多角形区間演算の改良と不確かさを含む 伝達関数の値集合の推定

非会員 磯 貝 晃 (神戸大)

正 員 太 田 有 三 (神戸大)

正 員 羽 根 田 博 正 (神戸大)

Improvement of Polygon Interval Arithmetic and Evaluation of Value Sets of Transfer Functions of Uncertain Systems

Akira Isogai, Non-member, Yuzo Ohta, Member, Hiromasa Haneda, Member (Kobe University)

The purpose of this paper is to propose an improved version of inversion operation of Polygon Interval Arithmetic (PIA) and demonstrate the usefulness of it to evaluate value sets of control systems containing parameters whose values exist in given intervals but their exact values are not known. By using PIA, a polygon containing a value set as a subset is computed very fast. To get a good evaluation of the value set, it is required to express a given transfer function in a class of special forms. Two examples, which can be modified so that they belong to such a class, are presented. Since these expressions require to execute the inversion operation of PIA many times, to reduce the enlargement of the uncertainty caused by that, it is necessary to improve the definition of the inversion operation of PIA. Moreover, to implement the new inversion operation, an algorithm to decompose a non-convex polygons into convex polygons is proposed.

キーワード：多角形区間演算，伝達関数の値集合

1. はじめに

不確かさを含む制御系において、不確かさのパラメータを含む伝達関数の値集合を算出することは、ナイキストの方法による安定判別、古典理論による解析・設計の際の指標となるバンド幅などの算定、更に H^∞ 理論に必要な加法(乗法)性摂動の算出などに有用である。従って、この値集合をできるだけ正確に求めることは、系の解析・設計を行ううえで非常に重要である。多角形区間演算(PIA)は、伝統的な区間演算を複素平面における多角形の集合に拡張したもので、不確かさを効率良く取り扱うための有効な手段である⁽¹⁾。本論文では、PIAを用いて不確かさを含む系の伝達関数の値集合を高速に、かつなるべく精確に推定する方法について述べる。

不確かさをパラメータとして含む伝達関数 $f(s, x_1, \dots, x_N) \equiv f(s, \mathbf{x})$, $\mathbf{x} = [x_1, \dots, x_N]^T$ と各パラメータの存在領域 $X_1, \dots, X_N$ が与えられたとき、値集合とは、複素平面上の領域

$$\left. \begin{aligned} f(s, \mathbf{X}) &\equiv f(s, X_1, X_2, \dots, X_N) \\ &\equiv \{f(s, \mathbf{x}) | x_i \in X_i, i=1, \dots, N\} \\ \mathbf{X} &= X_1 \times X_2 \times \dots \times X_N \end{aligned} \right\} \dots\dots\dots (1)$$

である。ここで、各パラメータの存在領域の直積で定義される $\mathbf{X}$ をパラメータ空間と呼ぶ。この値集合は、系の不確かさを表すと解釈することができる。PIAにおいては、各パラメータの存在領域をはじめ、すべての不確かさを複素平面上の多角形として取り扱う。よって、PIAを用いた値集合の推定とは、この値集

合の多角形による近似を求めることを意味する。

(1)式におけるパラメータの存在領域 X_i は実数直線上の領域(区間)を仮定すれば十分であるが、そのような領域は複素平面上の領域の特殊な場合であると考え、以下の議論ではパラメータの存在領域は複素平面上のものを前提とする。このような前提の下では、伝達関数の複素数 s が領域として与えられる場合にも対応することができる。

いうまでもなく、算出される値集合の推定は精密であるほど望ましい。しかし、PIA を用いると次の二つの理由から得られる結果が値集合よりも拡大してしまう傾向がある。

まず、伝達関数において同一パラメータが複数箇所に出現するような場合が存在することが、その一つである。これには、伝達関数の分母あるいは分子内で複数箇所に出現する場合と、分母、分子に同時に出現する場合とがある。実際の制御系においては、分母、あるいは分子の多項式はパラメータに関して多重線形となる場合が多いことが知られている⁽²⁾が、そのようなときには、前者に対しては端点集合の分割による方法⁽³⁾を用いることにより対応できるので、あまり深刻な問題とはならない。しかし、後者に関してはそのような方法を用いることはできず、そのまま計算したのでは本来同一であるはずのパラメータが別の独立したパラメータであるとして計算されてしまうために、演算結果は不確かさを大きく拡大したものになってしまう。

もう一つの理由は、PIA の演算自体が不確かさを拡大する性質をもつことである。PIA においては、和、積および逆集合の三つの演算が定義されているが、それらの演算のうち、積と逆集合の演算は不確かさを拡大してしまう可能性をもっている。積の演算による拡大は複雑な構造の下で生じており、その詳細についてはまだわかっておらず、本論文ではこの点に関する議論は差し控える。それに対して、逆集合の演算により生じる拡大は演算の定義に起因するものであり、これは定義を改良することにより防ぐことができる。

以下では、第1の原因に対してその対応策を述べ、さらにその対応策が従来よりも精確な逆集合の演算を必要とすることを示す。そして、第2の原因のうち、逆集合の演算による不確かさの拡大を防ぐため、従来の逆集合の演算を改良し、定義する。また、それらが算出する値集合の推定の改善に対して有効であることを示す。

