

# モデル予測制御の実システムへの応用 に関する研究

2014 年 11 月

中村 亘

モデル予測制御の実システムへの応用  
に関する研究

中村 亘

システム情報工学研究科  
筑波大学

2014 年 11 月

# 論文要旨

本研究は、モデル予測制御が未だ十分に活用されていないような実システムの制御問題に対する応用について述べる。モデル予測制御の応用によりそれらの制御問題を解決することにより、モデル予測制御の実用性をより高めることを目指す。まず、複数台の移動ロボットの走行制御問題を取り上げ、従来研究よりも厳しい制約条件を課した制御問題に対しモデル予測制御を用いた制御法を提案する。次に、モデル予測制御の導入が期待されている熱伝導システムの温度制御問題を取り上げ、モデル予測制御に不可欠な、同プラントの挙動を精度よく表現した一般的な制御系設計モデルの導出法について示す。最後に、従来より用いられてきた PID 制御にモデル予測制御の特長を付与することで、低コストに制御性能の向上を目指したモデル予測型 PID 制御について示す。

# 目次

論文要旨	i
第 1 章 序論	1
1.1 本論文の背景と目的	1
1.1 本論文の構成	2
第 2 章 モデル予測制御概説	3
2.1 モデル予測制御について	3
2.2 モデル予測制御アルゴリズム	4
2.3 モデル予測制御の一般的な定式化	5
第 3 章 移動ロボットによる協調運搬作業	10
3.1 緒言	10
3.1.1 背景と目的	10
3.1.2 本章の構成	11
3.2 移動ロボットによる協調運搬作業問題	11
3.2.1 制御対象	11
3.2.2 問題設定	14
3.3 協調運搬作業のための制御手法	15
3.3.1 目標値の生成	15
3.3.2 制約の変換	18
3.3.3 制約の幾何的意味	21
3.3.4 モデル予測制御による制御方式	21
3.4 数値例	23
3.4.1 直線走行路	25

3.4.2	円弧走行路 . . . . .	27
3.4.3	考察 . . . . .	28
3.5	結言 . . . . .	29
<b>第 4 章</b>	<b>熱伝導システムの制御系設計モデル導出</b>	<b>31</b>
4.1	緒言 . . . . .	31
4.1.1	背景と目的 . . . . .	31
4.1.2	本章の構成 . . . . .	33
4.2	問題設定 . . . . .	33
4.3	制御系設計モデルの導出手法 . . . . .	34
4.4	数値例 . . . . .	39
4.4.1	単入力システム (数値例 1) . . . . .	40
4.4.2	多入力システム (数値例 2) . . . . .	44
4.5	結言 . . . . .	46
<b>第 5 章</b>	<b>モデル予測型 PID 制御</b>	<b>49</b>
5.1	緒言 . . . . .	49
5.2	PID 制御について . . . . .	51
5.2.1	PID 制御 . . . . .	51
5.2.2	2 自由度 PID 制御 . . . . .	52
5.3	熱伝導システムのモデル . . . . .	54
5.3.1	線形差分式モデル . . . . .	54
5.3.2	制約 . . . . .	55
5.4	モデル予測型 PID 制御 . . . . .	55
5.4.1	制御系設計モデル . . . . .	56
5.4.2	MP-PID 制御系設計 . . . . .	57
5.5	数値例 . . . . .	60
5.5.1	熱伝導システムに対する適用結果 . . . . .	60
5.5.2	協調運搬作業問題に対する適用結果 . . . . .	71
5.6	モデル予測型 2 自由度 PID(MP-2DOF PID) 制御への拡張 . . . . .	73
5.7	結言 . . . . .	77
<b>第 6 章</b>	<b>結論</b>	<b>79</b>

謝辭	80
参考文献	82
研究業績一覽	86

# 目次

2.1	モデル予測制御の考え方 . . . . .	5
3.1	2 輪駆動型移動ロボット . . . . .	12
3.2	Leader と Follower . . . . .	14
3.3	極座標モデル . . . . .	16
3.4	変換前 . . . . .	18
3.5	変換後 . . . . .	18
3.6	並進速度制約の考え方 . . . . .	19
3.7	回転角速度制約の考え方 . . . . .	21
3.8	移動ロボットの到達可能領域 . . . . .	22
3.9	制御の流れ . . . . .	22
3.10	Leader の走行路 . . . . .	25
3.11	直線走行路の結果 (1) . . . . .	26
3.12	astolfi の制御則による直線走行路の結果 . . . . .	26
3.13	最適レギュレータによる直線走行路の結果 . . . . .	27
3.14	直線走行路の結果 (2) . . . . .	27
3.15	円弧走行路の結果 (1) . . . . .	28
3.16	円弧走行路の結果 (2) . . . . .	29
4.1	入力和の様子 . . . . .	35
4.2	線形補間の様子 . . . . .	37
4.3	プリンタの模式図 . . . . .	40
4.4	温度および入力の応答比較 (差分式モデルと制御系設計モデル) . . . . .	44
4.5	温度および入力の応答比較 (PID 制御とモデル予測制御) . . . . .	45
4.6	温度および入力の応答比較 (モデル予測制御：制約なし) . . . . .	47

4.7	温度および入力の応答比較 (モデル予測制御：制約あり) . . . . .	48
5.1	フィードバック制御系 . . . . .	52
5.2	フィードフォワード制御系 . . . . .	52
5.3	2 自由度制御系 . . . . .	53
5.4	プラントインデックス $i$ の時間変化 . . . . .	64
5.5	Case 1 での $y$ の時間応答 (上) と $y_M$ の時間応答 (下) . . . . .	66
5.6	Case 1 での入力 $u$ の時間応答 . . . . .	67
5.7	Case 2 での $y$ の時間応答 (上) と $y_M$ の時間応答 (下) . . . . .	68
5.8	Case 3 での $y$ の時間応答 (上) と $y_M$ の時間応答 (下) . . . . .	69
5.9	直線走行路におけるモデル予測追従制御と MP-PID 制御の比較 . . . . .	73
5.10	制御性能の比較 . . . . .	74
5.11	プラントインデックスの時間変化 . . . . .	77



# 表目次

3.1	移動ロボットのパラメータ . . . . .	13
3.2	シミュレーション環境 . . . . .	24
5.1	Case 1 における制御性能評価 . . . . .	65
5.2	Case 2 における制御性能評価 . . . . .	67
5.3	Case 3 における制御性能評価 . . . . .	70
5.4	MP-2DOF PID 制御の制御性能評価 . . . . .	75

# 第1章

## 序論

### 1.1 本論文の背景と目的

本研究では，モデル予測制御が従前には十分に活用されていない実システムの制御問題に対し，実用性をより高めることを目指した同制御の応用について述べる．

社会に存在している実システムには，その制御対象特性や制御目的に関係するさまざまな制約条件が存在する．例えば，システムへの物理的入力が発生させるアクチュエータの性能面や電力など制御にかかるコスト面に関する制約，あるいはシステムの安全性確保の面からの制約など様々な側面からの制約がある．また，システムによっては入力量が制御器の構造に依存して決定することから，既存システムの制御器構造も考慮する必要がある．このような点に配慮せずに制御を行うことは，システムの性能劣化や故障といったことを生じかねない．したがって，システムのもつ制約を満たしつつ諸性能の向上をはかる実用的制御手法の検討が求められている．

ここで，制約を持つシステムの制御手法として，モデル予測制御 [1] が高い注目を集めている．モデル予測制御とは，制御対象のダイナミクスを表現した制御系設計モデルを用いて現在時刻から有限長の未来にわたる区間における制御量の将来挙動を予測し，その挙動が目標軌道に近づき追従していくための最適入力を求めることを，制御周期ごと繰り返す制御手法である．最適入力を求めることは，制約付き最適化問題を解くことに帰着される．近年の計算機性能の発展に伴って，例えば機械系や電気系・メカトロニクス系といった動特性が高速に変化する対象などに対しても利用が広がっている．しかし，まだまだモデル予測制御が活用されていない対象や，あるいは現在その導入が期待されている対象が存在する．そのため，それぞれの制御対象やその制御目的など，それぞれの制御テーマに対し，(i) 制約条件を考慮できる，(ii) 時々刻々と最適な値を算出できるというモデル予

測制御の特長を活かした実用的制御手法の研究が重要となる。

本研究では、まず、モデル予測制御が十全に活用されていない対象として複数台の移動ロボットの走行制御問題を取り上げる。この問題には、先に述べるようなモデル予測制御の応用研究自体は存在するものの、より実用的な制御手法とするには移動ロボット間の相対距離制約に考慮する必要がある、まだ拡張の余地があるといえる。次に、モデル予測制御の導入が期待されている対象として熱伝導プラントの温度制御問題を取り上げる。モデル予測制御の適用に必要となる、同プラントの挙動を精度よく予測できる一般的な制御系設計モデルの導出について検討する。最後に、従来より用いられてきた PID 制御 [2] にモデル予測制御の特長を付与することで、低コストに制御性能を向上させることを目指したモデル予測型 PID 制御について示す。

## 1.1 本論文の構成

本論文の構成は以下の通りである。まず、2 章にてモデル予測制御について概説する。3 章では、移動ロボットの協調運搬作業を取り上げ、この問題に対して相対距離制約を考慮したモデル予測制御による制御手法について示す。4 章では、熱伝導システムに対するモデル予測制御導入のために、精度の良い制御系設計モデルの導出手法について述べる。5 章では、モデル予測制御の特長を活かして PID 制御の性能を向上させるモデル予測型 PID 制御について示す。最後に 6 章にて、本研究についてまとめを述べる。

## 第2章

# モデル予測制御概説

### 2.1 モデル予測制御について

モデル予測制御は 1970 年代後半に，Richalet らによって開発された ID-COM(Identification and Command) と Shell Oil の Cutler らによる DMC(Dynamic Matrix Control) と呼ばれる手法から始まり，石油化学産業等のプロセス産業界において最も適用されてきた．その理由のひとつとして，これらの手法はインパルス応答モデルやステップ応答モデルといった技術者に理解しやすい簡単なモデルでアルゴリズムが記述されていたことがある [3]．また他の理由として，プロセス制御においては制御周期が比較的長く，最適な入力を決断するためのオンライン計算に十分な時間がとれたことも挙げられ，現在では，計算機の高性能化に伴って，適用の範囲が大きく広がっている．

現在まで様々な理論・アルゴリズムが研究されているが，多入力多出力系を容易に扱えるという現代制御理論の利点をモデル予測制御の枠組みに取り入れるために，状態空間モデルが用いられるようになっている [4]．なお，その他には，一般的な伝達関数モデルを用いてモデル予測制御を構成する一般化予測制御と呼ばれる手法もあり，この手法は PID 制御ケインの調整への適用も研究されている [5]．モデル予測制御の大きな特長の一つが制約条件を考慮して入力を決定する点であり，いわゆる最適制御と大きく異なる点である．この場合，モデル予測制御は制約付き最適化問題として定式化される [6]．更に近年では，線形システムに限らず非線形なシステムに対する非線形モデル予測制御 [7] や，状態空間モデルのパラメータの不確かさを考慮するロバストモデル予測制御 [8]，連続的なダイナミクスに例えばスイッチの切替のような離散的現象を統合したハイブリッドシステムに対するモデル予測制御などが研究されている [9, 10]．

他制御との関連について，先に述べた一般化予測制御と PID 制御だけでなく，異なる

研究も行われている。PID 制御との関連では、モデル予測制御の時々刻々と最適な値を算出することを繰り返すというアルゴリズムを活用するモデル予測型 PID 制御があり、5 章にて述べる。また、スライディングモード制御 (SMC: Sliding Mode Control) に対する研究も行われている。SMC はモデル化誤差や外乱などシステムの不確かさに対し、ロバスト性を考慮した制御手法であり [11, 12, 13]、実用性が高い制御手法として近年産業応用が進んでいる。ここで、SMC の制御性能を高めるため、モデル予測制御の考え方を利用したパラメータ決定手法が提案されている [14, 15, 16]。[14, 15] は、離散時間非線形 MPC を用いスライディングサーフェスのパラメータを動的に決めるもので、[16] では、制約条件付き最適問題を解きながら切り替え関数を予測することで最適な入力を決定するものである。

## 2.2 モデル予測制御アルゴリズム

あらためて、モデル予測制御とは『制御対象のモデルを用いて将来の挙動を予測し、制約条件下において有限区間の最適化問題を解くことで、その挙動が出来るだけ望ましい挙動となるような入力を決定する手続きを、サンプリング時刻ごとに繰り返す制御手法』である。これを図 2.1 に示す。

$k$  を現在の時刻ステップとする。制御の目的は出力  $y[k]$  を目標値  $y_{\text{ref}}$  に収束させることである。コントローラは制御対象のモデルを有しており、 $H_p$  を予測区間とし、モデルを用いて  $k$  から  $k + H_p$  における出力の挙動  $\hat{y}[k + i]$ ,  $1 \leq i \leq H_p$  を予測する。この挙動は未来の入力に依存するものであるから、予測される出力が目標値  $y_{\text{ref}}$  に適切に近づくように、 $H_u$  を制御区間とした  $k$  から  $k + H_u$  にわたる未来の入力列  $\hat{u}[k + i]$ ,  $0 \leq i \leq H_u - 1$  を決定する。こうして得られた入力列のうち、実際の入力として用いられるのは図 2.1(b) に示すように最初の  $\hat{u}[k]$  のみである。

モデル予測制御の制御アルゴリズムを以下に示す。

- (step1) 現在の時刻ステップを  $k$  とし、状態変数  $x[k]$  の値を取得する。
- (step2) 有限の予測区間における、ある評価関数を最小にする未来の入力列  $\hat{u}[k + 0], \dots, \hat{u}[k + H_u - 1]$  を求める最適化計算を行う。
- (step3) (step2) で得られた入力列のうち、1 つ目の  $\hat{u}[k + 0]$  のみを現在時刻での入力として次のサンプリング時刻  $k + 1$  まで制御対象に加え、(step1) へ戻る。

このように、モデル予測制御は予測区間を 1 ステップずつずらしながら最適化計算を

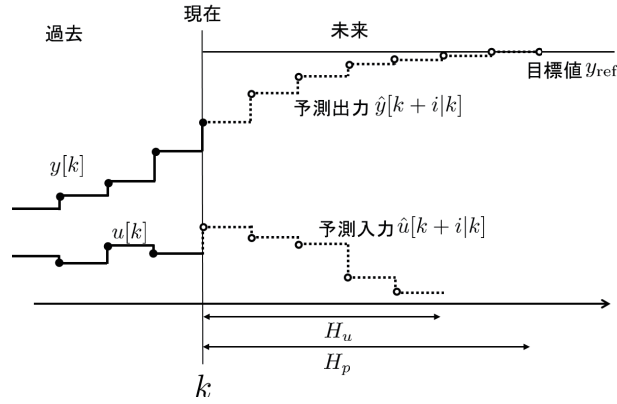
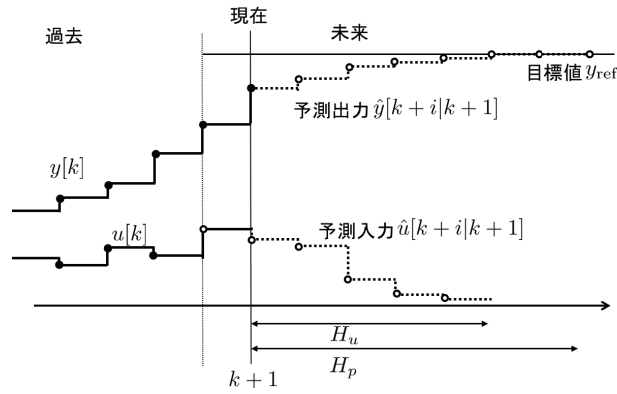
(a) 時刻  $k$  の時(b) 時刻  $k+1$  の時

図 2.1 モデル予測制御の考え方

行うことを繰り返す．これを Receding Horizon 方策，あるいは Moving Horizon 方策という．

## 2.3 モデル予測制御の一般的な定式化

ここではモデル予測制御の解法について説明する．制御対象が以下の離散時間系の状態空間モデルで表されているとする． $\mathbf{x} \in \mathbb{R}^{n_x}$ ,  $\mathbf{u} \in \mathbb{R}^{n_u}$ ,  $\mathbf{y} \in \mathbb{R}^{n_y}$  はそれぞれ制御対象の状態変数，制御入力，出力を示す． $A, B, C$  はシステムの特性を表すそれぞれ適当な大きさ

の行列である.

$$\mathbf{x}[k+1] = A\mathbf{x}[k] + B\mathbf{u}[k] \quad (2.1)$$

$$\mathbf{y}[k] = C\mathbf{x}[k] \quad (2.2)$$

時刻ステップ  $k$  における評価関数  $J[k]$  を以下の式で与える.  $\hat{\mathbf{x}}[k+i|k]$ ,  $\hat{\mathbf{u}}[k+i|k]$ ,  $\hat{\mathbf{y}}[k+i|k]$  といった表記は時刻  $k$  における未来の時刻  $k+i$  の予測値ということを表す.

$$J[k] = \sum_{i=1}^{H_p} \|\hat{\mathbf{y}}[k+i|k] - \mathbf{y}_{\text{ref}}[k+i|k]\|_{W_y}^2 + \sum_{i=0}^{H_u-1} \|\hat{\mathbf{u}}[k+i|k]\|_{W_u}^2 \quad (2.3)$$

ここで  $\|\mathbf{z}\|_P^2$  は  $\mathbf{z}^T P \mathbf{z}$  という二次形式を表す.

この評価関数の第 1 項は, 予測区間  $H_p$  にわたって予測出力  $\hat{\mathbf{y}}[k+i|k]$  と目標値軌道  $\mathbf{y}_{\text{ref}}[k+i|k]$  との偏差に重み  $W_y$  をかけたものである. この値を小さくすることは, 予測出力が目標値に近づく, すなわち制御目的が満たそうとすることを示す. また第 2 項は, 制御区間  $H_u$  における未来の入力  $\hat{\mathbf{u}}[k+i|k]$  の大きさに重み  $W_u$  をかけたものである. この項を小さくすることは, より小さなエネルギーで制御を行うことを意味する. したがって, モデル予測制御における最適入力とは, 予測出力を目標値へと制御することと, そのための制御にかかるコスト間のトレードオフの中で, 設計者が定めた評価指標において「最適」である入力を意味する.

以下では簡略化のため  $\mathbf{y}[k] = \mathbf{x}[k]$  (すなわち  $C = I$ ), 目標値  $\mathbf{y}_r[k+i|k] = 0$  とする. また,  $H_u \leq H_p$  とし,  $k+H_u$  から  $k+H_p$  までは入力は変化しないとする.

制御対象のモデルから未来の挙動を予測する. 式 (2.1) より, 予測区間  $H_p$  にわたる予測値は,

$$\begin{cases} \hat{\mathbf{x}}[k+1|k] &= A\mathbf{x}[k] + B\hat{\mathbf{u}}[k|k] \\ &\vdots \\ \hat{\mathbf{x}}[k+H_u|k] &= A^{H_u}\mathbf{x}[k] + A^{H_u-1}B\hat{\mathbf{u}}[k|k] + \cdots + B\hat{\mathbf{u}}[k+H_u-1|k] \\ \hat{\mathbf{x}}[k+H_u+1|k] &= A^{H_u+1}\mathbf{x}[k] + A^{H_u}B\hat{\mathbf{u}}[k|k] + \cdots + [AB+B]\hat{\mathbf{u}}[k+H_u-1|k] \\ &\vdots \\ \hat{\mathbf{x}}[k+H_p|k] &= A^{H_p}\mathbf{x}[k] + A^{H_p-1}B\hat{\mathbf{u}}[k|k] + \cdots + (\sum_{j=0}^{H_p-H_u} A^j B)\hat{\mathbf{u}}[k+H_u-1|k] \end{cases} \quad (2.4)$$

と表わされる．ここで， $\hat{X} = \begin{bmatrix} \hat{\mathbf{x}}[k+1|k] \\ \vdots \\ \hat{\mathbf{x}}[k+H_u|k] \\ \hat{\mathbf{x}}[k+H_u+1|k] \\ \vdots \\ \hat{\mathbf{x}}[k+H_p] \end{bmatrix}$ ， $\hat{U} = \begin{bmatrix} \hat{\mathbf{u}}[k|k] \\ \vdots \\ \hat{\mathbf{u}}[k+H_u-1|k] \end{bmatrix}$  を用い

て式 (2.4) は，

$$\hat{X} = \begin{bmatrix} A \\ \vdots \\ A^{H_u} \\ A^{H_u+1} \\ \vdots \\ A^{H_p} \end{bmatrix} \mathbf{x}[k] + \begin{bmatrix} B & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{H_u-1}B & A^{H_u-2}B & \cdots & B \\ A^{H_u}B & A^{H_u-1}B & \cdots & AB+B \\ \vdots & \vdots & \ddots & \vdots \\ A^{H_p-1}B & A^{H_p-2}B & \cdots & \sum_{j=0}^{H_p-H_u} A^j B \end{bmatrix} \hat{U} \quad (2.5)$$

と表される．これより，式 (2.3) の評価関数は，

$$J[k] = \hat{X}^T \begin{bmatrix} W_y & 0 & \cdots & 0 \\ 0 & W_y & & 0 \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & W_y \end{bmatrix} \hat{X} + \hat{U}^T \begin{bmatrix} W_u & 0 & \cdots & 0 \\ 0 & W_u & & 0 \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & W_u \end{bmatrix} \hat{U} \quad (2.6)$$

と表すことが出来る．式 (2.5) を式 (2.6) に代入すると，

$$\begin{aligned} J[k] &= \hat{X}^T \mathcal{Q} \hat{X} + \hat{U}^T \mathcal{R} \hat{U} \\ &= (\Phi \mathbf{x}[k] + \Theta \hat{U})^T \mathcal{Q} (\Phi \mathbf{x}[k] + \Theta \hat{U}) + \hat{U}^T \mathcal{R} \hat{U} \\ &= \mathbf{x}^T[k] \Phi^T \mathcal{Q} \Phi \mathbf{x}[k] + 2(\mathbf{x}^T[k] \Phi^T \mathcal{Q} \Theta) \hat{U} + \hat{U}^T (\Theta^T \mathcal{Q} \Theta + \mathcal{R}) \hat{U} \end{aligned} \quad (2.7)$$

となる．ここで  $\Phi = \begin{bmatrix} A \\ \vdots \\ A^{H_u} \\ A^{H_u+1} \\ \vdots \\ A^{H_p} \end{bmatrix}$ ， $\Theta = \begin{bmatrix} B & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{H_u-1}B & A^{H_u-2}B & \cdots & B \\ A^{H_u}B & A^{H_u-1}B & \cdots & AB+B \\ \vdots & \vdots & \ddots & \vdots \\ A^{H_p-1}B & A^{H_p-2}B & \cdots & \sum_{j=0}^{H_p-H_u} A^j B \end{bmatrix}$ ，



$$\mathcal{Q} = \begin{bmatrix} W_y & 0 & \cdots & 0 \\ 0 & W_y & & 0 \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & W_y \end{bmatrix}, \mathcal{R} = \begin{bmatrix} W_u & 0 & \cdots & 0 \\ 0 & W_u & & 0 \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & W_u \end{bmatrix} \text{としている.}$$

式 (2.7) は  $\mathcal{H} = \Theta^T \mathcal{Q} \Theta + \mathcal{R}$ ,  $\mathcal{G} = \mathbf{x}^T[k] \Phi^T \mathcal{Q} \Theta$  として,

$$J[\hat{U}] = \frac{1}{2} \hat{U}^T \mathcal{H} \hat{U} + \mathcal{G} \hat{U} (+ \text{定数}) \quad (2.8)$$

の形式をしている.

次に, 状態変数  $\mathbf{x}$  に関する制約条件が以下であるとする.

$$\mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max} \Leftrightarrow \begin{cases} \mathbf{x} \leq \mathbf{x}_{max} \\ -\mathbf{x} \leq -\mathbf{x}_{min} \end{cases} \quad (2.9)$$

これより,

$$\begin{aligned} \hat{X} &= \begin{bmatrix} \hat{\mathbf{x}}[k|k] \\ \vdots \\ \hat{\mathbf{x}}[k+H_p|k] \end{bmatrix} \leq \begin{bmatrix} \mathbf{x}_{max} \\ \vdots \\ \mathbf{x}_{max} \end{bmatrix} = X_{max} \\ -\hat{X} &\leq \begin{bmatrix} -\mathbf{x}_{min} \\ \vdots \\ -\mathbf{x}_{min} \end{bmatrix} = -X_{min} \end{aligned} \quad (2.10)$$

でなければならない. 式 (2.5) から,  $\hat{U}$  に関して,

$$\begin{aligned} \Theta \hat{U} &\leq X_{max} - \Phi \mathbf{x}[k] \\ -\Theta \hat{U} &\leq -X_{min} + \Phi \mathbf{x}[k] \end{aligned} \quad (2.11)$$

と変形できる.

更に, 入力  $\mathbf{u}$  に対する制約条件が,

$$|\mathbf{u}| \leq \mathbf{u}_0 \Leftrightarrow \begin{cases} \mathbf{u} \leq \mathbf{u}_0 \\ -\mathbf{u} \leq \mathbf{u}_0 \end{cases} \quad (2.12)$$

であるとする, と,

$$\begin{aligned} \hat{U} &= \begin{bmatrix} \hat{\mathbf{u}}[k|k] \\ \vdots \\ \hat{\mathbf{u}}[k+H_u-1|k] \end{bmatrix} \leq \begin{bmatrix} \mathbf{u}_0 \\ \vdots \\ \mathbf{u}_0 \end{bmatrix} = U_0 \\ -U &\leq U_0 \end{aligned} \quad (2.13)$$

と表せる．

$$\text{式 (2.11), (2.13) から, } \Gamma = \begin{bmatrix} \Theta \\ -\Theta \\ I \\ -I \end{bmatrix}, w = \begin{bmatrix} \hat{X}_{max} - \Phi \mathbf{x}[k] \\ -\hat{X}_{min} + \Phi \mathbf{x}[k] \\ U_0 \\ U_0 \end{bmatrix} \text{ を用いて,}$$

$$\Gamma \hat{U} \leq w \quad (2.14)$$

となる．

よって，式 (2.8), (2.14) から，モデル予測制御において最適な入力を求める問題は，

$$\min_{\hat{U}} \left[ \frac{1}{2} \hat{U}^T \mathcal{H} \hat{U} + \mathcal{G} \hat{U} \right] \quad \text{subject to} \quad \Gamma \hat{U} \leq w \quad (2.15)$$

という制約つき二次計画問題の形式に帰着できる．式 (2.15) 中の  $\mathcal{H}$  が半正定行列で，かつ制約がない場合にはこの問題は凸最適化問題となり，比較的容易に解くことができる．制約が存在する場合には，ラグランジュの未定乗数法を適用することで制約なし問題に変換して解くことができる．変換された問題が凸最適化問題であれば最適解の存在が保証される．

## 第3章

# 移動ロボットによる協調運搬作業

### 3.1 緒言

#### 3.1.1 背景と目的

近年、複数の移動ロボットをそれぞれ所定の相対位置へ自律走行させるフォーメーション制御が精力的に研究されている [17, 18, 19]. その背景には、制御理論・ロボット開発技術の発展に伴い、人が作業することが困難な状況・場所における作業代替・支援を目的としたロボットが期待されていることが挙げられる. 日本でも阪神大震災以降、人間が直接介入できないような環境下で人命を救助することを目的としたレスキューロボットの研究が行われており、その競技会も開かれている [20]. また、工場や建築現場等の作業環境下においてもこれまでに様々な作業用ロボットが開発されている中で、複数の移動ロボットが自律的に協調することによって指示の簡略化、作業効率の向上、柔軟な作業計画の構成が可能になる. このような観点から、特に、車輪を用いて移動する車輪型移動ロボットを対象とした複数台の走行制御として、ロボットを安定に他車へ追従させるための制御手法が様々な対象分野での応用を目指して盛んに研究されている. このような移動ロボットの制御に対してモデル予測制御を適用した研究として [21] がある. これは各移動ロボットが独立の最適制御問題を分散的に解くことでフォーメーション制御を行うものであるが、移動ロボット間の相対距離の制約は考慮していない. 相対距離の制約を考慮した制御としては [22] がある. しかし、これは衝突を回避することのみを条件としており、移動ロボット間の相対距離が離れすぎることややはり考慮していない. 更に、これらの手法では移動ロボットのダイナミクスを考慮していない.

