Dr Bob Davidov

Аналитическое конструирование линейно-квадратичного регулятора

Цель работы: Изучить принцип построения линейного квадратичного регулятора в MATLAB.

Задача работы: Средствами MATLAB спроектировать линейно-квадратичный регулятор.

Приборы и принадлежности: Персональный компьютер, интегрированная среда MATLAB.

Введение

Линейно-квадратичный регулятор обеспечивает оптимальное управление замкнутой системой при минимизации квадратичного функционала качества. В отличии от ПИД регулятора, квадратичный регулятор не вычисляет производные, он использует переменные состояния системы отслеживаемые наблюдателем. В этой работе рассматриваются варианты построения линейно-квадратичного регулятора (lqr) в интегрированной среде MATLAB.

ОБЩИЕ СВЕДЕНИЯ

Для определения возможности решения задачи управления в теории управления используются два важных понятия: управляемость и наблюдаемость, которые определяют перед разработкой регулятора для заданного объекта управления.

Управляемость (Controllability)

Управляемость, как возможность перевода системы из любого начального состояния в любое конечное состояние за конечный интервал времени при неограниченных входных воздействиях $\mathrm{u}(t)$. Для линейных систем управляемость эквивалентна условию равенства ранга матрицы управляемости

$C_o = [B\ AB\ A^2B \dots A^{n-1}B]$ порядку системы n (размеру матрицы A системы в формате пространства состояний):

$$rank(C_o) = size(A, 1)$$

В MATLAB матрицу управляемости C_o можно найти функцией ctrb(A,B) или ctrb(SYS), где SYS – модель в формате пространства состояний.

Наблюдаемость (Observability)

Наблюдаемость – возможность определения внутренних состояний системы по значениям ее выходов $\mathrm{y}(t)$. Подобно управляемости, для линейных систем наблюдаемость эквивалентна условию равенства ранга матрицы наблюдаемости

$O_b = [C;\ CA\ CA^2 \dots\ CA^{n-1}]$ порядку системы n:

$$rank(O_b) = size(A, 1)$$

В этом случае, вектор состояния может быть вычислен с помощью измеренных векторов входа и выхода и матрицы системы.

В MATLAB матрицу наблюдаемости C_o можно найти функцией obsv(A,B) или obsv(SYS), где SYS – модель в формате пространства состояний.

Асимптотическая устойчивость (Asymptotic stability)

Устойчивость объекта можно исследовать с помощью собственных значений матрицы A модели в формате пространства состояний. В динамическом режиме наблюдается стабильное поведение системы если действительные части всех корней системы отрицательны; кроме того, отсутствуют периодические колебания, если у корней нет мнимых значений.

Модель объекта

Модель объекта в формате пространства состояний включает систему дифференциальных уравнений:

$$\begin{cases} \frac{dx}{dt} = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}, \text{ (для непрерывных объектов).}$$

Блок-схема этой же системы показана на Рисунок 1.

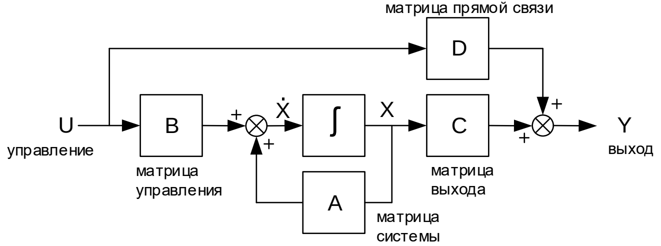


Рисунок 1. Модель системы в формате пространства состояний.

Ниже приведен пример построения дифференциальных уравнений и A, B, C, D матриц пространства состояний системы по ее блок схеме, показанной на Рисунок 2.

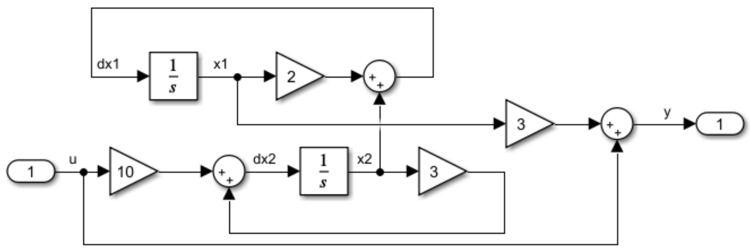


Рисунок 2. Пример динамической системы 2-го порядка.

