JAIST Repository
https://dspace.jaist.ac.jp/

| | |
|---|---|
| Title | 動的環境下におけるマルチビークルシステムの制御に関する研究 |
| Author(s) | 永見，琢朗 |
| Citation | |
| Issue Date | 2013-03 |
| Type | Thesis or Dissertation |
| Text version | author |
| URL | http://hdl.handle.net/10119/11344 |
| Rights | |
| Description | Supervisor:平石 邦彦，情報科学研究科，修士 |

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修 士 論 文

# 動的環境下における<br>マルチビークルシステムの制御に関する研究

北陸先端科学技術大学院大学
情報科学研究科

永見 琢朗

2013 年 3 月

修 士 論 文

# 動的環境下における<br>マルチビークルシステムの制御に関する研究

指導教官　平石邦彦 教授

審査委員主査　平石邦彦 教授<br>審査委員　緒方和博 准教授<br>審査委員　浅野文彦 准教授

北陸先端科学技術大学院大学<br>情報科学研究科

1110045 永見 琢朗

提出年月: 2013 年 2 月

Copyright © 2013 by Nagami Takuro

## 概 要

本論文では，マルチビークルシステムに対する協調動作，計算時間低減といった観点からの制御手法を提案する．

対象とするシステムとしてマルチビークルの障害物回避問題を扱う．提案手法では，ビークルの一部状態を離散化し，離散ダイナミクスを有向グラフを用いて記述する．最適制御問題として定式化された障害物回避問題は，混合整数二次計画（MIQP）問題に帰着される．

動的で複雑なマルチビークルシステムにおいては，ビークル間の協調動作をいかに記述するかが重要な課題である．そこで，動的な拘束条件を，線形時相論理式を用いて記述した．線形時相論理式で拘束条件を記述することにより，MIQP 問題への帰着が可能となる．

さらに，離散ダイナミクスを記述した有向グラフに着目し，有向グラフの不要な辺を事前に削除することで，計算時間が低減されることを示した．

# 目 次

**第 1 章　はじめに** **1**

**第 2 章　有向グラフによる拘束を持つシングルビークルシステムの最適制御** **5**
2.1 問題設定 . . . . . 5
2.2 解法 . . . . . 6

**第 3 章　マルチビークルシステムの最適制御** **9**
3.1 準備 . . . . . 9
3.2 問題設定 . . . . . 10
3.3 解法 . . . . . 12
3.3.1 問題 2 の解法 . . . . . 12
3.3.2 問題 3 の解法 . . . . . 12
3.3.3 問題 4 の解法 . . . . . 15

**第 4 章　有向グラフの辺削除による計算時間の低減** **17**
4.1 有向グラフの作成方法 . . . . . 17
4.2 初期状態を考慮した有向グラフの辺削除 . . . . . 19
4.3 ランデブポイントを考慮した有向グラフの辺削除 . . . . . 21
4.4 隣接行列作成，Algorithm 1，Algorithm 2 の計算時間 . . . . . 23

**第 5 章　計算例** **26**
5.1 制御対象 . . . . . 26
5.2 衝突回避に関する計算例 . . . . . 26
5.3 ランデブに関する計算例 . . . . . 28
5.4 計算時間に関する考察 . . . . . 31
5.4.1 予測ステップ数とビークル台数による計算時間の比較 . . . . . 31
5.4.2 各辺削除アルゴリズムによる計算時間の比較 . . . . . 32

**第 6 章　おわりに** **34**

# 図 目 次

4.1 有向グラフ . . . . . 18
4.2 初期位置からの有向グラフ . . . . . 21
4.3 2 台のビークルの有向グラフ . . . . . 22
4.4 ランデブポイントを考慮した有向グラフ . . . . . 24
4.5 有向グラフに関する計算時間 . . . . . 25

5.1 衝突回避 . . . . . 27
5.2 10 ステップ以内にランデブ . . . . . 29
5.3 4-6 ステップでランデブ . . . . . 30
5.4 ビークル台数と予測ステップ数による平均計算時間 . . . . . 31
5.5 辺削除による平均計算時間の比較 . . . . . 33

# 表 目 次

5.1 各協調動作条件の平均計算時間 . . . . . 28
5.2 ビークル台数と予測ステップ数による平均計算時間 . . . . . 32
5.3 ビークル台数による平均計算時間 . . . . . 32

# 第1章　はじめに

近年，ビークルフォーメーションやビークル間の協調動作・制御に関する研究が盛んに行われている [1, 3, 10, 12]．マルチビークルのフォーメーション制御や協調制御の応用例は，災害時における探索ロボットや，高度道路交通システム（ITS），ロボカップサッカーなど多岐に渡り，学術的な研究のみならず，産業界への応用が期待される．また，現代の技術的，社会的側面などから，マルチビークルシステムが有用となるケースは数多く存在する．これらマルチビークルシステムの大部分は，周囲の環境が時々刻々と変化する動的環境であるため，リアルタイム制御が必須である．リアルタイム制御の実現には，計算時間をいかに低減するかという問題が付随し，数多くの研究がなされている．また，マルチビークルシステムにおいては，ビークル間で共通の目的を達成するための協調動作・制御を行わなければならない．このため，協調動作を制御系にどのように記述・実装するか，といった問題も発生する．このように，ビークルシステムに関する研究は，計算時間と協調動作に焦点を当てられることがあり，これらの課題に対して，多数の研究報告がなされている．

マルチビークルシステムの具体例としては，障害物回避問題などがある．この問題は，シングルビークルシステムでは障害物を回避することのみを考えれば良い．しかしながら，マルチビークルシステムでは，ビークル同士の衝突を回避する必要があるなど，ビークル間の協調動作が必要となる．

障害物回避問題など，大規模かつ複雑なシステムに対する解析および制御法において，なんらかの近似による制御対象の簡単化が用いられることがある．制御対象の近似方法としては，時間および空間の離散化，低次元化などがあるが，あるクラスのハイブリッドシステムとして記述可能である場合が多い．典型的な例としては，非線形システムをハイブリッドシステムの一クラスである区分的アファイン（PWA）システムによって近似する方法である [11]．

井村らは，時間および空間を離散化することで，障害物回避問題に代表される動的な非凸状態拘束を有する連続時間線形システムの最適制御問題を，ハイブリッドシステムの一クラスであるスイッチドシステムの最適制御問題に帰着する方法を提案した [4, 5, 8, 14, 15]．この方法は，動的な非凸拘束を課された状態変数，および時間を離散化するものである．解法においては，サンプリング時刻毎に状態経由点を設け，状態経由点およびサンプル点間の区分的連続入力を同時に最適化している点が特徴である．状態経由点は，混合整数二次計画（MIQP）問題を解くことによって得られる．また，この手法では状態経由点の拘束（離散ダイナミクス）の記述に有向グラフを用いている．有向グラフを拘束条件として

問題へ加えることで，障害物への到達回避を行っている．

井村らが提案した手法はビークルの障害物回避問題への適用が可能である．障害物を回避する条件は一般に非凸状態拘束として記述される．サンプリング時刻のみに着目することとし，ビークルの位置を離散化することで，経路探索問題に帰着できる．このとき，サンプリング時刻でのビークルの最適な経路を表す離散ダイナミクス，およびサンプル点間のビークルの軌道を表す連続ダイナミクスが同時に最適化できる．実際，文献 [4, 5, 8, 14] において，ビークルシステムの障害物回避問題が数値例として示されている．

本論文では，上記の手法をマルチビークルシステムへ拡張することを考える．マルチビークルシステムでは，個々のビークルにおける障害物回避，最適制御を考えるだけでなく，ビークル同士の衝突を回避する，指定した時間区間でビークル同士を接近させるなどのさまざまな協調動作を考慮しなければならない．そこで本論文では，ビークル同士の協調動作に関する条件を線形時相論理として記述する．線形時相論理によって静的な拘束条件だけでなく，動的な拘束条件，つまり時間的な制約をもつ条件を記述することが可能となる．例えば，ある時刻ですべてのビークルの速度を 0 とする条件は線形時相論理として記述することができる．一方，ビークルの位置に関する協調動作の条件では，障害物回避も考慮しなければならない．この場合，ビークルの位置を離散化することで，線形時相論理として記述できる．線形時相論理は一般に混合整数線形不等式に帰着できる [6, 7] ことから，マルチビークルシステムの最適制御問題は MIQP 問題に帰着できる．本論文では，いくつかのビークル同士の協調動作を考え，線形時相論理式によって表現された最適制御問題を示す．それらの問題を MIQP 問題へ帰着するために，記述された線形時相論理式を混合整数線形不等式に変換する手順について示す．

また，有向グラフによってビークルの離散ダイナミクスを記述することにより，離散化されたビークルの状態における不要な状態遷移を除くことができる．すなわち，ビークルが移動する可能性のない位置への有向グラフの辺を削除した上で，MIQP 問題の拘束条件として加えることとする．これによって，MIQP 問題における解の候補を減らすことができ，計算時間を低減することが可能となる．本論文では，ビークルの初期状態を考慮した有向グラフの辺削除法および，ビークル同士の協調動作条件を考慮した辺削除法の 2 つのアルゴリズムを提案する．提案したアルゴリズムの有効性を計算例により示す．

本論文の構成は以下の通りである．

- 2 章では，シングルビークルシステムの最適制御問題に関する既存結果について紹介する．
- 3 章では，マルチビークルシステムの最適制御問題を定式化する．また，本論文で考えるビークル間の協調動作条件を説明し，線形時相論理として記述する方法を説明する．
- 4 章では，3 章で記述した各協調動作条件を含む問題を，MIQP 問題へ帰着させる方法を示す．

- 5 章では，ビークルの離散ダイナミクスを表す有向グラフの不要な辺を削除する方法の概要を示す.
- 6 章では，いくつかの協調動作条件について，実際の計算例を示す.
- 7 章はまとめである.

# 表記

- $\mathbf{R}$：実数全体の集合.
- $\mathcal{PC}$：区分的連続関数の集合.
- $I_n$：$n$ 次単位行列.
- $0_{m\times n}$：$m \times n$ の零行列．なお，サイズが明らかである場合は，簡単に $I$，$0$ とする.
- $e_n$：すべての要素が $1$ となる $n$ 次元ベクトル.

# 第2章　有向グラフによる拘束を持つシングルビークルシステムの最適制御

本章では，シングルビークルシステムの場合の結果を紹介する．詳細は文献 [4, 5, 8, 14, 15] などを参照されたい．

## 2.1　問題設定

シングルビークルシステムの場合の問題設定を考える．まず，ビークルのダイナミクスとして，次の可制御な線形システム

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.1}$$

を考える．ここで，$x(t) \in \mathbf{R}^n$ は状態，$u(t) \in \mathbf{R}^m$ は入力である．状態 $x(t)$ は

$$x(t) = \begin{bmatrix} x^c(t) \\ x^d(t) \end{bmatrix}, \quad x^c(t) \in \mathbf{R}^{n_c}, \quad x^d(t) \in \mathbf{R}^{n_d}$$

と分割することとし，$x^c(t)$ は連続値，$x^d(t)$ は離散値として扱うこととする．

次に，サンプリング時刻 $t_i, i = k, k+1, \ldots$ が与えられているとする．このとき，時刻 $t_i$ での状態 $x(t)$ の経由点を

$$x_i = \begin{bmatrix} x_i^c \\ x_i^d \end{bmatrix}, \quad x_i^c \in \mathbf{R}^{n_c}, \quad x_i^d \in \mathbf{R}^{n_d}$$

と表記し，$x(t_i) = x_i$ という拘束を課すこととする．離散値の状態 $x^d(t)$ の経由点 $x_i^d$ に対する拘束は

$$x_{i+1}^d \in \mathcal{E}_i(x_i^d) \tag{2.2}$$

として与えられているとする．ここで $\mathcal{E}_i : \mathcal{F}_i \to 2^{\mathcal{F}_i}$ は集合値関数である．$\mathcal{F}_i \subset \mathbf{R}^{n_d}$ は各サンプリング時刻ごとに与えられている有限集合，$2^{\mathcal{F}_i}$ は $\mathcal{F}_i$ のべき集合である．$\mathcal{E}_i$ は $\mathcal{F}_i$ の要素から要素への有向グラフとして表現されることから，(2.2) 式を有向グラフによる拘束と呼ぶこととする．

