Differentially Flat な非線形システムの 繰り返し軌道計画による安定化*

南口 雄一[†]・渡辺 亮[‡]

Stabilization of Differentially Flat Nonlinear Systems by Repeating Trajectory Planning*

Yuichi MINAMIGUCHI[†] and Ryo WATANABE[‡]

Trajectory planning is well-known control technique for differentially flat systems. However this technique requires an accurate state space model of the system, therefore it is difficult to apply this technique directly for the control of actual uncertain systems. In this paper, we propose a new control technique based on the trajectory planning for differentially flat systems with input uncertainty in which the trajectory planning is repeated to reduce the influence of the uncertainty. We verify the performance of the proposed control technique by the numerical case studies for a two wheel vehicle model with input uncertainty, and then discuss the stability and the performance of the control system from the theoretical point of view.

1. はじめに

非線形制御の分野において、多くの有用な結果をもたらした幾何学的な手法の一つに厳密な線形化がある。厳密な線形化とは、非線形システムを座標変換と静的/動的フィードバックを用いて線形システムに変換するもので、変換後の線形システムに従来の線形制御理論を適用することで、様々な設計仕様に応えることが可能となる。

近年、この手法の一般化である Differentially Flat な非線形システムに対する軌道計画アプローチが注目を集めている [2]。Differentially Flat な非線形システムとは、Flat outputs とよばれる特別な出力とその導関数によってシステムの状態と入力を表現することができるシステムである。Differentially Flat な非線形システムでは、Flat outputs の軌道を計画することがシステムの入力と状態の軌道を同時に計画することに相当し、システムの状態方程式から導出される代数方程式を解くことで、Flat outputs の軌道を容易に計画することができる [4]。

一般に多くの機械システムが Differentially Flat な非線形システムであることが知られており、飛行機、車両系などへの応用例が多数報告されている [3,7,8]。

しかし、Differentially Flat な非線形システムに対して軌道計画をおこなうには、対象とするシステムの正確な数式モデルが必要であり、不確かさが存在する実システムに直接適用することは困難である場合が多い。不確かさが存在する場合、軌道を計画しても状態が計画した軌道どおりには必ずしも遷移しない。

近年、有限区間の最適軌道計画を繰り返すことで不確かさの影響による目標軌道からの逸脱を修正する、モデル予測制御 (MPC) が注目されている [12,14]。本論文では軌道計画の繰り返しという MPC の基本的な考え方に注目、対象の Differentially Flat な特性に基づいた軌道計画をサンプリング時間ごとに繰り返すことによって不確かさの影響を低減する手法を提案する。また、提案手法の有効性を数値例を用いて検証する。

さらに、本論文では提案手法の有効性を理論面から議論する。対象とするシステムには不確かさが存在するため、不確かさが軌道計画に与える影響を解析し、制御系のロバスト性を評価する必要がある。一方、提案手法は軌道計画を繰り返すことから、これによって制御系の状態がどのように遷移していくかについても議論する必要がある。本論文では後者の問題について、軌道計画の繰り返しによって制御系がフィードバック制御系となるこ

* 原稿受付 2005 年 11 月 29 日

[†] 早稲田大学 大学院 理工学研究科 Graduate School of Science and Engineering, Waseda University; Okubo 3-4-1, Shinjuku, Tokyo 169-8555, JAPAN

[‡] 早稲田大学 理工学部 School of Science and Engineering, Waseda University; Okubo 3-4-1, Shinjuku, Tokyo 169-8555, JAPAN

Key Words: trajectory planning, differentially flat, nonlinear system, stability.

とに注目, 提案手法を適用した際に得られる制御系が, サンプル点上において離散時間線形時不変システムとして表現されることを示し, この結果に基づいて閉ループ系の安定性と制御性能を解析することを試みる. 非線形システムにMPCを適用した制御系の安定性と制御性能については, 個別の議論[10,11]がなされているものの, 現時点では一般的な議論の展開は困難であると認識されている. 本論文における解析はこの問題に対する一つのアプローチである. なお, 不確かさが存在する制御系のロバスト性についても理論的な面からの考察を与える.

本論文の構成は次の通りである. 2章では, Differentially Flat な非線形システムの概念について述べ, Differentially Flat な非線形システムに対する軌道計画手法とその問題点について述べる. 3章では, 繰り返し軌道計画手法[3,5,9]の考え方を述べ, 数値例を用いてその有効性を示す. 4章では, 提案手法を用いた場合における状態の遷移をフィードバック制御の観点から解析し, その有効性を理論面から検証する. 最後に5章でまとめる.

2. Differentially Flat な非線形システム

2.1 システムの無限次元化と等価性

Differentially Flat な非線形システムの定義方法には1) 微分代数での枠組みで定義する方法, 2) 微分幾何学の枠組みで微分形式で定義する方法, 3) 無次元のベクトル場を用いて定義する方法, の3通りがある[4]. 本論文では, 無限次元ベクトル場を用いた定義方法を採用するが, それを述べる準備として, 本節で無限次元システムの等価性についてまとめる.

