание параметров алгоритма оптимизации; необязательные параметры λ_0 и x_0 задают начальные приближения для $\lambda\min$ и *xopt* соответственно (игнорируются, если не удовлетворяют системе ограничений); target задает правило остановки – вычисления прекращаются, если $\lambda < \text{target}$ (по умолчанию $\text{target} = -10^5$).

Функция возвращает значения: $\lambda\min$ – минимум функции λ и *xopt* – оптимальное значение вектора скалярных переменных x. Для того, чтобы извлечь из него матричные переменные, требуется применить функцию *dec2mat*.

Функции *feasp*, mincx, *gevp* возвращают результат своей работы в виде массива скалярных переменных. Функция *dec2mat* позволяет извлечь из него значения матричных переменных. Вызов функции имеет вид:

$$X = dec2mat(lmisys, decvar\,s, k).$$

Здесь *lmisys* содержит внутреннее описание системы неравенств, $decvar\,s$ – массив (вектор) скалярных переменных, k – номер извлекаемой матричной переменной. Функция возвращает искомую переменную в виде матрицы. Пример использования функции *dec2mat* будет приведен ниже.

Следующая функция, реализованная в пакете MATLAB, решает частную задачу исследования линейного матричного неравенства

$$\psi + P^T\theta^T Q + Q^T\theta P < 0 \qquad (22)$$

которое используется выше. Здесь ψ, P, Q - заданные матрицы, причем ψ – симметричная матрица размера $(n \times n)$, P и Q – прямоугольные матрицы порядков $(l \times n)$ и $(k \times n)$ соответственно, θ – неизвестная прямоугольная матрица размера $(k \times l)$. Известно [4], что это неравенство разрешимо тогда и только тогда, когда

$$W_P^T \psi\ W_P < 0, \quad W_Q^T \psi\ W_Q < 0,$$

где столбцы матриц W_P и W_Q образуют базисы ядер матриц P и Q соответственно.

В первом частном случае, когда $rank\ P = n$, а $rank\ Q < n$ линейное матричное неравенство (22) разрешимо относительно матрицы θ тогда и только тогда, когда

$$W_Q^T \psi\ W_Q < 0,$$

где столбцы матрицы W_Q образуют базис ядра матрицы Q.

Во втором частном случае, когда $rank\ P < n$, а $rank\ Q = n$ линейное матричное неравенство (22) разрешимо относительно матрицы θ тогда и только тогда, когда

$$W_P^T \psi\ W_P < 0,$$

где столбцы матрицы W_P образуют базис ядра матрицы P.

Вызов функции имеет вид:

$$X = basiclmi(\psi, Q, P, option).$$

Если неравенство неразрешимо, то выходное значение $X = [\ ]$. Параметр *option* не обязательный. Если положить его равным $'X\,\min'$, то ищется значение X с наименьшей Фробениусовой нормой.

Все детали относительно использования этих и других команд в LMI Control Toolbox содержаться в [10,12].

6. Создание системы линейных матричных неравенств в MATLAB в виде программного кода

Создавать линейные матричные неравенства в MATLAB можно двумя разными способами: с помощью программы и используя графический пользовательский интерфейс. Рассмотрим первый способ.

Формирование системы линейных матричных неравенств в виде программного кода состоит в последовательном вызове таких функций как *setlmis*, *lmivar*, *lmiterm* и *getlmis*.

Перед тем, как начать описание новой системы линейных матричных неравенств с помощью функций *lmivar* и *lmiterm* применяют функцию

$$setlmis([])$$

для инициализации внутреннего представления системы. Это связано с тем, что большинство функций пакета LMILab, входящего в более крупный пакет LMI Control Toolbox пакета MATLAB, работает с внутренним представлением системы матричных неравенств. Именно это внутреннее представление и используется в данном случае. Для случая, когда добавление новых переменных и условий с помощью функции *lmivar* и *lmiterm* происходит в уже существующую систему линейных матричных неравенств, используют синтаксис

$$setlmis(lmisys),$$

где *lmisys* – внутреннее представление системы линейных матричных неравенств.

Функция *lmivar* служит для задания переменных в системе линейных матричных неравенств. Вызов функции имеет вид

$$X = \mathrm{lmivar}(\mathrm{type},\mathrm{struct}),$$

или в общем виде

$$[X, n, sX] = \mathrm{lmivar}(\mathrm{type},\mathrm{struct}).$$

Здесь X – новая матричная переменная, идентификатор, которой может быть использован для последующего использования этой матричной переменной. Первый аргумент функции *type* выбирается среди доступных типов переменных, второй аргумент *struct* дает дополнительную информацию о структуре X в зависимости от ее типа. Доступные типы переменных включают в себя:

$type=1$ – симметрические матрицы с блочно-диагональной структурой. Каждый диагональный блок может быть произвольной симметричной матрицей, скаляром, умноженным на единичную матрицу или нулевой матрицей. Если матричная переменная X имеет m диагональных блоков, то аргумент *struct* представляет собой матрицу размера $m \times 2$, где первое число каждой строки задает размер блока, а второе тип блока (1 – произвольная симметричная матрица, 0 – скаляр, умноженный на единичную матрицу, -1 – нулевая матрица).

$type=2$ – прямоугольная размера $m\times n$ матрица. В данном случае $struct=[m \quad n]$.

$type=3$ – используется для задания матриц другой структуры.

В качестве примера применения $type=1$ и $type=2$ рассмотрим создание матричных переменных X_1, X_2 и X_3 каких что

X_1 – симметричная матрица размера 3×3,

X_2 – прямоугольная матрица размера 2×4,

$$X_3=\begin{pmatrix}\Delta & 0 & 0\\ 0 & \delta_1 & 0\\ 0 & 0 & \delta_2 I\end{pmatrix},$$

где Δ – произвольная симметричная матрица размера 5×5, δ_1 и δ_2 – скаляры, а I – единичная матрица размера 2×2. Тогда следующие команды в MATLAB будут определять эти переменные.

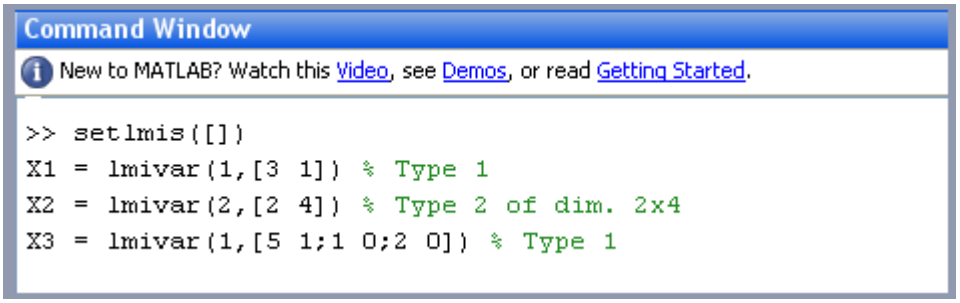

Рис. 2. Вид Command Window

В качестве примера применения $type=3$ создадим матрицу вида

$$X=\begin{pmatrix}X_1 & 0\\ 0 & X_2\end{pmatrix},$$

где

X_1 – прямоугольная матрица размера 2×3,

X_2 – прямоугольная матрица размера 3×2.

