

## 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

大川 功\*・野中 謙一郎\*\*

Model Predictive Control for Wheel Position of Leg/Wheel Robots considering Passage Width

Isao OKAWA\* and Kenichiro NONAKA\*\*

This paper describes a model predictive control (MPC) application which optimizes wheel positions of a mobile robot with geometry-variable wheels. Each wheel should be set apart from the others to keep enough stability for safety operation of mobile robots. However, this requirement tends to decrease the mobility. In this paper, we propose a method which optimizes wheel positions with respect to passage width using MPC. The proposed method realizes a good balance between stability and mobility by online optimization of wheel positions during automatic collision-free travel. Moreover, we reduced the computational cost of MPC by introduction of a control method for wheel positions with respect to travel distance. Thus it is capable to implement the control system into low speed computers, like embedded CPUs. We conducted experiments to verify the effectiveness; it is confirmed that the proposed method is useful to operate the mobile robot with enough stability and mobility.

**Key Words:** Mobile robot, Optimal control, Wheel position, Model predictive control, Embedded CPU

## 1. はじめに

車輪による移動方式は移動時のエネルギー効率が高く、自動車などの輸送機器や移動ロボットに広く採用されている。これらの車両は人や荷物の高速な輸送や各種のマニピュレータによる作業などをそれぞれ目的とする場合が多く、安定性を確保するためにホイールベース・トレッドを十分に広くする必要がある。この結果として、車体が大型化し狭隘な空間での移動・作業が困難となるといった問題が生じる。この問題は自動車などにおける車体の小型化に対する要求の高まりや、移動ロボットの高機能化に向けた搭載機器の増加に伴う高重心化によって顕著となることが予想され、何らかの対策を講じる必要がある。

この問題に対する一つの解決策として、これまでに車輪位置を可変とすることで状況に応じた走行特性を得る方法が提案されてきた。例を挙げると、先端に能動車輪を持つ2リンク脚を変形させることで車体安定化制御を行う研究<sup>1)</sup>では、車両の加速度に応じて車輪位置を変化させることで車体の安定化を行っている。また車両の幅を変更可能な全方向移動車両を車椅子に応用した研究<sup>2)</sup>では、重心の移動に応じてホイールベース・トレッドの比を変更することで転倒の危険性を低減させている。さらに脚と能動車輪から構成されるホイールモジュールを複数搭載することで車

輪位置の変更や脚による歩行を可能とした研究<sup>3)</sup>も報告されている。しかし車体安定性の更なる向上のためには現在時刻における加速度や重心位置などの静的な指標に基づく制御に加え、予測に基づいた制御を行い車輪位置を滑らかに変化させる必要がある。

そこでわれわれはこれまでに車輪位置を任意に変更可能な車両型ロボットを用いて、通路幅の予測に基づく車輪位置の最適化制御をオンラインで行う手法を提案してきた。提案手法は機体が十分に広い通路を走行する際は車輪間隔を可能な限り広げることで走行時の安定性を高め、障害物付近を走行する際や狭隘な通路では車輪間隔を狭めることで機動性を確保し、安定性と機動性の両立を図る。過去の研究<sup>4)</sup>では機体のトレッドを通路幅の予測に基づき数値的に最適化する手法を提案し、一定の安定性を維持しながら隘路を通過させることに成功した。しかし最大限の機体安定性を得るためには機体から離れた位置に置くべき車輪も機体側へ引き寄せられており、さらにデスクトップPCを計算に用いても制御周期は10 Hz程度と計算コストにも課題を残していた。

本論文では制御対象を各車輪の位置として独立に制御し、さらに車輪位置の最適化を計算コストを低減させたモデル予測制御によって行うことで、より高いレベルでの安定性と機動性の両立を実現する手法を示す。モデル予測制御については比較的計算コストの低い方法としてC/GMRESアルゴリズム<sup>5)</sup>や入力的时间微分を新たな入力として再定義する手法<sup>6)</sup>に加え、速度と精度を向上させた修正シューティング法<sup>7)</sup>など多くの手法が報告されているが、デスクトップPC程度の処理能力を有す計算機を必要とすることが多い。これに対してわれわれは脚の逆運動学計算をモデル予測制御から分離し、道程に対する車輪位置の変化を制御入力

\* 元 東京都市大学大学院工学研究科 世田谷区玉堤 1-28-1

\*\* 東京都市大学工学部 世田谷区玉堤 1-28-1

\* Graduate School of Engineering, Tokyo City University

\*\* Faculty of Engineering, Tokyo City University

(Email: knonaka@tcu.ac.jp)

(Received November 30, 2010)

とする手法を導入することで計算コストを大きく低減させ、組み込み CPU のみを制御に用いたオンライン計算の実装に成功した。

## 2. システム構成

### 2.1 実験機

本研究では Fig. 1 に示す、脚車輪型ロボット<sup>9),10)</sup>を実験機として使用している。Fig. 1 (a), (b) はそれぞれ機体全体の鳥瞰図、右前の脚部分を示す。本機体は先端に DC モータによって駆動する能動車輪を有する、3 リンクスカラ型の脚を 4 本持つ。この脚を変形させることで、各車輪を水平面上の可動範囲内で任意に配置しての走行や全方向移動が可能となっている。また本研究では車輪が常に機体の外郭に配置されるように制御を行うため、車輪位置の変化に伴い機体の縦横方向のサイズが変化する。具体的なサイズやその他のパラメータは Table 1 の通りである。なお脚関節のアクチュエータとして角度指示型の RC サーボモータを使用し、運動学に基づいて関節角度の制御を行っているため、詳細な質量や慣性モーメントの値は制御に必要としない。

制御用の演算装置としてはルネサスエレクトロニクス社製の組み込み CPU の SH7058F を搭載しており、動作周波数は 80 MHz、内蔵メモリは 1024 kB である。これは自動車の燃料噴射制御に用いられているものと同型であり、頑強で信頼性が高いがオンラインでの障害物回避経路生成やモデル予測制御を実装するためには計算コストの低減が必要である。本研究では提案手法の計算コストの低さを示すために、この CPU を 1 台のみ使用して提案制御手法を実現する。センサ類としてはレーザー測域センサ 2 台（北陽電機社製、URG-04LX-UG01）とセンサのドライブ用小型コンピュータで構成されるセンシングユニットを搭載しており、自己位置・姿勢の推定と障害物の検出が可能である。ただしセンサドライブ用小型コンピュータはセンシングの用途のみに使用し、制御には用いていない。

