

LMI に基づく制御系解析・設計

川田 昌克*・蛭原 義雄†

1. はじめに

近年の精力的な研究により、線形行列不等式 (LMI: Linear Matrix Inequality) に基づくロバスト制御系や多目的制御系、ゲインスケジューリング制御系の解析・設計の手法が確立された [1-7]。従来、LMI の数値解を得るために、商用の MATLAB ツールボックスである LMI Control Toolbox [8,9] (現在では Robust Control Toolbox に統合) が利用されることが多かった。このツールボックスには、半正定値計画問題 (SDP: Semidefinite Programming), すなわち、LMI を制約条件として線形目的関数を最大化 (最小化) する凸最適化問題のソルバ LMILAB が含まれているが、実行速度が速いとはいえず、サイズの大きい LMI を解く際に問題となる。また、そこで用いられる LMI の記述が煩雑であるという欠点もある。それに対し、現在では、MATLAB 上で使用可能なフリーウェアの SDP ソルバ SeDuMi [10] や SDPT3 [11] などにより実行速度の問題に対処できる。また、ユーザフレンドリなフリーウェアの LMI パーサである YALMIP [12] や CVX [13] を併用することで、LMI の記述が飛躍的に単純化される (本特集号の総説を参照)。

本解説では、まず、LMI に基づく制御系解析・設計について簡単に説明する。ついで、SDP ソルバ SeDuMi と LMI パーサ YALMIP のインストール方法について説明した後、いくつかの例を通じ、制御系解析・設計に LMI を「ツールとして使う」という立場から、解説する。

記述の簡単化のため、本解説では、正方行列 M に対して、 $\text{He}[M] := M + M^T$ と定義する。また、対称行列を以下のように記述する。

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{12}^T & M_{22} \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ * & M_{22} \end{bmatrix} \left(= \begin{bmatrix} M_{11} & * \\ M_{12}^T & M_{22} \end{bmatrix} \right)$$

2. LMI に基づく制御系解析・設計

2.1 LMI とは？

対称行列 M が正定であることを意味する “ $M \succ 0$ ” や負定であることを意味する “ $M \prec 0$ ” を行列不等式とよぶ。とくに、 M が定数行列と決定変数 ξ_i ($i =$

$1, \dots, k$) に関して 1 次の行列の和で表された行列不等式を、線形行列不等式 (LMI) とよぶ。つまり、LMI は既知の行列 $M_i = M_i^T$ ($i = 0, \dots, k$) と未知の決定変数 $\xi = [\xi_1 \ \dots \ \xi_k]^T$ を用いて、次式のように記述できる。

$$M(\xi) := M_0 + \sum_{i=1}^k \xi_i M_i \succ 0 \quad (\text{あるいは} \prec 0) \quad (1)$$

【例題 1】 (安定解析問題) 零入力の線形時不変システム

$$\dot{x}(t) = Ax(t), \quad x(0) = x_0 \quad (2)$$

が安定であるための必要十分条件は、Lyapunov 不等式

$$M := \text{He}[PA] \prec 0 \quad (3)$$

を満足する正定対称解

$$P = P^T \succ 0 \quad (4)$$

(この条件も行列不等式である) が存在することである。 P の要素を ξ_i とおけば、(3), (4) 式はともに (1) 式の形式で記述でき、これらが LMI であることがいえる。

実際の制御系解析や設計における行列不等式条件は、そのままでは LMI ではないことも多い。しかし、多くの場合、変数変換や以下に示す Schur の補題を利用することで、LMI に変換することが可能である。

【補題 1】 (Schur の補題) 以下の条件 (A) ~ (C) は等価である。

$$(A) \quad \begin{bmatrix} M_{11} & M_{12} \\ M_{12}^T & M_{22} \end{bmatrix} \succ 0 \quad (5)$$

$$(B) \quad M_{11} \succ 0 \text{ かつ } M_{22} - M_{12}^T M_{11}^{-1} M_{12} \succ 0 \quad (6)$$

$$(C) \quad M_{22} \succ 0 \text{ かつ } M_{11} - M_{12} M_{22}^{-1} M_{12}^T \succ 0 \quad (7)$$

【例題 2】 (コスト保証制御問題 [14]) 線形時不変システム

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0 \quad (8)$$

を考える。このとき、

$$J = \int_0^\infty (x^T(t)Qx(t) + u^T(t)Ru(t))dt$$

* 舞鶴工業高等専門学校 電子制御工学科

† 京都大学 大学院 工学研究科 電気工学専攻

Key Words: linear matrix inequality, robust control, gain scheduling control, numerical computation.

$$\leq \gamma x_0^T x_0 \quad (9)$$

を満足する、状態フィードバック形式のコントローラ

$$u(t) = Kx(t) \quad (10)$$

を設計する問題を考える。ただし、 $Q = Q^T \succ 0$, $R = R^T \succ 0$, $\gamma > 0$ である。

この問題の可解条件は、

$$\text{He}[P(A+BK)] + K^T R K + Q \prec 0 \quad (11)$$

$$P \prec \gamma I \quad (12)$$

を同時に満足する決定変数 $P = P^T \succ 0$, K , $\gamma > 0$ が存在することである。(11) 式は P , K^T と K との積を含むため、LMI ではない。そこで、 $X := P^{-1}$ (このとき、 $X = X^T \succ 0$ となる) を (11) 式の左右からかけ、変数変換 $F := KX$ を行うことにより、式変形する。さらに、 $\Phi := AX + BF$ とおき、 Q_h を $Q = Q_h^T Q_h$ を満足する正方行列として、Schur の補題を適宜、利用する。その結果、(11) 式を

$$\begin{aligned} & -P^{-1}\{\text{He}[P(A+BK)] + K^T R K + Q\}P^{-1} \\ &= -\text{He}[\Phi] - \begin{bmatrix} Q_h X \\ RF \end{bmatrix}^T \begin{bmatrix} I & * \\ 0 & R \end{bmatrix}^{-1} \begin{bmatrix} Q_h X \\ RF \end{bmatrix} \succ 0 \\ & \Downarrow \text{Schur の補題: (7) 式から (5) 式への変換} \\ & \begin{bmatrix} -\text{He}[\Phi] & * & * \\ Q_h X & I & * \\ RF & 0 & R \end{bmatrix} \succ 0 \end{aligned} \quad (13)$$