Система дифференциальных уравнений модели показанной на Рисунок 2:

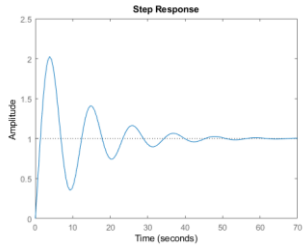
$$\begin{cases} \dfrac{dx_1}{dt} = -0.1667x_1 \ -0.6667x_2 \ +u \\ \dfrac{dx_2}{dt} = \qquad 0.5x_1 \\ \quad y = \quad 0.6667x_1 + 0.6667x_2 \end{cases}$$

Матрицы A, B, C, D системы уравнений:

$$A = \begin{bmatrix} -0.1667 & -0.6667 \\ 0.5 & 0 \end{bmatrix}, \ B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad C = [0.6667 \quad 0.6667], \qquad D = [0]$$

Далее приведен пример вычисления передаточной функции по матрицам A, B, C, D пространства состояний в MATLAB.

```
clear all
A = [-0.1667 -0.6667;0.5 0];
B = [1;0];
C = [0.6667 0.6667];
D = 0;
Wss = ss(A,B,C,D);

Wtf = tf(Wss)
      Wtf =

           2 s + 1
        -----------------
        3 s^2 + 0.5 s + 1

step(Wtf);
```



Регулятор

Требования устойчивости и качества процессов можно описывать в неявной форме как экстремали тех или иных функционалов. Наиболее часто применяют интегральные квадратичные функционалы. Задачей синтеза является определение матрицы коэффициентов обратной связи по состоянию K доставляющей минимум функционалу [1, стр. 147]:

$$I = \int_0^{\infty} (v^T Q v + r u^2) dt$$

Другое обозначение в MATLAB [2]: $u = -Kx$ минимизирует квадратичный функционал

$$J(u) = \int_0^{\infty} (x^T Q x + u^T R u + 2x^T N u) dt$$

где v, x– вектор состояния; Q – неотрицательно-определенная весовая матрица (например, единичная матрица); скаляр r или матрица R – положительный весовой коэффициент; N – весовая матица, если N не используется, ее значение принимается равным нулю; u – скалярное управление.

Матрица коэффициентов обратной связи (регулятора) находится из соотношения [1, стр. 148]: $K = B^T \widehat{K}$, где $\widehat{K}$ является решением нелинейного матричного уравнения Риккати (J. Riccati)

$$-\widehat{K}A - A^T \widehat{K} + \widehat{K} B B^T \widehat{K} \ / r - Q = 0$$

В MATLAB это уравнение обозначается как

$$A^T S + SA - (SB + N) R^{-1} (B^T S + N^T) + Q = 0, \qquad K = R^{-1}(B^T S + T^T)$$

Положение линейно-квадратичного интегрального регулятора Kn в составе системы управления показано на Рисунок 3. На вход регулятора подается вектор X - переменных состояния объекта.

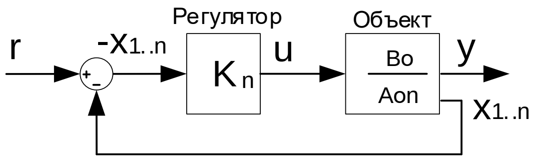


Рисунок 3. Замкнутая система управления с регулятором и объектом.

Коэффициенты **K** оптимального регулятора**,** решение **S** алгебраического уравнения Риккати и корни ***e*** характеристического полинома замкнутого контура можно найти библиотечной функцией MATLAB [2]

[K,S,e] = lqr(SYS,Q,R,N),

в которой аргумент SYS – модель объекта в формате пространства состояний, вычисляется как SYS = ss(A,B,C,D).

Пример применения стандартной функции MATLAB lqr()

Найдем параметры оптимального линейно-квадратичного регулятора для неустойчивого объекта с передаточной функцией

$W_o = \frac{1}{s^2+0.5s-1}$; матрицы $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ и $r = 1$

