

高速移動ロボットの機構と軌道追従制御

Mechanism and trajectory tracking control for High-speed wheeled mobile robot

○広地正樹, 海野隆充, 高橋隆行, 庄司道彦, 中野栄二

○Masaki Hirochi, Takamitsu Uno, Takayuki Takahashi,
Michihiko Shoji, Eiji Nakano

東北大学

Tohoku University

キーワード: 自律移動ロボット (autonomous mobile robot), 高速移動 (high-speed mobility),
軌道追従制御 (tracking control), 滑り (slippage), キャスタ (caster)

連絡先: 〒980-8579 仙台市青葉区荒巻字青葉01 東北大学大学院 情報科学研究科
知能ロボティクス論 (中野・高橋) 研究室 (青葉山キャンパス機械系)

広地 正樹, Tel.: (022)217-7025, Fax.: (022)217-7023, E-mail: hirochi@robotics.is.tohoku.ac.jp

1. はじめに

車輪型移動機構をもつロボットは, 歩行ロボットなどに比べ速く移動することに向いている. 知能ロボットにとって移動速度の向上は応用活躍範囲がより大きく広がることを意味する. 例えば, オフィスや工場など効率化が求められる場所での搬送用ロボットにとっては大きな利点であり, 不審者を発見追跡するような警備ロボット, 人間とテニス等を行うスポーツロボットなどへの応用が考えられる.

しかしながら, 現実には知能移動ロボットの多くは0.3~0.5[m/s]と人間以下である. これはロボットを高速かつ安全に移動させるには, 未解決の多くの問題点が存在するからである.

ロボットを高速で移動させるためには, 高い速度・加速度での運動が必要であり, そこで問題とな

るのは, 車輪の滑りである. この車輪の滑りは, 物理現象としてよくわかっていないのみならず, これを考慮にいった制御問題に関する研究は, 自動車の自動運転¹⁾などを除いてほとんど見受けられない.

滑りとは車輪の回転速度と対地速度に差が生じることである. 低速時には小さく問題にならないが, 急な加減速を行う場合はその差が急激に上昇する. そして上昇した後にある点を境に減少し始める. この非線形性が制御を不安定にする要因となる. この間の滑り状態は極めて短時間に変化するため, 短い割り込み周期でのセンシングなどを通して, 滑りを一定の範囲内に抑えるなどの滑りを考慮した制御が必要となる.

また滑りを検出することも大きな課題である. 一般的に移動ロボットは, 車輪の回転量をロータリーエンコーダにより計測, 推定を行うオドメト

りにより自己位置を推定しながら走行している。しかし、オドメトリは車輪の滑りにより計測誤差が生じる。

急加減速時や高速移動時には、この計測誤差は大きくなり、特に高速移動時にオドメトリに大きな誤差がでるのは、非常に危険なものとなる。さらに高速移動時にはコーナリング時に車輪の進行方向への滑りのみでなく横滑りが大きく生じるため、これを考慮した自己位置推定が必要となる。

本研究では、車輪と路面との摩擦係数など物理的な限界付近で走行することを高速移動と定義し、そのような状況下で安定した移動が可能な知能ロボットの実現を目指している。そこで車輪の横滑りの計測が可能で、高速移動に適した機構をもつ小型の実験ロボットの設計製作を行った。本報告では、ロボットの機構と横滑りの検出法について述べる。さらに、車輪の滑りを考慮しての制御法を提案し、シミュレーションによるその基礎的検討を行った。

2. ロボットの構成

2.1 ハードウェア構成

Fig. 1に設計製作した高速移動ロボットQuick2の写真を、またTable. 1に諸元を示す。回転時に無用なモーメントが発生することを避ける目的で、重心をロボットの中心付近に置くように対称な配置となるように注意してある。主要なメカニズム構成をFig. 2に示す。

駆動輪のモータ、計測輪、キャストにはロータリーエンコーダが取り付けられており、モータと計測輪の回転量、キャストの舵角の計測が可能であり、これらのロータリーエンコーダのみから車輪に生じる滑りを検出することを考える。この手法については、次章で述べる。

