

## 一般化 KYP 補題に基づく制御系設計ツール†

塩形大輔\*・原辰次\*

**ABSTRACT** The generalized Kalman-Yakubovich-Popov (GKYP) lemma provides a unified LMI characterization for finite frequency properties of dynamical systems. This paper is concerned with a soft wear tool for control system analysis and design based on the GKYP lemma. We first clarify the class of synthesis problems which can be reduced to LMI optimization problems. We then discuss about a tool developed on MATLAB for the purpose, where we explain the data structure and the GKYP-LMI translator which automatically creates LMIs corresponding to the design specifications. This enables control engineers to just write down design specifications directly without any knowledge about the GKYP theory and MATLAB syntax for multi-objective control system design. Finally the effectiveness is confirmed by showing a multi-objective PID controller design.

## 1. はじめに

近年, 計算機の普及と高速化によって, 制御系の解析・設計に関する問題を数値的に解くことが可能になり, そのための計算法が数多く提案されてきた. また, 制御系解析・設計のための支援パッケージやツールボックスに関しては, 幅広く使われている MATLAB ベースのもの以外に, 制御に関連する数値計算を集めたライブラリー SLICOT<sup>1)</sup> や制御理論で現れる式を可能な限り直接的に記述可能な MaTX<sup>2)</sup> など, フリーのソフトウェアも利用されている. 最新の情報は, IEEE Control Systems Society の Computer Aided Control System Design という technical committee の Web Page (<http://www.laas.fr/cacsd/numerics/index.shtml>) から得ることができる.

数値最適化による解析・設計という観点では, 特に 1990 年頃から, 線形行列不等式 (LMI: Linear Matrix Inequality) が制御系設計の道具として有用であるとして注目されるようになってきた<sup>3,4)</sup>. その理由は, LMI を制約条件とした数値最適化問題は凸計画問題 (半正定値計画法: SDP) として効率よく解くことができる点にある.

制御における LMI 条件のベースの一つは, KYP 補題 (Kalman-Yakubovich-Popov Lemma)<sup>5,6)</sup> である.  $H_\infty$  制御

A Soft Wear Tool for Control System Design based on Generalized KYP Lemma. By Daisuke Shiokata and Shinji Hara (The University of Tokyo).

\* 東京大学

† 2005 年 1 月 6 日受付 2005 年 6 月 28 日再受付

がその典型であり, 全周波数帯域に対して与えられた仕様を満たす制御器の設計条件を LMI 条件として書き表すことができる. しかし, 実際の設計では, 特定の周波数帯域に対してのみに何らかの特性を要求することが多い. 例えば, 感度関数は低周波域において小さくする必要があり, 相補感度関数は高周波域において小さいことが望ましい. そこで, これまでは適当な重み関数を用いることで, 必要ない帯域のゲインを相対的に低くして設計する必要があった.

しかし, この方法では特定の周波数帯域のみを厳密に取り出すことはできない. また, 重み関数を掛けたために対象のシステムの次数が高くなり, 計算量が増えるだけでなく, 設計されるコントローラも次数の高いものになってしまうという欠点があった. 近年, 著者らはこの欠点を克服するために一般化 KYP (Generalized KYP: GKYP) 補題を提案した. これにより, これまで解くことのできなかつた特定の周波数帯域のみに制約条件を与えたある種の制御系の解析・設計問題が LMI によって解けることを示した<sup>7~10)</sup>.

GKYP 補題は理論的にはかなり完成されたものではあるが, GKYP 補題に基づく制御系設計に関しては, 以下の 3 点において残されている課題は少なくない.

- LMI 最適化問題に帰着できる設計問題のクラスの明確化
  - LMI 最適化問題に帰着できない設計問題に対する解決法
  - 設計支援システムの開発
- そこで本論文では, 上記 3 点に着目し, GKYP 補題

に基づく制御系設計法の確立を目指す。本研究の主たる貢献は、GKYP補題に基づいて効率よく制御系の解析・設計を行なうためにMatlab上で開発した解析・設計ツール(LMI自動生成プログラム)にある。実際、現場の制御技術者がGKYP理論を理解し、それに基づいて設計プログラムを作成するのは非常に難しい。開発したツールは完全に構造化されており、各々の設計仕様を直接的に記述するだけで十分で、GKYP理論を習得する必要もない。また、必要となるプログラムステップ数はLMIツールボックスで直接記述する場合の10分の1以下であり、LMIツールボックスの文法を学ぶ必要もない。これらのことは、論文の最後の部分で、実際の設計例を通して具体的に示し、その有用性を確認する。

本論文の構成は以下の通りである。まず第2章において、GKYP補題の概要を説明する。次に第3章では、GKYP補題の制御系設計への応用について説明する。特に、LMI最適化問題になるシステムのクラスと設計仕様のクラスを明確にし、一巡伝達関数の周波数整形法を具体的に示す。さらに、LMI問題に帰着できない場合の取り扱いについての考察を行う。第4章では、Matlab上で開発したGKYP補題に基づいた制御設計解析・設計ツールについて説明する。全体の構成を述べた後、入出力のデータ構造を紹介し、LMI自動生成アルゴリズムについて説明する。第5章では、ツールを利用した制御系設計の一例としてPID制御系設計例をプログラムとともに紹介し、開発した解析・設計ツールの有用性を確認する。最後に第6章でまとめを行う。

本論文では、次のような表記を用いる。行列Aに対して、その転置行列、複素共役転置行列および零空間をそれぞれ $A^T, A^*, \mathcal{N}_A$ とする。エルミート行列Aに対して、 $A > (\geq) 0$ および $A < (\leq) 0$ は、それぞれAが(半)正定、(半)負定行列であることを意味する。行列Aの実部、虚部はそれぞれ $\Re(A), \Im(A)$ とする。行列A, Bのクロネッカー積を $A \otimes B$ とする。 $n \times n$ エルミート行列の集合を $\mathbf{H}_n$ とし、関数 $\sigma: \mathbf{C}^{p \times m} \times \mathbf{H}_{p+m} \rightarrow \mathbf{H}_m$ を次のように定義する。

$$\sigma(G, \Pi) = \begin{bmatrix} G \\ I_m \end{bmatrix}^* \Pi \begin{bmatrix} G \\ I_m \end{bmatrix} \quad (1)$$

