

経路生成形レギュレータによる 非ホロノミック車両のフィードバック制御*

高島昭彦*¹, 橋本幸男*², 堀 勝博*³
花島直彦*⁴, 足田弘光*⁴, 山下光久*⁴

Feedback Control of Non-holonomic Mobile Robots by Path-generating Regulator

Akihiko TAKASHIMA*⁵, Yukio HASHIMOTO, Katsuhiro HORI,
Naohiko HANAJIMA, Hiromitsu HIKITA and Mitsuhsa YAMASHITA

*⁵ Muroran Institute of Technology, Division of Production and Information Systems Engineering,
27-1 Mizumoto-cho, Muroran-shi, Hokkaido, 050-8585 Japan

In this paper, we propose a new control method for two-wheeled non-holonomic mobile robots. Since the non-holonomic mobile robots has non-integrable constraint conditions, it is difficult to derive a control law so that the robots are convergent to the target state. The proposed method designs a nonlinear regulator carrying out asymptotic convergence of non-holonomic mobile robots to a given trajectory family. The trajectory family is chosen so that it passes through the origin and its gradient at the origin is equal to zero. As the result, the mobile robots are convergent to the origin as the target state while generating their path. We name this method Path-generating Regulator. Neither the coordinate transformation nor the input transformation is needed in this method unlike other methods. Global asymptotic regulation property of this method is proven by a Lyapunov-based stability analysis. By numerical simulations and experiments, we show the validity of the proposed method.

Key Words: Mechatronics and Robotics, Nonlinear Control, Motion Control, Two-wheeled Non-holonomic Mobile Robot, Path-generating Regulator, Lyapunov-based Stability Analysis

1. はじめに

ロボットにおいて移動機能は基本機能の一つであり、移動機能を実現する機構として車輪機構は最も有力である。本論文では車輪機構を持つ移動ロボットにおける運動制御を扱う。

現実に移動ロボットを制御する場合には、車輪の滑り、磨耗、地面の凹凸などの外乱やモデル化誤差の下でも正確な運動を行うように、フィードバック制御系を構成する必要がある。しかしながら、車輪型移動ロボットの多くは非ホロノミックな拘束を有しており、フィードバック制御系を構築することが難しいとされている。ここで、非ホロノミックな拘束とは積分不可能な微分方程式で記述された拘束を指す。このような拘束を持つシステムは一般に非ホロノミックシステムと呼ばれている。

非ホロノミックシステムに対して既に幾つかのフィードバック制御法が提案され、文献⁽¹⁾⁽²⁾にもまとめられている。例えば、時変もしくは区分解析的なフィードバック則等を用いる方法^{(3)~(5)}、時間軸状態制御形制御方策⁽⁶⁾、非線形最適レギュレータによる方法⁽⁷⁾などがある。しかしながら、提案されてきた手法の多くは制御対象の数式モデルを Chained Form と呼ばれる形式に変数変換(座標変換と入力変換)してからフィードバック制御系の設計を行っている。このときの変数変換は大域的には定義されないため、導かれるフィードバック制御則も大域的に定義されたものではなく、結果としてこれらの手法の適用範囲を狭くしている。

本論文において提案する方法⁽⁸⁾は、運動制御の目的を達成できる経路関数群をあらかじめ用意しておき、移動ロボットの描く経路をこれらの関数の一つに収束させるというアイデアに基づいている。リアプノフの安定性定理を用いて経路関数群の中の一つの関数に漸近収束させる非線形状態フィードバック則が導ける点と、変数変換を用いることなく任意の初期状態から指定した最終状態へと移動させるレギュレータが設計できる点が特徴的であり、現実の移動ロボットへの実装

* 原稿受付 2003年3月19日。

¹ 正員, 室蘭工業大学大学院生産情報システム工学専攻(〒050-8585 室蘭市水元町27-1)。

² 室蘭工業大学電気電子工学科。

³ 苫小牧工業高等専門学校電気電子工学科(〒059-1275 苫小牧市錦岡443)。

⁴ 正員, 室蘭工業大学機械システム工学科。

E-mail: takasi-a@mondo.mech.muroran-it.ac.jp

も容易である。この制御によると、人が車を運転するのと似た車両の動きを実現可能で、目標地点が前方にない場合には切り返し動作も生成される。どの経路関数に収束するかは、初期状態や移動中の外乱の受け方によって変わる。外見上は進むべき経路をロボット自身が自律的に選んでいくように見えるので、この制御を従来の手法と区別する意味で経路生成形レギュレータと呼ぶことにする。

