

sample_cvx_***.m

```
1 % sample_cvx_default.m
2 % last modified: 2023/05/15 by Masakatsu KAWATA
3
4 plant % アクロボットに対してシステム行列などを定義した M ファイルの実行
5 % -----
6 cvx_begin sdp % SDP (半正定値計画問題; LMI を制約とした凸最適化問題) の開始
7     variable gamma(1,1) % 決定変数 gamma (ガンマ): スカラ
8     variable X(n,n) symmetric % 決定変数 X: n x n の対称行列
9     variable Z(n,m) % 決定変数 Z: n x m の長方形列
10 % -----
11     minimize(gamma) % 目的関数 E = gamma を最小化することを宣言
12 % -----
13     AX = A*X + B2*Z';
14     CX = C1*X + D12*Z';
15 % -----
16     M1 = [ r*X      AX-c*X
17            AX'-c*X  r*X ];
18     M1 > 0; % M1 > 0
19 % -----
20     M2 = [ AX+AX'  B1      CX'
21            B1'     -gamma*eye(q)  D11'
22            CX      D11      -gamma*eye(p) ];
23     M2 < 0; % M2 < 0
24 cvx_end % SDP の終了
25 % -----
26 K = Z'*inv(X) % コントローラゲイン K
```

説明

M ファイル sample_cvx_default.m を実行するには,

- LMI パーサ
 - CVX
- SDP ソルバ (CVX に含まれます)
 - SeDuMi
 - SDPT3

をインストールしている必要があります. インストール方法は,

<https://kawata.apps.kct.ac.jp/>

にある

SDP ソルバと LMI パーサのインストール
CVX のインストール

を参照してください.

この M ファイルは, 連立 LMI

$$M_1 := \begin{bmatrix} rX & A_X - cX \\ * & rX \end{bmatrix} \succ 0 \quad (1)$$

$$M_2 := \begin{bmatrix} A_X + A_X^\top & B_1 & * \\ * & -\gamma I & * \\ C_X & D_{11} & -\gamma I \end{bmatrix} \prec 0 \quad (2)$$

を満足する解

$$\mathbf{X} = \mathbf{X}^\top \succ 0, \mathbf{Z}, \gamma > 0$$

が存在する範囲で線形目的関数 $E = \gamma > 0$ を最小化し (線形目的関数 $E = \gamma > 0$ を最小化する凸最適化問題を解き), 得られた解を用い, コントローラを

$$u(t) = \mathbf{K}\mathbf{x}(t), \mathbf{K} = \mathbf{Z}^\top \mathbf{X}^{-1} \quad (3)$$

のように決定します. ただし,

$$\begin{cases} \mathbf{A}_X := \mathbf{A}\mathbf{X} + \mathbf{B}_2\mathbf{Z}^\top \\ \mathbf{C}_X := \mathbf{C}_1\mathbf{X} + \mathbf{D}_{12}\mathbf{Z}^\top \end{cases}$$

です.

ソルバを指定する (どのソルバを使用するかを明示する) ためには, 関数 `cvx_solver` を利用します.

`sample_yalmip_sedumi.m` ソルバ SeDuMi を使用

```
1 % sample_cvx_sedumi.m
2 % last modified: 2023/05/15 by Masakatsu KAWATA
3
4 plant                                % アクロボットに対してシステム行列などを定義した M ファイルの実行
5 % -----
6 cvx_solver('sedumi');                % ソルバとして SeDuMi を使用
7 % -----
8 cvx_begin sdp                        % SDP (半正定値計画問題; LMI を制約とした凸最適化問題) の開始
```

以下, `sample_cvx_default.m` と同じ

`sample_yalmip_sdpt3.m` ソルバ SDPT3 を使用

```
1 % sample_cvx_sdpt3.m
2 % last modified: 2023/05/15 by Masakatsu KAWATA
3
4 plant                                % アクロボットに対してシステム行列などを定義した M ファイルの実行
5 % -----
6 cvx_solver('sdpt3');                 % ソルバとして SDPT3 を使用
7 % -----
8 cvx_begin sdp                        % SDP (半正定値計画問題; LMI を制約とした凸最適化問題) の開始
```

以下, `sample_cvx_default.m` と同じ

実行結果

使用するソルバの違いにより, 以下のような実行結果が得られます.

