

不整地移動ロボットのための環境適応型制御システム

佐藤 雅紀*¹ 神田 敦司*² 石井 和男*²

Adaptive Control System for an Outdoor Mobile Robot

Masanori Sato*¹, Atsushi Kanda*² and Kazuo Ishii*²

This paper describes the development of an adaptive control system for an outdoor mobile robot. The adaptive control system is composed of an environment recognition system using a self-organizing map and hybrid-neural network controllers based on neural networks. The environment recognition system can recognize the environment in which the robot travels and can switch the hybrid-neural network controller. The hybrid-neural network controllers are tuned by experimental results for each environment. To evaluate the performance of target tracking and vibration suppression, an experiment using the wheeled mobile robot, “Zaurus”, was conducted in rough terrain. As a result, our proposed method could show less oscillatory motion in rough terrain and performed better than a well tuned PID controller.

Key Words: Outdoor Mobile Robot, Rough Terrain, Neural Network, Self-organizing Map

1. はじめに

これまで単純な繰り返し作業を行う労働力として扱われてきたロボットは、その発展とともに人間に代わり複雑な作業を行う労働力として期待されている。ロボットが人間社会において活躍するために必要な要素技術の一つとして、不整地移動機構が挙げられ、近年、不整地移動を目的として受動リンク機構と車輪を組み合わせた様々な移動ロボットが開発されている。筆者らも受動リンク機構に着目し、設計コンセプト [1] [2] や下水道管検査ロボットを開発してきた [3]。不整地走行ロボットとしては、広瀬らの3輪展開型ローバー Tri-Star II [4] や、NASA/JPLが開発したRocker-Bogie機構を有するSojourner [5] [6]、黒田らが開発したPEGASUS機構を有するMicro5 [7] [8]、EPFLが開発したShrimp [9]、CRAB [10]、吉田らのDune Explorer [11]が挙げられる。これらの移動ロボットは車輪・受動リンク機構を採用し、小径の車輪で高い不整地走破能力を有するという特徴をもつ。これは、車輪・受動リンク機構が小型、軽量化を実現し、走破環境に対して受動的に姿勢を変化させることで不整地移動を可能としている。

これまで筆者らはShrimpを参考に車輪型不整地移動ロボットZaurusを開発し、Zaurusが車輪直径の2倍の高さの段差昇降可能であることを示してきた。さらにニューラルネットワー

クを制御器として導入し、その有効性を示してきた [12]~[14]。

不整地移動ロボットがより効果的に不整地移動を実現するためには、ロボットの状態値や走破環境に応じて適応的に振る舞う必要がある。本論文では、不整地移動ロボットが走破中の環境を認識し、その環境に応じて制御器を切り替える環境適応型制御システムを提案する。提案システムは、環境認識システムと環境ごとに調整されたニューロコントローラから構成されている。提案システムの有効性を確認するため、階段昇降を対象として運動性能の評価を行った。本論文の構成は、2章で不整地移動ロボットのシステム構成、3章で提案する環境適応型制御システム、4章で実験を通じた有効性の検討、5章において実験結果の考察、最後に本論文を総括する。

2. 不整地移動ロボット

2.1 システム構成

筆者らが開発した不整地移動ロボットZaurusの外観図をFig. 1、リンク機構と車輪の定義をFig. 2、主な諸元をTable 1に示す。全長約0.7[m]、重量約13[kg]、6車輪独立駆動の車輪・受動リンク型移動ロボットである。移動機構部は、三つの受動リンク機構（フロントフォーク、サイドリンク、リアリンク）から構成され、走破面に対して受動的に姿勢が変化する構造となっている。フロントフォークに接続する車輪を前輪、サイドリンクの2車輪のうち前方に接続する車輪を側面の前輪、後方に接続する車輪を側面の後輪、リアリンクに接続する車輪を後輪と定義する。内界情報を得るためのセンサとして、速度センサ、姿勢センサ、角度センサ、電流センサを有し、コンピュータシステムとして小型ノートPCおよび2種類のマイクロコントロー

原稿受付 2008年12月21日

*¹福岡県産業科学技術振興財団

*²九州工業大学

*¹Fukuoka Industry, Science & Technology Foundation

*²Kyushu Institute of Technology

■ 本論文は有用性で評価されました。

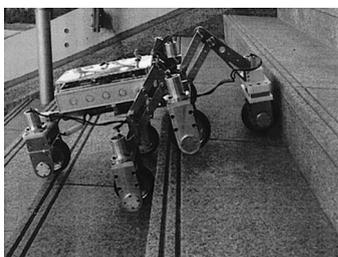


Fig. 1 Overview of Zaurus climbing up stairs

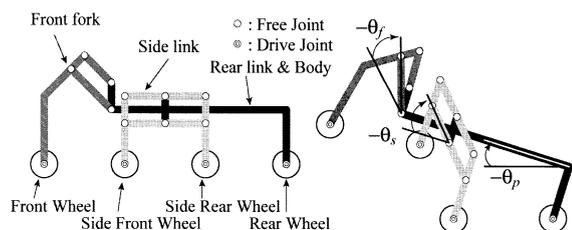


Fig. 2 Link mechanism of Zaurus

Table 1 Specification of Zaurus

Specification	Parameters
Size (L x W x H)	0.66 x 0.53 x 0.31 [m]
Weight	13 [kg]
Computer System	Laptop PC
	PIC18F8720, PIC18F252
Communication	Wireless LAN
	IrDA Remote Control
Sensor	Velocity Sensor
	Angular Sensor (x 3)
	Attitude Sensor
	Current Sensor (x 6)
Battery	Ni-MH 14.4 [V] 3300 [mAh]
Actuator	DC motor 12 [V] 10 [W] (x 6)
Wheel Diameter	0.10 [m]

ラ (PIC18F8720, PIC16F252) を搭載している。

不整地移動ロボットの走破面に対する移動速度を計測するために、経験上、不整地移動時に走破面に接地する可能性が高い側面前輪の横にフリーホイールとエンコーダから構成された速度センサを設置し、得られた速度を不整地移動ロボットの移動速度と定義した。姿勢センサは地面に対する車体のピッチ角 θ_p を計測する。角度センサはフロントフォークおよびサイドリンクの車体に対するそれぞれの角度 θ_f , θ_s を計測する。また電流センサは各車輪の駆動電流を計測する。

2.2 性能評価実験

これまでの実験結果から、車輪直径の2倍の高さ (0.20 [m])

Table 2 Parameters of evaluate function

Parameter	μ_1	μ_2	μ_3	μ_4
Value	10^3	10^5	10^4	10^{-3}

の段差昇降を確認しており、本論文では3段の階段 (踏み面 30 [cm], 蹴上げ高さ 15 [cm]) を対象として走破能力に関する性能評価実験を行った。ここで、不整地移動ロボットが安定した不整地走行を実現することを目標とし、次の評価項目を定義した。

(a) 目標値追従について

$$E_v = \frac{1}{N} \sum_{n=1}^N (v_r - v(n))^2 \quad (= E_1)$$

(b) 振動抑制について

$$E_v = \frac{1}{N} \sum_{n=1}^N (\dot{v}(n))^2 \quad (= E_2)$$

(c) 振動抑制について

$$E_{\dot{\theta}_p} = \frac{1}{N} \sum_{n=1}^N (\dot{\theta}_p(n))^2 \quad (= E_3)$$