加えて，各サンプリング時刻において，状態と入力の線形不等式拘束

$$Cx(t_i) + Du(t_i) \leq G \tag{2.3}$$

を課すこととする．

以上の準備のもとで，$[t, t+T)$ を評価区間とした次の最適制御問題を考える．

**問題 1** システム (2.1) が与えられているとする．また，$t_k = t$，$t_{k+N} = t+T$，および $x(t_k) = x_k$ が与えられているとする．このとき，拘束条件 $x(t_i) = x_i$，(2.2) 式，および (2.3) 式のもとで，次の評価関数

$$J(x_k, u(\cdot), \{x_i\}_{i=k+1,k+2,\ldots,k+N}) = \sum_{i=k}^{k+N-1} J_i(x_i, u(\cdot), x_{i+1}),$$

$$J_i(x_i, u(\cdot), x_{i+1}) = \int_{t_i}^{t_{i+1}} \left\{(x(t)-x_{i+1})^T Q(x(t)-x_{i+1}) + u^T(t) R u(t)\right\} dt$$

を最小化する状態フィードバック制御器 $u(t) \in \mathcal{PC}^m$，$t \in [t_k, t_{k+N})$ および状態経由点 $x_i \in \mathbf{R}^n$，$i = k+1, k+2, \ldots, k+N$ を求めよ．ここで $Q \geq 0$，$R > 0$ である．

問題 1 をビークルの障害物回避問題に適用することを考える．この場合，障害物やビークルの位置に関する拘束は一般に非凸となり，そのまま解くことは困難である．このとき，状態の一部である位置を離散化することで，障害物回避に関する拘束は (2.2) 式で表現できる．したがって，障害物回避問題は問題 1 で近似的に表現できる．詳細は文献 [4, 5, 8, 14] を参照されたい．また，状態の離散化については，すでにいくつかの結果（たとえば，文献 [13]）が得られているので，本論文では触れないこととする．

## 2.2 解法

問題 1 の最適な経由点を求める問題は，次の離散時間線形システムの最適制御問題に帰着される [5].

**問題 A**

$$\begin{aligned}
\min_{v_i,\ i=k,k+1,\ldots,k+N-1} \quad & \sum_{i=k}^{k+N-1} \begin{bmatrix} x_i \\ v_i \end{bmatrix}^T S^i \begin{bmatrix} x_i \\ v_i \end{bmatrix} \\
\text{subject to} \quad & x_{i+1} = v_i, \quad x_k : \text{given}, \\
& \hat{C}_i x_i + \hat{D}_i v_i \leq G, \\
& v_i = \begin{bmatrix} v_i^c \\ v_i^d \end{bmatrix}, \\
& v_i^c \in \mathbf{R}^{n_c}, \\
& v_i^d \in \mathcal{E}_i(v_{i-1}^d), \quad v_{k-1}^d = x_k^d
\end{aligned}$$

ここで

$$
\begin{aligned}
S^i &= \begin{bmatrix} S_1^i & S_2^i \\ (S_2^i)^T & S_3^i \end{bmatrix}, \\
\hat{C}_i &:= C - DR^{-1}B^T(P + e^{\tilde{A}^T h_i} W_i^{-1} \Phi_i), \\
\hat{D}_i &:= DR^{-1}B^T(P + \tilde{A}^{-T}PA + e^{\tilde{A}^T h_i} W_i^{-1} \Psi_i)
\end{aligned}
$$

および

$$
\begin{aligned}
S_1^i &:= \Phi_i^T W_i^{-1} \Phi_i + P \in \mathbf{R}^{n\times n}, \\
S_2^i &:= -\Phi_i^T W_i^{-1} \Psi_i - P - \tilde{A}^{-T}PA \in \mathbf{R}^{n\times n}, \\
S_3^i &:= \Psi_i^T W_i^{-1} \Psi_i + P + \tilde{A}^{-T}PA + A^T P \tilde{A}^{-1} + h_i A^T P \tilde{A}^{-1} P^{-1} Q P^{-1} \tilde{A}^{-T} PA \in \mathbf{R}^{n\times n}, \\
\Phi_i &:= e^{\tilde{A}h_i}, \\
\Psi_i &:= e^{\tilde{A}h_i} - \int_0^{h_i} e^{\tilde{A}\tau} d\tau (I + BR^{-1}B^T \tilde{A}^{-T} P) A
\end{aligned}
$$

である．また，$P$ および $W_i$ は次の Riccati 代数方程式および Lyapunov 方程式

$$
\begin{aligned}
&PA + A^T P - PBR^{-1}B^T P + Q = 0, \\
&W_i \tilde{A}^T + \tilde{A} W_i + BR^{-1}B^T - e^{\tilde{A}h_i} BR^{-1}B^T e^{\tilde{A}^T h_i} = 0
\end{aligned}
$$

の正定対称解である．ここで $h_i := t_{i+1} - t_i$ および $\tilde{A} := A - BR^{-1}B^T P$ である．

また，最適な経由点を $x_i^*$ とすると，最適な制御入力 $u^*(t)$ は

$$
\begin{aligned}
u^*(t) = &-R^{-1}B^T P(x(t) - x_{i+1}^*) + R^{-1}B^T \tilde{A}^{-T} PA x_{i+1}^* \\
&-R^{-1}B^T e^{\tilde{A}^T(t_{i+1}-t)} W_i^{-1}(\Phi_i x(t_i) - \Psi_i x_{i+1}^*), \\
&t \in [t_i, t_{i+1}),\ i = k, k+1, \ldots, k+N-1
\end{aligned}
$$

として得られる．

次に，問題 A を混合整数二次計画（MIQP）問題に帰着することを考える．文献 [15] と同様に，簡単のため，有限集合 $\mathcal{F}_i$ の要素数は一定として，$n_f$ として与えられているとする．また，有限集合 $\mathcal{F}_i$ は

$$
\mathcal{F}_i = \{\lambda_1^i, \lambda_2^i, \ldots, \lambda_{n_f}^i\} \tag{2.4}
$$

として与えられていると仮定する．ここで $\lambda_j^i \in \mathbf{R}^{n_d}$ である．このとき，離散値の状態の経由点 $x_i^d$ を

$$
x_i^d = \Lambda_i x_i^f, \quad \Lambda_i := \begin{bmatrix} \lambda_1^i & \lambda_2^i & \cdots & \lambda_{n_f}^i \end{bmatrix} \in \mathbf{R}^{n_d \times n_f} \tag{2.5}
$$

と表現する．ここで $x^f \in \{0,1\}^{n_f}$ は $e_{n_f}^T x_i^f = 1$ を満足する 0-1 変数であり，時刻 $t_i$ で離散値の状態が $\mathcal{F}_i$ のどの要素を取るかを表している．さらに，問題 A の $x_i$ および $v_i$ は

$$
\begin{bmatrix} x_i \\ v_i \end{bmatrix} = \begin{bmatrix} \Lambda_1^i & 0 \\ \Lambda_2^i & \Lambda_3^i \end{bmatrix} \begin{bmatrix} \tilde{x}_i \\ \tilde{v}_i \end{bmatrix} \tag{2.6}
$$

と表現できる．ここで

$$\tilde{x}_i := \begin{bmatrix} x_i^c \\ x_i^f \end{bmatrix} \in \mathbf{R}^{n_c} \times \{0,1\}^{n_f},$$
$$\tilde{v}_i := \begin{bmatrix} v_i^c \\ u_i^f \end{bmatrix} \in \mathbf{R}^{n_c} \times \{0,1\}^{m_i^f},$$
$$\Lambda_1^i := \text{block-diag}(I_{n_c}, \Lambda_i),$$
$$\Lambda_2^i := \text{block-diag}(0_{n_c \times n_c}, \Lambda_i A_i^f),$$
$$\Lambda_3^i := \text{block-diag}(I_{n_c}, \Lambda_i B_i^f)$$

である．$A_i^f$，$B_i^f$ は，(2.2) 式を表現する状態方程式

$$\begin{cases} x_{i+1}^f = A_i^f x_i^f + B_i^f u_i^f, \\ C_i^f x_i^f + D_i^f u_i^f \leq G_i^f, \\ x_i^f \in \mathbf{R}^{n_f}, \quad u_i^f \in \{0,1\}^{m_i^f}, \\ x_k^f \in \left\{ \eta \in \{0,1\}^{n_f} \mid e_{n_f}^T \eta = 1 \right\} \end{cases}$$

の係数行列である．本研究で扱うビークルシステムにおいては，$x_{i+1}^f = u_i^f$ として考える．よって，$A_i^f$ および $B_i^f$ は

$$A_i^f = 0_{n_f \times n_f}, \ B_i^f = I_{n_f \times n_f}$$

となる．$C_i^f$, $D_i^f$, $G_i^f$ は $e_{n_f}^T \eta = 1$ および拘束条件 (2.2) を表す．拘束条件 (2.2) は，隣接行列として表現する．例として，$n_f = 5$，時刻 $t_k$での 隣接行列 $H_k$ を次のように仮定する．

$$H_k = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

このとき，$C_k^f$, $D_k^f$, $G_k^f$ はそれぞれ

$$C_k^f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad D_k^f = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad G_k^f = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

となる．$i = k+1, k+2, \ldots, k+N$ も同様にして求めることができる．詳細は文献 [9, 16] を参照されたい．(2.6) 式を問題 A に代入することで，問題 A は MIQP 問題に帰着される．

# 第3章　マルチビークルシステムの最適制御

## 3.1　準備

2節の結果をマルチビークルシステムへ拡張することを考える．基本的には，ビークル毎に問題1を解くことを考えればよい．しかしながら，障害物回避を行いながら，協調動作を行う場合は，位置情報は各ビークルで同一でなければならない．そこで，(2.4) 式の有限集合 $\mathcal{F}_i$ および (2.5) 式の $\Lambda_i$ は各ビークルで同じ集合を考えることとする．

マルチビークルシステムの場合の問題設定を与える．ビークル数を $p$ とする．各ビークルに対して，可制御な線形システム

$$\dot{x}_j(t) = A_j x_j(t) + B_j u_j(t) \tag{3.1}$$

が与えられているとする．ここで $x_j(t) \in \mathbf{R}^n$，$u_j(t) \in \mathbf{R}^m$，$j = 1, 2, \ldots, p$ である．状態の分割および経由点は

$$x_j(t) = \begin{bmatrix} x_j^c(t) \\ x_j^d(t) \end{bmatrix}, \quad x_j^c(t) \in \mathbf{R}^{n_c}, \quad x_j^d(t) \in \mathbf{R}^{n_d},$$
$$x_i = \begin{bmatrix} x_{i,j}^c \\ x_{i,j}^d \end{bmatrix}, \quad x_{i,j}^c \in \mathbf{R}^{n_c}, \quad x_{i,j}^d \in \mathbf{R}^{n_d}$$

と表記する．なお，サンプリング時刻 $t_i$ はビークルに依存せず，共通で与えられていることとする．離散値の状態 $x_j^d(t)$ の経由点 $x_{i,j}^d$ に対する拘束は

$$x_{i+1,j}^d \in \mathcal{E}_{i,j}(x_{i,j}^d) \tag{3.2}$$

として与えられているとする．ここで $\mathcal{E}_{i,j} : \mathcal{F}_i \to 2^{\mathcal{F}_i}$ は集合値関数である．加えて，各サンプリング時刻において，状態と入力の線形不等式拘束

$$C_j x_j(t_i) + D_j u_j(t_i) \leq G_j \tag{3.3}$$

を課すこととする．

## 3.2 問題設定

以上の準備のもとで，$[t, t+T)$ を評価区間とした最適制御問題を考える．なお，本論文で扱うビークルシステムでは

$$x^c(t) = \begin{bmatrix} x_d(t) \\ \dot{x}_p(t) \end{bmatrix}, \quad x^d(t) = x_p(t)$$