のように、 X , F に関する LMI に変換できる。一方、(12) 式を X を用いた形式に書き換えると、

$$\begin{aligned} & \gamma I - P = \gamma I - I \cdot X^{-1} \cdot I^T \succ 0 \\ & \Downarrow \text{Schur の補題: (7) 式から (5) 式への変換} \\ & \begin{bmatrix} \gamma I & * \\ I & X \end{bmatrix} \succ 0 \end{aligned} \quad (14)$$

のように、 X , γ に関する LMI が得られる。したがって、LMI (13), (14) 式を満足する決定変数 X , F , γ が求まれば、(9) 式を満足する (10) 式のゲインは $K = FX^{-1}$ で与えられる。

2.2 数値計算問題

MATLAB 上で動作する SDP ソルバを利用することで、以下の問題を効率よく数値的に解くことができる。

- (i) 凸可解問題: LMI (1) 式を満足する決定変数 $\xi = [\xi_1 \dots \xi_k]^T$ を求める。
- (ii) 凸最適化問題: $c = [c_1 \dots c_k]^T$ が与えられたとき、LMI (1) 式を満足し、さらに、線形目的関数

$$E = c^T \xi = c_1 \xi_1 + \dots + c_k \xi_k \quad (15)$$

を最小化する決定変数 $\xi = [\xi_1 \dots \xi_k]^T$ および E

の最小値を求める。

上記の問題設定は汎用性が高く、多くの制御系解析・設計問題がこれらに帰着できる。たとえば、例題 1 の安定解析問題は、LMI を (3), (4) 式、決定変数を P とした凸可解問題である。また、例題 2 において $\gamma > 0$ の最小化を行うコスト保証制御問題は、LMI を (13), (14) 式、決定変数を X , F , γ , 線形目的関数を $E = \gamma$ とした凸最適化問題である。

2.3 LMI を利用した制御系解析・設計の利点

従来、Riccati 方程式などの方程式条件に基づく制御系解析・設計が主流であった。それに比べて、LMI に基づく制御系解析・設計は、以下のような利点がある。

- (i) 多目的制御 [7,3,5]: 複数の設計仕様を満足させる多目的制御の実現が容易である。たとえば、指定した極領域に閉ループ極を拘束したうえで、 H_∞ 制御などにおける評価関数の最小化を行うことが可能である。
- (ii) ロバスト制御 [3,5]: 線形パラメータ変動 (LPV: Linear Parameter Varying) モデルなどの多項式型モデルやポリトープ型モデルで記述できる、不確かな (あるいは時変) パラメータを含む制御対象に対し、ロバスト制御系の解析や設計が容易である。
- (iii) ゲインスケジューリング制御 [6,3,5]: 上述の制御対象に対し、コントローラゲインをパラメータに応じて変化させることで、制御性能の向上を目指すゲインスケジューリング (GS: Gain Scheduling) 制御系の設計が容易である。

3. SeDuMi/YALMIP のインストール

ここでは、Windows OS での使用を前提として、インストール方法を説明する。

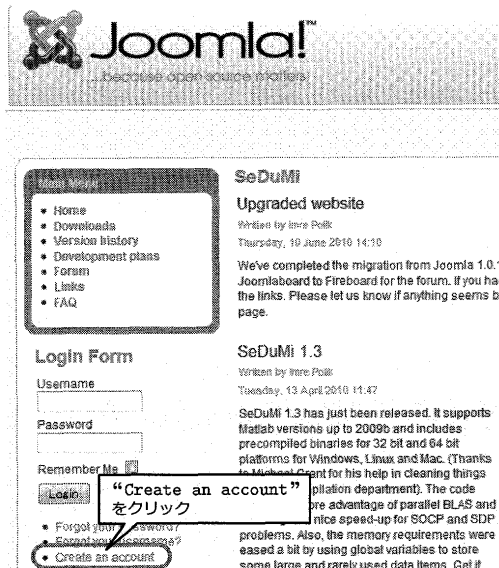
3.1 SDP ソルバ SeDuMi

SeDuMi (Self-Dual-Minimisation Package) は、Jos F. Sturm により開発されたフリーウェアの SDP ソルバである [10]。残念ながら彼は若くしてこの世を去ったが、現在、Lehigh 大学のグループに開発が引き継がれている。

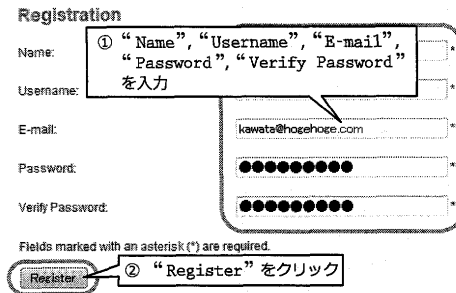
SeDuMi Ver.1.3 を入手したい場合、まず、

- <http://sedumi.ie.lehigh.edu/>

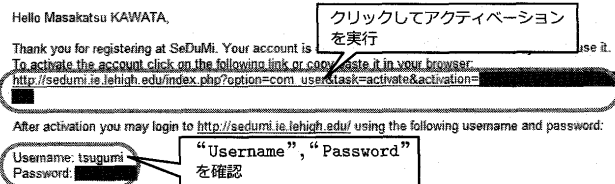
にアクセスし、ユーザ登録を行う必要がある。第 1 図にユーザ登録の手順を示す。ユーザ登録が完了した後、ログインし、Main Menu から “Downloads” を選択して Downloads ページへ移動する。つぎに、Categories から “Official SeDuMi releases” を選択して Releases ページへ移動する。最後に、Releases から “SeDuMi 1.3” を選択して Document license ページへ移動する。ライセンス内容を確認し、ページ最下部の “I agree.” をチェックした後、“Click here to proceed” をク



(a) ユーザ登録用のページへの移動



(b) ユーザ情報の入力



(c) 受信したメールとアクティベーションの実行

第1図 SuDeMi のユーザ登録の手順

リックすることで、zip ファイル “SeDuMi_1.3.zip” をダウンロードする。

3.2 LMI パーサ YALMIP

YALMIP (Yet Another LMI Parser) とは、LMI を MATLAB 上の簡便な記述で表現し、しかも、SeDuMi, SDPT3, LMILAB などといったさまざまな SDP ソルバを統一的に利用可能とする LMI パーサであり、Johan Löfberg により開発された [12].