入力を u , 状態を x とする次の非線形システムを考える.

$$\dot{x} = f(x, u), \quad x \in \mathbb{R}^n, u \in \mathbb{R}^m \quad (1)$$

ここで入力 u が十分滑らかであるとする, 上記のシステムの状態軌道は, 同じ入力と初期状態 $x(0)$ に対する, 次式で与えられる無限次元システムの x の軌道と等しい.

$$\frac{d}{dt} \begin{bmatrix} x \\ u \\ \dot{u} \\ \vdots \end{bmatrix} = \begin{bmatrix} f(x, u) \\ \dot{u} \\ \ddot{u} \\ \vdots \end{bmatrix} \quad (2)$$

上式に対し $\epsilon, F(\epsilon)$ を

$$\begin{aligned} \epsilon &= [x^T \ u^T \ \dot{u}^T \ \dots]^T \\ F(\epsilon) &= [f^T(x, u) \ \dot{u}^T \ \ddot{u}^T \ \dots]^T \end{aligned} \quad (3)$$

で定義すると, (2) 式は次のように表現される.

$$\dot{\epsilon} = F(\epsilon) \quad (4)$$

つぎに, 異なる二つの無限次元システム間の等価性を

定義する. 次式で与えられる無限次元システムを考える.

$$\dot{\eta} = G(\eta) \quad (5)$$

ここで $\eta, G(\eta)$ は次で与えられるものとする.

$$\begin{aligned} \eta &= [y^T \ w^T \ \dot{w}^T \ \dots]^T \\ G(\eta) &= [g^T(y, w) \ \dot{w}^T \ \ddot{w}^T \ \dots]^T \end{aligned} \quad (6)$$

(4), (5) 式で与えられる二つの無限次元のシステムは, 以下が成り立つときに等価であると定義する.

【定義 1】 (4) 式と (5) 式で与えられる二つのシステムは座標変換 $\eta = \Phi(\epsilon)$ および, その逆変換 $\epsilon = \Psi(\eta)$ が存在するとき等価であるという.

ここで $\Phi(\epsilon), \Psi(\eta)$ は Φ, Ψ の各要素が ϵ, η のうちの有限個の要素を引数とするものとする.

定義より, 次元の異なる二つの非線形システムからそれぞれ作られる無限次元システムが互いに等価となる場合がある. また変換 Φ, Ψ が可逆ならば, 写像 $\epsilon \mapsto \Psi(\eta)$, $\eta \mapsto \Phi(\epsilon)$ は微分同相写像となり, ϵ, η の状態軌道が 1 対 1 に対応する [6].

2.2 Differentially Flat な非線形システムの定義

Differentially Flat な非線形システムの定義を次で与える [4].

【定義 2】 システム (1) 式より与えられる無限次元システム (4) 式が次式の線形システムと等価であるとき, システム (1) 式は Differentially Flat な非線形システムであるという.

$$\frac{d}{dt} \begin{bmatrix} z \\ \dot{z} \\ \ddot{z} \\ \vdots \end{bmatrix} = \begin{bmatrix} \dot{z} \\ \ddot{z} \\ \dddot{z} \\ \vdots \end{bmatrix} \quad (7)$$

このときの z を Flat outputs とよぶ.

等価性の定義より, (1) 式が Differentially Flat な非線形システムであるとき, (4) 式の状態 ϵ と (7) 式の状態 η が座標変換 $\eta = \Phi(\epsilon), \epsilon = \Psi(\eta)$ で関係付けられる. 以上より Differentially Flat な非線形システムにおいて, 以下の関係が成り立つ [4].

【定理 1】 システム (1) 式が状態 x と入力 u に対して

$$x = w_1(z, \dot{z}, \dots, z^{(l)}) \quad (8)$$

$$u = w_2(z, \dot{z}, \dots, z^{(l)}) \quad (9)$$

を満たす

$$z = \zeta(x, u, \dot{u}, \dots, u^{(l)}), \quad z \in \mathbb{R}^m \quad (10)$$

が存在するとき, システム (1) 式は Differentially Flat な非線形システムである. ここで, $z^{(l)}$ は z の l 階微分を表すものとする. また, (10) 式の z は Flat outputs となる.

2.3 軌道計画手法

システムをある状態から目的の状態に移させるためには、初期値と目標値を結ぶ状態軌道と、それを実現する入力を求める必要があるが、一般の非線形システム (1) 式では状態 $x(t)$ の軌道を計画しても、それを実現する入力を得ることは容易ではない。一方、Differentially Flat な非線形システムのダイナミクスは Flat outputs $z(t)$ の軌跡に集約されており、 $z(t)$ の軌跡を計画することがそのままシステムの状態 $x(t)$ の軌跡と、それを実現する入力 $u(t)$ を同時に計画することに相当する [4]。

ここでシステムの Flat outputs の軌道を基底関数 $\phi_j(t) (j=1, \dots, N)$ の線形結合で表すことを考える。

$$z_i(t) = \zeta_i(x(t), u(t), \dot{u}(t), \dots, u(t)^{(l)}) = A_{ij} \phi_j(t) \quad (11)$$

A_{ij} は対応する基底関数の係数である。ここで、基底関数の数 N と係数 A_{ij} を設定すると、与えられた条件を満たす z の軌道を定めることができる。

与えられた初期状態と終端状態を実現する軌道の作り方の一例を示す。システムの時刻 τ_0 における初期状態から決定される Flat outputs を z_0 、時刻 τ_f における終端状態から決定される Flat outputs を z_f とする。ここで、以下の連立方程式を解く。

$$\begin{cases} z_i(\tau_0) = A_{ij} \phi_j(\tau_0) & z_i(\tau_f) = A_{ij} \phi_j(\tau_f) \\ \vdots & \vdots \\ z_i^{(p)}(\tau_0) = A_{ij} \phi_j^{(p)}(\tau_0) & z_i^{(q)}(\tau_f) = A_{ij} \phi_j^{(q)}(\tau_f) \end{cases} \quad (12)$$

ここで、 p と q は、初期時刻 τ_0 と終端時刻 τ_f で指定するシステムの状態 x と入力 u に対する条件に基づいて、(8) 式と (9) 式から定まる定数である。