なお、本論文では、複素平面を C で表し、 C 上の

すべての多角形からなるクラスおよびすべての凸多角形からなるクラスをそれぞれ P, P_{conv} で表す。また、 $\{x_1, \dots, x_m\}$ を C 上の点の集合、 $\text{conv}\{x_1, \dots, x_m\}$ をそれらの凸包とすると、凸多角形 X は $X = \text{conv}\{x_1, \dots, x_m\}$ で表される。凸多角形 X の端点は $\text{ext}(X)$ で表される。 C 上の点の集合 $\{p_1, \dots, p_n\}$ に対して、 $\text{poly}\{p_1, \dots, p_n\}$ は点 $p_1, p_2, \dots, p_n, p_1$ を順に結んだときにできる多角形を表す。また、集合 Z の要素数を $|Z|$ で表す。 $x, y \in C$ に対して、線分 $[x, y]$ は $[x, y] = \text{conv}\{x, y\}$ で定義される。そして、 $\text{aff}(\{x, y\})$ は線分 $[x, y]$ を含むアフィン包(ここでは直線)を表す。また、 $N(X, \varepsilon)$ は集合 X の ε -近傍を表す。更に、 $i < j$ であるようなある正の整数 i, j が与えられたとき、 $\{i..j\}$ は集合 $\{i, i+1, \dots, j\}$ を表す。

2. PIA による伝達関数の値集合の算出

ここでは、多角形区間演算(PIA)とその性質について簡単にまとめ、さきほど述べた PIA による不確かさの拡大の第1の原因への対応策について考察する。

PIA は伝統的な区間演算を平素平面上のすべての多角形からなるクラス P に対して適用できるように一般化したものである。ここで、 P の要素は必ずしも凸ではないことに注意する必要がある。このとき、このクラス P に対する和、積、逆集合を表す演算子をそれぞれ $+$, $\bullet$, $(\cdot)^{-1}$ であるとし、 P の要素に対してこれらの演算を施したものが再び P の要素となるように代数系 $\text{PIA}(P; +, \bullet, (\cdot)^{-1})$ は定義されている。PIA に関して、付録に示す[補題付2, 付3]の結果が成り立つことが示されている⁽⁴⁾。

この PIA を伝達関数の値集合の推定に用いるために、まず多角形の有理関数を定義する。 $f(x_1, \dots, x_n)$ を n 個の複素変数の有理関数とする。ただし、このときこの関数において和、積および商の演算を行う順序は定まっていると仮定する。ここでこのような仮定を設けるのは、同じ関数であっても演算の順序を変えることにより、得られる結果が異なるものになる可能性があるからである。この点については後に詳しく述べる。有理関数 $f(x_1, \dots, x_n)$ において、変数 x_i を多角形(P の要素) X_i に置き換え、通常の和、積および商の演算を P における和、積および逆集合の演算に置き換えたものを多角形の有理関数 $F_f(X_1, \dots, X_n)$ と定義する。なお、以下では多角形の有理関数 $F_f(X_1, \dots, X_n)$ に対して PIA を適用した結果得られる多角形(値集合の推定)を $F_f(X_1, \dots, X_n)$ で表現するが、さきほどの演算の順序の一意性より $X_1, \dots, X_n$ さえ与えられれば演算結果の多角形は一意に決まるの

で、このような表現が混乱をまねくことはないであろう。第1章でも述べたが、[補題付2, 付3]に示されるような性質と、分母、分子における同一パラメータの存在性により、値集合 $f(X_1, \dots, X_n)$ と PIA により求められるその推定 $F_I(X_1, \dots, X_n)$ との間には、

$$f(X_1, \dots, X_n) \equiv \{f(x_1, \dots, x_n) | x_i \in X_i, i \in [1..n]\} \subseteq F_I(X_1, \dots, X_n) \dots\dots\dots (2)$$

という関係が成り立つ。すなわち、PIA による推定は値集合よりも拡大したものである可能性があるわけである。この左辺と右辺の差をできるだけ小さくするための手法を以下に述べる。

さきほど演算の順序を変えることにより得られる結果が変わる可能性があることと述べたが、これはすなわち演算の順序を工夫することにより、より精密な推定を算出し得る可能性があるということである。そこで、演算の順序をどのように工夫すればよいかを次に考える。第1章で述べたように、代数系 PIA が不確かさを拡大するのは主に次のような理由による。

(i) (付15)式に示されるように分配律が成り立たないこと。つまり、同一パラメータはくり出してから計算しなければ演算結果が拡大してしまう可能性がある。

(ii) 分母と分子における互いに独立でないパラメータの存在に起因するもの。この場合にも、別の独立したパラメータであると計算されてしまうために、不確かさは拡大してしまう。

従って、PIA を用いて伝達関数の値集合をより正確に推定するためには、そのような状況を回避するような演算順序となるように関数を変形することが有効であると考えられる。その一つの方法として、Barmish らにより提案された TSD (Tree Structured Decomposition)⁽⁴⁾ や、それを改良した文献(5)の方法などがある。次に示す例は、文献(5)の方法が有効な例である。

