

実時間最適制御による移動物体の運動制御  
に関する研究

防衛大学校理工学研究科後期課程

装備・基盤工学系専攻 装備システム工学教育研究分野

渡辺 宗一郎

平成29年12月

# 目次

<b>第1章 序論</b>	<b>4</b>
1.1 本研究が望まれる状況と背景	4
1.2 移動物体の実時間最適制御	8
1.3 本研究の目的と意義	11
1.4 本論文で使用する主な記号	15
<b>第2章 Sample-and-Hold 型と Clock-Based 型の実時間最適制御の制御理論</b>	<b>18</b>
2.1 概要	18
2.2 モデル予測制御 (Model Predictive Control: MPC)	20
2.3 Carathéodory- $\pi$ 手法 (C- $\pi$ 手法)	25
2.3.1 概要	25
2.3.2 サンプリングタイムとダイナミクスとの関係	26
2.3.3 制御入力の生成方法	29
2.4 予測型 Carathéodory- $\pi$ 手法 (PC- $\pi$ 手法)	33
2.4.1 概要	33
2.4.2 制御入力の生成方法	33
2.4.3 C- $\pi$ 手法との比較	36
2.5 Gain 型 M・C- $\pi$ 手法と Gain 型 M・PC- $\pi$ 手法	38
2.5.1 概要	38
2.5.2 制御入力の生成方法	38
2.5.3 C- $\pi$ 手法との比較	39
2.5.4 安定性領域の解析	41
2.6 MPC 型 M・C- $\pi$ 手法と MPC 型 M・PC- $\pi$ 手法	42
2.6.1 概要	42
2.6.2 制御入力の生成方法	42
<b>第3章 実時間最適制御による運動制御の検証システム</b>	<b>44</b>
3.1 概要	44
3.2 数値シミュレーションを用いた検証方法	45
3.2.1 システムの構成	45
3.2.2 フルビークルモデルソフトウェア	47
3.3 実車両を用いた検証方法	48
3.3.1 システムの構成	48
3.3.2 実験車両	50
3.3.3 計測システム	51

<b>第 4 章</b>	<b>Clock-Based 実時間最適制御による MGV の運動制御</b>	<b>52</b>
4.1	概要	52
4.2	運動モデルの構築とパラメータ同定	53
4.3	制御系の設計	55
4.3.1	C- $\pi$ 手法における最適制御問題の定式化	55
4.3.2	Gain 型 M・C- $\pi$ 手法の制御系設計	58
4.3.3	MPC 型 M・C- $\pi$ 手法の制御系設計	60
4.3.4	MGV の運動制御で使用するパラメータの値 (共通)	62
4.4	数値シミュレーション	64
4.4.1	初期状態誤差に対する影響	64
4.4.2	路面外乱に対する影響	74
4.5	実機を用いた検証	82
4.5.1	直線走行	83
4.5.2	S 字走行	99
4.5.3	切り返し走行	115
4.5.4	障害物回避走行	130
4.6	まとめ	145
4.6.1	Clock-Based 実時間最適制御の適用可能性	145
4.6.2	予測情報を用いる PC- $\pi$ 手法の有用性	145
4.6.3	追従制御器を付加した提案手法の有用性	146
<b>第 5 章</b>	<b>Clock-Based 実時間最適制御による一般車両の運動制御</b>	<b>147</b>
5.1	概要	147
5.2	運動モデルの構築とパラメータ同定	148
5.2.1	運動モデルの構築	148
5.2.2	一般車両のパラメータ同定	151
5.3	制御系の設計	152
5.3.1	C- $\pi$ 手法における最適制御問題の定式化	152
5.3.2	Gain 型 M・C- $\pi$ 手法の制御系設計	154
5.3.3	MPC 型 M・C- $\pi$ 手法の制御系設計	156
5.3.4	一般車両の運動制御で使用するパラメータの値 (共通)	158
5.4	数値シミュレーション	160
5.4.1	レーンチェンジ走行	160
5.4.2	停止車両回避走行	170
5.5	まとめ	183
5.5.1	Clock-Based 実時間最適制御の適用可能性	183
5.5.2	予測情報を用いる PC- $\pi$ 手法の有用性	183
5.5.3	追従制御器を付加した提案手法の有用性	184
<b>第 6 章</b>	<b>結論</b>	<b>185</b>
6.1	本研究の総括	185
6.2	本研究で得られた今後の展望	186

謝 辞	188
参考文献	189
著者学会等発表実績	199

# 第1章 序論

## 1.1 本研究が望まれる状況と背景

本研究は、実時間最適制御を移動物体の運動制御に適用し、その有効性の検証を行う。はじめに無人システム・有人システムを含んだ移動物体の社会的な活用状況について概観する。次に、最適制御について、実例を用いてその有用性について示し、また問題点について指摘する。最後に、本研究において移動物体の運動制御に対して実時間最適制御を適用することに着目した背景について述べる。

様々な分野において、無人移動物体を活用した取り組みが行われている。無人移動物体はその活動領域により、陸上（地上）で活動する無人陸上車両 (Unmanned Ground Vehicle: UGV)、海中で活動する無人潜航艇 (Unmanned Underwater Vehicle: UUV)、空中で活動する無人航空機 (Unmanned Aerial Vehicle: UAV) などに大別することができる。また機体の大きさにより分類する場合もあり、小型の UGV であれば MGV (Micro Ground Vehicle)、小型の UAV であれば MAV (Micro Air Vehicle) などと呼ばれている。



図 1.1 510 PackBot\*<sup>1</sup>



図 1.2 RQ-16 T-Hawk\*<sup>2</sup>

\*<sup>1</sup> 画像引用 : iRobot HP (<http://www.irobot.com>)

\*<sup>2</sup> 画像引用 : Wikipedia (<https://ja.wikipedia.org/wiki>)

実用化された無人移動物体が社会的に活用された例としては、2011年3月11日に発生した東日本大震災による福島第一原子力発電所事故の調査において、図1.1に示すアメリカのiRobot社製軍用ロボット「510 PackBot」が投入されている。放射能の影響により人間が立ち入ることが困難な空間においても遠隔操作により活動することが可能であり、建屋内の放射線量、雰囲気温度、雰囲気湿度、酸素濃度の測定を行ったことが報告されている [1]。同じく、図1.2に示すHoneywell社製の「RQ-16 T-Hawk」についても調査に投入され、原子力発電所の空撮に従事している [2]。その他にも、農薬散布を行う産業用ヘリコプタや、軍事利用としての米軍の「RQ-1 Predator」、自衛隊の遠隔操縦観測システム「FFOS (Flying Forward Observation System)」などが挙げられる。無人移動物体の遠隔操縦は、カメラ等を通じて人間が操縦を行うため、有線・無線を問わずに通信が可能な距離での運用に制限されてしまうが、近年では自動化・自律化により運用の幅を広くすることが期待されている。

一方、一般的な乗用車などの有人の移動物体に対する自動化技術も発展している。一般車両は基本的には運転手がハンドルやペダルを操作することで移動するが、障害物を感知して衝突に備える衝突被害軽減ブレーキや、車線を逸脱しそうになると警報音にて運転手に知らせる車線逸脱警報、車速の制御を行うクルーズコントロールなどの運転手の操縦を補助する様々なシステムが取り入れられている。このようなシステムは安全運転支援システムに位置付けられるが、より高度な自動化技術としては自動運転システムがある。内閣官房IT総合戦略室では自動化のレベルを表1.1のように定めている [3]。

自動運転車両の実用化には様々な効果が期待されている。内閣府では「2020年までに交通事故死者数を2,500人以下とし、世界一安全な道路交通を実現する」ことを国家目標として掲げている [4]。交通事故死者数の低減や交通渋滞の緩和による環境負荷低減などの社会的意義や、自動車産業の競争力向上などの産業的意義がある。自動運転車両の実用化については、内閣府や国土交通省などの国主導のプロジェクトを中心として、各自動車メーカーやIT企業、大学、地方自治体などが取り組んでおり、2017年現在ではレベル3のロボットタクシー車両やレベル4の無人バス走行の実証実験を行った例も報告されている [5]。しかし、完全自動運転実現に対しては発展段階であり、制度面や技術面での多くの課題が示されている [6]。

このようなMGV (UGV) やMAV (UAV) などの無人移動物体と、自動運転車両などの有人移動物体の自動化を行う上では、移動する軌道を計画して運動を制御することは重要な機能の一つである。移動空間にポテンシャル場を設定して経路を生成するポテンシャルフィールド法や運転手の危険感覚を基にした危険感ポテンシャルドライバモデル、円弧の組み合わせにより経路を生成する手法など様々な軌道計画手法が

表 1.1 自動運転レベルの定義の概要 [3]

レベル	概要	安全運転に係る 監視, 対応主体
【 運転者が全てあるいは一部の運転タスクを実施 】		
SAE レベル 0 運転自動化なし	<ul style="list-style-type: none"> <li>・ 運転者が全ての運転タスクを実施</li> </ul>	運転者
SAE レベル 1 運転支援	<ul style="list-style-type: none"> <li>・ システムが前後, 左右のいずれかの車両制御に係る運転タスクのサブタスクを実施</li> </ul>	運転者
SAE レベル 2 部分運転自動化	<ul style="list-style-type: none"> <li>・ システムが前後, 左右の両方の車両制御に係る運転タスクのサブタスクを実施</li> </ul>	運転者
【 自動運転システムが全ての運転タスクを実施 】		
SAE レベル 3 条件付運転自動化	<ul style="list-style-type: none"> <li>・ システムが全ての運転タスクを実施 (領域※限定的)</li> <li>・ システムの介入要求等に対して, 予備対応時利用者は, 適切に応答することを期待</li> </ul>	システム (フォールバック中は運転者)
SAE レベル 4 高度運転自動化	<ul style="list-style-type: none"> <li>・ システムが全ての運転タスクを実施 (領域※限定的)</li> <li>・ 予備対応時において, 利用者が応答することは期待されない</li> </ul>	システム
SAE レベル 5 完全運転自動化	<ul style="list-style-type: none"> <li>・ システムが全ての運転タスクを実施 (領域※限定的ではない)</li> <li>・ 予備対応時において, 利用者が応答することは期待されない</li> </ul>	システム

※ここでの「領域」は, 必ずしも地理的な領域に限らず, 環境, 交通状況, 速度, 時間的な条件などを含む。

検討されており [7], 移動物体に適用する場合にはそれぞれの特徴に応じて適切に選定する必要がある。

移動物体を運用する際には, 運用場面に応じて効率的な移動が求められる場合が多い。例えば, 偵察活動や災害活動などの緊急性の高い任務に無人移動物体を投入する場合や, バッテリー等の積載量に制約を受けるため移動可能時間が短くなる MGV や MAV を使用する場合には, より短い時間での移動やエネルギー消費が最小となるような移動が求められる。また自動運転車両の場合においても同様な効率的な移動が求められるが, 乗車人員の負担も考慮した移動を行わなければならない。このような移動物体の運動の効率化を達成する手段として最適制御理論の活用がある。その場合には運用目的を最適制御問題の評価関数として設定し, 移動物体の運動特性や様々な制約等を考慮して評価関数が最小となる最適な制御入力を決定することになる。最適制御を移動物体の運動制御に応用した研究は数多く報告されている [8-43]。古くには, 超音速機 F4 ファントムの最短時間上昇問題として勾配法により最適解を導出し [44], 1962 年 1 月に Patuxent River Air Station にて得られた解を利用した試験飛行が行われている [26]。この飛行では, それ以前よりも飛行時間が大幅に短縮され, 移動物体への最適制御理論の適用が実システムの運用改善に有用であることが示されている。また日本国内においても, 航空自衛隊の航空実験団が超音速機 F-104J, F-4EJ, 国産の T-2 について最短時間上昇に関する理論と実機を対比した研究を行い, 良好な結果が得られていることが報告されている [8]。自動車の運動に適用した例としては, 文献 [19, 22-25, 34, 35, 39] において自動車の最短時間コーナリング問題として数値シミュレーションにより運動解析などを行っている。その他にも, 二輪車の運動に適用した研究 [31, 41] や, 実機の MGV・MAV に適用して検証した研究 [38, 40, 43] などがある。

このように最適制御が様々な移動物体の運動制御に対して有用であることが示されているが, 一方で制御系設計用のモデルと実制御対象とのギャップ, つまりモデル化誤差が存在する場合の安定性に関する問題も指摘されている [45]。また移動開始前には予期できない障害物の発生など, 状況が変化する場合における対処も必要となる。そのような問題に対応しつつ, より効率的な移動を実現する方法として, 移動間にも最適制御問題を繰り返し解き続ける方法が考えられている。これは実時間最適制御と呼ばれ, モデル化誤差などにより予測していた軌道から逸脱した場合でも新たな最適制御解を生成することで, その状態からの新たな最適軌道により移動することが可能となる。移動途中に認知した障害物などに対しても, それを考慮した新たな軌道を生成することで, 障害物を効率的に回避して目標まで移動できる。近年ではコンピュータの演算能力や最適化アルゴリズムの向上に伴い, 複雑な最適制御問題に対する計算時間の短縮化が図られており, 実時間最適制御による効果的な移動物体の運動制御が期待されている。

## 1.2 移動物体の実時間最適制御

従来の最適化計算時間は移動物体の制御時間に対して長く、最適化計算の高速化を図る必要があった。モデル予測制御 (Model Predictive Control: MPC) [46–55] は、サンプリング間の制御入力を一定値とし、また最適制御問題における評価区間を有限として高速な最適化計算を実現している。

MPC を移動物体の運動制御に適用した研究についても多数報告されている [56–80]。文献 [62] では、目標レーンを参照軌道とした自動車のレーンチェンジ走行を想定している。予測ホライズンにおける横方向位置の誤差と使用する制御入力の重み付き総和を評価関数とし、サンプリングタイムごとに制御入力を生成してレーンチェンジを行っている。文献 [70] では、直進する航空機が前方の突風領域を MPC により回避して飛行する状況を想定している。Laser LIDAR より短い距離（ここでは約 2.5 km）を予測ホライズンとして設定し、突風領域が予測ホライズンに入ったならば MPC によりその領域を回避して、その後もとの参照軌道に復帰している。文献 [71] では、前輪操舵式の非ホロノミック MGV について障害物回避と車庫入れ制御を行っている。切り返し操作を含んだ参照軌道上に静的・動的な障害物が発生することを想定し、実機の MGV を用いた実走行実験により検証している。

一方、近年ではコンピュータの演算能力や最適化アルゴリズムの向上に伴い、最適化計算に要する時間は短くなってきている。そのため、終端時間を自由とし、さらにサンプリング間の入力値も時刻参照として考える最適制御問題を移動途中に繰り返し解く実時間最適制御が現実的に可能となりつつある。文献 [81] では、MPC のようにサンプリング間の入力値を時間に対して不変、もしくは一定変化とする実時間最適制御を「Sample-and-Hold 型」、反対にサンプリング間の入力値を時刻参照として適用する実時間最適制御を「Clock-Based 型」と分類している。本論文においても実時間最適制御をそのような分類で取り扱う。

Clock-Based 実時間最適制御を移動物体の運動制御に適用した研究についてもいくつか報告されている [81–87]。文献 [81] では、サンプリングタイムごとにサンプリングした状態から目標状態までの解を生成する Clock-Based 実時間最適制御について考え、サンプリングタイムとダイナミクスを用いた安定性の理論的な解析を行っている。この手法により得られる制御解は Carathéodory- $\pi$  解と呼ばれているため、本論文では本手法を Carathéodory- $\pi$  手法 (C- $\pi$  手法) と呼ぶ。ここでは、C- $\pi$  手法を人工衛星の運動制御に適用し、地上での模型人工衛星による実験的検証も行っている。文献 [86] では、MGV の運動制御に C- $\pi$  手法を適用し、数値シミュレーションによる検証を行っている。また、Clock-Based 実時間最適制御と他の軌道計画手法についての性能を比較し、Clock-Based 実時間最適制御の有用性について表 1.2 のようにまとめている。そ

の他にも，宇宙往還機 [84]，MAV [87] を対象として数値シミュレーションによる検証を行った研究が報告されている．さらにサンプリングした状態と得られている制御解を用いたプロパゲーションにより初期条件を予測する手法も考えられている [82]．本論文では本手法を Propagation Carathéodory- $\pi$  手法 (PC- $\pi$  手法) と呼ぶ．PC- $\pi$  手法については，倒立振子の制御に適用し，数値シミュレーションを行った研究が報告されている [82]．

表 1.2 Hurmi による軌道計画手法の比較 [86]

	Path Planning and Control	Optimality	Obstacle Avoidance	Handling Vehicle Constraints	Global vs Local Info	Computational Complexity	Portability	Completeness	Multiple Vehicles	Error and Uncertainties
Bug Algorithms	C	F	F	C	C	A	C	A	F	C
Potential Fields	C	C	A	C	A	C	C	A	A	A
Roadmaps	C	F	A	A	A	C	C	A	A	A
Cell Decomposition	C	C	C	C	C	F	C	A	C	C
Optimal Control	A	A	A	A	A	C	A	A	A	A

### 1.3 本研究の目的と意義

Clock-Based 実時間最適制御である C- $\pi$  手法もしくは PC- $\pi$  手法は，サンプリングタイムごとに目標までの最適制御解を生成するため，より厳密な最適制御を行う手法であり，移動物体の更なるロバストで効率的な運動制御が期待される．さらに近年では最適化計算の高速化により，実システムの移動物体への適用が現実的に可能となっている．C- $\pi$  手法や PC- $\pi$  手法を移動物体に適用した研究は表 1.3 のようにまとめられる．

表 1.3 C- $\pi$  手法及び PC- $\pi$  手法を移動物体に適用した研究

移動物体の種類	C- $\pi$ 手法		PC- $\pi$ 手法	
	Sim	Exp	Sim	Exp
人工衛星	文献 [81]		なし (倒立振子のみ)	
宇宙往還機	文献 [84]	なし		
MGV	文献 [85, 86]	なし		
MAV	文献 [87-89]	文献 [88]		
一般車両	なし			
一般航空機	文献 [90]	なし	文献 [90]	

Sim : 簡易モデルを用いたシミュレーションによる検証

Exp : 実験による検証

表 1.3 に示すように，前述の MGV や MAV などに対して Clock-Based 実時間最適制御を適用した研究では，C- $\pi$  手法の簡易モデルを用いた数値シミュレーションによる検証のみである．自動運転車両など一般車両の運動制御を対象として検証した例はない．さらに，いずれについてもシステムの特性に依存する安定性が保証された短いサンプリング間隔内に最適化計算が終了する条件を設定している．

しかし実用上では，最適化で使用するモデルと実システムとの差異が存在し，制御対象の安定性に影響を及ぼす．そのため，実機による検証，もしくは厳密なシミュレーション環境による検証を行い，その適用可能性を明らかにする必要がある．また最適化計算時間を考慮してサンプリングタイムが長くなる可能性はあり，その場合においても移動物体をより安定して制御できることは実用化する上で大きな意義をもつ．

そこで本研究では，上記の移動物体の中でも，比較的検証が容易と考えられる地上移動物体の MGV 及び一般車両を対象として以下の課題について取り組む．なお，文献 [88-90] については著者が関連した研究であるが，本論文ではその内容について取り扱っていないため研究例として掲示している．

(1) Clock-Based 実時間最適制御を MGV に適用した研究は、C- $\pi$  手法についての数値シミュレーションによる検証のみである。しかし実機で運用する場合には、最適化計算において使用するモデルの誤差や、また小型であるという点から路面の起伏等の影響を受けやすいため、数値シミュレーションでは模擬できない不確定要素などが存在する。従って、Clock-Based 実時間最適制御の実機の MGV への適用可能性を示す必要がある。そこで本研究では、C- $\pi$  手法及び PC- $\pi$  手法のそれぞれを MGV に適用した制御システムの設計及び実験環境の構築を行い、実機を用いた検証により Clock-Based 実時間最適制御の MGV への適用可能性を示す。

(2) 一般車両は高速で、さらにローリングやピッチングなどの 3 次元的で複雑に運動するが、サンプリングタイムの短縮化のため最適化計算に簡易なモデルを使用することを想定しなければならない。そのため MGV とは異なる運動特性を持つ移動物体として考える必要があるが、一般車両に Clock-Based 実時間最適制御を適用して検証を行った研究は報告されていない。従って、Clock-Based 実時間最適制御の一般車両への適用可能性を示す必要がある。そこで本研究では、C- $\pi$  手法及び PC- $\pi$  手法のそれぞれを一般車両に適用した制御システムを設計し、最適化計算には簡易なモデルを、運動計算にはフルビークルモデルのシミュレーションソフトウェアを用いた数値シミュレーションを行うことで Clock-Based 実時間最適制御の一般車両への適用可能性を示す。

(3) 移動前には認知されていなかった障害物が移動中に発生するなど、状況が変化する場合では最適制御問題が複雑になる可能性がある。そのため最適化計算の負荷が増大する可能性があり、サンプリングタイムが長くなる場合を想定する必要がある。本研究では、C- $\pi$  手法及び PC- $\pi$  手法の制御系において逐次生成される最適状態軌道に対する追従制御器を組み込むことで、サンプリングタイムが長くなる場合においてもより安定性を保持できる制御アルゴリズムを提案し、それぞれの検証環境において有用性を確認する。

Clock-Based 実時間最適制御は、主に対象とする移動物体の動的変化と最適化計算時間の関係に依存する。本研究では移動物体として MGV および一般車両を扱っているが、他の動的システムに対しても容易に応用が可能な技術である。そのため、得られた知見は他のシステムに適用する際の指針ともなり、工学分野における制御技術の発展に寄与すると考える。

図 1.3 に本論文の構成を示す。

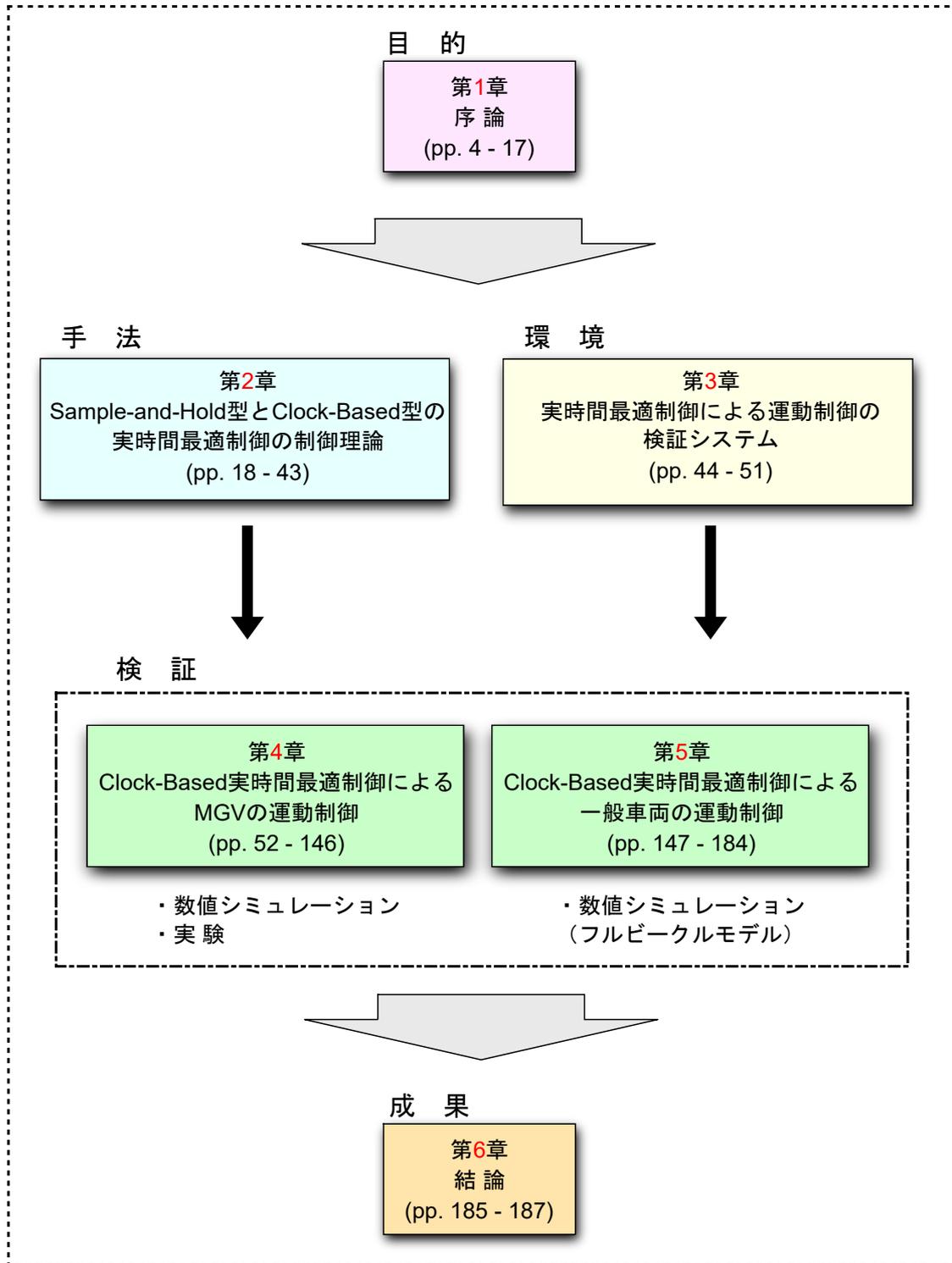


図 1.3 本論文の構成

第1章は本論文の序論であり、実時間最適制御による移動物体の運動制御に関する研究の背景や取り組むべき課題について述べている。さらに本研究の目的及び論文の概要を説明している。

第2章では、実時間最適制御のアルゴリズムについて述べている。最初に従来用いられている手法である、Sample-and-Hold型のMPCと、本研究で用いるClock-Based型のC- $\pi$ 手法及びPC- $\pi$ 手法における制御入力の生成方法について述べている。次にC- $\pi$ 手法及びPC- $\pi$ 手法におけるサンプリングタイムとダイナミクスの関係について述べている。続いて、C- $\pi$ 手法及びPC- $\pi$ 手法の制御系に最適状態軌道に対する追従機構を組み込んだ、Gain型M・C- $\pi$ (M・PC- $\pi$ )手法とMPC型M・C- $\pi$ (M・PC- $\pi$ )手法について提案している。

第3章では、シミュレーション及び実験による検証方法について述べている。シミュレーションによる検証では、サンプリングタイム以内に最適化計算が完了するものと仮定し、最適化計算と制御入力の適用を交互に行っている。実験による検証では、モーション・キャプチャシステムを含んだ屋内実験環境を構築し、最適化計算結果と制御入力の適用を並行して行うシステムを構築している。

第4章では、各制御手法をMGVの運動制御に適用した検証を行っている。ここでは、MGVが与えられた運用タスクを実行する際に高い頻度で用いられる直線走行、S字走行、繰り返し走行、障害物回避走行について想定し、それぞれの想定における各制御手法の性能をシミュレーション及び実験により比較・検証している。実機での走行を実現することでMGVに対する各制御手法の適用可能性を示し、さらにPC- $\pi$ 手法や提案手法の有用性を明らかにしている。

第5章では、各制御手法を一般車両の運動制御に適用した検証を行っている。ここでは、レーンチェンジ走行及び停止車両回避走行について想定し、それぞれの想定における各制御手法の性能をフルビークルモデルシミュレーションにより比較・検証している。実車両による検証が困難なため、厳密なモデルのシミュレーションを行うことにより一般車両に対する各制御手法の適用可能性を示し、さらにPC- $\pi$ 手法や提案手法の有用性を明らかにしている。

第6章は、本論文の結論であり、本研究の成果と今後の展望についてまとめている。

## 1.4 本論文で使用する主な記号

表 1.4~1.8 に本論文で使用する主な記号について示す。

表 1.4 本論文で使用する主な記号（その 1）

記号	意味
$a_x$	前後加速度
$a_y$	横加速度
$C_f$	前輪のコーナリングパワー
$C_r$	後輪のコーナリングパワー
$e_c$	最適制御問題 $\mathbf{P}_c$ で設定する境界条件ベクトル
$e_p$	最適制御問題 $\mathbf{P}_p$ で設定する境界条件ベクトル
$E$	平均化
$E_c$	最適制御問題 $\mathbf{P}_c$ で設定するメイヤーコスト
$E_m$	最適制御問題 $\mathbf{P}_m$ で設定するメイヤーコスト
$E_p$	最適制御問題 $\mathbf{P}_p$ で設定するメイヤーコスト
$E_x$	各サンプリング時刻における $x$ 方向誤差の平均値
$E_y$	各サンプリング時刻における $y$ 方向誤差の平均値
$E_\psi$	各サンプリング時刻における $\psi$ 方向誤差の平均値
$f$	運動モデル
$F_c$	最適制御問題 $\mathbf{P}_c$ で設定するラグランジュコスト
$F_m$	最適制御問題 $\mathbf{P}_m$ で設定するラグランジュコスト
$F_p$	最適制御問題 $\mathbf{P}_p$ で設定するラグランジュコスト
$h_c$	最適制御問題 $\mathbf{P}_c$ で設定する軌道拘束条件
$h_m$	最適制御問題 $\mathbf{P}_m$ で設定する軌道拘束条件
$h_p$	最適制御問題 $\mathbf{P}_p$ で設定する軌道拘束条件
$h_{obs}$	障害物の形状を表す関数
$H_p$	予測ホライズン
$H_u$	制御ホライズン
$I_z$	$z$ 軸まわりの慣性モーメント
$J_c$	最適制御問題 $\mathbf{P}_c$ で設定する評価関数
$J_m$	最適制御問題 $\mathbf{P}_m$ で設定する評価関数
$J_p$	最適制御問題 $\mathbf{P}_p$ で設定する評価関数
$k$	時変フィードバック制御則
$k_T$	追従制御の時変フィードバック制御則
$K$	フィードバックゲイン行列
$K_{PV}$	Gain 型追従制御器の前後方向誤差に対する比例ゲイン
$K_{P\delta}$	Gain 型追従制御器の横方向誤差に対する比例ゲイン
$K_V$	速度指令に対する速度応答のゲイン
$l_f$	重心から前輪までの距離

表 1.5 本論文で使用する主な記号（その2）

記号	意味
$l_r$	重心から後輪までの距離
$L$	MGVの車軸間距離
$L_p$	開始位置から目標位置までの距離
$Lipf$	下付き添え字に関するリップシツツ定数
$m$	車体質量
$N_s$	サンプリングを行った回数
$p$	実システムのパラメータ
$p_0$	モデルのパラメータ
$p_{obs}$	障害物のノルム
$\mathbf{P}_c$	C- $\pi$ 手法で設定する非線形最適制御問題
$\mathbf{P}_m$	MPCで設定する非線形最適制御問題
$\mathbf{P}_p$	PC- $\pi$ 手法で設定する非線形最適制御問題
$r$	ヨー角速度
$\mathbb{R}^N$	$N$ 次元の実数空間
$t$	時間
$T_p$	Gain型追従制御器の予測時間
$\mathbf{U}$	制御変数ベクトル
$\mathbf{U}_T$	フィードバック制御入力ベクトル
$V$	速度
$V_c$	速度指令
$W$	ランベルトの $W$ 関数
$W_{ddf}$	C- $\pi$ (PC- $\pi$ )手法の評価関数における重み
$W_{d\delta}$	MPCの評価関数における重み
$W_{obs}$	C- $\pi$ (PC- $\pi$ )手法の評価関数における重み
$W_M$	フィードフォワード制御入力とフィードバック制御入力の重み
$W_V$	MPCの評価関数における重み
$W_{Vlast}$	MPCの評価関数における重み
$W_x$	MPCの評価関数における重み
$W_y$	MPCの評価関数における重み
$W_{\delta last}$	MPCの評価関数における重み
$W_\psi$	MPCの評価関数における重み
$x$	$x$ 座標
$x_{obs}$	障害物の $x$ 座標
$\mathbf{X}$	状態変数ベクトル
$\mathbf{X}_M$	予測軌道ベクトル
$\mathbf{X}_R$	実軌道ベクトル
$\mathbf{X}_{Ref}$	参照軌道

表 1.6 本論文で使用する主な記号（その3）

記号	意味
$y$	$y$ 座標
$y_{obs}$	障害物の $y$ 座標
$\beta$	横滑り角
$\delta_f$	操舵角
$\delta_{fc}$	操舵角指令
$\xi$	障害物関数
$\eta$	外乱などの不確定要素
$\Lambda$	MGV の運動モデルの補正パラメータ
$\tau_c$	C- $\pi$ (PC- $\pi$ ) 手法のサンプリングタイム（固定値）
$\tau_{ci}$	C- $\pi$ (PC- $\pi$ ) 手法のサンプリングタイム（可変値）
$\tau_d$	操舵角指令に対する操舵角応答の時定数
$\tau_m$	MPC のサンプリングタイム
$\bar{\tau}$	C- $\pi$ 手法で安定性が保証されているサンプリングタイム
$\omega_n$	速度指令に対する速度応答の固有振動数
$\psi$	ヨー角
$\zeta$	速度指令に対する速度応答の減衰係数
$\ \cdot\ $	ノルム

表 1.7 本論文で使用する主な記号（上付き）

記号	意味
$L$	下限値
$U$	上限値
$*$	最適解
$\wedge$	MPC における予測状態，入力

表 1.8 本論文で使用する主な記号（下付き）

記号	意味
$i$	$i$ 番目
$f$	終端時刻，状態
$FB$	フィードバック制御入力
$FF$	フィードフォワード制御入力
$max$	変数の上限値
$min$	変数の下限値

# 第2章 Sample-and-Hold型と Clock-Based型の実時間最適制御の制御理論

## 2.1 概要

図 2.1 に実時間最適制御を移動物体 (System) へ適用する場合の制御システムの構成を示す。制御手順は、サンプリング時刻において Switch が閉じて、(1) 移動物体の状態 ( $\mathbf{X}(t)$ ) を計測し、次のサンプリング時刻までに (2) 最適化 (Optimization)、(3) 最適制御解 ( $\mathbf{X}^*(t), \mathbf{U}^*(t)$ ) の適用、の3つのステップを繰り返す。

ここでは達成すべき目標 (Mission Task) を評価関数 (Cost Function) として与え、移動物体の運動モデル (Model) や拘束条件 (Constraints) を考慮した最適制御問題を定式化して最適化計算を行う。運動モデルが厳密であれば、精度の高い最適制御解を得ることができるが、計算負荷が高くなるためサンプリング間隔も長くなる。そのため、精度と計算負荷のトレードオフを考慮した適切な運動モデルを構築する必要がある。拘束条件により移動物体の持つ性能の範囲内での最適化を行うため、性能を最大限に活

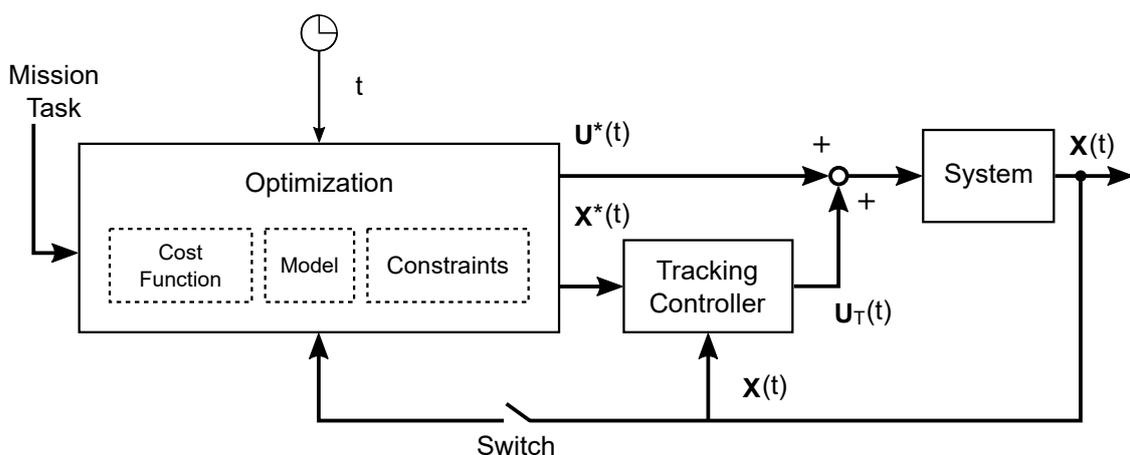


図 2.1 実時間最適制御の制御システムの構成

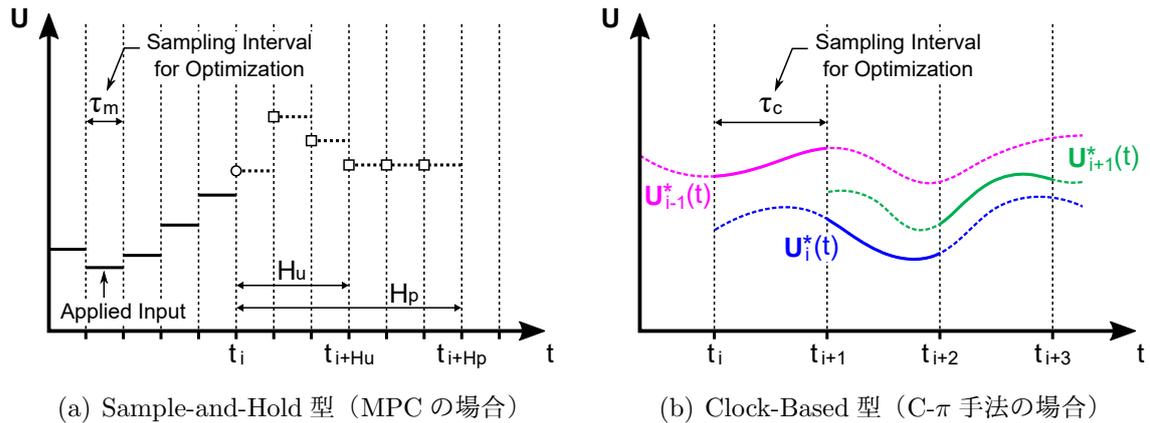


図 2.2 実時間最適制御における制御入力の模式的な時間応答

用した制御解を得ることができる。得られた最適制御解 ( $\mathbf{X}^*(t), \mathbf{U}^*(t)$ ) については、最適制御入力  $\mathbf{U}^*(t)$  のみを用いる場合、最適状態軌道  $\mathbf{X}^*(t)$  に追従させる場合、両方を利用する場合が考えられる。

第 1 章でも述べたが、実時間最適制御は最適化計算で生成する制御入力の形式により、「Sample-and-Hold 型」と「Clock-Based 型」に分類される。図 2.2 にそれぞれの型式における制御入力の模式的な時間応答を示す。

Sample-and-Hold 型はサンプリング間隔内の制御入力は一定である。そのため計算負荷が小さく、高速演算により短いサンプリング間隔での制御を行うという特徴がある。図 2.2(a) において一例として示している MPC では、一定の有限時間未来を評価区間とした最適化計算を行っている。

一方、Clock-Based 型はサンプリング間隔内の制御入力の時間的な変化を考慮している。そのため、より厳密な最適制御問題による制御入力を取得することができるが、一般的に計算負荷が大きくなる。図 2.2(b) において一例として示している C- $\pi$  手法では、サンプリングタイムごとに目標までの最適な参照軌道を再生成している。

本章では実時間最適制御のアルゴリズムとして、はじめに Sample-and-Hold 型の MPC について記述する。次に Clock-Based 型の C- $\pi$  手法と PC- $\pi$  手法、これらに追従制御器を付加した Gain 型 M · C- $\pi$  手法と MPC 型 M · C- $\pi$  手法について述べる。

## 2.2 モデル予測制御 (Model Predictive Control: MPC)

MPCは、評価関数や制御対象のモデル、拘束条件等を考慮した、サンプリングタイムごとの有限な評価区間における最適化により制御入力を再生成する。図2.3にサンプルした状態と適用する制御入力との関係を示す。サンプリング時刻 $t_i$ における最適な制御入力 $U_i(t)$ の生成過程は、大きく次の3ステップに分けられる。

Step 1 : 状態のサンプリング

時刻 $t_i$ における制御対象の状態 $X_R(t_i)$ を取得する。

Step 2 : 最適化演算

時刻 $t_i$ の情報をもとに有限時間の最適制御問題の最適化を行い、最適制御変数 $\{\hat{u}(t_i|t_i), \hat{u}(t_{i+1}|t_i), \dots, \hat{u}(t_{i+H_p-1}|t_i)\}$ を算出する。

Step 3 : 制御入力の適用

最適化で得られた最適制御変数の1ステップ目の値 $\hat{u}(t_i|t_i)$ を抽出し、 $U_i = \hat{u}(t_i|t_i)$ として適用して制御対象に入力する。次のサンプリング時刻になったならば、Step 1に戻り同様の操作を繰り返す。

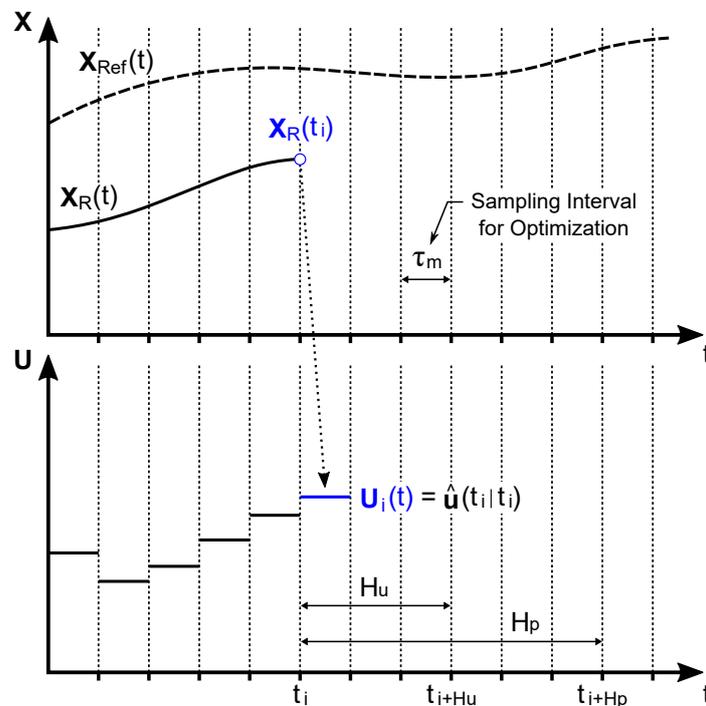


図 2.3 MPCにおけるサンプリングした状態と適用する制御入力との関係

以下に、参照軌道  $\mathbf{X}_{Ref}(t)$  に追従するための制御入力  $\mathbf{U}_i(t)$  の生成方法について、各ステップについて詳細に説明する。

### Step 1 : 状態のサンプリング

図 2.4 に MPC の模式的な時間応答におけるサンプリング時刻  $t_i$  の状態を示す。サンプリング時刻  $t_i$  における制御対象の状態  $\mathbf{X}_R(t_i)$  を取得する。この際、センサ等の制約により必要な状態が得られない場合は、状態推定を行う。

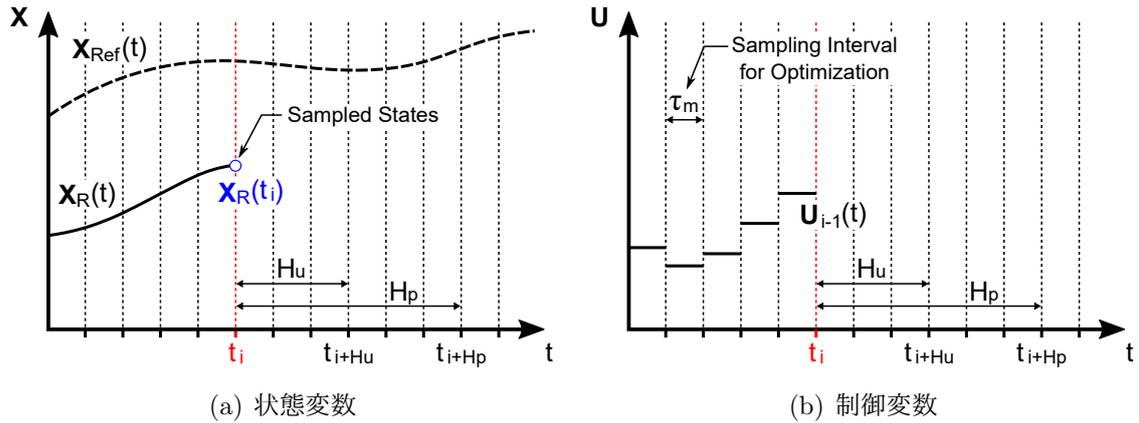


図 2.4 MPC の模式的な時間応答 (Step 1 : 状態のサンプリング)

### Step 2 : 最適化演算

図 2.5 にサンプリング時刻  $t_i$  における最適化計算で導出される軌道の概要を示す。Step 1 にて取得した  $\mathbf{X}_R(t_i)$  は最適化計算の初期条件として使用する。サンプリング時刻から  $H_p$  ステップ先まで (予測ホライズン) が最適化の評価区間であり、評価関数  $J_m$  を最小化する最適制御変数  $\{\hat{\mathbf{u}}(t_i|t_i), \hat{\mathbf{u}}(t_{i+1}|t_i), \dots, \hat{\mathbf{u}}(t_{i+H_p-1}|t_i)\}$  を算出する。サ

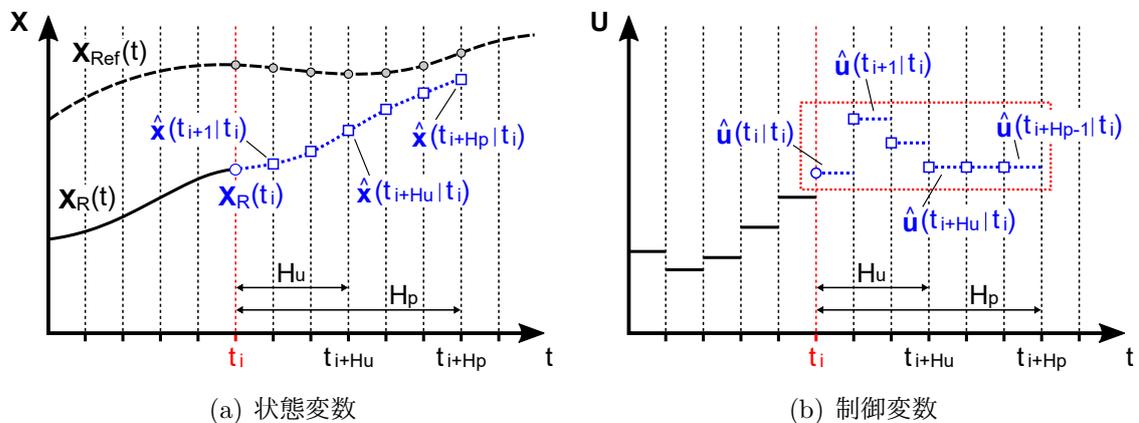


図 2.5 MPC の模式的な時間応答 (Step 2 : 最適化)

ンプリング間の入力は一定値,  $H_u$  ステップ (制御ホライズン) 以降の入力値は式 (2.1) のように全て同じと考える.

$$\hat{\mathbf{u}}(t_{i+H_u}|t_i) = \hat{\mathbf{u}}(t_{i+H_u+1}|t_i) = \cdots = \hat{\mathbf{u}}(t_{i+H_p-1}|t_i) \quad (2.1)$$

$H_p$  ステップ間の制御対象の状態  $\{\hat{\mathbf{x}}(t_{i+1}|t_i), \hat{\mathbf{x}}(t_{i+2}|t_i), \cdots, \hat{\mathbf{x}}(t_{i+H_p}|t_i)\}$  は, 式 (2.2) の運動モデル及びオイラー法に基づく式 (2.3) により予測する.

$$\dot{\mathbf{X}}(t) = \mathbf{f}(\mathbf{X}(t), \mathbf{U}(t)) \quad (2.2)$$

$$\begin{aligned} \hat{\mathbf{x}}(t_{i+1}|t_i) &= \mathbf{X}_R(t_i) + \mathbf{f}(\mathbf{X}_R(t_i), \hat{\mathbf{u}}(t_i|t_i))\tau_m \\ \hat{\mathbf{x}}(t_{i+2}|t_i) &= \hat{\mathbf{x}}(t_{i+1}|t_i) + \mathbf{f}(\hat{\mathbf{x}}(t_{i+1}|t_i), \hat{\mathbf{u}}(t_{i+1}|t_i))\tau_m \\ \hat{\mathbf{x}}(t_{i+3}|t_i) &= \hat{\mathbf{x}}(t_{i+2}|t_i) + \mathbf{f}(\hat{\mathbf{x}}(t_{i+2}|t_i), \hat{\mathbf{u}}(t_{i+2}|t_i))\tau_m \\ &\cdots \\ \hat{\mathbf{x}}(t_{i+H_p}|t_i) &= \hat{\mathbf{x}}(t_{i+H_p-1}|t_i) + \mathbf{f}(\hat{\mathbf{x}}(t_{i+H_p-1}|t_i), \hat{\mathbf{u}}(t_{i+H_p-1}|t_i))\tau_m \end{aligned} \quad (2.3)$$

制御対象の制御変数に制限がある場合には, 式 (2.4) の制約を与える.

$$\begin{aligned} \mathbf{h}_m^L &\leq \mathbf{h}_m(\hat{\mathbf{u}}(t_i|t_i)) \leq \mathbf{h}_m^U \\ \mathbf{h}_m^L &\leq \mathbf{h}_m(\hat{\mathbf{u}}(t_{i+1}|t_i)) \leq \mathbf{h}_m^U \\ &\vdots \\ \mathbf{h}_m^L &\leq \mathbf{h}_m(\hat{\mathbf{u}}(t_{i+H_u}|t_i)) \leq \mathbf{h}_m^U \end{aligned} \quad (2.4)$$

以上より, MPC で解くべき最適制御問題は,  $\{\hat{\mathbf{u}}(t_i|t_i), \hat{\mathbf{u}}(t_{i+1}|t_i), \cdots, \hat{\mathbf{u}}(t_{i+H_p-1}|t_i)\}$  を最適化パラメータとした式 (2.5) の非線形パラメータ最適化問題となる.

$$(\mathbf{P}_m) \left\{ \begin{array}{l} \text{Minimize} \quad J_m = E_m(\hat{\mathbf{x}}(t_{i+H_p}), t_{i+H_p}) \\ \quad \quad \quad + \sum_{k=0}^{H_p} F_m(\hat{\mathbf{x}}(t_{i+k}), \hat{\mathbf{u}}(t_{i+k}), t_{i+k})\tau_m \\ \text{Subject to} \quad \hat{\mathbf{x}}(t_{i+k}) - \hat{\mathbf{x}}(t_{i+k-1}) = \mathbf{f}(\hat{\mathbf{x}}(t_{i+k-1}), \hat{\mathbf{u}}(t_{i+k-1}))\tau_m \\ \quad \quad \quad k = 1, 2, \cdots, H_p \\ \quad \quad \quad \mathbf{X}_R(t_i) - \hat{\mathbf{x}}(t_i) = 0 \\ \quad \quad \quad \hat{\mathbf{u}}(t_{i+H_u}|t_i) = \hat{\mathbf{u}}(t_{i+H_u+1}|t_i) = \cdots = \hat{\mathbf{u}}(t_{i+H_p-1}|t_i) \\ \quad \quad \quad \mathbf{h}_m^L \leq \mathbf{h}_m(\hat{\mathbf{u}}(t_i|t_i)) \leq \mathbf{h}_m^U \\ \quad \quad \quad \vdots \\ \quad \quad \quad \mathbf{h}_m^L \leq \mathbf{h}_m(\hat{\mathbf{u}}(t_{i+H_u}|t_i)) \leq \mathbf{h}_m^U \end{array} \right. \quad (2.5)$$

問題  $\mathbf{P}_m$  は非線形計画問題 (Nonlinear Programming Problem: NLP) であり, 逐次二次計画 (Sequential Quadratic Programming: SQP) 法などの NLP ソルバーにて解くことができる。

Step 3 : 制御入力 of 適用

図 2.6 に示すように, Step 2 で得られた最適制御入力  $\{\hat{\mathbf{u}}(t_i|t_i), \hat{\mathbf{u}}(t_{i+1}|t_i), \dots, \hat{\mathbf{u}}(t_{i+H_p-1}|t_i)\}$  のうち, 1 ステップ目の値  $\hat{\mathbf{u}}(t_i|t_i)$  を制御対象への入力として適用する。従って,  $i$  番目の入力  $\mathbf{U}_i(t)$  は式 (2.6) で与えられ, この値を次の入力 that 得られるまで適用する。

$$\mathbf{U}_i(t) = \hat{\mathbf{u}}(t_i|t_i) \tag{2.6}$$

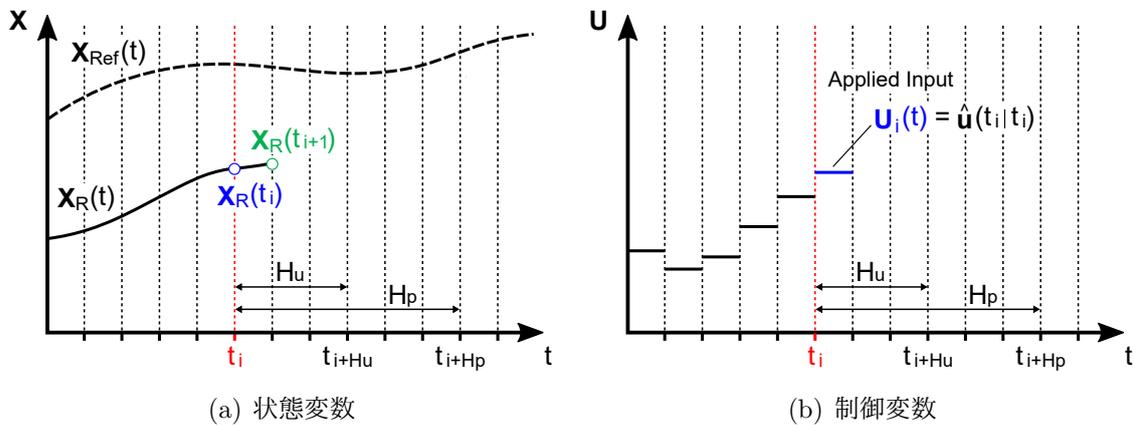


図 2.6 MPC の模式的な時間応答 (Step 3 : 制御入力の適用)

ただし, 図 2.7 に示すように, 実際には最適化計算による時間遅れ (Delay) が生じるため, 最適制御入力 that 適用されるのは計算完了以降となる。従って, サンプル間隔  $\tau_m$  に対して計算時間が十分に速いことが必要である。

次のサンプリング時刻になったならば, 新たに制御対象の状態を取得して Step 1 ~ Step 3 を繰り返す。同様の手順を目標 that 達成されるまで行う。

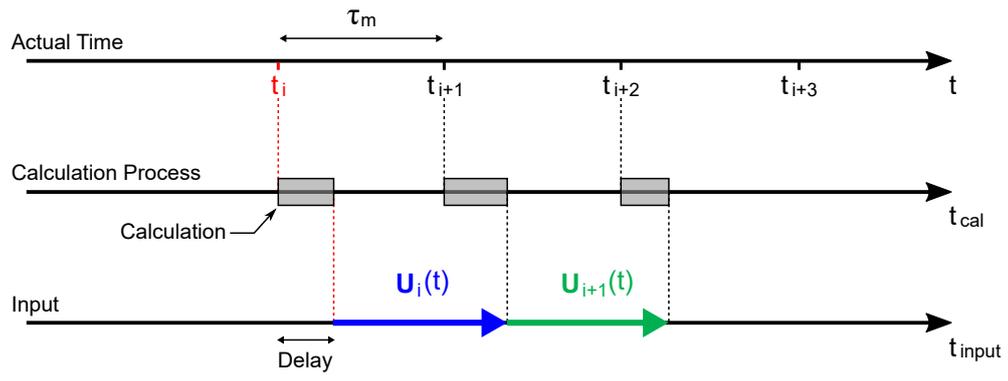


図 2.7 MPC の最適化計算による時間遅れと適用する制御入力の関係

## 2.3 Carathéodory- $\pi$ 手法 (C- $\pi$ 手法)

### 2.3.1 概要

ここでは、最初に Clock-Based 実時間最適制御の背景について述べる。式 (2.7) の非線形システムの状態フィードバック制御を考える。

$$\dot{x}(t) = f(x(t), u(t), t) \quad (2.7)$$

このシステムに対して、制御変数が式 (2.8) に示される状態変数の値を引数とする関数のフィードバック制御で与えられると考えると、制御システムは式 (2.9) となる。

$$u(t) = k(x(t)) \quad (2.8)$$

$$\dot{x}(t) = f(x(t), k(x(t)), t) \quad (2.9)$$

関数  $k(x(t))$  が与えられれば、システムの自動制御が可能となるが、この制御系の設計は非常に困難であると指摘されており [91, 92]、簡単な漸近安定化問題に対しても連続関数  $k(x(t))$  が存在しない場合があることが指摘されている [93, 94]。また与えられた評価関数を最小化する最適制御問題において、Hamilton-Jacobi-Bellman (HJB) 方程式を解くことができれば、式 (2.8) に対して最適フィードバック制御を実現できるが、HJB 方程式の解を得ることはより困難である [95]。そのため、連続関数の最適フィードバック制御では近似最適解を用いる方法が提案されている [96, 97]。

この問題に対して、区分化した制御区間ごとにフィードバック制御を行い、これをつなぎ合わせることで非線形フィードバック制御を実現させる方法が Clarke らによって提案されている [91, 94]。Clarke らは時間領域を変数  $\pi$  で表される区間に分割を行い、各区間の入り口における状態変数のサンプリング値を用いて  $\pi$  解 ( $\pi$ -solution) として制御変数を算出し制御入力として与えることを考えた。そして、 $\pi$ -solution を区間ごとに逐次与えることで得られる応答である  $\pi$  軌道 ( $\pi$ -trajectory) について、漸近安定性などの解析を行っている [91, 94]。しかしながら、各区間の  $\pi$ -solution は状態変数で定まる一定値とした Sample-and-Hold で与えられており、算出方法については明確な議論がなされていなかった。

Ross らは、この  $\pi$ -trajectory の方法を発展させた方法を提案している [81]。式 (2.7) に対して、式 (2.10) のように状態変数  $x(t)$  と時刻  $t$  を引数とする関数を考えると、式 (2.11) の非線形状態フィードバック制御系が得られる。

$$u(t) = k(t, x(t)) \quad (2.10)$$

$$\dot{x}(t) = f(x(t), k(t, x(t)), t) \quad (2.11)$$

この場合にも、連続関数  $k(t, x(t))$  の算出が困難なため、Clarke の考え方と同様に  $\pi$  区間に分割して考える。ただし、各区間で算出される制御値を得るための関数は hold 型ではなく、時間  $t$  による変化が許容されており、これは微分方程式の初期値問題と考えることができる。この初期値問題には「関数が不連続であっても初期値近傍において少なくとも一つ解を持つ」ことを示した Carathéodory の存在定理が知られており、その解は Carathéodory 解 (Carathéodory-solution) と呼ばれる。そこで、各  $\pi$  区間の入り口における状態変数のサンプリング値を初期値として算出した Carathéodory 解を新たに Carathéodory- $\pi$  解と呼称して、これを逐次制御入力として適用して得られる応答を、Carathéodory- $\pi$  軌道 (Carathéodory- $\pi$  trajectory) としている。Ross らは、この提案手法においてサンプリング間隔、システムの動特性、許容誤差の関係を解析的に示し [81]、さらに安定性の議論も行っている [82]。

時間に対して不変もしくは一定変化する Sample-and-Hold 型の実時間最適制御に対して、Carathéodory- $\pi$  解を制御入力として適用する場合は、時計 (Clock) により時刻  $t$  を参照するため「Clock-Based」実時間最適制御と呼ばれている。この手法では、各区間の Carathéodory- $\pi$  解である関数  $k(t, x(t))$  を生成する必要がある。Ross らは、各サンプリング時刻で得られた状態変数と、最終目的を終端条件とした非線形最適制御問題を設定して最適化を行い、得られた最適制御解の中から、必要な制御区間の部分を抽出する方法を提案している。従って、安定性や誤差に影響を与えるサンプリング間隔の設定は、この非線形最適制御問題の最適解を得るための計算速度に大きく依存する。

### 2.3.2 サンプリングタイムとダイナミクスの関係

式 (2.12) の非線形制御システムを考える。なお、本論文の C- $\pi$  手法等の理論解析ではベクトルの太字表記を省略している。 $p_0 \in \mathbb{R}^{N_p}$  は、動的システムであれば質量や慣性モーメントなどの時不変パラメータである。

$$\dot{x}_M(t) = f(x, u, t; p_0) \quad (2.12)$$

しかしながら、現実には制御対象の不完全なモデル化、未知もしくはモデル化されていない外乱、推定誤差などの多様な不確定要素が存在する。そこで、これらの要素の影響を可積分関数  $t \mapsto \eta(t)$  を用いて仮定すると、実システムのダイナミクスは式 (2.13) で記述される。

$$\dot{x}_R(t) = f(x, u, t; p) + \eta(t) \quad (2.13)$$

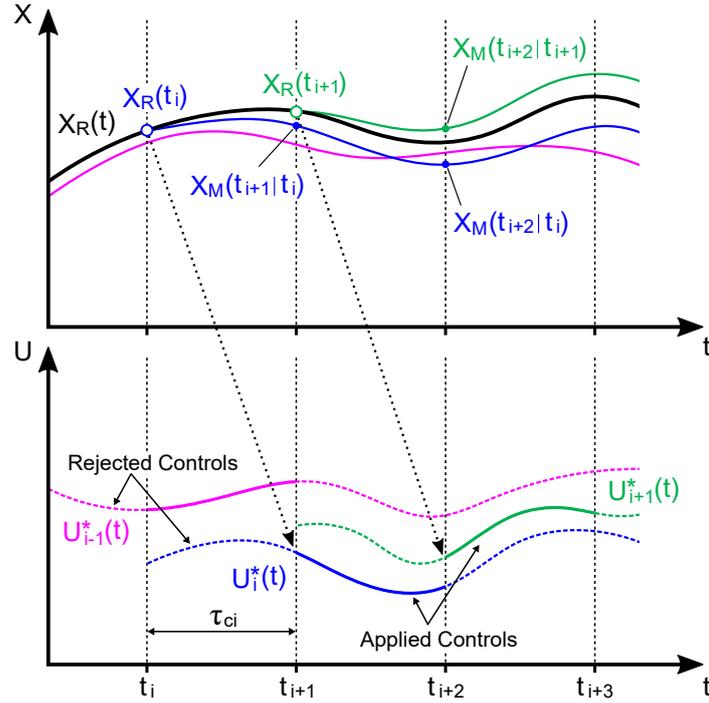


図 2.8 C- $\pi$  手法におけるサンプリングした状態と適用する制御入力の関係

式 (2.13) の  $p$  は、実システムにおけるパラメータの真値であり、完全なモデル化の場合は  $p = p_0$  及び  $\|\eta\| = 0$  となる。

図 2.8 に、C- $\pi$  手法におけるサンプリングした状態と適用する制御入力の関係を示す。各サンプリング時刻において取得した状態を初期条件として最適制御解を生成している。実システムでは、制御対象の状態をサンプリングした後から最適化計算を行う時間が必要であるため、MPC の様に取得した状態を反映した最適制御入力を直後に適用することは不可能である。そこで C- $\pi$  手法では、1 つ前の区間において得られた最適制御入力の対応する区間を抽出して適用する方法を用いる。

ここで、サンプリング時刻  $t_i$  において計測した  $x_R(t_i)$  をもとに予測される次のサンプリング時刻  $t_{i+1}$  の状態  $x_M(t_{i+1}|t_i)$  と、 $t_{i+1}$  における実軌道の値  $x_R(t_{i+1})$  の誤差について考える。得られた制御入力を用いた開ループ制御  $[t_i, t_{i+1}] \mapsto k(t, x_R(t_i))$  により、 $t_{i+1}$  におけるモデルの状態  $x_M(t_{i+1}|t_i)$  は式 (2.14) で与えられる。

$$x_M(t_{i+1}|t_i) = x_R(t_i) + \int_{t_i}^{t_{i+1}} f(x_M(t), k(t, x_R(t_i)), t; p_0) dt \quad (2.14)$$

サンプリング間隔  $\tau_{ci}$  は式 (2.15) であり、必ずしも一定である必要はない。制御解が

得られるまでの時間や一定の間隔などに設定できる.

$$\tau_{ci} := t_{i+1} - t_i \quad (2.15)$$

時刻  $t_{i+1}$  における状態  $x_R(t_{i+1})$  は, 時刻  $t_{i-1}$  の状態  $x_R(t_{i-1})$  を用いて得られた制御則  $[t_i, t_{i+1}] \mapsto k(t, x_R(t_{i-1}))$  の入力に対する応答となるため, 式 (2.16) により得られる.

$$x_R(t_{i+1}) = x_R(t_i) + \int_{t_i}^{t_{i+1}} f(x_R(t), k(t, x_R(t_{i-1})), t; p) dt + \int_{t_i}^{t_{i+1}} \eta(t) dt \quad (2.16)$$

従って, 時刻  $t_{i+1}$  における実システムとモデルの誤差は式 (2.14) 及び (2.16) より式 (2.17) となる.

$$\begin{aligned} x_R(t_{i+1}) - x_M(t_{i+1}|t_i) &= \int_{t_i}^{t_{i+1}} f(x_R(t), k(t, x_R(t_{i-1})), t; p) dt \\ &\quad - \int_{t_i}^{t_{i+1}} f(x_M(t), k(t, x_R(t_i)), t; p_0) dt + \int_{t_i}^{t_{i+1}} \eta(t) dt \end{aligned} \quad (2.17)$$

関数  $(u, p) \mapsto f(x, u, t; p)$  がリプシッツ連続と考えると, 式 (2.18) が成り立つ.

$$\begin{aligned} \|x_R(t_{i+1}) - x_M(t_{i+1}|t_i)\| &\leq \text{Lip}f_x \int_{t_i}^{t_{i+1}} \|x_R(t) - x_M(t)\| dt \\ &\quad + \text{Lip}f_u \int_{t_i}^{t_{i+1}} \|k(t, x_R(t_{i-1})) - k(t, x_R(t_i))\| dt \\ &\quad + \text{Lip}f_p \|p - p_0\| \tau_{ci} + \|\eta\| \tau_{ci} \end{aligned} \quad (2.18)$$

ただし,  $\text{Lip}f$  は下付きの添え字に関するリプシッツ定数を表す.

さらに, Ross らは以下の補題及び証明を与えている.

【補題】

$\sum_i \alpha_i = 1$  となる  $\alpha \in (0, 1)$  に対して,  $\delta$  が式 (2.19)~(2.21) を満たす微小な正の数とし,

$$\|k(\cdot, x_R(t_{i-1})) - k(\cdot, x_R(t_i))\| \leq \alpha_1 \delta \frac{\text{Lip}f_x}{\text{Lip}f_u} \quad (2.19)$$

$$\|\eta\| \leq \alpha_2 \delta \text{Lip}f_x \quad (2.20)$$

$$\|p - p_0\| \leq \alpha_3 \delta \frac{\text{Lip}f_x}{\text{Lip}f_p} \quad (2.21)$$

ある正の数  $\epsilon$  に対して式 (2.22) であるならば,

$$\tau_{ci} \leq \frac{W(r)}{\text{Lip}f_x} \quad (2.22)$$

式 (2.23) が成り立つ.

$$\|x_R(t_{i+1}) - x_M(t_{i+1}|t_i)\| \leq \epsilon \quad (2.23)$$

ただし,  $W(r)$  は  $r := \epsilon/\delta$  としたランベルトの  $W$  関数である.

(証明)

式 (2.18)~(2.21) より式 (2.24) となる.

$$\|x_R(t_{i+1}) - x_M(t_{i+1}|t_i)\| \leq \text{Lip}f_x \int_{t_i}^{t_{i+1}} \|x_R(t) - x_M(t)\| dt + \delta \text{Lip}f_x \tau_{ci} \quad (2.24)$$

さらに Gronwall の不等式より, 式 (2.25) が成り立つ.

$$\|x_R(t_{i+1}) - x_M(t_{i+1}|t_i)\| \leq \delta \text{Lip}f_x \tau_{ci} \exp(\text{Lip}f_x \tau_{ci}) \quad (2.25)$$

式 (2.22) より  $\text{Lip}f_x \tau_{ci} \exp(\text{Lip}f_x \tau_{ci}) \leq W(r) \exp(W(r))$  となり, ランベルトの  $W$  関数は  $r = W(r) \exp(W(r))$  であるため式 (2.26) が成り立つ.

$$\|x_R(t_{i+1}) - x_M(t_{i+1}|t_i)\| \leq r\delta = (\epsilon/\delta)\delta = \epsilon \quad (2.26)$$

(証明終わり)

式 (2.23) は, 実際の軌道と予測した軌道の誤差はある  $\epsilon$  以内に収まることを意味している. つまり式 (2.22) は, 誤差を小さくするためにはサンプリングタイムを短くする必要があり, また  $\text{Lip}f_x$  の小さいシステムではサンプリングタイムを長くできることを定性的に表している.

### 2.3.3 制御入力の生成方法

C- $\pi$  手法は, MPC と異なり, サンプリングした状態から目標状態までの全区間を評価区間として最適化計算を行う. サンプリング時刻  $t_i$  の情報を利用した  $t_{i+1} \sim t_{i+2}$  で制御対象に適用する  $U_{i+1}(t)$  の生成過程は, 大きく次の 3 ステップに分けられる.

Step 1 : 状態のサンプリング

時刻  $t_i$  における制御対象の状態  $\mathbf{X}_R(t_i)$  を取得する.

Step 2 : 最適化演算

$t_i \sim t_{i+1}$  において, 目標状態までの最適制御解  $\{\mathbf{X}_i^*(t), \mathbf{U}_i^*(t)\}$  を生成する.

Step 3 : 制御入力の実用

$t_{i+1} \sim t_{i+2}$  では, 最適制御入力  $\mathbf{U}_i^*(t)$  の対応する区間を抽出し,  $\mathbf{U}_{i+1}(t)$  として適用する. 次のサンプリング時刻になったならば, Step 1 に戻り同様の操作を繰り返す.

以下に, サンプリング時刻  $t_i$  の情報をもととした制御入力  $\mathbf{U}_{i+1}(t)$  の生成方法について, 各ステップについて詳細に説明する.

Step 1 : 状態のサンプリング

図 2.9 に C- $\pi$  手法の模式的な時間応答におけるサンプリング時刻  $t_i$  の状態を示す. サンプリング時刻  $t_i$  における制御対象の状態  $\mathbf{X}_R(t_i)$  を取得する.

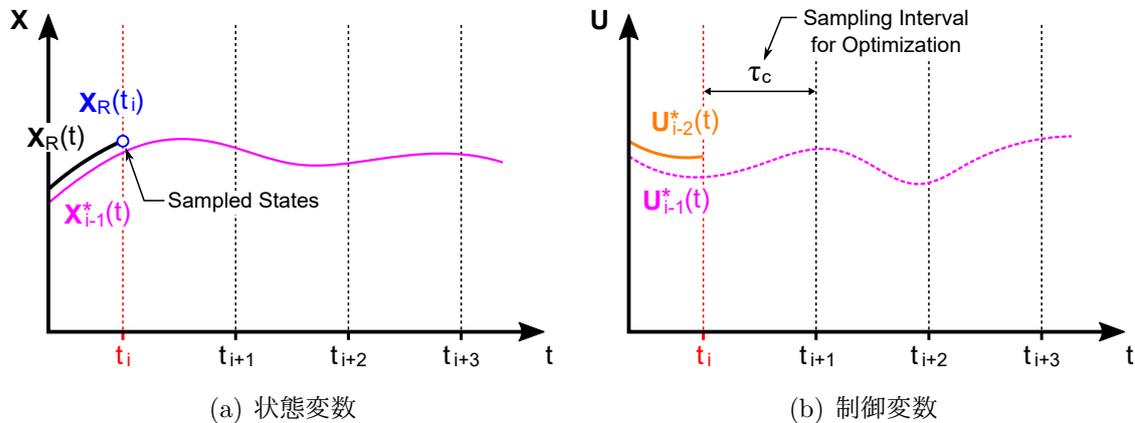


図 2.9 C- $\pi$  手法の模式的な時間応答 (STEP 1 : 状態のサンプリング)

Step 2 : 最適化演算

図 2.10 に  $t_i \sim t_{i+1}$  における最適化計算で導出される最適状態軌道及び最適制御入力の概要を示す. Step 1 にて取得した  $\mathbf{X}_R(t_i)$  は最適化計算の初期条件として使用する. 制御対象の運動モデル, 境界条件, 及び拘束条件が式 (2.27)~(2.29) で与えられるとすると, C- $\pi$  手法における最適制御問題  $\mathbf{P}_c$  は式 (2.30) のように定式化することができる. C- $\pi$  手法では, 状態をサンプリングした時刻から最終目標の到達状態までを最適化する評価区間としている. 時刻  $t_i \sim t_{i+1}$  では,  $\mathbf{X}_R(t_i)$  を初期条件とした最適化計算を行っているため, 図 2.10 に示すように制御入力には  $\mathbf{U}_{i-1}^*(t)$  の対応する区間を使用

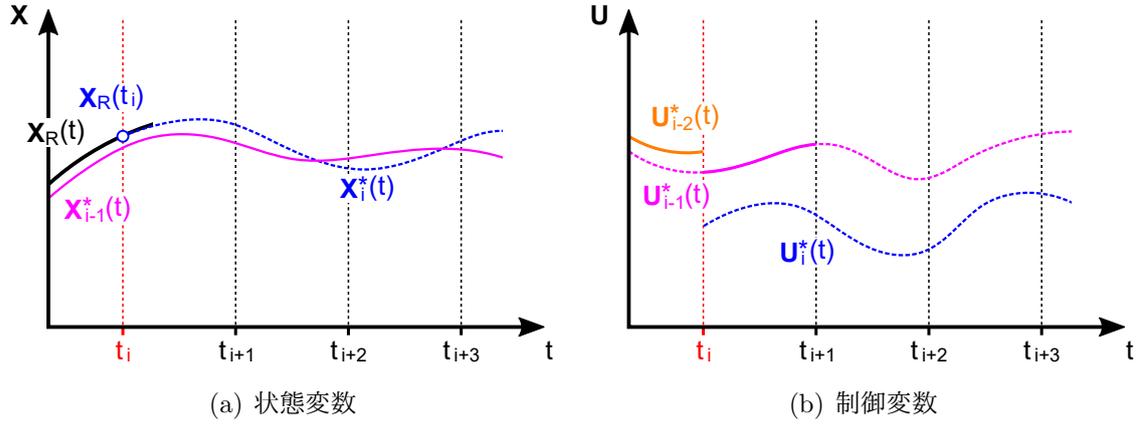


図 2.10 C- $\pi$  手法の模式的な時間応答 (STEP 2 : 最適化)

する. 定式化した最適制御問題を解くことにより, 最適状態軌道  $\mathbf{X}_i^*(t)$  及び最適制御入力  $\mathbf{U}_i^*(t)$  が導出される.

$$\dot{\mathbf{X}}_M(t) = \mathbf{f}(\mathbf{X}_M(t), \mathbf{U}(t)) \quad (2.27)$$

$$\mathbf{e}_c^L \leq \mathbf{e}_c(\mathbf{X}_R(t_i), \mathbf{X}_R(t_f)) \leq \mathbf{e}_c^U \quad (2.28)$$

$$\mathbf{h}_c^L \leq \mathbf{h}_c(\mathbf{X}_M(t), \mathbf{U}(t)) \leq \mathbf{h}_c^U \quad (2.29)$$

$$(\mathbf{P}_c) \left\{ \begin{array}{l} \text{Minimize} \quad J_c = E_c(\mathbf{X}_M(t_i), \mathbf{X}(t_f), t_i, t_f) \\ \quad \quad \quad + \int_{t_i}^{t_f} F_c(\mathbf{X}_M(t), \mathbf{U}(t), t) dt \\ \text{Subject to} \quad \dot{\mathbf{X}}_M(t) - \mathbf{f}(\mathbf{X}_M(t), \mathbf{U}(t)) = \mathbf{0} \\ \quad \quad \quad \mathbf{e}_c^L \leq \mathbf{e}_c(\mathbf{X}_R(t_i), \mathbf{X}_R(t_f)) \leq \mathbf{e}_c^U \\ \quad \quad \quad \mathbf{h}_c^L \leq \mathbf{h}_c(\mathbf{X}_M(t), \mathbf{U}(t)) \leq \mathbf{h}_c^U \end{array} \right. \quad (2.30)$$

### Step 3 : 制御入力の適用

図 2.11 に時刻  $t_{i+1}$  の状態を示す. 時刻が  $t_{i+1}$  になったならば, 再び制御対象の状態  $\mathbf{X}_R(t_{i+1})$  を取得して最適化計算を行う.  $t_{i+1}$  から  $t_{i+2}$  では, Step 2 の最適化により  $\mathbf{X}_R(t_i)$  を初期条件とした最適状態軌道  $\mathbf{X}_i^*(t)$  及び最適制御入力  $\mathbf{U}_i^*(t)$  が得られているため,  $\mathbf{U}_i^*(t)$  の  $t_{i+1} \sim t_{i+2}$  の区間を使用する. 次のサンプリング時刻になったならば, 新たに制御対象の状態を取得して Step 1~Step 3 を繰り返す. 同様の手順を目標が達成されるまで行う.