Определим матрицы X_1 и X_2 как

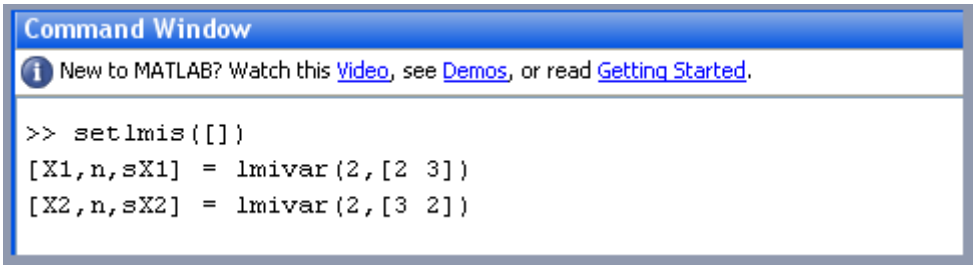

Рис. 3. Вид Command Window

При этом структура матриц имеет следующий вид

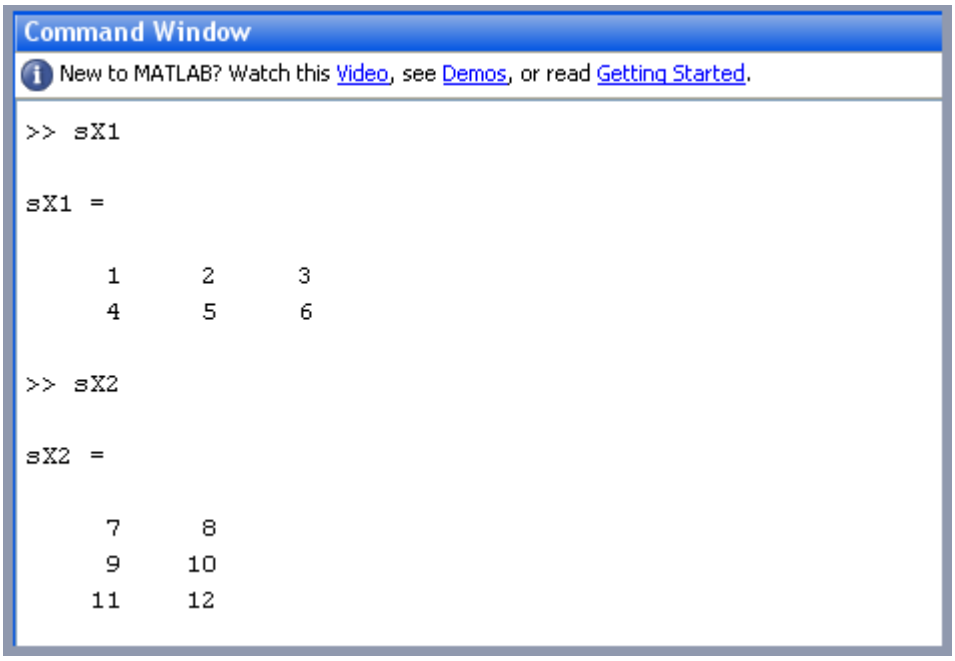

Рис. 4. Вид Command Window

Выходы sX1 и sX2 дают положение решения для переменных X1 и X2. Например, равенство элемента sX2(1,1)=7 означает, что элемент X2(1,1) расположен в седьмом компоненте вектора решения.

Для задания матрицы X применим команду

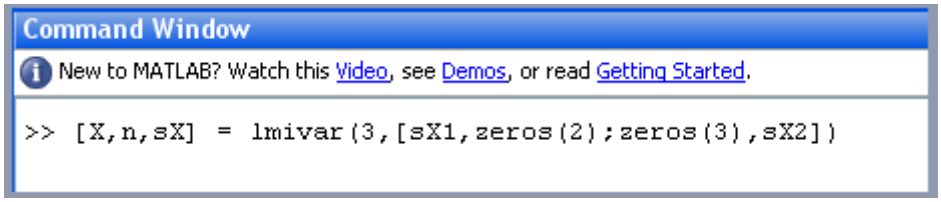

Рис. 5. Вид Command Window

Тогда выход sX будет иметь требуемый вид

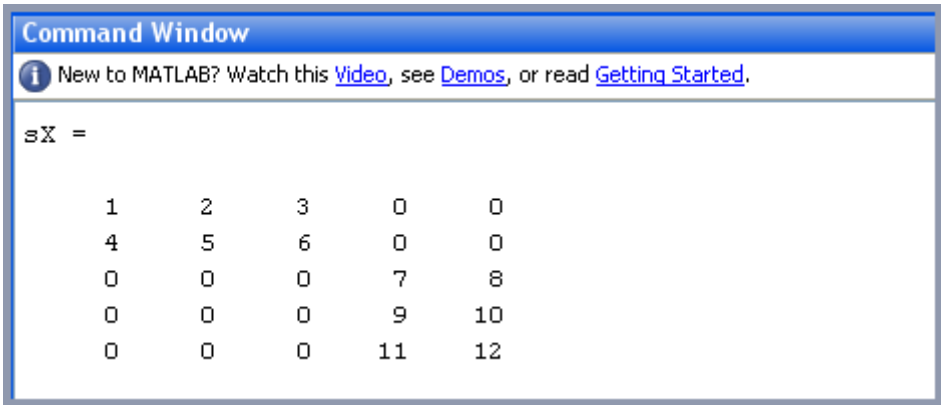

Рис. 6. Вид Command Window

Функция *lmiterm* позволяет добавить один элемент в систему матричных неравенств. Перед использованием *lmiterm* описание системы матричных неравенств должно быть инициализировано с помощью функции *setlmis*, а матричные переменные системы должны быть объявлены с помощью функции *lmi*var .

Вызов функции *lmi*var имеет вид:

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

Здесь *termID* представляет собой вектор, состоящий из четырех компонент. Первая компонента вектора определяет номер линейного матричного неравенства, в которое добавляем элемент, а знак компоненты указывает на расположение добавляемого элемента слева (число отрицательно) или справа (число положительно) в данном матричном неравенстве. Вторая и третья компонента *termID* определяют номер блока матричного неравенства, в который добавляем элемент. Четвертая компонента определяет тип добавляемого элемента: постоянная матрица, переменная матрица X в блоке вида AXB или переменная матрица X в блоке вида AX^TB. Матрицы A и B указываются в параметрах *termID*. Параметр flag – необязательный параметр, позволяющий с учетом структуры диагональных блоков упрощать описание, используя одну команду *lmi*var .

Например, описание левой части линейного матричного неравенства

$$\begin{pmatrix} 2AX_2A^T - x_3E + DD^T & B^TX_1 \\ X_1^TB & -I \end{pmatrix} < M^T \begin{pmatrix} CX_1C^T + CX_1^TC^T & 0 \\ 0 & -fX_2 \end{pmatrix} M$$

имеет вид

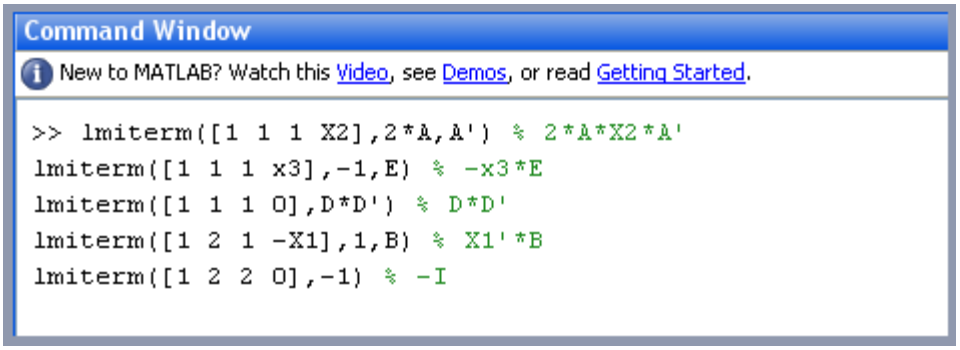

Рис. 7. Вид Command Window

Описание правой части неравенства имеет вид

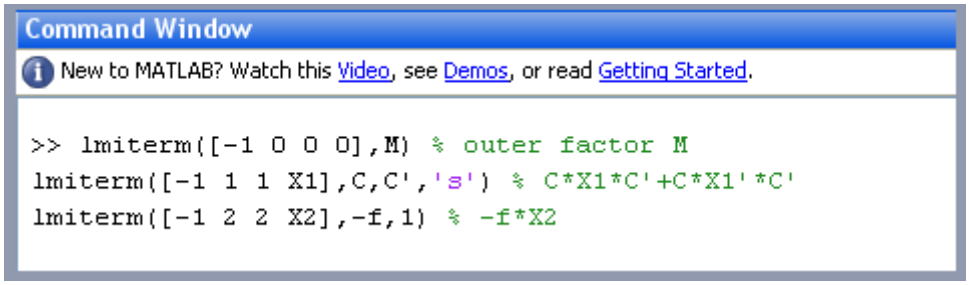

Рис. 8. Вид Command Window

Функция *getlmis* служит для внешнего описания системы линейных матричных неравенств. После завершения описания системы линейных матричных неравенств с помощью функций *lmivar* и *lmiterm*, внутреннее представление системы *lmisys* получают с помощью команды

$$lmisys = getlmis.$$

Это внутреннее описание системы матричных неравенств *lmisys* может быть в дальнейшем использовано в MATLAB для решения задачи разрешимости линейных матричных неравенств, задачи оптимизации с линейными матричными ограничениями, задачи на обобщенное собственное значение или может использоваться для последующей обработки.

