

狭路走行のためのスーパー楕円を用いた Model Predictive Control による軌道計画の障害物制約の改良

殿谷 徳和, 向井 滋春, 齊藤 ゆみ

本稿では、Model Predictive Control (MPC) を用いた、自律走行ロボットの軌道計画技術に関する手法を述べる。MPC では、ロボットが与えられた目標を達成するために、時間とともに位置や姿勢を変化させるような速度指令を算出する。MPC の軌道計画では、ロボットの外形が微分可能であることを前提に周辺環境に対する障害物制約を設定する。そのため、ロボットの外形が微分可能ではない正方形等の多角形の場合、円や楕円といったプリミティブ形状で包含することで対応している。しかしこれでは、実際のロボットにサイズに対してクリアランスが過剰に大きくなり、本来通れるはずの狭路を走行できないことがある。そこで本稿では、スーパー楕円の特殊ケースであるスーパー円による近似を採用する。さらに、スーパー円を用いた際に生じる最適化計算の不安定性を解消するための制約式の改良を提案する。シミュレーション評価により、本手法が従来手法と比較してより狭路走行に適していることを示す。さらに、提案手法の平均処理時間は従来手法と同等であり、リアルタイム性があることを示す。

Refinement of Obstacle Constraints in Trajectory Planning Using Super Ellipses for Narrow Passage Navigation with Model Predictive Control

TONOGAI Norikazu, MUKAI Shigeharu and SAITOH Yumi

This paper presents a refinement of obstacle constraints in trajectory planning for autonomous mobile robots using Model Predictive Control (MPC). MPC generates velocity commands that guide the robot to achieve a given goal by adjusting its position and orientation overtime. In MPC formulations, obstacle constraints require differentiable representations of the robot's shape, making it difficult to directly incorporate polygonal shapes such as squares. As a result, robots are typically approximated using primitive shapes like circles or ellipses, which introduce excessive clearance and reduce flexibility in narrow environments. To address this limitation, we adopt a shape approximation method based on super ellipses. However, high-order super ellipses often lead to instability in numerical optimization. To overcome this, we introduce a refined constraints formulation that stabilizes the optimization process. The effectiveness of this method is evaluated through simulations, demonstrating that it is more suitable for narrow passage navigation compared to conventional methods. Furthermore, the average processing time of the proposed method is equivalent to that of conventional methods, ensuring real-time performance.

1. まえがき

製造現場や物流倉庫において、物品を特定の地点から別の地点へ搬送する作業が多く行われている。例えば、電気機器の組立工程では、部品棚から組立てセルまで部品を運

ぶ作業などが挙げられる。このような現場では多くの作業員が搬送作業に従事している。しかし、近年の労働人口の減少により、こうした単純作業の自動化が求められている。そのニーズに応える形で、自律走行ロボットを用いた搬送作業の自動化が進んでいる。搬送作業を自動化するにあたり、その利用範囲を拡大するための課題の一つに狭路

Contact : TONOGAI Norikazu norikazu.tonogai@omron.com

走行がある。工場内はスペースが限られているため、搬送を担うロボットは、設備の複雑なレイアウトによって屈曲し狭い通路を走行する能力が求められる。自動走行ロボットの狭路走行が可能になると、こうした限られたスペースでもスムーズな運用が可能となり、工場での適用範囲が広がるため普及が加速すると考えられる。

自律走行ロボットを走行させるためには自己位置推定、経路計画、軌道計画等の様々な技術が必要になる。本稿では、狭路走行に影響が大きい軌道計画技術に焦点をあてる。自律走行ロボットの軌道計画手法の代表例として、Model Predictive Control (MPC)¹⁾ と呼ばれる手法がある。MPC はロボットの走行を目的関数や制約式としてモデル化し最適化手法を用いて速度指令を算出する。MPC では、予測期間内の速度指令が軌道の滑らかさを考慮して算出される。しかしこの手法は演算量が多いことから、敬遠されることが多かった。近年では自律走行ロボットに搭載される計算リソースの向上により MPC を用いた研究が増加している²⁻⁵⁾。