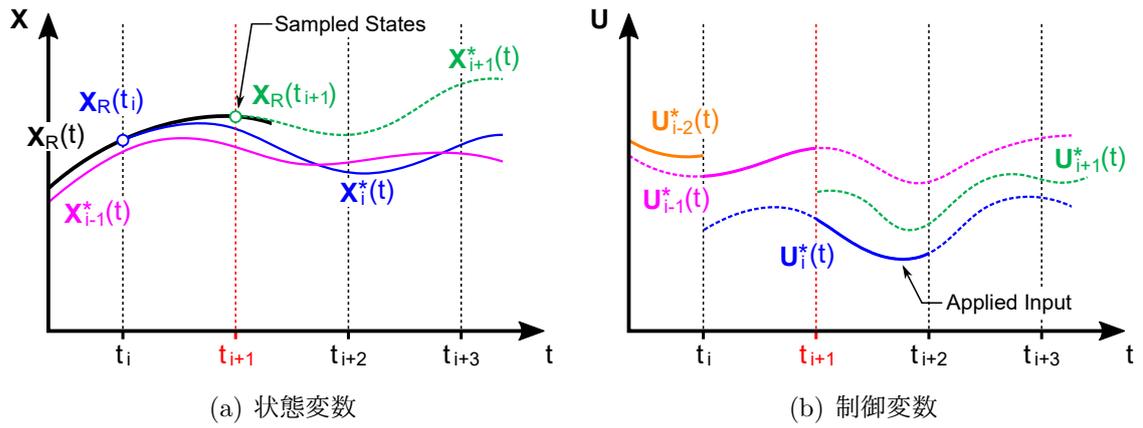


図 2.11 C- $\pi$  手法の模式的な時間応答 (STEP 3: 制御入力 of 適用)

## 2.4 予測型 Carathéodory- $\pi$ 手法 (PC- $\pi$ 手法)

### 2.4.1 概要

C- $\pi$  手法では, サンプルングした状態を最適制御問題の初期条件として与えている. 一方, 得られた最適制御解と運動モデルを用いて次のサンプルング時刻における制御対象の状態を予測 (プロパゲーション) し, 算出した状態を初期条件として最適化計算を行う手法も提案されている [82]. 本研究では, 本手法を Propagation Carathéodory- $\pi$  手法 (PC- $\pi$  手法) と呼ぶ. ここでは, まず PC- $\pi$  手法による制御入力の生成方法について述べ, 次に C- $\pi$  手法との比較を行う.

### 2.4.2 制御入力の生成方法

PC- $\pi$  手法は, 次のサンプルング時刻における予測状態から目標状態までを評価区間として最適化計算を行う. サンプルング時刻  $t_i$  の情報をもととした  $t_{i+1} \sim t_{i+2}$  で制御対象に適用する  $U_{i+1}(t)$  の生成過程は, 大きく次の 4 ステップに分けられる.

Step 1 : 状態のサンプルング

時刻  $t_i$  における制御対象の状態  $X_R(t_i)$  を取得する.

Step 2 : 次のサンプルング時刻における状態の予測

既に得られている最適制御入力  $U_{i-1}^*(t)$  及び運動モデルを用いて, 次のサンプルング時刻における状態を予測する.

Step 3 : 最適化演算

$t_i \sim t_{i+1}$  において, 目標状態までの最適制御解  $\{X_i^*(t), U_i^*(t)\}$  を生成する.

Step 4 : 制御入力の適用

Step 3 で得られた最適制御解の制御区間は  $t_{i+1} \sim t_f$  であるため,  $t_{i+1} \sim t_{i+2}$  ではそのまま  $U_i^*(t)$  を  $U_{i+1}(t)$  として適用する. 次のサンプルング時刻になったならば, Step 1 に戻り同様の操作を繰り返し行う.

以下に, サンプルング時刻  $t_i$  の情報を利用した制御入力  $U_{i+1}(t)$  の生成方法について, 各ステップについて詳細に説明する.

### Step 1 : 状態のサンプリング

図 2.12 に PC- $\pi$  手法の模式的な時間応答におけるサンプリング時刻  $t_i$  の状態を示す。サンプリング時刻  $t_i$  における制御対象の状態  $\mathbf{X}_R(t_i)$  を取得する。

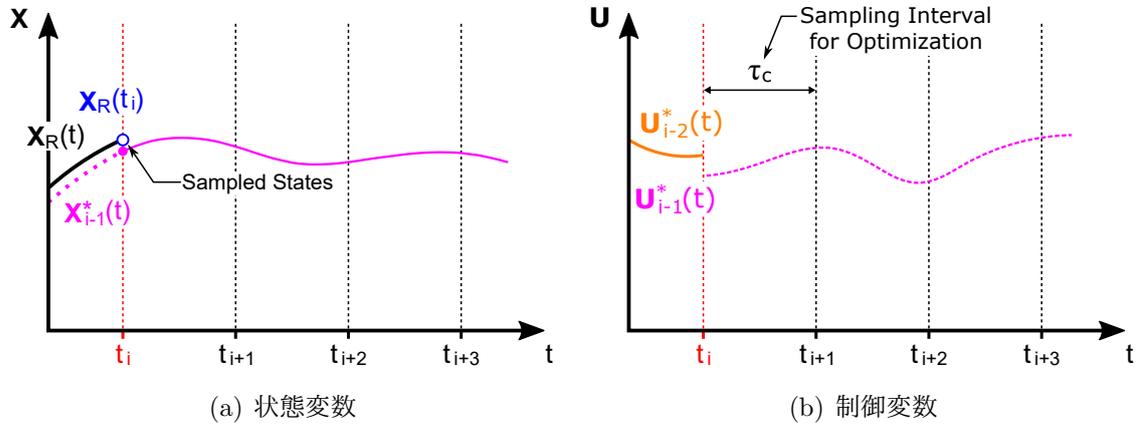


図 2.12 PC- $\pi$  手法の模式的な時間応答 (STEP 1 : 状態のサンプリング)

### Step 2 : 次のサンプリング時刻における状態の予測

図 2.13 に PC- $\pi$  手法におけるプロパゲーションの概要を示す。時刻  $t_i$  においては、1 つ前の最適化計算により最適制御入力  $\mathbf{U}_{i-1}^*(t)$  が得られている。Step 1 で取得した  $\mathbf{X}_R(t_i)$  を初期値として、 $\mathbf{U}_{i-1}^*(t)$  及び運動モデルを用いて時刻  $t_{i+1}$  における制御対象の状態  $\mathbf{X}_M(t_{i+1}|t_i)$  を開ループ制御で予測する。

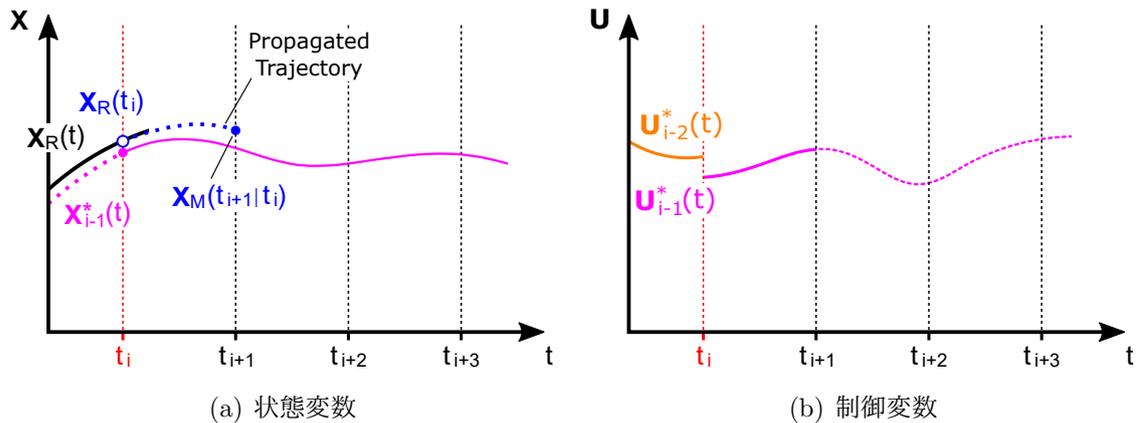


図 2.13 PC- $\pi$  手法の模式的な時間応答 (STEP 2 : プロパゲーション)

### Step 3 : 最適化演算

図 2.14 に  $t_i \sim t_{i+1}$  における最適化計算で導出される最適状態軌道及び最適制御入力の概要を示す。Step 2 にて算出した  $\mathbf{X}_M(t_{i+1}|t_i)$  は最適化計算の初期条件として使用する。制御対象の運動モデル、境界条件、及び拘束条件が式 (2.31) ~ (2.33) で与えられる



グ時刻になったならば、新たに制御対象の状態を取得して Step 1~Step 4 を繰り返す。同様の手順を目標が達成されるまで行う。

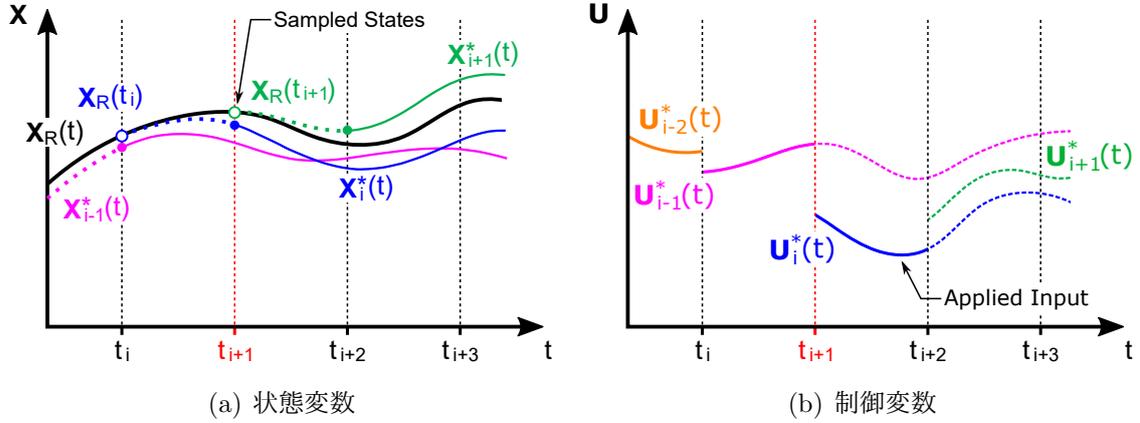


図 2.15 PC- $\pi$  手法の模式的な時間応答 (STEP 4: 制御入力適用)

### 2.4.3 C- $\pi$ 手法との比較

PC- $\pi$  手法では、得られた制御入力を用いた開ループ制御  $[t_i, t_{i+1}] \mapsto k(t, x_R(t_{i-1}))$  により、 $t_{i+1}$  におけるモデルの状態  $x_M(t_{i+1}|t_i)$  は式 (2.35) で与えられる。

$$x_M(t_{i+1}|t_i) = x_R(t_i) + \int_{t_i}^{t_{i+1}} f(x_M(t), k(t, x_R(t_{i-1})), t; p_0) dt \quad (2.35)$$

時刻  $t_{i+1}$  における状態  $x_R(t_{i+1})$  は、同じ制御則  $[t_i, t_{i+1}] \mapsto k(t, x_R(t_{i-1}))$  の入力に対する応答となるため、式 (2.36) により得られる。

$$x_R(t_{i+1}) = x_R(t_i) + \int_{t_i}^{t_{i+1}} f(x_R(t), k(t, x_R(t_{i-1})), t; p) dt + \int_{t_i}^{t_{i+1}} \eta(t) dt \quad (2.36)$$

従って、実システムとモデルの誤差は式 (2.35) 及び (2.36) より式 (2.37) となる。

$$\begin{aligned} x_R(t_{i+1}) - x_M(t_{i+1}|t_i) &= \int_{t_i}^{t_{i+1}} f(x_R(t), k(t, x_R(t_{i-1})), t; p) dt \\ &\quad - \int_{t_i}^{t_{i+1}} f(x_M(t), k(t, x_R(t_{i-1})), t; p_0) dt + \int_{t_i}^{t_{i+1}} \eta(t) dt \end{aligned} \quad (2.37)$$

C- $\pi$  手法の場合と同様に、関数  $(u, p) \mapsto f(x, u, t; p)$  がリプシッツ連続と考えると、式 (2.38) が成り立つ。

$$\begin{aligned}
\|x_R(t_{i+1}) - x_M(t_{i+1}|t_i)\| &\leq \text{Lip}f_x \int_{t_i}^{t_{i+1}} \|x_R(t) - x_M(t)\| dt \\
&+ \text{Lip}f_u \int_{t_i}^{t_{i+1}} \|k(t, x_R(t_{i-1})) - k(t, x_R(t_{i-1}))\| dt \quad (2.38) \\
&+ \text{Lip}f_p \|p - p_0\| \tau_{ci} + \|\eta\| \tau_{ci}
\end{aligned}$$

式 (2.38) の右辺第 2 項では被積分関数は 0 となるため、C- $\pi$  手法における式 (2.18) と比較すると、測定値とモデル予測値の誤差は入力誤差の影響を受けない分だけ小さくなるのが分かる。しかしながら、PC- $\pi$  手法ではサンプリングタイム後の状態を予測するアルゴリズムを追加し、またサンプリングタイムは必ず固定値として扱わなければならない制約を受ける。

## 2.5 Gain型 M・C- $\pi$ 手法と Gain型 M・PC- $\pi$ 手法

### 2.5.1 概要

C- $\pi$ 手法やPC- $\pi$ 手法では、サンプリング間隔において、逐次生成した最適制御解のうち最適制御入力を用いて開ループ制御を行う。サンプリング間隔が短い時間では実軌道と最適状態軌道との誤差は微小であるが、最適化計算に要する時間の制約によりサンプリング間隔が長くなる場合では、誤差が大きくなる。そこでC- $\pi$ 手法及びPC- $\pi$ 手法の制御系に最適化計算で得られる最適状態軌道に対する追従制御器を組み込む方法を提案する。本節では追従制御器として固定ゲインの状態フィードバック制御を用いる場合をGain型と呼称して、それぞれの手法をGain Type Modified Carathéodory- $\pi$ 手法（Gain型 M・C- $\pi$ 手法）及びGain Type Modified Propagation Carathéodory- $\pi$ 手法（Gain型 M・PC- $\pi$ 手法）と呼ぶ。ここでは、まずGain型 M・C- $\pi$ 手法による制御入力の生成方法について述べ、次にC- $\pi$ 手法との比較を行う。最後にGain型 M・C- $\pi$ 手法の安定性の領域について述べる。Gain型 M・PC- $\pi$ 手法については、同様の考え方であるため省略する。

### 2.5.2 制御入力の生成方法

図 2.16 に Gain型 M・C- $\pi$ 手法における制御系のブロック線図を示す。C- $\pi$ 手法と同様に、サンプリングタイムごとに最適制御入力  $U^*(t)$  と最適状態軌道  $X^*(t)$  を生成する。さらにGain型 M・C- $\pi$ 手法では、 $X(t)$  と  $X^*(t)$  の誤差から追従制御器により  $U_T(t)$  を生成する。 $U^*(t)$  に  $U_T(t)$  を加えた値を制御対象へ入力する。

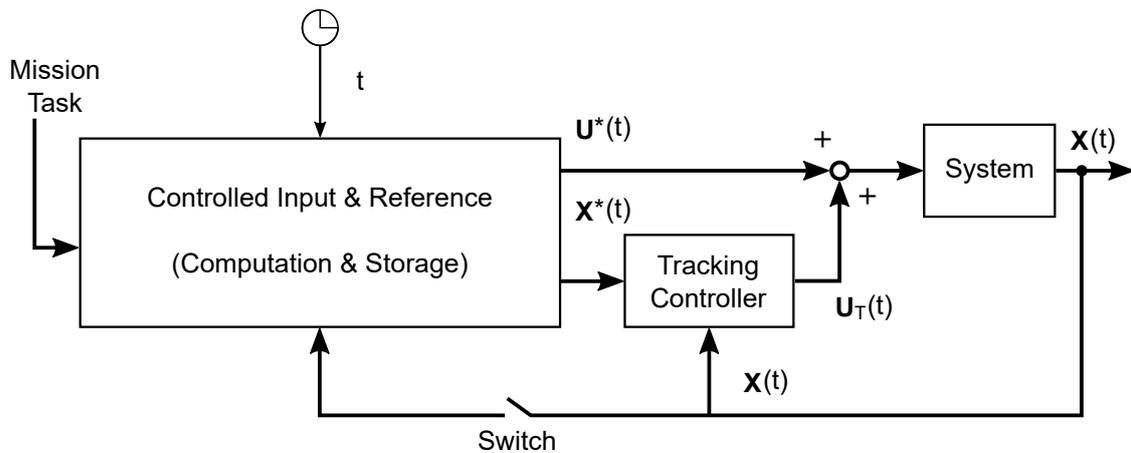


図 2.16 Gain型 M・C- $\pi$ 手法のブロック線図

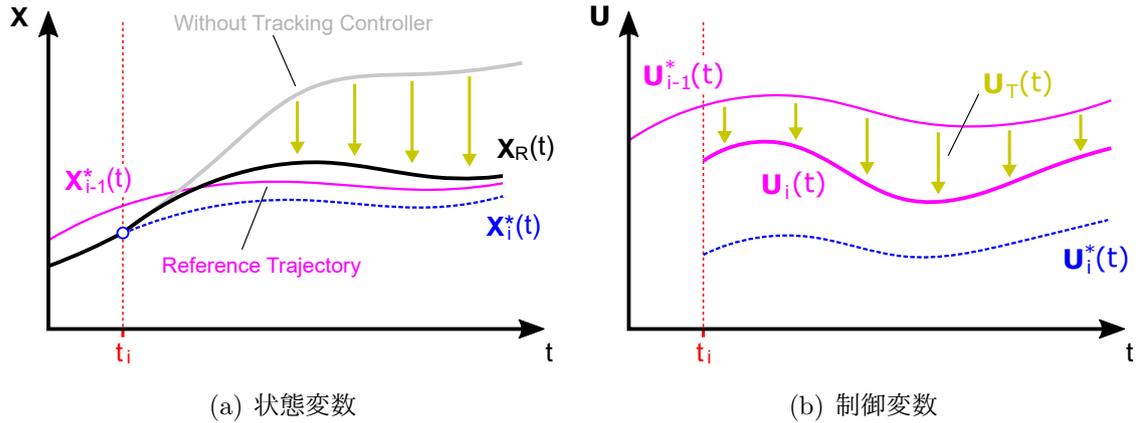


図 2.17 Gain 型 M · C- $\pi$  手法の模式的な時間応答

図 2.17 に Gain 型 M · C- $\pi$  手法の  $t_i \sim t_{i+1}$  における模式的な時間応答を示す。  $t_i \sim t_{i+1}$  では、  $\mathbf{X}_R(t_i)$  を初期条件とした最適化計算を行っているため、直前に生成した最適制御解として  $\mathbf{X}_{i-1}^*(t)$  及び  $\mathbf{U}_{i-1}^*(t)$  が存在する。そのため  $\mathbf{X}_{i-1}^*(t)$  を参照軌道として  $\mathbf{X}_R(t)$  との誤差から追従制御器により  $\mathbf{U}_T(t)$  を生成し、制御対象に入力する  $\mathbf{U}_i(t)$  を式 (2.39) で与える。

$$\mathbf{U}_i(t) = \mathbf{U}_{i-1}^*(t) + \mathbf{U}_T(t) \quad (2.39)$$

合成された制御入力  $\mathbf{U}_i(t)$  の具体的な生成方法については、制御対象によって異なるため、後述の 4 章 (MGV), 5 章 (一般車両) にてそれぞれ設計する。

### 2.5.3 C- $\pi$ 手法との比較

C- $\pi$  手法では時刻  $t_{i+1}$  における実システムの状態は式 (2.16) で与えられていたが、Gain 型 M · C- $\pi$  手法では誤差を修正するための入力として  $k_T(t, x_R(t), x_M(t))$  が加えられるため、時刻  $t_{i+1}$  における実システムの状態は式 (2.40) で表される。

$$\begin{aligned} x_R(t_{i+1}) &= x_R(t_i) \\ &+ \int_{t_i}^{t_{i+1}} f(x_R(t), k(t, x_R(t_{i-1})), k_T(t, x_R(t), x_M(t)), t; p) dt \\ &+ \int_{t_i}^{t_{i+1}} \eta(t) dt \end{aligned} \quad (2.40)$$

従って、実システムとモデルの誤差は式 (2.14) 及び (2.40) より式 (2.41) となる。

$$\begin{aligned}
& x_R(t_{i+1}) - x_M(t_{i+1}|t_i) \\
&= \int_{t_i}^{t_{i+1}} f(x_R(t), k(t, x_R(t_{i-1})), k_T(t, x_R(t), x_M(t)), t; p) dt \\
&- \int_{t_i}^{t_{i+1}} f(x_M(t), k(t, x_R(t_i)), t; p_0) dt + \int_{t_i}^{t_{i+1}} \eta(t) dt
\end{aligned} \tag{2.41}$$

関数  $(u, p) \mapsto f(x, u, t; p)$  がリプシッツ連続と考えると、式 (2.42) が成り立つ。

$$\begin{aligned}
\|x_R(t_{i+1}) - x_M(t_{i+1}|t_i)\| &\leq \text{Lip}f_x \int_{t_i}^{t_{i+1}} \|x_R(t) - x_M(t)\| dt \\
&+ \text{Lip}f_u \int_{t_i}^{t_{i+1}} \|k(t, x_R(t_{i-1})) - k(t, x_R(t_i))\| dt \\
&+ \text{Lip}f_u \int_{t_i}^{t_{i+1}} \|k_T(t, x_R(t), x_M(t)) - 0\| dt \\
&+ \text{Lip}f_p \|p - p_0\| \tau_{ci} + \|\eta\| \tau_{ci}
\end{aligned} \tag{2.42}$$

ここで、 $k_T(t, x_R(t), x_M(t))$  は  $x_R(t)$  と  $x_M(t)$  の誤差を減少させる追従制御器であるため、式 (2.42) の右辺第3項は式 (2.43) に変換することができる。ただし、 $K$  はフィードバックゲイン行列を表す。

$$\begin{aligned}
\int_{t_i}^{t_{i+1}} \|k_T(t, x_R(t), x_M(t))\| dt &= - \int_{t_i}^{t_{i+1}} \|K(x_R(t) - x_M(t))\| dt \\
&= -\|K\| \int_{t_i}^{t_{i+1}} \|x_R(t) - x_M(t)\| dt
\end{aligned} \tag{2.43}$$

式 (2.42) 及び (2.43) より式 (2.44) となる。

$$\begin{aligned}
\|x_R(t_{i+1}) - x_M(t_{i+1}|t_i)\| &\leq (\text{Lip}f_x - \text{Lip}f_u \|K\|) \int_{t_i}^{t_{i+1}} \|x_R(t) - x_M(t)\| dt \\
&+ \text{Lip}f_u \int_{t_i}^{t_{i+1}} \|k(t, x_R(t_{i-1})) - k(t, x_R(t_i))\| dt \\
&+ \text{Lip}f_p \|p - p_0\| \tau_{ci} + \|\eta\| \tau_{ci}
\end{aligned} \tag{2.44}$$

C- $\pi$  手法における式 (2.18)～(2.26) と同様の考え方により式 (2.45) を得る。

$$\begin{aligned}
\|x_R(t_{i+1}) - x_M(t_{i+1}|t_i)\| &\leq (\text{Lip}f_x - \text{Lip}f_u \|K\|) \int_{t_i}^{t_{i+1}} \|x_R(t) - x_M(t)\| dt \\
&+ \delta \text{Lip}f_x \tau_{ci}
\end{aligned} \tag{2.45}$$

Gronwall の不等式より、式 (2.46) が成り立つ。

$$\|x_R(t_{i+1}) - x_M(t_{i+1}|t_i)\| \leq \delta \text{Lip}f_x \tau_{ci} \exp((\text{Lip}f_x - \text{Lip}f_u \|K\|) \tau_{ci}) \tag{2.46}$$

式(2.46)において  $\|K\| = 0$  であるならば C- $\pi$  手法における式(2.25)と同じであり,  $\|K\|$  が値をもつならば  $x_R(t)$  と  $x_M(t_{i+1}|t_i)$  の誤差の上限値が C- $\pi$  手法における式(2.25)よりも減少することが分かる.

## 2.5.4 安定性領域の解析

ゲイン  $K$  は設計パラメータであるが, その指針として  $\|K\|$  の値と安定性領域の関係について示す [98]. ある動的システムに対して C- $\pi$  手法を適用した場合に安定となるサンプリングタイムを  $\bar{\tau}$  とする. 同じシステムに対して Gain 型 M · C- $\pi$  手法を用いる場合には, 式(2.47)を満たす  $\|K\|$  及び  $\tau$  であれば同等以上の安定性をもつ.

$$\delta \text{Lip} f_x \tau \exp((\text{Lip} f_x - \text{Lip} f_u \|K\|)\tau) \leq \delta \text{Lip} f_x \bar{\tau} \exp(\text{Lip} f_x \bar{\tau}) \quad (2.47)$$

式(2.47)を整理すると式(2.48)が導かれる.

$$\|K\| \geq \frac{\text{Lip} f_x (\tau - \bar{\tau}) + \log \gamma}{\text{Lip} f_u \tau} \quad (2.48)$$

ただし,  $\tau = \gamma \bar{\tau}$  である.

式(2.48)を  $\|K\|$  と  $\tau$  の関係により概念的に表すと図 2.18 のようになる. 図の赤色の  $\circ$  プロットは  $\bar{\tau}$  を表し, それ以下の  $\tau$  であればゲインは 0 でも安定するが, それ以上の領域ではある一定以上の  $\|K\|$  の値であれば安定することが示されている. ただし, Gain 型では式(2.39)で得られた制御入力の値が制御対象の制限値を超えてしまう場合が考えられる. そのため設定するゲインの値は必ずしも大きくすることはできない.

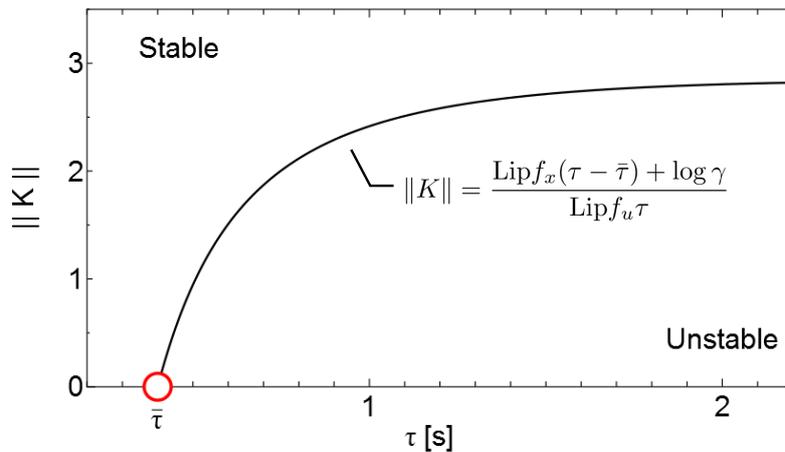


図 2.18 Gain 型 M · C- $\pi$  手法の安定性領域の概念図

## 2.6 MPC型 M・C- $\pi$ 手法と MPC型 M・PC- $\pi$ 手法

### 2.6.1 概要

Gain型による最適状態軌道に対する追従は状態フィードバック制御であるため、制御入力の生成アルゴリズムは単純であり、実装も容易である。一方で、MPCではフィードバック制御入力を最適化により決定するため Gain型よりは複雑な追従制御器となるが、制御対象の拘束条件など性能を考慮することができる利点がある。ここではC- $\pi$ 手法及びPC- $\pi$ 手法の制御系においてMPCにより最適状態軌道に対する追従を行う方法を提案する。本研究では、それぞれの手法をMPC Type Modified Carathéodory- $\pi$ 手法（MPC型 M・C- $\pi$ 手法）及びMPC Type Modified Propagation Carathéodory- $\pi$ 手法（MPC型 M・PC- $\pi$ 手法）と呼ぶ。ここではMPC型 M・C- $\pi$ 手法による制御入力の生成方法について述べる。MPC型 M・PC- $\pi$ 手法については、同様の考え方であるため省略する。

### 2.6.2 制御入力の生成方法

図2.19にMPC型 M・C- $\pi$ 手法における制御系のブロック線図を示す。MPC型 M・C- $\pi$ 手法では、サンプリングタイム $\tau_c$ ごとに最適化計算により最適制御入力 $U^*(t)$ 及び最適状態軌道 $X^*(t)$ を生成するとともに、サンプリングタイム $\tau_m$ ごとにMPCによる最適化も行う。 $U^*(t)$ 及びMPCにより生成したフィードバック制御入力 $U_T(t)$ は、どちらも制御入力の最大値・最小値を取り得るため、それぞれに重み付きで足し合わせた入力を最終的に制御対象へ適用する。

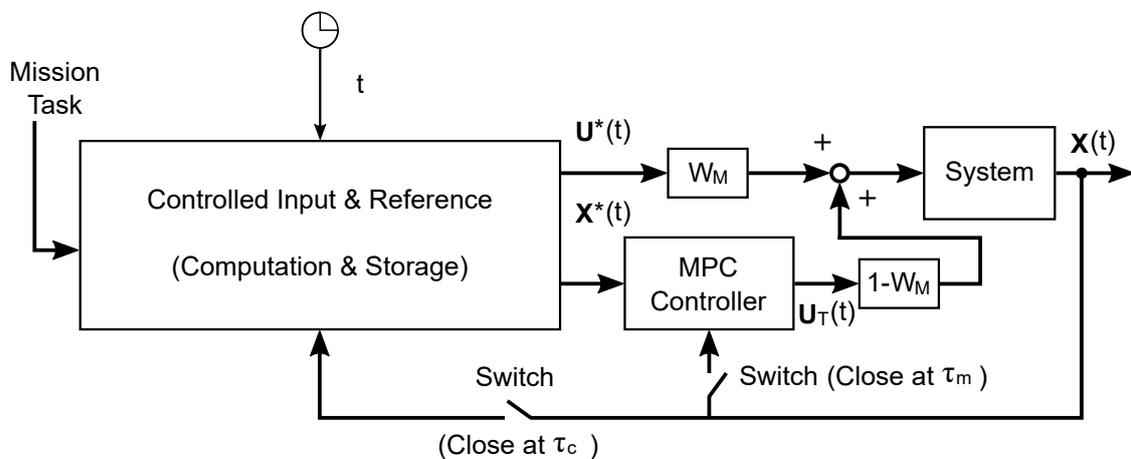


図 2.19 MPC型 M・C- $\pi$ 手法のブロック線図

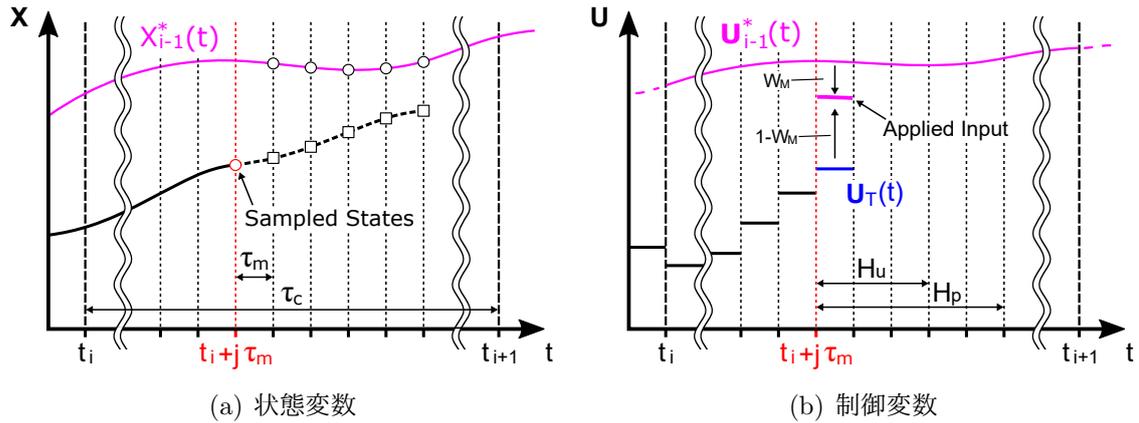


図 2.20 MPC 型 M · C- $\pi$  手法の模式的な時間応答

図 2.20 に MPC 型 M · C- $\pi$  手法の  $t_i \sim t_{i+1}$  における模式的な時間応答を示す。  $t_i \sim t_{i+1}$  では、  $\mathbf{X}_R(t_i)$  を初期条件とした最適化計算を行っているため、直前に生成した最適制御解として  $\mathbf{X}_{i-1}^*(t)$  及び  $\mathbf{U}_{i-1}^*(t)$  が存在する。 MPC 型 M · C- $\pi$  手法では、  $\tau_c$  ごとに C- $\pi$  手法で行う最適化計算に加えて、さらに短いサンプリングタイム  $\tau_m$  ごとに MPC による最適化計算も行う。  $t_i \sim t_{i+1}$  の MPC におけるサンプリング時刻  $t_i + j\tau_m$  にて制御対象の状態をサンプリングし、その時刻から  $H_p$  ステップ間を評価区間とし最適化を行う。 MPC によるフィードバック制御入力  $\mathbf{U}_T(t)$  の具体的な生成手順は、第 2.2 節モデル予測制御 (Model Predictive Control: MPC) にて示しているのここでは省略する。既に得られている制御入力  $\mathbf{U}_{i-1}^*(t)$  と MPC により生成したフィードバック制御入力  $\mathbf{U}_T(t)$  を、重みを用いた式 (2.49) により算出した  $\mathbf{U}_i(t)$  を制御対象に入力する。

$$\mathbf{U}_i(t) = W_M \mathbf{U}_{i-1}^*(t) + (1 - W_M) \mathbf{U}_T(t) \quad (2.49)$$

ただし、  $0 \leq W_M \leq 1$  である。

合成された制御入力  $\mathbf{U}_i(t)$  の具体的な生成方法については、制御対象によって異なるため、後述の 4 章 (MGV)、5 章 (一般車両) にてそれぞれ設計する。

# 第3章 実時間最適制御による運動制御 の検証システム

## 3.1 概要

移動物体の運動制御では，構築した制御アルゴリズムの制御性能は数値シミュレーションや実機により検証が行われる．数値シミュレーションは，構築したアルゴリズムの妥当性の確認や，同じ条件の反復した計算によるパラメータの決定，また様々なパターンなどの検証に有用である．一方，実機を用いる場合は，シミュレーションでは考慮することが困難なモデル誤差や外乱など未知の要素を含めた検証を行うことができる．

本研究ではMGVと一般車両を対象として，数値シミュレーションや実車両を用いた制御アルゴリズムの検証を行う．しかし，一般車両サイズの実車両を用いる場合には，場所，時間，費用，安全管理など様々な制約がある．そこで一般車両に対する検証では，フルビークルモデルのシミュレーションソフトウェアを用いる．使用するソフトウェアは，高自由度のモデリング等による実車両に近い挙動を再現することが可能あり，多くの研究において車両運動の解析や評価に利用されている．

本章では，第2章で記述した実時間最適制御手法を移動物体の運動制御に適用し，その制御性能を検証するためのシステムについて述べる．数値シミュレーションでは，MGVと一般車両の運動計算に用いるソフトウェアは異なるが，同様の計算手順により検証を行う．また実車両を用いた検証はMGVのみである．従って，ここでは(1)数値シミュレーションを用いた検証方法と，(2)実車両を用いた検証方法に分けて説明する．

## 3.2 数値シミュレーションを用いた検証方法

### 3.2.1 システムの構成

図3.1に計算の時間の流れを示す。Optimizationにおいて最適制御解を生成し、データを保存する。保存したデータから必要な時刻の最適制御解を抽出し、Model Simulationにおいて運動計算を行って、計算後の運動状態は保存及び最適化計算に使用される。各ステップの最適化の計算時間は設定したサンプリングタイム以内に終了して最適制御解が得られることを前提とし、以下の手順により数値シミュレーションを行う。

1. 制御開始前にあらかじめオフラインで最適化計算を行い初期解を得る (#1 OPT).  
↓
2. 制御開始後に、初期状態の情報をもとに最適化計算を行う (#2 OPT).  
↓
3. 最適化計算が完了したならば、#1 OPTの最適制御解により運動計算を行って運動状態を算出する。  
↓
4. 得られた運動状態をもとに最適化計算を行う (#3 OPT).  
↓
5. #2 OPTの最適制御解により運動計算を行って運動状態を算出する。  
↓
6. 得られた運動状態をもとに最適化計算を行う (#4 OPT).  
↓
- ...

Optimizationの最適制御解の算出には、ルジャンドル擬スペクトル法を用いる。その際の離散点数は、実車両の実験システムにおいて1秒以内で計算が可能な $N = 21$ に合わせて設定した。運動計算に用いる制御入力値は、最適化計算より得られる離散データをスプライン補間して算出する。各移動物体の検証では表3.1の構成で数値シミュレーションを行った。

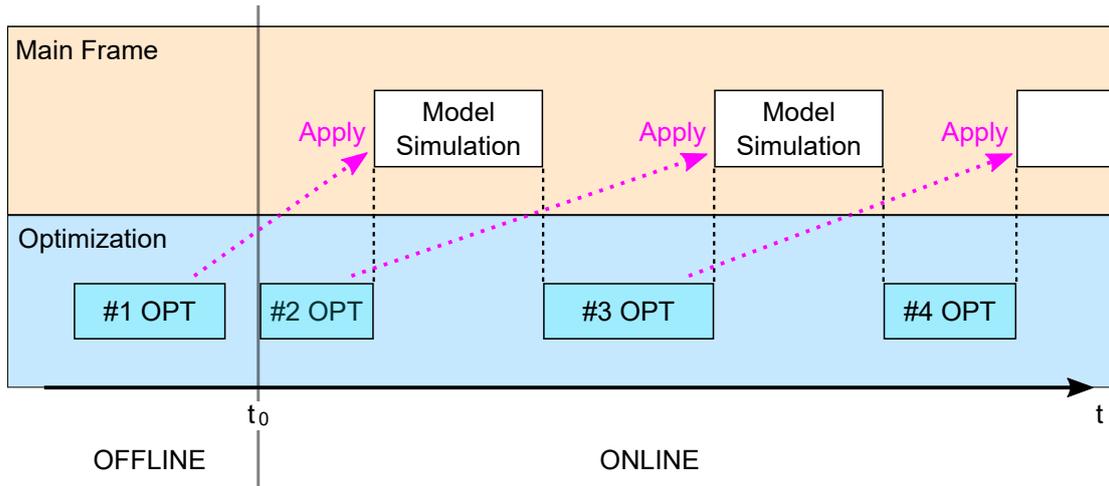


図 3.1 シミュレーションを用いた検証の計算手順

表 3.1 シミュレーションで使用するソフトウェアの構成

	ソフトウェア		
	Main Frame	Optimization	Model Simulation
MGV	MATLAB	MATLAB	Simulink
一般車両	MATLAB	MATLAB	CarSim

### 3.2.2 フルビークルモデルソフトウェア

フルビークルモデルソフトウェアは、高自由度のモデリング等による実車両に近い挙動を再現することが可能であり、検証システムとして使用するには難易度の高い実機の一般車両の代替として、車両運動の解析や評価に用いられている。フルビークルモデルによる運動計算には、実車両ベースのシミュレーションソフトウェアとして多くの研究 [99–102] で用いられている Mechanical Simulation 社製「CarSim」を使用する。本ソフトウェアは27自由度フルビークル動力学モデル及びMATLAB/Simulinkとの接続インターフェースを有する。さらに車両諸元や運転条件、環境条件などの変更も可能である。図3.2に操作画面、図3.3に走行結果のアニメーションの様子を示す。

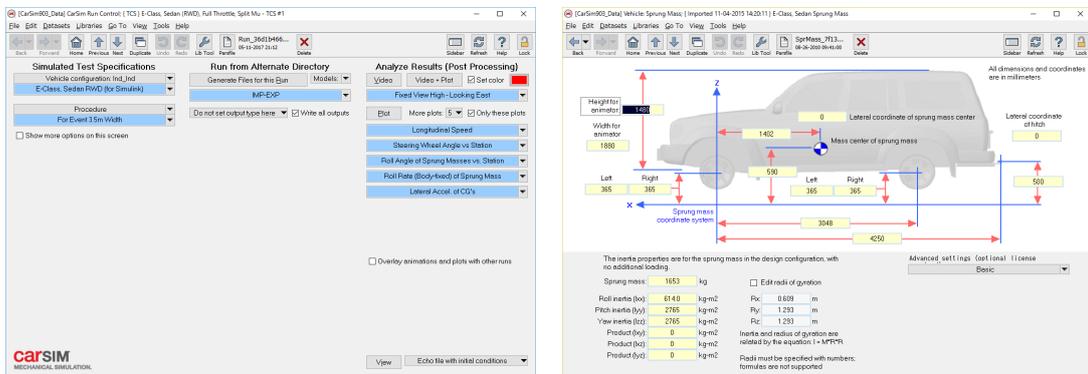


図 3.2 フルビークルモデルソフトウェアの操作画面



図 3.3 フルビークルモデルソフトウェアのアニメーションの一例

### 3.3 実車両を用いた検証方法

#### 3.3.1 システムの構成

図 3.4 に実験システムの構成を示す。構築した制御システムによる MGV の走行結果を高精度に取得し、また外乱を可能な限り排除した理想的な環境において基礎的な検証を行うため、モーション・キャプチャシステムを利用した実験設備において走行実験を行う。

実機の MGV に赤外線反射マーカを 3 つ装着し、赤外線カメラによりそれぞれのマーカの位置座標を取得する (Markers Calibration)。取得データの精度向上や撮影死角の排除のため、7 台のカメラが設置されている。取得データをコントロール PC に送り、位置や速度、姿勢など必要な状態を推定する (State Estimation)。取得・推定した移動物体の状態は走行結果として保存する (Storage) とともに、設計した実時間制御器 (Real-Time Controller) の初期条件などとしても使用する。実時間制御器により移動物体の制御に必要な指令値を生成し、Wi-Fi を介して用いて制御指令を送信する。Storage はデータのアクセスがハードディスクよりも高速な RAM ディスクを用いた。

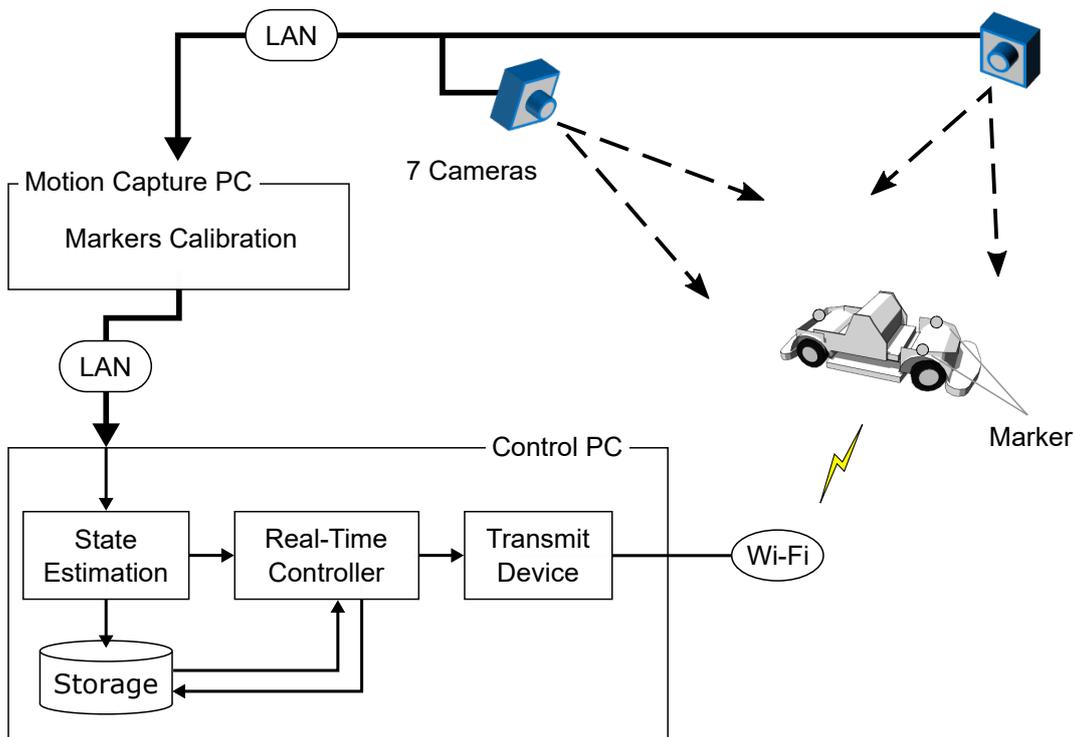


図 3.4 実験システムの構成

図 3.5 に実車両を用いた検証における，最適制御解の更新及び適用の流れを示す．数値シミュレーションとは異なり，実車両を用いた検証では最適化計算と制御入力の実用を交互に行うことはできない．そのため，MATLAB で動作する Main Frame と，C 言語化により最適制御解の高速演算が可能な Optimization を並列に実行し，Storage にてデータを媒介することで実時間最適制御による実車両の走行を実現する．Main Frame と Optimization ではそれぞれの一定の時間間隔で Storage を読み込む．Main Frame では，最も新しい最適制御解を読み込んで制御指令を MGV に入力し，またサンプリングタイムごとに MGV の状態を保存する．Optimization では，サンプリングタイムごとに更新された MGV の状態を用いて最適化計算を行い，生成した最適制御解を Storage に保存する．

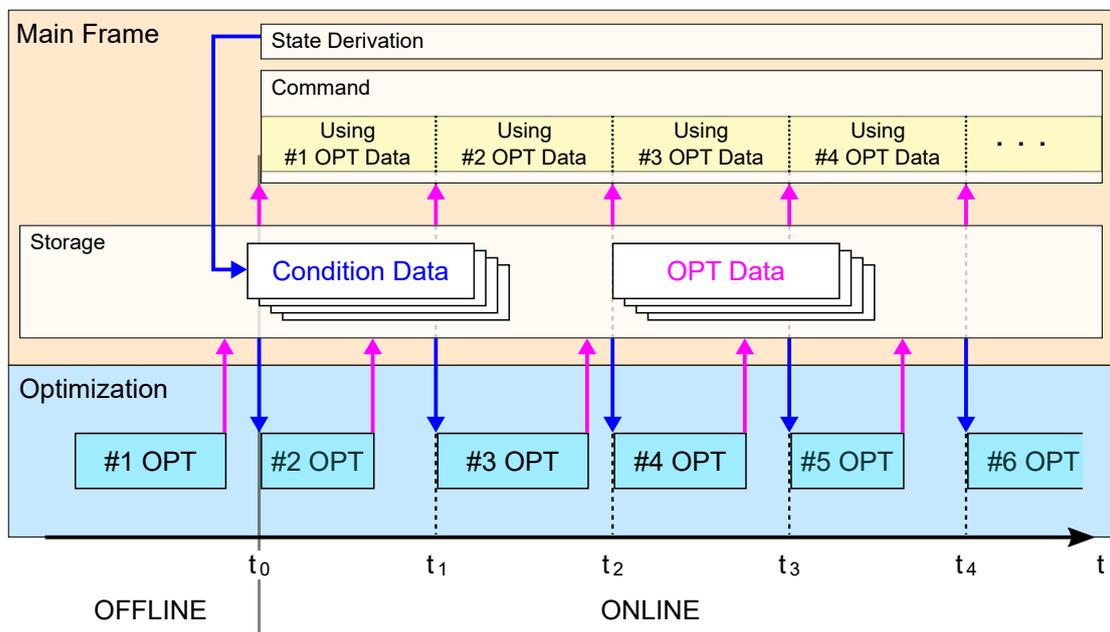


図 3.5 実車両を用いた検証における最適制御解の更新及び適用の流れ

### 3.3.2 実験車両

実験車両として、図3.6に示すZMP社製「RoboCar<sup>®</sup> 1/10」を使用する。このMGVは新しいカー・ロボティクス分野における研究開発のためのプラットフォームとして開発され、広い分野の研究に対し提供されている。RoboCarの特徴として、1/10スケール自動車型であることや、本格的な駆動系、サスペンション、操舵系、無線通信によりPCとの連携ができることなどがある。モーション・キャプチャシステムを使用して車両の位置座標を計測するため、1/10スケールRCカーの未塗装の外装を装着し、3点の白色赤外線反射マーカを取り付けている。

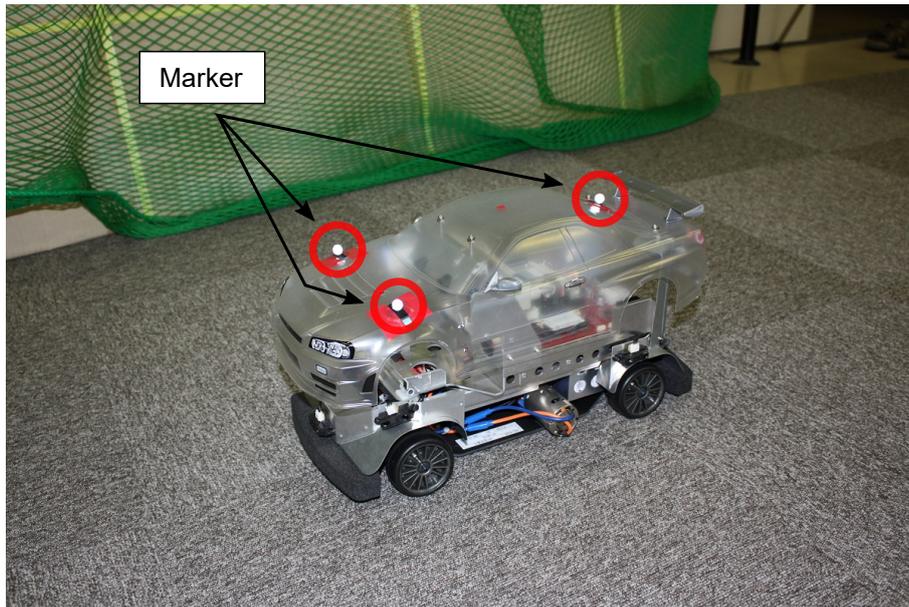


図 3.6 実験車両の外観

### 3.3.3 計測システム

3次元空間内の物体の計測には、リアルタイム光学式モーション・キャプチャシステム（Motion Analysis社製 MAC 3D System）を用いる。図3.7にキャプチャ画面を示す。7台設置した赤外線カメラにより、3.0 m (W) × 3.0 m (L) × 1.8 m (H) の測定空間において、対象物に取り付けた赤外線反射マーカの3次元位置を検知することができる。マーカの位置座標  $(x, y, z)$  の検出精度は、1 mm 以下であり、サンプリング周波数は最大 250 Hz である。また実験区域は、図3.8のように、床面にはカーペットを使用し、より平坦かつ光の反射によるモーション・キャプチャシステムの誤検知防止と、タイヤの滑りも抑制している。

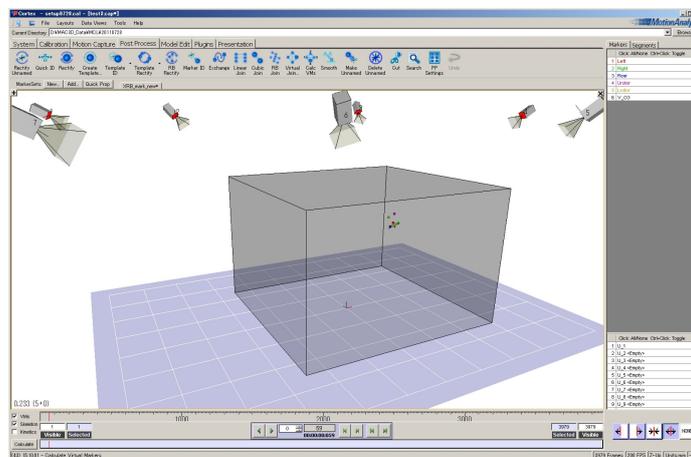


図 3.7 キャプチャ画面



図 3.8 実験区域

# 第4章 Clock-Based 実時間最適制御によるMGVの運動制御

## 4.1 概要

小型無人車両 (Micro Ground Vehicle: MGV) は、偵察活動や災害活動などの緊急性の高い任務に投入されることが想定され、より効率的な移動が求められる。効率的な移動を達成する手段としては最適制御理論の活用があるが、モデル化誤差による安定性の問題や移動開始前には予期できない障害物の発生など、変化する状況にも対応する必要がある。

そのような問題に対応しつつ、効率的な移動を実現する方法としては、移動間にも最適制御問題を繰り返し解き続ける実時間最適制御がある。しかし、従来の最適化計算時間は移動物体の制御時間に対して長く、最適化計算の高速化を図る必要があったため、サンプリング間の制御入力を一定値、最適制御問題における評価区間を有限として高速な最適化計算を実現するモデル予測制御 (MPC) がMGVの運動制御にもよく用いられてきた。

一方で、近年ではコンピュータの演算能力や最適化アルゴリズムの向上に伴い、最適化計算に要する時間は短くなってきている。そのため、終端時間を自由とし、さらにサンプリング間の入力も時刻参照として考える Clock-Based 実時間最適制御を用いたMGVの運動制御が可能となりつつあり、より効率的な移動が期待される。しかし、Clock-Based 実時間最適制御をMGVに適用した研究は数値シミュレーションによる検証のみであり、また安定性が保証されたサンプリング間隔内に最適化計算が終了する条件で設定されている。実機を用いた検証により Clock-Based 実時間最適制御のMGVへの適用可能性を示し、さらにサンプリングタイムが長くなる場合においても安定性を向上させる必要がある。

本章では、第2章で記述した制御アルゴリズムをMGVの運動制御に適用する。まずMGVの運動モデルと制御系を設計する。続いて数値シミュレーション及び実機を用いた検証を行い、Clock-Based 実時間最適制御のMGVへの適用可能性と提案手法の有用性を示す。

## 4.2 運動モデルの構築とパラメータ同定

実時間最適制御では最適制御問題の最適化計算を逐次行う必要がある。その際には厳密性と計算速度のトレードオフを考えた適切な運動モデルを用いることが望まれる。先行研究 [103] において、Dubins モデル [104] や Reeds-Shepp モデル [105] を基にした複数のモデルを構築し、最適化計算に用いる妥当なモデルについて数値シミュレーションにより比較・検証を行った。そこで本研究では詳細なモデルの導出は省略し、先行研究で選定した運動モデルを使用する。

図 4.1 に本研究で用いる MGV の 2 次元直交座標系を示す。状態変数は式 (4.1)、制御変数は式 (4.2) とする。

$$\mathbf{X}(t) = [x(t) \ y(t) \ \psi(t) \ V(t) \ \delta_f(t) \ a_x(t)]^T \quad (4.1)$$

$$\mathbf{U}(t) = [V_c(t) \ \delta_{fc}(t)]^T \quad (4.2)$$

本研究で使用する MGV システムは速度及び操舵角の指令値を与えることで操作することができる。計算負荷を軽減させるため、速度指令に対する実際の速度の応答を 2 次遅れ系、操舵角指令に対する操舵角の応答を 1 次遅れ系と仮定した簡易なモデルを用いる。式 (4.3) に運動方程式を示す。ただし、 $\Lambda(\delta_f)$  はステアリング特性を考慮した補正パラメータであり、式 (4.4) で表される。

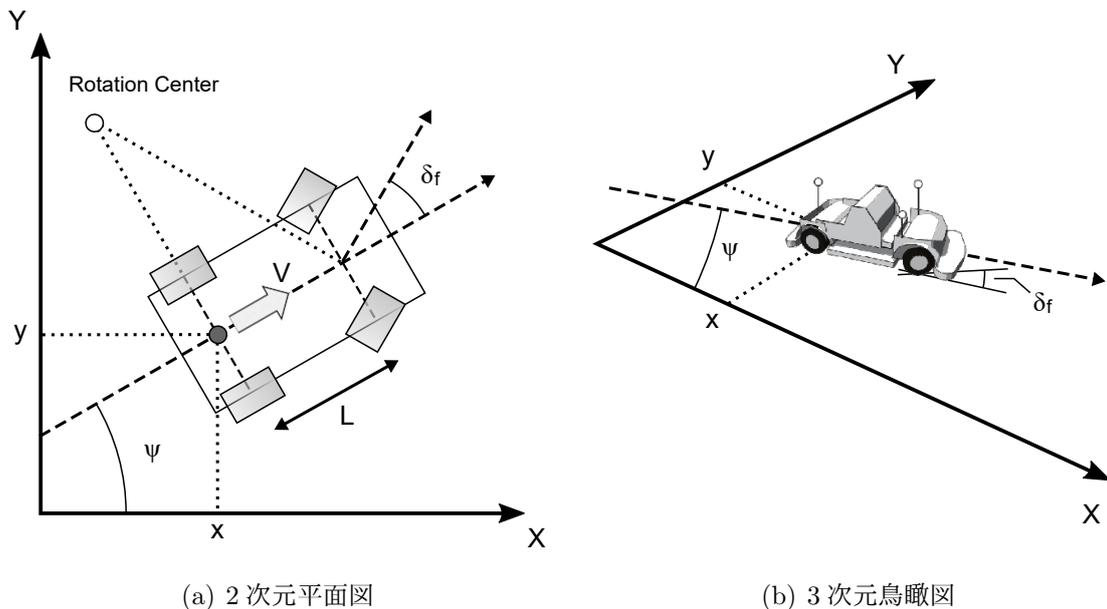


図 4.1 MGV の座標系

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{V} \\ \dot{\delta}_f \\ \dot{a}_x \end{bmatrix} = \begin{bmatrix} V \cos \psi \\ V \sin \psi \\ \frac{V}{L + \Lambda(\delta_f)} \tan \delta_f \\ a_x \\ \frac{1}{\tau_d} (\delta_{fc} - \delta_f) \\ -\omega_n^2 V - 2\zeta\omega_n a_x + K_V \omega_n^2 V_c \end{bmatrix} \quad (4.3)$$

$$\Lambda(\delta_f) = A \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} \delta_f^2\right) \quad (4.4)$$

先行研究 [103] において，式 (4.3) 及び (4.4) で用いられている  $K$ ， $\zeta$ ， $\omega_n$ ， $\tau_d$ ， $A$ ，及び  $\sigma$  の各値を，実験データによりパラメータ同定を行っている．速度指令に対する速度応答の同定は，速度指令値を通常のスレップ入力や多段階のスレップ入力により得られた応答を用いた．操舵角指令に対する操舵角応答の同定は，一定の操舵角指令値による定常円旋回のデータを用いた．表 4.1 にパラメータ同定により得られた値を示す．

表 4.1 MGV の運動モデルで用いるパラメータの値 [103]

$K_V$ [-]	$\zeta$ [-]	$\omega_n$ [rad/s]	$\tau_d$ [s]	$A$ [m/deg]	$\sigma$ [-]
0.94	0.20	9.42	0.1	22.00	2.80

## 4.3 制御系の設計

### 4.3.1 C- $\pi$ 手法における最適制御問題の定式化

p. 31 の式 (2.30) で示した時刻  $t_i \sim t_{i+1}$  における最適制御問題  $\mathbf{P}_c$  の定式化を行う．最初に拘束条件について与えて，続いて評価関数を設定する．

式 (2.27) の運動モデルには前節で構築した式 (4.3) を用いる．式 (2.28) の境界条件は式 (4.5) で与える．

$$\begin{bmatrix} x_i \\ y_i \\ \psi_i \\ V_i \\ \delta_{fi} \\ a_{xi} \\ x_f \\ y_f \\ \psi_f \\ V_f \\ \delta_{ff} \\ a_{xf} \end{bmatrix} \leq \begin{bmatrix} x(t_i) \\ y(t_i) \\ \psi(t_i) \\ V(t_i) \\ \delta_f(t_i) \\ a_x(t_i) \\ x(t_f) \\ y(t_f) \\ \psi(t_f) \\ V(t_f) \\ \delta_f(t_f) \\ a_x(t_f) \end{bmatrix} \leq \begin{bmatrix} x_i \\ y_i \\ \psi_i \\ V_i \\ \delta_{fi} \\ a_{xi} \\ x_f \\ y_f \\ \psi_f \\ V_f \\ \delta_{ff} \\ a_{xf} \end{bmatrix} \quad (4.5)$$

ここで  $x_i, y_i, \psi_i, V_i, \delta_{fi}, a_{xi}$  はサンプリングした状態， $x_f, y_f, \psi_f, V_f, \delta_{ff}, a_{xf}$  は目標状態を表す．

式 (2.29) の軌道拘束条件には本研究で使用する MGV システムの機械的制約や測定領域を考慮して，式 (4.6)～(4.7) の拘束条件を与える．

$$V_{minFF} \leq V_c(t) \leq V_{maxFF} \quad (4.6)$$

$$\delta_{fminFF} \leq \delta_{fc}(t) \leq \delta_{fmaxFF} \quad (4.7)$$

評価関数は式 (4.8) のボルザ型で与える．終端時刻  $t_f$  とサンプリングした時刻  $t_i$  の差を最小化することによる最短時間移動を基本として，積分項の第 1 項にて急激な操舵角の変化を抑えている．また障害物回避では，積分項の第 2 項まで考え，式 (4.9) 及び (4.10) の障害物を回避する軌道を生成するための関数を与える．図 4.2 のように障害物付近で  $\xi(t)$  の値が急増するため回避軌道が生成される．

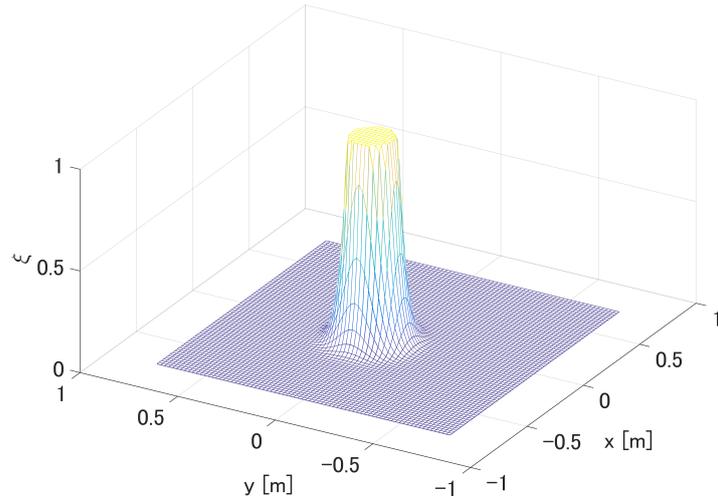


図 4.2 障害物関数の値の変化（一例）

$$J_c = t_f - t_i + \int_{t_i}^{t_f} \left( W_{ddf} \dot{\delta}_f^2(t) + \xi(t) \right) dt \quad (4.8)$$

$$\xi(t) = e^{(-W_{obs} h_{obs}(x(t), y(t)))} - 1 \quad (4.9)$$

$$h_{obs}(x(t), y(t)) = (x - x_{obs})^{p_{obs}} + (y - y_{obs})^{p_{obs}} - r_{obs}^{p_{obs}} \quad (4.10)$$

以上の設定により，MGV の実時間最適制御で用いる最適制御問題  $\mathbf{P}_c$  は以下の通り定式化される．

$$(\mathbf{P}_c) \left\{ \begin{array}{ll} \text{評価関数} & \text{式 (4.8)} \\ \text{運動モデル} & \text{式 (4.3)} \\ \text{境界条件} & \text{式 (4.5)} \\ \text{軌道拘束条件} & \text{式 (4.6)~(4.7)} \end{array} \right.$$

定式化した最適制御問題  $\mathbf{P}_c$  から擬スペクトル法を用いた最適化計算ソフトウェア [106]

により式 (4.11) 及び (4.12) の最適状態軌道  $\mathbf{X}_i^*(t)$  及び最適制御入力  $\mathbf{U}_i^*(t)$  を導出する.

$$\mathbf{X}_i^*(t) = [ x_i^*(t) \ y_i^*(t) \ \psi_i^*(t) \ V_i^*(t) \ \delta_{fi}^*(t) \ a_{xi}^*(t) ]^T \quad (4.11)$$

$$\mathbf{U}_i^*(t) = [ V_{ci}^*(t) \ \delta_{fci}^*(t) ]^T \quad (4.12)$$

得られた式 (4.12) の最適制御入力は, 式 (4.13) 及び (4.14) のフィードフォワード制御入力として設定する. C- $\pi$  手法では,  $V_{cFF}$  及び  $\delta_{fcFF}$  が MGV に制御指令として与えられる.

$$V_{cFF} = V_{ci}^*(t) \quad (4.13)$$

$$\delta_{fcFF} = \delta_{fci}^*(t) \quad (4.14)$$

### 4.3.2 Gain型 M・C- $\pi$ 手法の制御系設計

Gain型 M・C- $\pi$  手法のMGVへの適用は、図4.3に示す前方誤差修正モデルに基づいて追従制御器のフィードバック制御入力を生成する。ある時刻においてMGVの位置が $(x_0, y_0)$ であるとき、式(4.15)及び(4.16)により予測時間 $T_p$ 秒後のMGVの位置を算出する。

$$x_p = x_0 + \dot{x}_0 T_p + \frac{1}{2} \ddot{x}_0 T_p^2 \quad (4.15)$$

$$y_p = y_0 + \dot{y}_0 T_p + \frac{1}{2} \ddot{y}_0 T_p^2 \quad (4.16)$$

式(4.15)及び(4.16)の右辺第2項までを考えた場合は1次予測モデル[107]、右辺第3項までを考えた場合は2次予測モデル[108]となる。

算出した $(x_p, y_p)$ と $T_p$ 秒後の目標位置 $(x_c, y_c)$ の誤差は、回転行列を用いると式(4.17)の車体固定座標系における前後方向の誤差 $\epsilon_x$ 及び横方向の誤差 $\epsilon_y$ に変換できる。

$$\begin{bmatrix} \epsilon_x \\ \epsilon_y \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} x_c - x_p \\ y_c - y_p \end{bmatrix} \quad (4.17)$$

$T_p$ 秒後の目標位置 $(x_c, y_c)$ は最適化計算により目標軌道が時間の関数 $(x^*(t), y^*(t))$ で

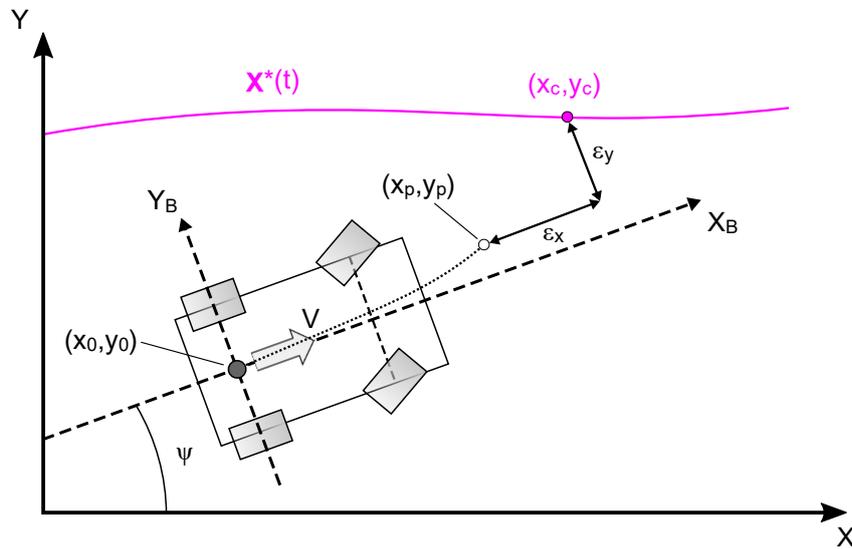


図 4.3 MGV の運動制御における前方誤差修正モデルによるフィードバック制御入力の生成

与えられているので，式 (4.18) 及び (4.19) となる．

$$x_c = x^*(t + T_p) \quad (4.18)$$

$$y_c = y^*(t + T_p) \quad (4.19)$$

速度指令及び操舵角指令のフィードバック制御入力値は，前後方向誤差及び横方向誤差を補償する式 (4.20) 及び (4.21) の比例制御で与える．ここで  $K_{PV}$  及び  $K_{P\delta}$  はそれぞれのフィードバックゲインである．

$$V_{cFB} = K_{PV}\epsilon_x \quad (4.20)$$

$$\delta_{fcFB} = K_{P\delta}\epsilon_y \quad (4.21)$$

よって Gain 型 M・C- $\pi$  手法では，フィードフォワード制御入力とフィードバック制御入力を合わせた式 (4.22) 及び (4.23) で得られる  $V_c$  及び  $\delta_{fc}$  を MGV へ入力する．

$$V_c = V_{cFF} + V_{cFB} \quad (4.22)$$

$$\delta_{fc} = \delta_{fcFF} + \delta_{fcFB} \quad (4.23)$$

ただし，目標との誤差が大きいとフィードバック制御入力の値が大きくなる可能性があるので，式 (4.24) 及び (4.25) により入力値に制限を与える．

$$V_{cmin} \leq V_c(t) \leq V_{cmax} \quad (4.24)$$

$$\delta_{fcmin} \leq \delta_{fc}(t) \leq \delta_{fcmax} \quad (4.25)$$

なお，Gain 型 M・PC- $\pi$  手法の制御系設計については同様であるため，本論文では省略する．

### 4.3.3 MPC 型 M・C- $\pi$ 手法の制御系設計

図 4.4 に時刻  $t_i + \tau_m j$  における MPC 型 M・C- $\pi$  手法を用いた追従制御器の、フィードバック制御入力の生成の概要を示す。式 (2.5) のパラメータ最適化問題  $\mathbf{P}_m$  の定式化を行う。式 (4.26) を初期値とし、式 (4.3) の運動モデルを用いて式 (4.27) の評価関数を最小化する最適制御入力  $\hat{V}_c$  及び  $\hat{\delta}_{fc}$  を求める。評価関数は、 $H_p$  ステップ後の  $x$ ,  $y$ ,  $\psi$ ,  $V$  の目標値との誤差を評価する。右辺第 5 項及び第 6 項では、生成される各制御入力の一つ前の値と比べて大きく変化することを抑制する効果がある。

$$\mathbf{X}_R(t_i + \tau_m j) = \begin{bmatrix} x(t_i + \tau_m j) \\ y(t_i + \tau_m j) \\ \psi(t_i + \tau_m j) \\ V(t_i + \tau_m j) \\ \delta_f(t_i + \tau_m j) \\ a_x(t_i + \tau_m j) \end{bmatrix} \quad (4.26)$$

$$\begin{aligned} J_m = & W_x(x^*(t_i + \tau_m(j + H_p)) - x(t_i + \tau_m(j + H_p)))^2 \\ & + W_y(y^*(t_i + \tau_m(j + H_p)) - y(t_i + \tau_m(j + H_p)))^2 \\ & + W_\psi(\psi^*(t_i + \tau_m(j + H_p)) - \psi(t_i + \tau_m(j + H_p)))^2 \\ & + W_V(V^*(t_i + \tau_m(j + H_p)) - V(t_i + \tau_m(j + H_p)))^2 \\ & + W_{Vlast}(\hat{V}_c(t_i + \tau_m j) - \hat{V}_c(t_i + \tau_m(j - 1)))^2 \\ & + W_{\delta last}(\hat{\delta}_{fc}(t_i + \tau_m j) - \hat{\delta}_{fc}(t_i + \tau_m(j - 1)))^2 \end{aligned} \quad (4.27)$$

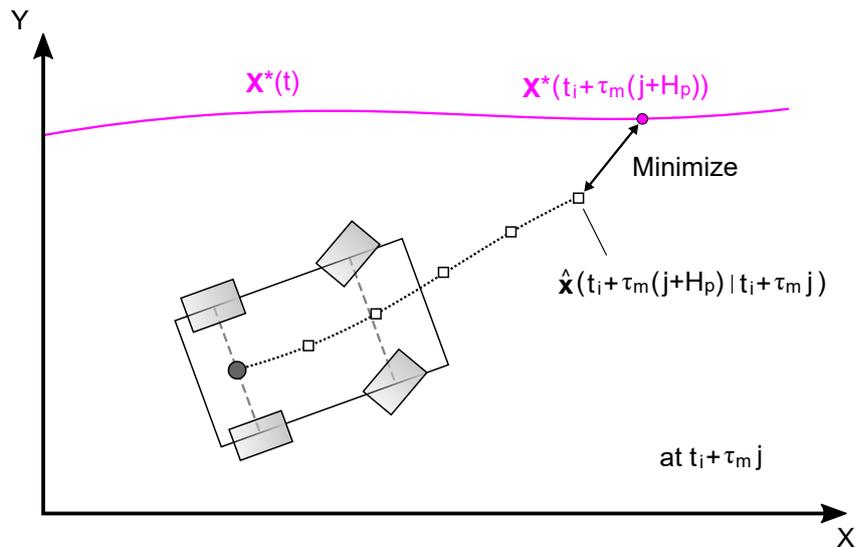


図 4.4 MGV の運動制御における MPC によるフィードバック制御入力の生成

MPC は  $H_u$  ステップ以降の入力値は一定であるので、式 (4.28) 及び (4.29) とする.

$$\hat{V}_c(t_i + \tau_m(j + H_u)|t_i + \tau_m j) = \cdots = \hat{V}_c(t_i + \tau_m(j + H_p)|t_i + \tau_m j) \quad (4.28)$$

$$\hat{\delta}_{fc}(t_i + \tau_m(j + H_u)|t_i + \tau_m j) = \cdots = \hat{\delta}_{fc}(t_i + \tau_m(j + H_p)|t_i + \tau_m j) \quad (4.29)$$

また式 (4.30) 及び (4.31) により制御変数を制限する.

$$V_{cmin} \leq \hat{V}_c \leq V_{cmax} \quad (4.30)$$

$$\delta_{fcmin} \leq \hat{\delta}_{fc} \leq \delta_{fcmax} \quad (4.31)$$

以上の設定により、MGV の実時間最適制御で用いる最適制御問題  $\mathbf{P}_m$  は以下の通り定式化される.

$$(\mathbf{P}_m) \left\{ \begin{array}{ll} \text{評価関数} & \text{式 (4.27)} \\ \text{運動モデル} & \text{式 (4.3)} \\ \text{境界条件} & \text{式 (4.26)} \\ \text{軌道拘束条件} & \text{式 (4.30)~(4.31)} \end{array} \right.$$

定式化した非線形計画問題である最適制御問題  $\mathbf{P}_m$  を SQP で解を得て、最適制御入力  $\hat{V}_c$  及び  $\hat{\delta}_{fc}$  を決定する. 導出した  $\hat{V}_c$  及び  $\hat{\delta}_{fc}$  を式 (4.32) 及び (4.33) のフィードバック制御入力として定め、式 (4.34) 及び (4.35) の重み付き合成で得られる  $V_c$  及び  $\delta_{fc}$  を MGV へ入力する.

$$V_{cFB} = \hat{V}_c(t_i + \tau_m j) \quad (4.32)$$

$$\delta_{fcFB} = \hat{\delta}_{fc}(t_i + \tau_m j) \quad (4.33)$$

$$V_c = W_M V_{cFF} + (1 - W_M) V_{cFB} \quad (4.34)$$

$$\delta_{fc} = W_M \delta_{fcFF} + (1 - W_M) \delta_{fcFB} \quad (4.35)$$

ただし、 $W_M$  は合成の割合を調整する重みであり、 $0 \leq W_M \leq 1$  を満たす. なお、MPC 型 M・PC- $\pi$  手法の制御系設計については同様であるため、本論文では省略する.