とする．ここで，$x_p(t),\ \dot{x}_p(t),\ x_d(t) \in \mathbf{R}$ はそれぞれビークルの位置，速度，駆動力である．

以下ではビークルの台数を $p = 2$ とした，以下の 3 つの問題を考える．

**問題 2** 各ビークルに対して，システム (3.1) が与えられているとする．また，$t_k = t$，$t_{k+N} = t + T$，および $x_j(t_k) = x_{k,j}$，$j = 1, 2$ が与えられているとする．このとき，拘束条件 $x_j(t_i) = x_{i,j}$，(3.2) 式，(3.3) 式，および「サンプリング時刻でビークル同士が衝突しない」という意味を持つ次の協調動作条件

$$x^d_{i,1} \neq x^d_{i,2}, \quad i = k, k+1, \ldots, k+N-1 \tag{3.4}$$

のもとで，次の評価関数

$$\begin{aligned} &J(x_{k,j}, u_j(\cdot), \{x_{i,j}\}_{i=k+1,k+2,\ldots,k+N, j=1,2,\ldots,p}) \\ &\qquad = \sum_{j=1}^{p} J_j(x_{k,j}, u_j(\cdot), \{x_{i,j}\}_{i=k+1,k+2,\ldots,k+N}), \end{aligned} \tag{3.5}$$

$$J_j(x_{k,j}, u_j(\cdot), \{x_{i,j}\}_{i=k+1,k+2,\ldots,k+N}) = \sum_{i=k}^{k+N-1} J_{i,j}(x_{i,j}, u_j(\cdot), x_{i+1,j}),$$

$$J_{i,j}(x_{i,j}, u_j(\cdot), x_{i+1,j}) = \int_{t_i}^{t_{i+1}} \left\{ (x_j(t) - x_{i+1,j})^T Q_j (x_j(t) - x_{i+1,j}) \ + u_j^T(t) R_j u_j(t) \right\} dt$$

を最小化する各ビークルの状態フィードバック制御器 $u_j(t) \in \mathcal{PC}^m$，$t \in [t_k, t_{k+N})$ および状態経由点 $x_{i,j} \in \mathbf{R}^n$，$i = k+1, k+2, \ldots, k+N$ を求めよ．ここで $Q_j \geq 0$，$R_j > 0$ である．

問題 2 は，静的な協調動作条件を達成するような問題である．以下の問題では，動的な協調動作条件を考える．時刻に依存した協調動作条件を記述するために，線形時相論理 (LTL) 式を用いる．

LTL 式の演算子は，命題論理の演算子，論理和 ($\vee$)，論理積 ($\wedge$)，含意 ($\Rightarrow$)，否定 ($\neg$) に加えて，時間に関する性質を表現する演算子，next($\circ$)，until($\mathcal{U}$)，always($\Box$)，eventually($\diamond$) が用いられる．

また，上記の演算子に加えて，以下の演算子（$\diamond_r$，$\diamond_r^1$，$\circ_r$）を定義する．まず，$\diamond_r \varphi$ が真とは「$r$ ステップ以内に命題 $\varphi$ が真になる」を意味する．$r = 2$ とすると

$$\diamond_2 \varphi := \varphi \vee \circ\varphi \vee \circ \circ \varphi \tag{3.6}$$

と表される．次に，$\diamond_r^1\varphi$ が真とは「$r$ ステップ以内に一度だけ命題 $\varphi$ が真になる」を意味する．この演算子は next 演算子 ($\circ$) を用いて表現できる．例として「2 ステップ以内に命題 $\varphi$ が一度だけ真となる」は

$$\diamond_2^1\varphi := (\varphi \wedge \neg \circ \varphi \wedge \neg \circ \circ\varphi) \vee (\neg\varphi \wedge \circ\varphi \wedge \neg \circ \circ\varphi) \vee (\neg\varphi \wedge \neg \circ \varphi \wedge \circ \circ \varphi) \tag{3.7}$$

と表現される．最後に，$\circ_r\varphi$ が真とは「$r$ ステップ後に命題 $\varphi$ が真になる」を意味する．例として $r = 3$ の場合は

$$\circ_3\varphi := \circ \circ \circ\varphi \tag{3.8}$$

と表される．

さらに,「2 台のビークルの位置が同一時刻で一致し，このとき各ビークルの速度が 0 となる」という協調動作を「ランデブ」と呼ぶこととする．以下では，$\varphi$ は「ランデブをしているとき真，そうでなければ偽」を意味する命題とする．

以上を踏まえて，次の 2 つの問題を考える．

**問題 3** 各ビークルに対して，システム (3.1) が与えられているとする．また，$t_k = t$, $t_{k+N} = t + T$，$[t_k, t_{k+N}]$ 内の時刻 $t_{k+r}$，および $x_j(t_k) = x_{k,j}$，$i = k, k+1, \ldots, k+N$, $j = 1, 2$ が与えられているとする．このとき，拘束条件 $x_j(t_i) = x_{i,j}$，(3.2) 式，(3.3) 式，および「現在時刻 $t_k$ から $t_{k+r}$ 以内に一度だけランデブをする」という意味をもつ次の LTL 式

$$\phi = \diamond_r^1\varphi \wedge \circ_{r+1}\Box\neg\varphi \tag{3.9}$$

が真となる拘束条件のもとで，評価関数 (3.5) を最小化する各ビークルの状態フィードバック制御器 $u_j(t) \in \mathcal{PC}^m$, $t \in [t_k, t_{k+N})$ および状態経由点 $x_{i,j} \in \mathbf{R}^n$, $i = k+1, k+2, \ldots, k+N$ を求めよ．ここで $Q_j \geq 0$，$R_j > 0$ である．

**問題 4** 各ビークルに対して，システム (3.1) が与えられているとする．また，$t_k = t$, $t_{k+N} = t + T$，$[t_k, t_{k+N}]$ 内の時刻 $t_{k+r_1}, t_{k+r_2}$ $(r_1 \leq r_2)$，および $x_j(t_k) = x_{k,j}$，$i = k, k+1, \ldots, k+N$，$j = 1, 2$ が与えられているとする．このとき，拘束条件 $x_j(t_i) = x_{i,j}$, (3.2) 式，(3.3) 式，および「現在時刻 $t_k$ において $t_{k+r_1}$ から $t_{k+r_2}$ の間に一度だけランデブをする」という意味をもつ次の LTL 式

$$\phi = \neg \diamond_{r_1-1} \varphi \wedge \diamond_{r_2}^1\varphi \wedge \circ_{r_2+1}\Box\neg\varphi \tag{3.10}$$

が真となる拘束条件のもとで，評価関数 (3.5) を最小化する各ビークルの状態フィードバック制御器 $u_j(t) \in \mathcal{PC}^m$, $t \in [t_k, t_{k+N})$ および状態経由点 $x_{i,j} \in \mathcal{R}^n$, $i = k+1, k+2, \ldots, k+N$ を求めよ．

## 3.3 解法

### 3.3.1 問題 2 の解法

協調動作条件 (3.4) 式は，(2.5) 式の $x_i^f$ を用いると表現できる．$x_{i,j}^f$ をビークル $j$ の経由点 $x_{i,j}^p$（ビークルの位置）を表す 0-1 変数のベクトルとする．このとき，(3.4) 式は，次の線形不等式

$$\sum_{j=1}^{p} x_{i,j}^f \leq 1, \quad i = k, k+1, \ldots, k+N-1$$

と等価である．したがって，問題 2 は MIQP 問題に帰着される．

### 3.3.2 問題 3 の解法

LTL 式によって記述された拘束条件は MIQP 問題に帰着できることが Karaman らの研究によって示されている [6, 7]．そこで，LTL 式によって記述された協調動作条件 (3.9) を MIQP 問題に帰着させることを考える．

まず，LTL 式 (3.9) の意味について説明する．LTL 式 (3.9) の第一項

$$\phi = \diamond_r^1 \varphi$$

はすでに説明したように,「現在時刻 $t_k$ から $t_{k+r}$ で一度だけランデブをする」を意味する．区間 $t_k - t_{k+r}$ 以外では，ランデブを禁止する拘束条件を課す必要がある．第二項

$$\phi = \circ_{r+1} \Box \neg \varphi$$

は，ランデブが許可される時刻 $t_{k+r}$ 以降でのランデブを禁止する意味を持つ．$\Box\phi$ は「今後ずっと $\phi$ が成り立つ」という意味を持つ LTL 演算子である．一方，$\circ_{r+1}\phi$ は「現在時刻 $t_k$ において，時刻 $t_{k+r+1}$ で $\phi$ が成り立つ」であるから，LTL 式 (3.9) の第二項は「時刻 $t_{k+r+1}$ 以降ずっとランデブをしない」という論理式となる．よって，LTL 式 (3.9) は「現在時刻 $t_k$ から $t_{k+r}$ 以内に一度だけランデブをする」という意味をもつ論理式である．

次に，ランデブの動作を線形不等式に変換する．時刻 $t_i$ および位置 $\lambda_s^i \in \mathcal{F}_i$, $s \in \{1, 2, \ldots, n_f\}$ （$n_f$ は位置の離散点数に対応する）で各ビークルの位置を一致させる拘束条件は 0-1 変数 $\delta_{i,s} \in \{0, 1\}$ を用いて

$$\delta_{i,s} = x_{i,1,s}^f \wedge x_{i,2,s}^f \tag{3.11}$$

という論理式で表すことができる．ここで，$x_{i,j,s}^f$ はベクトル $x_{i,j}^f$ の $s$ 番目の要素を表す．拘束条件 (3.11) は，$\delta_{i,s} = 1$ のとき各ビークルの位置が $x_i^d = \lambda_s^i$ となることを意味する．論理式 (3.11) は

$$\begin{aligned} -x_{i,1,s}^f + \delta_{i,s} &\leq 0, \\ -x_{i,2,s}^f + \delta_{i,s} &\leq 0, \\ x_{i,1,s}^f + x_{i,2,s}^f - \delta_{i,s} &\leq 1 \end{aligned}$$

という線形不等式で表すことができる [2]．したがって「各ビークルがランデブをする」という命題 $\varphi$ は

$$\varphi = \bigvee_{s\in\{1,2,\ldots,n_f\}} \delta_{i,s}$$

として表現できる．命題 $\varphi$ の時間発展を考える．$P^i_\varphi \in \{0,1\}$ は,「$t = t_i$ のとき命題 $\varphi$ が真であれば 1，そうでなければ 0」を意味する 0-1 変数である．このとき，$P^i_\varphi \in \{0,1\}$ の時系列を

$$\boldsymbol{P}_\varphi = (P^k_\varphi, P^{k+1}_\varphi, \ldots, P^{k+N}_\varphi)$$

と表す．$P^i_\varphi$，$i = k, k+1, \ldots, k+N$ は，0-1 変数であることに注意すると

$$\begin{cases} P^i_\varphi \le \sum_{s=1}^{n_f} \delta_{i,s}, \\ P^i_\varphi \ge \delta_{i,s}, \quad s = 1, 2, \ldots, n_f \end{cases}$$

として得ることができる．次に，$P^i_\varphi = 1$ のとき各ビークルの速度を $\dot{x}^p_{i,j} = 0$ とする拘束条件は

$$(1 - P^i_\varphi)\dot{x}^p_{\min} \le \dot{x}_{p,j} \le (1 - P^i_\varphi)\dot{x}^p_{\max}, \qquad j \in \{1, 2\}$$

で表される．ここで，$\dot{x}^p_{\min}$ および $\dot{x}^p_{\max}$ はサンプリング時刻での速度の上下限を表し，あらかじめ与えられているとする．

次に，LTL 式で記述された協調動作条件 (3.9) を線形不等式で記述することを考える．$t = t_k$ での命題 $\phi$ の真偽を $P^k_\phi \in \{0,1\}$ で表すこととする（真を 1，偽を 0 とする）．

論理式 (3.9) を $\psi_1 = \diamond^1_r\varphi,\ \psi_2 = \circ_r\Box\neg\varphi$ とおく．簡単のため，$r = 2$ の場合について示すこととする．

まず，$\psi_1 = \diamond^1_2\varphi$，すなわち論理式 (3.7) を線形不等式に変換する手順を示す．まず，$\psi_3 = \varphi,\ \psi_4 = \circ\varphi,\ \psi_5 = \circ\circ\varphi$ とし，各々を線形不等式に変換する．論理式 $\psi_3 = \varphi$ はすでに示したように

$$P^k_{\psi_3} = P^k_\varphi$$

で表される．論理式 $\psi_4 = \circ\varphi,\ \psi_5 = \circ\circ\varphi$ はそれぞれ

$$P^k_{\psi_4} = P^{k+1}_\varphi,$$
$$P^k_{\psi_5} = P^{k+2}_\varphi$$

となる．論理式 (3.7) は

$$\diamond^1_2 p = (\psi_3 \wedge \neg\psi_4 \wedge \neg\psi_5) \vee (\neg\psi_3 \wedge \psi_4 \wedge \neg\psi_5) \vee (\neg\psi_3 \wedge \neg\psi_4 \wedge \psi_5)$$