一般的に、MPC ではロボットの形状を正方形などの多角形で表現することが難しく、障害物との干渉を直接制約式に設定することが困難である。その理由は、MPC で使用される最適化手法が制約式の勾配を必要とし、制約式が微分可能である必要があるためである。このため、ロボットの形状は円や楕円などで近似され制約式に組み込まれる。しかし単純な外接円などの近似では実際のロボットの形状との間に無駄な領域が生じ狭路走行には不向きである。無駄な領域を削減する方法として、複数の円を用いるアプローチもあるが、円の数が増えると計算量が増加する課題がある。一方で円の次数を高くすることで円が正方形に近づく特性を利用し、スーパー楕円の特長である次数の高い円（スーパー円）で近似する方法もあるが、次数を高くすると最適化計算が不安定になる問題が生じる。本稿では、計算量の増加を抑えつつ、近似による無駄な領域を削減するためにスーパー円を用いた際に生じる最適化計算の不安定性を解消するための制約式の改良を提案する。提案手法は、形状の正確な表現と計算量のトレードオフに対応しており、リアルタイム処理が可能で狭路走行に適したアプローチである。

2. 関連研究

MPC で正方形等の多角形のロボット形状を取り扱う方法として、以下の3つのよく知られている方法がある。以下ではその3つの方法について述べる。

2.1 プリミティブ形状を用いた方法

プリミティブ形状を用いた方法では、ロボットの形状を円や楕円等のプリミティブ形状で近似しそれを制約式に設定することで干渉回避を行う。例えば、長方形のロボット

形状では2つの円を組合せて形状を近似するものが多い^{2,3)}。このメリットは、表現が簡単で計算量が軽いことである。デメリットは長方形を複数のプリミティブ形状で近似しても無駄な部分があり、狭路走行において不利となることである。

2.2 混合整数最適化を用いた方法

混合整数最適化を用いた方法^{6,7)} では、連続値と整数値の2種類の変数を組合せて速度指令を求める。ロボットの外形の境界線を直線で表現し、その境界線の左右を整数値で表現しその組み合わせでロボットの内外を判定する。そして、ロボットの内側が壁や周囲の障害物に干渉しないように軌道を最適化する。この手法のメリットは、直線で形状が表現できるためロボット形状との間に無駄な領域がないことである。しかしデメリットとして障害物の数や通路の複雑さが増加すると最適化の組み合わせの数に伴い、計算時間が増大し、リアルタイム処理が困難になる。

2.3 双対性を用いた方法

ロボット形状と障害物の2つの凸集合間の最小距離を求める主問題とその双対問題の双対性を用いた方法では、制御入力算出する通常の MPC のモデルとロボット形状内の点と障害物形状内の点の最小距離を算出するモデルの2つを同時に解く⁴⁾。ロボット形状や障害物形状が多角形の場合、微分可能でないため勾配法で解くことができないが、双対問題に変換することによって双対変数は微分可能となる。通常の MPC のモデルとその双対問題を組合せて解くことで、ロボット形状を近似なく表現できるため無駄はない。しかし、解くべき変数が多くなるため計算量が多いデメリットがある。近年では Control Barrier Function (CBF) を用いてそのデメリットを削減した方法もある⁵⁾。しかし依然として計算量が多い点と前処理として障害物を多角形で近似する必要があるため、障害物の形状が変化すると前処理を再度実施する必要がありその分計算量が増える。

3. 提案手法

3.1 Model Predictive Control

今回の提案手法は、第2章関連研究²⁾ の2.1で紹介したプリミティブ形状を用いた方法を改良したものである。改良点を述べる前に、ベースとなる Model Predictive Control (MPC) のモデルについて説明する。モデルには目的関数と制約式が含まれている。

まず、目的関数は移動ロボットに実現させたい動作を定義するもので、目的関数を最小化することでその動作を実現する。今回設定した目的関数は式(1)の ϕ で表され、 x と y は移動ロボットが存在する2次元平面の x 軸と y 軸上の

位置を表し、 θ はその平面上の回転角度である。 v は並進速度、 ω は角速度を表している。予測ホライズン N_p は、時刻 t を時間間隔 dt で分割した個数で t 秒後までの予測を行うことを示している。式(1)の右辺の第一項と第二項は、台車の状態と現場のレイアウトなどの情報を元に経路計画等の手法を用いて得られた参照経路との差分を示す参照経路追従項を表している。これらの項を最小化することで、移動ロボットが参照経路をなぞるように走行する。参照経路追従項はターミナルコスト（第一項）とステージコスト（第二項）の二つに分解することができ、ターミナルコストは予測ホライズンの終点の台車の状態と参照経路との差から計算されるコストを表している。ステージコストは、予測ホライズンの始点から N_p-1 までの台車の状態と参照経路との差から計算されるコストを示す。参照経路追従項の概念図を図1に示す。参照経路上にリファレンス点（図中緑点）を設定し、目的関数を最小化する軌道を求め、その1ステップ目の指令値を実行する。これを繰り返すことで参照経路を追従するような走行が可能となる。