#### 4.3.4 MGV の運動制御で使用するパラメータの値（共通）

表 4.2 にサンプリングタイムごとに解く最適制御問題の拘束条件として与えているパラメータの値を示す。速度及び速度指令の制約は、室内の限られた空間での走行実験を行うため、安全面を考慮して与えている。この際、正方形の測定領域の対角線方向に移動する際に最適解の更新回数が 10 回程度以上となる様に制限速度を設定した。操舵角及び操舵角指令の制約は、使用する MGV の入力可能な最大操舵角が左右 30 deg であるため、FF 制御入力を生成する最適制御問題では余裕を持たせた左右 25 deg とし、FB 制御による補正を考慮した最終的に入力する指令値の限界を左右 30 deg とした。

表 4.2 MGV の運動制御で使用するパラメータの値（拘束条件）

$V_{minFF}$ [m/s]	$V_{maxFF}$ [m/s]	$V_{cmin}$ [m/s]	$V_{cmax}$ [m/s]
-0.2	0.2	-0.2	0.2
$\delta_{fminFF}$ [deg]	$\delta_{fmaxFF}$ [deg]	$\delta_{fcmin}$ [deg]	$\delta_{fcmax}$ [deg]
-25	25	-30	30

表 4.3 に Gain 型の追従制御器に使用したパラメータの値を示す。Gain 型の追従制御器では、一般的な前方誤差修正モデルを用いた追従制御と異なり、サンプリングタイムごとの最適化計算により既に基本となる FF 制御入力が得られている。そのため、ここでは FF 制御入力を補正する範囲での追従を目的として予測時間やゲインの値を決定する。先行研究 [103] では、それぞれのパラメータを変更した FF 制御と FB 制御の 2 自由度制御系によるオフライン最適状態軌道への追従制御実験を行って、最も結果が良かった値について採用している。本章の Gain 型追従制御器のパラメータには同じ値を使用している。

表 4.3 MGV の運動制御で使用するパラメータの値（Gain 型） [103]

$T_p$ [s]	$K_{PV}$ [1/s]	$K_{P\delta}$ [rad/m]
1.0	0.1	2.0

表 4.4 に MPC 型の追従制御器に使用したパラメータの値を示す。MPC は、高速に最適化計算を繰り返し行う手法であり、サンプリングタイムを 0.02 秒程度に設定している研究もあるが、本研究の実験システムでは最適化計算時間に加えて最適解の書き込み及び読み込みを行うため、それらを実行できる十分な時間として 0.2 秒に設定した。 $H_p$  は 0.5 秒から 1.0 秒先までを評価区間とすることを目的として 3 から 5 の範囲

で、それに伴って  $H_u$  は 2 から 4 の範囲で検討した。MPC により生成した FB 制御入力  
 は、FF 制御入力と重み付きで足し合わせるが、Gain 型の追従制御器と同様に、FF  
 制御入力を補正する範囲での追従を目的として、 $W_M$  は 0.8 とした。以上の  $\tau_m$ ,  $H_p$ ,  
 $H_u$ ,  $W_M$  の範囲のもと評価関数のそれぞれの重みを調整したオフライン最適状態軌道  
 への追従制御実験を行い、各パラメータの値を決定した。

表 4.4 MGV の運動制御で使用するパラメータの値 (MPC 型)

$\tau_m$ [s]	$H_p$ [-]	$H_u$ [-]	$W_M$ [-]
0.2	3	2	0.8

$W_x$ [-]	$W_y$ [-]	$W_\psi$ [-]	$W_V$ [-]	$W_{Vlast}$ [-]	$W_{\delta last}$ [-]
100	100	10	10	0.1	0.1

## 4.4 数値シミュレーション

### 4.4.1 初期状態誤差に対する影響

#### ・想定及びシミュレーション条件

Clock-Based 実時間最適制御では、走行途中に予測していた状態から逸脱した場合においても、新たな最適軌道を生成することで目標まで移動を行う。そのため、ここでは図 4.5 に示すように、走行開始前に行った最適化計算の条件とは異なる初期状態から移動を開始させ、その誤差に対して新たな軌道を生成することで目標まで移動できるかを検証する。また第 2 章で示した通り、Clock-Based 実時間最適制御に対してサンプリングタイム  $\tau_c$  の長さは安定性へ大きく影響する。よって  $\tau_c$  を変化させたことによる走行結果への影響についても検証する。ただし、いずれの場合でも移動間の最適制御解を更新する回数を同程度に保持するため、初期誤差がない場合での開始位置と目標位置の距離  $L_p$  を  $\tau_c$  で正規化する。ここでは、 $L_p/\tau_c = 2$  とし、約 10 回程度の更新を行う。各手法の制御性能については、2 点間距離をもとに正規化した走行軌跡と、各サンプリング時刻における実軌道と予測軌道の誤差を用いて評価を行う。数値シミュレーションは、目標位置から 0.15 m 以内かつ目標ヨー角との誤差が 10 deg 以内を満たすか、オフライン解の終端時刻から 3 秒超えた場合に終了する。表 4.5 にシミュレーションの条件をまとめる。

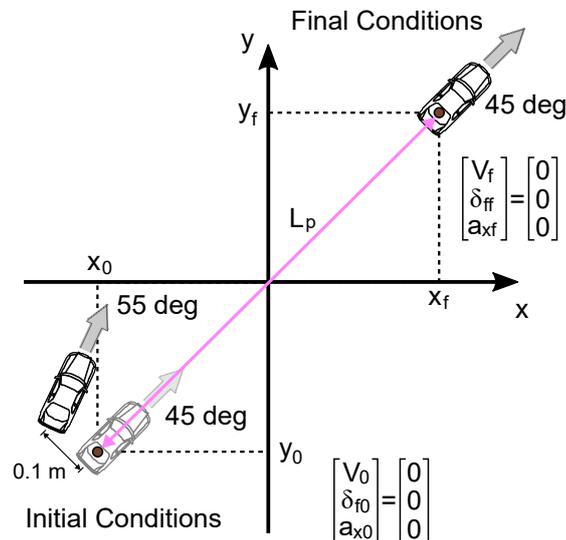


図 4.5 想定の概要（初期状態誤差）

表 4.5 シミュレーション条件（初期状態誤差）

		$\tau_c$ [s]	$L_p/\tau_c$ [m/s]			
		0.5, 1.0, $\dots$ , 2.5	2			
$x_0$ [m]	$y_0$ [m]	$\psi_0$ [deg]	$V_0$ [m/s]	$\delta_{f0}$ [deg]	$a_{x0}$ [m/s <sup>2</sup> ]	
$-(L_p/2\sqrt{2})$	$-(L_p/2\sqrt{2})$	45	0	0	0	
$x_f$ [m]	$y_f$ [m]	$\psi_f$ [deg]	$V_f$ [m/s]	$\delta_{ff}$ [deg]	$a_{xf}$ [m/s <sup>2</sup> ]	
$L_p/2\sqrt{2}$	$L_p/2\sqrt{2}$	45	0	0	0	

・数値シミュレーション結果

図 4.6 に、終端  $x$  座標  $x_f$  で  $x$  軸及び  $y$  軸を正規化した走行軌跡を示す。黒色の破線はオフラインで得られた最適状態軌道，それぞれの色の実線が走行軌跡を表している。 $\tau_c = 0.5$  s の場合では，いずれの手法においても初期状態誤差を修正し目標まで移動している。通常のフィードバック制御であればオフライン軌道に戻るよう右へ旋回し，以後は軌道に沿って目標まで移動するが，Clock-Based 実時間最適制御では目標までの新たな軌道を再生成するため直線的に目標まで移動している。しかし，C- $\pi$  手法を用いた場合では  $\tau_c$  が長くなるにつれて大きく左へ迂回している。MPC 型 M・C- $\pi$  手法を用いた場合でも他手法と比較すると左への迂回がみられるが，Gain 型や MPC 型の M・C- $\pi$  手法を用いた場合では C- $\pi$  手法における左への迂回が軽減されている。一方，PC- $\pi$  手法を用いた場合では C- $\pi$  手法で発生した左への迂回は見られず，Gain 型や MPC 型の M・PC- $\pi$  手法を用いた場合でも同等の軌道が得られている。

図 4.7 に，それぞれの走行における各サンプリング時刻での実軌道と予測軌道の誤差を整理した結果を示す。○プロットは誤差の平均値を表し， $x$  方向， $y$  方向， $\psi$  方向に分けてまとめている。なお，結果の重複を避けて見やすくするため，C- $\pi$  手法及び PC- $\pi$  手法は負の方向，MPC 型の M・C- $\pi$  手法及び M・PC- $\pi$  手法は正の方向に横軸をずらしている。それぞれの誤差の平均値  $E_x$ ， $E_y$ ， $E_\psi$  は式 (4.36)～(4.38) にて算出する。

$$E_x = E \left( \sum_{i=1}^{N_s} \|x_R(t_i) - x_{i-1}^*(t_i)\| \right) \quad (4.36)$$

$$E_y = E \left( \sum_{i=1}^{N_s} \|y_R(t_i) - y_{i-1}^*(t_i)\| \right) \quad (4.37)$$

$$E_\psi = E \left( \sum_{i=1}^{N_s} \|\psi_R(t_i) - \psi_{i-1}^*(t_i)\| \right) \quad (4.38)$$

ここで、 $E$ は平均、 $N_s$ はサンプリングを行った回数を表す。また $i = 1$ の場合の $x_0^*(t)$ ,  $y_0^*(t)$ ,  $\psi_0^*(t)$ は走行開始時 $t = 0$  sにおいて生成する最適状態軌道となる。

C- $\pi$ 手法を用いた場合では $\tau_c$ が長くなるにつれて誤差が増大し走行が不安定になる傾向があるが、Gain型やMPC型のM・C- $\pi$ 手法を用いた場合では誤差の増大を抑制することができる。特にGain型M・C- $\pi$ 手法ではよく抑えることができる。一方、PC- $\pi$ 手法を用いた場合では $\tau_c$ が長くなることに伴う誤差の増大する傾向はみられない。PC- $\pi$ 手法やGain型M・PC- $\pi$ 手法と比較するとMPC型M・PC- $\pi$ 手法を用いた場合に若干の誤差の増加があるが、 $\tau_c$ の変化による影響はない。

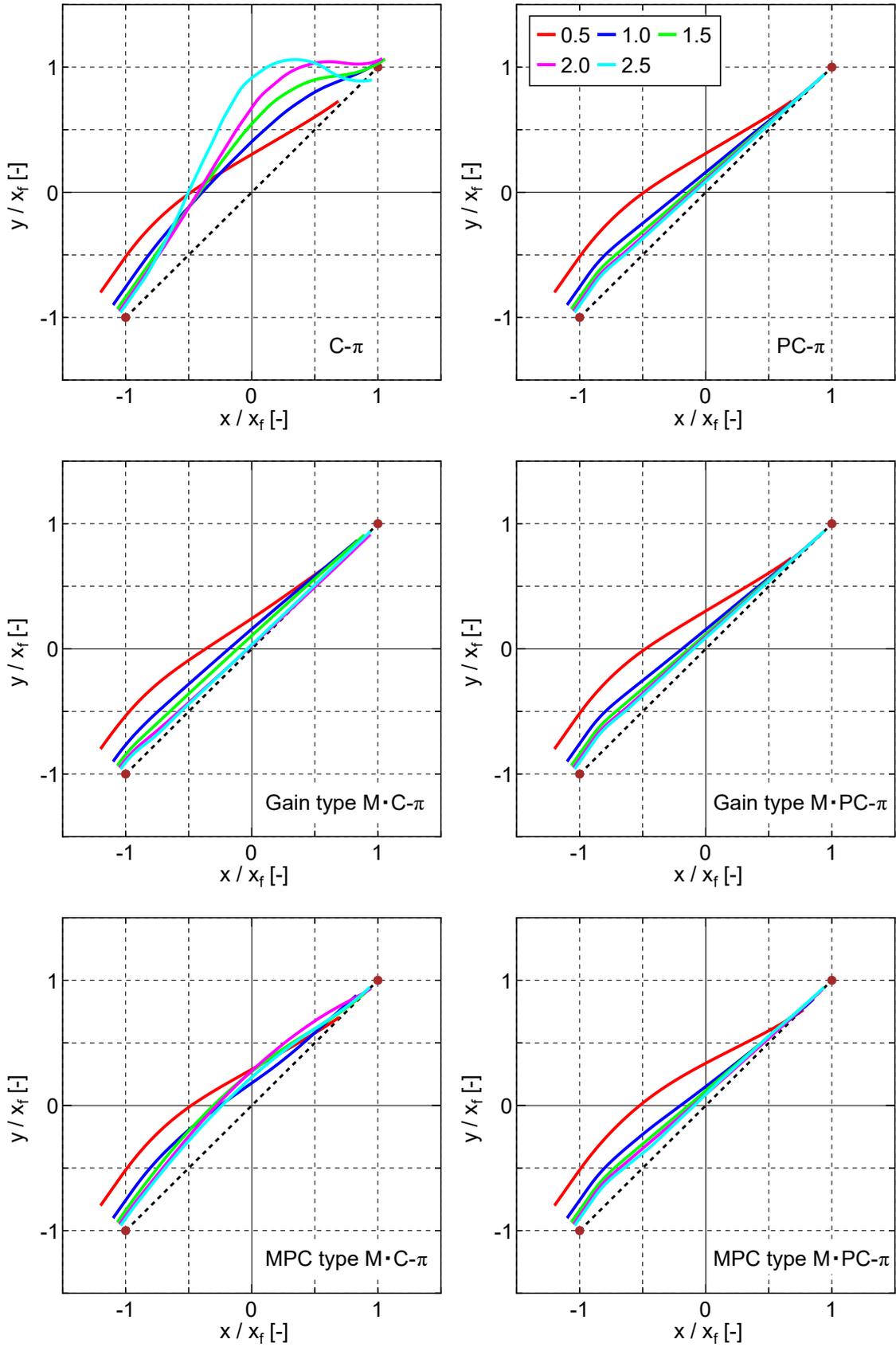
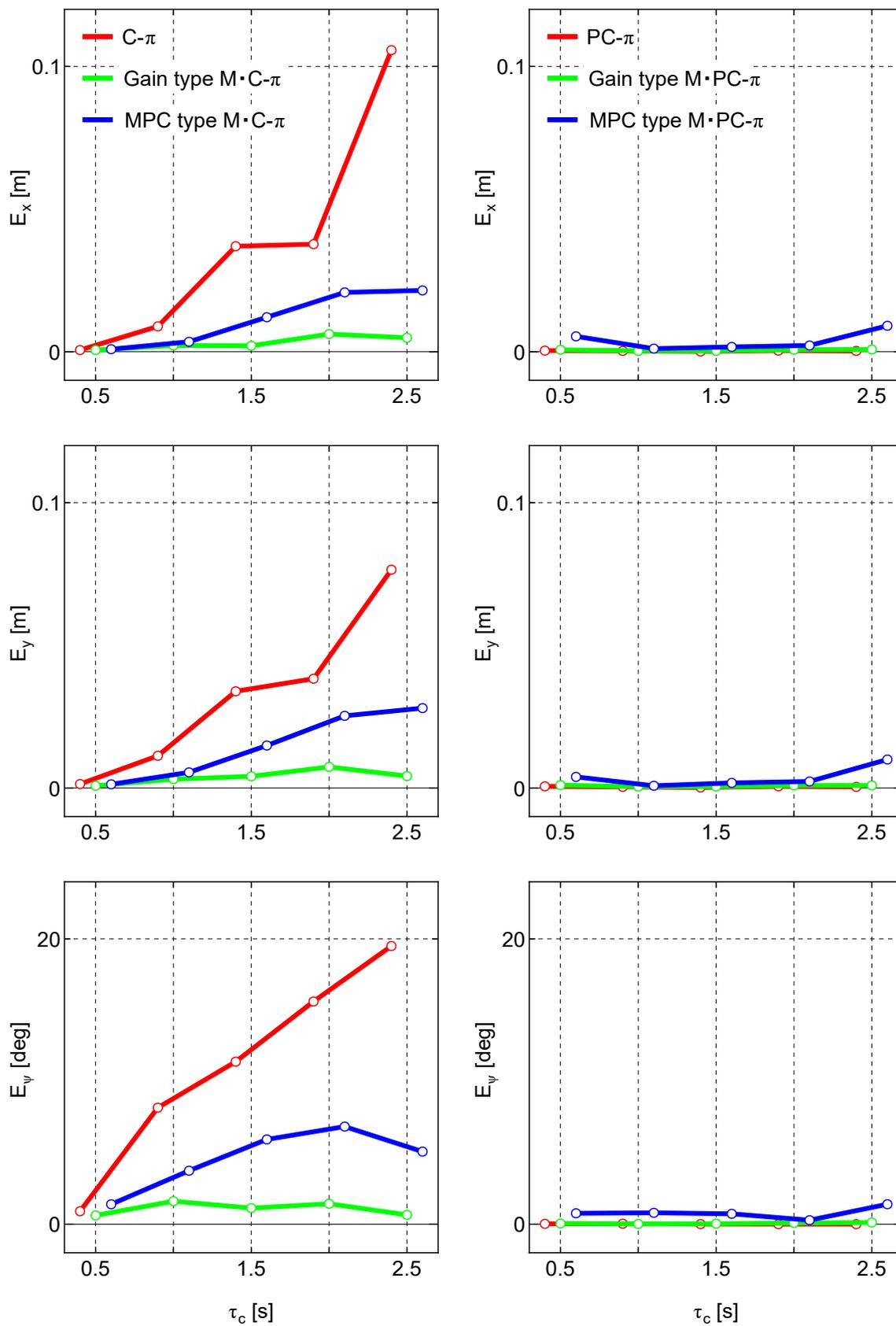


図 4.6 初期状態誤差に対する走行結果（正規化走行軌跡）



(a) C- $\pi$  手法等

(b) PC- $\pi$  手法等

図 4.7 初期状態誤差に対する走行結果（実軌道と予測軌道の誤差）

図 4.6 及び 4.7 により, C- $\pi$  手法に対して Gain 型や MPC 型の追従制御器を組み込むか, もしくはプロパゲーション型にすることで安定性が向上することが示された. その要因について  $\tau_c = 2.5$  s における各手法の走行結果を一例として用いて考察する. 図 4.8~4.10 に各手法での最適状態軌道の生成状況と, 操舵角指令の時間履歴における最適制御解の生成状況を示す. 赤色は実軌道及び実入力, 桃色, 空色, 若草色はそれぞれのサンプリング時刻で生成した最適制御解, シアンはフィードバック制御入力を示す.

本想定では初期状態誤差の一つとしてヨー角誤差を含んでいるため, 走行開始時の  $t = 0$  s の最適化計算では最初に右旋回して目標へ直進していく解が得られている. しかし C- $\pi$  手法では, 最初のサンプリング区間分の  $t = 0 \sim 2.5$  s ではオフラインで計算した解が適用されるため, ヨー角誤差をもったまま直進している.  $t = 2.5, 5.0, \dots$  の 2 回目以降も最初に右旋回を行う解が生成されているが, 適用されない最初のサンプリング区間に右操舵入力の大部分が含まれている. 従って C- $\pi$  手法を用いた場合では, 初期状態誤差を修正することができず, 目標とは異なる方向に直進する結果となっている. ただし, 後半では長い右旋回を必要とする解が生成されているため, 目標方向に移動することができている.

Gain 型や MPC 型の M · C- $\pi$  手法を用いた場合でも走行開始直後の  $t = 0 \sim 2.5$  s のフィードフォワード制御入力は  $\delta_{fc} = 0$  deg が適用されるが, この区間ではオフライン最適軌道が追従の参照軌道となるため, 右旋回を行うフィードバック制御入力も生成されている. Gain 型 M · C- $\pi$  手法を用いた場合では最初の修正で予測軌道へ収束し, 以降は生成した最適軌道に沿って走行することができている. MPC 型 M · C- $\pi$  手法を用いた場合ではフィードフォワード制御入力とフィードバック制御入力の合成割合が 8:2 で適用されるため, フィードバック制御入力は Gain 型 M · C- $\pi$  手法よりも小さくなり左へ迂回する影響を受けている. しかし, 以後のサンプリング時刻においても最適状態軌道との誤差を修正するフィードバック制御入力も生成されているため, 徐々に影響が軽減されている. 従って, C- $\pi$  手法の制御系に Gain 型や MPC 型の追従制御器を組み込むことで, 初期状態誤差に対する安定性の向上に寄与することが示された.

PC- $\pi$  手法を用いた場合については,  $t = 0 \sim 2.5$  s ではオフラインで生成した最適制御入力を用いることを前提としてサンプリングタイム後の状態を予測しているため,  $t = 2.5$  s から右旋回を行う最適制御解が生成されている. そのため,  $t = 0 \sim 2.5$  s においてヨー角誤差をもったまま直進した後に,  $t = 2.5 \sim 5$  s において生成した右操舵入力を適用して右旋回を行っている. これは第 2.4 節で示した入力誤差を要因とする安定性への影響を排除している. 従って PC- $\pi$  手法を用いた場合でも, 初期状態誤差に対する安定性が向上することが示された.

なお, 数値シミュレーションでは最適化計算と MGV の運動のシミュレーションは同じ運動モデルを使用しているためフィードフォワード制御入力による実軌道と予測

軌道との誤差は微小である。そのため Gain 型  $M \cdot PC-\pi$  手法を用いた場合では、生成されるフィードバック制御入力の値も微小であり、 $PC-\pi$  手法と同等の走行結果が得られている。一方、MPC 型  $M \cdot PC-\pi$  手法を用いた場合については、フィードフォワード制御入力の 20 % がフィードバック制御入力の 20 % に変換される。そのため誤差が発生していなくても入力値は微小に変化し、図 4.7(b) で示しているように、 $PC-\pi$  手法や Gain 型  $M \cdot PC-\pi$  手法と比較して誤差が生じている。

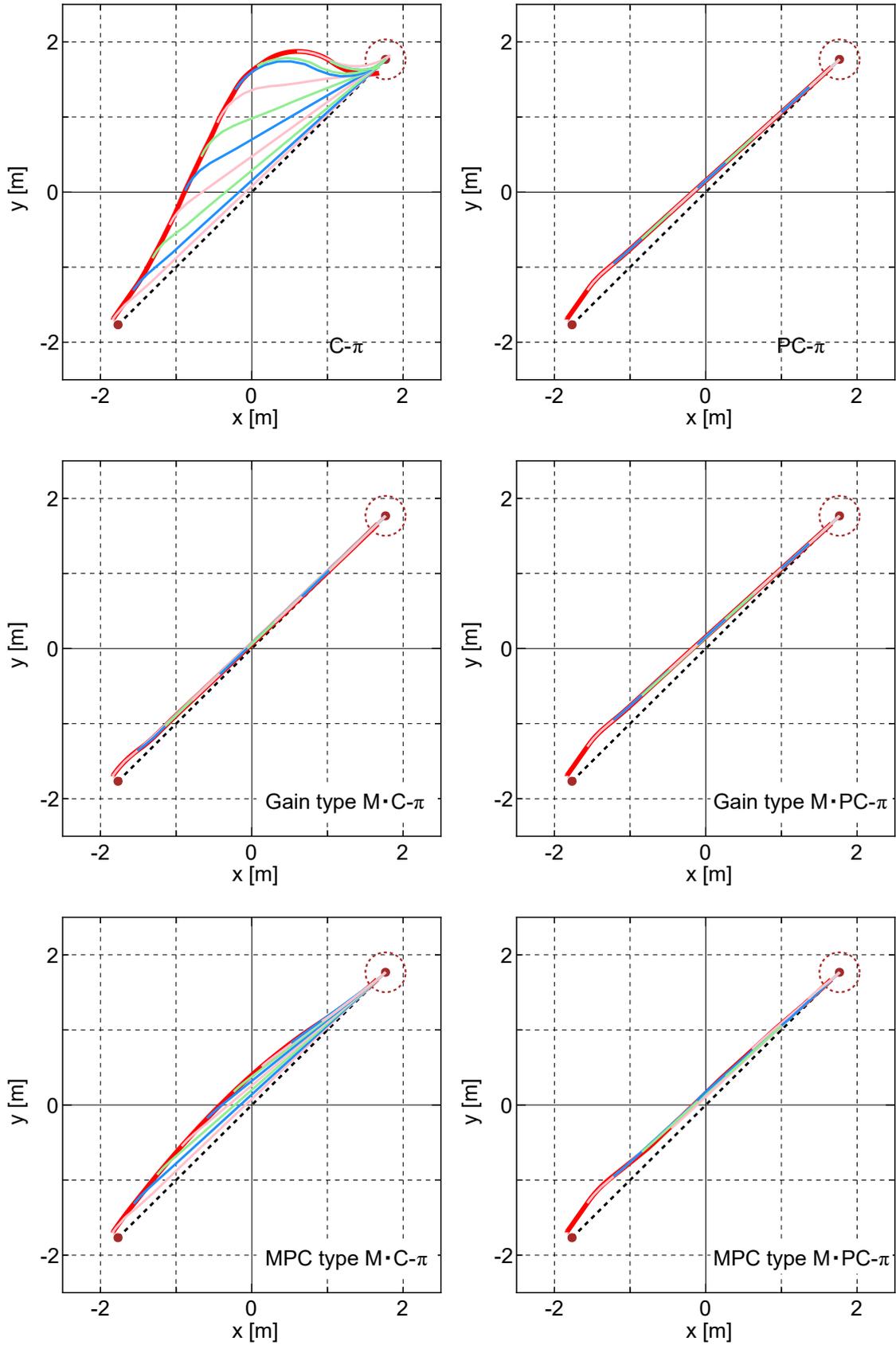
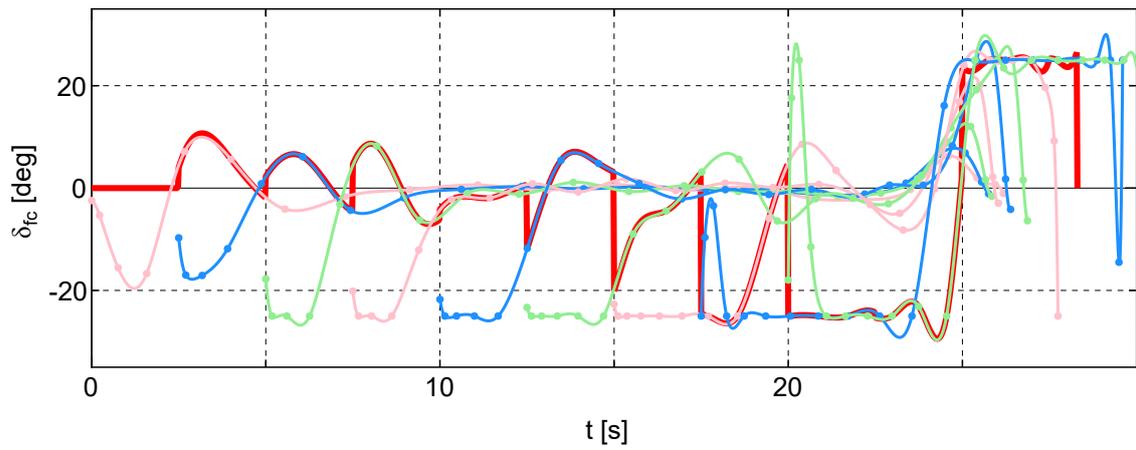
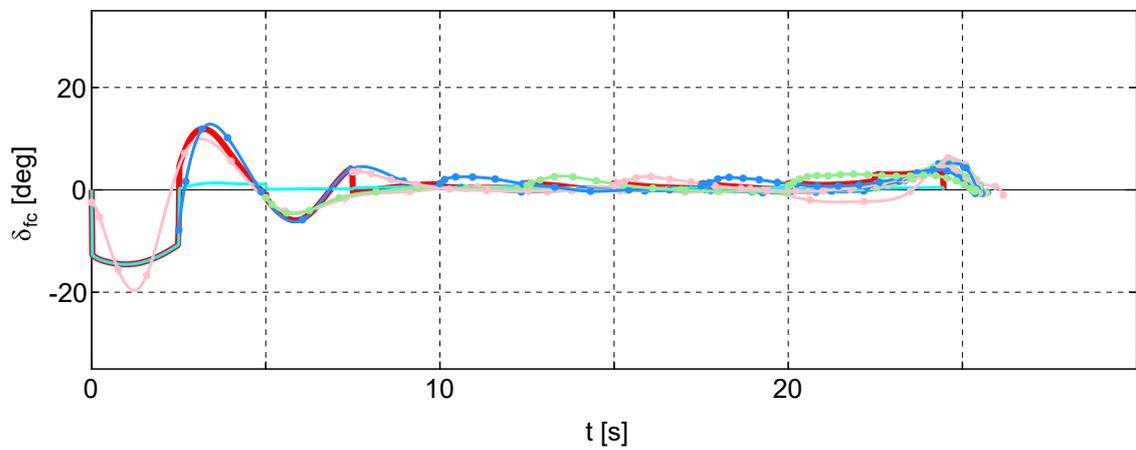


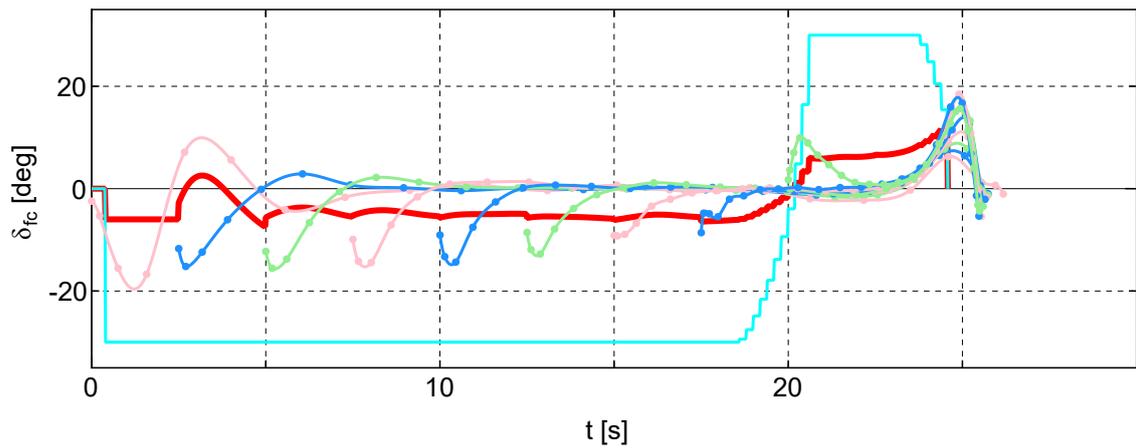
図 4.8 初期状態誤差に対する走行結果（最適解の生成状況， $\tau_c = 2.5$ ，走行軌跡）



(a) C- $\pi$  手法

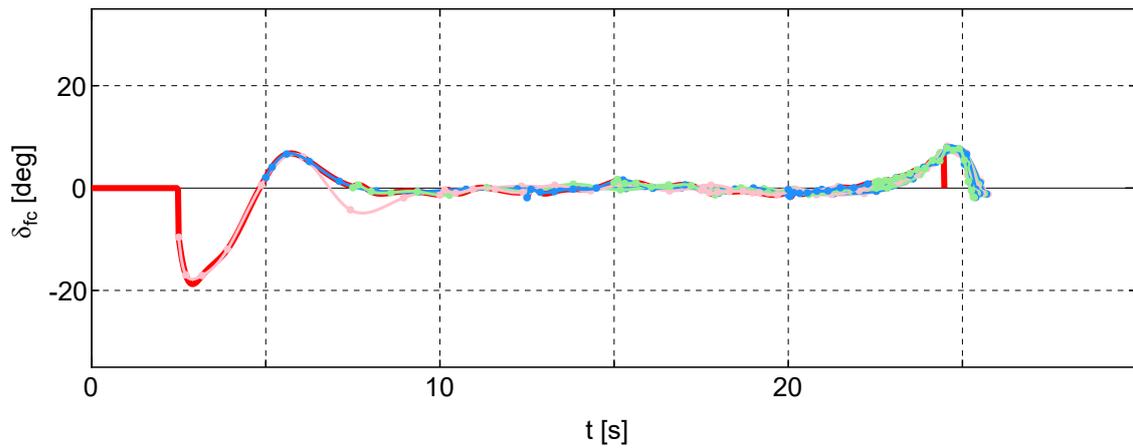


(b) Gain 型 M · C- $\pi$  手法

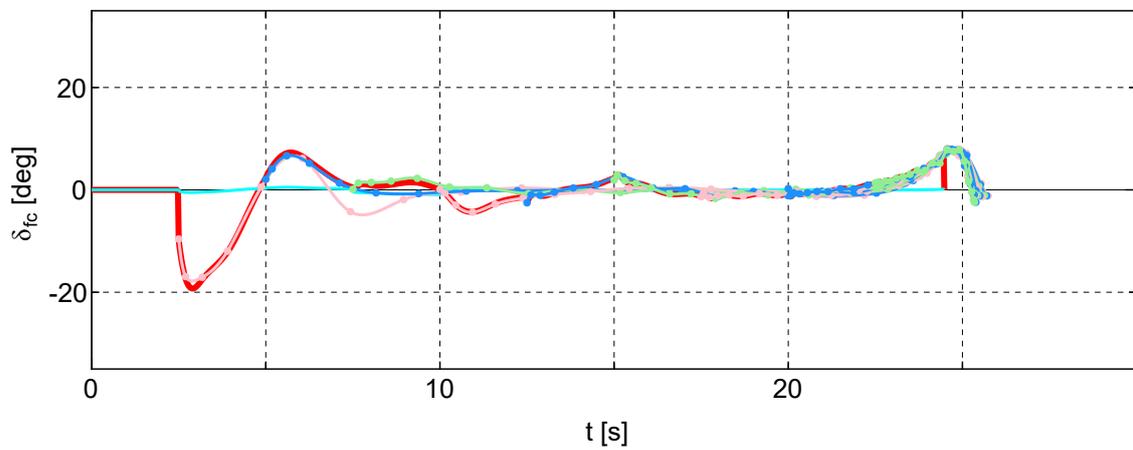


(c) MPC 型 M · C- $\pi$  手法

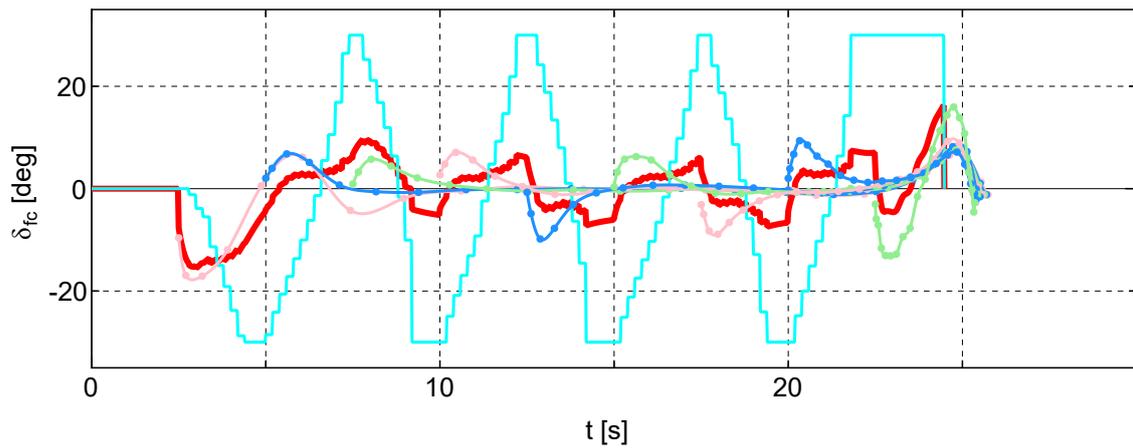
図 4.9 初期状態誤差に対する走行結果（最適解の生成状況， $\tau_c = 2.5$ ， $\delta_{fc}$  の時間履歴，その 1）



(a) PC- $\pi$  手法



(b) Gain 型 M · PC- $\pi$  手法



(c) MPC 型 M · PC- $\pi$  手法

図 4.10 初期状態誤差に対する走行結果（最適解の生成状況， $\tau_c = 2.5$ ， $\delta_{fc}$  の時間履歴，その 2）

## 4.4.2 路面外乱に対する影響

### ・想定及びシミュレーション条件

初期状態誤差に対する影響の検証では、最適化計算と運動計算は同じモデルを使用している。しかし、実際の走行では最適化計算で使用しているモデルと厳密には挙動が一致することはない、外乱の影響などによっては大きく異なることもある。特にMGVは小型・軽量であるため路面の影響を受けやすい。従って、ここでは路面外乱が存在する場合を想定して、それに対する Clock-Based 実時間最適制御の性能を評価する。本研究では、基本段階として、単純かつ明瞭な外乱を設定することで、外乱に対する各手法の基本的な性能を明らかにすることを着眼とする。従って、図 4.11 に示すような、一定の勾配があるような領域を走行するものと仮定する。数値シミュレーションの運動計算においてヨー角速度を算出する部分に対して定常値外乱を印加する。この外乱については、サンプリングタイムごとに行う最適化計算では考慮されない。また、初期状態誤差に対する影響での検証と同様に、 $L_p/\tau_c = 2$  として  $\tau_c$  を変化させたことによる走行結果への影響についても検証する。各手法の制御性能については、2 点間距離をもとに正規化した走行軌跡と、各サンプリング時刻における実軌道と予測軌道の誤差を用いて評価を行う。数値シミュレーションは、目標位置から 0.15 m 以内かつ目標ヨー角との誤差が 10 deg 以内を満たすか、オフライン解の終端時刻から 3 秒超えた場合に終了する。シミュレーションの条件は、初期状態誤差に対する影響の検証と同じく、表 4.5 の値を使用する。

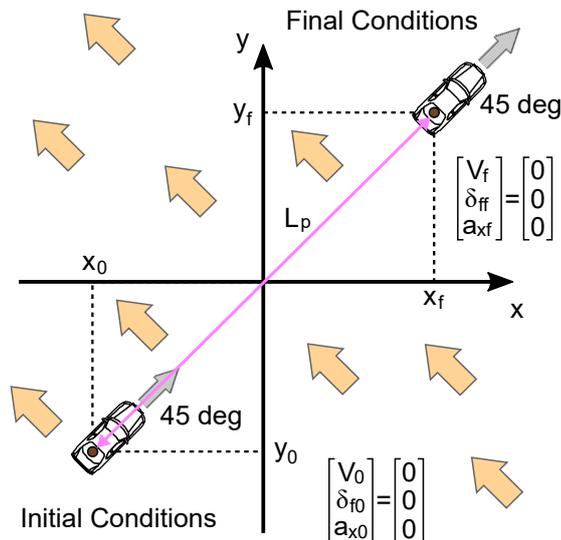


図 4.11 想定概要 (路面外乱)

#### ・数値シミュレーション結果

図 4.12 に、終端  $x$  座標  $x_f$  で  $x$  軸及び  $y$  軸を正規化した走行軌跡を示す。黒色の破線はオフラインで得られた最適状態軌道、それぞれの色の実線が走行軌跡を表している。いずれの手法を用いた場合においても、 $\tau_c$  が長くなるにつれて左へ逸れる影響が大きくなっている。特に C- $\pi$  手法のサンプリングタイムが長い場合においては、数値シミュレーションの制限時間以内に目標まで到達できていない。一方、C- $\pi$  手法や PC- $\pi$  手法に Gain 型や MPC 型の追従制御器を組み込んだ M・C- $\pi$  手法や M・PC- $\pi$  手法を用いた場合では、 $\tau_c$  が長くなることに伴う左へ逸れる影響の増大を抑制できている。

図 4.13 に、それぞれの走行における各サンプリング時刻での実軌道と予測軌道の誤差を整理した結果を示す。○プロットは誤差の平均値を表し、 $x$  方向、 $y$  方向、 $\psi$  方向に分けてまとめている。なお、結果の重複を避けて見やすくするため、C- $\pi$  手法及び PC- $\pi$  手法は負の方向、MPC 型の M・C- $\pi$  手法及び M・PC- $\pi$  手法は正の方向に横軸をずらしている。

C- $\pi$  手法を用いた場合では  $\tau_c$  が長くなるにつれて誤差が増大し走行が不安定になる傾向があるが、Gain 型や MPC 型の M・C- $\pi$  手法を用いた場合では誤差の増大を抑制することができている。PC- $\pi$  手法を用いた場合では、初期状態誤差に対する影響の検証ではサンプリングタイムの増加に伴う誤差の増大はなかったが、路面外乱の影響がある場合では  $\tau_c$  が長くなるにつれて誤差が増大している。しかし、Gain 型や MPC 型の M・PC- $\pi$  手法を用いることで誤差の増大を抑制することができている。特に MPC 型 M・PC- $\pi$  手法は初期状態誤差の場合の図 4.7(b) と比較すると、Gain 型 M・PC- $\pi$  手法よりも抑制していることが分かる。

図 4.12 及び 4.13 により、C- $\pi$  手法や PC- $\pi$  手法に対して Gain 型や MPC 型の追従制御器を組み込むことで安定性が向上することを確認したが、その要因について  $\tau_c = 2.5$  s における各手法の走行結果を一例として用いて考察する。図 4.14~4.16 に各手法での最適状態軌道の生成状況と、操舵角指令の時間履歴における最適制御解の生成状況を示す。赤色は実軌道及び実入力、桃色、空色、若草色はそれぞれのサンプリング時刻で生成した最適制御解、シアンはフィードバック制御入力を示す。

C- $\pi$  手法を用いた場合では、 $t = 0$  s の走行開始から  $t = 2.5$  s までオフライン解の  $\delta_{fc} = 0$  deg が入力されるため、左へ逸れている。 $t = 2.5$  s 以降の各サンプリング時刻ではいずれも右旋回を行う最適解を生成することを試みているが、初期状態誤差がある場合と同様に、最初のサンプリング区間の入力は適用されないため、そのまま大きく左へ逸れている。

Gain 型や MPC 型の M・C- $\pi$  手法を用いた場合では  $t = 0 \sim 2.5$  s においても、フィードバック制御入力を生成し右操舵を行ってオフライン最適軌道との誤差を修正している。ただし路面外乱の影響を完全には修正していないため、各サンプリング時刻の最

適解では最初に右旋回を行う最適解が生成されている。C- $\pi$ 手法と同様にこの区間の入力は適用されないが、フィードバック制御入力を生成することで左へ逸れる影響を軽減している。従って、C- $\pi$ 手法の制御系に Gain 型や MPC 型の追従制御器を組み込むことで路面外乱に対する安定性の向上に寄与することが示された。

PC- $\pi$ 手法を用いた場合については、C- $\pi$ 手法と異なり右操舵がそのまま適用されるため、左へ逸れる影響が軽減している。しかし、最適化計算では路面の影響は考慮できないため、十分な右旋回を行うことはできない。一方、Gain 型 M・PC- $\pi$ 手法を用いた場合では、不足している右への操舵入力に対して発生した最適軌道との誤差を修正する入力が増えられる。MPC 型 M・PC- $\pi$ 手法を用いた場合でも、フィードバック制御入力はフィードフォワード制御入力よりも大きな値となっているため、より大きな右への操舵入力が増え適用されることになり、左へ逸れる影響が軽減している。PC- $\pi$ 手法の制御系に Gain 型や MPC 型の追従制御器を組み込むことで、路面外乱に対する安定性が向上することが示された。

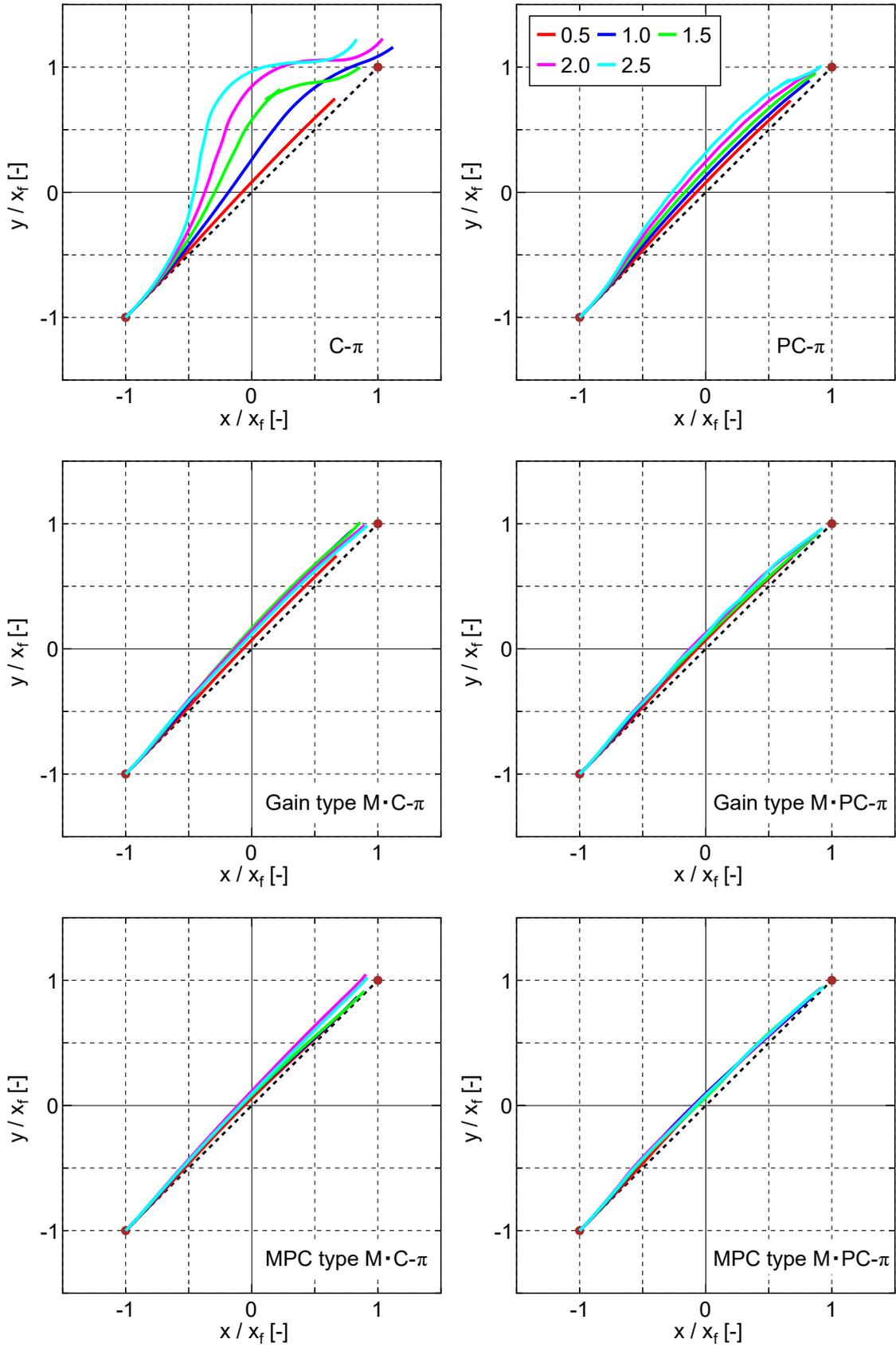
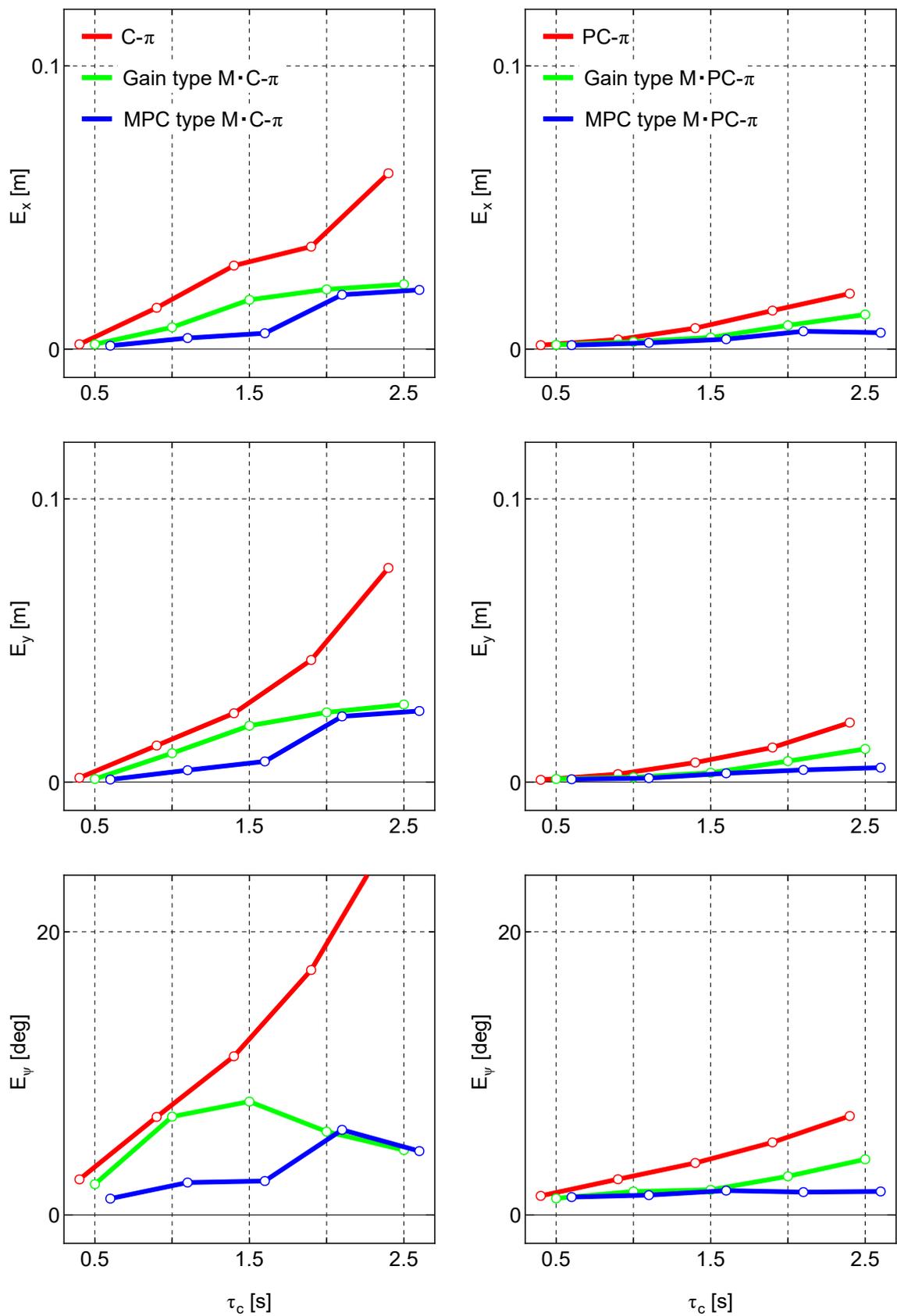


図 4.12 路面外乱に対する走行結果（正規化走行軌跡）



(a) C- $\pi$  手法等

(b) PC- $\pi$  手法等

図 4.13 路面外乱に対する走行結果 (実軌道と予測軌道の誤差)

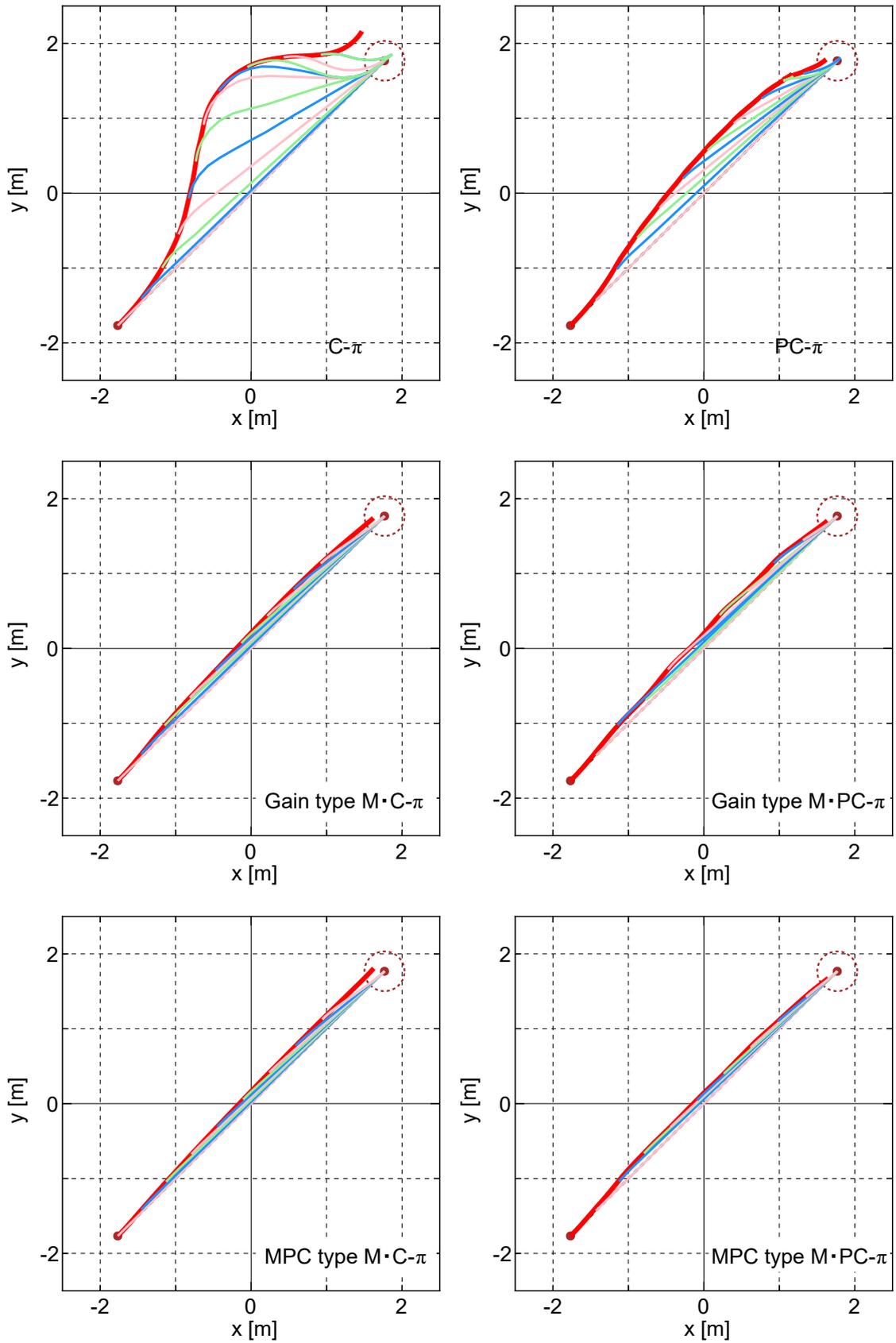
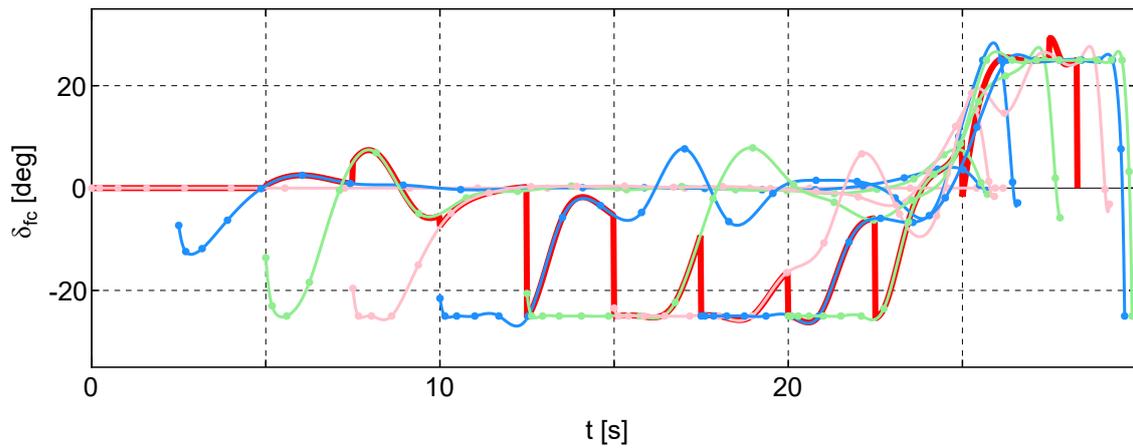
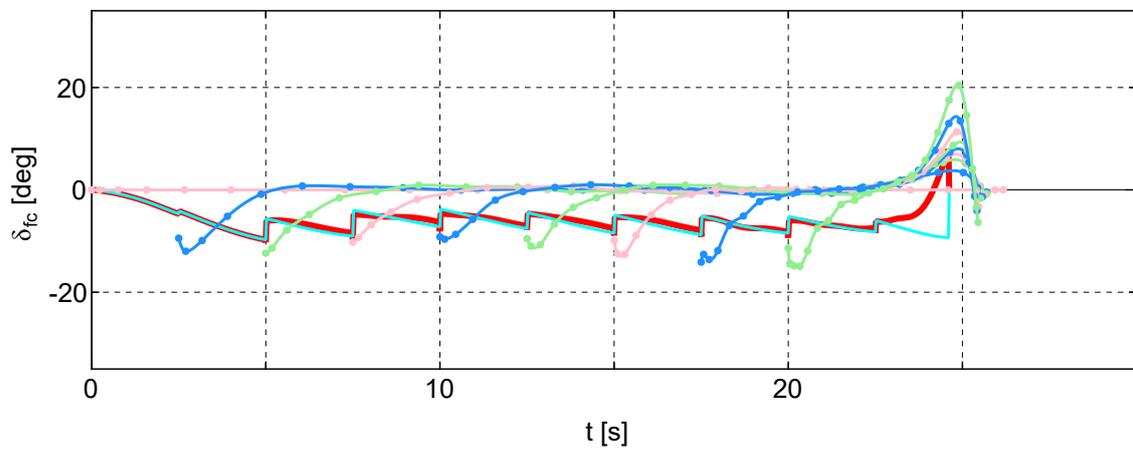


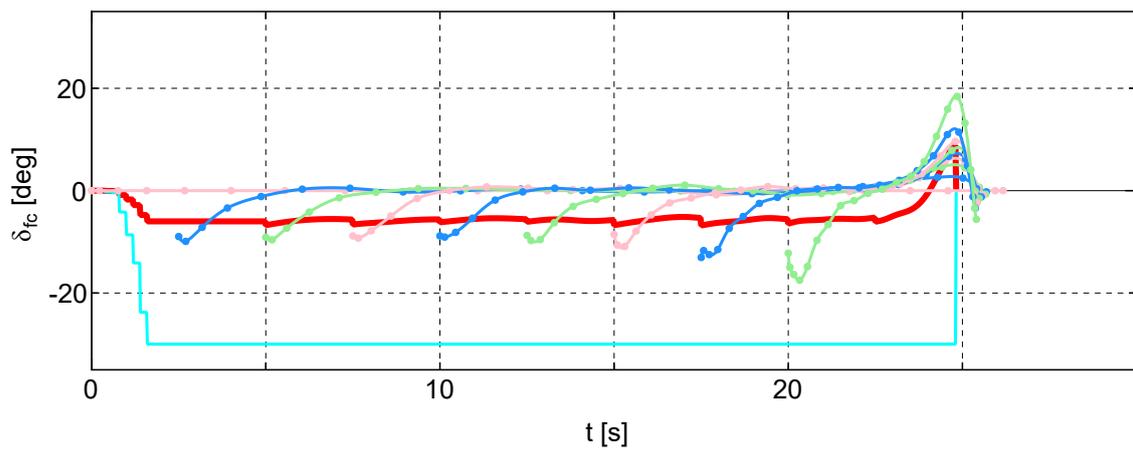
図 4.14 路面外乱に対する走行結果（最適解の生成状況， $\tau_c = 2.5$ ，走行軌跡）



(a) C- $\pi$  手法

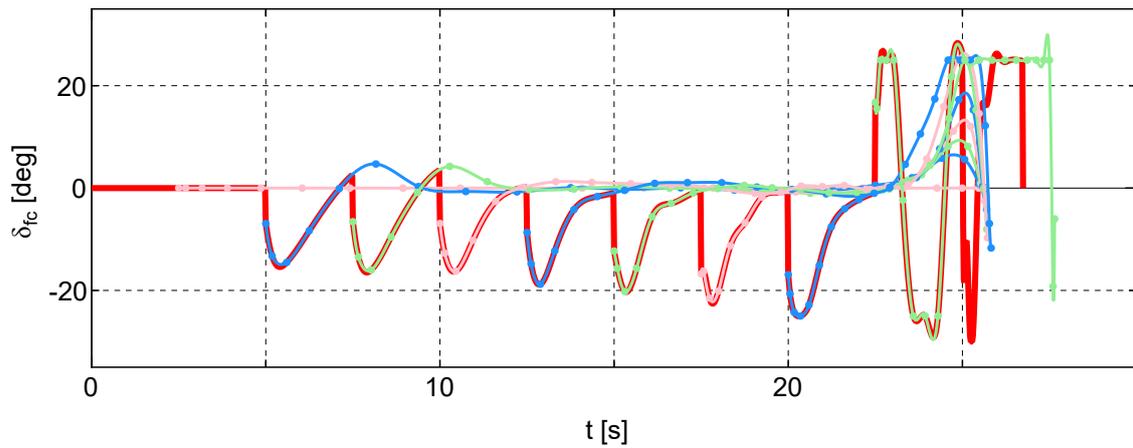


(b) Gain 型 M · C- $\pi$  手法

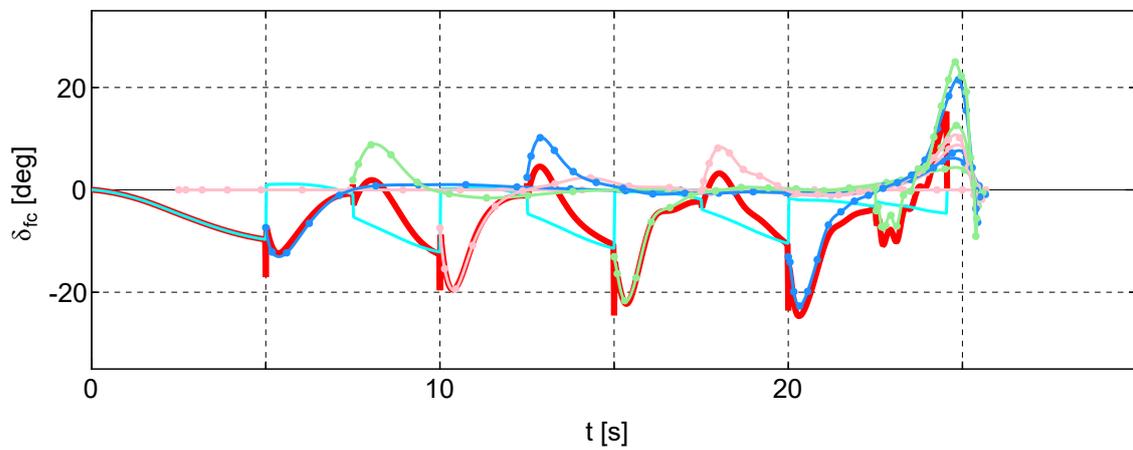


(c) MPC 型 M · C- $\pi$  手法

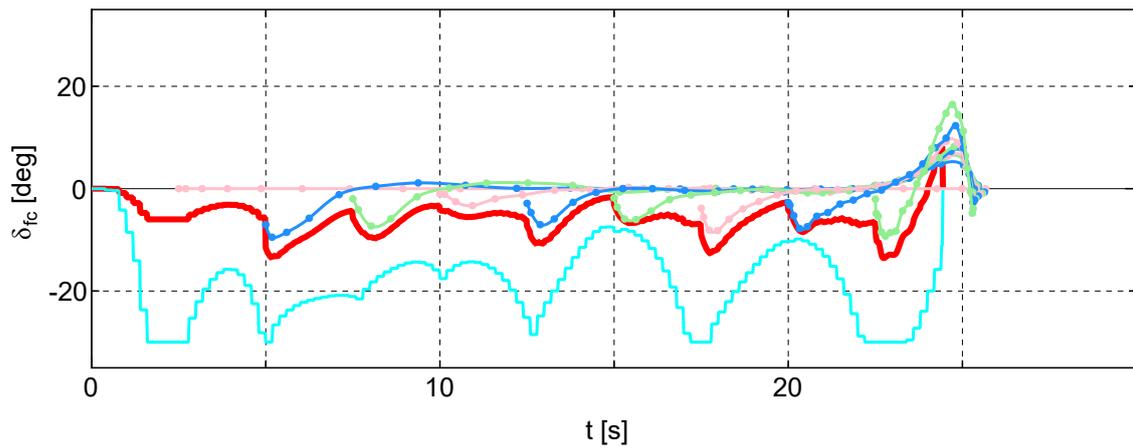
図 4.15 路面外乱に対する走行結果（最適解の生成状況， $\tau_c = 2.5$ ， $\delta_{fc}$  の時間履歴，その 1）



(a) PC- $\pi$  手法



(b) Gain 型 M · PC- $\pi$  手法



(c) MPC 型 M · PC- $\pi$  手法

図 4.16 路面外乱に対する走行結果（最適解の生成状況， $\tau_c = 2.5$ ， $\delta_{fc}$  の時間履歴，その 2）

## 4.5 実機を用いた検証

MGV の運動制御実験では、各手法を用いる場合の基本的な走行性能と、応用的な状況に対する性能を検証するため、以下の4種類の走行を想定する。

1. 直線走行
2. S字走行
3. 繰り返し走行
4. 障害物回避走行

さらに直線走行，S字走行，及び繰り返し走行では初期状態誤差がある場合についても考え，障害物回避走行では障害物の発生時刻を変化させる。

表 4.6 に MGV の運動制御における共通の走行条件を示す。サンプリングタイム  $\tau_c$  については，最適制御解を概ね得ることができる 1.0 秒と，長くなった場合として 2.0 秒の 2 パターンについて行う。

表 4.6 MGV の運動制御における共通の走行条件

$\tau_c$ [s]	$W_{ddf}$ [-]	$W_{obs}$ [-]
1.0, 2.0	0.005	0, 50

Clock-Based 実時間最適制御は走行時の状態に誤差の発生が許容されているので，目標位置との距離が 0.15 m 以内かつ目標ヨー角との誤差が 10 deg 以内になったならば走行実験を終了する。また安全上，オフラインの最適化計算で得られた走行時間より 5 秒以上経過した場合には強制的に終了する。最適化計算における離散点数は  $N = 21$  とし，走行結果の最適解の生成状況では最適状態軌道と併せて表示している。

### 4.5.1 直線走行

・想定

図 4.17 に直線走行における想定の概要及びオフラインの最適化計算で得られた走行軌跡を示す。直線走行では、座標  $(-1, -1)$ ，ヨー角  $45 \text{ deg}$  の状態から座標  $(1, 1)$ ，ヨー角  $45 \text{ deg}$  の状態まで移動する。その他の状態は 0 としている。

MGV の最も基本的な走行としては前方の目標地点へ移動することが想定される。しかし、走行開始位置やヨー角に誤差が生じて必ずしも進行方向に目標地点が位置するとは限らず、実時間最適制御によりその誤差に対応して目標まで到達しなければならない。そのため、直線走行では以下の 3 つの初期状態を想定した走行実験を行う。

- ・ CASE 1 : 初期状態誤差なし
- ・ CASE 2 : 想定した初期位置の 0.1 m 左側から走行開始
- ・ CASE 3 : CASE 2 にさらに  $15 \text{ deg}$  の外側へヨー角誤差を持たせて走行開始

いずれの場合においても機動初期に用いるオフライン解は初期状態誤差なしの解を用いる。仮に CASE 2 及び CASE 3 の初期状態が既知の場合に得られる最適軌道は図 4.17(b) の赤色及び青色の実線のようになる。

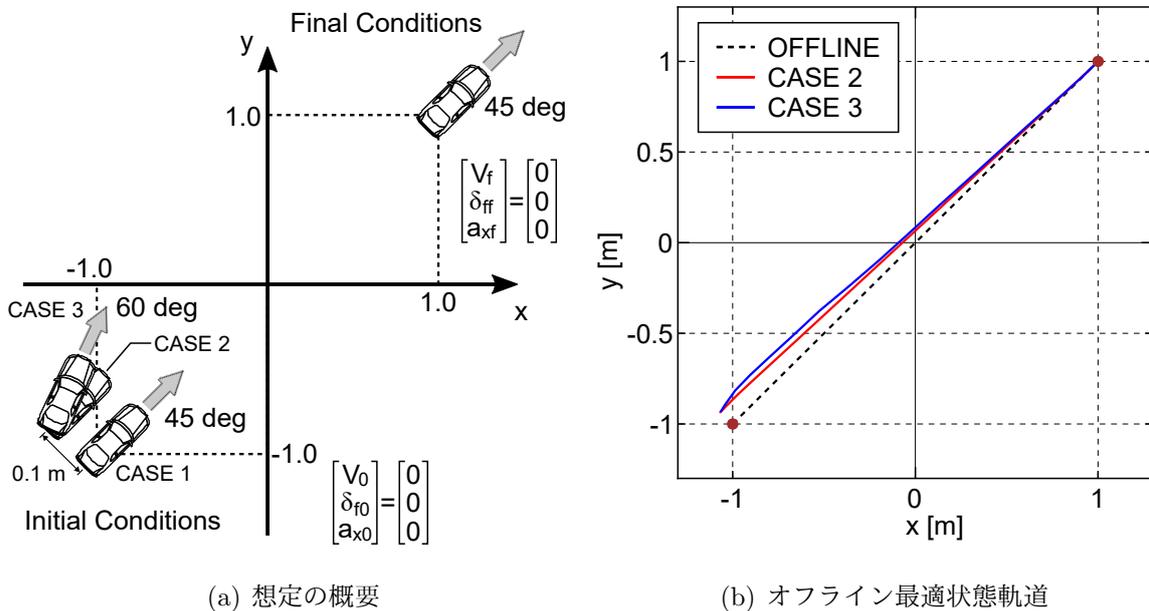


図 4.17 MGVS の運動制御の想定：直線走行

・実験結果

図 4.18 に直線走行における全ての走行パターンと、本節で示している実験結果の関係を示す。

- ① 最適解の生成状況では、時系列で最適軌道の生成の様子を示し、直線走行において移動しながら最適解を更新できていることを確認する。
- ② 各手法の時間履歴では、MGV の状態や実際に適用した制御入力、追従制御器により生成した制御入力の時間履歴を示し、直線走行に対する各手法の制御状況を確認する。
- ③ 走行軌跡では、走行パターンごとの3回ずつの走行軌跡を示し、直線走行における走行実験の再現性を確認する。
- ④ 予測軌道との誤差では、直線走行の全ての走行パターンにおける各サンプリングタイム・各手法での予測軌道と実軌道との誤差の平均を状態別に整理し、各手法の制御性能を比較する。

図 4.19 及び 4.20 に、直線走行，CASE 3， $\tau_c = 1.0$  s における 2 秒ごとの C- $\pi$  手法及び PC- $\pi$  手法による最適解の生成状況を示す。黒色の破線がオフライン最適軌道，赤色の実線が実軌道，マゼンダの実線が各時刻で生成した最適状態軌道を表している。各時刻において，C- $\pi$  手法ではサンプリングした状態から，PC- $\pi$  手法では予測した状態から目標状態までの最適解が生成できている。一般的な追従制御の PD 制御や MPC を用いた場合ではオフラインで生成した直線軌道に復帰することで初期状態誤差に対するロバスト性を保持するが，C- $\pi$  手法及び PC- $\pi$  手法ではヨー角を目標方向へ向けてから目標まで直線的に走行し，かつ目標まで到達している。

	$\tau_c$	C- $\pi$	PC- $\pi$	Gain型 M・C- $\pi$	Gain型 M・PC- $\pi$	MPC型 M・C- $\pi$	MPC型 M・PC- $\pi$
CASE 1 (なし)	1.0						
	2.0						
CASE 2 (左側)	1.0						
	2.0						
CASE 3 (ヨー角)	1.0						
	2.0						

- ① 最適解の生成状況 (pp. 85 - 86)
- ② 各手法の時間履歴 (pp. 87 - 89)
- ③ 走行軌跡 (pp. 90 - 95)
- ④ 予測軌道との誤差 (p. 96)

図 4.18 直線走行実験の走行パターンと結果の一覧

図 4.21～4.23 に、直線走行、CASE 3,  $\tau_c = 2.0$  s における各手法の時間履歴を示す。黒色の短鎖線はオフラインでの最適制御解、赤色の実線は実軌道、マゼンダ、緑色、シアンの実線は各サンプリング時刻において生成された走行軌跡、空色の実線は追従機構により生成された制御入力を示す。

図 4.21(a) の C- $\pi$  手法を用いた場合では、各サンプリング時刻において目標までの最適制御解を生成しているが、左へ大きく迂回している。それぞれの最適制御解では、最初に右操舵を行ってヨー角を目標方向へ向け、目標まで直進する解となっている。しかしサンプリングの長い  $\tau_c = 2.0$  s の場合では、右操舵の大部分が適用されない最初のサンプリング区間に含まれているため、ヨー角を目標方向へ向けることができていない。走行の終盤では単純な旋回のみでは目標に到達できない位置へ移動したため、切り返し操作を含んだ解が生成されているが、制限時間になったため強制的に走行実験を終了している。

