

# ロボカップサッカー小型リーグロボットに対するモデル予測制御の 設計パラメータと制御性能との関係

Relation between Control Performance and Tuning Parameters of Model Predictive Control for  
Robots in RoboCup Soccer Small Size League

日下部 宏幸・伊藤 正英・成瀬 正

Hiroyuki KUSAKABE, Masahide ITO, and Tadashi NARUSE

愛知県立大学 情報科学部

School of Information Science and Technology, Aichi Prefectural University

is121032@cis.aichi-pu.ac.jp, {masa-ito, naruse}@ist.aichi-pu.ac.jp

## Abstract

ロボカップサッカー小型リーグは現在、より動的なパワークへと発展する途上にあり、個々のロボットには高精度な運動制御が要求されている。予測を取り入れた高精度な制御手法の一つとしてモデル予測制御があるものの、一般に設計パラメータの調整指針が明らかでない。本稿では、小型リーグロボットの運動制御に対し後退ホライズン方策に基づくモデル予測制御器を設計し、数値実験に基づく検討によって、設計パラメータと制御性能との関係を明らかにした。

## 1 背景と目的

ロボカップサッカー小型リーグでは、近年の活発な研究・開発の結果、ロボットは高速で正確な動きが可能となり、戦略もますます高度化している。特に、複数台ロボットによる連携プレーに着目すると、各ロボットが受け身でボールを回すというこれまでの主流から、動きながらのボール回しに現在シフトしつつある。この発展にはより高精度な運動制御が要求される。

モデルに基づく予測を取り入れた制御手法として、モデル予測制御 (Model Predictive Control; MPC) [1] が注目されている。ロボカップサッカーでは小型リーグと中型リーグにおいてモデル予測制御の適用例がある [2, 3]。それらの適用例では、従来の制御手法と比べて制御性能が向上したという結果は示されているものの、設計パラメータと制御性能との関係性が明らかになっていない。そこで本研究では小型リーグで用いられている車輪型全方向移動ロボットを対象として、設計した MPC 制御器のパラメータと制御性能との関係を明らかにすることを目的とする。

## 2 小型リーグシステムの概要

小型リーグでは一般的に、オムニホイール 4 輪から構成される全方向移動ロボットが用いられる。ロボットやボールの幾何情報は、各フィールドの全体を俯瞰するように取り付けられた 4 つのカメラによって観測される。一方、ロボットの動きはホスト PC から無線通信によって指令される。

カメラから得られた画像データは SSL-Vision サーバにおいて適切に画像処理され、各ロボットの位置と姿勢、ボール位置などが得られる。それらの情報は各チームのホスト PC へ配信され、試合状況の分析や戦略の決定が行われたのち、ロボット個々の目標軌道が計画される。各ロボットには目標軌道に基づいた速度指令値が無線通信によって送信され、ロボット内部では指令値に対するフィードバック制御が行われる。なお、ホスト PC で速度指令値が送信されてから実際のロボットがそれに応じた動作が開始するまでには、一定のむだ時間が生じる。

## 3 車輪型全方向移動ロボットの運動モデル

車輪型全方向移動ロボットの速度に基づく運動モデルは連続時間表現として次式で与えられる [2]。

$$\begin{cases} \dot{x} = \alpha_x v_x(t - t_0) \\ \dot{y} = \alpha_y v_y(t - t_0) \\ \dot{\theta} = \alpha_\omega \omega(t - t_0) \end{cases} \quad (1)$$

ただし、 $(x, y, \theta)$  はロボットの位置・姿勢、 $(v_x, v_y, \omega)$  は各速度を表している。 $t_0$  は通信によるむだ時間を表しており、入力の影響が実際にロボットの出力に表れるのは時刻  $t_0$  後ということの意味する。速度のスケールパラメータ  $(\alpha_x, \alpha_y, \alpha_\omega)$  はロボットを実際に走行させ、入力として指令した速度と発生した速度との比較から同定することができる。状態ベクトルを  $\mathbf{z} = (x, y, \theta)^T$ 、入力ベクトルを  $\mathbf{v} = (v_x, v_y, \omega)^T$  とすると式 (1) は次式のように表現し