2. 一般化KYP補題

本章では、文献10)で導かれている一般化KYP補題を紹介する。まず、周波数帯域を表わす集合 $\Lambda$ を、

$$\Lambda = \{ \lambda \in \mathbf{C} \mid \sigma(\lambda, \Phi) = 0, \sigma(\lambda, \Psi) \geq 0 \} \quad (2)$$

表1 連続時間システムの周波数帯域

帯域	$\omega$	中間 $\Psi$
低域	$ \omega  < \omega_l$	$\begin{bmatrix} -1 & 0 \\ 0 & \omega_l^2 \end{bmatrix}$
中間	$\omega_1 < \omega < \omega_2$	$\begin{bmatrix} -1 & j\omega_c \\ -j\omega_c & -\omega_1\omega_2 \end{bmatrix}$
高域	$ \omega  > \omega_h$	$\begin{bmatrix} 1 & 0 \\ 0 & -\omega_h^2 \end{bmatrix}$

ただし  $\omega_c = (\omega_1 + \omega_2)/2$

表2 離散時間システムの周波数帯域

帯域	$\theta$	$\Psi$
低域	$ \theta  < \theta_l$	$\begin{bmatrix} 0 & 1 \\ 1 & -2 \cos \theta_l \end{bmatrix}$
中間	$\theta_1 < \theta < \theta_2$	$\begin{bmatrix} 0 & e^{j\theta_c} \\ e^{-j\theta_c} & 2 \cos \theta_d \end{bmatrix}$
高域	$ \theta  > \theta_h$	$\begin{bmatrix} 0 & -1 \\ -1 & 2 \cos \theta_h \end{bmatrix}$

ただし  $\theta_c = (\omega_1 + \omega_2)/2, \theta_d = (\omega_2 - \omega_1)/2$

と定義する。ここで行列 $\Phi, \Psi \in \mathbf{H}_2$ を適切に定めることにより、連続時間・離散時間の低・中間・高周波帯域に対応する集合 $\Lambda$ が定義される(表1, 2を参照)。いま、集合 $\bar{\Lambda}$ を

$$\bar{\Lambda} = \begin{cases} \Lambda & (\Lambda \text{ が有界}) \\ \Lambda \cup \{\infty\} & (\text{それ以外}) \end{cases} \quad (3)$$

と定義すると、以下の定理が成り立つ。

定理1. (一般化KYP補題)

行列 $A \in \mathbf{C}^{n \times n}, B \in \mathbf{C}^{n \times m}, \Theta \in \mathbf{H}_{n+m}$ および $\Psi, \Phi \in \mathbf{H}_2$ が与えられたとする。このとき、式(3)によって $\bar{\Lambda}$ を定義し、 $\Omega$ をAの固有値のうち $\Lambda$ に含まれるものの集合とすると、次の二つの条件が同値となる。

- すべての $\lambda \in \bar{\Lambda} \setminus \Omega$ に対して

$$\begin{bmatrix} (\lambda I - A)^{-1} B \\ I \end{bmatrix}^* \Theta \begin{bmatrix} (\lambda I - A)^{-1} B \\ I \end{bmatrix} \leq 0 \quad (4)$$

が成り立つ。

- $Q \geq 0$ となるような $P, Q \in \mathbf{H}_n$ が存在し

$$\begin{bmatrix} A & B \\ I & 0 \end{bmatrix}^* (\Phi \otimes P + \Psi \otimes Q) \begin{bmatrix} A & B \\ I & 0 \end{bmatrix} + \Theta \leq 0 \quad (5)$$

が成り立つ。

式(5)は変数行列P, Qに対する線形行列不等式(LMI)となっている。また、システムの伝達関数が

$$G(\lambda) = C(\lambda I - A)^{-1} B + D \quad (6)$$

ただし,  $C \in \mathbf{C}^{p \times n}$ ,  $D \in \mathbf{C}^{p \times m}$ , で与えられるとする. このとき, 行列  $\Pi \in \mathbf{H}_{p+m}$  に対して  $\Theta$  を

$$\Theta = \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^* \Pi \begin{bmatrix} C & D \\ 0 & I \end{bmatrix} \quad (7)$$

と定義すると, 式(4)の条件は

$$\sigma(G(\lambda), \Pi) \leq 0 \quad (8)$$

と同値である. すなわち,  $G(\lambda)$  のベクトル軌跡が特定の周波数帯域  $\Lambda$  において, 複素平面上の式(8)によって表される領域に含まれているかどうかを式(5)のLMIを解くことによって知ることができる. ここで,

$$\Pi_{PR} = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}, \quad \Pi_{BR} = \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \quad (9)$$

とおくと, それぞれ正実性・有界実性の条件となる. したがって, システムのパラメータ  $(A, B, C, D)$  が与えられているときの解析問題(周波数を限定した場合の正実性や有界実性のチェック)はLMI可解問題となる. 特に, 有界実性において  $\gamma^2$  を最小化する問題はLMI最適化問題に帰着できる. しかし,  $(A, B, C, D)$  と  $\Pi$  の一部が設計パラメータを含んでいる設計問題を考える場合, 常にLMI最適化問題に帰着されるとは限らない. どのような条件のもとでLMI問題になるかについて次章で考察する.

一般化KYP補題の優れている点は, 単にLMIであることだけではなく, 統一的な表現になっている点である. すなわち, 周波数帯域ごとの各々の設計仕様は, 3つのエルミート行列  $\Phi, \Psi, \Pi$  を与えることと等価であるという点である. このことが4章におけるツールの開発に非常に役に立っている.