一方、図 4.22(a) 及び 4.23(a) の Gain 型 M・C- $\pi$  手法及び MPC 型 M・C- $\pi$  手法を用いた場合では、 $t = 0 \sim 2$  s においてオフライン軌道との誤差を修正するための右操舵のフィードバック制御入力生成され、ヨー角を目標方向へ向けることができていく。MPC 型 M・C- $\pi$  手法については、生成したフィードバック制御入力はフィードフォワード制御入力と 8:2 の割合で合成されるため、緩やかにヨー角を目標方向へ向けている。以降はフィードバック制御入力を生成してヨー角を目標方向に保ちながら目標まで移動している。

図 4.21(b), 4.22(b), 4.23(b) の PC- $\pi$  手法, Gain 型 M・PC- $\pi$  手法, MPC 型 M・PC- $\pi$  手法を用いた場合では、 $t = 0 \sim 2$  s は直進することを前提として  $t = 2$  s から右操舵を行う入力を生成し、そのまま適用しているため、ヨー角を目標方向へ向けることができていく。また Gain 型 M・PC- $\pi$  手法や MPC 型 M・PC- $\pi$  手法では、フィードバック制御入力を生成して最適状態軌道との誤差を修正しながら目標まで移動している。

図 4.24～4.29 に直線走行の全走行パターンにおける走行軌跡の結果を示す。いずれの走行パターンにおいても 3 回ずつ走行を行っており、概ね同様の結果が得られている。CASE 3 では CASE 1 及び CASE 2 と比較して走行結果に差異が生じる傾向がある。CASE 3 の初期状態誤差の設定は、第 4.4.1 項と同様の位置誤差及びヨー角誤差を与えている場合である。図 4.28 及び 4.29 の各手法の様子は、図 4.6 と類似していることがわかる。

ここで、各サンプリング時刻で生成した最適軌道と実軌道との誤差により各制御手法の性能を比較して評価する。誤差の算出方法は第 4.4 節の数値シミュレーションの場合と同様に行う。図 4.30 に各手法、各サンプリングタイムにおいて予測軌道と実軌道との誤差の平均値を状態変数別に整理した結果を示す。CASE 1 及び CASE 2 では

各手法とも同等の結果を得ているが、CASE 3のC- $\pi$ 手法を用いた場合においてサンプリングタイムが長くなると誤差が大きく増加している。一方で、Gain型M・C- $\pi$ 手法やMPC型M・C- $\pi$ 手法、プロパゲーション型のそれぞれの手法では誤差の増加が抑制されている。Gain型やMPC型の追従制御器を組み込むこと、または初期条件を予測するプロパゲーション型を用いることで安定性が向上することが定量的に示されている。この傾向は図4.7の数値シミュレーションと類似の結果となっている。

以上より直線走行における走行実験で以下の結果を得た。

- (1) 直線走行の各想定において、設計した各手法により最適制御解を更新しながら実機のMGVを制御することができた。
- (2) Gain型やMPC型の追従制御器を組み込むこと、または初期条件を予測するプロパゲーション型を用いることで安定性が向上することを示した。

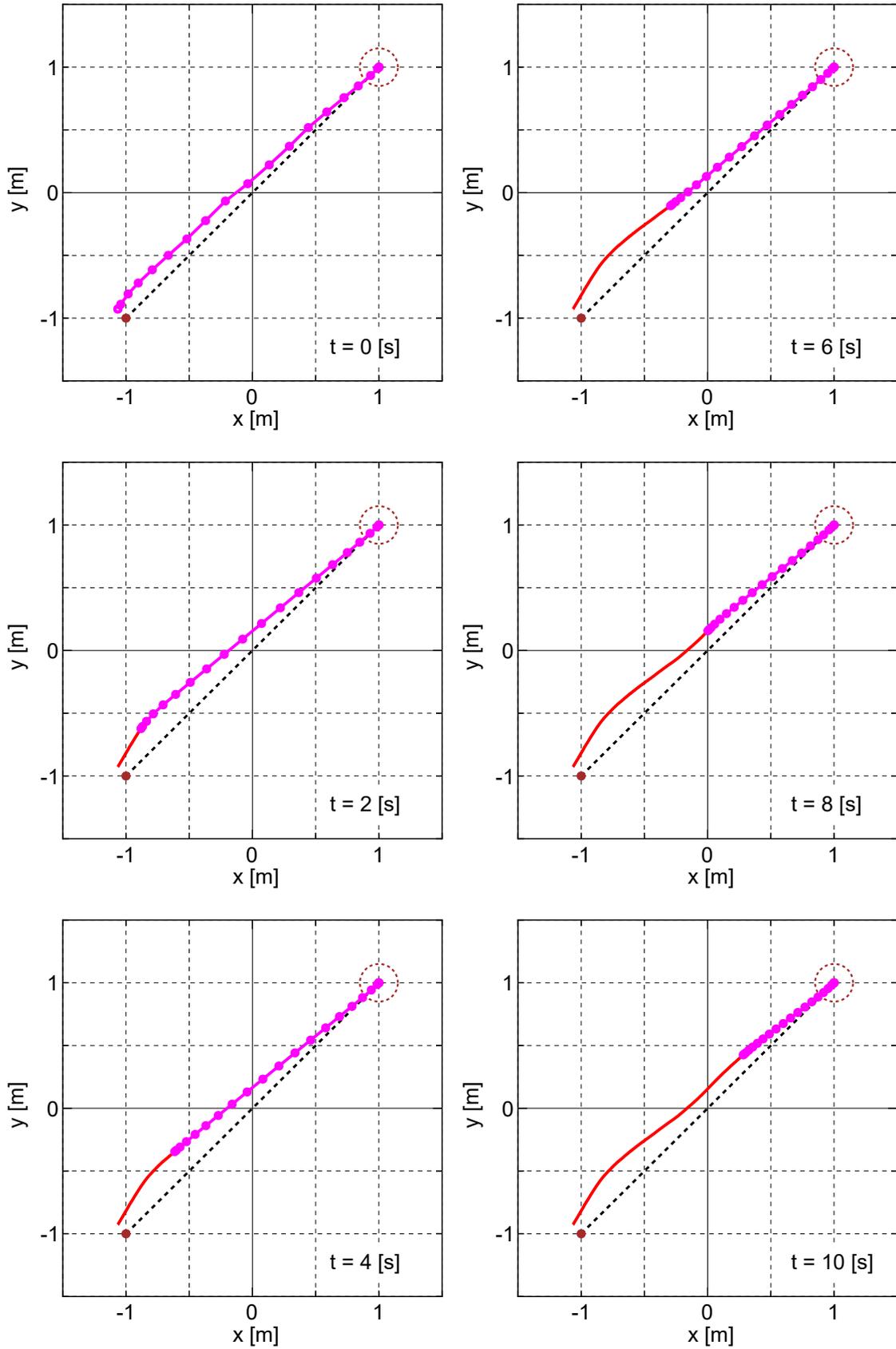


図 4.19 直線走行における最適解の生成状況 (C- $\pi$ 手法, CASE 3,  $\tau_c=1.0$ )

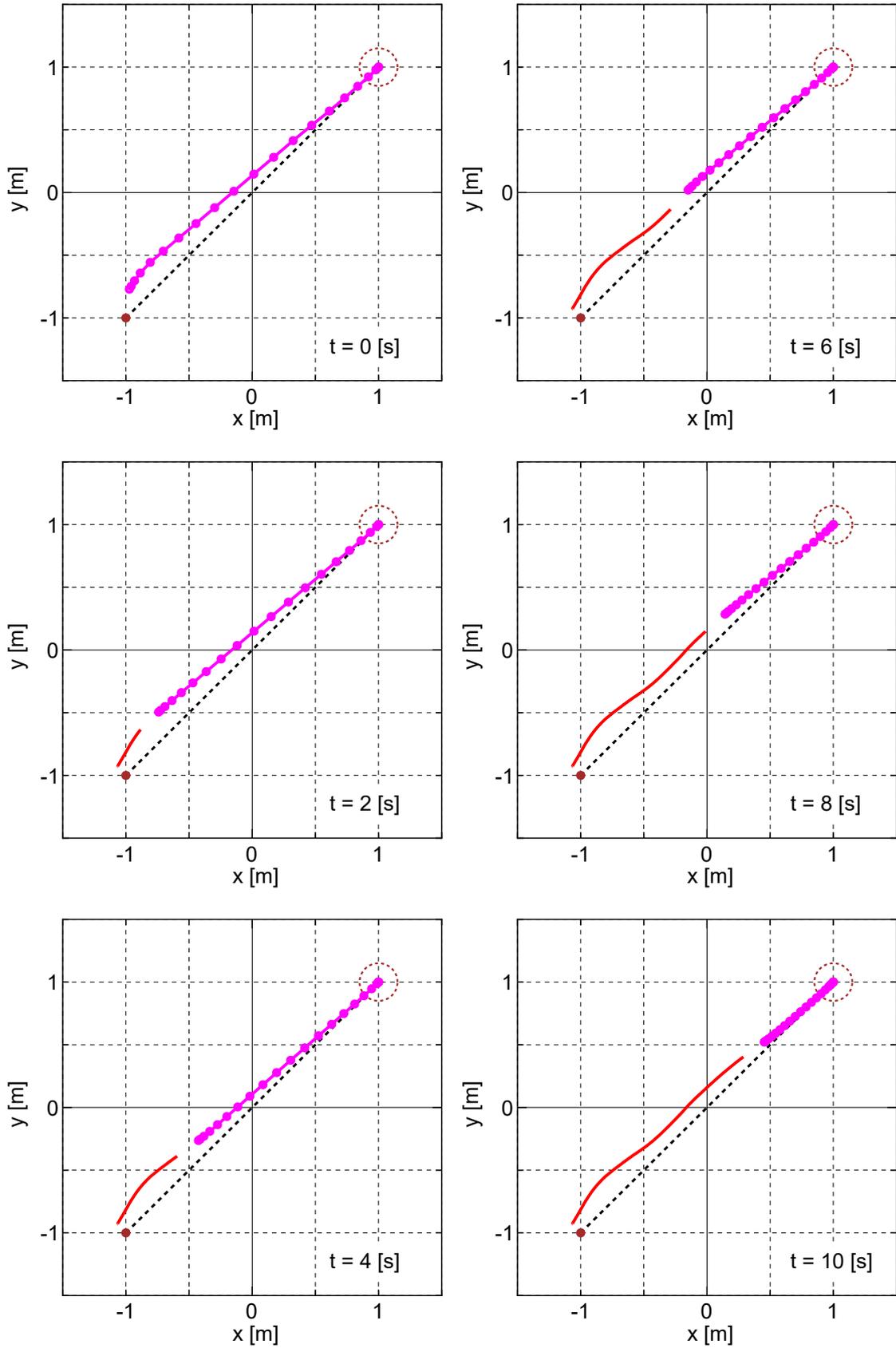


図 4.20 直線走行における最適解の生成状況 (PC- $\pi$ 手法, CASE 3,  $\tau_c=1.0$ )

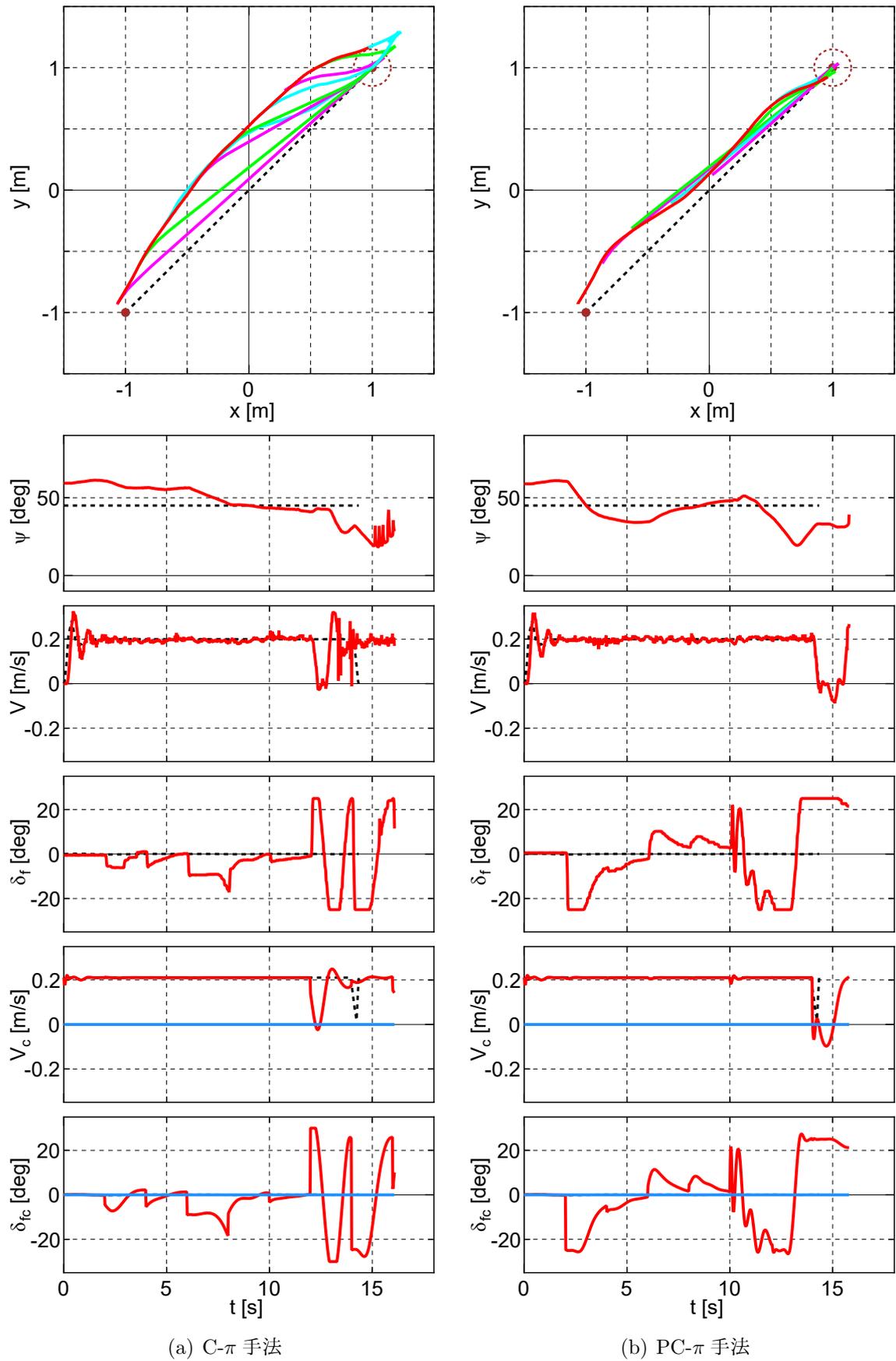


図 4.21 直線走行における走行軌跡と各変数の時間履歴 (CASE 3,  $\tau_c = 2.0$ , その1)

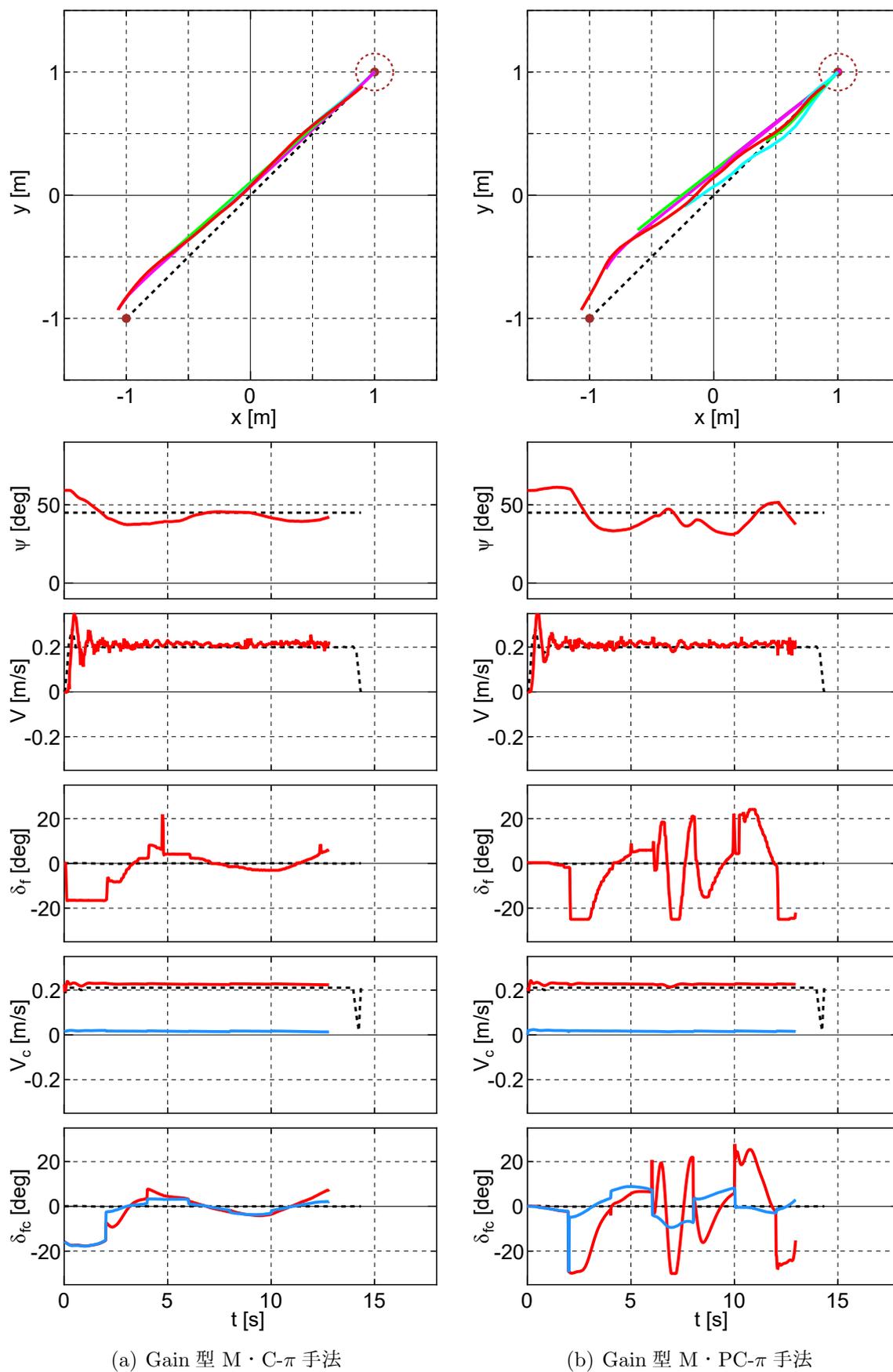


図 4.22 直線走行における走行軌跡と各変数の時間履歴 (CASE 3,  $\tau_c = 2.0$ , その 2)

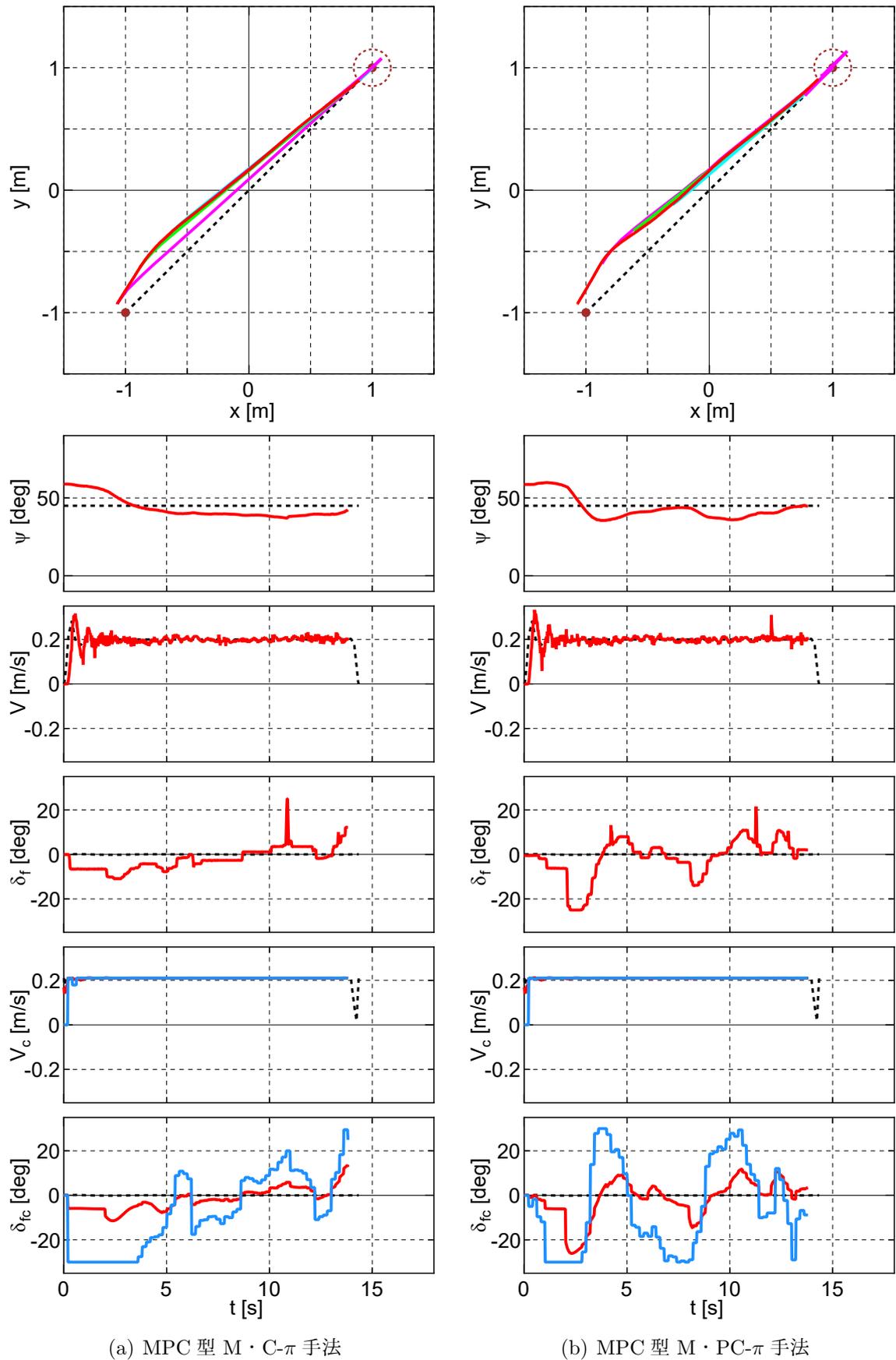


図 4.23 直線走行における走行軌跡と各変数の時間履歴 (CASE 3,  $\tau_c = 2.0$ , その 3)

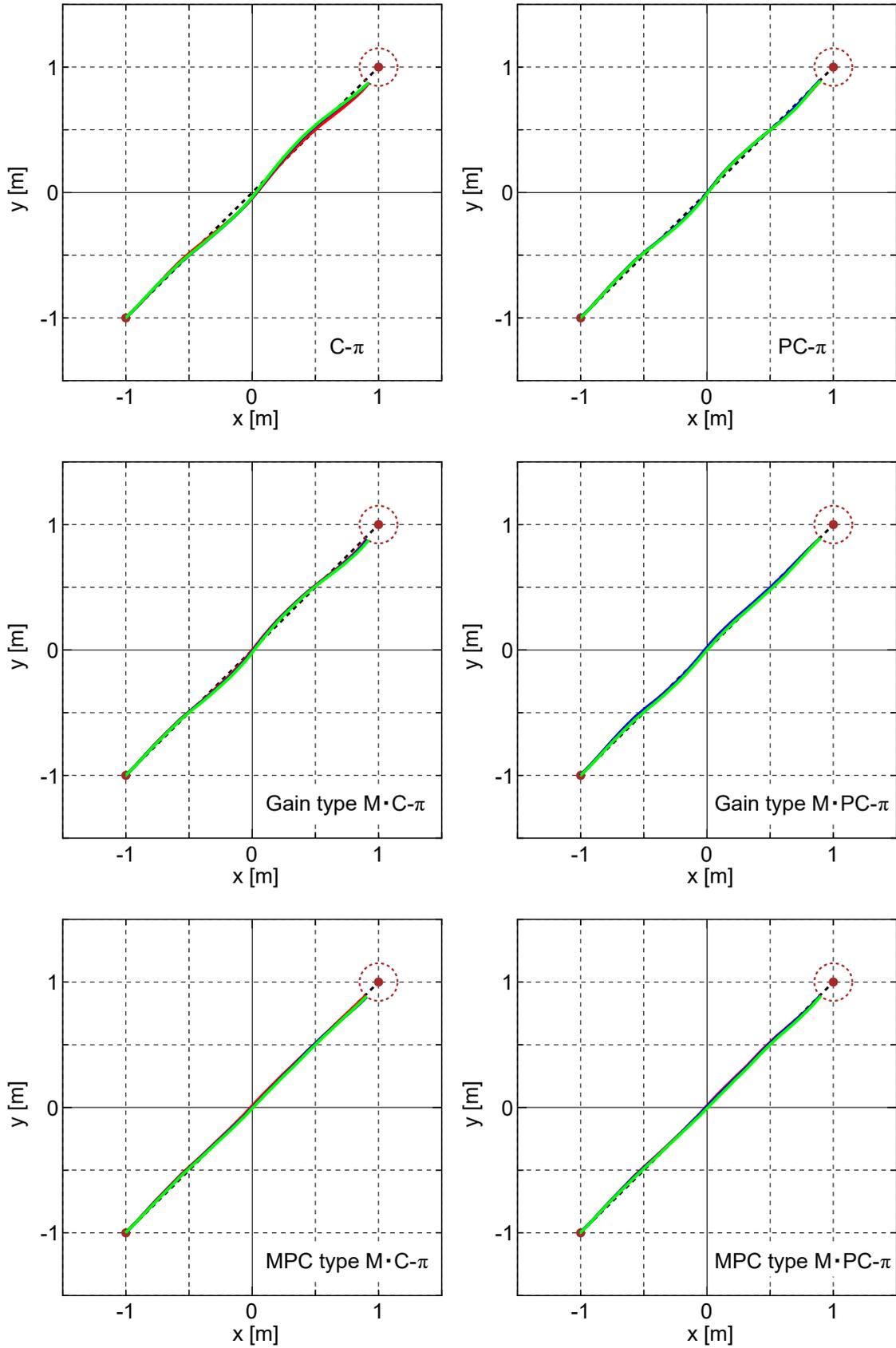


図 4.24 直線走行における実験の再現性の確認 (CASE 1,  $\tau_c=1.0$ )

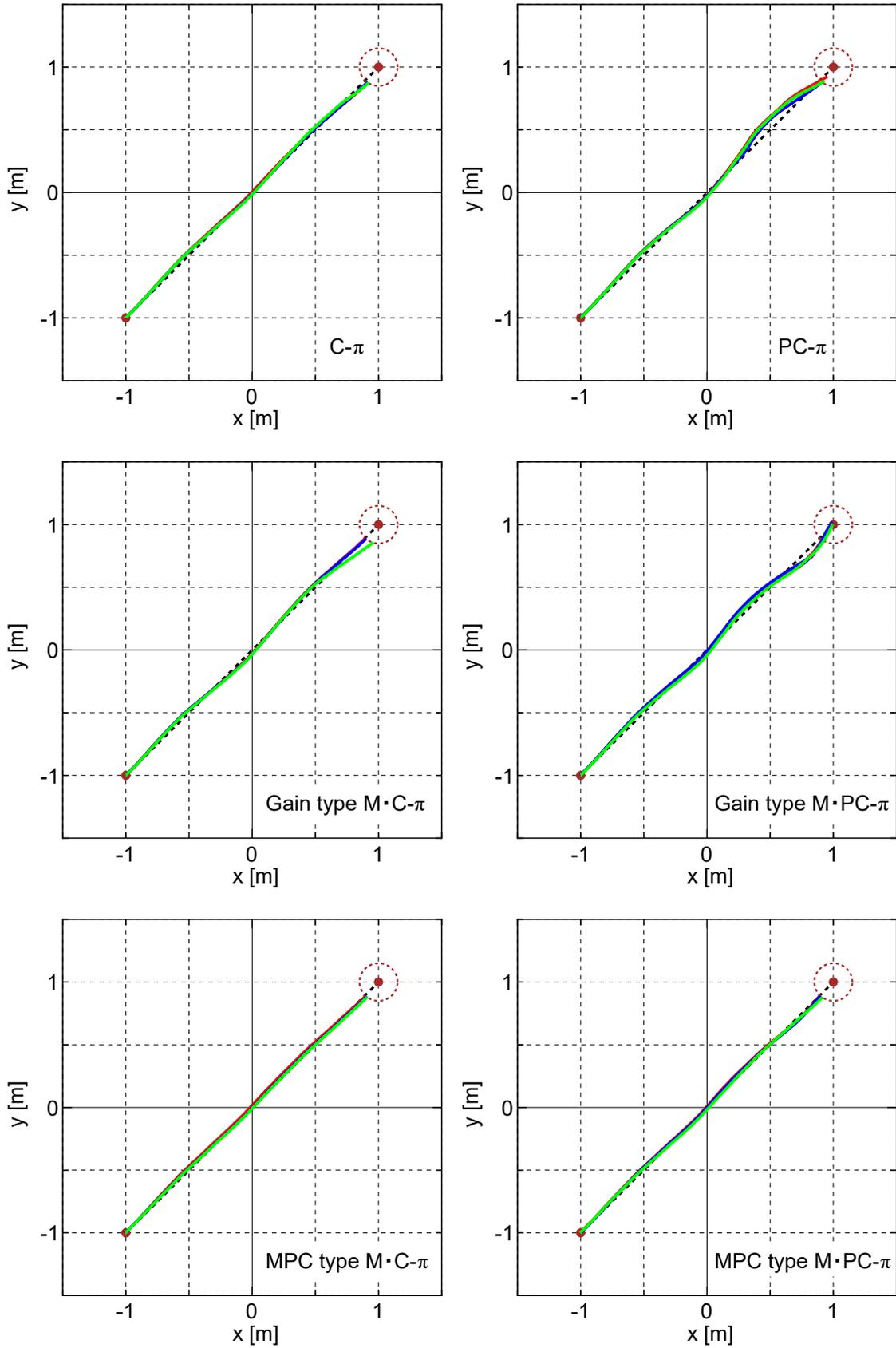


図 4.25 直線走行における実験の再現性の確認 (CASE 1,  $\tau_c=2.0$ )

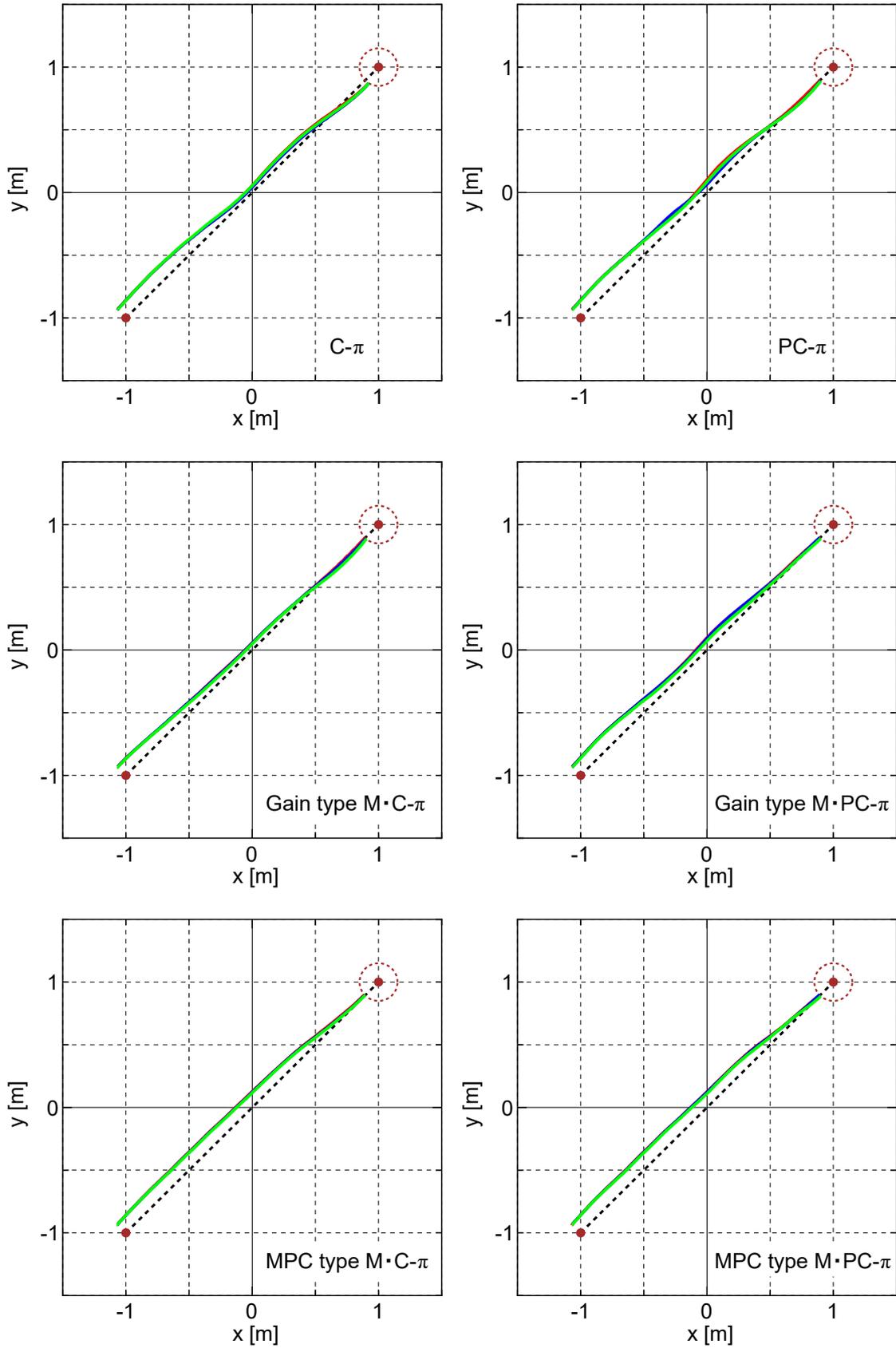


図 4.26 直線走行における実験の再現性の確認 (CASE 2,  $\tau_c=1.0$ )

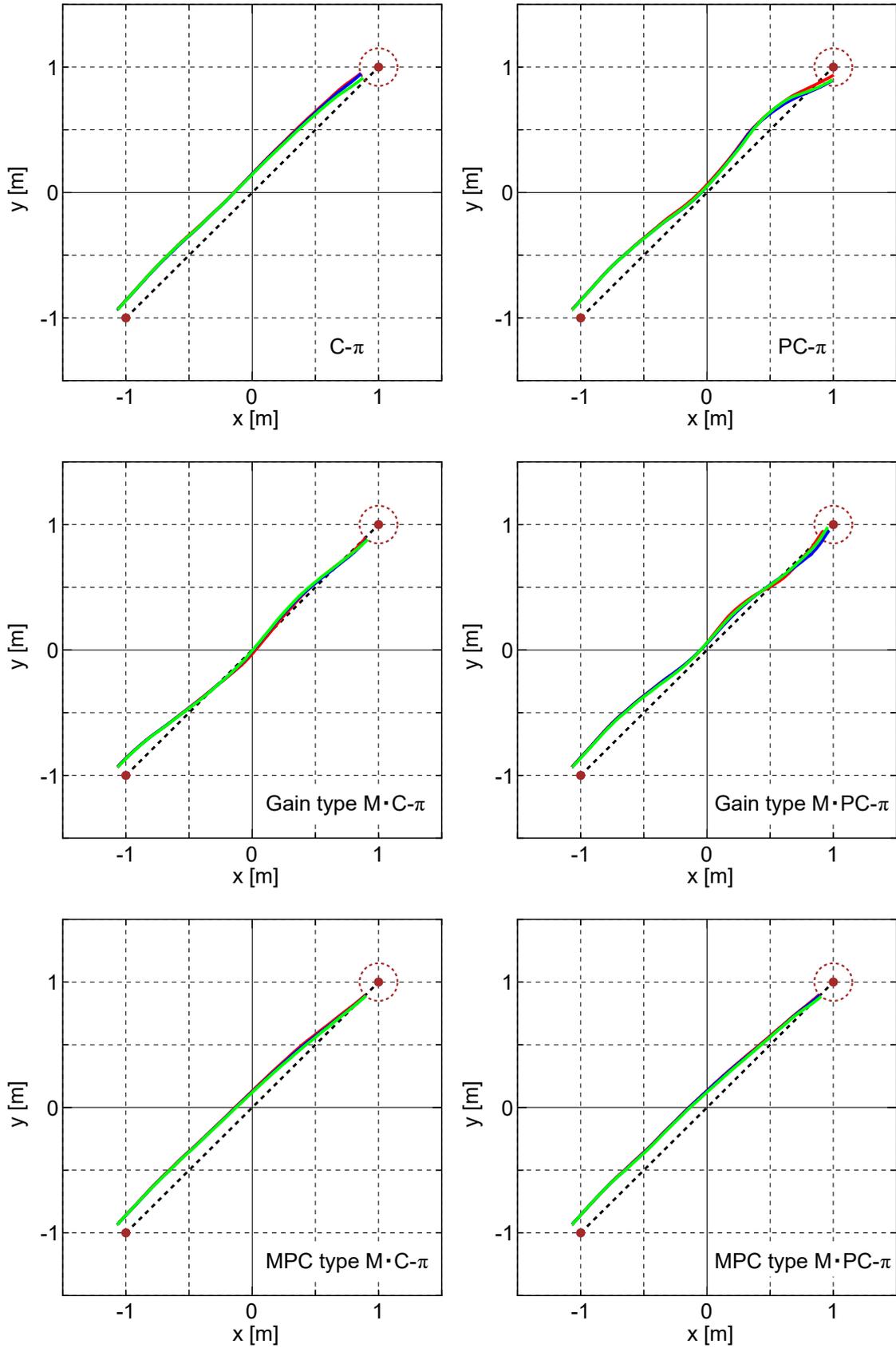


図 4.27 直線走行における実験の再現性の確認 (CASE 2,  $\tau_c=2.0$ )

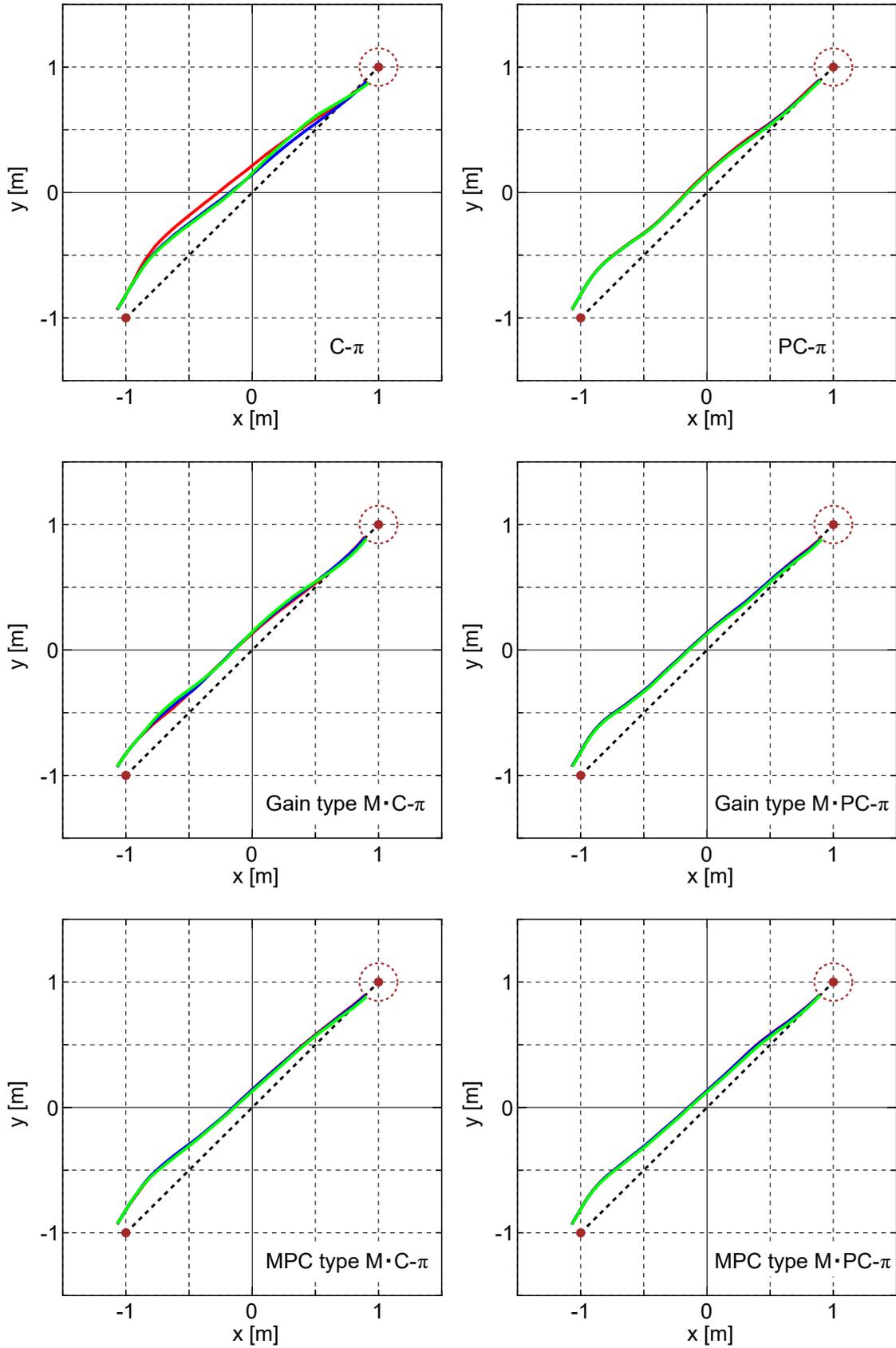


図 4.28 直線走行における実験の再現性の確認 (CASE 3,  $\tau_c=1.0$ )

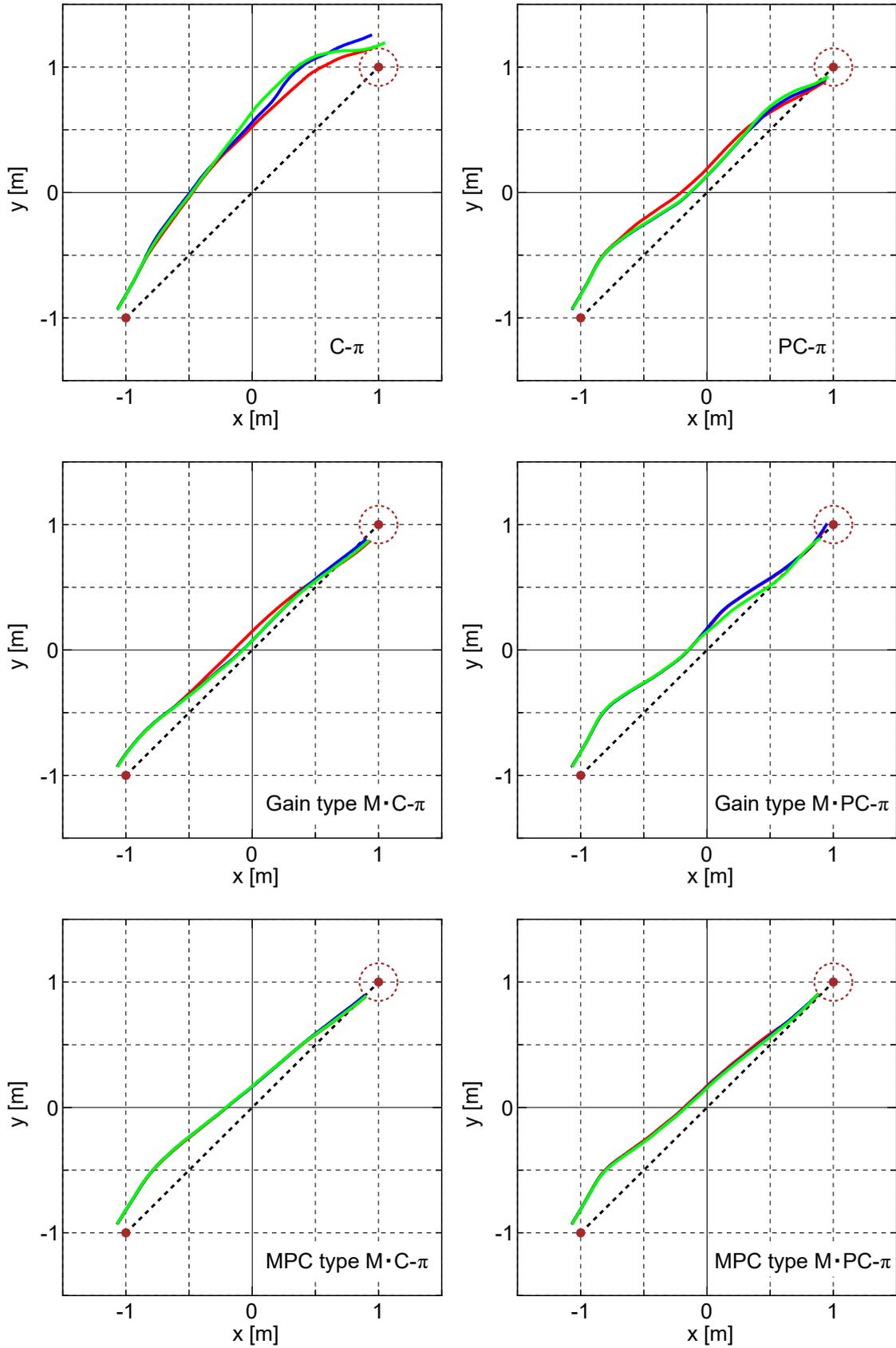
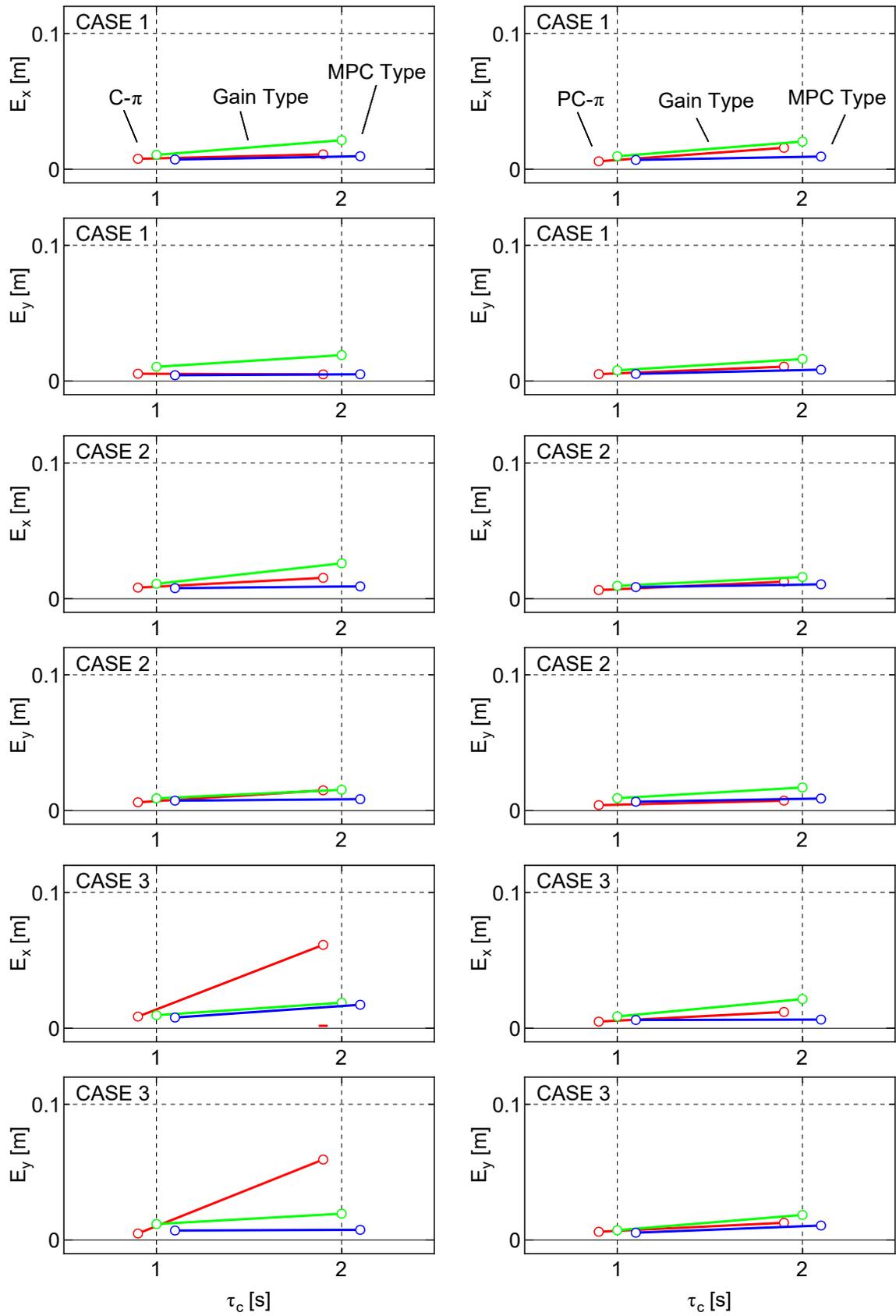


図 4.29 直線走行における実験の再現性の確認 (CASE 3,  $\tau_c=2.0$ )



(a) C- $\pi$  手法等

(b) PC- $\pi$  手法等

図 4.30 直線走行における予測軌道と実軌道の誤差

## 4.5.2 S字走行

### ・想定

図 4.31 に S 字走行における想定の概要及びオフラインの最適化計算で得られた走行軌跡を示す。S 字走行では、座標  $(-1, -1)$ ，ヨー角  $0 \text{ deg}$  の状態から座標  $(1, 1)$ ，ヨー角  $0 \text{ deg}$  の状態まで移動する。その他の状態は 0 としている。

直線走行では MGV の前方付近の目標地点に対する移動を基本的な想定としたが，次の段階としては MGV の進行方向とは異なる地点に移動することが想定される。さらに目標地点でのヨー角が指定されるならば，非ホロノミック車両は進行方向とは異なる方向への直線的な移動はできないため，左右の大きな旋回を伴って移動することが考えられる。また MGV の横方向に走行開始位置の誤差が存在すると旋回の曲率は変化し，実時間最適制御によりその誤差に対応して目標まで到達しなければならない。そのため，S 字走行では以下の 3 ケースを想定した走行実験を行う。

- ・ CASE 1 : 初期状態誤差なし
- ・ CASE 2 : 想定した初期位置の 0.1 m 右側から走行開始
- ・ CASE 3 : 想定した初期位置の 0.1 m 左側から走行開始

いずれの場合においても機動初期に用いるオフライン解は初期状態誤差なしの解を用いる。仮に CASE 2 及び CASE 3 の初期状態が既知の場合に得られる最適軌道は図 4.31(b) の赤色及び青色の実線のようにになる。

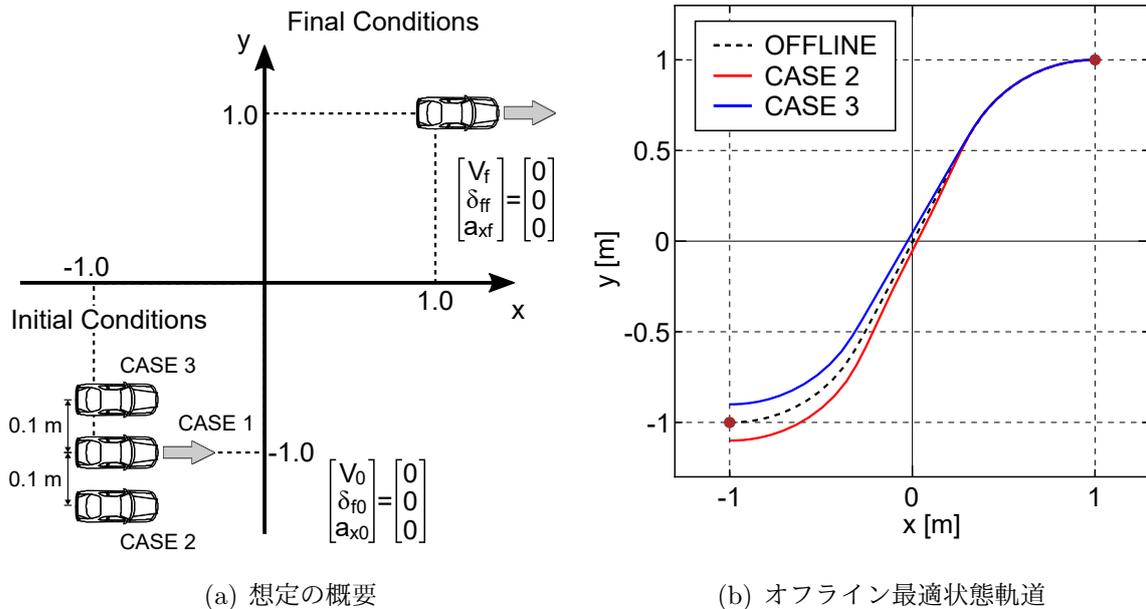


図 4.31 MGV の運動制御の想定：S 字走行

・実験結果

図 4.32 に S 字走行における全ての走行パターンと、本節で示している実験結果の関係を示す。

- ① 最適解の生成状況では、時系列で最適軌道の生成の様子を示し、S 字走行において移動しながら最適解を更新できていることを確認する。
- ② 各手法の時間履歴では、MGV の状態や実際に適用した制御入力、追従制御器により生成した制御入力の時間履歴を示し、S 字走行における各手法の制御状況を確認する。
- ③ 走行軌跡では、走行パターンごとの 3 回ずつの走行軌跡を示し、S 字走行における走行実験の再現性を確認する。
- ④ 予測軌道との誤差では、S 字走行の全ての走行パターンにおける各サンプリングタイム・各手法での予測軌道と実軌道との誤差の平均を状態別に整理し、各手法の制御性能を比較する。

図 4.33 及び 4.34 に、S 字走行、CASE 3,  $\tau_c = 1.0$  s における 2 秒ごとの C- $\pi$  手法及び PC- $\pi$  手法による最適解の生成状況を示す。黒色の破線がオフライン最適軌道、赤色の実線が実軌道、マゼンダの実線が各時刻で生成した最適状態軌道を表している。各時刻において、C- $\pi$  手法ではサンプリングした状態から、PC- $\pi$  手法では予測した状態から目標状態までの最適解が生成できている。一般的な追従制御の PD 制御や MPC を用いた場合にはオフラインで生成した S 字軌道に復帰することで初期状態誤差に対するロバスト性を保持するが、C- $\pi$  手法及び PC- $\pi$  手法では目標までの新たな S 字軌道を生成して目標まで到達している。

	$\tau_c$	C- $\pi$	PC- $\pi$	Gain型 M・C- $\pi$	Gain型 M・PC- $\pi$	MPC型 M・C- $\pi$	MPC型 M・PC- $\pi$
CASE 1 (なし)	1.0						
	2.0						
CASE 2 (外側)	1.0						
	2.0						
CASE 3 (内側)	1.0						
	2.0						

- ① 最適解の生成状況 (pp. 101 - 102)
- ② 各手法の時間履歴 (pp. 103 - 105)
- ③ 走行軌跡 (pp. 106 - 111)
- ④ 予測軌道との誤差 (p. 112)

図 4.32 S 字走行実験の走行パターンと結果の一覧

図 4.35～4.37 に、S 字走行、CASE 3、 $\tau_c = 2.0$  s における各手法の時間履歴を示す。黒色の短鎖線はオフラインでの最適制御解、赤色の実線は実軌道、マゼンダ、緑色、シアンの実線は各サンプリング時刻において生成された走行軌跡、空色の実線は追従機構により生成された制御入力を示す。

図 4.35(a) の C- $\pi$  手法を用いた場合には、各サンプリング時刻において目標までの最適制御解を生成しているが、左へ大きく迂回している。各最適制御解では最初に右旋回を行う操舵指令が得られているが、適用されない最初のサンプリング区間に含まれているため、修正を行うことができていない。走行の終盤では切り返し操作により目標へ移動する解が得られているが、この場合も同様に適用されていない。制限時間を超えたため強制的に走行実験を終了している。

図 4.36(a) 及び 4.37(a) の Gain 型 M・C- $\pi$  手法及び MPC 型 M・C- $\pi$  手法を用いた場合には、左へ迂回する傾向が発生する  $t = 4$  s 付近にて右操舵のフィードバック制御入力生成されている。そのため左へ迂回することなく目標まで移動することができる。

図 4.35(a) の C- $\pi$  手法を用いた場合には  $t = 4 \sim 5$  s 付近の入力は緩やかに左操舵から  $\delta_{fc} = 0$  deg へ変化している。一方、図 4.35(b) の PC- $\pi$  手法を用いた場合には、 $t = 4$  s までに十分なヨー角が得られることを予測し、新しく生成した最適制御解では余分な左旋回を行う入力が排除されているため、左へ迂回する傾向を抑えている。また、図 4.36(b) 及び 4.37(b) の Gain 型 M・PC- $\pi$  手法や MPC 型 M・PC- $\pi$  手法では、フィードバック制御入力を生成して最適状態軌道との誤差を修正しながら目標まで移動している。

図 4.38～4.43 に S 字走行の全走行パターンにおける走行軌跡の結果を示す。いずれの走行パターンにおいても 3 回ずつ走行を行っており、概ね同様の結果が得られている。

ここで、各サンプリング時刻で生成した最適軌道と実軌道との誤差により各制御手法の性能を比較して評価する。誤差の算出方法は第 4.4 節の数値シミュレーションの場合と同様に行う。図 4.44 に各手法、各サンプリングタイムにおいて予測軌道と実軌道との誤差の平均値を状態変数別に整理した結果を示す。S 字走行ではいずれのケースでも C- $\pi$  手法を用いた場合には、サンプリングが長くなることに伴い誤差が大きく増加している。一方で、Gain 型 M・C- $\pi$  手法や MPC 型 M・C- $\pi$  手法、プロパゲーション型のそれぞれの手法では誤差の増加が抑制されている。S 字走行においても、Gain 型や MPC 型の追従制御器を組み込むこと、または初期条件を予測するプロパゲーション型を用いることで安定性が向上することが定量的に示されている。

以上より S 字走行における走行実験で以下の結果を得た。

(1) S 字走行の各想定において、設計した各手法により最適制御解を更新しながら実機の MGV を制御することができた。

(2) Gain 型や MPC 型の追従制御器を組み込むこと，または初期条件を予測するプロパゲーション型を用いることで安定性が向上することを示した。

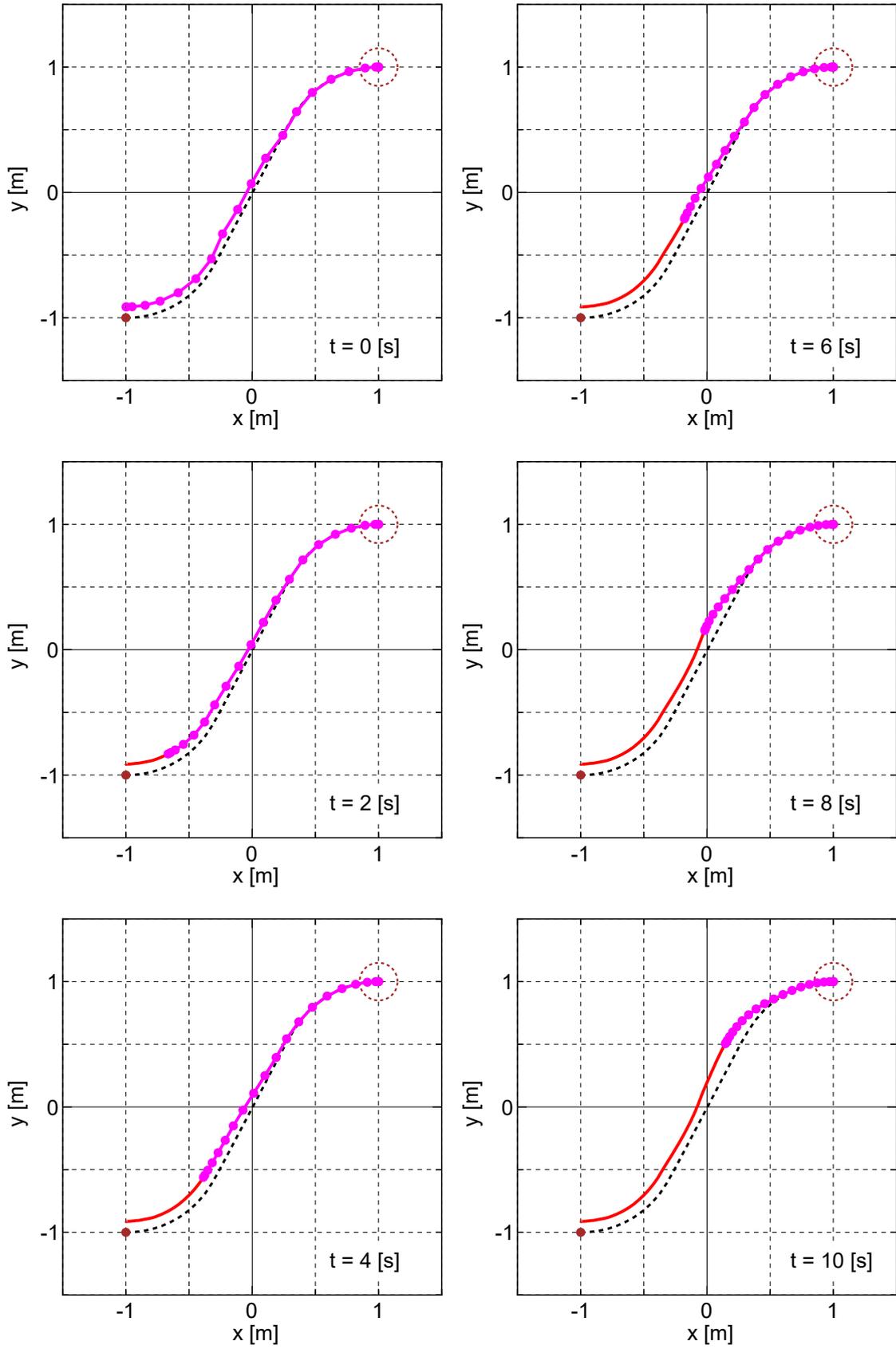


図 4.33 S 字走行における最適解の生成状況 (C- $\pi$  手法, CASE 3,  $\tau_c=1.0$ )

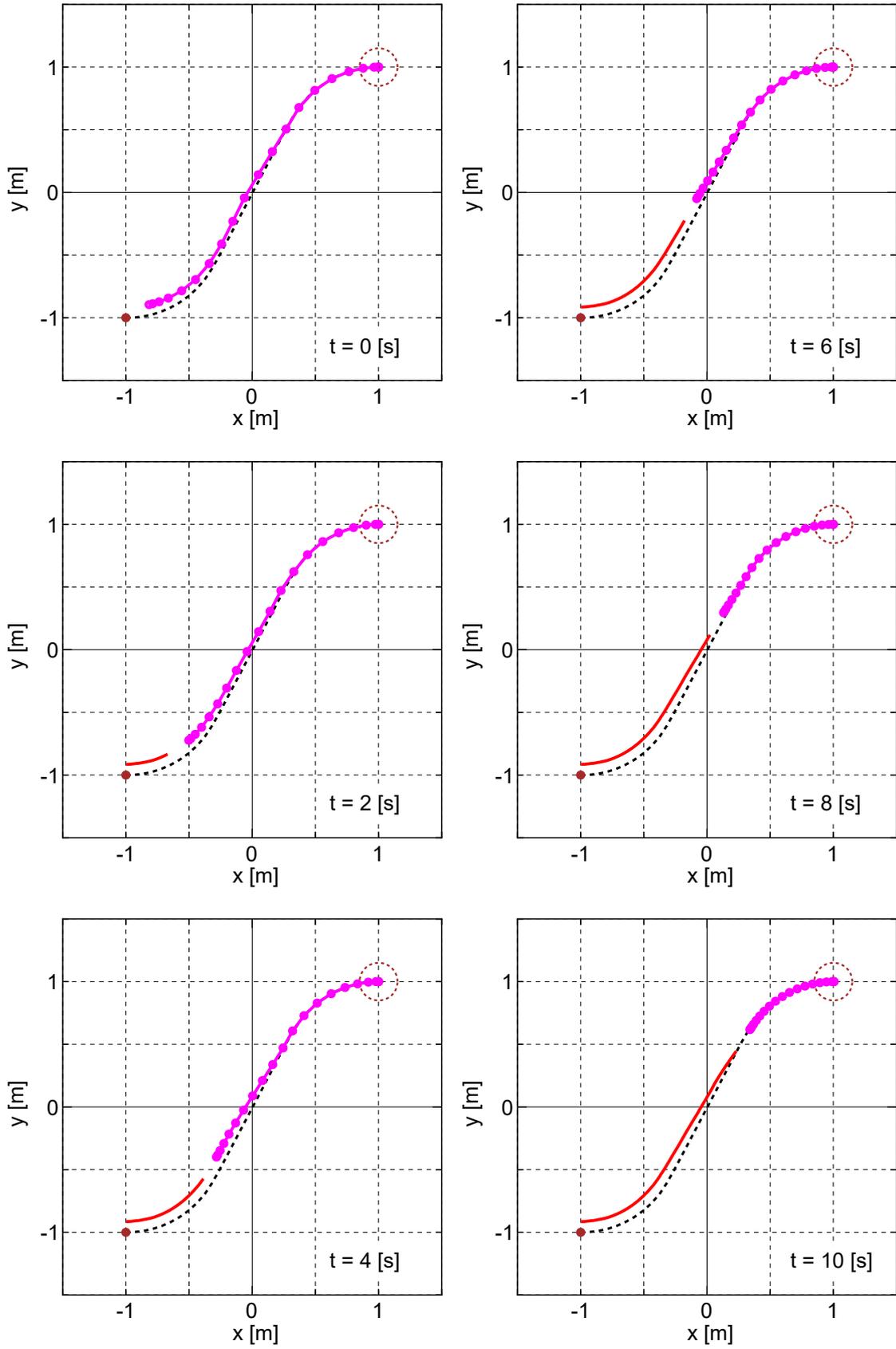


図 4.34 S字走行における最適解の生成状況 (PC- $\pi$ 手法, CASE 3,  $\tau_c=1.0$ )

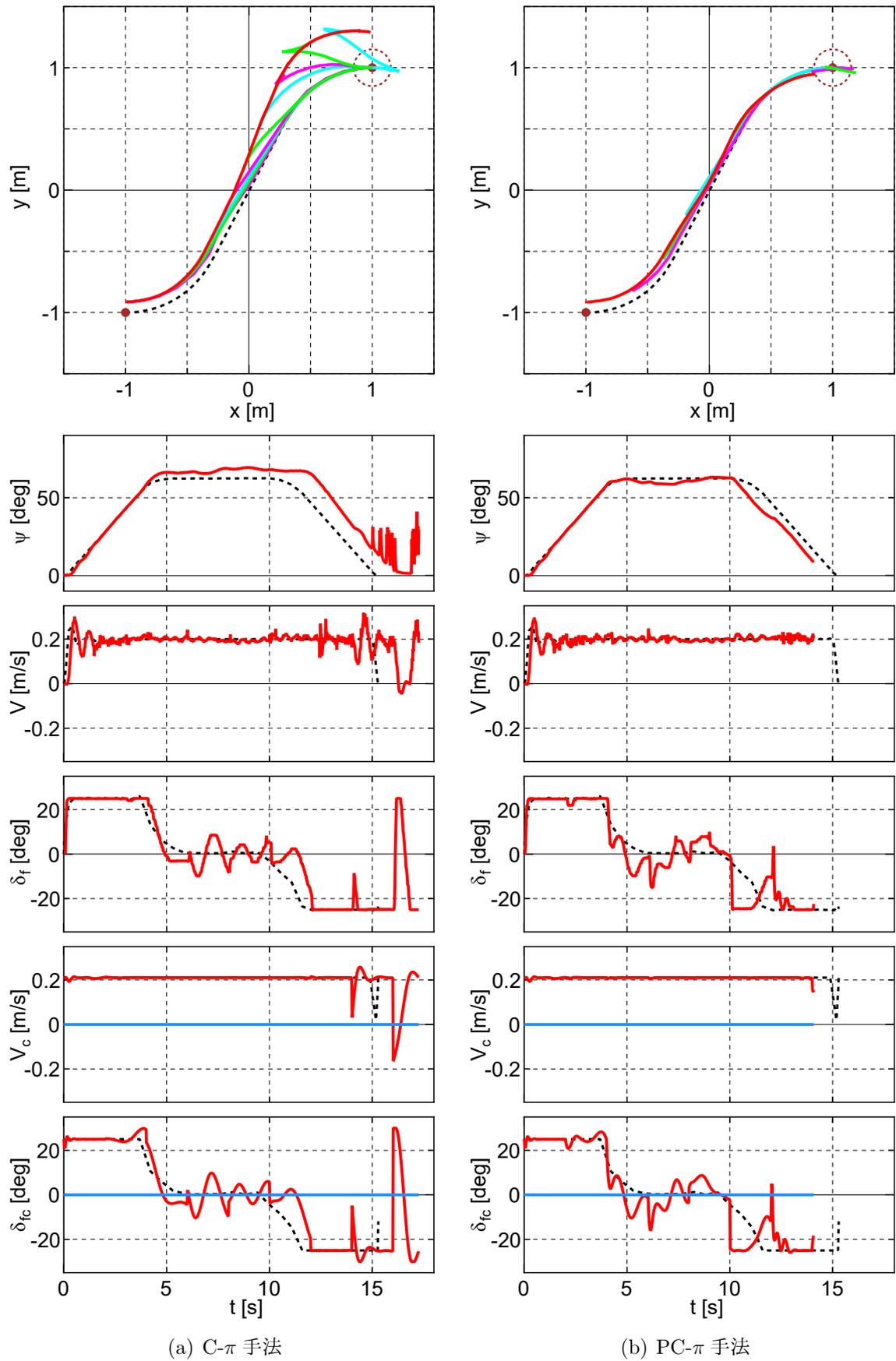


図 4.35 S字走行における走行軌跡と各変数の時間履歴 (CASE 3,  $\tau_c = 2.0$ , その1)

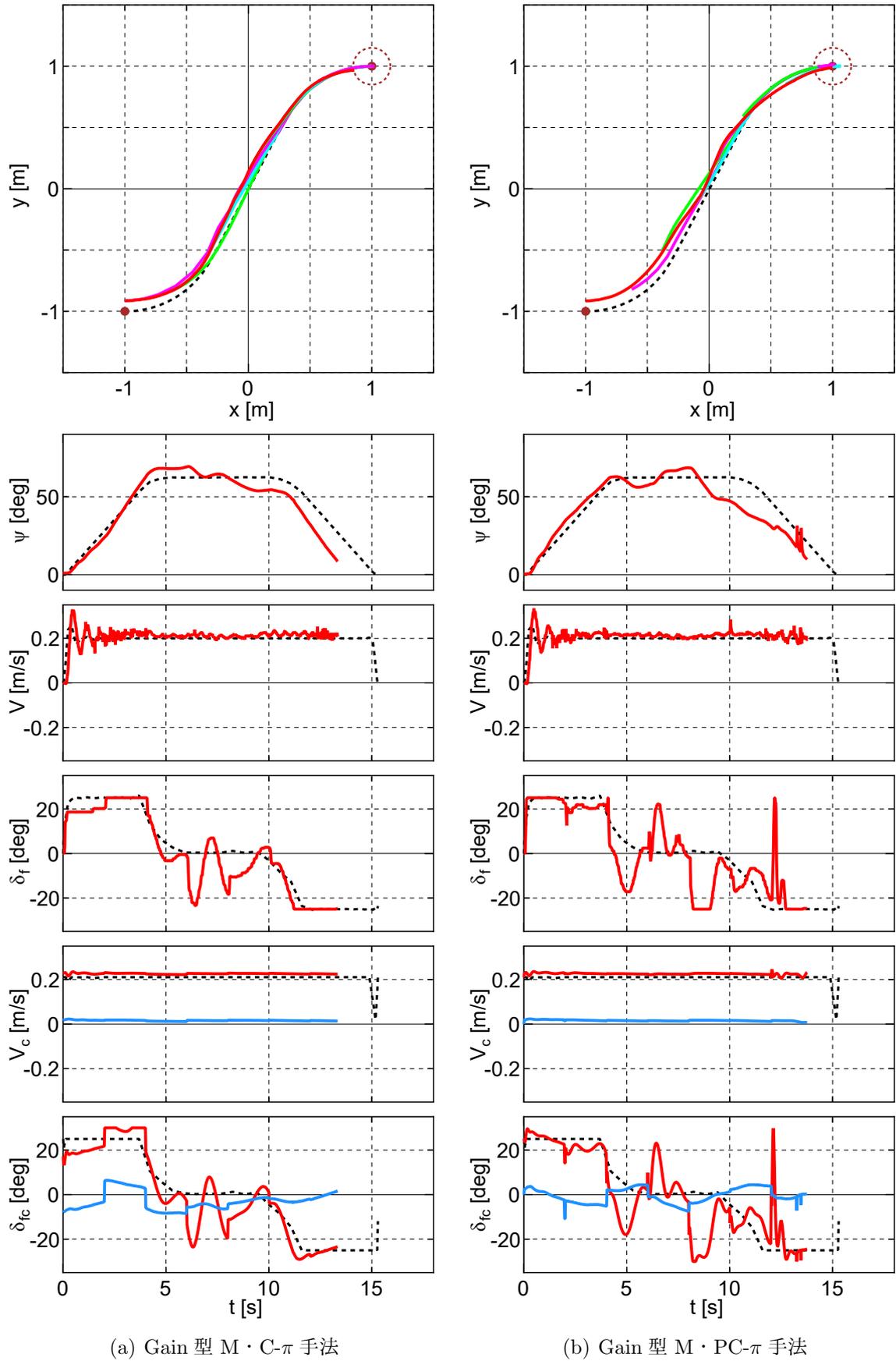
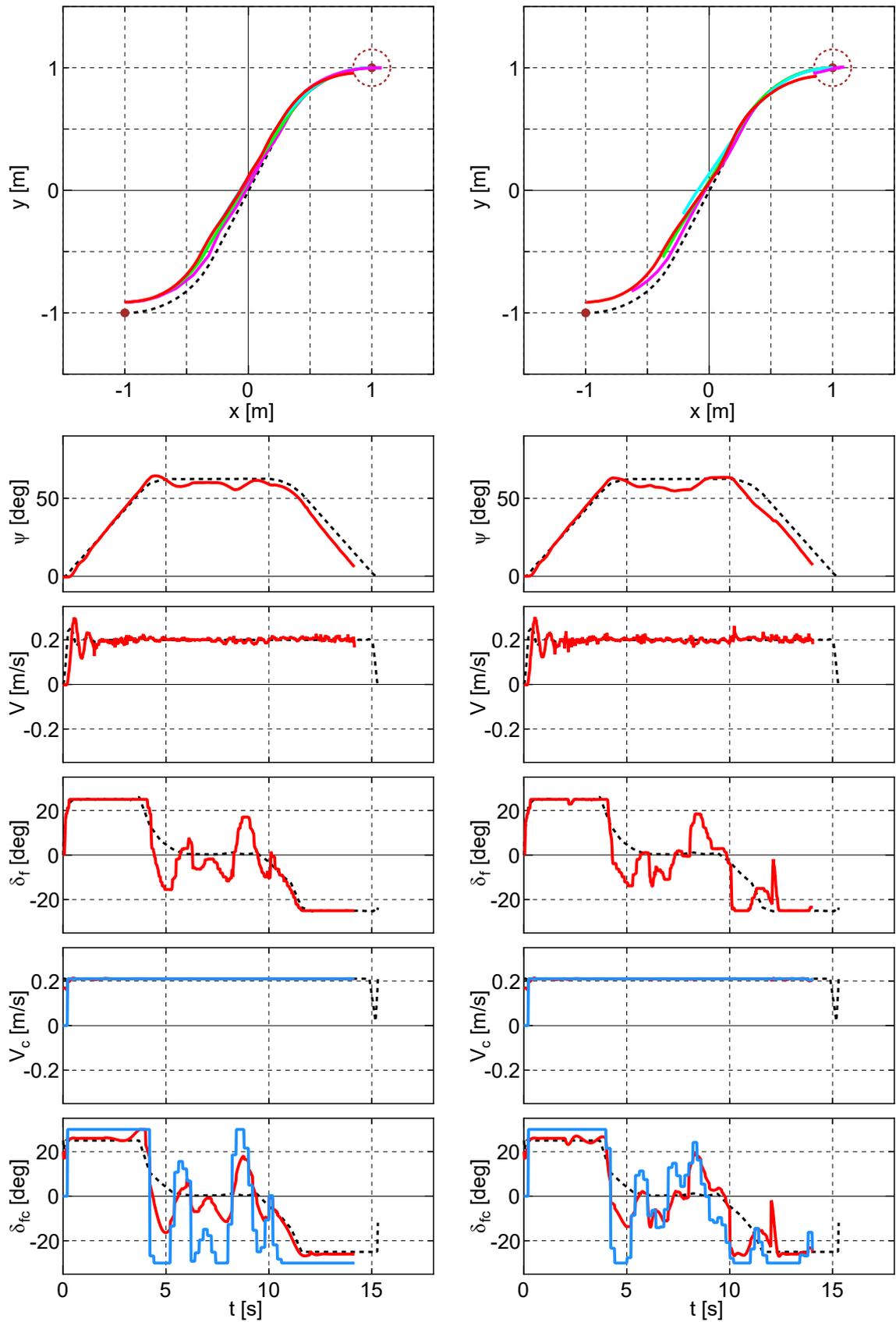