### 2.2 制御システムの概要

提案手法は最適な回避経路の生成と追従、モデル予測制御による通路幅に対して最適な参照車輪位置の計算、そして参照車輪位置を達成しながら回避経路の追従を可能とする最適な関節角度の計算という、3 つの最適化問題を階層的に行う。これらを一つの最適化問題として扱うことは計算コストの観点から困難な

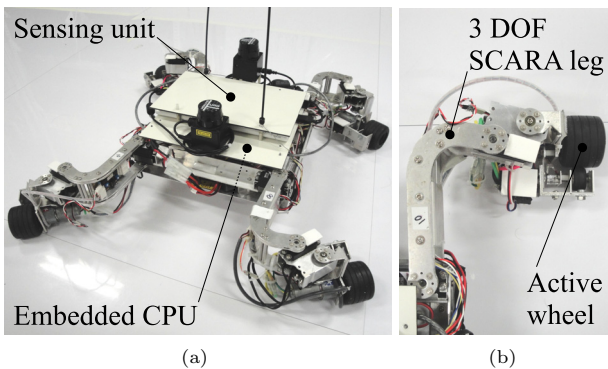


Fig. 1 Experimental robot

め、本研究では 3 つの最適化問題について独立に最適性を考える。Fig. 2 は制御システムのフローチャートである。まずセンシングユニットによって自己位置・姿勢の推定と共に障害物の検出とポリゴンへの近似を行い、この結果を組み込み CPU へ送信する。障害物の検出は回避動作を必要とする機体周辺の障害物のみに対し、測域センサによって得られた走査データを基に行う。測域センサは 240° の範囲に対して最長 5 m までの距離を 100 ms の周期で走査可能であり、これを 2 台使用して全方向の障害物を検出する。つぎに組み込み CPU において、障害物の配置に対して最適な回避経路の生成を行い<sup>8)</sup>、この結果に基づいて経路を追従可能な機体速度、角速度と通路幅を計算する。なお実験機の姿勢角は任意に指定することが可能であるが、本論文では姿勢角が常に回避軌道の接線方向を向くように角速度を制御している<sup>9)</sup>。続いて通路幅に対して最適な参照車輪位置をモデル予測制御によって計算し、経路を追従可能な各車輪の速度と参照車輪位置を実現する最適な脚の関節角度を逆運動学計算によって決定する<sup>10)</sup>。そしてサーボモータへ関節角度を送信し、DC モータへは各車輪の速度から計算したトルクの値をそれぞれ送信する。このように制御に関する計算はすべて組み込み CPU において計算される。本論文で扱う内容は障害物回避経路の生成と通路幅の計算、そして参照車輪位置のモデル予測制御による最適化とする。その他の要素については著者らの過去の研究<sup>8)~10)</sup>を参照していただきたい。

後者の参照車輪位置の最適化においては、通路幅に対して最適な関節角度を直接求めることも考えられるが非線形システムの最適制御問題を解くことになり、さらに車輪位置の変更には一定の時間を必要とするため入力に関する拘束条件を考慮する必要があ

Table 1 Specification of experimental robot

Parameter	Value
Length, Width, Height	500~700, 290~700, 230 mm
Tire diameter	57 mm
Weight	7.7 kg
Max. longitudinal velocity	1.0 m/s
Max. lateral velocity	1.0 m/s
Max. angular velocity	3.0 rad/s

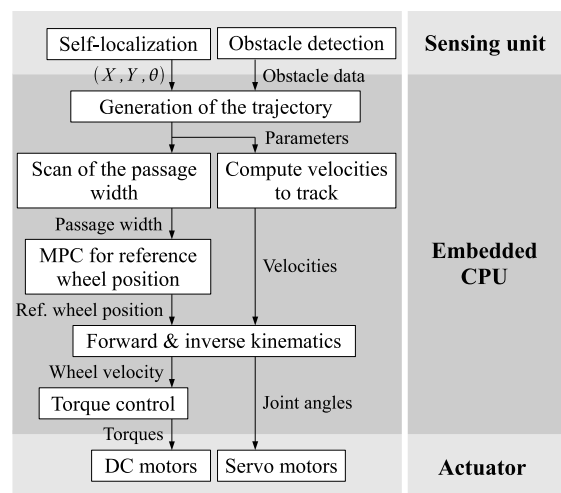


Fig. 2 Flowchart of the control system

る．そのため既存の数値解法を用いて回避経路の生成と共にオンライン計算を行うことは困難であると考えられる．そこで本研究では計算コストを低減するために、通路幅に対して最適な関節角度を求める問題を、通路幅に対する最適な参照車輪位置をモデル予測制御によって求める問題と、参照車輪位置に対して最適な関節角度を逆運動学に基づいて数值的に計算する問題に分離させて扱う．モデル予測制御は後述する道程に対する車輪位置の変化を制御入力とする手法の導入によって線形システムの2次形式評価関数に対する拘束条件なしの最適制御問題に変換することが可能であり、繰り返し計算を行わずに最適解を得ることが可能となる．また逆運動学計算は操舵角に関する拘束や複数の解が存在するなどの特徴から数值的に行う必要があるが計算コストの低減が行われており、組み込みCPUのような小規模な演算装置においても十分高速に計算可能である．

### 3. 座標変数の設定

Fig. 3 に以下の議論で使用する座標系と変数を示す．まず Fig. 3(a) において、 $X$ - $Y$  座標は慣性座標系であり、 $x$ - $y$  座標はロボット固定座標系を表す．また慣性座標系に対するロボット固定座標系の回転角を  $\theta$  とし、ロボットの慣性座標を  $(X, Y, \theta)$  と表す． $(x^{ij}, y^{ij})$  は車輪のロボット固定座標であり、右肩の添え字は各車輪について、00:左前, 01:右前, 10:左後, 11:右後を表す．つぎに  $\tilde{x}$ - $\tilde{y}$  座標は原点が常に機体中心にあり、方向が慣性座標と一致する座標系である．本研究では、たとえば廊下を一方方向に走行する場合のように、慣性座標の少なくとも一つが単調増加する状況を想定している．この想定の下では障害物回避経路を曲線  $\tilde{y}(\tilde{x})$  として  $\tilde{x}$ - $\tilde{y}$  座標上で生成することで計算が簡単化される．続いて Fig. 3(b) において、 $s^{ij}$  軸は各車輪の中心を原点とする障害物回避経路に沿った座標軸であり、4つの車輪についてそれぞれ設定する．また  $y_s(\tilde{x})$  は任意の  $\tilde{x}$  座標において障害物回避経路に直交する座標である．