### 3. GKYP補題に基づく制御系設計

#### 3.1 GKYP補題による設計問題のLMI化

前章において, システム  $G(\lambda)$  が与えられた解析問題はLMIに帰着されることを示した. しかし, 指定した周波数帯域において  $G(\lambda)$  のベクトル軌跡が式(8)によって示される領域に含まれるような  $G(\lambda)$  を求めるという設計問題を考えるときは, 式(5)および式(7)がシステムのパラメータ  $(A, B, C, D)$  に対して一般には二次になる. このため, このままでは解くことはできないが, 文献10)では, 次のようにしてLMIに直すことができる」と述べられている.

$G(\lambda)$  が  $p \times m$  行列, すなわち  $m$  入力  $p$  出力のシステムであるとき, 式(7)における  $\Pi \in \mathbf{H}_{m+p}$  を,

$$\Pi = \begin{bmatrix} \Pi_{11} & \Pi_{12} \\ \Pi_{21} & \Pi_{22} \end{bmatrix}, \quad \Pi_{11} \in \mathbf{H}_p, \quad \Pi_{11} \geq 0 \quad (10)$$

とする. ここで新たに加えられた条件  $\Pi_{11} \geq 0$  は, 式(8)を満たす領域が凸であることを意味する. たとえば  $m=p=1$ , すなわち  $G(\lambda)$  がSISOであるときこの領域は,  $\Pi_{11} = 0$  のときは直線の片側,  $\Pi_{11} > 0$  のときは円の内側を意味し, どちらの領域も凸である. また, 有界実性や正実性などシステムの設計において必要な特性の多くは, この形で記述することができる.

$\Theta$  が式(7)によって定義され, さらに  $\Pi$  について式(10)の条件を満たしているとする. このとき, Schur complementを適用すると, 式(5)は次の式と同値であることが分かる.

$$\begin{bmatrix} \Gamma(P, Q, C, D) & [CD]^* \Pi_{11} \\ \Pi_{11} [CD] & -\Pi_{11} \end{bmatrix} \leq 0 \quad (11)$$

ここで,  $\Gamma(P, Q, C, D)$  は次の式で表される.

$$\Gamma(P, Q, C, D) := \begin{bmatrix} A & B \\ I & 0 \end{bmatrix}^* (\Phi \otimes P + \Psi \otimes Q) \begin{bmatrix} A & B \\ I & 0 \end{bmatrix} + \begin{bmatrix} 0 & C^* \Pi_{12} \\ \Pi_{12}^* C & D^* \Pi_{12} + \Pi_{12}^* D + \Pi_{22} \end{bmatrix}$$

この式(11)は, 式(5)に比べ, 行列のサイズが大きくなるため, システムのパラメータ  $(A, B, C, D)$  が固定されている解析問題では式(5)を解くほうが効率よく計算することができる. そのため, 与えられたシステムに関する解析問題を解く際には, 式(11)を考える必要はない. しかし, 式(11)は,  $C, D$  についても一次となるため, これらのパラメータをLMIの変数として解くことができる. 例えば,  $G(\lambda)$  を可制御正準形で表現したとすると,  $C, D$  は分子多項式の係数に対して一次になるので, 特定の周波数帯域において  $\sigma(G(\lambda), \Pi) \leq 0$  を満たすように固定した分母多項式をもつ伝達関数の分子多項式を設計することができる. これが, 設計に対する基本アイデアである.

• **Πの最適化**: 行列  $A, B, C, D$  が定数となる解析問題では, 式(5), (11)は  $\Pi$  に対して1次のLMIとなるので,  $\Pi$  を変数として計算することができる. 一方, 式(11)において  $C, D$  を変数として解く設計問題については,  $C, D$  の係数とならない  $\Pi_{22}$  のみLMIの変数として解くことができる.

例として, SISOシステム  $G(\lambda)$  を考える. 境界が直線となる条件,

$$a\Re G(\lambda) + b\Im G(\lambda) + c \leq 0, \quad a, b, c \in \mathbf{R}$$

に対して,  $\sigma(G(\lambda), \Pi) \leq 0$  となる  $\Pi$  は

$$\Pi = \frac{1}{2} \begin{bmatrix} 0 & a + jb \\ a - jb & 2c \end{bmatrix}$$

と書ける. よって解析問題では  $a, b, c$  のすべてを変数

とすることができ、設計問題では  $c$  のみを変数とすることができる。また、境界が円となる条件

$$|G(\lambda) - c| \leq r, \quad c \in \mathbf{C}, r \in \mathbf{R}$$

に対しては、 $\sigma(G(\lambda), \Pi) \leq 0$  となる  $\Pi$  は

$$\Pi = \begin{bmatrix} 1 & -c \\ -c^* & c^*c - r^2 \end{bmatrix}$$

と書ける。したがって、解析問題、設計問題ともに  $r^2$  を変数とすることができる。

### 3.2 取り扱えるシステムのクラス

図1で表されるコントローラ  $K$  によって制御される制御系を考える。このとき、 $w \rightarrow z$  の伝達関数は、

$$G(\lambda) = P_{11}(\lambda) + K(\lambda)P_{21}(\lambda) \quad (12)$$

と書け、 $K$  に関しアフィンとなる。

いま、 $P(\lambda)$  と  $K(\lambda)$  の状態空間実現をそれぞれ、

$$P : \begin{bmatrix} \lambda x_p \\ z \\ y \end{bmatrix} = \begin{bmatrix} A_p & B_{p1} & O \\ C_{p1} & D_{p11} & I \\ C_{p2} & D_{p12} & O \end{bmatrix} \begin{bmatrix} x_p \\ w \\ u \end{bmatrix}$$

$$K : \begin{bmatrix} \lambda x_c \\ u \\ \xi \end{bmatrix} = \begin{bmatrix} A_c & B_{c1} & O \\ C_{c1} & D_{c11} & D_{c12} \\ C_{c2} & D_{c12} & O \end{bmatrix} \begin{bmatrix} x_c \\ y \\ K\xi \end{bmatrix}$$

とする。ここで、 $K$  はコントローラの設計パラメータを表す対角行列である。これが、設計問題で取り扱えるシステムのクラスである。

例えば、 $P_{11}(\lambda) = 0$ ,  $G_{21}(\lambda) = P$  とすると、一巡伝達関数の周波数整形問題に適用することができる(詳しくは3.3を参照)。また、SISO系で  $H_\infty$  制御問題の準最適解を与えるパラメトリゼーションに基づく設計法にも適用できる。このとき、中心解を含めた閉ループ系を  $P$ 、フリーパラメータを  $K$  とし、 $K$  の分母を固定すればこの枠組みに入る。