図 4.36 S 字走行における走行軌跡と各変数の時間履歴 (CASE 3,  $\tau_c = 2.0$ , その 2)



(a) MPC 型 M · C- $\pi$  手法

(b) MPC 型 M · PC- $\pi$  手法

図 4.37 S 字走行における走行軌跡と各変数の時間履歴 (CASE 3,  $\tau_c = 2.0$ , その 3)

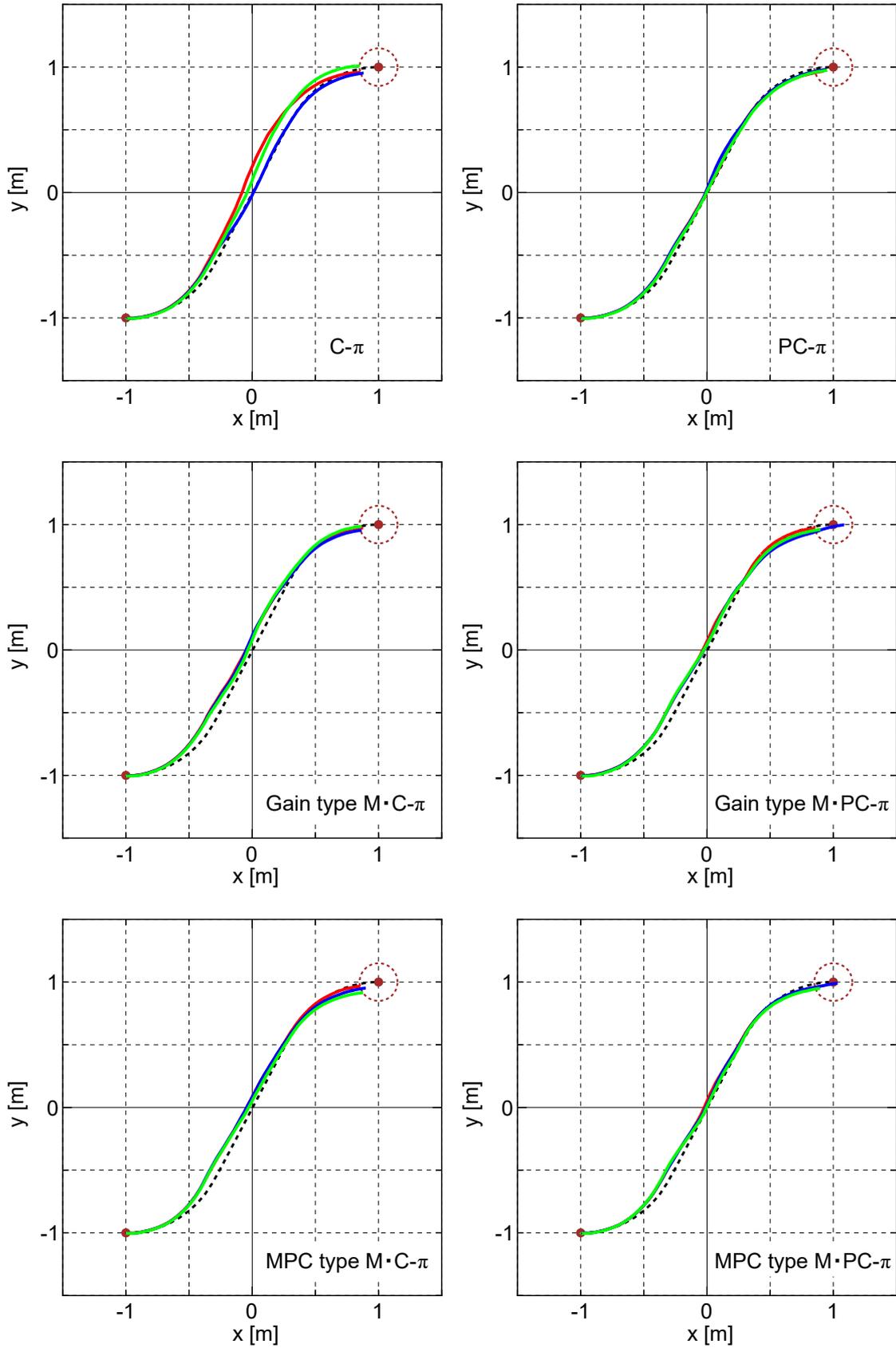


図 4.38 S 字走行における実験の再現性の確認 (CASE 1,  $\tau_c=1.0$ )

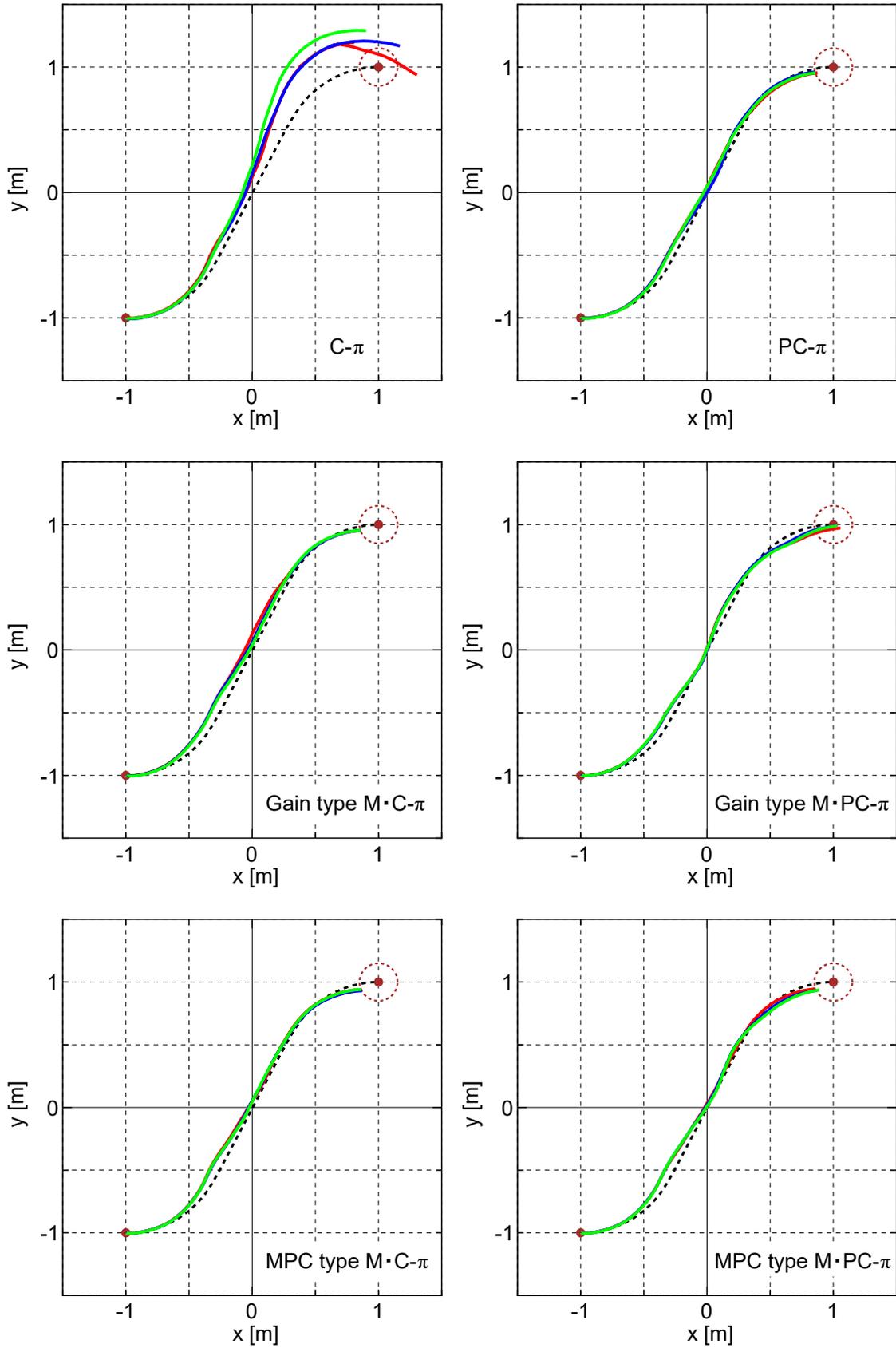


図 4.39 S字走行における実験の再現性の確認 (CASE 1,  $\tau_c=2.0$ )

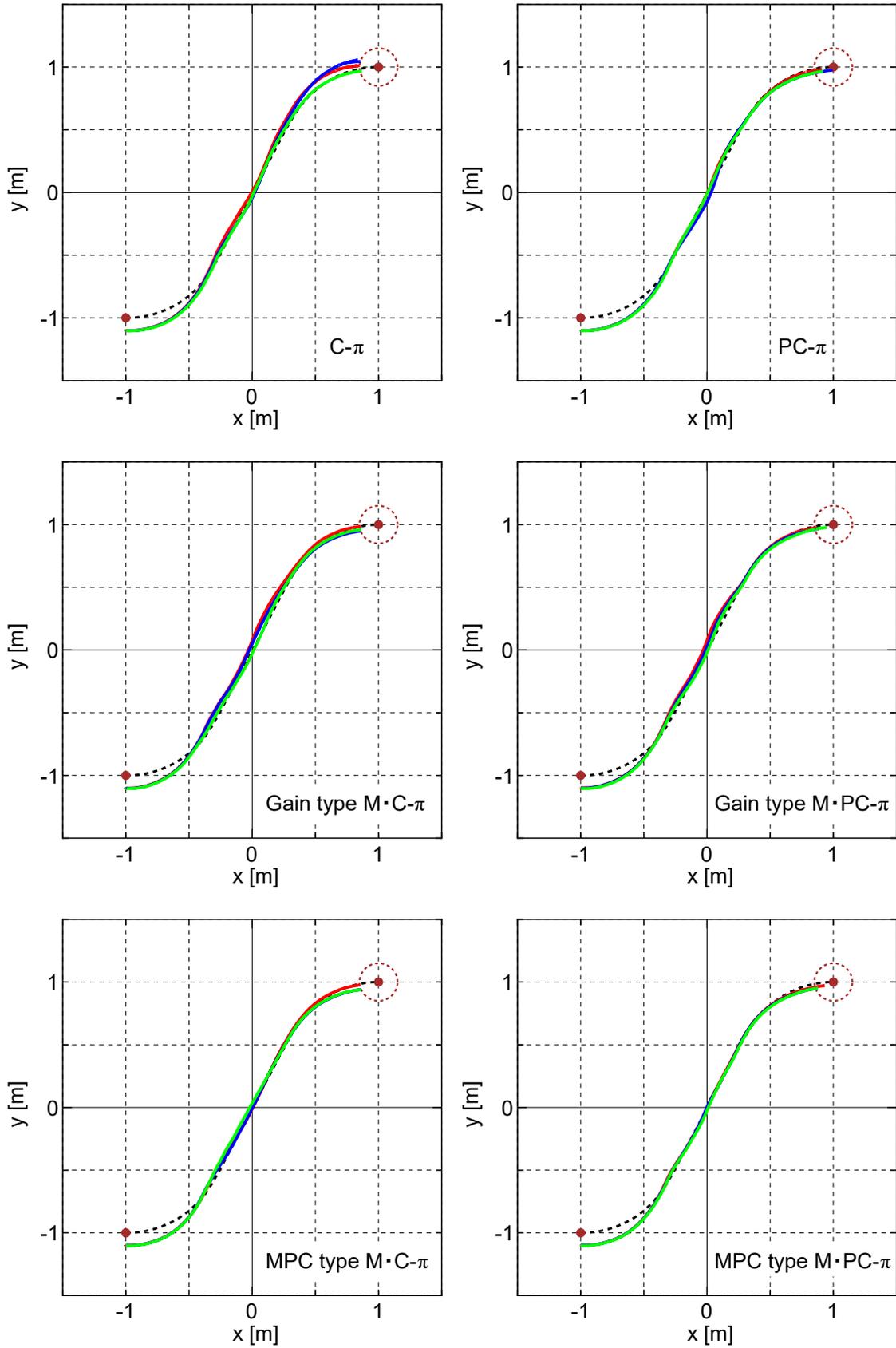


図 4.40 S字走行における実験の再現性の確認 (CASE 2,  $\tau_c=1.0$ )

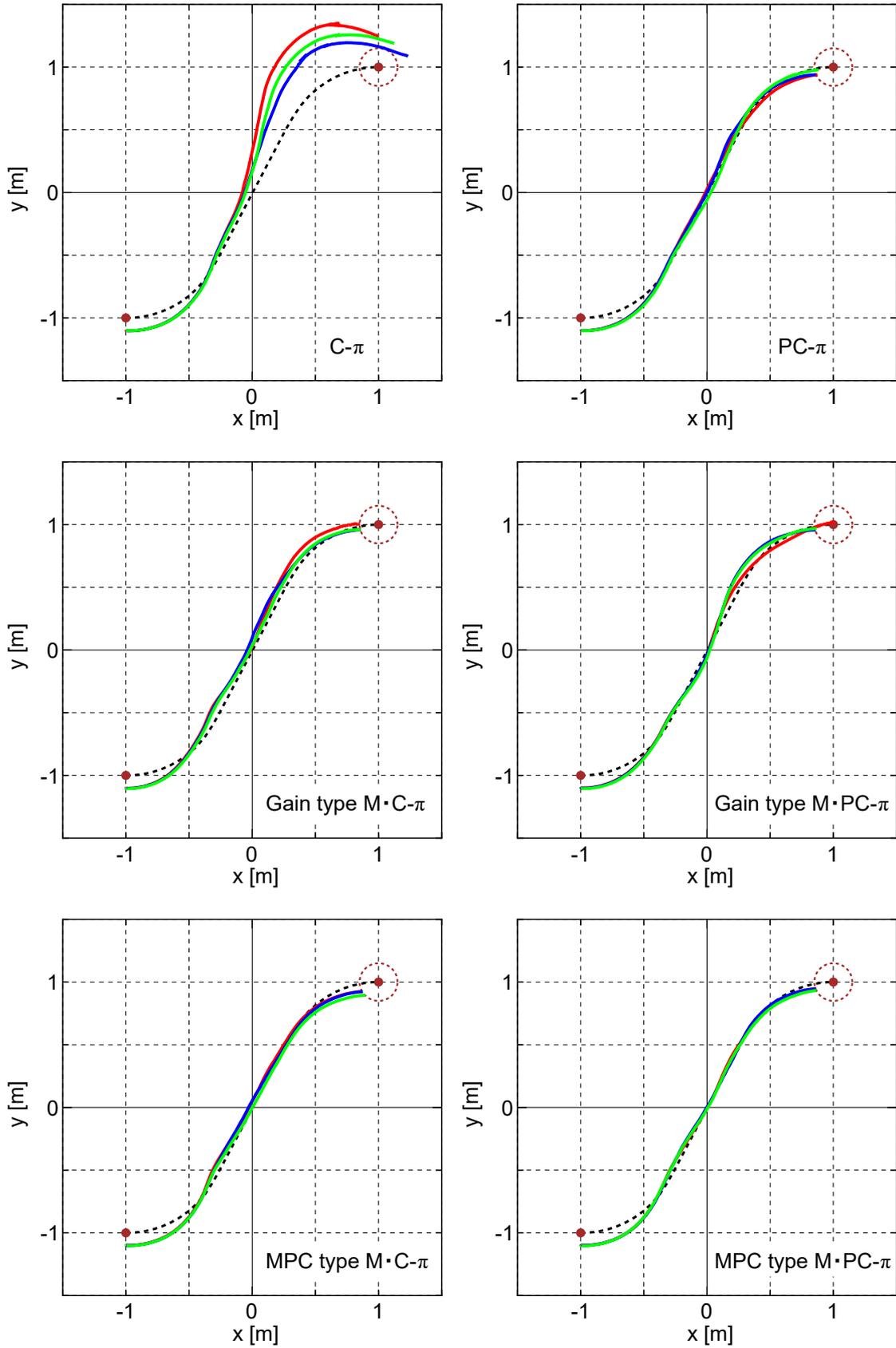


図 4.41 S 字走行における実験の再現性の確認 (CASE 2,  $\tau_c=2.0$ )

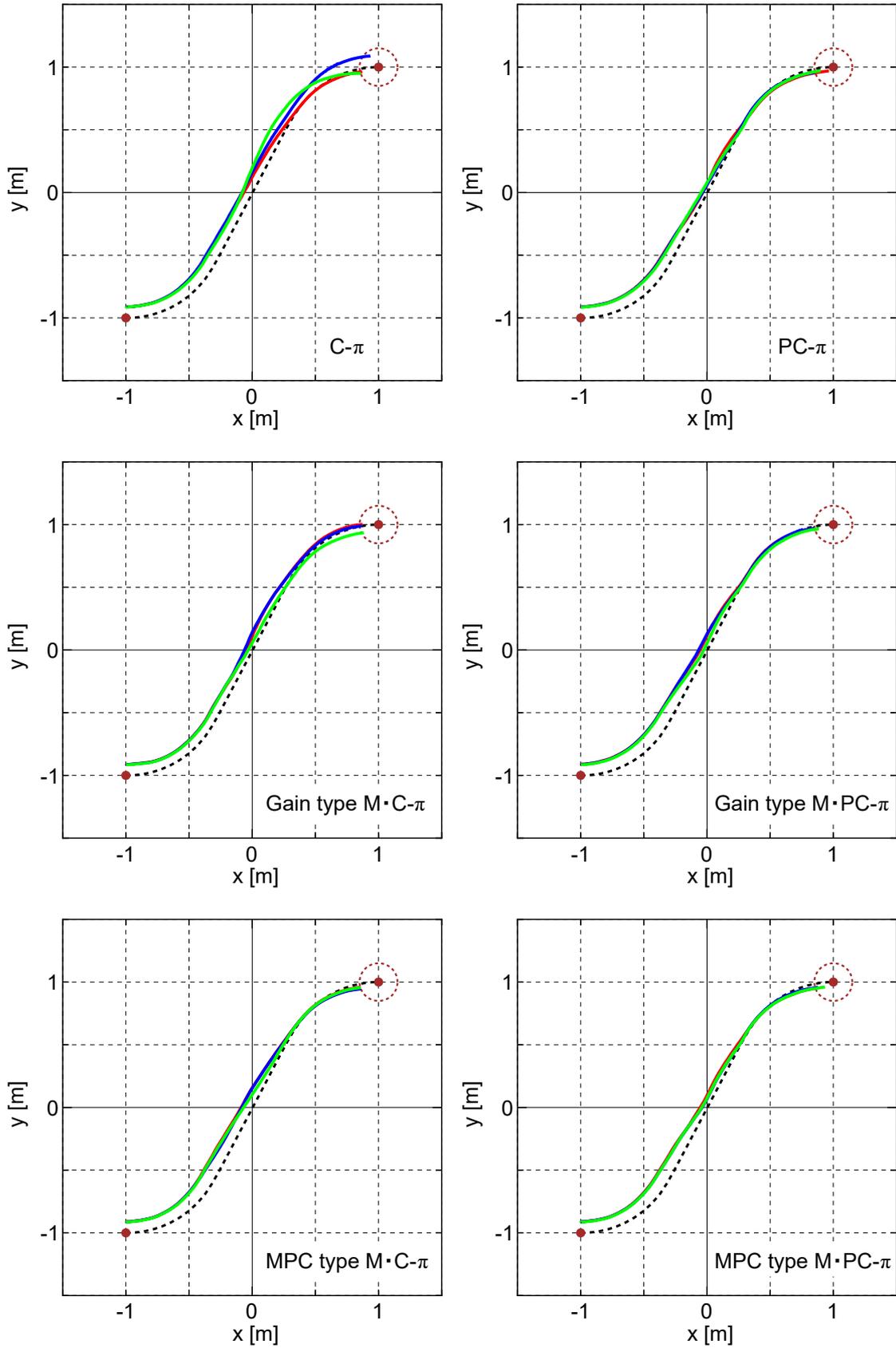


図 4.42 S字走行における実験の再現性の確認 (CASE 3,  $\tau_c=1.0$ )

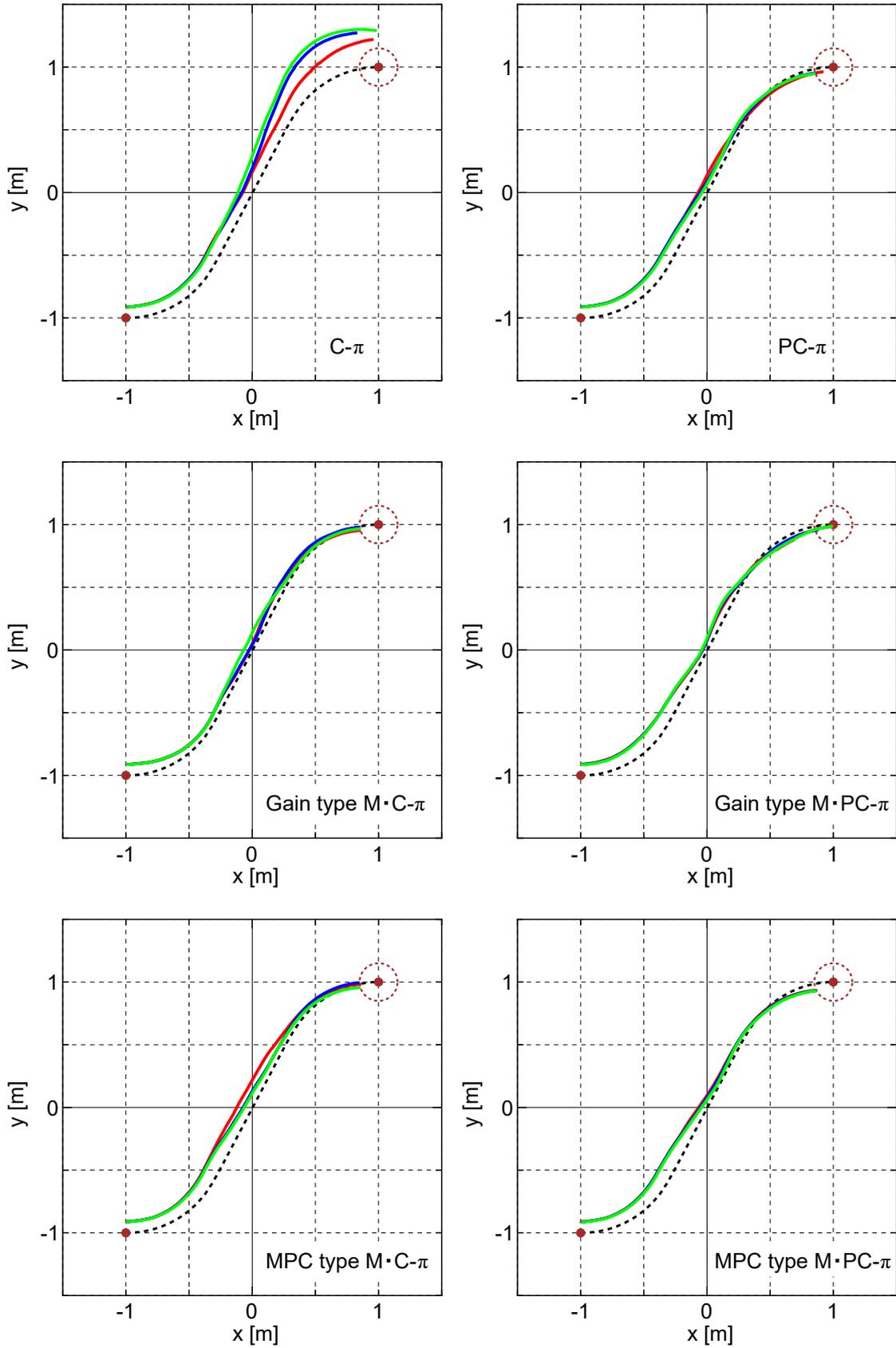
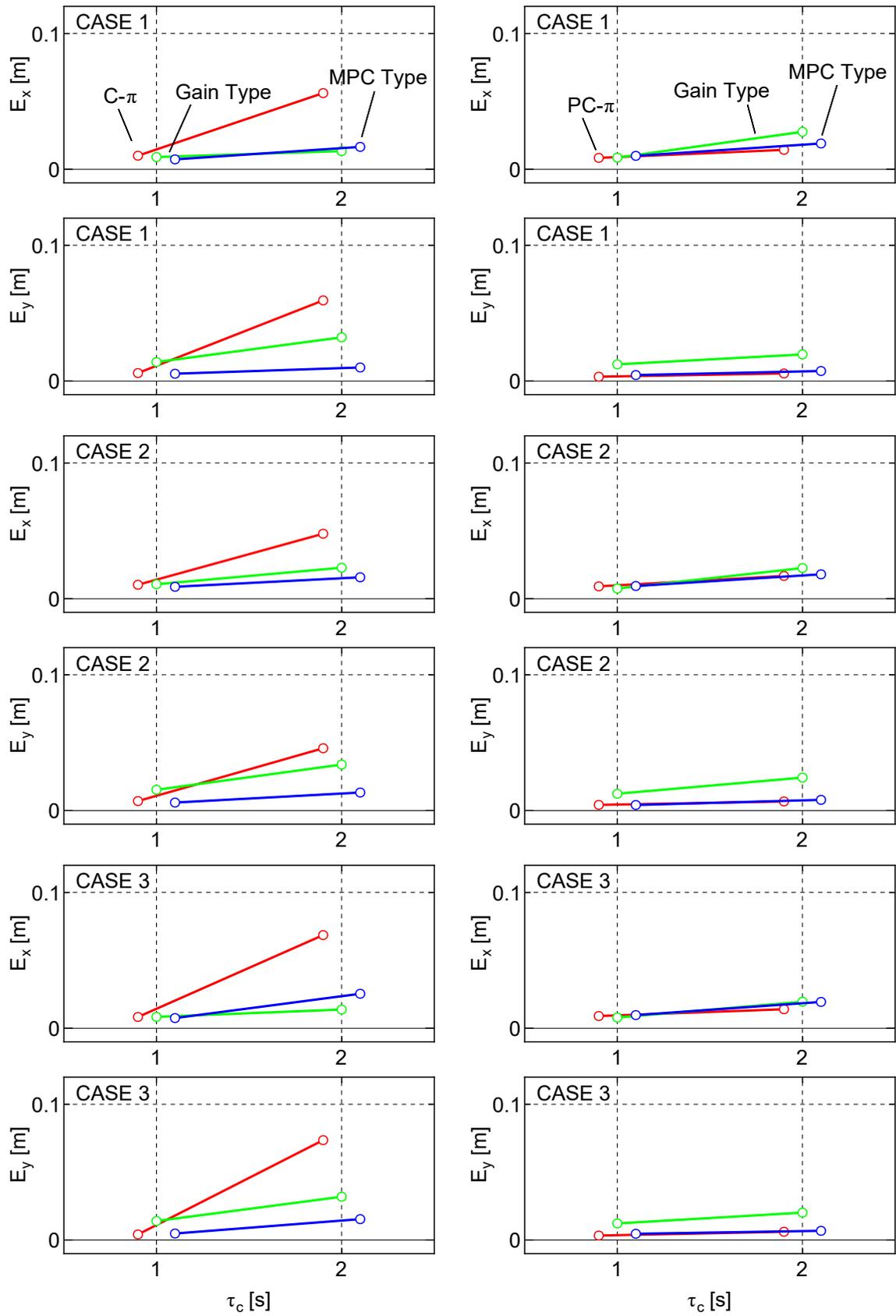


図 4.43 S字走行における実験の再現性の確認 (CASE 3,  $\tau_c=2.0$ )



(a) C- $\pi$  手法等

(b) PC- $\pi$  手法等

図 4.44 S 字走行における予測軌道と実軌道の誤差

### 4.5.3 切り返し走行

・想定

図 4.45 に切り返し走行における想定の概要及びオフラインの最適化計算で得られた走行軌跡を示す。切り返し走行では、座標  $(0, -0.3)$ 、ヨー角  $0 \text{ deg}$  の状態から座標  $(0, 0.3)$ 、ヨー角  $180 \text{ deg}$  の状態まで移動する。その他の状態は  $0$  としている。

MGV を運用する際に限られた空間において逆方向へ転換することは想定される。しかし、MGV の横方向で旋回半径以内の地点に対しては単純な旋回のみでは目標まで到達することができないため、前進や後退を組み合わせた切り返し操作を行う必要がある。また MGV の横方向に走行開始位置の誤差が存在すると途中の切り返し点などが変わることが考えられ、実時間最適制御によりその誤差に対応して目標まで到達しなければならない。そのため、切り返し走行では以下の 3 ケースを想定した走行実験を行う。

- ・ CASE 1 : 初期状態誤差なし
- ・ CASE 2 : 想定した初期位置の 0.1 m 右側から走行開始
- ・ CASE 3 : 想定した初期位置の 0.1 m 左側から走行開始

いずれの場合においても機動初期に用いるオフライン解は初期状態誤差なしの解を用いる。仮に CASE 2 及び CASE 3 の初期状態が既知の場合に得られる最適軌道は図 4.45(b) の赤色及び青色の実線のようにになる。

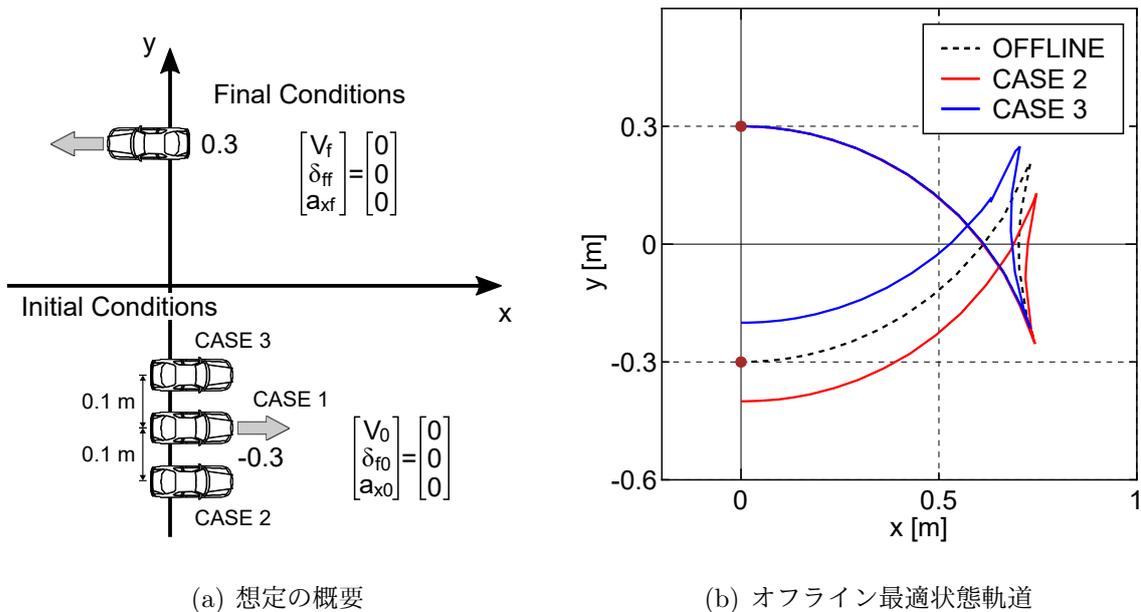


図 4.45 MGV の運動制御の想定：切り返し走行

・実験結果

図 4.46 に繰り返し走行における全ての走行パターンと、本節で示している実験結果の関係を示す。

- ① 最適解の生成状況では、時系列で最適軌道の生成の様子を示し、繰り返し走行において移動しながら最適解を更新できていることを確認する。
- ② 各手法の時間履歴では、MGV の状態や実際に適用した制御入力、追従制御器により生成した制御入力の時間履歴を示し、繰り返し走行における各手法の制御状況を確認する。
- ③ 走行軌跡では、走行パターンごとの3回ずつの走行軌跡を示し、繰り返し走行における走行実験の再現性を確認する。
- ④ 予測軌道との誤差では、繰り返し走行の全ての走行パターンにおける各サンプリングタイム・各手法での予測軌道と実軌道との誤差の平均を状態別に整理し、各手法の制御性能を比較する。

図 4.47 及び 4.48 に、繰り返し走行，CASE 3， $\tau_c = 1.0$  s における 2 秒ごとの C- $\pi$  手法及び PC- $\pi$  手法による最適解の生成状況を示す。黒色の破線がオフライン最適軌道，赤色の実線が実軌道，マゼンダの実線が各時刻で生成した最適状態軌道を表している。各時刻において，C- $\pi$  手法ではサンプリングした状態から，PC- $\pi$  手法では予測した状態から目標状態までの最適解が生成できている。一般的な追従制御の PD 制御や MPC を用いた場合ではオフラインで生成した繰り返し軌道に復帰することで初期状態誤差に対するロバスト性を保持するが，C- $\pi$  手法及び PC- $\pi$  手法では目標までの新たな繰り返し軌道を生成して目標まで到達している。

	$\tau_c$	C- $\pi$	PC- $\pi$	Gain型 M・C- $\pi$	Gain型 M・PC- $\pi$	MPC型 M・C- $\pi$	MPC型 M・PC- $\pi$
CASE 1 (なし)	1.0						
	2.0						
CASE 2 (外側)	1.0						
	2.0						
CASE 3 (内側)	1.0						
	2.0						

① 最適解の生成状況  
(pp. 116 - 117)

② 各手法の時間履歴  
(pp. 118 - 120)

③ 走行軌跡 (pp. 121 - 126)  
④ 予測軌道との誤差 (p. 127)

図 4.46 繰り返し走行実験の走行パターンと結果の一覧

図 4.49～4.51 に、繰り返し走行，CASE 3， $\tau_c = 2.0$  s における各手法の時間履歴を示す．黒色の短鎖線はオフラインでの最適制御解，赤色の実線は実軌道，マゼンダ，緑色，シアンの実線は各サンプリング時刻において生成された走行軌跡，空色の実線は追従機構により生成された制御入力を示す．

図 4.49(a) の C- $\pi$  手法を用いた場合では，まず  $t = 0$  s において内側の初期位置誤差を初期条件として最適解を生成している．オフライン解と同じく，左旋回で前進，右旋回で後退，左旋回で前進することにより目標まで到達する解となっている．オフラインの最適解と  $t = 0$  s で生成した解はどちらも最大操舵角・最大速度で移動する解であり，移動開始直後から更新した軌道に沿って移動することができているため，目標まで到達することができている．他の手法においても同様の結果が得られた．

図 4.52～4.57 に繰り返し走行の全走行パターンにおける走行軌跡の結果を示す．いずれの走行パターンにおいても 3 回ずつ走行を行っており，直線走行や S 字走行と比較するとばらつきが見られるが，概ね同様の結果が得られている．

ここで，各サンプリング時刻で生成した最適軌道と実軌道との誤差により各制御手法の性能を比較して評価する．誤差の算出方法は第 4.4 節の数値シミュレーションの場合と同様に行う．

図 4.58 に各手法，各サンプリングタイムにおいて予測軌道と実軌道との誤差の平均値を状態変数別に整理した結果を示す．いずれの手法もサンプリングタイムが長くなることに伴う誤差が増大する傾向はなく，同等の結果が得られた．直線走行及び S 字走行では直線部分の走行が含まれるが，想定した繰り返し走行では左右の最大操舵角の組み合わせによる曲線をつなぐ走行となるため，各手法の差が小さくなっている．

以上より繰り返し走行における走行実験で以下の結果を得た．

- (1) 繰り返し走行の各想定において，設計した各手法により最適制御解を更新しながら実機の MGV を制御することができた．
- (2) 手法による大きな差異はなく，各手法・各サンプリングタイムにおいて目標まで到達することができた．

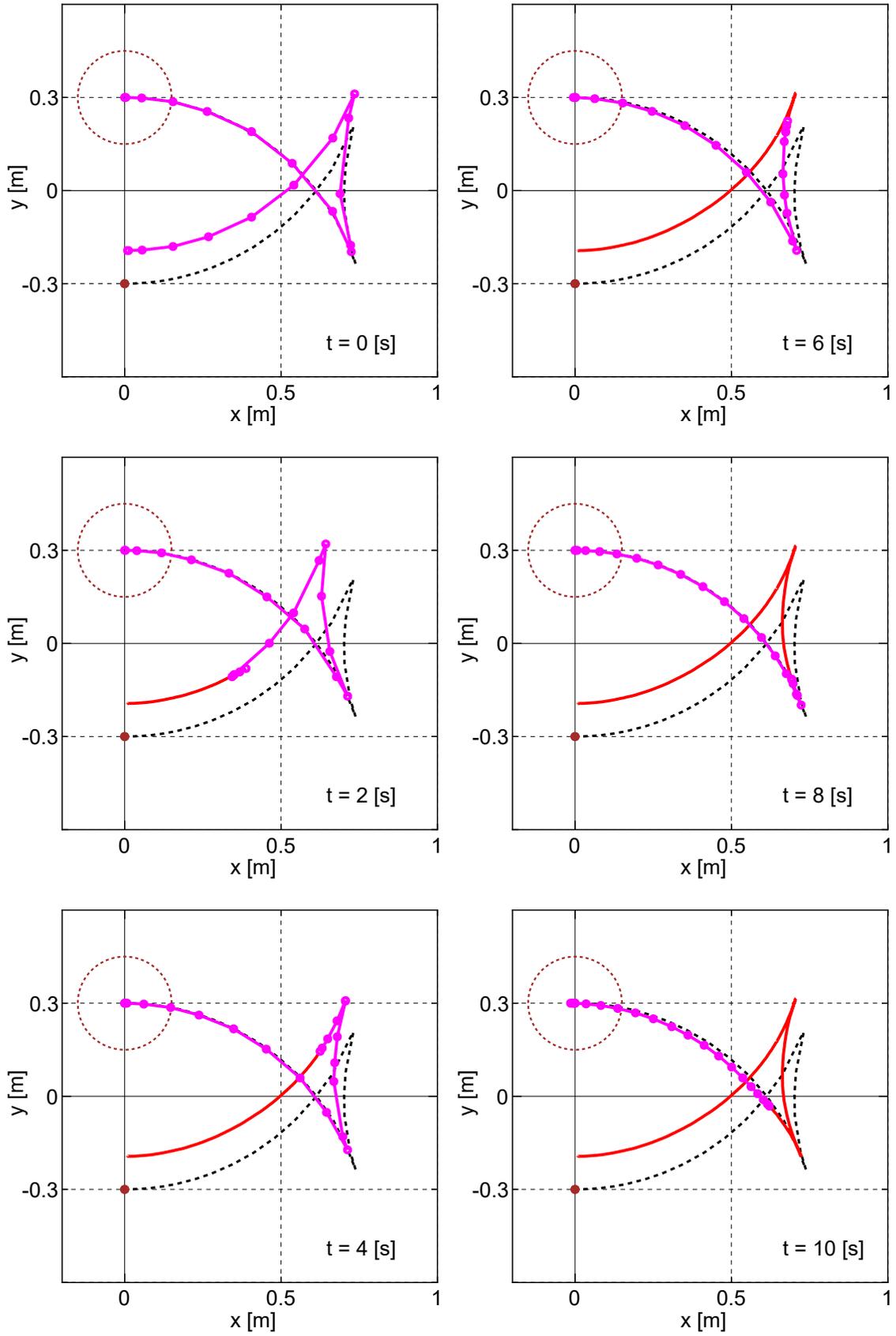


図 4.47 繰り返し走行における最適解の生成状況 (C- $\pi$ 手法, CASE 3,  $\tau_c=1.0$ )

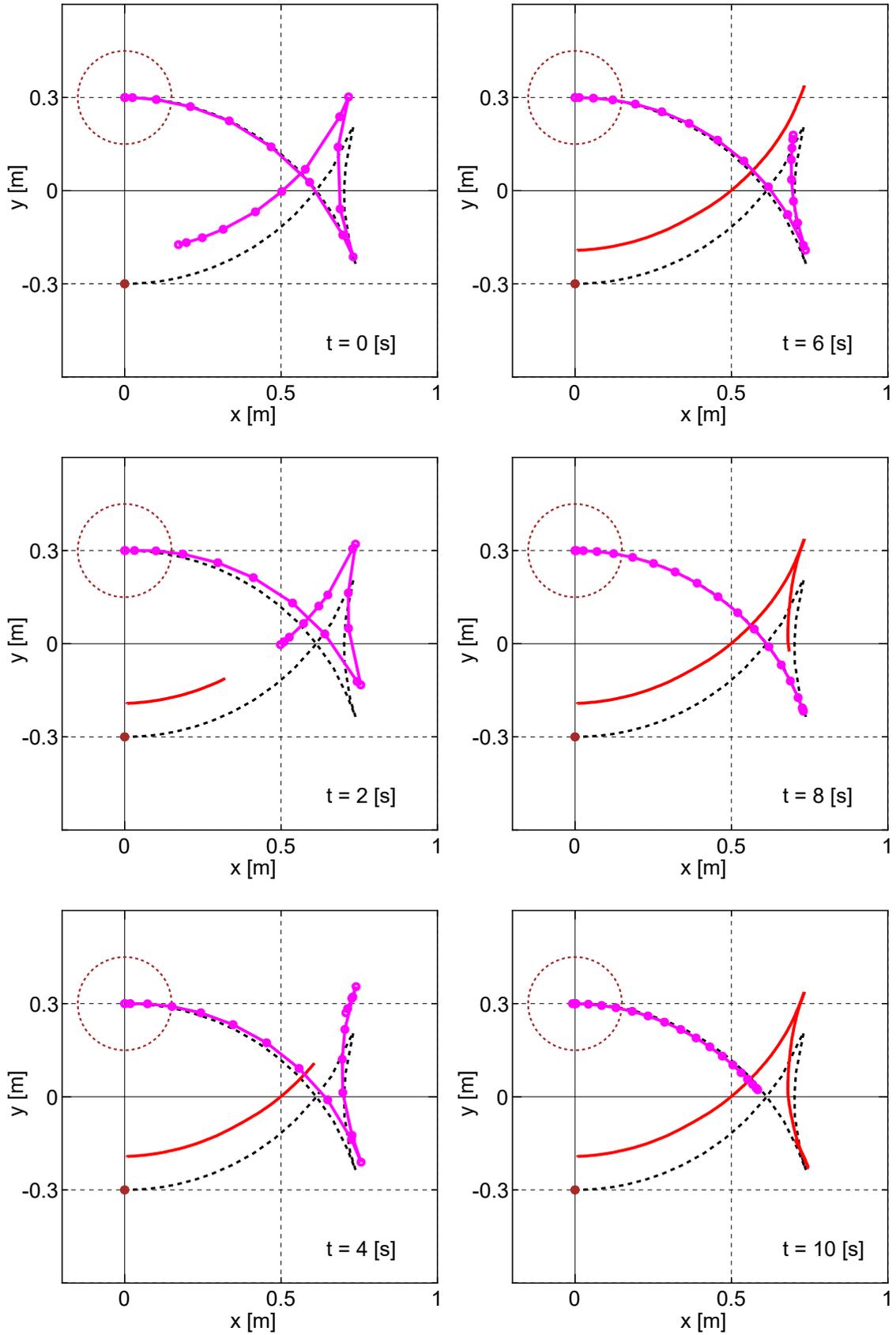


図 4.48 繰り返し走行における最適解の生成状況 (PC- $\pi$ 手法, CASE 3,  $\tau_c=1.0$ )

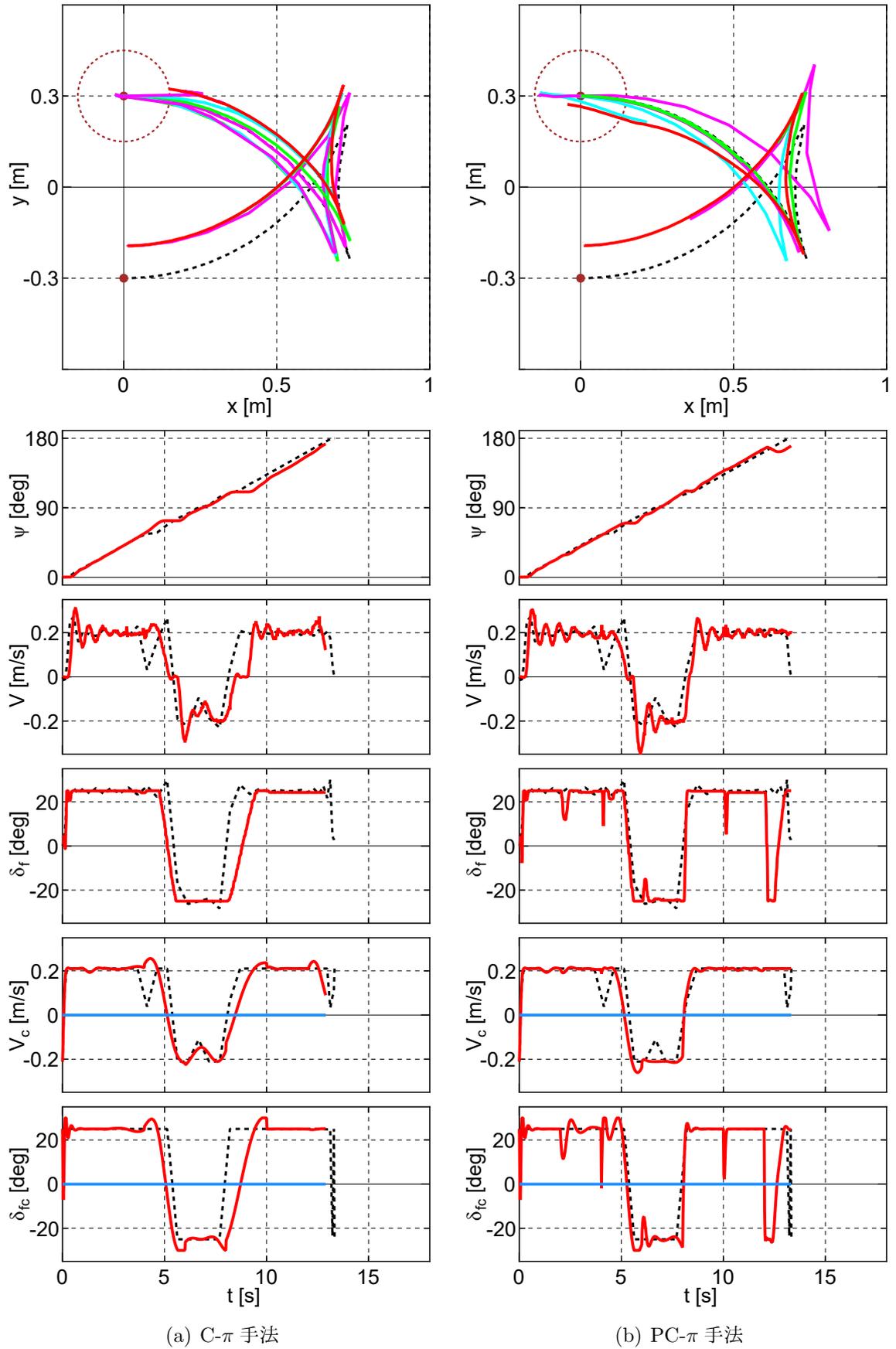


図 4.49 繰り返し走行における走行軌跡と各変数の時間履歴 (CASE 3,  $\tau_c = 2.0$ , その 1)

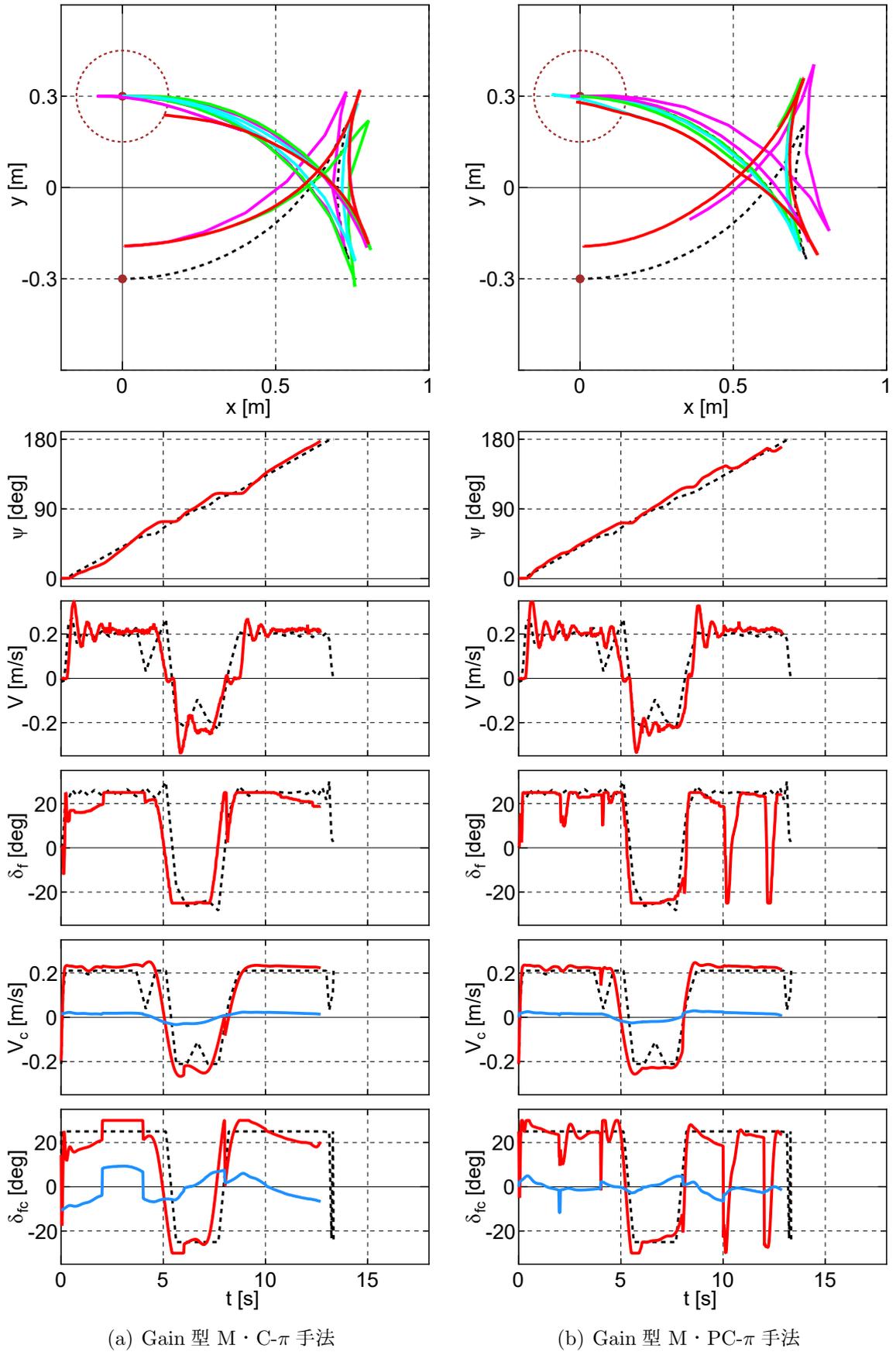


図 4.50 繰り返し走行における走行軌跡と各変数の時間履歴 (CASE 3,  $\tau_c = 2.0$ , その 2)

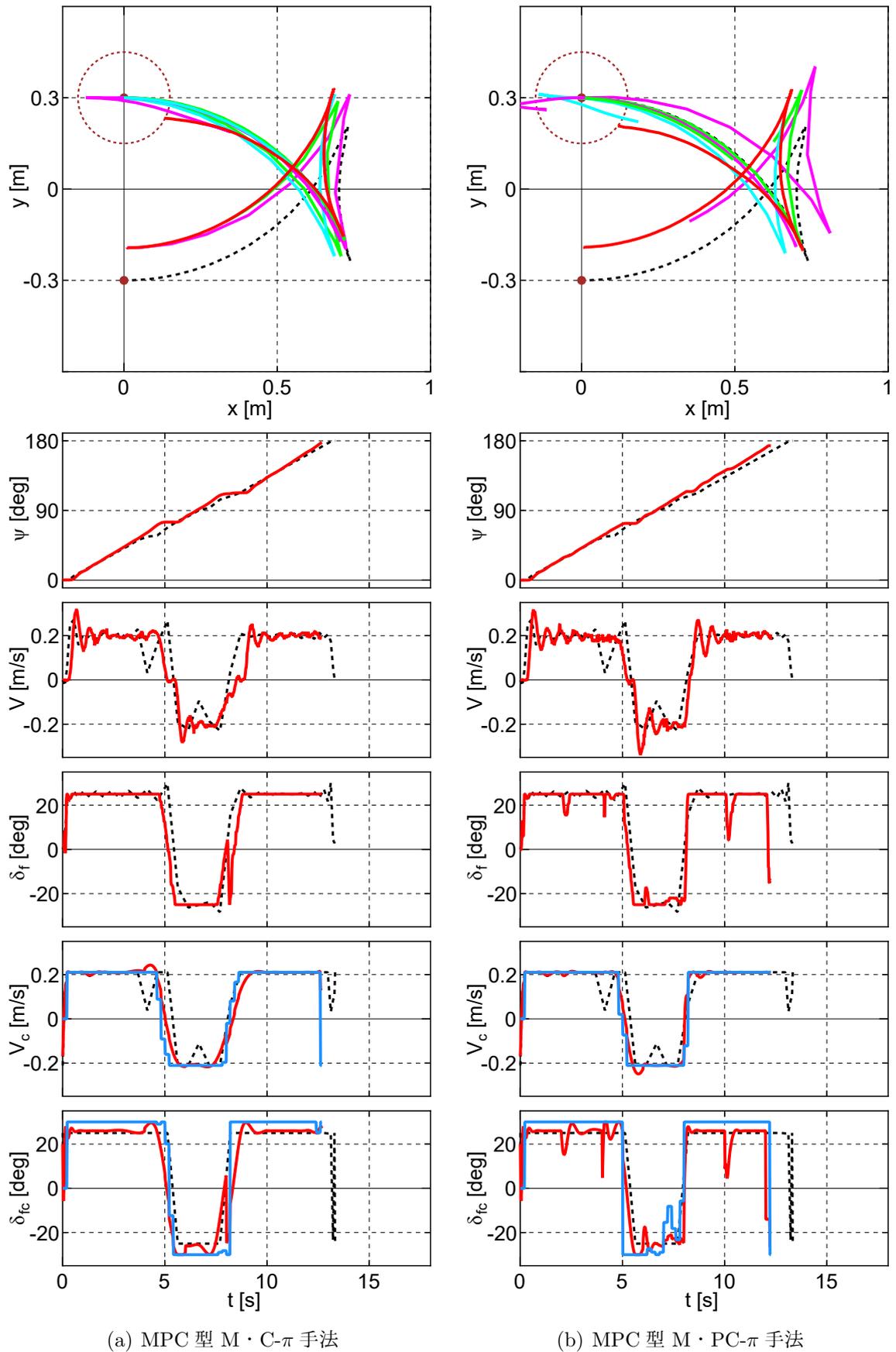


図 4.51 繰り返し走行における走行軌跡と各変数の時間履歴 (CASE 3,  $\tau_c = 2.0$ , その 3)

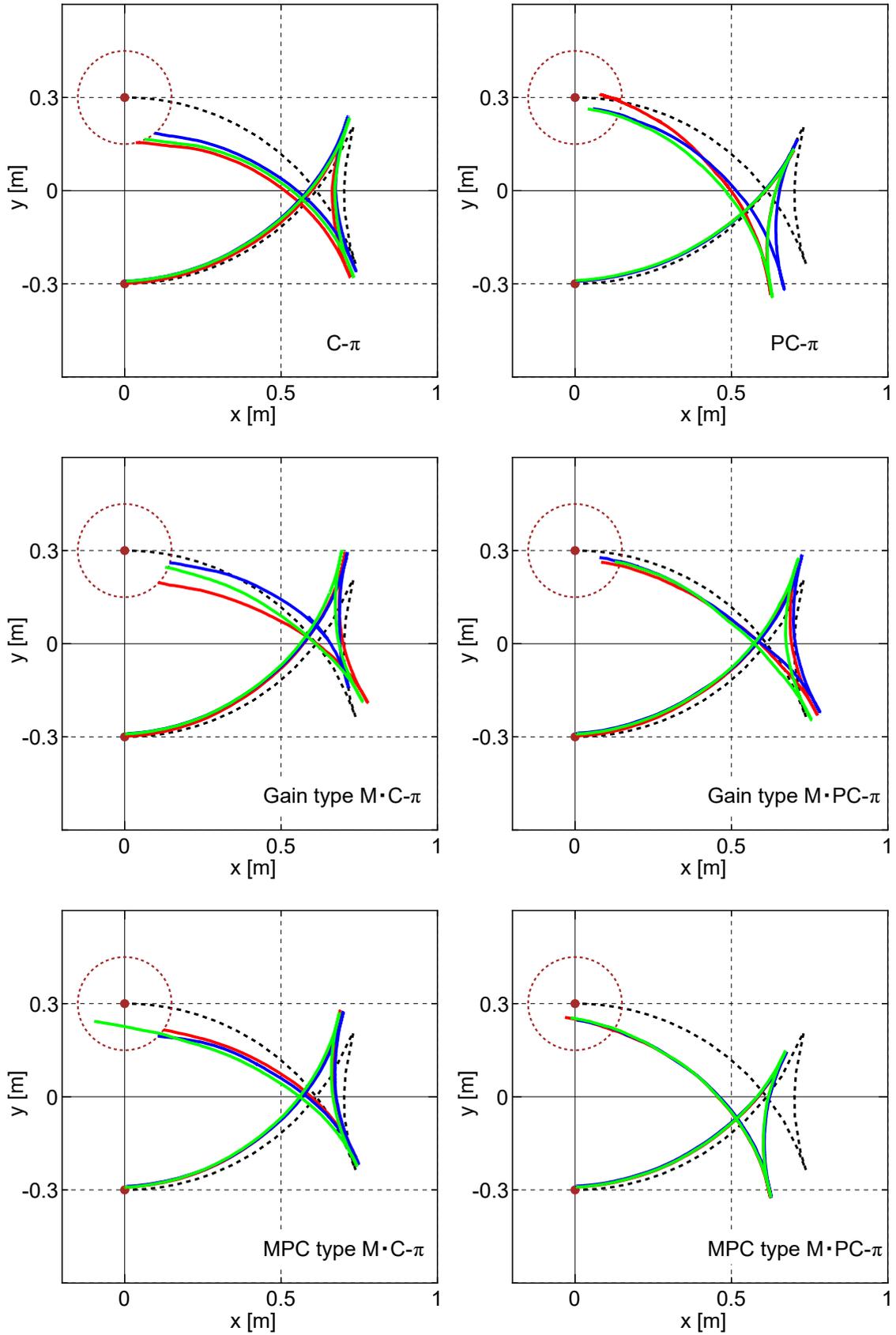


図 4.52 切り返し走行における実験の再現性の確認 (CASE 1,  $\tau_c=1.0$ )

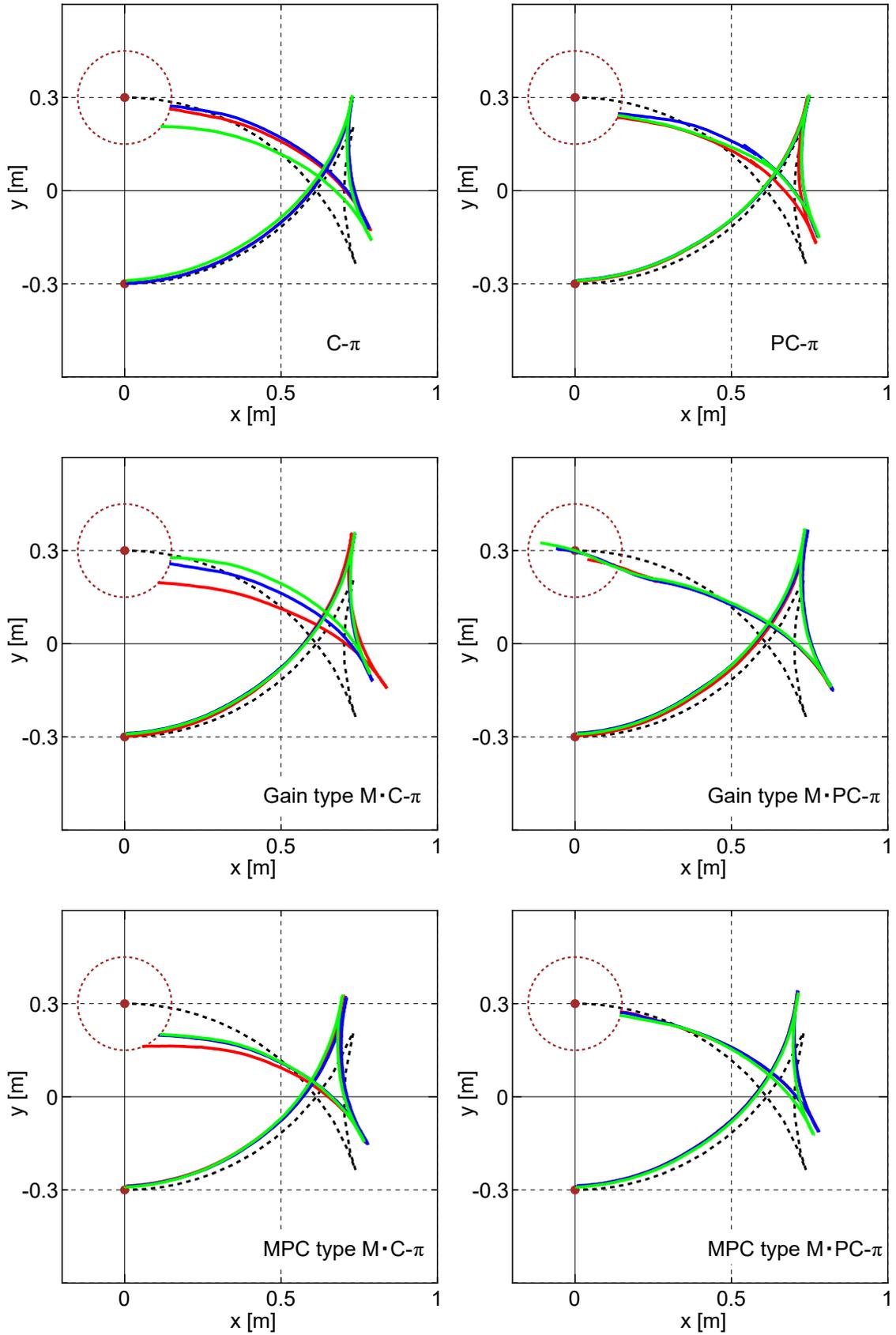


図 4.53 繰り返し走行における実験の再現性の確認 (CASE 1,  $\tau_c=2.0$ )

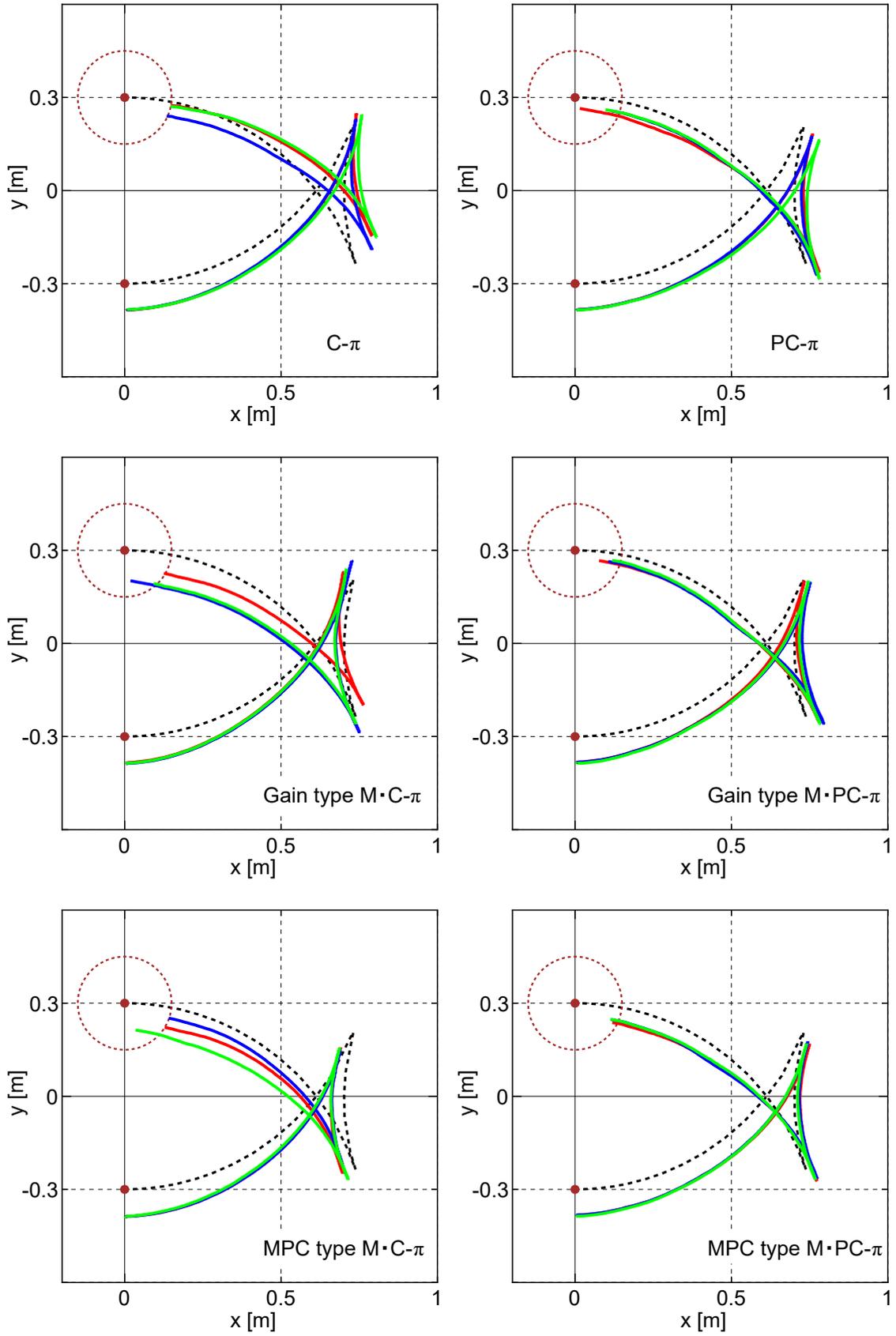


図 4.54 切り返し走行における実験の再現性の確認 (CASE 2,  $\tau_c=1.0$ )

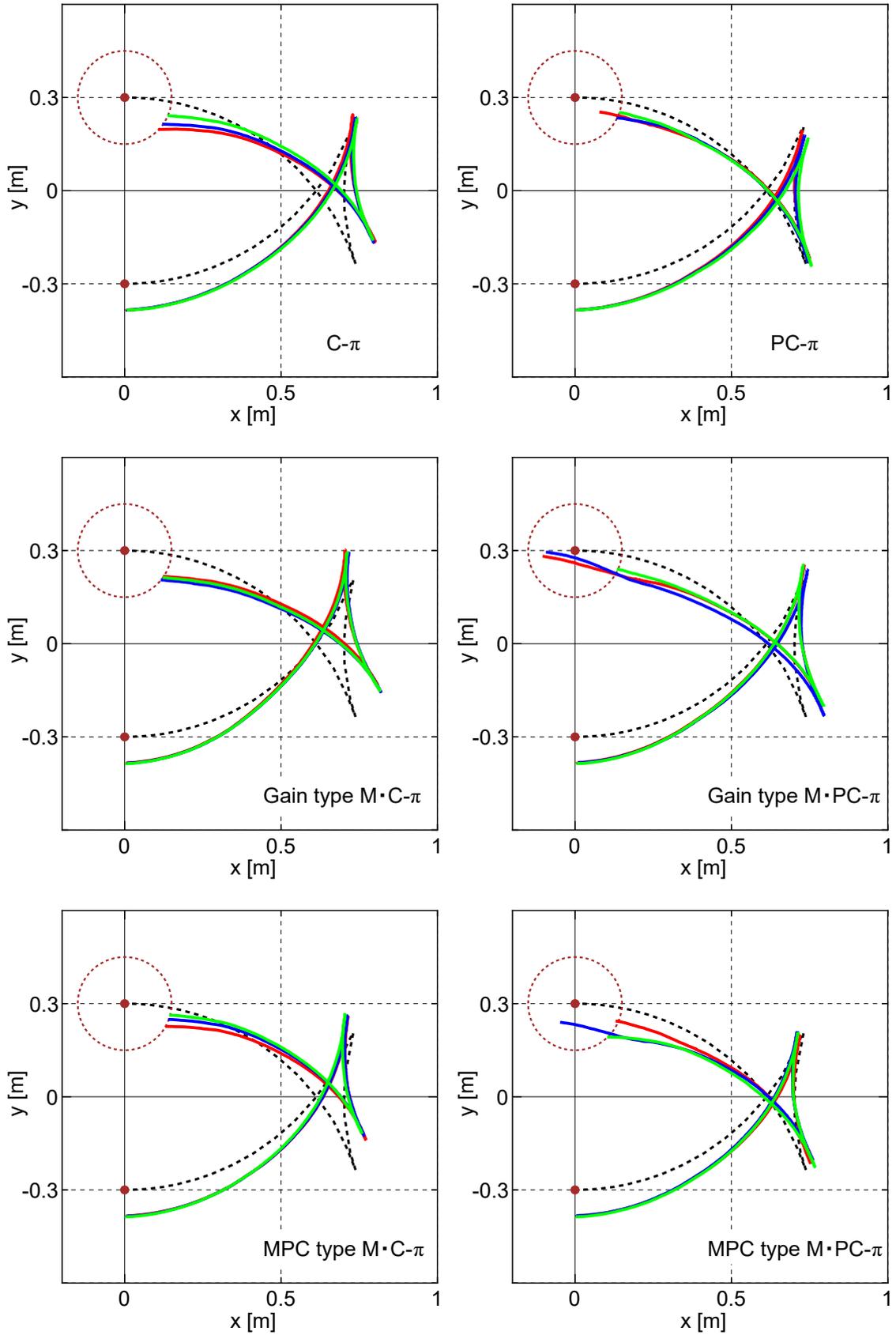


図 4.55 切り返し走行における実験の再現性の確認 (CASE 2,  $\tau_c=2.0$ )

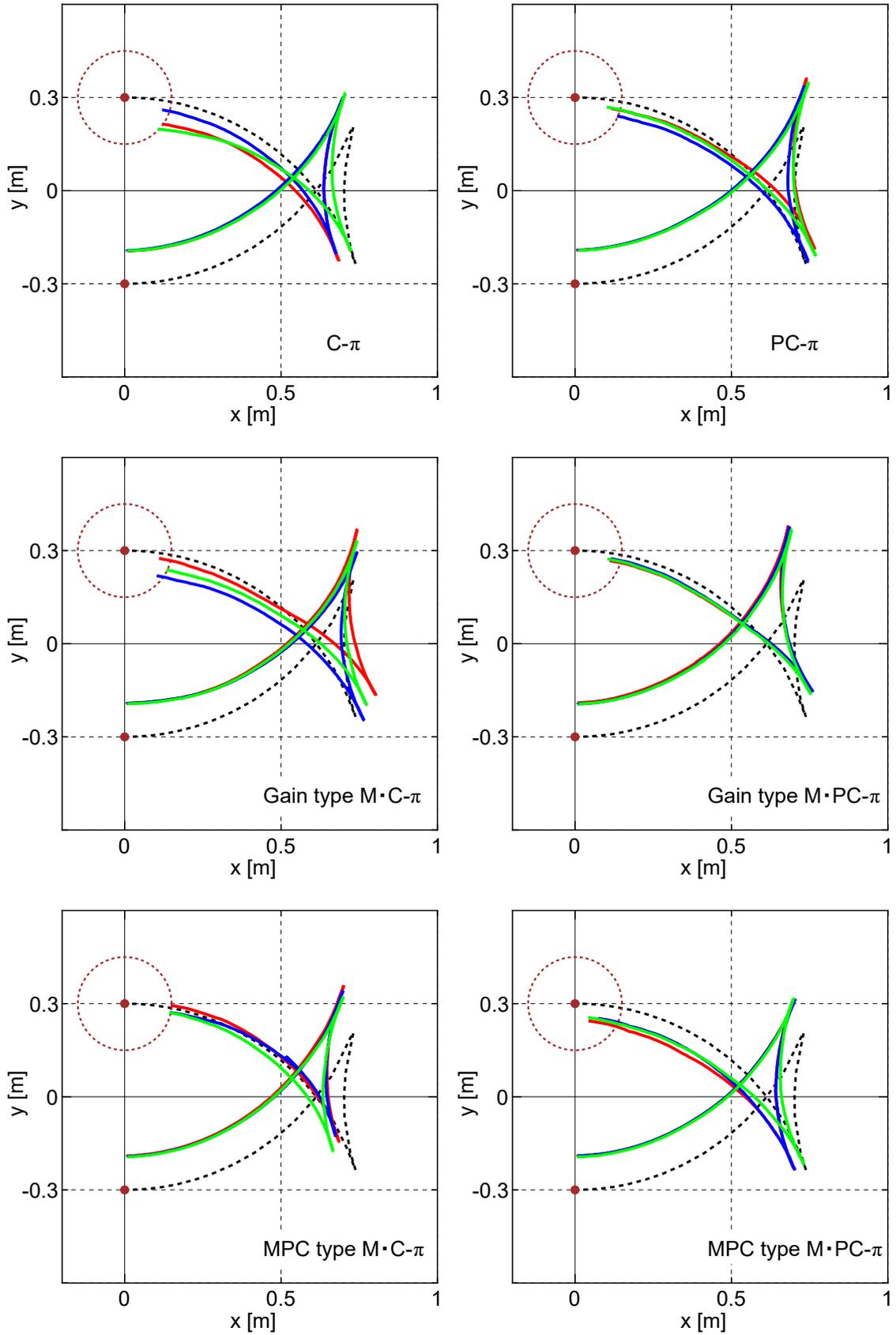


図 4.56 切り返し走行における実験の再現性の確認 (CASE 3,  $\tau_c=1.0$ )