右辺第三項は、制御入力と参照速度との差分を計算する入力追従項である。この入力追従項を設定することで、陽に移動速度を調節することができる。例えばゴール付近で減速させたい場合、この項の参照速度に減速指令を設定すればよい。

$$\begin{aligned} \phi(x, y, \theta, v, \omega) = & e_1^T(k+N_p)Pe_1(k+N_p) \\ & + \sum_{i=1}^{N_p-1} e_1^T(k+i)Qe_1(k+i) \\ & + \sum_{i=1}^{N_p-1} e_2^T(k+i-1)Re_2(k+i-1) \end{aligned} \quad (1)$$

$$e_1(k+i) = \begin{pmatrix} x(k+i) - x_{ref}(k+i) \\ y(k+i) - y_{ref}(k+i) \\ \theta(k+i) - \theta_{ref}(k+i) \end{pmatrix}$$

$$e_2(k+i-1) = \begin{pmatrix} v(k+i-1) - v_{ref}(k+i-1) \\ \omega(k+i-1) - \omega_{ref}(k+i-1) \end{pmatrix}$$

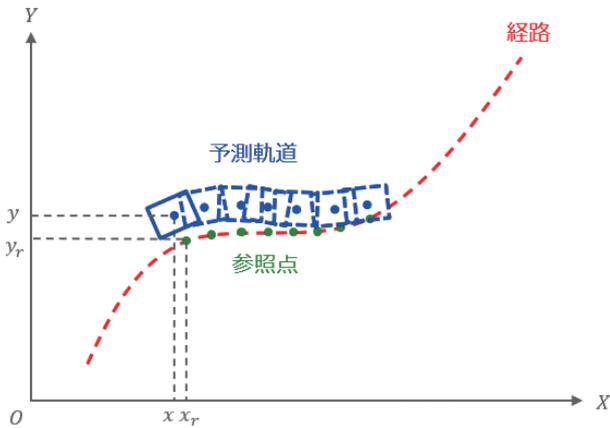


図1 目的関数のイメージ図

制約は次の4つの制約式からなる。キネマティクス制約は自社製品である自動搬送モバイルロボット LD シリーズと同じ構造の差動二輪モデルを用いた。差動二輪モデルは式(2)で表される。速度制約は式(3)のような不等式で表され、ロボットが実現可能な速度の上下限値を決めている。加減速制約は式(4)のように数値微分を用い、式(3)と同様に設定する。障害物制約は式(5)で表されセンサで計測された障害物の点群がロボットを被覆する円の外側にあること、つまり障害物とロボットが干渉していないことを表す。また、ロボットを被覆した時の無駄が少なくなるように、複数円を用いて障害物制約を設定する。図2のように赤い枠で示したロボットの外側に障害物があればこの制約は満たされていることとなる。

$$\begin{cases} x(k+i) = x(k+i-1) + v(k+i-1)\cos(\theta(k+i-1))dt \\ y(k+i) = y(k+i-1) + v(k+i-1)\sin(\theta(k+i-1))dt \\ \theta(k+i) = \theta(k+i-1) + \omega(k+i-1)dt \end{cases} \quad i=1,2,\dots,N_p \quad (2)$$

$$\begin{cases} v_{min} \leq v(k+i-1) \leq v_{max} \\ \omega_{min} \leq \omega(k+i-1) \leq \omega_{max} \end{cases} \quad i=1,2,\dots,N_p \quad (3)$$

$$\begin{cases} \dot{v}_{min} \leq \frac{v(k+i-1) - v(k+i-2)}{dt} \leq \dot{v}_{max} \\ \dot{\omega}_{min} \leq \frac{\omega(k+i-1) - \omega(k+i-2)}{dt} \leq \dot{\omega}_{max} \end{cases} \quad i=1,2,\dots,N_p \quad (4)$$

$$(x_{obsj} - x_i)^2 + (y_{obsj} - y_i)^2 \geq r^2 \quad i=1,2,\dots,N_p \quad j=1,2,\dots,N_{obs} \quad (5)$$