### 3.3 一巡伝達関数の周波数整形

プラント  $P$  の状態空間実現を、

$$P = \begin{bmatrix} A_p & B_p \\ C_p & D_p \end{bmatrix}$$

とし、設計パラメータ  $K$  を持つコントローラ  $K$  の状態空間実現を、

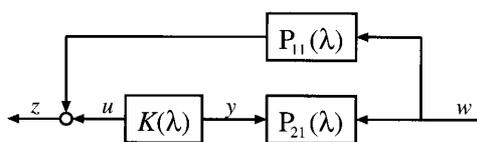


図1 取り扱えるシステム

$$K = \left[ \begin{array}{c|c} A_1 & B_1 \\ \hline C_1 & D_{11} \end{array} \right] + \left[ \begin{array}{c} B_2 \\ \hline D_{12} \end{array} \right] K [C_2 \mid D_{21}], \quad C_2 = 0 \quad (13)$$

によって定義する。ここで、 $C_2 = 0$  という条件は、コントローラの  $A, C$  行列は固定されていることを意味する。このとき一巡伝達関数  $\mathcal{L} = PK$  は、

$$\mathcal{L} = \left[ \begin{array}{cc|c} A_1 & 0 & B_1 + B_2KD_{21} \\ \hline B_pC_1 & A_p & B_p(D_1 + D_{12}KD_{21}) \\ \hline D_pC_1 & C_p & D_p(D_1 + D_{12}KD_{21}) \end{array} \right] \quad (14)$$

の状態空間実現を持つ。したがって、 $\hat{\mathcal{L}} = \mathcal{L}^T$  と定義し、この  $\hat{\mathcal{L}}$  に対して式(11)を適用することでパラメータ  $K$  に対して1次のLMIとなる。この方法によって、プラント  $P$  に対し一巡伝達関数  $L(\lambda)$  のベクトル軌跡が特定の周波数帯域で  $\sigma(L(\lambda)\Pi) \leq 0$  となる領域に存在するようなコントローラ  $K$  を設計することができる。例えば

$$K(s) = k_p + \frac{k_i}{s} + \frac{k_d}{1 + T_d s} \quad (15)$$

で表されるPIDコントローラ的设计を考えてみる。ここで、 $k_p, k_i, k_d$  はそれぞれ比例ゲイン、微分ゲイン、積分ゲインを表しており、 $T_d$  は伝達関数をプロパーにするための微小な固定係数である。このコントローラを可観測標準形で表すと、

$$K = \left[ \begin{array}{cc|c} 0 & 0 & \frac{k_i}{T_d} \\ \hline 1 & -\frac{1}{T_d} & k_i - \frac{k_d}{T_d^2} \\ \hline 0 & 1 & k_p + \frac{k_d}{T_d} \end{array} \right] \quad (16)$$

となり、式(13)と同じ形で表現できる。よって、 $\mathcal{L} = PK$  の設計を行うことができる。

・二次曲線への拡張: 文献10)では、伝達関数  $H(\lambda)$  に対する条件を表す境界線が2次曲線となるときに GKYP 補題を適用するには、

$$G(\lambda) = \begin{bmatrix} \tilde{H}(\lambda) \\ H(\lambda) \end{bmatrix}$$

とすればよいと述べられている。ここで、 $\tilde{H}$  は  $H$  の共役システム表している。これを一巡伝達関数の設計に用いることを考える。式(14)の  $\mathcal{L}$  が、

$$\mathcal{L} = \left[ \begin{array}{c|c} A_L & B_L(K) \\ \hline C_L & D_L(K) \end{array} \right]$$

となるように  $A_L, B_L(K), C_L, D_L(K)$  を定義する。 $\mathcal{L}$  の入力と出力の次元が等しいとき、 $\lambda \in \Lambda$  に対して  $H(\lambda), \tilde{H}(\lambda)$  を、

$$H(\lambda) = \hat{L}(\lambda), \quad \tilde{H}(\lambda) = L(\lambda)^*$$

と定義すると,  $G(\lambda)$ は連続時間システムの場合

$$G^T = \begin{bmatrix} -A_L^T & 0 & C_L^T \\ 0 & A_L^T & C_L^T \\ -B_L(K)^T & 0 & D_L(K)^T \\ 0 & B_L(K)^T & D_L(K)^T \end{bmatrix} \quad (17)$$

となる. また, 離散時間システムの場合は,

$$G^T = \begin{bmatrix} A_L^{-T} & 0 & -A_L^{-T} C_L^T \\ 0 & A_L^T & C_L^T \\ B_L(K)^T A_L^{-T} & 0 & D_L(K)^T - B_L(K)^T A_L^{-T} C_L^T \\ 0 & B_L(K)^T & D_L(K)^T \end{bmatrix} \quad (18)$$

となる. したがって, 連続時間・離散時間どちらの場合も式(11)によって  $K$  に関するLMIとなり,  $K$  の設計が可能になる.

### 3.4 分母多項式を含めた設計

前節において述べた設計法は, 式(11)の  $C, D$  行列をLMIの変数として解く方法をもとにしている. これは式(11)が  $A, B$  に対してはLMIでないため, 変数とすることができないからである. 特に  $A$  を変数にできないということは, 伝達関数の分母を設計することができないということを意味する. 本節では, この問題を解決するための一手法を紹介する.

プラント  $P(\lambda)$  およびコントローラ  $K(\lambda)$  はSISOであるとする.  $K(\lambda)$  を多項式  $n(\lambda), d(\lambda)$  によって,  $K(\lambda) = n(\lambda)/d(\lambda)$  と定義し,  $L(\lambda) = P(\lambda)K(\lambda)$  に対して,

$$\left| P(\lambda) \frac{n(\lambda)}{d(\lambda)} - c \right| \leq \gamma, \quad \lambda \in \Lambda \quad (19)$$

という条件を考える. ここで,  $c \in \mathbf{C}, \gamma > 0$  とする.