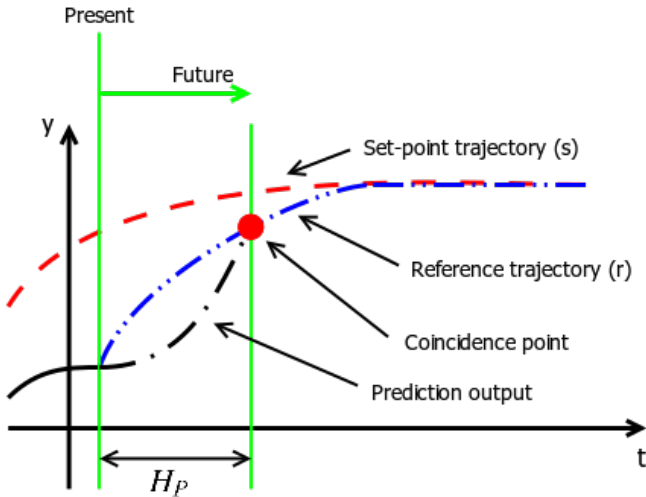


Figure 2: Parameters of MPC

直せる。

$$\dot{z} = \alpha v(t - t_0) \quad (2)$$

#### 4 モデル予測制御

近年注目されている MPC は、制御対象のモデルを用いて将来の動きを予測し、制御入力を決める制御方法である。基本的な考え方は、有限の未来の時間区間の動きを予測し、その動きを最適化するように現在の入力を決定するというものである。また、外乱補償器の付加が容易で、むだ時間やモデル誤差にロバストな制御系を構成することができる。

本稿では、離散時間モデルに基づく最も基本的な MPC として後退ホライズン方策 [1] による制御器を設計する。Figure 1 にブロック線図を示す。

##### 4.1 MPC の設計パラメータ

MPC で用いる 2 種類の軌道と設計パラメータとを Figure 2 に示す。設定値軌道 (set-point trajectory) は、ロボットに与える最終的な所望軌道である。一方、参照軌道 (reference trajectory) は現在の制御対象の出力から設定値軌道に向かうまでの軌道である。参照軌道と制御対象の予測出力が一致する点を一致点 (coincidence point; CP) といい、MPC 制御器はこの一致点に向かうような入力を生成する。MPC 制御器の性能は、以下の設計パラメータに依存する。

- $T_{ref}$  : 設定値軌道に対する参照軌道の時定数。
- $H_p$  : 予測ステップ数の長さ。何ステップ先まで予測を行うか。これは許容される計算時間に依存する。
- 入力変化の数 :  $H_p$  ステップ間で入力が何ステップ変化するか。

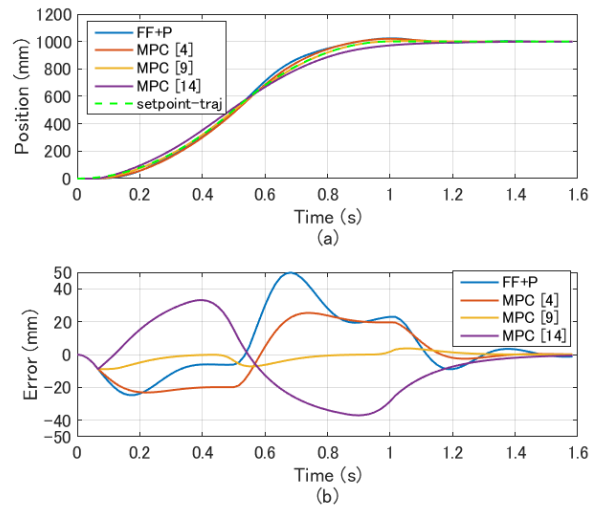


Figure 3: Comparison between FF+P and MPC

- 一致点の数 :  $H_p$  ステップ間で制御対象の予測出力と参照軌道との一致点をいくつ持つか。

##### 4.2 MPC で使用するモデル

MPC 制御器内部には制御対象の出力を予測するためにモデルが必要になる。制御対象としているロボットには第 3 節に述べたとおりむだ時間  $t_0$  が存在するが、これはモデル化誤差と考える。つまり、式 (2) においてむだ時間  $t_0$  を  $t_0 = 0$  としたものをモデルとして考えることにする。