2. 経路生成形レギュレータ

ここでは2輪独立駆動型の移動ロボットを考察の対象とする。以降ではこのロボットを簡単に車両または車体と呼ぶ。図1のような車軸中心点の座標を (x, y) 、 x 軸から反時計回りに測った車体軸角度を ϕ とする。ここで車体軸方向の速度を u_1 及び、操舵角速度 $\dot{\phi} = u_2$ が制御できるとすると、車両のキネマティクスは式(1),(2),(3)で記述される⁽²⁾。この微分方程式を制御対象と考える。

$$\frac{dx(t)}{dt} = \cos(\phi(t))u_1(t) \quad (1)$$

$$\frac{dy(t)}{dt} = \sin(\phi(t))u_1(t) \quad (2)$$

$$\frac{d\phi(t)}{dt} = u_2(t) \quad (3)$$

状態変数ベクトル $z = [x(t), y(t), \phi(t)]^T$ を定義し(以下 t は省略する)、任意の初期値ベクトル $z_0 = [x_0, y_0, \phi_0]^T$ から原点 $z_e = [0, 0, 0]^T$ に整定させるレギュレータ問題を考える。 z_e 以外の目標状態ベクトル z_r に整定させたい場合には、 $z - z_r$ を改めて z と置けばよい。

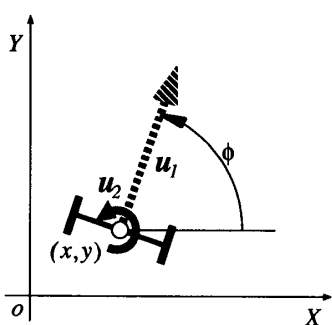


Fig. 1 State variables and velocity inputs of the two-wheeled mobile robot

最初に車両がある位置座標 (x, y) において向かうべき目標姿勢角度 ϕ_r を次のような一階偏微分可能な関数として定義する。

$$\{\phi_r = \phi_r(x, y) : \phi_r(0, 0) = 0\} \quad (4)$$

この ϕ_r に車両の向きを拘束しながら原点 $(0, 0)$ へ移動させると上記の制御目的が達成される。 ϕ_r に沿う経路関数は無限にあり、 $x-y$ 平面上で関数群となる。

目標姿勢角度 ϕ_r と車両の姿勢角 ϕ の偏差を δ とおく。

$$\delta = \phi - \phi_r \quad (5)$$

そして δ に対して式(6)を要請する。

$$\dot{\delta} = -\lambda\delta \quad (6)$$

ここで $\lambda > 0$ である。

さて、式(6)の左辺は式(3)を用いて展開すると次のようになる。

$$\begin{aligned} \dot{\delta} &= \dot{\phi} - \dot{\phi}_r \\ &= u_2 - \frac{\partial \phi_r}{\partial x} \dot{x} - \frac{\partial \phi_r}{\partial y} \dot{y} \end{aligned} \quad (7)$$

これより u_2 が以下のようなになれば、式(6)が満足される。

$$u_2 = -\lambda\delta + \frac{\partial \phi_r}{\partial x} \dot{x} + \frac{\partial \phi_r}{\partial y} \dot{y} \quad (8)$$

式(1),(2)は δ を用いると次のように書ける。

$$\dot{x} = \cos(\delta + \phi_r)u_1 \quad (9)$$

$$\dot{y} = \sin(\delta + \phi_r)u_1 \quad (10)$$

よって式(8)は次式のように書き直せる。

$$u_2 = -\lambda\delta + \left\{ \frac{\partial \phi_r}{\partial x} \cos(\delta + \phi_r) + \frac{\partial \phi_r}{\partial y} \sin(\delta + \phi_r) \right\} u_1 \quad (11)$$

したがって制御対象の状態方程式(1),(2),(3)はそれぞれ式(9),(10),(6)で置き直してよいことがわかった。

次に、 (x, y, δ) に関するシステム、すなわち状態方程式(9),(10),(6)の原点 $((x, y, \delta) = (0, 0, 0))$ が漸近安定となるように車速制御則 u_1 を求める。リアプノフ関数の候補を以下のようにおく。

$$V = \frac{1}{2} \{ \delta^2 + \lambda_1 x^2 + \lambda_2 y^2 \} \quad (12)$$

ここで $\lambda_1, \lambda_2 > 0$ である。

状態方程式(9),(10),(6)の解に沿って式(12)を時間微分すると次式を得る。

$$\begin{aligned} \frac{d}{dt} V &= \delta \dot{\delta} + \lambda_1 x \dot{x} + \lambda_2 y \dot{y} \\ &= -\lambda \delta^2 + \{ \lambda_1 x \cos(\delta + \phi_r) + \lambda_2 y \sin(\delta + \phi_r) \} u_1 \end{aligned} \quad (13)$$