本章ではより実用的な制御手法を確立するため、これまでに述べた従来の制御手法では

扱われていない厳しい制約条件を要求される移動ロボットの制御問題を取り上げ、その問題に対しモデル予測制御を応用した制御手法を提案する。

具体的には、移動ロボットによる協調制御問題、ここでは 2 台の移動ロボットによる搬送物の運搬作業を取り上げる。このような作業を行うには、後続車は先行車に対してある一定距離の相対距離を保ちつつ、追従走行を行うことになる。通常の追従制御では、後続車が先行車から大きく離れすぎないことを制約とするのみであるが、運搬作業を想定した場合、積載している運搬物の落下や回転による破損などの恐れがあるため、近づきすぎても離れ過ぎてもいけない。従って、一定範囲内の距離差を保つことが必要となり、その分制約として厳しいものになる。このように厳しい制約を伴うため、この制御問題に対して、モデル予測制御を応用することとする。モデル予測制御では、制御対象のモデルを用いて、制御量をあらかじめ定められた目標値へ収束させる。同時に、満たすべき制約がモデル予測制御則の枠組みに沿った形で表現されている必要がある。そこで、制御対象となる移動ロボットはよく用いられるキネマティックモデルよりも詳細なダイナミクスを表現したモデルを用いて表現する。また、車輪型の移動ロボットは非ホロノミック制約という特性を持つため、Astolfi による制御則を元に、協調運搬作業のための追従走行を実現するための目標値生成則を導出する。相対距離制約に関しては、最適化計算時に扱えるロボットの並進速度・回転角速度の制約へと変換する手法を示す。これらのモデル・目標値生成・制約変換により、協調運搬作業問題に対する制御手法を提案する。

### 3.1.2 本章の構成

本章の構成は以下の通りである。2 節では制御対象とする 2 輪の車輪型移動ロボットのモデルと、協調運搬作業問題の定式化について述べる。3 節で協調運搬作業問題に対して提案する制御手法の詳細を示す。制約変換、目標値生成、それらを用いたモデル予測制御の適用について述べる。4 節では数値例により提案手法の有効性を検証する。最後に 5 節で結論を述べる。

## 3.2 移動ロボットによる協調運搬作業問題

### 3.2.1 制御対象

本研究では、図 3.1 のような左右輪が独立に回転する 2 輪の車輪型移動ロボットを制御対象とする。

このようにな車輪によって走行する移動ロボットは構造としては非常に簡潔であるが、

車輪が横滑りしないため、非ホロミック制約と呼ばれる速度制約が存在する [23]．そのためこの点を考慮した制御が必要になる．この問題については次節にて述べる．なお、この制約を考慮した制御入力として、前節で上げた研究と同様に移動ロボットのキネマティックモデルから同様に並進速度  $p_v$  と回転角速度  $p_\omega$  を用いる手法がある [24]．しかし、より実用的な制御のためには移動ロボットのダイナミクスをモデルとして表した動力学モデルを用いて左右輪のモータートルクを制御する必要がある．そのための手法も研究されている [25, 26] がこれらは相対距離制約をしておらず、想定している協調運搬に適用することはできない．

さて、本研究においても、上記の観点から移動ロボットの動力学モデルを用いる．移動ロボットの動特性は以下の式 (3.1)～(3.6) によって表されるものとする [27, 28]．

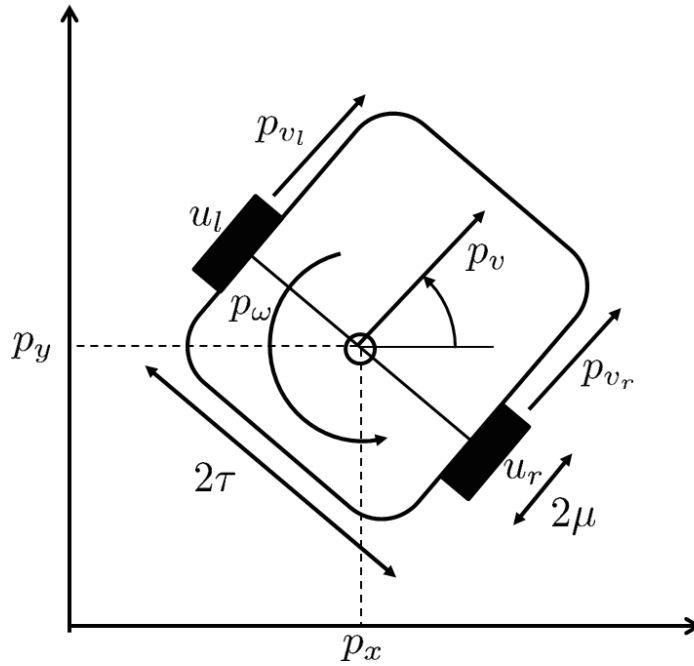


図 3.1 2 輪駆動型移動ロボット

移動ロボットの並進運動，回転運動は以下の式によって表わされる．

$$M\dot{p}_v = D_r + D_l \quad (3.1)$$

$$I_v\dot{p}_\omega = D_r\tau - D_l\tau \quad (3.2)$$

左右輪の回転に関して以下のように表わされる.

$$I_w \ddot{\phi}_r + \sigma \dot{\phi}_r = u_r - \mu D_r \quad (3.3)$$

$$I_w \ddot{\phi}_l + \sigma \dot{\phi}_l = u_l - \mu D_l \quad (3.4)$$

左右輪の速度とロボットの並進速度, 回転角速度の関係は以下のように表わされる.

$$\mu \dot{\phi}_r = p_v + \tau p_\omega \quad (3.5)$$

$$\mu \dot{\phi}_l = p_v - \tau p_\omega \quad (3.6)$$

各パラメータ, 及び変数は表 3.1 のように定義されているものとする. また, ロボットの重心位置は左右輪間の中心であるものとし, その位置をロボットの位置  $(p_x, p_y)$  とする.

表 3.1 移動ロボットのパラメータ

$I_w$	車両慣性モーメント	$M$	重量
$I_v$	重心周りの慣性モーメント	$\tau$	車輪と重心の距離
$\sigma$	粘性摩擦係数	$\mu$	車輪回転半径
$D_r, D_l$	左右輪駆動力	$\phi_r, \phi_l$	左右輪回転角
$p_v$	ロボットの並進速度	$p_\omega$	ロボットの回転角速度

これより移動ロボットの動力学モデルは, 状態変数を  $p_v$  と  $p_\omega$ , 制御入力を左右輪への入力トルク  $u_r, u_l$  とした状態空間モデルとして式 (3.7) のように表せる.

$$\begin{bmatrix} \dot{p}_v \\ \dot{p}_\omega \end{bmatrix} = \begin{bmatrix} a_1 & 0 \\ 0 & a_2 \end{bmatrix} \begin{bmatrix} p_v \\ p_\omega \end{bmatrix} + \begin{bmatrix} b_1 & b_1 \\ b_2 & -b_2 \end{bmatrix} \begin{bmatrix} u_r \\ u_l \end{bmatrix} \quad (3.7)$$

ただし

$$a_1 = \frac{-2\sigma}{M\mu^2 + 2I_w}, \quad a_2 = \frac{-2\sigma\tau^2}{I_v\mu^2 + 2I_w\tau^2}$$

$$b_1 = \frac{r}{M\mu^2 + 2I_w}, \quad b_2 = \frac{\mu\tau}{I_v\mu^2 + 2I_w\tau^2}$$

である. 式 (3.7) を

$$\dot{\mathbf{x}} = A_c \mathbf{x} + B_c \mathbf{u} \quad (3.8)$$

と表すこととする.

ロボットの並進速度  $p_v$ 、回転角速度  $p_\omega$  と、絶対座標系における位置・進行方位  $(p_x, p_y, p_\theta)$  は式 (3.9) のキネマティックモデルによって表される関係になる。

$$\begin{aligned}\dot{p}_x &= p_v \cos p_\theta \\ \dot{p}_y &= p_v \sin p_\theta \\ \dot{p}_\theta &= p_\omega\end{aligned}\tag{3.9}$$

式 (3.7), (3.9) から、入力トルク  $u_r, u_l$  を移動ロボットに入力すると  $p_v, p_\omega$  が変化し、これによって  $p_x, p_y, p_\theta$  を制御することになる。

### 3.2.2 問題設定

図 3.2 は 2 台の移動ロボットによる搬送物の協調運搬作業の想定図である。作業環境は、障害物回避を考慮する必要の無い水平 2 次元平面であるとする。搬送物は 1 台の移動ロボットでは運搬困難な長さ  $L$  の棒状の物体とし、これを Leader, Follower と呼ばれる 2 台の移動ロボットによって運搬する。搬送物の移動経路は Leader が先導し、Follower が追従するものとする。また、移動ロボットは自己の位置・進行方位  $(p_x, p_y, p_\theta)$ 、並進速度  $p_v$ 、回転角速度  $p_\omega$  を取得、送信できるものとする。

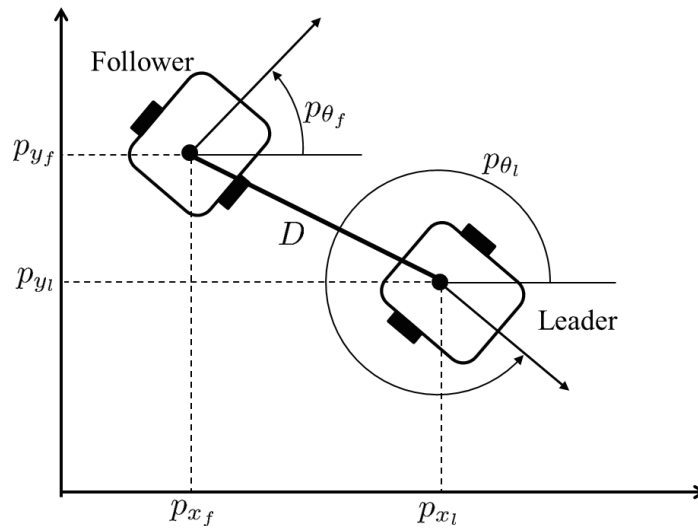


図 3.2 Leader と Follower

本研究における協調運搬作業のための制御の目的は、Leader と Follower 間の相対距離  $D$  に関して、目標距離差  $D_{\text{ref}}$  に収束させ、また、Leader の軌道が変化してもある一定の

範囲

$$D_{\min} \leq D \leq D_{\max} \quad (3.10)$$

に収まりながら Follower が Leader に追従していくように，Follower の入力トルク  $u_{r_f}, u_{l_f}$  を制御することである．

以下に移動ロボットの動作についてまとめる．

**Leader** 任意の軌道を走行する．制御周期  $T_s$  秒ごとに自身の位置・進行方位  $(p_{x_l}, p_{y_l}, p_{\theta_l})$ ，並進速度と回転角速度  $(p_{v_l}, p_{\omega_l})$  を Follower に送信する．

**Follower** 自身の位置・進行方位  $(p_{x_f}, p_{y_f}, p_{\theta_f})$  と  $T_s$  秒毎に送られてくる Leader の情報から，式 (3.10) の制約を保つような制御トルク入力によって走行することで Follower に追従する．

### 3.3 協調運搬作業のための制御手法

この節では，前節で示した協調運搬作業問題に対して，モデル制御を応用した制御手法の詳細について述べる．

#### 3.3.1 目標値の生成

モデル予測制御では，解が存在すれば，与えられた制約を必ず満たしたまま，状態変数を見目標値へ収束させることができる．ここでは，Follower を制御するための目標値の生成について述べる．式 (3.7) で表わされた制御対象となる移動ロボットのモデルで，状態変数としてロボットの並進速度  $p_v$ ，回転角速度  $p_\omega$  をとったため，この 2 つの速度の目標値の導出を行う．制御対象とする車輪型移動ロボットでは，前節で述べたように車輪が横滑りできないため非ホロノミック制約が存在する．このとき連続フィードバック制御ではロボットを見目標座標に制御することが出来ないことが知られている．これに対し，Astolfi は以下のような制御則を提案した [29]．

#### Astolfi の制御則

ロボットの位置・進行方位  $(p_x, p_y, p_\theta)$  を式 (3.13) に従い， $p_\rho, p_\lambda, p_\phi$  によって表わされ



る図 3.3 のような極座標モデルに変換する.

$$\begin{aligned} p_\rho &= \sqrt{p_x^2 + p_y^2} \\ p_\lambda &= -p_\theta + \arctan \frac{-p_y}{-p_x} \\ p_\phi &= \frac{\pi}{2} - p_\theta \end{aligned} \quad (3.11)$$

このとき, 適切な制御パラメータ  $K_\rho$ ,  $K_\lambda$ ,  $K_\phi$  を用いて, 式 (3.9) において,

$$\begin{aligned} p_v &= K_\rho p_\rho \\ p_\omega &= K_\lambda p_\lambda + K_\phi p_\phi \end{aligned} \quad (3.12)$$

とすることで  $p_\rho, p_\lambda, p_\phi$  はそれぞれ 0 に収束し, 移動ロボットは原点  $(p_x, p_y, p_\theta) = (0, 0, \frac{\pi}{2})$  へ収束する.

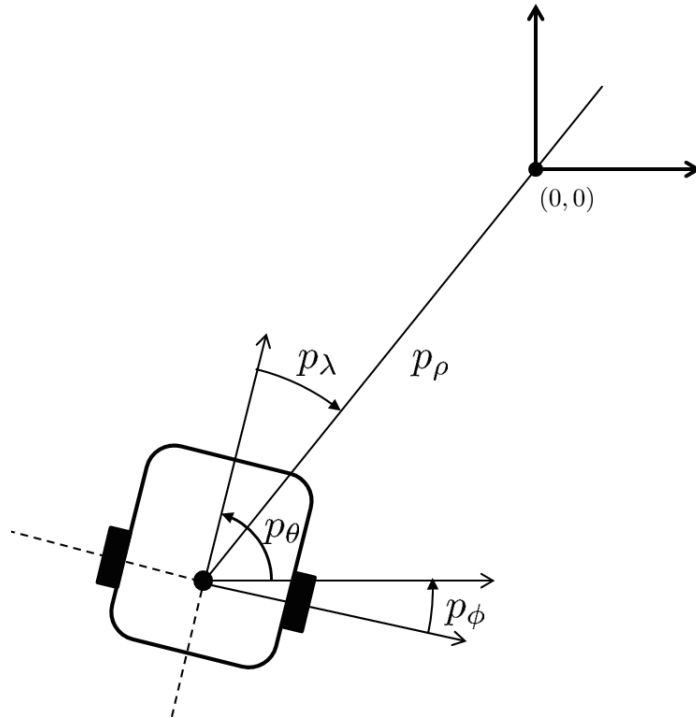


図 3.3 極座標モデル

Astolfi の制御則は, 非ホロノミック制約を持つ移動ロボットが任意の位置・進行方位にあるとき, 絶対座標系の原点に収束させるものである. 従って,  $T_s$  秒ごとに得られる

Leader の位置を原点とした座標変換を行い、その後 Follower に対して Astolfi の制御則式 (3.12) を適用し、得られた  $p_v, p_\omega$  を Follower の並進速度、回転角速度  $p_{v_f}, p_{\omega_f}$  の目標値とすることで Leader に追従すると考えられる。しかし、Follower は Leader から  $D_{\text{ref}}$  だけ距離を保って追従する点と、Leader が並進速度  $p_{v_l}$  で移動している点から、特に速度に関して Astolfi の制御則をそのまま用いることは適当ではない。このため、以下のようにして目標速度を生成する。

### 提案する目標速度生成法

図 3.4 のように 2 台の移動ロボットがあるとき、Leader の位置・進行方位  $(p_{x_l}, p_{y_l}, p_{\theta_l})$  を原点  $(0, 0, \frac{\pi}{2})$  として、Follower の位置・進行方位  $(p_{x_f}, p_{y_f}, p_{\theta_f})$  を変換した値を  $(p'_{x_f}, p'_{y_f}, p'_{\theta_f})$  とする。式 (3.13) に従って変換すると、

$$\begin{aligned} p'_\rho &= \sqrt{(p'_{x_f})^2 + (p'_{y_f})^2} \\ p'_\lambda &= -p'_{\theta_f} + \arctan \frac{-p'_{y_f}}{-p'_{x_f}} \\ p'_\phi &= \frac{\pi}{2} - p'_{\theta_f} \end{aligned} \quad (3.13)$$

が得られる。この様子は図 3.5 のようになる。 $p'_\rho$  の値は、相対距離を表す。よって  $p'_\rho$  を  $D_{\text{ref}}$  に収束させることが目的となる。従って Astolfi の制御則においては  $p_\rho$  が 0 に収束していたように、 $p'_\rho - D_{\text{ref}}$  (図 3.5 の太線部) を 0 を収束させる。また、図 3.5 の原点にある Leader は速度  $p_{v_l}$  で移動していくことから、第 2 項として  $p_{v_l}$  を加えることで Leader の速度を考慮した目標速度  $p_{v_{\text{ref}}}$  を生成するものとする。これより目標速度は式 (3.14) のようになる。

$$p_{v_{\text{ref}}} = K_\rho (p'_\rho - D_{\text{ref}}) + p_{v_l} \quad (3.14)$$

$$p_{\omega_{\text{ref}}} = K_\lambda p'_\lambda + K_\phi p'_\phi \quad (3.15)$$

回転角速度  $p_{\omega_f}$  に関しては、式 (3.15) のように式 (3.12) の形ををそのまま用いて目標値  $p_{\omega_{\text{ref}}}$  を求めるものとする。

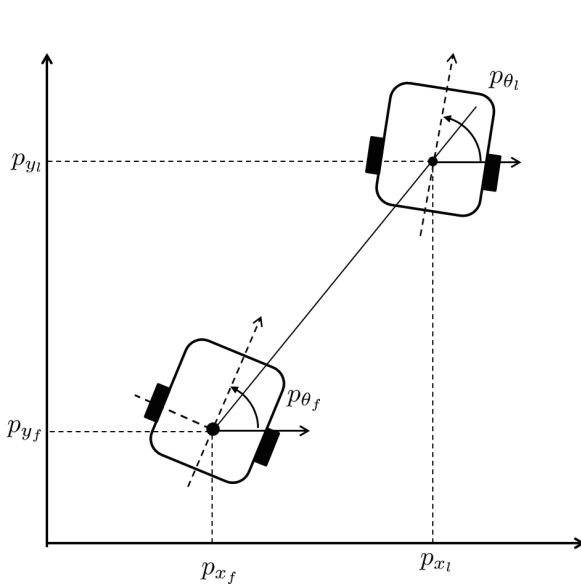


図 3.4 変換前

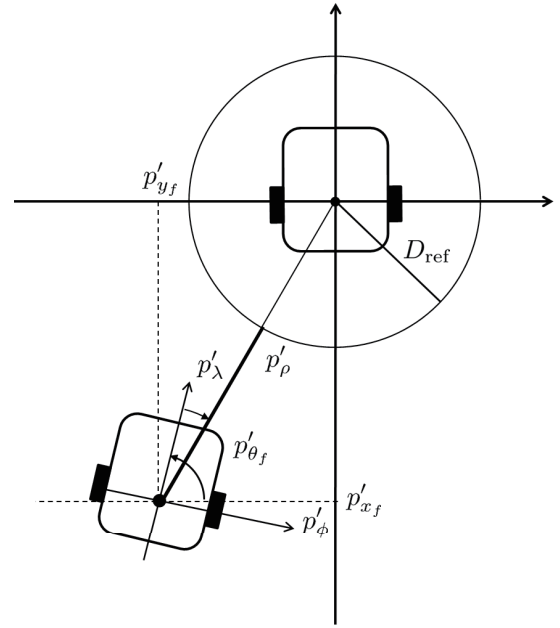


図 3.5 変換後

### 3.3.2 制約の変換

運搬制御の問題設定から，Follower は各時刻ステップにおいて式 (3.10) で示される Leader との相対距離制約を守らなくてはならない．すなわち，現在のステップを  $k$  とすると，Follower は，次ステップ  $k+1$  における Leader の位置・進行方位を予測し， $k+1$  ステップにおいてその予測位置を中心とした半径  $D_{\max}$  の円内，かつ半径  $D_{\min}$  の円外にいないてはならない．この幾何的条件から以下のような Follower の並進速度  $p_{v_f}$ ，回転角速度  $p_{\omega_f}$  に関する制約を導出する．

#### 並進速度制約

図 3.6 のように，現在の Follower の位置  $(p_{x_f}, p_{y_f})$  と，Leader の次ステップにおける予測位置・進行方位  $(\hat{p}_{x_l}, \hat{p}_{y_l}, \hat{p}_{\theta_l})$  を結ぶ線分を考える．この線分の長さは 2 台の相対距離を表す． $(\hat{p}_{x_l}, \hat{p}_{y_l}, \hat{p}_{\theta_l})$  を中心とした半径  $D_{\min}$  の円  $C_1$  と相対距離を示す線分の交点のうち，Follower に近い方を点  $P_1(p_{x_{P1}}, p_{y_{P1}})$  とする．また， $(p_{x_f}, p_{y_f})$  から円  $C_1$  への接線と， $(\hat{p}_{x_l}, \hat{p}_{y_l}, \hat{p}_{\theta_l})$  を中心とした半径  $D_{\max}$  の円  $C_2$  との交点で Follower に近い方を点  $P_2(p_{x_{P2}}, p_{y_{P2}})$  とする．

このとき,  $P_1$  と  $(p_{x_f}, p_{y_f})$  との距離を単位ステップ時間  $T_s$  で割ったものを  $p_{v_f}$  の上限  $p_{v_{\max}}$ ,  $P_2$  との距離を  $T_s$  で割ったものを  $p_{v_f}$  の下限  $p_{v_{\min}}$  とする. これを式 (3.16), (3.17) に示す.

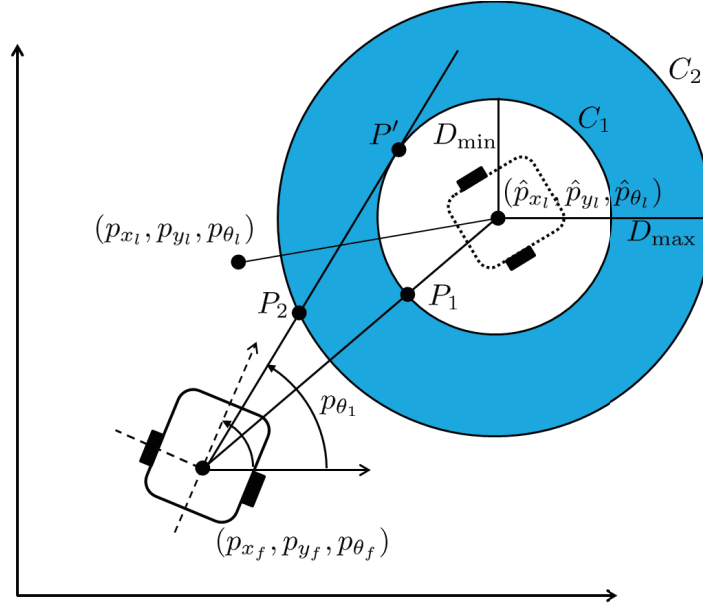


図 3.6 並進速度制約の考え方

$$p_{v_{\max}} = \frac{\sqrt{(p_{x_{P1}} - p_{x_f})^2 + (p_{y_{P1}} - p_{y_f})^2}}{T_s} \quad (3.16)$$

$$p_{v_{\min}} = \frac{\sqrt{(p_{x_{P2}} - p_{x_f})^2 + (p_{y_{P2}} - p_{y_f})^2}}{T_s} \quad (3.17)$$

ここで,  $P_1$  は以下の式を連立することで求まる.

$$\sqrt{(\hat{p}_{x_l} - p_{x_f})^2 + (\hat{p}_{y_l} - p_{y_f})^2} = D \quad (3.18)$$

$$(p_{x_{P1}} - \hat{p}_{x_l})^2 + (p_{y_{P1}} - \hat{p}_{y_l})^2 = D_{\min}^2 \quad (3.19)$$

$$(p_{x_{P1}} - p_{x_f})^2 + (p_{y_{P1}} - p_{y_f})^2 = (D - D_{\min})^2 \quad (3.20)$$

ただし, Leader の予測位置に関して, サンプリング時間を  $T_s$  として,

$$\hat{p}_{x_l} = p_{x_l} + p_{v_l} \cos p_{\theta_l} T_s \quad (3.21)$$

$$\hat{p}_{y_l} = p_{y_l} + p_{v_l} \sin p_{\theta_l} T_s \quad (3.22)$$

とする.

$P_2$  を求めるためにまず接点  $P'(p'_{x_P}, p'_{y_P})$  を求める.  $P'$  は以下の式を連立することによって求まる解のうちの 1 つである.

$$(p'_{x_P} - \hat{p}_{x_l})^2 + (p'_{y_P} - \hat{p}_{y_l})^2 = D_{min}^2 \quad (3.23)$$

$$(p'_{x_P} - \hat{p}_{x_l})(p_{x_f} - \hat{p}_{x_l}) + (p'_{y_P} - \hat{p}_{y_l})(p_{y_f} - \hat{p}_{y_l}) = D_{min}^2 \quad (3.24)$$

ここで,  $\tan p_{\theta_1} = \frac{p'_{y_P} - p_{y_f}}{p'_{x_P} - p_{x_f}}$  となる角度  $p_{\theta_1}$  を用いて,  $P_2$  は以下の式を連立することで求まる.

$$(p_{x_{P_2}} - \hat{p}_{x_l})^2 + (p_{y_{P_2}} - \hat{p}_{y_l})^2 = D_{max}^2 \quad (3.25)$$

$$p_{y_{P_2}} - p_{y_f} = \tan \theta_1 (p_{x_{P_2}} - p_{x_f}) \quad (3.26)$$

### 回転角速度制約

点  $P_2$  と同様にして,  $(p_{x_f}, p_{y_f})$  から円  $C_1$  へのもう一本の接線と円  $C_2$  との交点のうち, Follower に近い方を  $P_3(p_{x_{P_3}}, p_{y_{P_3}})$  とする.

このとき, 図 3.7 に示すように,  $p_{\theta_f}$  から  $P_2, P_3$  を通る接線までのそれぞれのなす角度のうち, 小さい方を進行方位の下限值  $p_{\theta_{min}}$ , 大きい方を上限値  $p_{\theta_{max}}$  とし,  $T_s$  で割ることで回転角速度制約の上限値  $p_{\omega_{max}}$ , 下限値  $p_{\omega_{min}}$  とする. これを式 (3.27), (3.28) に示す.

$$p_{\omega_{max}} = \frac{p_{\theta_{max}}}{T_s} = \frac{\arctan \frac{(p_{y_{P_2}} - p_{y_f})}{(p_{x_{P_2}} - p_{x_f})} - p_{\theta_f}}{T_s} \quad (3.27)$$

$$p_{\omega_{min}} = \frac{p_{\theta_{min}}}{T_s} = \frac{\arctan \frac{(p_{y_{P_3}} - p_{y_f})}{(p_{x_{P_3}} - p_{x_f})} - p_{\theta_f}}{T_s} \quad (3.28)$$

$P_3$  は式 (3.23), (3.24) の解のうち,  $P'$  でない解  $P''$  を用いることで,  $P_2$  と同様に求めることができる.