式 (2) の連続時間モデルを離散化すると、

$$z(k+1) = z(k) + \alpha T_s v(k) \quad (3)$$

となる。ただし、サンプリング周期を  $T_s$ 、離散時間を  $k$  と表している。

##### 4.3 設計した制御器の性能

設計した制御器の性能を数値実験により示す。ここでは、 $x(0) = 0 \text{ mm}$  から  $x(t_f) = 1000 \text{ mm}$  までの移動を考え、この 2 点間をつなぐ多項式軌道を設定値軌道とする。ただし、 $t_f$  は理想的な移動し終える時間とする。 $t_f = 1.0 \text{ s}$ ,  $T_{ref} = 5/60 \text{ s}$ ,  $H_p = 4, 9, 14$ , 入力変化の数を 1、一致点の数を 1 としたときの結果を Figure 3 に示す。

比較のため、従来制御手法の一つとして、フィードフォワード補償器つき比例 (FF+P) 制御による結果も合わせて示してある。Figure 3(a) はロボットの実際の出力と設定値軌道の時間による変化、(b) は設定値軌道と実際の出力との誤差を示している。

Figure 3(b) から  $H_p$  の値が大きくなると誤差は小さくなっているが、 $H_p = 14$  の場合は逆に誤差が増えることがわかる。したがって、MPC には最適な予測区間が存在すると思われる。

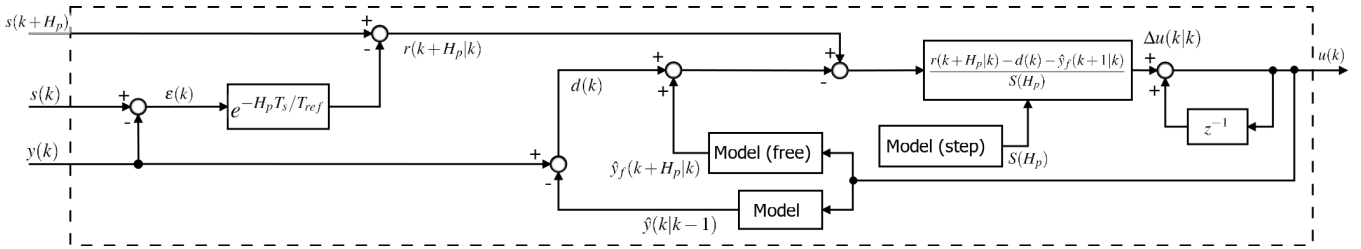


Figure 1: Block diagram of MPC

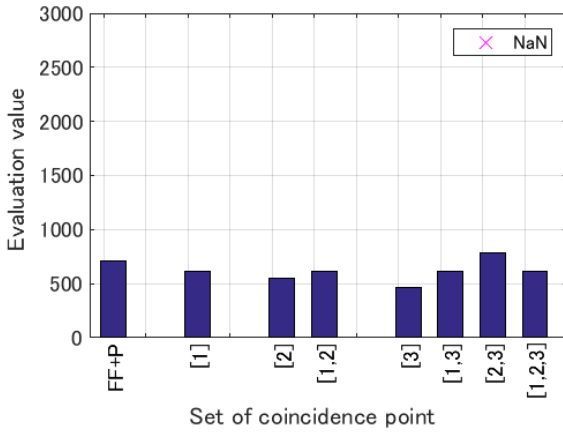


Figure 4: Performance with  $H_p = 1, 2, 3$  steps

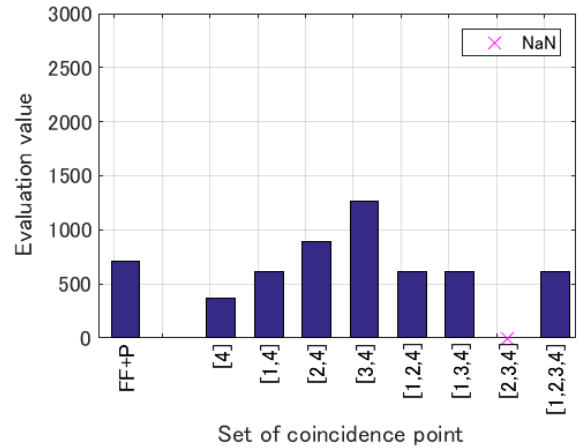


Figure 5: Performance with  $H_p = 4$  steps

## 5 設計パラメータと制御性能との関係

第 4.3 節と同じ問題設定のもと MPC 制御器の設計パラメータと制御性能との関係を明らかにする。評価値として、ロボットの実際の軌道と設定値軌道との差の二乗和を移動にかかる時間ステップ数で規格化したものを考える。誤差の二乗和を評価値としているので評価値が小さいほどよい制御性能が得られていることを意味する。