(d) 消費エネルギーについて

$$E_J = \sum_{l=1}^6 \sum_{n=1}^N V |I_l(n)| \Delta t \quad (= E_4)$$

(e) 総合評価について

$$E_{Total} = \sum_i \mu_i E_i$$

v_r は不整地移動ロボットの目標移動速度、 $v(n)$ は時刻 n における移動速度、 n はサンプリング数、 N は総データ数、 $\theta_p(n)$ は時刻 n における不整地移動ロボットのピッチ角、 V はモータの駆動電圧、 $I_l(n)$ は時刻 n における l 番目の車輪の駆動電流、 Δt はサンプリング周期 (0.05 [s])、 μ_i はそれぞれの評価値の重みを表す (Table 2)。(a) は目標移動速度と各時刻の移動速度の平均二乗誤差であり、目標値追従性能の評価、(b) は不整地移動ロボットの移動方向における加速度の平均二乗和であり、移動方向の振動に関する評価、(c) は不整地移動ロボットのピッチ角速度の平均二乗和であり、ピッチ角方向の振動に関する評価、(d) は各車輪が不整地移動に要した総消費エネルギーをモータの駆動電流から求めた評価である。(e) は (a) ~ (d) の重み付き和を取った総合評価とした。

Zaurusの車輪は、モータドライバの速度制御により、一定の回転速度に制御される。車輪半径 r および目標移動速度 v_r からモータドライバへ与える目標角速度 $\omega_r (v_r = r \times \omega_r)$ を求める。モータを電源から直接駆動した場合、および、目標移動速度を 0.1, 0.2, 0.3, 0.4 [m/s] とした場合の実験結果を Table 3 に示す。目標移動速度 0.1 [m/s] 以外の場合において3段の階段走破を実現したが、各評価項目において大きな値を示す結果となった。また目標移動速度 0.2 [m/s] における Zaurus の移動速度の時間変化を Fig. 3 に示す。階段走破を実現する一方で、目標値追従が実現できていない。

Table 3 Comparison with climbing up the three step stairs by motor battery direct connection and the constant velocity control using motor drivers

Ctrl. Target [m/s]	E_v ($\times 10^{-3}$)	$E_{\dot{v}}$ ($\times 10^{-5}$)	$E_{\dot{\theta}_p}$ ($\times 10^{-4}$)	E_J ($\times 10^3$)	E_{Total}
w/o ctrl.	-	3.66	13.4	3.78	-
Const. Vel. 0.1	Failure	Failure	Failure	Failure	Failure
Const. Vel. 0.2	25.9	7.0	2.17	1.26	36.33
Const. Vel. 0.3	38.6	18.6	6.95	0.446	64.60
Const. Vel. 0.4	77.4	38.8	14.4	0.355	118.00

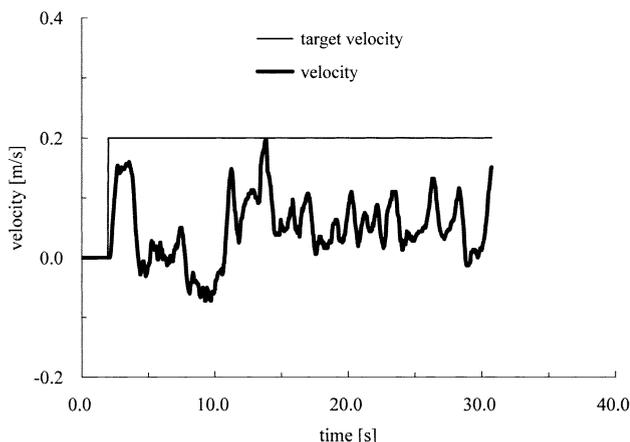


Fig. 3 The experimental results climbing up three step stairs by motor drivers controlled on 0.2 [m/s]

3. 環境適応型制御システム

3.1 システム概要

2章において Zaurus が高い不整地移動能力を有することを示す一方で、モータドライバ単体による速度制御のみでは目標値追従や振動抑制が困難であることが分かる。そこで不整地移動ロボットの状態や走破環境に応じてモータドライバへ与える目標角速度 ω_r を適応的に変化させる制御系が必要となる。

ここで、走破環境である不整地は無数に存在するため、単一の制御器でそれらすべてを考慮することは困難である。これまで筆者らは、不整地移動ロボットの走破環境を認識する環境認識システムの開発 [15] や、環境ごとに調整可能なニューロコントローラの開発 [16] を行ってきた。そこで本論文は、走破環境を認識し環境に応じて制御器を切り替える環境認識システムと、環境ごとに調整されたニューロコントローラを統合し、不整地移動ロボットのための環境適応型制御システムを提案する。

提案する制御システムのコンセプト図を Fig. 4 に示す。不整地移動ロボットから環境を認識するための環境情報が環境認識システムに入力される。本論文では、不整地移動ロボットの受動リンク機構が走破面に対して受動的に変化することに着目し、受動リンク機構のリンク角を環境情報とする。環境認識システムは、あらかじめ代表的な環境をデータベース化しておき、入

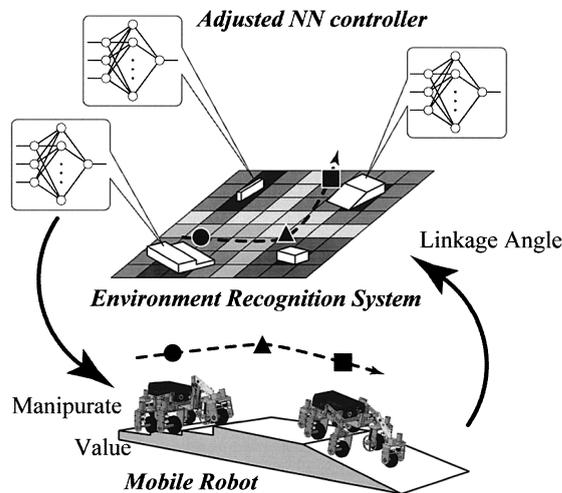


Fig. 4 The concept image of the adaptive control system for the outdoor mobile robot

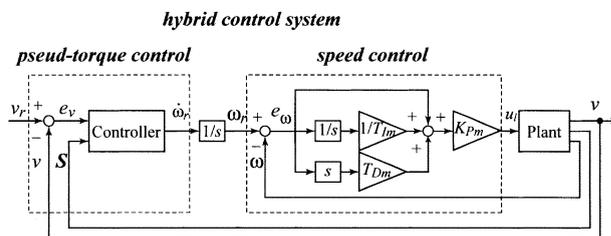


Fig. 5 Hybrid control system for an outdoor mobile robot

力された環境情報をもとに現在の走破環境を認識する。本システムの運用方法は、認識した走破環境に応じて行動決定や地図作成、自己位置推定などのアプリケーションが挙げられる。本論文では環境ごとに調整された制御器の切り替えを行う。環境ごとに調整された制御器はロボットの状態値や走破環境に応じてロボットを制御する。

3.2 ニューロコントローラ

3.2.1 設計方針

本論文で設計するニューロコントローラは、目標移動速度およびロボットの状態値を入力とし、モータドライバへ与える目標角加速度を出力とする擬似的なトルク制御器であり、環境に応じて調整する。この擬似的なトルク制御器と、モータドライバによる速度制御器を組み合わせたハイブリッド制御システムを構成し、効率的な不整地移動の実現を目指す。Fig. 5 にハイブリッド制御システムを示す。モータドライバの速度制御器のモデルを式 (1) に示す。

$$\begin{cases} u_l = K_{Pm} \left(e_\omega + \frac{1}{T_{Im}} \int e_\omega dt + T_{Dm} \dot{e}_\omega \right) \\ e_\omega = \omega_r - \omega_l \end{cases} \quad (1)$$

ここで $K_{Pm} = 0.5$, $T_{Im} = 0.1$, $T_{Dm} = 0.00005$ で、 l は車輪数を示す。制御ゲインはステップ応答法を用い設計した。