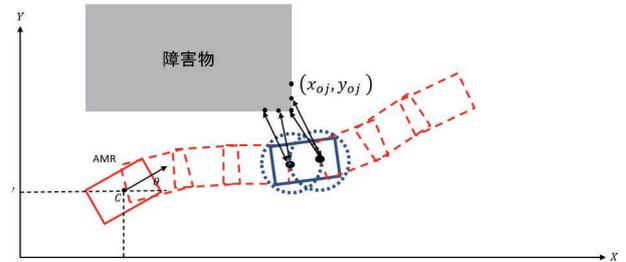


図2 障害物項のイメージ図

3.2 スーパー楕円（スーパー円）

式(6)のように、スーパー楕円の長軸と短軸を同じ長さにしたスーパー楕円の特殊ケースをスーパー円として導入する。式(6)の次数を大きくしていくと図3のように円が正方形に近づいていく。この性質を利用し障害物制約を複数円で設定している部分をスーパー円に置き換える。スーパー円の次数 p を上げることで正方形のロボット形状でも実形状からの乖離を限りなく削減できる。

$$\begin{aligned} (x_{\text{obs}j} - x_i)^{2\rho} + (y_{\text{obs}j} - y_i)^{2\rho} &\geq r^{2\rho}, \\ i = 1, 2, \dots, N_p, j = 1, 2, \dots, N_{\text{obs}} \end{aligned} \quad (6)$$

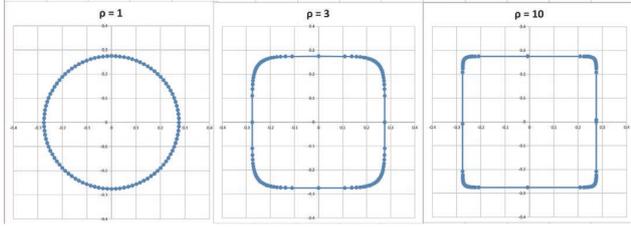


図3 スーパー円

3.3 スーパー円に対する課題と対策

スーパー円を用いる上での課題は、次数を大きくした時に障害物制約の値が発散したり潰れたりすることである。ここで“潰れ”とは、値が1以下の場合に次数が高くなるにつれて値が0に近づき、小さくなる現象を指す。発散や潰れが発生すると数値計算が安定せず最適化を解くことができない。例えば、次数6の場合ある制約の値が527.2だが次数を12にすると527482623.1となる。このように急激に値が大きくなると数値計算が不安定になりやすい。先行研究でもスーパー楕円を用いたものがある^{8,9)}が次数は小さい。この課題を解決するためにスーパー円を使った制約式(6)を式(7)のように変形することを考案した。変形のポイントは次の3つである。まず1つ目は数値の潰れを防ぐために式(6)の両辺をスーパー円の半径の 2ρ 乗である $r^{2\rho}$ で割る。2つ目は対数の値がマイナス無限大になるのを防ぐために両辺に1を足す。3つ目は数値的发散を防ぐために両辺の対数をとる。以上の変形を実施することで数値的な安定性を確保する。式(7)の変換は両辺に対して行われ不等式の大小関係は変換前と変わらないためこの変形を行っても障害物の干渉を判定することが可能である。

$$\begin{aligned} \log\left(\left(\frac{x_{\text{obs}j} - x_i}{r}\right)^{2\rho} + \left(\frac{y_{\text{obs}j} - y_i}{r}\right)^{2\rho} + 1\right) &\geq \log(2), \\ i = 1, 2, \dots, N_p, j = 1, 2, \dots, N_{\text{obs}} \end{aligned} \quad (7)$$

次にスーパー円を用いた際の効果について考察する。例えば、1辺45 cmの正方形に両側3 cmの安全のためのマージンを設けた1辺51 cmの正方形に外接円を設定した場合と、同じ正方形に次数20のスーパー円を適用した場合を比較する。図4より外接円を使用すると1辺45 cmの正方形から最大で13.5 cmの無駄な幅が発生する。スーパー円を適用した場合、無駄な幅は最大で設定したマージンとほぼ一致する3 cmに抑えられる。この結果、スーパー円を利用することで約10 cmの無駄な幅を削減できるため外接円に比べて20 cm狭い通路でも走行可能となる。

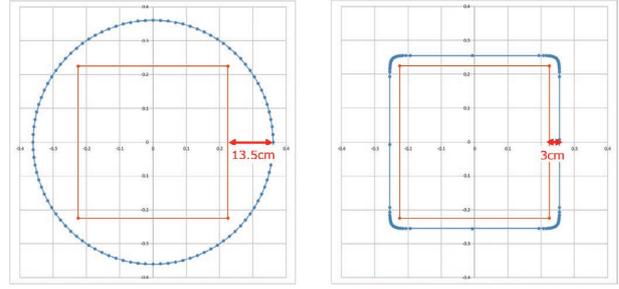


図4 スーパー円の効果

次に制約式の改良の効果を示すために最適化の計算に式(6)や式(7)の不等式制約がどのように影響するのかを述べる。まず最初に最適化手法の1つである内点法について述べる。最初に式(8)のような最適化問題が与えられたとする。

$$\begin{aligned} \min_x \quad &\frac{1}{2}x^T Qx + c^T x \\ \text{s.t.} \quad &Ax = b \\ &Gx \geq d \end{aligned} \quad (8)$$