ここで u_1 を次のようにおく。

$$u_1 = -\{ \lambda_1 x \cos(\delta + \phi_r) + \lambda_2 y \sin(\delta + \phi_r) \} \quad (14)$$

式(14)を式(13)へ代入すると,

$$\frac{d}{dt}V = -\lambda\delta^2 - \{\lambda_1 x \cos(\delta + \phi_r) + \lambda_2 y \sin(\delta + \phi_r)\}^2 \quad (15)$$

式(15)は原点以外で半負定になり, 式(12)はリアプノフ関数である.

次に式(15)が原点以外で負定であるかどうかを調べる. $\delta \neq 0$ の場合は負定であることは明らか. $\delta = 0$ の場合, $x \neq 0$ または $y \neq 0$ であっても式(16)を満たす ϕ_r があると負定にならない.

$$\lambda_1 x \cos(\phi_r) + \lambda_2 y \sin(\phi_r) = 0 \quad (16)$$

$\phi_r(x, y)$ を選ぶ際に, 原点以外で式(16)が常に成り立たないものを選べば, 式(15)が原点以外で負定となる. このとき, 車速制御則 u_1 を式(14)とすれば, システムの原点は大域的に漸近安定な平衡点となる.

さて, $\phi_r(x, y)$ の選び方について考える. 式(16)を成り立たせる ϕ_r を求めると, 以下の関係式を得る.

$$\tan(\phi_r) = -\frac{\xi x}{y} \quad (17)$$

ここで $\xi = \lambda_1/\lambda_2 > 0$ である. ところで, 式(2)を式(1)で割ると次式を得る.

$$\tan(\phi_r) = \frac{dy}{dx} \quad (18)$$

式(18)より式(17)を満たす (x, y) は式(19)を満たさなければならない.

$$\frac{dy}{dx} = -\frac{\xi x}{y} \quad (19)$$

式(19)は容易に解けて, 式(20)の解を得る.

$$\xi x^2 + y^2 = C \quad (20)$$

ここで C は積分定数で, x, y は実数なので $C > 0$. したがって, 式(20)を満たす (x, y) は原点を中心にもつ楕円群の上の点となり, 式(17)を満たす ϕ_r は式(20)の楕円上の点 (x, y) における接線の傾きを表す. ところで, ここでの制御は車両の位置を原点の位置に収束させるのが目的であったから, 目標とする経路を生成するための関数は $x-y$ 座標系の原点を通り原点での接線の傾きが0になる関数で, しかもいずれかの方向に開いた関数とすればよい. このような経路関数群を使う場合, 生成された経路関数は楕円群と座標原点以外の点で必ず交差する. それゆえ, $\delta = 0$ のとき, すなわち車両がこれらのうちの一つの関数の上に乗って移動しているときには原点以外のどの位置においても楕円の接線方向を向くことはない.

3. べき関数による経路生成

経路関数群として次式(21)のようなべき関数を用いる.

$$y = ax^n \quad n > 1, a \text{ は任意} \quad (21)$$

式(21)の両辺を x で微分すると式(22)を得る.

$$\frac{dy}{dx} = anx^{n-1} \quad (22)$$

式(22)に式(21)を代入し, 式(18)を用いると,

$$\frac{ny}{x} = \tan(\phi_r) \quad (23)$$

つまり ϕ_r は,

$$\phi_r = \tan^{-1}\left(\frac{ny}{x}\right) \quad (24)$$

となる.

式(24)の表現では $x=0$ は許されないが, 式(22)において, $x=0$ のとき $dy/dx=0$ となることがわかり, ϕ_r は条件式(4)における $\phi_r(0,0)=0$ を満足している.

図2は $n=2$ としたときの格子代表点における目標姿勢角 ϕ_r を矢印で表したものである.