(例1) 次のように関数を考える。

$$\left. \begin{aligned} f_1(x) &= \frac{x_1 x_2 + x_1 x_3}{x_2 + x_3 + x_4} \\ x &= [x_1, x_2, x_3, x_4]^T \end{aligned} \right\} \dots\dots\dots (3)$$

この式は

$$\left. \begin{aligned} f_2(x) &= \frac{x_1}{1 + \{x_4 / (x_2 + x_3)\}} \\ x &= [x_1, x_2, x_3, x_4]^T \end{aligned} \right\} \dots\dots\dots (4)$$

と変形できる。これらはもちろん同じ関数であるが、 $f_1(x)$, $f_2(x)$ にそれぞれ PIA を適用すると、さきほど

述べたような理由により結果は異なるものとなる。すなわち、同じパラメータ空間 $X = X_1 \times \dots \times X_4$ に対して値集合を推定すると、 $F_{I2}(X)$ は $F_{I1}(X)$ に比べてより精密なものになると考えられる。

実際に(3), (4)式において、 $X_1, \dots, X_4$ に領域を割り当て、値集合を推定する。C 上の点 $p_1, p_2, \dots, p_8$ を

$$\begin{aligned} p_1 &= 1+j, & p_2 &= 2+j, & p_3 &= 3+j, \\ p_4 &= 1+2j, & p_5 &= 2+2j, & p_6 &= 3+2j, \\ p_7 &= 1+3j, & p_8 &= 2+3j, & p_8 &= 3+3j \end{aligned}$$

とし、 $X_1, \dots, X_4$ を

$$\begin{aligned} X_1 &= \text{conv}\{p_1, p_2, p_5, p_4\} \\ X_2 &= \text{conv}\{p_3, p_5\} \\ X_3 &= \text{conv}\{p_4, p_5, p_7\} \\ X_4 &= \text{conv}\{p_6, p_8, p_8\} \end{aligned}$$

として(3), (4)式の値集合の推定 $F_{I1}(X)$, $F_{I2}(X)$ をそれぞれ算出すると、図1のようになった。

図中の点は、各パラメータの領域を細かくグリッド状に区切り、その各グリッドの点のすべてに対して関数の値を求めたものである。これらの点の集合の境界は値集合のそれにほぼ等しいものと考えられる。このようにして値集合を算出する方法を、以下ではグリッド法と呼ぶことにする。PIA により算出した値集合の推定とグリッド法により求めたものとを比較すると、式変形を行うことによりほぼ正確に値集合が推定できていることがわかる。 ■

次に、文献(5)の方法だけでは十分でない例として、DC モータの例を示す。

(例2) 次式は DC モータの電機子電圧からトルクへの伝達関数である⁽⁶⁾。

図1 値集合の推定 $F_{I1}(X)$, $F_{I2}(X)$
Fig.1. Evaluation of the value sets $F_{I1}(X)$, $F_{I2}(X)$.

$$f_1(s, \mathbf{x}) = \frac{K_T(J_L s + B_L)}{(Ls + R)(J_m s + J_L s + B_m + B_L) + K_V K_T} \quad \mathbf{x} = [J_m, B_m, L, R, J_L, B_L, K_V, K_T]^T \quad (5)$$

上式において、パラメータ J_L, B_L, K_T が分母、分子にともに現れている。そこで、この式を

$$f_2(s, \mathbf{x}) = \frac{1}{\frac{Ls + R}{K_T} + \frac{K_V + ((Ls + R)(J_m s + B_m)/K_T)}{J_L s + B_L}} \quad (6)$$

と変形すると、どの分母、分子にも共通なパラメータは現れていない。ここで、各パラメータの値を

$$\begin{aligned} J_L &\in [10^{-5}, 3 \cdot 10^{-5}], & K_V &\in [0.2, 0.6], \\ K_T &\in [0.2, 0.6], & J_m &= 2 \cdot 10^{-3}, \\ B_m &= B_L = 2 \cdot 10^{-5}, & L &= 10^{-2}, \\ R &= 1 \end{aligned}$$

に設定し、 $s = j\omega$ として $\omega = 50, 100$ rad/s で変形前および変形後の式を用いて PIA より計算する。

変形前の式では、分母、分子の多項式の値集合の推定をそれぞれ求め、分子の推定と、分母の推定の逆集合との積を計算することにより、値集合の推定 $F_{11}(j\omega, \mathbf{X})$ を求める。

変形後の式は、文献(4)、(5)でいう分解可能の条件を満たしていない。しかし、

$$\begin{aligned} x_1 &= \frac{Ls + R}{K_T}, & x_2 &= \frac{1}{J_L s + B_L}, \\ x_3 &= K_V, & c_1 &= J_m s + B_m \end{aligned}$$

と変数を置き換えて(6)式を書き直すと、

$$\begin{aligned} f_2(x_1, x_2, x_3, c_1) &= f_2(s, \mathbf{x}) \\ &= \frac{1}{x_1 + x_2(x_3 + c_1 x_1)} \quad (7) \end{aligned}$$