### 4. 障害物回避経路の生成

#### 4.1 多項式経路の導入

障害物回避を伴う移動ロボットの自律移動経路としては障害物との衝突を避けることに加え、障害物が存在しない場合に本来の

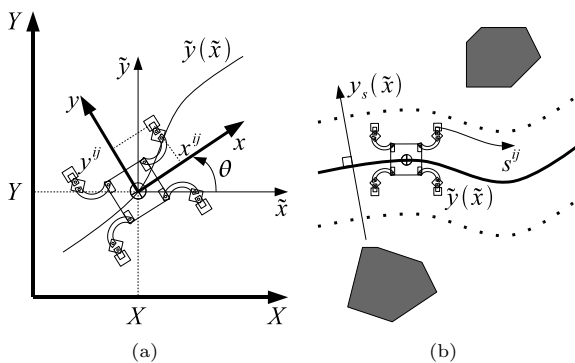


Fig. 3 Coordinates and variables

望ましい経路へ復帰することや、急激な変化が無く滑らかな経路であることが望まれる．これらの要求に対しては、たとえば人工ポテンシャル場法<sup>11)</sup>のような既存の方法によっても達成可能である．しかし本研究では回避経路の法線方向に対して通路幅の測定を行うため、通路幅の測定範囲において区分的に滑らかな関数で回避経路を表現する必要がある．また通路幅の測定範囲を最適な参照車輪位置を計算する際の評価区間以上とする必要から、回避経路も機体中心から一定の範囲における障害物の配置を考慮して決定する必要がある．このような経路を既存の手法によってオンラインで生成することは難しいと考えられる．そこで本研究では回避経路を多項式曲線として表現し、各項の係数を障害物の配置に対して最適な回避経路となるように数値計算によって決定する．本論文では3次多項式関数を回避経路として導入する．

$$\tilde{y}(\tilde{x}) = k_3 \tilde{x}^3 + k_2 \tilde{x}^2 + k_1 \tilde{x} + k_0 \quad (1)$$

最適化の対象は  $k_3 \sim k_0$  である．多項式の次数は任意に設定することが可能であるが、計算コストの観点から3次とした．多項式曲線は少ないパラメータで滑らかな経路の表現が可能で、3次以上の多項式は障害物回避と目標経路への復帰が可能なので、(1)の係数を最適化すれば上記の要求は全て満たされる．

また経路の更新は毎回の制御周期ごとに行われ、更新の際に前回の結果と連続した経路を生成するためには  $k_1$  と  $k_0$  が以下の境界条件を満たす必要がある．なお  $\tilde{y}_p(\tilde{x})$  は前回生成した経路であり、 $\Delta X$  は経路を更新するまでの機体  $X$  座標の変化量である．

$$k_1 = \left. \frac{d\tilde{y}_p(\tilde{x})}{d\tilde{x}} \right|_{\tilde{x}=\Delta X} \quad (2)$$

$$k_0 = \tilde{y}_p(\Delta X) \quad (3)$$

このように低次の2項については解析的に係数が決定されるため、数値計算の対象は  $k_3$  と  $k_2$  の2つとなる．なお経路を慣性座標系における曲線として生成することも可能であるが、(2)、(3)式に対応する計算が複雑化する．

#### 4.2 パラメータの数値最適化

$k_3$  と  $k_2$  の最適化は評価関数を最小化する値を探索することで行う．評価区間を  $\tilde{x} \in [0, \tilde{x}_h]$  として  $N_{\tilde{x}}$  個の評価点に離散化し、 $\tilde{x}^k = \tilde{x}_h \cdot k / N_{\tilde{x}}$  ( $k = 1, 2, \dots, N_{\tilde{x}}$ ) を  $k$  番目の評価点における  $\tilde{x}$  座標の値とする．また  $\tilde{y}_d(\tilde{x})$  を機体の周囲に障害物が存在しない場合に最適な経路となる基準経路とし、 $d^n(\tilde{x})$  ( $n = 1, 2, \dots$ ) を Fig. 4 に示す  $n$  番目の障害物までの最短距離<sup>4)</sup>として、 $k_3$ 、 $k_2$  をつぎのように更新する．

$$J_1 = (\tilde{y}(\tilde{x}^k) - \tilde{y}_d(\tilde{x}^k))^2 \quad (4)$$

$$J_2 = \left( \frac{d\tilde{y}(\tilde{x})}{d\tilde{x}} \right)^2 \Bigg|_{\tilde{x}=\tilde{x}^k} \quad (5)$$

$$J_3 = \sum_{n=1}^{N_o} \frac{1}{d^n(\tilde{x}^k)^2} \quad (6)$$

$$J(k_3, k_2) := \sum_{k=1}^{N_{\tilde{x}}} (J_1 + w_2 J_2 + w_3 J_3) \quad (7)$$

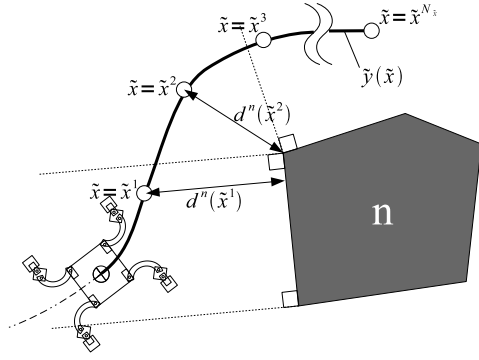


Fig. 4 The shortest distance to an obstacle

$$(k_3, k_2) = \arg \min J(k_3, k_2) \quad (8)$$

$J_1$  は経路を基準経路に近づけるための評価,  $J_2$  は経路の急激な変化を抑えるための評価,  $J_3$  は障害物への接近に対するペナルティをそれぞれ表している. また  $w_2, w_3$  は評価の重みを表す正の定数である. (7) 式の評価関数を最小にする  $k_3$  と  $k_2$  が,  $J_1 \sim J_3$  の各評価をバランスよく小さくする最適な係数となる. (8) 式の数値計算は BFGS アルゴリズムによる準ニュートン法<sup>12)</sup>によって実装しており, (7) 式の評価関数は値や勾配の計算において初等関数の計算を一切必要としない. また障害物が静止しており検出が理想的に行われる場合,  $k_3 \sim k_0$  がステップ状に変化することは無いため数値計算の初期値に前回の結果を用いることで計算の高速化が見込める. 特に, 浮動小数点演算ユニットの実装が十分ではない組み込み CPU においても高速な演算が可能である.