Quick2は高加速・高速移動を実現するために、重量に比して大きなDCサーボモータ(70[W]×2)を使

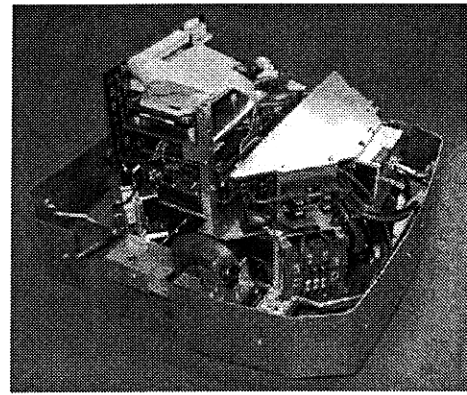


Fig. 1 高速移動ロボット:Quick2

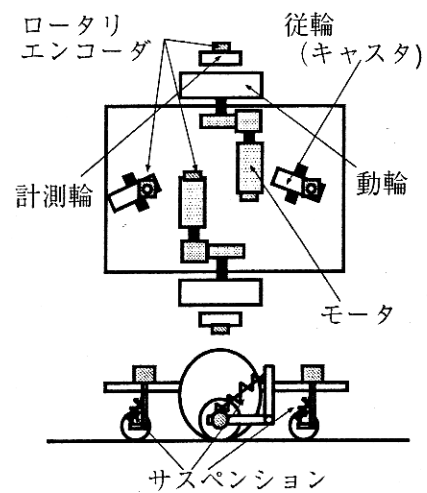


Fig. 2 ロボットのハードウェア構成

Table 1 機械部分諸元

重量	10[kg]
全長	400[mm]
全幅	400[mm]
全高	300[mm]
駆動輪径	100[mm]
計測輪径	66[mm]
駆動ギア比	30/98
搭載コンピュータ	PC/AT互換機
アクチュエータ	DCサーボモータ70[W]×2
バッテリー	小型鉛蓄電池12[V]3.2[Ah]×2

用し、設計最大速度が10.0[m/s]、最大推力が25.6[N]となっている。

加速度の限界は、以下のように、ロボット重量に無関係に、駆動輪の荷重分担率と路面の摩擦により決定されてしまう。車輪と路面との間に働く

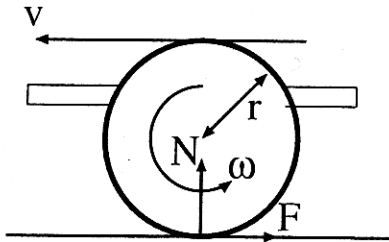


Fig. 3 車輪と地面の間に働く摩擦力

力をFig. 3のようにモデル化しよう。 m :質量, a :加速度, k :駆動輪の荷重分担率(車体の重量を駆動輪が支える割合), g :重力加速度, μ :車輪-路面間の摩擦係数とすると, 垂直抗力 N , 駆動力 F_1 , 摩擦力 F_2 は次の式で表される。

$$N = kmg \quad (1)$$

$$F_1 = ma \quad (2)$$

$$F_2 = \mu N \quad (3)$$

ここで空気抵抗などの小さな抵抗力を無視すると $F_1 = F_2$ が成り立ち, 加速度を表す式 (4) が得られる。

$$a = \mu kg \quad (4)$$

この式から, ロボットの最大加速度は μ と k で決まることがわかる。加減速時でも k を大きくするために重心をできる限り低くし, 車軸のそばにくるように前後左右で対称に近くなるように設計してある。

また定速走行時に k を大きくすることと加減速時の車体のピッチング抑制を両立させるために, 非線型ばねを取り付けたサスペンションをキャストに装備している。

2.2 ソフトウェア構成

次に実験機に搭載したコンピュータの諸元をTable. 2, 制御システムの構成をFig. 4に示す。高速移動時に要求される短い割り込みでのセンシングが可能になるような, リアルタイムOSであるRT-

Table 2 搭載コンピュータ諸元

OS	RT-Linux version 3.1 (Linux version 2.2.19 base)
FPGAボード	Altera社製 EPF6016QC208-3
CPU	Intel社製 PentiumIII 800MHz
メモリ	SDRAM 256MB
M/B	ICP Electronics社製 JUKI-3711P