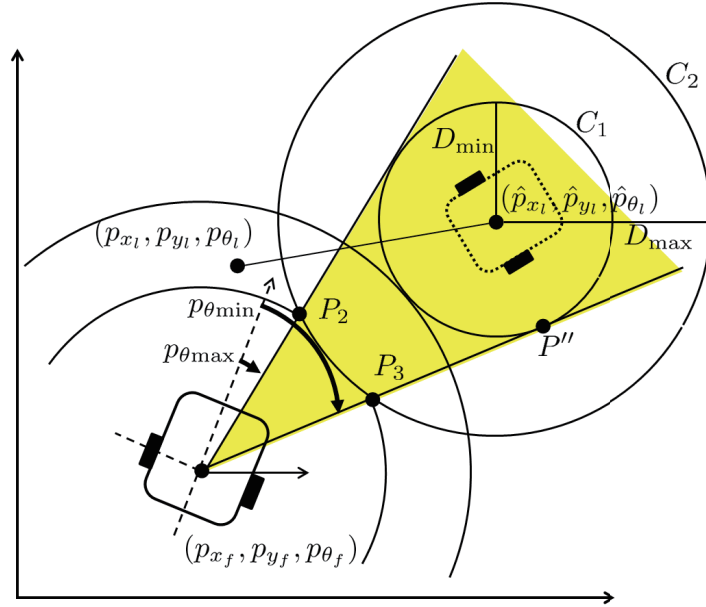


図 3.7 回転角速度制約の考え方

### 3.3.3 制約の幾何的意味

移動ロボットの並進速度と回転角速度に制約を与えることで、 $T_s$  秒後におけるロボットの到達可能領域は制限される。言い換えると、3.2.1 小節、3.2.2 小節にて述べた並進速度と回転角速度の制約変換法は、 $T_s$  秒後の Follower の到達可能領域が、2 台の相対距離に関して  $D_{\min}$  以上  $D_{\max}$  以下になるように制限する制約を求めるものである。

この制約により、Follower の到達可能領域は図 3.8 中に図示されている点  $P_1, P_2, P_3$  からなる緑色の領域内となる。Follower の並進速度と回転角速度の組  $(v_f, \omega_f)$  がどのような値の組となっても、それぞれの制約を満足する限りこの領域内部にしか移動しない。

### 3.3.4 モデル予測制御による制御方式

協調運搬作業のためのモデル予測制御のアルゴリズムは以下ようになる。図 3.9 はこれを図示したものである。

**(step1)** コントローラが、現在の時刻における Leader の位置・進行方位  $(p_{x_l}, p_{y_l}, p_{\theta_l})$ 、並進速度・回転角速度  $(p_{v_l}, p_{\omega_l})$ 、Follower の座標  $(p_{x_f}, p_{y_f}, p_{\theta_f})$  の情報を取得する。

**(step2-1)** 目標値生成則と制約変換則に従って Follower の目標並進速度  $p_{v_f \text{ref}}$ 、目標回

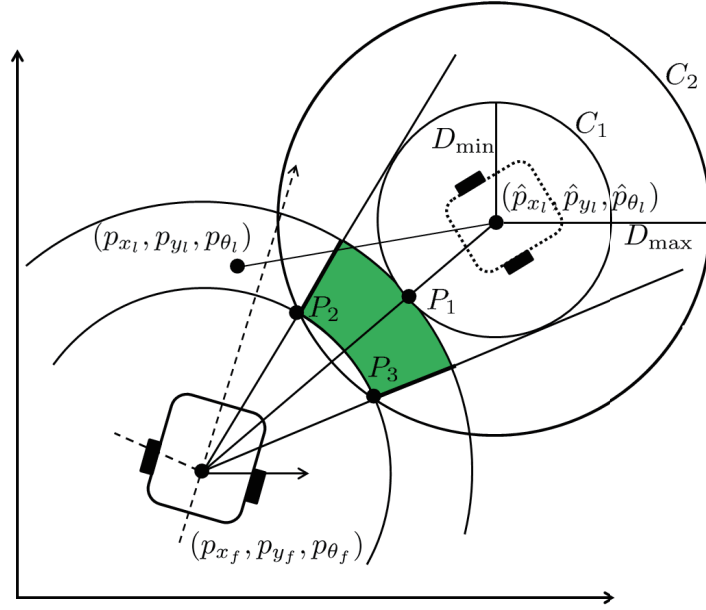


図 3.8 移動ロボットの到達可能領域

- 転角速度  $p_{\omega_{f\text{ref}}}$ , それぞれの制約  $p_{v_{f\text{max}}}$ ,  $p_{v_{f\text{min}}}$ ,  $p_{\omega_{f\text{max}}}$ ,  $p_{\omega_{f\text{min}}}$  を求める.  
**(step2-2)** 式 (3.7) で表されるモデルと (Step2-1) において得られた目標値, 制約から MPC コントローラが Follower への最適な入力トルク  $u_{r_f}$ ,  $u_{l_f}$  を計算する.  
**(step3)** Step 2 における入力トルクによって Follower が走行する. 単位時間  $T_s$  後, 再び Step 1 に戻る.

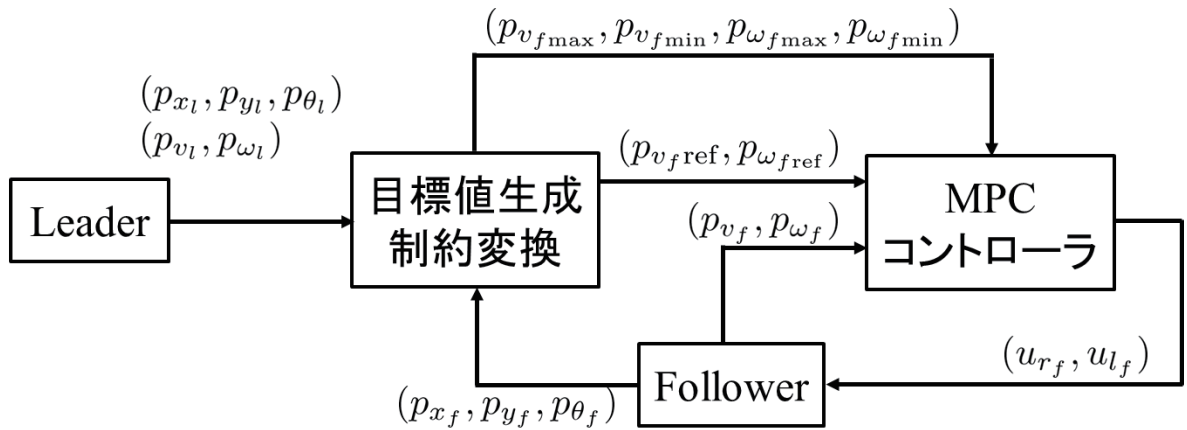


図 3.9 制御の流れ

(Step 2-2) において、時刻ステップ  $k$  のとき MPC コントローラはモデル制御則に基づき、以下の最適化問題を解く。連続時間の式 (3.8) 式を単位ステップ時間  $T_s$  で離散化した式を以下の式 (3.29) で表す。

$$\mathbf{x}[k+1] = A\mathbf{x}[k] + B\mathbf{u}[k] \quad (3.29)$$

ただし、

$$A = e^{A_c T_s}, B = \left( \int_0^{T_s} e^{A\tau} d\tau \right) B_c, \mathbf{x}[k] := \begin{bmatrix} p_{v_f}[k] \\ p_{\omega_f}[k] \end{bmatrix}, \mathbf{u}[k] := \begin{bmatrix} u_{r_f}[k] \\ u_{l_f}[k] \end{bmatrix}$$

である。これより、以下の最適化問題の解により入力トルクが求まる。

$$\min_{\hat{\mathbf{u}}_f} \sum_{i=1}^{Hp} \|\hat{\mathbf{x}}[k+i|k] - \mathbf{x}_{\text{ref}}[k]\|_{W_y}^2 + \sum_{i=0}^{Hu-1} \|\hat{\mathbf{u}}[k+i|k]\|_{W_u}^2 \quad (3.30)$$

subject to

$$\hat{\mathbf{x}}[k+i+1|k] = A\hat{\mathbf{x}}[k+i|k] + B\hat{\mathbf{u}}[k+i|k] \quad (3.31)$$

$$\mathbf{x}_{\min}[k] \leq \hat{\mathbf{x}}[k+i|k] \leq \mathbf{x}_{\max}[k] \quad (3.32)$$

$$i = 0, 1, \dots, Hp$$

ここで、 $\hat{\mathbf{x}}[k+i|k]$  は  $k$  ステップ時における  $k+i$  ステップの予測値を示しており、また、

$$\hat{\mathbf{x}}[k|k] = \mathbf{x}[k], \quad \mathbf{x}_{\text{ref}}[k] = \begin{bmatrix} p_{v_f \text{ref}}[k] \\ p_{\omega_f \text{ref}}[k] \end{bmatrix},$$

$$\mathbf{x}_{\min}[k] = \begin{bmatrix} p_{v_f \min}[k] \\ p_{\omega_f \min}[k] \end{bmatrix}, \mathbf{x}_{\max}[k] = \begin{bmatrix} p_{v_f \max}[k] \\ p_{\omega_f \max}[k] \end{bmatrix}$$

である。

### 3.4 数値例

本節では、前節で示した制御手法の有効性を確認するために計算機上でシミュレーションを行った結果を示す。シミュレーションを行った環境は表 3.2 の通りである。

表 3.1 のパラメータは、 $M = 10$ ,  $\sigma = 0.05$ ,  $\mu = 0.05$ ,  $I_v = 0.5$ ,  $I_w = 5 \times 10^{-3}$ ,  $\tau = 0.1$  とした。また、目標値生成のための制御パラメータの値は試行錯誤的に  $K_\rho = 1.5$ ,  $K_\lambda = 1.5$ ,  $K_\phi = -0.3$  とした。評価関数の重みは  $W_y = \begin{bmatrix} 100 & 0 \\ 0 & 10 \end{bmatrix}$ ,  $W_u = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$  と



表 3.2 シミュレーション環境

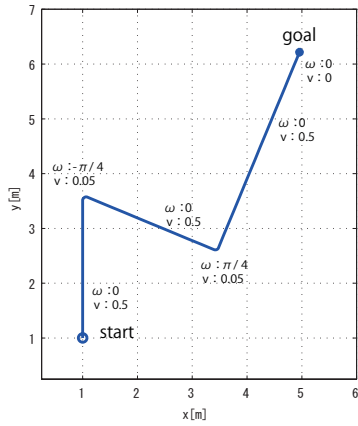
CPU	Intel(R) Core(TM)i7 2600 CPU
Memory	16GB
OS	Windows 7 Professional
software	MATLAB(R2012a)

した．式 (3.7), (3.9) をサンプリング時間  $T_s = 0.5[s]$  で離散化したモデルを用いた提案手法を行う．

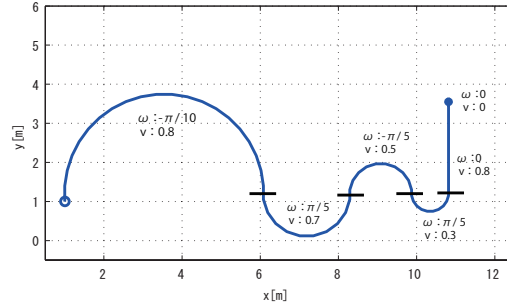
目標距離差  $D_{\text{ref}}$  は  $0.5[\text{m}]$  とし，2 種類の走行経路を用いてシミュレーションを行った結果を以下に示す．

図 3.10(a) は直線をつなぎ合わせた経路で，Leader が走行する時の軌道と並進速度・回転角速度を示している．これを直線走行路とする．同様に図 3.10(b) は円弧と直線をつなぎ合わせた軌道で，Leader が走行する時の経路と速度・角速度を示している．これを円弧走行路とする．初期位置は  $(p_{x_l}, p_{y_l}, p_{\theta_l}) = \left(1, 1, \frac{\pi}{2}\right)$ ， $(p_{x_f}, p_{y_f}, p_{\theta_f}) = \left(1, 0.5, \frac{\pi}{2}\right)$  とした．

次小節より直線走行路でのシミュレーション結果と円弧走行路でのシミュレーション結果をそれぞれ示す．



(a) 直線走行路



(b) 円弧走行路

図 3.10 Leader の走行路

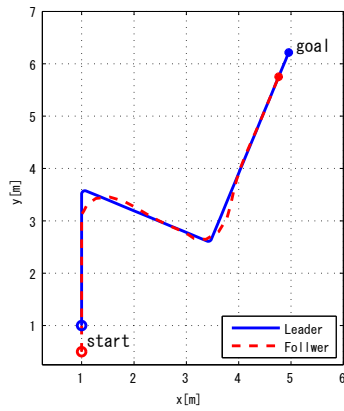
### 3.4.1 直線走行路

図 3.10(a) で示されている Leader の経路に対して、相対距離制約を  $d_{\min} = 0.4[\text{m}]$ ,  $d_{\max} = 0.6[\text{m}]$  とした場合 (相対距離の変動幅  $0.2[\text{m}]$ ) の結果を図 3.11 に示す。

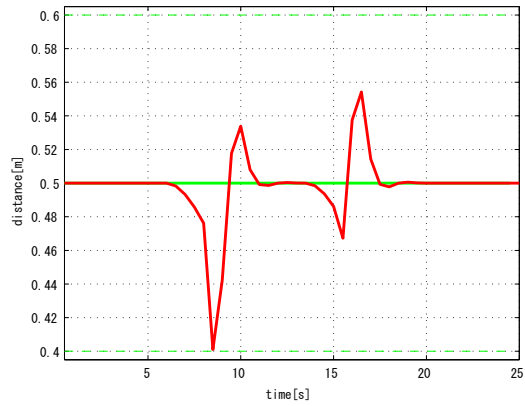
図 5.9(a) は Leader と Follower の走行軌道を  $xy$  平面上に示したもので、これより follower は Leader を追従出来ている様子がわかる。図 5.9(b) はこのときの相対距離の時間変動を示している。この結果において相対距離の最大値は  $0.554[\text{m}]$ , 最小値は  $0.401[\text{m}]$  となったことから相対距離制約を満たしていることがわかる。

ここで、比較として astolfi の制御則による結果を図 3.12 に、偏差系に対する最適レギュレータによる制御による結果を図 3.13 にそれぞれ示す。なお、これらの制御はモデル予測制御の応用である提案手法と異なり、Leader の予測走行情報を利用しない手法である。これらの結果から、距離制約を考慮していない制御手法では適切な走行ができていないことが確認できた。

次に、制約を陽に扱った、すなわち制約の大きさに応じた制御ができていないか検証するために、同じ軌道で  $d_{\min} = 0.45[\text{m}]$ ,  $d_{\max} = 0.55[\text{m}]$  と制約を厳しくした場合 (相対距離の変動幅  $0.1[\text{m}]$ ) の結果を図 3.14 に示す。図 5.9(a) と図 3.14(a) を比べて、Follower の走行軌道に大きな差は見られないが、図 5.9(b) と図 3.14(b) を見ると、相対距離制約の

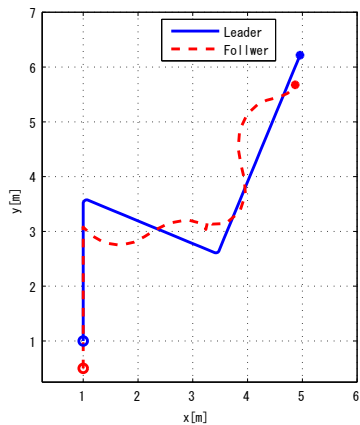


(a) Leader と Follower の軌道

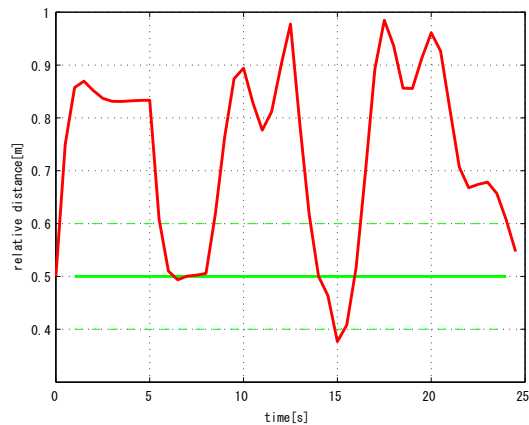


(b) 相対距離の時間変動

図 3.11 直線走行路の結果 (1)



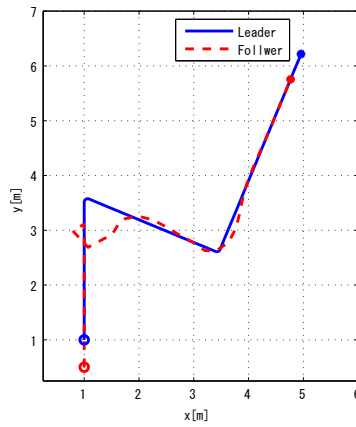
(a) Leader と Follower の軌道



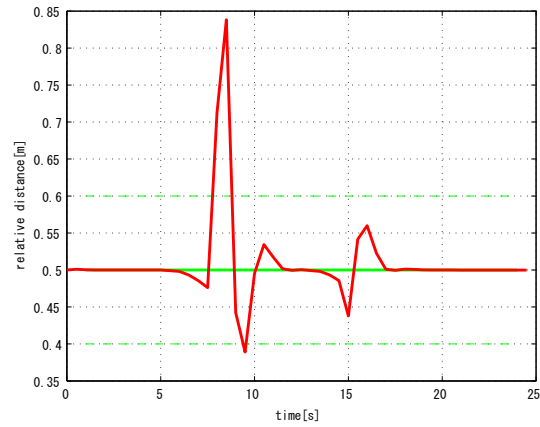
(b) 相対距離の時間変動

図 3.12 astolfi の制御則による直線走行路の結果

影響で変化している様子がわかる．相対距離の最大値は  $0.545[\text{m}]$ ，最小値は  $0.450[\text{m}]$  となったことから，厳しくなった相対距離制約を満たしており，制約生成の効果を示している．

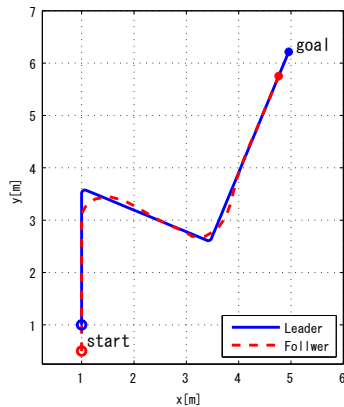


(a) Leader と Follower の軌道

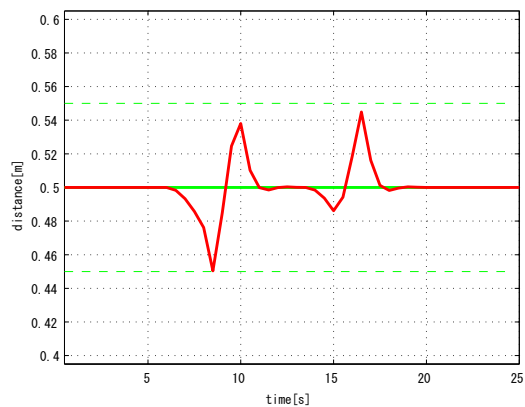


(b) 相対距離の時間変動

図 3.13 最適レギュレータによる直線走行路の結果



(a) Leader と Follower の軌道



(b) 相対距離の時間変動

図 3.14 直線走行路の結果 (2)

### 3.4.2 円弧走行路

図 3.10(b) で示されている Leader の経路に対して，走行軌道と相対距離の変動の結果を図 3.15 に示す．目標距離差は  $0.5[\text{m}]$ ，相対距離制約は  $d_{\min} = 0.4[\text{m}]$ ， $d_{\max} = 0.6[\text{m}]$

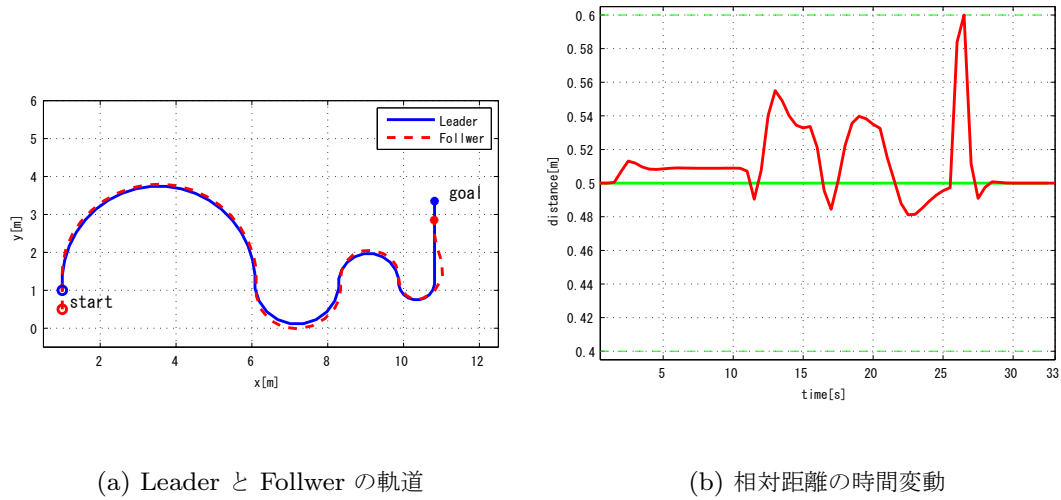


図 3.15 円弧走行路の結果 (1)

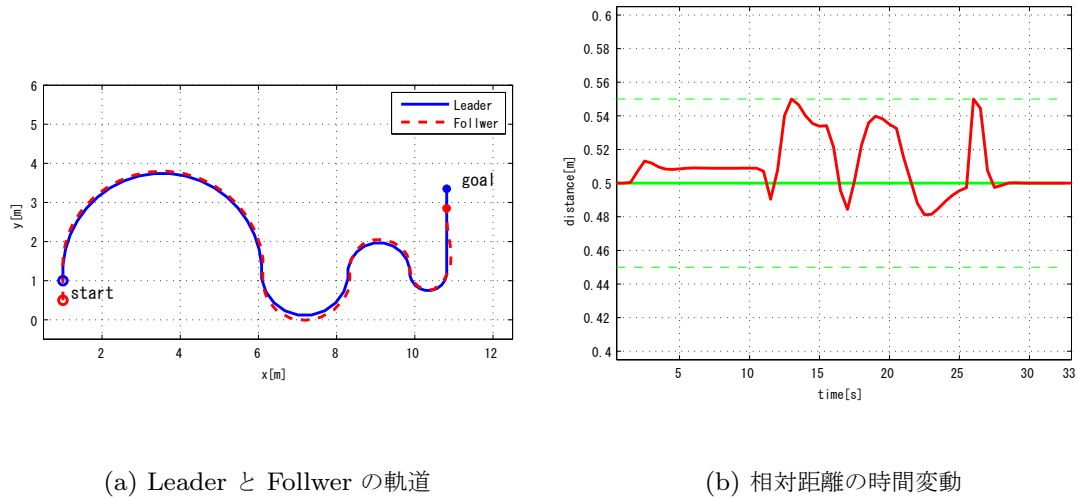
とした．Follower の軌道は図 3.15(a) のようになった．これより直線走行路のときと同様に Leader の軌道を追従できていることがわかる．このときの 2 台の相対距離の時間変動は図 3.15(b) のようになり，最大値  $0.600[\text{m}]$ ，最小値  $0.4812[\text{m}]$  をとより制約を満たしていることがわかる．

直線走行路と同様に，同じ円弧走行路で  $d_{\min} = 0.45[\text{m}]$ ， $d_{\max} = 0.55[\text{m}]$  と制約を厳しくした場合の結果を図 3.16 に示す．このとき，相対距離の最大値  $0.550[\text{m}]$ ，最小値  $0.4812[\text{m}]$  となり，相対距離の最大値が厳しくした制約の範囲内に収まっており，円弧走行路においても制約生成の効果を確認できた．

### 3.4.3 考察

直線走行路，円弧走行路における提案手法による追従走行のシミュレーション結果の総合的考察を述べる．

ここまで示した Leader と Follower 走行軌道と，その際の相対距離の時間変動の結果から，提案手法が，相対距離に関してその変動幅の制約を満たしつつ追従走行し，目標距離差に収束させるという制御の目的を満足していることが示された．並進速度と回転角速度の制約は，相対距離制約の値を用いて導出されているため，相対距離制約が厳しくなれば，それに対応するように並進速度と回転角速度の制約の値も変化する．これにより，同じ軌道を走行していても相対距離の時間変動の様子は変化する．直線走行路での追従に



(a) Leader と Follower の軌道

(b) 相対距離の時間変動

図 3.16 円弧走行路の結果 (2)

において顕著に見られるように、Leader の急激な軌道変化によって Follower との相対距離が大きく変動している。このような状況においては、並進速度と回転角速度の制約によってそれぞれの速度が距離制約を満足する値となるように入力トルクが与えられ、相対距離制約の侵害を防いでいる。

円弧走行路での追従走行でも、直線走行路と同様に、並進速度と回転角速度の制約によって相対距離制約が守られている様子が見られた。一方、円弧走行路においては Leader の走行軌道より Follower の軌道の方がやや大きく旋回していて、初めの 10 秒ほど半円を走行している際に目標とする相対距離差からややずれたまま走行している様子が見られた。これは直線走行時には見られないことから、目標値生成において考慮していなかった Leader の回転角速度が影響しているものと推察される。

### 3.5 結言

本章では、従前の研究で扱われていた制約より厳しい制約が課されるような移動ロボットの協調運搬作業に対し、モデル予測制御の応用による制御手法について述べた。これは、2 台の車輪型移動ロボットを用いて棒状の物体を搬送することを目的として、モデル予測制御を用いて後続車 Follower のトルクを制御する手法である。搬送を目的とすることから、従来の追従制御のように Follower を先行車 Leader に追従させ、その相対距離

を目標値に収束させるだけでなく、搬送物の長さによる相対距離の最小値と最大値の制約を考慮する必要がある。提案手法はこのような相対距離に制約がある場合に、これを幾何的条件から Follower の並進速度、回転角速度に関する制約に変換する。また、移動ロボットのモデルとして動力学モデルを用いて、追従走行での並進速度、回転角速度の目標値生成のために Astolfi の制御則の改良を行った。これにより非ホロノミック制約を持つ Follower が Leader の経路に対して追従走行し、目標距離差に収束させることができた。これらの導出した、制約変換手法、目標値生成手法、動力学モデルを用いてモデル予測制御を応用することで、移動ロボットの協調運搬作業を実現する、モデル予測制御による Follower のトルク制御手法を提案した。

提案手法による直線、円弧軌道での追従走行をシミュレーションした結果、いずれの軌道においても、Follower は相対距離制約を満たしながら、Leader の走行経路を追従した。同じ走行軌道において、相対距離の変動幅を狭めてより厳しい制約とした場合でも、同様に 2 台の相対距離制約は満たされており、この結果から提案手法の有効性を確認した。

## 第4章

# 熱伝導システムの制御系設計モデル 導出

### 4.1 緒言

#### 4.1.1 背景と目的

一般に熱伝導システムに対する温度制御問題を考える際、対象システムのモデリングは制御性能に関わる重要な要素である。たとえば電子写真の定着プロセス [30] において高い品質を得るためには、トナーと媒体を望ましい温度に制御することが必要になる。そこで定着プロセスを熱伝導方程式 [31, 32] として表し、これに有限差分法による離散化 [30] を行うことでコンピュータによるシミュレーション計算が可能となるが、このモデリング方法では多数の状態変数が必要で制御系設計用モデルとしては扱いにくい。そこで、熱伝導システムにおいて熱の伝わる部材が複数ある際に、個々の部材の特性や熱の移動時間を考慮し、状態変数は部材上の適当な点を取ることにすると、モデルサイズを小さくでき、制御系設計には適している。しかし、部材間の熱のやり取りなどによって加わる熱が制御したい箇所まで移動していく時間の長さや、逆にしばらく時間が経過しても残る影響を考慮する必要が生じる。このような動特性はシステム同定によって差分式モデルとしてモデリングすることで対処できるが、この場合、複数のむだ時間 [33, 34] を持つ複雑なモデルとなってしまう。制御対象システムがむだ時間を含むということは、制御入力効果がすぐに出力に現れないということであり、したがって、現時刻で観測できる出力だけでは制御のための十分な情報量にならない恐れがある。また、前述の入力の影響がしばらく残るような場合には、入力と出力の関係を正しく把握することが難しくなる。このよ