PID 制御は Ziegler-Nichols 法などに代表されるいくつかのゲイン調整法が提案されているが、無数に存在する不整地環境

に適用するには多大な労力を要する。本論文では、ニューラルネットワークがもつ非線形システムの表現能力、学習能力、補完能力に着目した制御系設計法を提案する。

提案する制御系設計法のフローチャートを **Fig. 6** に示す。提案する制御系設計法は、まず代表的な不整地における制御対象のダイナミクスを複数学習した逆モデルを構築し、「不整地移動に関する常識」を有するニューロコントローラを構築する。本論文では、代表的な不整地（基本環境）を平地、段差（0.05, 0.10, 0.15, 0.20 [m]）、斜面（10, 20, 30 [deg]）と定義する。複数のダイナミクスを学習したニューロコントローラを基本制御器と定義する。

次にこの基本制御器を各環境へ適用し、追加学習することで環境ごとに調整されたニューロコントローラを得る。このニューロコントローラを専用制御器と定義する。専用制御器は基本制御器を元に構築されているため、同じく「不整地移動に関する常識」を有することが期待される。つまり広く不整地移動について学んだ上で、ある環境に対して特化させる。また基本制御器をもとに設計するため、零からの設計と比較して設計時間の短縮が期待できる。

専用制御器は制御性能に優れる一方で、適用する環境に応じてニューラルネットワークの結合加重などの制御パラメータをあらかじめ保存しておく必要がある。またニューラルネットワークの結合加重や非線形関数は計算コストを要するためマイコンによる実装は困難である。そこで専用制御器が特定の環境下で用いられ、その入出力領域を中心点近傍に限定し、専用制御器のニューラルネットワークが構成する超平面を線形近似した簡易制御器を設計する。簡易制御器は、専用制御器の入力と線形近似して得られた制御パラメータの線形結合により構成される制御器であるため、計算コストに優れる。また基本制御器、専用制御器から派生した制御器であるため、「不整地移動に関する常識」を有することが期待される。

専用制御器と簡易制御器は制御性能と計算コストに関してトレードオフの関係にあり、条件やニーズに応じて使い分ければよい。

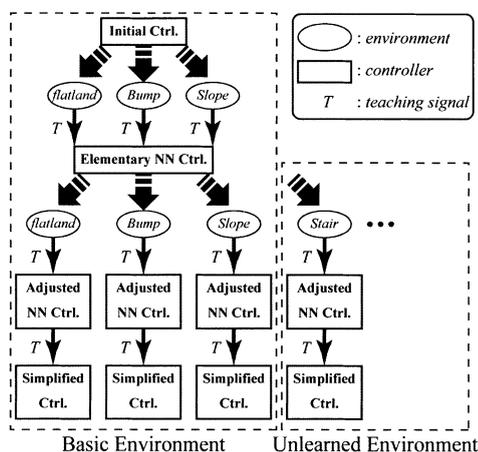


Fig. 6 The proposed controller design method

3.2.2 基本制御器 (Elementary Neural Network Controller)

ニューラルネットワークの学習において重要な課題の一つが教師信号の取得である。石井らは水中ロボットのニューロコントローラ設計において、ロボットの入出力特性よりロボットのエミュレータを構築し、エミュレータとコントローラを一つのニューラルネットワークとして学習することで良好な結果を得ている [17]~[20]。

本論文では、機構解析ソフトウェア DADS (Dynamics Analysis and Design System) および Matlab/Simulink を用いたシミュレータより教師信号を取得し、直接ニューラルネットワークの学習を行い逆モデルの構築を行う。教師信号を得るための初期制御器は設計が容易な比例制御を用いた。式 (2) に初期制御器を示す。目標速度 v_r は 0.05 [m/s]、比例ゲインは平地においてステップ応答法を適用し、試行錯誤的な調整を行い $K_P = 30$ とした。

$$\begin{cases} \dot{\omega}_r = K_P(v_r - v) \\ \omega_r = \int \dot{\omega}_r dt \\ (K_P = 30) \end{cases} \quad (2)$$

教師信号は、初期制御器を基礎環境において適用した際のモータドライバへ与えた目標角加速度 T 、およびシミュレータから得られるロボットの状態値 S の時系列データから構成される。

$$\begin{cases} S = [e_v \int e_v \dot{e}_v v \theta_p \dot{\theta}_p \theta_f \theta_s] \\ (e_v = v_r - v) \\ T = \dot{\omega}_r \end{cases} \quad (3)$$

S は一般の PID 制御系で用いられる入力、および内界センサの出力から構成される。

得られた教師信号から誤差逆伝播法を用いてニューラルネットワークの学習を行う。学習を終えたニューラルネットワークを基本制御器として用いる。**Fig. 7** に基本制御器のブロック線図を示す。目標移動速度 v_r 、およびロボットの状態値 S を入力とし、モータドライバへの目標角加速度 $\dot{\omega}_r$ を出力とする擬似的なトルク制御を行う基本制御器と、目標角速度 ω_r から速度制御を行うモータドライバによるハイブリッド制御系となっている。

3.2.3 専用制御器 (Adjusted Neural Network Controller)

前項で設計した基本制御器を、逆モデルの学習に使用した基礎環境や未学習の環境（例えば階段など）など様々な環境に適用し、式 (4) で示す評価関数によりニューラルネットワークの

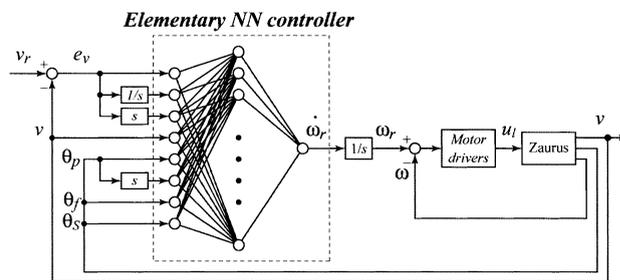


Fig. 7 The block diagram of the elementary controller for Zaurus

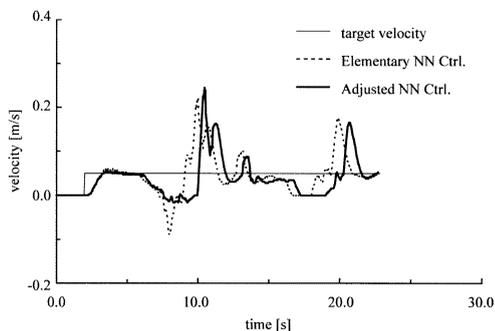
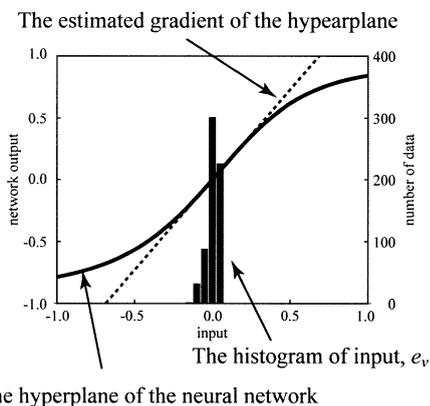


Fig. 8 The experimental result climbing over the 0.18 [m] height of bump. The solid line shows the result of the adjusted controller and the dashed line shows the elementary controller



The hyperplane of the neural network

Fig. 9 Input-Output characteristic of the adjusted controller input e_v . The bars show the histogram of the input. The thick line shows the hyperplane of the adjusted controller, and the dashed line shows the estimated gradient by least square method

結合加重を調整する.