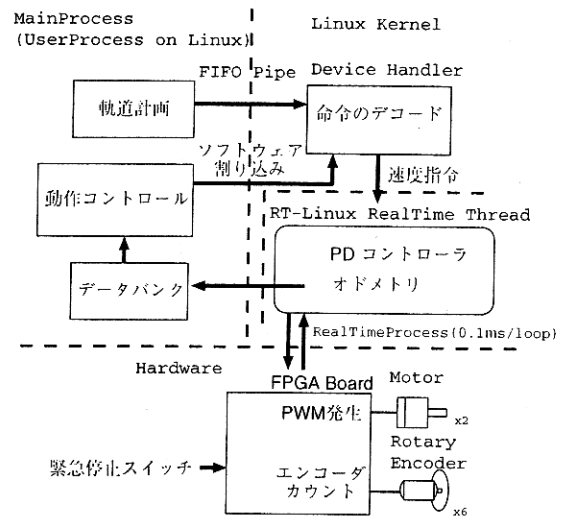


Fig. 4 制御システム構成

Linuxを用いている。RT-Linuxによるリアルタイムプロセス中に,

- 駆動輪PDコントローラ
- 計測輪オドメトリ演算

が実装されている。ロボットの制御はすべてこのリアルタイムプロセス中で行い, ユーザプロセスからの命令によって, 制御目標を変更することでロボットを動作させる。リアルタイムプロセスとユーザプロセスの間はRT-Linuxによって提供されているプロセス間FIFOで繋がれており, これを通して命令の受け渡しを行う。

2.3 モータの高速精密制御

モータの精密な制御のために, CET (Constant Elapsed Time) 法^{2, 3)}を用いてロータリーエンコー

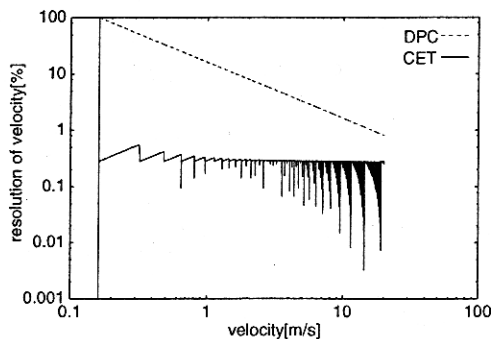


Fig. 5 CET法とDPC法の分解能の違い

ダから速度を求めている。ロボットの回転速度の計算は、割り込み周期当たりのエンコーダのパルスをカウントから計算する方法 (Direct Pulse Counting : DPC) が簡単で広く用いられている。実験機に実装したときに得られるCET法とDPC法での速度分解能の理論値の違いをFig. 5に示す。横軸は、ロボットの進行方向への速度、縦軸は進行方向への速度に対する測定可能な最小速度の割合を表す。DPC法は割り込み周期を短くする程分解能が下がるという短所がある。この問題を解決するために、割り込み周期あたりの、パルスのカウントとパルス間の時間を測定する回路をFPGAボードに実装させてCET法でモータの制御を行った。DPC法では、5[ms]よりも短い制御周期で必要な制御性能を得ることは困難であった。一方CET法では、0.1[ms]での制御が可能であった。制御性能を比較するため、ロボットの片方の車輪を静止させ、もう片方の車輪の目標移動量を正弦波で与えたときの目標値と実験値の誤差を比較する実験を行った。この結果をFig. 6に示す。CET法を用いた場合には、DPC法に比べ誤差を1/7程度に抑え、車輪の目標に対する追従精度を高くすることができた。