**命題1.** 多項式  $\hat{d}(\lambda)$  および  $0 < \epsilon < 1$  が与えられたとする.  $\forall \lambda \in \Lambda$  に対して次の2つの式がともに成り立つことは, 式(19)が成り立つための十分条件である.

$$\left| P(\lambda) \frac{n(\lambda)}{\hat{d}(\lambda)} - \frac{d(\lambda)c}{\hat{d}(\lambda)} \right| \leq (1-\epsilon)\gamma \quad (20)$$

$$\left| 1 - \frac{d(\lambda)}{\hat{d}(\lambda)} \right| \leq \epsilon \quad (21)$$

**証明.** 三角不等式より,

$$1 - \left| \frac{d(\lambda)}{\hat{d}(\lambda)} \right| \leq \left| 1 - \frac{d(\lambda)}{\hat{d}(\lambda)} \right|$$

が成立する. この式と式(21)より,

$$1 - \epsilon \leq \left| \frac{d(\lambda)}{\hat{d}(\lambda)} \right|$$

を得る. さらに, この式と式(20)より,

$$\left| P(\lambda) \frac{n(\lambda)}{\hat{d}(\lambda)} - \frac{d(\lambda)c}{\hat{d}(\lambda)} \right| \leq \left| \frac{d(\lambda)}{\hat{d}(\lambda)} \right| \gamma$$

が成り立つ. 式(19)はこの式の両辺に  $\left| \frac{\hat{d}(\lambda)}{d(\lambda)} \right|$  を掛けたものである. 式(20), (21)の両方が成り立てば式(19)を満たすことがわかる.  $\square$

式(20), (21)は, 多項式  $\hat{d}(\lambda)$  を固定すれば変数は分子にしか存在しないため, 式(11)の形に直すことで解くことができる.

式(20)は式(19)の半径を少し小さくして, 中心を少し移動させたものである. また, 式(21)は分母多項式  $d(\lambda)$  を  $\hat{d}(\lambda)$  の近傍で動かすことができるという意味である. すなわちこの命題は, 式(19)の条件を少しだけ厳しくしたために, 分母多項式に狭い範囲ではあるが設計できる余地が生まれたということを意味している. この命題を利用した繰り返し計算アルゴリズムが, 文献11)において提案されている.

## 4. 制御系設計ツール

### 4.1 概要

前節までにおいて, 与えられた制御対象に対して周波数ごとに要求される複数の仕様を満たすコントローラをLMI最適化問題として数値的に解くことができることを示した. しかし, 実際に Matlab の LMI Control ToolboxによってこのLMI条件を記述するには, 1つの仕様を記述するのに50行を越えるプログラムが必要である. さらに, ある制御対象に対して周波数帯域ごとに異なる性能が要求されることが一般的であるため, GKYP補題に基づく制御問題を解くためには, LMIを定義するだけで数百行のプログラムを記述しなければならない.

この問題を解決するために, Matlab の LMI Control Toolboxを利用してGKYP制御系解析・設計ツール(LMI自動生成プログラム)を作成した. このツールを利用することで, 1つの設計仕様をわずか4行で直接的に記述することができ, 現場の制御技術者がGKYP理論を習得する必要なくGKYPに基づく制御問題を容易に解くことが可能になる. 実際, 必要となるプログラムステップ数はLMIツールボックスで直接記述する場合の10分の1以下であり, LMIツールボックスの文法を学ぶ必要もない.

このツールの構造は, 図2のようになる. 以下において, それぞれのブロックについて, 簡単に説明する.

• **GKYP Tool**: GKYP Toolは, 対象のシステムと周波数帯域ごとに要求される仕様を入力とし, GKYP補題の式(5), (11)を解くために必要な  $\Phi, \Psi, \Pi$  を計算して

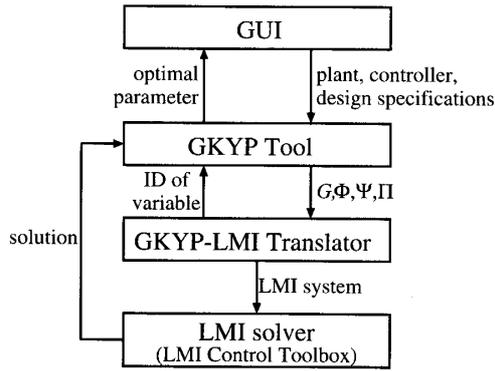


図2 GKYP制御解析・設計ツール

GKYP-LMI Translatorに出力する。さらに、LMIを解いて得られた最適解と、GKYP-LMI Translatorから返される変数の識別子をもとに、コントローラのパラメータと最適化された性能を構成して設計者に返す。

• **GKYP-LMI Translator**: GKYP-LMI Translatorは、GKYP Toolから渡されるシステムと $\Phi, \Psi, \Pi$ から、式(5), (11)に基づいてLMIを生成し、LMI Control ToolboxのLMI systemに変換する。また、計算結果から周波数帯域ごとの性能を構成するために必要な情報をGKYP Toolに返す。

• **LMI solver**: LMI solverはGKYP-LMI Translatorによって構築されたLMIを解き、最適解をGKYP Toolに返す。このブロックは、MatlabのLMI Control Toolboxの機能である。

• **GUI**: GUIは、GKYP Toolに必要な入力を自動的に生成し、計算結果を視覚的に表示する。

GKYP Toolの入出力が構造化されていれば、前述のGUIとGKYP Toolを分離して開発を行なうことができる。また、GUIを利用せず直接システムと条件をプログラムに記述してGKYP Toolを利用することを考えると、GKYP Toolの入出力は設計者の表現に近いほうが望ましい。これらの要求をできるだけ満たすようにGKYP制御系解析・設計ツールを開発した。以下の節では、図2の各ブロック間でやりとりされるデータの構造とGKYP ToolおよびGKYP-LMI Translator内部でのデータ変換アルゴリズムを説明する。