YALMIP の最新バージョンを入手したい場合、まず、

- <http://users.isy.liu.se/johanl/yalmip/>

から、“Download” ページに移動する。そこで、Latest release の “YALMIP R*****” (***** にはリリースされた年月日が入る) をクリックし、zip ファイル “YALMIP.zip” をダウンロードする。

3.3 パスの設定

ダウンロードした zip ファイル “SeDuMi_1.3.zip” および “YALMIP.zip” を適当なフォルダに解凍する。たとえば、C ドライブのフォルダ “hoge” の中にこれらのファイルが解凍され、“SeDuMi_1.3”、“yalmip” というフォルダが生成されているものとする。フォルダ内のファイルを利用できるようにするには、M ファイル

M ファイル “lmi_path.m”：パスの設定

```
1 addpath(genpath('C:\hoge\SeDuMi_1.3'))
2 addpath(genpath('C:\hoge\yalmip'))
```

を実行して、これらフォルダにパスを通す必要がある。M ファイル “lmi_path.m” を実行した後、

```
>> path
```

と入力することで、パスが通ったことを確認できる。

また、ユーザが管理者権限をもっているのであれば、MATLAB の起動時にパスを自動的に通すことができる。たとえば、MATLAB が C ドライブのフォルダ

“MATLAB_R2010b” にインストールされているのであれば、“C:\MATLAB_R2010b\toolbox\local” にある M ファイル “matlabrc.m” の最下部に M ファイル “lmi_path.m” の内容を追加すればよい。

4. SeDuMi/YALMIP を利用した LMI の求解

4.1 LMI 求解の手順

YALMIP を利用した LMI の記述と LMI の求解のための手順を以下に示す。

ステップ1 システム行列（状態方程式 (8) 式における A, B など）や重み行列（評価関数 (9) 式における Q, R など）を定義する。

ステップ2 関数 “sdpvar” を利用して、決定変数を定義する。たとえば、スカラーの決定変数 $\gamma \in \mathbb{R}$ を定義するには、

- $\gamma = \text{sdpvar}(1);$
- $\gamma = \text{sdpvar};$

のいずれかを記述する。対称行列 $X = X^T \in \mathbb{R}^{n \times n}$ の決定変数を定義するには、

- $X = \text{sdpvar}(n,n,'symmetric');$
- $X = \text{sdpvar}(n,n,'sy');$
- $X = \text{sdpvar}(n,n);$
- $X = \text{sdpvar}(n);$

のいずれかを、対称とは限らない正方行列 $Y \in \mathbb{R}^{n \times n}$ の決定変数を定義するには、

- $Y = \text{sdpvar}(n,n,'full');$
- $Y = \text{sdpvar}(n,n,'f');$

のいずれかを、長方形行列 $Z \in \mathbb{R}^{n \times m}$ の決定変数を定義するには、以下のいずれかを記述する。

- $Z = \text{sdpvar}(n,m,'full');$
- $Z = \text{sdpvar}(n,m,'f');$
- $Z = \text{sdpvar}(n,m); (n \neq m)$

また, $S_{11} = S_{11}^T \in \mathbb{R}^{n \times n}$, $S_{21} \in \mathbb{R}^{m \times n}$, $S_{22} \in \mathbb{R}^{m \times m}$ により構成される行列

$$S = \begin{bmatrix} S_{11} & 0 \\ S_{21} & S_{22} \end{bmatrix}$$

を定義するには, 以下のように記述する.

- $$\begin{cases} S_{11} = \text{sdpvar}(n,n,'sy'); \\ S_{21} = \text{sdpvar}(m,n,'f'); \\ S_{22} = \text{sdpvar}(m,m,'f'); \\ S = [S_{11} \text{zeros}(n,m) \\ S_{21} \quad S_{22} \quad]; \end{cases}$$

ステップ3 関数“set”を利用して, LMI を定義する. まず, LMI の定義を初期化するために

- LMI = set([]);

と記述する. そのうえで, たとえば,

- $$\begin{cases} \text{LMI} = \text{LMI} + \text{set}(P \cdot A + A' \cdot P < 0); \\ \text{LMI} = \text{LMI} + \text{set}(P > 0); \end{cases}$$

のように記述すると, LMI (3), (4) 式を定義することができる.

ステップ4 関数“solvesdp”を利用して, LMI の決定変数を求める. たとえば, 定義された LMI に対して, デフォルトの SDP ソルバにより凸可解問題を解くには,

- solvesdp(LMI)

と記述し, 線形目的関数を $E = \gamma$ (gamma) とした凸最適化問題を解くには,

- solvesdp(LMI, gamma)

と記述する. なお, SeDuMi がインストールされている場合, これがデフォルトの SDP ソルバとなる. SDP ソルバの種類, 反復計算の最大回数などの設定を行いたい場合, 関数“sdpsettings”を使用する. たとえば, SDP ソルバの種類を SeDuMi や LMILAB, 反復計算の最大回数を 1,000 とするには,

- $$\begin{cases} \text{ops} = \text{sdpsettings}; \\ \text{ops.solver} = 'sedumi'; \\ \text{ops.sedumi.maxiter} = 1000; \end{cases}$$
- $$\begin{cases} \text{ops} = \text{sdpsettings}; \\ \text{ops.solver} = 'lmilab'; \\ \text{ops.lmilab.maxiter} = 1000; \end{cases}$$

のように設定した後, 凸可解問題の場合,

- solvesdp(LMI, [], ops)

と記述し, 凸最適化問題の場合,

- solvesdp(LMI, gamma, ops)

と記述する.

ステップ5 関数“double”を利用して, 得られた決定変数を倍精度に変換する. たとえば, 関数“solvesdp”により得られた決定変数 P は

- P_feas = double(P)

のように入力することで, 値を確認できる.

ここで示した関数について, さらに詳しい情報を知り

たい場合は, コマンドウィンドウで“help sdpvar”などと入力するか, ホームページ [12] を参照されたい.

以上の説明を基に, 例題 1 の安定解析問題に対する M ファイルを作成する. (3) 式は定数行列項を含まない LMI であり, A が不安定行列の場合でも, $P \equiv 0$ とすれば $\text{He}[PA] \equiv 0$ となる. したがって, このままでは, 正確な判定が期待できない. そこで, (4) 式の代わりに

$$P > I (> 0) \quad (16)$$

を用いた M ファイルを以下に示す¹.

