

# 様々な軌道に対するモデル予測制御を用いた追従制御の挙動の検証

2017SC070 諏訪佑季

指導教員：陳 幹

## 1 はじめに

コンピュータの計算スピード増加が原因で、モデル予測制御は近年自動運転の研究にもよく使われるようになってきている。本研究では、モデル予測制御を用いて車両を目標軌道に追従制御させることを目的としている。また、様々な軌道に対してその制御を行ったときにも、しっかりと追従制御できるのか、制御のクオリティにムラがあるかないかを検証したい。

## 2 モデリング

車両のモデルには2輪幾何モデルを用いる。車両の中心（重心）を後輪中心とする。また、低速な状態しか扱わないので、滑りを考えない。車両の運動学は次のようなものになる [1]。

$$\dot{x} = v \cos \theta, \dot{y} = v \sin \theta$$

$$\dot{v} = a, \dot{\theta} = v \frac{\tan \delta}{W}$$

$v[m/s]$  は車両の後輪中心の速度、 $\theta[rad]$  は固定された座標系の  $x$  軸を基準とした車両の回転角度を表す。 $\delta[rad]$  は操舵角であり、ハンドル操作によって生まれる車軸の傾きである。 $a[m/s^2]$  は車両の加速度であり、どれだけアクセルを踏むかを表している。 $W[m]$  は車両の前輪の中心と後輪の中心を結ぶ長さ（ホイールベース）である。状態ベクトルを  $z = [x, y, v, \theta]^T$ 、入力ベクトルを  $u = [a, \delta]^T$  として、 $\dot{z} = f(z, u)$  と表す。制御対象の車両が追従する目標軌道は別の車両（以下、ターゲットと呼ぶ）が通った軌道であるとする。ターゲットの運動学も同様に、 $\dot{z}_r = f(z_r, u_r)$  と表す。 $\dot{z} = f(z, u)$  を  $(z_r, u_r)$  の近傍で、2次の項以降を近似的に0とみなしてテイラー展開して線形化を行い、

$$\begin{aligned} \dot{z} &= f(z_r, u_r) + \frac{\partial f(z, u)}{\partial z} (z - z_r) + \frac{\partial f(z, u)}{\partial u} (u - u_r) \\ &= f(z_r, u_r) + A'(z - z_r) + B'(u - u_r) \end{aligned} \quad (1)$$

を得る。さらに(1)式をサンプリングタイム  $T_s$  で離散化すると、最終的にシステムを次のような形で表すことができる [2]。

$$z_{k+1} = A_k z_k + B_k u_k + C_k \quad (2)$$

$$A_k = \begin{bmatrix} 1 & 0 & \cos \theta_{r,k} T_s & -v_{r,k} \sin \theta_{r,k} T_s \\ 0 & 1 & \sin \theta_{r,k} T_s & v_{r,k} \cos \theta_{r,k} T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{\tan \delta_{r,k}}{W} T_s & 1 \end{bmatrix}$$

$$B_k = \begin{bmatrix} 0 & 0 \\ T_s & 0 \\ 0 & \frac{v_{r,k}}{W \cos^2 \delta_{r,k}} T_s \end{bmatrix}$$

$$C_k = \begin{bmatrix} v_{r,k} \theta_{r,k} \sin \theta_{r,k} T_s \\ -v_{r,k} \theta_{r,k} \cos \theta_{r,k} T_s \\ 0 \\ \frac{v_{r,k} \delta_{r,k}}{W \cos^2 \delta_{r,k}} T_s \end{bmatrix}$$

$k$  は自然数であり、 $z(T_s), z(2T_s), \dots, z(kT_s)$  をそれぞれ  $z_1, z_2, \dots, z_k$  と定義する。同様に、 $u_k, z_{r,k}, u_{r,k}$  も定義する。

## 3 モデル予測制御について

本研究ではモデル予測制御を用いて軌道追従制御を行う。この論文では、receding horizon policy と呼ばれる方針を用いる [3]。(a) 制御を考える有限の時間区間を定め、目標値追従、システムの制約を考慮した制御問題を解き、得られた制御入力列の初期値を制御対象に加える。(b) 次のサンプル時刻が来たら、(a) の制御問題をシフトさせた時間区間で解く。つまり、時間が進むのに合わせて、終端時刻を後退させながら制御問題を解きなおすということである。そして、制御問題を解いて得られた入力列の先頭の値を制御対象に加える。この論文において、MPC コントローラで最小化するコスト関数は

$$\begin{aligned} & (z_{r,k+N} - z_{k+N})^T Q_f (z_{r,k+N} - z_{k+N}) \\ & + \sum_{j=1}^{N-1} (z_{r,k+j} - z_{k+j})^T Q (z_{r,k+j} - z_{k+j}) \\ & + \sum_{j=1}^N u_{k+j}^T R u_{k+j} \\ & + \sum_{j=1}^N (u_{k+j+1} - u_{k+j})^T R_d (u_{k+j+1} - u_{k+j}) \end{aligned} \quad (3)$$