となり、分母は多重線形となっているので、端点集合の分割による方法を用いることができる。分母の多項式を $d_2(x_1, x_2, x_3, c_1)$ とすると、その値集合 $d_2(X_1, X_2, X_3, c_1)$ には、

$$\begin{aligned} d_2(X_1, X_2, X_3, c_1) &= \{d_2(x_1, x_2, x_3, c_1) | x_i \in X_i, i \in \{1, 2, 3\}\} \\ &\subseteq \text{conv}\{\{d_2(x_1, x_2, x_3, c_1) | x_i \in X_i, \\ &\quad i \in \{1, 2, 3\}\}\} \\ &= \text{conv}\{\{D_{12}(x_1, X_2, X_3, c_1) | x_1 \in \text{ext}(X_1)\}\} \quad (8) \end{aligned}$$

という関係がある。ここで、(8)式は値集合よりも拡

図2 値集合の推定 $F_{11}(j\omega, \mathbf{X}), F_{12}(j\omega, \mathbf{X})$ ($\omega=50$)

Fig. 2. Evaluation of the value sets $F_{11}(j\omega, \mathbf{X}), F_{12}(j\omega, \mathbf{X})$ ($\omega=50$).

図3 値集合の推定 $F_{11}(j\omega, \mathbf{X}), F_{12}(j\omega, \mathbf{X})$ ($\omega=100$)

Fig. 3. Evaluation of the value sets $F_{11}(j\omega, \mathbf{X}), F_{12}(j\omega, \mathbf{X})$ ($\omega=100$).

大している可能性があるが、その拡大はPIAの積の演算の際に生じるものなので、避けることはできない。よって、(8)式はPIAを用いた場合の最良の結果であるので、これにより分母の値集合の推定を算出し、その逆集合を求めることにより値集合の推定 $F_{12}(j\omega, \mathbf{X})$ を求める。

$F_{11}(j\omega, \mathbf{X}), F_{12}(j\omega, \mathbf{X})$ は図2、図3のようになった。グリッド法により算出したものと比較すると、式変形を行うことの有効性は明らかである。■

一般に、全パラメータが1箇所にしか出現しないような式変形が常に可能であるとは限らない。例2では端点集合の分割による方法を用いることができたが、この方法が常に使えるわけではない。そのような場合の対応策として、領域の細分化による方法⁹⁾が考えられる。この方法は、パラメータの存在領域(パラメータ空間)を細分化し、各々のパラメータ空間に対し

で算出したものの和集合として値集合を推定する方法である。この方法により、値集合の限りなく精密な近似を得ることが可能であることが知られている⁽³⁾。

また、TSD に関しては文献(5)にそのアルゴリズムが提案されているが、更に効率的な分解アルゴリズムが望まれる。なお、端点集合の分割による方法に関しては、文献(2)に経験的なアルゴリズムが提案されている。

本章では、伝達関数を同一パラメータがなるべく複数箇所に現れないように変形し、その式に対してPIAを適用することで、第1章で述べた値集合の推定が拡大する第1の原因を取り除くことができることを示した。

3. 逆集合の演算の改良

前章では、伝達関数の値集合をPIAにより推定する際の式変形の重要性について述べた。このように変形した結果得られる式では、逆集合の演算を数回行わなければならないことがわかる。(付14)式に示されるように逆集合の演算は不確かさを拡大してしまうので、2度以上の逆集合の演算によって得られる結果が更に拡大し、精度さが失われてしまう可能性がある。よって、より正確な演算結果を得るためには、逆集合の演算による拡大をできるだけ抑えることが肝要である。そこで、ここでは演算の定義を改良することによりこの拡大を抑えることを考える。なお、従来のPIAでは、すべての演算はまず P_{conv} の要素に対して定義され、それを P の要素にまで拡張した形で定義しているが、ここでは P_{conv} の要素に対する定義のみ改良し、 P の要素への拡張に関しては従来と同様の定義とする。

まず、従来の逆集合の演算が不確かさを拡大してしまう原因について述べる。 P_{conv} の要素 X の真の逆集合 $W=\{w=x^{-1}|x\in X\}$ は、複数の円弧に囲まれた領域となり、 P_{conv} にも P にも属さないようなものである(〔補題付1〕)。従って、これをPIAにより取り扱うためには、多角形として近似しなければならない。この近似の仕方により、得られる逆集合の精度が決まるといえる。従来提案していた逆集合の定義は、 P_{conv} の要素の逆集合がそのまま P_{conv} に属するようなものであった。これは真の逆集合の凸包に対する近似となるので、拡大は免れないものとなっていた。

そこで、新たな逆集合の定義では真の逆集合をより正確に近似するために、集合の凹の部分の円弧に関しても、拡大がある与えられた正の数 ϵ で抑えられるように折れ線で近似することを考える。すなわち、新

たな定義ではある与えられた正数 ϵ に対して、 P_{conv} の要素である凸多角形 X の逆集合 X^{-1} を、

$$W\subseteq X^{-1}\subseteq N(W, \epsilon) \dots\dots\dots (9)$$

となるように定義する。従来の定義は、(付12)式に示されるように、 W の凸包 W' に対して $W'\subseteq X^{-1}\subseteq N(W', \epsilon)$ となるようなものであった。この定義との比較より、新たな定義がより精密なものであることは明らかである。