$$E_w = \frac{1}{2} \sum_{n=1}^N (v_r - v(n))^2 \quad (4)$$

式 (4) で示す評価関数は、目標移動速度と不整地移動ロボットの移動速度の時系列データの誤差を小さくすることを目的とする。これはゆっくりとした目標移動速度に対し、十分に目標値追従が実現されることで振動抑制も可能であると考えたからである。

基本制御器を段差高さ 0.18 [m] の上りに適用し、式 (4) で調整した専用制御器を同じ環境で適用した際の移動速度の変化を **Fig. 8** に示す。Fig. 8 に示すように、8 [s] 付近にみられたオーバーシュートが追加学習によって軽減していることが分かる。

3.2.4 簡易制御器 (Simplified Controller)

前項にて設計した専用制御器を線形近似し、計算コストの軽減を図る。線形近似を行う手順は、まず専用制御器がある環境において適用し、専用制御器の入出力特性を得る。専用制御器は特定の環境下で使用されるため、ある程度入力範囲が限定される。中心となる入力範囲の近傍において線形近似し簡易制御器の制御パラメータを導出する。着目する入力状態量を x_c 、その

Table 4 Comparison with the experimental results using four kinds of hyperplane gradient estimated methods. (a) Average of the five trials. (b) Variance of the five trials

(a)					
Estimated method	E_v ($\times 10^{-3}$)	$E_{\dot{v}}$ ($\times 10^{-5}$)	$E_{\dot{\theta}_p}$ ($\times 10^{-4}$)	E_J ($\times 10^3$)	E_{Total}
I	1.85	5.56	2.11	1.66	11.18
II	2.04	6.16	2.34	1.59	12.13
III	2.06	6.19	1.63	1.44	11.32
IV	7.12	17.1	6.02	0.76	31.00

(b)					
Estimated method	σ_v ($\times 10^{-8}$)	$\sigma_{\dot{v}}$ ($\times 10^{-11}$)	$\sigma_{\dot{\theta}_p}$ ($\times 10^{-9}$)	σ_J ($\times 10^3$)	σ_{Total}
I	4.19	9.79	1.92	5.02	20.92
II	12.4	10.3	16.9	4.43	44.03
III	5.98	2.38	0.65	0.22	9.23
IV	27.2	33.1	21.9	1.04	83.24

他の状態量を x_i 、その出力を y_c とすると、専用制御器はニューラルネットワークの写像関数 f を用いて、 $y_c = f(x_c, x_i)$ とする f の線形近似に関して以下の 4 とおりを考える。

- I. 入出力関数: $y_c = \kappa \cdot x_c, \forall x_i = 0$
- II. 入出力関数: $y_c = \kappa \cdot x_c + \lambda, \forall x_i = 0$
- III. 入出力関数: $y_c = \kappa \cdot x_c, \forall x_i = \bar{x}_i$
- IV. 入出力関数: $y_c = \kappa \cdot x_c + \lambda, \forall x_i = \bar{x}_i$

ここで、 κ は傾き、 λ は切片、 \bar{x}_i は各入力の時系列データの平均値を意味する。線形近似する範囲は、入力 e_v の平均値 \bar{e}_v と標準偏差 σ より $[\bar{e}_v - 2\sigma, \bar{e}_v + 2\sigma]$ とする。線形近似は最小二乗法を用いる。

Fig. 9 に階段上りにおける専用制御器の入力の一つ e_v に関する入出力特性を示す。ヒストグラムは 3 段の階段上りにおける入力データ e_v の分布を示す。ヒストグラムで示されるように、入力 e_v は入力範囲 $[-1, 1]$ に対してはほぼ零付近に分布している。ニューラルネットワークの入力 e_v に対する写像関数を線形近似する。線形近似は以下の方法で行った。入力 e_v には $[-1, 1]$ の範囲を入力する。得られた専用制御器の出力、すなわち専用制御器が構成する入出力の写像関係を Fig. 9 の太線で示し、線形近似した直線を破線で示している。

本論文では、四通りの導出法を階段上りについて適用し比較実験を行った。実験は各条件で 5 回ずつ行い、その平均値と分散値によって評価した。実験の結果を **Table 4** に示す。手法 IV を除く三つの手法は、Table 4 (a) に示す平均値に関してほぼ同等の制御性能を示した。そこで各評価値の分散に着目し、評価を行ったのが Table 4 (b) である。Table 4 (b) の σ_{Total} は式 (5) および **Table 5** で示す評価値である。

Table 5 Parameters of evaluate function for variance

Parameter	η_1	η_2	η_3	η_4
Value	10^8	10^{11}	10^9	10^{-3}

Table 6 The estimated gradient of the adjusted controller's hyperplane for the three step stairs

	κ	λ
e	1.4589	-0.0006
$\int e$	0.0266	0.0102
\dot{e}	0.0557	-0.0006
v	-0.6840	0.0002
θ_p	0.0099	-0.0028
$\dot{\theta}_p$	0.0173	-0.0008
θ_f	-0.0162	-0.0006
θ_s	0.0050	0.0001

$$\sigma_{Total} = \sum_i \eta_i \sigma_i \quad (5)$$

手法 III は他の手法と比較して分散値が小さい、つまり不整地走破において安定したパフォーマンスを示したことを意味する。

以上より、簡易制御器の制御パラメータの導出方法として手法 III を採用する。手法 III により得られた制御パラメータを **Table 6** に示す。簡易制御器は得られた制御パラメータと専用制御器の入力の線形結合により、式 (6) で表される。

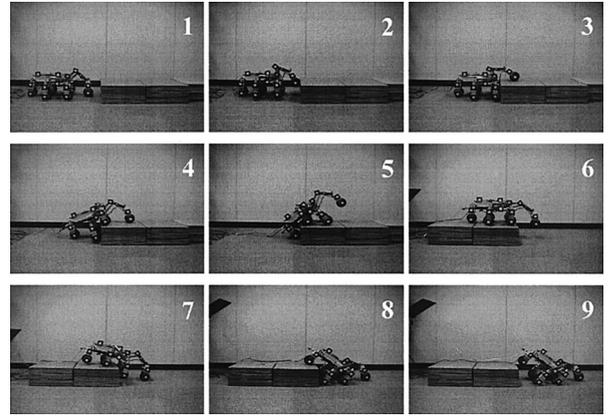
$$\begin{cases} \dot{\omega} = \kappa_P e_v + \kappa_I \int e_v + \kappa_D \dot{e}_v + \kappa_v v + \kappa_{\theta_p} \theta_p + \kappa_{\dot{\theta}_p} \dot{\theta}_p \\ \quad + \kappa_{\theta_f} \theta_f + \kappa_{\theta_s} \theta_s \\ \omega_r = \int \dot{\omega}_r dt \end{cases} \quad (6)$$

3.3 環境認識システム

3.3.1 環境変数の定義

外部環境を認識するための従来手法として、レーザレンジファインダを用いた手法やステレオカメラを用いた手法が提案されている。本論文で提案する環境認識システムは、受動リンク機構が走破面に対して受動的に姿勢を変化させることに着目し、受動リンクのリンク角情報を環境情報と定義する。環境認識システムはあらかじめ基本的な環境情報をデータベース化しておき、移動ロボットから得られる環境情報を基に走破中の環境を認識する。

環境情報として用いるリンク角情報は、フロントフォーク角 θ_f 、サイドリンク角 θ_s 、ピッチ角 θ_p に加え、それぞれの角速度を用いる。データのサンプリング周期を ΔT ($=0.25$ [s]) とし、時刻 t から $t-3\Delta T$ 時刻までの 1 秒間のデータを 1 セットとする。これは 1 ステップ時間のセンサデータは静的な入出力関係のみを表し、移動ロボットのような動的システムに対応できないためである。式 (7) のベクトルは各センサ情報を示す。