ここで1行目の式は目的関数、2行目の式は等式制約、3行目の式は不等式制約を表している。この最適化問題では等式、不等式制約をみたくつづつ目的関数を最小する x を求める問題である。最適な点では式(9)のKarush-Kuhn-Tucker条件(KKT条件)を満たしている必要があるためこのKKT条件を使って最適な x を求める。

$$\begin{aligned} -Qx - c + A^T \lambda + G^T \mu &= 0 \\ Ax &= b \\ Gx - d &\geq 0 \\ \mu &\geq 0 \\ (Gx - d)_i \mu_i &= 0 \end{aligned} \quad (9)$$

ここで λ 、 μ はラグランジュ乗数である。式(9)において x 、 λ 、 μ の3つが変数となる。KKT条件を直接解いて3つの変数を求めることは難しいので次の式(10)のように各変数の微小量で展開する。この際不等式は一旦無視し0と置く。

$$\begin{aligned} -Q(x+dx) - c + A^T(\lambda+d\lambda) + (\mu+d\mu) &= 0 \\ A(x+dx) &= b \\ \Sigma\mu + \Sigma d\mu + d\Sigma\mu &= 0 \\ \Sigma &= \text{diag}((Gx-d)_1, \dots, (Gx-d)_n) \end{aligned} \quad (10)$$

変化が微小量である必要があるために第3式に $(Gx-d)_i \mu_i$ の平均値 v で計算される緩和項を追加すると式(11)となる。

$$\begin{aligned} A^T(\lambda+d\lambda) + (\mu+d\mu) &= Q(x+dx) + c \\ A(x+dx) &= b \\ X\mu + Xd\mu + \mu dX &= \sigma v e \end{aligned} \quad (11)$$

ここで $\sigma \in [0,1)$ のスケール項、 e は要素がすべて 1 のベクトルである。式(11)を微小量とそれ以外でまとめると式(12)となる。式(12)の微小量にかかる行列を KKT 行列と言う。

$$\begin{bmatrix} -Q & A^T & G^T \\ A & 0 & 0 \\ MG & 0 & \Sigma \end{bmatrix} \begin{bmatrix} dx \\ d\lambda \\ d\mu \end{bmatrix} = \begin{bmatrix} Qx+c-A^T\lambda-\mu \\ b-Ax \\ \sigma ve-\Sigma\mu \end{bmatrix} \quad (12)$$

$$\Sigma = \text{diag}((Gx-d)_1, \dots, (Gx-d)_n)$$

$$M = \text{diag}(\mu_1, \dots, \mu_n)$$

$$e = [1, \dots, 1]^T$$

これより微小量を求めるには KKT 行列の逆行列をもとめ以下の式(13)を解けばよいことがわかる。

$$\begin{bmatrix} dx \\ d\lambda \\ d\mu \end{bmatrix} = \begin{bmatrix} -Q & A^T & G^T \\ A & 0 & 0 \\ MG & 0 & \Sigma \end{bmatrix}^{-1} \begin{bmatrix} Qx+c-A^T\lambda-\mu \\ b-Ax \\ \sigma ve-\Sigma\mu \end{bmatrix} \quad (13)$$

上記より求めた微小量をもとに x 、 λ 、 μ を逐次的に更新することで最適な x を算出する。上記に対して式(6)や式(7)の制約式を適用する場合は式(6)、式(7)が非線形制約であるためある点 x の周りで線形近似したものを代入し解く。そしてそのある点を徐々に更新し最適変数を求める。式(6)や式(7)を、次数20とし、 \bar{x} 、 \bar{y} 周りで線形化したものは以下の式(14)、式(15)になる。ここで簡単のために障害物は1点であるとした。

$$G = \begin{bmatrix} -20(x_{obsj} - \bar{x}_i)^{19} & -20(y_{obsj} - \bar{y}_i)^{19} \end{bmatrix} \quad (14)$$

$$d = (x_{obsj} - \bar{x}_i)^{20} + (y_{obsj} - \bar{y}_i)^{20} - r^{20}$$

$$G = \begin{bmatrix} \frac{-20r}{\left(\frac{x_{obsj} - \bar{x}_i}{r}\right)^{20}} \left(\frac{x_{obsj} - \bar{x}_i}{r}\right)^{19} & \frac{-20r}{\left(\frac{y_{obsj} - \bar{y}_i}{r}\right)^{20}} \left(\frac{y_{obsj} - \bar{y}_i}{r}\right)^{19} \end{bmatrix} \quad (15)$$

$$d = \log \left[\left(\frac{x_{obsj} - \bar{x}_i}{r}\right)^{20} + \left(\frac{y_{obsj} - \bar{y}_i}{r}\right)^{20} + 1 \right] - \log(2)$$