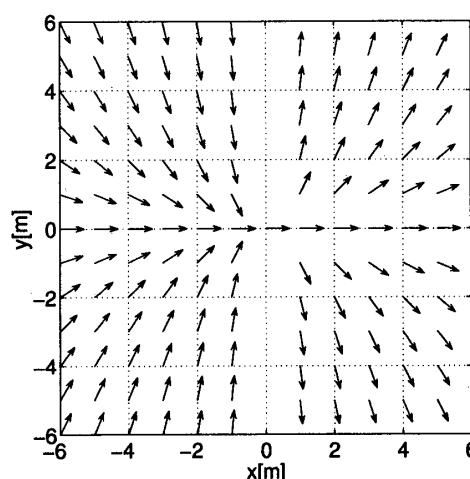


Fig. 2 Attitude angle $\phi_r = \tan^{-1}(ny/x)$ on the lattice points when $n = 2$. Inclination of arrows represents ϕ_r .

制御則である式(11), (14)は式(5), (24)を用いると, それぞれ次のようになる.

$$\begin{aligned} u_2 = & -\lambda \left\{ \phi - \tan^{-1}\left(\frac{ny}{x}\right) \right\} \\ & + \left\{ \frac{\partial}{\partial x} \tan^{-1}\left(\frac{ny}{x}\right) \right\} \cos(\phi) \\ & + \left\{ \frac{\partial}{\partial y} \tan^{-1}\left(\frac{ny}{x}\right) \right\} \sin(\phi) \right\} u_1 \\ = & -\lambda \left\{ \phi - \tan^{-1}\left(\frac{ny}{x}\right) \right\} \\ & + \frac{n \{ x \sin(\phi) - y \cos(\phi) \}}{x^2 + n^2 y^2} u_1 \end{aligned} \quad (25)$$

$$u_1 = -\{\lambda_1 x \cos(\phi) + \lambda_2 y \sin(\phi)\} \quad (26)$$

車両が任意の位置と姿勢にある場合、式 (25) は目標姿勢角 $\phi_r = \tan^{-1}(ny/x)$ に漸近する入力を与え、式 (26) は車両を原点方向へ移動させる進行方向速度入力 u_1 を与える。

本手法で難点と考えられるのは、制御則 u_2 の中に ny/x が入ってきてしまう点であり、したがって数学的には原点を除く y 軸上で制御できないことになる。しかしながら、制御を実装する立場で見た場合には以下のような操作を施すことによって、この問題は解決できる。すなわち、観測した位置 x が 0 と見なせるほど小さい値であれば位置 y の値を調べ、 y の値が 0 とは見なせない有限値であったなら ny/x を計算せずに $\tan^{-1}(ny/x)$ の値を x, y の符号に応じて直接 $\pi/2$ あるいは $-\pi/2$ と近似する。また、 y の値が十分 0 に近ければ、車両の最終目標点である原点に収束したと判断して、制御を打ち切ることとする。このような実装上の工夫を許すのであれば、提案した制御法は原点への収束を $x=0$ の場合でも保証できるようになる。このような近似を採用し、 y 軸を横切るような動きが現れる場合の車両の制御結果については次節 4.2 の数値シミュレーションの中で触れることとする。

4. シミュレーション

先に示した制御則 (式 (25), (26)) を制御対象 (式 (1),(2),(3)) に適用した場合の数値シミュレーション結果を示す。

4.1 ノミナル応答 図 3 において車両初期状態 1~5 は式 (25), (26) 内のパラメータを $(n, \lambda, \lambda_1, \lambda_2) = (2, 4, 2, 1)$ と設定し、初期状態を種々に変えた計算結果である。また車両初期状態 6 は n だけを変更した場合の計算結果である。いずれの場合も原点 $((x, y, \phi) = (0, 0, 0))$ への収束をみることができる。また、 n を変更することで経路の形状を変更することができるのわかる。

さらに同じパラメータの制御器であっても、初期状態の違いによって生成される経路は異なったものになる。ここでは経路関数群として $y = ax^2$ を用いているが、例えば、車両初期状態 2,3,4 では、それぞれ $a = -1.1092$, $a = -0.1304$, $a = 0.1470$ という係数の 2 次関数に収束する。これらの例では車両は一度前進したのち、切り返しを行ってから後退姿勢で原点へと移動を行う。これらの切り返しは意図的に指示したのではなく、式 (25), (26) を実行すると自動的に現れるものである。図 4 は図 3 の車両初期状態 2 に関する時間応答である。 $u_1 = 0$ となる 0.16 sec 付近で切り返しを行っているのがわかる。