## 5. モデル予測制御による車輪位置最適化

### 5.1 通路幅の計算

まず通路幅の計算方法について述べる. 通路幅は回避経路上の測定点における法線の方に, 経路の左右それぞれに対して計算を行う. 計算を行う範囲はロボットの後輪位置周辺の通路幅も考慮するために回避経路生成における評価区間よりも後方に広く設定し, 計算点の総数を  $N_D$  とする. ここで浮動小数点演算ユニットの実装が十分ではない計算機においても高速な演算を実現するためには, 初等関数の計算回数を可能な限り減らす必要がある. 最も単純な計算方法として法線の関数を解析的に導出して障害物を近似した多角形の辺との交点を求める方法が考えられるが, この方法では平方根の計算を必要とする交点-計算点間の距離計算を左右合わせて  $2N_D$  回行う必要がある. そこで, われわれは比較的高速に計算可能な方法として, 以下のように計算を行う.

まず回避経路上の任意の計算点  $(\tilde{x}, \tilde{y})$  における法線に沿った, 回避経路の左側を向いたベクトルを  $\mathbf{a}(\tilde{x}) = [\Delta\tilde{x}, \Delta\tilde{y}]^T$  とする. このとき  $\mathbf{a}(\tilde{x})$  と  $\|\mathbf{a}(\tilde{x})\|$ , さらに  $\rho$  を媒介変数とするパラメータ表示での法線の方程式は, つぎのように表現できる. なお  $(\tilde{x}_n, \tilde{y}_n)$  は法線上の点の座標を表し,  $\Delta\tilde{y}$  は経路の左側では正, 右側では負の値として必要とする計算精度に応じて設定する.

$$\mathbf{a}(\tilde{x}) = [\Delta\tilde{x}, \Delta\tilde{y}]^T = \left[ -\frac{d\tilde{y}(\tilde{x})}{d\tilde{x}} \Delta\tilde{y}, \Delta\tilde{y} \right]^T \quad (9)$$

$$\|\mathbf{a}(\tilde{x})\| = \sqrt{1 + \left( \frac{d\tilde{y}(\tilde{x})}{d\tilde{x}} \right)^2} |\Delta\tilde{y}| \quad (10)$$

$$\begin{bmatrix} \tilde{x}_n \\ \tilde{y}_n \end{bmatrix} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} + \mathbf{a}(\tilde{x}) \cdot \rho \quad (11)$$

つぎに polygon  $(\tilde{x}_n, \tilde{y}_n)$  を点  $(\tilde{x}_n, \tilde{y}_n)$  が障害物の内側ならば 1, 外側ならば 0 を返す関数とし, 計算点  $(\tilde{x}, \tilde{y})$  において polygon  $(\tilde{x}_n, \tilde{y}_n) = 1$  を満たす  $\rho$  を回避経路の左右に対して以下のアルゴリズムで求める. ただし  $|D_{max}|$  は通路幅の絶対値の上限であり, 車輪位置の可動範囲に対して十分に大きく設定する.

**Algorithm 1** Find  $\rho$  such that polygon  $(\tilde{x}_n, \tilde{y}_n) = 1$

**Input:**  $\tilde{x}, \tilde{y}$

**Output:**  $\rho$

- 1: Initialize as  $\rho = 1$
- 2: Calculate  $\mathbf{a}(\tilde{x})$  by equation (9)
- 3: Calculate  $\|\mathbf{a}(\tilde{x})\|$  by equation (10)
- 4: **while**  $\|\mathbf{a}(\tilde{x})\| \cdot \rho \leq |D_{max}|$  **do**
- 5:   Calculate  $\tilde{x}_n, \tilde{y}_n$  by equation (11)
- 6:   **if** polygon  $(\tilde{x}_n, \tilde{y}_n) = 1$  **then**
- 7:     **return**  $\rho$
- 8:   **end if**
- 9:    $\rho \leftarrow \rho + 1$
- 10: **end while**
- 11: **return**  $\rho$

ここで計算点  $(\tilde{x}, \tilde{y})$  における通路幅を  $D(\tilde{x})$  とする. このとき  $D(\tilde{x})$  は先の計算で求めた  $\rho$  を用いて次式で計算できる.

$$D(\tilde{x}) = \pm \|\mathbf{a}(\tilde{x})\| \cdot \rho \quad (12)$$

なお (12) 式の符号は  $\Delta\tilde{y}$  と同様に回避経路の左側に対して正, 右側に対して負とする. 以上の計算をモデル予測制御で考慮される範囲に対応した計算点において行うことで, 通路幅が計算される. 本手法は平方根の計算回数が  $N_D$  回で実行可能であり, 比較的高速に計算を行うことができる. また (10) 式の平方根部分をマクローリン展開することでより高速に計算を行うことも可能であるが, 誤差が拡大する. よって本研究では精度を優先し, すべての計算点において平方根の計算を行う.

### 5.2 車輪位置の最適化

続いて最適な参照車輪位置をモデル予測制御によって求める方法について述べる. 本論文では機体のロール方向の安定性に対して支配的な  $y^{ij}$  座標のみを制御の対象とし,  $x^{ij}$  座標は固定する. また車輪は可動範囲内で任意の配置とすることが可能であり, 車輪位置の通路幅に対する最適性と参照車輪位置に対する最適性を同一と見なせると仮定する. ここでつぎのシステムを考える.

$$\frac{dy^{ij}(t)}{dt} = u^{ij}(t) \quad (13)$$

しかし車輪位置の変化には一定の時間を要するため,  $u^{ij}(t)$  はつぎの不等式拘束条件を満足する必要がある. ただし  $|u_{max}|$  は車輪位置変化速度の絶対値の上限である.

$$\left| u^{ij}(t) \right| \leq |u_{max}| \quad (14)$$

不等式拘束条件をモデル予測制御で扱う方法としては、(i) ダミー入力を導入して等式拘束条件に変換した後にラグランジュの未定乗数法を用いる方法、(ii) ペナルティ関数を評価関数に加える方法、などが挙げられる。しかしこれらの方法では非線形な関数を利用することが多く、解析的なアプローチが困難になる。