ここで仮に $x_{obsj} - \bar{x}_i$ と $y_{obsj} - \bar{y}_i$ の値を 2、 r の値を 1 としてそれぞれの G と d を計算すると以下の表 1 の値になる。

表 1 制約式の改良の効果

次数	パラメータ	従来手法	提案手法
次数 2	G	(-40, -40)	(-10, -10)
	d	7	0.653212514
次数 10	G	(-10240, -10240)	(-10, -10)
	d	2047	3.010511963
次数 20	G	(-10485760, -10485760)	(-10, -10)
	d	2097151	6.02060012

上記の表 1 に示すとおり従来手法では次数が高くなるにつれて G と d の値が急激に大きくなっている。このような大きな値で式(13)の計算を行おうとすると G や Σ の値が大きいため式(13)の KKT 行列の逆行列が計算できず微小量を求めることができない。また仮に計算できたとしても微小量が大きな値になり更新がうまくいかず最適変数を求めることができない。しかし、提案手法では次数が大きくなっても G と d の値の急激な変化が抑えられている。これにより最適解を安定して求めることが可能となる。上記が最適化手法に制約式の変形を導入した時の効果の例である。

3.4 スーパー円の障害物制約に改良を加えた定式化

スーパー円の障害物制約に改良を加えて最終的な定式化は式(16)になる。この問題設定は目的関数が二次式、キネマティクス制約式と障害物制約式が非線形関数であるため非線形最適化手法を用いて最適化を行った。今回は非線形最適化手法の一種である Sequential least squares quadratic programming (SLSQP) を用いた。

$$\phi = \min \left(e_1^T(k+N_p)Pe_1(k+N_p) + \sum_{i=1}^{N_p-1} e_1^T(k+i)Qe_1(k+i) + \sum_{i=1}^{N_p-1} e_2^T(k+i-1)Re_2(k+i-1) \right) \quad (16)$$

s.t.

$$\begin{cases} x(k+i) = x(k+i-1) + v(k+i-1)\cos(\theta(k+i-1))dt \\ y(k+i) = y(k+i-1) + v(k+i-1)\sin(\theta(k+i-1))dt \quad i=1,2,\dots,N_p \\ \theta(k+i) = \theta(k+i-1) + \omega(k+i-1)dt \end{cases}$$

$$\begin{cases} v_{\min} \leq v(k+i-1) \leq v_{\max} \\ \omega_{\min} \leq \omega(k+i-1) \leq \omega_{\max} \end{cases} \quad i=1,2,\dots,N_p$$

$$\begin{cases} \dot{v}_{\min} \leq \frac{v(k+i-1) - v(k+i-2)}{dt} \leq \dot{v}_{\max} \\ \dot{\omega}_{\min} \leq \frac{\omega(k+i-1) - \omega(k+i-2)}{dt} \leq \dot{\omega}_{\max} \end{cases} \quad i=1,2,\dots,N_p$$

$$\log \left(\left(\frac{x_{obsj} - x_i}{r} \right)^{2\rho} + \left(\frac{y_{obsj} - y_i}{r} \right)^{2\rho} \right) \geq \log(2),$$

$$i = 1, 2, \dots, N_p, \quad j = 1, 2, \dots, N_{obs}$$

4. 評価実験

4.1 評価方法

評価実験として、シミュレーションで狭路をスタートからゴールまで走行可能か評価し提案手法の有効性を確認した、今回の評価実験ではロボットの形状を長辺 65 cm、短辺 45 cm の長方形とし、走行環境として狭路 1 と狭路 2 の 2 種類を用意した。図 5 に狭路 1 と狭路 2 を示す。狭路 1 では狭路に対して旋回し侵入することを想定している。狭路 2 は狭路内にクランクがある。各狭路に対して狭路幅 70 cm、と 80 cm の 2 種類を用意した。