このように定義を改良することにより、逆集合の演算による拡大を正数 ϵ で抑えることが可能である。しかし、この場合逆集合自身は一般に凸多角形とはならない。PIAでは、凸でない多角形は幾つかの凸多角形に分割し、それらの和集合として表すことにより取り扱う。従って、逆集合の演算を、真の逆集合の一般に凸でない多角形への近似と、その多角形の幾つかの凸多角形への分割という2段階で実行する。凸でない多角形を凸多角形へ分割する方法は様々なものが提案されているが、ここでは次のような分割アルゴリズムを用いることにする。

まず、このアルゴリズムで用いる語彙の定義をする。

(i) notch 多角形 P の端点 p_i の内角が 180° より大きいとき、 p_i はnotchであるという。また、 $\angle p_i p_j p_k$ が 180° より大きいとき、 $\angle p_i p_j p_k$ はnotchであるという。

(ii) visible 多角形 P の二つの端点 p_i, p_k を結ぶ線分が P の内部(境界線を含む)に完全に含まれるとき、 p_i から p_j (p_j から p_k)はvisibleであるという。

このアルゴリズムは、多角形のある辺に対して、それを含むような極大の凸多角形を切り出す操作を、次々と行うようなものである。

凸多角形へ分割すべき凸でない多角形を P とし、 P を分割した結果得られる凸多角形の集合を ν とすると、アルゴリズムは次のように記述される。

手続き $CD(P, \nu)$

(i) P の端点数 n を求め、辞書式順序の最初の端点を p_0 として反時計回りに p_{n-1} まで名前を付ける。

図4 notchとvisible
Fig. 4. Notch and visible.

- (ii) $\nu := \phi$.
 (iii) $i := 1$.
 (iv) $[(p_i \text{ は notch でない}) \wedge (p_0 \text{ から } p_{i+1} \text{ は visible}) \wedge (\angle p_{i+1} p_0 p_i \text{ は notch でない}) \wedge (i < n-1)]$ ならば $i := i+1$ として (iv) へ。
 (v) $i = n-1$ ならば P を凸多角形 V として ν に加え、終了。
 (vi) $[(p_0 \text{ は notch}) \vee (p_i \text{ から } p_{n-1} \text{ は visible でない}) \vee (\angle p_{i-1} p_i p_{n-1} \text{ は notch})]$ ならば、(vii) へ。そうでないならば、(vi-1) へ。
 (vi-1) $j := n-1$.
 (vi-2) $[(p_j \text{ は notch でない}) \wedge (p_i \text{ から } p_{j-1} \text{ は visible}) \wedge (\angle p_{i-1} p_i p_{j-1} \text{ は notch でない}) \wedge (j-1 > i)]$ ならば $j := j-1$ として (vi-2) へ。
 (vi-3) P から凸多角形 $V = \text{conv}\{p_i, \dots, p_{n-1}, p_0, \dots, p_j\}$ を切り離し、 $n := j-i+1$ とし、 p_i を p_0 として反時計回りに名前を付け変えたものを P とおく。 V を ν に加え、(iii) へ。
 (vii) $i < 2$ ならば (viii) へ。 $i \geq 2$ ならば P から凸多角形 $V = \text{conv}\{p_0, \dots, p_i\}$ を切り離し、 $n := n-i+1$ とし、 p_i を p_0 として反時計回りに名前を付け変えたものを P とおく。 V を ν に加え、(iii) へ。
 (viii) p_i を P の最初の端点 p_0 として、(iii) へ。

◇

(例3) 図5に示される多角形 P に対して、手続き $CD(P, \nu)$ を適用する。

- (1) 手順(i)により、 $p_0 \sim p_7$ を図のように決める ($n=8$)。
 (2) 手順(iv)では、端点 p_0 を含む凸多角形を反時計回りに大きくしてゆく操作を行っている。条件を満たさなくなるまで i をインクリメントしてゆく。
 (3) $i=3$ のとき、手順(iv)の「 p_i は notch でない」という条件を満たさないので、手順(v)へ。
 (4) 手順(v)の条件「 $i=n-1$ 」を満たさないので手順(vi)へ。

図5 多角形 P の凸分割

Fig. 5. Convex decomposition of the polygon P .

(5) 手順(vi)では、手順(iv)と逆の方向に凸多角形を大きくしてゆく。 $j=7, i=3$ で手順(vi-2)の「 $\angle p_{i-1} p_i p_{j-1}$ は notch ではない」という条件を満たさないので、 P から $V = \text{conv}\{p_7, p_0, \dots, p_3\}$ を切り離し、これを ν に加える。新たな P は、 $n=5$ として、 p_3 を p_0 とし、 $p_0 \sim p_4$ まで反時計回りに名前を付け変える。

(6) 新たな P に対し、同様の操作を行う。

(7) 最終的に、 ν の要素として三つの凸多角形 $\text{conv}\{p_7, p_0, \dots, p_3\}$, $\text{conv}\{p_6, p_7, p_3, p_4\}$, $\text{conv}\{p_4, p_5, p_6\}$ を得る。ただし、ここで端点の名前 $p_i (i \in \{0..7\})$ は初めに付けたものである。■