となる。 $N$  は予測ステップ数である。また、 $z_{k+j} (j = 1, \dots, N), u_{k+j} (j = 1, \dots, N)$  は  $k$  時点から  $N$  ステップ先までの制御対象の振る舞いを考えて最適軌道を計算したことで現れる数値である。 $z_{r,k+j} (j = 1, \dots, N), u_{r,k+j} (j = 1, \dots, N)$  についても同様である。第1項と第2項は追従偏差についての式であり、第3項は制御入力の変化量についての式である。制約条件はつぎである。

$$\delta_{min} < \delta_k < \delta_{max}, v_{min} < v_k < v_{max} \quad (4)$$

$$u_{min} < u_k < u_{max}, z_0 = z_{ini} \quad (5)$$

(4) 式はステアリング角度と車両速度の制限、(5) 式はそれぞれ入力の制限と初期値の設定をしている。また、

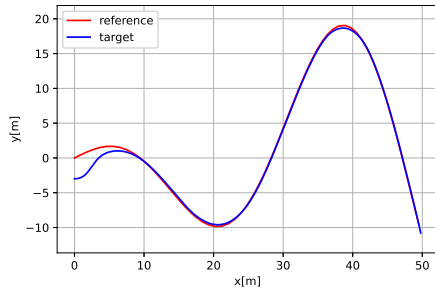


図1 1つ目の軌道に対する MPC

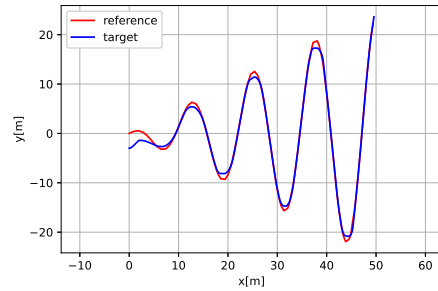


図3 2つ目の軌道に対する MPC (条件変更後)

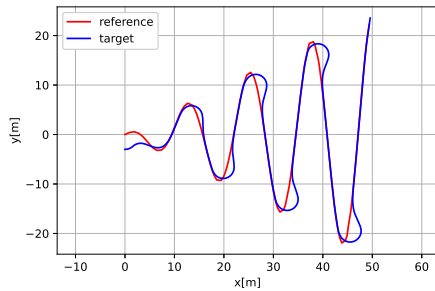


図2 2つ目の軌道に対する MPC

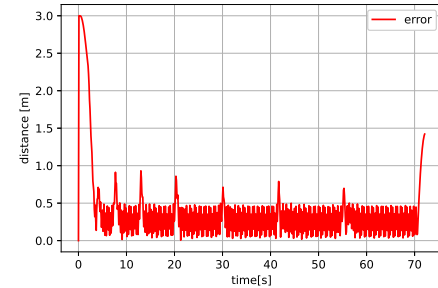


図4 2つ目の軌道に対する MPC の偏差 (条件変更後)

$Q_f, Q, R, R_d$  は重み行列であり、それぞれ半正定及び正定行列である。コスト関数 (3) 式を制約条件内で最小にするような入力  $u_k$  を求め、(2) 式によって次の状態が決まる。評価関数の形は 2 次形式のかたちをしており、二次計画法によって解くことができる。

#### 4 シミュレーション

本研究では、sin 関数や cos 関数を使い曲線を作り、それらを目標軌道とする。ここでは、様々な形の軌道に車両をモデル予測制御を用いて追従させ、その挙動を観察する。パラメータの値は  $W = 2.2$ 、 $N = 10$ 、 $v_{max} = 15.3$ 、 $v_{min} = 0.0$ 、 $\delta$  の上限が  $\frac{\pi}{4}$ 、下限を  $-\frac{\pi}{4}$  としている。また、プログラムはこのサイト [4] に書かれたコードを使用している。一つ目の軌道への MPC の結果を図 1 に示す。reference が目標軌道を表し、target が車両の軌道を表す。

同様に、2つ目の軌道への MPC による制御の結果を図 2 に示す。ここで、2つ目の軌道は1つ目の軌道よりもさらに急カーブ、180 度に近いような曲がり方ではどのような挙動になるのかを観察する意図で作った。このとき、180 度近い急カーブには対応できていないことが分かる。ここで、軌道そのままに制約条件を変えてシミュレーションを行う。 $\delta$  の上限と下限の値を  $\pm\frac{\pi}{4}$  としていたのを、上限、下限ともに  $\pm\frac{4\pi}{9}$  に変更した。これらの変更を加えたうえで、2つ目の軌道への MPC のシミュレーションを行い、結果を図 3 に示し、追従偏差のデータを図 4 に示す。これによって、実際にきれいに目標軌道に沿って追従できていることが分かる。

#### 5 おわりに

MPC では制約条件を変えることで車両が描く軌道がかなり変わってくることが分かった。たとえば、ステアリング角の制約条件値を変えることで、180 度近い急カーブにもきちんと追従できるようになった。しかし、実際の自動運転でステアリング角などの制約条件を  $\pm\frac{4\pi}{9}$  [rad] に設定して急カーブを曲がらせれば、とても無茶な挙動が出てしまい、乗り心地や安定性に欠けた状態になってしまう。今後の課題としては、モデル予測制御できちんと軌道追従制御しつつ乗り心地を高いレベルで安定させる研究などが挙げられる。

#### 参考文献

- [1] 佐藤瞳、村松鋭一「移動ロボットの軌道追従制御」計測自動制御学会東北支部第 275 回研究集会 (2012.10.26) 資料番号 275 10
- [2] 「Path Tracking」PythonRobotics github [https://pythonrobotics.readthedocs.io/en/latest/modules/path\\_tracking/path\\_tracking.html#model-predictive-speed-and-steering-control](https://pythonrobotics.readthedocs.io/en/latest/modules/path_tracking/path_tracking.html#model-predictive-speed-and-steering-control) (最終閲覧日 2022 年 3 月 4 日)
- [3] 児島 晃・大塚 敏之「モデル予測制御の考え方」計測と制御 第 42 巻 第 4 号 2003 年 4 月号 p p.310-312
- [4] Atsushi Sakai 「PythonRobotics」github <https://github.com/AtsushiSakai/PythonRobotics> (最終閲覧日 2022 年 3 月 3 日)