宇宙機のスラスト制御系へのファジィ技術の適用^{*1}

An Application of Fuzzy Technique to Thruster Control System of Spacecraft

鈴木 孝^{*2}・安田 国治^{*2}

Takashi SUZUKI, Kuniharu YASUDA,

吉河 章二^{*3}・山田 克彦^{*3}

Shoji YOSHIKAWA, Katsuhiko YAMADA,

吉田 憲正^{*4}

and Norimasa YOSHIDA

Key Words: Spacecraft, Guidance and Control, Rendezvous and Docking, Fuzzy Control, Fuzzy Set Theory, Thruster Control System

Abstract

Thruster assignment to generate ideal forces and torques is important for a chaser spacecraft approaching another spacecraft (a target spacecraft), for example, during rendezvous and docking operations. The thruster selection is more important in the proximity area of the target spacecraft where the chaser spacecraft controls both relative position and attitude. The conventional method is based on a table look-up. It is easy to execute (calculation time minimum) but tends to generate a wasteful pattern of thruster select. Two new methods based on fuzzy logic are proposed: one removes such wasteful patterns using fuzzy pattern matching and the other selects thrusters one by one while it reduces the residual control forces and torques with smaller fuel consumption using fuzzy inference. Numerical simulations are given to compare the conventional method, the proposed fuzzy methods and the linear programming (fuel minimum). The results show that the proposed fuzzy methods realize a thruster select logic with low fuel-consumption in allowable calculation time and that they provide a flexibility in the trade off between calculation time and fuel consumption.

1. はじめに

将来の宇宙活動では、システムが大型化・複雑化し、またミッションが多様化・同時並行化することが予想される。このような場合には、従来手法のみでは解決が困難であったり、設計・運用コストが爆発的に

増大してしまうような問題点も抱えている¹⁾。一方、ファジィ技術の適用により、従来手法に比べると比較的小規模なシステムで解決策を提示できる可能性があることが、家電製品をはじめとする地上の産業分野において報告されている^{2,3)}。そのため、近年、宇宙分野においても、誘導制御系等へのファジィ技術の適用が検討されつつある^{1,4-9)}。

本論文では、宇宙機の誘導制御系へのファジィ技術適用の一例として、スラスト制御系におけるスラスト選択則をとりあげる。

宇宙機のスラスト選択則の高機能化が求められる背景をまとめると以下ようになる。一般の宇宙機、とりわけランデブ・ドッキングする能力を備えた移動型の宇宙機は、その位置・姿勢の制御に、ガスジェットを噴射するスラスト制御系を用いることが多い。任意の制御力・トルクに対して適切なスラストを選択し噴射量を決めるのが、スラスト選択則の役割である。ところが、太陽電池パドル、アンテナ、マニピュレータームなどの搭載機器やターゲット宇宙機へのブルーム外乱を回避するために、スラスト配置が複雑にならざるを得ない。また同じ理由で、偶力のみを発生するようにスラストを配置することが困難である。結果的に、各スラストが力とトルクの両方を発生したり、複数の軸方向に力を発生したり、複数の軸まわりのトルクを発生したりするため、合力として同じ制御力・トルクを生成するようなスラストの組み合わせが一般に複数組存在する。それらの中では推奨消費量が最も小さい組み合わせが望ましい。さらに、宇宙環境が過酷で搭載計算機能力が限られているため、スラスト選択に必要な計算量やプログラムサイズが小さいことが望

^{*1} 平成7年11月27日原稿受理^{*2} 宇宙開発事業団筑波宇宙センター^{*3} 三菱電機株式会社先端技術総合研究所^{*4} 三菱電機株式会社鎌倉製作所

ましい。

通常の宇宙機では、スラストの配分をテーブルルックアップ方式¹⁰⁾で行う。この方法は実装性が高い反面、後述するように噴射効率が低いという問題があった。

一方、線形計画法を用いてオンボードで推薬消費量を最小とする解を求める方法が、米国のスペースシャトルの誘導制御系に提案されている¹¹⁾。しかし、計算量が大きく実装性に難があるといわざるを得なかった。