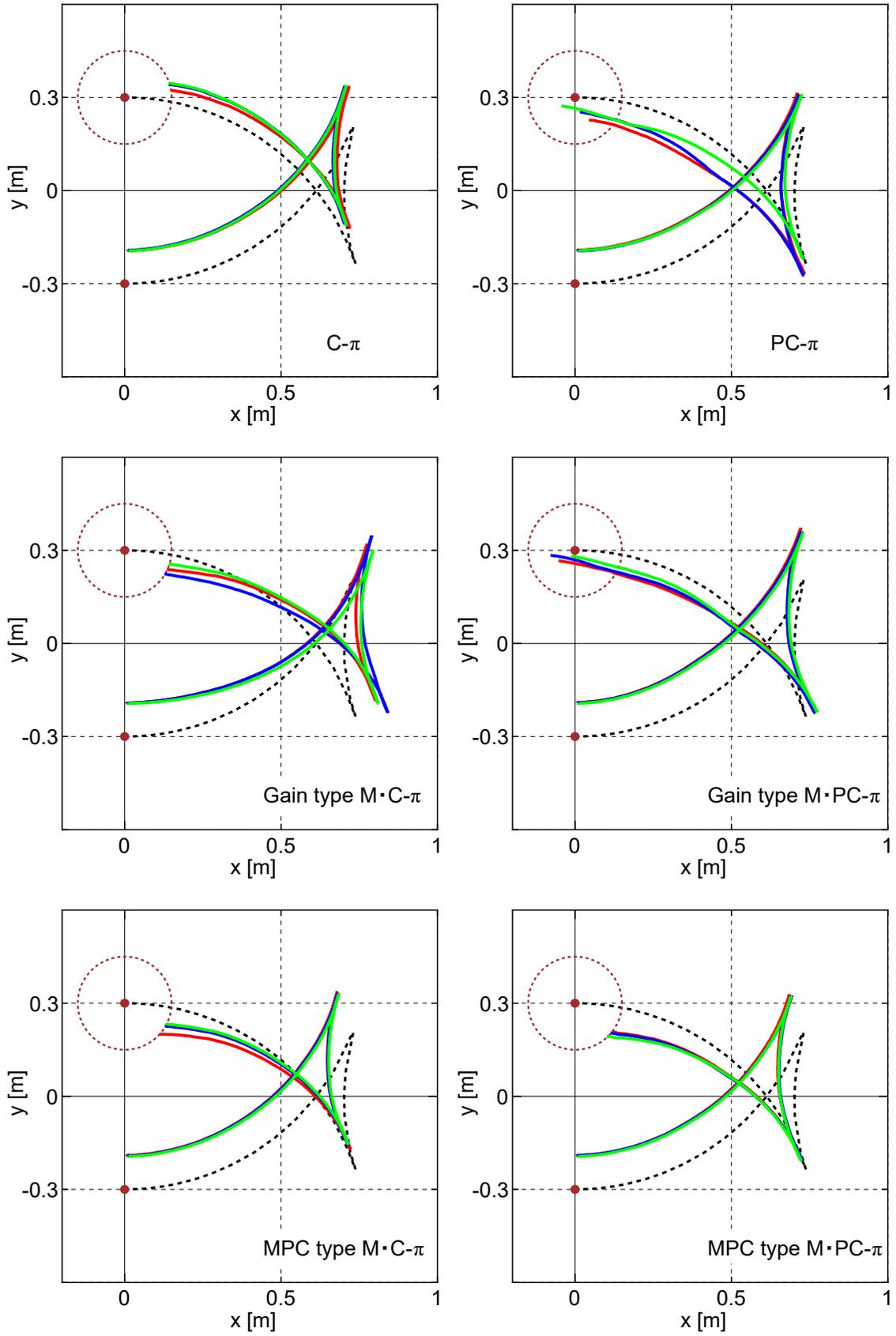
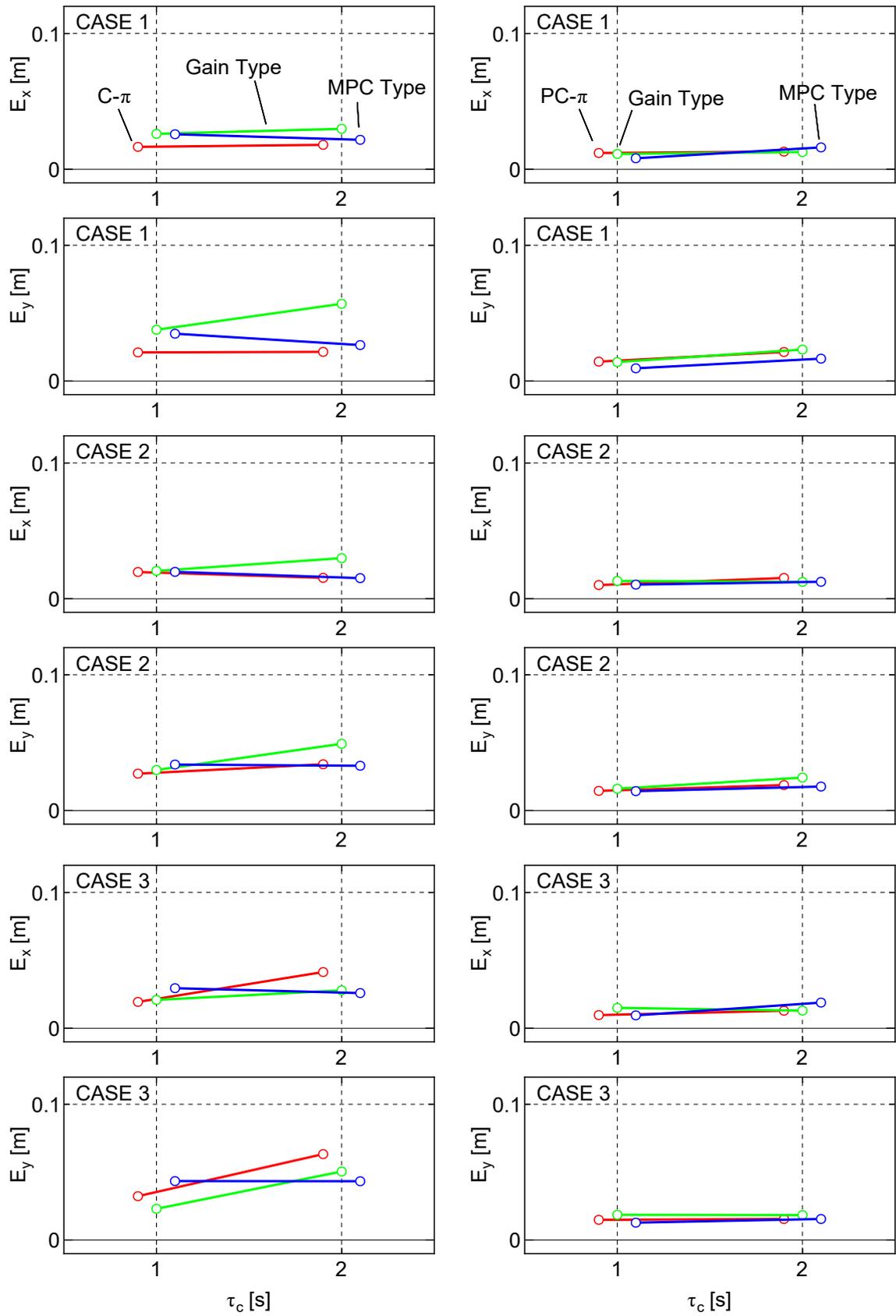


図 4.57 切り返し走行における実験の再現性の確認 (CASE 3,  $\tau_c=2.0$ )



(a) C- $\pi$  手法等

(b) PC- $\pi$  手法等

図 4.58 繰り返し走行における予測軌道と実軌道の誤差

#### 4.5.4 障害物回避走行

・想定

図 4.59 に障害物回避走行における想定の概要及びオフラインの最適化計算で得られた走行軌跡を示す。障害物回避走行では、直線走行と同じく、座標  $(-1, -1)$ ，ヨー角  $45 \text{ deg}$  の状態から座標  $(1, 1)$ ，ヨー角  $45 \text{ deg}$  の状態まで移動する。その他の状態は 0 としている。

直線走行のように前方の目標地点への単純な移動を行う場合であっても、走行途中に障害物などが存在すればそれを回避して目標まで移動しなければならない。さらに障害物を発見するタイミングが異なれば MGV と障害物の相対位置が変わり、最適な回避軌道は変化する。そのため、障害物回避走行では以下の 4 ケースを想定した走行実験を行う。

- ・ CASE 1 : 走行開始後の  $t = 0.9 \text{ s}$  にて障害物が発生
- ・ CASE 2 : 走行開始後の  $t = 1.9 \text{ s}$  にて障害物が発生
- ・ CASE 3 : 走行開始後の  $t = 2.9 \text{ s}$  にて障害物が発生
- ・ CASE 4 : 走行開始後の  $t = 3.9 \text{ s}$  にて障害物が発生

図 4.60 に障害物の発生するタイミングについてケースごとの概略を示す。ここで障害物は中心座標  $(-0.05, 0.05)$ ，半径  $0.15 \text{ m}$  の円形障害物としてオフライン最適軌道上

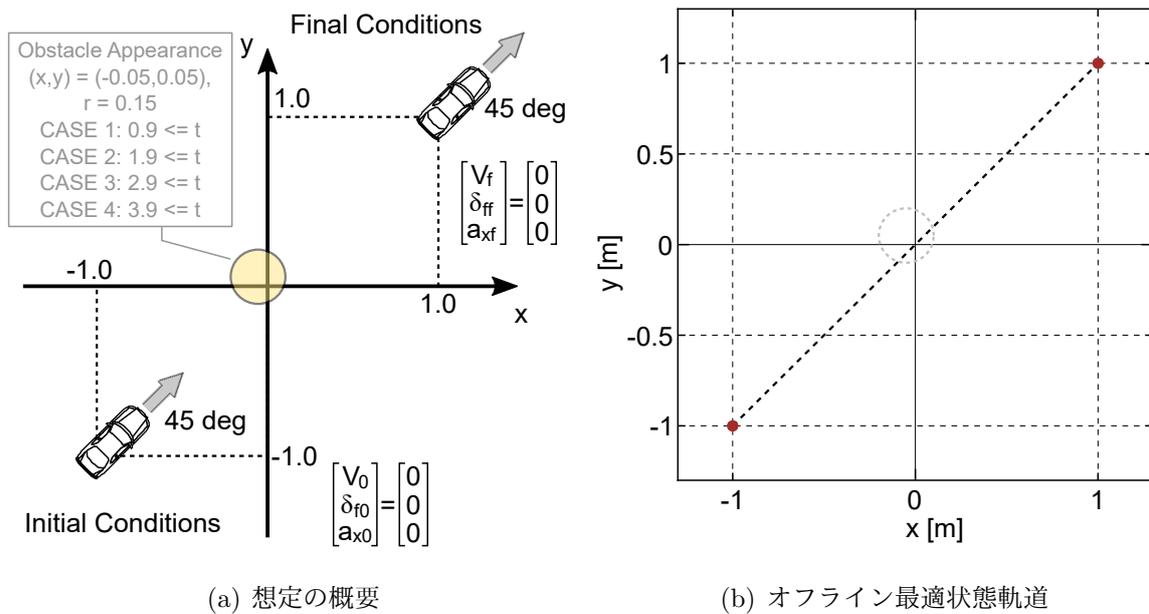


図 4.59 MGV の運動制御の想定：障害物回避走行

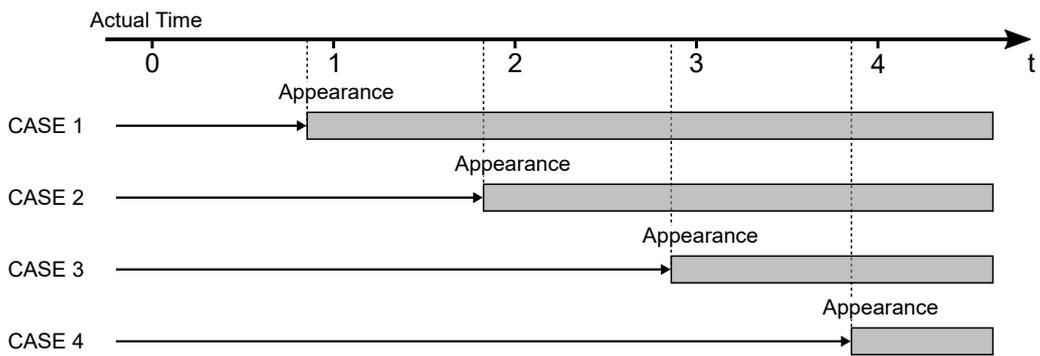


図 4.60 障害物発生タイミング

に発生させる．いずれの場合においてもオフラインで行う最適化計算では障害物の発生を考慮しない．

・実験結果

図 4.61 に障害物回避走行における全ての走行パターンと、本節で示している実験結果の関係を示す．

- ① 最適解の生成状況では、時系列で最適軌道の生成の様子を示し、障害物回避走行において移動しながら最適解を更新できていることを確認する．
- ② 走行軌跡では、走行パターンごとの3回ずつの走行軌跡を示し、障害物回避走行における走行実験の再現性を確認する．
- ③ 各手法の時間履歴では、MGVの状態や実際に適用した制御入力、追従制御器によ

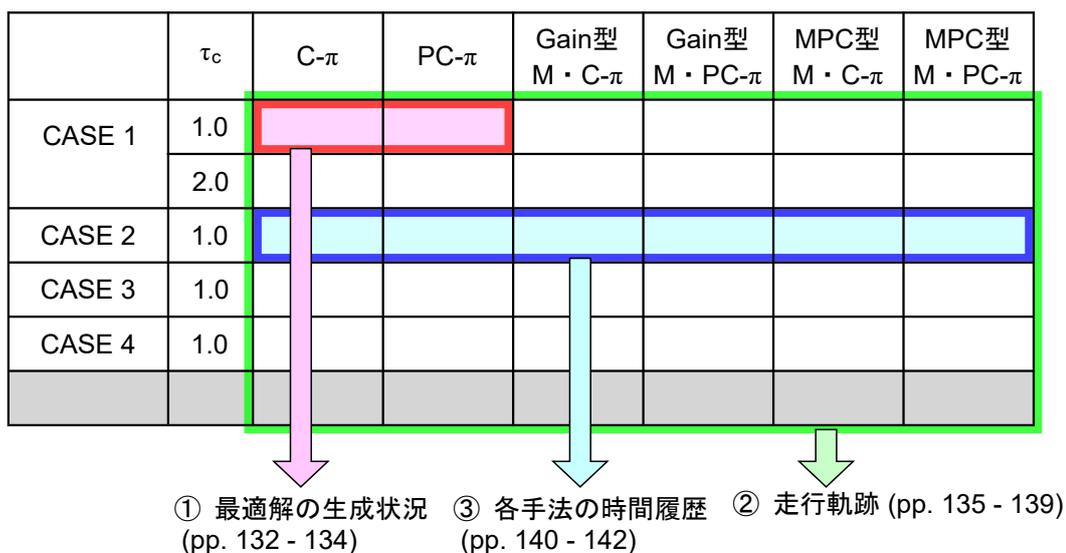


図 4.61 障害物回避走行実験の走行パターンと結果の一覧

り生成した制御入力の時履歴を示し、障害物回避走行における各手法の制御状況を確認する。

図 4.62 及び 4.63 に、障害物回避走行，CASE 1， $\tau_c = 1.0$  s における 2 秒ごとの C- $\pi$  手法及び PC- $\pi$  手法による最適解の生成状況を示す。黒色の破線がオフライン最適軌道，赤色の実線が実軌道，マゼンダの実線が各時刻で生成した最適状態軌道を表している。  $t = 0$  s の走行開始時には障害物が発生していないため目標まで直線的な軌道を生成しているが，障害物発生以降では各時刻において，C- $\pi$  手法ではサンプリングした状態から，PC- $\pi$  手法では予測した状態から目標状態までの回避軌道が生成できている。

図 4.64 に同じ想定における評価関数値の内訳を示す。図 4.64(a) は障害物発生後も C- $\pi$  手法を用いて直進軌道を生成し続けた場合の結果，図 4.64(b) 及び 4.64(c) は，それぞれ C- $\pi$  手法及び PC- $\pi$  手法を用いて回避軌道を生成した場合の結果である。緑色は  $t_f - t_i$ ，青色は  $\delta_f$ ，赤色は障害物に関する評価関数値を表し，サンプリングタイム別に示している。障害物が発生している状況でオフライン最適軌道のように障害物を通過する軌道を生成した場合，障害物に関する評価関数値が大きくなる。一方，回避軌道を生成している場合では， $t_f - t_i$  及び  $\delta_f$  の評価関数値に大きな変動がないが，障害物に関しては大幅に減少している。ただし，PC- $\pi$  手法を用いた場合では，初期条件を予測している分だけ  $t_f - t_i$  の値が低くなる。急激な操舵角の変化の抑制及び障害物の回避を行いつつ，より短時間での移動する最適制御解が生成されている。

図 4.65～4.69 に障害物回避走行の全走行パターンにおける走行軌跡の結果を示す。いずれの走行パターンにおいても 3 回ずつ走行を行っており，概ね同様の結果が得られている。MGV の車体寸法を踏まえた回避結果をまとめると表 4.7 の通りとなる。CASE 1 の  $\tau_c = 2.0$  s の場合及び CASE 4 の場合はどの手法を用いても障害物に衝突しているが，CASE 1 の  $\tau_c = 1.0$  s の場合及び CASE 2 の場合では C- $\pi$  手法以外の全ての手法，CASE 3 の場合では PC- $\pi$  手法，Gain 型 M・PC- $\pi$  手法，及び MPC 型 M・PC- $\pi$  手法が回避に成功し，手法により差異が生じた。

図 4.70～4.72 に、障害物回避走行，CASE 2 における各手法の時履歴を示す。黒色の短鎖線はオフラインでの最適制御解，赤色の実線は実軌道，マゼンダ，緑色，シアンの実線は各サンプリング時刻において生成された走行軌跡，空色の実線は追従機構により生成された制御入力を示す。

図 4.70(a) の C- $\pi$  手法を用いた場合では， $t = 2$  s でのサンプリングにおいて最初の回避軌道を生成している。 $t = 2$  s から  $t = 4.5$  s 付近において右へ操舵を行うことにより回避する解が得られてるが，右操舵の大部分が制御入力の適用されない最初の 1 秒間に含まれているため，十分な回避が行えず障害物に衝突している。

図 4.71(a) 及び 4.72(a) の Gain 型 M・C- $\pi$  手法及び MPC 型 M・C- $\pi$  手法を用いた

表 4.7 障害物回避結果

	$\tau_c$	C- $\pi$	Gain 型 M · C- $\pi$	MPC 型 M · C- $\pi$	PC- $\pi$	Gain 型 M · PC- $\pi$	MPC 型 M · PC- $\pi$
CASE 1	1.0	×	○	○	○	○	○
	2.0	×	×	×	×	×	×
CASE 2	1.0	×	○	○	○	○	○
CASE 3	1.0	×	×	×	○	○	○
CASE 4	1.0	×	×	×	×	×	×

場合では、C- $\pi$  手法と同様に、 $t = 2$  s から目標までの回避軌道は生成されているがフィードフォワード制御入力としての右操舵の大部分は適用されていない。しかしながら、 $t = 3$  s 付近では回避軌道との誤差を修正するためにフィードバック制御入力として右操舵が生成されており、最適軌道に追従することで障害物を回避して目標まで到達することができている。

図 4.70(b), 4.71(b), 4.72(b) の PC- $\pi$  手法, Gain 型 M · PC- $\pi$  手法, MPC 型 M · PC- $\pi$  手法を用いた場合には、 $t = 2$  s でサンプリングした情報をもとに  $t = 3$  s から目標までの最適制御解を生成している。生成した制御解は  $t = 3$  s から大きな右操舵を行って障害物を回避し、目標へ到達する解となっている。右への操舵は  $t = 3$  s において実入力としてそのまま適用され、障害物を回避して目標まで到達することができている。

CASE 4 についてはいずれの手法を用いても障害物に衝突している。CASE 4 では  $t = 3.9$  s で障害物を認知し、 $t = 5$  s から回避を行う制御解が適用される。MGV が約 0.2 m/s で移動する場合、その間に約 1 m 前進する。走行開始位置から障害物の淵までは約 1.25 m あるが、後輪車軸中央から前面まで 0.3 m 以上あるため MGV の大きさを考慮すると、障害物を認知してから回避軌道を生成するまでに必ず衝突することになる。そのような場合には強制的に停止するような別のアルゴリズムを追加する必要がある。

以上より障害物回避走行における走行実験で以下の結果を得た。

- (1) 障害物走行の各想定において、設計した各手法により最適制御解を更新しながら実機の MGV を制御することができた。
- (2) Gain 型や MPC 型の追従制御器を組み込むこと、特に初期条件を予測するプロパゲーション型を用いることで回避性能が向上することを示した。
- (3) 障害物を認知してから回避軌道を生成するまでに別のアルゴリズムを追加することで衝突を回避する必要がある。

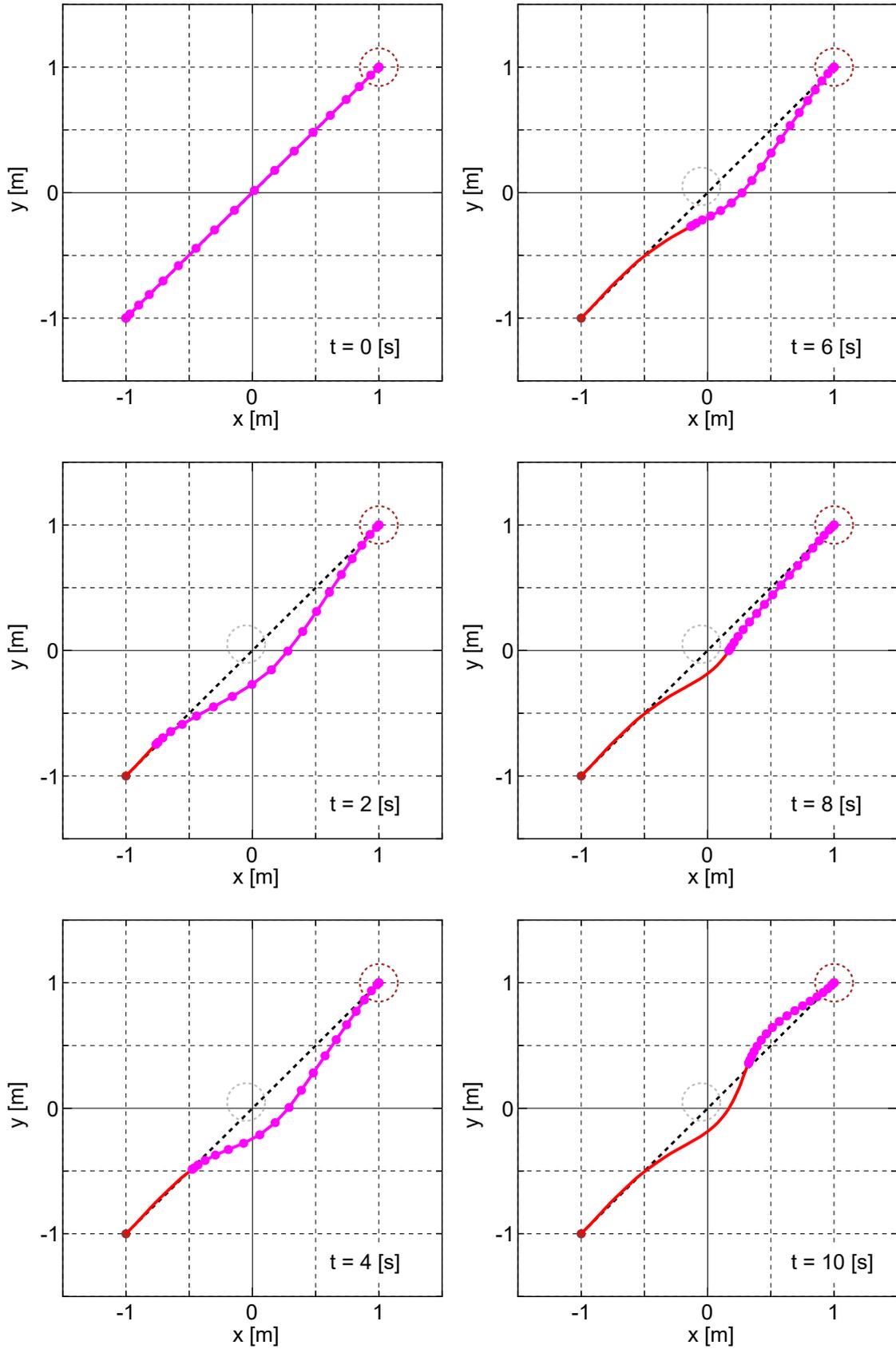


図 4.62 障害物回避走行における最適解の生成状況 (C- $\pi$ 手法, CASE 1,  $\tau_c=1.0$ )

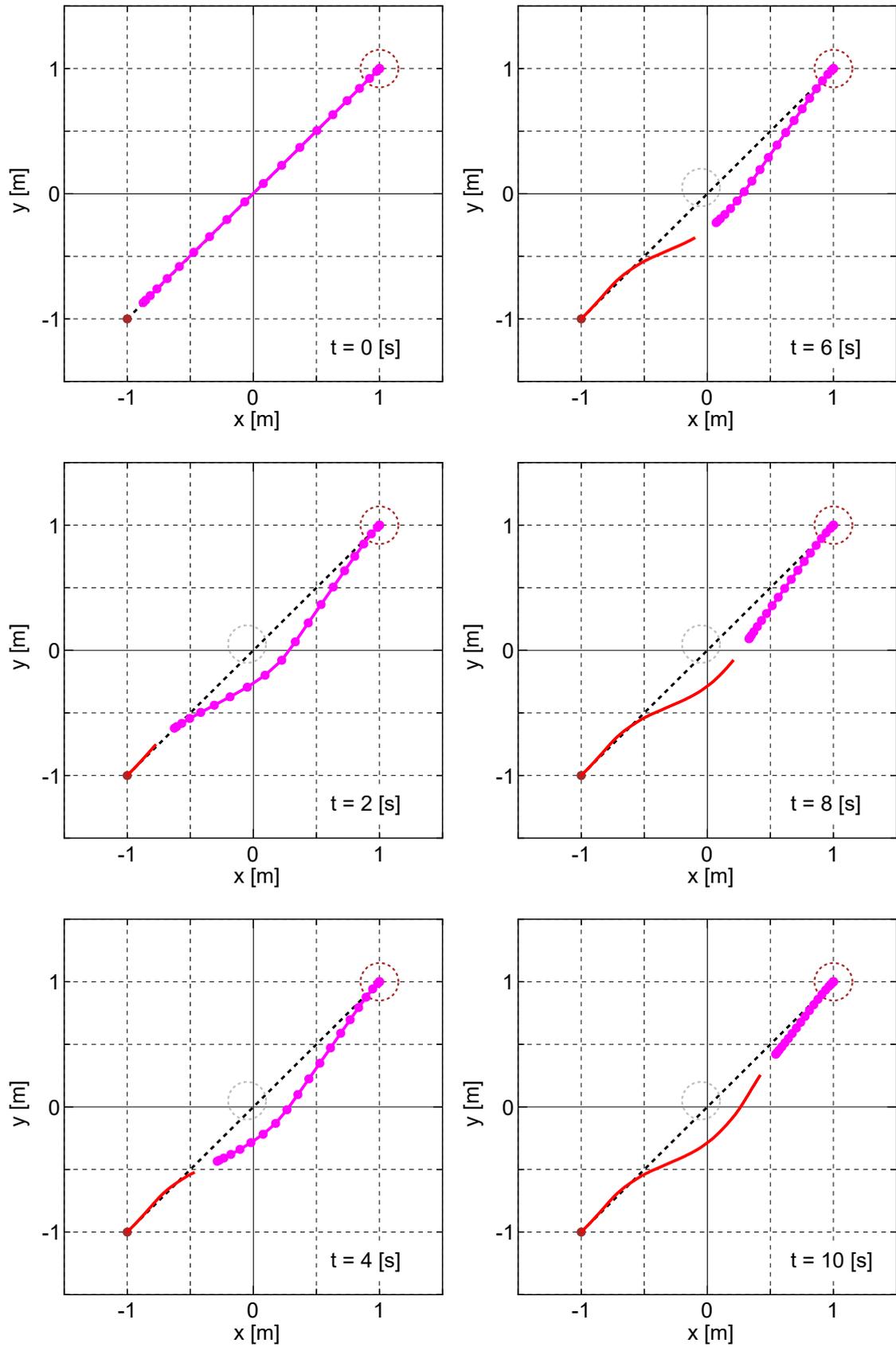
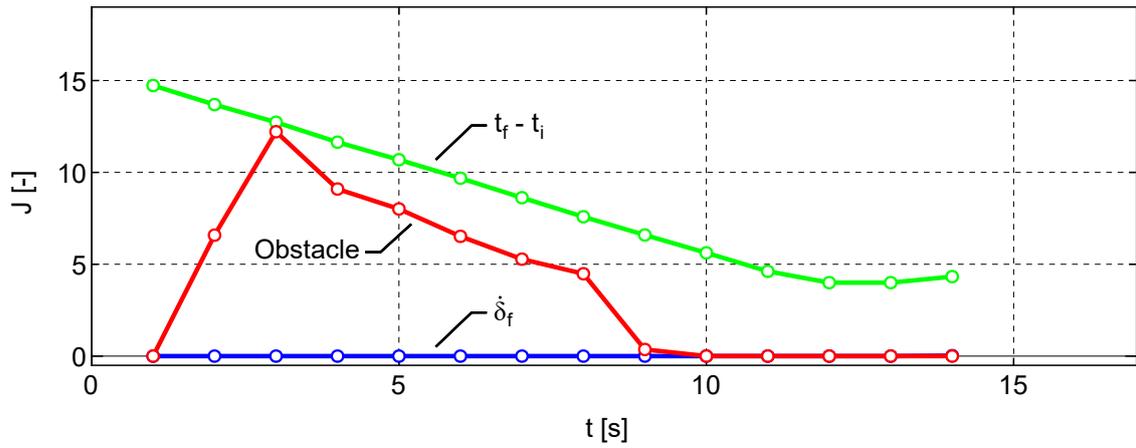
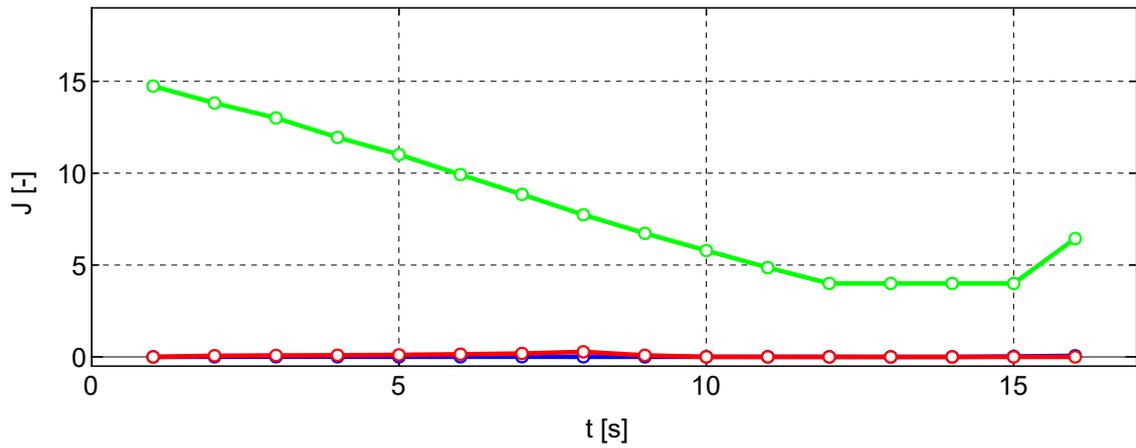


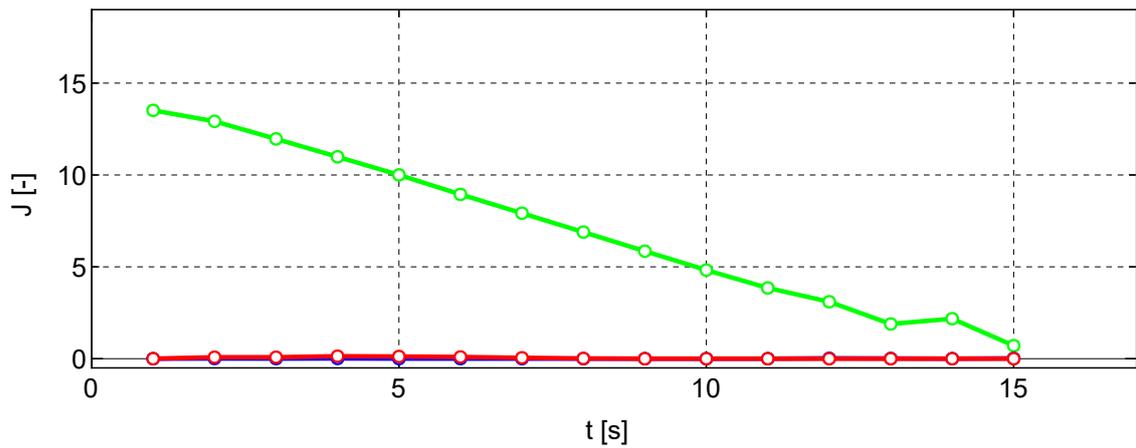
図 4.63 障害物回避走行における最適解の生成状況 (PC- $\pi$ 手法, CASE 1,  $\tau_c=1.0$ )



(a) 直進した場合 (C- $\pi$  手法)



(b) 回避した場合 (C- $\pi$  手法)



(c) 回避した場合 (PC- $\pi$  手法)

図 4.64 評価関数値の内訳 (CASE 1,  $\tau_c=1.0$ )

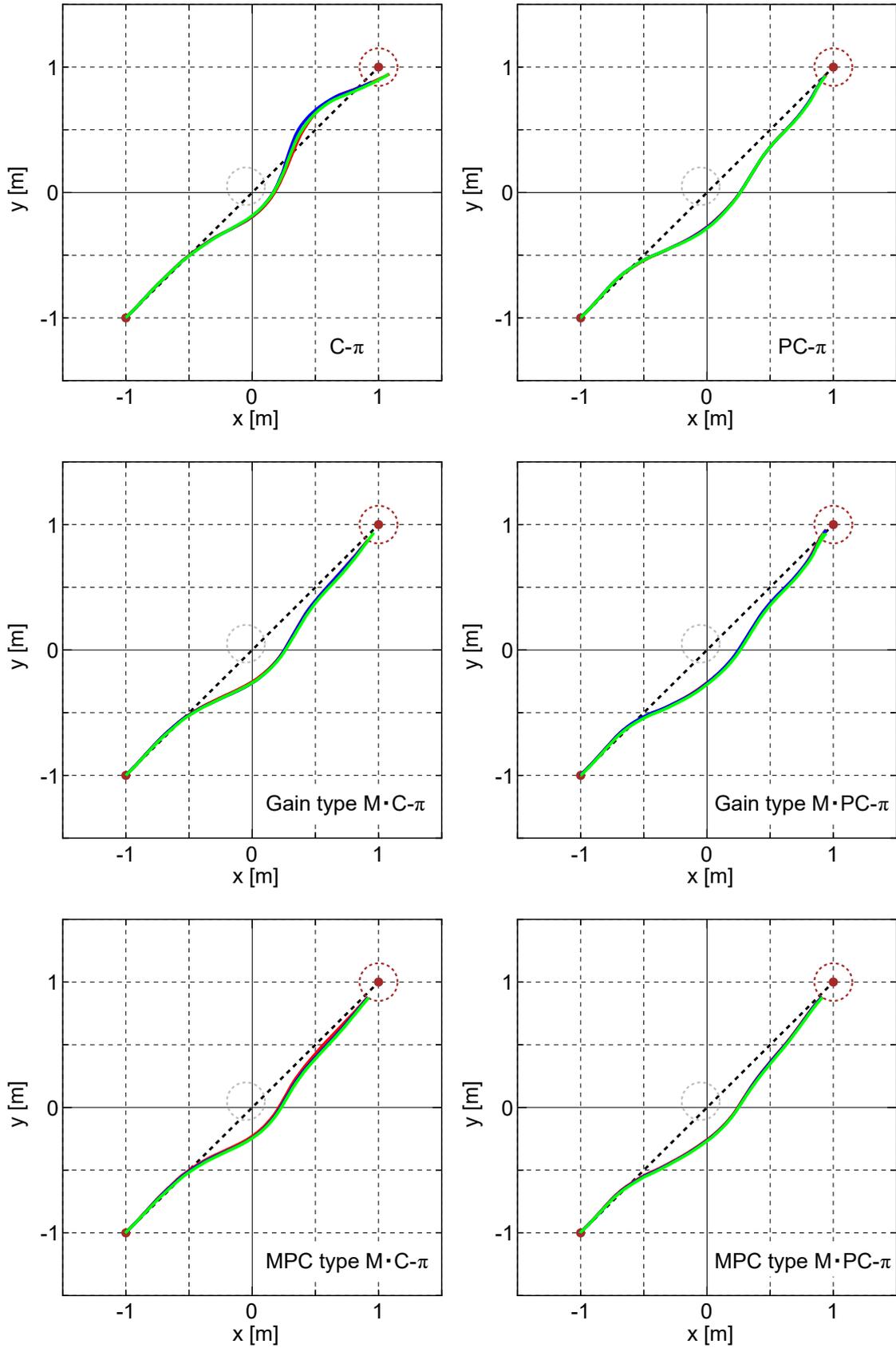


図 4.65 障害物走行における実験の再現性の確認 (CASE 1,  $\tau_c=1.0$ )

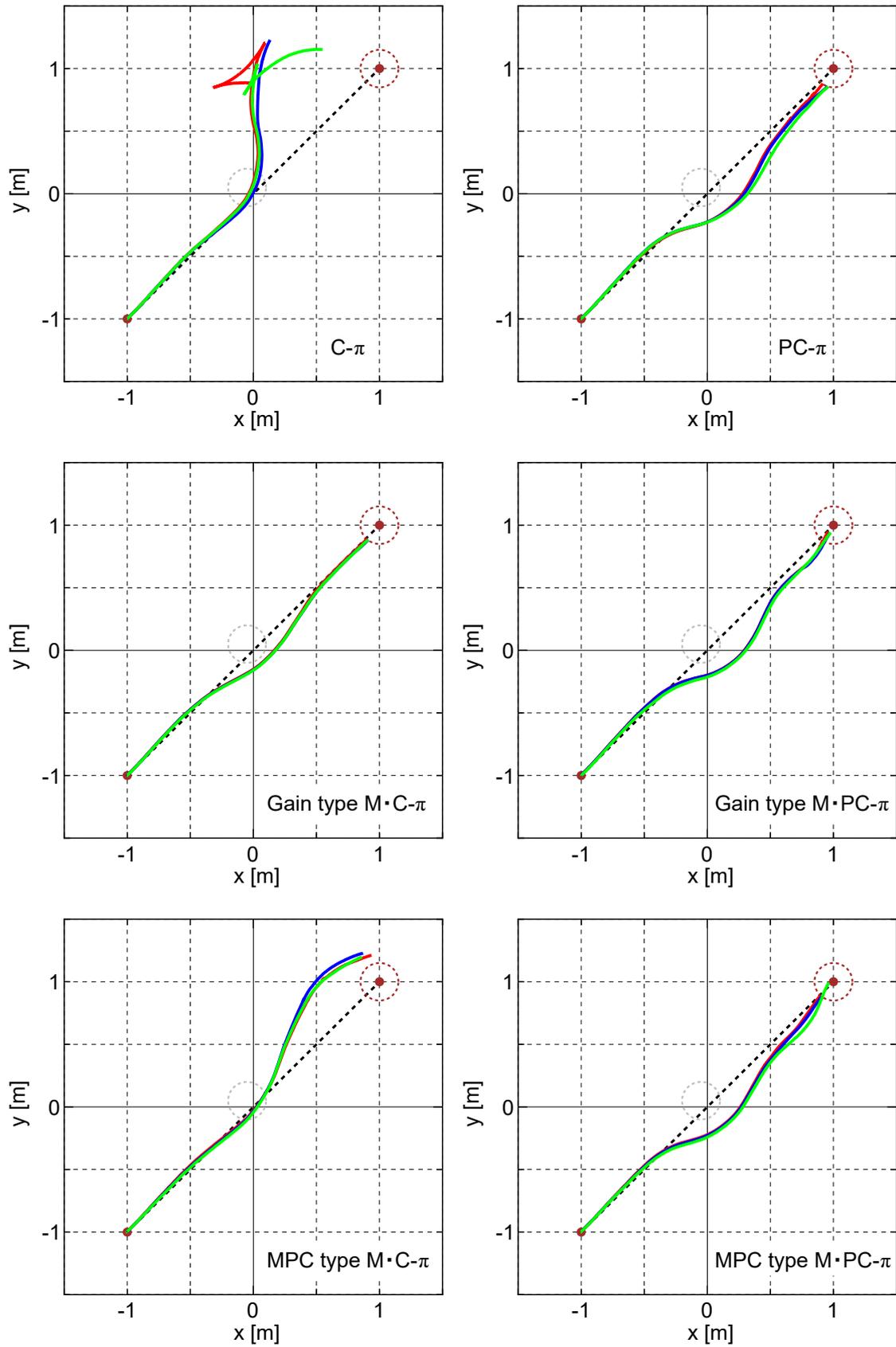


図 4.66 障害物走行における実験の再現性の確認 (CASE 1,  $\tau_c=2.0$ )

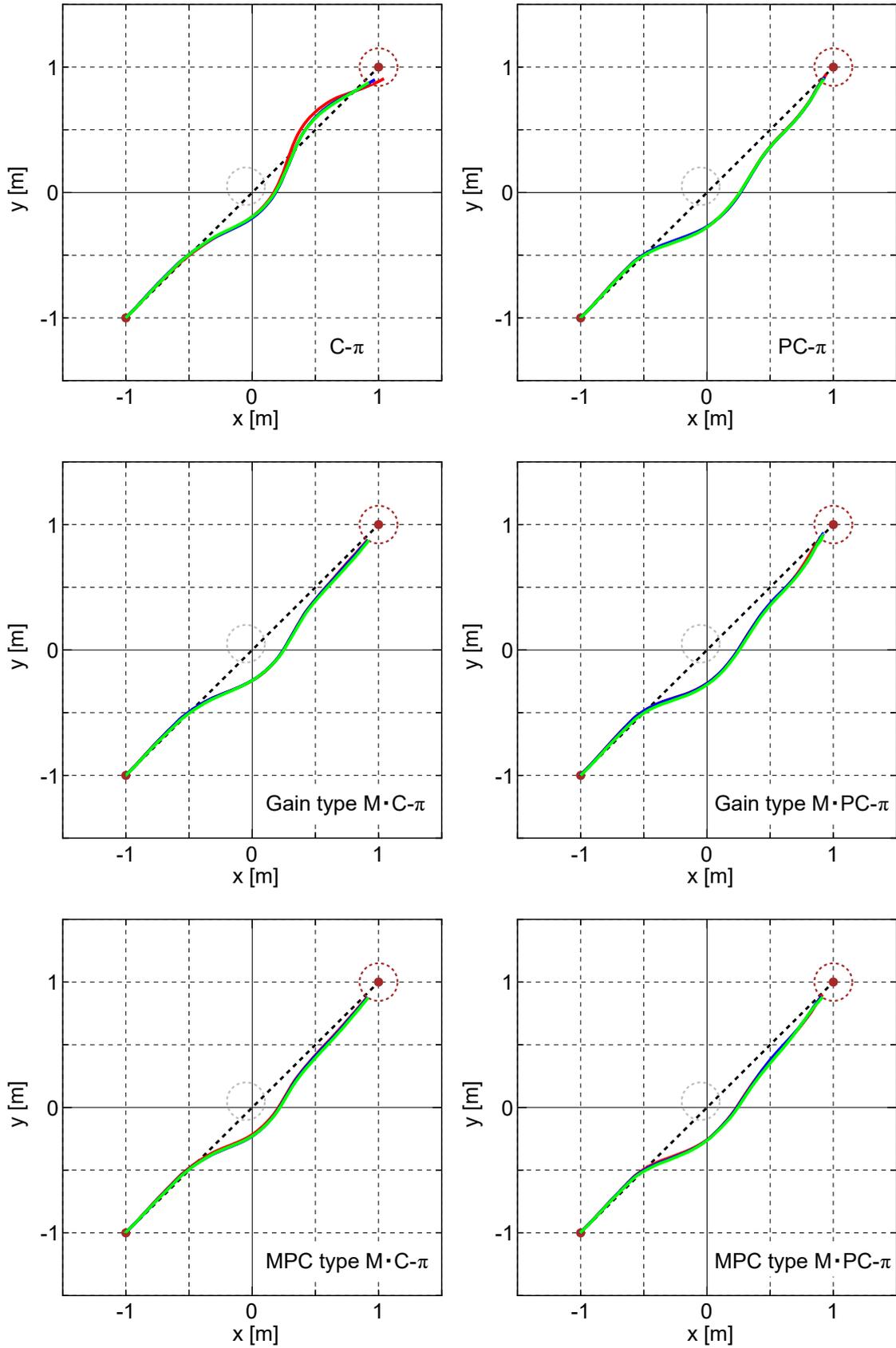


図 4.67 障害物走行における実験の再現性の確認 (CASE 2,  $\tau_c=1.0$ )

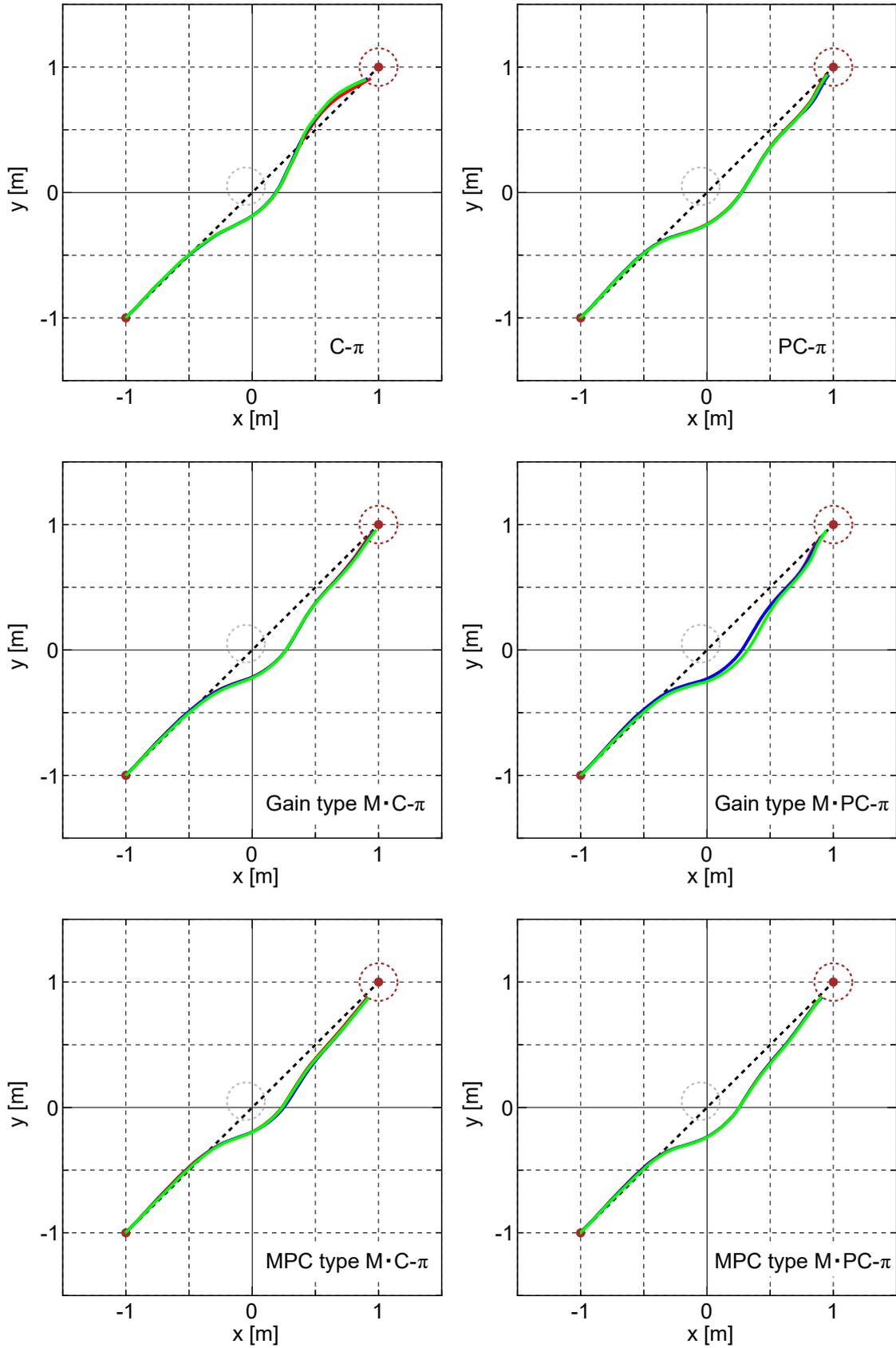


図 4.68 障害物走行における実験の再現性の確認 (CASE 3,  $\tau_c=1.0$ )

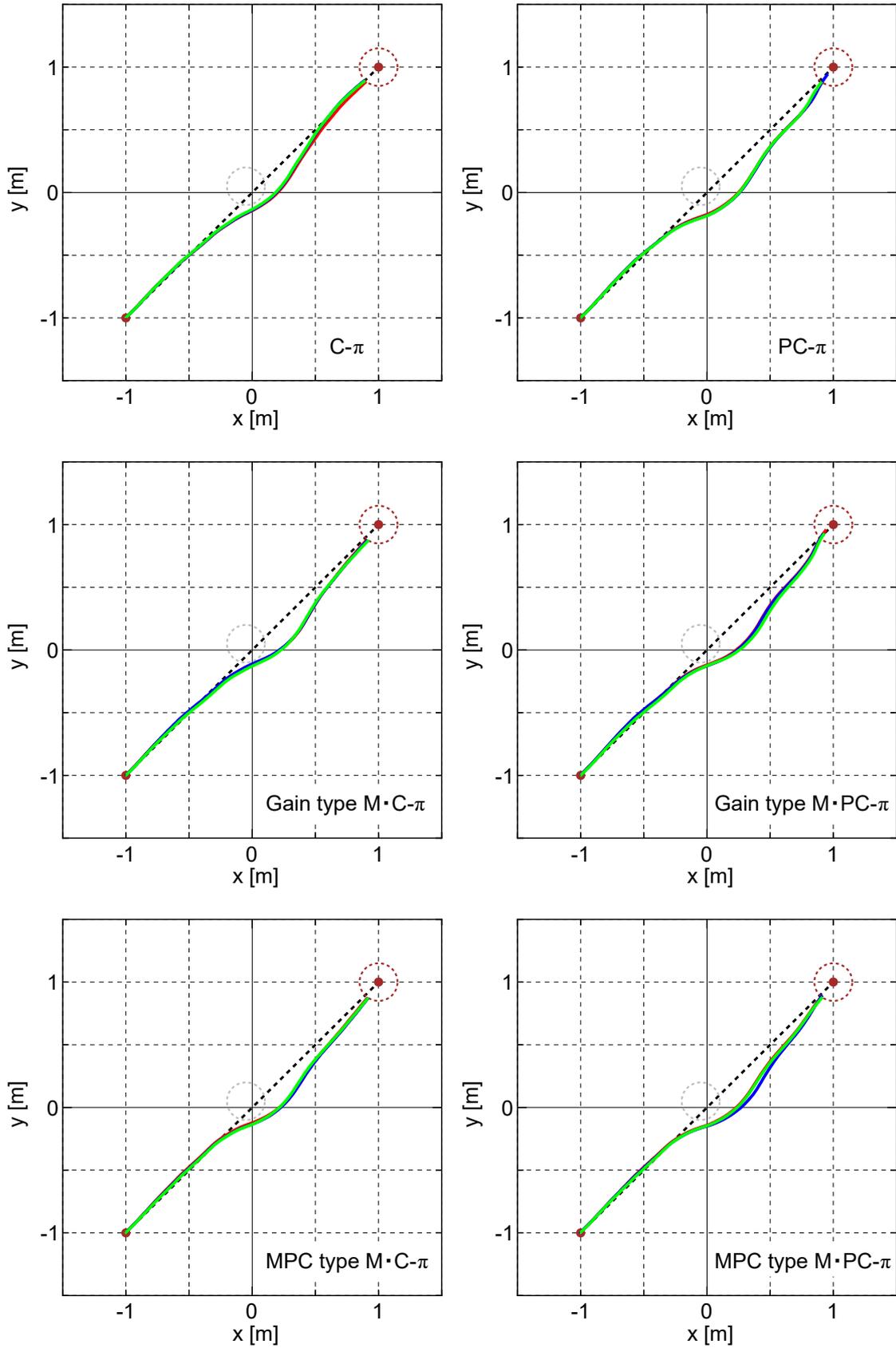


図 4.69 障害物走行における実験の再現性の確認 (CASE 4,  $\tau_c=1.0$ )

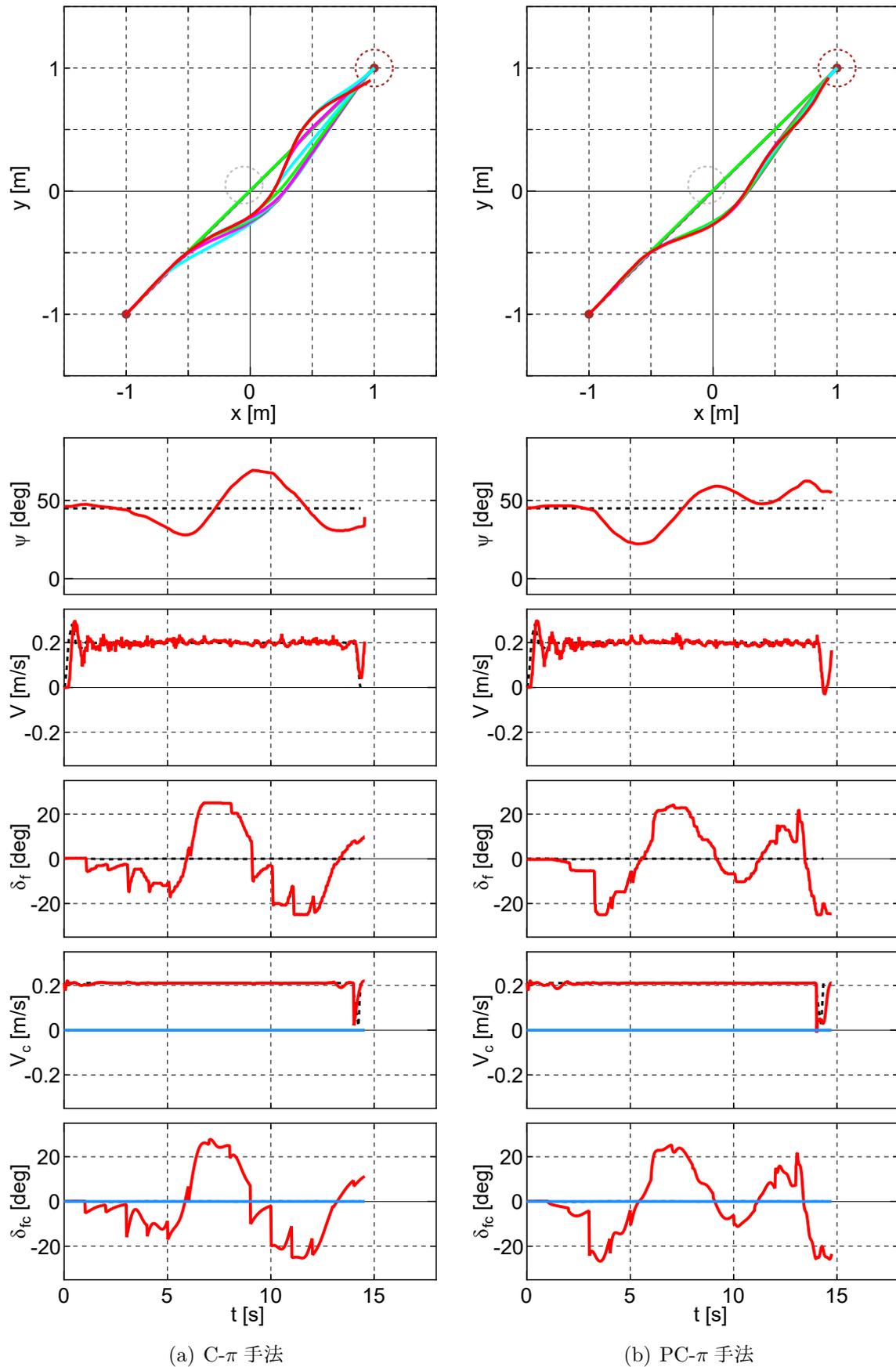


図 4.70 障害物回避走行における走行軌跡と各変数の時間履歴 (CASE 2, その 1)

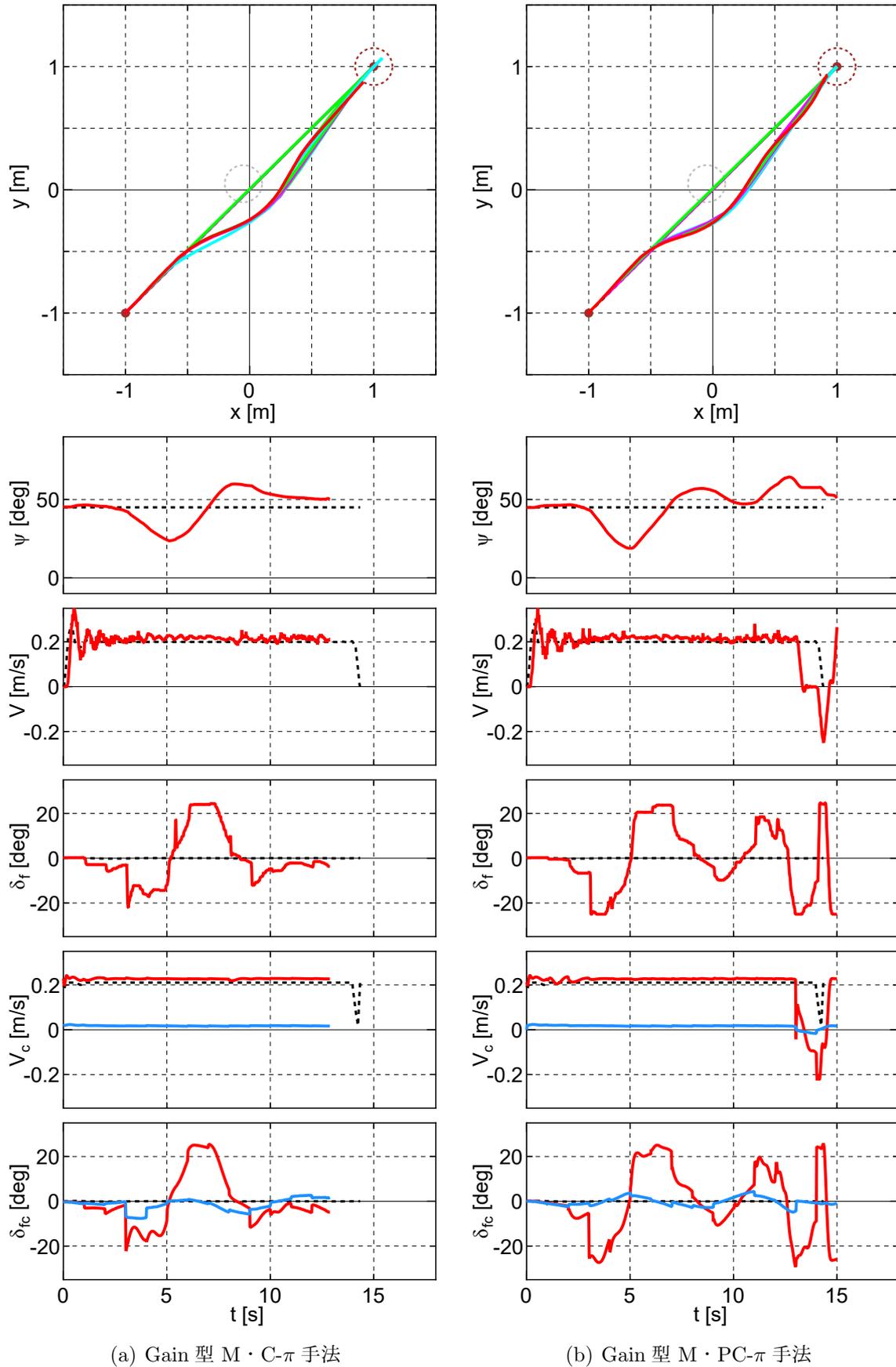
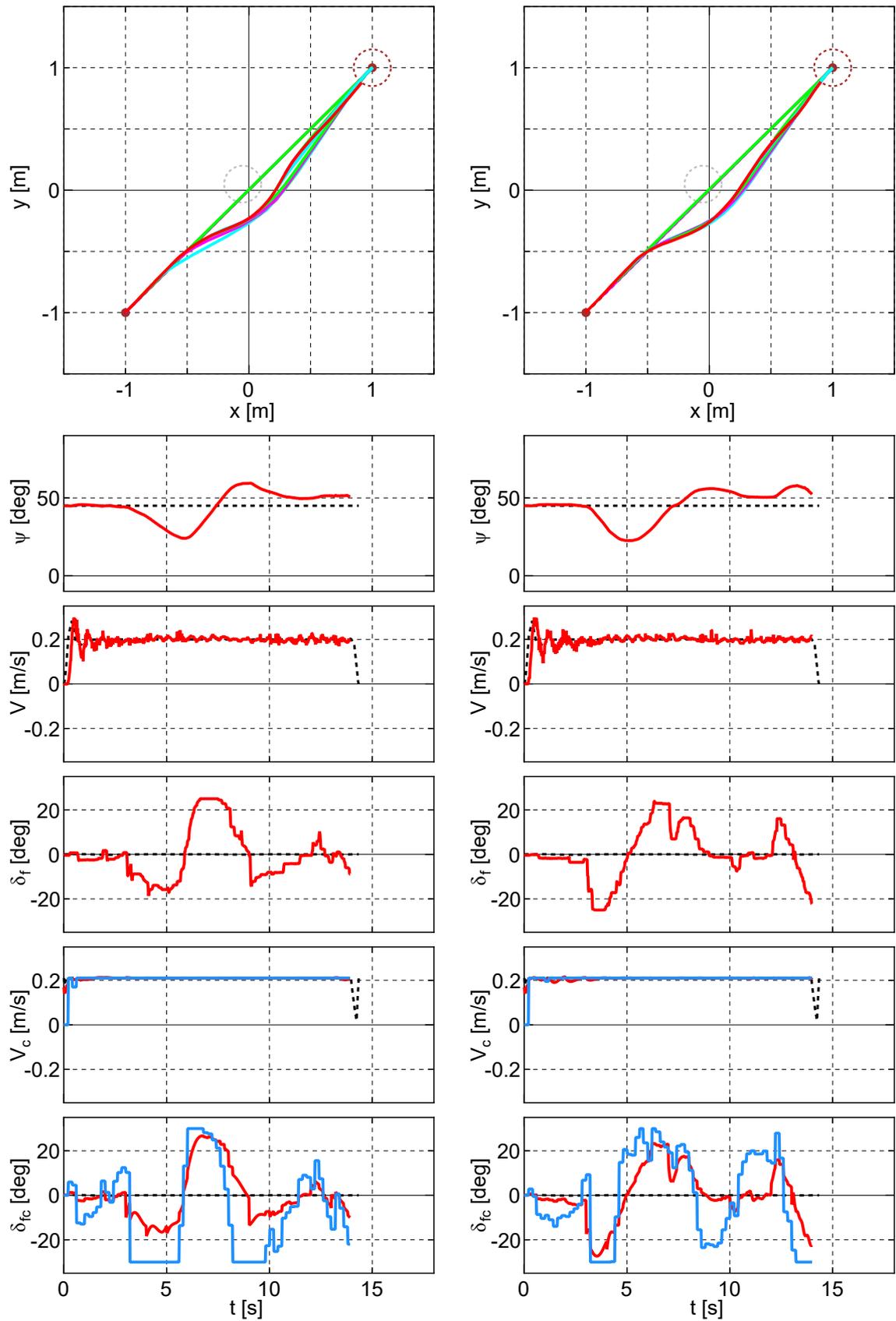


図 4.71 障害物回避走行における走行軌跡と各変数の時間履歴 (CASE 2, その 2)



(a) MPC 型 M · C- $\pi$  手法

(b) MPC 型 M · PC- $\pi$  手法

図 4.72 障害物回避走行における走行軌跡と各変数の時間履歴 (CASE 2, その 3)

## 4.6 まとめ

本章では、第2章で記述した各手法をもとにMGVの運動制御へ適用した制御系を設計し、数値シミュレーション及び実験による比較・検証を行い、以下の成果を得た。

### 4.6.1 Clock-Based 実時間最適制御の適用可能性

従来では、MGVの運動制御に対してはC- $\pi$ 手法を用いた数値シミュレーションによる検証しか行われていなかった。そこで、実車両のMGVにC- $\pi$ 手法及びPC- $\pi$ 手法による実時間最適制御を適用した走行を実現させ、各制御手法の適用可能性の確認を本研究における課題とした。

実際に要する最適化計算時間を考慮したサンプリングタイムの下で、更新と走行を並列実行する実験システムを構築した。そして、実場面で想定される直線走行、S字走行、繰り返し走行、障害物回避走行について走行実験を行った。構築した制御系により移動しながら最終の目標状態まで最適制御解が逐次更新され目標まで到達できることを確認した。

以上より、Clock-Based 実時間最適制御であるC- $\pi$ 手法とPC- $\pi$ 手法のMGVの運動制御に対する適用可能性について実証した。

### 4.6.2 予測情報を用いるPC- $\pi$ 手法の有用性

PC- $\pi$ 手法は移動物体の運動制御に対して適用された例は報告されていなかったが、本研究ではC- $\pi$ 手法に対するPC- $\pi$ 手法の有用性について注目し、本章では数値シミュレーション及び実験による検証を試みた。

初期状態誤差がある直線走行などにおいてC- $\pi$ 手法を適用した場合には、誤差を修正する重要な制御指令がアルゴリズムの制約で入力を使用されない最初のサンプリング区間に生成されるため誤差を修正できないことを示した。一方で、PC- $\pi$ 手法を適用した場合には、サンプリングタイム後の状態を予測して最適制御問題の初期条件とすることで、修正する制御指令は次のサンプリング時刻からMGVにそのまま入力され、誤差を修正して目標まで到達することができることを確認した。

以上により、MGVの運動制御におけるC- $\pi$ 手法に対するPC- $\pi$ 手法の有用性について実証した。

### 4.6.3 追従制御器を付加した提案手法の有用性

計算機の能力や最適制御問題の複雑さなどにより最適化計算に要する時間は長くなるため、それに伴ってサンプリングタイムが長くなることは考えられ、そのような場合でもより安定性を保持する制御系の提案を本研究における課題とした。

C- $\pi$ 手法に対する有用性が示されたPC- $\pi$ 手法を用いた場合でも、サンプリング間ではFF制御のため路面外乱などに対しては、サンプリングタイムが長くなるにつれてその影響を大きく受けることが予想されていたが、本研究においてそれを確認した。一方で、サンプリングタイムごとに生成される最適状態軌道に対する追従制御器を組み込んだ提案手法を用いることで、サンプリングタイムごとの誤差を低減させ安定性が向上することを確認した。

以上により、MGVの運動制御におけるC- $\pi$ 手法及びPC- $\pi$ 手法の安定性の低下の問題点を示し、それを改善する提案手法の有用性について実証した。

# 第5章 Clock-Based 実時間最適制御による一般車両の運動制御

## 5.1 概要

自動運転車両は移動時間や燃費の観点から、より効率的な移動が求められるが、制限速度や走行領域、乗車人員の負担などを考慮する必要がある。最適制御は運動特性や制約等を考慮して最適な制御入力を決定することができるため効率的な移動に寄与できる。しかし、モデル化誤差による安定性の問題や移動開始前には予期できない障害物の発生など、変化する状況にも対応しなければならない。

そのような問題に対応しつつ効率的な移動を実現する方法として、移動間にも最適制御問題を繰り返し解き続ける実時間最適制御がある。しかし従来の最適化計算時間は移動物体の制御時間に対して長く最適化計算の高速化を図る必要があったため、サンプリング間の制御入力を一定値、最適制御問題の評価区間を有限として高速な最適化計算を実現するモデル予測制御 (MPC) が一般車両の運動制御に用いられてきた。

一方で、近年ではコンピュータの演算能力や最適化アルゴリズムの向上に伴い、最適化計算に要する時間は短くなってきている。そのため、終端時間を自由とし、さらにサンプリング間の入力も時刻参照として考える Clock-Based 実時間最適制御を用いた一般車両の運動制御が可能となりつつあり、より効率的な移動が期待される。しかし、Clock-Based 実時間最適制御を一般車両の運動制御に適用した研究はなく、実機もしくは厳密な数値シミュレーション環境による検証を行うことで Clock-Based 実時間最適制御の一般車両への適用可能性を示す必要がある。さらに最適化計算に要する時間の変化も想定して、サンプリングタイムが長くなる場合に対しても安定性の向上を図る必要がある。

本章では、第2章で記述した制御アルゴリズムを一般車両の運動制御に適用するため、最初に一般車両の運動モデルと制御系を設計する。続いて、最も基本的な想定として直線道路を一定車速で走行する状況を考え、緊急ではない通常走行の範囲でのレーンチェンジ及び前方の停止車両の回避を対象としたフルビークルモデルのシミュレーションソフトウェアを用いた検証を行い、Clock-Based 実時間最適制御の一般車両への適用可能性と提案手法の有用性を示す。

## 5.2 運動モデルの構築とパラメータ同定

実時間最適制御では、逐次最適化問題の計算を行う必要がある。その際には厳密性と計算速度のトレードオフを考えた適切な運動モデルが望まれる。

### 5.2.1 運動モデルの構築

一般車両の運動解析では、ペダリングによる前後方向の運動や、不整路面走行による上下方向の運動、ハンドリングによる横方向の運動などを想定した様々なモデルがあり、想定する状況に応じて適切なモデルを選定しなければならない。本研究における一般車両の運動制御では、基本的な想定として直線道路を一定車速で走行する状況を考え、緊急ではない通常走行の範囲でのレーンチェンジ及び前方の停止車両の回避を対象としている。そのため、急激な加減速やハンドル操作はないものとし、車両運動を地面に投影された剛体としての2次的な運動と考えることができる。従って最適化計算では、本想定における車両運動の基本的な性質を保持しつつ、かつより低い次数として取り扱える等価二輪モデルを基に運動モデルを構築する。

図 5.1 に本研究で用いる一般車両の座標系を示す。状態変数は式 (5.1)、制御変数は式 (5.2) とする。

$$\mathbf{X}(t) = [x(t) \ y(t) \ \psi(t) \ \beta(t) \ r(t) \ \delta_f(t)]^T \quad (5.1)$$

$$\mathbf{U}(t) = \delta_{fc}(t) \quad (5.2)$$

ヨー角を  $\psi$ 、速度を  $V$  とすると、慣性座標系の各軸方向の変化率は式 (5.3) 及び (5.4) で与えられる。

$$\dot{x} = V \cos(\psi + \beta) \quad (5.3)$$

$$\dot{y} = V \sin(\psi + \beta) \quad (5.4)$$

ヨー角  $\psi$  を微分するとヨー角速度  $r$  となるため、式 (5.5) が与えられる。

$$\dot{\psi} = r \quad (5.5)$$

横滑り角とタイヤの横力の関係が線形の範囲では、車両の水平面内の運動は式 (5.6)

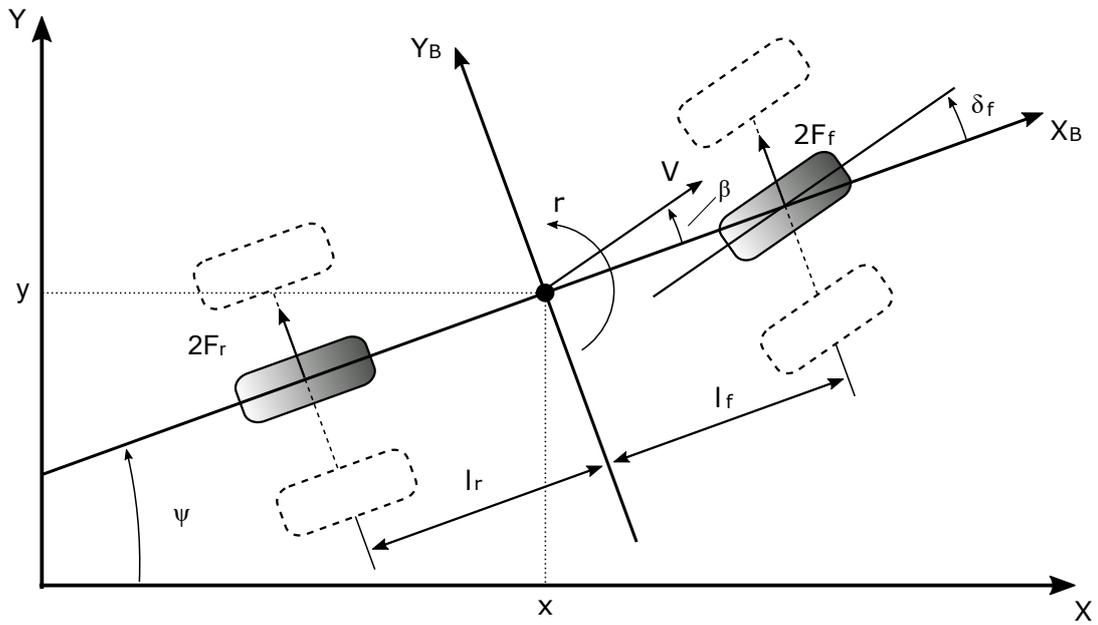


図 5.1 一般車両の座標系

及び (5.7) で表される [109].