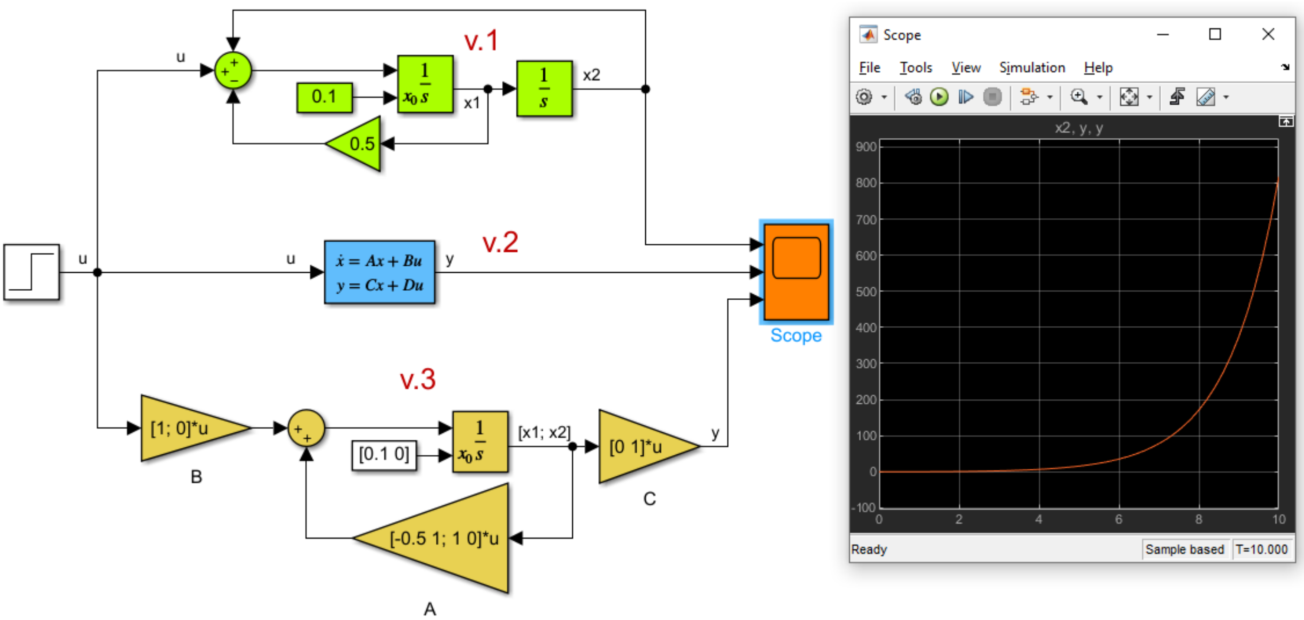


Рисунок 4. Три варианта построения неустойчивого объекта $W_o = \frac{1}{s^2+0.5s-1}$;

Решение:

```
Wo_tf = tf(1,[1 0.5 -1]) % ПФ объекта
      Wo_tf =
               1
        ---------------
        s^2 + 0.5 s - 1
      Continuous-time transfer function

Wo_ss = ss(Wo_tf)  % объект в формате пространства состояний
      Wo_ss =
        A =
               x1    x2
         x1  -0.5     1
         x2     1     0
        B =
             u1
         x1   1
         x2   0
        C =
             x1  x2
         y1   0   1
        D =
             u1
         y1   0
      Continuous-time state-space model

A = Wo_ss.A; % матрица A пространства состояний
eig(A) % полюса ПФ объекта
         -1.2808
          0.7808

[K,S,e] = lqr(Wo_ss,Q,r) % K - параметры регулятора; S - решение алгебраического уравнения Риккати; e - корни Характеристического Полинома замкнутой системы

      K =
          1.6089    2.0488
      S =
         16.0894   20.4881
         20.4881   27.1187
      e =
         -0.8033
         -1.3057

eig(A-B*K) % проверка корней Характеристического Полинома замкнутой системы
         -1.3057
         -0.8033
```

Два варианта построения идентичных системы управления с lqr регулятором в Simulink показаны на Рисунок 5.