#### 4.2 入出力のデータ構造

本節では、前節の図2においてGKYP ToolとGUIをつなぐデータの構造について説明する。GUIを使わずにこのツールを使うときは、この構造体によってシステムや条件の定義と結果の出力を行なう。

##### 4.2.1 システムの定義

GKYP補題の枠組みにおいて扱うことのできるシス

表3 プラントと一巡伝達関数を表す構造体

変数	内容	変数	内容
A	行列 A	B2	行列 B <sub>2</sub>
B1	行列 B <sub>1</sub>	C2	行列 C <sub>2</sub>
C1	行列 C <sub>1</sub>	D12	行列 D <sub>12</sub>
D11	行列 D <sub>11</sub>	D22	行列 D <sub>22</sub> (常に0)
D21	行列 D <sub>21</sub>	nx	システムの次数
nx	システムの次数	nk	パラメータの数
lim	パラメータの範囲	ts	単位時間

テムについては、3.2節において議論されている。ここで、プラントとコントローラが接続されたシステム全体の状態空間実現が次のように表されているとする。

$$\begin{bmatrix} \lambda x \\ z \\ \xi \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x_c \\ w \\ K\xi \end{bmatrix}$$

この行列の各要素に加え、コントローラの次数、パラメータの自由度、各パラメータのとりうる値を一つの構造体にまとめたものをコントローラとして定義する。ただし、GKYP補題を解く上で連続時間システムと離散時間システムを区別する必要があるため、単位時間を構造体に加える。単位時間は離散時間システムにおける刻み時間幅を表し、連続時間システムでは0となる。

これらをまとめたものを表3に示す。システムの定義を容易にするために、頻繁に使われる可能性のあるコントローラを自動的に生成する関数K\_lti, K\_pid, K\_num, K\_diffを用意した。K\_ltiは、MatlabのLTIシステムからコントローラを生成し、K\_pid, K\_numはそれぞれ、PIDコントローラ、分母多項式が固定された伝達関数を持つコントローラを生成する。また、K\_diffは2つのコントローラ $K_1(\lambda)$ と $K_2(\lambda)$ の差分 $K_1(\lambda) - K_2(\lambda)$ を生成し、コントローラの低次元化に用いることができる。さらに、関数make\_LTFによって、これらの関数によって生成されたコントローラとプラントから容易に一巡伝達関数を生成することができる。

##### 4.2.2 条件の定義

帯域ごとの条件は構造体配列によって定義する。この構造体にはomega\_l, omega\_h, type, paramという4つのフィールドがある。それぞれのフィールドについて以下で説明する。

• **周波数帯域**: omega\_lは周波数の下限, omega\_hは周波数の上限を表し, omega\_h = 0としたときは上限が有界でないことを意味する(表4を参照)。例えば, m番目の仕様に $1 < \omega < 10$ という帯域を設定するときには,

表4 周波数帯域の設定

帯域	$ \omega  < \omega_1$	$\omega_2 < \omega < \omega_3$	$ \omega  > \omega_4$
omega_l	0	$\omega_2$	$\omega_4$
omega_h	$\omega_1$	$\omega_3$	0

表5 条件の設定

境界	直線	円	楕円
type	'hp'	'oval'	'oval'
paramの順序	$a, b, c$	$c, r$	$c, r, a, \theta$
最適化可能(解析)	$a, b, c$	$r$	$r$
最適化可能(設計)	$c$	$r$	$r$

```
specs(m).omega_l = 1 ;
specs(m).omega_h = 10 ;
```

とすればよい。ただし、低周波域と高周波域では、以下の2つの注意が必要である。

1) 実システムするとき  $\omega$  と  $-\omega$  に対応するベクトル軌跡は実軸対称になるので、その帯域に与えられた条件が実軸に対称であれば、 $0 < \omega < \omega_1$  あるいは  $\omega > \omega_4$  の代わりに  $|\omega| < \omega_1$  あるいは  $|\omega| > \omega_4$  という帯域が使われる。このとき、GKYP補題の式(5),(11)の  $\Pi$  と  $\Phi$  が実行列になることから、このLMIは実LMIとなるため計算を短時間でこなうことができるからである。この帯域処理は、GKYP Translatorが自動的にこなされている。

2) コントローラに積分器が含まれるなどの理由で、 $\lim_{s \rightarrow 0} |L(s)| = \infty$  となる場合には  $0 < \omega < \omega_l$  という条件では正しく計算ができない。そのため十分小さい  $\omega_\epsilon$  を用いて、 $\omega_\epsilon < \omega < \omega_l$  という条件に直した上で、中間帯域の条件として解く必要がある。

• **帯域ごとの条件** : type, param は、それぞれの帯域における仕様を具体的に表す(表5を参照)。param は、それぞれの仕様におけるパラメータを2行2~4列の行列で設定する。この行列の1行目はそれぞれのパラメータの定数部分に、2行目はパラメータを最適化するときの目的関数の係数に対応している(係数が正であれば最小化、負であれば最大化される)。これらのパラメータは、楕円の中心のみ複素数で、それ以外は実数となる。また、これらのパラメータは、そのすべてが最適化できるわけではない。最適化できないパラメータについては、対応する2行目の要素を0にしなくてはならない。どのようなパラメータが最適化できるのかについては、楕円の内側条件に対し、具体的に説明する。

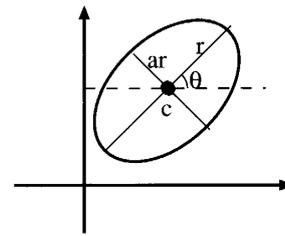


図3 複素平面上的楕円

type = 'oval' のとき、複素平面上的楕円が図3のように表されているとすると、paramの1列目は楕円の中心  $c$ 、2列目は一方の軸の半径  $r$ 、3列目は2軸の長さの比  $a$ 、4列目は軸と実軸とのなす角  $\theta$  に対応している。また、3列目、4列目は省略することができ、その場合はそれぞれ  $a=1$  (真円)、 $\theta=0$  (軸は実軸・虚軸にそれぞれ平行) になる。また、最適化できるパラメータは、解析・設計問題ともに半径  $r$  のみである。ただし、その場合においても  $r$  のかわりに  $r^2$  が最適化の変数として用いられる。たとえば、図3のような楕円の内側を表すには、

```
specs(m).type = 'oval' ;
specs(m).param = [ c, r, a, theta ;
                  0, 0, 0, 0 ] ;
```