3. 車輪の滑りの検出方法

ロボットが直進しているときは、車輪の横に取りつけられた計測輪で滑りの測定が可能である。駆動輪が路面と滑りを生じるときも、計測輪には

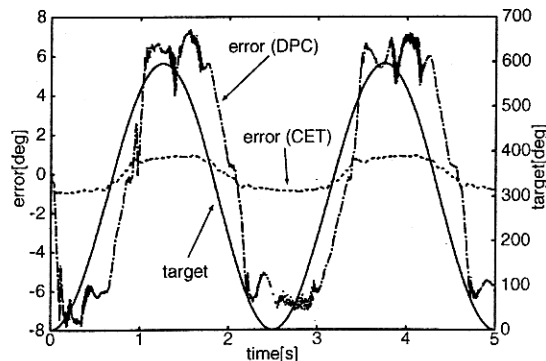


Fig. 6 CET法とDPC法での制御性能の比較

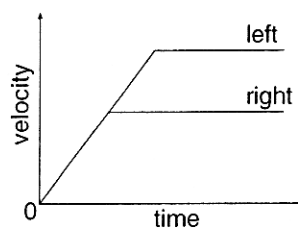


Fig. 7 目標速度線図

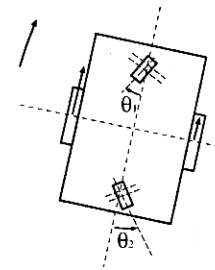


Fig. 8 キャスタの舵角の座標系

サスペンションが装備され常に路面と接触して動くようにしてあり、進行方向への滑りを検出することができるからである。

次にロボットがコーナリングしているときの横滑りの検出を考える。ロボットの前後に取りつけたキャスタの舵角をFig. 8のように θ_1 と θ_2 と取る。ロボットの重心が車軸上より若干後ろ側となっている場合には、高速移動でのコーナリング時には、慣性力によるモーメントを生じてロボットはFig. 9の波線のように横滑りを起こす。このときのキャスタの動きは、横滑りが大きい程 θ_1 は小さくなり、 θ_2 は大きくなると予想される。

実際にロボットが一定の旋回半径の円を描くように、左右の車輪にFig. 7のような目標速度を与えたときの θ_1 と θ_2 を測定する実験を行った。実験は進行方向への最終速度が、0.1, 0.2, 0.5, 1.2, 1.25, 1.3[m/s]となる場合で行った。低速度での実験結果をFig. 10に速度を上昇させたときの結果をFig. 11

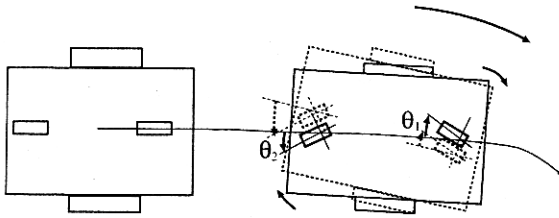


Fig. 9 横滑り時のキャストの動き

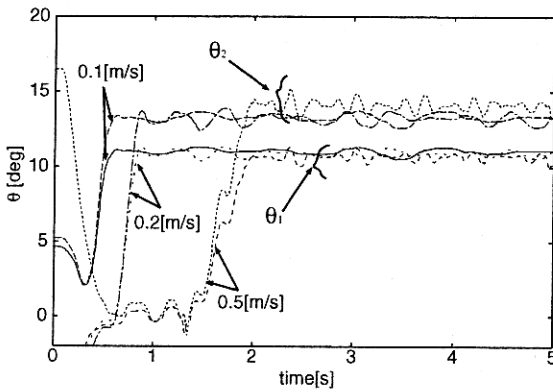


Fig. 10 低速度でのキャストの動き

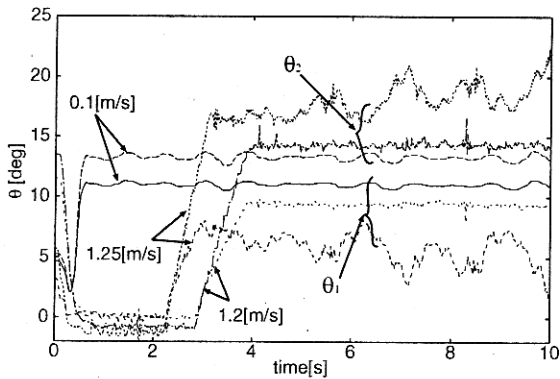


Fig. 11 速度を上昇させたときのキャストの動き

に示す。低速度 (0.1, 0.2, 0.5[m/s]) では θ_1 , θ_2 の値は差がほとんどない一定値に落ち着く。しかし, 1.2, 1.25[m/s]のときの値は, 低速度のときの値に比べ θ_1 は小さく, θ_2 の値は大きくなっている。また1.3[m/s]では, ロボットは滑りその場でスピンしてしまった。これは速度が大きくなるにつれ, 少しずつ車輪が横滑りしていることを示している。