次の記号を導入する。

$$\bar{z}_0 = [z_i(\tau_0) \dots z_i^{(p)}(\tau_0)]^T \quad (13)$$

$$\bar{z}_f = [z_i(\tau_f) \dots z_i^{(q)}(\tau_f)]^T \quad (14)$$

$$\bar{z}_i = [\bar{z}_0^T \ \bar{z}_f^T]^T \quad (15)$$

また $\Phi(t)$ を $((p+q+2) \times N)$ 行列、 $\Phi_{ij}(t) = \phi_j^{(i-1)}$ とすると (12) 式より次式が成立する。

$$\bar{z}_i = \begin{bmatrix} \Phi(\tau_0) \\ \Phi(\tau_f) \end{bmatrix} A_i = \Phi A_i \quad (16)$$

$$\bar{z}_i : ((p+q+2) \times 1), \Phi : ((p+q+2) \times N), A_i : (N \times 1)$$

Φ がフルランクであれば、(16) 式より係数 A_{ij} が一意に定まり、与えられた初期状態と終端状態を結ぶ Flat outputs の軌道を定めることができる。

2.4 2 輪車両モデルを用いた例

Differentially Flat な非線形システムの典型的な例である 2 輪車両モデルを用いて軌道計画の具体的な例を示す [1,2]。2 輪車両モデルは Fig. 1 のように一般化座標 x, y とし、車体の姿勢を θ とする。

Fig. 1 Two wheel vehicle model

ここで車軸方向の速度 $v = u_1$ と操舵角速度 $\dot{\theta} = u_2$ をシステムの制御入力とすると、次式のモデルを得る。

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (17)$$

(17) 式の 2 輪車両モデルに対して、 $z_1 = x$, $z_2 = y$ とすると各状態変数、入力に対して次式が成立する。

$$\begin{aligned} x = z_1, y = z_2, \theta = \arctan \frac{\dot{z}_2}{\dot{z}_1}, u_1 &= \sqrt{\dot{z}_1^2 + \dot{z}_2^2} \\ u_2 &= \frac{d}{dt} \left(\arctan \frac{\dot{z}_2}{\dot{z}_1} \right) = \frac{\ddot{z}_2 \dot{z}_1 - \dot{z}_1 \ddot{z}_2}{\dot{z}_1^2 + \dot{z}_2^2} \end{aligned} \quad (18)$$

よって z_1, z_2 が (8), (9) 式を満たすことから、2 輪車両モデルは Differentially Flat な非線形システムであり、 z_1, z_2 が Flat outputs となる。

Differentially Flat なシステム (17) 式に対して z の軌道が満たすべき条件を次で与えたとする。

$$\begin{aligned} (z_1(0), \dot{z}_1(0), \ddot{z}_1(0), z_1(5), \dot{z}_1(5), z_1(10), \dot{z}_1(10)) \\ = (0, 0.5, 0, 0, 0.25, 0, 0) \\ (z_2(0), \dot{z}_2(0), \ddot{z}_2(0), z_2(5), \dot{z}_2(5), z_2(10), \dot{z}_2(10)) \\ = (1, 0, 0, 0.5, 0, 0, 0) \end{aligned} \quad (19)$$

これらの条件は、(17) 式のシステムに対し、時刻 $t=0$ において $x(0), y(0), \theta(0), u_1(0), u_2(0)$ を、時刻 $t=5$ において $x(5), y(5), \theta(5), u_1(5)$ を、時刻 $t=10$ において $x(10), y(10), \theta(10), u_1(10)$ を指定していることを意味する。(19) 式は、それぞれの z について七つの条件を与えていることから、 z の軌道を次式の 6 次多項式で表すことにする。

$$\begin{aligned} z_1(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 + a_6 t^6 \\ z_2(t) &= b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5 + b_6 t^6 \end{aligned} \quad (20)$$

(19) 式の条件を満たすように (20) 式の係数を定めることで、Flat outputs の軌道 $z_1(t), z_2(t)$ が定まる。そして (18) 式から求まる u_1, u_2 をシステム (17) 式に入力する。なお、 $z_1(t)$ と $z_2(t)$ の軌道は独立に計画されることに注意する。

このときのシミュレーション結果を Fig. 2 に示す。Fig. 2 では上段は左から順に x, y, θ の応答、下段は左から順に入力の u_1, u_2 の応答、 xy に平面における軌道を表

Fig. 2 Simulation without uncertainty

し、また図中の黒丸は目標値を表している。\$\theta\$ に関しては目標値は設定していない。

Fig. 2 から状態は正確に設定した目標値に到達していることがわかる。

2.5 軌道計画の問題点

前節の2輪車両モデルにおいて、制御入力として \$v=u_1\$, \$\dot{\theta}=u_2\$ がシステムに入力できると仮定した。しかし実際の機械システムでは、入力できるのが速度ではなくトルクである場合が多い。トルクが制御入力である場合の対処法として、Fig. 3 左のように速度サーボ系を構成し、全体の制御系に付加することが考えられるが、この場合、サーボ系に過渡特性が存在するため Fig. 3 右のように、実際にシステムに入力される速度 \$v_{in}\$ と速度指令値 \$v_{ref}\$ に差が生じる。