このような背景をふまえ、本論文ではファジィ技術の適用を二通り提案する。

第一の方法は、テーブルルックアップ法に基づいて力・トルク要求を満足するようなスラスト配分を求め、それから計算量とのトレードオフを考慮しつつ、パターンマッチングにより、噴射効率の低い配分を効率的な配分へと置き換えるものである。置き換えは常に合計噴射量を減らす方向に行うため、減算法と呼ぶことにする。ファジィ技術は、パターンマッチングのところに適用する。

第二の方法は、合計噴射量が0のところから出発して、できるだけ合計噴射量を増やさないようにしながら、力・トルク要求を満足するまで、スラストのデューティー比を増やしていく方法である。第一の方法と対比して、加算法と呼ぶことにする。ファジィ技術は、できるだけ合計噴射量を増やさないようにスラストの組み合わせを決めるところに適用する。

本論文の構成は、以下の通り

第2章で、スラスト選択問題を定式化する。第3章で、従来のスラスト選択則として、テーブルルックアップ法および線形計画法のアルゴリズムを整理する。第4章で、ファジィ技術(減算法、加算法)を用いたスラスト選択則のアルゴリズムを提案する。第5章で、これら四手法を比較するために、宇宙機(チェイサ)が別の宇宙機(ターゲット)の位置と姿勢へ追従する近傍域制御の場合について数値シミュレーション

を行う。第6章で、結論を述べる。

2. スラスト選択問題の定式化

位置・姿勢の制御にガスジェットを用いる宇宙機にとって、必要な制御力およびトルクを合力として生成するようなスラスト配分を求めることが必要である(=スラスト選択問題)。

スラスト制御系におけるスラスト選択則の位置づけを第1図に示す。図において、スラスト選択則への入力は、生成すべき制御力 f_c と制御トルク n_c を併せたベクトル $b = [f_c^T \ n_c^T]^T$ である。スラスト本数を n とすると、スラスト# i に対する配分 x_i ($i=1, \dots, n$) は、最大推力に対する要求推力の割合 (=デューティー比) を表す。定義からデューティー比 x_i は0以上1以下でなければならないが、運用時にデューティー比が1を越えるような制御力あるいはマヌーバ量が要求されるとは考えにくい。実用上は、

$$0 \leq \forall x_i \quad (i=1, \dots, n) \quad (1)$$

が拘束条件である。

スラスト選択則からの出力は、各スラストへの配分 $x = [x_1, x_2, \dots, x_n]$ である。実際には、各スラストはパルスの力しか発生できないので、モジュレータでスラストの噴射間隔の疎密を調整することで任意の制御力・トルクを生成する。このようなモジュレーションを、制御力・トルクの成分毎に行う場合と、スラスト毎に行う場合があるが、本論文ではスラスト毎にモジュレーションを行う場合について考察する。

スラスト# i を駆動することにより生成される制御力・トルクをまとめてベクトル a_i と書くと

$$a_i = \begin{bmatrix} F_i d_i \\ p_i \times F_i d_i \end{bmatrix} \quad \begin{matrix} \text{(制御力)} \\ \text{(制御トルク)} \end{matrix} \quad (2)$$

を得る。ここで、 F_i はスラストの最大推力、 d_i は推力の向きを表す単位ベクトル(無次元量)、 p_i は宇宙機の質量中心 O_{BC} に対するスラスト# i の位置ベクトルである。合力として得られる制御力・トルクは

$$x_1 a_1 + x_2 a_2 + \dots + x_n a_n = A x \quad (3)$$

第1図 スラスト制御系におけるスラスト選択則の位置づけ

で与えられる。ここで、 $A=[a_1 \ a_2 \ \cdots \ a_n]$ はスラスト行列である。

全推進消費量は、次式に比例する。

$$\sum_{i=1}^n F_i x_i$$

スラスト選択問題とは、次の問題を解くことである。

$$Ax=b, \quad 0 \leq \forall x_i, \quad (i=1, \cdots, n)$$

を満たす解 x を (1つ) 求めよ。

ただし、全推進消費量は少ない方が望ましい

A の次元 (数値シミュレーション例で 6×14) からわかるように、一般に解 x は複数存在する。解を選ぶにあたって、合計噴射量が少ない方が望ましいしかし、そのような解を求めるには、一般に計算量が增大するため実装性が悪化する恐れがある。本論文は、合計噴射量と計算機負荷のバランスをとった効率の良いスラスト選択則を提案するものである。