**Fig. 10** The attitude of Zaurus climbing over/down the 0.18 [m] height of bump. The numbers show the typical attitude of Zaurus

$$\begin{aligned} \theta_*(t) &= [\theta_*(t) \ \theta_*(t-\Delta T) \ \theta_*(t-2\Delta T) \ \theta_*(t-3\Delta T)] \\ (* &= f, s, p) \end{aligned} \quad (7)$$

環境情報を $\mathbf{x}(t)$ とすると、 $\mathbf{x}(t)$ は各リンク角度と各リンク角速度から構成され、式 (8) で表される。

$$\mathbf{x}(t) = [\theta_f(t) \ \theta_s(t) \ \theta_p(t) \ \dot{\theta}_f(t) \ \dot{\theta}_s(t) \ \dot{\theta}_p(t)] \quad (8)$$

あらかじめデータベース化する環境情報は、平地、0.06, 0.12, 0.18 [m] の段差を走破した際のリンク角情報を用いる。データ取得のための制御器は文献 [14] で示す PID 制御器を用いた。高さ 0.18 [m] の段差昇降の流れを **Fig. 10** に示す。図中の番号は代表的な Zaurus の姿勢を示す。Fig. 10 の 1 が平地、2~5 が段差上り、6~9 が段差下りに対応する。以降、式 (13) で示す環境情報 $\mathbf{x}(t)$ から走破環境の特徴を抽出し環境の分類を行う際に、各番号と姿勢を一致させる。Fig. 10 の 4, 5 において、Zaurus の前輪が宙に浮いてしまっている。これはフロントフォークの機構的な制約によるものであり、今後改善していく必要がある。

環境を分類するための手法として、主成分分析、 k -means 法、自己組織化マップについて比較する。

3.3.2 主成分分析 (PCA: Principle Component Analysis)

主成分分析は、多次元データの情報損失をできるだけ少なくしてデータの縮約をする手法の一つである。PCA による分析結果を **Fig. 11** に示す。横軸は第 1 主成分、縦軸は第 2 主成分である。Fig. 11 (a) より段差昇降における姿勢変化について一定の傾向がみられず、Zaurus の姿勢の分類が困難であることが分かる。Fig. 11 (b) より段差の高さに関して関連性がみられず、環境の分類が困難である。以上より、主成分分析による解析は、Zaurus の走破環境の分類が困難であるといえる。

3.3.3 k -means 法

k -means 法は、代表的な非階層的クラスタ分析手法である [21]。 k -means 法による分析結果を **Fig. 12** に示す。横軸はステップ数、縦軸がクラスタ番号である。図中の番号および帯は Fig. 10 に示す代表的な姿勢の大まかな区間を表す。Fig. 12 (a) は Fig. 10 に示す九つの姿勢ごとにクラスタリングされること

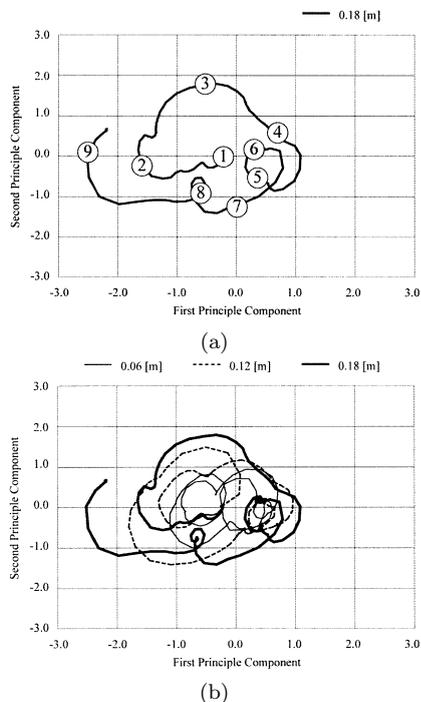


Fig. 11 The results of the PCA. (a) The result of climbing over/down the 0.18 [m] height of bump. (b) The results of climbing over/down the 0.06, 0.12, and 0.18 [m] height of bump

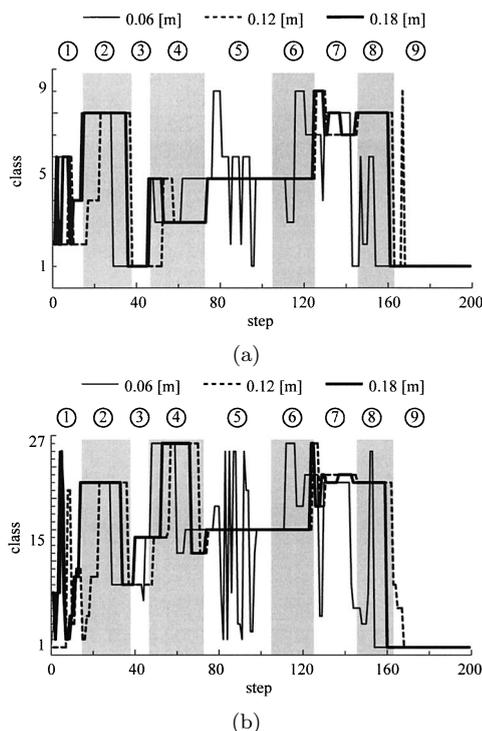


Fig. 12 The results of the k -means method. (a) Clustering number is 9. (b) Clustering number 27

を期待し、クラスタ数を 9 とした。Fig. 12 (b) は九つの姿勢に加え、3 種類の段差高さごとにクラスタリングされることを期待し、クラスタ数を 27 とした。

Fig. 12 (a) より、すべての段差高さのデータが時系列に沿ってほぼ同じクラスタに分類されていることが分かる。したがって、 k -means 法は Zaurus の姿勢変化について分類可能であるといえる。一方 Fig. 12 (b) において、すべての段差高さのデータが時系列に沿って同じクラスタに分類されているため、段差高さについて分類ができていない。以上より、 k -means 法は Zaurus の姿勢変化について分類可能であるが、走破環境の分類が困難であるといえる。

3.3.4 自己組織化マップ (SOM: Self-Organizing Map)

自己組織化マップは、T. Kohonen により提案された教師なし学習を用いたニューラルネットワーク [22] である。自己組織化マップは、位相を保存しつつ写像することでデータ間の類似性を視覚的に捉えるマップを生成する。さらに入力されたベクトルデータを自己組織的に内挿することが可能である。

自己組織化マップは入力層と競合層から構成され、競合層は入力空間の位相を保存しつつ写像された低次元空間である。自己組織化マップのアルゴリズムは、「評価」、「競合」、「協調」、「適応」の四つのプロセスから構成される。競合層の k 番目の参照ベクトル w^k と入力ベクトル x_i のクラス i 間のユークリッド距離を求め (評価)、最小となる勝者ユニット k^* を式 (9) より求める (競合)。

$$k^* = \arg_k \min \|w^k - x_i\|^2 \quad (9)$$

次にそれぞれの参照ベクトルが入力データをどれくらい学習するか (学習率 ψ^k) を式 (10) ~ (12) より求める (協調)。

$$\sigma = \sigma_{\min} + (\sigma_{\max} - \sigma_{\min}) \exp\left(-\frac{t}{\tau}\right) \quad (10)$$

$$\phi_i^k = \exp\left(\frac{-d(k, k^*)}{2\sigma^2}\right) \quad (11)$$

$$\psi_i^k = \frac{\phi_i^k}{\sum_i \phi_i^k} \quad (12)$$

最後に求めた学習率 ψ^k を用いて式 (13) より参照ベクトル w の更新を行う (適応)。

$$w^k = \sum_i \psi_i^k x_i \quad (13)$$

ここで σ は近傍半径を表し、 τ は時定数を表す。

学習後に得られた特徴マップを Fig. 13 に示す。特徴マップのサイズを 30×30 とし、学習回数 1,000 回、最大近傍半径 $\sigma_{\max} = 45$ 、最小近傍半径 $\sigma_{\min} = 2$ 、時定数 $\tau = 50$ とした。本研究では自己組織化マップをオフライン学習しているため、時定数や近傍半径を試行錯誤的に決定したが、今後オンライン調整をする際には、近傍半径を小さくし、時定数がデータ更新のタイミングを決定することを考慮する必要がある。各ユニットは近傍ユニットとの類似度に応じて色づけされている。白色は

