

## 準最短時間機械運動探索のための 低計算オーダーによる位相空間上探索木枝刈り手法

金 天海<sup>†</sup> 辻野 広司<sup>†</sup>

ホンダ・リサーチインスティテュート・ジャパン

埼玉県和光市本町 8-1

{tenkai, tsujino}@jp.honda-ri.com

菅原 志門<sup>‡</sup> 菅野 重樹<sup>‡</sup>

早稲田大学 理工学術院

総合機械工学専攻 菅野重樹研究室

東京都新宿区大久保 3-4-1

{s\_sugawara, sugano}@sugano.mech.waseda.ac.jp

**Abstract** - 我々は準最短時間機械運動探索を高速化するための位相空間上探索木の枝刈り手法を提案する。位相空間の特性を生かした提案法は、メッシュを用いることで探索木上の類似する枝を定数オーダーで削除できるため、低計算コストであり解の精度も良い。二重倒立振り子モデルを用いて準最短時間運動探索シミュレーションを行った。提案法の結果は他の手法と比べて計算速度が 58 倍以上となり、目標状態到達誤差は半分以下であった。

### I. はじめに

機械全般に共通して使用可能な運動動作計画・実行システムを構成することは機械制御分野全体の課題と言える。本稿では、運動方程式をもとに指定された目標姿勢・速度への準最短時間動作を高速・高精度で計画・実行するシステムに関する汎用的な解法を取り扱う。なお運動方程式などの機械運動モデルの同定に関する問題は扱わず正確なモデルは既知とする。

従来、運動計画を行うための手法として A\*法やダイクストラ法をベースとした手法が用いられてきた。これらの手法は機械状態を機械姿勢の関数  $S:=S(q)$  として考え、姿勢空間上における距離に基づいて初期状態から目標状態間へと最短距離で移動するための遷移に関する計算を行うが、速度  $\dot{q}$  を考慮していないため、通常得られる解は最短時間軌道とは異なる。

我々は位置と速度の二つの変数からなる位相空間上で運動計画を行うための手法を扱う。

### II. 提案法

位相空間上において機械の状態  $S$  は  $S:=S(q, \dot{q})$  と定義でき、位相空間上における状態  $S_a$  と状態  $S_b$  の最短遷移時間を  $T^*(S_a, S_b)$  とする。また、状態  $S_a$  より状態  $S_b$  への最短時間軌道を  $P^*(S_a, S_b)$  とする。この際、 $P^*$  は複数存在する可能性があるが本稿ではそれらを区別しない。

この問題は、機械モデルの運動方程式に基づき、機械を初期状態  $S_0$  から目標状態  $S_g$  へと最短時間  $T^*(S_0, S_g)$  で遷移させるための制御入力系列  $I^*(S_0, S_g)=\{I(S_0, S_1), \dots, I(S_{g-1}, S_g)\}$  となるべく早く正確に見つけるという問題とも考えられる。

このような位相空間上で運動計画を行う際に [1] では 2 状態間における状態遷移を線形と仮定している。しかし、実際の位相空間上における状態遷移は非線形であるため、計画した運動を実行した際には誤差が生じてしまう。そこで、計画した運動を正確に実行するために状態探索木 [2] を用いる手法が提案されている。

本稿では位相空間における準最短時間軌道計画を状態探索木を用いて速く正確に行う方法を提案する。Fig.1 にフローを示し、以下に各手順の説明を示す。

- 1) 機械の初期状態  $S_0$  と目標状態  $S_g$  を取得する。
- 2) 機械の初期状態  $S_0$  を探索木の根として登録する。
- 3) 探索木に対し、幅優先探索により次状態を探索し、各状態から所定のトルク  $\{\tau_a, \tau_b, \tau_c, \dots\}$  を与え、一定時間  $\Delta t$  経過した後の次状態  $S_{found}$  を推定し、状態探索木へと登録する。
- 4) 3) で推定した次状態  $S_{found}$  を枝刈りする。
- 5) 探索木の深さが所定の値であれば終了する。