そこで、つぎの  $s^{ij}$  軸に関する微分方程式を導入する。なお  $s^{ij}$  座標は機体の移動に伴う各車輪の道程に対応する。

$$\frac{dy^{ij}(s^{ij})}{ds^{ij}} = \mu^{ij}(s^{ij}) \quad (15)$$

このとき  $u^{ij}(t)$  と  $\mu^{ij}(s^{ij})$  の関係は  $V (= ds^{ij}/dt)$  を機体の移動する速さとして、次式で表される。

$$u^{ij}(t) = \mu^{ij}(s^{ij}) \cdot V \quad (16)$$

(16) 式より、 $V$  を変化させることで  $u^{ij}(t)$  を調節することが可能になる。よって (14) 式の拘束をすべての車輪に対して満足する上限の  $V$  は、 $V^{ij}(s^{ij})$  を各車輪について拘束を満足する上限の速さ、 $V_d$  を速さの指令値として次式で計算できる。

$$V^{ij}(s^{ij}) = \begin{cases} V_d & \text{if } |u_{max}| \geq |\mu^{ij}(s^{ij})| \cdot V_d \\ \left| \frac{u_{max}}{\mu^{ij}(s^{ij})} \right| & \text{if } |u_{max}| < |\mu^{ij}(s^{ij})| \cdot V_d \end{cases} \quad (17)$$

$$V = \min_{i,j} V^{ij}(s^{ij}) \quad (18)$$

すなわち (15) 式のシステムを使用することで、線形システムの拘束条件なしの最適制御問題に変換することができる。以上より、本研究では (15) 式のシステムに対してモデル予測制御を適用し、以降では  $y^{ij}(s^{ij})$  を状態量、 $\mu^{ij}(s^{ij})$  を入力として扱う。

まずはオンライン計算における繰り返しの冒頭にて、評価区間を状態量のモデル予測制御における初期値  $y^{ij}(0)$  に応じて動的かつ各車輪に対して独立に設定する。これは評価区間を一定すると障害物が評価区間よりも僅かに広い間隔を置いて点するような環境において車輪位置が振動し、かえって機体の安定性を失うことを避けるために必要な手順である。障害物を検出して車輪を機体側に引き寄せせる際は複数の障害物が連続して存在する可能性を考慮し、評価区間を伸ばした上で最小通路幅を検出し、車輪位置が可能な限り振動しないようにする。一方で障害物が検出されない場合は、不必要に評価区間が長いと遥かに前方の障害物を検出して車輪位置が縮むことになるので、評価区間を必要最低限に縮める。以上より評価区間を  $s^{ij} \in [0, s_h^{ij}]$  とし、 $s_h^{ij}$  を  $y^{ij}(0)$  の関数として次式のように変化させる。

$$s_h^{ij} = \begin{cases} -\frac{s_{max} - s_{min}}{y_{max}^{ij} - y_{min}^{ij}} \left( y^{ij}(0) - y_{min}^{ij} \right) + s_{max} & \text{(w.r.t. left side)} \\ \frac{s_{max} - s_{min}}{y_{max}^{ij} - y_{min}^{ij}} \left( y^{ij}(0) - y_{max}^{ij} \right) + s_{max} & \text{(w.r.t. right side)} \end{cases} \quad (19)$$

ただし  $s_{min}$ ,  $s_{max}$  はそれぞれ最短、最長の  $s_h^{ij}$  であり、すべての車輪に対して同一の正の値とする。また  $y_{min}^{ij}$ ,  $y_{max}^{ij}$  はそれぞ

れ  $y^{ij}(s^{ij})$  の可動範囲における最小、最大の値であり、左右の車輪で符号と大きさが対称の値とする。

つぎに評価区間に対応する範囲の通路幅をもとに、目標状態量の計算を行う。本研究では目標状態量を評価区間全体に渡って障害物や通路壁面への衝突を避けられる一定の値として求めることで、ステップ状に変化する目標値に追従させる問題とする。(12) 式で求めた通路幅は  $\tilde{x}$  座標に対する関数として表現されるため、評価区間に対応する範囲の通路幅を得るためには  $\tilde{x}$  座標 -  $s^{ij}$  座標間の関係を考える必要があり、この関係は次式のようになる。

$$\frac{ds^{ij}}{d\tilde{x}} = \sqrt{1 + \left( \frac{d\tilde{y}(\tilde{x})}{d\tilde{x}} \right)^2} \quad (20)$$

また  $s$  軸の原点は車輪の中心に設定しているため、 $s^{ij} = 0$  に対応する  $\tilde{x}$  座標はつぎのように求まる。

$$\tilde{x} = x^{ij} \cos \theta - y^{ij} \sin \theta \quad (21)$$

ただし本研究ではロボットが (5) 式の効果によって傾きが小さくなった回避経路の接線方向を向いて走行するものとして、初等関数の計算回数を減らすために  $\theta \simeq 0$  と見なす。一般の  $\theta \neq 0$  はルックアップテーブルで高速に近似計算可能である。このとき  $\tilde{x}$  座標から  $s^{ij}$  座標への変換は (20) 式を積分することで次式として得られる。

$$s^{ij} = \int_{x^{ij}}^{\tilde{x}} \sqrt{1 + \left( \frac{d\tilde{y}(\tilde{x})}{d\tilde{x}} \right)^2} d\tilde{x} \quad (22)$$

ここで通路幅の測定を行う  $\tilde{x}$  座標を  $\tilde{x}^l$  ( $l = -1, 0, 1, 2, \dots$ ) として、 $\Delta \tilde{x}$  ごとに測定するものとする。まずは  $\theta \simeq 0$  の近似による誤差を考慮して  $\tilde{x} = x^{ij} - \Delta \tilde{x}$  における通路幅を  $l = -1$  の結果として保存し、そして  $l = 0$  以降の通路幅を (22) 式を用いて以下のアルゴリズムで求める。

---

#### Algorithm 2 Measure $D(\tilde{x}^l)$

---

**Input:**  $x^{ij}$

**Output:** Series of  $D(\tilde{x}^l)$

- 1: Initialize as  $s^{ij} = 0$ ,  $l = 0$ ,  $\tilde{x}^l = x^{ij}$
  - 2: **while**  $s^{ij} \leq s_h^{ij}$  **do**
  - 3:   Calculate  $D(\tilde{x}^l)$  by equation (12)
  - 4:   Store  $D(\tilde{x}^l)$
  - 5:   Calculate  $ds^{ij}/d\tilde{x}$  by equation (20)
  - 6:    $s^{ij} \leftarrow s^{ij} + ds^{ij}/d\tilde{x} \Delta \tilde{x}$
  - 7:    $\tilde{x}^{l+1} \leftarrow \tilde{x}^l + \Delta \tilde{x}$
  - 8:    $l \leftarrow l + 1$
  - 9: **end while**
- 