3. 従来手法

3.1 テーブルルックアップ法 従来は、実装性と簡便性などからテーブルルックアップ法が採用されてきた。

テーブルの中身は、単軸 (純力または純トルク) 毎に求めた最適なスラストの組み合わせと分配量である。各スラストがそれぞれ一方向の推力しか生成できないので、テーブルも各軸毎に正負両方用意することが必要で、合計で $6 \times 2 = 12$ 組必要である。

この方法ならば、必要とされる制御力およびトルクに対応してテーブルの値をスカラー倍してスラスト毎に和を取れば良く、計算が容易である。なお、各軸毎に用意するスラストの組み合わせと分配量もルール的一种であり、テーブルルックアップ法は12個のルールを利用すると考えることができる。

このように、テーブルルックアップ法は実装も計算も容易である反面、欠点が二つある。

第一に、複数軸の制御を行う場合には、スラストが対称性良く配置されていても無駄噴きを生じやすい例えば、第2図に示すように、スラストが宇宙機の軸に沿って配置されている単純な例を考える。図において、純トルクを生成するには(a)に示すように#1と#2の2本のスラスト噴射を必要とし、純力を生成するには(b)に示すように#3の1本のスラスト噴射を必要とする。同時に純力と純トルクの両方を生成するためには、(c)に示すように単純にそれぞれのスラスト配分を加える場合 (テーブルルックアップ法) と、(d)に示すようにスラストの組み合わせを変更する場合があります。いずれも同じ制御が可能である。しかし、(c)では打ち消し合って合力としては制御に寄与しな

第2図 単純なスラスト配置

いような噴射パターン (=無駄噴き) を含んでおり、(d)に比べて噴射効率が低い。

第二に、複数軸の場合に対応してテーブルを拡張することが、困難である。例えば、同じ軸 (y 軸と z 軸) についてのルールを用意しても、制御力の大きさ (F_y と F_z の内訳) によって最適なスラストの組み合わせが一般に異なるため、ルールの数を単純に増やしても推進の効果的な節約にはなかなか結びつかない。

このような二つの欠点のために、二軸、三軸のように制御力と制御トルクの複数の成分の組み合わせについて適切なスラストの組み合わせを与えるようなルールを用意しようとすると、必要なルールは爆発的に増大し、テーブルルックアップ法の持つ実装性という利点が失われてしまう

3.2 線形計画法 スラスト選択問題を線形計画法で再定式化すると次のようになる。

$$Ax=b, \quad 0 \leq \forall x_i, \quad (i=1, \cdots, n)$$

を満たす x で

$$\sum_{i=1}^n c_i x_i, \quad c=[c_1, \cdots, c_n]$$

を最小にする x を求めよ。

線形計画法の解法には、シンプレックス法¹²⁾を用いた。全推進消費量を最小にするために、重みベクトル c を次式のように設定する。

$$c=[F_1, F_2, \cdots, F_n]$$

4. ファジロジック

すでに述べたように、テーブルルックアップ法では実装性が高い反面、無駄噴きを部分的に含んでしまうことが多く噴射効率が低いという問題があった。一方、線形計画法は消費推進量の意味で最適な解が得られる反面、計算機負荷が過大であるという問題があった。

そこでファジィ技術を適用し、以下に詳述する減算法と加算法の二つの方法を提案する。

4.1 減算法 スラストの噴射パターンに「無駄」

があるということは、その「無駄」なスラストの組が新たな制御力・トルク成分を生じていないということと等価で、そのようなスラストの組み合わせが1次従属の関係にあるということである。

スラストの i 番目の組み合わせ $\{i_1, i_2, \dots, i_m\}$ が1次従属の関係にあるとき

$$e_1 \mathbf{a}_{i_1} + e_2 \mathbf{a}_{i_2} + \dots + e_m \mathbf{a}_{i_m} = \mathbf{0}, \quad \|\mathbf{e}_i\| \neq 0 \quad (4)$$

となるような係数ベクトル $\mathbf{e}_i = [e_1, e_2, \dots, e_m]^T$ が存在する。書き直して

$$\mathbf{A}_m \mathbf{e}_i = \mathbf{0}, \quad \|\mathbf{e}_i\| \neq 0 \quad (5)$$

