

自動運転における MPC を用いた動的障害物の回避

2020SC003 青野隆太 2020SC025 井上陸

指導教員：中島明

1 はじめに

近年では、車両の自動運転の研究が進められているが現時点では完全自動運転を行う車は認可されておらず、いまだ研究段階であるといえる。しかし、この完全自動運転が現代の車社会が抱える様々な問題を解決する糸口となる。その問題の一つが人のミスに起因する交通事故である。現代の自動車社会では、移動手段として当たり前のように多くの人が自動車を運転している。それに伴い多くの自動車事故も発生しているが、それもそれらの事故のほとんどが人間のミスや不注意によって起こるものである。しかし、完全自動運転が完成し普及すればこのような事故を 0 に減らすことも不可能ではない。その完全自動運転の実現のための第一歩として、通常の運転だけではなく、イレギュラーに対応する技術の研究も必要不可欠である。本研究ではその中でも、自動運転における障害物の回避のためのシミュレーションを行う。

2 車両の軌跡追従

ここでは、車両の軌跡追従について説明する。

2.1 運動学モデル

運動学モデルとは、運動の結果としての車両の位置と速度のみを考えるモデルである。このモデルはパラメータが少なく、計算も容易であるという特徴を持つ。そのため、車載用途や機械の学習モデルなどに多く使われる。ここで、運動学モデルを用いた車両の運動を図 1 に表す。また、今回使用するパラメータを表 1 に示す。ただし、このモデルは四輪車両の左右のタイヤ特性や重量バランスが完全に対象であることを仮定した場合に成り立つ二輪モデルを前提とする。

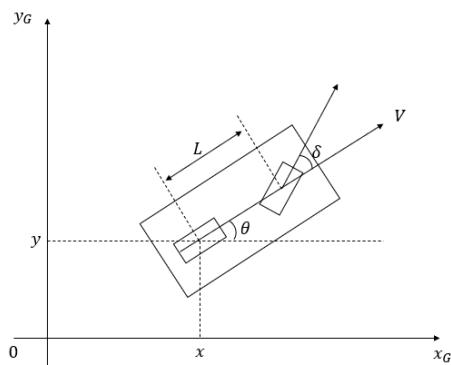


図 1 運動学モデル

表 1 パラメータ

定義	記号	単位
車両位置	x	m
車両位置	y	m
車体の長さ	L	m
直進速度	V	m/s
姿勢角	θ	rad
転舵角	δ	rad

ここで、反時計回りを正とした車両の向きを θ [rad] とするとこのモデルは次のように表すことができる。

$$\begin{cases} \frac{dx}{dt} = V \cos \theta \\ \frac{dy}{dt} = V \sin \theta \\ \frac{d\theta}{dt} = \frac{V}{R} = \frac{V \tan \delta}{L} \end{cases} \quad (1)$$

3 MPC (モデル予測制御)

MPC とは、有限時間先までの動きをモデルを使って予測し、最適化問題を解き、制御入力を計算するという制御方法である。MPC では、PID 制御の D 制御と比べてより細やかな予測モデルを使うため、PID 制御より正確に予測できるという特徴がある。MPC の考え方についてを次の図 2 に示す。

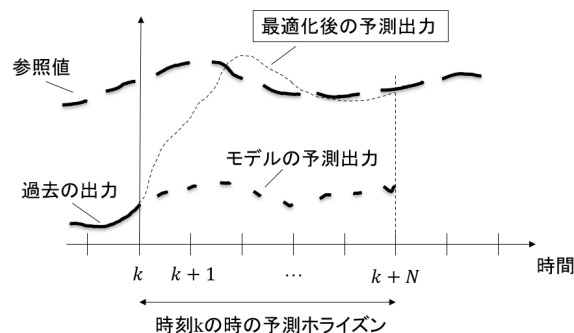


図 2 MPC の考え方

図 2 に示した考え方のように、MPC では、各制御周期において、予測 horizon と呼ばれる N ステップ先までの有限時間を考えて予測モデルをもとに予測出力を計算する。そして、予測出力と参照値の差分や、入力などで構成する評価関数を最小化する最適入力を求める。得られた最適入力の初期値を、制御周期における制御入力とする。このように、MPC ではある程度先までの時間を考えて、毎周期、最適な入力を計算することで高い精度の制御が可能となる。