Теперь рассмотрим второй способ создания в MATLAB системы линейных матричных неравенств

7. Создание системы линейных матричных неравенств в MATLAB с использованием графического пользовательского интерфейса LMIEDIT

Удобным средством задания линейных матричных неравенств является встроенный в MATLAB графический пользовательский интерфейс LMIEDIT, который позволяет представить матричные неравенства в символьном виде. Набрав в командной строке MATLAB «lmiedit» пользователь вызывает окно с двумя редактируемыми текстовыми полями и с набором кнопок. В верхней части окна указывается название системы линейных матричных неравенств.

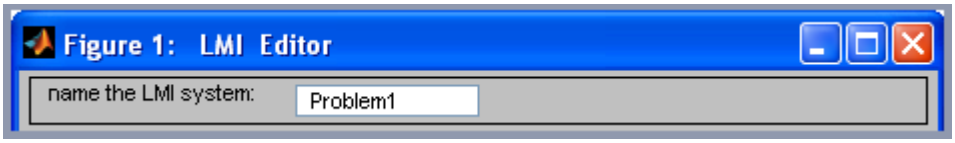

Рис. 9. Окно для задания названия системы линейных матричных неравенств в LMIEDIT

Первое редактируемое текстовое поле состоит из трех частей. В первой части указываются имена матричных переменных. Во второй части первого текстового окна указывается тип матричных переменных. Он может быть трех видов S, R и G. Тип S указывается для симметричных блок диагональных матриц, тип R для неструктурированных прямоугольных матриц и тип G используется для матриц других структур. В третьей части первого текстового окна указывается дополнительная структура матрицы. Например, описание вида

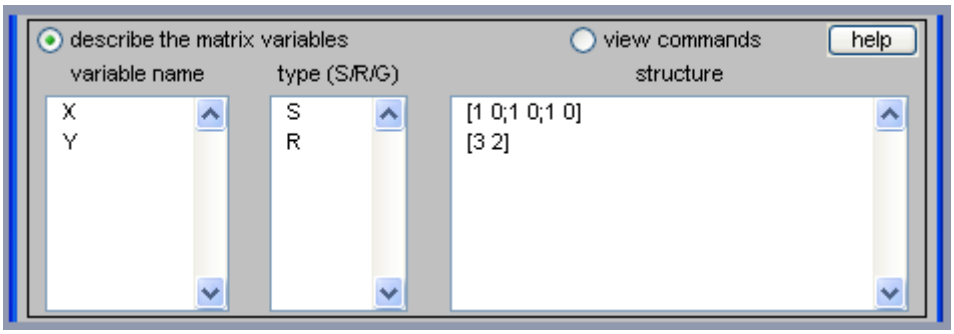

Рис. 10. Первое редактируемое текстовое поле LMIEDIT

задает 3×3 диагональную матрицу X и 3×2 прямоугольную матрицу Y.

Во втором редактируемом текстовом поле указываются линейные матричные неравенства в символьном виде, например

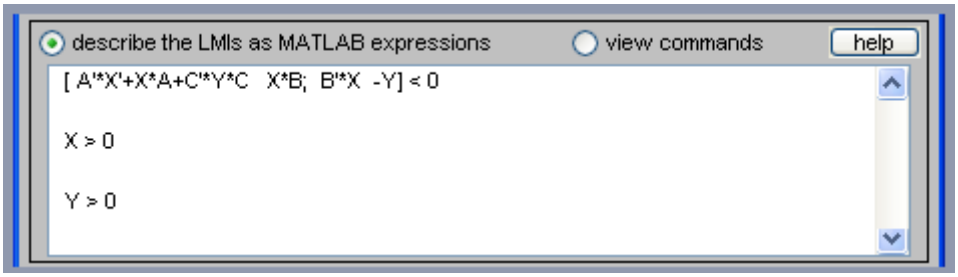

Рис. 11. Второе редактируемое текстовое поле LMIEDIT

После того как линейные матричные неравенства полностью определены, можно представить их в виде последовательности команд языка MATLAB с помощью нажатия на соответствующую кнопку «view commands»

Расположенные в нижней части графического пользовательского интерфейса LMIEDIT кнопки позволяют сохранить описание линейных матричных неравенств в символьном виде с помощью кнопка «Save», загрузить символьное описание помощью кнопка «Load», прочитать последовательность команд на языке MATLAB с помощью кнопки «Read» или сохранить их с помощью кнопки «Write». Кнопка «Create» служит для создания внутреннего представления системы линейных матричных неравенств в MATLAB. Результат записывается в массив с таким же именем, как и имя линейных матричных неравенств. Полученный массив может быть передан другим функциям пакета. Кнопка «Clear all» позволяет очистить все поля графического пользовательского интерфейса LMIEDIT, а кнопка «Close» закрывает его.

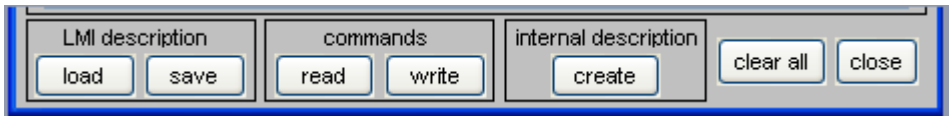

Рис. 12 Нижняя часть графического пользовательского интерфейса LMIEDIT

8. Пример работы с матричными неравенствами в MATLAB

В качестве примера работы с матричными неравенствами в MATLAB решим описанную выше задачу синтеза динамического регулятора полного порядка, обеспечивающего стабилизацию перевернутого маятника на тележке при фазовых ограничениях.

Построим динамический регулятор полного порядка вида (2) для объекта (19), (20), обеспечивающий асимптотическую устойчивость замкнутой системы со степенью устойчивости не меньшей $\beta = 0.1$, при котором для траекторий системы достигается значение $\gamma_0 = 0.5$ такое, что

$$\max_{t \geq 0} \left| x_H(t) \right| \leq \gamma_0,$$

а также удовлетворяется фазовое ограничение

$$\max_{t \geq 0} \left| \varphi_1(t) \right| \leq \pi / 60.$$

Шаг 1. Запустим программу MATLAB в операционной системе WINDOWS, нажав для этого соответствующую иконку 

на рабочем столе. Введем исходные данные. Пусть параметр $\mu = 0.1$. Зададим матрицы A, B, описывающие модель перевернутого маятника на тележке (19), матрицу C, описывающую измеряемый выход системы (20). Для этого наберем матрицы в командном окне MATLAB и нажмем клавишу «Enter».

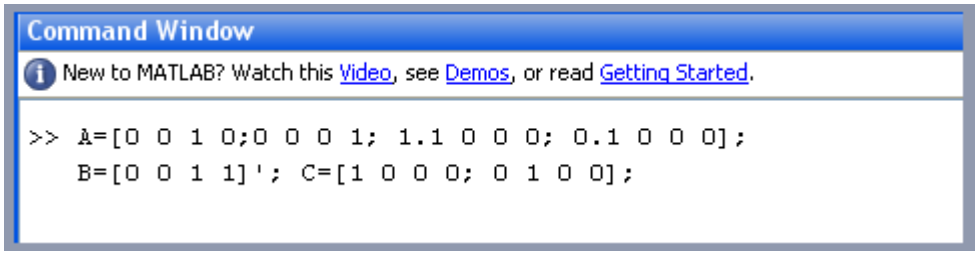

Рис. 13. Вид Command Window на первом шаге

Команда «whos» в MATLAB позволяет выводить на экран информацию об именах матричных переменных, их размерности и типе. Для просмотра матричной переменной в командном окне MATLAB достаточно набрать ее имя.