となる．次に，論理式 $\psi_6 = \neg\psi_3,\ \psi_7 = \neg\psi_4,\ \psi_8 = \neg\psi_5$ は以下の線形不等式で表される．

$$\begin{aligned} P^k_{\psi_6} &= (1 - P^k_{\psi_3}), \\ P^k_{\psi_7} &= (1 - P^k_{\psi_4}), \\ P^k_{\psi_8} &= (1 - P^k_{\psi_5}) \end{aligned}$$

論理式 (3.7) は

$$\diamond^1_2 p = (\psi_3 \wedge \psi_7 \wedge \psi_8) \vee (\psi_6 \wedge \psi_4 \wedge \psi_8) \vee (\psi_6 \wedge \psi_7 \wedge \psi_5)$$

と書き換えられる．$\psi_9 = \psi_3 \wedge \psi_7 \wedge \psi_8,\ \psi_{10} = \psi_6 \wedge \psi_4 \wedge \psi_8,\ \psi_{11} = \psi_6 \wedge \psi_7 \wedge \psi_5$ とすると，各論理式はそれぞれ次の線形不等式に変換できる [2].

$$\begin{cases} P^k_{\psi_9} \le P^k_{\psi_q}, \quad q \in \{3,7,8\}, \\ P^k_{\psi_9} \ge \sum_{q\in\{3,7,8\}} P^k_{\psi_q} - 2 \end{cases}$$
$$\begin{cases} P^k_{\psi_{10}} \le P^k_{\psi_q}, \quad q \in \{4,6,8\}, \\ P^k_{\psi_{10}} \ge \sum_{q\in\{4,6,8\}} P^k_{\psi_q} - 2 \end{cases}$$
$$\begin{cases} P^k_{\psi_{11}} \le P^k_{\psi_q}, \quad q \in \{5,6,7\}, \\ P^k_{\psi_{11}} \ge \sum_{q\in\{5,6,7\}} P^k_{\psi_q} - 2 \end{cases}$$

ゆえに，論理式 (3.7) は

$$\diamond^1_2 \varphi = \psi_9 \vee \psi_{10} \vee \psi_{11}$$

となり，線形不等式

$$\begin{cases} P^k_{\psi_1} \le \sum_{q\in\{9,10,11\}} P^k_{\psi_q}, \\ P^k_{\psi_1} \ge P^k_{\psi_q}, \quad q \in \{9,10,11\} \end{cases}$$

に変換できる．

次に，$\psi_2 = \circ_3\Box\neg\varphi$，すなわち論理式 (3.8) を線形不等式に変換する．まず，$\psi_{12} = \neg\varphi$ を線形不等式で表すと

$$P^i_{\psi_{12}} = (1 - P^i_{\varphi})$$

となる．次に，$\psi_{13} = \Box\psi_{12}$ は

$$\begin{cases} P^i_{\psi_{13}} \le P^\tau_{\psi_{12}},\ \tau \in \{i, \ldots, k+N\},\ i \in \{k, \ldots, k+N\}, \\ P^i_{\psi_{13}} \ge \sum_{\tau=i}^{k+N} P^\tau_{\psi_{12}} - (k+N-i),\ i \in \{k, \ldots, k+N\} \end{cases}$$

となる．よって，論理式 (3.8) は以下の線形不等式に変換される．

$$P^k_{\psi_2} = P^{k+3}_{\psi_{13}}$$

最後に，論理式 (3.9) は

$$\phi = \psi_1 \wedge \psi_2$$

と表せることから，以下の線形不等式に変換される．

$$\begin{cases} P^k_\phi \leq P^k_{\psi_q}, \quad q \in \{1,2\} \\ P^k_\phi \geq \sum_{q\in\{1,2\}} P^k_{\psi_q} - 1 \end{cases}$$

以上より，問題 3 は MIQP 問題に帰着される．

### 3.3.3 問題 4 の解法

LTL 式で記述された協調動作条件 (3.10) を線形不等式で表現することを考える．まず，LTL 式 (3.10) の意味について説明する．LTL 式 (3.10) の第二項および第三項は，問題 3 と同様の意味で，「現在時刻 $t_k$ から $t_{k+r_2}$ で一度だけランデブをする」という論理式である．$t_{k+r_1}$ から $t_{k+r_2}$ の時間区間内のみで一度だけランデブを許可しなければならないため，それ以外の区間ではランデブを禁止する必要がある．LTL 式 (3.10) の第一項

$$\phi = \neg \diamond_{r_1-1} \varphi$$

は，「時刻 $t_{k+r_1}$ 以前ではランデブを禁止する」を意味する論理式である．この論理式によって LTL 式 (3.10) は「現在時刻 $t_k$ において $t_{k+r_1}$ から $t_{k+r_2}$ の間に一度だけランデブをする」という意味を持つ論理式となる．

次に，協調動作条件 (3.10) を線形不等式に変換する手順を示す．$\psi_1 = \diamond_{r_1-1}\varphi$, $\psi_2 = \diamond^1_{r_2}\varphi \wedge \circ_{r_2+1}\Box\neg\varphi$ とすれば，論理式 (3.10) は

$$\phi = \neg\psi_1 \wedge \psi_2$$

となる．$\psi_2$ は問題 3 と同様の手順で線形不等式変換できる．次に，論理式 $\psi_1$ を線形不等式に変換することを考える．ここでは，$r_1 = 3$ の場合すなわち論理式 (3.6) を線形不等式に変換することを考える．$\psi_3 = \varphi$, $\psi_4 = \circ\varphi$, $\psi_5 = \circ\circ\varphi$ とする．各々の変換については，問題 3 の解法で示したので省略する．論理式 (3.6) は

$$\diamond_2\varphi = \psi_3 \vee \psi_4 \vee \psi_5$$

と書き換えることができる．よって，論理式 (3.6) は以下の線形不等式に変換される．

$$\begin{cases} P^k_{\psi_1} \leq \sum_{q\in\{3,4,5\}} P^k_{\psi_q}, \\ P^k_{\psi_1} \geq P^k_{\psi_q}, \quad q \in \{3,4,5\} \end{cases}$$

ここで，$\psi_6 = \neg\psi_1$ とすると，$\psi_6$ は

$$P^k_{\psi_6} = (1 - P^k_{\psi_1})$$

のような線形不等式に変換される．したがって，論理式 (3.10) は以下の線形不等式に変換できる．

$$\begin{cases} P_{\phi}^{k} \leq P_{\psi_q}^{k}, \quad q \in \{2, 6\}, \\ P_{\phi}^{k} \geq \sum_{q \in \{2,6\}} P_{\psi_q}^{k} - 1 \end{cases}$$

以上より，問題 4 は MIQP 問題に帰着される．

# 第4章　有向グラフの辺削除による計算時間の低減

離散ダイナミクスを表す有向グラフは，ビークルの初期状態からの状態遷移を考慮することによって，不要な辺を削除することができる．また，2台のビークルが協調動作としてランデブを行う場合，ランデブを行う時刻と位置を考慮することにより，有向グラフ上で到達の可能性がない頂点が発生する．本章では，有向グラフの作成方法，初期状態およびランデブポイントを考慮した辺削除のアルゴリズムについて述べる．

## 4.1　有向グラフの作成方法

まず，(3.2) 式の有向グラフの作成方法について，例題を用いて説明する．

有向グラフは障害物の情報から決定される．ここで，$\mathcal{F}_i = \{1, 2, \ldots, 11\},\ n_f = 11$ と仮定し，$i = k, k+1, \ldots, k+N-1$，$j = 1, 2, \ldots, p$ が与えられているとする．このとき (3.2) 式の $\mathcal{E}_{i,j}(x^d_{i,j})$ は

$$
\begin{aligned}
&x^d_{i+1,j} \in \{1, 2\} \cap \mathcal{V}_{i+1} \quad \text{if } x^d_{i,j} = 1 \\
&x^d_{i+1,j} \in \{10, 11\} \cap \mathcal{V}_{i+1} \quad \text{if } x^d_{i,j} = 11 \\
&x^d_{i+1,j} \in \{x^d_{i,j} - 1, x^d_{i,j}, x^d_{i,j} + 1\} \cap \mathcal{V}_{i+1} \quad \text{otherwise}
\end{aligned}
$$

で与えられる．$\mathcal{V}_i$ は時刻 $t_i$ における許容領域，すなわち障害物の存在しない領域である．例えば，時刻 $t_i$ において，障害物が $1, 2, 5, 8, 11 \in \mathcal{F}_i$ に存在するとき，$\mathcal{V}_i = \{3, 4, 6, 7, 9, 10\}$ となる．

この有向グラフを隣接行列 $H_i \in \mathbf{R}^{n_f \times n_f}$ を用いて表現する．例として，Fig. 4.1 のグラフ構造の場合，$t_k - t_{k+1}$ 間の隣接行列は

$$
H_k = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

と表現できる．この隣接行列を各時刻間で作成することで有向グラフを表現する．

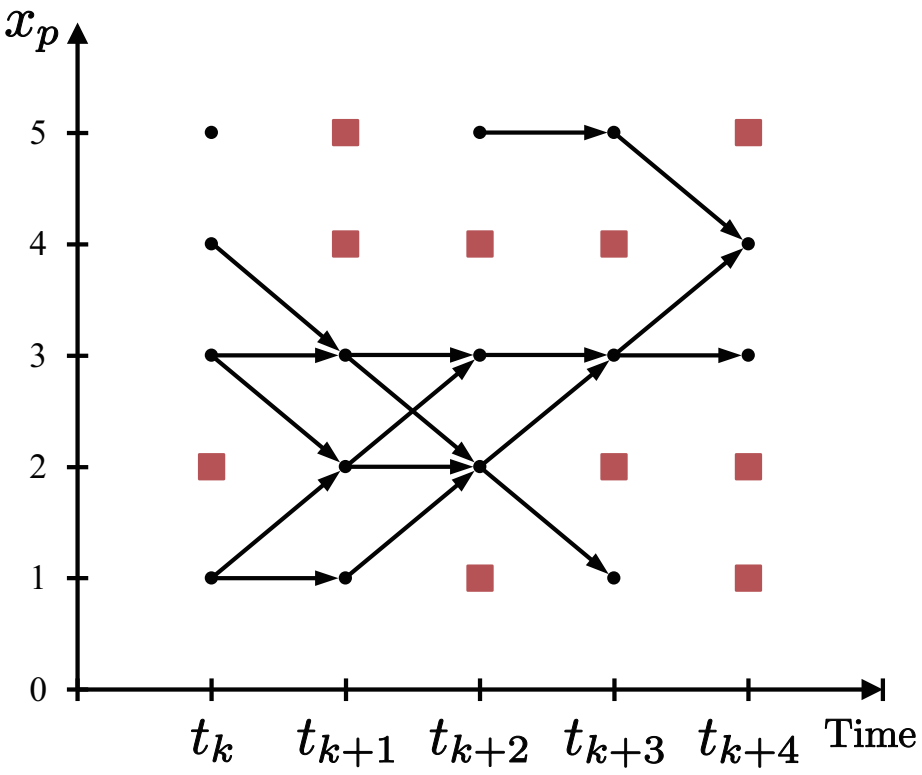



図 4.1: 有向グラフ

## 4.2 初期状態を考慮した有向グラフの辺削除

上記の方法によって作成された有向グラフは，状態遷移を表す辺をすべて記述している．しかし，辺の数が多ければ経路数も多くなるため，最適経路の探索に時間がかかってしまう．そこで，各ビークルの初期位置を考慮することによって，有向グラフにおける不要な辺を削除する．