とすればよく、この条件において  $r$  を最小化させるときは、

```
specs(m).type = 'oval' ;
specs(m).param = [ c, 0, a, theta ;
                  0, 1, 0, 0 ] ;
```

とすればよい。

• **複数パラメータの線形和の最適化** : 最適化させるパラメータが複数あるとき、その線形和を最適化させることができる。このようなときは、paramの2行目に目的関数の係数を与えればよい。例えば、

$$\begin{aligned} \min \quad & -x_1 + 2x_2 \\ \text{s.t.} \quad & -\Re[L(\lambda)] + \Im[L(\lambda)] + x_1 < 0 \\ & |L(\lambda)|^2 < x_2 \end{aligned}$$

のような問題にを解くときは、1つ目の条件に対して

```
specs(1).type = 'hp' ;
specs(1).param = [ -1, 1, 0 ;
                  0, 0, -1 ] ;
```

とした上で、2つ目の条件に対して

```
specs(1).type = 'oval' ;
specs(1).param = [ 0, 0 ;
                  0, 2 ] ;
```

とすれば、この最適化問題を解くことができる。

表6 結果を表す構造体

変数	内容
feasible	解があれば1, なければ0
optval	目的関数の最適値
k	コントローラのパラメータ
sysL	一巡伝達関数の状態空間モデル
specs	最適化された条件を表す構造体

#### 4.2.3 結果の出力

結果resultはfeasible, optval, k, sysL, specsという5つのフィールドを持つ構造体である(表6を参照)。

### 4.3 LMI自動生成アルゴリズム

#### 4.3.1 GKYP Toolのアルゴリズム

GKYP Toolの役割は2つある。1つは与えられたシステムと条件をGKYP補題の行列(A, B, C, D, Φ, Ψ, Π)に変換してGKYP Translatorに送ることと, GKYP Translatorによって作られたLMIの解から最適化されたシステムと条件を再構築して結果を出力することである。それぞれのブロックについて, 以下で説明する。

- (1) **解析・設計の判別**: まず, 与えられたシステムにパラメータが存在するかを判別する。パラメータが存在しなければ解析問題, 存在すれば設計問題となる。
- (2) **パラメータの定義**: つぎに, システムにパラメータが存在するときは, そのパラメータに対応するLMI変数を定義する。この変数は, すべての条件に対して共通に使われるため, GKYP Translatorによって条件をLMIに直す前に定義する必要がある。また, パラメータのとりうる範囲に制約があるときは, ここで不等式としてLMIに最初に定義される。
- (3) **条件の分解・Ψ, Πの計算**: 条件として入力された構造体に対して, Ψ, Πを計算する。ただし, Πには最適化すべき変数を含んでいることがある。そのためΠは $x_k$ を変数として

$$\Pi = \Pi_0 + \sum_{k=1}^m x_k \Pi_k$$

と表現したときの $\Pi_0, \dots, \Pi_m$ と変数の数 $m$ を全て計算して構造化している。

- (4) **LMIの定義**: (3)で計算して構造化したΨ, Πを, システムの構造体とコントローラのパラメータを表す識別子とともにGKYP Translatorに送り, その条件に対応するLMIを構成する。このとき,

GKYP Translatorからは変数を表す識別子が返される。この識別子には, 実際に作られたLMIのどの変数が条件として最適化すべき変数なのかを表している。

- (5) **LMIを解く**: すべての条件をGKYP TranslatorでLMIに直したのち, GKYP Translatorの返り値と条件をもとに目的関数の係数を計算してLMI ToolboxのLMI solverによって解く。
- (6) **LMI結果の再構成**: LMIを解いた結果から最適なシステムと条件を構成して, 計算結果として出力する。

#### 4.3.2 GKYP Translatorアルゴリズム

GKYP Translatorは, 与えられたシステムとΨ, Πを式(5), (11)のLMIとして定義する。このとき, 解析問題では行列の次数が少ない式(5)が, 設計問題ではC, Dに対して1次になっている(11)が使われる。

(1) まず, 行列Πとシステムの次数から二次曲線の拡張が必要かを判別して, 必要なときは式(17), (18)によってシステムを拡張する。2次曲線に拡張する必要があるときは $\Pi_{11}$ の次数とシステムの出力の次数が一致しないことから判別できる。

(2) つぎに, ΠとΨからLMIが実か複素かを判別する。そして, 実LMIであれば式(5), (11)を表すLMIを作り, 複素LMIであれば $A \leq 0$ というLMIを

$$\begin{bmatrix} \Re(A) & \Im(A) \\ -\Im(A) & \Re(A) \end{bmatrix} \leq 0$$

とすることで実LMIに直す。

(3) 最後に, この過程において定義されたΠの変数部分に対応する識別子をLMI Toolsに返す。

### 5. 設計ツールを用いた制御系設計例

本章では, GKYP制御解析・設計ツールを利用したプログラムとして, 一巡伝達関数の周波数整形をあげる。この例題は, プラント $P(s)$ が

$$P(s) = \frac{9}{(s+1)(s^2+s+9)}$$

であるとき, 一巡伝達関数 $L(s) = P(s)K(s)$ が次の条件を満たしつつ,  $\gamma$ を最小とするPIDコントローラを見つけるものである。

$$\begin{aligned} |L(s)| &< 0.1 & |\omega| &> 5 \\ -3\Re[L(s)] + \Im[L(s)] &< \gamma & \omega &> 0.01 \\ \Im[L(s)] + 2 &< 0 & 0.01 &< \omega < 0.5 \\ \left\{ \frac{\Re(L_1(j\omega))}{0.6} \right\}^2 + \left\{ \frac{\Im(L_1(j\omega))}{0.3} \right\}^2 &< 1 & 0.8 &< \omega < 1 \end{aligned}$$