この結果からキャストの動きから逆に横滑りを検出することが可能であると言える。現在のところ, 外部から横滑り量を定量的に計測する手段を

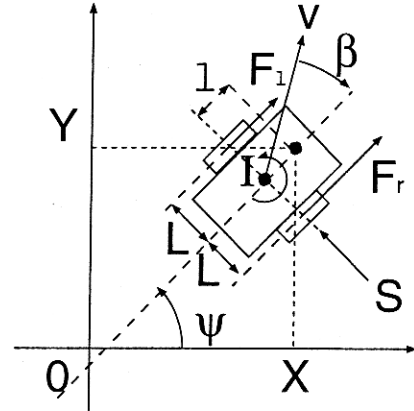


Fig. 12 ロボットのモデリング

持ち合わせていないため, 上記の結果を定量的に考察することはできていない。この点は, 今後の課題である。

4. 滑りを考慮した制御法

横滑り角が検知できているとしたときの, ロボットが動力学を考慮し, 横滑りを起こした限界状態に対応した制御系を考える。 Freundらが行った4輪自動車での滑りを考慮した制御⁴⁾を対向二輪型移動機構での場合に当てはめ, 数値シミュレーションを行いその基礎検討を行った。

4.1 原理

4.1.1 ロボットの数学モデル

ロボットの数学モデルを作成するためにFig. 12に示すように記号を定義した。ここで車軸の中央上に重心があるものとし, 参照点を l だけ前の位置に取っている。ここでそれぞれの値は,

- F_l : 左車輪の推進力
- F_r : 右車輪の推進力
- S : 車軸方向に加わる力
- v : 重心の速度
- β : 横滑り角
- X : 参照点のX座標
- Y : 参照点のY座標
- ψ : ロボットの姿勢角
- I : 重心回りの慣性モーメント
- L : 重心から車輪までの長さ
- l : 重心から参照点までの長さ

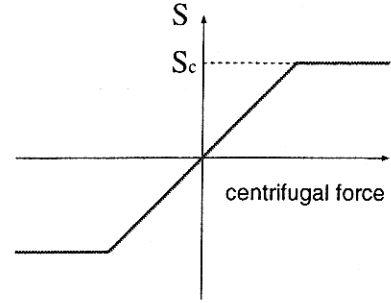


Fig. 13 車軸方向に働く力

である。

ロボットの滑り角 β は、前述の実験機で前後のキャスト角度および計測輪の値を用いることで直接計測できることに注意する。

ロボットの重心の並進運動、重心回りの回転運動の運動方程式は以下のように表される。

$$m\dot{v} = (F_l + F_r) \cos \beta - S \sin \beta \quad (5)$$

$$I\ddot{\psi} = L(F_r - F_l) \quad (6)$$

またロボットの運動を、瞬間回転中心回りの円運動と見なすと、遠心力と向心力の釣り合いより、次式が成り立つ。

$$mv(\dot{\psi} - \dot{\beta}) = -(F_l + F_r) \sin \beta + S \cos \beta \quad (7)$$

ここで車軸方向に働く力 S については、Fig. 13のように、ロボットが横滑りしていない $\beta = 0$ 時には、遠心力と等しく、ロボットが滑っているときには、遠心力より小さくなる特性を持つものとする。そして参照点の速度を表すと次式のように表される。

$$\dot{X} = v \cos(\psi - \beta) - l\dot{\psi} \sin \psi \quad (8)$$

$$\dot{Y} = v \sin(\psi - \beta) - l\dot{\psi} \cos \psi \quad (9)$$

式(5)~(9)を

$$\begin{aligned} \mathbf{x} &= [\beta \quad \psi \quad \psi' \quad v \quad X \quad Y]^T \\ \mathbf{u} &= [F_l \quad F_r]^T \\ \mathbf{y} &= [X \quad Y]^T \end{aligned} \quad (10)$$

と置いて、

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}) \end{aligned} \quad (11)$$