`sample_yalmip_sedumi.m`

```
>> sample_cvx_sedumi

Calling sedumi: 91 variables, 21 equality constraints
For improved efficiency, sedumi is solving the dual problem.
-----
SeDuMi 1.21 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
Alg = 2: xz-corrector, Adaptive Step-Differentiation, theta = 0.250, beta = 0.500
eqs m = 21, order n = 19, dim = 165, blocks = 3
nnz(A) = 207 + 0, nnz(ADA) = 441, nnz(L) = 231
it :      b*y      gap    delta rate   t/tP*   t/tD*   feas cg cg   prec
0 :              6.71E+01 0.000
1 :  -1.37E+00  1.63E+01 0.000 0.2434 0.9000 0.9000   1.30 1 1  1.7E+01
2 :  -2.25E+00  4.92E+00 0.000 0.3013 0.9000 0.9000   0.42 1 1  7.7E+00
```

```

 3 : -1.97E+00 1.26E+00 0.000 0.2571 0.9000 0.9000 0.66 1 1 2.3E+00
 4 : -1.63E+00 3.41E-01 0.000 0.2697 0.9000 0.9000 0.58 1 1 8.1E-01
 5 : -1.38E+00 9.37E-02 0.000 0.2747 0.9000 0.9000 0.46 1 1 3.1E-01
 6 : -1.22E+00 2.65E-02 0.000 0.2826 0.9000 0.9000 0.46 1 1 1.2E-01
 7 : -1.09E+00 7.03E-03 0.000 0.2652 0.9000 0.9000 0.36 1 1 4.8E-02
 8 : -9.79E-01 1.78E-03 0.000 0.2537 0.9000 0.9000 0.26 1 1 2.0E-02
 9 : -9.06E-01 4.68E-04 0.000 0.2624 0.9000 0.9000 0.24 1 1 8.6E-03
10 : -8.63E-01 1.41E-04 0.000 0.3010 0.9000 0.9000 0.30 1 1 3.8E-03
11 : -8.32E-01 4.26E-05 0.000 0.3026 0.9000 0.9000 0.36 1 1 1.7E-03
12 : -8.12E-01 1.53E-05 0.000 0.3583 0.9000 0.9000 0.41 1 1 8.4E-04
13 : -7.99E-01 5.69E-06 0.000 0.3731 0.9000 0.9000 0.50 1 1 4.1E-04
14 : -7.92E-01 2.78E-06 0.000 0.4880 0.9000 0.9000 0.49 1 1 2.6E-04
15 : -7.86E-01 1.11E-06 0.000 0.4003 0.9000 0.9000 0.64 1 1 1.2E-04
16 : -7.82E-01 5.55E-07 0.000 0.4991 0.9000 0.9000 0.54 1 1 7.7E-05
17 : -7.79E-01 2.20E-07 0.000 0.3957 0.9000 0.9000 0.67 1 1 3.6E-05
18 : -7.77E-01 1.05E-07 0.000 0.4771 0.9000 0.9000 0.55 1 1 2.2E-05
19 : -7.76E-01 3.86E-08 0.000 0.3680 0.9000 0.9000 0.69 1 1 9.4E-06
20 : -7.75E-01 2.04E-08 0.000 0.5298 0.9000 0.9000 0.70 2 2 5.5E-06
21 : -7.75E-01 5.63E-10 0.000 0.0276 0.9890 0.9900 0.97 1 1 1.7E-07
22 : -7.75E-01 4.92E-12 0.000 0.0087 0.9937 0.9990 0.99 2 2 1.5E-09

```

```

iter seconds digits      c*x          b*y
22      0.0   Inf -7.7494092147e-01 -7.7494090478e-01
|Ax-b| = 1.6e-09, [Ay-c]_+ = 0.0E+00, |x|= 1.1e+00, |y|= 4.0e+03

```

Detailed timing (sec)

```

      Pre      IPM      Post
4.688E-02  3.125E-02  1.562E-02
Max-norms: ||b||=1, ||c|| = 2,
Cholesky |add|=0, |skip| = 0, ||L.L|| = 13129.6.

```

Status: Solved

Optimal value (cvx_optval): +0.774941

K =

```

      6.5233      -7.7435      1.7719      -0.2351      -6.8743

```

sample_cvx_sdpt3.m

```
>> sample_cvx_sdpt3
```

```

Calling sdpt3: 91 variables, 21 equality constraints
For improved efficiency, sdpt3 is solving the dual problem.