例として、制御対象が次に示すような n 次元 m 入力離散時間であるとする。

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k) + \boldsymbol{\omega}_d(k)$$

離散的な時刻を $k \in \mathbb{Z}$ とし、時刻 k における、システムの状態変数ベクトル、入力ベクトル、外乱ベクトルをそれぞれ、 $\mathbf{x}(k) \in \mathbb{R}^n$ 、 $\mathbf{u}(k) \in \mathbb{R}^m$ 、 $\boldsymbol{\omega}(k) \in \mathbb{R}^n$ とする。また、 $\mathbf{F} \in \mathbb{R}^n \times n$ ($n \times n$ の実行列) $\mathbf{G} \in \mathbb{R}^n \times m$ である。このとき、評価関数は次のように表すことができる。

$$J(\mathbf{x}, \mathbf{u}) = \frac{1}{2} \mathbf{x}^T(N) \mathbf{Q}_f \mathbf{x}(N)$$

$$+ \frac{1}{2} \sum_{k=0}^{N-1} (\mathbf{x}^T(k) \mathbf{Q} \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{R} \mathbf{u}(k))$$

ただし、時刻 $k = N$ の状態変数ベクトルの重みを準正定行列 $\mathbf{Q}_f \in \mathbb{R}^n \times n$ 、時刻 $k = 0 \dots, N-1$ の状態変数ベクトルの重みを準正定行列 $\mathbf{Q} \in \mathbb{R}^n \times n$ 、入力ベクトルの重みを正定行列 $\mathbf{R} \in \mathbb{R}^m \times m$ とする。

4 MPC による円軌跡追従

線形 MPC による円軌跡追従についてシミュレーションを行っていく。

4.1 モデル設定

シミュレーションモデルを、図3のように設定する。

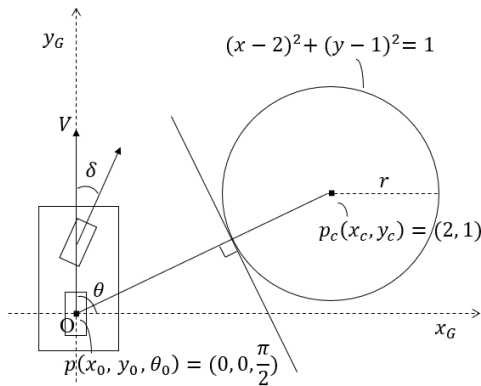


図3 MPCのシミュレーションモデル

また、横偏差と姿勢偏差については図4に示すように、それぞれ e_{lat} 、 e_θ と表す。

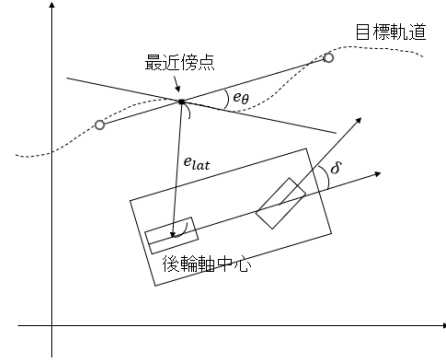


図4 横偏差と姿勢偏差

このとき、 e_{lat} は横偏差であり、後輪軸中心から目標軌道上の最近傍点までの距離を、 e_θ は姿勢偏差であり、最近傍点での接線方向と車両の方位角との角度差を表している。