ここで

$$\mathbf{A}_m = [\mathbf{a}_{i_1}, \mathbf{a}_{i_2}, \dots, \mathbf{a}_{i_m}]$$

を得る。ここで、 $\|\cdot\|$ は通常のユークリッドノルムを表す

式(5)は、 $\mathbf{e}_i$ が行列 $\mathbf{A}$ の部分行列 $\mathbf{A}_m$ の零空間に属することを表す。この $\mathbf{e}_i$ をスラスト配分と解釈すると、符号によって無駄噴きを次の二つのパターンに分類できる。

- (1) 相殺噴射パターン：複数のスラストが生成する制御力およびトルクの合力を求めたときに、相殺しあってその値が $\mathbf{0}$ になる噴射パターン
- (2) 置換噴射パターン：生成する制御力およびトルクの合力は同じで、別のスラストの組み合わせに置き換えた方が合計噴射量が少なくて済む噴射パターン

ベクトル $\mathbf{e}_i$ の各要素が、同符号であれば相殺噴射パターン、異符号であれば置換噴射パターンに対応する。また、相殺噴射パターンと置換噴射パターンを抽出することは、行列 $\mathbf{A}$ の部分行列の零空間を張るベクトルを求めることと等価である。ここで得られたベクトル $\mathbf{e}_i$ を

$$\mathbf{e}_i \leftarrow \frac{\mathbf{e}_i}{\sum_{j=1}^m e_j} \quad (6)$$

のように、要素の和が1に等しくなるようにスケールリングする。このようにスケールリングを行うと、後述するルールマッチングにおいてマッチングの度合いがルール適用により節約できる噴射量を表す利点がある。なお、置換噴射パターンの特殊なケースとして、式(6)の右辺の分母が0になる場合がある。これは、スラストの組み合わせを置換しても合計噴射量が変わらないことを表すもので、ここではルールから除く。

こうして得られた i 番目のルール $\mathbf{e}_i$ のサイズを $n \times 1$ に揃えて $\mathbf{r}_i$ とし、ルールセットに格納する。ここで、 $\mathbf{r}_i$ の j 番目の要素を次式で与える。

$$r_{ij} = \begin{cases} 0 & (j \in \{i_1, i_2, \dots, i_m\}) \\ e_j & (j \in \{i_1, i_2, \dots, i_m\}) \end{cases} \quad (7)$$

無駄噴きの除去にあたり、このようなルールセットの中のどのルールから適用していくかが問題である。ルールの適用で節約できる合計噴射量が多いものを優先するというのが、ルール適用の順番として合理的であると考えられる。

ところで、スラスト配分 $\mathbf{x}$ が与えられたとき、各ルールが適用できるか否かは、ベクトル $\mathbf{x}$ がある特定のスラストの組み合わせを含んでいるか否か (“Yes” または “No” の2値) で判断される。一方、スラスト配分 $\mathbf{x}$ の各要素は0~1の連続量である。

ルールのマッチングを行うにあたって、スラスト配分 $\mathbf{x}$ の要素が閾値 α より大きいものについてのみ抜き出せば、マッチングに必要な組み合わせも少なく済みまたマッチングがとれた場合に節約できる合計噴射量も大きいと考えられる。

このために、 α -カット法¹³⁾を導入する。 α -カット法はファジィ集合理論の手法の一つで、次式のようにファジィパラメータ α を用いて集合 P から部分集合 P_α を抽出する。

$$P_\alpha = \{x | \mu_P(x) \geq \alpha\}, \quad (8)$$

$$x \in P, \quad \mu_P \in [0, 1], \quad \alpha \in [0, 1]$$

ここで、 $\mu_P(x)$ は x がどれだけ集合 P に属するかを表すメンバシップ関数である。

α -カット法を用いた減算法によるスラスト選択則は、次の4ステップで行う(第3図)。

- S1: テーブルルックアップ法で、スラスト配分 $\mathbf{x}$ を求める。
- S2: ファジィパラメータ α 以上の要素をベクトル $\mathbf{x}$ から抜き出して新しいベクトル $\mathbf{x}^\alpha$ を求める (α -カット法)
- S3: 各ルールの適合度を計算する。 i 番目のルールの要素が $\mathbf{r}_i = [r_{i1}, \dots, r_{in}]$ で与えられるとき、 i 番目のルールの適合度 h_i ($i=1, \dots, N$) を

$$h_i = \min_{j \in \{1, \dots, n\}} \frac{x_j^\alpha}{r_{ij}} \quad (r_{ij} > 0) \quad (9)$$