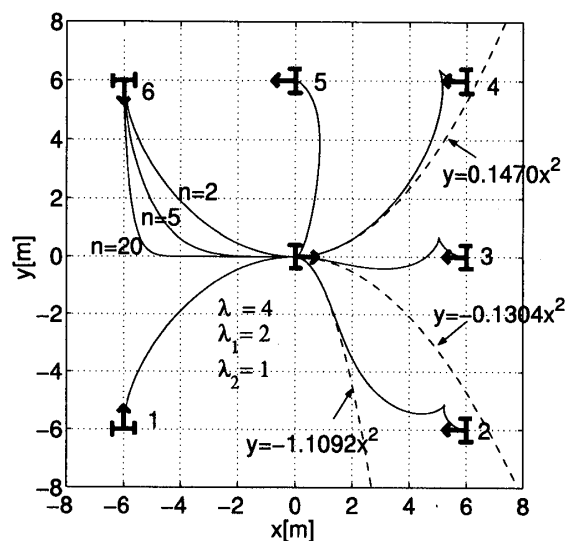


Fig. 3 Generated paths with respect to six initial states

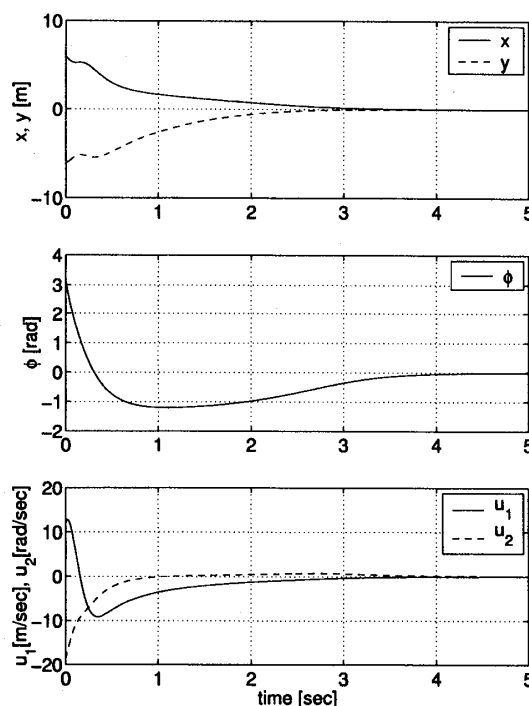


Fig. 4 Time responses of x, y, ϕ, u_1 , and u_2 for initial condition of No.2

4.2 y 軸近傍の挙動 前節 3 で指摘した y 軸近傍の挙動を確認するために y 軸上からスタートさせたときの車両の応答を図 5 の左図に示す。初期状態は $(x_0, y_0, \phi_0) = (0, 5, \pi/2)$ 、パラメータは $(n, \lambda, \lambda_1, \lambda_2) = (2, 10, 5, 5)$ である。車両は切り返ししながら y 軸に沿って原点へ向かい、制御が不能になることはない。文献⁽⁵⁾の制御と同様の動きが現れる点で興味深い、実際の制御ではこのような連続した切り返しの経路を描く

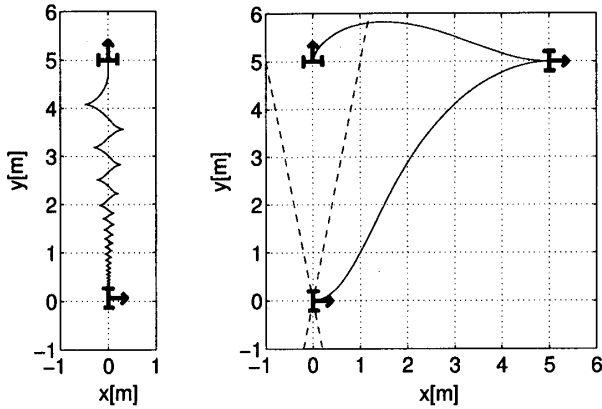


Fig. 5 Generated paths at the initial states $(x_0, y_0, \phi_0) = (0, 0.5, \pi/2)$. Original strategy(left) and improved strategy(right).