Fig. 3 Input uncertainty

前節で述べた2輪車両の例で、実際にシステムに入力できるのがトルクである場合、速度サーボ系の付加が制御性能に及ぼす影響についてシミュレーションで検証する。

シミュレーションで用いる2輪車両のアクチュエータを次のように仮定する。2輪車両の両タイヤに、それぞれ独立に制御できるモータが取り付けられており、モータのトルク \$\tau_1, \tau_2\$ が制御入力であるとする。また、車軸からタイヤまでの距離を \$R\$、左右のタイヤ半径をそれぞれ \$r_1, r_2\$、角速度を \$\omega_1, \omega_2\$、慣性モーメントを \$J_1, J_2\$、粘性摩擦係数を \$c_1, c_2\$ とする。このとき、角速度 \$\omega_1, \omega_2\$ のダイナミクスは以下で与えられる。

$$\begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \end{bmatrix} = \begin{bmatrix} -c_1/J_1 & 0 \\ 0 & -c_2/J_2 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} + \begin{bmatrix} -1/J_1 & 0 \\ 0 & -1/J_2 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (21)$$

また速度 \$u_1\$、操舵角速度 \$u_2\$ とそれぞれのタイヤの角速度 \$\omega_1, \omega_2\$ について、次の関係が成り立つ。

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} r_1/2 & r_2/2 \\ r_1/2R & -r_2/2R \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = T \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} \quad (22)$$

速度サーボ系はPIコントローラ \$K(s)=K_p+K_i/s\$ を用いて構成する。全体の制御系を Fig. 4 に示す。

Fig. 4 Block diagram with servo

Fig. 4 において、(19), (20) 式の条件から求まる \$u_1, u_2\$ を速度指令値 \$u_{1ref}, u_{2ref}\$ としてシステムに入力したときのシミュレーション結果を Fig. 5 に示す。ただし、アクチュエータのパラメータは \$r_1=r_2=0.1\$, \$R_1=R_2=0.3\$, \$c_1=c_2=0.5\$, \$J_1=J_2=1.5\$、コントローラのパラメータは \$P_1=P_2=2\$, \$I_1=I_2=1\$ とした。なお、図の配置は Fig. 2 と同じである。

Fig. 5 Simulation with uncertainty

Fig. 5 から、Fig. 3 右に示されるように、実際にシステムに入力される速度が生成した軌道を実現するための入力と異なる場合、状態を目標値まで到達させることができないことがわかる。

3. 繰り返し軌道計画

3.1 繰り返し軌道計画アルゴリズム

2.5 節の例のように速度サーボ系を付加した場合は、Fig. 6 に示すように、入力に不確かさが存在する系とみなすことができる。このような場合、計画した Flat outputs の軌道を実現する入力 u_{ref} を正確にシステムに加えることができないため、2.5 節で示したように設定した目標値に状態を到達させることができない。そこで本節では、不確かさが存在しても状態を目標値に到達させることを目的に、軌道計画を繰り返すことを考える [3,5,9]。

Fig. 6 Block diagram of system with input uncertainty

はじめに、基本的なアイデアを述べる。いま、時刻 $t=0$ における初期値から与えられた目標値まで時間 T で到達するように、Flat outputs の軌道を計画したとする。入力に不確かさが存在する場合、システムの状態は計画した軌道からずれていくが、十分短い時刻 $t=\Delta t$ においては、計画した軌道からの状態のずれは小さいと考えられる。そこで、 Δt ごとに現在の状態と目標値を結ぶ軌道を繰り返し計画すれば、状態がそれぞれ計画した軌道からある程度離れることがあっても、設定した目標値に状態を近づけることができると考えられる。

上記のアイデアに基づく繰り返し軌道計画の方法を Fig. 7 を用いて説明する。まず Flat outputs の初期値 $z_0 = z(0)$ と目標値 z_l に対して、時刻 $t=T$ で目標値に到達するよう、 z_0 と z_l を結ぶ Flat outputs の軌道 z_{1p} を計画する。入力に不確かさが存在する場合、 $t=\Delta t$ において状態は計画した軌道上の点 z_{1p} からずれた点 z_{1a} に存在する。そこで、 $t=\Delta t$ における現在値 z_{1a} から目標値 z_l に時刻 $t=\Delta t+T$ で到達するような軌道 z_{2p} を新たに計画する。さらに $t=2\Delta t$ における状態は z_{2p} から少しずれた点 z_{2a} に存在するため、新たに z_{2a} から目標値 z_l に時刻 $t=2\Delta t+T$ で到達する軌道を計画する。この処理を時間 Δt ごとに繰り返すことで目標値に状態を近づけていく。

ここで、目標値に到達するまでの時間 T は軌道計画の繰り返しに応じた時刻の経過に従って減少させるのが自然である。しかし現在時刻 $t=i\Delta t$ に対して、到着するまでの時間を $T-i\Delta t$ として軌道計画を繰り返すと、 $T-i\Delta t$ が小さくかつ目標値との距離が大きい場合に急激な入力が発生するため、到達するまでの時間は常に T として軌道計画を行う。

繰り返し軌道計画の制御系のブロック線図を Fig. 8 に示す。

Fig. 7 Trajectory planning algorithm

Fig. 8 Block diagram of trajectory planning algorithm

Fig. 8 において、現在値は Δt 時間ごとにサンプルされ、軌道計画ブロックにフィードバックされる。また、Fig. 6 の制御系がフィードフォワード制御系であるのに対し、Fig. 8 の制御系はフィードバック制御系となっていることに注意する。

一般に、繰り返し軌道計画を適用することによって期待されるフィードバック制御系としての効果は、サンプリング時間に依存する [13]。本手法の軌道計画でオンラインで解くべき計算は簡単な行列計算だけであることを鑑みると、十分短いサンプリングが必要とされる場合においても広く対応が可能であると考えられる。

3.2 繰り返し軌道計画の具体例

2.5 節で述べたアクチュエータのダイナミクスを考慮した 2 輪車両モデルを対象に、繰り返し軌道計画の有効性を検証する。なお、Flat outputs z_1 と z_2 の軌道計画は同じ手順で行うため、本節では z_1 の軌道計画についてのみ記述し、また記号の簡略化のために $z_1(t)$ を $z(t)$ と表記する。

i 回目の軌道計画における Flat output $z(t)$ の軌道を次の 4 次多項式で与える。

$$z(t) = a_0 + a_1(t-i\Delta t) + a_2(t-i\Delta t)^2 + a_3(t-i\Delta t)^3 + a_4(t-i\Delta t)^4 \quad (23)$$

$a_i (i=0, \dots, 4)$ は軌道 $z(t)$ を定める係数である。

係数 $a_i (i=0, \dots, 4)$ は i 回目の軌道計画において、現在値、目標値に関する次の五つの条件を満たすように決

定する.