(20) 式は (10) 式の平方根部分と一致するため実際には計算する必要がなく、初等関数の計算回数を抑えることができる。以上で求めた通路幅をもとに、目標状態量  $y_d^{ij}$  を次式で決定する。

$$y_d^{ij} = \min_l D(\tilde{x}^l) + \epsilon \quad (23)$$

$\epsilon$  は通路幅に対する余裕を表すパラメータであり、機体左側の車輪に対しては正、右側の車輪に対しては負の値を設定する。また

目標状態量が車輪の可動範囲を超えた場合は値を飽和させる．

以上の計算の後に,最適入力の計算を行う．ここで  $e(s^{ij}) := y^{ij}(s^{ij}) - y_d^{ij}$  を状態量と目標状態量の誤差として定義し,評価関数をつぎのように設定する．なお以降では標記を簡単にするために添え字  $ij$  は省略する．

$$J := \phi(e(s_h)) + \int_0^{s_h} L(e(s), \mu(s)) ds \quad (24)$$

$$\phi(e(s_h)) = \frac{1}{2} S_f e(s_h)^2 \quad (25)$$

$$L(e(s), \mu(s)) = \frac{1}{2} Q e(s)^2 + \frac{1}{2} R \mu(s)^2 \quad (26)$$

(24) 式の評価関数に対して最適な入力,滑らかな車輪位置の変化によって状態量を目標状態量に追従させる入力となる．また  $S_f, Q, R$  は重みを表す正の定数である．続いて  $\lambda$  を共状態量として,ハミルトニアンをつぎのように定義する．

$$H(e(s), \mu(s), \lambda(s)) = L(e(s), \mu(s)) + \lambda(s) \mu(s) \quad (27)$$

ここで変分法に基づき 1 次の最適性の必要条件を求めると,次式が得られる．

$$\frac{de(s)}{ds} = \mu(s), \quad e(0) = y_r(0) - y_d \quad (28)$$

$$\frac{d\lambda(s)}{ds} = -Qe(s), \quad \lambda(s_h) = S_f e(s_h) \quad (29)$$

$$\frac{\partial H}{\partial \mu} = R\mu(s) + \lambda(s) = 0 \quad (30)$$

(28)~(30) 式の連立微分方程式の解となる  $\mu(s)$  が最適な入力となる．(28)~(30) 式を解くために (30) 式を  $\mu(s)$  について解いた結果を (28) 式に代入すると,つぎの  $e(s)$  と  $\lambda(s)$  に関する常微分方程式が得られる．

$$\frac{d}{ds} \begin{bmatrix} e(s) \\ \lambda(s) \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{R} \\ -Q & 0 \end{bmatrix} \begin{bmatrix} e(s) \\ \lambda(s) \end{bmatrix} \quad (31)$$

$$e(0) = y_r(0) - y_d, \quad \lambda(s_h) = S_f e(s_h) \quad (31)$$

(31) 式はリカッチ微分方程式の解  $S(s)$  を用いて繰り返し計算を行わずに解くことが可能であり,本手法の場合  $S(s)$  は次式を数値積分することで得られる．

$$\frac{dS(s)}{ds} = \frac{S(s)^2}{R} - Q \quad (32)$$

なお (32) 式は解析的に  $S(s)$  について解くことも可能であるが,解に現れる双曲線関数の計算を避けて以降の計算を簡単化するために本研究では数値的に解く．このとき  $e(s)$  は  $S(s)$  を含む次式を数値積分することで得られる．

$$\frac{de(s)}{ds} = -\frac{S(s)}{R} e(s) \quad (33)$$

そして  $\lambda(s)$  は  $e(s)$  と  $S(s)$  から,つぎのように計算できる．

$$\lambda(s) = S(s) e(s) \quad (34)$$

$e(s)$  と  $\lambda(s)$  を (30) 式に代入し,再び  $\mu(s)$  について解いたものが最適な入力となる．

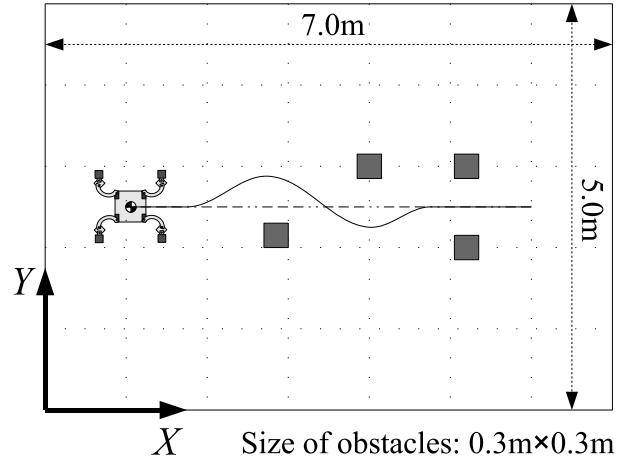


Fig. 5 Experimental environment

Table 2 Experimental parameters

Parameter	Value	Parameter	Value
$\tilde{x}_h$	1.0 m	$y_{min}^{ij}$	0.05 m (L) -0.35 m (R)
$N_{\tilde{x}}$	5	$y_{max}^{ij}$	0.35 m (L) -0.05 m (R)
$w_2$	0.8	$\Delta \tilde{x}$	0.2 m
$w_3$	0.01	$\epsilon$	-0.2 m (L) 0.2 m (R)
$\Delta \tilde{y}$	0.05 m (L) -0.05 m (R)	$S_f$	0.1
$ D_{max} $	0.5 m	$Q$	1.0
$ u_{max} $	0.2 m/s	$R$	0.05
$V_d$	0.8 m/s	$\Delta t$	1 ms
$s_{min}$	0.7 m		
$s_{max}$	0.9 m		

参照車輪位置は最適な入力を  $\mu_o := \mu(0)$  として次式のように更新することで,最適な値へと更新される．なお  $\Delta t$  は参照車輪位置の更新周期である．

$$y^{ij} \leftarrow y^{ij} + \mu_o V \Delta t \quad (35)$$

以上のように,本手法はモデル予測制御の最適な入力を求めるために繰り返し計算を必要としない．また初等関数の計算回数も少なく,組み込み CPU においても高速に最適な参照車輪位置を計算可能である．