で定義する。ここで N はルールの数である。適合度 h_i は、 i 番目のルールで削減しうる最大の無駄噴きの大きさを表し、 h_i が大きいほどルール $\mathbf{r}_i$ の適用が望ましい。なお、もし、 $h_i=0$ であれば、スラスト配分 $\mathbf{x}^\alpha$ はルール $\mathbf{r}_i$ で表される無駄噴きパターンを含んでおらず、ルール $\mathbf{r}_i$ は適用できない。適合度の計算をすべてのルールについて行い、その最大値 $h_{\max}$ を求める。また、このときのルール番号を $i_{\max}$ とする。

- S4: ベクトル $\mathbf{x}$ を

$$\mathbf{x} \leftarrow \mathbf{x} - h_{\max} \mathbf{r}_{i_{\max}} \quad (10)$$

で更新する。

第3図 ファジィ (減算法) によるスラスト選択則

第4図 ファジィ (加算法) によるスラスト選択則

すべてのスラストの最大推力が等しい ($F_j = F$, $j = 1, \dots, n$) とき, 消費推薬の節約量は

$$\frac{1}{I_{SP}} \sum_{j=1}^n F h_{\max} r_{i_{\max} j} = \frac{1}{I_{SP}} F h_{\max} \quad (11)$$

となる。ここで I_{SP} は推薬の比推力である。このように適合度 $h_{\max}$ は推薬消費量の節約の大きさを直接的に表している。

実用上、これらの4ステップを繰り返してより効率的なスラスト配分を求める。

4.2 加算法 加算法では、合計噴射量が0のところから出発して、できるだけ合計噴射量を増やさないようにスラストの組み合わせを決め、その組み合わせに対して噴射量を配分する。

今、目標とする力・トルクを b とし、加算法の $i-1$ 番目のステップの解として、スラスト配分ベクトル $x^{(i-1)}$ が求められているとする。すなわち、 $i-1$ 本のスラストに、噴射配分しているとする。この時点

での残りの力・トルク (= 残差ベクトル) を $e^{(i-1)}$ とすると、 $e^{(i-1)} = b - Ax^{(i-1)}$ である。

加算法によるスラスト選択則は、次の4ステップで行う (第4図)。

S1: 残差ベクトル $e^{(i-1)}$ と、各スラストの推力ベクトルを正規化したベクトル $\hat{a}_1, \dots, \hat{a}_n$ との内積を求める。 j 番目のスラストを候補とする場合に、内積 r_j 、噴射量の増分 s_j および残差ベクトル $e_j^{(i)}$ はそれぞれ次式で与えられる。

$$r_j = \hat{e}^{(i-1)} \cdot \hat{a}_j \quad (12)$$

$$\hat{e}^{(i-1)} = \frac{e^{(i-1)}}{\|e^{(i-1)}\|}, \quad \hat{a}_j = \frac{a_j}{\|a_j\|}$$

$$s_j = \frac{e^{(i-1)} \cdot \hat{a}_j}{\|a_j\|} \quad (13)$$

$$e_j^{(i)} = e^{(i-1)} - s_j a_j \quad (14)$$

$$\|e_j^{(i)}\|^2 = \|e^{(i-1)}\|^2 (1 - r_j^2) \quad (15)$$

式(13)において、右辺の分子は、残差ベクトル

$e^{(i-1)}$ の j 番目のスラストによる制御力・トルクの方向の成分を表し、それをスラストによる制御力・トルクの大きさ $\|\mathbf{a}_j\|$ で除することで、 s_j として噴射量の増分を求めている。また、式(12)で定義した内積 r_j は、式(15)より、その値が大きいほど、 j 番目のスラストを候補とした場合の残差ベクトルの大きさが小さい。

S2: 残差ベクトルの大きさ (誤差) をできるだけ小さくすると同時に、そのために必要な噴射量をできるだけ小さくするために、内積 r_j が大きい (big) こと、および噴射量 s_j が小さい (small) ことをそれぞれ第5図(a)(b)に示すようなメンバシップ関数で表し、次のように、スラストの「望ましき」 h の評価をファジィルール化する。