ことは現実的ではないし、収束にも時間がかかる。そこで、車両を一度 y 軸から遠ざける方策を考える。例えば、 $x-y$ 平面上に次のような領域を設ける。

$$y^2 - 25x^2 > 0 \quad (27)$$

式 (27) を満たす位置に車両が存在した場合、その位置を (x_e, y_e) とする。そして車両の目標状態を、 $x_e = 0$ の場合は $(0, 0, 0) \rightarrow (y_e, y_e, 0)$ 、 $x_e \neq 0$ の場合は $(0, 0, 0) \rightarrow (\text{sgn}(x_e)y_e, y_e, 0)$ と変更し、この新たな目標状態に車両が到達した時点で再び目標状態を $(0, 0, 0)$ へと戻す。

このような方策の下に図 5 の左図と同じ初期状態およびパラメータで行ったシミュレーションの結果を図 5 の右図に示す。切り返しの回数が 1 回で済み、収束時間が短縮され、この方法が有効であることがわかる。

4.3 誤差・雑音がある場合の応答 ここでは、文献⁽¹⁾において比較検討されているモデル化誤差と観測雑音に関して、同様の条件でシミュレーションを実行し、影響を調べる。

4.3.1 モデル化誤差 文献⁽¹⁾と同様に、モデル化誤差として右の車輪の半径が設計値より 50% 大きいとしたときの結果を示す。

各車輪角速度指令に対して十分追従速度の速い速度サーボ系が組まれているものとする。左右の車輪の半径をそれぞれ r_l, r_r 、左右の車輪への車輪角速度指令値をそれぞれ ω_l, ω_r とする。式 (25)、(26) で演算される $u = [u_1, u_2]^T$ と $\omega = [\omega_r, \omega_l]^T$ の関係は

$$u = A\omega = \begin{bmatrix} r_r & r_l \\ \frac{2}{2R} & \frac{2}{2R} \\ r_r & -r_l \\ \frac{2}{2R} & \frac{2}{2R} \end{bmatrix} \omega \quad (28)$$

ここで $2R$ は車輪間の距離である。右車輪の半径の誤差を導入して、 r_r を ar_r とした場合の変換行列 A を A' とする。式 (25)、(26) で演算された u と誤差が混入した u' の関係は

$$\begin{aligned} u' &= A'A^{-1}u \\ &= \begin{bmatrix} ar_r & r_l \\ \frac{2}{2R} & \frac{2}{2R} \\ ar_r & -r_l \\ \frac{2}{2R} & \frac{2}{2R} \end{bmatrix} \begin{bmatrix} \frac{1}{r_r} & \frac{R}{r_r} \\ \frac{1}{r_l} & -\frac{R}{r_l} \end{bmatrix} u \\ &= \begin{bmatrix} a+1 & R(a-1) \\ \frac{2}{2R} & \frac{2}{2R} \\ a-1 & a+1 \\ \frac{2}{2R} & \frac{2}{2R} \end{bmatrix} u \end{aligned} \quad (29)$$

この関係式をシミュレーションに組み込んで計算した。初期状態 $(x_0, y_0, \phi_0) = (3, 2, 0)$ 、パラメータは $(n, \lambda, \lambda_1, \lambda_2) = (2, 10, 1, 1)$ である。式 (29) での車輪間距離 $2R$ は 0.3m 、 a は 1.5 とした。シミュレーション結果を図 6 に示す。生成される経路は異なるが、モデル化誤差があっても、定常偏差なく原点へ収束している。