うに熱伝導システムの差分式モデルはそのままでは制御系設計モデルとして扱いにくい。

対象システムをモデリングする目的は、高い性能を発揮する自動制御系の構築が主要な目的である [35] が、近年では、モデルベース開発 (MBD)[36] の観点からも対象システムのモデリングは重要なファクターとなっている。MBD は自動車産業を中心に広がりを見せており、その他 OA 機器開発にも適用されている [37]。モデルという共通言語をもつことにより、メカ/エレキ/ソフトといった開発部署間の密接な連携、開発の効率化を図ることが可能となる。また、対象システムをモデルとして表現することにより、実機の試作や実験を行う前にシミュレーションソフトによってその応答を調べられることや、実際の計測が困難な応答の可視化ができるというメリットもある。このことは言い換えるとモデルの品質、すなわち精度良いモデル構築が MBD においても非常に重要であるということを示している。しかし、精密なモデリングを行っても、先の熱伝導方程式の有限差分化によるモデルに見られるように、精度のよい詳細モデルには必要となる情報が必然的に増加してしまうことから現実的な制御系設計には必ずしも有効であるとはいえない。したがって、高性能な制御系設計や、更に制御系設計を含めた開発を進める上では、対象システムを適度に疑似できるレベルでできる限り簡略な設計用モデルが求められる。この点において状態方程式表現は、これまでに多くの制御理論がこれを基に体系化され、更に Matlab/Simulink といったソフトウェア上でツール化されていることから、制御系の構成や対象システムの制御シミュレーションが容易であり、MBD のメリットが生かしやすいといえる。

特に、近年はシステムの構成部材においてその熱容量を低減させ熱しやすくすることで、例えば省エネルギー性を向上させる志向があり、このようなシステムでは緻密な加熱ヒータ制御が必要となり、従来用いられてきた PID 制御では温度制御性能要求を満足できなくなりつつある。これに対し、モデル予測制御の適用が有効であると考えられ、この観点からもシステムをある程度低次の状態方程式として表現することは有益である。

以上のことから、本論文では熱伝導システムにおける複数むだ時間を含んだ差分式モデルから、精度のよい線形状態方程式表現を導出することを目的とする。そのために、差分式モデルのサンプリング周期と導出する状態方程式表現における制御周期との関係に着目し、現在の時刻ステップからむだ時間ステップ経過後の温度応答値までの線形補間を行い、補間した線上の値からつぎの制御周期における値を決定するという処理を行う方法を提案する。この手法により複雑なモデルから簡潔な設計用モデルを得ることができ、対象システムに対して柔軟に各種制御手法の適用が可能となる。

### 4.1.2 本章の構成

本章は、以下 2 節で今回取り扱う問題を定式化し、その詳細について述べる。3 節でその問題に対して提案する制御系設計モデルの導出手法について述べる。4 節では熱伝導システムの数値例に対して提案手法を適用した際の制御シミュレーションを行った結果を示す。最後に 5 節でまとめを述べる。

## 4.2 問題設定

複数部材から構成されるある種の熱伝導システムに対し、差分式モデルを導出すると、(4.1) 式に示す複数の異なるむだ時間を持つ形で表現される。

$$\begin{bmatrix} x_1(l+d_1) \\ x_2(l+d_2) \\ \vdots \\ x_n(l+d_n) \end{bmatrix} = A\mathbf{x}(l) + \sum_{j=j_s}^{j_f} B^{(j)}\mathbf{u}(l-j) \quad (4.1)$$

(4.1) 式について、状態変数は  $\mathbf{x}(l) = (x_1(l), \dots, x_{n_x}(l))^T$  の  $n_x$  個 (本論文においてはベクトル表記は  $\mathbf{x}$  のように太字で表すものとする)、熱源は  $n_u$  個あるものとして入力ベクトル  $\mathbf{u} = (u_1(l), \dots, u_{n_u}(l))^T$  で表す。行列  $A$  は  $n_x$  次正方行列である。サンプリング周期を  $T_l$  とし、周期ごとの時刻を時刻ステップ  $l$  を用いて表す。また、制御周期は長さを  $T_k$  とし、周期ごとの時刻を時刻ステップ  $k$  を用いて表す。これより、現在時刻ステップは  $l$  または  $k$  で表す。つぎに状態変数と入力のむだ時間について、まず  $d_1, d_2, \dots, d_{n_x}$  はそれぞれ各状態変数  $x_1(l), \dots, x_{n_x}(l)$  の時刻ステップ  $l$  によるむだ時間ステップ数を表す。また、右辺第 2 項にあるように入力むだ時間ステップ  $j$  は  $j_s, \dots, j_f$  まで複数あるものとする。行列  $B$  についても入力むだ時間ステップ  $j$  によって異なるものとして、入力  $\mathbf{u}(l-j)$  の係数となる  $B$  行列を  $B^{(j)}$  のように行列の右上に括弧つき添字 ( $j$ ) を付けて表す。これらは  $n_x \times n_u$  行列となる。

このような差分式モデルでは、前節で述べたようにこのままの形では一般的な線形制御理論を適用することができない。この問題を解消するため、(4.1) 式のように制御系設計時に扱いにくい差分式モデルに対して、ある種の変換を施した状態変数  $\xi$  を用いて

$$\xi[k+1] = \tilde{A}\xi[k] + \tilde{B}\mathbf{u}[k] \quad (4.2)$$

という形で表せる一般的な線形の制御系設計モデルを導出する。このことは、導出する設

計モデルは制御系設計用であるためモデリングのサンプル点間隔について式 (4.1) のサンプリング周期  $T_l$  から制御周期  $T_k$  に変更するということである．見やすさのため，(1) 式のような元々のサンプリング周期上にサンプル点をとる式中の状態変数  $\mathbf{x}$  と入力  $\mathbf{u}$  にはその時刻ステップを  $l$ ，入力むだ時間ステップを  $j$  で表して丸括弧 ( ) を用いるのに対し，サンプル点を制御周期とする式中の  $\mathbf{x}, \mathbf{u}$  には時刻ステップを  $k$ ，入力むだ時間ステップを  $i$  で表し角括弧 [ ] を用いることとする．したがって (4.2) 式は角括弧を用いている．

ここで，その制御周期の長さ  $T_k$  は  $T_l$  よりも長く，1 以上の整数  $r$  を用いて

$$r = \frac{T_k}{T_l} \quad (4.3)$$

として表現できるものと仮定する．たとえば OA 機器などの組み込みシステムにおいては，コストや消費電力の点と，熱源の制御だけではなくその他の様々なタスクも処理しなければならない点から，必ずしも十分な演算能力をもたせることができず，實際上，制御周期  $T_k$  を大きく取らざるを得ないことがありうる．このような制約とむだ時間の取り扱いのため，差分式モデルのサンプリング周期に比べて設計モデルの制御周期が大きいと仮定した．(4.3) 式から， $r$  は (4.1) 式の差分式モデルのサンプリング周期と (4.2) 式の線形制御系設計モデルの制御周期との比を表している．すなわち，それぞれの時刻ステップ  $l, k$  について，時刻ステップ  $l$  が  $l+r$  まで達した時に  $k$  はつぎの時刻ステップ  $k+1$  に移るという関係を意味している．

### 4.3 制御系設計モデルの導出手法

この節では，線形な制御系設計モデルの導出手法について述べる．提案する導出手法は以下の 3 つの処理によって構成される．

#### 第 1 処理：入力列の処理

まず始めに，サンプリング周期の時刻ステップ  $l$  とその入力むだ時間ステップ数  $j$  によって表現されている，(4.1) 式の右辺第 2 項に表れる入力列  $\mathbf{u}(l-j_s), \dots, \mathbf{u}(l-j_f)$  について，制御周期の時刻ステップ  $k$  とその入力むだ時間ステップ数  $i_s, \dots, i_f$  によって表記される入力列  $\mathbf{u}[k-i_s], \dots, \mathbf{u}[k-i_f]$  を用いた表現にする処理を行う．

図 4.1 は (4.1) 式の右辺第 2 項  $\sum_{j=j_s}^{j_f} \mathbf{u}(l-j)$  について，ある  $s$  番目の要素  $u_s$  の入力

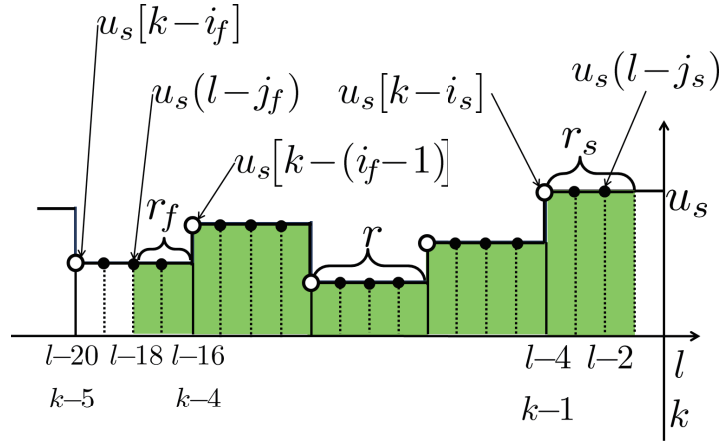


図 4.1 入力和の様子

列の和  $\sum_{j=j_s}^{j_f} u_s(l-j)$ ,  $s = 1, \dots, m$  を模式的に表現した図である．ここでは  $j_s = 2$ ,  $j_f = 18$ , 更に  $r = 4$  とした時の様子を示している．

図 4.1 から，入力列の両端となる  $i_s$ ,  $i_f$  は以下の (4.4), (4.5) 式として表せる．ここで， $\text{ceil}$  関数は与えられた数を正の無限大方向の最も近い整数に丸めるものである．

$$i_s = \text{ceil} \left( \frac{j_s}{r} \right) \quad (4.4)$$

$$i_f = \text{ceil} \left( \frac{j_f}{r} \right) \quad (4.5)$$

これを用いて，4.1 に示したように  $\sum_{j=j_s}^{j_f} u_s(l-j)$  に含まれる  $u_s[k-i_s]$ ,  $u_s[k-i_f]$  の幅をそれぞれ  $r_s$ ,  $r_f$  とすると，それぞれ  $r_s = i_s r - (j_s - 1)$ ,  $r_f = j_f - r(i_f - 1)$  である．

さて，図 4.1 では  $u_s$  のみ模式化した，それぞれの離散時間ステップ  $k$  と  $l$  の関係

から,

$$\begin{aligned}
 \mathbf{u}[k-i_s] &= \underbrace{\mathbf{u}(l-j_s) = \cdots = \mathbf{u}(l-i_s r)}_{r_s \text{ 個}} \\
 \mathbf{u}[k-i] &= \underbrace{\mathbf{u}(l-((i-1)r+1)) = \cdots = \mathbf{u}(l-ir)}_{r \text{ 個}} \\
 &\quad i = i_s + 1, \dots, i_f - 1 \\
 \mathbf{u}[k-i_f] &= \underbrace{\mathbf{u}(l-((i_f-1)r+1)) = \cdots = \mathbf{u}(l-j_f)}_{r_f \text{ 個}}
 \end{aligned}$$

であるので,  $\sum_{j=j_s}^{j_f} B^{(j)} \mathbf{u}(l-j)$  は  $i_s, \dots, i_f$  を用いて, 時刻ステップを  $k$  で考えた以下の (4.6) 式として表現できることがわかる.

$$\begin{aligned}
 \sum_{j=j_s}^{j_f} B^{(j)} \mathbf{u}(l-j) &= \left( B^{(j_s)} + \cdots + B^{(i_s r)} \right) \mathbf{u}[k-i_s] \\
 &\quad + \cdots \\
 &\quad + \left( B^{((i-1)r+1)} + \cdots + B^{(ir)} \right) \mathbf{u}[k-i] \\
 &\quad + \cdots \\
 &\quad + \left( B^{((i_f-1)r+1)} + \cdots + B^{(j_f)} \right) \mathbf{u}[k-i_f] \\
 &= \sum_{i=i_s}^{i_f} B^{[i]} \mathbf{u}[k-i]
 \end{aligned} \tag{4.6}$$

ここで,

$$\begin{aligned}
 B^{[i_s]} &= \sum_{h=0}^{r_s-1} B^{(j_s+h)}, \quad B^{[i_f]} = \sum_{h=0}^{r_f-1} B^{(j_f-h)} \\
 B^{[i]} &= \sum_{h=1}^r B^{((i-1)r+h)} \quad (i = i_s + 1, \dots, i_f - 1)
 \end{aligned}$$

としている. この  $B^{[i]}$ ,  $(i = i_s, \dots, i_f)$  は入力むだ時間ステップ  $i$  を持つ入力  $\mathbf{u}[k-i]$  の係数行列を表すもので,  $\mathbf{u}(l-j)$  の係数行列を  $B^{(j)}$  で表記したように  $B$  行列右上に角括弧つき添字  $[i]$  を付けて表している.

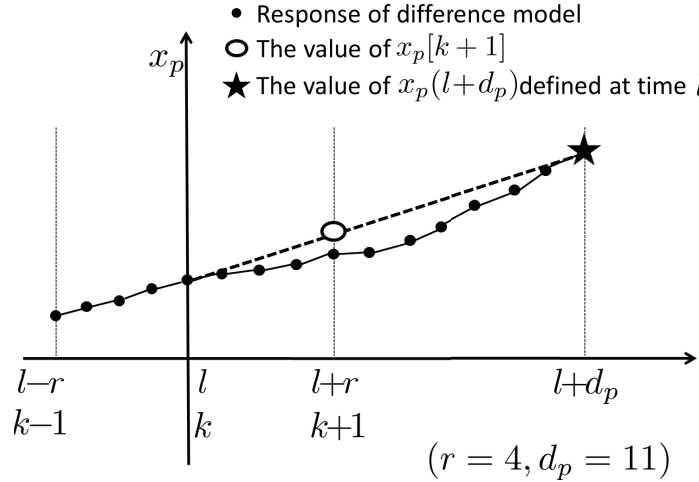


図 4.2 線形補間の様子

## 第 2 処理：状態変数に対する処理

続いて、各状態変数について線形補間により 1 制御周期時間経過後の値を導出する処理を行う。図 4.2 はその様子を模式的に示した図である。

行列  $A$  の  $(p, q)$  要素を  $a_{pq}$ 、行列  $B^{(j)}$  の  $p$  行目の行ベクトルを  $\mathbf{b}_p^{(j)}$  で表す。今、現在時刻ステップでの入力には入力むだ時間が存在していることから、現在時刻ステップにおける状態変数  $\mathbf{x}(l) = \mathbf{x}[k]$  が計測されると、それぞれの状態変数  $x_p$  ( $p = 1, \dots, n_x$ ) のむだ時間経過後の値  $x_p(l + d_p)$  (図 4.2 における星印) について、(4.1) 式を各状態変数ごとに書き下すことで各状態変数と各入力要素についての線形結合和として (4.7) 式にしたがって計算することができる。

$$x_p[l + d_p] = \sum_{q=1}^n a_{pq} x_q(l) + \sum_{j=j_s}^{j_f} \mathbf{b}_p^{(j)} \mathbf{u}(l - j) \quad (4.7)$$

このとき、 $x_p(l) = x_p[k]$  を用いてむだ時間  $d_p$  ステップ経過後の状態変数  $x_p(l + d_p)$  までの間の線形補間を考える。この補間は、4.2 における黒丸から星印までを結んだ太点線として図示できる。制御周期  $T_k$  経過した時の本来の  $x_p$  の値は  $x_p(l + r)$  であるが、ここで、この太点線上における値 (時刻ステップ  $k + 1$  上の中抜き黒丸) を設計モデルの  $x_p[k + 1]$  の値とする。

このことにより、状態変数についても時刻ステップを  $k$  で表した式の導出が可能とな

る．これは，(4.6)，(4.7) 式を用いて

$$\begin{aligned}
 x_p[k+1] &= \frac{r}{d_p} \{x_p(l+d_p) - x_p[k]\} + x_p[k] \\
 &= \left(\frac{d_p-r}{d_p}\right) x_p[k] + \frac{r}{d_p} \sum_{q=1}^n a_{pq} x_q[k] \\
 &\quad + \frac{r}{d_p} \sum_{i=i_s}^{i_f} \mathbf{b}_p^{[i]} \mathbf{u}[k-i]
 \end{aligned} \tag{4.8}$$

として表される．(4.8) 式の処理をすべての状態変数  $x_p$  に対して行い ( $p = 1, \dots, n_x$ )，その結果をまとめると (4.9) 式の状態方程式表現になる．

$$\mathbf{x}[k+1] = \hat{\mathbf{A}}\mathbf{x}[k] + \hat{\mathbf{B}}\hat{\mathbf{u}}[k] \tag{4.9}$$

ここで， $\hat{\mathbf{A}}$  の  $(p, q)$  要素  $\hat{a}_{pq}$  は，

$$\hat{a}_{pq} = \begin{cases} \frac{d_p-r}{d_p} + \frac{r}{d_p} a_{pp}, & q = p \\ \frac{r}{d_p} a_{pq}, & q \neq p \end{cases}$$

となり，

$$\hat{\mathbf{B}} = \begin{bmatrix} \bar{\mathbf{B}}^{[i_s]} & \bar{\mathbf{B}}^{[i_s+1]} & \dots & \bar{\mathbf{B}}^{[i_f]} \end{bmatrix}$$

である． $\bar{\mathbf{B}}^{[i]} (i = i_s, \dots, i_f)$  は，その  $p$  行目の行ベクトル  $\bar{\mathbf{b}}_p^{[i]}$  が  $\bar{\mathbf{b}}_p^{[i]} = \frac{r}{d_p} \mathbf{b}_p^{[i]}$  である行列である．また，

$$\hat{\mathbf{u}}[k] = \begin{bmatrix} \mathbf{u}^T[k-i_s] & \mathbf{u}^T[k-(i_s+1)] & \dots & \mathbf{u}^T[k-i_f] \end{bmatrix}^T \tag{4.10}$$

である．

### 第 3 処理：拡大系による状態方程式表現

(4.9) 式は状態方程式表現にはなっているものの，(4.10) 式から入力ベクトル  $\hat{\mathbf{u}}[k]$  の中身は過去の入力ベクトルを時系列順に並べた形となっている．したがって， $\hat{\mathbf{u}}[k]$  中の  $i$  番目のベクトルを  $\hat{\mathbf{u}}^i[k]$  と書くと，時刻ステップ  $k$  と  $k+1$  の間でたとえば  $\hat{\mathbf{u}}^2[k+1] = \hat{\mathbf{u}}^1[k]$  という関係が生じている．すなわち  $\hat{\mathbf{u}}[k]$  の全要素を自由に決定できるわけではないため，

(4.9) 式のままではまだ一般的な線形制御理論は適用しにくい．そこで最後に，(4.2) 式の形で表現される状態方程式として表すために，過去の入力を状態変数に含めた拡大系を構成する．

拡大系の状態変数  $\xi[k]$  を，以下の (4.11) 式のように定める．

$$\xi[k] = [\mathbf{x}^T[k] \quad \mathbf{u}^T[k-1] \quad \cdots \quad \mathbf{u}^T[k-i_f]]^T \quad (4.11)$$

ここで， $\hat{\mathbf{u}}[k]$  だけでなく  $\mathbf{u}[k-1]$  から  $\mathbf{u}[k-i_f]$  までを状態変数に含めたのは， $\hat{\mathbf{u}}[k]$  中の  $\mathbf{u}[k-i_s]$  について  $i_s \neq 1$  の場合には入力むだ時間が残ってしまい，(4.2) 式の形で現在時刻における入力  $\mathbf{u}[k]$  を状態方程式表現に反映できなくなるためである．

先に例示した様な時刻ステップ間での関係があることを考慮すると (4.9) 式は，

$$\tilde{A} = \begin{bmatrix} \hat{A} & \mathbf{0}_{*1} & \hat{B} \\ \mathbf{0}_{*3} & I_m & \mathbf{0}_{*4} & \cdots & \mathbf{0}_{*4} & \mathbf{0}_{*4} \\ \mathbf{0}_{*3} & \mathbf{0}_{*4} & I_m & \cdots & \mathbf{0}_{*4} & \mathbf{0}_{*4} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_{*3} & \mathbf{0}_{*4} & \mathbf{0}_{*4} & \cdots & I_m & \mathbf{0}_{*4} \end{bmatrix}$$

$$\tilde{B} = [\mathbf{0}_{*3} \quad I_m \quad \mathbf{0}_{*4} \quad \mathbf{0}_{*4} \quad \cdots \quad \mathbf{0}_{*4}]^T$$

という行列  $\tilde{A}$  と  $\tilde{B}$  を用いて (4.2) 式の形で表せる．ここで， $\mathbf{0}_{*1}$ ， $\mathbf{0}_{*2}$ ， $\mathbf{0}_{*3}$ ， $\mathbf{0}_{*4}$  はそれぞれ  $n_x \times n_u(i_s-1)$ ， $n_u \times (n_x + i_f n_u)$ ， $n_u \times n_x$ ， $n_u \times n_u$  の零行列である．

## 4.4 数値例

この節では，熱伝導システム的具体例として文献 [38] に示されているような部材構成によるトナー定着制御を取りあげる．図 4.3 はレーザプリンタにおける温度制御の様子を模式的に示したものであり，左回りに回転する定着ベルトと，右回りに回転する圧力ローラの接触面において印刷が行われる．印刷品質を保つためにこの点の定着ベルト温度が目標温度になるようにヒータを制御することを目的とする．状態変数として定着ベルトと圧力ローラ上のいくつかの点の温度と，印刷機内温度，環境温度 (外気温)，制御入力を熱源 (ヒータ) の熱入力 (Duty) とした．これに対し，前節で示した導出手法により得られた制御系設計モデルによる制御の数値シミュレーションを行った結果を 2 例示す．



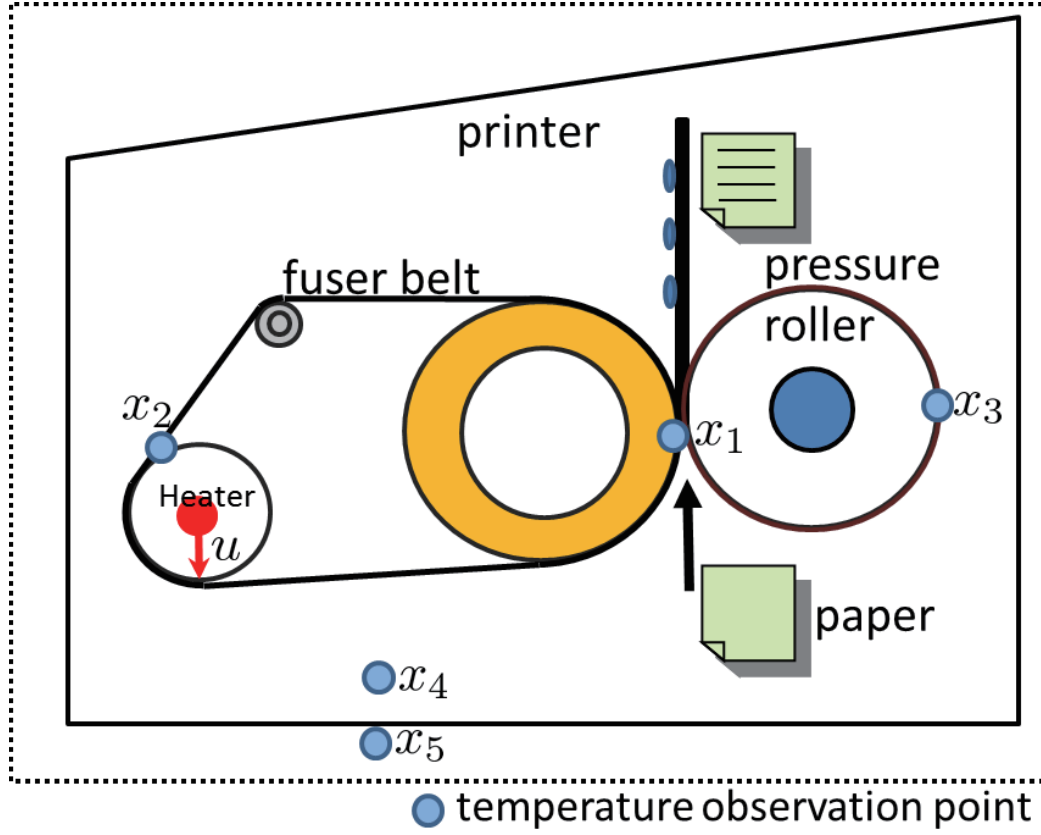


図 4.3 プリンタの模式図

#### 4.4.1 単入力システム (数値例 1)

熱伝導システムの差分式モデルとして、(4.12) 式が得られたものとする。数値例 1 では、各状態変数について、 $x_1$ ,  $x_2$ ,  $x_3$  をそれぞれ計測可能な各部材温度とし、 $x_4$  を機内温度、 $x_5$  としてシステム近傍の外気温をとる。 $x_1$  を目標温度に保つことが制御の目的となる。この様子は図 4.3 のようになり、ヒータは 1 つとする、すなわち、このシステムは単入力システムである。また、 $T_l = 0.1[\text{s}]$  とする。

$$\begin{bmatrix} x_1(l+11) \\ x_2(l+17) \\ x_3(l+8) \\ x_4(l+19) \\ x_5(l+10) \end{bmatrix} = A\mathbf{x}(l) + \sum_{j=2}^{30} \mathbf{b}^{(j)} u(l-j) \quad (4.12)$$

ただし,

$$A = \begin{bmatrix} 0 & 0.995 & 0 & 0.005 & 0 \\ 0.853 & 0 & 0.139 & 0.008 & 0 \\ 0.031 & 0 & 0.960 & 0.009 & 0 \\ 0 & 0.017 & 0 & 0.960 & 0.023 \\ 0 & 0 & 0 & 2.5 \times 10^{-6} & 1 - 2.5 \times 10^{-6} \end{bmatrix}$$

であり  $\mathbf{b}^{(j)}, j = 2, \dots, 30$  については, それぞれの第 1 要素  $b_1^{(j)}$  を

$$\begin{aligned} & \begin{bmatrix} b_1^{(2)} & b_1^{(3)} & \dots & b_1^{(30)} \end{bmatrix} \\ &= \begin{bmatrix} 55 & 60 & 65 & 71 & 77 & 82 & 88 & 93 & 98 \\ 103 & 107 & 110 & 112 & 114 & 114 & 114 & 112 \\ 110 & 107 & 103 & 98 & 93 & 88 & 82 & 77 \\ 71 & 65 & 60 & 55 \end{bmatrix} \times 10^{-4} \end{aligned}$$

とし, それ以外の要素  $b_n^{(j)}, n = 2, \dots, 5$  はすべて 0 とした. たとえば,

$$\mathbf{b}^{(j_s)} = \mathbf{b}^{(2)} = \begin{bmatrix} 55 \times 10^{-4} & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

である.

この差分式モデル (4.12) 式に対して, 制御周期の大きさを  $T_k = 0.4[\text{s}]$ , すなわち  $r = 4$  として制御系設計モデルの導出手法を適用する.