- 現在値: $\bar{z}_0 = [z(i\Delta t) \dot{z}(i\Delta t) \ddot{z}(i\Delta t)]^T$
- 目標値: $\bar{z}_T = [z_T \dot{z}_T]^T$

ここで z_T は時刻 $t = i\Delta t + T$ における $z(t)$ の目標値を表す. また, 目標値までの到達時間は T とする.

このとき, 繰り返し軌道計画は以下の手順によって行われる.

Step 1 現在の状態から Flat outputs の始点, 終点を与える.

Step 2 次式を Λ について解く.

$$\begin{bmatrix} \bar{z}_0 \\ \bar{z}_T \end{bmatrix} = \begin{bmatrix} \Phi(i\Delta t) \\ \Phi(i\Delta t + T) \end{bmatrix} \Lambda \quad (24)$$

Step 3 Λ から Flat outputs の軌道, (23) 式を計画する.

Step 4 計画された Flat outputs の軌道を用いて (18) 式から入力 u_1, u_2 を導出し, Δt の間システムに入力する.

Step 5 Step1 から Step4 の処理をサンプリング時間 Δt ごとに繰り返す.

2.5 節のアクチュエータのダイナミクスを考慮した2輪車両モデルに対し上記のアルゴリズムを適用した場合について, シミュレーションを行う. シミュレーションに用いたアクチュエータのパラメータを Table 1 に示す. ケース1ではサーボ系のパラメータを追従性能が高くなるように設定し, ケース2では低くなるように設定した. またケース1, 2ともに満たすべき Flat outputs の条件と到達時間, サンプリング時間は Table 2 に示す値を用いた. ケース1, ケース2のシミュレーション結果をそれぞれ Figs. 9, 10 に示す. なお, 図の配置は Fig. 2 と同じである.

Table 1 Actuator parameter

	r_1, r_2	R_1, R_2	c_1, c_2	J_1, J_2	P_1, P_2	I_1, I_2
case1	0.1	0.3	0.1	0.5	2	1
case2	0.1	0.3	0.5	1.5	2	1

Table 2 Condition on Flat outputs

T	Δt	$(z_1(0), \dot{z}_1(0), \ddot{z}_1(0))$	$(z_1T, \dot{z}_1T)$	$(z_2(0), \dot{z}_2(0), \ddot{z}_2(0))$	$(z_2T, \dot{z}_2T)$
5	1	(1, 0.5, 0)	(0, 0)	(1, 0, 0)	(0, 0)

Figs. 9, 10 から, 入力に不確かさが存在しても, 状態は設定した目標値に収束していることがわかる. また Fig. 10 ではサーボ系の性能が低いことから入力の不確かさが大きく, 計画した軌道から大きくずれるが, 目標値に収束している.

なお, 上記の数値例において Flat outputs の現在値 $\bar{z}_0$ を決定するためには, $z_1, \dot{z}_1, \ddot{z}_1, z_2, \dot{z}_2, \ddot{z}_2$ の現在値が必要である. 2輪車両モデルの場合, $z_1 = x, z_2 = y$ であることから, $x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}$ が観測できれば, $\bar{z}_0$ を決定することができる. これらの物理量は, たとえば, $\ddot{x}$

Fig. 9 Simulation using condition1 parameter

Fig. 10 Simulation using condition2 parameter

と $\ddot{y}$ は車体に取り付けた加速度センサによって, $\dot{x}$ と $\dot{y}$ は車軸に取り付けたタコメータによって, x と y は外部に取り付けた画像センサによって観測することが可能となる.

4. 安定性と制御性能

4.1 安定性と制御性能の解析方法

前節では繰り返し軌道計画アルゴリズムの有効性を数値例を用いて検証したが, この方法によってシステムを与えられた目標状態に到達させることができるかについては保証されていない. 繰り返し軌道計画の有効性を理論的に検証するためには, Fig. 8 に示される制御系の安定性と制御性能を解析することが必要である.

Fig. 8 の制御系は入力に不確かさを有することから, それが Flat outputs の軌道計画に与える影響を調べることで, 不確かさに対するロバスト性を評価する必要がある. 一方, 3.1 節で述べたように, 提案した制御手法は軌道計画の繰り返しによって Fig. 8 に示されるフィードバック制御系となることから, 閉ループ系の状態がどのように遷移をするかについてもフィードバック制御の観点から議論する必要がある.