比較手法としては、プリミティブ形状を用いた手法の複数円を用いた手法と比較した。走行に必要な経路は二段階 Fast Marching Method (FMM)¹⁰ と呼ばれる方法で算出し、自己位置は差動二輪のキネマティクスを元に理想的に動いたとして算出した。安全のために設けるマージンはロボットの形状の長方形に対して、その長方形の外側に 3 cm とした。主要なパラメータの設定は、予測ホライズンは 6、ホライズンの時間間隔 dt は 0.2 s、並進速度の上下限は ± 1.0 m/s、並進加減速の上下限は ± 0.5 m/s²、旋回速度の上下限は ± 180 度/s、旋回加速度の上下限は ± 180 度/s² である。スーパー円の半径は 25.5 cm、次数は 20 とし複数円の半径は長方形を包含するために 36.1 cm とした。評価は Intel (R) Core (TM) i7-7700@3.60 GHz の CPU を搭載した計算機を用いて処理時間の計測を行った。

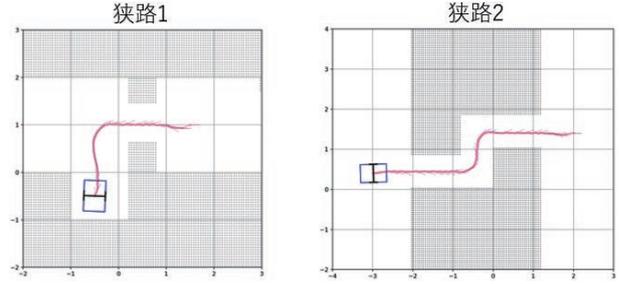


図 5 狭路

図中の長方形が移動するロボット、グレー部分が障害物、赤線が移動ルートである。赤線の終端が移動の終点を表す。

4.2 評価結果

図 5 左側に示すように、狭路 1 はおよそ直角に狭路に侵入するケースである。図 6 左側に示す通り、狭路幅が 80 cm の場合は複数円を用いた方法と提案手法どちらもゴールまでたどり着けている。狭路幅が 70 cm になると複数円を用いた方法では、途中で最適化が解けず停留してしまいゴールまでたどり着けなかった。一方で提案手法はゴールまで到達できている。次に図 5 右の狭路 2 のクランク状の経路に関しては、図 6 右側のように狭路幅が 80 cm の場合は複数円を用いた方法と提案手法はどちらもゴールまで走行できている。狭路幅が 70 cm の場合は複数円を用いた方法は途中で停留してしまいが、提案手法はゴールまで走行することができた。この結果から従来手法よりも提案手法のほうが狭路走行に適していると言える。その理由は提案手法の方がロボットを包含する干渉モデルの無駄が少なく狭路に対して余裕があることで、最適化問題が簡単に