このアルゴリズムを用いて、[定義付2]は次のように改良される。

[定義1] 任意に $\varepsilon > 0$ を与える。 $\Gamma_{p,s}(\varepsilon)$ を中心が O_p 、半径が $(R_p + \varepsilon)$ である円弧とする。 O_p, R_p はそれぞれ $\Gamma_{p,s}$ をその一部として含む円の中心、半径である。凸多角形 $X = \text{conv}\{x_1, x_2, \dots, x_n\}$ を与え、

$$\begin{aligned} P &= \text{poly}\{w_1, x_{1,1}, \dots, x_{1,r_1}, w_2, \dots, x_{q,r_q}, \\ &\quad w_{q+1}, x_{q+1,1}, \dots, w_n, x_{n,1}, \dots, x_{n,r_{n-1}}\} \\ x_{p,k} &= O_p + (1 + \varepsilon_p/R_p)(w_p - O_p) \\ &\quad \times \exp[j(2k-1)\alpha_p] \\ p &\in [1..q], k \in [1..r_p] \\ x_{p,k} &= O_p + (w_p - O_p) \exp[-j2k\alpha_p] \\ p &\in [q+1..n], k \in [1..r_p-1] \\ r_p &\leq (\angle w_p O_p w_{p+1})/2 \cos^{-1}\{R_p/(R_p + \varepsilon)\} \\ p &\in [1..q] \\ r_p &\leq (\angle w_p O_p w_{p+1})/2 \cos^{-1}\{(R_p - \varepsilon)/R_p\} \\ p &\in [q+1..n] \\ \alpha_p &= (\angle w_p O_p w_{p+1})/2r_p \\ \varepsilon_p &= R_p(1 - \cos \alpha_p)/\cos \alpha_p \end{aligned}$$

とする。更に、この P に対して手続き $CD(P, \nu)$ を施したとき得られる凸多角形の集合 ν を用いると、 X の逆集合 X^{-1} は次式で与えられる。

$$X^{-1} = \bigcup_{i \in [1..|\nu|]} V_i, V_i \in \nu \dots\dots\dots (10)$$

■

また、逆集合の演算を改良したことにより、[補題付2]は次のように書き換えられる。

[補題1] X, Y を(付2)式、または(付7)式で与えられる多角形とする。このとき、以下のことが成り立つ。

- (i) $\{w = x + y | x \in X, y \in Y\} = X + Y$
 (ii) $\{w = x \cdot y | x \in X, y \in Y\} \subseteq X \bullet Y$

$$\dots\dots\dots (11)$$

図 6 値集合の推定 $F_{I2}(X)$
Fig. 6. Evaluation of the value set $F_{I2}(X)$.

図 7 値集合の推定 $F_{I2}(j\omega, X)$ ($\omega=50$)
Fig. 7. Evaluation of the value set $F_{I2}(j\omega, X)$ ($\omega=50$).

(iii) X が 0 を含まないとする。任意に $\varepsilon > 0$ を与えたとき、

$$W = \{w = x^{-1} | x \in X\} \subseteq X^{-1} \subseteq N(W, \varepsilon) \quad \dots\dots\dots (12)$$

が成り立つ。 ■

次に、逆集合を改良することによって値集合がより精確に推定される様子を例を用いて示す。

(例 4) 例 1 で挙げた例において、改良した逆集合の演算を用いて値集合を推定すると図 6 のようになった。さきの例と比較すると、逆集合の演算を改良することにより、より精確な推定が得られていることがわかる。 ■

(例 5) 例 2 で挙げた例において、各パラメータの値を同様に設定する。このとき、各周波数で改良前および改良後の逆集合の演算を用いた PIA より計算すると、図 7、図 8 のようになった。図より、改良前の逆集合を用いた場合に値集合よりも拡大していた

図 8 値集合の推定 $F_{I2}(j\omega, X)$ ($\omega=100$)
Fig. 8. Evaluation of the value set $F_{I2}(j\omega, X)$ ($\omega=100$).

部分が、改良後には排除されていることがわかる。 ■

PIA より算出される値集合の推定が、値集合そのものよりも拡大してしまう第 2 の原因は、PIA の演算自体によるものであった。しかし、逆集合の演算を改良することにより、残る課題は積の演算の改善だけとなった。また、逆集合の演算の改良は、第 2 章で述べた手法をより有効なものとするためにも必要不可欠であるといえる。

4. む す び

本論文では、PIA を適用した場合に不確かさの拡大を生ずる二つの理由を挙げ、計算式において同一パラメータになるべく複数箇所に同時に出現しないように伝達関数を変形すること、および逆集合の演算を改良することで、従来よりも値集合を精確に推定することが可能となることを示した。今後は次のような点について更に検討する必要があると考えられる。

まず、本論文では凸でない多角形を凸多角形に分割するアルゴリズムを示したが、このアルゴリズムによって分割した凸多角形の数、必ずしも最小となるとは限らない。この結果、PIA の演算の効率が低下する可能性がある。よって、これに関しては更に改良の余地があると考えられる。また、凸多角形の集合を取り扱う際に、他の凸多角形の和集合の内に含まれてしまうために不必要となる凸多角形を集合から取り除くことで、不要な演算をなくし、演算の高速化を図ることができると考えられる。更に、積の演算に対して、その不確かさの拡大の構造の解明と、拡大を抑えるための演算の改良が求められる。積の演算の拡大が任意に小さく抑えられれば、PIA より限りなく精確に