## 6. 検証実験

### 6.1 実験環境

提案手法を実験機へ実装し,システムの有効性を制御性能と計算コストの低さの観点から検証する．Fig. 5 は屋内に設置した実験フィールドと障害物の配置を表しており,車輪位置を最適化しながら 2 つの障害物を回避した後,ボトルネック部分を通過させる実験を行う．回避経路の生成における基準経路を  $\tilde{y}_d(\tilde{x}) = 2.5 - Y$  m ( $Y = 2.5$  m) の直線とし,他の各種パラメータについては Table 2 のように設定した．なお表中の (L), (R) はそれぞれ機体の左側,右側の車輪に対する値であることを表している．



## 6.2 実験結果

実験結果を Fig. 6~Fig. 10 に示す．機体と障害物との位置関係の比較を容易にするために，Fig. 6~10 の横軸はすべて機体中心の  $X$  座標として統一している．まず Fig. 6 は慣性座標における機体中心と各車輪の軌道を表しており，中央の曲線が機体中心の軌道，上下の曲線が各車輪の軌道である．また軌道の上下に描かれた長方形は障害物の輪郭を表しており，軌道を跨ぐ四角形は各車輪の中心を直線で結んだものである．Fig. 6 から車輪位置を通路幅に対して変化させながら障害物回避を行っていることが確認でき，障害物付近を走行する際は障害物の表面に対して  $\epsilon = \pm 0.2\text{m}$  程度の余裕をもって通過していることがわかる．また 2 つ目の障害物とボトルネックを形成する障害物との間隔は  $0.85\text{m}$  であるが，この区間において左側の車輪の位置は変化していない．この際は評価区間が  $s_h^{ij} = s_{min} = 0.7\text{m}$  から伸長されており， $s_h^{ij}$  を固定とすると車輪位置が振動する．したがって，評価区間を変化させることの効果が確認できた．

つぎに Fig. 7 は回避経路のパラメータ  $k_3, k_2, k_1, k_0$  の変化を表し，Fig. 8 はモデル予測制御における最適な入力  $\mu_o^{ij}$  の変化を表す．Fig. 7 において  $k_3, k_2, k_1, k_0$  がステップ状に変化することはなく，更新を行う前後では連続した経路が生成されていることがわかる．Fig. 8 は Fig. 6 との比較から，最適解の変動が回避経路の生成とモデル予測制御の評価区間に障害物が入り出す瞬間において大きくなっていることがわかる．また Fig. 8 において  $\mu_o^{ij}$  は対応する車輪が障害物近傍に達するまでに収束しており，車輪位置の変化が通路幅の減少する位置においてすでに完了していることがわかる．

続いて Fig. 9 は  $\mu_o^{ij}$  から求めたロボットの移動する速さ  $V$  の理論値と，(14) 式の拘束条件を満たす上限の速さを表している．Fig. 9 より，減速が必要な場合には拘束条件を満たす最大の速さで走行させていることが確認できる．また Fig. 8 との比較から，減速させる位置は車輪位置の変化が急激な位置と対応していることがわかる．以上より，滑らかに障害物を回避しながら基準軌道に追従する経路が生成され，車輪位置も目標位置に追従するように最適化されているため，目標とする制御が達成されていることが確認できる．よって提案手法は，制御性能の上では目的とするシステムを十分に実現していると判断できる．

そして Fig. 10 は回避経路最適化，参照車輪位置のモデル予測制御，脚関節の数値計算による逆運動学計算の全体での制御周波数を表している．最小の値は  $X = 2.7, 4.0\text{m}$  付近における  $50\text{Hz}$  であり，やはり評価区間に障害物が入り出す瞬間において低下がみられる．しかし本研究では動力学に基づく制御のような高い制御周波数を必要とする制御手法を用いずにシステムを構築しており，また例として  $V = 0.8\text{m/s}$  で走行しながら  $50\text{Hz}$  の周波数で制御を行った場合には機体が  $16\text{mm}$  進むごとに最適解が更新される．よって計算コストの観点からも実用上十分な性能が達成されていると考えられ，提案手法は組み込み CPU のような小規模な演算装置への実装が可能であり，さらには回避経路の生成や車輪位置の制御を伴いながらも比較的速い速度による移動