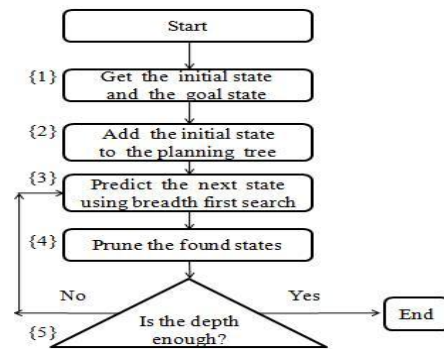


Fig.1 General flow to search paths

以上の手順に従って位相空間上の探索を行った後に準最短時間軌道を作成するには、探索で見つかった状態群の中から目標状態  $S_g$  へと十分に近い状態群  $S_{near}(S|dist(S, S_g) < Const.)$  を探す。さらに到達するまでの所要時間が最も短い  $S_{near}$  から初期状態  $S_0$  へ向けて状態探索木をたどることで準最短時間遷移系列  $P^*(S_0, S_g)$  を特定する。

この軌道探索において、見つかった状態群  $S_{all}$  の中に位相空間上のユークリッド距離に近い 2 状態  $S_p, S_q$  が含まれる場合、それらの 2 状態  $S_p, S_q$  から再び軌道探索を開始すると次状態  $S'_p, S'_q$  はほぼ同じ結果が得られる。

$$dist(S_p, S_q) \approx 0 \Rightarrow dist(S'_p, S'_q) \approx 0 \quad (1)$$

よってこれらの 2 状態  $S_p, S_q$  のうち 1 状態を削除することが可能であり、計算効率上望ましい。また、 $S_p$  と  $S_q$  の類似度が高い場合、この枝刈りによる解の精度は悪化しないと考えられる。この考えに基づき、我々は位相空間上の軌道計画において、十分な枝刈りを行うことで計算速度を保ちつつ解の精度も保証する枝刈り手法を提案する。ここで 4) の枝刈りに関し、以下の三種類で比較し、考察する。

- ①  $S_{found}$  全てに対して枝刈りを行わない
- ②  $S_{found}$  全てに対し、既に探索木へ登録されている全ての状態  $S_{all}$  とのユークリッド距離を比較し、そのユークリッド距離が閾値以下の場合には  $S_{found}$  を破棄する。
- ③  $S_{found}$  全てに対し、以下の条件で枝刈りを行う。

A Low Calculation Order Pruning Method in Phase Space for Searching Semi-Shortest Time Mechanical Motion

<sup>†</sup> Chyon Hae Kim, Hiroshi Tsujino (Honda Research Institute Japan)

<sup>‡</sup> Shimon Sugawara, Shigeki Sugano (Waseda University)

A:初期化処理

- A-1 位相空間を格子状に分割する
- A-2 分割した格子の各セル  $C$  を参照するための既知状態登録木を作成し、各セル  $C$  を既知状態登録木の枝として登録する

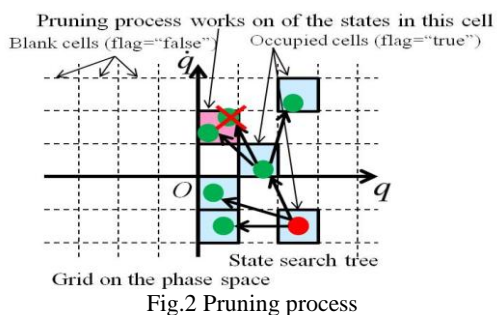
B:枝刈り処理

- B-1 3)において新しく状態探索木に登録した状態である  $S_{found}$  を、 $S_{found}$  の属する各セル  $C$  に対応した既知状態登録木の枝として登録を行うが、既に別の状態が登録されている場合には  $S_{found}$  を破棄する(例外として  $dist(S \in S_{found}, S_g)$  が  $dist(S \in S_{all}, S_g)$  の中で最小の場合には  $S$  を保持する)

①の手法は位相空間上で発見された  $S_{found}$  を全て状態探索木に登録する全探索の手法である。

②の手法では  $S_{found}$  と  $S_{all}$  のユークリッド距離比較を行う際に  $S_{all}$  の数に比例した計算コストがかかるため、 $S_{all}$  が大きくなると依然として計算量が多くなる。

③の手法は  $S_{all}$  の数によらず定計算コストで枝刈りを可能とするために提案する手法で、計算量の発散を低減した探索が行える。また、B-1により削除される状態  $S_{found}$  よりも既知状態登録木に既に登録済みの  $S \in S_{all}$  の方が早く同じ状態へ到達できる解となっており、準最短時間軌道生成に有効な解を保存することができる。③の枝刈りの概要図を Fig.2 に示す。



III. 実験

提案法により位相空間上における高速かつ正確な準最短時間軌道計画が行えることを確認するために、二重倒立振子の軌道計画について2通りの実験を行った。いずれの実験も  $(q_1, \dot{q}_1, q_2, \dot{q}_2) = (\pi, 0, 0, 0)$  を初期状態  $S_0$ ,