式 (1) において、むだ時間  $t_0$  の値は文献 [4] の推定法により  $t_0 = 0.07206$  と求まった。また、スケールパラメータ  $\alpha$  は予備的な実験により  $\alpha = 0.9$  とした。

本節では一致点の数と予測区間中に許される入力変化の数とが等しい場合に限定して、1) 一致点のパターンと制御性能との関係、2) 予測区間の長さとの関係、3) 時定数と制御性能との関係に着目する。なお、本節では一致点のパターンの表記について次のように定義する。

### 一致点の表記

ある時刻  $k$  において、予測区間が  $N_n$  ステップ。一致点の時刻が  $k + N_1, k + N_2, \dots, k + N_n$  であるとき、一致点の集合を  $[N_1, N_2, \dots, N_n]$  で表す。

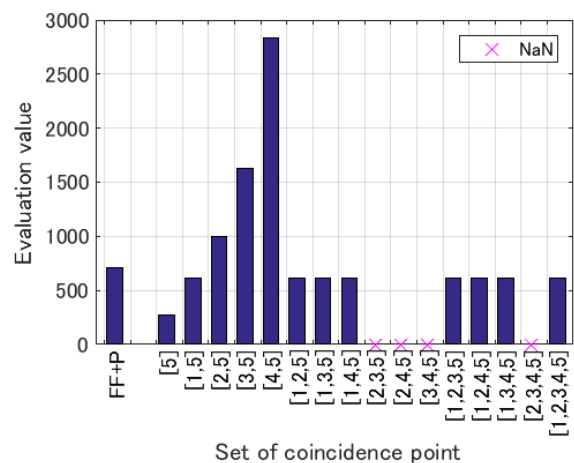


Figure 6: Performance with  $H_p = 5$  steps

### 5.1 一致点のパターンと制御性能との関係

予測区間の長さを5ステップまでとし、その中で考えられる一致点のすべてのパターンに対し、評価を行った。結果を Figures 4~6 に示す。図中の「NaN」は制御で入力を求めるための行列が特異となり、値が不定であることを示している。Figures 4~6 から次の2点がいえる。

- (A) : 一致点の集合に [1] を含む場合、評価値はすべて同じである。
- (B) : 一致点が複数個あり、一致点の集合に [1] を含まない場合、最初の一致点の位置に比例して評価値が大きくなる。

(A) に関して、一致点の集合が [1] と [1,3] の場合を例として考える。このときの入力生成の様子を Figure 7 に示す。Figure 7 のように、MPC では設定値軌道に収束するような参照軌道が時刻  $k$  において生成されるが、設定値軌道と制御対象の出力が等しければ一致点の集合によらず、生成される参照軌道は同じである。参照軌道は同じであるから、一致点の位置が同じであれば Figure 7 のように1ステップ目の入力は一致する。よって、入力が一致するので制御対象の出力も一致し、誤差も等しくなる。Figure 8 に一致点が [1] と [1,3] の入力と出力の変化を示す。Figure 8 から入力と出力が一致しているのが分かる。