の形に変形すると、

$$\begin{aligned} \mathbf{f}(\mathbf{x}) &= \begin{bmatrix} \psi' - \frac{1}{mv} S \cos \beta \\ \psi' \\ 0 \\ -\frac{1}{m} S \sin \beta \\ v \cos(\psi - \beta) - l\dot{\psi} \sin \psi \\ v \sin(\psi - \beta) + l\dot{\psi} \cos \psi \end{bmatrix} \\ \mathbf{g}(\mathbf{x}) &= \begin{bmatrix} \frac{1}{mv} \sin \beta & \frac{1}{mv} \sin \beta \\ 0 & 0 \\ -\frac{L}{I} & \frac{L}{I} \\ \frac{1}{m} \cos \beta & \frac{1}{m} \cos \beta \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \end{aligned} \quad (12)$$

を得る。

4.1.2 経路制御系

前節で与えられた系の状態方程式は、明らかに非線形で干渉がある。これを線形化して制御系を設計するのは様々な問題が生じる。ここでは、微分幾何に基づく非線形フィードバックによって線形・非干渉化し制御を行う。次差数を次式で定義する。

$$p_i \stackrel{def}{=} \min\{j \mid L_g L_f^{j-1} h_i(\mathbf{x}) \neq 0, (j = 1, 2, \dots)\} \quad (13)$$

本システムでは, $p_1 = 2, p_2 = 2$ となる.ここで次の行列を定義する.

$$C^* \stackrel{def}{=} \begin{bmatrix} L_f^{p_1} h_1(\mathbf{x}) \\ L_f^{p_2} h_2(\mathbf{x}) \end{bmatrix} \quad (14)$$

$$D^* \stackrel{def}{=} \begin{bmatrix} L_g L_f^{p_1-1} h_1(\mathbf{x}) \\ L_g L_f^{p_2-1} h_2(\mathbf{x}) \end{bmatrix} \quad (15)$$

$$\Lambda \stackrel{def}{=} \text{diag}\{\lambda_1, \lambda_2\} \quad (16)$$

$$M^* \stackrel{def}{=} \begin{bmatrix} \sum_{k=0}^{p_1-1} \alpha_{k1} L_f^k h_1(\mathbf{x}) \\ \sum_{k=0}^{p_2-1} \alpha_{k1} L_f^k h_1(\mathbf{x}) \end{bmatrix} \quad (17)$$

これらを用いて制御入力 \mathbf{u} を次のように定める.

$$\mathbf{u}(t) = D(\mathbf{x})^{*-1} \{-C^*(\mathbf{x}) + \Lambda \boldsymbol{\omega}(t) - M^*(\mathbf{x})\} \quad (18)$$

ここで $\boldsymbol{\omega}(t) = [\omega_1 \ \omega_2]^T$ であり, それぞれ $\mathbf{y}(t)$ の各要素 X と Y に対応し, 目標軌道となる. ここで制御入力 \mathbf{u} により状態方程式は線形化され, X, Y は次式に従う.

$$\ddot{X} + \alpha_{11}\dot{X} + \alpha_{01}X = \lambda_1 \omega_1(t) \quad (19)$$

$$\ddot{Y} + \alpha_{12}\dot{Y} + \alpha_{02}Y = \lambda_2 \omega_2(t) \quad (20)$$

目標軌道 $w_1(t), w_2(t)$ を次式で与える.

$$\lambda_1 \omega_1(t) = \ddot{X}_d + \alpha_{11}\dot{X}_d + \alpha_{01}X_d \quad (21)$$

$$\lambda_2 \omega_2(t) = \ddot{Y}_d + \alpha_{12}\dot{Y}_d + \alpha_{02}Y_d \quad (22)$$

X, Y の軌道誤差 e_1, e_2 を次式で定義する

$$e_1 \stackrel{def}{=} X_d - X \quad (23)$$

$$e_2 \stackrel{def}{=} Y_d - Y \quad (24)$$

そして, 目標軌道を線形化された X, Y の式に代入すると e_1, e_2 は次の二次遅れ系になる.