$(q_1, \dot{q}_1, q_2, \dot{q}_2) = (0, 0, 0, 0)$  を目標状態  $S_g$  として、初期状態から0.1秒間隔で最大5秒後までの状態を探索した ( $\Delta t = 0.1[s]$ , 最大探索木深さ =  $5[s] / \Delta t = 50$ )。実験1では振り子に伝達可能なトルク最大値  $\max(|\tau|)$  を十分に大きくして実験を行い、実験2では、前腕の最大トルクを  $0(\max(|\tau_2|) = 0)$  として実験を行った(Table 1)。また、軌道計画システムのパラメータは Table 2 のように設定した。

計算時間を測定するため、Intel Core2 Duo T7500搭載のうち1コアのみを使用して1スレッドによる計算を実行した。その結果を Table 3 に示す。提案手法である③は他の手法に比べ高速かつ正確に目標状態へと軌道計画が可能な枝刈り手法であることがわかる。

Table 1 The parameters of the pendulums

Exp.	Joint	Length [m]	Mass [kg]	Maximum torque [Nm]
1	1	0.5	0.5	2
1	2	0.5	0.5	1
2	1	0.5	0.5	1.5
2	2	0.5	0.5	0

Table 2 The parameters of the path planning system

Exp.	Joint	Torque resolution	Angular resolution	Angular velocity resolution
1	1	10	12	12
1	2	10	12	20
2	1	10	12	12
2	2	1 (since $\tau_2=0$ )	12	20

Table 3 Time cost and error

Exp.	Arg.	Num. of searches	Num. of $S_{all}$	Time [s]	Error
1	①	$(\sum_{n=1}^{50} 100^n)$	-	-	-
1	②	6983800	69838	$1.073 \times 10^5$	1.942
1	③	2730700	27307	59.4	0.264
2	①	$(\sum_{n=1}^{50} 10^n)$	-	-	-
2	②	100480	10048	325	1.470
2	③	123860	12386	5.6	0.505

IV. 考察

三種類の枝刈り手法における計算オーダと精度に関する考察を行う。

まず①の手法について考える。これは  $S_{found}$  の各状態に対してトルク粒度の数だけ次状態を探索するものである。つまり、探索深さ  $d$  における  $S_{found}$  の数を  $f$ 、トルク粒度を  $k$  とすると  $f = k^d$  となり、探索深さ  $d+1$  における状態を計算するための計算オーダは  $O(k \times k^d) = O(k^d)$  となるため容易に計算不能に陥ってしまう。

次に②の手法について考える。Table 3 の実験 1, 2 の比較からわかるように  $S_{all}$  が増えると探索にかかる時間が急激に増えていることがわかる。これは現在までの状態探索木に登録された状態数  $S_{all}$  の数を  $a$  とすると枝刈りを行うためのユークリッド距離比較回数に基づく計算オーダは  $O(af)$  となり、探索が進むと  $a$  が大きくなるため計算量が発散しやすい。

最後に、提案した③の手法について考える。これは  $S_{found}$  の属する各セル  $C$  が既知状態登録木の枝として登録されているか否かを調べるだけであるため、 $S_{all}$  の数によらず計算オーダは  $O(f)$  となる。この手法を Table 3 において②の手法とで比較してみると、 $S_{all}$  の数が計算時間に及ぼす影響が少なく絶対的な時間も少ない。また、目標状態との誤差も③の手法のほうが小さく、より有効な状態を残して探索されていることがわかる。なお、③の手法において Table 2 の角度と角速度の分解能を無限大とした場合が①に相当する。

V. まとめ

本稿では、運動方程式をもとに指定された目標姿勢・速度への準最短時間軌道計画を高速に実行するために、低計算オーダの枝刈りによる軌道計画手法を提案した。提案法を用いて二重倒立振子に関する軌道計画を行ったところ、初期状態より状態空間上の各状態への遷移方法が短時間で計画できることがわかった。また、提案法は位相空間を均等に探索する手法であり、効率のよい探索手法であると言える。

REFERENCES

[1] H.Ritter and K.Schulten: "Planning Dynamic Trajectory via Path Finding in Discretized Phase Space", Lecture Notes in Computer Science, Vol.253, pp.29-39, (1987).  
 [2] Pedro S. Huang: "Planning For Dynamic Motions Using a Search Tree", a graduate thesis of Toronto Univ., (1996).