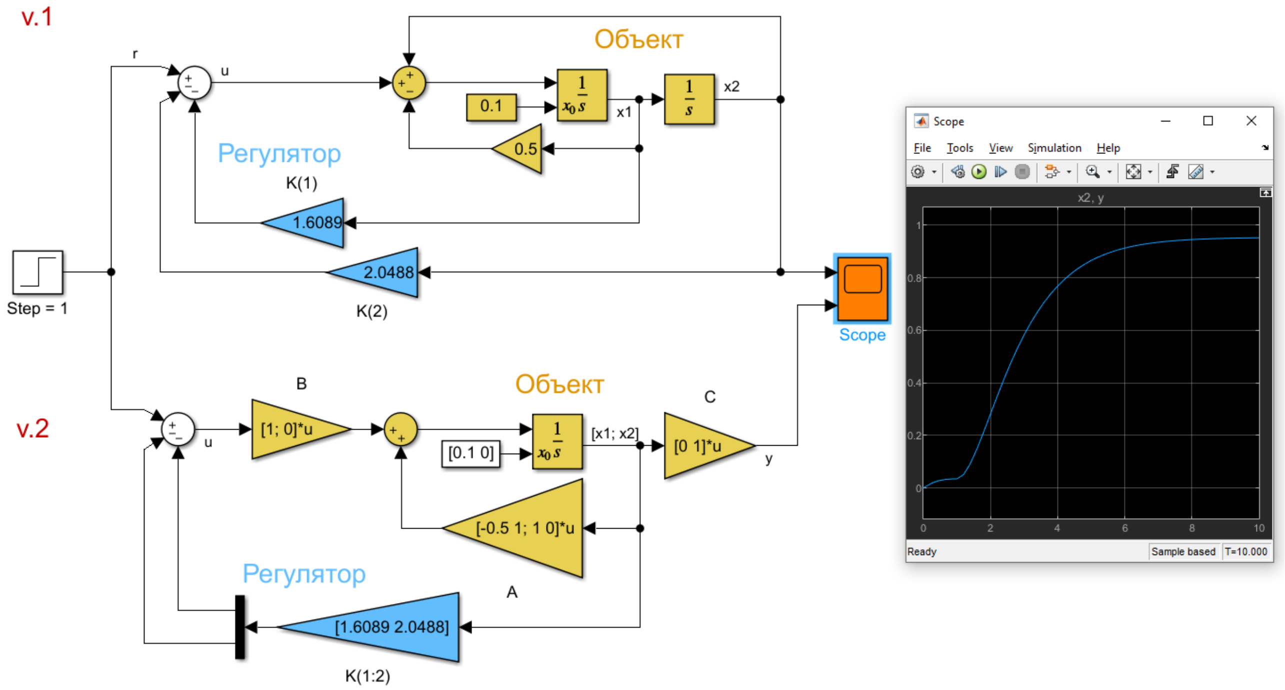


Рисунок 5. Два варианта системы управления с lqr регулятором, ПФ объекта $Wo = \frac{1}{s^2+0.5s-1}$. $Q = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$, $r = 1$. Блоки объекта выделены желтым, блоки регулятора – синим цветом.

Аналитическое конструирование линейно-квадратичного регулятора

Вычислим в MATLAB параметры оптимального линейно-квадратичного регулятора для того же объекта: $W_o = \frac{1}{s^2+0.5s-1}$, $Q = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$, $r = 1$ без применения стандартной функции lqr.

1. Перевод передаточной функции в формат пространства состояния

```
Wo_tf = tf(1,[1 0.5 -1]) % структура объекта в формате ПФ
      Wo_tf =
               1
        ---------------
        s^2 + 0.5 s - 1
      Continuous-time transfer function

Wo_ss = ss(Wo_tf)   % объект в формате пространства состояний
      Wo_ss =
        A =
               x1     x2
         x1  -0.5      1
         x2     1      0
        B =
             u1
         x1   1
         x2   0
        C =
             x1   x2
         y1   0    1
```

```
D =
      u1
 y1   0

Continuous-time state-space model

A = Wo_ss.A; B = Wo_ss.B; C = Wo_ss.C;
```

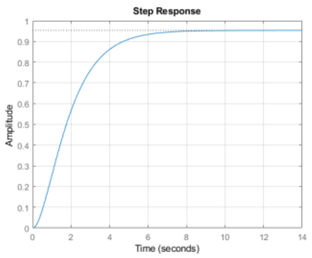
2. Проверка объекта на управляемость, наблюдаемость и устойчивость

```
Co = [B A*B] % контролируемость
      Co =
          1.0000   -0.5000
               0    1.0000
rank(Co) % ранг матрицы равен порядку системы ?
      ans =
           2


Ob = [C; C*A] % наблюдаемость
      Ob =

           0     1
           1     0
rank(Ob) % ранг матрицы равен порядку системы?
      ans =
           2

eig(A) % полюса ПФ объекта
      ans =
         -1.2808
          0.7808
```