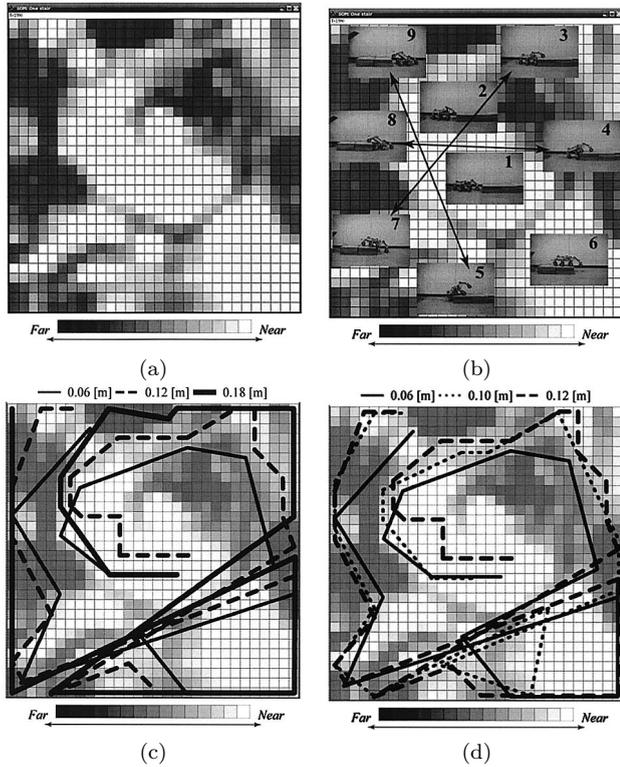


Fig. 13 The results of Self-Organizing Map. (a) The feature map. (b) Classification regard as the attitude of Zaurus. (c) Trajectories of the data set of 0.06, 0.12, and 0.18 [m] height of bumps. (d) Trajectory of the 0.06, 0.10, and 0.12 [m] height of bumps

近傍ユニットのデータベクトルとのユークリッド距離が近いことを示し、色が黒くなるほど距離が遠くなることを示す。

Fig. 13 (b) に Zaurus の姿勢に関する分類を示す。各車輪の段差上りと段差下りの姿勢はマップの対極に配置され、平地が中央に配置された。自己組織化マップは、Zaurus の姿勢ごとに分類が可能である。

Fig. 13 (c) に各段差高さ (0.06, 0.12, 0.18 [m]) の段差昇降データの軌跡を示す。マップの中央に平地が配置され、外周に近づくほど段差の高さが高くなる等高線状の軌跡となっている。つまり自己組織化マップは段差高さについて分類が可能である。

自己組織化マップがもつ補完機能について検証する。未学習環境である高さ 0.10 [m] の段差昇降に関する実験データを自己組織化マップに入力した結果を Fig. 13 (d) に示す。未学習データ (高さ 0.10 [m]) の軌跡は、おおよそ 0.06 [m] と 0.12 [m] の間に配置された。これは未学習データに対して補完機能を有することを示す。

以上より、自己組織化マップは自己組織的に環境情報のクラスタリングを行い、Zaurus の姿勢変化、走破環境の推定が可能であることが分かった。また未学習の環境情報に対しても補完機能によりロボットの姿勢変化、走破環境の推定が可能であることが分かった。したがって本論文では自己組織化マップを環境認識システムとして採用する。

Table 7 Comparison with the experimental results the elementary controller, adjusted controller, simplified controller, and PID controller. (a) The experimental results of climbing over the 0.18 [m] bump. (b) The experimental results of climbing the three step stairs

(a)

Controller	E_v ($\times 10^{-3}$)	$E_{\dot{v}}$ ($\times 10^{-5}$)	$E_{\dot{\theta}_p}$ ($\times 10^{-4}$)	E_J ($\times 10^3$)	E_{Total}
Elementary Ctrl.	2.62	12.7	2.00	7.35	24.67
Adjusted Ctrl.	2.27	5.58	2.19	8.04	18.08
Simplified Ctrl.	2.19	5.01	2.31	6.57	16.08
PID Ctrl.	5.67	4.75	3.03	5.60	19.05

(b)

Controller	E_v ($\times 10^{-3}$)	$E_{\dot{v}}$ ($\times 10^{-5}$)	$E_{\dot{\theta}_p}$ ($\times 10^{-4}$)	E_J ($\times 10^3$)	E_{Total}
Elementary Ctrl.	1.86	6.23	1.76	7.35	11.53
Adjusted Ctrl.	1.53	4.63	1.39	8.04	9.25
Simplified Ctrl.	2.06	6.19	1.63	6.57	11.32
PID Ctrl.	2.87	7.90	2.41	5.60	14.83

4. 提案手法による不整地移動

3章においてニューロコントローラおよび環境認識システムを提案した。本章では、ニューロコントローラ、および環境認識システムを融合した環境適応型制御システムに関する性能評価を行う。

4.1 ニューロコントローラ

提案した制御器 (基本制御器, 専用制御器, 簡易制御器) と従来手法の一つである PID 制御器を用いた不整地移動実験を行う。PID 制御器を式 (14) に示す。ゲインは式 (2) と同様、平地においてステップ応答法を適用し、試行錯誤的な調整を行い決定した ($K_P = 30$, $T_I = 50$, $T_D = 0.01$)。目標速度は $v_r = 0.05$ [m/s] である。

$$\begin{cases} \dot{\omega}_r = K_P \left\{ e + \frac{1}{T_I} \int e dt + T_D \dot{e} \right\}, & e = (v_r - v) \\ \omega_r = \int \dot{\omega}_r dt \end{cases} \quad (14)$$

実験環境を、ニューラルネットワークの学習データ取得に用いた 0.18 [m] の段差上り、および未学習環境である 3 段の階段上りとし、2.2 節で述べた評価関数にて性能評価を行う。実験の結果を Table 7 に示す。Table 7 (a) より、高さ 0.18 [m] の段差乗り越えでは、総合評価において簡易制御器が最も優れた成績となった。また、Table 7 (b) より、未学習環境の 3 段の階段上りについては、総合評価において専用制御器が最も優れた成績となった。