M ファイル“lyapunov.m”: 安定解析問題

```
1 clear; format compact      ... 初期化
2 % --- ステップ 1 -----
3 A = [ 0 1;
4       -2 -3 ];              ... A = [ 0 1;
5                               ...      -2 -3 ]
6 % --- ステップ 2 -----    (A の固有値: -1, -2)
7 P = sdpvar(2,2,'sy');      ... P = P^T = [ p11 p12;
8                               ...      p12 p22 ]
9 % --- ステップ 3 -----
10 LMI = set([]);              ... LMI の初期化
11 M = P*A + A'*P;             ... M := He[PA]
12 LMI = LMI + set(M < 0);      ... LMI (3) 式
13 LMI = LMI + set(P > eye(2)); ... LMI (16) 式
14 % --- ステップ 4 -----
15 sol = solvesdp(LMI);         ... LMI の求解 (凸可解問題)
16 % -----
17 if sol.problem ~= 1         ... problem ≠ 1 であれば,
18     % --- ステップ 5 -----
19     P_feas = double(P);      ... 得られた解を倍精度に変換
20     sign(eig(double(-M)))    ... -M の固有値の符号を計算
21     sign(eig(double(P)))     ... P の固有値の符号を計算
22 end
```

18, 19 行目では, LMI (3), (16) 式が可解であった場合, 関数“eig”により P , $-M$ の固有値を求め, 関数“sign”でその符号を調べている. M ファイル“lyapunov.m”を実行した結果を以下に示す.

```
SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
: <省略>
sol =
  yalmiptime: 0.2400
  solvertime: 0.0680
  info: 'No problems detected (SeDuMi-1.3)'
  problem: 0      ... 凸可解問題は可解である
```

¹別の対処方法を以下に示す. まず, “lyapunov.m”の 11 行目の後に

```
LMI = LMI + set(trace(P) == 1); ... tr[P] = 1
```

を書き加え, 制約条件 $\text{tr}[P] = 1$ を加えることで, $P \equiv 0$ となることを回避する. つぎに, “lyapunov.m”の 10, 11 行目を

```
ep = 1e-7;                      ... ε = 10-7
LMI = LMI + set(M < - ep*eye(2)); ... M < -εI (< 0)
LMI = LMI + set(P > ep*eye(2)); ... P > εI (> 0)
```

と書き換えるか, あるいは, “lyapunov.m”の 13 行目を

```
ops = sdpsettings; ops.shift = 1e-7;
solvesdp(LMI, [], ops)
```

と書き換える (関数“solvesdp”のオプション shift を設定する). このように記述すると, 十分小さな正数 ε を用いて M や P の固有値をわずかにシフトさせることにより, 数値計算による正定性や負定性の判定誤差に対処することができる.


```

dimacs: [1.0427e-012 0 ... <省略> ... 2.8037e-012]
P_feas = ... Lyapunov 不等式 (3) 式の正定対称解 P
7.9115 1.1877
1.1877 2.4300
ans = -M の固有値の符号がすべて正なので M < 0 (-M > 0)
1 1
ans = P の固有値の符号がすべて正なので P > 0
1 1

```

この例の場合，“info”に“No problems detected” (“problem”に“0”)と表示される。これらのメッセージは，LMI (3), (4) 式を満足する解 P が得られ，システムが安定であることを示している。また，得られた P と $-M = -\text{He}[PA]$ が正定であることは，これらの固有値がすべて正であることから確認できる。

つぎに，“lyapunov.m”の3, 4行目を

```

3 A = [ 0 1 ... A = [ 0 1
4 2 -3 ]; ... [ 2 -3 ]
(A の固有値: (-3 ± √17)/2)

```

に変更して実行すると，以下ようになる。

```

SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
<省略>
info: 'Infeasible problem (SeDuMi-1.3)'
problem: 1 ... 凸可解問題は可解ではない

```

この例の場合，“info”に“Infeasible problem” (“problem”に“1”)と表示される。これらのメッセージは，LMI (3), (4) 式を満足する解 P が存在せず，システムが不安定であることを示している。

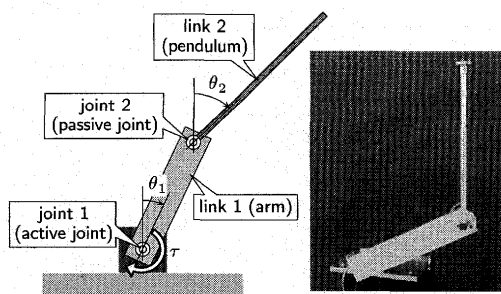
4.2 アーム型倒立振子の GS 制御

4.2.1 LPV モデル

第2図に示すアーム型倒立振子は，速度制御型モータドライバにより駆動されており，その非線形モデルは，

$$\begin{cases} \ddot{\theta}_1 + a_s \dot{\theta}_1 = b_s v \\ \alpha_3 \cos \theta_{12} \ddot{\theta}_1 + \alpha_2 \ddot{\theta}_2 \\ -\alpha_3 \dot{\theta}_1^2 \sin \theta_{12} - \alpha_5 \sin \theta_2 - c_2 \dot{\theta}_{12} = 0 \end{cases} \quad (17)$$

となる。ただし， $\theta_{12} = \theta_1 - \theta_2$ であり， v は速度指令電圧である。また，物理パラメータの意味については後述の M ファイル “adip.m” に示す。ここで，振子が倒立状態 ($\theta_2 = 0$) の近傍で動作すると仮定し， $\cos \theta_{12} \doteq \cos \theta_1$ ， $\sin \theta_2 \doteq \theta_2$ のように近似する。さらに，遠心力・コリオリ力の項を無視し， $\rho(t) := 1 - \cos \theta_1(t)$ と定義すると，状態変数を $x = [\theta^T \dot{\theta}^T]^T$ ($\theta = [\theta_1 \ \theta_2]^T$)，操作量を $u = v$ ，係数行列を



第2図 アーム型倒立振子

$$\begin{aligned} [A(\rho) \ B(\rho)] &= [A_{(0)} \ B_{(0)}] + \rho [A_{(1)} \ B_{(1)}] \\ &= \begin{bmatrix} 0 & I & 0 \\ A_{21} & A_{22(0)} & B_{2(0)} \end{bmatrix} + \rho \begin{bmatrix} 0 & 0 & 0 \\ 0 & A_{22(1)} & B_{2(1)} \end{bmatrix} \end{aligned}$$

