Лабораторная работа №9

Задание 1

```
A = [-7 0 0 0;
0 3 0 0;
0 0 2 7;
0 0 -7 2]
```

```
A = 4×4
    -7     0     0     0
     0     3     0     0
     0     0     2     7
     0     0    -7     2
```

```
B = [0; 7; 0; 6]
```

```
B = 4×1
     0
     7
     0
     6
```

Определяем собственные числа

```
eig(A)
```

```
ans = 4×1 complex
   2.0000 + 7.0000i
   2.0000 - 7.0000i
  -7.0000 + 0.0000i
   3.0000 + 0.0000i
```

Выберем различные значения желаемой степени устойчивости α

```
a_1 = 7
```

```
a_1 = 7
```

```
a_2 = 2
```

```
a_2 = 2
```

```
a_3 = 0.05
```

```
a_3 = 0.0500
```

Далее решаем неравенства Ляпунова

```
x_0 = [1; 1; 1; 1]
```

```
x_0 = 4×1
     1
     1
     1
     1
```

```
%m = 2500

cvx_begin sdp
```

```
variable Q(4, 4)
variable Y(4, 2)
variable P(4, 4)
variable Y1(1, 4)
%variable m
%minimize m
%Q > 0.00001*eye(4);
P > 0.00001*eye(4);
```

```
Warning: The use of strict inequalities in CVX is strongly discouraged,
    because solvers treat them as non-strict inequalities. Please
    consider using ">=" instead.
Warning: This linear matrix inequality appears to be unsymmetric. This is
very likely an error that will produce unexpected results. Please check
the LMI; and, if necessary, re-enter the model.
```

```
%A'*Q + Q*A + 2*a_3*Q + C'*Y'+Y*C <= 0;
P*A' + A*P + 2*a_3*P + Y1'*B' + B*Y1 <= 0;
```

```
Warning: This linear matrix inequality appears to be unsymmetric. This is
very likely an error that will produce unexpected results. Please check
the LMI; and, if necessary, re-enter the model.
```

```
%[P x_0;
%     x_0' 1] > 0;
%[P Y';
%     Y m] > 0;
cvx_end
```

```
Calling SDPT3 4.0: 44 variables, 8 equality constraints
------------------------------------------------------------

 num. of constraints =  8
 dim. of sdp    var  =  8,   num. of sdp  blk  =  2
 dim. of free   var  = 24 *** convert ublk to lblk
 number of nearly dependent constraints = 1
 To remove these constraints, re-run sqlp.m with OPTIONS.rmdepconstr = 1.
*******************************************************************
   SDPT3: Infeasible path-following algorithms
*******************************************************************
 version  predcorr  gam  expon  scale_data
   HKM      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj    cputime
-------------------------------------------------------------------
 0|0.000|0.000|1.7e+02|3.3e+02|1.1e+05| 0.000000e+00  0.000000e+00| 0:0:00| chol  1  1
 1|0.931|0.558|1.2e+01|1.5e+02|1.2e+04| 0.000000e+00  4.496711e-04| 0:0:00| chol  1  1
 2|1.000|0.936|9.0e-05|9.5e+00|2.0e+02| 0.000000e+00  3.854664e-05| 0:0:00| chol  1  1
 3|1.000|0.989|1.4e-05|1.1e-01|1.0e+00| 0.000000e+00  4.499138e-07| 0:0:00| chol  1  1
 4|1.000|0.990|1.0e-06|1.2e-03|1.0e-02| 0.000000e+00  5.915853e-09| 0:0:00| chol  1  1
 5|1.000|0.998|2.9e-08|1.3e-05|2.5e-04| 0.000000e+00  1.543233e-10| 0:0:00| chol  1  1
 6|1.000|1.000|3.7e-10|1.6e-05|1.9e-05| 0.000000e+00  1.379703e-11| 0:0:00| chol  1  1
 7|1.000|0.989|2.2e-11|1.2e-06|2.2e-07| 0.000000e+00  1.552343e-13| 0:0:00| chol  1  1
 8|1.000|0.989|2.0e-13|1.4e-08|2.5e-09| 0.000000e+00  1.761101e-15| 0:0:00|
  stop: max(relative gap, infeasibilities) < 1.49e-08
-------------------------------------------------------------------
 number of iterations   =  8
 primal objective value =  0.00000000e+00
 dual   objective value =  1.76110129e-15
 gap := trace(XZ)       = 2.53e-09
 relative gap           = 2.53e-09
 actual relative gap    = -1.76e-15
 rel. primal infeas (scaled problem)   = 2.01e-13
 rel. dual     "        "       "      = 1.36e-08
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "        "       "      = 0.00e+00
 norm(X), norm(y), norm(Z) = 5.9e+01, 4.4e-04, 9.6e-09
 norm(A), norm(b), norm(C) = 2.9e+01, 1.0e+00, 1.0e+00
 Total CPU time (secs)  = 0.19
 CPU time per iteration = 0.02
 termination code       =  0
 DIMACS: 2.0e-13  0.0e+00  1.4e-08  0.0e+00  -1.8e-15  2.5e-09
-------------------------------------------------------------------

------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +0
```

```
%m
```

И находим матриу реглятора K:

```
K = Y1*inv(P)
```

```
K = 1×4
   -0.0000   -5.0430   -5.2166    0.6592
```

```
%L = inv(Q)*Y
```

Далее определим корни матрицы A+BK:

```
%LC = eig(A+L*C)
BK = eig(A+B*K)
```

```
BK = 4×1 complex
  -2.8847 + 6.8685i
  -2.8847 - 6.8685i
 -18.5760 + 0.0000i
  -7.0000 + 0.0000i
```