現在時刻 $t_k$ において，ビークルの初期位置 $x^d_{k,j}$ が一意に与えられたと仮定する．このとき，次時刻 $t_{k+1}$ への辺は，頂点 $x^d_{k+1,j} \in \mathcal{E}_{k,j}(x^d_{k,j})$ へ結ばれるものに限られる．さらに $t_{k+2}$ では，頂点 $x^d_{k+1,j} \in \mathcal{E}_{k,j}(x^d_{k,j})$ から頂点 $x^d_{k+2,j} \in \mathcal{E}_{k+1,j}(x^d_{k+1,j})$ へ結ばれる辺のみを残す．これを $i = k + N - 1$ まで繰り返し，有向グラフ $\mathcal{E}'_i$ を作成する．作成された有向グラフは，初期位置からの辺が結ばれた構造となる．

また，最終時刻 $t_{k+N}$ への到達可能性を考慮することによっても，有向グラフの辺を削除することができる．

Algorithm 1 は隣接行列を初期状態以外からの辺および，最終時刻 $t_{k+N}$ に到達不可能な辺を削除するアルゴリズムである．

**Algorithm 1** 初期状態および到達可能性を考慮した有向グラフ生成

1: $x_c := x^f_k$
2: $P_x := x_c \cdot (x_c)^T$
3: **for** $i = k$ to $k + N - 1$ **do**
4: $\quad H'_i := P_x \cdot H_i$
5: $\quad x_c = (H'_i)^T \cdot e_{n_f}$
6: $\quad P_x = \mathrm{and}(x_c \cdot (x_c)^T,\ I_{n_f})$
7: **end for**
8: $x_c := (H'_{k+N-1})^T \cdot e_{n_f}$
9: **for** $i = k + N - 1$ to $k$ **do**
10: $\quad P_x = \mathrm{and}(x_c \cdot (x_c)^T,\ I_{n_f})$
11: $\quad H'_i := H'_i \cdot P_x$
12: $\quad x_c = H'_i \cdot e_{n_f}$
13: **end for**

Algorithm 1 において，and(A, B) は，同じサイズの行列 $A \in \{0,1\}^{n\times m}, B \in \{0,1\}^{n\times m}$ の各要素で論理積をとる関数である．行列 $A, B$ の $(i,j)$ 成分をそれぞれ $a_{i,j}, b_{i,j},\ 1 \le i \le n,\ 1 \le j \le m$ とすると，$C = \mathrm{and}(A,\ B)$ の $(i,j)$ 成分は

$$c_{i,j} = a_{i,j} \wedge b_{i,j}$$

で表される．

Algorithm 1 を，1 行目から 7 行目の初期状態を考慮した辺削除と 8 行目から 13 行目の最終時刻を考慮した辺削除に分けて説明する．

- 初期状態を考慮した辺削除

  ビークルの初期状態 $x_k^d = \Lambda_k x_k^f$ および隣接行列 $H_i$ が与えられているものとする．まず，対角成分 $(x_i^d, x_i^d)$ のみが 1 となる行列 $P_x \in \{0,1\}^{n_f \times n_f}$ を作成する (2 行目)．ここで，$\cdot$ は二つの行列のブール積を表す二項演算子である．例として，$t_{k+1}$ で $x_{k+1}^d \in \{2,5\}$ ならば

$$P_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

  となる．この $P_x$ を用いることで，隣接行列 $H_i$ は $x_i^d$ からの辺のみを残した隣接行列 $H_i'$ に変換される (4 行目)．次に，次時刻 $t_{i+1}$ における状態経由点の候補を表したベクトル $x_c \in \{0,1\}^{n_f}$ を作成する (5 行目)．ベクトル $x_c$ は，状態経由点の要素のみが非ゼロとなり，それ以外の要素はゼロとなるベクトルである．$x_i^d \in \{2,5\}$ であれば

$$x_c = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

  のように，ベクトル $x_c$ の 2 番目と 5 番目の要素が非ゼロとなり，それ以外の要素はゼロとなる．$x_c$ を用いて $P_x$ を更新し，次時刻の隣接行列 $H_i$ も同様に変換する．これを $i = k + N$ まで繰り返すことで，初期状態を考慮した有向グラフを生成する (6 行目)．

- 最終時刻への到達可能性を考慮した辺削除

  次に，最終時刻 $t_{k+N}$ に到達不可能な有向グラフの辺を削除することを考える．まず，ベクトル $x_c$ によって，$t_{k+N}$ における許容領域を表す (8 行目)．初期状態からの辺の削除と同様の方法で，行列 $P_x$ を作成する (11 行目)．$P_x$ を用いて，$t_{k+N}$ へ向かう辺のみを残した隣接行列に変換する (12 行目)．この手順を繰り返し，$t_{k+N-1}, t_{k+N-2}, \ldots, t_k$ のように，最終時刻から初期時刻に向かう辺が残るような隣接行列を作成する．

以上の手順によって，初期状態から最終状態へ到達可能な有向グラフが生成される．Fig. 4.1 の例において，初期位置を $x_k^d = 4$ と仮定すると，そのときの有向グラフは，Fig. 4.2 のようにビークルの初期位置から辺が結ばれたグラフ構造となる．また，$t_k - t_{k+1}$ 間の隣

接行列は

$$H'_k = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

となる．

この結果，最適状態の探索量が減少するため，計算時間を低減することができる．

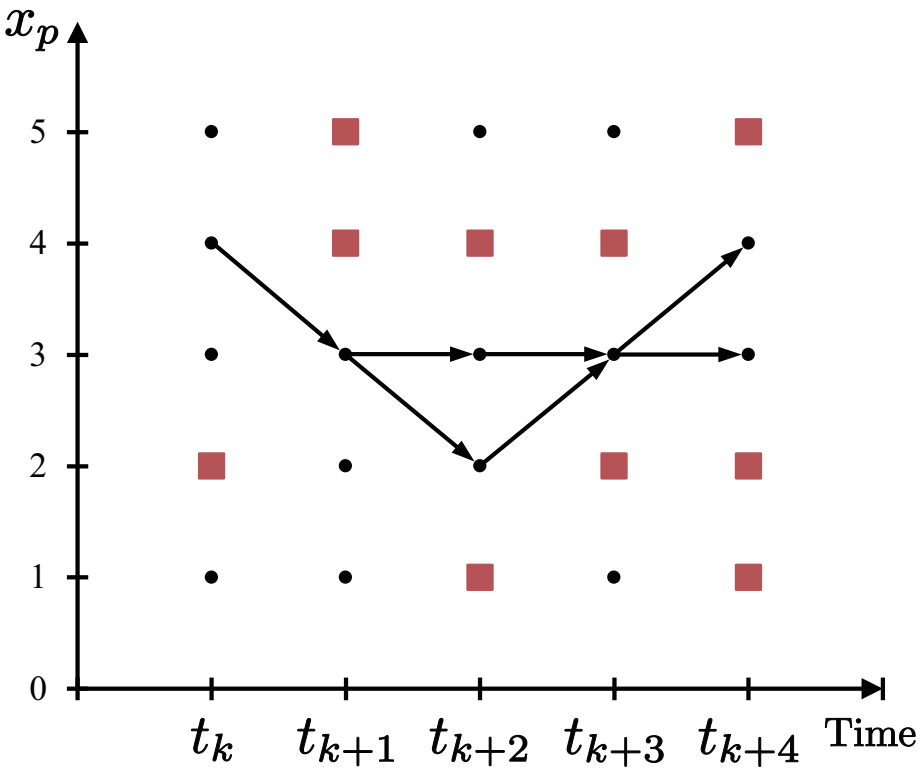



図 4.2: 初期位置からの有向グラフ

## 4.3 ランデブポイントを考慮した有向グラフの辺削除

次に，ランデブポイントを考慮した辺削除について説明する．「時刻 $t_{k+r}$, $(0 \leq r \leq N)$ 以内にランデブを行わなければならない」という拘束条件が与えられているのもとする．このとき，2 台のビークルは $t_{k+r}$ までに同じ位置に到達可能でなければならない．このことを考慮することにより，有向グラフの辺を削除することができる．

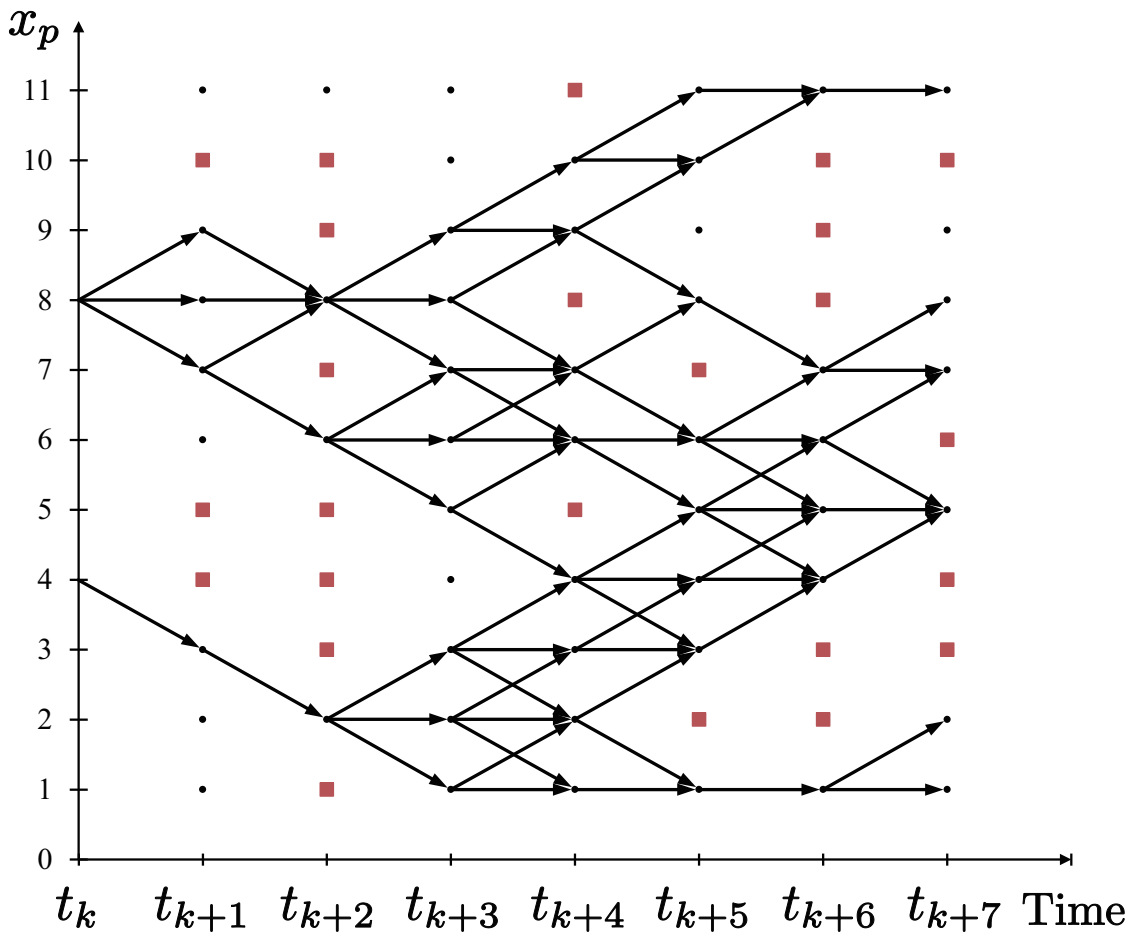



図 4.3: 2 台のビークルの有向グラフ

例として，図 4.3 のような有向グラフを考える．2 台のビークルの有向グラフは，Algorithm 1 によって初期状態を考慮した有向グラフに変換されたものである．図 4.3 において，$r = 5$，すなわち，「$t_{k+5}$ 以内にランデブを行わなければならない」という拘束条件を仮定する．そのときのランデブ候補点は，$x_i^d \in \emptyset,\ i = k, k+1, k+2, k+3,\ x_{k+4}^d \in \{4\},\ x_{k+5}^d \in \{3, 4, 5\}$ となる．これらの内，いずれかのポイントへ到達しなければならない．すなわち，これらのランデブ候補点へ到達しない有向グラフの辺は事前に削除することができる．

Algorithm 2 はランデブ候補点へ向かわない辺を削除するためのアルゴリズムである．

Algorithm 2 について説明する．Algorithm 1 によって生成された隣接行列 $H'_{i,j},\ j \in \{1, 2\}$ が与えられているものとする．Algorithm 2 では，ランデブを行う時刻 $t_{k+r}$ より前の時刻 $t_i < t_{k+r}$ と後の時刻 $t_i \geq t_{k+r}$ に分けて考える．