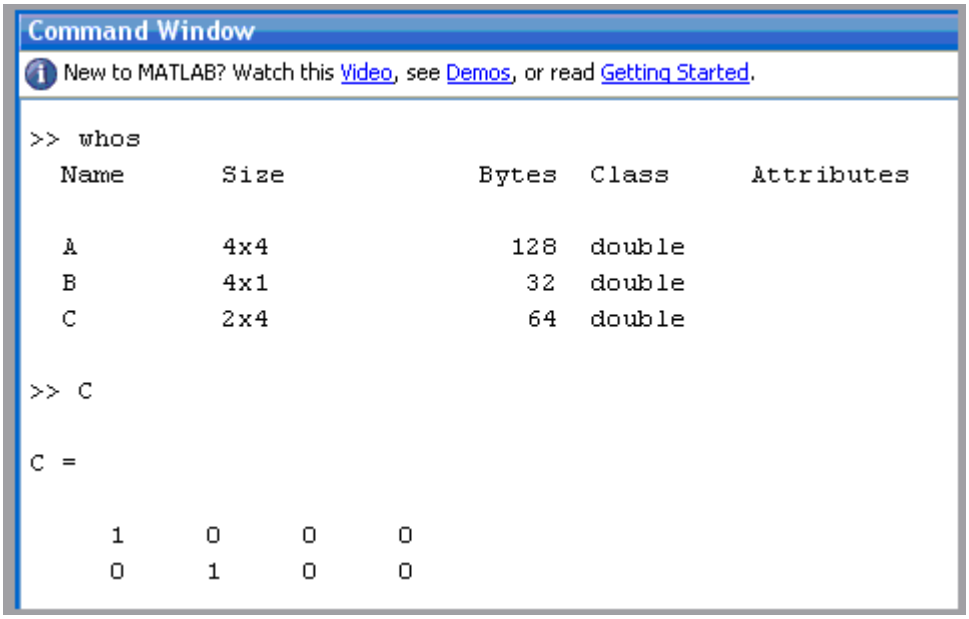

Рис. 14. Вид Command Window

Аналогично зададим матрицы C_0 и C_1,, соответственно для управляемого выхода $z_0 = C_0 x$ и для выхода $z_1 = C_1 x$, определяющего фазовое ограничение.

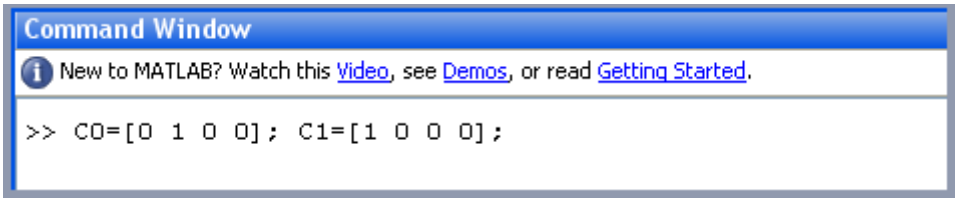

Рис. 15. Вид Command Window

Получим матрицы W_C и W_{B^T}, входящие в матричные неравенства (11) и задающие ноль пространство матриц C и B^T соответственно. Для этого воспользуемся командой «null» пакета MATLAB.

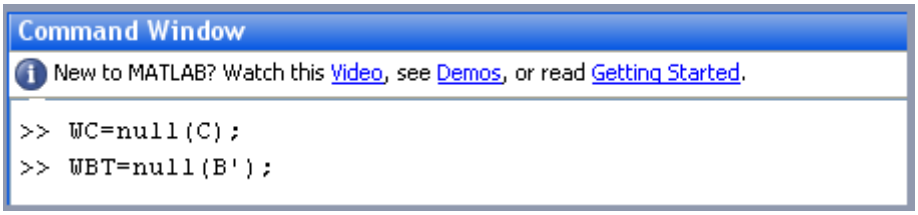

Рис. 16. Вид Command Window

Пусть начальное отклонение звена маятника $\varphi = 0.035$. Тогда входящая в матричное неравенство (13) величина x_0 имеет вид

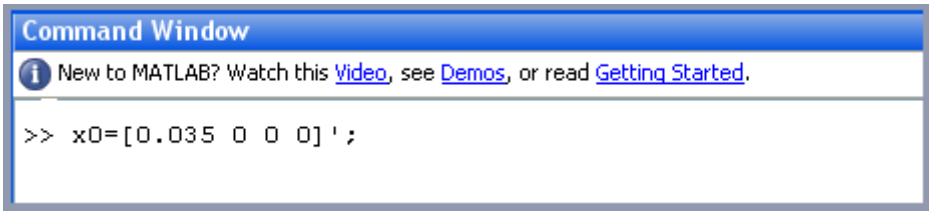

Рис. 17. Вид Command Window

Зададим требуемую степень устойчивости замкнутой системы

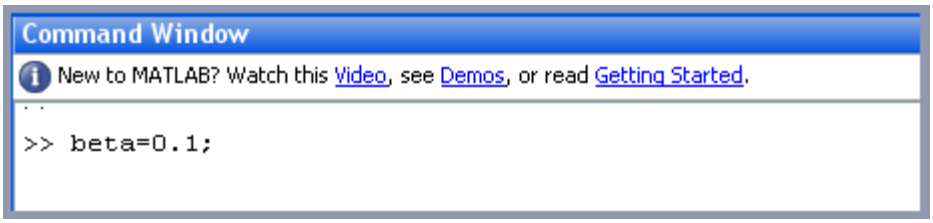

Рис. 18. Вид Command Window

Шаг 2. Запустим графический пользовательский интерфейс LMIEDIT, набрав в командной строке MATLAB «lmiedit». Введем название системы линейных матричных неравенств, В текстовых полях укажем имена, тип матричных переменных и линейные матричные неравенства (11)-(14) в символьном виде как это показано на рис. 19.

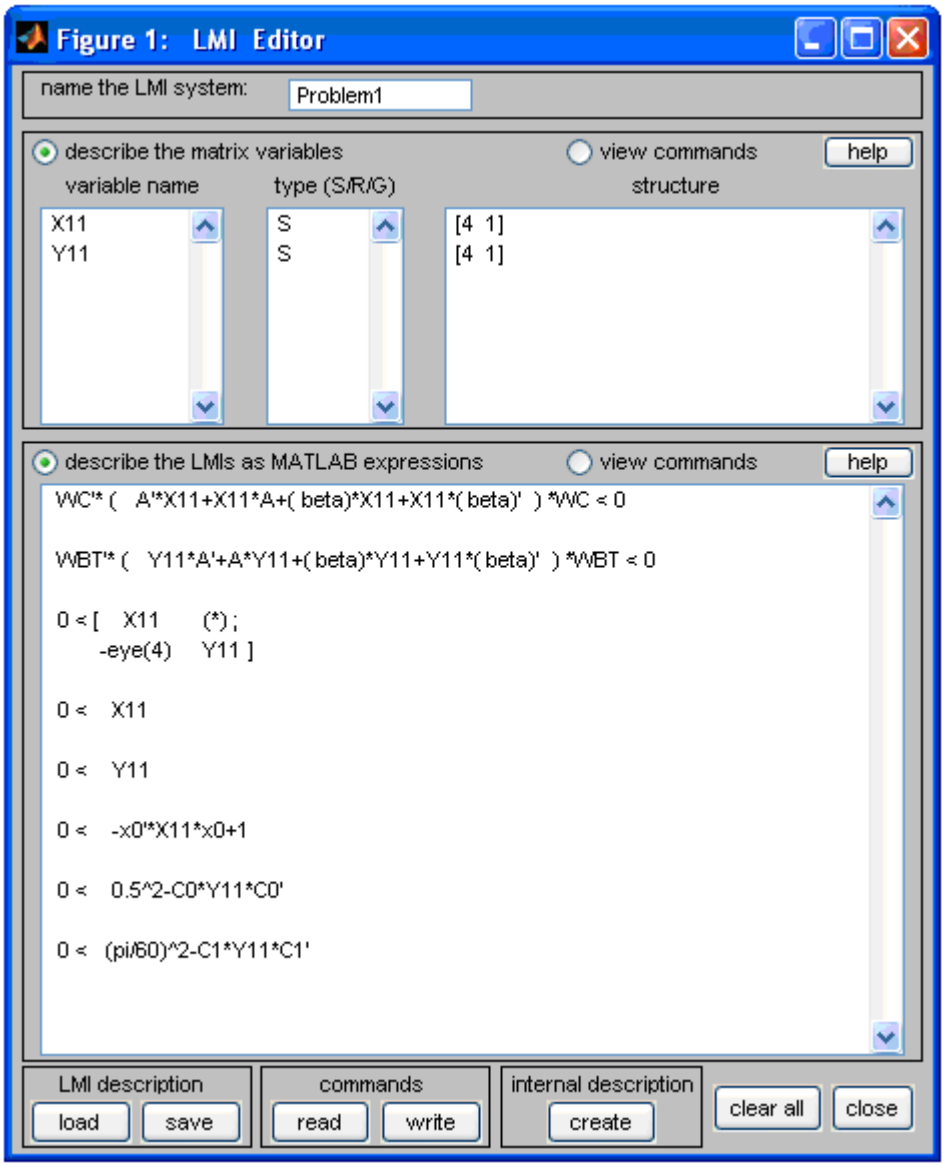

Рис. 19. Символьное представление линейных матричных неравенств в LMIEDIT на втором шаге