とした LPV モデル

$$\dot{x}(t) = A(\rho(t))x(t) + B(\rho(t))u(t), \quad x(0) = x_0 \quad (18)$$

が得られる。この LPV モデルの係数行列を定義した M ファイルを以下に示す。

```

M ファイル “adip.m”：アーム型倒立振子のモデル
1 m1 = 3.06e-001; ... m1: アームの質量
2 l1 = 1.20e-001; ... l1: アームの軸から重心までの長さ
3 L1 = 2.27e-001; ... L1: アームの長さ
4 J2 = 1.38e-003; ... J2: 振子の重心まわりの慣性モーメント
5 m2 = 1.04e-001; ... m2: 振子の質量
6 l2 = 1.75e-001; ... l2: 振子の軸から重心までの長さ
7 c2 = 1.86e-004; ... c2: 振子の軸の粘性摩擦係数
8 g = 9.81e+000; ... g: 重力加速度
9 as = 6.25e+000; ... as, bs: アーム, モータ, モータドライバ
bs = 1.56e+001; ... の特性より決まる定数
11 %
12 a2 = J2 + m2*l2^2; ... a2 = J2 + m2*l2^2
13 a3 = m2*L1*l2; ... a3 = m2*L1*l2
14 a4 = (m1*l1 + m2*L1)*g; ... a4 = (m1*l1 + m2*L1)*g
15 a5 = m2*l2*g; ... a5 = m2*l2*g
16 %
17 A21 = [ 0 0 ... A21
18 0 a5/a2 ];
19 A22_0 = [ -as 0 ... A22(0)
20 (as*a3+c2)/a2 -c2/a2 ];
21 A22_1 = [ 0 0 ... A22(1)
22 -as*a3/a2 0 ];
23 B2_0 = [ bs ... B2(0)
24 -bs*a3/a2 ];
25 B2_1 = [ 0 ... B2(1)
26 bs*a3/a2 ];
27 %
28 A_0 = [ zeros(2,2) eye(2) ... A(0) = [ 0 I
29 A21 A22_0 ]; ... [ A21 A22(0)
30 A_1 = [ zeros(2,2) zeros(2,2) ... A(1) = [ 0 0
31 zeros(2,2) A22_1 ]; ... [ 0 A22(1)
32 B_0 = [ zeros(2,1) ... B(0) = [ 0
33 B2_0 ]; ... [ B2(0)
34 B_1 = [ zeros(2,1) ... B(1) = [ 0
35 B2_1 ]; ... [ B2(1)

```

4.2.2 GS コントローラ設計

アーム角度が $|\theta_1(t)| \leq \bar{\theta}_1$ で動作するとき，時変パラメータ $\rho(t)$ の範囲は $0 \leq \rho(t) \leq \bar{\rho} := 1 - \cos \bar{\theta}_1$ となる。同様に， $|\theta_1(0)| \leq \bar{\theta}_{10}$ に対して，初期値 $\rho_0 := \rho(0)$ の範囲は $0 \leq \rho_0 \leq \bar{\rho}_0 := 1 - \cos \bar{\theta}_{10}$ となる。また， $|\dot{\rho}(t)| \leq \delta$ であるとする。このとき，LPV モデル (18) 式に対して，例題2のコスト保証制御問題における有界条件 (9) 式を満足する，状態フィードバック形式の GS コントローラ

$$u(t) = K(\rho(t))x(t) \quad (19)$$

を設計する問題を考える。

この問題の可解条件は，LMI (13), (14) 式をパラメータに依存させた，パラメータ依存 LMI (PDLMI: Parameter Dependent LMI) に帰着できる。具体的には，多項式の行列値関数

$$X(\rho) = \sum_{i=0}^{k_1} \rho^i X_{(i)}, \quad F(\rho) = \sum_{i=0}^{k_2} \rho^i F_{(i)} \quad (20)$$

に対して, PDLMI は,

$$M_1(\xi, \rho, \rho^2, \dots, \rho^{k_1}) := X(\rho) \succ 0 \quad (21)$$

$$M_2(\xi, \rho, \rho^2, \dots, \rho^k, \dot{\rho}) := \begin{bmatrix} -(\text{He}[\Phi(\rho)] - \dot{X}(\rho)) & * & * \\ Q_h X(\rho) & I & * \\ RF(\rho) & 0 & R \end{bmatrix} \succ 0 \quad (22)$$

$$M_3(\xi, \rho_0, \rho_0^2, \dots, \rho_0^{k_1}) := \begin{bmatrix} \gamma I & * \\ I & X(\rho_0) \end{bmatrix} \succ 0 \quad (23)$$

となる. ただし, k は k_1+1, k_2+1 のうちで大きい方の数, 決定変数 ξ は $X_{(i)}, F_{(i)}$ の要素と γ である. とくに, $k_1=k_2=1$ とすると, PDLMI (21)–(23) 式は,

$$M_1(\xi, \rho) = X_{(0)} + \rho X_{(1)} \succ 0 \quad (24)$$

$$M_2(\xi, \rho, \rho^2, \dot{\rho}) = \sum_{i=0}^2 \rho^i M_{2(i0)}(\xi) + \dot{\rho} M_{2(01)}(\xi) \succ 0 \quad (25)$$

$$M_3(\xi, \rho_0) = M_{3(0)}(\xi) + \rho_0 M_{3(1)}(\xi) \succ 0 \quad (26)$$

という形式となる. ここで, PDLMI (24)–(26) 式はそれぞれ, $\rho, \dot{\rho}, \rho_0$ に関して 1 次であるから, これらは変動 $0 \leq \rho \leq \bar{\rho}, -\delta \leq \dot{\rho} \leq \delta, 0 \leq \rho_0 \leq \bar{\rho}_0$ の端点条件

$$M_1(\xi, 0) \succ 0, M_1(\xi, \bar{\rho}) \succ 0 \quad (27)$$

$$M_2(\xi, \rho, \rho^2, -\delta) \succ 0, M_2(\xi, \rho, \rho^2, \delta) \succ 0 \quad (28)$$

$$M_3(\xi, 0) \succ 0, M_3(\xi, \bar{\rho}_0) \succ 0 \quad (29)$$

で置き換えることができる. さらに, PDLMI (28) 式は, パラメータ ρ に関して 2 次であるが, 以下の補題を利用し, 有限個の LMI に帰着できることが渡辺らにより示されている [15,6].

【補題 2】 決定変数が ξ , パラメータ ρ の変動範囲が $0 \leq \rho \leq \bar{\rho}$ であるような次式の PDLMI を考える.

$$M(\xi, \rho, \rho^2, \dots, \rho^k) = \sum_{i=0}^k \rho^i M_{(i)}(\xi) \succ 0 \quad (30)$$

このとき, 変動 $\mathcal{U} = \{(\rho, \rho^2, \dots, \rho^k) \mid 0 \leq \rho \leq \bar{\rho}\}$ を囲む凸多面体を \mathcal{T} とすると, (30) 式の $(\rho, \rho^2, \dots, \rho^k)$ を \mathcal{T} のすべての頂点で置き換えた連立 LMI を同時に満足する決定変数 ξ は, PDLMI (30) 式を満足する.