- $t_i \geq t_{k+r}$

  ランデブを行う時刻 $t_i \geq t_{k+r}$ 以降では，2 台のビークルの持つ有向グラフは同じにな

**Algorithm 2** ランデブポイントを考慮した有向グラフ生成

1: $H'_{i,j},\ j \in \{1,2\}$ を与える
2: **for** $i = k + r$ to $k + N - 1$ **do**
3: $\quad H''_{i,j} := \mathrm{and}(H'_{i,1},\ H'_{i,2})$
4: **end for**
5: $x^r_j := x^d_{k+r,1} \cap x^d_{k+r,2}$
6: **for** $i = k + r - 1$ to $k$ **do**
7: $\quad P_{x,j} = \mathrm{and}(x^r_j \cdot (x^r_j)^T,\ I_{n_f})$
8: $\quad H''_{i,j} := H'_{i,j} \cdot P_{x,j}$
9: $\quad x^r_j = H'_{i+1,j} \cdot e_{n_f}$
10: **end for**

らなければならない．すなわち，$t_{k+r}$ 以降では 2 台のビークルの隣接行列 $H'_{i,1},\ H'_{i,2}$ の共通部分を残した隣接行列が各ビークルもつグラフとなる (3 行目)．

- $t_i < t_{k+r}$

  ランデブを行う以前の時刻 $t_i < t_{k+r}$ では，時刻 $t_{k+r}$ で 2 台のビークルが同時に到達可能な位置 $x^r_j = x^d_{k+r,1} \cap x^d_{k+r,2}$ を抽出する (5 行目)．その後，Algorithm 1 の最終時刻への到達不可能な辺を削除する手順と同様に，抽出した位置 $x^r_j$ に到達不可能な辺を削除していく．

これを $t_k$ まで繰り返すことで，ランデブを考慮した有向グラフ $H''_{i,j}$ が生成される．「$t_{k+5}$ 以内にランデブを行わなければならない」という拘束条件を仮定したとき，Algorithm 2 によって，図 4.3 の有向グラフは図 4.4 のように変換される．

## 4.4 隣接行列作成，Algorithm 1，Algorithm 2の計算時間

隣接行列の作成，Algorithm 1，Algorithm 2 の平均計算時間を図 4.5 に示す．ランダムに生成した障害物配置を 100 パターン用意し，各障害物配置における計算時間の平均値を平均計算時間とした．Algorithm 1 での 2 台のビークルの初期状態は $x_p = 4$ および $x_p = 8$ とし，Algorithm 2 では $t_i = 4$ でランデブを行うものとした．また，実行環境は，Inter Core 2 Duo Processor 3.00GHz，メモリ 4.00GB の PC を使用し，各アルゴリズムは MathWorks MATLAB(R2007b) にてプログラムを作成した．

横軸は予測ステップ数であり，いずれの計算時間も予測ステップ数に比例して増加している．特に，Algorithm 1 の予測ステップ数に対する増加率が大きいことがわかる．しかしながら，サンプリング時間を 1.0[sec] と仮定すると，隣接行列作成から Algorithm 1 および Algorithm 2 を実行した際の合計平均計算時間は $N = 5, 6, \ldots 15$ でサンプリング時間の 0.07%〜0.17%程度である．MIQP 問題を解く際の計算時間と比較して非常に小さい値であることから，制御性能に大きな影響を与えるものではないと判断できる．

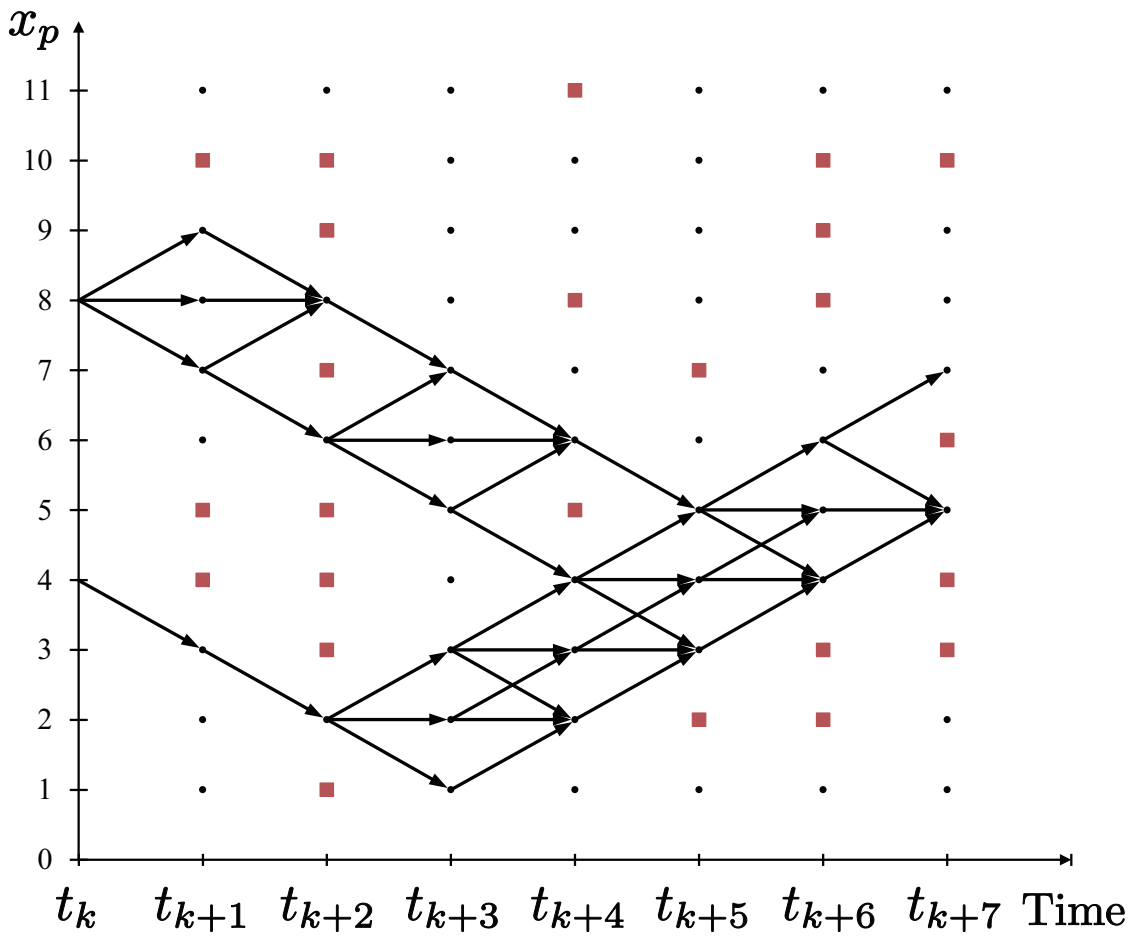



図 4.4: ランデブポイントを考慮した有向グラフ

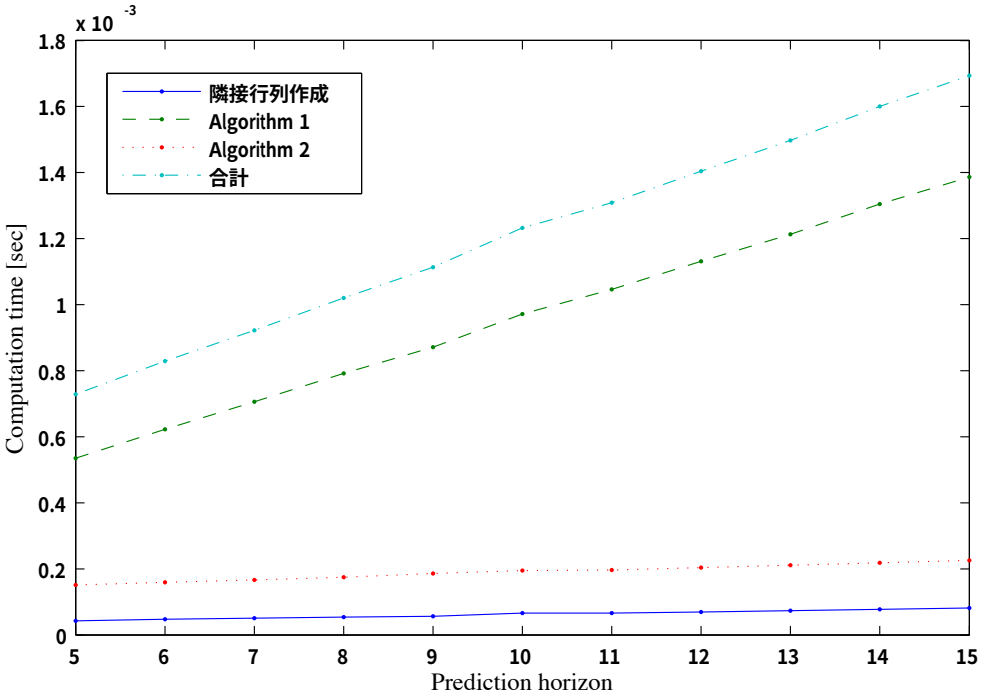



図 4.5: 有向グラフに関する計算時間

# 第5章　計算例

本章では3章で定式化したマルチビークルシステムに対するいくつかの計算例を示す．ビークル同士の協調動作条件として3.2節で示した，時相論理による拘束を持つ協調動作を考える．また，それぞれの協調動作条件に対して，4章で示した有向グラフの辺削除により計算時間が低減されることを示す．

## 5.1　制御対象

対象とする制御対象について説明する．使用したPCは4.4節と同様で，ソルバはIBM ILOG CPLEX v11.0.0を使用した．ビークルのダイナミクスは

$$\ddot{x}_p = -a\dot{x}_p + x_d, \quad \dot{x}_d = -bx_d + u$$

のように定めた．上式より(2.1)式の$A$および$B$は次のように表される．

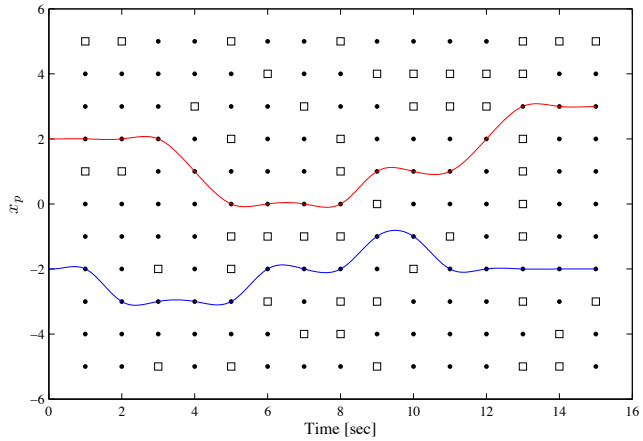
$$A = \begin{bmatrix} -b & 0 & 0 \\ 1 & -a & 0 \\ 0 & 1 & 0 \end{bmatrix}, \ B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$x$はすでに述べたように$x := [\ x_d\ \dot{x}_p\ x_p\ ]^T$で表され，それぞれ速度，駆動力，位置を表し，$u$は制御入力である．状態制約および制御入力制約はそれぞれ，$-10 \leq x_d(t_i) \leq 10,\ -10 \leq \dot{x}_p(t_i) \leq 10,\ -40 \leq u(t_i) \leq 40$と定めた．また，評価関数は$Q = I_3,\ R = 1$，とし，予測ステップ数は$N = 15$，$a = 3,\ b = 5$とした．

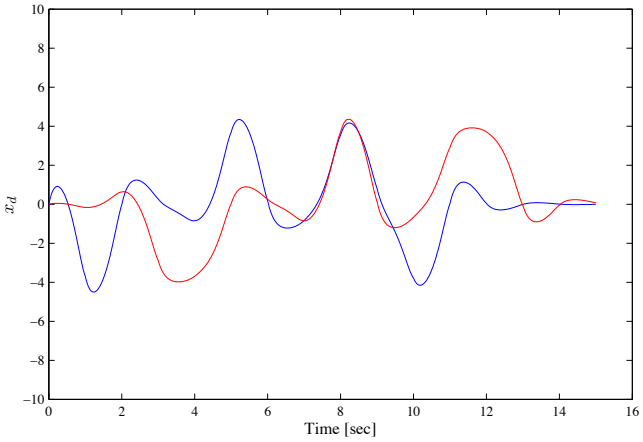
## 5.2　衝突回避に関する計算例

問題2の衝突回避についての計算例を示す．計算結果を図5.1に示す．有向グラフの辺削除には，Algorithm 1を適用している．図5.1aがビークルの位置の時間的変化を表し，図5.1b，図5.1cはそれぞれ，ビークルの駆動力と速度である．