内積が大きくて、かつ噴射量が小さい
 ならば、そのスラストは望ましい

↓

if r_j is big and s_j is small,

then $h_j=1$ (16)

また、第5図(c)に示すように、前件部のメンバシップ関数の値の積を用いて、「望ましき」を非ファジィ化する。このような、「望ましき」をスラスト毎に計算し、その最大値を与えるスラストを候補として選択し、スラストの組み合わせに付加する。

S3: 加算法では、スラストを選び噴射を割り当てることを繰り返して、力・トルク要求を満足するデューティーを逐次的に求める。任意の力・トルク要求に対して、解が存在するならばスラ

タの本数は最大6本で構成できる。また、この繰り返し (ループ) の回数が多いほど計算量が增大するので、残差ベクトルが十分小さければ、繰り返しは打ち切ることが望ましい。そこで、選んだスラストの本数が6本になるとき、または残差ベクトルがスラストの分解能以下になる場合には、繰り返しを終了する。

S4: 終了判定後は、残差ベクトルをテーブルルックアップ法によって埋め合わせる。そのために、加算法ではテーブルルックアップ法で用いるスラスト・アサイン・マトリクスを必要とする。

4.3 ファジィロジックの特徴 以上述べた減算法および加算法に共通の長所は、次の2点である。

- (1) 計算量やプログラムサイズの増加を抑えつつ、無駄噴きの少ないスラスト選択が可能。
- (2) スラスト故障時のスラスト再構成にも適用可能。

ただし、減算法では初期の解を求めるために、加算法では繰り返し終了後に残差ベクトルを埋め合わせるために、従来のテーブルが必要である。また、各方法の主な長所短所を以下に挙げる。

(1) 減算法

長所: マッチングの途中で得られる解も、力・トルク要求を満足。

短所: テーブルルックアップで求めた初期の解から全く異なるスラスト組み合わせを求めることは困難。

(2) 加算法

長所: 初期の解を用いないので、減算法では求めることができないようなスラスト組み合わせを求めることが可能。

短所: 積み上げの途中で得られる解が、力・トルク要求を満足しない。

5. 数値シミュレーション

本章では、これまで述べた四手法によるスラスト選択則を用いてターゲットに対するチェイサの近傍域制御を想定して数値シミュレーションを行い、各手法を比較検討する。

5.1 シミュレーションモデル チェイサのターゲットに対する並進運動の方程式は、Hillの座標系¹⁴⁾で次のように表される。

$$\begin{pmatrix} \ddot{x}_c \\ \ddot{y}_c \\ \ddot{z}_c \end{pmatrix} = 2\omega \begin{pmatrix} \dot{z}_c \\ 0 \\ -\dot{x}_c \end{pmatrix} + \omega^2 \begin{pmatrix} 0 \\ -y_c \\ 3z_c \end{pmatrix} + \frac{1}{M} \mathbf{f}_H \quad (17)$$

ここで、 $\mathbf{p}_c = [x_c, y_c, z_c]^T$ はターゲットに対するチェイサの質量中心位置、 ω はターゲットの軌道角速度、 M はチェイサの質量、 $\mathbf{f}_H$ は制御力を表す

第5図 加算法におけるファジィルール

第6図 スラスタ配置

チェイサおよびターゲットの回転運動の方程式は、次のように表される。

$$I_c \dot{\omega}_c + \omega_c \times I_c \omega_c = n_c \quad (18)$$

$$I_T \dot{\omega}_T + \omega_T \times I_T \omega_T = 0 \quad (19)$$

ここで、 ω_c はチェイサの角速度、 I_c はチェイサの慣性テンソル、 n_c は制御トルクを表し、各成分をチェイサ機体座標系で表現する。 ω_T はターゲットの角速度、 I_T はターゲットの慣性テンソルを表し、各成分をターゲット機体座標系で表現する。

チェイサおよびターゲットに働く外力は無視できるものとし、またすべての状態変数を観測できるものとする。

相対位置および姿勢に対して、次のPD制御則を採用する。

$$\begin{aligned} f_H &= M\{K_p(p_d - p_c) + K_d(\dot{p}_d - \dot{p}_c)\} \\ n_c &= I_c\{K_p\theta_{CT} + K_d(C_T^c\omega_T - \omega_c)\} \end{aligned} \quad (20)$$