簡単のため, 変動 $\mathcal{U} = \{(\rho, \rho^2) \mid 0 \leq \rho \leq \bar{\rho}\}$ を囲む凸多面体 \mathcal{T} として, 頂点を $(0,0), (\bar{\rho},0), (0,\bar{\rho}^2), (\bar{\rho},\bar{\rho}^2)$ とした長方形を考える. このとき, PDLMI (28) 式を解く問題は, この長方形の頂点における連立 LMI

$$\begin{cases} M_2(\xi, 0, 0, -\delta) \succ 0, M_2(\xi, 0, 0, \delta) \succ 0 \\ M_2(\xi, \bar{\rho}, 0, -\delta) \succ 0, M_2(\xi, \bar{\rho}, 0, \delta) \succ 0 \\ M_2(\xi, 0, \bar{\rho}^2, -\delta) \succ 0, M_2(\xi, 0, \bar{\rho}^2, \delta) \succ 0 \\ M_2(\xi, \bar{\rho}, \bar{\rho}^2, -\delta) \succ 0, M_2(\xi, \bar{\rho}, \bar{\rho}^2, \delta) \succ 0 \end{cases} \quad (31)$$

を解く問題に帰着される.

以上のことをまとめると, GS コントローラを設計するための M ファイルは以下ようになる.

M ファイル “gs.m”: GS コントローラの設計

```
1 clear; format compact      ... 初期化
2 % -----
3 adip                        ... “adip.m” の実行
4 % -----
5 q1 = 5;                     ... q1 > 0 の設定
6 q2 = 1;                     ... q2 > 0 の設定
7 q3 = 0.01;                  ... q3 > 0 の設定
8 q4 = 0.01;                  ... q4 > 0 の設定
9 R = 1;                      ... R > 0 の設定
10 Qh = diag([sqrt(q1) sqrt(q2) sqrt(q3) sqrt(q4)]);
11 Q = Qh'*Qh;                 ... Q = Qh^T Qh > 0
12 % -----
13 th1_max = 60*pi/180;        ... θ1
14 th10_max = 50*pi/180;       ... θ10
15 drho_max = 1.4;             ... δ
16 rho_max = 1 - cos(th1_max); ... ρ = 1 - cosθ1
17 rho0_max = 1 - cos(th10_max); ... ρ0 = 1 - cosθ10
18 % -----
19 n = 4; p = 1;               ... n: x の次数, p: u の次数
20 X_0 = sdpvar(n,n,'sy');     ... X(0) ∈ ℝ^{n×n}: 対称行列
21 X_1 = sdpvar(n,n,'sy');     ... X(1) ∈ ℝ^{n×n}: 対称行列
22 F_0 = sdpvar(p,n,'f');      ... F(0) ∈ ℝ^{p×n}
23 F_1 = sdpvar(p,n,'f');      ... F(1) ∈ ℝ^{p×n}
24 gamma = sdpvar(1);          ... γ ∈ ℝ
25 % -----
26 LMI = set([]);              ... LMI の初期化
27 % -----
28 rho = [0 1]*rho_max;        ... 端点: ρ = 0, ρ̄
29
30 for i = 1:2
31     X = X_0 + rho(i)*X_1;    ... X(ρ) = X(0) + ρX(1)
32     M1 = X;
33     LMI = LMI + set(M1 > 0); ... LMI (27) 式の追加
34 end
35 % -----
36 rho1 = [0 1 0 1]*rho_max;   ... T の頂点: (ρ, ρ^2) = (0, 0),
37 rho2 = [0 0 1 1]*rho_max^2; ... (ρ̄, 0), (0, ρ̄^2), (ρ̄, ρ̄^2)
38 drho = [-1 1]*drho_max;     ... 端点: ρ̇ = -δ, δ
39
40 Phi_0 = A_0*X_0 + B_0*F_0;   ... Φ(0)
41 Phi_1 = A_1*X_0 + B_1*F_0 + A_0*X_1 + B_0*F_1; ... Φ(1)
42 Phi_2 = A_1*X_1 + B_1*F_1;   ... Φ(2)
43
44 for i = 1:4
45     for j = 1:2
46         Phi = Phi_0 + rho1(i)*Phi_1 + rho2(i)*Phi_2;
47         ... Φ(ρ) = ∑_{i=0}^2 ρ^i Φ(i)
48         X = X_0 + rho1(i)*X_1; ... X(ρ) = X(0) + ρX(1)
49         F = F_0 + rho1(i)*F_1; ... F(ρ) = F(0) + ρF(1)
50         dX = drho(j)*X_1;     ... X(ρ) = ρ̇X(1)
51
52         M2 = [-(Phi+Phi'-dX) X*Qh F'*R
53               Qh*X eye(n) zeros(n,p)
54               R'*F zeros(p,n) R];
55         LMI = LMI + set(M2 > 0); ... LMI (31) 式の追加
56     end
57 end
58 % -----
59 rho0 = [0 1]*rho0_max;       ... 端点: ρ0 = 0, ρ̄0
60
61 for i = 1:2
62     X_rho0 = X_0 + rho0(i)*X_1; ... X(ρ0) = X(0) + ρ0X(1)
63     M3 = [gamma*eye(n) eye(n)
64           eye(n) X_rho0];
65     LMI = LMI + set(M3 > 0); ... LMI (29) 式の追加
66 end
67 % -----
68 sol = solvesdp(LMI,gamma)    ... LMI の求解 (γ を線形目的
69 % -----
70 if sol.problem ~= 1          ... problem ≠ 1 であれば,
71     gamma_opt = double(gamma) ... γ を倍精度で表示
72     X_0_opt = double(X_0)    ... X(0) を倍精度で表示
73     X_1_opt = double(X_1)    ... X(1) を倍精度で表示
74     F_0_opt = double(F_0)    ... F(0) を倍精度で表示
75     F_1_opt = double(F_1)    ... F(1) を倍精度で表示
```

```

78 % -----
79 format short e
80 pres = checkset(LMI)    ... 連立 LMI (27), (31), (29) 式
81 format short           ... の左辺の最小固有値を計算
82 end