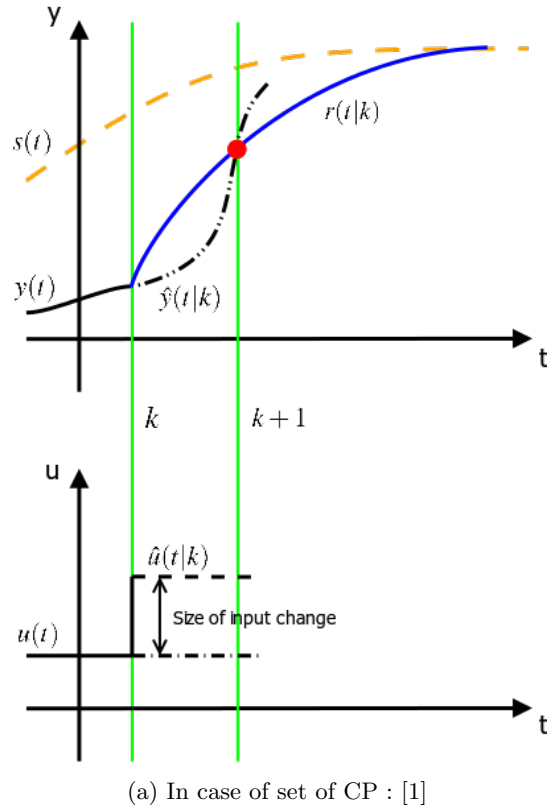
(B) は、例として一致点が [3,4] の場合を考える。このときの MPC 制御器による入力生成は Figure 9 のようになると考えられる。一致点の集合が [3,4] の場合、3, 4 ステップ目で参照軌道と制御対象の予測出力が一致するよう、最初の2ステップが変化する入力が生成される。しかし、制御対象に実際印加される入力は1段階目の入力だけである。その結果、Figure 9 のように1段階目の入力のみによる出力と1段階目と2段階目の入力による出力では差が生じる。1段階目の入力のみでは一致点で参照軌道と制御対象の予測出力は一致しないため誤差が生じたと考えられる。最初の一致点の位置が伸びるほど誤差が大きくなるのは1段階目の入力のみによる出力によって生じる誤差がステップが進むにつれ大きくなるからだろうと思われる<sup>1</sup>。

予測区間の長さごとに誤差を見ると、どの予測区間の長さにおいても一致点の数が1つの場合が最も評価値が小さくなる結果となった。

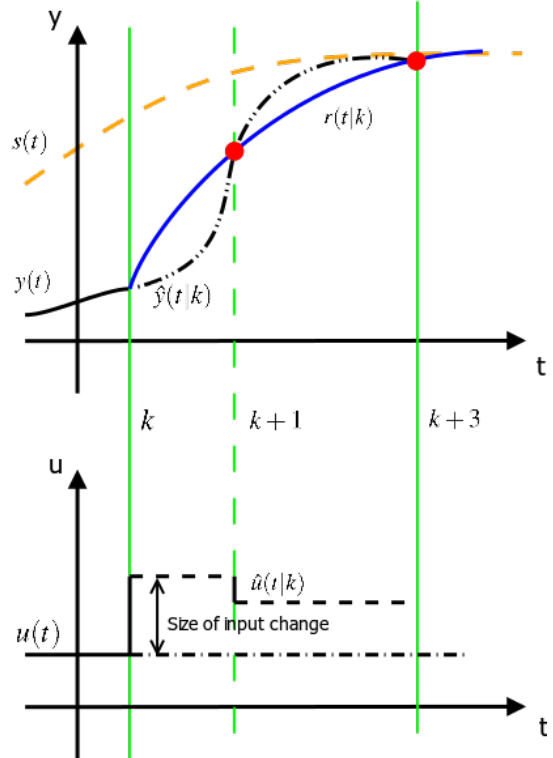
### 5.2 予測区間の長さとの関係

一致点の数を一つに限定して予測区間の長さについて評価を行った結果を Figure 10 に示す。Figure 中から予測区間の長さが9ステップまでは評価値が単調減少する一方、10ステップ以降は評価値が大きくなる結果になった。

<sup>1</sup>Figure 9 の時刻  $k+3$  と  $k+4$  では後者のほうが誤差が大きい。



(a) In case of set of CP : [1]



(b) In case of set of CPs : [1,3]

Figure 7: MPC with set of CPs [1] and [1,3]

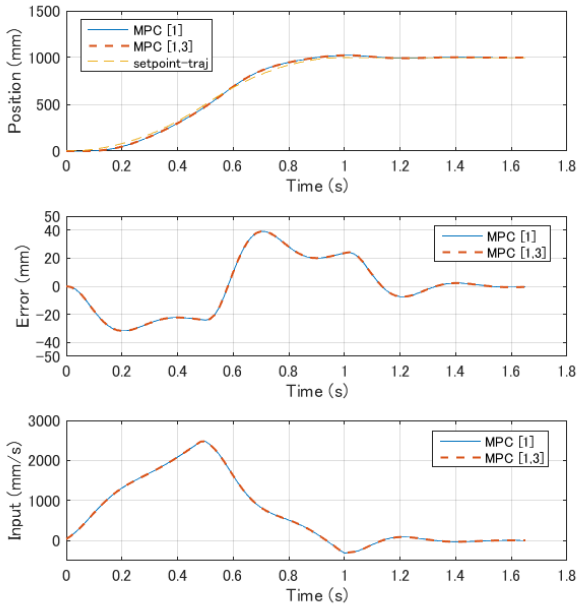


Figure 8: MPC with set of CPs : [1] vs [1,3]