ここで,  $L_1(s)$ は次のように定義される。

```

% --- define plant and controller ---
sysP = tf(9,[1,2,11,9]);
sysK = K_pid([1,1,1],0.01,[10,10,10; 0,0,0]);
sysL = make_LTF(sysP,sysK);

% --- set specs ---
specs(1).omega_l = 5;
specs(1).omega_h = 0;
specs(1).type = 'oval';
specs(1).param = [0,0.1; 0,0];

specs(2).omega_l = 0.01;
specs(2).omega_h = 0;
specs(2).type = 'hp';
specs(2).param = [-3,1,0; 0,0,-1];

specs(3).omega_l = 0.01;
specs(3).omega_h = 0.5;
specs(3).type = 'hp';
specs(3).param = [0,1,2; 0,0,0];

specs(4).omega_l = 0.8;
specs(4).omega_h = 1;
specs(4).type = 'oval';
specs(4).param = [exp(i*pi*4/3),0.3,2,pi/3;
                  0, 0, 0, 0 ];

% --- solve LMI baed on GKYP ---
result = gkyp(sysL, specs);

% --- display results ---
kp = result.k(1)
ki = result.k(2)
kd = result.k(3)
nyq_specs(result);

```

図4 周波数整形プログラム

$$L_1(s) = e^{\frac{\pi}{6}j} L(s) + j$$

また、PIDコントローラは式(15)で定義されるものとし、 $T_d = 0.01$ とする。さらに、PIDのパラメータはともに  $0 < k_p, k_i, k_d < 10$  を満たすように条件を加えておく。この問題は、図4のプログラムにより解くことができる。

このプログラムについて、順に説明する。まず、プラントを 'tf' コマンドによって定義し、次にコントローラを定義する。この 'K\_pid' コマンドは、 $T_d = 0.01$  で、各パラメータの上限が10、下限が0となるようなPIDコントローラであることを意味している。さらに、プラント、コントローラを接続したループ伝達関数  $L(s) = P(s)K(s)$  に対応したモデルを 'make\_LTF' で定義する。

つぎに、条件を構造体配列で定義する。例題における4つの条件がそれぞれ specs(1), specs(2), specs(3), specs(4) に対応している。これらを定義した後に、'gkyp(sysL, specs)' によって、GKYP Toolを実行すれば定義したシステム、条件に対応したLMIを解いた上で、結果 result に返される。

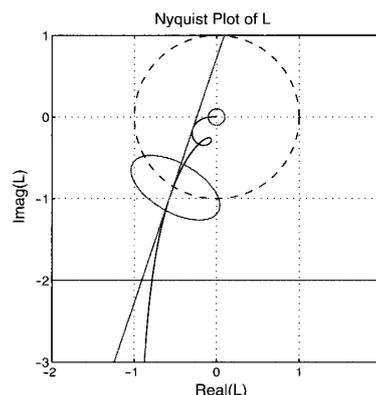


図5 例題の実行結果

最後に、構造体 result からPIDパラメータを取り出した後、'nyq\_specs(result)' によって、一巡伝達関数のベクトル軌跡と設定した条件を描画する。

このプログラムを実行すると、図5のような図が表示される。この結果、設計されたPIDパラメータは、

$$k_p = 0.3406, \quad k_i = 1.2127, \quad k_d = 0.2217$$

また、 $\gamma$ の最小値は、0.7201となった。

## 6. おわりに

本論文では、GKYP補題に基づく制御系設計法の確立を目指し、特にそれに基づいて効率よく制御系の解析・設計を行なうためにMatlab上で開発した解析・設計ツールについて説明した。開発したツールを利用すると、現場の制御技術者は単に各々の設計仕様を直接的に記述するだけで十分である。また、このツールは完全に構造化されているので、対応するGUIの作成により、全くプログラミングを必要とせず、多目的制御系設計が可能となる。ただし、LMI最適化問題に帰着される制御問題は制限されており、3.4節において説明した分母多項式を含めた最適設計やロバストGKYP補題<sup>1)</sup>に基づく設計への対応が今後の課題である。

なお、本研究の一部は、(独)科学技術振興機構CRESTの支援を受けて行った。また、本研究を進めるに当り貴重なご意見をいただいた岩崎徹也教授(バージニア大学)に感謝いたします。

## 参考文献

- 1) SLICOT: <http://www.win.tue.nl/niconet/NIC2/slicot.html>
- 2) 古賀雅伸: 制御・数値解析のためのMaTX, 東京電機大学出版局(2000)
- 3) S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan: *Lin-*

- ear Matrix Inequalities in System and Control Theory*, SIAM Studies in Applied Mathematics (1994)
- 4) 岩崎徹也: LMI と制御, 昭晃堂 (1997)
  - 5) B. D. O. Anderson: A system theory criterion for positive real matrices, *SIAM J. Contr.*, **5**, 171/182 (1967)
  - 6) A. Rantzer: On the Kalman-Yakubovich-Popov lemma, *Sys. Contr. Lett.*, **28-1**, 7/10 (1996)
  - 7) T. Iwasaki, G. Meinsma, and M. Fu: Generalized S-procedure and finite frequency KYP lemma, *Mathematical Problems in Engineering*, **6**, 305/320 (2000)
  - 8) T. Iwasaki, S. Hara, and H. Yamauchi: Dynamical system design from a control perspective: Finite frequency positive-realness approach, *IEEE Trans. Auto. Contr.*, **48-8**, 1337/1354 (2003)
  - 9) T. Iwasaki and S. Hara: Generalization of Kalman-Yakubovich-Popov lemma for restricted frequency inequalities, *Proc. American Contr. Conf.* (2003)
  - 10) T. Iwasaki and S. Hara: Generalized KYP Lemma: Unified Frequency Domain Inequalities with Design Applications, *IEEE Trans. Auto. Contr.*, **50-1**, (2005)
  - 11) S. Hara, D. Shiokata, and T. Iwasaki: Fixed Order Controller Design via Generalized KYP Lemma, Proc. of 2004 IEEE CCA/ISIC/CACSD (2004)
-