本章では Fig. 8 の制御系を解析するための第一歩として, Fig. 8 の制御系をフィードバック制御系ととらえ, 閉

ループ系の状態の遷移を解析することで、安定性と制御性能について考察する．そこで Fig. 8 の制御系において、不確かさが存在しない場合について考察を進めることにする．

不確かさが存在しない場合、システムの Differentially Flat な特性から、計画した軌道と出力される軌道が一致し、Fig. 8 において $z_{in} = z_{out}$ が成り立つ．このとき、Fig. 8 のシステムは Fig. 11 と等価になる．

Fig. 11 Block diagram without uncertainty

Fig. 11 から、システムに不確かさが存在しない場合、システムの状態を調べるには、サンプル時間ごとに繰り返し計画される Flat outputs の軌道に着目すればよいことがわかる．そこで Fig. 11 のシステムに対して、Flat outputs $z(t)$ を状態変数とする状態方程式を導出し、その安定性と定常特性を解析することで、繰り返し軌道計画を用いたときの制御性能を考察する．

4.2 状態方程式の導出と制御性能の解析

本節では繰り返し軌道計画を用いたときの制御系の状態方程式を導出する．繰り返し軌道計画ではすべての Flat outputs は同じ手順で生成されるため、本節で Flat outputs z の一つ、 z_j の挙動に着目し議論をおこなう．また、3.2 節と同様に記号の簡略化のため、 $z_j(t)$ を $z(t)$ と表記する．

計画される軌道は $z(t)$ は基底関数 $\Phi(t)$ と係数 Λ を用いて次で与えられているものとする．

$$z(t) = \Phi(t)\Lambda \quad (25)$$

係数 Λ は、現在値と目標値をそれぞれ時刻 $i\Delta t$, $i\Delta t + T$ で満たすように設定されるものとする．ここで時間軸を $i\Delta t$ ずらして考えると、 $z(t)$ は時刻 0 で現在値 $z(i\Delta t)$ を満たし、時刻 T で目標値 $z(i\Delta t + T)$ を満たすように決定される．これより、係数 Λ について次の関係が成り立つ．

$$\begin{bmatrix} z(i\Delta t) \\ \vdots \\ z^{(p)}(i\Delta t) \\ z(i\Delta t + T) \\ \vdots \\ z^{(q)}(i\Delta t + T) \end{bmatrix} = \begin{bmatrix} \Phi(0) \\ \vdots \\ \Phi^{(p)}(0) \\ \Phi(T) \\ \vdots \\ \Phi^{(q)}(T) \end{bmatrix} \Lambda \quad (26)$$

(26) 式を Λ について解き、時間軸を $i\Delta t$ ずらしていることを考慮すると、(25) 式より次式を得る．

$$\begin{bmatrix} z(t) \\ \vdots \\ z^{(p)}(t) \end{bmatrix} = \begin{bmatrix} \Phi(t - i\Delta t) \\ \vdots \\ \Phi^{(p)}(t - i\Delta t) \end{bmatrix} \begin{bmatrix} \Phi(0) \\ \vdots \\ \Phi^{(p)}(0) \\ \Phi(T) \\ \vdots \\ \Phi^{(q)}(T) \end{bmatrix}^{-1} \begin{bmatrix} z(i\Delta t) \\ \vdots \\ z^{(p)}(i\Delta t) \\ z(i\Delta t + T) \\ \vdots \\ z^{(q)}(i\Delta t + T) \end{bmatrix} \quad (27)$$

このとき、(27) 式に $t = (i+1)\Delta t$ を代入すると、隣り合うサンプル点上、 $i\Delta t$, $(i+1)\Delta t$ 、における z の挙動は次式で与えられる．

$$\begin{bmatrix} z((i+1)\Delta t) \\ \vdots \\ z^{(p)}((i+1)\Delta t) \end{bmatrix} = \begin{bmatrix} \Phi(\Delta t) \\ \vdots \\ \Phi^{(p)}(\Delta t) \end{bmatrix} \begin{bmatrix} \Phi(0) \\ \vdots \\ \Phi^{(p)}(0) \\ \Phi(T) \\ \vdots \\ \Phi^{(q)}(T) \end{bmatrix}^{-1} \begin{bmatrix} z(i\Delta t) \\ \vdots \\ z^{(p)}(i\Delta t) \\ z(i\Delta t + T) \\ \vdots \\ z^{(q)}(i\Delta t + T) \end{bmatrix} \quad (28)$$

ここで次の記号を定義する．

$$\xi_j(i) = \begin{bmatrix} z(i\Delta t) \dots z^{(p)}(i\Delta t) \end{bmatrix}^T \quad (29)$$

$$r_j(i) = \begin{bmatrix} z(i\Delta t + T) \dots z^{(q)}(i\Delta t + T) \end{bmatrix}^T \quad (30)$$

このとき ξ_j に関する状態方程式は (28) 式より次式で与えられる．

$$\xi_j(i+1) = A\xi_j(i) + Br_j(i) \quad (31)$$

行列 A, B は Δt と T の行列関数であることに注意する．

他の Flat output に関する状態方程式も同様の手順で導出され、これより、Fig. 11 に示される制御系の状態方程式は次で与えられる．

$$\xi(i+1) = \bar{A}\xi(i) + \bar{B}r(i) \quad (32)$$

ただし

$$\xi(i) = \begin{bmatrix} \xi_1^T(i) \dots \xi_m^T(i) \end{bmatrix}^T \quad (33)$$

$$r(i) = \begin{bmatrix} r_1^T(i) \dots r_m^T(i) \end{bmatrix}^T \quad (34)$$

$$\bar{A} = \text{diag}(A, \dots, A) \quad (35)$$

$$\bar{B} = \text{diag}(B, \dots, B) \quad (36)$$

である．

(32)式で表される制御系の安定性と制御性能は、線形制御理論を用いることで検証できる。システム行列がブロック対角であることから、安定性は行列 A の固有値を求めることで評価できる。また、目標値 $r(i)$ が一定値 r_T の場合は、 A が安定行列ならば、定常特性は (32) 式より次式で与えられる。

$$\xi(\infty) = (I - \bar{A})^{-1} \bar{B} r_T \quad (37)$$

4.3 安定性と制御性能の検証例

例として 3.2 節の具体例について状態方程式を導出する。Flat outputs z_1 の挙動を z で表し、状態変数 ξ_1 と目標値 r_1 を次で定義する。

$$\xi_1 = \begin{bmatrix} z & \dot{z} & \ddot{z} \end{bmatrix}^T \quad (38)$$

$$r_1 = \begin{bmatrix} z_T & \dot{z}_T \end{bmatrix}^T = \begin{bmatrix} z_T & 0 \end{bmatrix}^T \quad (39)$$