$$\dot{e}_1 + \alpha_{11}e_1 + \alpha_{01}e_1 = 0 \quad (25)$$

$$\dot{e}_2 + \alpha_{12}e_2 + \alpha_{02}e_2 = 0 \quad (26)$$

4.2 シミュレーションによる基礎検討

シミュレーションの物理パラメータを実験機に近い値を入力し, 制御パラメータを以下のように

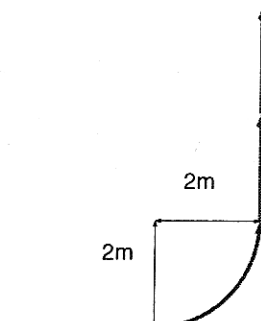


Fig. 14 目標軌道

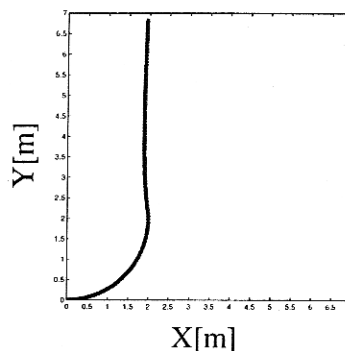


Fig. 15 シミュレーション結果

設定し, シミュレーションを行いロボットの軌道を得た.

$$\alpha_{01} = \alpha_{02} = \lambda_1 = \lambda_2 = 1$$

$$\alpha_{11} = \alpha_{12} = 2 \quad (27)$$

目標軌道は直線と円弧で作成した. 円弧を用いることで遠心力は一定となり, 車軸方向に働く力 S との比較が容易になる. ロボットは, 一定速度1.43[m/s]でFig. 14のように半径2[m]の円弧を90[deg]描き, その後直進させた. このときの速度と半径は, S の最大値 $S_c=10$ [N]の時に, ロボットが横滑りを起こすが, 滑りすぎて制御不能にはならない限界の値に設定してある. シミュレーション結果をFig. 15に示す. ロボットの円弧の横滑りの部分を詳しく描いたのがFig. 16である. 横滑りをして, 参照点を軌道上に維持する動作を見ることができた. また, 円弧が終わると横滑りがなくなり, 非ホロ

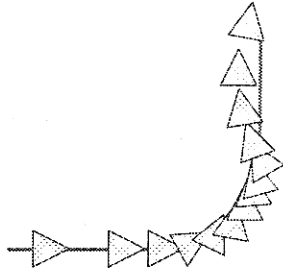


Fig. 16 円弧の横滑りの様子

ノミクな拘束が働くため軌道からずれてしまうが、その後に移動への復帰動作がみられる。以上のことから、本研究のモデルが効果的で横滑りを起こしながらの、限界での円弧と直線を移動する制御をかけることを可能にしたと言える。

5. まとめ

本研究では、高速移動の挙動を調べるための実験ロボットを設計、製作し基礎実験を行った。

本研究で得られた主な結果は以下の通りである。

- 高速移動に適した、ロボットの機構を考え、実験機で実現させた。
- キャスタに取りつけた舵角から、横滑りを検出することが可能であることを確認した。
- 滑り角を得たときに限界状態に対応できる制御系を考案し、数値シミュレーションにより、円弧と直線を移動する制御が可能であることを確認した。

今後は、横滑りを滑り角として定量的に検出し、実験機で滑りを考慮した軌道追従制御の実現を行う。

参考文献

- 1) 片岡寛暁, 佐渡秀夫, 坂井真一郎, 堀 洋一, ファジィ推論を用いた電気自動車用トラクションコントロールシステムのための最適スリップ率推定器, 電気学会産業応用部門誌, Vol.120-D, No.4, pp.581-586, 2000.4

- 2) R.Bonert: Design of a high performance digital tachometer with a microcontroller, IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL.IM-38, pp. 1104-1108, DEC. 1989
- 3) John N.Lygouras: High-Performance Position Detection and Velocity Adaptive Measurement for Closed-Loop Position Control, IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 47, pp. 978-985 NO. 4, AUGUST 1998
- 4) E.Freund, R.Mayr.: Nonlinear path control in automated vehicle guidance, IEEE TRANSACTIONS ON Robotics and Automation, VOL. 13, pp. 49-60 NO.1, 1997