3. Решение уравнения Риккати

```
Q = eye(size(A,1))
      Q =
           1     0
           0     1
r = 10;

syms sr [1,4]
Sr = [sr1 sr2; sr3 sr4]
      Sr =
      [ sr1, sr2]
      [ sr3, sr4]

S = Sr*Ar+Ar'*Sr-Sr*Br*Br'*Sr/r+Q % уравнение Риккати
      S =
      [   sr1^2/10 + sr1 - sr2 - sr3 - 1, sr2/2 - sr1 - sr4 + (sr1*sr2)/10]
      [ sr3/2 - sr1 - sr4 + (sr1*sr3)/10,      (sr2*sr3)/10 - sr3 - sr2 - 1]

Kr = solve(S) % решение уравнения Риккати
        struct with fields:
          sr1: [6×1 sym]
          sr2: [6×1 sym]
          sr3: [6×1 sym]
          sr4: [6×1 sym]
```

```
% среди множества решений системы уравнений Kr выбираем вариант i_n, у
которого
% первый член матрицы имеет максимальное значение ?
i_n = 1;
Kr_sr1_max = vpa(Kr.sr1(i_n))
for i = 2:size(Kr.sr1,1) %
    if vpa(Kr.sr1(i)) > Kr_sr1_max
        Kr_sr1_max = vpa(Kr.sr1(i));
        i_n = i;
    end
end
      i_n =
           6
% Матрица S из [K,S,e] = lqi(Wo_ss,Q,r,0)
Kr = vpa([Kr.sr1(i_n) Kr.sr2(i_n); ...
          Kr.sr3(i_n) Kr.sr4(i_n)])
      Kr =
      [  16.089375752592353803823956346664, 20.488088481701515469914535136799]
      [ 20.488088481701515469914535136799, 27.118723891703909515984659849446]
% Параметры регулятора
K = B'*Kr/r
      K =
      [ 1.6089375752592353803823956346664, 2.0488088481701515469914535136799]
K = eval(K) % перевод символьных данных в численные

% Матрица пространства состояний замкнутой системы
A = [A(1)-K(1) A(2)-K(2); 1 0]
      A =
         -2.1089   -1.0488
          1.0000         0
D = 0;

Wtf = tf(ss(A,B,C,D)) % ПФ замкнутой системы
      Wtf =
                  1
        ---------------------
        s^2 + 2.109 s + 1.049

pole(Wtf) % полюса (корни) системы управления
         -1.3057
         -0.8033
% Реакция системы управления на единичное ступенчатое воздействие
```




```
step(Wtf)
grid
```

Применение наблюдателя в структурах с линейно-квадратичными регуляторами

Если объект наблюдаем полностью, то по измеренным значениям переменной выхода y можно вычислить текущее состояние объекта [1, стр. 148]. При этом управляющее воздействие на объект формируется по оценкам вектора состояния.

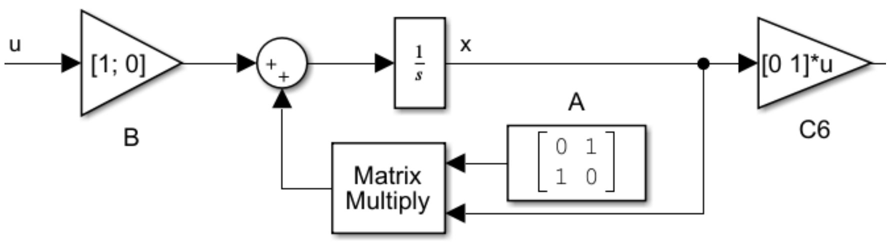
Для оценки вектора состояния в систему включают модель наблюдателя, внутри которого находится модель объекта управления. Поскольку, оценка может отличаться от состояния объекта из-за различия начальных условий, действующих на объект возмущений, а также неточности описания объекта, то наблюдатель дополняют обратной связью, при которой оценка должна асимптотически стремиться к состоянию объекта.

Синтез наблюдателя поясняется в соответствующей работе курса [3].

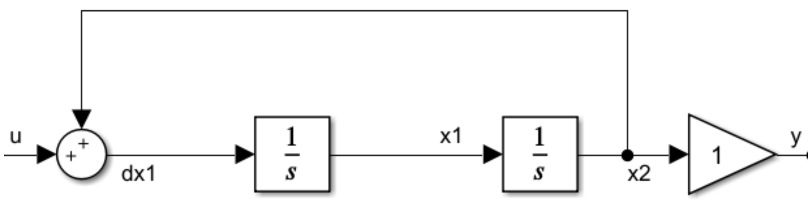
ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Задание 1. Конструирование линейно-квадратичного регулятора для управления неустойчивым объектом.