```

```

num. of constraints = 21
dim. of sdp var = 18, num. of sdp blk = 2
*****
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|2.1e+02|6.2e+01|9.3e+03| 0.000000e+00 0.000000e+00| 0:0:00| chol 1 1
1|0.850|0.929|3.2e+01|4.5e+00|1.2e+03|-4.193559e-01 -3.671447e+01| 0:0:00| chol 1 1
2|0.668|0.892|1.1e+01|5.0e-01|4.4e+02|-7.489591e-01 -3.955435e+01| 0:0:00| chol 1 1
3|0.606|0.856|4.2e+00|7.2e-02|2.0e+02|-3.959338e-01 -2.684330e+01| 0:0:00| chol 1 1
4|0.581|1.000|1.8e+00|1.0e-04|1.0e+02|-3.479096e-01 -1.324491e+01| 0:0:00| chol 1 1
5|0.920|0.893|1.4e-01|2.0e-05|1.1e+01|-4.609823e-01 -4.521248e+00| 0:0:00| chol 1 1
6|0.545|1.000|6.5e-02|1.0e-06|6.5e+00|-8.152513e-01 -2.706342e+00| 0:0:00| chol 1 1
7|0.846|0.979|1.0e-02|1.3e-02|1.3e+00|-7.343153e-01 -1.195423e+00| 0:0:00| chol 1 1
8|0.694|1.000|3.0e-03|2.0e-03|6.0e-01|-7.427374e-01 -9.801743e-01| 0:0:00| chol 1 1
9|0.714|0.904|8.7e-04|8.0e-04|2.7e-01|-7.596073e-01 -8.783238e-01| 0:0:00| chol 1 1
10|0.435|1.000|4.9e-04|1.7e-04|1.8e-01|-7.601901e-01 -8.312874e-01| 0:0:00| chol 1 1

```

```

11|0.626|0.764|1.8e-04|1.4e-04|1.1e-01|-7.572344e-01 -8.095710e-01| 0:0:00| chol 1 1
12|0.579|1.000|7.7e-05|3.7e-05|6.8e-02|-7.583239e-01 -7.954089e-01| 0:0:00| chol 2 2
13|0.669|0.575|2.6e-05|3.1e-05|4.2e-02|-7.597438e-01 -7.882043e-01| 0:0:00| chol 1 2
14|0.540|1.000|1.2e-05|5.1e-06|2.6e-02|-7.637057e-01 -7.821747e-01| 0:0:00| chol 2 2
15|0.728|0.825|3.2e-06|3.2e-06|1.5e-02|-7.664635e-01 -7.783843e-01| 0:0:00| chol 2 2
16|0.868|1.000|4.2e-07|6.4e-07|2.4e-03|-7.733526e-01 -7.753346e-01| 0:0:00| chol 2 2
17|1.000|1.000|1.2e-12|8.5e-08|3.9e-04|-7.746785e-01 -7.750680e-01| 0:0:00| chol 2 2
18|0.980|0.985|3.4e-13|1.2e-09|7.2e-06|-7.749357e-01 -7.749429e-01| 0:0:00| chol 2 2
19|1.000|1.000|4.2e-13|9.6e-13|3.1e-07|-7.749406e-01 -7.749409e-01| 0:0:00| chol 2 2
20|1.000|1.000|1.5e-12|1.1e-12|1.4e-08|-7.749408e-01 -7.749408e-01| 0:0:00|

```

stop: max(relative gap, infeasibilities) < 1.49e-08

```

-----
number of iterations    = 20
primal objective value = -7.74940787e-01
dual  objective value = -7.74940801e-01
gap := trace(XZ)       = 1.37e-08
relative gap           = 5.36e-09
actual relative gap    = 5.48e-09
rel. primal infeas     = 1.45e-12
rel. dual  infeas     = 1.10e-12
norm(X), norm(y), norm(Z) = 1.1e+00, 4.0e+03, 7.1e+03
norm(A), norm(b), norm(C) = 9.9e+01, 2.0e+00, 2.4e+00
Total CPU time (secs) = 0.45
CPU time per iteration = 0.02
termination code      = 0
DIMACS: 1.5e-12  0.0e+00  1.3e-12  0.0e+00  5.5e-09  5.4e-09
-----

```

Status: Solved

Optimal value (cvx_optval): +0.774941

K =

```

6.5233  -7.7435  1.7719  -0.2351  -6.8743

```