4.2 モデルの線形化

次の図5に示すモデルにおいて線形化を行う。

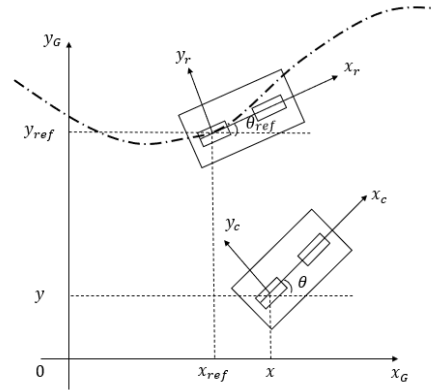


図5 MPCの偏差のモデル

図5における x_{ref} 、 y_{ref} 、 θ_{ref} はそれぞれ、目標軌道上の最近傍点における車両の目標座標と方位を表している。ここで、 θ_{ref} を基準とした座標系に $x-y$ 座標系を変換するため、次の示す式のようにもとの座標系に対し θ_{ref} 回転させる。

$$\left\{ \begin{bmatrix} x - x_{ref} \\ y - y_{ref} \end{bmatrix} = \mathbf{R}(\theta_{ref}) \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} \right.$$

\Leftrightarrow

$$\left. \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \mathbf{R}^T(\theta_{ref}) \begin{bmatrix} x - x_{ref} \\ y - y_{ref} \end{bmatrix} \right\}$$

$$\mathbf{R}(\theta_{ref}) = \begin{bmatrix} \cos(\theta_{ref}) & -\sin(\theta_{ref}) \\ \sin(\theta_{ref}) & \cos(\theta_{ref}) \end{bmatrix}$$

$$\bar{\theta} = \theta - \theta_{ref}$$

先述した通り、 θ_{ref} は最近傍点における座標であるため時間に依存しないという点を考慮すると、軌跡座標系のモデルでは次式が成り立つ。

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} &= \mathbf{R}^T(\theta_{ref}) \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \\ &= \begin{bmatrix} V \cos \bar{\theta} \\ V \sin \bar{\theta} \end{bmatrix} \end{aligned}$$

目標軌跡上の最近傍点における、目標角速度を ω_{ref} 、目標旋回半径 R_{ref} とすると、車両の方位 $\bar{\theta}$ は、 $\theta - \theta_{ref}$ であるため次の式のように表せる。

$$\begin{aligned} \frac{d\bar{\theta}}{dt} &= \frac{d}{dt}(\theta - \theta_{ref}) = \frac{V \tan \delta}{L} - \omega_{ref} \\ \omega_{ref} &= \frac{V}{R_{ref}} \end{aligned}$$

これをまとめると、

$$\frac{d}{dt} \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{\theta} \end{bmatrix} = \begin{bmatrix} V \cos \bar{\theta} \\ V \sin \bar{\theta} \\ \frac{V \tan \delta}{L} - \frac{V}{R_{ref}} \end{bmatrix}$$

となる。ここでは軌跡追従の目的より、車両の方位が目標方位から大きくずれていないこと、すなわち $\bar{\theta}$ が微小であることと、前輪転舵角 δ が微小であることの2つを仮定すると、

$$\frac{d}{dt} \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{\theta} \end{bmatrix} \approx \begin{bmatrix} V \\ V\bar{\theta} \\ \frac{V\delta}{L} - \frac{V}{R_{ref}} \end{bmatrix}$$

と線形化できる。また、 \bar{x} 方向には速度が一定であるためこれを省略して、 \bar{y} 、 $\bar{\theta}$ について状態方程式で表すと次の式のようになる。

$$\frac{dx}{dt} = \mathbf{A}x + \mathbf{b}u + \omega_c$$

$$\mathbf{A} = \begin{bmatrix} 0 & V \\ 0 & 0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 & \frac{V}{L} \end{bmatrix}^T, \omega_c = \begin{bmatrix} 0 & -\frac{V}{R_{ref}} \end{bmatrix}^T$$