このとき x_1 に関する状態方程式は次で与えられる。

$$\xi_1(i+1) = A\xi_1(i) + Br_1 \quad (40)$$

ここで、 A, B は次で与えられる。

$$A = \begin{bmatrix} \Gamma_1\Psi_1 & \Gamma_1\Psi_2 & \Gamma_1\Psi_3 \\ \Gamma_2\Psi_1 & \Gamma_2\Psi_2 & \Gamma_2\Psi_3 \\ \Gamma_3\Psi_1 & \Gamma_3\Psi_2 & \Gamma_3\Psi_3 \end{bmatrix} \quad (41)$$

$$B = \begin{bmatrix} \Gamma_1\Psi_4 & \Gamma_1\Psi_5 \\ \Gamma_2\Psi_4 & \Gamma_2\Psi_5 \\ \Gamma_3\Psi_4 & \Gamma_3\Psi_5 \end{bmatrix} \quad (42)$$

$$\Gamma_1(\Delta t) = \begin{bmatrix} 1 & \Delta t & \Delta t^2 & \Delta t^3 & \Delta t^4 \end{bmatrix} \quad (43)$$

$$\Gamma_2(\Delta t) = \begin{bmatrix} 0 & 1 & 2\Delta t & 3\Delta t^2 & 4\Delta t^3 \end{bmatrix} \quad (44)$$

$$\Gamma_3(\Delta t) = \begin{bmatrix} 0 & 0 & 2 & 6\Delta t & 12\Delta t^2 \end{bmatrix} \quad (45)$$

$$\begin{aligned} \Psi(T) &= \begin{bmatrix} \Psi_1(T) & \Psi_2(T) & \dots & \Psi_5(T) \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 \\ -4/T^3 & -3/T^2 & -1/T & 4/T^3 & -1/T^2 \\ 3/T^4 & 2/T^3 & 1/2T^2 & -3/T^4 & 1/T^3 \end{bmatrix} \end{aligned} \quad (46)$$

Flat output z_2 に関する状態方程式も同様の手順で導出される。この場合 z_2 の挙動は z_1 と等しくなるため、 ξ_2 を z_2 に関する状態変数、 r_2 を z_2 の目標値とすると、状態方程式は次で与えられる。

$$\begin{aligned} \begin{bmatrix} \xi_1(i+1) \\ \xi_2(i+1) \end{bmatrix} &= \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} \xi_1(i) \\ \xi_2(i) \end{bmatrix} \\ &+ \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \end{aligned} \quad (47)$$

パラメータを 3.2 節と同様に設定した場合の (47) 式の Flat outputs の応答を、3.2 節のシミュレーションにおける応答と比較した結果を Fig. 12 に示す。また z_1, z_2 平面の軌道を比較した結果を Fig. 13 に示す。Fig. 12 において実線は (47) 式の応答、点線は 3.2 節の軌道計画の応答を表している。

Fig. 12 Comparison of Flat outputs response

Fig. 13 Comparison of Flat outputs trajectory

Figs. 12, 13 から両者の応答が一致していることが確認できる。

つぎに、システムの安定性について検証する。例として $\Delta t = 1$ の条件のもとで、 T を 5 から 50 まで変化した場合の A の根軌跡を Fig. 14 に示す。

Fig. 14 Root locus

Fig. 14 から, 上記の条件において A の固有値はすべて原点中心の単位円内に収まっており, システムは安定であることがわかる. また, $T=10, \Delta t=1$ とした場合, (37) 式より $\xi_1(\infty)$ は次で与えられる.

$$\xi_1(\infty) = \begin{bmatrix} z(\infty) \\ \dot{z}(\infty) \\ \ddot{z}(\infty) \end{bmatrix} = \begin{bmatrix} 1 & * \\ 0 & * \\ 0 & * \end{bmatrix} \begin{bmatrix} z_T \\ 0 \end{bmatrix} \quad (48)$$

上式より $[z_1(t) \ \dot{z}_1(t)]^T$ は定めた目標値 r_1 に収束することがわかる. 同様の計算により $[z_2(t) \ \dot{z}_2(t)]^T$ が目標値 r_2 に収束することが確かめられる.

4.4 ロバスト性に関する考察

本節では, 制御系のロバスト性を, 観測に不確かさを持つ 2 輪車両モデルを対象に考察する.

2 輪車両モデルの場合, 制御系のブロック図は Fig. 15 で与えられる.

Fig. 15 Control system

ここで, Fig. 15 において $\delta(i\Delta t) = \text{diag}[\delta_1(i\Delta t), \delta_2(i\Delta t)]$ は $\xi(i\Delta t)$ の観測における不確かさ, $\tilde{\xi}(i\Delta t)$ は $\xi(t)$ の観測値, $\xi_r(t)$ は $\tilde{\xi}(i\Delta t)$ を用いて計画される $\xi(t)$ の軌道を表す.