```

M ファイル “gs.m” を実行すると、

```

SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
: <省略>
info: 'Numerical problems (SeDuMi-1.3)'
problem: 4    ... 凸最適化問題の解は数値的な問題があるか
: <省略>
gamma_opt =   ... 線形目的関数  $\gamma$  の最小値 ( $\gamma = 193.8820$ )
193.8820
: <省略>
pres =
7.5018e-003    ...  $M_1(\xi, 0)$  の最小固有値
6.1005e-004    ...  $M_1(\xi, \bar{p})$  の最小固有値
-8.8716e-010   ...  $M_2(\xi, 0, 0, -\delta)$  の最小固有値
4.8249e-009    ...  $M_2(\xi, 0, 0, \delta)$  の最小固有値
-7.4369e-010   ...  $M_2(\xi, \bar{p}, 0, -\delta)$  の最小固有値
1.3066e-009    ...  $M_2(\xi, \bar{p}, 0, \delta)$  の最小固有値
2.8326e-009    ...  $M_2(\xi, 0, \bar{p}^2, -\delta)$  の最小固有値
1.7832e-009    ...  $M_2(\xi, 0, \bar{p}^2, \delta)$  の最小固有値
-5.1840e-010   ...  $M_2(\xi, \bar{p}, \bar{p}^2, -\delta)$  の最小固有値
-8.3788e-010   ...  $M_2(\xi, \bar{p}, \bar{p}^2, \delta)$  の最小固有値
2.3439e-003    ...  $M_3(\xi, 0)$  の最小固有値
-1.3232e-009   ...  $M_3(\xi, \bar{p}_0)$  の最小固有値

```

という結果が得られる。このように、SDP ソルバ SeDuMi を用いて規模の大きな凸最適化問題を解くと、可解であるはずの問題であっても、“info” に “Numerical problems” (“problem” に “4”) と表示され、数値的に問題がある可能性を指摘されることがある¹。実際、上記の実行結果より、連立 LMI (27), (31), (29) 式の左辺の中で、 $M_2(\xi, 0, 0, -\delta)$, $M_2(\xi, \bar{p}, 0, -\delta)$, $M_2(\xi, \bar{p}, \bar{p}^2, -\delta)$, $M_2(\xi, \bar{p}, \bar{p}^2, \delta)$, $M_3(\xi, \bar{p}_0)$ の最小固有値は負であり、これらは正定ではない。

コスト保証制御問題において、このような結果に対処するための一つの方法は、凸最適化問題で得られた最小値 193.8820 よりも若干、大きな線形目的関数 γ の値を与えて、再度、凸可解問題を解くというものである。M ファイル “gs.m” の 24, 70 行目を書き換えた

M ファイル “gs_cfp.m” : GS コントローラ的设计

```

: <“gs.m” の 1~23 行目>
24 gamma = 194;    ...  $\gamma = 194 (> 193.8820)$  に設定
: <“gs.m” の 25~69 行目>
70 sol = solvesdp(LMI) ... LMI の求解 (凸可解問題)
: <“gs.m” の 71~82 行目>

```

を実行すると、

```

SeDuMi 1.3 by AdvOL, 2005-2008 and Jos F. Sturm, 1998-2003.
: <省略>

```

¹LMILAB は実行速度が遅いという欠点はあるが、信頼性が高い。実際、M ファイル “gs.m” の 70 行目を

```

ops = sdpsettings; ops.solver = 'lmilab';
sol = solvesdp(LMI, gamma, ops)

```

と書き換えて実行すると、連立 LMI (27), (31), (29) 式の解が得られる。このときの γ の最小値は 193.9675 となり、SeDuMi による結果よりもわずかに大きい。

```

info: 'No problems detected (SeDuMi-1.3)'
problem: 0    ... 凸可解問題は可解である
: <省略>
pres =       ... 最小固有値はすべて正
: <省略>

```

となり、連立 LMI (27), (31), (29) 式の解が得られる。

別の対処方法としては、先に説明した関数 “solvesdp” のオプション shift を設定し、凸最適化問題を解くことが考えられる。たとえば、“gs.m” の 70 行目を

```

ops = sdpsettings; ops.shift = 1e-7;
sol = solvesdp(LMI, gamma, ops)

```

に変更して実行すると、 $\gamma = 193.8861$ が得られる。このとき、“info” に “Numerical problems” (“problem” に “4”) と表示されるが、pres の値をみると、最小固有値がすべて正であり、連立 LMI (27), (31), (29) 式を満足する解が得られたことを確認できる。

4.2.3 非線形シミュレーション

Simulink を利用し、制御対象を非線形モデル (17) 式、GS コントローラを M ファイル “gs_cfp.m” により設計された (19) 式とした非線形シミュレーションを行った。シミュレーション結果を、第 3 図に示す。第 3 図には、比較のため、原点近傍における線形化モデル

$$\dot{x}(t) = A_{(0)}x(t) + B_{(0)}u(t) \quad (32)$$

に対して、最適レギュレータ理論により

```

>> K_LQ = -lqr(A_0,B_0,Q,R)
K_LQ =
2.2361    20.1077    1.8179    3.2024

```

のように設計された、線形コントローラ

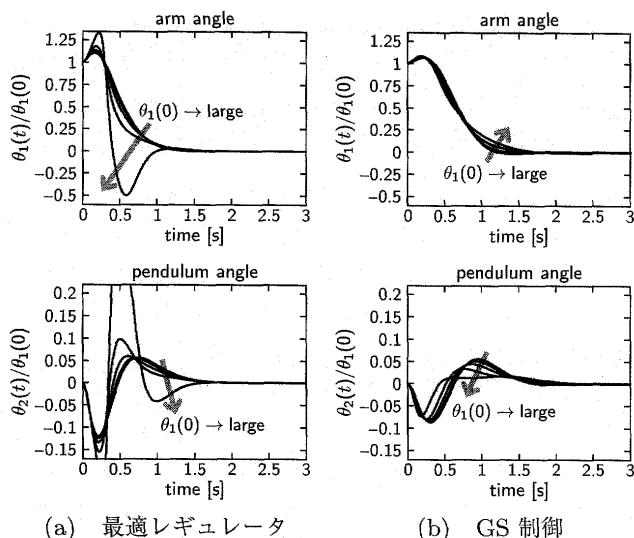
$$u(t) = K_{LQ}x(t) \quad (33)$$

を用いた結果も示している。

第 3 図 (a) に示すように、最適レギュレータでは、アームの初期角度 $\theta_1(0)$ が 30 [deg] 程度までは応答波形がほとんど変わらないが、40 [deg] を超えて 50 [deg] に近づくと振動的な応答となり、制御性能が著しく劣化する。それに対し、第 3 図 (b) に示すように、GS 制御では、アームの初期角度 $\theta_1(0)$ が 50 [deg] であっても制御性能の劣化は抑えられている。