ここで、 K_p は比例ゲイン、 K_d は微分ゲイン、 p_d は軌道プロファイル、 θ_{CT} はチェイサに対するターゲットの微小オイラー角、 $C_T^c\omega_T$ はターゲット角速度（チェイサ機体座標系表現）を表す。

スラスト行列 A はチェイサ機体座標系で表現されているから、スラスト選択則への入力 b は

$$b = \begin{pmatrix} f_c \\ n_c \end{pmatrix}, \quad f_c = C_H^c f_H \quad (21)$$

で与えられる。ここで、 C_H^c はHill座標系からチェイサ機体座標系への方向余弦行列である。

スラスト配置は、ランデブ・ドッキングを行う技術試験衛星 VII 型の初期の設計フェーズにおけるコン

第1表 初期条件

項目	単位	$x(\phi)$	$y(\theta)$	$z(\phi)$
チェイサ				
相対位置	m	5.05	0.05	0.05
相対速度	m/s	0	0	0
姿勢	deg	3.0	2.0	2.0
角速度	deg/s	0.5	0.5	0.5
ターゲット				
姿勢	deg	-0.5	-0.5	-0.5
角速度	deg/s	-0.05	-0.05	-0.05

第7図 速度プロファイル

フィギュレーションをベースに第6図のものでモデル化した。

配分ベクトルの成分 x_i が1を越える場合には、最大成分が1になるように x をスケーリングし直す。この操作は、デューティ比が1を越えないように制御ゲインを一時的に小さくすることに相当する。

チェイサは、質量が2000 kgで x , y , z 軸まわりの慣性モーメントがそれぞれ7500, 2500, 8700 kgm² とする。ターゲットは、高度550 kmの円軌道を周回するものとする。

5.2 初期条件 両宇宙機の初期条件を第1表に示す。角速度が0でないのは誘導制御誤差を考慮したためである。ターゲットに対する接近は、Hill座標系の x_c 軸に沿ってノミナルで相対位置5m地点から出発し2m地点で終了する。速度プロファイルを第7図に示す。

5.3 シミュレーション結果 チェイサの相対速度の x_c 成分および合計噴射量の時間履歴を、それぞれ第8図の(a)と(b)に示す。図において、実線の細かい順にテーブルルックアップ法、減算法 (Fuzzy 1)、加算法 (Fuzzy 2) に対応し、点線は線形計画法に対応する。図より以下のことがわかる。

(1) (a)より、制御性能においては、四手法とも差異はほとんどみられない。追従開始時など速度プロファイルが大きく変わるところでは制御量が大きくなり多少のオーバーシュートが見られるが、十分な制御精度を達成している。

(a) 相対速度 (x)

(b) 合計噴射時間

第8図 相対速度および合計噴射時間の比較

第9図 計算機負荷および推薬消費量の比較

- (2) (b)より, 合計噴射時間においては, 四手法で大きな差が生じている. 線形計画法が最も少なく, 次いでファジィロジック (加算法, 減算法) が少ない. 速度プロファイルが大きく変わるところで, 大きな噴射量が要求されている理由は上と同じ.

次に, 計算機負荷と推薬消費量の観点からこれら四手法のトレードオフを行い, 第9図に示す. 図において, 横軸は, テーブルルックアップ法での値を1としたときのスラスト選択則部分の計算時間の比を対数目盛で示す. 縦軸は, テーブルルックアップ法での値を1としたときの合計噴射時間の比を表す. 減算法については, ファジィパラメータ α を0, 0.1, 0.2の3通りに変えた場合の結果を示す. α の値が小さいほど適合度をチェックする組み合わせが増え, 演算量が増加するが無駄噴きの除去効果も大きい.

図より, 以下のことがわかる.

- (1) ファジィロジック (減算法, 加算法) は, テーブルルックアップ法 (計算時間が短い) あるいは線形計画法 (合計噴射時間が最小) と比較して, 計算時間と合計噴射時間のバランスをとったものになっている.
- (2) 減算法の場合は, 計算時間と合計噴射時間のトレードオフを, ファジィパラメータ α によって容易に調整可能である.