#### [第 1 処理]

ヒータ入力  $u$  に関して, (4.4), (4.5) 式から,

$$i_s = \text{ceil} \left( \frac{2}{4} \right) = 1 \quad (4.13)$$

$$i_f = \text{ceil} \left( \frac{30}{4} \right) = 8 \quad (4.14)$$

となる. これより, 設計モデルの状態変数の数は  $5 + 8 = 13$  となる.

更に, (4.6) 式の  $B^{[i]}$  について計算すると (この例では  $u$  がスカラーなので  $\mathbf{b}^{[i]}$ ), それ

それ以下の形になる.

$$\begin{aligned}
 & \begin{bmatrix} \mathbf{b}^{[1]} & \dots & \mathbf{b}^{[8]} \end{bmatrix} \\
 &= \begin{bmatrix} 180 & 318 & 401 & 450 & 443 & 382 & 295 & 115 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times 10^{-4}
 \end{aligned} \tag{4.15}$$

### [第 2 処理]

$x_1$  に対して (4.8) 式を計算すると, (4.15) 式を用いて以下の (4.16) 式となる.

$$\begin{aligned}
 x_1[k+1] &= \frac{4}{11} \{x_1(l+11) - x_1[k]\} + x_1[k] \\
 &= \frac{7}{11} + \frac{4}{11}a_{12} + \frac{4}{11}a_{14} + \frac{4}{11} \sum_{i=1}^8 \mathbf{b}_p^{[i]} u[k-i]
 \end{aligned} \tag{4.16}$$

$x_2, \dots, x_5$  についても同様に計算することで, 以下の (4.17) 式が得られる.

$$\mathbf{x}[k+1] = \hat{A}\mathbf{x}[k] + \hat{B}\hat{\mathbf{u}}[k] \tag{4.17}$$

ここで,

$$\begin{aligned}
 \hat{A} &= \begin{bmatrix} \frac{7}{11} & \frac{4}{11}a_{12} & 0 & \frac{4}{11}a_{14} & 0 \\ \frac{4}{17}a_{21} & \frac{13}{17} & \frac{4}{17}a_{23} & \frac{4}{17}a_{24} & 0 \\ \frac{4}{8}a_{31} & 0 & \frac{4}{8} + \frac{4}{8}a_{33} & \frac{4}{8}a_{34} & 0 \\ 0 & \frac{4}{19}a_{41} & 0 & \frac{15}{19} + \frac{4}{19}a_{44} & \frac{4}{19}a_{45} \\ 0 & 0 & 0 & \frac{4}{10}a_{54} & \frac{6}{10} + \frac{4}{10}a_{55} \end{bmatrix} \\
 \hat{B} &= \begin{bmatrix} \bar{\mathbf{b}}^{[1]} & \bar{\mathbf{b}}^{[2]} & \dots & \bar{\mathbf{b}}^{[8]} \end{bmatrix} \\
 \hat{\mathbf{u}}[k] &= [u[k-1] \quad \dots \quad u[k-8]]^T
 \end{aligned}$$

である. ただし,  $\bar{\mathbf{b}}^{[i]}$  は, その第 1 要素について  $\bar{\mathbf{b}}_1^{[i]} = \frac{4}{11}\mathbf{b}_1^{[i]}$ , その他の要素はすべて 0 であるベクトルである.

## [第 3 処理]

(4.17) 式から，入力むだ時間を排した状態方程式表現にするための拡大系は以下の (4.18) 式となる．

$$\boldsymbol{\xi}[k+1] = \tilde{A}\boldsymbol{\xi}[k] + \tilde{\mathbf{b}}u[k] \quad (4.18)$$

ただし，

$$\boldsymbol{\xi}[k] = \begin{bmatrix} \mathbf{x}[k] & u[k-1] & u[k-2] & u[k-3] & \cdots & u[k-8] \end{bmatrix}^T$$

$$\tilde{A} = \begin{pmatrix} \hat{A} & \bar{\mathbf{b}}^{[1]} & \bar{\mathbf{b}}^{[2]} & \cdots & \bar{\mathbf{b}}^{[7]} & \bar{\mathbf{b}}^{[8]} \\ \mathbf{0} & 0 & 0 & \cdots & 0 & 0 \\ \mathbf{0} & 1 & 0 & \cdots & 0 & 0 \\ \mathbf{0} & 0 & 1 & \cdots & 0 & 0 \\ & & & \ddots & & \\ \mathbf{0} & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

$$\tilde{\mathbf{b}} = (\mathbf{0} \quad 1 \quad 0 \quad 0 \quad \cdots \quad 0)^T$$

である．ここで， $\mathbf{0}$  は 5 次の零行ベクトルである．

もとの差分式モデル (4.12) と，提案手法による導出モデル (4.18) に対して，同じ入力を加えた際の各部材温度の応答を比較したグラフを図 4.4 に示す．初期温度はすべて 15[°C] とした．実線が導出モデルの温度応答，点線が差分式モデルにおける温度応答である．なお， $x_5$  の温度応答はほとんど変化しないため割愛した．ここで与えている入力は，出力目標値を 180[°C] とし，(4.12) 式において，入力を  $u[k]$ ，出力温度を  $x_1[k]$  とし構成した速度型 PID 制御によって得られたものを用いている．その入力応答も図 4.4 中に示している．これより，27[s] あたりまで Duty100[%] を維持し，その後徐々に低下していく様子が見られるが，むだ時間の影響によって  $x_1$  の応答が 27[s] 以降も上昇していき目標温度に対してオーバーシュートを生じている様子がわかる．図 4.4 の結果から，全体的にモデルの違いによる応答差はほとんど見られないが，特に温度上昇やオーバーシュートを生じている箇所など各状態変数の温度応答に若干の差が見られる．

つぎに，図 4.4 で示した速度型 PID 制御と，導出モデル (4.18) をモデル予測制御 (MPC) の予測モデルとして適用した際の，それぞれの  $x_1$  の温度応答，入力 Duty 応答を比較した結果を図 4.5 に示す．モデル予測制御のパラメータは予測区間を 10 ステップ，制御区間を 2 ステップとした．どちらのシミュレーションもシステムモデルとして差分式モデル (4.12) を用いた．今回も制御目的は  $x_1$  の温度を目標温度 180[°C] (図 4.5 上の細点

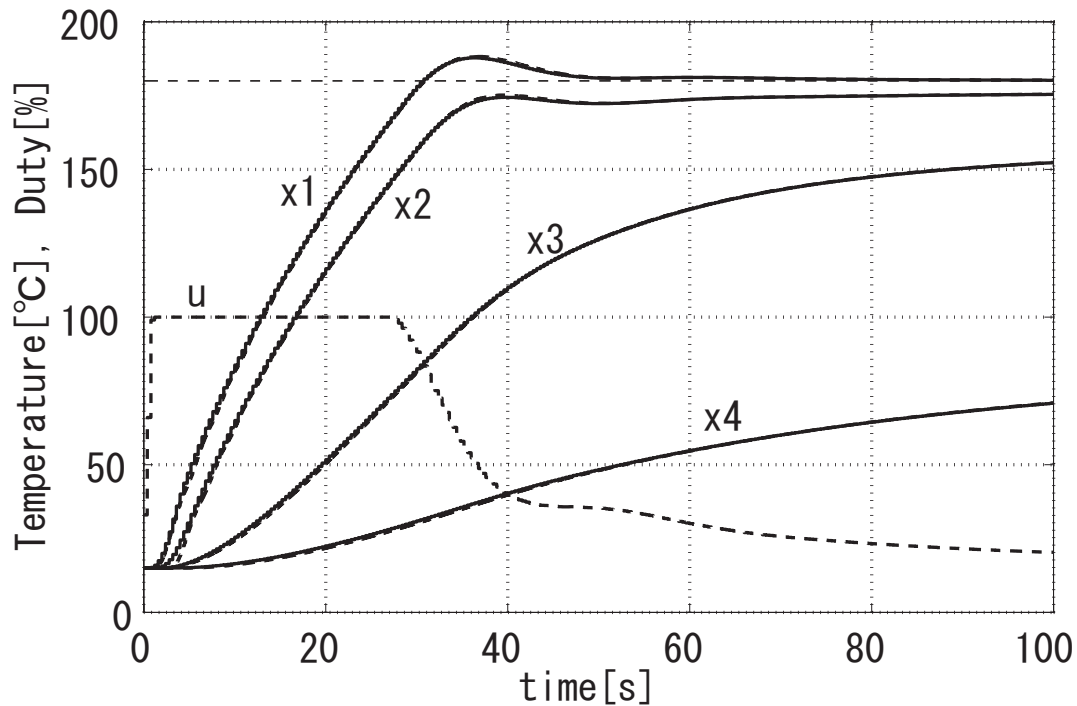


図 4.4 温度および入力の応答比較 (差分式モデルと制御系設計モデル)

線) へ制御するものとしている。この結果より、モデル予測制御による温度応答のほうがわずかに立ち上がりがよく、また目標温度への収束も早い様子がわかる。

このことは、差分式モデルそのままでは PID 制御のような制御則しか適用できなかったものが、提案する導出法による設計モデルによってモデル予測制御が容易に適用できるようになったことで制御性能を向上させることができたものであり、提案手法の有用性を示しているといえる。

#### 4.4.2 多入力システム (数値例 2)

次に、入力熱源が複数ある熱伝導システムを対象とする。この場合多入力系となり PID 制御系設計が行いにくい。このような対象に対して提案手法を適用し、温度制約を与えたモデル予測制御を適用した結果を示す。

制御対象の差分式モデルについて、以下の (4.19) 式に示す。例 1 と同様に、状態変数として  $x_1$  から  $x_7$  までそれぞれ各部材上のいくつかの箇所の温度をとり、 $x_8$  は外気温と

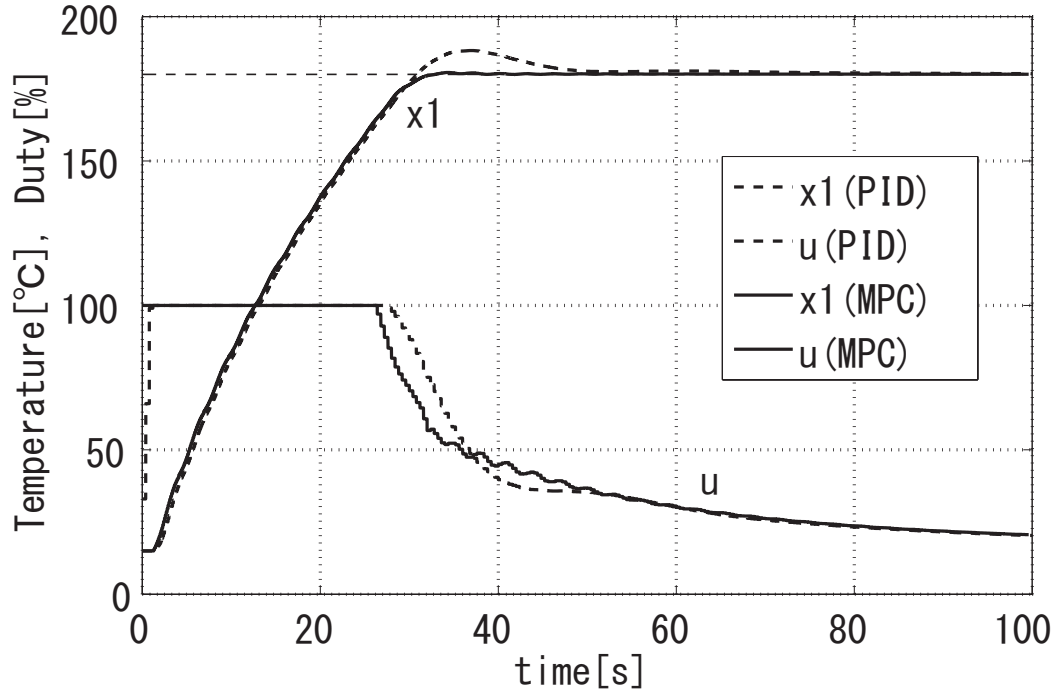


図 4.5 温度および入力の応答比較 (PID 制御とモデル予測制御)

する．入力は熱源の数を 3 とし ( $\mathbf{u} = [u_1 \ u_2 \ u_3]^T$ ), それぞれの熱入力 (Duty) をとるものとする．

$$\begin{bmatrix} x_1(l+7) \\ x_2(l+15) \\ x_3(l+12) \\ x_4(l+9) \\ x_5(l+14) \\ x_6(l+20) \\ x_7(l+18) \\ x_8(l+10) \end{bmatrix} = A\mathbf{x}(l) + \sum_{j=5}^{23} B^{(j)}\mathbf{u}(l-j) \quad (4.19)$$

ただし,

$$A = \begin{bmatrix} 0 & 0.875 & 0 & 0 & 0 & 0 & 0.125 & 0 \\ 0.589 & 0 & 0.190 & 0 & 0.099 & 0 & 0.122 & 0 \\ 0.083 & 0 & 0.868 & 0 & 0 & 0 & 0.049 & 0 \\ 0 & 0.386 & 0.301 & 0.189 & 0 & 0 & 0.124 & 0 \\ 0 & 0 & 0 & 0.083 & 0.820 & 0 & 0.097 & 0 \\ 0 & 0 & 0.013 & 0 & 0.071 & 0.692 & 0.187 & 0.037 \\ 0 & 0.013 & 0 & 0 & 0 & 0.010 & 0.970 & 0.007 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \times 10^{-6} & 1 - 1 \times 10^{-6} \end{bmatrix}$$

であり,  $B^{(j)}$  については, 1,4 行目を非零, それ以外を零とし, 非零要素の値はむだ時間ステップ  $j$  によって変わるものとした. たとえば,  $B^{(5)}$ ,  $B^{(6)}$  は以下のように表せる.

$$B^{(5)} = \begin{bmatrix} 0.0160 & 0.0144 & 0.0062 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.0139 & 0.0126 & 0.0057 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B^{(6)} = \begin{bmatrix} 0.0171 & 0.0154 & 0.0067 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.0149 & 0.0134 & 0.0061 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

この差分式モデル (4.19) に対して, 第 3 処理までを行い線形モデルを導出した. 今回は  $x_3$  を目標温度  $180[^\circ\text{C}]$  に制御するものとし, 例 1 と同様に導出モデルによるモデル予測制御を用いたシミュレーションを行った. このときの  $x_1$ ,  $x_3$ ,  $x_4$  と  $u$  の各応答を図 4.6 に示す. この結果から,  $x_3$  が  $180[^\circ\text{C}]$  に制御出来ていることがわかる. しかし,  $x_1$  と  $x_4$  が  $270[^\circ\text{C}]$  を超える高温になっていたため, 今度は部材許容温度  $260[^\circ\text{C}]$  以下という温度制約を課したモデル予測制御を行った. このときの応答を示した図 4.7 の結果から, 温度制約のない図 4.6 に比べて制約を考慮した制御を行えている様子がわかる.  $x_1$  の最高温度は  $260.68[^\circ\text{C}]$  となり, モデル誤差に起因して温度制約を満たさない箇所があった. しかし, 温度制約を考慮した制御が行えるので, たとえば安全性の面などを考慮し許容温度よりマージンを取るなど制約条件を見直すことで回避できる.

## 4.5 結言

本章では異なるむだ時間が複数存在する複雑な差分式モデルで表される熱伝導システムを対象として, むだ時間ステップ分経過後までの温度応答変化に線形補間を適用することによる線形の制御系設計モデルの導出法について述べた. 更に, この導出法による設計モ

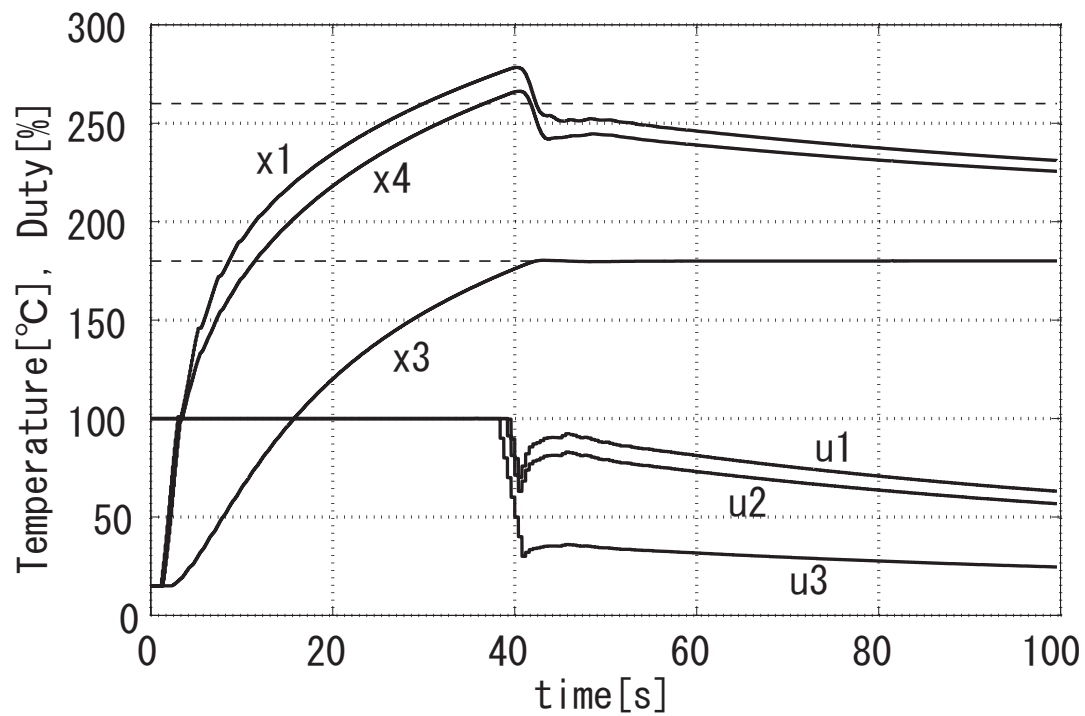


図 4.6 温度および入力の変答比較 (モデル予測制御：制約なし)

デルを用いることで、もとの差分式モデルでは適用できなかったモデル予測制御のような高度で実用的な制御手法を適用することが可能となり、数値例ではその制御性能が差分式モデルでも適用できる PID 制御のそれを上回ったことや、温度制約を考慮した多入力系の制御が容易に行えたことから導出法の有用性を示すことができた。



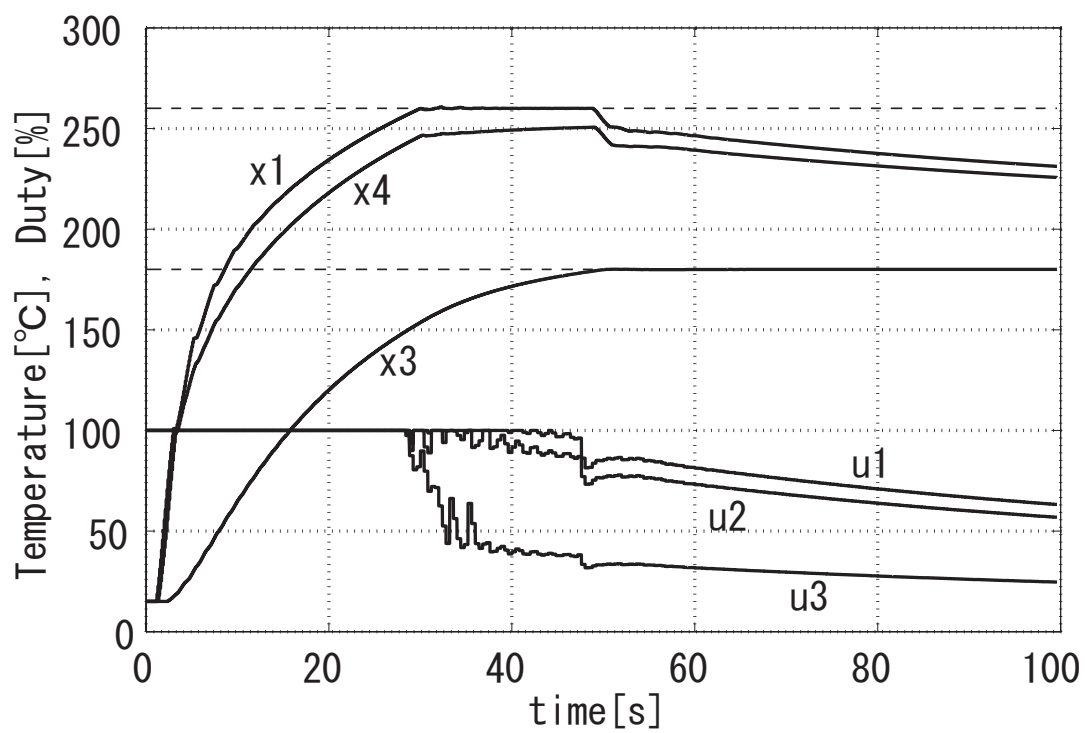


図 4.7 温度および入力の応答比較 (モデル予測制御：制約あり)

## 第5章

# モデル予測型PID制御

### 5.1 緒言

まず，PID 制御が広く用いられている熱伝導システムを改めて取り上げる．

例えばフィルム包装機やプラスチック成形機など，製品加工システムには熱伝導システムが多い．このような熱伝導システムでは，システム外部からの被加熱部材が投入されてシステムの温度動特性が変化しても，特定の部材・部位の温度を高精度に制御することが求められる．この際に，与える熱量に過不足が生じて目標温度へうまく制御できないと製品品質が劣化してしまう．特に，過加熱はエネルギーの無駄であるばかりか，構成部材の限度をこえる過昇温を招く恐れがあり，システムの安全性や耐久性に悪影響を及ぼす恐れがある．したがって，高品質化・省エネルギー化や更なる安全性の向上のためには，最高温度制約を考慮しながら適切な熱量を与えるようヒータを高精度に温度制御することが求められる．

従来より，このヒータ制御には主として PID 制御が用いられている．PID 制御は信頼性が高く，かつ構造が簡単であるため，古くから産業界で幅広く用いられてきた．PID 制御器の性能は3つの PID ゲインの値のチューニングにより大きく左右されるため，そのチューニング方法については古くからさまざまな手法が提案されてきた [39] しかし，熱伝導システムに対しては，主として以下の2つの理由により PID ゲインのチューニングは容易ではない．

#### 1. 使用環境による伝熱特性の変化

例えば，熱伝導システムを冬の朝に始動させる場合には低温の状態から目標温度まで加熱する．一方，一度加熱して一時停止後に再度始動する場合，蓄熱された状態

から目標温度まで加熱することになる．このように部材温度が使用環境によって異なると伝熱特性が変わるため，前者と後者を共に精度よく温度制御するための固定 PID ゲインの設計は困難である．

## 2. 部材の低熱容量化

最近の熱伝導システムでは，構成部材を低温から目標温度まで加熱する際の所要時間を短縮するために，低熱容量の部材を用いることが多い．この場合，熱しやすく冷めやすいため，高精度な温度制御が難しい．

これらのことから，近年の高性能な熱伝導システムの実現に対しては，従来の固定ゲインによる PID 制御器では対応が難しくなっている．また，従来の PID 制御器では過剰に温度が上昇することを防止するための温度制約を陽に考慮した設計を行うこともできない．よって，これらに対する解決策が求められている．

そこで，制約を陽に考慮でき，オンライン (制御周期ごと) で最適な入力を計算するというモデル予測制御 (MPC) の特長が，上に述べた制御課題に有効であることが期待される．しかし，現在稼働しているシステムでは PID 制御器が広く使われており，これに対して構造の異なる MPC を新たに導入するためには，手法だけでなく制御器の入れ替えも含めた各種条件下での実用性の検証が必要となり，多大なコストや労力がかかる．それゆえ，従来用いられている PID 制御器を即座に MPC による制御器に切り替えることは容易ではない．

そこで，本論文では MPC のアルゴリズムに基づいた PID ゲイン設計を行うモデル予測型 PID (MP-PID) 制御手法を提案する．なお，モデル予測型 PID 制御としては，これまでも例えば，一般化モデル予測制御則 (Generalized Predictive Control: GPC) に基づく固定 PID ゲインを導出するもの [5, 40]，制御周期ごとのオンラインチューニングを行うもの [41, 42] などが提案されている．しかし，これらは制御対象を入出力モデルで表しており入力や状態変数の制約について考慮していない点，または連続時間システムを対象としている点から本論文で提案するものとは本質的に異なる．また，電気自動車のスリップ抑制問題に対して離散時間状態空間モデルを用いた最適 PID ゲイン導出を行うもの [43] があるが，この手法では制御対象の状態空間モデルのシステム行列や出力行列の逆行列が必要であるためにスカラーなシステム以外で適用できるシステムが限られており，汎用的な手法として確立されていない．そこで本論文では，前章の手法 (文献 [44]) により導出されるような制御系設計用の離散時間状態空間モデルに対する，より汎用的なモデル予測型 PID 制御手法を提案する．すなわち，制御器の構造は従来と同じ PID 制御を用い，設計手法として MPC アルゴリズムを用いることで，制約条件を陽に考慮した最適な

PID ゲインを制御周期ごとに再計算しながら PID 制御を行うものである。

以下、本論文は次の通り構成される。2 節では主たる対象とする熱伝導システムのモデリングとシステムの制約について概説し、3 節で提案するモデル予測型 PID 制御について述べる。4 節では、提案するモデル予測型 PID 制御の有効性と汎用性について、熱伝導システムと 3 章の移動ロボットの協調運搬作業問題に適用した数値例により示す。さらに、5 節にてモデル予測型 2 自由度制御について述べ、同じく数値例を示す。6 節にてまとめと述べる。

## 5.2 PID 制御について

### 5.2.1 PID 制御

PID 制御は古くから実用化され、今なお多くの制御で用いられている制御手法である [45]。ここで、PID とは比例 (Proportional)、積分 (Integral)、微分 (Derivative) の頭字語を並べたもので、その名の通り比例動作、積分動作、微分動作という 3 つの動作を組み合わせた制御である。

まず、連続時間において基本形 PID 制御は、時刻  $t$  における入力を  $u(t)$  が、

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (5.1)$$

と表される。ここで、 $e(t)$  は目標値と制御対象の出力との間の偏差を示しており、目標値を  $y_{\text{ref}}(t)$ 、制御対象の出力を  $y(t)$  とすると  $e(t) = y_{\text{ref}}(t) - y(t)$  となる。

式 (5.1) の各項について説明する。右辺第一項  $K_P e(t)$  は比例項で、これは偏差に比例した制御入力を行う。すなわち、比例動作は偏差が大きければ大きいほど比例した大きな制御入力を与えることで偏差を縮めようと動作する。第二項の  $K_I \int_0^t e(\tau) d\tau$  は積分項で、過去からの偏差の積分値に応じて制御入力を行うことを示している。例えば、比例要素だけでは補償できない定常偏差が残る場合には偏差の積分値が増大するので、積分動作は定常偏差をなくすように動作する。三項目の  $K_D \frac{de(t)}{dt}$  は微分項で、偏差の変化量に応じて制御入力を行うことを示している。つまり将来の偏差の変化に対して作用し、これを抑制するような動作となる。各項に表れる 3 つの係数  $K_P, K_I, k_d$  を PID ゲインと呼ぶ。

さて、PID 制御器をディジタル計算機上で実装する場合、積分および微分を総和および差分にすることで離散化した式を用いる。これを離散時間 PID 制御とよび、時刻  $k$  の制

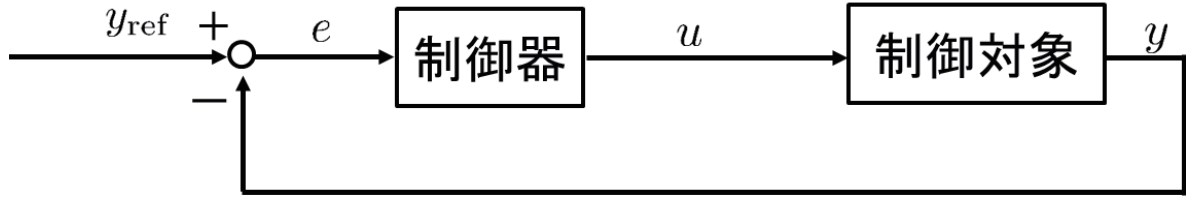


図 5.1 フィードバック制御系

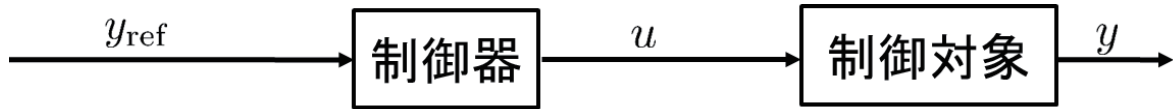


図 5.2 フィードフォワード制御系

御入力を  $u[k]$  とすると,

$$u[k] = K_P e[k] + K_I \sum_{i=0}^k e[i] + K_D (e[k] - e[k-1]) \quad (5.2)$$

となる.