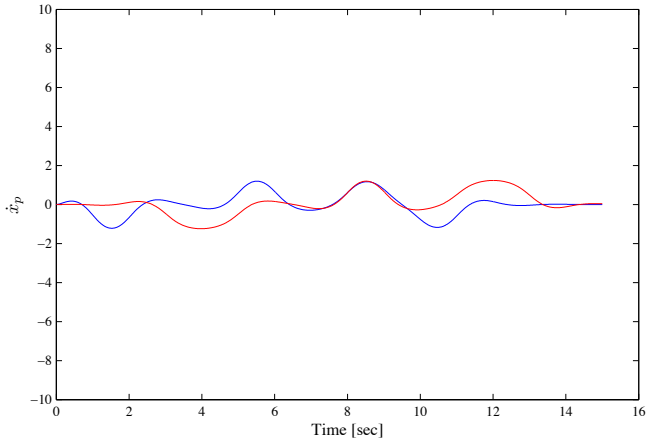
結果を見ると，拘束条件(3.4)に従い，2台のビークルが同時刻で同じ位置に到達していないことがわかる．この計算例での計算時間は5.7542 [sec]である．また，ランダムな障害物配置で100個の問題を解き，その際の平均計算時間を求めた．平均計算時間は10.2259 [sec]，標準偏差は7.5537 [sec]となった．なお，平均計算時間は外れ値を除くために，結果の上下10%ずつを除いた10%調整平均とし，障害物配置は可解性のあるものに限った．




(a) $x_p$




(b) $x_d$




(c) $\dot{x}_p$

図 5.1: 衝突回避

## 5.3 ランデブに関する計算例

問題 3 において，10 ステップ以内に一度だけランデブをするという拘束条件

$$\phi = \diamond_{10}^{1}\varphi \wedge \circ_{11}\Box\neg\varphi \tag{5.1}$$

を考えた場合の計算例を示す．実行環境や実行条件は前節と同じとした．また，有向グラフの辺削除には Algorithm 1 および Algorithm 2 を適用した．結果のグラフを図 5.2 に示す.

図 5.2 の結果では，$t_i = 6$ でランデブが行われた．計算時間は 20.1009 [sec] で，平均計算時間は 22.4326 [sec]，そのときの標準偏差は 25.0683 [sec] であった．5.2 節の衝突回避と比較すると，平均計算時間が 2 倍以上かかっている．この原因として考えられるのは，協調動作条件が衝突回避に比べると複雑であり，変換した線形不等式の本数や，変換する際の 0-1 変数の数が多いことが挙げられ，このことが計算時間の増加に影響していると考えられる．また，協調動作条件 (5.1) では，ランデブ候補点が数多く存在するため，最適なランデブポイントの探索量が多いことなども原因の一つと考えられる．

次に,「現在時刻 $t_k$ において $t_{k+4}$ から $t_{k+6}$ の間に一度だけランデブをする」という拘束条件を考える．すなわち，問題 4 において，協調動作条件 (3.10) が $r_1 = 4,\ r_2 = 6$ のときの条件

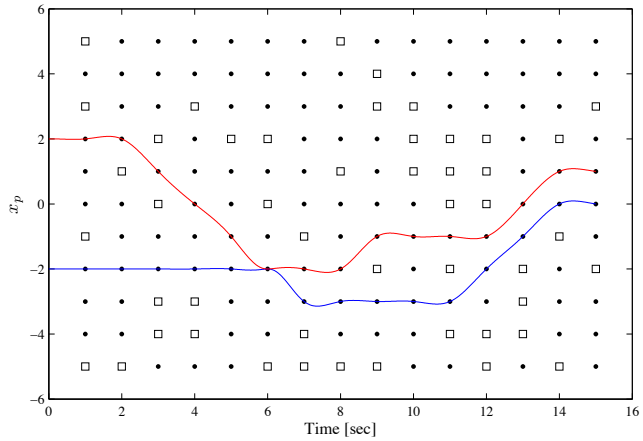
$$\phi = \neg \diamond_3 \varphi \wedge \diamond_6^1\varphi \wedge \circ_7\Box\neg\varphi \tag{5.2}$$

である．計算結果を図 5.3 に示す．実行環境および実行条件は 5.2 節と同じとし，有向グラフの辺削除には Algorithm 1 および Algorithm 2 を適用した．図 5.3 を見ると，$t_i = 6$ でランデブが起こっている．本計算例での計算時間は，9.2475 [sec] であり，平均計算時間は 9.0222 [sec]，また標準偏差は 7.5835 [sec] であった.

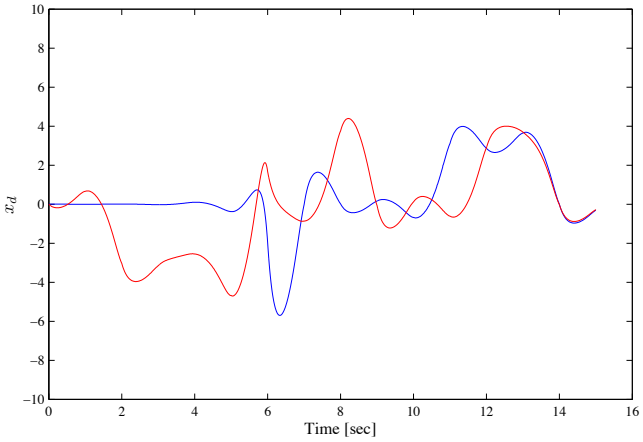
本計算例と協調動作条件 (5.1) の平均計算時間および標準偏差を表 5.1 に示す．協調動作条件 (5.1) の計算時間と比較すると，平均計算時間が約 13 秒短くなった．これは，協調動作条件 (5.1) が,「$t_k \le t_i \le t_{k+10}$ の時間区間内に一度だけランデブをする」だったのに対し，協調動作条件 (5.2) は,「$t_{k+4} \le t_i \le t_{k+6}$ の時間区間内で一度だけランデブをする」であり，より短い区間でのランデブ条件であったことが原因として挙げられる．ランデブを行う区間が短くなるということは，ランデブ候補点が少なくなるということである．よって，最適なランデブポイントの探索量が少なくて済むことになり，計算時間が短くなったと考えられる.

表 5.1: 各協調動作条件の平均計算時間

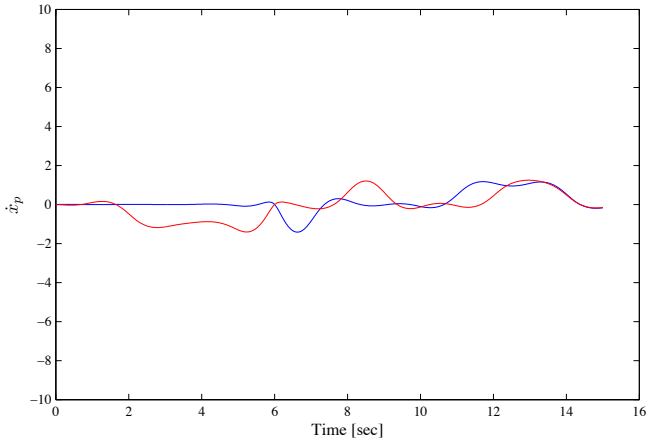
| 協調動作条件 | 10 ステップ以内 | 4-6 ステップ |
|---|---|---|
| 平均計算時間 [sec] | 22.4326 | 9.0222 |
| 標準偏差 [sec] | 25.0683 | 7.5835 |




(a) $x_p$

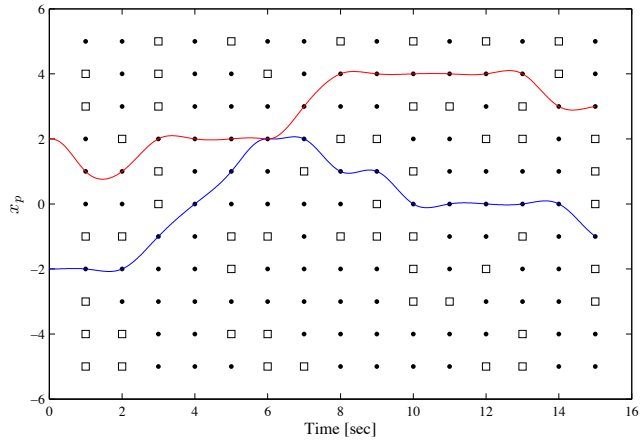



(b) $x_d$

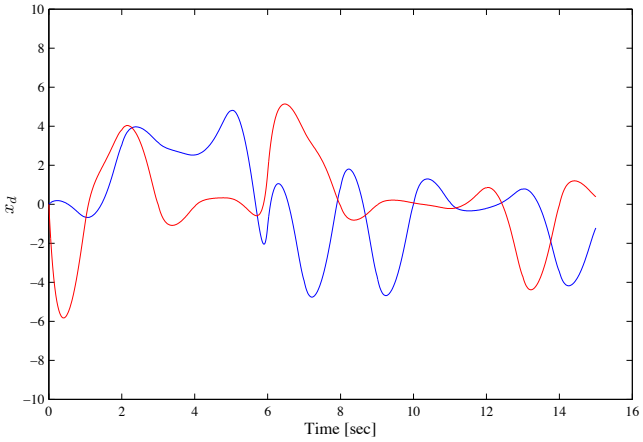



(c) $\dot{x}_p$

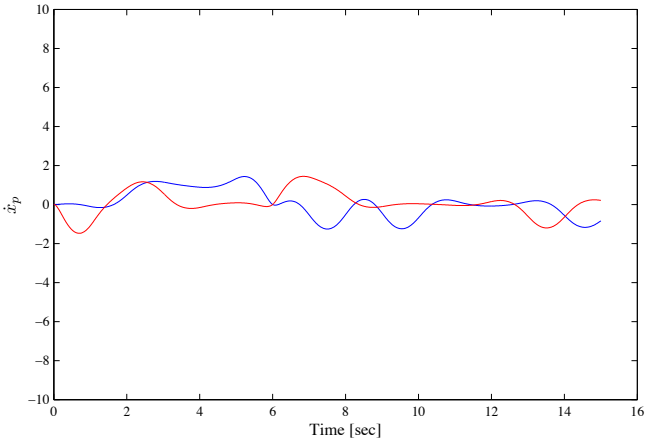
図 5.2: 10 ステップ以内にランデブ




(a) $x_p$




(b) $x_d$




(c) $\dot{x}_p$

図 5.3: 4-6 ステップでランデブ

## 5.4　計算時間に関する考察

### 5.4.1　予測ステップ数とビークル台数による計算時間の比較

ビークル間の協調動作条件として衝突回避を与えた際の，予測ステップ数 $N$ によるビークル台数別の平均計算時間の変化を表したグラフを図 5.4 に，$N = 5, 10, 15$ の計算時間を表 5.2 に示す．ビークルの台数が 1 台の場合と，2 台で 4.2 節で示した Algorithm 1 を適用した場合，しなかった場合の結果である．いずれの結果も予測ステップ数 $N$ に比例して指数関数的に増加していることがわかる．また，ビークルの台数に従って決定変数の数が 2 倍ずつ増加していくため，1 台，2 台,... とビークルの台数を増やしていくことによっても指数オーダーで計算時間が増加することが予想される．図 5.4 によってもこのことが示唆されている．

しかし，有向グラフの辺削除を行った場合と行わなかった場合を比較すると，削除した場合では計算時間が $N = 15$ では約 7 秒短縮できるなど，計算時間の低減に有効であることが示された．辺削除を行うことによって，より大きな予測ステップ数を取ることが可能となる．モデル予測制御などを用いる際には，サンプリング時間以下の計算時間が必要である．このとき，予測ステップ数を大きく取れることにより，制御性能の向上が期待できる．