Шаг 3. Переведем линейные матричные неравенства из символьного представления в последовательность команд языка MATLAB нажатием на кнопку «view commands»

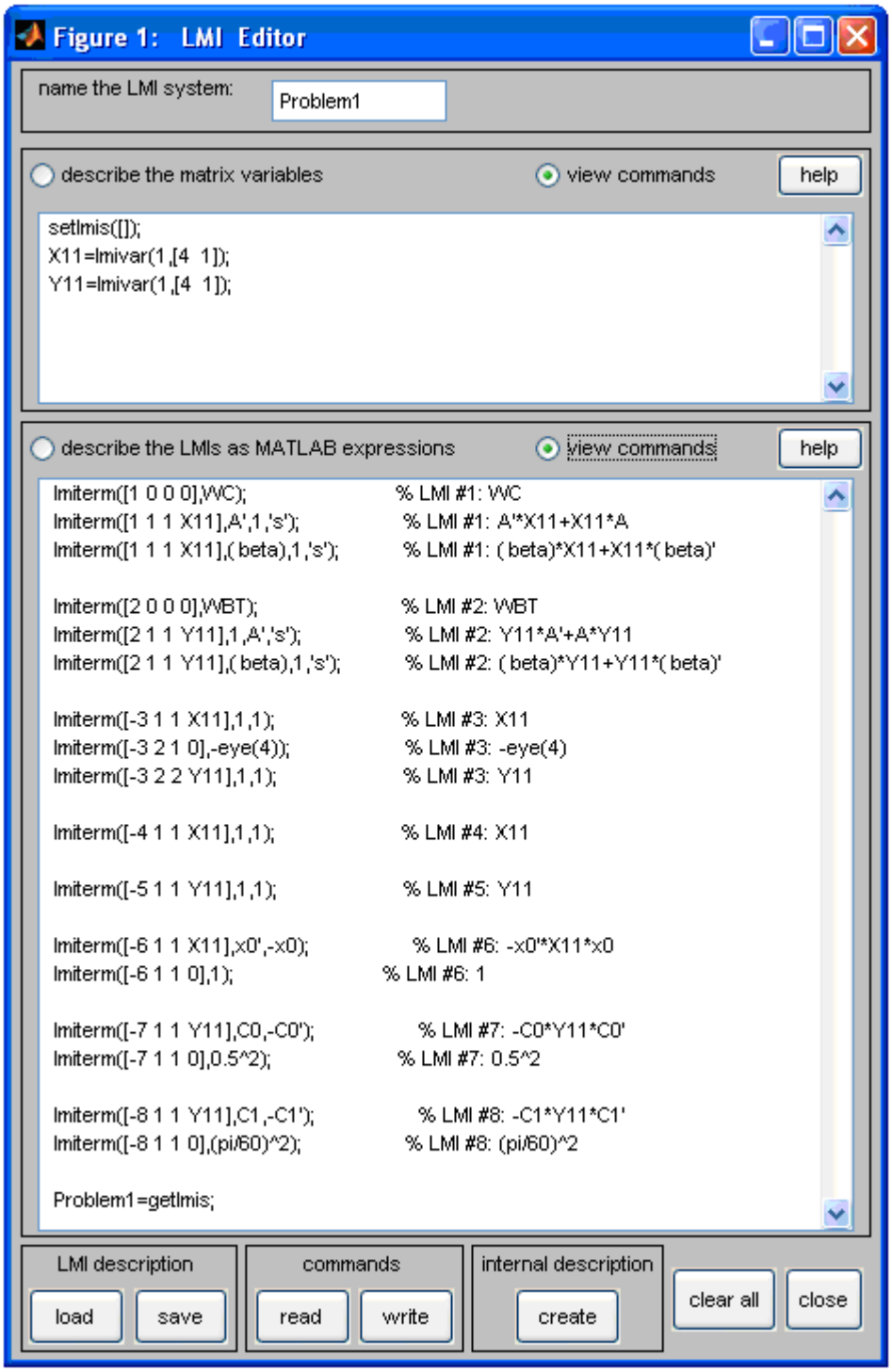

Рис. 20. Вид графического пользовательского интерфейса LMIEDIT на третьем шаге

Нажатие кнопки «Create» равносильно выполнению в командном окне MATLAB следующих команд:

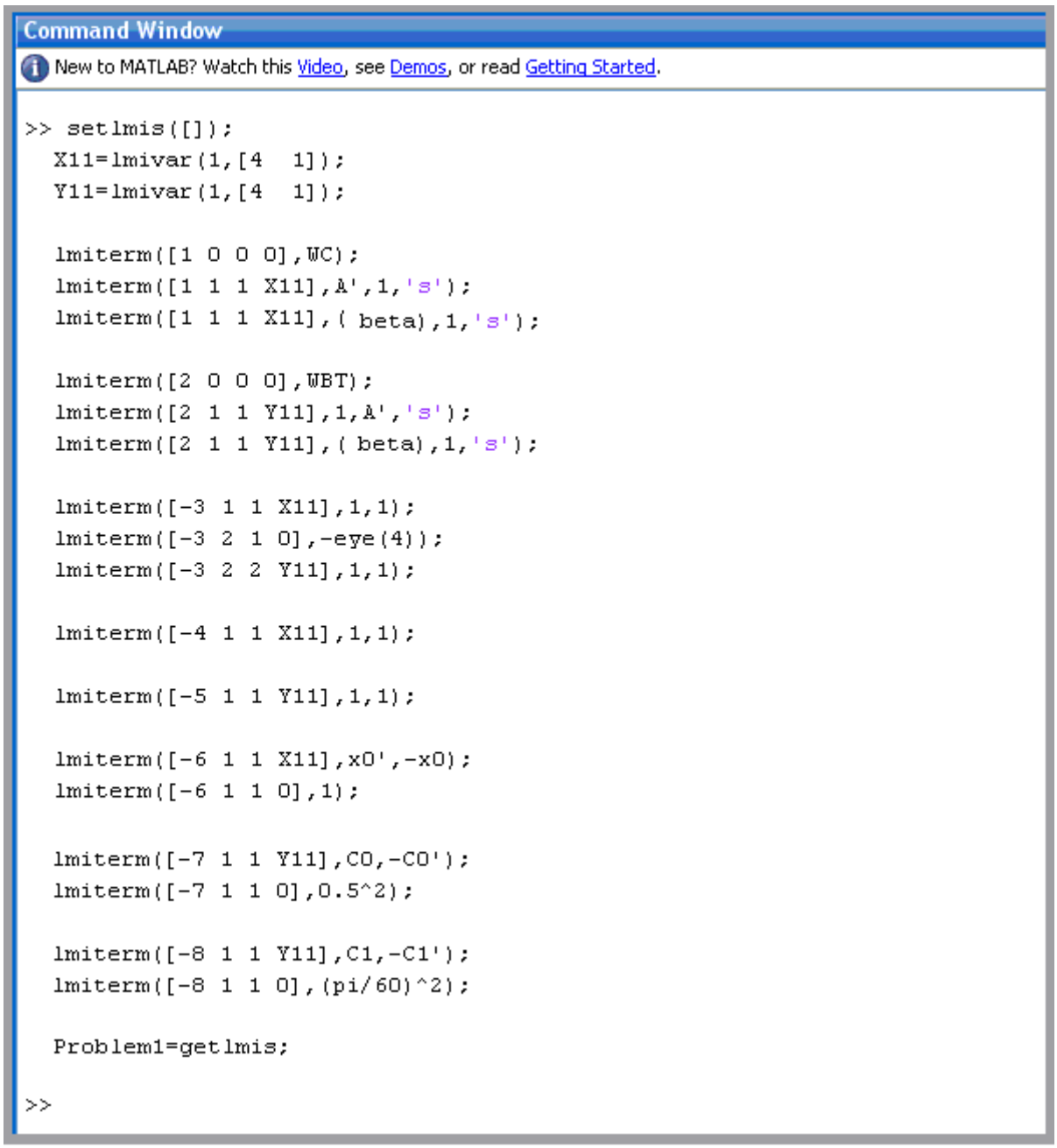

Рис. 21. Программное задание линейных матричных неравенств в Command Window на третьем шаге

Шаг 4. Решим систему линейных матричных неравенств, используя функцию *feasp* .