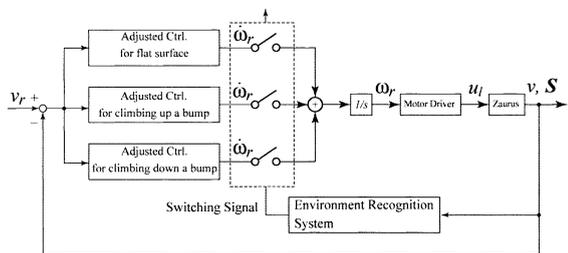


Fig. 14 The block diagram of the adaptive control system

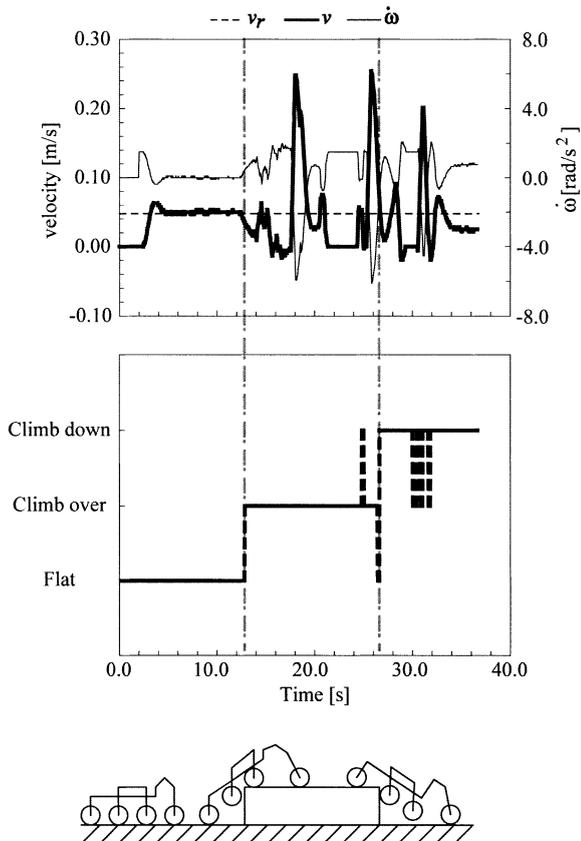


Fig. 15 The experimental results using the adaptive control system. The top graph shows the target velocity, traveling velocity, and adjusted controller's output. The middle graph shows the output of the environment recognition system

4.2 環境適応型制御システム

環境認識システムと専用制御器を併用した環境適応型制御システムを検証する。実験環境は平地、長さ 0.8 [m]、高さ 0.18 [m] の段差から構成される不整地とする。環境認識システムが 3 種類の走破環境（平地、段差上り、段差下り）を認識し、専用制御器を切り替えることで効率的な不整地移動が可能であるか検証する。Fig. 14 に環境適応型制御システムのブロック線図を示す。環境認識システムは環境情報に応じて制御器の切り替えを行う。制御器の切り替え時に制御量の急激な変化を避けるため、コントローラの出力を共通の積分器で積分し、モータドライバへ与える目標角速度 ω_r としている。

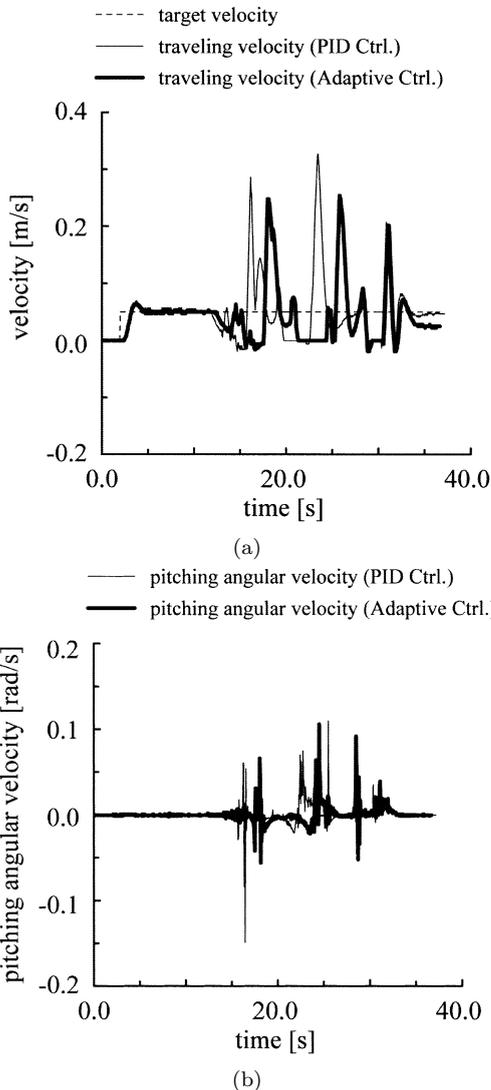


Fig. 16 The experimental results using the adaptive control system and single PID control system

Table 8 The comparison with the adaptive control system and single PID control system

Controller	E_v ($\times 10^{-3}$)	$E_{\dot{v}}$ ($\times 10^{-5}$)	$E_{\dot{\theta}_p}$ ($\times 10^{-4}$)	E_J ($\times 10^3$)	E_{Total}
Adaptive Ctrl.	2.50	5.59	1.17	0.86	10.12
PID Ctrl.	2.82	6.52	1.61	0.72	11.67

実験結果を Fig. 15 に示す。上段のグラフは Zaurus の移動速度および専用制御器からの出力を表す。中段のグラフは環境認識システムの出力を示す。環境認識システムが環境を認識し制御器を切り替えていることが分かる。段差の上で平地を認識しないのは、段差の長さがほぼ Zaurus の全長と等しく、段差上りの状態から段差下りの状態へすぐに遷移するからである。また、25 [s] 以降で見られる環境の誤認識は、Fig. 13 (b)において、段差の上り始めと段差の下り終わりが近くに配置されて

いることに起因する。今後オンライン調整を行うことで改善されると考えている。Fig. 16 に提案手法と式 (12) に示す従来手法による Zaurus の移動速度および車体のピッチ角速度を示す。Fig. 16 (a) に示すように、提案手法は従来手法と比較してオーバーシュートが低減されていることが分かる。それに伴い、Fig. 16 (b) に示すように 16 [s] 付近に見られる振動が低減されている。一方、35 [s] 付近から提案手法は速度追従が実現できていない。これは段差を下りたあと、平地を走行する際の姿勢が未学習の姿勢であるからで、今後追加調整することで改善されると考えている。また 2.2 節で述べた評価関数による評価を Table 8 に示す。総合評価において提案手法は良好な結果を示した。

5. 考 察

提案手法は、従来手法と比較して目標値追従や振動抑制に関して良好な結果を示した。特に専用制御器は追加学習の効果により基本制御器より優れた結果を示した。また簡易制御器は線形近似の影響により専用制御器に劣るものの、従来手法と比較して良好な結果を示した。一方、消費エネルギーに関しては、提案手法の消費エネルギーが従来手法を上回る結果となった。これは目標値追従や振動抑制が消費エネルギーとトレードオフの関係にあるためと考えられる。本論文では、目標値追従を優先した追加学習を行っているため消費エネルギーが増加したと考えられる。また環境認識システムは、環境情報より走破環境を認識し専用制御器の切り替えを実現した。制御器の切り替えを行うことにより、単一の制御器による不整地移動と比較して目標値追従や振動抑制に関して良好な結果を示したといえる。

本論文で提案する環境適応型制御システムは、ニューロコントローラと環境認識システムから構成され、車輪型不整地移動ロボットの制御システムとして有効である。

6. お わ り に

本論文では、車輪型の不整地移動ロボット Zaurus に用いる環境適応型制御システムを提案した。環境適応型制御システムは、走破環境を認識しアプリケーションの切り替えを行う環境認識システムと、環境ごとに調整可能なニューロコントローラから構成される。

環境ごとに調整可能なニューロコントローラは、未学習の環境に対して適応可能な基本制御器、追加学習により制御性能に優れた専用制御器、線形近似により計算コストに優れた簡易制御器から構成される。学習済みの環境および未学習の環境における性能評価実験において、従来手法の一つである PID 制御と比較し、目標値追従および振動抑制に関して良好な結果を示し、総合的な評価により提案手法の有効性を示した。