1. Постройте в Simulink модель неустойчивого объекта со следующим распределением корней p = [-1 +1].



или

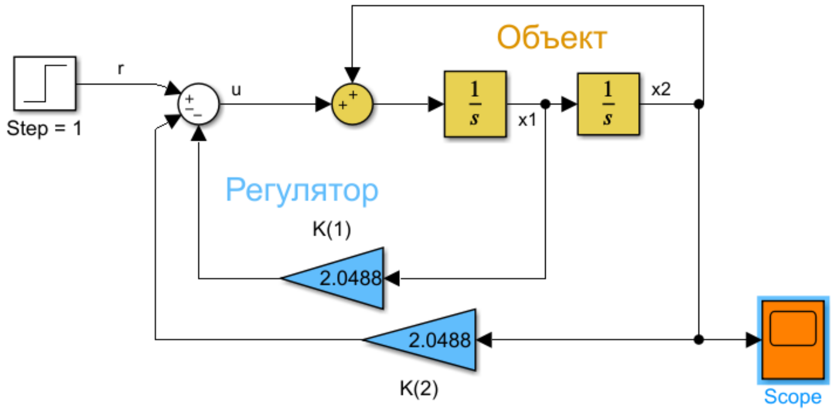


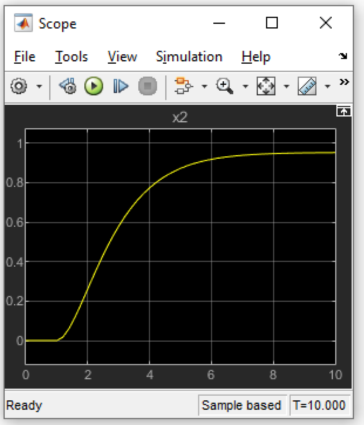
Для выполнения этого пункта можно использовать функции MATLAB:

W_zpk = zpk([],[1 -1],1); ss(W_zpk)

2. Рассчитайте коэффициенты регулятора “**K**” и определите корни замкнутой системы управления “**e**”, как показано в разделе “Пример применения стандартной функции MATLAB lqr()”, [**K**,S,**e**] – результат выполнения функции lqr.
3. Рассчитайте коэффициенты регулятора, как показано в разделе “Аналитическое конструирование линейно-квадратичного регулятора”.
4. Сравните результаты, полученные в п.2 и п.3.

5. Постройте модель системы управления с квадратичным регулятором и график реакции системы на единичное ступенчатое воздействие.





6. По блок-схеме п.5 постройте систему дифференциальных уравнений системы в формате пространства состояний.

$$\dot{x}1 = -K(1) * x1 + (-K2) * x2 + 1 * u$$
$$\dot{x}2 = 1 * x1 + 0 * x2 + 0 * u$$
$$y = 0 * x1 + 1 * x2 + 0 * u$$

7. Из системы дифференциальных уравнений выделите матрицы A, B, C, D пространства состояний замкнутой системы.

$$A_3 = \begin{bmatrix} -K(1) & -K(2) \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} A_o(1,1) - K1 & A_o(2,1) - K2 \\ A_o(2,1) & A_o(2,2) \end{bmatrix};$$

$$B_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix};$$

$$C_3 = [0 \quad 1];$$

$$D_3 = [0].$$

8. Сравните A_3 с результирующий матрицей: A-B*K.
9. По матрице Az = A_3 найдите корни характеристического полинома замкнутой системы.

```
>> eig(Az)
    -1.0488
    -1.0000
```

10. Сравните корни замкнутой системы управления п.9 с результатами выполнения п.2.
11. Изменяя коэффициенты весовой матрицы Q и скаляра r определите их влияние на качество управления.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие преимущества дает оптимальный линейно-квадратичный регулятор (lqr) в сравнении с ПИД регулятором?
2. Влияют ли параметры весовых матриц Q и R на коэффициенты регулятора и, как следствие, на качество управления?
3. Какими способами можно получить переменные состояния, поступающие на вход регулятора?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. А.А.Алексеев, Д.Х.Имаев, Н.Н.Кузьмин, В.Б.Яковлев. Теория управления: Учеб./СПб.: Изд-во СПбГЭТУ “ЛЭТИ”, 1999. – 435 с.
2. Help MATLAB.
3. Dr. Bob Davidov. Компьютерные технологии управления в технических системах http://portalnp.ru/author/bobdavidov