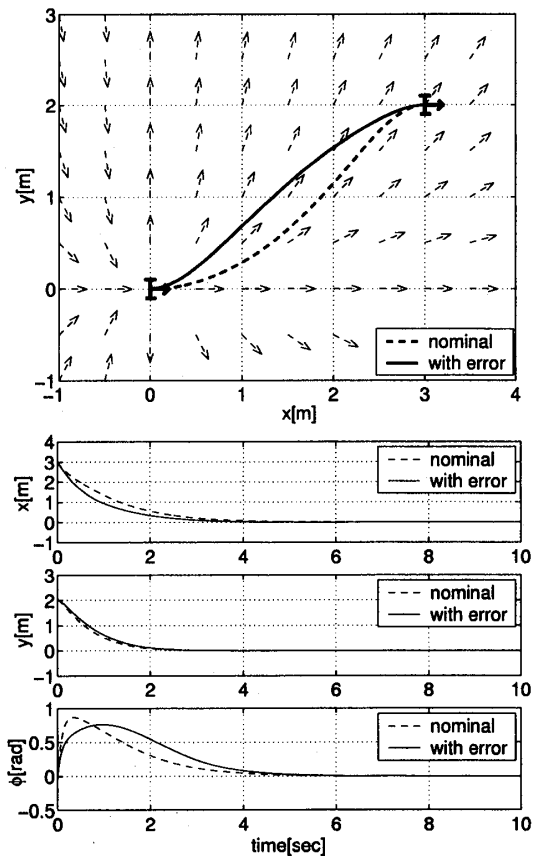


Fig. 6 Generated paths and time responses with/without parameter error

4.3.2 観測誤差 文献⁽¹⁾と同様に、観測誤差として x, y, ϕ の測定に平均 0、分散 1、標準偏差 1 の正規分

布乱数を加えた場合の応答結果を示す(図7). パラメータが $(n, \lambda, \lambda_1, \lambda_2) = (2, 20, 1, 1)$ の場合, y 軸へ先に到達してしまい, y 軸上を振動を伴いながら原点へ収束する. もしこれを避けたい場合は, λ の値を影響の無い程度に小さく設定し, x 軸への到達を速めるために n を大きくすればよい. パラメータが $(n, \lambda, \lambda_1, \lambda_2) = (20, 5, 1, 1)$ の場合, x 軸への到達が速くなり, このような振動が軽減できている.

ϕ の時間応答からわかるように, 雑音のため原点付近においても振動は持続しているが, このようなときは $\phi = 0$ となったところで制御を停止させる.

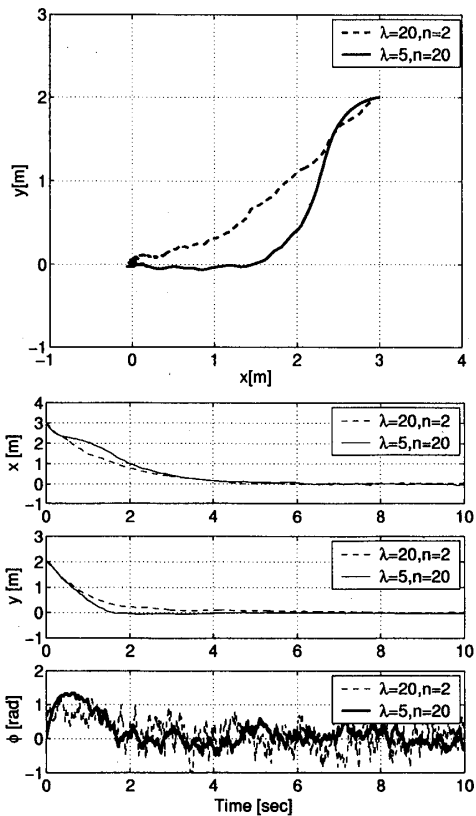


Fig. 7 Generated paths and time responses with measurement noise

5. 実験

図8に示すような移動ロボットに対し, 本手法を適用した結果を示す. 移動ロボットはステッピングモーター(分解能 0.1π rad/step)を駆動力とし, 制御用のマイクロコンピュータとして日立製 H8/3664F⁽⁹⁾を用いている. 車輪間距離は 0.071 m, 車輪半径は 0.027 m である. ロボットの上部に赤外線 LED を取り付け, 頭上に2次元 PSD(位置検出素子)である浜松ホトニクス製 S1880 を設置し, ロボットの絶対位置を観測した(図9). 図10に実験結果の一例を示す. 初期