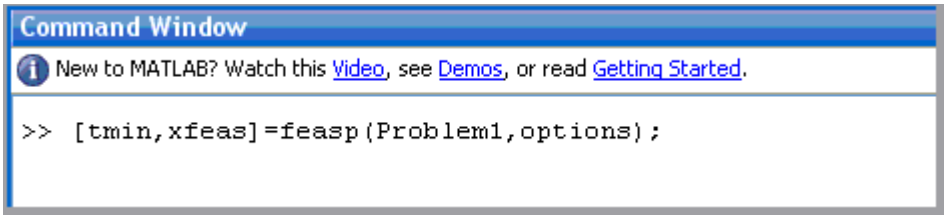

Рис. 22. Вид Command Window на четвертом шаге

Шаг 5. Получим значение матриц X_{11} и Y_{11}

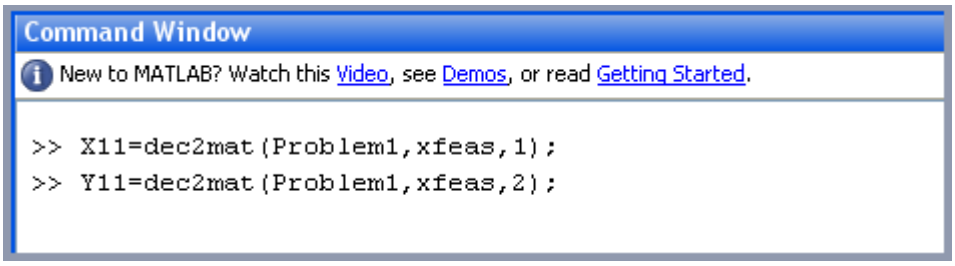

Рис. 23. Вид Command Window на пятом шаге

Шаг 6. Восстановим матрицу $Y = \begin{pmatrix} Y_{11} & Y_{11} - X_{11}^{-1} \\ Y_{11} - X_{11}^{-1} & Y_{11} - X_{11}^{-1} \end{pmatrix}$ и матрицу $X = Y^{-1}$.

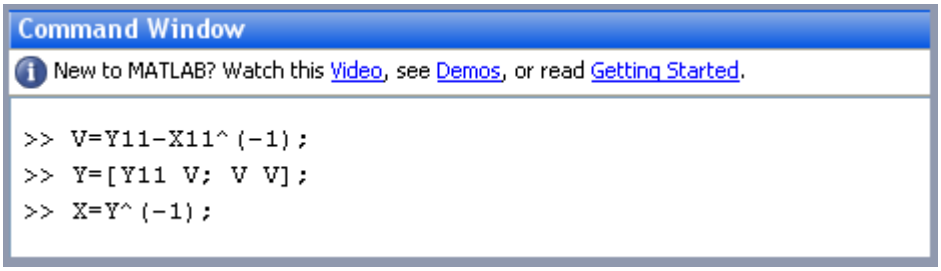

Рис. 24. Вид Command Window на шестом шаге

На рис. 25 приведены полученные взаимно обратные матрицы X и Y, которые обеспечивают разрешимость матричных неравенств (9).

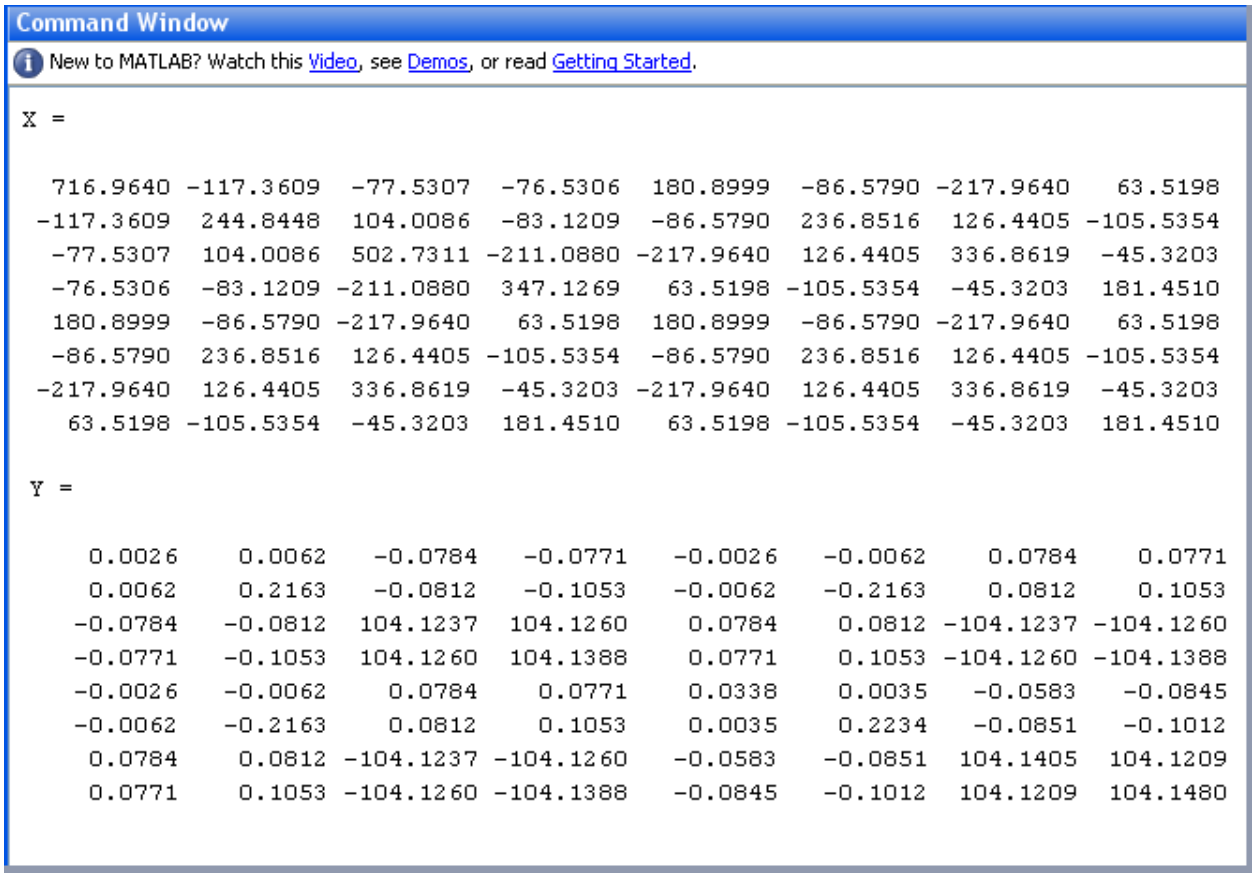

Рис. 25. Взаимно обратные матрицы X и Y, полученные на шестом шаге

Шаг 7. Подставим матрицу Y в неравенство (7) и, решив линейное матричное неравенство относительно неизвестной матрицы θ, найдем параметры регулятора (6).

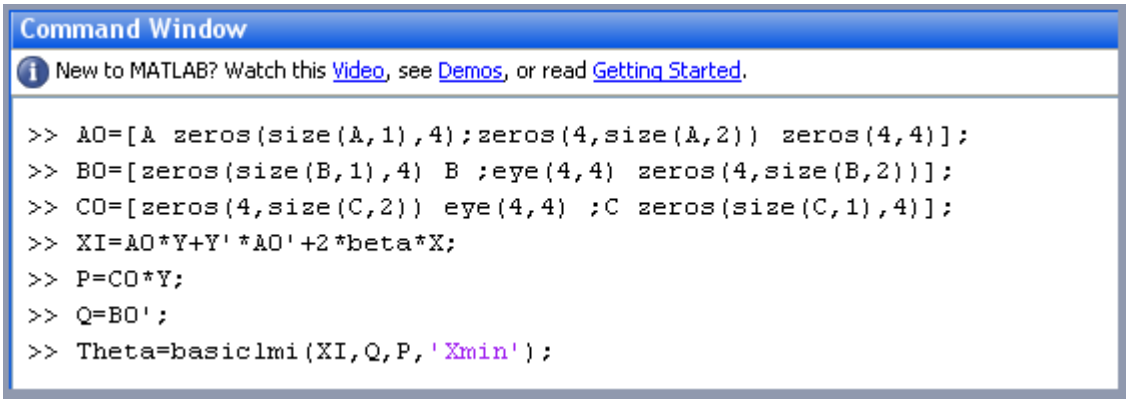

Рис. 26. Вид Command Window на седьмом шаге

Для матрицы Y, полученной выше, найдена следующая матрица динамический регулятор полного порядка (6) который гарантирует стабилизацию объекта при фазовых ограничениях