6. 結 論

本論文では, 宇宙機のスラスト制御系のスラスト選択則について考察した. 従来用いられてきたテーブルルックアップ法は, 無駄噴きを含みやすく噴射効率が低いという問題があり, 線形計画法で推薬消費量の最小解を求める方法は, 計算量が大きく実装性に問題があった. そこで, ファジィ技術の適用を二通り提案した.

第一の方法 (減算法) は, テーブルルックアップ法で求めたスラスト配分から, パターンマッチングによって合力として制御力およびトルクを発生しないような無駄噴きパターンを検出し取り除く. このパターンマッチングにおいて, α -カット法と呼ばれるファジィ手法を適用し, 合計噴射時間 (推薬消費量) と計算時間 (計算機負荷) のトレードオフを実施する.

第二の方法 (加算法) は, 合計噴射量が0のところから出発して, ファジィ推論によってできるだけ合計噴射量を増やさないようにしながら, 力・トルク要求を満足するまでスラストのデューティー比を増やす.

宇宙機が別の宇宙機の位置と姿勢へ追従する近傍域制御の場合について数値シミュレーションを行った. その結果, 従来のテーブルルックアップ法に比べて, 減算法は, ファジィパラメータ $\alpha=0\sim0.1$ で推薬消費量が14~20%減少し計算時間が6~11倍になった. 加算法は, 推薬消費量が29%減少し計算時間が17倍になった. 一方, 線形計画法では, 推薬消費量が30%減少し, 計算時間が33倍になった. このように, 減算法および加算法では, 計算量の増加を抑えつつ, 無駄噴きを減らしている.

これらより, 提案するファジィロジックによって, 実装性を損わずに効率的なスラスト選択則の実現が可能であると考えられる. また, 推薬消費量と計算時間のトレードオフは, 減算法または加算法の選択および, 減算法におけるファジィパラメータ α によって容易に実施可能である.

参考文献

- 1) 安田国治, 横田清美: 宇宙分野でのファジィ技術適用に関する一考察, 第11回誘導制御シンポジウム, 東京, 1994, pp. 43-50.
- 2) 廣田 薫編: ファジィ活用事例集, 工業調査会, 東京, 1991, pp. 57-195.
- 3) 廣田 薫: ファジィ制御の最近の動向, システム/制御/情報学会誌, 38 (1994), pp. 585-589.
- 4) 河野 功, 鈴木崇弘: Fuzzy 制御の宇宙機制御への応用の研究, 第6回宇宙航空の誘導制御シンポジウム, 東京, 1989, pp. 67-70.
- 5) 石本真二, 加藤寛一郎: ランデブの軌道制御における予見ファジィ制御と最適制御の比較, 日本航空宇宙学会誌, 39 (1991), pp. 44-50.
- 6) 吉桑英二: むだ時間を考慮したファジィ PD 制御系の安定解析に関する一考察, 第35回宇宙科学技術連合講演会, 長岡, 1991, pp. 509-510.
- 7) Lea, R. and Jani, Y.: Fuzzy Logic in Autonomous Orbital Operations, J. Approximate Reasoning, 6 (1992), pp. 151-184.
- 8) 鈴木 孝, 安田国治, 吉河章二, 山田克彦, 吉田憲正: 宇宙機の誘導制御技術へのファジィアルゴリズム適用の検討, 第37回宇宙科学技術連合講演会, 北九州, 1993, pp. 21-22.
- 9) 鈴木 孝, 安田国治, 峯 正弥, 山口慶剛, 吉沢 剛, 玉川浩一郎: 宇宙機マニピュレータシステムへのファジィ技術適用検討, 第38回宇宙科学技術連合講演会, 東京, 1994, pp. 211-212.
- 10) Cook, G.: Thruster Select Logic Determination for Attitude Control of Skylab by Teleoperator Retrieval System, AAS Paper, 79-003, 1979.
- 11) Bergmann, E. and Weiler, P.: Accommodation of Practical Constraints by a Linear Programming Jet Select, AIAA Paper, 83-2209, 1983.
- 12) 茨木俊秀, 福島雅夫: FORTRAN77 最適化プログラミング, 岩波書店, 東京, 1991, pp. 13-58.
- 13) 寺野寿郎, 浅居喜代治, 菅野道夫: ファジィシステム入門, オーム社, 東京, 1987, pp. 27-29.
- 14) 室津義定: 宇宙航行力学, 共立出版, 東京, 1993, pp. 144-146.