$$mV\dot{\beta} + 2(C_f + C_r)\beta + \left(mV + \frac{2}{V}(C_f l_f - C_r l_r)\right)r = 2C_f \delta_f \quad (5.6)$$

$$2(C_f l_f - C_r l_r)\beta + I_z \dot{r} + \frac{2(C_f l_f^2 + C_r l_r^2)}{V}r = 2C_f l_f \delta_f \quad (5.7)$$

式 (5.6) 及び (5.7) を整理すると、式 (5.8) 及び (5.9) が与えられる。

$$\dot{\beta} = -\frac{2(C_f + C_r)}{mV}\beta - \left(1 + \frac{2(C_f l_f - C_r l_r)}{mV^2}\right)r + \frac{2C_f}{mV}\delta_f \quad (5.8)$$

$$\dot{r} = -\frac{2(C_f l_f - C_r l_r)}{I_z}\beta - \frac{2(C_f l_f^2 + C_r l_r^2)}{I_z V}r + \frac{2C_f l_f}{I_z}\delta_f \quad (5.9)$$

実際の一般車両では前輪操舵角は連続的に変化するが、前輪操舵角を制御変数とすると最適化計算において不連続な変化が許容されるので、不連続な解が適用される場合も発生し、非現実的な挙動となる。従って前輪操舵角は状態変数として扱い、操舵角指令  $\delta_{fc}$  に対して一次遅れ系で応答するものと仮定して前輪操舵角の微分方程式を式 (5.10) で与える。

$$\dot{\delta}_f = \frac{1}{\tau_d}(\delta_{fc} - \delta_f) \quad (5.10)$$

式 (5.3), (5.4), (5.5), (5.8), (5.9), 及び (5.10) をまとめると, 一般車両の運動モデルは式 (5.11) となる.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{\beta} \\ \dot{r} \\ \dot{\delta}_f \end{bmatrix} = \begin{bmatrix} V \cos(\psi + \beta) \\ V \sin(\psi + \beta) \\ r \\ -\frac{2(C_f + C_r)}{mV} \beta - \left(1 + \frac{2(C_f l_f - C_r l_r)}{mV^2}\right) r + \frac{2C_f}{mV} \delta_f \\ -\frac{2(C_f l_f - C_r l_r)}{I_z} \beta - \frac{2(C_f l_f^2 + C_r l_r^2)}{I_z V} r + \frac{2C_f l_f}{I_z} \delta_f \\ \frac{1}{\tau_d} (\delta_{fc} - \delta_f) \end{bmatrix} \quad (5.11)$$

## 5.2.2 一般車両のパラメータ同定

車体の諸元は表 5.1 で与えられている．そこで未知のパラメータ  $C_f$ ,  $C_r$  の値をパラメータ同定により導出する．式 (5.8) 及び (5.9) より式 (5.12) 及び (5.13) が求められる．

$$C_f = \frac{mVl_r(\dot{\beta} + r) + I_z\dot{r}}{2(l_f + l_r)(\delta_f - \beta - \frac{l_f}{V}r)} \quad (5.12)$$

$$C_r = \frac{mVl_f(\dot{\beta} + r) - I_z\dot{r}}{2(l_f + l_r)(-\beta - \frac{l_r}{V}r)} \quad (5.13)$$

ここで，フルビークルモデルソフトウェア上にて車速一定でハンドル角を固定した定常円旋回を行う．車速は 60 km/h とし，ハンドル角は 5, 10, 15, 20, 25, 30 deg と変更する． $\beta$  及び  $r$  はシミュレーション結果から得られ， $\dot{\beta}$  及び  $\dot{r}$  は得られた  $\beta$  及び  $r$  の値より数値微分で算出する．

表 5.1 一般車両の運動制御，車体諸元

$m$ [kg]	$l_f$ [m]	$l_r$ [m]	$I_z$ [kg · m <sup>2</sup> ]
1653	1.402	1.646	2765

図 5.2 にハンドル角別の  $C_f$  及び  $C_r$  の推定結果を示す．ハンドル角のステップ入力により  $t = 4$  s 付近以降から定常円旋回中となり  $C_f$  及び  $C_r$  は一定値となっている．しかし，タイヤの垂直荷重の変化や路面摩擦係数などの要因によって  $C_f$  及び  $C_r$  は変動するので， $C_f$  及び  $C_r$  の値は意図的に推定結果の平均値に誤差を持たせた値を使用する． $C_f$  は約 5 % 増加方向に誤差を持たせて  $C_f = 115360$  N/rad， $C_r$  は約 10 % 減少方向に誤差を持たせて  $C_r = 135110$  N/rad とする．

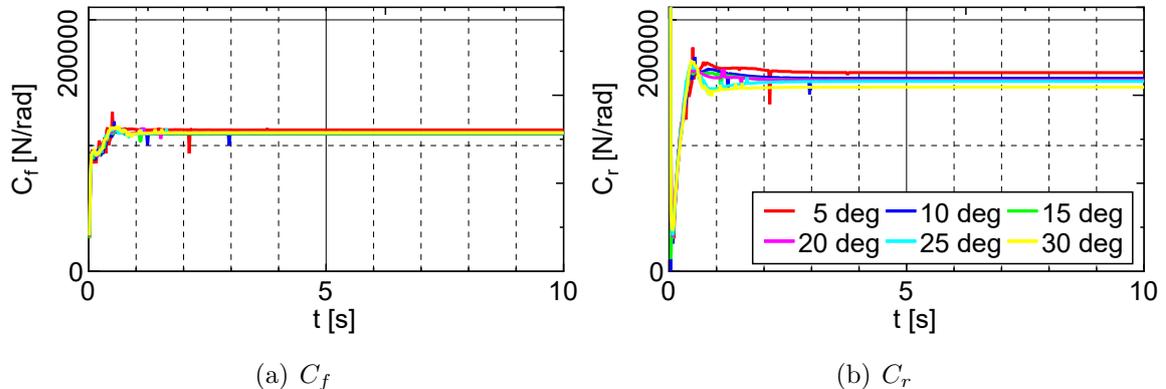


図 5.2 一般車両のパラメータ同定結果

## 5.3 制御系の設計

### 5.3.1 C- $\pi$ 手法における最適制御問題の定式化

p. 31 の式 (2.30) で示した時刻  $t_i \sim t_{i+1}$  における最適制御問題  $\mathbf{P}_c$  の定式化を行う．最初に拘束条件について与えて，続いて評価関数を設定する．

式 (2.27) の運動モデルには前節で構築した式 (5.11) を用いる．式 (2.28) の境界条件は式 (5.14) で与える．

$$\begin{bmatrix} x_i \\ y_i \\ \psi_i \\ \beta_i \\ r_i \\ \delta_{fi} \\ x_{fmin} \\ y_f \\ \psi_f \\ \beta_f \\ r_f \\ \delta_{ff} \end{bmatrix} \leq \begin{bmatrix} x(t_i) \\ y(t_i) \\ \psi(t_i) \\ \beta(t_i) \\ r(t_i) \\ \delta_f(t_i) \\ x(t_f) \\ y(t_f) \\ \psi(t_f) \\ \beta(t_f) \\ r(t_f) \\ \delta_f(t_f) \end{bmatrix} \leq \begin{bmatrix} x_i \\ y_i \\ \psi_i \\ \beta_i \\ r_i \\ \delta_{fi} \\ x_{fmax} \\ y_f \\ \psi_f \\ \beta_f \\ r_f \\ \delta_{ff} \end{bmatrix} \quad (5.14)$$

ここで  $x_i, y_i, \psi_i, \beta_i, r_i, \delta_{fi}$  はサンプリングした状態， $y_f, \psi_f, \beta_f, r_f, \delta_{ff}$  は各想定における目標状態を表す．レーンチェンジ走行や停止車両回避走行では走行条件等により最適な直進方向の移動距離は変化するため， $x$  の終端条件は式 (5.14) のように制限範囲  $x_{fmin} \leq x(t_f) \leq x_{fmax}$  で与える．式 (2.29) の軌道拘束条件は，過大な操舵角となることを防ぐため，式 (5.15) で与える．

$$\delta_{fminFF} \leq \delta_{fc}(t) \leq \delta_{fmaxFF} \quad (5.15)$$

評価関数は式 (5.16) のボルザ型で与える．終端時刻  $t_f$  とサンプリングした時刻  $t_i$  の差を最小化することによる最短時間移動を基本として，積分項の第 1 項にて急激な操舵角の変化を抑えている．また障害物回避では積分項の第 2 項まで考え，式 (5.17) 及び (5.18) の障害物を回避する軌道を生成するための関数を与える．

$$J_c = t_f - t_i + \int_{t_i}^{t_f} \left( W_{ddf} \dot{\delta}_f^2(t) + \xi(t) \right) dt \quad (5.16)$$

$$\xi(t) = e^{(-W_{obs} h_{obs}(x(t), y(t)))} - 1 \quad (5.17)$$

$$h_{obs}(x(t), y(t)) = (x - x_{obs})^p + (y - y_{obs})^p - r_{obs}^p \quad (5.18)$$

以上の設定により、一般車両の実時間最適制御で用いる最適制御問題  $\mathbf{P}_c$  は以下の通り定式化される。

$$(\mathbf{P}_c) \left\{ \begin{array}{ll} \text{評価関数} & \text{式 (5.16)} \\ \text{運動モデル} & \text{式 (5.11)} \\ \text{境界条件} & \text{式 (5.14)} \\ \text{軌道拘束条件} & \text{式 (5.15)} \end{array} \right.$$

擬スペクトル法を用いた最適化計算ソフトウェア [106] にて式 (5.19) 及び (5.20) の最適状態軌道  $\mathbf{X}_i^*(t)$  及び最適制御入力  $\mathbf{U}_i^*(t)$  を導出する。

$$\mathbf{X}_i^*(t) = [ x_i^*(t) \ y_i^*(t) \ \psi_i^*(t) \ \beta_i^*(t) \ r_i^*(t) \ \delta_{fi}^*(t) ]^T \quad (5.19)$$

$$\mathbf{U}_i^*(t) = \delta_{fci}^*(t) \quad (5.20)$$

本研究では式 (5.20) の最適制御入力は、式 (5.21) のフィードフォワード制御入力として定める。

$$\delta_{fcFF} = \delta_{fci}^*(t) \quad (5.21)$$

### 5.3.2 Gain型 M・C- $\pi$ 手法の制御系設計

Gain型 M・C- $\pi$  手法の一般車両への適用では、MGVの場合と同様に、図5.3に示す前方誤差修正モデルに基づいて追従制御器のフィードバック制御入力を生成する。ある時刻において一般車両の位置が  $(x_0, y_0)$  であるとき、式(5.22)及び(5.23)により予測時間  $T_p$  秒後の一般車両の位置を算出する。

$$x_p = x_0 + \dot{x}_0 T_p + \frac{1}{2} \ddot{x}_0 T_p^2 \quad (5.22)$$

$$y_p = y_0 + \dot{y}_0 T_p + \frac{1}{2} \ddot{y}_0 T_p^2 \quad (5.23)$$

算出した  $(x_p, y_p)$  と  $T_p$  秒後の目標位置  $(x_c, y_c)$  の誤差は、回転行列を用いると式(5.24)の車体固定座標系における前後方向の誤差  $\epsilon_x$  及び横方向の誤差  $\epsilon_y$  に変換できる。

$$\begin{bmatrix} \epsilon_x \\ \epsilon_y \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{bmatrix} \begin{bmatrix} x_c - x_p \\ y_c - y_p \end{bmatrix} \quad (5.24)$$

$T_p$  秒後の目標位置  $(x_c, y_c)$  は最適化計算により目標軌道が時間の関数  $(x^*(t), y^*(t))$  で

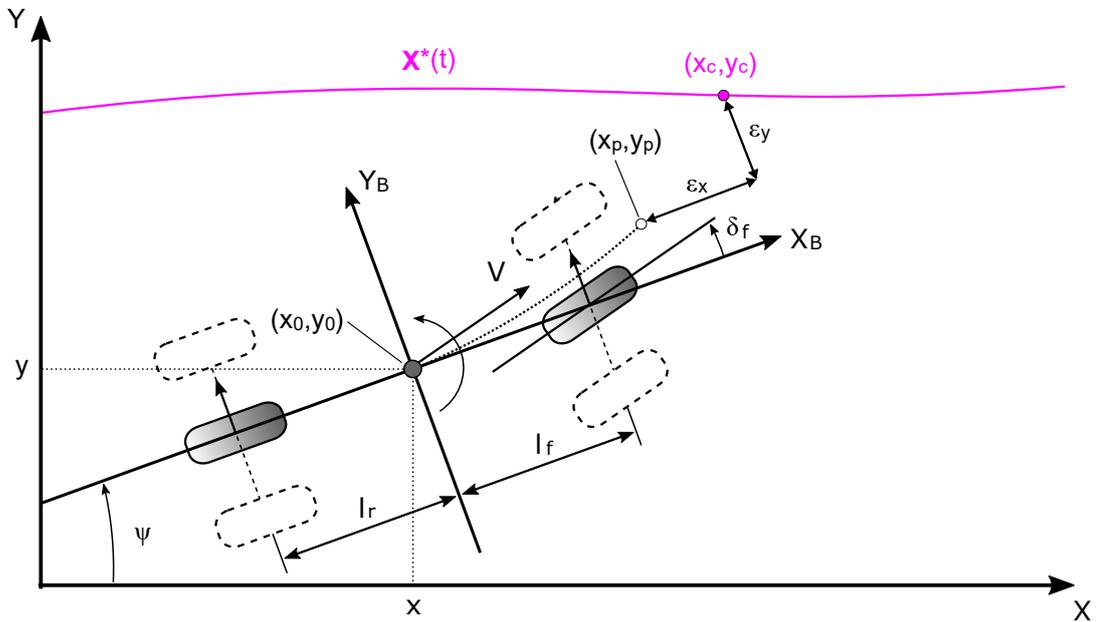


図 5.3 一般車両の運動制御における前方誤差修正モデルによるフィードバック制御入力の生成

与えられているので，式 (5.25) 及び (5.26) となる．

$$x_c = x^*(t + T_p) \quad (5.25)$$

$$y_c = y^*(t + T_p) \quad (5.26)$$

本研究で使用する一般車両は操舵角指令により制御することができるため，式 (5.24) の横方向の誤差を補償する式 (5.27) の比例制御で与える．ここで  $K_{P\delta}$  はフィードバックゲインである．

$$\delta_{fcFB} = K_{P\delta}\epsilon_y \quad (5.27)$$

よって Gain 型 M・C- $\pi$  手法では，フィードフォワード制御入力とフィードバック制御入力を合わせた式 (5.28) で得られる  $\delta_{fc}$  を一般車両へ入力する．

$$\delta_{fc} = \delta_{fcFF} + \delta_{fcFB} \quad (5.28)$$

ただし，目標との誤差が大きいとフィードバック制御入力の値が大きくなる可能性があるので，式 (5.29) により入力値に制限を与える．

$$\delta_{fcmin} \leq \delta_{fc}(t) \leq \delta_{fcmax} \quad (5.29)$$

なお，Gain 型 M・PC- $\pi$  手法の制御系設計については同様であるため，本論文では省略する．

### 5.3.3 MPC 型 M・C- $\pi$ 手法の制御系設計

図 5.4 に時刻  $t_i + \tau_m j$  における MPC 型 M・C- $\pi$  手法を用いた追従制御器の、フィードバック制御入力の生成についての概要を示す。式 (2.5) のパラメータ最適化問題  $\mathbf{P}_m$  の定式化を行う。式 (5.30) を初期値とし、式 (5.11) の運動モデルを用いて式 (5.31) の評価関数を最小化する最適制御入力  $\hat{\delta}_{fc}$  を求める。評価関数は、 $H_p$  ステップ後の  $x$ ,  $y$ ,  $\psi$  の目標値との誤差を評価する。右辺第 4 項では、評価区間内における  $\delta_f$  の変化を抑制する効果がある。

$$\mathbf{X}_R(t_i + \tau_m j) = \begin{bmatrix} x(t_i + \tau_m j) \\ y(t_i + \tau_m j) \\ \psi(t_i + \tau_m j) \\ \beta(t_i + \tau_m j) \\ r(t_i + \tau_m j) \\ \delta_f(t_i + \tau_m j) \end{bmatrix} \quad (5.30)$$

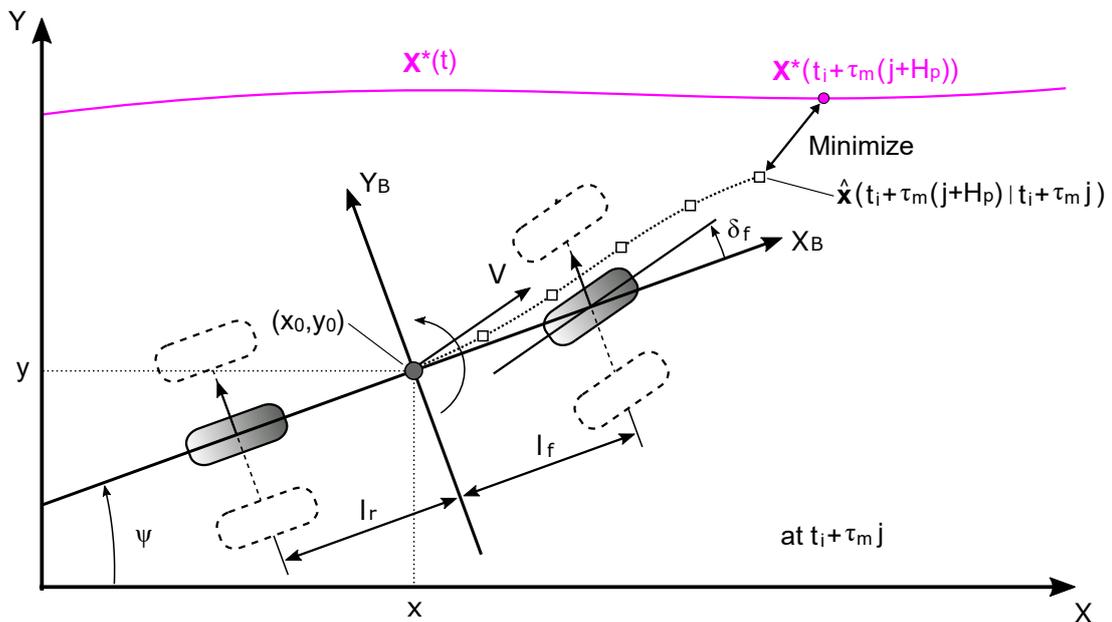


図 5.4 一般車両の運動制御における MPC によるフィードバック制御入力の生成

$$\begin{aligned}
J_m &= W_x(x^*(t_i + \tau_m(j + H_p)) - x(t_i + \tau_m(j + H_p)))^2 \\
&+ W_y(y^*(t_i + \tau_m(j + H_p)) - y(t_i + \tau_m(j + H_p)))^2 \\
&+ W_\psi(\psi^*(t_i + \tau_m(j + H_p)) - \psi(t_i + \tau_m(j + H_p)))^2 \\
&+ W_{d\delta_f} \sum_{k=1}^{H_p} (\hat{\delta}_f(t_i + \tau_m(j + k)|t_i + \tau_m j))^2
\end{aligned} \tag{5.31}$$

MPC は  $H_u$  ステップ以降の入力値は一定であるので、式 (5.32) とする。

$$\hat{\delta}_{fc}(t_i + \tau_m(j + H_u)|t_i + \tau_m j) = \dots = \hat{\delta}_{fc}(t_i + \tau_m(j + H_p)|t_i + \tau_m j) \tag{5.32}$$

また式 (5.33) により制御変数を制限する。

$$\delta_{fcmin} \leq \hat{\delta}_{fc} \leq \delta_{fcmax} \tag{5.33}$$

以上の設定により、一般車両の実時間最適制御で用いる最適制御問題  $\mathbf{P}_m$  は以下の通り定式化される。

$$(\mathbf{P}_m) \left\{ \begin{array}{ll} \text{評価関数} & \text{式 (5.31)} \\ \text{運動モデル} & \text{式 (5.11)} \\ \text{境界条件} & \text{式 (5.30)} \\ \text{軌道拘束条件} & \text{式 (5.33)} \end{array} \right.$$

定式化した非線形計画問題である最適制御問題  $\mathbf{P}_m$  を SQP で解を得て、最適制御入力  $\hat{\delta}_{fc}$  を決定する。導出した  $\hat{\delta}_{fc}$  を式 (5.34) のフィードバック制御入力として定め、式 (5.35) の重み付き合成で得られる  $\delta_{fc}$  を一般車両へ入力する。

$$\delta_{fcFB} = \hat{\delta}_{fc}(t_i + \tau_m j) \tag{5.34}$$

$$\delta_{fc} = W_M \delta_{fcFB} + (1 - W_M) \delta_{fcFF} \tag{5.35}$$

ただし、 $W_M$  は合成の割合を調整する重みであり、 $0 \leq W_M \leq 1$  を満たす。なお、MPC 型 M・PC- $\pi$  手法の制御系設計については同様であるため、本論文では省略する。

### 5.3.4 一般車両の運動制御で使用するパラメータの値（共通）

表 5.2 にサンプリングタイムごとに解く最適制御問題の拘束条件として与えているパラメータの値を示す。本研究では直線道路を一定車速 (60 km/h) で走行する状況を想定しているため、過大な操舵角は大きな横加速度を発生させ、乗員の負担に繋がる。そのため本研究では、FF 制御入力を生成する最適制御問題や最終的に入力する指令では左右 5 deg を限界としている。

表 5.2 一般の運動制御で使用するパラメータの値（拘束条件）

$\delta_{fminFF}$ [deg]	$\delta_{fmaxFF}$ [deg]	$\delta_{fcmin}$ [deg]	$\delta_{fcmax}$ [deg]
-5	5	-5	5

表 5.3 に Gain 型の追従制御器に使用したパラメータの値を示す。MGV の運動制御と同様に、予測時間やゲインの値については FF 制御入力を補正する範囲での追従を目的として決定する。 $T_p$  は設定したサンプリングタイム以下、 $K_{P\delta}$  は FF 制御入力を補正する範囲での追従を目的とした。本章では、それぞれのパラメータを変更した数値シミュレーションを行い、最も結果が良かった値について採用している。

表 5.3 一般車両の運動制御で使用するパラメータの値（Gain 型）

$T_p$ [s]	$K_{P\delta}$ [rad/m]
0.5	3.0

表 5.4 に MPC 型の追従制御器に使用したパラメータの値を示す。MGV の運動制御における検証では実験システムの制約上  $\tau_m$  を 0.2 秒で設定したが、一般車両の運動制御では、システムの制約を受けないものと仮定して MPC のサンプリングタイムは 0.1 秒に設定した。 $H_p$  は 0.5 秒から 1.0 秒先までを評価区間とすることを目的として 5 から 10 の範囲で、それに伴って  $H_u$  は 2 から 8 の範囲で検討した。MPC により生成した FB 制御入力は、FF 制御入力と重み付きで足し合わさせるが、Gain 型の追従制御器と同様に、FF 制御入力を補正する範囲での追従を目的として、 $W_M$  は 0.8 とした。以上の  $\tau_m$ ,  $H_p$ ,  $H_u$ ,  $W_M$  の範囲のもと評価関数のそれぞれの重みを調整した数値シミュレーションを行い、最も結果が良かった値について採用している。

表 5.4 一般車両の運動制御で使用するパラメータの値 (MPC 型)

$\tau_m$ [s]	$H_p$ [-]	$H_u$ [-]	$W_M$ [-]
0.1	5	3	0.8
$W_x$ [-]	$W_y$ [-]	$W_\psi$ [-]	$W_{d\delta}$ [-]
60	60	10	0.15

## 5.4 数値シミュレーション

### 5.4.1 レーンチェンジ走行

・想定及びシミュレーション条件

図 5.5 にレーンチェンジ走行における想定の概要を示す。片側 2 車線の直線道路の右車線を直進し，途中から左車線へレーンチェンジを行う走行を想定する。1 つの車線の道幅は一般的な幅の 3.5 m とし，その中央 ( $y_{c1} = -1.75$ ,  $y_{c2} = 1.75$ ) を目標として走行する。具体的には以下の要領で数値シミュレーションを行う。

1.  $t = 0$  s において  $(x, y) = (0, -1.75)$  から車速  $V$  km/h (一定) で走行開始
2.  $t = 2$  s からレーンチェンジを行うための最適化計算を開始
3.  $t = 15$  s になったならばシミュレーションを終了
4. 道路から逸脱した場合については強制的にシミュレーションを終了

その他の目標状態  $\psi_f$ ,  $\beta_f$ ,  $r_f$  及び  $\delta_{ff}$  はレーンチェンジ前後ともに全て 0 としている。

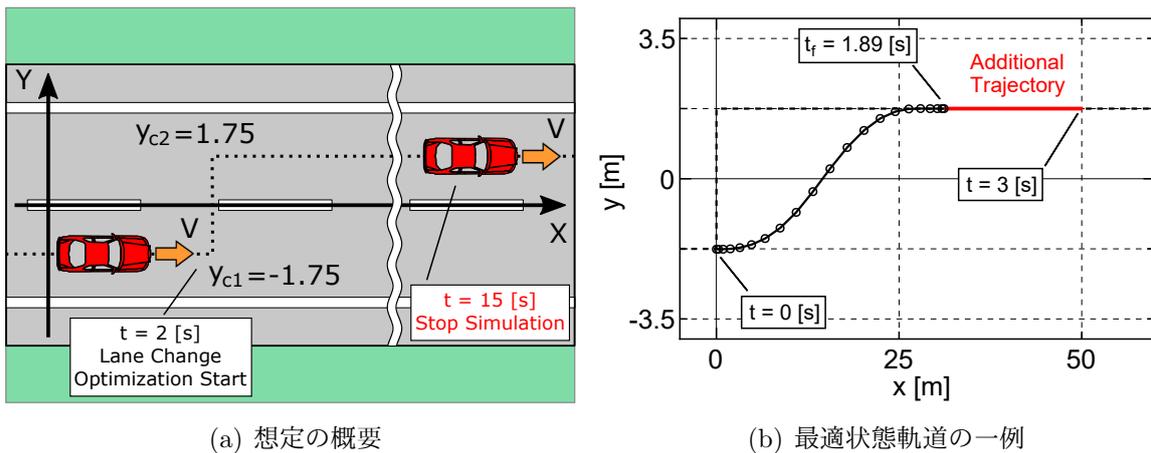


図 5.5 想定概要 (レーンチェンジ走行)

ここでは最短時間でのレーンチェンジ走行を基本としているため，急激な操舵指令が生成される可能性があり，制御系が不安定になる場合が考えられる。また急な旋回は乗車人員に大きな身体的負担を与える。そのため，評価関数の  $W_{df}$  を変化させることで制御系の安定化や急旋回の抑制を図り，それぞれの制御手法を用いた場合における走行への影響について検証を行う。

第 5.2.2 項で行ったパラメータ同定では，線形近似モデルにおいて車速 60 km/h を基準とした定常円旋回によりパラメータ値を導出している。実際の走行において車速が異なる場合も考えられるため，車速を  $V = 40, 50, 70, 80$  km/h と変更した場合についても数値シミュレーションを行って，速度変化に対するロバスト性も検証する。

各制御手法では最適化計算に要する時間も考慮して最適制御解を適用しているが、最短時間問題の最適化計算を行った場合には、制御入力として抽出する定められた長さの時間よりも短い解が生成されることが考えられる。この問題に対して次の対処を行った。図 5.5(b) に  $W_{ddf} = 0$  の最短時間問題で最適化計算を行った際の最適状態軌道の結果を示す。例えば、C- $\pi$ 手法を用いて  $\tau_c = 1.0$  s で走行している場合では、 $t = 1.0 \sim 2.0$  s の区間の制御入力が必要となるが、制御解は約 1.89 秒でレーンチェンジを完了しているため、 $t = 1.89 \sim 2.0$  s の部分の入力が存在しない。そのため制御解が不足する場合には、図 5.5(b) に示すように、最適状態軌道は終端状態を継続的に満たすような軌道を与え、最適制御入力そのまま直進すれば良いので  $\delta_{fc} = 0$  deg を与える。

・数値シミュレーション結果

ここでは、最初にC- $\pi$ 手法、続いてPC- $\pi$ 手法について、サンプリングタイムや $W_{ddf}$ を変更した場合の影響、また追従制御器を組み込んだ場合の効果を検証する。

図 5.6 に、C- $\pi$ 手法を用いた $W_{ddf} = 0$ 、 $\tau_c = 1.0$  s の場合の各時刻における最適軌道の生成状況を示す。レーンチェンジを開始する $t = 2$  s 以降において目標レーンまでのレーンチェンジ軌道は生成されているが、目標レーンとは反対方向に移動し、走行範囲内から逸脱して数値シミュレーションは終了している。

図 5.7 に、 $W_{ddf} = 0$ 、 $\tau_c = 1.0, 2.0$  s の場合の操舵角指令の時間履歴を示す。図 5.7(a) の $\tau_c = 1.0$  s の場合では、 $t = 2$  s から左旋回を行うことでレーンチェンジを開始する解が得られている。しかし左操舵は適用されない最初のサンプリング区間に含まれており、反対に右操舵のみが適用されているため、車両は逆方向へ移動している。図 5.7(b) の $\tau_c = 2.0$  s の場合においても、同様に $t = 2$  s から左旋回を行う解が得られて

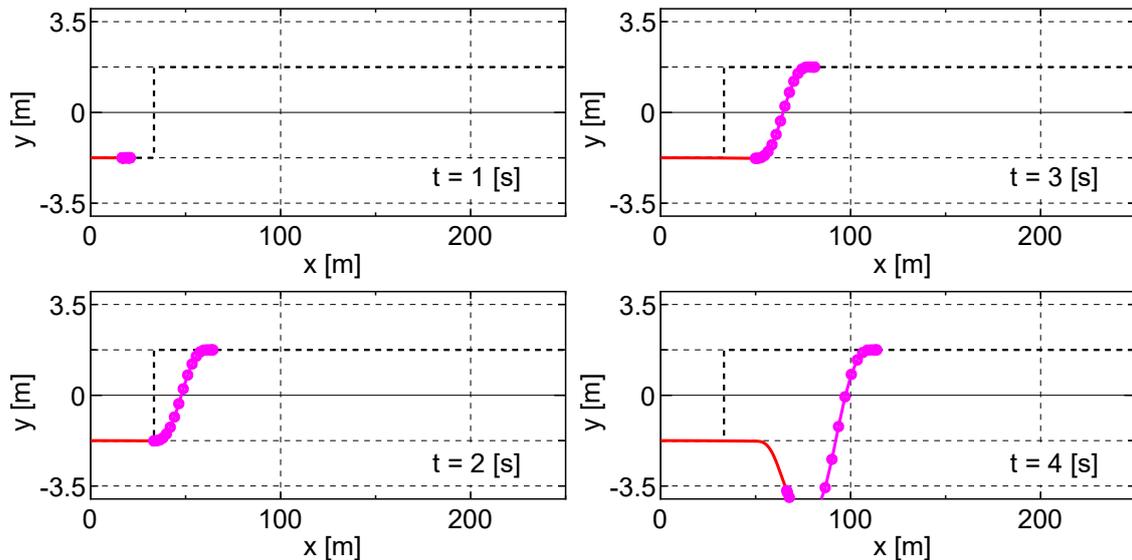


図 5.6 レーンチェンジ走行における最適解の生成状況 (C- $\pi$ 手法,  $W_{ddf} = 0$ ,  $\tau_c=1.0$ )

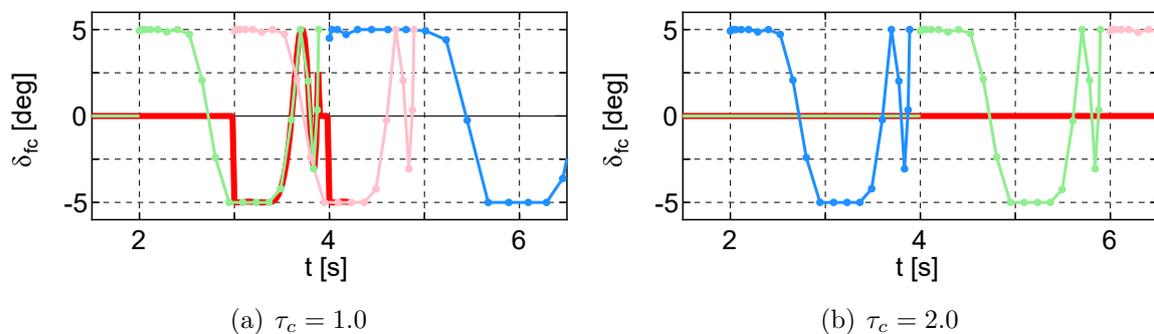


図 5.7 レーンチェンジ走行における操舵角指令の時間履歴 (C- $\pi$ 手法,  $W_{ddf}=0$ )

いるが、適用されない最初の2秒間でレーンチェンジが完了しているため、 $t = 4 \sim 6$  sでは $\delta_{fc} = 0$  degが適用され、もはや制御されていない。

これら2つのケースについては、レーンチェンジに必要な操舵の大部分が適用されない最初のサンプリング区間に含まれているため不安定な挙動となっている。ここで操舵角の変化を緩和することで次のサンプリング区間までレーンチェンジに必要な操舵を残すことができると考え、 $W_{ddf} \neq 0$ として車両運動の安定化を図る。

図5.8及び5.9に $W_{ddf} \neq 0$ とした場合の走行軌跡及び操舵角指令の時間履歴を示す。 $\tau_c = 1.0$  sでは $W_{ddf}$ を1, 2, 3, 4, 5と変化させた。 $W_{ddf}$ の値が大きくなるにつれ目標レーンへの収束性を向上することができている。生成された最適制御入力 $W_{ddf} = 0$ の場合と比較して緩やかになり、 $t = 3$  sから左操舵が適用されレーンチェンジを開始している。 $\tau_c = 2.0$  sでは $W_{ddf}$ を5, 10, 30, 50, 100と変化させた。左操舵が適用されレーンチェンジを開始することはできているが、数値シミュレーションの時間以内には目標レーンに収束することはできなかった。 $W_{ddf}$ が大きい場合には応答が遅くなり、レーンチェンジに必要な時間・距離が長くなる。そのため $W_{ddf}$ がより小さな値で安定化を実現することが望ましい。

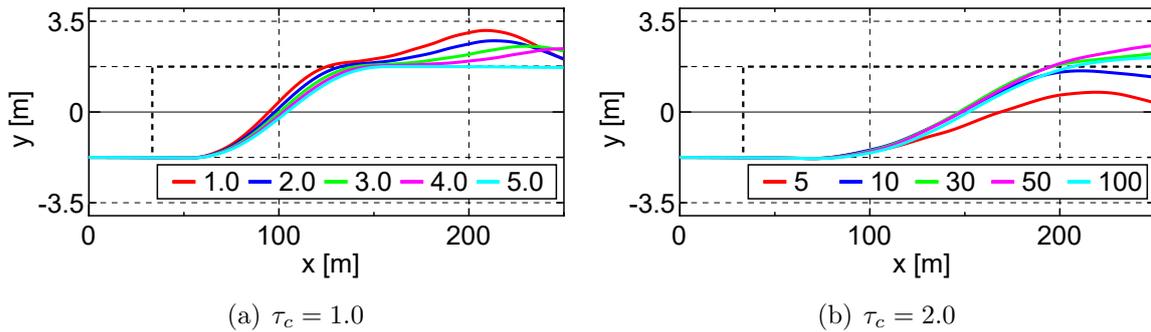


図 5.8 レーンチェンジ走行における走行軌跡 (C- $\pi$ 手法,  $W_{ddf} \neq 0$ )

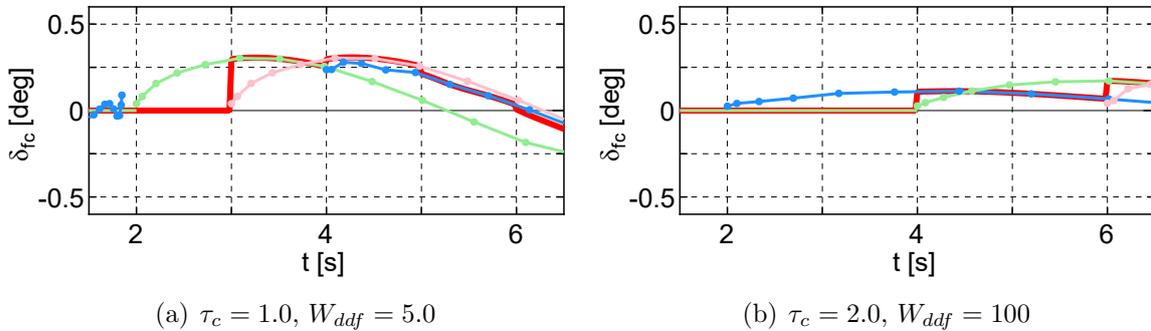


図 5.9 レーンチェンジ走行における操舵角指令の時間履歴 (C- $\pi$ 手法,  $W_{ddf} \neq 0$ )

続いて、Gain型やMPC型の追従制御器を組み込んだM・C- $\pi$ 手法を用いて、最適軌道との誤差を修正することによる $W_{ddf}$ がより小さな値での安定化を試みる。図5.10及び5.11にそれぞれの手法・サンプリングタイムにおいて $W_{ddf}$ を変化させた場合の走行結果を示す。Gain型M・C- $\pi$ 手法を用いた場合には、 $\tau_c = 1.0$  sの $W_{ddf} = 3$ 以上、 $\tau_c = 2.0$  sの $W_{ddf} = 5$ 、MPC型M・C- $\pi$ 手法を用いた場合には全ての走行においてレーンチェンジを行うことができています。

特にMPC型M・C- $\pi$ 手法を用いた場合には、Gain型M・C- $\pi$ 手法と比較して $W_{ddf}$ が小さい場合でもレーンチェンジを行うことができています。図5.7(b)のように $W_{ddf} = 0$ では左右のどちらの操舵も適用されないが、 $W_{ddf} = 1 \sim 2$ ではレーンチェンジ終了時の修正操舵（あて舵）が多くなるため、Gain型M・C- $\pi$ 手法を用いた場合でも最適軌道の誤差の修正できていない。しかしながら、MPC型M・C- $\pi$ 手法を用いた場合には、修正操舵の部分が80%に減少し、代わりにMPCにより生成された操舵角指令として制限値の20%が適用される。その結果、左操舵となるため $W_{ddf}$ が小さい場合でもレーンチェンジを行うことができています。

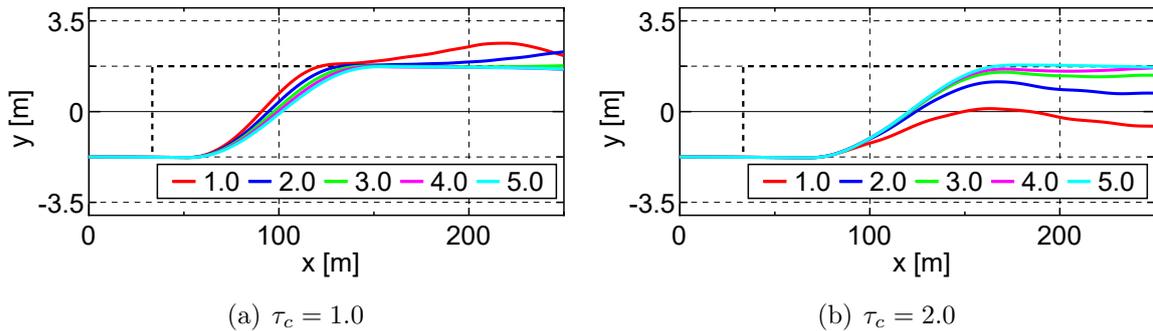


図 5.10 レーンチェンジ走行における走行軌跡（Gain型M・C- $\pi$ 手法， $W_{ddf} \neq 0$ ）

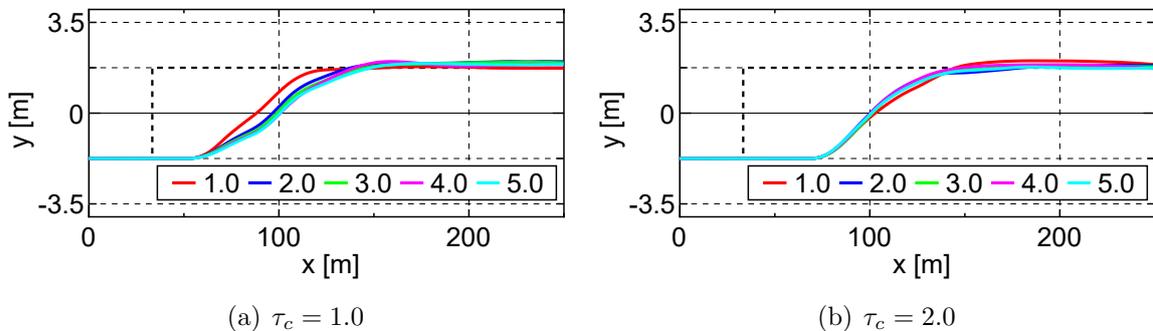


図 5.11 レーンチェンジ走行における走行軌跡（MPC型M・C- $\pi$ 手法， $W_{ddf} \neq 0$ ）

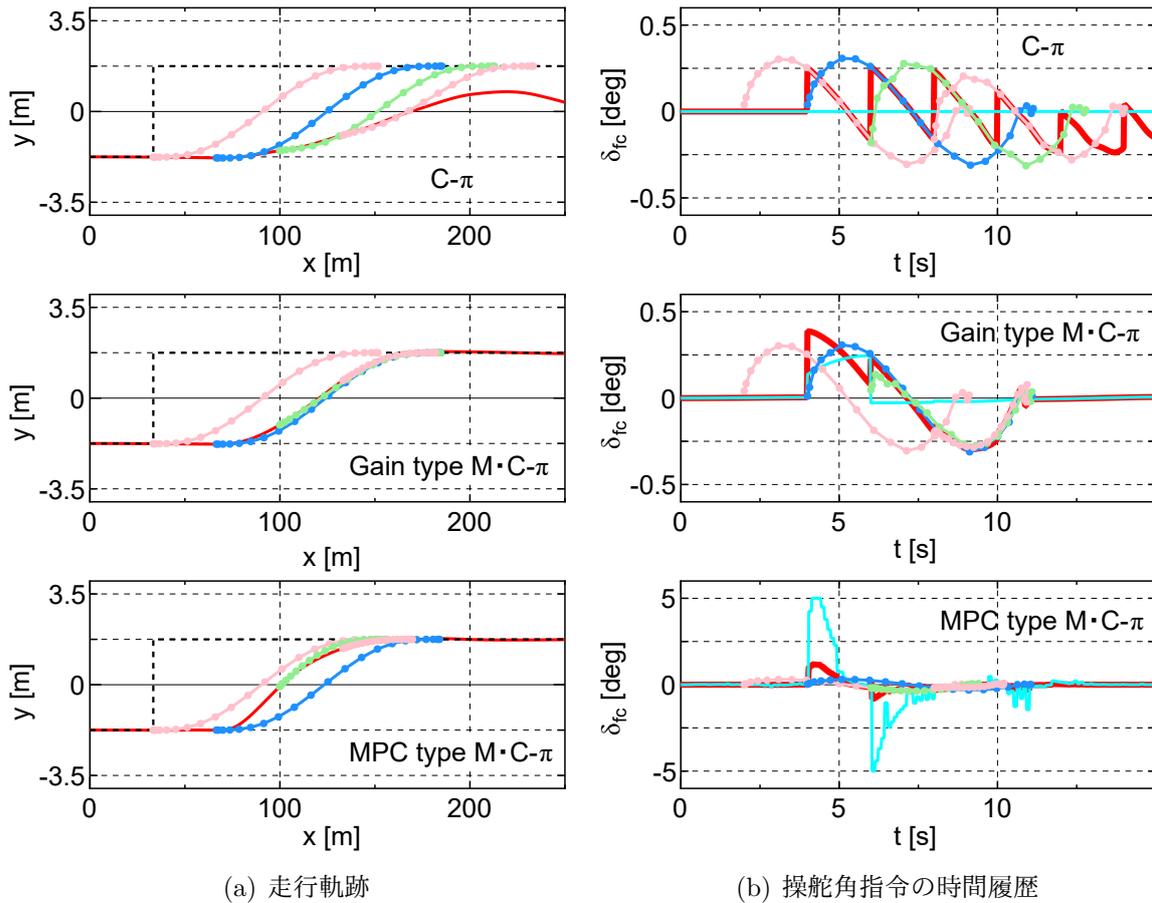


図 5.12 レーンチェンジ走行における走行軌跡及び操舵角指令の時間履歴 ( $W_{ddf} = 5$ ,  $\tau_c = 2.0$ )

図 5.12 に、 $\tau_c = 2.0$  s,  $W_{ddf} = 5$  における各手法の走行軌跡及び操舵角指令の時間履歴を示す。図中のシアンの実線はそれぞれの追従制御器により生成したフィードバック制御入力を表している。MPC 型 M·C- $\pi$  手法の操舵角の時間履歴のみフィードバック制御入力を表示するため、縦軸を 10 倍のスケールとしている。

レーンチェンジを行う 1 つ目の最適制御解は、いずれの手法においても同じ解が得られている。最初のサンプリング区間に左操舵の大部分が含まれており、C- $\pi$  手法を用いた場合では各サンプリング間隔で左右の操舵の一部が入力されているため、十分な左旋回ができていない。一方、Gain 型 M·C- $\pi$  手法や MPC 型 M·C- $\pi$  手法を用いた場合では、最適軌道との誤差を修正するための左操舵のフィードバック制御入力生成されている。サンプリング時刻ごとに予測軌道との誤差が小さくなり、目標レーンに収束することができている。

図 5.13 に PC- $\pi$  手法を用いた  $W_{ddf} = 0$ ,  $\tau_c = 1.0$  s の場合の各時刻における最適軌道の生成状況を示す。サンプリングタイムごとに予測した次のサンプリング時刻における状態から、目標レーンまでの最適軌道が生成されてレーンチェンジを行っている。

図 5.14 に操舵角指令の適用状況、と車両の横加速度  $a_y$  の時間履歴を示す。  $t = 2$  s 付近の操舵角指令は、レーンチェンジ前の直進走行中における目標レーンとの微小な

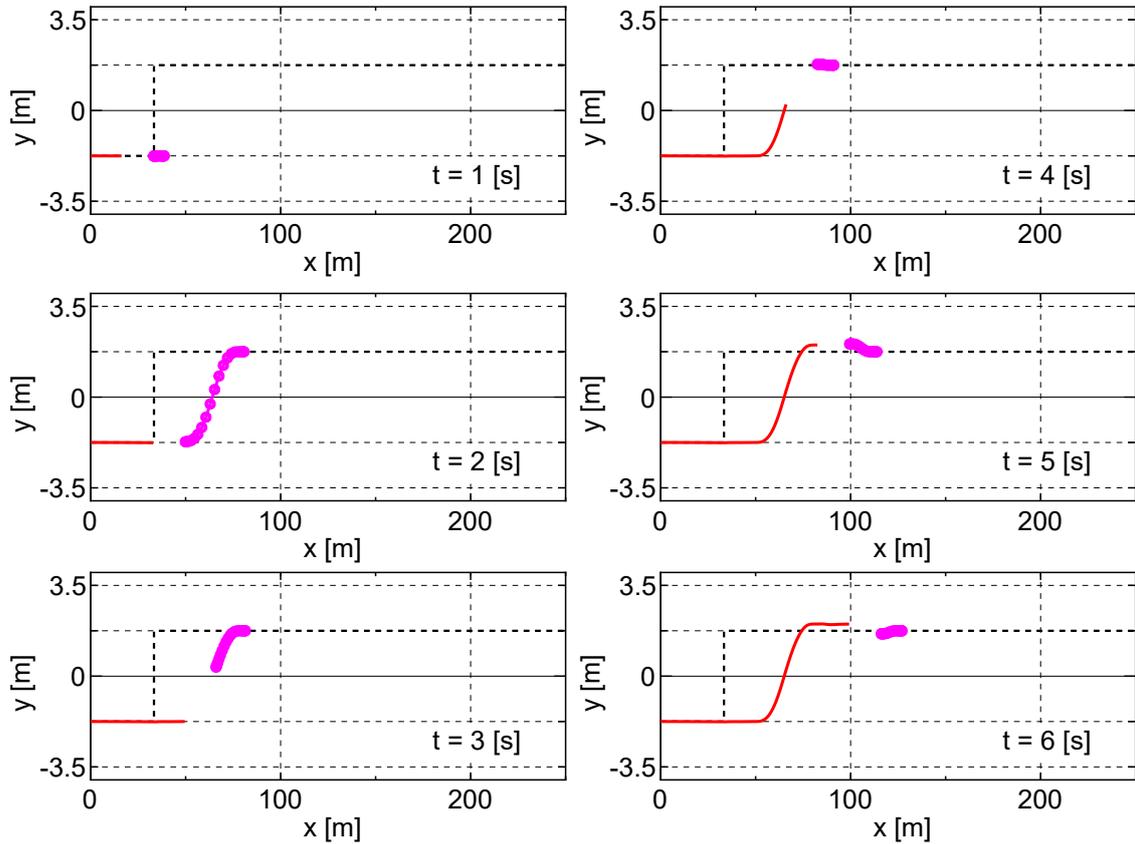


図 5.13 レーンチェンジ走行における最適解の生成状況 (PC- $\pi$  手法,  $W_{ddf} = 0$ ,  $\tau_c=1.0$ )

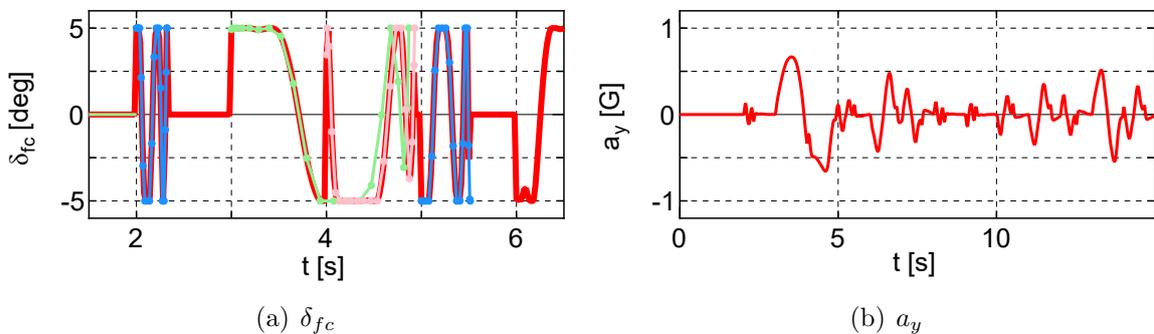


図 5.14 レーンチェンジ走行における操舵角指令及び横加速度の時間履歴 (PC- $\pi$  手法,  $W_{ddf} = 0$ ,  $\tau_c = 1.0$ )

誤差を修正するために生成されているが、 $W_{ddf} = 0$  であるため急激な変化が許容されている。PC- $\pi$  手法を用いた場合には、 $t = 2$  s のサンプリングにより生成した  $t = 3$  s からのレーンチェンジのための左操舵がそのまま適用されているため、C- $\pi$  手法を用いた場合に発生した修正操舵による影響は見られないが、最大横加速度が約 0.67 G と大きな値となっている。そこで  $W_{ddf}$  に値を設定して操舵角の変化を抑制することで横加速度を小さくすることを考える。

図 5.15 及び 5.16 に  $W_{ddf}$  を変更した場合の走行軌跡と横加速度の変化を示す。 $W_{ddf}$  を 0.01, 0.02, 0.03, 0.04, 0.05 と増やすことで、最大横加速度を 0.37, 0.29, 0.25, 0.22, 0.21 G と抑えることができている。

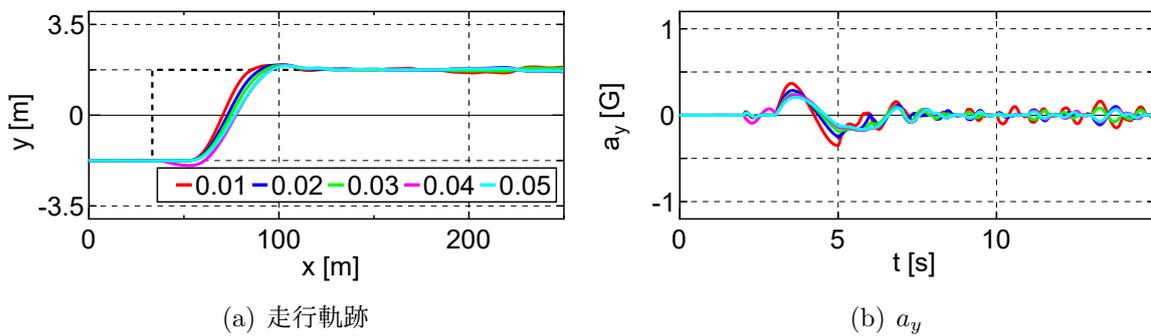


図 5.15 レーンチェンジ走行における走行軌跡及び横加速度の時間履歴 (PC- $\pi$  手法,  $W_{ddf} \neq 0$ ,  $\tau_c = 1.0$ )

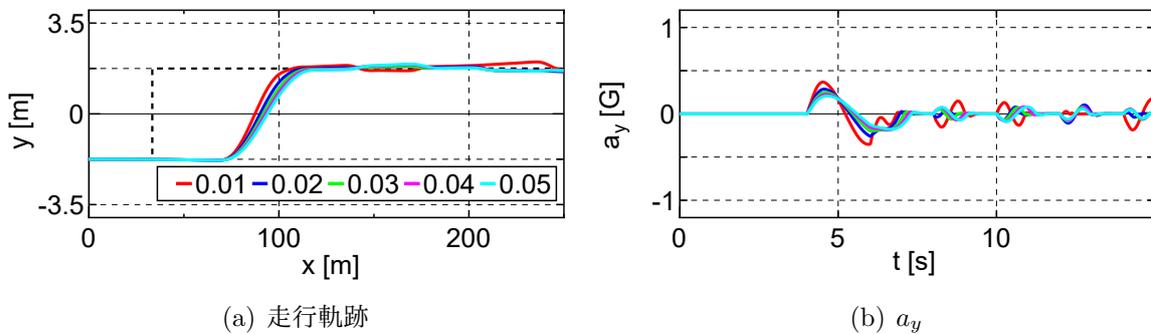


図 5.16 レーンチェンジ走行における走行軌跡及び横加速度の時間履歴 (PC- $\pi$  手法,  $W_{ddf} \neq 0$ ,  $\tau_c = 2.0$ )

PC- $\pi$  手法を用いた場合には、Gain 型や MPC 型の追従制御器を組み込まなくてもレーンチェンジを行うことができ、また  $W_{ddf}$  に値を設定することで最大横加速度を抑えることができた。図 5.17 及び 5.18 に Gain 型 M・PC- $\pi$  手法及び MPC 型 M・PC- $\pi$  手法を用いた場合の結果を示す。どちらの手法も PC- $\pi$  手法と同等の結果が得られている。

図 5.19 に走行速度を変更した場合の走行結果を示す。C- $\pi$  手法で安定化することの

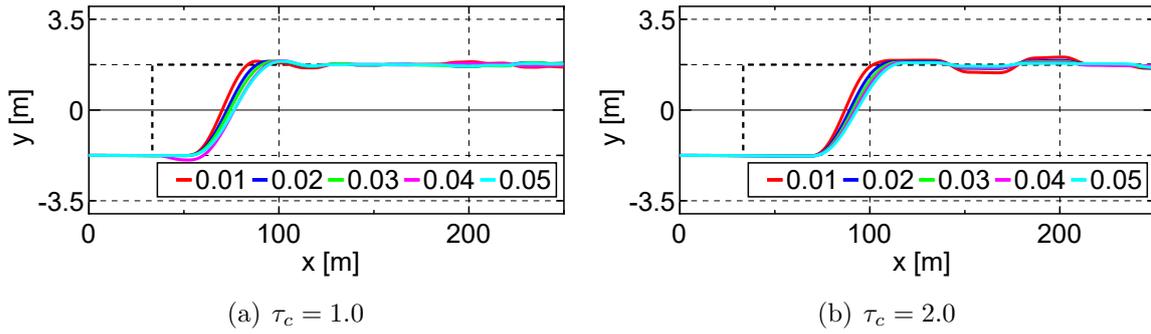


図 5.17 レーンチェンジ走行における走行軌跡 (Gain 型 M・PC- $\pi$  手法,  $W_{ddf} \neq 0$ )

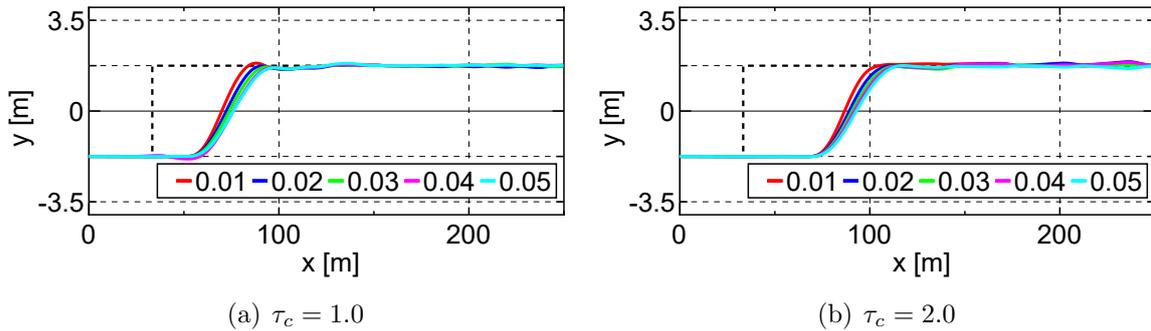
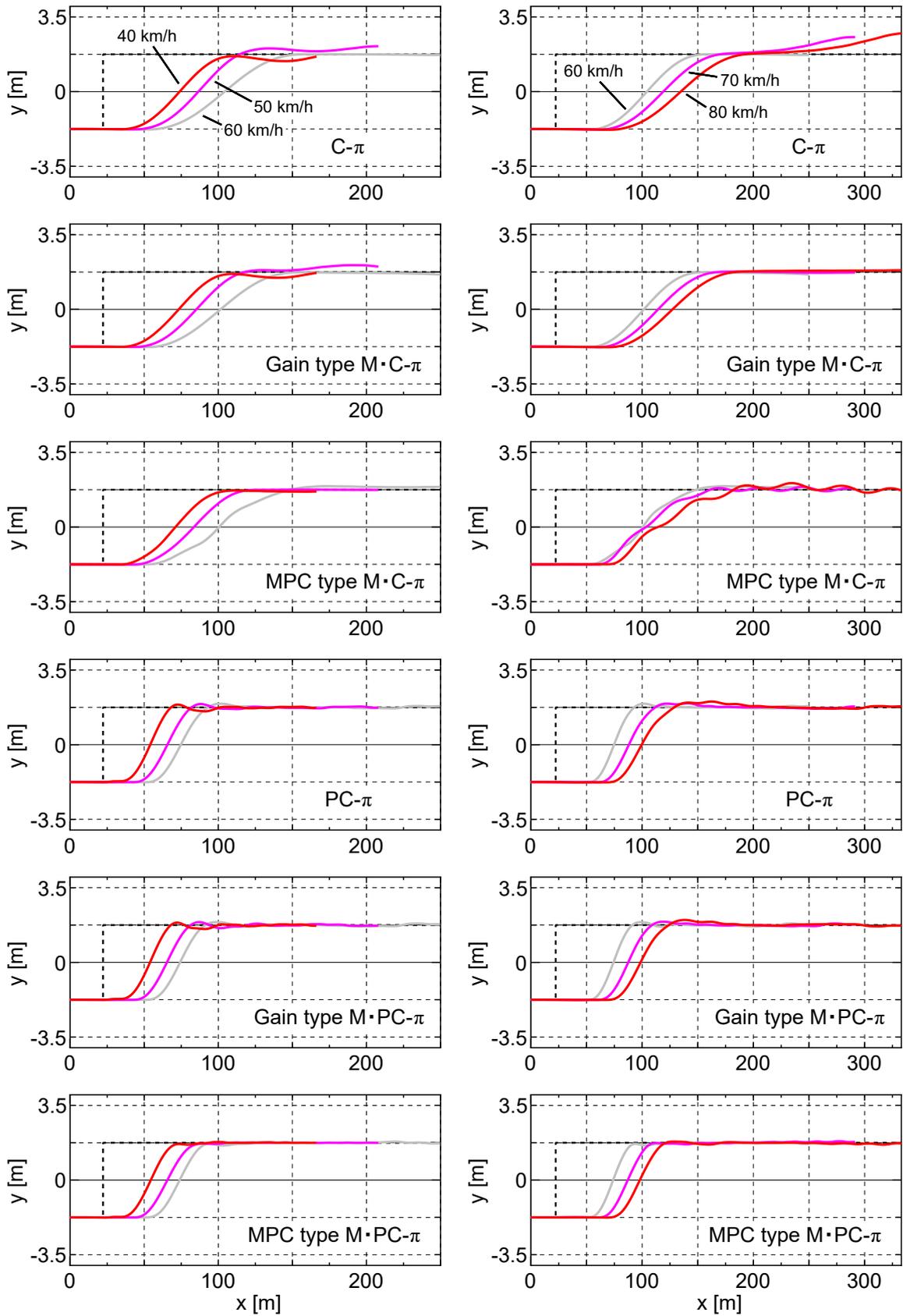


図 5.18 レーンチェンジ走行における走行軌跡 (MPC 型 M・PC- $\pi$  手法,  $W_{ddf} \neq 0$ )

できた  $\tau_c = 1.0$  s の場合について, C- $\pi$  手法, Gain 型 M・C- $\pi$  手法, MPC 型 M・C- $\pi$  手法は  $W_{ddf} = 5$ , PC- $\pi$  手法, Gain 型 M・PC- $\pi$  手法, MPC 型 M・PC- $\pi$  手法は  $W_{ddf} = 0.03$  の走行結果を示している.

C- $\pi$  手法を用いた場合では, 低速側, 高速側のどちらにおいても車速 60 km/h と比べて車両の挙動は不安定になった. 一方, Gain 型 M・C- $\pi$  手法を用いた場合では, 高速側への変更は安定となるが, 低速側への変更は挙動が不安定となった. 低速側では, 予測位置が近くなるため誤差が小さくなり, 同じゲインではフィードバック制御入力が小さくなるため, C- $\pi$  手法と同様な挙動をしている. 反対に, MPC 型 M・C- $\pi$  手法を用いた場合では, 低速側への変更は安定となるが, 高速側への変更は挙動が不安定となった. その理由として, 追従する最適状態軌道と次の最適状態軌道の誤差が高速になるほど大きくなるため, 予測位置が交互に大きく変わり, 振動的な走行となるためである. Gain 型や MPC 型の M・C- $\pi$  手法を用いた場合については, 速度変更に対する安定した走行という面での課題があり, 今後は様々な速度域に対する安定化について検討する必要がある.

PC- $\pi$  手法, Gain 型 M・PC- $\pi$  手法, MPC 型 M・PC- $\pi$  手法を用いた場合では, 速度が変わるためレーンチェンジを開始する位置は変化しているが, 低速側, 高速側のどちらにおいても車速 60 km/h と比べて同等の走行結果となった.



(a)  $V = 40, 50, 60$

(b)  $V = 60, 70, 80$

図 5.19 車速変更を変更した場合のレーンチェンジ走行結果,  $\tau_c = 1.0$

## 5.4.2 停止車両回避走行

### ・想定及びシミュレーション条件

図 5.20 に停止車両回避走行における想定の概要を示す。片側 2 車線の直線道路の右車線を直進し、走行途中で認知した前方の停止車両を回避する走行を想定する。具体的には以下の要領で数値シミュレーションを行う。

1.  $t = 0$  s において  $(x, y) = (0, -1.75)$  から車速  $V$  km/h (一定) で走行開始
2.  $t = 2$  s から停止車両を回避するための最適化計算を開始
3.  $t = 15$  s になったならばシミュレーションを終了
4. 道路から逸脱した場合には強制的にシミュレーションを終了

その他の目標状態  $\psi_f$ ,  $\beta_f$ ,  $r_f$  及び  $\delta_{ff}$  は全て 0 としている。

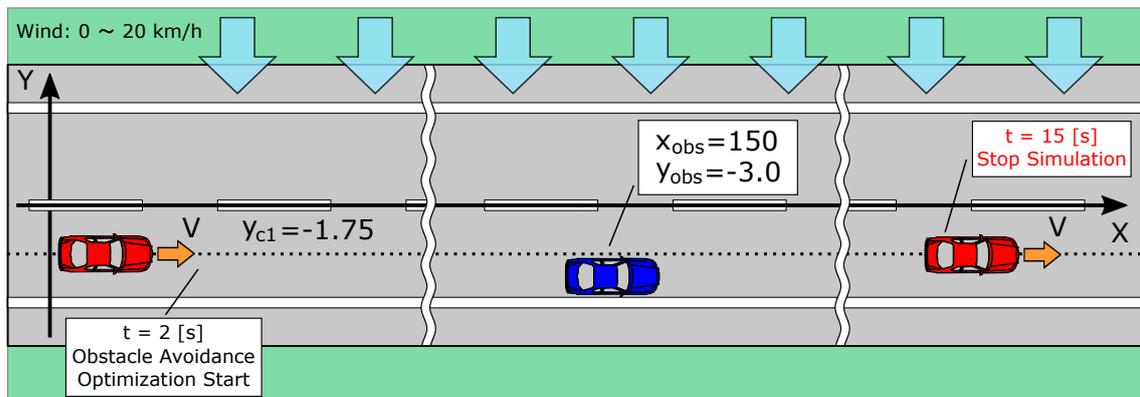


図 5.20 想定概要 (停止車両回避走行)

また、最適化計算では考慮されていない外乱に対する影響についても検証する。MGV の運動制御における路面外乱に対する影響の場合と同様に、単純かつ明瞭な外乱を設定することで、外乱に対する各手法の基本的な性能を明らかにすることを着眼とする。ここでは、走行車両を停止車両に近づけるような一定風速の横風の影響を定常的に受ける状況を仮定する。

・数値シミュレーション結果

【横風なし】

図 5.21 に、C- $\pi$  手法を用いた場合の各時刻における最適軌道の生成状況を示す。黒色の破線が目標とする走行レーン，赤色の実線が走行車両の重心の走行軌跡，マゼンダがサンプリングタイムごとに生成した最適軌道，茶色の実線が停止車両，茶色の破線が衝突エリアを表す。表示している座標の縦横比の関係で，停止車両は縦長の枠となっている。

停止車両を確認することができる  $t = 2 \text{ s}$  以降において，停止車両より遠方までの回避軌道を生成している。ただし， $t = 2 \text{ s}$  及び  $t = 4 \text{ s}$  では衝突エリアに最適軌道が通過している。擬スペクトル法では，離散点を境界付近で密，中央付近で疎になるような離散化を行うことが一つの特徴である。そのため，本想定のような場合において離散点が障害物を跨いでしまうことがある。しかしながら，本研究では実時間で最適制御解を更新しており，車両が障害物に近づくにつれ最適軌道の障害物付近の離散点は密になるため， $t = 5 \text{ s}$  以降では衝突エリアを回避した軌道を得ることができ，停止車両を回避して元の走行レーンに復帰している。

図 5.22 にサンプリングタイム別の各手法の走行結果を示す。 $\tau_c = 1.0 \text{ s}$  では全ての手法で回避できているが， $\tau_c = 2.0 \text{ s}$  では C- $\pi$  手法のみ衝突エリアに進入している。PC- $\pi$  手法，Gain 型 M・PC- $\pi$  手法，及び MPC 型 M・PC- $\pi$  手法を用いた場合では，実軌道と予測軌道の誤差は少なく，追従制御器を組み込んでいない PC- $\pi$  手法でも  $\tau_c = 2.0 \text{ s}$  において停止車両を回避することができている。図 5.23 に C- $\pi$  手法，Gain 型 M・C- $\pi$  手法，及び MPC 型 M・C- $\pi$  手法を用いた場合における操舵角指令の適用状況を示す。いずれの手法でも停止車両を確認した  $t = 2.0 \text{ s}$  から左旋回を開始し，右旋回，左旋回の順で元の走行レーンに復帰している。解が適用される  $t = 4.0 \sim 6.0 \text{ s}$  の部分では回避後の右操舵が含まれており，C- $\pi$  手法を用いた場合ではそのまま逆操舵が適用されている。一方，Gain 型 M・C- $\pi$  手法及び MPC 型 M・C- $\pi$  手法を用いた場合では，最適軌道との誤差を修正する左操舵のフィードバック制御入力生成されており， $t = 4.0 \sim 6.0 \text{ s}$  では左操舵が入力されている。サンプリング間において Gain 型や MPC 型などの追従制御により最適軌道に追従を行うことで安定性が向上し，停止車両を回避することができている。

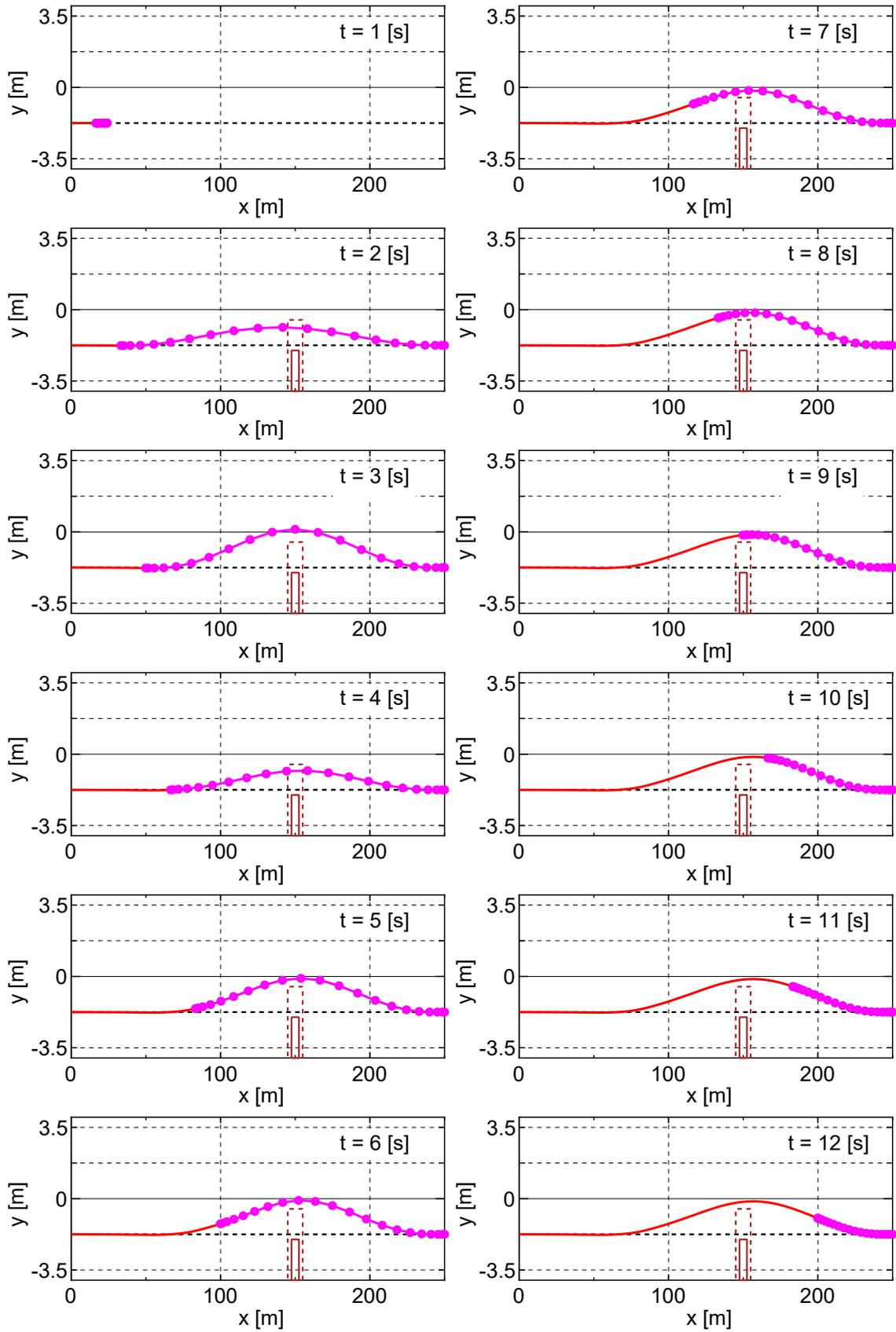


図 5.21 停止車両回避走行，横風なし， $C-\pi$ 手法の最適解の生成状況， $\tau_c=1.0$

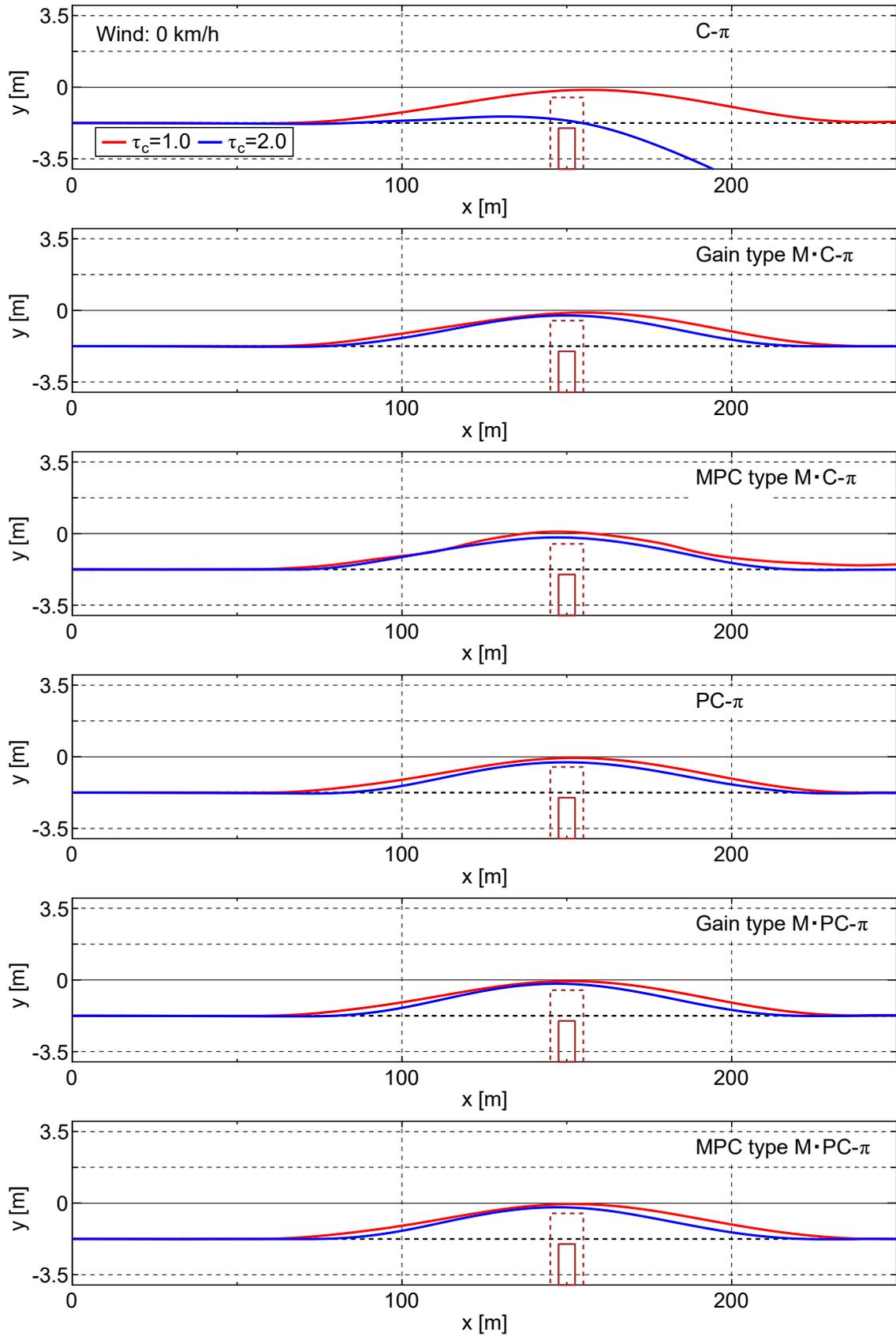
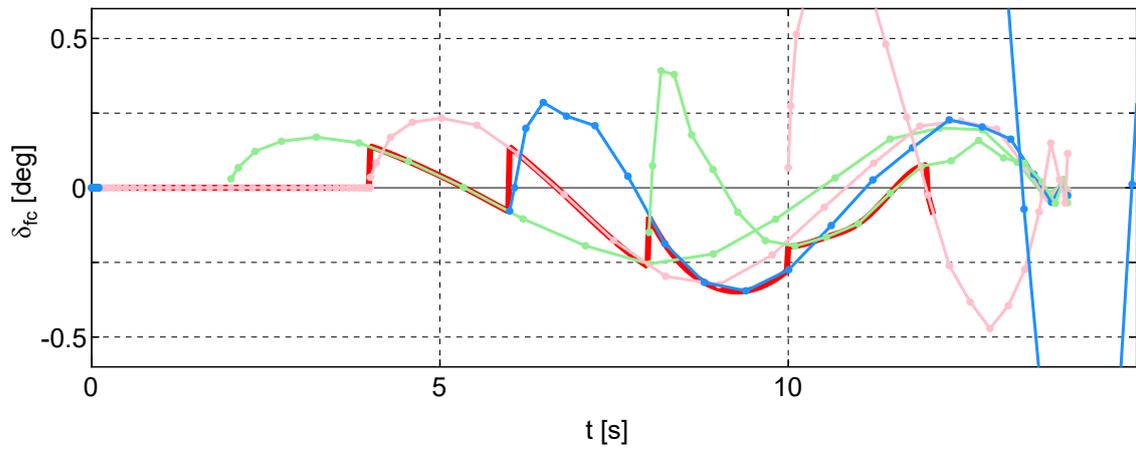
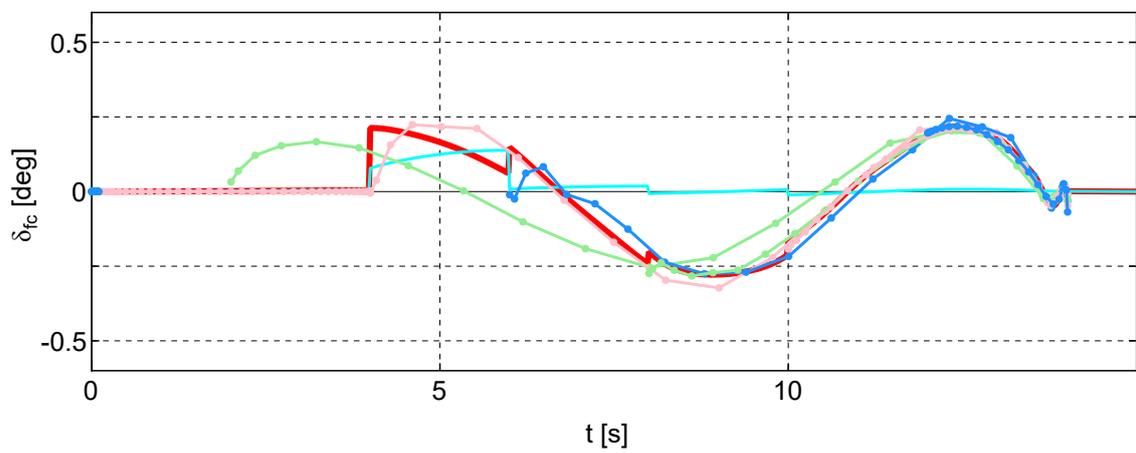