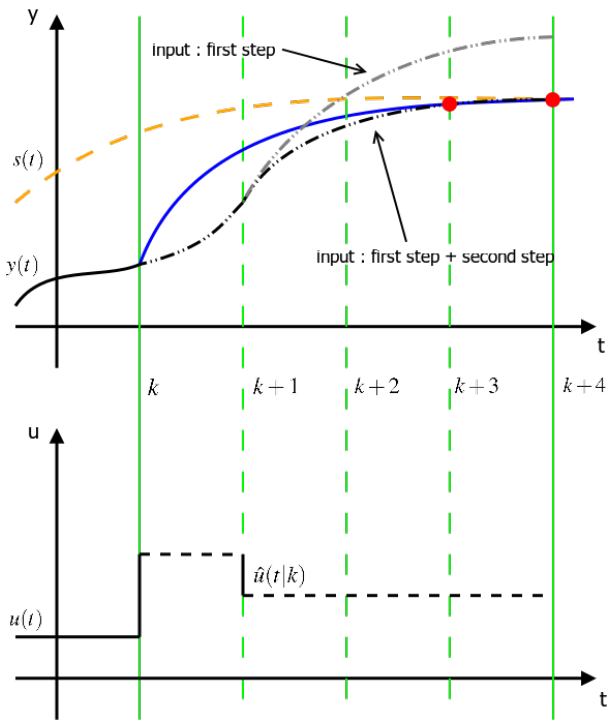


Figure 9: MPC with set of CPs [3,4]

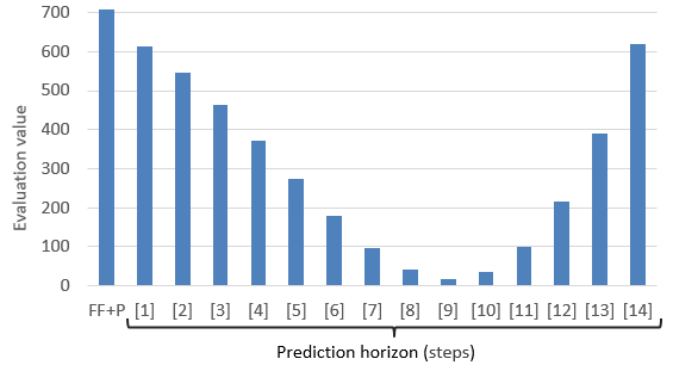


Figure 10: Performance w.r.t. various prediction horizon

この誤差は以下に示す二つの要因から生じるものだと分析することができる (Figure 11(a))。第一の要因は、設定値軌道  $s$  と参照軌道  $r$  から生じる誤差  $r - s$  である。参照軌道が設定値軌道に収束する速さは MPC の設計パラメータの一つである時定数によって決まる。時定数の値が大きいと参照軌道が設定値軌道へ収束するのに時間がかかるため、予測区間の終点では参照軌道と設定値軌道が一致せずその差が誤差となって生じてしまう。第二の要因は実際の出力  $y_p$  と参照軌道  $r$  から生じる誤差  $y_p - r$  である。この差はむだ時間などモデル化誤差が影響に起因する。

これら二つの誤差を予測区間の長さごとに示したのが Figure 11(b) である。Figure 11(b) から 2 つの誤差の合計が最も少ない予測区間の長さ 9 のときが Figure 10 で最も誤差が少ない場合と一致しているのがわかる。また、誤差  $r - s$  は予測区間の長さが伸びれば参照軌道は設定値軌道に収束していくので、予測区間が長くなるほど誤差が少なくなっていることがわかる。誤差  $y_p - r$  について見ると、予測区間の長さが 10 以降で急激に誤差が増えていることがわかるが、これは一致点を予測区間の終点の一点のみとしているため、予測区間中の軌道追従については考慮されず、そこから誤差が生じるためである。