値集合を推定することが可能となる。以上のような点に対する研究が今後の課題である。

(平成 5 年 2 月 25 日受付)

文 献

- (1) Y. Ohta, L. Gong & H. Haneda: "Polygon interval arithmetic and design of robust control systems", Proc. of 29th Conf. on Decision and Control, Honolulu, Hawaii, p. 1065 (1990)
- (2) 松村・太田・羽根田: 「多重線形関数に関する一考察: 制御系のロバスト安定解析への応用」, 信学技報, 91, No. 411, CAS 91-100 (平 3)
- (3) 太田・安井・羽根田: 「多角形区間演算を用いた不確かさをもつた時間制御系の安定度の解析」, 計測自動制御学論, 27, No. 6, p. 620 (平 3)
- (4) B. R. Barmish, J. Ackermann & H. Hu: "The tree structured decomposition: a new approach to robust stability analysis", Proc. Conf. on Information Sciences and Systems, Princeton, NJ (1990)
- (5) W. Sienel: "Algorithms for tree structured decomposition", Proc. of 31st Conf. on Decision and Control, Tucson, Arizona, p. 739 (1992)
- (6) F. N. Bailey, D. Panzer & G. Gu: "Two algorithms for frequency domain design of robust control systems", Int. J. Control, 48, No. 5, p. 1787 (1988)

付 録

1. 多角形区間演算

代数系 PIA ($P; +, \bullet, (\cdot)^{-1}$) は次のように定義される⁽¹⁾。

まず初めに, P_{conv} の要素について定義する。

[定義付 1] 端点の集合が

$$\begin{aligned} \text{ext}(X) &= \{x_1, x_2, \dots, x_m\} \\ \text{ext}(Y) &= \{y_1, y_2, \dots, y_n\} \end{aligned} \quad \dots\dots\dots (\text{付 1})$$

で与えられるような二つの凸多角形

$$\begin{aligned} X &= \text{conv}\{x_1, x_2, \dots, x_m\} \\ Y &= \text{conv}\{y_1, y_2, \dots, y_n\} \end{aligned} \quad \dots\dots\dots (\text{付 2})$$

を考える。このとき, X と Y の和と積を

$$X + Y = \text{conv}\{x_1 + y_1, \dots, x_1 + y_n, x_2 + y_1, \dots, x_m + y_n\} \quad \dots\dots\dots (\text{付 3})$$

$$X \bullet Y = \text{conv}\{x_1 y_1, \dots, x_1 y_n, x_2 y_1, \dots, x_m y_n\} \quad \dots\dots\dots (\text{付 4})$$

で定義する。

次に, X の逆集合 X^{-1} を定義するが, この定義は少し複雑になる。その前にまず, 次の補題を示す⁽¹⁾。

[補題付 1] $\{x_1, x_2, \dots, x_n\}$ を凸多角形 X の端点の集合とする。 $L_{p,s} = \text{conv}\{x_p, x_s\}$ を x_p と x_s を結ぶ線分とする。ただし, $s = (p \bmod n) + 1$ である。 X は原点 0 を含まないものとし, X の境界は線分の集合 $\{L_{1,2}, \dots, L_{n-1,n}, L_{n,1}\}$ で与えられるものとする。また, $\Gamma_{p,s}$ を $\Gamma_{p,s} = \{w = x^{-1} \in C | x \in L_{p,s}\}$ で定義し,

$w_p = x_p^{-1}, w_s = x_s^{-1}$ とする。このとき, 以下のことが成り立つ。

(i) $0 \notin \text{aff}(\{x_p, x_s\})$ ならば, $\Gamma_{p,s} = \text{conv}\{w_p, w_s\}$ となる。

(ii) $0 \in \text{aff}(\{x_p, x_s\})$ ならば, $\Gamma_{p,s}$ は 3 点 $w_p, w_s, 0$ を通る円の一部分となる。円弧 $\Gamma_{p,s}$ は w_p, w_s を両端の点とし, 0 を含まない。

(iii) 集合 W を,

$$W = \{w = x^{-1} | x \in X\} \quad \dots\dots\dots (\text{付 5})$$

で定義すると, W は円弧 $\Gamma_{1,2}, \Gamma_{2,3}, \dots, \Gamma_{n,1}$ で囲まれた領域となる。

$\text{aff}(\{0, x_1\}), \text{aff}(\{0, x_{q+1}\})$ をそれぞれ, 0 から凸多角形 X への左接線, 右接線とすると, 円弧 $\Gamma_{p,p+1} (p \in [1..q])$ は W の境界の凸となる部分であり, 円弧 $\Gamma_{p,p+1} (p \in [q+1..n-1])$ と $\Gamma_{n,1}$ は W の境界の凹となる部分である(線分は, 凸とも凹ともみることができる)。 W の凸包を W' とすると, W' の境界は円弧と線分からなっているのので, W' は P の要素にならない。そこで, ある与えられた正数 ε に対して, 凸多角形 X の逆集合 X^{-1} を, $W \subseteq X^{-1} \subseteq N(W', \varepsilon)$ となるように定義する。