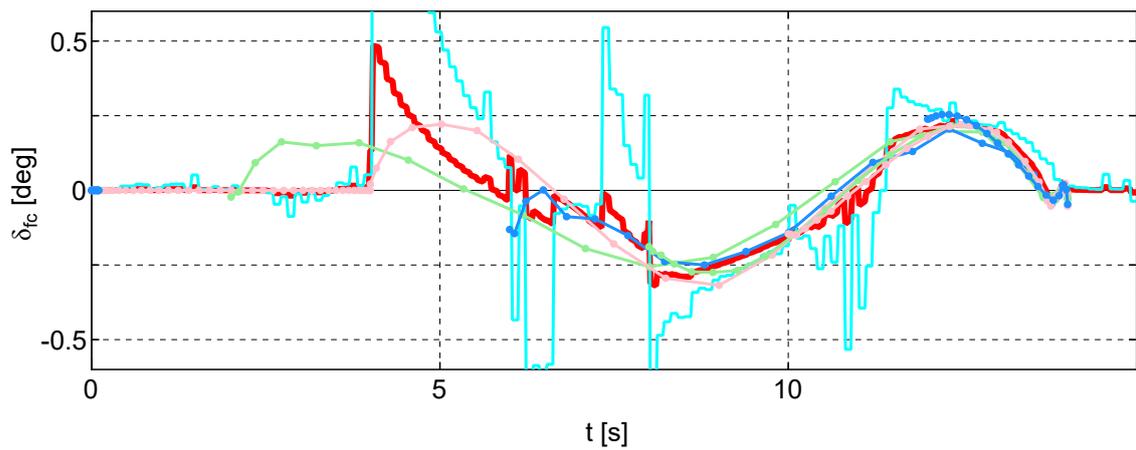
図 5.22 停止車両回避走行，横風なし，各手法の走行結果， $\tau_c = 1.0, 2.0$



(a) C- $\pi$  手法



(b) Gain 型 M · C- $\pi$  手法



(c) MPC 型 M · C- $\pi$  手法

図 5.23 停止車両回避走行，横風なし，操舵角指令の時間履歴， $\tau_c = 2.0$

### 【横風あり】

横風が無い場合は、PC- $\pi$ 手法は実軌道と予測軌道の誤差は小さく制御性能は高かった。しかし、横風がある場合では、PC- $\pi$ 手法を用いた場合でも実軌道は横風の影響を受けるので、予測軌道との誤差が生じて安定性が低下することが考えられる。図5.24に $\tau_c = 1.0$  sで風速を5 ~ 20 km/hと変更した場合の各手法の走行結果を示す。 $\tau_c = 1.0$  sで横風がない場合では全ての手法で停止車両を回避できていたが、横風の速度が速くなるにつれて走行車両は障害物へ近づき、衝突エリアに進入するケースも確認できる。しかしながら、C- $\pi$ 手法やPC- $\pi$ 手法と比べて、Gain型やMPC型の追従制御器を組み込んだ場合では風の影響が軽減されており、さらにMPC型では全ての風速で回避することができている。

図5.25にサンプリングタイムが長くなった $\tau_c = 2.0$  sの場合の走行結果を示す。C- $\pi$ 手法を用いた場合では、横風の影響に加え、前述した修正操舵の影響により全ての風速で衝突エリアに進入している。PC- $\pi$ 手法を用いた場合でも、 $\tau_c = 1.0$  sの場合と比べて、風の影響を大きく受けている。Gain型M・C- $\pi$ 手法やGain型M・PC- $\pi$ 手法を用いた場合でも風の影響は大きくなっているが、C- $\pi$ 手法やPC- $\pi$ 手法と比較すると影響が軽減している。MPC型M・C- $\pi$ 手法やMPC型M・PC- $\pi$ 手法を用いた場合では、 $\tau_c = 1.0$  sと同様に、 $\tau_c = 2.0$  sでも全ての風速で停止車両を回避することができている。

図5.26及び5.27に、風速20 km/h、 $\tau_c = 2.0$  sにおけるPC- $\pi$ 手法、Gain型M・PC- $\pi$ 手法、及びMPC型M・PC- $\pi$ 手法を用いた場合の最適制御解の生成状況を示す。PC- $\pi$ 手法を用いた場合では、図5.27(a)より $t = 4.0$  sからの左操舵はそのまま適用されているが、図5.26(a)において風の影響で予測軌道より停止車両へ近づく方向へ誤差が生じていることが分かる。Gain型M・PC- $\pi$ 手法を用いた場合では、横風の影響による予測軌道との誤差を補償するため、左操舵のフィードバック制御入力生成されている。そのため、PC- $\pi$ 手法よりは予測軌道との誤差は小さくなっているが、風速が最も大きい20 km/hでは発生した誤差に対する十分なフィードバック制御入力が得られず衝突エリアに進入している。MPC型M・PC- $\pi$ 手法を用いた場合においてもフィードバック制御入力として左操舵が生成されている。その結果、大きな修正を行うフィードバック制御入力を適用することができ、風速20 km/hにおいても停止車両を回避することができている。

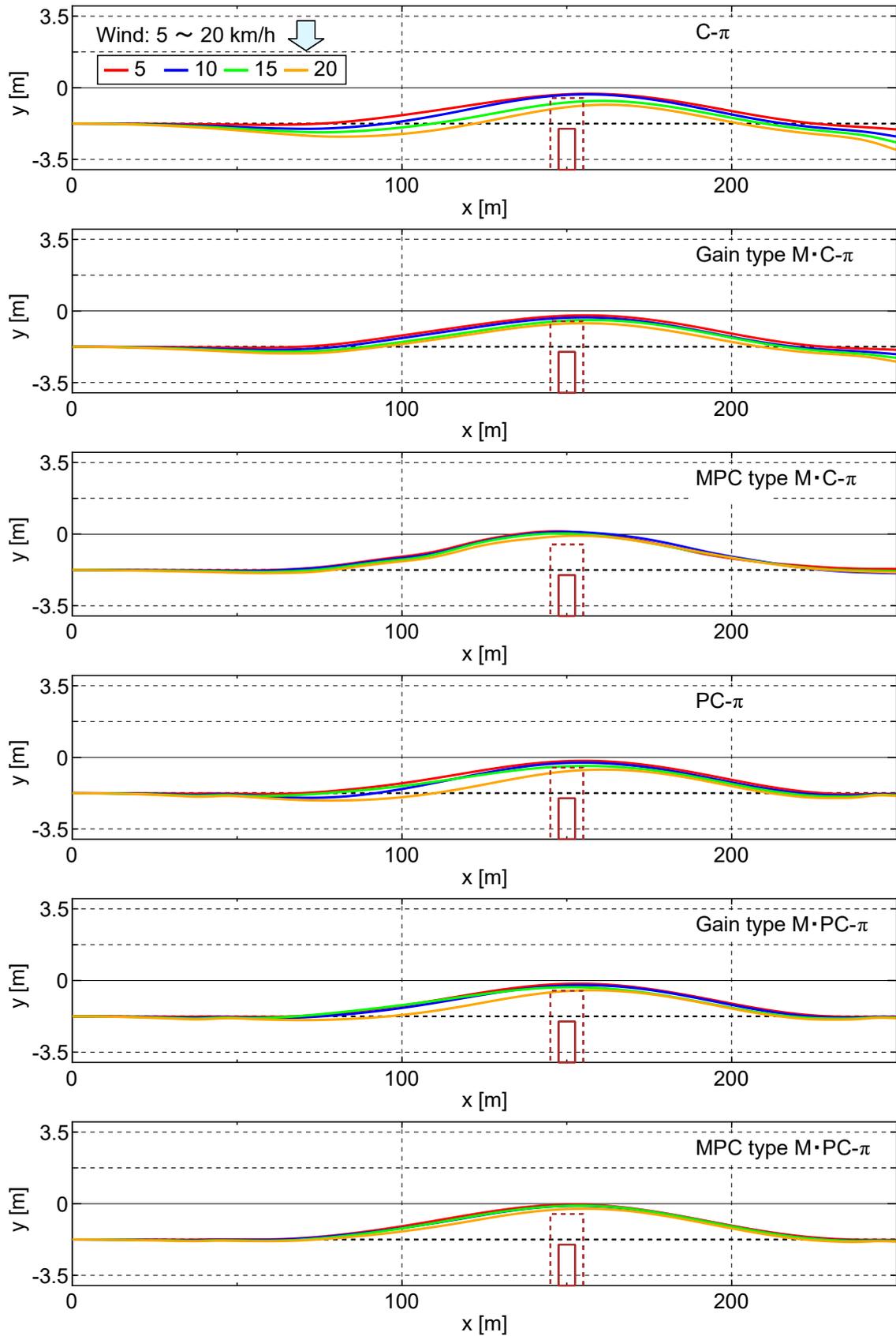


図 5.24 停止車両回避走行，横風あり，各手法の走行結果， $\tau_c = 1.0$

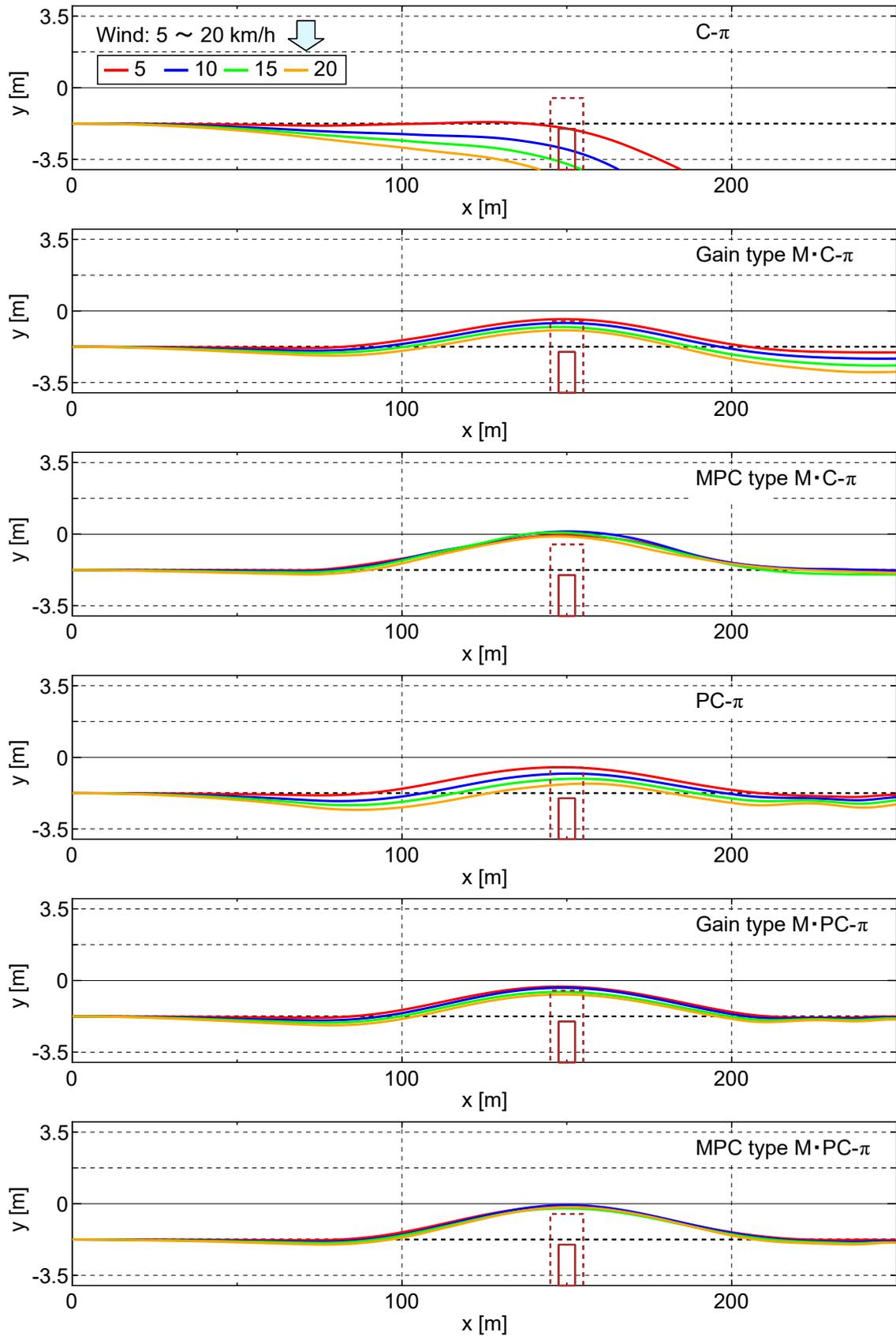
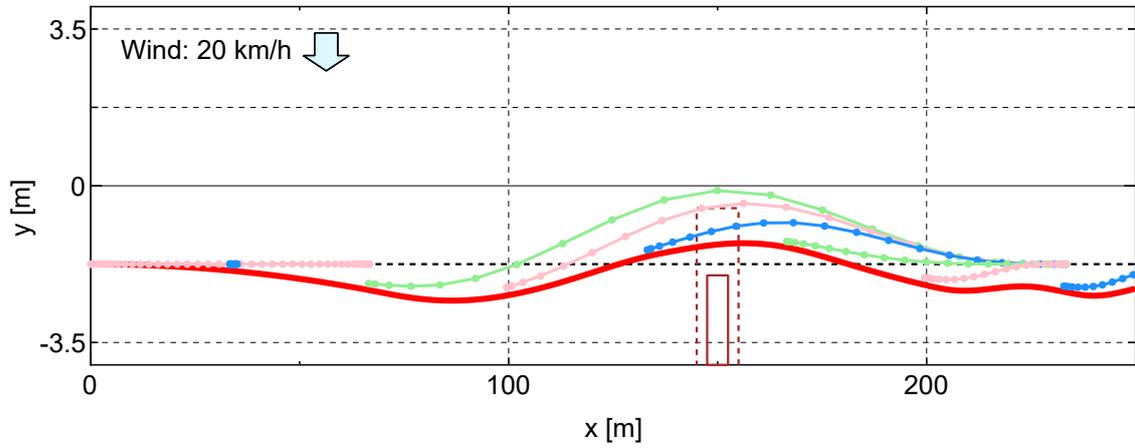
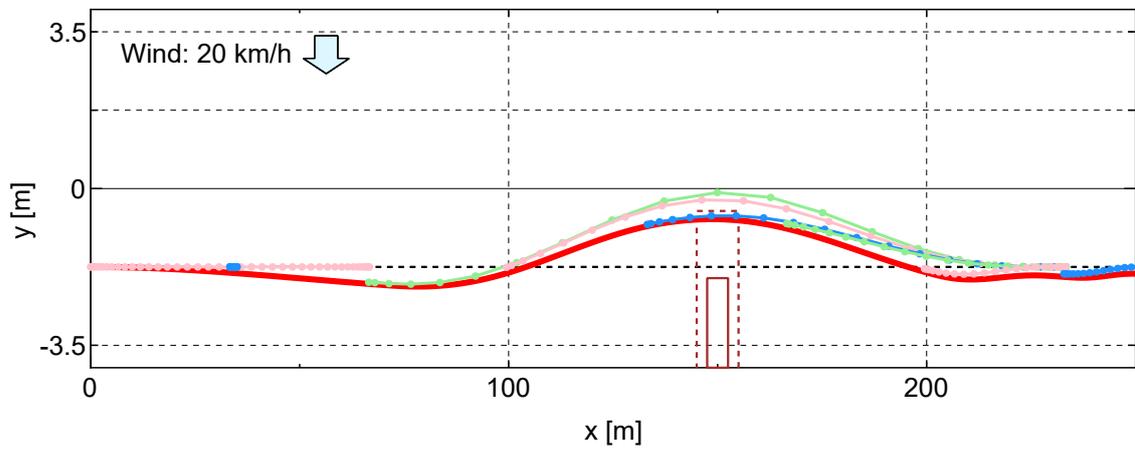


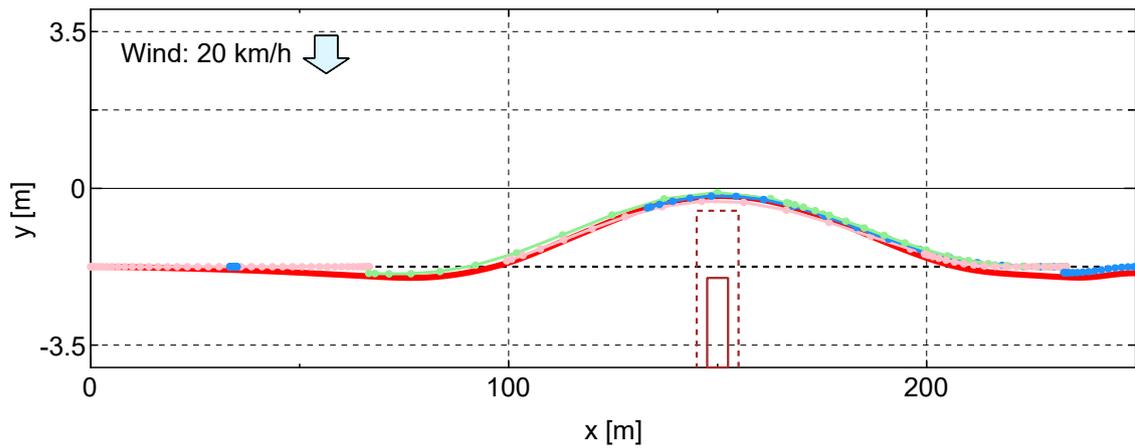
図 5.25 停止車両回避走行，横風あり，各手法の走行結果， $\tau_c = 2.0$



(a) PC- $\pi$  手法

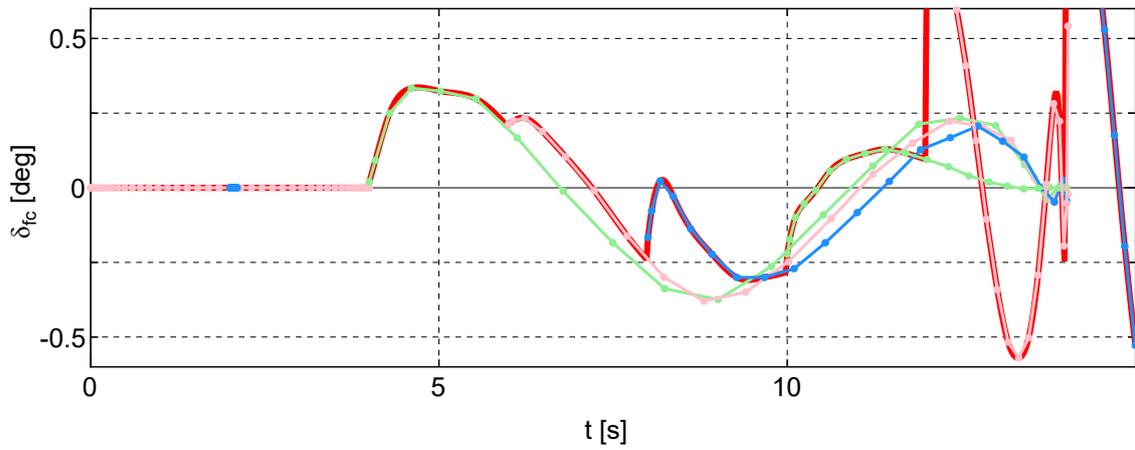


(b) Gain 型 M · PC- $\pi$  手法

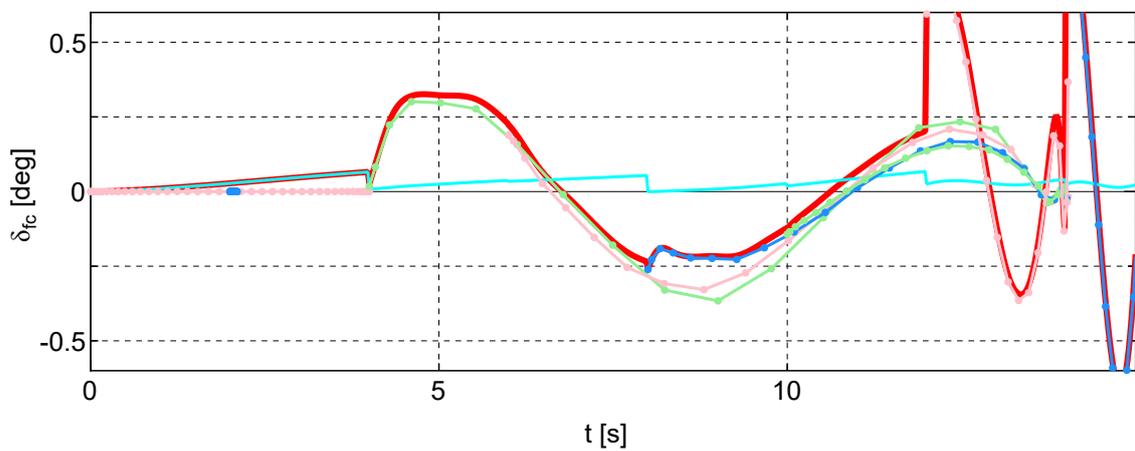


(c) MPC 型 M · PC- $\pi$  手法

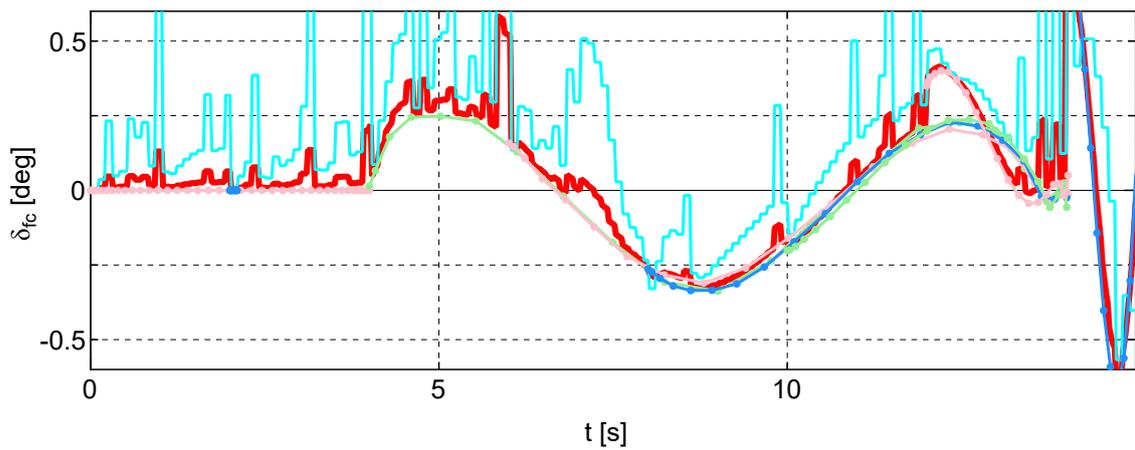
図 5.26 停止車両回避走行，横風あり，走行軌跡， $\tau_c = 2.0$



(a) PC- $\pi$  手法



(b) Gain 型 M · PC- $\pi$  手法



(c) MPC 型 M · PC- $\pi$  手法

図 5.27 停止車両回避走行，横風あり，操舵角指令の時間履歴， $\tau_c = 2.0$

図 5.28 に、各手法・各サンプリングタイムにおける停止車両真横通過時の走行車両の  $y$  座標をまとめる。□プロットが赤色の破線 ( $y = -0.5$  m) より低い値になると停止車両に衝突したと判定し、赤抜き□で示している。  $\tau_c = 2.0$  s の C- $\pi$  手法を用いた場合については、大きく負の方向へ移動している。ここでは表示のため、 $y$  軸のスケールを変更している。いずれの場合でもサンプリングタイムや風速の増大に伴って停止車両に接近する傾向が現れているが、追従制御器を組み込むことで回避性能は向上している。特に MPC 型では、全ての場合で回避することができている。

図 5.29 に停止車両回避走行におけるフルビークルモデルシミュレーションの様子を示す。横風 20 km/h,  $\tau_c = 2.0$  s において、MPC 型 M・PC- $\pi$  手法を用いた場合の走行結果を一例として取り上げている。停止車両を確認後、緩やかな旋回を行って回避し、元のレーンに復帰していることが、アニメーションでも確認できる。

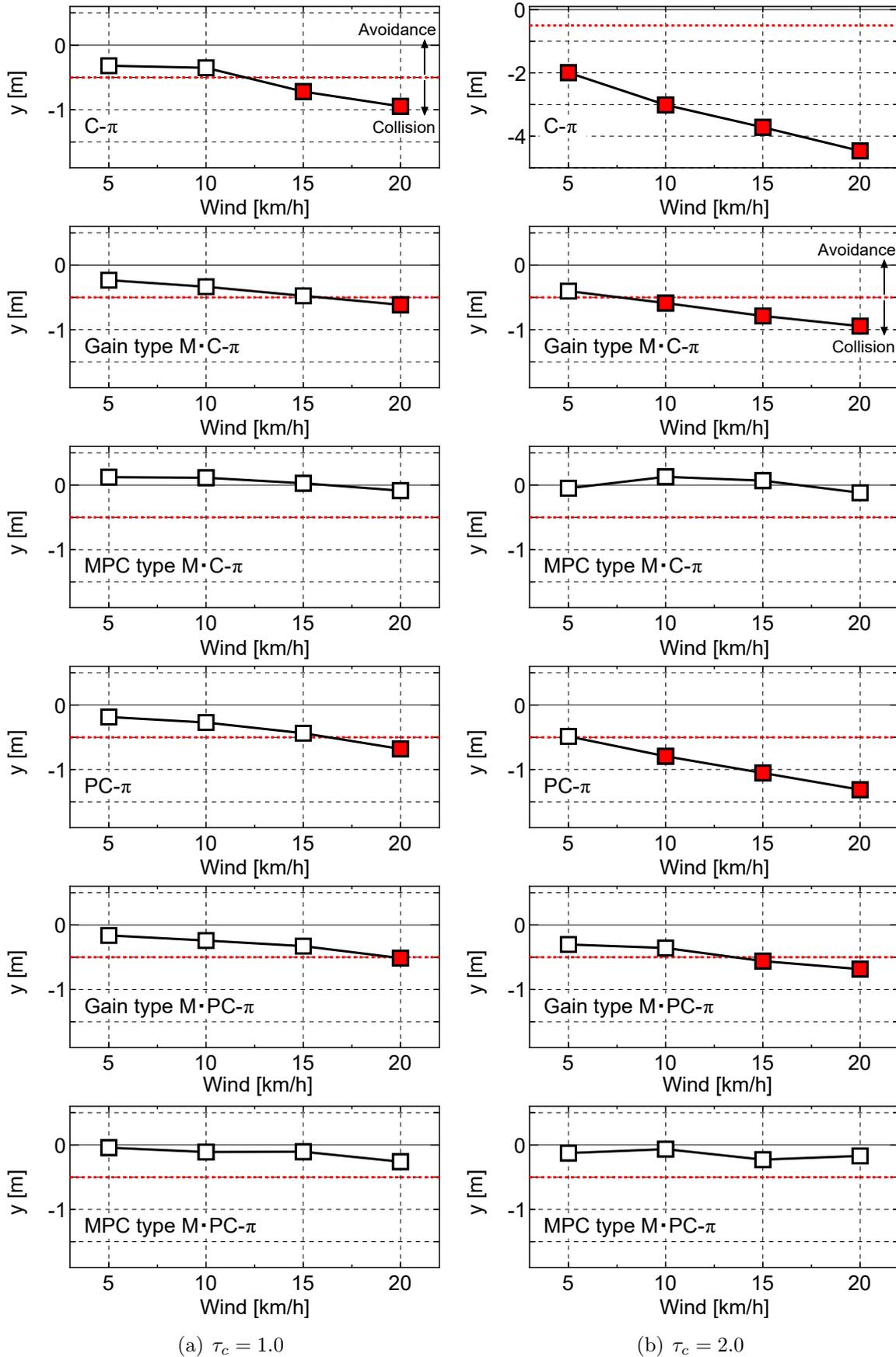


図 5.28 停止車両回避走行，横風あり，停止車両との接近距離

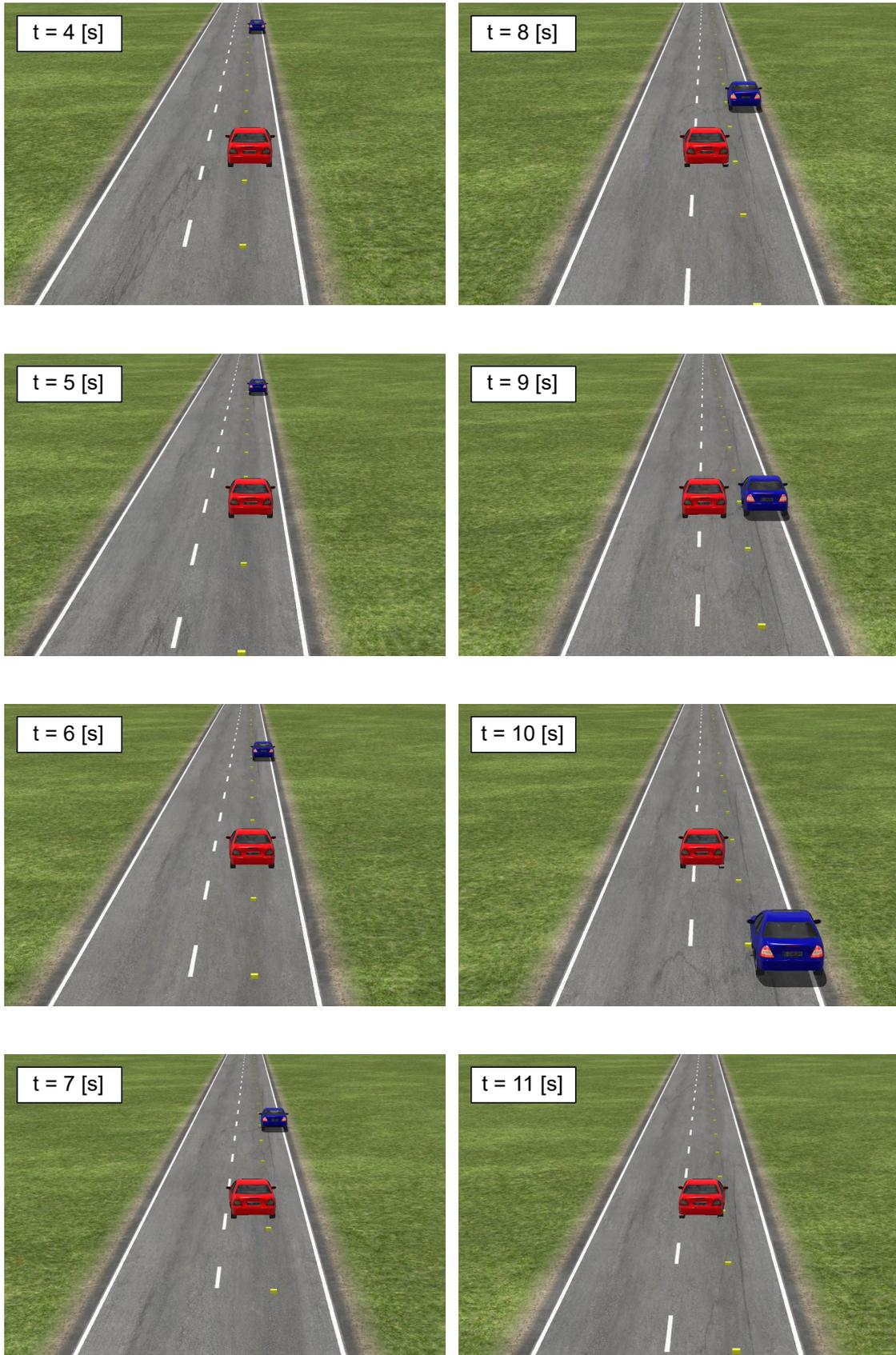


図 5.29 停止車両回避走行，横風あり，フルビークルモデルシミュレーションの様子

## 5.5 まとめ

本章では、第2章で記述した各手法をもとに一般車両の運動制御へ適用した制御系を設計し、フルビークルモデルのシミュレーションソフトウェアを用いた数値シミュレーションによる比較・検証を行い、以下の成果を得た。

### 5.5.1 Clock-Based 実時間最適制御の適用可能性

従来では、一般車両の運動制御を対象とした検証は行われていなかった。そこで、フルビークルモデルのシミュレーションソフトウェア上の一般車両に C- $\pi$  手法及び PC- $\pi$  手法による実時間最適制御を適用した走行を実現させ、各制御手法の適用可能性の確認を本研究における課題とした。

サンプリングタイムごとに解く最適制御問題において用いる運動モデルには計算負荷を軽減させるために簡易なモデルを使用する必要があるため、そこで得られた最適制御解を用いてフルビークルモデルにより運動計算を行う検証システムを構築した。本研究ではまず基本段階として、直線道路におけるレーンチェンジ走行と停止車両回避走行について数値シミュレーションを行った。構築した制御系により移動しながら最終の目標状態までの最適制御解が更新され目標を達成できることを確認した。

以上により、Clock-Based 実時間最適制御である C- $\pi$  手法と PC- $\pi$  手法の一般車両の運動制御に対する適用可能性について示すことができた。

### 5.5.2 予測情報を用いる PC- $\pi$ 手法の有用性

PC- $\pi$  手法は移動物体の運動制御に対して適用された例は報告されていなかったが、本研究では C- $\pi$  手法に対する PC- $\pi$  手法の有用性について注目し、本章では数値シミュレーションによる検証を試みた。

レーンチェンジ走行において C- $\pi$  手法を用いた場合では、走行の安定化をするため評価関数の設定により操舵角の急激な変化を抑制する必要があったが、PC- $\pi$  手法を用いた場合では評価関数の設定を変更しない場合でもレーンチェンジが可能であった。また停止車両回避走行においても、C- $\pi$  手法を用いた場合では停止車両に衝突する想定でも、PC- $\pi$  手法を用いた場合では回避できた。

以上により、一般車両の運動制御における C- $\pi$  手法に対する PC- $\pi$  手法の有用性について示すことができた。

### 5.5.3 追従制御器を付加した提案手法の有用性

計算機の能力や最適制御問題の複雑さなどにより最適化計算に要する時間は長くなるため、それに伴ってサンプリングタイムが長くなることは考えられ、そのような場合でもより安定性を保持する制御系の提案を本研究における課題とした。

C- $\pi$ 手法に対する有用性が示されたPC- $\pi$ 手法を用いた場合でも、停止車両回避走行の横風が存在する場合などでは、サンプリングタイムが長くなるにつれてその影響を大きく受けることが確認された。一方で、サンプリングタイムごとに生成される最適状態軌道に対する追従制御器を組み込んだ提案手法を用いることで、停止車両を回避する最適状態軌道との誤差を修正し、より長いサンプリングタイムでも回避性能が向上することを確認した。

以上により、一般車両の運動制御におけるC- $\pi$ 手法及びPC- $\pi$ 手法に対する提案手法の有用性について示すことができた。

# 第6章 結論

## 6.1 本研究の総括

本研究では Clock-Based 実時間最適制御の  $C-\pi$  手法と  $PC-\pi$  手法について、MGV と一般車両の運動制御に対する適用可能性を検証し、さらにサンプリングタイムが長くなる場合に追従制御器を付加することで安定性を向上させる新しい手法を提案した。MGV の運動制御に対しては、最初にそれぞれの手法を適用した制御システムを設計し、実機の MGV を用いた屋内走行実験環境を構築して数値シミュレーションと併せて実車両での検証を行った。一般車両の運動制御に対しては、最初にそれぞれの手法を適用した制御システムを設計し、フルビークルモデルのシミュレーションソフトウェアを用いた数値シミュレーション環境を構築して検証を行った。次の3点が実時間最適制御に関しての本研究による主な研究成果として挙げられる。

### Clock-Based 実時間最適制御の MGV に対する適用可能性の明示

Clock-Based 実時間最適制御を MGV の運動制御に適用した研究は、 $C-\pi$  手法についての数値シミュレーションによる検証のみであり、 $PC-\pi$  手法も含めて実機の MGV に対する適用可能性を示す必要があった。本研究では  $C-\pi$  手法や  $PC-\pi$  手法を適用した制御システムを設計し、実機の MGV を用いた走行を実現した。直線走行、S 字走行、繰り返し走行において基本的な移動性能を確認し、障害物回避走行では走行途中に発生する障害物を新たな最適制御解を生成することで回避して目標まで移動することができた。これらより、Clock-Based 実時間最適制御の MGV に対する適用可能性を示した。

### Clock-Based 実時間最適制御の一般車両に対する適用可能性の明示

Clock-Based 実時間最適制御を一般車両の運動制御に適用した研究はなく、 $C-\pi$  手法や  $PC-\pi$  手法の一般車両に対する適用可能性を示す必要があった。本研究では  $C-\pi$  手法や  $PC-\pi$  手法を適用した制御システムを設計し、フルビークルモデルのシミュレーションソフトウェアを用いた数値シミュレーションによる検証を行った。片側2車線

を一定速度で移動する想定において、レーンチェンジ走行に対する基本的な性能を確認し、停止車両回避走行では走行途中に確認した前方の停止車両を新たな最適制御解を生成することで回避することができた。これらより、Clock-Based 実時間最適制御の一般車両に対する適用可能性を示した。

### 提案手法による安定性の向上

従来ではシステムの特性に依存する安定性が保証されたサンプリング間隔内に最適化計算が終了することを前提とした検証しか行われておらず、サンプリングタイムが長くなる場合でもより安定して移動できる必要があった。本研究では、C- $\pi$  手法やPC- $\pi$  手法の制御系に Gain 型や MPC 型の最適状態軌道に追従する機構を組み込む手法を提案し、各制御対象の検証を行った。多くの場合でPC- $\pi$  手法と提案手法が有効であったが、MGV では路面外乱の影響がある場合、一般車両では横風の影響がある場合などではPC- $\pi$  手法と比較しても提案手法が有効であった。サンプリングタイムが長くなる場合でも安定性が向上することを示した。

## 6.2 本研究で得られた今後の展望

本研究に対して、以下の展望が挙げられる。

### Gain 型や MPC 型の追従制御器の適切なパラメータの決定

本研究では、それぞれの制御対象において想定が変わる場合でも追従制御器では同一のパラメータ値を使用している。今回想定したほとんどの状況では Gain 型や MPC 型の追従制御器を組み込みことで安定性が向上したが、一般車両のレーンチェンジ走行において速度変更した場合には不安定な挙動が確認された。そのため、今後はより多様な走行状況を想定した検証を行い、走行状況に応じてパラメータをスケジューリングするアルゴリズムの追加や、より多くの場面で安定できるパラメータ値の選定などを行っていく必要がある。

### MGV の低精度状態取得下における検証

本研究の MGV の走行実験ではモーション・キャプチャシステムを用いて高精度な状態推定を行った。しかし実際の運用では GPS やレーザーレンジファインダなど本システムより状態推定の精度は劣化する。そのため、それぞれのセンサを用いた状態取

得を行う実験システムを構築し，低精度状態取得下における Clock-Based 実時間最適制御の適用可能性や提案手法の効果について検証していく必要がある．

#### 一般車両のコーナリング走行中の制御

本研究の一般車両の運動制御では，基本的な想定として直線道路での走行を考えた．しかし曲線路の走行では，旋回に伴うロール運動の発生など本研究で使用した等価二輪モデルとは運動特性が異なってくる．そのため，曲線路走行に対する本研究で構築したシステムのロバスト性や，別のモデルを用いた制御系設計の必要性などについて検証していく必要がある．

# 謝 辞

本論文は著者が防衛大学校理工学研究科後期課程に在籍中の研究成果をまとめたものである。

本研究を行うにあたり、全般にわたって終始懇切、丁寧なご指導、ご鞭撻を賜りました原田正範教授に心より感謝致します。本科4学年の卒業研究よりお世話になり、卒業後3年の部隊勤務を経て研究科前期課程学生として研究室に戻ってくる機会を与えていただきました。この5年間では、講座内に先輩にあたる研究科学生がいなかったこともあり、勉学や研究の進め方、学会発表、論文投稿の研究活動など数多くの場面で原田教授より直接ご指導をいただくことができ、非常に有意義かつ恵まれた環境で過ごすことができました。そして本博士論文執筆にあたっては、執筆に関する全てにおいて幾度となくご指導していただき、完成に至ることができました。深く御礼申し上げます。

審査教官の山川教授、藤原准教授、横浜国立大学の上野教授には、予備審査より本論文の審査にあたり御指摘や御助言をいただきました。研究の位置付けや研究成果のまとめ方などについて修正を行い、本論文をより良くすることができました。深く感謝致します。

防衛大学校システム工学群機械工学科及び機械システム工学科の教職員の皆様には、学業のみならず、様々な行事にも参加させていただき、大変お世話になりました。誠にありがとうございました。

著者の所属機関である防衛省陸上自衛隊には、本研究を行う機会を与えていただきました。日頃より多大なる御支援をいただき、防衛大学校理工学研究科の後期課程教育を専念して受けることができました。厚く御礼申し上げます。

同研究室の市川竜太1等陸尉（理工学研究科前期課程第53期学生）、大久保重李沙1等陸尉（理工学研究科前期課程第55期学生）、稲垣汰一学生（本科第62期学生）、大坪秀斗学生（本科第62期学生）、中前拓也学生（本科第62期学生）には、研究実験や論文作成に協力していただきました。大変ありがとうございました。

最後に、研究を継続する上で心の支えとなってくれた妻 悠季と、長男 誠士郎に感謝の意を表します。家に帰らない日々が続いても、文句も言わずにいつも応援してくれて支えてもらいました。本当にありがとうございました。

## 参考文献

- [1] 日本ロボット学会, 原子力ロボット記録と提言, 東日本大震災関連調査研究委員会/原子力関係記録作成分科会最終報告書, 2014.
- [2] 福島第一原発 無人ヘリ T-Hawk 空撮, Response, 2011,  
<https://response.jp/article/2011/04/12/154697.html>, 最終アクセス日:2017/8/24.
- [3] 内閣官房 IT 総合戦略室, 自動運転レベルの定義を巡る動きと今後の対応(案), 第1回道路交通ワーキングチーム・第26回SIP自動走行システム推進員会合同会議配布資料, 2016.
- [4] 内閣府政策統括官(科学技術・イノベーション担当), 戦略的イノベーション創造プログラム(SIP)自動走行システム研究開発計画, 2017.
- [5] 内閣官房 IT 総合戦略室, ITS・自動運転を巡る最近の動向(国内の動向を中心に), 第2回道路交通ワーキングチーム配布資料, 2017.
- [6] 内閣官房 IT 総合戦略室, 完全自動運転実現へのシナリオと制度的課題(案), 第2回道路交通ワーキングチーム配布資料, 2017.
- [7] Siegwart, R. and Nourbakhsh, I. R., Introduction to Autonomous Mobile Robots, The MIT Press, 2004.
- [8] 上原祥雄, 最近の戦闘機の運動性における最適制御理論の応用, 計測と制御, Vol. 20, No. 10, pp. 14-20, 1981.
- [9] 加藤寛一郎, 飛行力学と最適制御, 日本航空宇宙学会誌, Vol. 31, No. 359, pp. 667-671, 1983.
- [10] 加藤寛一郎, 中島陸博, 風がある場合の最短時間経路, 日本航空宇宙学会誌, Vol. 32, No. 360, pp. 66-71, 1984.
- [11] 加藤寛一郎, 最適制御からみた航空機, 計測と制御, Vol. 24, No. 5, pp. 15-21, 1985.

- [12] 藤岡健彦, 石川毅彦, 最適制御理論の車両運動力学への応用 -四輪操舵による車両運動性能向上の可能性-, 自動車技術会論文集, No. 35, pp. 112-119, 1987.
- [13] 割石義典, 加藤寛一郎, 極超音速機の最適飛行経路に関する一覚書, 日本航空宇宙学会誌, Vol. 36, No. 416, pp. 427-432, 1988.
- [14] 大林和弘, 加藤寛一郎, 最短時間旋回に関する一覚書, 日本航空宇宙学会誌, Vol. 36, No. 416, pp. 433-439, 1988.
- [15] 上野誠也, 周期最適化による最小燃料巡航, 日本航空宇宙学会誌, Vol. 37, No. 429, pp. 502-506, 1989.
- [16] 高野博行, 空力加熱の制限を含む AOTV の最適帰還に関する研究, 日本航空宇宙学会誌, Vol. 39, No. 453, pp. 522-530, 1991.
- [17] 石本真二, AOTV の大気圏飛行における最適軌道について, 日本航空宇宙学会誌, Vol. 40, No. 463, pp. 437-442, 1992.
- [18] 杉本啓, 加藤寛一郎, 航空機の進入経路に関する一考察, 日本航空宇宙学会誌, Vol. 40, No. 467, pp. 674-681, 1992.
- [19] 藤岡健彦, 木村健, 最短時間コーナリング法に関する理論的研究 -第1報 4WD・4WS-, 自動車技術会論文集, Vol. 23, No. 2, pp. 75-80, 1992.
- [20] 猫橋敏文, 佐々木茂, 長島知有, ヘリコプタの最短時間旋回, 日本航空宇宙学会誌, Vol. 41, No. 472, pp. 293-299, 1993.
- [21] 杉本啓, 二機の航空機の進入経路に関する一考察, 日本航空宇宙学会誌, Vol. 41, No. 476, pp. 537-542, 1993.
- [22] 藤岡健彦, 木村健, 最短時間コーナリング法に関する理論的研究 -第3報 荷重移動の影響-, 自動車技術会論文集, Vol. 24, No. 2, pp. 88-93, 1993.
- [23] 藤岡健彦, 木村健, 最短時間コーナリング法に関する理論的研究 -第2報 操舵・駆動方式が異なる車両におけるコーナリング運動の比較-, 自動車技術会論文集, Vol. 24, No. 3, pp. 94-99, 1993.
- [24] 藤岡健彦, 江守大昌, 最短時間コーナリング法に関する理論的研究 -第4報 状態量不等式拘束を用いた道路条件の導入-, 自動車技術会論文集, Vol. 24, No. 3, pp. 106-111, 1993.

- [25] Fujioka, T. and Kato, M., Numerical Analysis of Minimum-Time Cornering, AVEC'94, No. 9438367, pp. 260-265, 1994.
- [26] Bryson, A. E., Optimal Control 1950 to 1985, IEEE Control Systems Magazine, pp. 26-33, 1996.
- [27] 鈴木真二, 最適制御は航空機を救えるか -マイクロバースト下での航空機最適制御-, 日本航空宇宙学会誌, Vol. 44, No. 504, pp. 21-27, 1996.
- [28] 原田正範, 別府護郎, 東昭, 滑空機の距離最大飛行に関する研究, 日本航空宇宙学会誌, Vol. 44, No. 513, pp. 582-645, 1996.
- [29] 高江康彦, 石川芳男, 山口雄仁, 遺伝的アルゴリズムによる最適飛行経路の生成 -スペースプレーンの最大滑空距離問題-, 日本航空宇宙学会誌, Vol. 44, No. 514, pp. 637-645, 1996.
- [30] 井前讓, 平山信一郎, 古館貴, 非ホロミックシステムの軌道計画問題 (最適制御アルゴリズムからの接近), 日本機械学会論文集 (C編), Vol. 62, No. 600, pp. 228-234, 1996.
- [31] 藤岡健彦, 横山篤, 今泉博英, 二輪車の最短時間コーナリング法に関する理論的研究, 自動車技術会論文集, Vol. 28, No. 3, pp. 125-129, 1997.
- [32] 土屋武司, 鈴木真二, スペースプレーンの機体設計と飛行経路の同時最適化に関する数値解法, 日本航空宇宙学会誌, Vol. 46, No. 533, pp. 346-353, 1998.
- [33] 原田正範, 別府護郎, STOL機の着陸進入に関する研究, 日本航空宇宙学会誌, Vol. 46, No. 537, pp. 558-564, 1998.
- [34] Casanova, D., Sharp, R. S., and Symonds, P., Minimum Time Manoeuvring: The Significance of Yaw Inertia, Vehicle System Dynamics, Vol. 34, pp. 77-115, 2000.
- [35] 原田正範, 競技車両の最適周回走行の解析, 自動車技術会論文集, Vol. 39, No. 6, pp. 17-22, 2008.
- [36] 藤岡健彦, 柴田温史, 塚椅祐一郎, 津田慎司, 最適制御計算による衝突危険値を最小にする車両運動制御, 自動車技術会 2008 年春季大会学術講演会前刷集, No. 8-08, pp. 21-26, 2008.
- [37] Gong, Q., Lewis, L. R., and Ross, I. M., Pseudospectral Motion Planning for Autonomous Vehicles, Journal of Guidance, Control, and Dynamics, Vol. 32, No. 3, pp. 1039-1045, 2009.

- [38] Hurni, M. A., Sekhavat, P., Karpenko, M., and Ross, I. M., A Pseudospectral Optimal Motion Planner for Autonomous Unmanned Vehicles, Proceedings of American Control Conference, No. WeC01.1, pp. 1591-1598, 2010.
- [39] 原田正範, 最短時間コーナリングの解析における車両モデルの影響, 自動車技術会論文集, Vol. 41, No. 3, pp. 581-586, 2010.
- [40] 伊藤正貴, ロータ型無人機の最適制御に関する研究, 平成 23 年度防衛大学校前期課程卒業論文, 2011.
- [41] 原田正範, 二輪車の最短時間走行経路, 自動車技術会論文集, Vol. 42, No. 2, pp. 301-306, 2011.
- [42] 原田正範, 擬スペクトル最適化手法を用いた二輪車の衝突回避運動の解析, 自動車技術会 2013 年春季大会学術講演会前刷集, No. 16-13, pp. 1-6, 2013.
- [43] 山村信夫, 車両の最適軌道制御に関する研究, 平成 24 年度防衛大学校本科卒業論文, 2013.
- [44] Bryson, A. E. and Denham, W. F., A Steepest-Ascent Method for Solving Optimum Programming Problems, ASME Journal of Applied Mechanics, Vol. 29, No. 2, pp. 247-257, 1962.
- [45] 原辰次, ロバスト制御理論の回顧と展望, 計測と制御, Vol. 40, No. 1, pp. 63-69, 2001.
- [46] 大嶋正裕, モデル予測制御 -理論の誕生・展開・発展-, 計測と制御, Vol. 39, No. 5, pp. 321-325, 2000.
- [47] 大嶋正裕, 小河守正, 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.
- [48] 加納学, 大嶋正裕, 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.
- [49] 増田士朗, 山本透, 大嶋正裕, 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.
- [50] 大嶋正裕, 加納学, 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.

- [51] 大嶋正裕, 関宏也, 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.
- [52] 藤田政之, 大嶋正裕, 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.
- [53] 児島晃, 大塚敏之, モデル予測制御の考え方, 計測と制御, Vol. 42, No. 4, pp. 310-312, 2003.
- [54] Maciejowski, J. M. (著), 足立修一, 菅野政明 (訳), 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.
- [55] 足立修一, モデル予測制御の基礎, 日本ロボット学会誌, Vol. 32, No. 6, pp. 499-502, 2014.
- [56] Kim, H. J., Shim, D. H., and Sastry, S., Nonlinear Model Predictive Tracking Control for Rotorcraft-based Unmanned Aerial Vehicles, Proceedings of the American Control Conference, pp. 3576-3581, 2002.
- [57] 向井正和, 川邊武俊, 西羅光, 高木良貴, 出口欣高, モデル予測制御による自動車の障害物回避へのアプローチ, 第49回自動制御連合講演会, No. os05, 2006.
- [58] 根和幸, 福島宏明, 松野文俊, 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. 42, No. 8, pp. 877-883, 2006.
- [59] 向井正和, 川邊武俊, 混合整数計画問題を用いたモデル予測制御, 第51回自動制御連合講演会, No. 531, pp. 900-903, 2008.
- [60] 朴達, 大塚敏之, C/GMRES アルゴリズムによる非線形モデル予測制御を用いた四輪車両の衝突回避, 第51回自動制御連合講演会, No. 535, pp. 916-919, 2008.
- [61] 中澤大輔, 鈴木智, 廣谷和之, 野波健蔵, 規範モデル追従型モデル予測制御を用いた小型無人ヘリコプタの編隊飛行制御, 第51回自動制御連合講演会, No. 706, pp. 385-390, 2008.
- [62] Yoshida, H., Shinohara, S., and Nagai, M., Lane change steering manoeuvre using model predictive control theory, Vehicle System Dynamics, Vol. 46, pp. 669-681, 2002.
- [63] 中澤大輔, 鈴木智, 野波健蔵, 規範モデル追従型モデル予測制御による小型無人ヘリコプタの軌道追従制御, 日本機械学会論文集 (C編), Vol. 74, No. 746, pp. 170-177, 2008.

- [64] Park, J. M., Kim, D. W., Yoon, Y. S., Kim, H. J., and Yi, K. S., Obstacle avoidance of autonomous vehicles based on model predictive control, Proceedings of the Institution of Mechanical Engineers, Vol. 223, No. 12, pp. 1499-1516, 2009.
- [65] 根和幸, 福島宏明, 松野文俊, 予測時刻間の障害物回避を考慮したモデル予測制御に基づく軌道計画法, 計測自動制御学会論文集, Vol. 45, No. 8, pp. 406-413, 2009.
- [66] Gao, Y., Lin, T., Borrelli, F., Tseng, T., and Hrovat, D., Predictive Control of Autonomous Ground Vehicles with Obstacle Avoidance on Slippery Roads, Proceedings of ASME 2010 Dynamic Systems and Control Conference, No. DSCC2010-4263, pp. 265-272, 2010.
- [67] 大川功, 野中謙一郎, 経路の道幅を考慮した脚車輪型ロボット車輪位置のモデル予測制御, 計測自動制御学会論文集, Vol. 10, No. 23, pp. 190-197, 2011.
- [68] 伊藤洸, 中山弘隆, 多目的モデル予測制御によるロボットの移動経路決定, 第54回自動制御連合講演会, No. 2M301, pp. 1781-1782, 2011.
- [69] 丸茂喜高, 野見山龍介, 二輪車の衝突回避システムに関する研究 (モデル予測制御を用いた自動操舵制御), 日本機械学会論文集 (C編), Vol. 77, No. 781, pp. 97-108, 2011.
- [70] 土屋武司, 航空機の突風回避飛行に対する実時間最適制御, 計測自動制御学会論文集, Vol. 50, No. 1, pp. 68-74, 2014.
- [71] 小山健太郎, 野中謙一郎, 障害物回避と切り返し点の自動調整によるモデル予測車庫入れ制御, 計測自動制御学会論文集, Vol. 50, No. 1, pp. 9-17, 2014.
- [72] 濱田吉郎, 塚本太郎, 石本真二, モデル予測制御による小規模飛行実験機の水平面内誘導-直線軌道追従ケースでの飛行試験結果-, 計測自動制御学会論文集, Vol. 50, No. 3, pp. 236-244, 2014.
- [73] 氷野康平, 橋本智昭, 大塚敏之, タイヤ力の飽和と荷重移動を伴う四輪車両の非線形モデル予測制御, 計測自動制御学会論文集, Vol. 50, No. 5, pp. 432-440, 2014.
- [74] 小山健太郎, 関口和真, 野中謙一郎, 走行距離を時間軸とする時間軸状態制御形による車両の経路追従制御, 計測自動制御学会論文集, Vol. 50, No. 10, pp. 764-754, 2014.

- [75] Liu, J., Jeffrey, P., Stein, L., and Ersal, T., A Nonlinear Model Predictive Control Algorithm for Obstacle Avoidance in Autonomous Ground Vehicles within Unknown Environments, Defense Technical Information Center, No. ADA615460, 2015.
- [76] Franze, G. and Lucia, W., An Obstacle Avoidance Model Predictive Control Scheme for Mobile Robots Subject to Nonholonomic Constraints: A Sum-of-squares Approach, *Journal of the Franklin Institute*, Vol. 352, No. 6, pp. 2358-2380, 2015.
- [77] 小田貴嗣, 野中謙一郎, 関口和真, モデル予測制御とスライディングモード制御による四輪操舵駆動車両のロバスト経路追従制御, *計測自動制御学会論文集*, Vol. 51, No. 7, pp. 484-493, 2015.
- [78] 向井正和, 川邊武俊, 省燃費のための自動車の速度制御, *計測と制御*, Vol. 54, No. 11, pp. 824-827, 2015.
- [79] 豊口将之, 松實良祐, 林隆三, 障害物自動回避システムにおけるモデル予測制御を用いた軌道追従制御, *日本機械学会 Dynamics and Design Conference 2015 USB 論文集*, No. 15-7, 2015.
- [80] 萩森夕紀, 高橋直樹, 野中謙一郎, 操舵角度の可動範囲と連続性を陽に考慮した車輪型移動ロボットのモデル予測軌道追従制御, *日本機械学会論文集*, Vol. 81, No. 825, p. 14-00484, 2015.
- [81] Ross, I. M., Sekahavat, P., Felming, A., and Gong, Q., Optimal Feedback Control: Foundations, Examples, and Experimental Results for a New Approach, *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 2, pp. 307-321, 2008.
- [82] Ross, I. M., Gong, Q., Felming, A., and Kang, W., Practical Stabilization Through Real-Time Optimal Control, *Proceedings of the 2006 American Control Conference*, WeA09.3, pp. 304-309, 2006.
- [83] Ross, I. M., Sekahavat, P., Felming, A., and Gong, Q., Pseudospectral Feedback Control: Foundations, Examples, and Experimental Results for a New Approach, *AIAA Guidance, Navigation, and Control Conference*, AIAA 2006-6354, pp. 1-23, 2006.
- [84] Kevin, P. B., High-Fidelity Real-Time Trajectory Optimization for Reusable Launch Vehicles, Ph. D. thesis, Naval Postgraduate School, 2006.

- [85] Hurni, M. A., Sekhavat, P., and Ross, I. M., Autonomous Trajectory Planning Using Real-Time Information Updates, AIAA Guidance, Navigation, and Control Conference, AIAA 2008-6305, pp. 1-19, 2008.
- [86] Hurni, M. A., An Information-Centric Approach to Autonomous Trajectory Planning Utilizing Optimal Control Techniques, Ph. D. thesis, Naval Postgraduate School, 2009.
- [87] Karpenko, M., Sekhavat, P., Park, J., and Ross, I. M., Closed-Loop Optimal Guidance and Control of Quadrotor Unmanned Aerial Vehicle, AIAA Guidance, Navigation, and Control Conference, AIAA 2010-8051, pp. 1-11, 2010.
- [88] 市川竜太,  $L_1$  適応制御および実時間最適制御を用いた同軸反転ロータ機の誘導制御に関する研究, 防衛大学校理工学研究科前期課程第 53 期修士論文, 2016.
- [89] 渡辺宗一郎, 原田正範, モデル予測制御を用いた Clock-Based 実時間最適制御の安定性向上, 日本航空宇宙学会第 48 期年会講演会講演論文集, JSASS-2017-1080, 2017.
- [90] 原田正範, 渡辺宗一郎, 下降気流の影響下における着陸進入に実時間最適制御, 日本航空宇宙学会第 55 回飛行機シンポジウム (発表予定).
- [91] Clarke, F. H., Ledyaev, Y. S., Stern, R. J., and Wolenski, P. R., Nonsmooth Analysis and Control Theory, Springer-Verlag, New York, 1998.
- [92] Vinter, R. B., Optimal Control, Birkhäuser, Boston, 2000.
- [93] Brockett, R., Asymptotic Stability and Feedback Stabilization, Differential Geometric Control Theory, edited by R. Brockett, R. Millman, and H. Sussmann, Birkhäuser, Boston, pp. 181-191, 1983.
- [94] Clarke, F. H., Ledyaev, Y. S., Sontag, E. D., and Subbotin, A. I., Asymptotic Controllability Implies Feedback Stabilization, IEEE Transactions on Automatic Control, Vol. 42, No. 10, pp. 1394-1406, 1997.
- [95] Bryson, A. E. and Ho, Y. C., Applied Optimal Control, Taylor&Francis, 1975.
- [96] 上野誠也, フィードバック誘導制御則による航空機の最短時間旋回, 日本航空宇宙学会誌, Vol. 45, No. 516, pp. 29-35, 1997.

- [97] Ghosh, P., Conway, B. A., Near-Optimal Feedback Strategies Synthesized Using a Spatial Statistical Approach, *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, pp. 905-919, 2013.
- [98] Harada, M. and Watanabe, S., Stability Analysis of Modified Clock-Based Real-Time Optimal Control, APISAT2016, APISAT-2016-R7-2, 2016.
- [99] Dahmani, H., Chadli, M., Rabhi, A., and Hajjaji, A. El., Driver attention warning system based on a fuzzy representation of the vehicle model, IFAC The International Federation of Automatic Control, Proceedings of the 18th World Congress, pp. 6260-6265, 2011.
- [100] Wang, Y., Bevly, D. M., and Rajamani, R., Interval observer design for LPV systems with parametric uncertainty, *Automatica*, Vol. 60, pp. 79-85, 2015.
- [101] Alipour, H., Sabahi, M., Sharifian, M. B. B., Lateral stabilization of a four wheel independent drive electric vehicle on slippery roads, *Mechatronics*, Vol. 30, pp. 275-285, 2015.
- [102] 浜口雅史, 武野真典, 谷口隆雄, 自動走行車両が緊急事態を回避するための制御系 (実数型免疫アルゴリズムを用いた操縦波形生成法), *日本機械学会論文集*, Vol. 81, No. 829, pp. 1-12, 2015.
- [103] 渡辺宗一郎, 小型無人車両の実時間最適誘導制御に関する研究, 防衛大学校理工学研究科前期課程第 52 期修士論文, 2015.
- [104] Dubins, L. E., On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents, *American Journal of Mathematics*, Vol. 79, pp. 497-516, 1957.
- [105] Reeds, J. A. and Shepp, L. A., Optimal Paths for a Car that Goes Both Forwards and Backwards, *Pacific Journal of Mathematics*, Vol. 145, No. 2, pp. 367-393, 1990.
- [106] 原田正範, 高次ガウス・ロボット則の重み関数を用いたヤコビ擬スペクトル法による軌道最適化, *日本機械学会論文集 (C 編)*, Vol. 73, No. 728, pp. 1075-1080, 2007.
- [107] 近藤政市, 自動車の操舵と運動間に存在する基礎的關係について, *自動車技術会論文集*, No. 5, pp. 40-43, 1958.

- [108] 吉本堅一, 予測を含む操舵モデルによる人間-自動車系のシミュレーション, 日本機械学会誌, Vol. 71, No. 596, pp. 1181-1186, 1968.
- [109] 安部正人, 自動車の運動と制御 (第二版), 山海堂, ISBN 4-381-08822, 1992.

# 著者学会等発表実績

## 学術論文

1. Watanabe, S. and Harada, M.,  
Experimental Study on Optimal Tracking Control of a Micro Ground Vehicle,  
Journal of Robotics and Mechatronics, Vol. 29, No. 4, pp. 757-765, 2017.8.
2. 渡辺宗一郎, 原田正範,  
自動運転のための予測型 Clock-Based 実時間最適制御,  
自動車技術会論文集, Vol. 48, No. 4, pp. 953-958, 2017.7.
3. 渡辺宗一郎, 原田正範,  
小型無人車両の繰り返し操作を含む実時間最適誘導制御,  
自動車技術会論文集, Vol. 47, No. 4, pp. 985-990, 2016.7.
4. Watanabe, S. and Harada, M.,  
Optimal Tracking Control of an Micro Ground Vehicle,  
Journal of Robotics and Mechatronics, Vol. 27, No. 6, pp. 653-659, 2015.12.
5. 渡辺宗一郎, 原田正範,  
障害物配置法を用いた最適走行の解析,  
日本機械学会論文集, Vol. 80, No. 817, pp. 1-12, 2014.9.

## 国際会議

1. Watanabe, S. and Harada, M.,  
Clock-Based Real-Time Optimal Control of a Skid-Steered Vehicle,  
2017 Fourth Asian Conference on Defence Technology (4th ACDT),  
2017.11.29.
2. Harada, M. and Watanabe, S.,  
Stability Analysis of Modified Clock-Based Real-Time Optimal Control,  
The 2016 Asia-Pacific International Symposium on Aerospace Technology  
(APISAT2016), 2016.10.26.
3. Watanabe, S., Okubo, A., and Harada, M.,  
Stability Augmentation of an SCR MAV with Slung Load,  
The 2016 Asia-Pacific International Symposium on Aerospace Technology  
(APISAT2016), 2016.10.25.
4. Watanabe, S. and Harada, M.,  
Optimal Trajectory Tracking Control of an SCR MAV Using Model Predictive  
Control,  
The 2016 Asia-Pacific International Symposium on Aerospace Technology  
(APISAT2016), 2016.10.25.
5. Watanabe, S. and Harada, M.,  
Real-Time Optimal Obstacle Avoidance of MGW Using Modified Carathéodory-  
 $\pi$  Solutions,  
13th International Symposium on Advanced Vehicle Control (AVEC'16),  
2016.9.15.
6. Harada, M., Watanabe, S., Ichikawa, R., and Bollino, K.,  
Stability Augmentation for Rotor MAV Takeoff and Landing using a Meshed  
Platform,  
The AIAA Science and Technology Forum and Exposition 2016  
(AIAA SCITECH2016), 2016.1.7.

7. Harada, M., Ichikawa, R., Watanabe, S., and Bollino, K.,  
 $\mathcal{L}_1$  Adaptive Control for a Single Coaxial Rotor MAV,  
The AIAA Science and Technology Forum and Exposition 2016  
(AIAA SCITECH2016), 2016.1.5.
8. Okubo, A., Watanabe, S., and Harada, M.,  
Three-Dimensional Trajectory Optimization Considering Arrival Time at the  
Merging Point,  
The 4th ENRI International Workshop on ATM/CNS (EIWAC2015), 2015.11.18.
9. Watanabe, S. and Harada, M.,  
Real-Time Optimal Feedback Control of UGVs Using Modified Carathéodory- $\pi$   
Solutions,  
12th International Symposium on Advanced Vehicle Control (AVEC'14),  
2014.9.23.
10. Harada, M. and Watanabe, S.,  
Optimal Landing Approach Trajectory in Two Microburst Cells,  
The 2013 Asia-Pacific International Symposium on Aerospace Technology  
(APISAT2013), 2013.11.21.
11. Watanabe, S. and Harada, M.,  
Stability Augmentation Landing Approach of a Rotor MAV by a Meshed Plat-  
form,  
The 2013 Asia-Pacific International Symposium on Aerospace Technology  
(APISAT2013), 2013.11.20.

## 学会発表等

1. 大久保亜李沙, 渡辺宗一郎, 原田正範,  
飛行経路および高度拘束下における最適進入軌道,  
日本航空宇宙学会第 55 回飛行機シンポジウム, 2017.11.22.
2. 原田正範, 渡辺宗一郎,  
下降気流の影響下における着陸進入の実時間最適制御,  
日本航空宇宙学会第 55 回飛行機シンポジウム, 2017.11.21.
3. 原田正範, 渡辺宗一郎,  
フリップ機動によるクワッドロータ機の最適周辺監視,  
日本航空宇宙学会第 48 期年会講演会, 2017.4.14.
4. 渡辺宗一郎, 原田正範,  
モデル予測制御を用いた Clock-Based 実時間最適制御の安定性向上,  
日本航空宇宙学会第 48 期年会講演会, 2017.4.14.
5. 渡辺宗一郎, 原田正範,  
MGV-MAV 協調運用のための光学式移動離着陸システムの紹介,  
協調ビークル分科会, 2017.3.27.
6. 大久保亜李沙, 市川竜太, 渡辺宗一郎, 原田正範,  
SCR 機の実時間最適制御による実飛行,  
日本航空宇宙学会第 47 期年会講演会, 2016.4.14.
7. 渡辺宗一郎, 市川竜太, 大久保亜李沙, 原田正範,  
床面及び障害物付近の擾乱が SCR 機に与える影響,  
日本航空宇宙学会第 47 期年会講演会, 2016.4.14.
8. 渡辺宗一郎, 市川竜太, 大久保亜李沙, 原田正範,  
MPC を用いた最適軌道に対する MGV の追従制御,  
協調ビークル分科会, 2016.3.22.

9. 渡辺宗一郎, 市川竜太, 原田正範,  
最適軌道に対するスキッドステア式小型無人車両の追従制御,  
日本機械学会第 24 回交通・物流部門大会 (TRANSLOG2015), 2015.12.11.
10. 市川竜太, 渡辺宗一郎, 大久保亜李沙, 原田正範,  
SCR 機の実時間最適制御,  
日本航空宇宙学会第 53 回飛行機シンポジウム, 2015.11.11.
11. 市川竜太, 渡辺宗一郎, 原田正範,  
スキッドステア車両の最適フィードフォワード制御,  
自動車技術会 2015 年春季大会, 2015.5.20.
12. 渡辺宗一郎, 原田正範,  
小型無人車両の繰り返し操作を含む実時間最適誘導制御,  
自動車技術会 2015 年春季大会, 2015.5.20.
13. 渡辺宗一郎, 原田正範,  
高高度無人機の偏西風下における制約空間内の上昇経路最適化,  
日本航空宇宙学会第 46 期年会講演会, 2015.4.17.
14. 市川竜太, 渡辺宗一郎, 原田正範,  
同軸反転ロータ機の自動離着陸制御,  
日本航空宇宙学会第 46 期年会講演会, 2015.4.17.
15. 原田正範, 渡辺宗一郎,  
合流地点到着時間を考慮した経路最適化,  
日本航空宇宙学会第 46 期年会講演会, 2015.4.17.
16. 市川竜太, 渡辺宗一郎, 原田正範,  
 $L_1$  適応制御を用いた SCR 機の誘導制御,  
日本航空宇宙学会第 52 回飛行機シンポジウム, 2014.10.10.
17. 渡辺宗一郎, 市川竜太, 原田正範,  
MAV 用メッシュ状着陸台における着陸面形状の影響の検証,  
日本航空宇宙学会第 52 回飛行機シンポジウム, 2014.10.10.

18. 渡辺宗一郎, 原田正範,  
小型無人車両の最適誘導制御,  
自動車技術会 2014 年春季大会, 2014.5.22.
19. 原田正範, 渡辺宗一郎,  
障害物配置法を用いた競技車両の最適周回走行の解析,  
自動車技術会 2014 年春季大会, 2014.5.21.
20. 渡辺宗一郎, 原田正範,  
MAV 用メッシュ状着陸台まわりの流れの可視化,  
日本航空宇宙学会第 45 期年会講演会, 2014.4.10.
21. 原田正範, 渡辺宗一郎,  
高高度無人機の制約空間内における上昇経路最適化,  
日本航空宇宙学会第 45 期年会講演会, 2014.4.10.
22. 渡辺宗一郎, 原田正範,  
回転型振子のフィードバック線形化制御,  
日本機械学会第 22 回交通・物流部門大会 (TRANSLOG2013), 2013.12.12.
23. 原田正範, 渡辺宗一郎,  
開ループ制御による回転型振子の最短時間運動,  
日本機械学会第 22 回交通・物流部門大会 (TRANSLOG2013), 2013.12.12.
24. 渡辺宗一郎, 原田正範,  
障害物設置法を用いた最適走行,  
日本機械学会第 22 回交通・物流部門大会 (TRANSLOG2013), 2013.12.10.