制御系設計者が決められる部分は右辺 3 項に表れる PID ゲインのみである. このことから, 調整する部分が少なく, かつその意味も分かりやすく, 現場で試行錯誤的に設計するのに都合が良い. その反面, モデル予測制御のように制約条件を陽に考慮することは出来ない. また, 式 (5.1), (5.2) とともに制御入力の計算には偏差  $e$  しか用いないため, 出力  $y$  にすぐには反映されないような状況の変化があっても制御入力を変更できない. そのため, 熱伝導システムに見られるようなむだ時間の大きな制御対象に対し適切な制御が困難な場合がある.

### 5.2.2 2 自由度 PID 制御

PID 制御を含むフィードバック制御系のブロック線図を図 5.1 に, フィードフォワード制御系のブロック線図を図 5.2 に示す.

図 5.1 のフィードバック制御系では偏差をフィードバックする, つまり制御器は偏差に応じて制御入力を決定する. そのため, 適切な制御器を設計できれば, 予期しない外乱が発生して制御対象の出力が変化した場合に, その悪影響を抑制するよう制御器は偏差を小さくするよう動作する. 一方で, 出力の変化が目標値変更による適切な変化なのか, 外乱

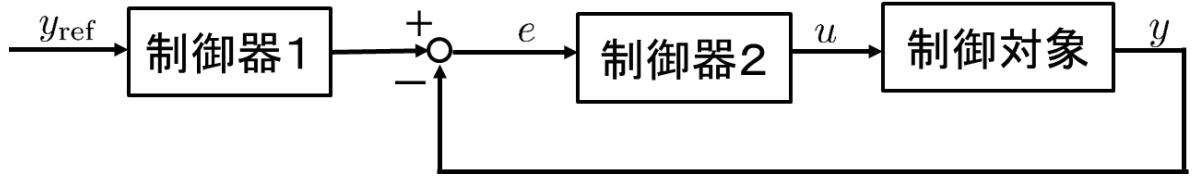


図 5.3 2 自由度制御系

による不要な影響なのかの区別できない．特に目標値変更は偏差の変化に対する影響が大きいため，本来不要な抑制機能が働きすぎてオーバーシュートを起こす恐れがある．

図 5.2 のフィードフォワード制御系では，制御器は目標値に応じて制御入力を決める．そのため，理想的には制御器として制御対象の逆システムを実装することで制御器と制御対象の動特性は相殺され，制御対象の出力は目標値と完全に一致する．しかし，外乱や制御対象にモデル誤差が生じている場合（これは現実的には回避できない）には，例えばフィードバック制御のように偏差情報からそれを知ることができないためにその影響を抑制できない．

これらフィードバック制御系，フィードフォワード制御系の問題点は，制御入力を決める制御器が偏差に対して，あるいは目標値に対してという 1 か所でしか設計できないことに起因する．このような制御系を 1 自由度制御系と呼ぶ．これに対し，図 5.3 のブロック線図に示すような構成の制御系を 2 自由度制御系 [46] という．これは偏差および目標値という 2 か所それぞれに対し制御器を設計できる制御系である．

2 自由度 PID 制御 [39] とは，従来の 1 自由度 PID 制御がフィードバック制御系であることによってトレードオフの関係にあった目標値追従性能と外乱抑制性能の 2 つの性能を同時に改善できるよう，前述の 2 自由度制御系の考え方に基づき拡張された制御手法である．2 自由度 PID 制御には様々な定式化があるが，本研究では比例動作および微分動作に対し 2 自由度化する構成 [47] を対象とする．基本形 PID 制御の式を直接変更してシンプルに表現した式を以下に示す．これは要素分離形 2 自由度 PID 制御とよばれる．

$$\begin{aligned}
 u[k] = & K_P \{ (1 - K_{rp})r[k] - y[k] \} + \\
 & K_I \sum_{i=0}^k \{ r[i] - y[i] \} + \\
 & K_D \{ (1 - K_{rd})r[k] - y[k] - (1 - K_{rd})r[k-1] + y[k-1] \}
 \end{aligned} \tag{5.3}$$

この 2 自由度 PID 制御の式では，1 自由度の基本形 PID 制御の制御入力の式 (5.2) と

比べると、比例項および微分項について目標値  $r[k]$ ,  $r[k-1]$  に対する設計係数  $K_{rp}$ ,  $K_{rd}$  が追加されている。この各係数について  $K_{rp} = 0, K_{rd} = 0$  の場合を考えると、式 (5.3) は基本形 PID 制御の式 (5.2) と同一になることから、2 自由度 PID 制御が基本形 PID 制御を含んだものであることがわかる。また、 $K_{rp} = 0, K_{rd} = 1$  のときは微分項に関して目標値が消去され微分先行形 PID 制御に、 $K_{rp} = 1, K_{rd} = 1$  のときは微分項および比例項に関して目標値が消去され I-PD 制御になることから、 $K_{rp}, K_{rd}$  により PID 制御の構造が変化することがわかる。 $K_{rp}, K_{rd}$  を 2 自由度化パラメータと呼ぶ。この 2 自由度 PID 制御の式は、等価変換を行うことで 2 自由度 PID 制御系を従来の PID 制御系に対する要素追加で構成できる [47]。例えば、図 5.3 の様に基本形 PID 制御系に目標値に対するフィルタ要素を追加することで構成できる。

## 5.3 熱伝導システムのモデル

### 5.3.1 線形差分式モデル

本節では、複数部材から構成されるある種の熱伝導システムを制御対象とする。熱の伝わる部材が複数ある際に、個々の部材の特性や熱の移動時間を考慮し、状態変数は部材上の適当な点の温度をとることになると、部材間の熱のやり取りなどによる熱の移動時間や、逆にしばらく時間が経過しても残る影響を考慮する必要があるが生じる。このような温度動特性は、システムの熱伝導方程式 [48, 49] に対する有限差分法により導出される離散時間の温度挙動モデルを用いたシステム同定により、(5.4) 式に示す複数の異なるむだ時間を持つ線形差分式モデルとして表すことができる。

$$\begin{bmatrix} x_1(l+d_1) \\ x_2(l+d_2) \\ \vdots \\ x_n(l+d_n) \end{bmatrix} = A(l)\mathbf{x}(l) + \mathbf{b}(l)u(l) + D(l)\boldsymbol{\alpha} \quad (5.4)$$

$$y(l) = \mathbf{c}\mathbf{x}(l) \quad (5.5)$$

$$\begin{aligned} y_M(l) &= \mathbf{c}_M\mathbf{x}(l) \\ &= \max \{x_1(l), x_2(l), \dots, x_n(l)\} \end{aligned} \quad (5.6)$$

$l$  は離散時刻ステップを表し、その離散化ステップ幅を  $t_l$  とする。(5.4) 式について、状態変数ベクトル  $\mathbf{x}(l) = (x_1(l), \dots, x_n(l))^T$  として熱伝導システムの適当な部位の温度をとるものとする (本論文においてはベクトル表記は  $\mathbf{x}$  のように太字で表す)。このとき、状態変数の各要素  $x_i (i = 1, \dots, n)$  はむだ時間ステップ  $d_i$  を持つ。また、制御入力  $u(l)$

はスカラーで、単一の加熱ヒータのデューティ比 (1 制御周期時間に対するヒータ ON の時間の割合) をとる.  $\alpha$  は、例えば外気温や外部から投入される被加熱部材の温度のように一定温度とみなせるある温度からなるベクトルである. (5.5) 式の出力方程式における制御出力  $y(l)$  は制御したい部位の温度を表す. (5.6) 式の出力方程式の  $y_M(l)$  は  $x(l)$  の表すシステム各部位の温度のうち最高温度となるものを表す. どの部位が最高温度になるかは熱伝導システムの構造などからあらかじめ既知であるとする.

熱伝導システムに被加熱部材が投入されるとシステム全体の伝熱特性が変化する. 投入される被加熱部材類がそれぞれの材質などにより異なる熱特性を持つとすると, 被加熱部材投入の有無やどのような部材が投入されたかにより, 熱伝導システム全体の伝熱特性にはいくつかのパターンが生じる. パターンが  $p$  通りのとき, これらをシステム行列  $A$ , 入力行列  $b$  および定数温度の係数行列  $D$  の組に対し, 上付き添え字  $(i)$  を用いて (5.7) 式のように表すものとする. また,  $i$  をプラントインデックスと呼ぶものとする.

$$(A(l), b(l), D(l)) \in \left\{ (A^{(i)}, b^{(i)}, D^{(i)}) \mid i = 1, \dots, p \right\} \quad (5.7)$$

### 5.3.2 制約

想定する熱伝導システムには, 考慮すべき以下 2 つの制約条件が存在する.

第一に最高温度制約である. 加熱を生じる熱伝導システムにおいては, 先に述べたように構成部材の許容する温度制約を遵守することが重要であり,  $T_M$  を許容される最高温度とすると, (5.8) 式を常に満足しなければならない.

$$y_M(l) \leq T_M \quad (5.8)$$

第二に入力制約である. 上述の (5.4) 式において制御入力  $u(l)$  はデューティ比であり, したがって 0 以上 100 以下の値であるから (5.9) 式で表される制約を持つ.

$$0 \leq u(l) \leq 100 \quad (5.9)$$

## 5.4 モデル予測型 PID 制御

本節では, 提案するモデル予測型 PID 制御の設計法について述べる.



### 5.4.1 制御系設計モデル

モデル予測型 PID 制御では、MPC のアルゴリズムに従い、現在時刻  $l$  からある予測区間  $\eta H_p$  にわたる  $l + \eta H_p$  までの温度挙動を制御系設計用モデルを用いて予測する ( $\eta$  については後に述べる). しかし、(5.4) 式の左辺は状態変数  $\mathbf{x}$  の各要素それぞれが異なるむだ時間をもっており、一般的な離散時間状態方程式  $\mathbf{x}(l+1) = A^{(i)}\mathbf{x}(l) + \mathbf{b}^{(i)}u(l) + D^{(i)}\boldsymbol{\alpha}$  の形ではないため、そのままでは設計モデルとして扱いにくい. (5.4) 式は、必要な過去の温度まで状態変数に含めた拡大系を考えることで、一般的な離散時間状態方程式として表現することはできる. しかし、熱伝導システムでは熱の移動時間をむだ時間ステップ  $d_i$  として表すため、システムによっては拡大系の次数が元の (5.4) 式の次数より非常に大きくなってしまふことがあり、この場合、やはり設計モデルとして好ましくない. これを回避するために、文献 [44] の手法により導出した近似モデルを制御系設計モデルとして利用する. この方法は熱伝導システムの温度挙動のように状態変数の変動がそれほど激しくない場合に精度良い近似モデルを導出でき、今回の場合に有効であることが期待できる. 文献 [44] では定数項  $D^{(i)}\boldsymbol{\alpha}$  を扱っていないが、 $\boldsymbol{\alpha}$  を状態変数に含めた拡大系

$$\begin{bmatrix} \mathbf{x}(l+1) \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} A^{(i)} & D^{(i)} \\ \mathbf{0} & I \end{bmatrix} \begin{bmatrix} \mathbf{x}(l) \\ \boldsymbol{\alpha} \end{bmatrix} + \begin{bmatrix} \mathbf{b}^{(i)} \\ \mathbf{0} \end{bmatrix} u(l)$$

に対して近似モデルを導出した後、状態変数から定数項を外せばよい. (5.4) 式は入力にむだ時間を持たないため、これより (5.4) 式と同じ次数の近似モデル (5.10) が得られる.

$$\mathbf{x}(l+1) = A_m^{(i)}\mathbf{x}(l) + \mathbf{b}_m^{(i)}u(l) + D_m^{(i)}\boldsymbol{\alpha} \quad (5.10)$$

なお、(5.4) 式および (5.10) 式の時刻ステップ幅  $t_l$  はモデリング精度を高めるため微小な時間をとる. したがって、ヒータ制御周期  $t_k$  に対して  $t_l$  がごく短時間となることがありうる. ある自然数  $\eta$  を用いて  $\eta = t_k/t_l$  であるとき、(5.10) 式のステップ幅をヒータの制御周期に合わせた以下の (5.11) 式をあらためて設計モデルとすることもできる. この場合、サンプリング時刻を離散時刻ステップ幅  $t_l$  としたときの現在時刻ステップ  $l$  に対し、制御周期  $t_k$  としたときの現在時刻ステップを  $k$  で表せば、時刻ステップ  $l + \eta p$  は

$k + p$  と等しい．混同しないように，前者は丸括弧  $(\cdot)$  で，後者を角括弧  $[\cdot]$  で表記する．

$$\begin{aligned}
 \boldsymbol{x}[k+1] &= G_m[k]\boldsymbol{x}[k] + \boldsymbol{h}_m[k]u[k] + F_m[k]\boldsymbol{\alpha} \\
 &= G_m^{(i)}\boldsymbol{x}[k] + \boldsymbol{h}_m^{(i)}u[k] + F_m^{(i)}\boldsymbol{\alpha} \\
 &\quad i = 1, \dots, p \\
 G_m^{(i)} &= \left(A_m^{(i)}\right)^\eta \\
 \boldsymbol{h}_m^{(i)} &= \sum_{j=0}^{\eta-1} \left(A_m^{(i)}\right)^j \boldsymbol{b}^{(i)} \\
 F_m^{(i)} &= \sum_{j=0}^{\eta-1} \left(A_m^{(i)}\right)^j D^{(i)}
 \end{aligned} \tag{5.11}$$

ここで， $\left(A_m^{(i)}\right)^0$  は単位行列とする．(5.10)，(5.11) 式はともに状態変数が (5.4) 式と同一なので，それぞれの出力方程式は (5.5)，(5.6) 式と同じである．以下では (5.11) 式を設計モデルとして用いる．

### 5.4.2 MP-PID 制御系設計

現在時刻ステップが  $k$  のとき，予測区間  $j = 0$  から  $j = H_p - 1$  にわたる将来の予測状態変数  $\boldsymbol{x}[k + j + 1]$  は，(5.11) 式の左辺を右边第一項に繰り返し代入することで，

$$\prod_{p=p_1}^{p_2} G_m[k+p] = G_m[k+p_2]G_m[k+p_2-1] \cdots G_m[k+p_1]$$

を用いて (5.12) 式となる.

$$\begin{aligned}
 & \cdot j = 0 \text{ のとき} \\
 & \quad \mathbf{x}[k+j+1] = G_m[k]\mathbf{x}[k] + \mathbf{h}_m[k]u[k] + F_m[k]\boldsymbol{\alpha} \\
 & \cdot 1 \leq j \leq H_p - 1 \text{ のとき} \\
 & \quad \mathbf{x}[k+j+1] = \\
 & \quad \left( \prod_{p=0}^j G_m[k+p] \right) \mathbf{x}[k] + \mathbf{h}_m[k+j]u[k+j] \\
 & \quad + \sum_{q=0}^{j-1} \left( \prod_{p=q+1}^j G_m[k+p] \right) \mathbf{h}_m[k+q]u[k+q] \\
 & \quad + \sum_{q=0}^{j-1} \left( \prod_{p=q+1}^j G_m[k+p] \right) F_m[k+q]\boldsymbol{\alpha} + F_m[k+j]\boldsymbol{\alpha}
 \end{aligned} \tag{5.12}$$

これより, 制御出力温度と最高温度はそれぞれ以下の式 (5.13), (5.14) で表される.

$$y[k+j+1] = \mathbf{c}\mathbf{x}[k+j+1] \tag{5.13}$$

$$y_M[k+j+1] = \mathbf{c}_M\mathbf{x}[k+j+1] \tag{5.14}$$

さて,  $j = 0, \dots, H_p - 1$  における, PID 制御による各時刻の制御入力  $u[k+j]$  は, 制御出力  $y[k+j]$  とその目標温度  $y_{\text{ref}}[k+j]$  に PID ゲイン ( $K_P, K_I, K_D$ ) を用いて,

$$e[k+j] = y_{\text{ref}}[k+j] - y[k+j] \tag{5.15}$$

$$\begin{aligned}
 u[k+j] &= K_P e[k+j] + K_I \sum_{s=0}^{k+j} e[s] \\
 &\quad + K_D (e[k+j] - e[k+j-1])
 \end{aligned} \tag{5.16}$$

となる. これを速度型 PID 制御の式に変形すると,

$$\begin{aligned}
 \Delta u[k+j] &= u[k+j] - u[k+j-1] \\
 &= K_P (e[k+j] - e[k+j-1]) + K_I e[k+j] \\
 &\quad + K_D (e[k+j] - 2e[k+j-1] + e[k+j-2]) \\
 &= K_\alpha e[k+j] + K_\beta e[k+j-1] + K_\gamma e[k+j-2]
 \end{aligned} \tag{5.17}$$

と表せる．ただし，以下の関係式 (5.18) を用いた．

$$\begin{bmatrix} K_\alpha \\ K_\beta \\ K_\gamma \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} K_P \\ K_I \\ K_D \end{bmatrix} \quad (5.18)$$

モデル予測型 PID 制御では，まず一時的なゲイン  $\tilde{\theta} = (K_\alpha \ K_\beta \ K_\gamma)^T$  について最適ゲインを計算する．時刻ステップ  $k$  における最適化問題の評価関数としては，MPC でよく用いられる  $e$  と  $u$  の重み付き二乗和  $J_{LQ}$  を用いた．これを (5.19) 式に示す． $e$  と  $u$  を両方評価することで，制御性能と省エネルギー性のトレードオフを考慮した制御を行える．

$$J_{LQ}[k] = \sum_{j=0}^{H_p-1} (q_j e[k+j+1]^2 + r_j u[k+j]^2) \quad (5.19)$$

$\mathbf{x}[k]$ ,  $\boldsymbol{\alpha}$  が既知のとき，(5.12) 式から (5.15) 式より， $e[k+j+1]$ ,  $y_M[k+j+1]$  は  $u[k+q]$  ( $q = 0, \dots, j$ ) によって定まることがわかる．さらに，(5.16), (5.17) 式から， $u[k+q]$  は  $e[k+q-2]$  から  $e[k+q]$  までの 3 つの値を用いて， $\tilde{\theta}$  の関数になっている．したがって， $e[k+j+1]$ ,  $y_M[k+j+1]$ ,  $u[k+j]$ , ( $j = 0, \dots, H_p - 1$ ) および  $J_{LQ}$  は全て  $\tilde{\theta}$  の関数となる．以上より，時刻  $k$  で解くべき PID ゲイン最適化問題は，

$$\begin{aligned} & \min_{\tilde{\theta}} J_{LQ}(\tilde{\theta}) \\ & s.t. \quad \text{Eqs. (5.12) - (5.17)} \\ & \quad y_M[k+j+1] \leq T_M \\ & \quad 0 \leq u[k+j] \leq 100 \\ & \quad j = 0, \dots, H_p - 1 \end{aligned} \quad (5.20)$$

となる． $J_{LQ}$  は  $\tilde{\theta}$  の要素に関して非線形関数になるので，非線形計画法や，あるいは計算機の能力に応じた刻み幅で離散化される  $\tilde{\theta}$  を全探索することで解くことができる．時刻  $k$  における最適化問題 (5.20) の解  $\tilde{\theta}_{opt} = (K_{\alpha opt} \ K_{\beta opt} \ K_{\gamma opt})^T$  が得られたら，(5.18) 式より

$$\begin{bmatrix} K_{Popt} \\ K_{Iopt} \\ K_{Dopt} \end{bmatrix} = \left( \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & -2 \\ 0 & 0 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} K_{\alpha opt} \\ K_{\beta opt} \\ K_{\gamma opt} \end{bmatrix} \quad (5.21)$$

により，時刻  $k$  における最適 PID ゲイン ( $K_{Popt}$ ,  $K_{Iopt}$ ,  $K_{Dopt}$ ) が得られる．次の制御周期時刻  $k+1$  に移ったら，その時刻をふたたび時刻  $k$  として改めて最適化問題 (5.20)

を解き、(5.21) 式により最適ゲインを求めることを制御周期ごとに繰り返す。この手法では文献 [43] で用いていた  $\mathbf{c}^{-1}, \left(G_m^{(i)}\right)^{-1}$  が不要なので、より一般的なシステムに適用できる。

## 5.5 数値例

本節では、前章で示したモデル予測型 PID 制御を、最高温度制約を持つある種の熱伝導システム例に適用した数値シミュレーションの結果を示す。そのうち、3 章にて示した移動ロボットの強調運搬作業問題に対してこのモデル予測型 PID 制御を適用した数値シミュレーションの結果を示す。

### 5.5.1 熱伝導システムに対する適用結果

制御の目的は、熱伝導システム中の特定部位の温度を制御出力温度  $y$  として、これをある目標温度  $y_{\text{ref}}$  へ制御することである。ただし、加熱途中で、異なる伝熱特性を持つ 2 種類の被加熱部材がある周期でシステム中へ投入される。これによりシステム内の伝熱特性に変化が生じるが、そのような状況下においてもできるだけ早く目標温度へ到達し、かつ目標温度付近で安定した温度挙動となることが求められる。そこで性能指標として、およそ目標温度まで到達する 5[s] 以降の温度応答に対し、制御出力温度  $y$  と目標温度  $y_{\text{ref}}$  との偏差の絶対値  $|e|$  の時間和、および  $y$  の温度分散値を用いて、提案する MP-PID 制御と固定ゲインを用いる従来の PID 制御、スタンダードな MPC の性能を比較する。

システムの温度動特性を示す差分式モデル、制御出力温度  $y$  と最高温度  $y_M$  を示す出力方程式はそれぞれ (5.22)～(5.24) 式の通りである。2 種の被加熱部材のいずれかが投入されているときと、どちらも投入されていないときがあるので伝熱特性は 3 パターンとなる。また、差分方程式の離散化ステップ幅  $t_l$  は 40[ms]、システムの制御周期  $t_k$  は 200[ms] であるとする。

$$\begin{aligned}
 & \begin{bmatrix} x_1(l+2) & x_2(l+4) & x_3(l+3) & x_4(l+7) & x_5(l+2) \\ x_6(l+15) & x_7(l+5) & x_8(l+1) & x_9(l+1) \end{bmatrix}^T \\
 & = A(l)\mathbf{x}(l) + \mathbf{b}(l)u(l) + D(l) \begin{bmatrix} T_{\text{mat}} \\ T_{\text{amb}} \end{bmatrix}
 \end{aligned} \tag{5.22}$$

$$\left( A(l), \mathbf{b}(l), D(l) \right) \in \left\{ \left( A^{(i)}, \mathbf{b}^{(i)}, D^{(i)} \right) \mid i = 1, 2, 3 \right\}$$

$$y(l) = \mathbf{c}\mathbf{x}(l) = x_1(l) \tag{5.23}$$

$$y_M(l) = \mathbf{c}_M\mathbf{x}(l) = x_8(l) \tag{5.24}$$

ここで,  $T_{\text{mat}}$  は被加熱部材のシステム投入前温度,  $T_{\text{amb}}$  は外気温である.  $\left( A^{(i)}, \mathbf{b}^{(i)}, D^{(i)} \right)$

の値は以下の通りとした.

$$A^{(1)} = \begin{bmatrix} 0 & 0.779221 & 0 & 0 & 0 & 0 & 0 & 0 \\ 182 & 0 & 0 & 0 & 0 & 0.367451 & 0 & 0 \\ 0.980 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.847149 & 0 & 0 & 0 & 0 \\ 0 & 0.282718 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.998 & 0 & 0 & 0 \\ 23 & 0 & 0 & 0 & 0 & 0.973 & 3 & 0 \\ 409 & 0 & 0 & 0 & 0 & 0 & 0.409182 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15.985 \end{bmatrix} \times 10^{-3}$$

$$A^{(2)} = \begin{bmatrix} 0 & 0.720212 & 0 & 0 & 0 & 0 & 0 & 0 \\ 182 & 0 & 0 & 0 & 0 & 0.367451 & 0 & 0 \\ 0.980 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.847149 & 0 & 0 & 0 & 0 \\ 0 & 0.102800 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.998 & 0 & 0 & 0 \\ 23 & 0 & 0 & 0 & 0 & 0.973 & 3 & 0 \\ 409 & 0 & 0 & 0 & 0 & 0 & 0.409182 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15.985 \end{bmatrix} \times 10^{-3}$$

$$A^{(3)} = \begin{bmatrix} 0 & 0.75077 & 0 & 0 & 0 & 0 & 0 & 0 \\ 182 & 0 & 0 & 0 & 0 & 0.367451 & 0 & 0 \\ 0.980 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.847149 & 0 & 0 & 0 & 0 \\ 0 & 0.100772 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.998 & 0 & 0 & 0 \\ 23 & 0 & 0 & 0 & 0 & 0.973 & 3 & 0 \\ 409 & 0 & 0 & 0 & 0 & 0 & 0.409182 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15.985 \end{bmatrix} \times 10^{-3}$$

$$\mathbf{b}^{(1)} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 662 \ 0]^T \times 10^{-3}$$

$$\mathbf{b}^{(2)} = \mathbf{b}^{(1)}$$

$$\mathbf{b}^{(3)} = \mathbf{b}^{(1)}$$

$$D^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & 4 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T \times 10^{-3}$$

$$D^{(2)} = \begin{bmatrix} 68 & 0 & 0 & 0 & 98 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & 4 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T \times 10^{-3}$$

$$D^{(3)} = \begin{bmatrix} 173 & 0 & 0 & 0 & 128 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & 4 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T \times 10^{-3}$$