[定義付 2] 任意に $\varepsilon > 0$ を与える。 $\Gamma_{p,s}(\varepsilon)$ を, 中心が O_p , 半径が $(R_p + \varepsilon)$ である円弧とする。 O_p, R_p はそれぞれ $\Gamma_{p,s}$ をその一部として含む円の中心, 半径である。このとき, 凸多角形 $X = \text{conv}\{x_1, x_2, \dots, x_n\}$ の逆集合 X^{-1} は, 次式で与えられる。

$$\begin{aligned} X^{-1} &= \text{conv}\{w_1, x_{1,1}, \dots, x_{1,r_1}, w_2, \dots, \\ &\quad x_{q,r_q}, w_{q+1}, \dots, w_n\} \quad \dots\dots\dots (\text{付 6}) \\ x_{p,k} &= O_p + (1 + \varepsilon_p / R_p)(w_p - O_p) \\ &\quad \times \exp[j(2k-1)\alpha_p] \\ p &\in [1..q], k \in [1..r_p] \\ r_p &\leq (\angle w_p O_p w_{p+1}) / 2 \cos^{-1}\{R_p / (R_p + \varepsilon)\} \\ \alpha_q &= (\angle w_p O_p w_{p+1}) / 2 r_p \\ \varepsilon_p &= R_p [1 - \cos \alpha_p] / \cos \alpha_p \end{aligned} \quad \blacksquare$$

次に, 一般に P の要素について定義する。

[定義付 3] $\{X_i, i \in [1..m]\}, \{Y_j, j \in [1..n]\}$ が凸多角形であり, X, Y がそれぞれ

$$\begin{aligned} X &= \bigcup_{i \in [1..m]} X_i \\ Y &= \bigcup_{j \in [1..n]} Y_j \end{aligned} \quad \dots\dots\dots (\text{付 7})$$

で与えられる凸でない多角形であるとする。このとき, X と Y の和と積を

$$X + Y = \bigcup_{i \in [1..m], j \in [1..n]} (X_i + Y_j) \quad \dots\dots (\text{付 8})$$

$$X \bullet Y = \bigcup_{i \in [1..m], j \in [1..n]} (X_i \bullet Y_j) \quad \dots\dots (\text{付 9})$$

で定義する。また, X の逆集合を

$X^{-1} = \bigcup_{i \in \{1..m\}} X_i^{-1}$ (付 10)
 で定義する。 ■

PIA に関して、次の結果が成り立つことが示されている⁽¹⁾。

〔補題付 2〕 X, Y を (付 2) 式、または (付 7) 式で与えられる多角形とする。このとき、以下のことが成り立つ。

- (i) $\{w = x + y | x \in X, y \in Y\} = X + Y$
- (ii) $\{w = x \cdot y | x \in X, y \in Y\} \subseteq X \bullet Y$
 (付 11)

(iii) X が 0 を含まないとする。任意に $\varepsilon > 0$ を与えたとき、

$$W = \{w = x^{-1} | x \in X\} \subseteq W' \\ \subseteq X^{-1} \subseteq N(W', \varepsilon) \text{ (付 12)}$$

が成り立つ。ただし、 W' は、 X が (付 2) 式で与えられる凸多角形のときは W の凸包であり、(付 7) 式で与えられるときは、

$W' = \bigcup_{i \in \{1..m\}} W'_i$ (付 13)
 である。ここで、 W'_i は、 $W_i = \{w = x^{-1} | x \in X_i\}$ の凸包である。 ■

〔補題付 3〕 X, Y, W を P の要素とし、 z を 0 でない複素数とする。このとき、以下の結果が成り立つ。

- (i) $X + Y = Y + X, X \bullet Y = Y \bullet X$
 (交換律)
- (ii) $(X + Y) + W = X + (Y + W)$ (結合律)
 $(X \bullet Y) \bullet W = X \bullet (Y \bullet W)$
- (iii) $X + 0 = 0 + X = X, X \bullet 1 = 1 \bullet X = X$
- (iv) $(z + X) + (-z) = X, z \bullet X \bullet z^{-1} = X$
- (v) $z \bullet (X + Y) = z \bullet X + z \bullet Y$
- (vi) $\{0\} \subseteq X + (-1) \bullet X$

(vii) $\{1\} \subseteq X \bullet (X^{-1})$ (付 14)

(viii) $X \bullet (Y + W) \subseteq X \bullet Y + X \bullet W$
 (準分配律)
 (付 15)

磯貝 晃 (非会員)

平成 4 年神戸大学工学部電子工学科卒業。現在、同修士課程在学中。ロバスト制御系の設計、解析の研究に従事。電子情報通信学会会員。

太田 有三 (正員)

昭和 52 年大阪大学大学院博士課程修了。同年福井大学助手。62 年神戸大学工学部電子工学科助教授、現在に至る。工学博士。ロバスト制御、計算機援用システム設計、解析の研究などに従事。電子情報通信学会、システム制御情報学会、計測自動制御学会、IEEE 会員。

羽根田 博正 (正員)

昭和 47 年カリフォルニア大学(パークレー校)大学院博士課程修了。同年神戸大学工学部電子工学科助教授、57 年同教授、現在に至る。Ph.D. 計算機援用システム設計、解析に関する研究などに従事。電子情報通信学会、システム制御情報学会、計測自動制御学会、IEEE、ACM 会員。