を実現可能な，計算コストの低い手法であると判断できる．

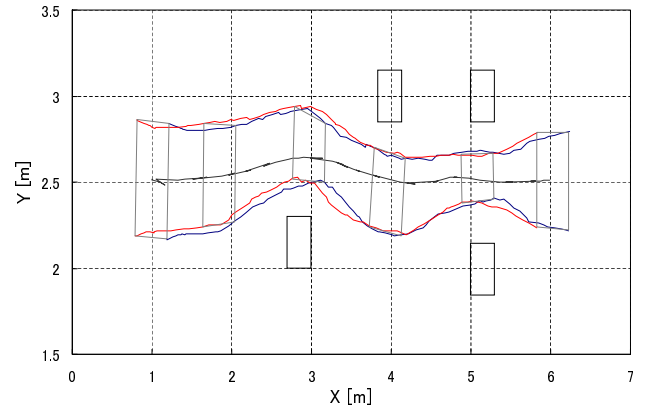


Fig. 6 Trajectories of the robot and wheels

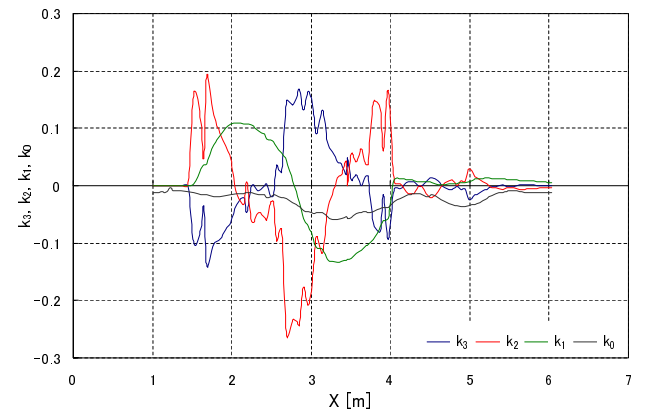


Fig. 7  $k_3, k_2, k_1, k_0$  of the collision-free trajectory

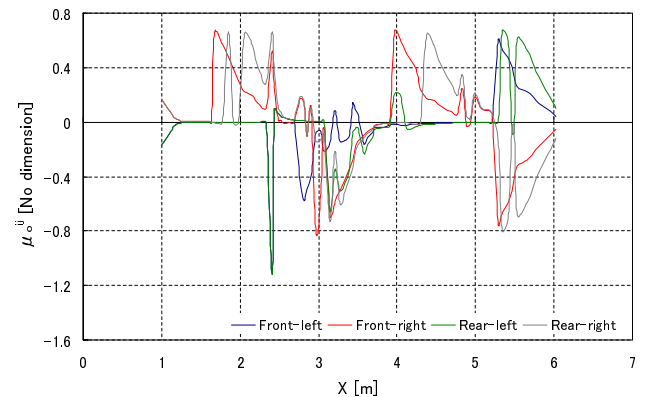


Fig. 8 Optimal inputs computed by MPC

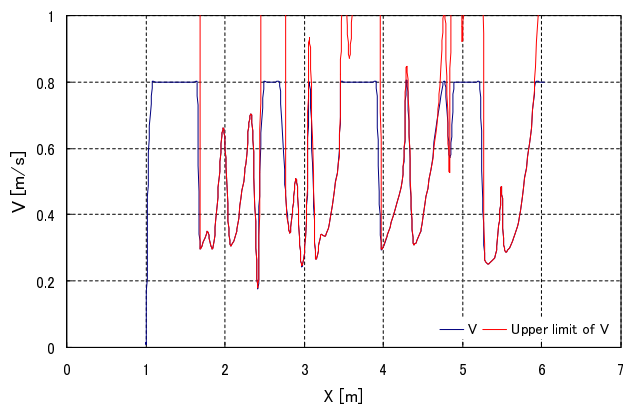


Fig. 9 Locomotion speed

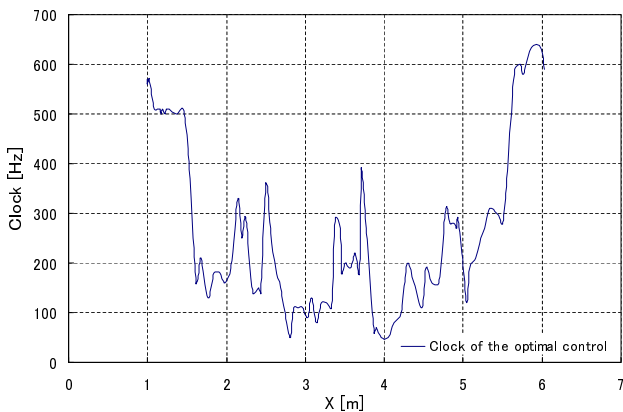


Fig. 10 Computational clock of the optimization

## 7. まとめ

本論文では車両の安定性と機動性の両立を目的として、車輪位置が可変な車両型ロボットを用いた障害物回避経路の生成手法とモデル予測制御を用いた車輪位置の通路幅に対する最適化手法を提案した。また道程に対する車輪位置の変化を制御入力とすることで最適解を繰り返し計算なしで得られる問題への変換を行い、計算コストの低いシステムを実現した。そして提案手法を組み込みCPUのみを制御に使用する実験機へ実装し、障害物を回避しながらモデル予測制御によって車輪位置を最適化する実験を行った。この結果、提案手法は制御性能・計算コストともに実用上十分なものであることが確認され、車両の安定性と機動性の両立を期待できる結果が得られた。

謝辞 本研究は平成 22 年度科研費補助金（基盤研究 C, 21560475）により実施しました。また研究に際して多くの助力を頂いた東京都市大学の鈴木勝正教授、実験に協力して頂いた同大学の高橋直樹氏に深く感謝を申し上げます。

## 参考文献

- 1) 高橋隆行, 横田尚大, 王志東, 森善一, 中野栄二: 車両幅可変型全方向移動ロボットの機構と制御, 日本ロボット学会誌, 19-5, 638/645 (2001)
- 2) M. Wada and H. Asada: A Holonomic Omnidirectional Vehicle with a Reconfigurable Footprint Mechanism and Its Application to Wheelchairs, Proceedings of 1998 IEEE International Conference on Robotics and Automation, 774/780(1998)
- 3) 大和秀彰, 奥村悠, 古田貴之: ホイールモジュールによる移動機構を備えた次世代自律移動ロボット 'Hallucige-nia01', 計測と制御, 45-3, 254/259 (2006)
- 4) 大川功, 野中謙一郎: 多関節脚を有する車両のリアルタイムトレッド最適化を用いた障害物回避, 高速信号処理応用技術学会誌, 12-1, 25/30 (2009)
- 5) T. Ohtsuka: A continuation/GMRES method for fast computation of nonlinear receding horizon control, Automatica, 40-4, 563/574 (2004)
- 6) 村山明弘, 山北昌毅: 非線形モデル予測制御による可変バルブ機構付きエンジンの制御, 電学論 C, 130-5, 828/833 (2010)
- 7) A. Trent, R. Venkataraman and D. Doman: Trajectory Generation Using A Modified Simple Shooting Method, Proceedings of 2004 IEEE Aerospace Conference, 2723/2729(2004)
- 8) I. Okawa and K. Nonaka: Optimal Online Generation of Obstacle Avoidance Trajectory running on a Low Speed Embedded CPU for Vehicle, Proceedings of 2010 IEEE Multi-Conference on Systems and Control, 1257/1262(2010)
- 9) 野中謙一郎, 河村哲郎: 3 軸スカラー型の脚を有する車両ロボットの最適脚配置と非線形軌道追従制御, 日本機械学会 第 10 回「運動と振動の制御」シンポジウム講演論文集, 334/339(2007)
- 10) 野中謙一郎: 3 軸スカラー脚を有する車両型ロボットの連続な操舵の実現, 計測自動制御学会 第 8 回 制御部門大会資料 (2008)
- 11) C. I. Connolly, J. B. Burns and R. Weiss: Path Planning Using Laplace's Equation, Proceedings of 1990 IEEE International Conference on Robotics and Automation, 2102/2106(1990)
- 12) 福島雅夫: 数理計画入門, 朝倉書店 (2000)

## [著者紹介]

大川 功 (正会員)



2011 年 3 月東京都市大学大学院工学研究科機械システム工学専攻修了。在学中は車両型ロボットを対象とした障害物回避, 最適制御, 組み込みシステムによる制御の研究に従事。現在は (株) デンソーに勤務し, 車両制御システムの研究・開発を行う。

野中 謙一郎 (正会員)



1997 年 3 月東京工業大学情報理工学研究科博士課程修了。同年 4 月に武蔵工業大学 (現, 東京都市大学) 工学部機械システム工学科助手, 講師を経て, 2008 年 4 月から准教授。2000 年 4 月から 1 年間ボストン大学客員研究員。開ループ振動安定化, スライディングモード制御, ビークル制御に関する研究を行う。