$\xi_r(t)$, $t \in [i\Delta t, (i+1)\Delta t]$ は, 不確かさの影響を受けた観測値 $\tilde{\xi}(i\Delta t)$ を用いて計画される. このことから, $\xi_{rj}((i+1)\Delta t)$ は次で与えられる.

$$\xi_{rj}(i+1) = A_j \tilde{\xi}_j(i) + B_j r_j(i) \quad (49)$$

一方, $\xi_r(t)$ は $\tilde{\xi}(i\Delta t)$ を用いて計画されていることから, $t = i\Delta t$ において次が成立する.

$$\xi_{rj}(i\Delta t) = \tilde{\xi}_j(i\Delta t) = \xi_j(i\Delta t) + \delta_j(i\Delta t) \xi_j(i\Delta t) \quad (50)$$

このとき, 2 輪車両 (17) 式の動特性が

$$[\dot{x} \ \dot{y}]^T = f(\theta) u_1 \quad (51)$$

で与えられることから, $t = (i+1)\Delta t$ において次が成立する.

$$\xi_{rj}((i+1)\Delta t) = \xi_j((i+1)\Delta t) + \delta_j(i\Delta t) \xi_j(i\Delta t) \quad (52)$$

(52) 式, (49) 式, および $\tilde{\xi}_j(i\Delta t)$ が

$$\tilde{\xi}_j(i\Delta t) = \xi_j(i\Delta t) + \delta_j(i\Delta t) \xi_j(i\Delta t) \quad (53)$$

で与えられることから, 次式が成り立つ.

$$\begin{aligned} \xi_j(i+1) + \delta_j(i) \xi_j(i) \\ = A_j [\xi_j(i) + \delta_j(i) \xi_j(i)] + B_j r_j(i) \end{aligned} \quad (54)$$

上式を整理すると, $\xi_j(i)$ に関する差分方程式として次を与える.

$$\xi_j(i+1) = [A_j + (A_j - I) \delta_j(i)] \xi_j(i) + B_j r_j(i) \quad (55)$$

この差分方程式は, システム行列に時変なパラメータ変動 $\delta_j(i\Delta t)$ を持つ離散時間線形システムを表していることから, 線形ロバスト制御理論によってその解析が可能となる.

5. おわりに

本論文では入力に不確かさを有する Differentially Flat な非線形システムに対して, 状態を与えられた目標値に収束させることを目的とする繰り返し軌道計画手法を提案し, シミュレーションによりその有効性を確認した. また提案した手法を用いた場合, 制御系がフィードバック制御系となることに着目し, 不確かさが存在しない場合について, 離散時間システムとして閉ループ系の状態方程式を導出した. 導出した状態方程式に基づき制御系の安定性と制御性能について考察し, 提案した手法の有効性を確認した. そして 2 輪車両モデルを対象に, 制御系のロバスト性解析について議論し, この例においては, システム行列にパラメータ変動を持つ離散時間線形システムのロバスト性解析問題に帰着されることを示した.

今後の課題としては, より一般的なクラスの Differentially Flat な非線形システムに対して, 不確かさが存在する場合についての議論があげられる.

参考文献

- [1] 美多：非線形制御入門 - 劣駆動ロボットの技能制御論-, 昭晃堂 (2000)
- [2] E. Yang and T. Mita: Flatness-based trajectory planning and control of a car-like mobile robot; 第 3 回制御部門大会予稿集, pp. 607–610 (2003)
- [3] M. J. van Nieuwstadt and R. M. Murray: Real time trajectory generation for differentially flat systems; *International Journal of Robust and Nonlinear Control*, Vol. 8, No. 11, pp. 995–1020 (1998)
- [4] 藤本, 杉江：厳密な線形化からフラットネスへ; システム/制御/情報, Vol. 43, No. 2, pp. 87–93 (1999)
- [5] 村田：不確かさを持つモデルを用いた船舶の軌道計画アルゴリズム-Differential flatness 特性に基づくアプローチ-, 早稲田大学大学院理工学研究科修士論文 (2005)
- [6] M. Fliess, J. Lévine, P. Martin and P. Rouchon: A Lie-Bäcklund approach to equivalence and flatness of nonlinear systems; *IEEE Transactions on Automatic Control*, Vol. 44, No. 5, pp. 922–937 (1999)
- [7] Ph. Martin, R. M. Murray and P. Rouchon: Flat Systems, Equivalence and Trajectory Generation, CDS2003-008, California Institute and Technology

- Technical Report (2003)
- [8] M. J. van Nieuwstadt and R. M. Murray: Approximate trajectory generation for differentially flat systems with zero dynamics; *Proc. IEEE Control and Decision Conf.*, Vol. 4, pp. 4224–4230 (1995)
- [9] M. B. Milam, K. Mushambi and R. M. Murray: A new computation approach to real-time trajectory generation for constrained mechanical systems; *Proc. of 34th IEEE Conference on Decision and Control*, pp. 845–851 (2000)
- [10] A. Jadbabaie, J. Yu and J. Hauser: Unconstrained receding-horizon control of nonlinear systems; *IEEE Transactions on Automatic Control*, Vol. 46, No. 5, pp. 776–783 (2001)
- [11] L. Magni and R. Sepulchre: Stability margins of nonlinear receding-horizon control via inverse optimality; *Systems and Control Letters*, Vol. 32, pp. 241–245 (1997)
- [12] J. M. Maciejowski: モデル予測制御-制約のもとでの最適制御-, 東京電機大学出版局 (2005)
- [13] 小林, 孫野, 井前: 非線形最適制御問題の数値解法アルゴリズムに基づく実時間制御の検討; 第4回制御部門大会予稿集, pp. 347–350 (2004)
- [14] E. F. Camacho and C. Bordons: *Model Predictive Control*, Springer (1998)
-