### 5.3 時定数と制御性能との比較

予測区間の長さを 1 ステップとし、時定数を変化させて評価した結果が Figure 12 である。Figure 12 から評価値は時定数が小さくなるほど減少する傾向にあるが、 $T_{ref} = 0.02s$  では大幅に増加している。このとき MPC 制御器は制御対象を制御できておらず出力が発散振動をしていた。この結果から制御対象の特性に依存して臨界点が決まっていると考えられる。しかしながら、具体的にどんな点が原因となっているかについては今後、検討を進める必要がある。

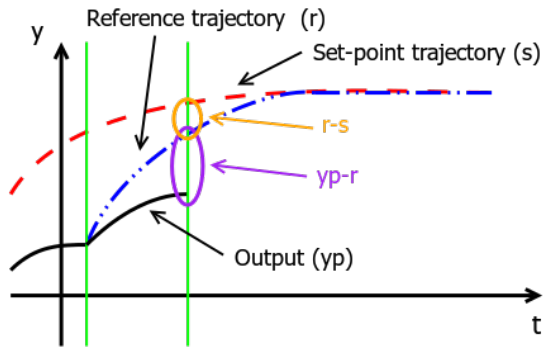
## 6 おわりに

本稿では、設計した MPC 制御器に対して様々な設計パラメータを与え数値実験を行ったことによって、本稿で考えた運動モデルとロボットに対して、最適な予測区間が存在するなどの設計パラメータと制御性能との関係を示すことができた。また、従来の制御手法と比べて軌道追従誤差が減るなどのより高精度な制御が可能であることを示すことができた。これによって従来より高精度なボールのパス回しが可能となるだろう。

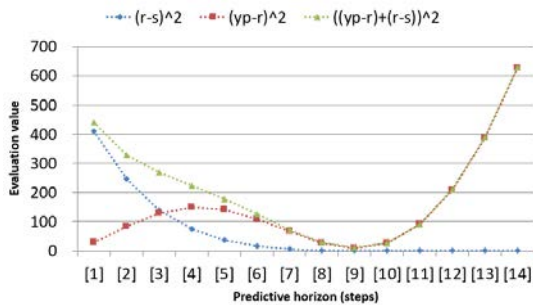
今後は、実機システムへの実装、パス精度での評価などが課題である。

### 参考文献

- [1] J. M. Maciejowski : “モデル予測制御 — 制約のもとでの最適制御—”, 足立・菅野 訳, 東京電機大学出版局, 2005.
- [2] M. Zarghami, A. Fakharian, A. Ganjali Poudeh and A. Adhami-Mirhosseini: “Fast and precise positioning of wheeled omni-directional robot with input delay using model-based predictive control,” Proceedings of the 33rd Chinese Control Conference (CCC'14), pp.7800–7804, 2014.
- [3] K. Kanjanawanishkul and A. Zell: “Path following for an omnidirectional mobile robot based on model predictive control,” Proceedings of IEEE International Conference on Robotics and Automation (ICRA'09), pp.3341–3346, 2009.
- [4] Y. Wu, X. Qiu, G. Yu, J. Chen, X. Rie, Y. Wu, and R. Xiong: “Extended TDP of ZjuNlict 2009,” Proceedings of RoboCup2009, pp.14–16, 2009.



(a) Causes of tracking error



(b) Decomposition of tracking error

Figure 11: Analysis of tracking error

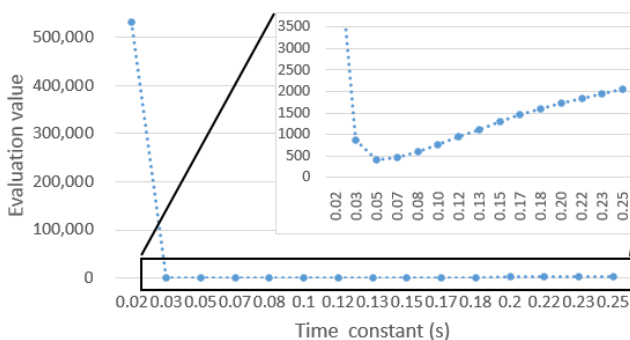


Figure 12: Performance w.r.t. various time constants