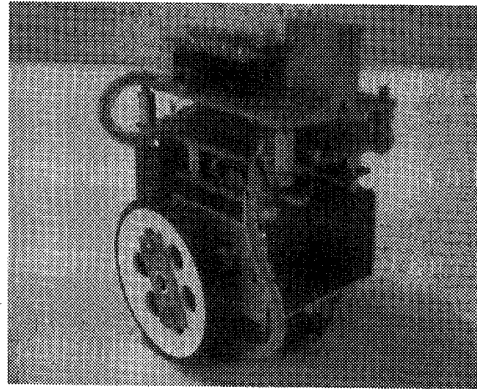


Fig. 8 Two-wheeled mobile robot for evaluation of proposed controller

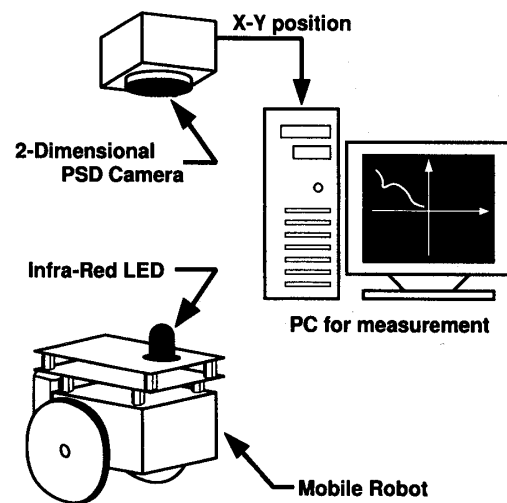


Fig. 9 Experimental setup

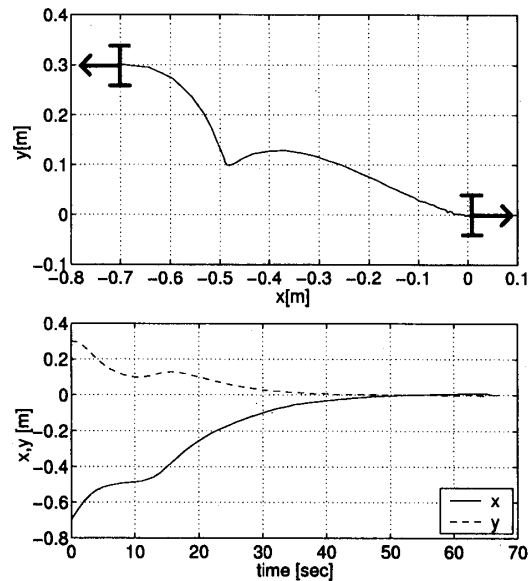


Fig. 10 Experimental result

状態は $(x_0, y_0, \phi_0) = (-0.7 \text{ m}, 0.3 \text{ m}, \pi \text{ rad})$, 制御周期は 0.016 sec , パラメータは $(n, \lambda, \lambda_1, \lambda_2) = (2, 0.1, 0.1, 0.1)$ である. $x = -0.5 \text{ m}$ 付近で切り返し, 原点に収束したことがわかる.

パラメータの値を大きくすると, 収束速度を速められるが, ステッピングモータの出力上限からあまり大きくはできなかった. 本手法は制御にかかる計算量が比較的少なく, ワンチップマイコンを使った比較的簡素な装置構成であっても十分適用可能である.

6. おわりに

非ホロノミック車両を目標地点に移動する非線形レギュレータである経路生成形レギュレータを提案した. リアプノフの安定性定理を用いてあらかじめ与えた経路関数群の中の一つの関数に漸近安定させる非線形状態フィードバック則を導いた. 数値シミュレーションにより任意の初期値からのレギュレータを達成できることを示した. また, モデル化誤差および観測誤差に対する影響を示し, それに対する具体的な方策も示した. 本手法を実際の2輪独立駆動型移動ロボットに適用し, 実験により有効性を確認した. 本論文では経路関数群としてべき関数を用いたが条件を満たせば他の

関数も利用可能である.

文 献

- (1) 三平満司, 非ホロノミック系のフィードバック制御, 計測と制御, **36-6**(1997), 396-403.
- (2) 美田勉, 非線形制御入門, (2000), 昭晃堂.
- (3) C.Samson, Control of Chained System Application to Path Following and Time-Varying Point-Stabilization of Mobile Robots, *IEEE Trans. Autom. Control*, **40-1**(1995), 64-77.
- (4) O.J.Sordalen and O.Egeland, Exponential Stabilization of Nonholonomic Chained Systems, *IEEE Trans. Autom. Control*, **40-1**(1995), 35-49.
- (5) J-B.Pomet, Explicit Design of Time-varying Stabilizing Control Laws for a Class of Controllable Systems without Drift, *System & Control Letters*, **18**(1992), 147-158.
- (6) 清田洋光, 三平満司, 時間軸状態制御形によるドリフト項を持たない非ホロノミックシステムの安定化, システム制御情報学会誌, **12-11**(1999), 647-654.
- (7) 今福啓, 山下裕, 西谷紘一, Hamilton-Jacobi 偏微分方程式の粘性解を用いた三輪移動体の制御, 日本ロボット学会誌, **17-5**(1999), 85-91.
- (8) 橋本幸男, 堀勝博, 土谷武士, 経路生成形レギュレータと非ホロノミック車両の制御, 第31回計測自動制御学会北海道支部学術講演会, (1999), 59-60.
- (9) H8/3664 シリーズハードウェアマニュアル, (2000), 日立製作所.