ここで離散ステップ時間を Δt として、上の式をオイラー法を用いて離散化すると以下のような式が求まる。

$$x(k+1) = \mathbf{F}x(k) + \mathbf{G}u(k) + \omega_d(k)$$

$$\begin{cases} \mathbf{F} = \begin{bmatrix} 1 & V\Delta t \\ 0 & 1 \end{bmatrix} \\ \mathbf{G} = \begin{bmatrix} 0 & \frac{V}{L}\Delta t \end{bmatrix}^T \\ \omega_d = \begin{bmatrix} 0 & -\frac{V}{R_{ref}}\Delta t \end{bmatrix}^T \end{cases}$$

これによって、MPC を適応することが可能となる。

4.3 シミュレーション

4.2 節に基づいて、MATLAB 上で CVXGEN を用いて軌跡追従を行った [2]。そのときのシミュレーション結果を次の図6と図7に示す。なお、今回は $(x-2)^2 + (y-1)^2 = 1$ の円を追従することを目的としている。また、このシミュレーションに用いた車速と車体の長さは次のように設定した。

$$V = 1 \quad L = 0.2$$

また、初期値は次のように設定した。

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \pi/2 \end{bmatrix}$$

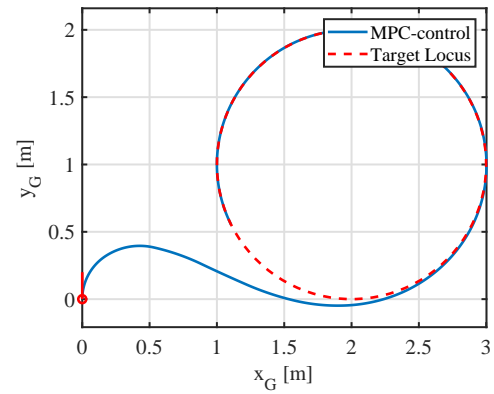


図6 MPC を用いた軌跡追従

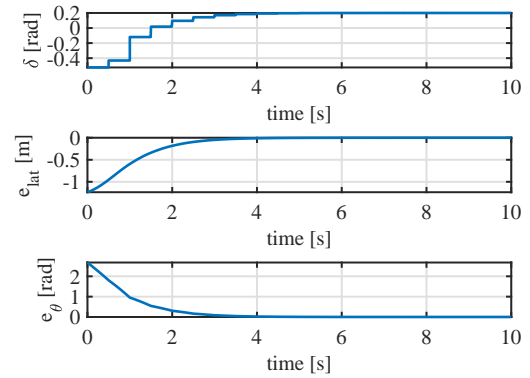


図7 軌跡追従における操舵角と偏差

5 MPC における動的障害物回避

ここでは、非線形 MPC における動的障害物回避について、ライブラリは CasADi を、ソルバーは ipopt を用いてシミュレーションを行う [3]。

5.1 モデル設定

今回は車両の追い越しを想定した次の図8のような状況でシミュレーションを行う。

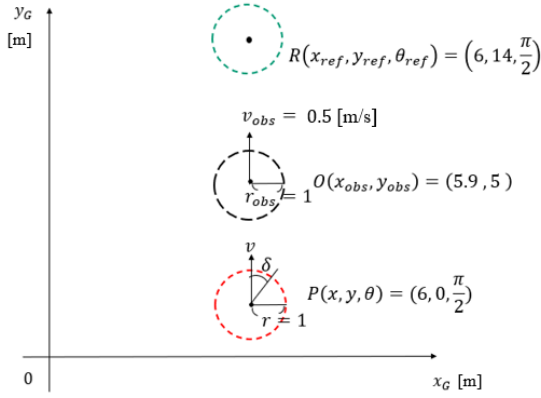


図8 動的障害物回避の状況設定

ここでの動的障害物は自車の前を走行する他車両を想定している。また、この他車両は一定の速度で前進し、途中での進路変更がないと仮定している。

5.2 モデルの導出

システムモデルは(1)と同様のものである。これを、オイラー近似すると以下の式ようになる。

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} + \Delta T \begin{bmatrix} V(k) \cos \theta(k) \\ V(k) \sin \theta(k) \\ \frac{V(k) \tan \delta(k)}{L} \end{bmatrix}$$