環境認識システムは、受動リンク機構が走破面に対して受動的に姿勢変化することに着目し、走破環境を示す環境情報としてリンク角情報を用いた。得られた環境情報は、自己組織化マップがもつクラスタリング能力や補完能力によってデータベース化し、学習済みの環境だけでなく未学習の環境についても分類、補完可能であることを確かめた。

環境適応型制御システムは、平地、段差から構成される不整

地走行実験において走破環境を認識し、専用制御器を切り替えることで単一の制御器と比較して目標値追従や振動抑制に関して良好な結果を示し、総合的な評価により提案手法の有効性を示した。

今後の展望として、提案手法のオンライン学習が挙げられる。ニューロコントローラは、計算コストに優れた簡易制御器をマイコンに実装し、上位 PC において専用制御器の更新、簡易制御器の設計を行うという運用が考えられる。また環境認識システムは、様々な環境を走破し未学習の環境情報を取り入れることで、より詳細な環境認識システムとして運用することが可能となると考えられる。

謝 辞 本研究の一部は文部科学省の知的クラスター創成事業 (第 II 期)、および九州工業大学 21 世紀 COE プロジェクト「生物とロボットの織りなす脳情報工学の世界」の支援を受けた。ここに謝意を表す。

参 考 文 献

- [1] A.A.F. Nassiraei and K. Ishii: "Concept of Intelligent Mechanical Design for Autonomous Mobile Robots," *Journal of Bionic Engineering*, vol.4, no.4, pp.281-289, 2007.
- [2] A.A.F. Nassiraei and K. Ishii: "How Does "Intelligent Mechanical Design Concept" Help Us to Enhance Robot's Function?," *Intelligent Unmanned Systems: Theory and Applications, Studies in Computational Intelligence*, vol.192/2009, pp.155-178, 2009.
- [3] A.A.F. Nassiraei, Y. Kawamura, A. Aharary, Y. Mikuriya and K. Ishii: "Concept and Design of A Fully Autonomous Sewer Pipe Inspection Mobile Robot: "KANTARO"," *Proc. of ICRA'07*, pp.136-143, 2007.
- [4] S. Hirose, N. Ootsukasa, T. Shirasu, H. Kuwahara and K. Yoneda: "Fundamental Considerations for the Design of a Planetary Rover," *Proc. of ICRA'95*, pp. 1939-1944, 1995.
- [5] R. Volpe, J. Balaram, T. Ohm and R. Ivlev: "Rocky 7: a next generation Mars rover prototype," *Advanced Robotics*, vol.11, no.4, pp.341-358, 1997.
- [6] S. Hayati, R. Volpe, P. Backes, J. Balaram, R. Welch, R. Ivlev, G. Tharp, S. Peters, T. Ohm, R. petras and S. Laubach: "The Rocky 7 Rover: A Mars Sciencecraft Prototype," *Proc. of ICRA'97*, pp. 2458-2464, 1997.
- [7] T. Kubota, Y. Kuroda, Y. Kunii and I. Nakatani: "Lunar exploration rover: Micro5," *Advanced Robotics*, vol.14, no.5, pp.443-444, 2000.
- [8] Y. Kuroda, T. Teshima, Y. Sato and T. Kubota: "Mobility performance evaluation of planetary rover with similarity model experiment," *Proc. of ICRA'04*, pp.2098-2103, 2004.
- [9] R. Siegwart, P. Lamon, T. Estier, M. Lauria and R. Piguett: "Innovative design for wheeled locomotion in rough terrain," *Robotics and Autonomous System*, vol.40, no.2, pp.151-162, 2002.
- [10] T. Thueer, A. Krebs and R. Siegwart: "Comprehensive Locomotion Performance Evaluation of All-Terrain Robots," *Proc. of IROS'06*, pp.4260-4265, 2006.
- [11] G. Ishigami, A. Miwa, K. Nagatani and K. Yoshida: "Terramechanics-based Model for Steering Maneuver of Planetary Exploration Rovers on Loose Soil," *Journal of Field Robotics*, vol.24, issue 3, pp.233-250, 2007.
- [12] M. Sato and K. Ishii: "A Neural Network Based Control System for a Mobile Robot Employing Link Mechanism," *Proc. of the AIM'05*, pp.426-431, 2005.
- [13] M. Sato and K. Ishii: "A Neural Network Based Controller System for a Wheel Type Mobile Robot," *Brain-Inspired IT II*,

- pp.261–264, Elsevier, 2006.
- [14] M. Sato, A. Kanda and K. Ishii: ‘Performance evaluation of a neural network controller system for a wheel type mobile robot,’ Brain-Inspired IT III, pp.160–163, Elsevier, 2007.
- [15] A. Kanda, M. Sato and K. Ishii: ‘Environment Recognition System Based on Multiple Classification Analyses for Mobile Robots,’ Journal of Bionic Engineering, vol.5, suppl.1, pp.113–120, 2008.
- [16] M. Sato, A. Kanda and K. Ishii: ‘A Controller Design Method Based on a Neural Network for an Outdoor Mobile Robot,’ Journal of Bionic Engineering, vol.5, suppl.1, pp.130–137, 2008.
- [17] 藤井, 浦, 須藤, 石井: ‘自己訓練と学習に基づく海中ロボットの運動制御’, 日本ロボット学会誌, vol.13, no.7, pp.1006–1019, 1995.
- [18] K. Ishii, T. Fujii and T. Ura: ‘On-line Adaptive Method in a Neural Network Based Control System for AUVs,’ IEEE Journal of Oceanic Engineering, pp.221–228, 1995.
- [19] K. Ishii and T. Ura: ‘An adaptive neural-net controller system for an underwater vehicle,’ Control Engineering Practice, vol.8, pp.177–184, 2000.
- [20] 西田, 石井, 古川: ‘水中ロボットにおける自己組織的行動獲得システム (第1報: 自己組織化マップを用いた運動制御システムの提案)’, 日本船舶海洋工学会論文集, vol.3, pp.205–213, 2006.
- [21] A. Tarsitano: ‘A computational study of several relocation methods for k-means algorithms,’ Pattern Recognition, vol.36, issue 12, pp.2955–2966, 2003.
- [22] T. Kohonen: Self-organizing maps. Springer, 2001.



佐藤雅紀 (Masanori Sato)

2007年九州工業大学大学院生命体工学研究科脳情報専攻, 博士後期課程単位取得満期退学, 同年同大学研究員. 博士 (工学). 2008年福岡産業・科学技術振興財団研究員となり現在に至る. 不整地走行ロボットや脳型情報処理技術に関する研究に従事. 計測自動制御学会, IEEE等の会員.

(日本ロボット学会正会員)



神田敦司 (Atsushi Kanda)

2007年九州工業大学大学院生命体工学研究科脳情報専攻, 博士前期課程修了. 同年同大学同研究科同専攻博士後期課程に進学, 在学中. 自律型水中ロボット, 不整地走行ロボット, 脳型情報処理技術に関する研究に従事. (日本ロボット学会学生会員)



石井和男 (Kazuo Ishii)

1996年東京大学大学院工学系研究科船舶海洋工学専攻, 博士後期課程修了, 博士 (工学). 同年九州工業大学情報工学部講師, 同助教授を経て, 現在九州工業大学大学院生命体工学研究科准教授. 自律型水中ロボット, 不整地走行ロボット, 跳躍ロボットの開発, 脳型情報処理技術のロボットへの応用等について研究をすすめている. 日本機械学会, 日本船舶海洋工学会, SICE, IEEE等の会員.

(日本ロボット学会正会員)