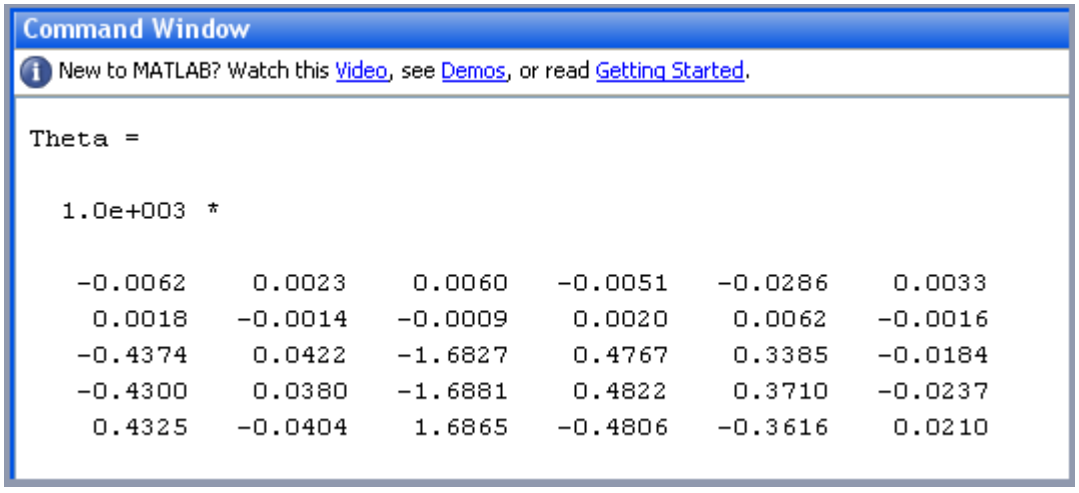

Рис. 27. Параметры регулятора $Theta$, полученные на седьмом шаге

Замечание 1. Численное решение в пакете MATLAB линейных матричных неравенств (11)-(14) и последующее восстановление матрицы Y, может привести к ситуации, когда матрицы Y или $X = Y^{-1}$ имеют собственные числа близкие к мнимой оси. В этом случае возможна ситуация, когда неравенство (22)

$$\psi + P^T \theta^T Q + Q^T \theta P < 0,$$

где $\psi = YA_0^T + A_0 Y + 2\beta Y$, $P = C_0 Y$ и $Q = B_0^T$ неразрешимо. Поскольку данное неравенство разрешимо тогда и только тогда, когда

$$W_P^T \psi\ W_P < 0, \quad W_Q^T \psi\ W_Q < 0, \tag{23}$$

где столбцы матриц W_P и W_Q образуют базисы ядер матриц P и Q соответственно, то можно предложить следующий вариант решения этой вычислительной проблемы. Решим линейные матричные неравенства

$$W_P^T (\psi + \varepsilon_1 I)\ W_P < 0, \quad W_Q^T (\psi + \varepsilon_1 I)\ W_Q < 0, \tag{24}$$

где $\varepsilon_1 > 0$ – некоторое заданное значение, символ «I» обозначает единичную матрицу соответствующего размера. Тогда матрица ψ являющаяся решением неравенства (24) так же будет являться и решением неравенства (23). Для нее справедливо, что

$$\psi + P^T \theta^T Q + Q^T \theta P < -\varepsilon_1 I .$$

Это означает, что все собственные числа матрицы $\psi + P^T \theta^T Q + Q^T \theta P$ будут меньше, чем $-\varepsilon_1$, а значит, будет выполнено неравенство (22).

Шаг 8. Матрица $X = Y^{-1}$ определяет область в 8-и мерном фазовом пространстве вида $E = \{x:\ x^T X x \le 1\}$, при выборе в любой точке которой начального состояния и, используя регулятор θ, гарантируем стабилизацию маятника на тележке при соблюдении фазовых ограничений. Сечение области плоскостями $x_H = 0$, $\dot{x}_H = 0$, $x_r = 0$ и $\dot{x}_r = 0$ представлено на рис. 28.

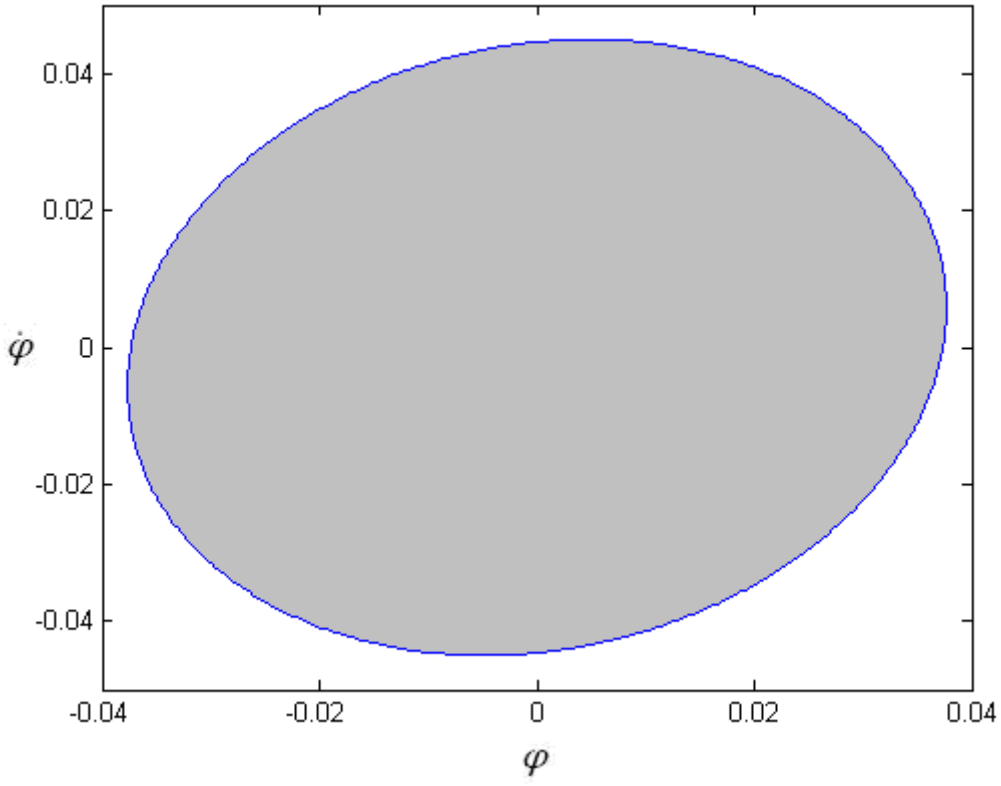

Рис. 28. Сечение области плоскостями $x_H = 0$, $\dot{x}_H = 0$, $x_r = 0$ и $\dot{x}_r = 0$

Шаг 9. Построим график угла отклонения $\varphi(\bar{t})$ звена маятника (в радианах) и график $x_H(\bar{t})$ для заданного начального состояния объекта. Подставим найденный регулятор (2) в уравнение управляемого объекта (19) и проинтегрируем полученное дифференциальное уравнение, применяя функцию-решатель ode45 пакета MATLAB. Данная функция может решать системы уравнений явного вида $y' = f(t, y)$, использует одношаговые явные методы Рунге-Кутта 4-го и 5-го порядка. Это классический метод, который во многих случаях дает хорошие результаты;
Вызов функции ode45 имеет вид:

$$[\text{tout}, yout] = ode45(\text{odefun}, tspan, y0).$$

Ее входные значения: odefin – название ODE-файла, т.е. функции от t и y, которая возвращает вектор-столбец. Вектор $tspan$, определяет интервал интегрирования [$t0$ $tfinal$]. Для получения решений в конкретные моменты времени $t0$, $t1$, …, $tfinal$ (расположенные в порядке уменьшения или увеличения), нужно использовать $tspan$=[$t0$ $t1$ ... $tfinal$]. Величина $y0$ – вектор начальных условий.
Выходные значения функции $yout$ – матрица решений дифференциального уравнения, где каждая строка матрицы соответствует времени, возвращенному в вектор-столбце tout.