ただし、 k を時間ステップ、 ΔT をサンプリング時間とする。

次に、評価関数を以下のように設定する。

$$J_N(\mathbf{x}_0, \mathbf{u}) = \sum_{k=0}^{N-1} \|\mathbf{x}_u - \mathbf{x}_{ref}\|_Q^2 + \|\mathbf{u} - \mathbf{u}_{ref}\|_R^2 \quad (2)$$

ただし、 N をホライズン数、状態変数ベクトルの重みを準正定行列 $\mathbf{Q} \in \mathbb{R}^n \times n$ 、入力ベクトルの重みを正定行列 $\mathbf{R} \in \mathbb{R}^m \times m$ とする。

次に、評価関数 J_N を用いて、制約条件を以下のように設定する。

$$\begin{aligned} & \underset{\mathbf{u}}{\text{minimize}} && J_N(\mathbf{x}_0, \mathbf{u}) \\ & \text{subject to} && \mathbf{x}_u(k+1) = \mathbf{f}(\mathbf{x}_u(k), \mathbf{u}(k)), \\ & && \mathbf{x}_u(0) = \mathbf{x}_0, \\ & && \mathbf{u}(k) \in U, \quad \forall k \in [0, N-1], \\ & && \mathbf{x}(k) \in X, \quad \forall k \in [0, N] \end{aligned} \quad (3)$$

また、図8のような状況において、障害物回避を行うために(3)の制約条件に次の制約条件を追加する。

$$-\sqrt{(x - x_{obs})^2 + (y - y_{obs})^2} + (r + r_{obs}) \leq 0$$

5.3 シミュレーション

5.2節の制約条件をもとに、ホライズン数 N を14として行ったシミュレーションの結果を次の図9に示す。このとき、サンプリング時間は0.2sで重みは次のように設定した。

$$Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad R = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.05 \end{bmatrix} \quad (4)$$

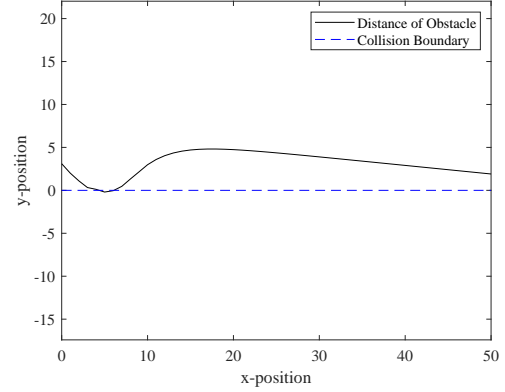


図9 シミュレーションにおける車両と障害物の距離

図9を見ると、障害物回避の途中で自車と障害物との接触が見られる。

5.4 考察

前節で見受けられた自車と障害物との接触については、MPCは荒い時間で荒い近似をしているのに対し、プログラムは現実に近いもので解いているため、そこにズレが生じ接触してしまったのではないかと考察する。

6 おわりに

本研究では、MPCを用いた円軌跡追従と、非線形MPCを用いた動的障害物回避について行ってきた。今後は、より現実に近い状況設定で実機実験を行っていきたい。

参考文献

- [1] [日本ロボット学会編著] 自動運転技術入門 AIx ロボティクスによる自動車の進化
- [2] 西山 輝：『軌跡追従による自動運転の制御』，南山大学理工学部機械電子制御工学科，愛知，2021
- [3] Mohamed W. Mehrez, Phd, Optimization based Solutions for Control and State Estimation in Dynamical Systems(Implementation to Mobile Robots) A Workshop, https://github.com/MMehrez/MPC-and-MHE-implementation-in-MATLAB-using-Casadi/blob/master/workshop_github/MPC_MHE_slides.pdf