このモデルに対し、文献 [44] の手法を適用して導出される近似モデルは (5.10) 式の形で

$$A_m^{(1)} = \begin{bmatrix} 500 & 0.389111 & 0 & 0 & 0 & 0 & 0 & 0 \\ 46750 & 0 & 0 & 0 & 0 & 0 & 91113 & 0 \\ 0.326667 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.857121 & 21 & 0 & 0 & 0 & 0 \\ 0 & 0.141359 & 500 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1999 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0.994 & 1 & 0 \\ 409 & 0 & 0 & 0 & 0 & 0 & 0.409182 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15985 \end{bmatrix} \times 10^{-3}$$

$$A_m^{(2)} = \begin{bmatrix} 500 & 0.360106 & 0 & 0 & 0 & 0 & 0 & 0 \\ 46750 & 0 & 0 & 0 & 0 & 0 & 91113 & 0 \\ 0.326667 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.857121 & 21 & 0 & 0 & 0 & 0 \\ 0 & 0 & 51400 & 500 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1999 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0.994 & 1 & 0 \\ 409 & 0 & 0 & 0 & 0 & 0 & 0.409182 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15985 \end{bmatrix} \times 10^{-3}$$

$$A_m^{(3)} = \begin{bmatrix} 500 & 0.37539 & 0 & 0 & 0 & 0 & 0 & 0 \\ 46750 & 0 & 0 & 0 & 0 & 0 & 91113 & 0 \\ 0.326667 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.857121 & 21 & 0 & 0 & 0 & 0 \\ 0 & 0 & 50386 & 500 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1999 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0.994 & 1 & 0 \\ 409 & 0 & 0 & 0 & 0 & 0 & 0.409182 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 15985 \end{bmatrix} \times 10^{-3}$$

$$\mathbf{b}_m^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 662 & 0 \end{bmatrix}^T \times 10^{-3}$$

$$\mathbf{b}_m^{(2)} = \mathbf{b}_m^{(1)}$$

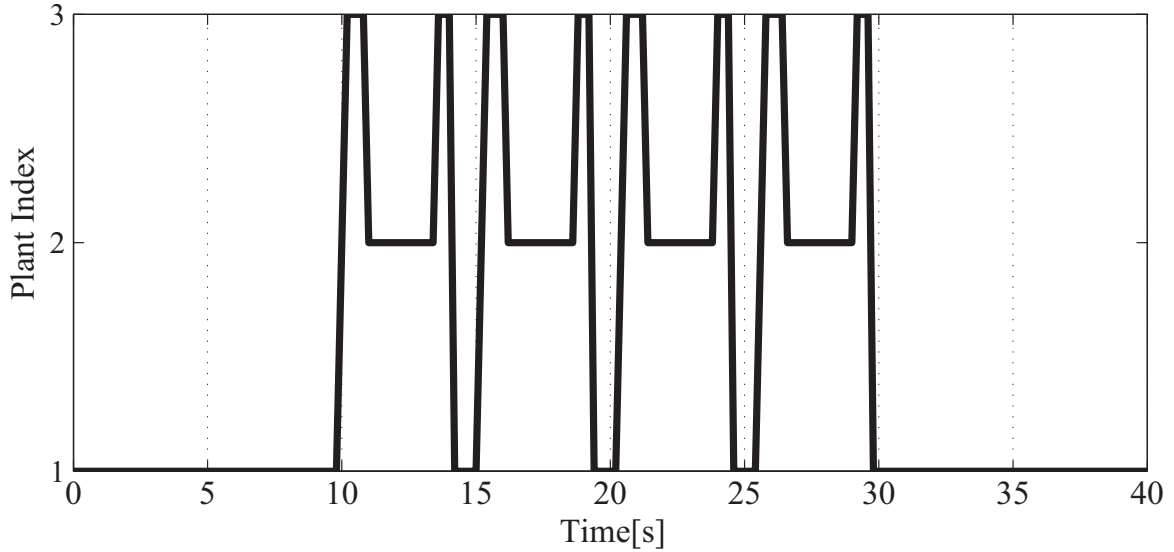
$$\mathbf{b}_m^{(3)} = \mathbf{b}_m^{(1)}$$

$$D_m^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 1 & 0 & 0 & 2 & 0 & 0 \end{bmatrix}^T \times 10^{-3}$$

$$D_m^{(2)} = \begin{bmatrix} 34 & 0 & 0 & 0 & 49 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 1 & 0 & 0 & 2 & 0 & 0 \end{bmatrix}^T \times 10^{-3}$$

$$D_m^{(3)} = \begin{bmatrix} 86 & 0 & 0 & 0 & 64 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 1 & 0 & 0 & 2 & 0 & 0 \end{bmatrix}^T \times 10^{-3}$$



図 5.4 プラントインデックス  $i$  の時間変化

となる．さらに，時刻ステップ幅が制御周期と一致するように  $\eta = t_k/t_l = 5$  として (5.11) 式を適用した以下の (5.25) 式を設計モデルとする．

$$\mathbf{x}[k+1] = \mathbf{G}_m[k]\mathbf{x}[k] + \mathbf{h}_m[k]u[k] + \mathbf{F}_m[k] \begin{bmatrix} T_{\text{mat}} \\ T_{\text{amb}} \end{bmatrix} \quad (5.25)$$

なお，システムモデル (5.22) と設計モデル (5.25) との各出力温度  $y(=x_1)$ ,  $y_M(=x_8)$  の適合率 [50] はそれぞれ 90.419[%], 97.033[%] となり，その他の各温度についても 83.582[%] ~ 99.490[%] の範囲の値となった．以下のシミュレーション結果からもわかる通り，設計モデルとして十分な精度が得られた．

以上の準備のもと，性能検証として以下では 3 通りの数値シミュレーション結果を示す．プラント伝熱特性を示すプラントインデックス  $i \in \{1, 2, 3\}$  の時間変化グラフは図 5.4 に示す通りで，3 通り全てのシミュレーションで同じとする．モデル予測型 PID 制御では， $-5 \leq K_P \leq 5$ ,  $5 \leq K_I \leq 5$ ,  $5 \leq K_D \leq 5$  という各ゲインの取りうる領域を与え，これをそれぞれ幅 0.2 のグリッドに区切って全探索により最適ゲインを求めた．予測区間は制御入力  $u$  が制御出力温度  $y$  に反映されるまでの時間を考慮して  $H_p = 15$  とし，(5.19) 式における重みは，制御性能を重視して  $w_q = 1, w_u = 0.001$  ( $j = 0, \dots, 14$ ) とした．一方，従来の PID 制御については PID ゲインを試行錯誤的に求めた．

#### Case 1. 最高温度制約無し

まず，目標値追従性についてシミュレーションを行った．目標温度を  $y_{\text{ref}} = 120[^\circ\text{C}]$ ,

初期温度は全て  $20[^\circ\text{C}]$  とし、最高温度制約 (5.8) は設けていない。MP-PID 制御と MPC では入力制約 (5.9) を与え、PID 制御においては (5.16) 式による入力が 0 以下または 100 以上を超える場合にはそれぞれ 0, 100 とした。PID ゲインは  $K_P = 2.6, K_I = 1.4, K_D = 0.8$  とした。以上において、MP-PID 制御と PID 制御、MPC を適用した際の制御出力温度ならびに最高温度の時間応答グラフを図 5.5 に、制御ごとの評価指標値を表 5.1 に示す。

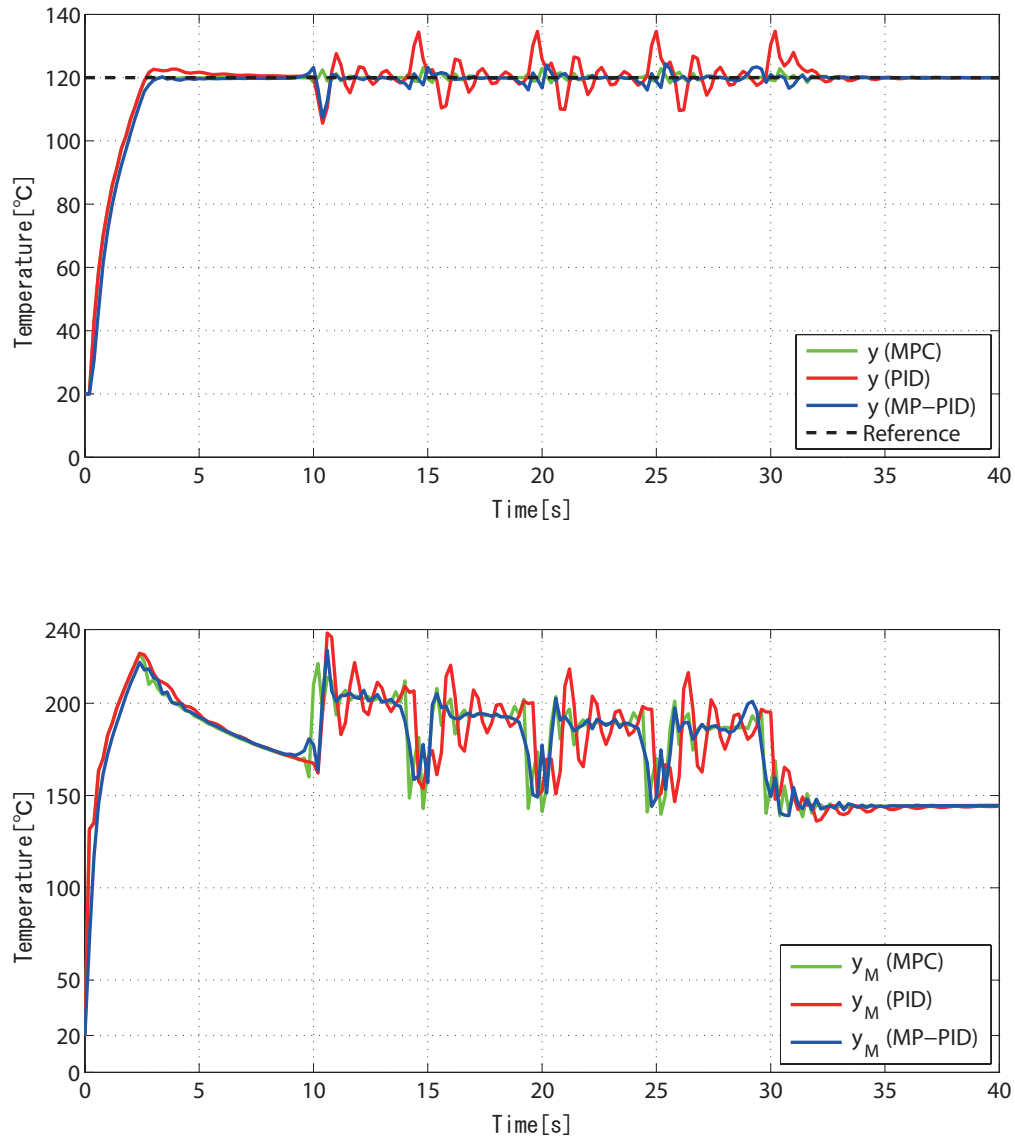
図 5.5 の結果から、特に PID 制御ではシステムの伝熱特性が切り替わるタイミングで温度変動が生じているが、表 5.1 の数値を見ても MP-PID 制御では MPC ほどではないが PID 制御よりも指標値が小さくなっており、変動が抑えられていることがわかる。これは適切なゲインが制御周期ごとに選ばれたためである。また、図 5.6 は制御入力とした各制御ごとのデューティ比を示している。これより、MP-PID 制御で入力制約 (5.9) が守られていることがわかる。

表 5.1 Case 1 における制御性能評価

	MP-PID	PID	MPC
Average of absolute error $[^\circ\text{C}]$	0.192	0.668	0.043
Standard deviation $[^\circ\text{C}]$	1.623	4.202	0.753
Maximum temperature $[^\circ\text{C}]$	228.767	238.079	227.059

## Case 2. 最高温度制約の付加

次に、Case 1 の設定に、許容最高温度が  $210[^\circ\text{C}]$  であるという制約を加えたシミュレーションを行った。制御対象が加熱するシステムであることから安全マージンを考慮し、MPC と MP-PID 制御では  $T_M$  を  $210[^\circ\text{C}]$  ではなく  $209[^\circ\text{C}]$  とした (5.8) 式と (5.9) 式の制約のもとで制御した。このときのそれぞれの制御における制御出力温度ならびに最高温度の時間応答グラフを図 5.7、評価指標値を表 5.2 に示す。なお、PID 制御におけるゲインは Case 1 と同様に試行錯誤的に  $K_P = 3.0, K_I = 0.12, K_D = 0.6$  とした。これは、(5.8) 式を満足するゲインとして選んでいる。しかし、PID 制御では一般に 1 入出力系を扱うため、今回のように  $y_M$  の制約を満足しつつ  $y$  の温度挙動を適切にするという 2 出力を同時に所望する挙動とするようなゲインを求めることは大変困難となる可能性があり、実際、今回の数値例では  $y_M$  の制約を満たす PID ゲインでは  $y$  の制御性能は大きく劣化している。これに対して、図 5.7 および表 5.2 の結果から、Case 1 と同様に、MP-PID 制御では PID 制御よりも制御指標値が良い値になっている。さらに、最高温度は  $210[^\circ\text{C}]$

図 5.5 Case 1 での  $y$  の時間応答 (上) と  $y_M$  の時間応答 (下)

を超えておらず (5.8) 式を満足していることも確認できた。

### Case 3. 初期温度の変更

最後に、Case 2 の設定で初期温度が 20[°C] から 10[°C] に変動した場合のシミュレーションを行った。このときのそれぞれの制御における制御出力温度ならびに最高温度の時

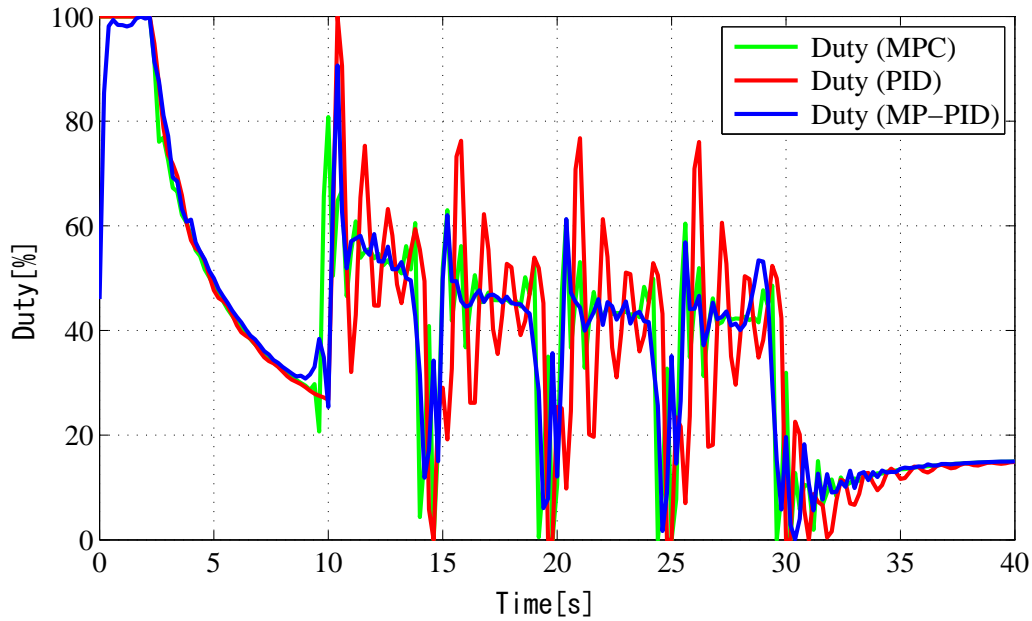
図 5.6 Case 1 での入力  $u$  の時間応答

表 5.2 Case 2 における制御性能評価

	MP-PID	PID	MPC
Average of absolute error [°C]	0.016	4.792	0.031
Standard deviation [°C]	2.503	11.739	0.703
Maximum temperature [°C]	209.003	209.096	209.384

間応答グラフを図 5.8, 評価指標値を表 5.3 に示す.

ここで, PID 制御におけるゲインは Case 2 と同じ  $K_P = 3.0, K_I = 0.12, K_D = 0.6$  とした. このゲインは初期温度  $20[^\circ\text{C}]$  の条件下におけるゲインであり, Case 3 では初期温度が  $10[^\circ\text{C}]$  と異なっているために最高温度が  $210[^\circ\text{C}]$  を超過してしまっている. これに対し, MP-PID 制御では初期温度が変化しても温度制約を満足している.

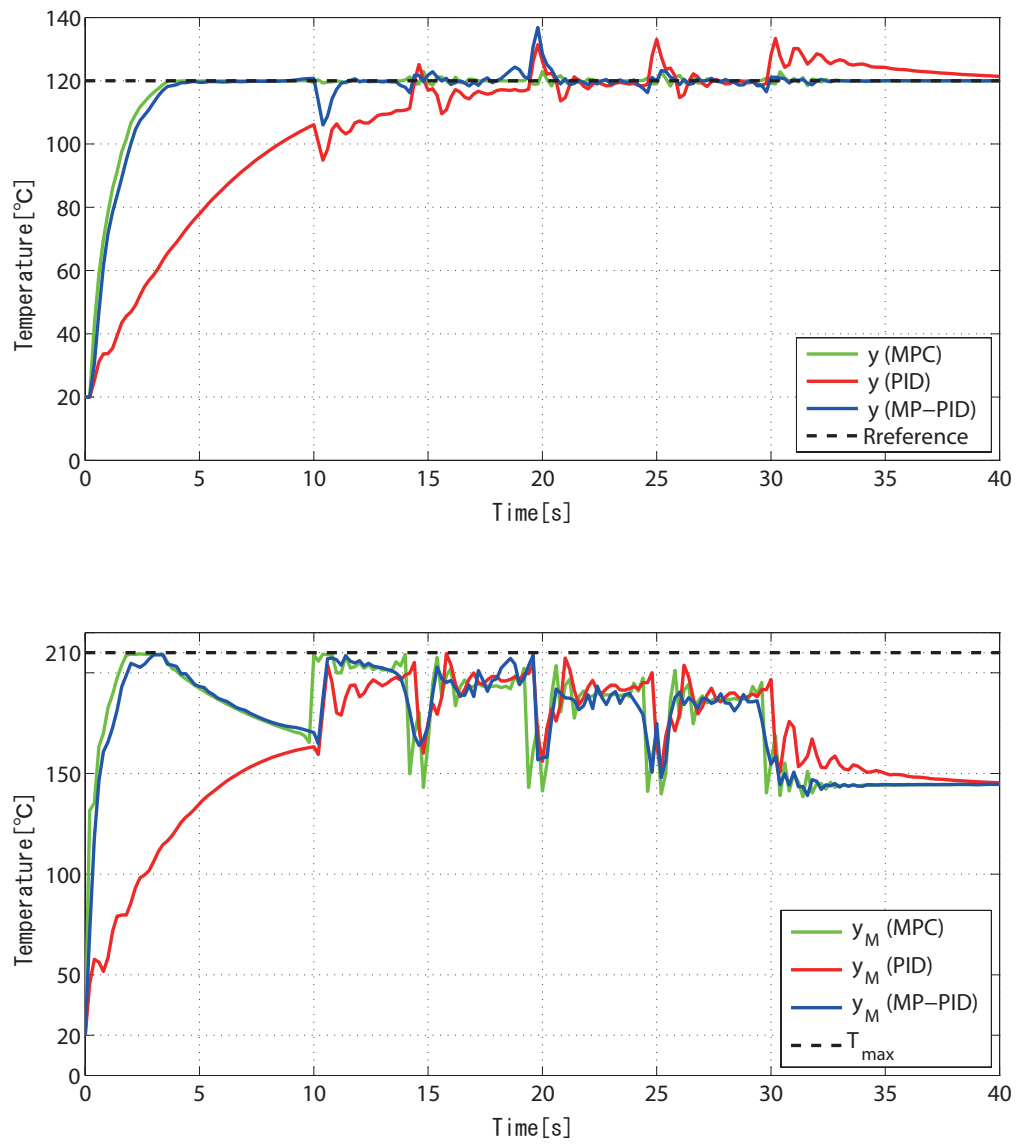


図 5.7 Case 2 での  $y$  の時間応答 (上) と  $y_M$  の時間応答 (下)

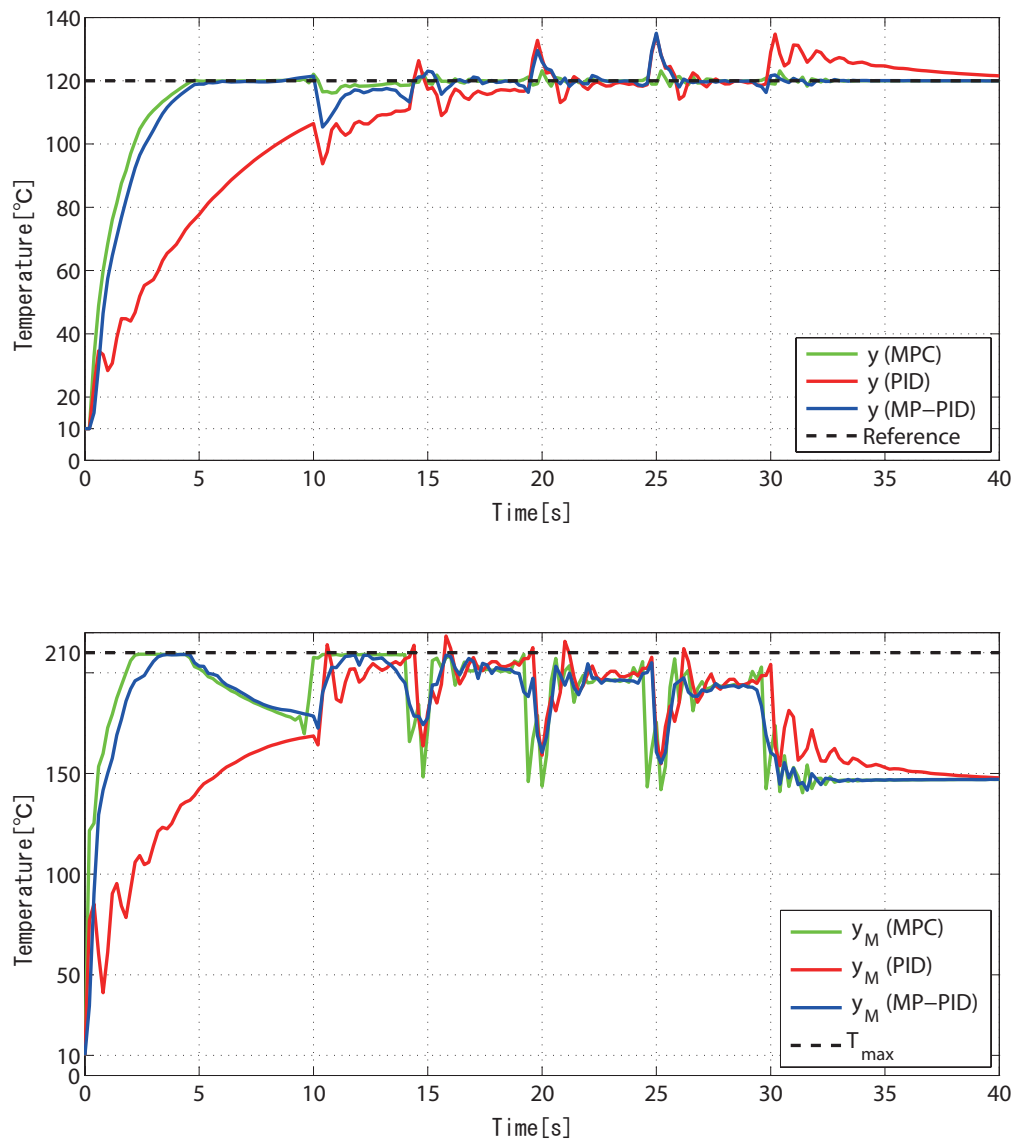


図 5.8 Case 3 での  $y$  の時間応答 (上) と  $y_M$  の時間応答 (下)

表 5.3 Case 3 における制御性能評価

	MP-PID	PID	MPC
Average of absolute error [°C]	0.409	4.61	0.154
Standard deviation [°C]	2.968	11.927	0.979
Maximum temperature [°C]	209.147	218.320	209.295

### 5.5.2 協調運搬作業問題に対する適用結果

次に、3 章の協調運搬作業問題に対してモデル予測型 PID 制御を適用する。左右輪の各トルク制御にそれぞれ PID 制御器を用いることとし、右輪トルクは右輪目標速度との偏差、左輪トルクは左輪目標速度との偏差をそれぞれの PID 制御器への入力とした。

$$u_i[k] = K_{Pi}e_i[k] + K_{Ii} \sum_{j=0}^k e_i[j] + K_{Di}(e_i[i] - e_i[k-1]) \quad i = r, l$$

$$e_r = p_{v_{\text{ref}},r} - p_{v_f,r}$$

$$e_l = p_{v_{\text{ref}},l} - p_{v_f,l}$$

ここで、右輪速度  $p_{v_f,r}$ 、左輪速度  $p_{v_f,l}$  は、重心と車輪の距離  $\tau$  を用いて以下のように並進速度  $p_{v_f}$ 、回転角速度  $p_{\omega_f}$  から算出できる。

$$p_{v_f,r} = p_{v_f} + \tau p_{\omega_f}$$

$$p_{v_f,l} = p_{v_f} - \tau p_{\omega_f}$$

同様の計算で、右輪目標速度  $p_{v_{\text{ref}},r}$ 、左輪目標速度  $p_{v_{\text{ref}},l}$  は、3 章にて述べた提案手法から導出される目標並進速度  $p_{v_{\text{ref}}}$ 、目標回転角速度  $p_{\omega_{\text{ref}}}$  から算出できる。

$$p_{v_{\text{ref}},r} = p_{v_{\text{ref}}} + \tau p_{\omega_{\text{ref}}}$$

$$p_{v_{\text{ref}},l} = p_{v_{\text{ref}}} - \tau p_{\omega_{\text{ref}}}$$

これより、評価関数を、

$$J_{LQ} = \sum_{h=0}^{H_p} \|\hat{\mathbf{x}}_{\text{ref}}[k+h] - \hat{\mathbf{x}}_l[k+h]\|_{Q_y}^2 + \sum_{h=0}^{H_p} \left\| \begin{bmatrix} \hat{u}_r[k+h-1] \\ \hat{u}_l[k+h-1] \end{bmatrix} \right\|_{R_u}^2$$

とすると、 $J_{LQ}$  は  $\tilde{\boldsymbol{\theta}} = (K_{\alpha r}, K_{\beta r}, K_{\gamma r}, K_{\alpha l}, K_{\beta l}, K_{\gamma l})$  の関数である。これより、時刻  $k$  における最適化問題は、



$$\min_{\tilde{\boldsymbol{\theta}}[k]} J_{LQ}(\tilde{\boldsymbol{\theta}}[k])$$

s.t.

$$\begin{aligned} \hat{u}_i[k+h] &= K_{Pi}\hat{e}_i[k+h] + K_{Ii} \sum_{j=0}^{k+h} \hat{e}_i[j] + K_{Di}(\hat{e}_i[k+h] - \hat{e}_i[k+h-1]) \quad i = r, l \\ \hat{e}[k+h] &= \hat{x}_{\text{ref}}[k+h] - \hat{x}[k+h] \\ \hat{\mathbf{x}}[k+h] &= A\hat{\mathbf{x}}[k+h-1] + B\hat{\mathbf{u}}[k+h-1] \\ \mathbf{x}_{\min}[k] &\leq \hat{\mathbf{x}}[k+h] \leq \mathbf{x}_{\max}[k] \\ h &= 0, 1, \dots, Hp \end{aligned}$$

ただし,

$$\begin{aligned} \hat{\mathbf{x}}[k] = \mathbf{x}[k] &= \begin{bmatrix} p_{v_f, r}[k] \\ p_{v_f, l}[k] \end{bmatrix}, \quad \mathbf{x}_{\text{ref}}[k] = \begin{bmatrix} p_{v_{\text{ref}}, r}[k] \\ p_{v_{\text{ref}}, l}[k] \end{bmatrix}, \\ \mathbf{x}_{\min}[k] &= \begin{bmatrix} p_{v_{f\min}, r}[k] \\ p_{v_{f\min}, l}[k] \end{bmatrix}, \quad \mathbf{x}_{\max}[k] = \begin{bmatrix} p_{v_{f\max}, r}[k] \\ p_{v_{f\max}, l}[k] \end{bmatrix} \end{aligned}$$

と表すことができる.  $\mathbf{x}_{\max}$ ,  $\mathbf{x}_{\min}$  については, やはり 3 章の制約変換手法による並進速度, 回転角速度の最大値及び最小値をもちいて

$$\begin{aligned} p_{v_{\max}, r} &= p_{v_{\max}} + \tau p_{\omega_{\max}} \\ p_{v_{\max}, l} &= p_{v_{\max}} - \tau p_{\omega_{\max}} \\ p_{v_{\min}, r} &= p_{v_{\min}} + \tau p_{\omega_{\min}} \\ p_{v_{\min}, l} &= p_{v_{\min}} - \tau p_{\omega_{\min}} \end{aligned}$$

と算出できる.

図 5.9 に数値例による結果を示す. Leader の走行路は直線走行路, 相対距離制約は 0.45[m] 以上 0.55[m] 以下とした. MP-PID 制御では, 左折している付近で若干 3 章における手法とは軌道が異なる程度であった. この結果より, 本制御手法が熱伝導システムだけでなく, 移動ロボットの制御にも有効であり, 制御対象に対する汎用性を持っていることが確認できた.