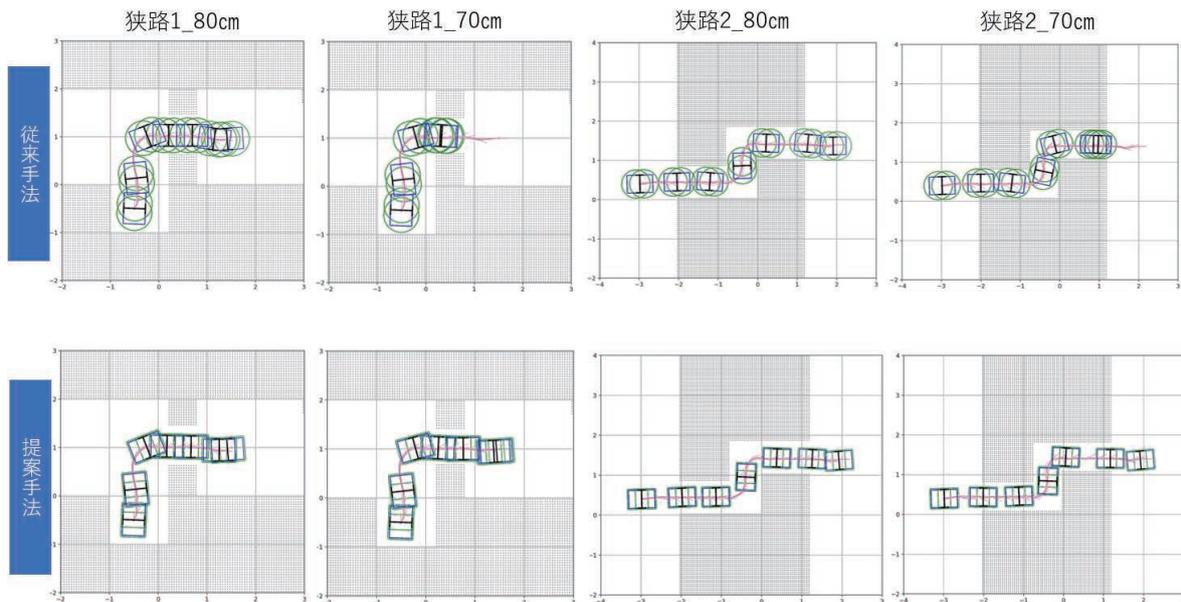


図 6 狭路の走行結果

なったことが原因だと考える。処理時間の結果は表 2 に示す。最適化処理はロボットが経路を進む度に逐次的に実施しており、障害物との近さなどに応じて処理時間が変動する。よって、スタートからゴールまで走行する間に実施した最適化処理の平均と最大の処理時間を算出した。比較手法と提案手法で平均処理時間はほぼ差がない。Max 値に関しては提案手法の方が多少長くなっている。これは次数部分の計算が影響していると考えられる。従来手法も提案手法も Max 値が約 100 ms 程度であるためリアルタイム性を保持していると言える。

表 2 処理時間

狭路	処理時間	従来手法	提案手法
狭路 1 80 cm	平均 [ms]	21	31
	最大 [ms]	64	98
狭路 1 70 cm	平均 [ms]	53	29
	最大 [ms]	123	90
狭路 2 80 cm	平均 [ms]	18	25
	最大 [ms]	70	73
狭路 2 70 cm	平均 [ms]	31	18
	最大 [ms]	110	118

5. むすび

本研究では、MPC で狭路エリアを走行するためのスーパー円の障害物制約を改良した手法を提案した。従来手法との比較では、よりシビアな条件の狭路に対応できることが示された。処理時間に関しても従来手法と比較して同程度の水準であり、リアルタイム性が必要なロボット制御に活用できると考えられる。

今後の課題としては、実機を用いた動作検証の実施に加え、スーパー円をスーパー楕円に戻すことで、必要な円の個数を削減し、計算効率の向上を図ることが挙げられる。また、本手法は障害物制約式の改良であるため他の問題設定への応用が期待される。例えば、オフライン軌道計画や経路計画の手法である。今後はこれらの拡張を行う予定である。

本研究は、中外製薬株式会社との共同研究の一環として行った。

参考文献

- 1) K. Felipe et al., "Model predictive control of a mobile robot using linearization," in *Proc. Mechatron. Robot.*, 2004, vol. 4, pp. 525-530.
- 2) B. Brito et al., "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robot.*

Automat. Lett., vol. 4, pp. 4459-4466, 2019.

- 3) T. M. Cho et al., "Model predictive control of autonomous vehicles with integrated barriers using occupancy grid maps," *IEEE Robot. Automat. Lett.*, vol. 8, pp. 2006-2013, 2023.
- 4) X. Zhang et al., "Optimization-based collision avoidance," *IEEE Trans. Control Syst. Tech.*, vol. 29, pp. 972-983, 2021.
- 5) A. Thirugnanam et al., "Safety-critical control and planning for obstacle avoidance between polytopes with control barrier functions," in *ICRA*, 2022, pp. 286-292.
- 6) A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proc. 2002 Amer. Cont. Conf.*, 2002, vol. 3, pp. 1936-1941.
- 7) L. Blackmore et al., "Chance-constrained optimal path planning with obstacles," *IEEE Trans. Robot.*, vol. 27, pp. 1080-1094, 2011.
- 8) H. Febbo et al., "Moving obstacle avoidance for large, high-speed autonomous ground vehicles," in *Amer. Cont. Conf.*, 2017, pp. 5568-5573.
- 9) P. Vlantis et al., "Quadrotor landing on an inclined platform of a moving ground vehicle," in *ICRA*, 2015, pp. 2202-2207.
- 10) S. Garrido et al., "FM2: A real fast marching sensor-based motion planner," *Int. J. Robot. Automa.*, pp. 1-6, 2008.

執筆者紹介



殿谷 徳和 TONOGAI Norikazu

技術・知財本部
ロボティクス R&D センタ
Voyager Project 部
専門：軌道計画、画像処理



向井 滋春 MUKAI Shigeharu

技術・知財本部
ロボティクス R&D センタ
Voyager Project 部
専門：機械学習、経路計画



齊藤 ゆみ SAITOH Yumi

技術・知財本部
ロボティクス R&D センタ
専門：制御工学
所属学会：システム制御情報学会、電子情報通信学会、電気学会

本文に掲載の商品の名称は、各社が商標としている場合があります。