5. おわりに

本解説では、まず、制御理論の分野で広く普及している LMI に基づく制御系解析・設計の概要を説明した。ついで、ユーザが実際に LMI を用いて制御系解析・設計を行うためのソフトウェア環境について説明した。そこでは、数値計算ソフトウェア MATLAB 上で動作し、この分野の研究者の間で最も使用されている、フリーウェアの SDP ソルバ SeDuMi や LMI パーサ YALMIP のインストール手順を示した。また、安定性解析や GS 制御系設計といった具体例を通じ、その活用の方法を示した。



第3図 シミュレーション結果 (縦軸はアームの初期角度 ($\theta_1(0) = 10, 20, 30, 40, 50$ [deg]) を基準に正規化)

本解説では、数値最適化の分野の基本的な事項のみ説明したが、二乗和 (SOS: Sum of Squares) やランダムイズドアルゴリズムといった最近の結果と数値計算ソフトウェアの活用については、本特集号の市原氏の解説[16]や、和田氏・藤崎氏の解説[17]で説明されているので、そちらを参照されたい。

なお、本解説の脱稿後、YALMIPの開発者である Dr. Johan Löfberg より、関数“set”を利用した LMI の記述 (4.1 のステップ 3 を参照) が将来のバージョンでは保証されない可能性があるとのことご指摘をいただいた。新しい記述方法では、関数“set”の代わりに、括弧 “[”, “]” を利用し、第 1 表のように記述する。現時点での最新バージョン (Ver.R20110318) では、関数“set”を利用可能であり、本解説だけでなく本特集号の総説および解説[16,17]における M ファイルは、LMI が古い記述形式で表現されていることを、念のため付記しておく。

本解説で使った MATLAB の M ファイルや Simulink モデルおよび補足事項は、

- <http://www.maizuru-ct.ac.jp/control/kawata/iscie/iscie.html>

で公開する。本解説が LMI に基づく制御系解析・設計を導入するための一助となれば幸いである。

(2011 年 1 月 11 日受付)

参考文献

- [1] S. Boyd, L. El Ghaoui, E. Feron and V. Balakrishnan: *Linear Matrix Inequalities in System and Control Theory*, SIAM (1998) (available from <http://www.stanford.edu/~boyd/lmibook/lmibook.pdf>)
- [2] 岩崎: LMI と制御, 昭晃堂 (1997)
- [3] 藤森: ロバスト制御, コロナ社 (2001)
- [4] 増淵, 小原: LMI によるシンセシスと半正定値計画問

- 題; 計測と制御, Vol. 35, No. 10, pp. 743–750 (1996)
- [5] 浅井: リレー解説 「ロバスト制御の基礎から最先端まで」 第 4 回: 線形行列不等式, 第 5 回: 線形 H_∞ 制御系設計, 第 6 回: LMI に基づく線形ロバスト制御系解析・設計; 計測と制御, Vol. 42, No. 10–12, pp. 859–866, 958–964, 1032–1038 (2003)
- [6] 渡辺, 内田: ゲインスケジューリング — 適応/非線形制御への展開; システム/制御/情報, Vol. 42, No. 6, pp. 306–311 (1998)
- [7] C. Scherer, P. Gahinet and M. Chilali: Multiobjective output-feedback control via LMI optimization; *IEEE Transactions on Automatic Control*, Vol. 42, No. 7, pp. 896–911 (1997)
- [8] P. Gahinet, A. Nemirovski, A. J. Laub and M. Chilali (梶原 監訳): *LMI Control Toolbox for Use with MATLAB*, The MathWorks (1995)
- [9] 梶原: LMI ベース制御系 CAD とその応用; 計測と制御, Vol. 35, No. 10, pp. 737–742 (1996)
- [10] J. F. Sturm: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones (Updated for Version 1.05); *Optimization Methods and Software*, Vol. 11, No. 1, pp. 625–653 (1999) (available from <http://sedumi.ie.lehigh.edu/>)
- [11] K. C. Toh, M. J. Todd and R. H. Tütüncü: SDPT3 — a MATLAB software package for semidefinite programming, version 1.3; *Optimization Methods and Software*, Vol. 11, No. 1, pp. 545–581 (1999) (available from <http://www.math.nus.edu.sg/~matttohkc/sdpt3.html>)
- [12] J. Löfberg: YALMIP: a toolbox for modeling and optimization in MATLAB; *Proceedings of 2004 IEEE International Symposium on Computer Aided Control Systems Design*, pp. 284–289 (2004) (available from <http://users.isy.liu.se/johanl/yalmip/>)
- [13] M. Grant, S. Boyd and Y. Ye: Disciplined convex programming (Chapter in *Global Optimization: from Theory to Implementation*), Springer, pp. 155–210 (2006) (available from <http://cvxr.com/cvx/>)
- [14] Y. Fujisaki and Y. Oishi: Guaranteed cost regulator design: a probabilistic solution and a randomized algorithm; *Automatica*, Vol. 43, No. 2, pp. 317–324 (2007)
- [15] R. Watanabe, K. Uchida and M. Fujita: A new LMI approach to analysis of linear systems with scheduling parameter — reduction to finite number of LMI conditions —; *Proceedings of the 35th IEEE Conference on Decision and Control*, pp. 1663–1665 (1996)
- [16] 市原: 二乗和に基づく制御系解析・設計; システム/制御/情報, Vol. 55, No. 5, pp. 174–180 (2011)
- [17] 和田, 藤崎: ランダムイズドアルゴリズムによるロバスト制御系解析・設計; システム/制御/情報, Vol. 55, No. 5, pp. 181–188 (2011)

第1表 YALMIP における LMI の新しい記述形式

古い記述形式 (関数 “set” を利用)	新しい記述形式 (括弧 “[”, “]” を利用)	備考
$LMI = \text{set}([]);$	$LMI = [];$	LMI の初期化
$LMI = LMI + \text{set}(M1 > 0);$ $LMI = LMI + \text{set}(M2 < 0);$	$LMI = [LMI, M1 > 0];$ $LMI = [LMI, M2 < 0];$	$LMI \begin{cases} M1 > 0 \\ M2 < 0 \end{cases}$ の追加

著者略歴

かわ た
川 田まさ かつ
昌 克 (正会員)

本学会誌 p. 164 参照

えび はら よし お
蛭 原 義 雄 (正会員)

2002 年 3 月京都大学大学院工学研究科
電気工学専攻博士後期課程修了。同年 4 月
より京都大学大学院工学研究科助手，同研
究科講師を経て，2010 年 3 月同研究科准
教授となり現在に至る。数値最適化手法を
用いた制御系の解析，設計に関する研究に

従事。京都大学博士（工学）。計測自動制御学会，IEEE など
の会員。