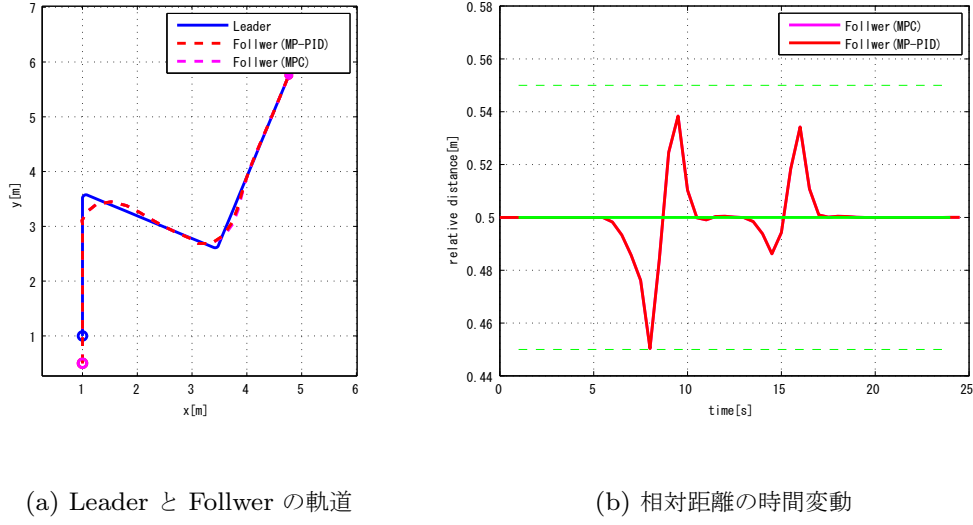


図 5.9 直線走行路におけるモデル予測追従制御と MP-PID 制御の比較

## 5.6 モデル予測型 2 自由度 PID(MP-2DOF PID) 制御への拡張

本節では、前節のモデル予測型 PID 制御の 2 自由度系への拡張を考える．入力  $u$  は、2 自由度 PID 制御の入力へ拡張すると、PID ゲイン  $K_P, K_I, K_D$  に加えて 2 自由度パラメータ  $K_{rp}, K_{rd}$  を用いて (5.26) 式で表すことができる．

$$\begin{aligned}
 u[k] = & K_P \{ (1 - K_{rp}) T_r[k] - y[k] \} + \\
 & K_I \sum_{i=0}^k \{ T_r[i] - y[i] \} + \\
 & K_D \{ (1 - K_{rd}) T_r[k] - y[k] - (1 - K_{rd}) T_r[k-1] + y[k-1] \}
 \end{aligned} \tag{5.26}$$

従って、MP-2DOF PID 制御は 5 パラメータ  $\theta_{2dof} = (K_P, K_I, K_D, K_{rp}, K_{rd})$  に関する最適化問題に定式化できる．すなわち、 $u[k]$  が  $\theta_{2dof}$  の関数であることから、5.4 小節にて示した 3 パラメータからなる MP-PID 制御の定式化と同様に  $e[k+j+1]$ ,  $y_M[k+j+1]$ ,  $u[k+j]$ , ( $j = 0, \dots, H_p - 1$ ) および (5.16) 式の  $J_{LQ}$  は全て  $\theta_{2dof}$  の関

数となる．以上より，時刻  $k$  で解くべき最適化問題は，

$$\begin{aligned}
 & \min_{\theta_{2dof}} J_{LQ}(\theta_{2dof}) \\
 & s.t. \quad \text{Eqs. (5.12) - (5.17)} \\
 & \quad y_M[k+j+1] \leq T_M \\
 & \quad 0 \leq u[k+j] \leq 100 \\
 & \quad j = 0, \dots, H_p - 1
 \end{aligned} \tag{5.27}$$

となる．ここでも， $J_{LQ}$  は  $\theta_{2dof}$  の要素に関して非線形関数になるので，非線形計画法や，あるいは計算機的能力に応じた刻み幅で離散化した  $\theta_{2dof}$  を全探索することで最適化問題 (5.27) は解くことができる．

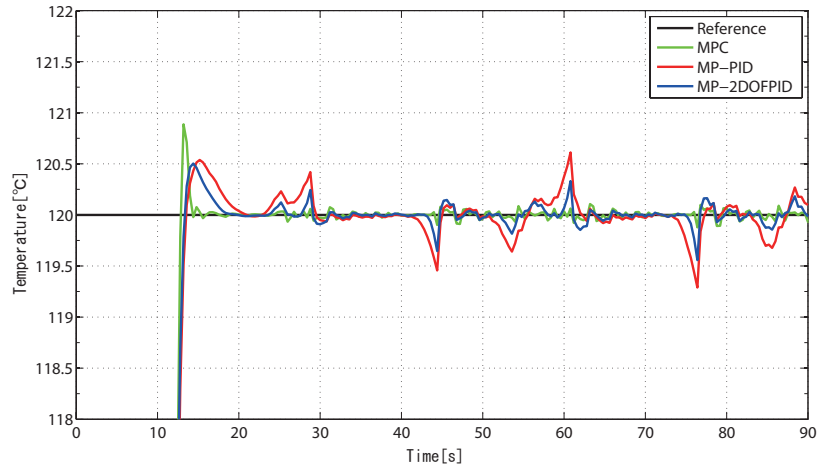


図 5.10 制御性能の比較

## 数値例

前節とは異なる熱伝導プラントに対し，MPC，MP-PID 制御，MP-2DOF PID 制御を適用しその温度制御性能を比較したのが図 5.10 で，縦軸は温度 [°C]，横軸は時間 [s] である．また，表 5.4 に制御ごとの評価指標値 (各制御がおよそ目標温度付近に達する 12[s] 以降の応答から算出) を示す．この数値例においては前節の Case1 と同様に最高温度制約は与えていない．MP-PID，MP-2DOFPID の最適化問題は MATLAB で非線形制約問題をとくための関数 `fmincon` を用い，最適化問題の初期解は全て 0 とし局所的最適解を PID ゲイン，2 自由度パラメータとして用いた．ここで，制御周期

$t_k = 400[\text{ms}]$  であり, 全ての制御において予測区間  $H_p = 10$ , 重みは制御性能のみを考慮して  $w_q = 1, w_r = 0$  ( $j = 0, \dots, 9$ ) とした.

表 5.4 MP-2DOF PID 制御の制御性能評価

	MP-PID	MP-2DOF PID	MPC
Average of absolute error [°C]	0.043	0.038	0.036
Standard deviation [°C]	0.547	0.495	0.473

表 5.4 より, MP-2DOF PID 制御は MP-PID 制御に比べて性能指標値を低減できている. この結果より, MP-2DOFPID は MP-PID より性能が向上していることがわかる. なお, ここでのシミュレーションに用いたプラントの差分式モデルを以下に示す. ここで,  $y = x_1$ ,  $T_{\text{mat}}$  は被加熱部材のシステム投入前温度,  $T_{\text{amb}}$  は外気温で,  $20[^\circ\text{C}]$  とした.  $(A^{(i)}, \mathbf{b}^{(i)}, D^{(i)})$  の値は以下の通りとした. また, 差分方程式の離散化ステップ幅  $t_l$  は  $50[\text{ms}]$  とした. 本数値例のプラントインデックスの時間変化は図 5.11 の通りである.

$$\begin{aligned}
 & \begin{bmatrix} x_1(l+16) & x_2(l+26) & x_3(l+21) & x_4(l+57) & x_5(l+6) \\ x_6(l+63) & x_7(l+47) & x_8(l+1) & x_9(l+1) & x_{10}(l+6) \end{bmatrix}^T \\
 & = A(l)\mathbf{x}(l) + \mathbf{b}(l)u(l) + D(l) \begin{bmatrix} T_{\text{mat}} \\ T_{\text{amb}} \end{bmatrix} \\
 & (A(l), \mathbf{b}(l), D(l)) \in \left\{ (A^{(i)}, \mathbf{b}^{(i)}, D^{(i)}) \mid i = 1, 2, 3 \right\}
 \end{aligned} \tag{5.28}$$

$$y(l) = \mathbf{c}\mathbf{x}(l) = x_3(l) \tag{5.29}$$

$$y_M(l) = \mathbf{c}_M\mathbf{x}(l) = x_9(l) \tag{5.30}$$

$$A^{(1)} = \begin{bmatrix} 0 & 0.673327 & 0 & 0 & 0 & 0 & 0 & 0 \\ 104 & 0 & 0 & 0 & 0 & 0.821 & 0 & 750 \\ 0 & 0.950 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.789191 & 0 & 0 & 0 & 0 \\ 0 & 0.276724 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 14.986 & 0 & 0 & 0 \\ 44 & 0 & 0 & 0 & 0 & 0.921 & 0 & 250 \\ 26 & 0 & 0 & 0 & 0 & 0 & 17.920 & 370 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10.9900 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times 10^{-3}$$

$$A^{(2)} = \begin{bmatrix} 0 & 0.732222 & 0 & 0 & 0 & 0 & 0 & 0 \\ 104 & 0 & 0 & 0 & 0 & 0.821 & 0 & 750 \\ 0 & 0.950 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.789191 & 0 & 0 & 0 & 0 \\ 0 & 0.157815 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 14.986 & 0 & 0 & 0 \\ 44 & 0 & 0 & 0 & 0 & 0.921 & 0 & 250 \\ 26 & 0 & 0 & 0 & 0 & 0 & 17.920 & 370 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10.9900 \\ 0 & 0.405373 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times 10^{-3}$$

$$A^{(3)} = \begin{bmatrix} 0 & 0.76181 & 0 & 0 & 0 & 0 & 0 & 0 \\ 104 & 0 & 0 & 0 & 0 & 0.821 & 0 & 750 \\ 0 & 0.950 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.789191 & 0 & 0 & 0 & 0 \\ 0 & 0.191808 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 14.986 & 0 & 0 & 0 \\ 44 & 0 & 0 & 0 & 0 & 0.921 & 0 & 250 \\ 26 & 0 & 0 & 0 & 0 & 0 & 17.920 & 370 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10.9900 \\ 0 & 0.368375 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times 10^{-3}$$

$$\mathbf{b}^{(1)} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 94 \ 0 \ 0]^T \times 10^{-3}$$

$$\mathbf{b}^{(2)} = \mathbf{b}^{(1)}$$

$$\mathbf{b}^{(3)} = \mathbf{b}^{(1)}$$

$$D^{(1)} = \begin{bmatrix} 46 & 0 & 0 & 0 & 28 & 0 & 0 & 0 & 0 & 222 \\ 0 & 0 & 50 & 20 & 0 & 0 & 10 & 0 & 0 & 0 \end{bmatrix}^T \times 10^{-3}$$

$$D^{(2)} = \begin{bmatrix} 46 & 0 & 0 & 0 & 28 & 0 & 0 & 0 & 0 & 222 \\ 0 & 0 & 50 & 20 & 0 & 0 & 10 & 0 & 0 & 0 \end{bmatrix}^T \times 10^{-3}$$

$$D^{(3)} = \begin{bmatrix} 158 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 257 \\ 0 & 0 & 50 & 20 & 0.76 & -10 & 0 & 0 & 0 & 0 \end{bmatrix}^T \times 10^{-3}$$

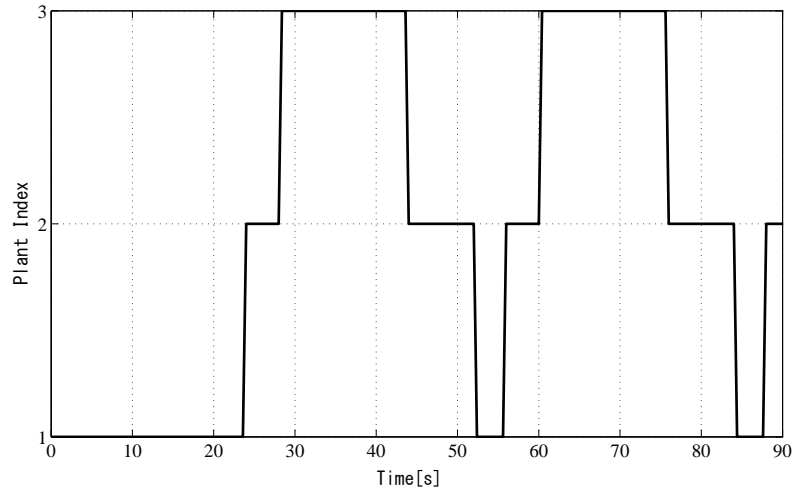


図 5.11 プラントインデックスの時間変化

## 5.7 結言

本章では，モデル予測制御のアルゴリズムに基づき，最適化問題を解くことで PID 制御のゲインを制御周期ごとに最適化するモデル予測型 PID 制御手法を提案した．複数部材から構成される熱伝導システムに対する数値例により，入力制約および最高温度制約の存在下や，初期温度変動に対しても従来の PID 制御より性能が改善されており，提案手法の有効性が確認できた．更に，移動ロボットの協調運搬作業問題にも有効であった．これにより，実製品においても従来の PID 制御よりも製品の品質向上が期待できるだけでなく，制御器には従来用いられてきた PID 制御器の構造をそのまま用いることができるため導入へのハードルも低いといえる．更なる性能改善のために，導入コストが低い，あるいは既に設置済みであることが期待できる FF 制御器を含めた 2 自由度 PID 制御系に対するモデル予測型 PID 制御系の設計への拡張を検討し，同じく数値例によりその有効性を確認した．更なる検討課題として，最適化問題の解法の検討が挙げられる．特に移動ロボットのような高速なダイナミクスを持つ対象に対しては制御周期が長すぎると制御性能が劣化してしまう可能性がある．その場合には，定式化した最適化問題を高速に解くことが求められる．また，ここでは評価関数として一般的である偏差と入力の二乗誤差和を用いて性能改善が見られたが，評価関数の与え方には工夫の余地がある．例えば，熱伝導システムの制御においては今回考慮していないオーバーシュートの低減を加えるなど，複

数の目標に対する多目的最適化が課題となる.

## 第6章

# 結論

本研究では，モデル予測制御の実用性を高めるため，同制御が未だ適用されていない，あるいは更なる活用の余地があるような各種の実システムに対する応用手法について述べた．

2章にてモデル予測制御について概説し，その歴史や発展，また一般的な線形状態空間モデルに対する問題の定式化について示したのち，3章では，移動ロボットの制御問題として未検討であった，近づきすぎても離れすぎても行けないという相対距離制約が課される協調運搬作業問題を取り上げた．これに対し，幾何学的な距離制約を等価な速度制約に変換することによるモデル予測制御を応用した制御手法について示した．数値例によって，相対距離制約を考慮した協調運搬作業を満足する走行制御ができていることを確認した．

4章では，PID制御が主流の熱伝導システムに対し，モデル予測制御を適用するための制御系設計モデル導出手法について示した．提案手法について，モデル予測制御を適用した数値例によりその有効性を確認した．近年，複数の異なるむだ時間をもち複雑な差分式モデルで表現される熱伝導プラントに対しては温度制御の高性能化が求められている．提案手法によりモデル予測制御の容易な適用することが可能となれば，そのような要求に答え，更なる制御性能向上の一助になることが期待できる．

5章では，PID制御の更なる性能改善をはかるため，モデル予測型PID制御を提案した．時々刻々と最適な値を計算すること，制約を考慮した値を計算すること，この2つのモデル予測制御の特長をPID制御や2自由度PID制御という，既に一般に広く用いられている制御器に加えることができた．これにより，制御器の入れ替えや検証にかかるコストを低減し，PID制御器の信頼性を保ちつつ性能を向上させることができる．本論文で扱った移動ロボットの強調運搬作業問題，熱伝導システムの温度制御問題という異なる対



象に有効であることを数値例により示し，その汎用性を確認できた．今後の課題として，最適化問題の解法の検討と，特にモデル予測型 PID 制御においては最適化問題の評価指標の取り方が挙げられる．

以上，本研究は，実システムにおいてこれまでモデル予測制御が十全に活用されてこなかった領域に対する新たな実用的モデル予測制御を示し，モデル予測制御の実用性を高め，実社会におけるモデル予測制御の更なる応用促進に貢献するものである．

# 謝辞

本研究を行うにあたり、筑波大学システム情報系 河辺徹教授には、素晴らしい研究環境を調えて頂き、数多くの御指導、御助言を頂きました。また、研究活動のみならず様々な御配慮も頂きました。心より御礼を申し上げます。幾度と無く御叱咤を頂きながら、それでも自分の至らなさゆえに繰り返し御迷惑をおかけしてしまいましたが、最後まで面倒を見て頂き誠にありがとうございました。

筑波大学システム情報系 久野誉人 教授、徳永隆治 教授、大矢晃久 教授、亀山啓輔 教授には、大変お忙しい中本論文の審査をお引き受け頂きました。心より感謝致します。

長きにわたる在室期間中には、多くの卒業生、修了生とともに学び、知見を得ることができました。また、現在の研究室在室メンバーの皆様にも様々な助言を頂きました。特に李少博君に多く相談させてもらったことが自分にとっては大きな助けとなりました。更には、本研究を通じて学内のみならず学外の方々とも交流を持つ機会を多く得る事ができました。ここでも御迷惑をおかけすることが少なくなかったかと思いますが、自分にとっては社会に出て課題に取り組むことの何たるかを実践的に学ぶ良い機会でありました。この研究室を通じて出会い、自身を成長させて下さった皆様に心より感謝致します。

最後に、10年を超える非常に長い大学生活を温かく見守り、常に支えて下さった両親に心より感謝致します。本当にありがとうございました。

## 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, 東京電機大学出版局, 2005.
- [2] 山本重彦ら, PID 制御の基礎と応用, 朝倉書店, 2005.
- [3] 大嶋正裕ら, モデル予測制御-I-基礎編: 発展の歴史と現状, システム/制御/情報, Vol. 46, No. 5, pp. 286-293, 2002.
- [4] 加納学ら, モデル予測制御-II-線形モデル予測制御, システム/制御/情報, Vol. 46, No. 7, pp. 418-424, 2002.
- [5] 増田士朗ら, モデル予測制御-III-一般化予測制御 (GPC) とその周辺, システム/制御/情報, Vol. 46, No. 9, pp. 578-584, 2002.
- [6] 大嶋正裕ら, モデル予測制御-IV-制約条件付きのモデル予測制御, システム/制御/情報, Vol. 46, No. 11, pp. 702-708, 2002.
- [7] 大嶋正裕ら, モデル予測制御-V-非線形モデル予測制御, システム/制御/情報, Vol. 47, No. 1, pp. 52-57, 2003.
- [8] K.V. Kothare et al., Robust Constrained Model Predictive Control using Linear Matrix Inequalities, *Automatica*, Vol. 32, No. 10, pp. 1361-1379, 1996.
- [9] 藤田政之ら, モデル予測制御-VI-ハイブリッドモデル予測制御, システム/制御/情報, Vol. 47, No. 3, pp. 146-152, 2003.
- [10] 井村順一, モデル予測制御の基礎から最前線まで -ハイブリッドシステムの制御を中心に-, SICE セミナー「ポストロバスト制御」テキスト, 2008.
- [11] K. David Young et al., A Control Engineer's Guide to Sliding Mode Control, *IEEE Transactions on Control Systems Technology*, Vol. 7, No. 3, pp. 328-342, 1999.
- [12] S. Tokat et al., Sliding Mode Control Using a Nonlinear Time-varying Sliding Surface, *Proceedings of the 10th IEEE International Mediterranean Conference*

- on Control and Automation*, Lisbon, Portugal, 2002.
- [13] J. Jacques et al., *Applied Nonlinear Control*, Prentice Hall, New Jersey, 1999.
  - [14] L. C. McNinch et al., Predictive and Sliding Mode Cascade Control for Unmanned Surface Vessels, *American Control Conference*, San Francisco, USA, pp. 184-189, 2011.
  - [15] S. Li and T. Kawabe, A Sliding Mode Control Based on MPC Algorithm for Slip Ratio of Electric Vehicle, *Proceedings of the SICE Annual Conference 2014*, Sapporo, Japan, 2014.
  - [16] J. Zhou et al., A New Nonlinear Model Predictive Control Scheme for Discrete-Time System Based on Sliding Mode Control, *Proceedings of the American Control Conference*, Arlington, USA, pp. 3079-3084, 2001.
  - [17] J. P. Desai et al., Controlling formations of multiple mobile robots, *Proceeding of IEEE International Conference on Robotics and Automation*, pp. 2864-2869, 1998.
  - [18] A. K. Das et al., A Vision-Based Formation Control Framework, *IEEE Transaction on Robotics and Automation*, Vol. 8, No. 5, pp.813-825, 2002.
  - [19] R. M. Murray, Recent Research in Cooperative Control of Multivehicle Systems, *Journal of Dynamic Systems, Measurement, and Control*, Vol. 129, pp.571-583, 2007.
  - [20] Rescue Robot Contest : <http://www.rescue-robot-contest.org> (2014 年 9 月 2 日 閲覧).
  - [21] W.B. Dunbar et al., Distributed receding horizon control for multi-vehicle formation stabilization, *Automatica*, Vol.42, pp. 549-558, 2006.
  - [22] 根和幸ら, 衝突回避を考慮した複数移動体のモデル予測編隊制御, 計測自動制御学会論文集, Vol.428, No. 8, pp. 877-883, 2006.
  - [23] 美多勉, 非線形制御入門-劣駆動ロボットの技能制御論-, 昭晃堂, 2000.
  - [24] 三平満司ら, 非ホロミック Driftless システムのフィードバック制御, 1998.
  - [25] 高橋正徳, 二輪移動ロボットのロードマップ作成と経路追従, 北海道大学大学院工学研究科システム情報工学専攻 修士論文, 2002.
  - [26] R. Fierro et al., Control of a Nonholonomic Mobile Robot : Backstepping Kinematics into Dynamics, *Journal of Robotic Systems*, Vol.14, No. 3, pp. 149-163, 1997.
  - [27] 斉藤守ら, 移動ロボット群における衝突回避 -非線形計画法に基づく局所的解法-, シ

- システム制御情報学会論文集, Vol. 3, No. 8, pp. 252-260, 1990.
- [28] 陽軍ら, ファジィ・ガウシアン・ニューラルネットワーク制御器とその移動ロボット車の制御への応用, 日本機械学会論文集 C, Vol. 59, No. 564, pp. 2290-2297, 1993.
- [29] A. Astolfi, Exponential stabilization of a wheeled mobile robot via discontinuous control, *ASME Journal of Dynamic Systems Measurement and Control*, Vol. 121, No. 1, pp. 121-125, 1999.
- [30] 伊藤朋之, 有限差分法による定着プロセスの熱伝導計算 (非定常問題), 第 67 回日本画像学会技術講習会資料, スプレッドシートを用いた電子写真シミュレーション実習, 2009.
- [31] 日本機械学会編, 伝熱工学 (J S M E テキストシリーズ), 丸善, 2005.
- [32] 清水明, 熱力学の基礎, 東京大学出版, 2007.
- [33] 阿部直人ら, むだ時間・分布定数系の制御, コロナ社, 2007.
- [34] 渡部慶二, むだ時間システムの制御, 計測自動制御学会, 1993.
- [35] 増淵正美ら, システムのモデリングと非線形制御 (コンピュータ制御機械システムシリーズ 9), コロナ社, 1996.
- [36] 工藤啓治, プラントモデル統合環境に求められる技術, 第 2 回制御部門 MBD フォーラム資料, SICE 組込み制御システムのモデルベース開発調査研究会, pp. 49-68, 2008.
- [37] 杉浦英樹, OA 機器開発におけるメカ・エレキ・ソフト・情報統合開発, 第 2 回制御部門 MBD フォーラム資料, SICE 組込み制御システムのモデルベース開発調査研究会, pp. 121-128, 2008.
- [38] 山田哲ら, ベルトを用いたフルカラー定着システム, *Proceeding of Japan Hardcopy Fall Meeting*, pp. 65-68, 1997.
- [39] 須田信英, PID 制御, 朝倉書店, 1992.
- [40] T. Sato, Design of a GPC-based PID controller for controlling a weigh feeder, *Control Engineering Practice*, Vol. 18, pp. 105-113, 2010.
- [41] M. Xu, et al., Auto-tuning of PID controller parameters with supervised receding horizon optimization, *ISA Transactions*, Vol. 44, pp. 105-113, 2005.
- [42] 井前ら, ディスクリプタ形式に基づく P I D 制御系設計, 日本機械学会論文集 C 編, Vol. 77, No. 775, pp. 755-763, 2011.
- [43] 河辺徹ら, モデル予測型 PID 制御による電気自動車のトラクションコントロール, 日本機械学会論文集 C 編, vol. 77, No. 781, pp. 3375-3385, 2011.
- [44] 中村亘ら, 熱伝導システムに対する制御系設計モデルの一導出法, 計測自動制御学会

- 論文集, vol. 47, No. 11, pp.527-533, 2011.
- [45] 山本重彦, 加藤尚武. PID 制御の基礎と応用, 朝倉書店, 東京, 2005.
- [46] 前田肇, 杉江俊治, アドバンスト制御のためのシステム制御理論, 朝倉書店, 東京, 1990.
- [47] 荒木光彦, 田口秀文, 2 自由度 PID 制御装置, システム/制御/情報, Vol.42, No.1, pp.18-25, 1998.
- [48] 日本機械学会編: 伝熱工学 (J S M E テキストシリーズ), 丸善, 2005.
- [49] 清水明, 熱力学の基礎, 東京大学出版, 2007
- [50] 足立修一, システム同定の基礎, 東京電機大学出版局, 2009.

# 研究業績一覧

## 主要業績

### 査読付き論文

1. 中村亘, 河辺徹, 熱伝導プラントに対する制御系設計モデルの一導出法, 計測自動制御学会論文集, Vol. 47, No. 11, pp. 527-533, 2011.
2. K. Nakamura and T. Kawabe, Cooperative Carrying Task Control based on Receding Horizon Control for Mobile Robots, *International Journal of Circuits, Systems and Signal Processing*, Issue 3, Vol. 4, pp. 129-136, 2010.

### 査読付き国際会議

1. K. Nakamura, Y. Hagiwara, S. Li and T. Kawabe, Model Predictive 2DOF PID Control for Heat Conduction System, *Proceedings of SICE Annual Conference 2013*, Nagoya, September, 2013.
2. K. Nakamura, R. Yamashina, S. Li and T. Kawabe, A Configuration of Model Predictive PID Control for Heat Conduction System, *Proceedings of SICE Annual Conference 2012*, pp. 187-191, Akita, August, 2012.
3. K. Nakamura and T. Kawabe, Model Predictive Control Method for Cooperative Carrying Task of Mobile Robots, *Proceedings of the IFToMM 1. International Symposium on Robotics and Mechatronics (ISRM 2009)*, pp. 186-193, Hanoi, September, 2009.

## 査読無し国内発表

1. 中村亘, 河辺徹, 熱伝導プラントに対する制御系設計モデルの一導出法, 計測自動制御学会 システム・情報部門学術講演会 2010, 京都, 2010.

## その他の業績

### 査読付き国際会議

1. S. Li, K. Nakamura and T. Kawabe, A Sliding Mode Control Based on MPC Algorithm for Slip Ratio of Electric Vehicle, *Proceedings of SICE Annual Conference 2014*, Sapporo, September, 2014.
2. S. Li, K. Nakamura, T. Kawabe and K. Morikawa, A Sliding Mode Control for Slip Ratio of Electric Vehicle, *Proceedings of SICE Annual Conference 2012*, pp. 1974-1979, Akita, August, 2012.
3. T. Kawabe and K. Nakamura, Cooperative Carrying Task Control based on RHC for Mobile Robots, *Proceedings of the International Conference on CSS2010 (Circuits, Systems, Signals 2010)*, pp. 164-169, Malta, September, 2010.