Рисунок 29 представляет собой график угла отклонения $\varphi(\bar{t})$ звена маятника (в радианах). На рисунке 30 приведен график $x_H(\bar{t})$ для начального состояния объекта $\varphi = 0.035$, $\dot{\varphi} = 0$, $x_H = 0$, $\dot{x}_H = 0$, $x_r = 0$ и $\dot{x}_r = 0$.

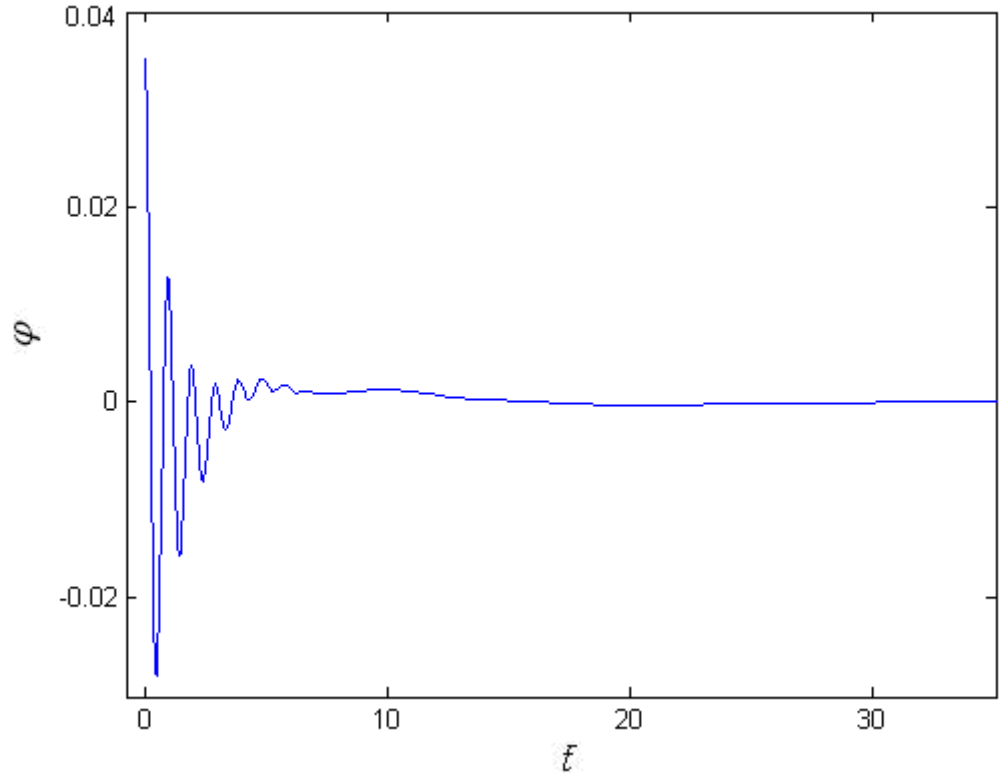

Рис. 29. График угла отклонения $\varphi(\bar{t})$ звена маятника

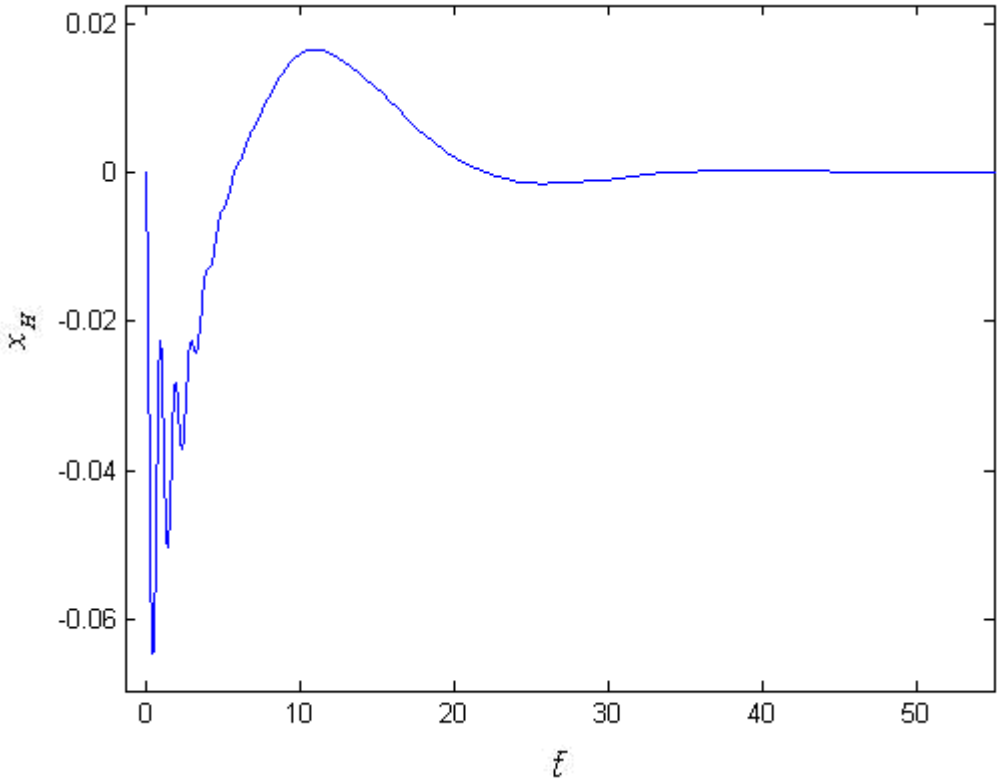

Рис. 30. График $x_H(\bar{t})$

Литература

1. Неймарк Ю.И. Математические модели в естествознании и технике. Нижний Новгород: ННГУ, 2004. 401 с.

2. Неймарк Ю.И. Динамические системы и управляемые процессы. М.: Наука, 1978. 336 с.

3. Баландин Д.В., Городецкий С.Ю. Классические и современные методы построения регуляторов в примерах. Электронное учебно-методическое пособие. Нижний Новгород: Нижегородский госуниверситет, 2012. – 122 с.

4. Баландин Д.В., Коган М.М. Синтез законов управления на основе линейных матричных неравенств. М.: Физматлит, 2007. 280 с.

5. L. E. Ghaoui, F. Oustry, and M. AitRami, “A cone complementarity linearization algorithm for static output-feedback and related problems”, IEEE Trans. Automatic Control, vol. 42, no. 8, pp 1171-1176, Aug. 1997.

6. Yong He, Qing-Guo Wang “An improved ILMI method for static output feedback control with application to multivariable PID control”, IEEE Trans. Automatic Control, vol. 51, no. 10, pp 1678-1683, October. 2006.

7. Федюков А.А. Стабилизация по измеряемому выходу двухзвенного перевернутого маятника. Н. Новгород: Вестник ННГУ, Издательство ННГУ, 2012, No 2. С 177-183.

8. Баландин Д.В., Коган М.М. Синтез линейных законов управления при фазовых ограничениях. Автоматика и телемеханика, 2009, No 6, С 48-57.

9. Федюков А.А. Синтез динамических регуляторов, обеспечивающих стабилизацию систем с ограничениями на фазовые переменные. Н. Новгород: Вестник ННГУ им. Н.И. Лобачевского, № 2. 2013. С. 152-159.

10.MATLAB R2006/2007/2008 + Simulink 5/6/7. Основы применения. 2-е изд., перераб. и доп. / В. П. Дьяконов. М.: СОЛОН-ПРЕСС, 2008. 800 с.

11.Nesterov Y.E., Nemirovski A. Interior-Point Polynomial Algorithms in Convex Programming. Philadelphia: SIAM, 1994.

12.Чурилов А. Н., Гессен А. В. Исследование линейных матричных неравенств. Путеводитель по программным пакетам. СПб.: Изд-во С.-Петерб. ун-та, 2004. 148 с

Александр Анатольевич **Федюков**

ПРИМЕНЕНИЕ СРЕДСТВ ПАКЕТА MATALAB ДЛЯ ЧИСЛЕННОГО РЕШЕНИЯ ЗАДАЧ СТАБИЛИЗАЦИИ ПО ВЫХОДУ ДИНАМИЧЕСКИХ СИСИТЕМ С ФАЗОВЫМИ ОГРАНИЧЕНИЯМИ

Учебно-методическое пособие

Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
«Нижегородский государственный университет
им. Н.И. Лобачевского».
603950, Нижний Новгород, пр. Гагарина, 23.