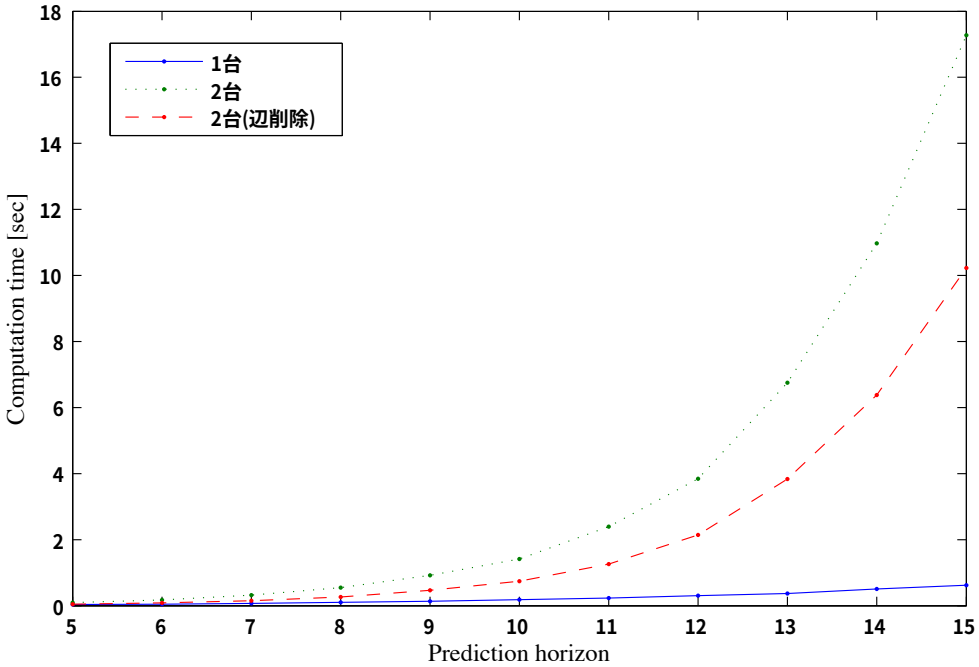



図 5.4: ビークル台数と予測ステップ数による平均計算時間

表 5.2: ビークル台数と予測ステップ数による平均計算時間

| | ビークル台数 | | |
|---|---|---|---|
| $N$ | 1 台 [sec] | 2 台 [sec] | 2 台 (辺削除) [sec] |
| 5 | 0.0378 | 0.0964 | 0.0486 |
| 10 | 0.1832 | 1.4157 | 0.7414 |
| 15 | 0.6220 | 17.2734 | 10.2259 |

### 5.4.2 各辺削除アルゴリズムによる計算時間の比較

次に，各辺削除アルゴリズムを適用した場合の各協調動作条件における計算時間を比較する．平均計算時間を比較したグラフを図 5.5 に，また平均計算時間を表 5.3 に示す．

グラフを見ると，Algorithm 1 を適用することによる効果が，すべての協調動作条件で最も大きいことがわかる．Algorithm 1 を適用することで，3〜43 秒程度の計算時間の低減が達成できた．特に，協調動作条件 (5.1) においては，計算時間が 50%以上低減された．これは，初期時刻を含む $t_k \leq t_i \leq t_{k+10}$ の時間区間内でランデブを行うため，初期状態を考慮した辺削除によって，ランデブ候補点が大幅に減少したことが要因として考えられる．すなわち，ランデブを行う時間区間が初期時刻に近いほど Algorithm 1 による計算時間低減の効果が大きいと考えられる．

また，Algorithm 2 によって，協調動作条件 (5.1) および (5.2) の計算時間は，Algorithm 1 のみの場合よりもそれぞれ 2 秒および 2.7 秒程度短くなった．Algorithm 2 では，協調動作条件 (5.2) に対する効果が大きいことがわかった．協調動作条件 (5.1) の $t_k \leq t_i \leq t_{k+10}$ の時間区間内でのランデブ条件では，時刻 $t_{k+10}$ における 2 台のビークルの共通部分

$$x_j^r := x_{k+10,1}^d \cap x_{k+10,2}^d$$

が，協調動作条件 (5.2) の $t_{k+4} \leq t_i \leq t_{k+6}$ の時間区間内でのランデブ条件より多くなる．これによって，各ビークルの有向グラフ $\mathcal{E}_{i,j}(\cdot)$ の $x_j^r$ に向かう辺が多くなるため，Algorithm 2 によって削除される辺が少なくなる．このことが，協調動作条件 (5.2) に対して，Algorithm 2 の効果がより大きかったことの要因と考えられる．すなわち，ランデブを行う時間区間が初期時刻より遠くなると，Algorithm 2 による辺削除の効果が小さくなることを意味する．

表 5.3: ビークル台数による平均計算時間

| 辺削除の条件 | 削除なし [sec] | Algorithm 1 [sec] | Algorithm 2 [sec] |
|---|---|---|---|
| 衝突回避 | 17.2734 | 14.3359 | - |
| 10 ステップ以内 | 67.2556 | 24.4989 | 22.4326 |
| 4-6 ステップ | 25.7686 | 11.7827 | 9.0222 |

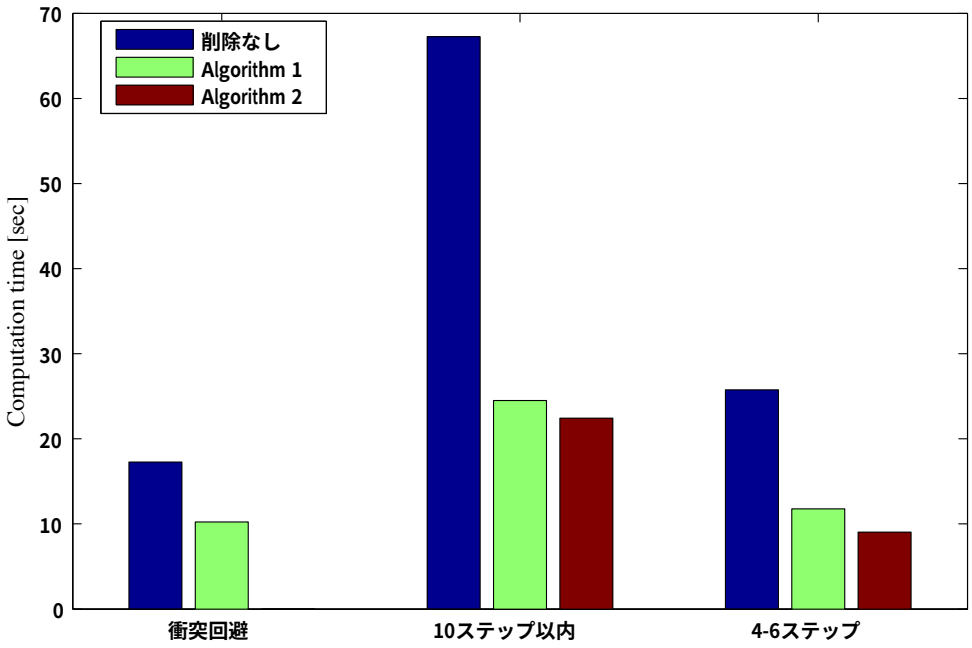



図 5.5: 辺削除による平均計算時間の比較

# 第6章　おわりに

本論文では，状態の離散化を用いたマルチビークルシステムの最適制御を考えた．複雑な拘束をもつ状態の要素のみを離散化することで，MIQP 問題に帰着される．また，ビークル同士の協調動作条件をより一般的に線形時相論理によって記述した．線形時相論理によって記述することで，ランデブという時間的な拘束をもつ協調動作を記述することができた．また，線形時相論理による拘束を含む問題を MIQP 問題に帰着させるために，線形不等式への変換手順について示した．

さらに，ビークルの離散ダイナミクスを表す有向グラフの不要な辺削除を行った．これに，ビークルの初期位置を考慮した辺削除と，ランデブの拘束条件を考慮した辺削除の 2 つのアルゴリズムを提案した．そして，いくつかの計算例を示すことにより，提案したアルゴリズムで計算時間が低減されることを示した．

今後の課題としては，ビークル数の増加に伴う計算時間の増加がある．ビークル数に比例して指数関数的に計算時間が増加することが予想される．複数のビークルの問題を同時に解く現在の手法では実用可能な計算時間を達成することは困難である．したがって，ビークル毎に計算を分散化するなど，計算時間の低減を図ることが重要な課題である．

# 謝辞

本研究を進めるにあたり，指導教員である平石邦彦教授には終始熱心にご指導，ご支援頂きました．ここに深い感謝の意を表します．研究に関して数多くのご助言を頂きました小林孝一助教，崔舜星特任助教に感謝致します．また大学院での生活を支えて頂いた平石研究室の皆様に感謝申し上げます．

# 参考文献


[1] P. Abichandani, H. Y. Benson and M. Kam: Decentralized multi-vehicle path coordination under communication constraints, *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2306/2313（2011）

[2] T. M. Cavalier, P. M. Pardalos and A. L. Soyster: Modeling and integer programming techniques applied to propositional calculus, *Computer & Operations Research*, vol. 17, no. 6, 561/570 (1990)

[3] M. Franceschelli, D. Rosa, C. Seatzu and F. Bullo: A gossip algorithm for heterogeneous multi-vehicle routing problems, *Proc. 4th IFAC Conf. on Analysis and Design of Hybrid Systems*, 325/332（2012）

[4] H. L. Hagenaars, J. Imura and H. Nijmeijer: Approximate continuous-time optimal control in obstacle avoidance by time/space discretization of non-convex state constraints, *Proc. of the IEEE Conf. on Control Application*, 878/883（2004）

[5] J. Imura and H. Matsushima: Simultaneous optimization of continuous control inputs and discrete state waypoints, *Proc. 9th Int'l Workshop on Hybrid Systems: Computation and Control*, *LNCS* 3927, 302/317（2006）

[6] S. Karaman, R. G. Sanfelice and E. Frazzoli: Optimal control of mixed logical dynamical systems with linear temporal logic specifications, *Proc. of the 47th IEEE Conf. on Decision and Control*, 2117/2122（2008）

[7] S. Karaman and E. Frazzoli: Linear temporal logic vehicle routing with applications to multi-UAV mission planning, *Int'l Journal of Robust and Nonlinear Control*, vol. 21, no. 12, 1372/1395（2011）

[8] K. Kobayashi and J. Imura: Model predictive control of directed-graph type hybrid systems, *Proc. of the 46th IEEE Conf. on Decision and Control*, 3196/3201（2007）

[9] K. Kobayashi and J. Imura: Deterministic finite automata representation for model predictive control of hybrid systems, *Journal of Process Control*, vol. 22, no. 9, 1670/1680（2012）

[10] Y. Kuwata, A. Richards, T. Schouwenaars and J.P. How: Distributed robust receding horizon control for multivehicle guidance, *IEEE Trans. on Control Systems Technology*, vol. 15, no. 4 627/641（2007）

[11] A. Rantzer and M. Johansson: Piecewise linear quadratic optimal control, *IEEE Trans. on Automatic Control*, vol. 45, no. 4, 629/637（2000）

[12] W. Ren and E Atkins: Distributed multi-vehicle coordinated control via local information exchange, *Int'l Journal of Robust and Nonlinear Control*, vol. 17, no. 10-11, 1002/1033 （2007）

[13] Y. Tazaki and J. Imura: Discrete abstractions of nonlinear systems based on error propagation analysis, *IEEE Trans. on Automatic Control*, vol. 57, no. 3, 550/564（2012）

[14] 井村順一：ハイブリッドシステムのリアルタイム最適制御，計測と制御，vol. 44, no. 7，464/470（2005）

[15] 小林孝一，井村順一：モデル予測制御における有限オートマトンの時系列ベーストリアルタイムモデリング，計測自動制御学会第 7 回制御部門大会資料，73-1-4（2007）

[16] 小林孝一：ハイブリッドシステムにおける離散ダイナミクスのモデリング，システム/制御/情報，vol. 55, no. 3, 75/81（2011）

# 本研究に関する発表論文

### 国内会議

[1] 永見琢朗，小林孝一，平石邦彦：状態離散化を用いたマルチエージェントシステムの最適制御，第 55 回自動制御連合講演会講演論文集，526/529（2012）

[2] 永見琢朗，小林孝一，平石邦彦：時相論理による拘束をもつマルチビークルシステムの最適制御，計測自動制御学会第 52 回離散事象システム研究会講演論文集，21/26（2013）