

# 人間と計算機の協調による移動ロボット群の制御

細川 翼<sup>\*</sup>・添田 満

Cooperative control of multi mobile robots between the human operator and computer

Tsubasa HOSOKAWA<sup>\*</sup> and Mitsuru SOEDA

## Abstract

In this paper, we propose an agent system to support human operation of multi mobile robots, which is performed by cooperation between the human operator and computer. Specifically, we provide several support functions to the agent system. The first function is to show multi-robots as one robot (virtual robot) to the operator. The second function is to intervene in computer operation to change the formation of multi-robots manually. The third function is to avoid collisions between mobile robots. The last function is to adjust the proportion of autonomous operation by automatic and manual operation. In order to examine the effectiveness of these agent systems, we simulate the proposed system on computer.

**Key words:** Mobile robots, cooperative control, multi-robots, agent system, virtual robot, human operation

## 1. はじめに

単数台の自律ロボットに多くの機能を持たせることは、コスト面や技術面で限界がある。そこで、複数台のロボットが協調して作業を行うことが有効だと考えられる<sup>1)</sup>。

一方、ロボットの自律操作において、予め計算機に与えられた作業情報にない環境変化等が発生し自律制御のみで対応することが困難な場合は、あらゆる変化に柔軟に対応することができる人間がロボットを操作することが好ましいと考えられる<sup>2)</sup>。

そこで、通常は計算機操作による自律制御でロボットを動かし、計算機操作では対応できない事象が発生したときに人間が操作を加え対応する、計算機と人間の協調型操作システムが提案されている<sup>3)4)5)</sup>。

しかし、複数台の自律ロボットが協調して作業を行っている時、計算機の操作では対応できない事象が発生した場合、一人の人間が計算機操作に介入し、複数台のロボットを同時に操作するには相当な熟練を要する。そのため、人間が介入して複数台のロボットを操縦するための操作支援システムが有効だと考えられる。

そこで本研究では、人間と計算機が協調して行う移動ロボット群操作系において人間が自律操作に介入し複数台のロボット（ロボット群）を操作するための支援システムを付加した、計算機・人間協調型ロボット群制御系を提案する。

## 2. システム構成

本研究のシステム構成図を Fig.1 に示す。本研究では、6 台の移動ロボット群を想定してシステムを構築した。本システムでは、通常は各移動型ロボットは自律操作により目標位置の追従を行い、計算機はロボット全体の制御を行う。支援システム（エージェント）は人間オペレータが介入操作する場合、ロボット群全体を 1 つのロボットと見立てた仮想ロボットとして操作画面上に提示する。仮想ロボットとは、オペレータが複数のロボット全体の位置を把握し、群ロボットの操作を簡単に行うことができるようにするため、Fig.2 のように、複数台の移動型ロボットから成る群ロボットを全て覆う

ように外形線で囲み、1 台の大きな移動型ロボットとみなしたものである。ロボット群の各ロボットの移動時における位置により仮想ロボットの形状は変化する。オペレータは状況に応じて仮想ロボットの形状を変えるように操作を加えることができる。

オペレータは仮想ロボットを見ることでロボット群全体を把握し必要に応じて、ジョイスティックを使用して自律操作への操作介入を行う。操作介入が行われた場合、エージェントが状況に応じて手動操作と自律操作の割合を調整したうえで、両指令を融合し、オペレータが指示した仮想ロボットの動作になるように各ロボットに指令を分配し、ロボット群の隊形変更や障害物回避などの移動動作を行う。各ロボットは、Fig.3 に示すような 2 輪駆動 1 キースタ方式の移動型ロボットである。このシステムでは、オペレータがエージェントから提示される仮想ロボットに対して操作を加えることにより群ロボット全体を操作する事ができる。

## 3. 仮想ロボットの制御

### 3.1 仮想ロボットの自律操作

本研究では、ロボットの自律操作において、ロボットの並進速度を決定するためと、ロボットの左右車輪の速度を決定するために簡略ファジィ推論を用いている。

ロボットから目標地点までの距離を  $r$  とする時、以下のようルール化した。但し、 $dv$  は速度修正量である。

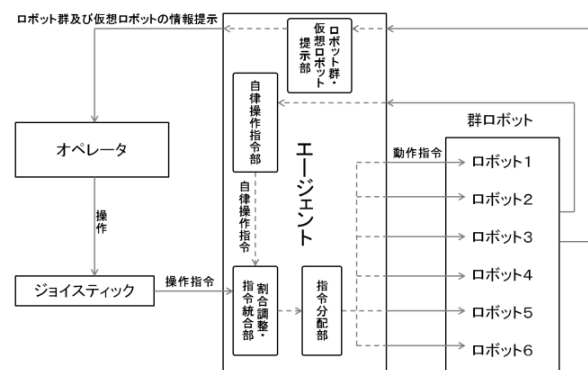


Fig.1 システム構成図

if  $r$  is  $A_i$  and  $V$  is  $B_i$  Then  $dv$  is  $C_i$   
 この時,  $A_i$ ,  $B_i$ はファジイ集合,  $C_i$ は後件部のメンバーシップ関数のシングルトンの実数値である。仮想ロボットの並進速度  $V$  は,

$$V = V + V_{\max} \cdot dv \quad (1)$$

で表せる。但し,  $V_{\max}$ は車輪の最大速度である。

また, 仮想ロボットの進行方向を基準とした時の目標地点の角度を $\varphi$ , ロボットの角速度を $\omega$ とする時, 角速度の修正量  $d\omega$ を以下のようにルール化した。

if  $\varphi$  is  $A_i$  and  $\omega$  is  $B_i$  Then  $d\omega$  is  $C_i$   
 メンバシップ関数とファジイルールは, 並進速度を決定する際と同様に定める。最終的に左右の車輪速度  $V_R$ ,  $V_L$ は式(1)を踏まえてそれぞれ,

$$V_R = V + (\omega + \omega_{\max} \cdot d\omega) \cdot d \quad (2)$$

$$V_L = V - (\omega + \omega_{\max} \cdot d\omega) \cdot d \quad (3)$$

となる。このとき,  $\omega_{\max}$ は角速度の最大値,  $d$ は仮想ロボットの中心から車輪までの距離である。

### 3. 2 仮想ロボットの手动操作

本システムでは, オペレータがジョイスティックを使用して, 仮想ロボットを手动操作することができる。ジョイスティックの操作により指令を与えることにより, 自律操作の時と同様に簡略ファジイ推論で, ロボットの並進速度と角速度を更新する。ジョイスティックを前に倒すと前進, 左右に倒すと方向転換というような, 人間が直感的に操作し易いシステムになっている。

## 4. 群ロボットの制御

### 4. 1 群ロボットの目標位置追従

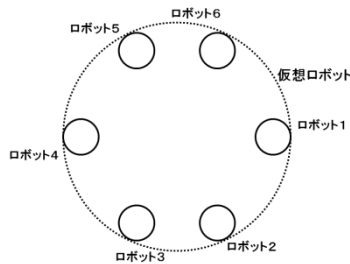


Fig.2 仮想ロボット

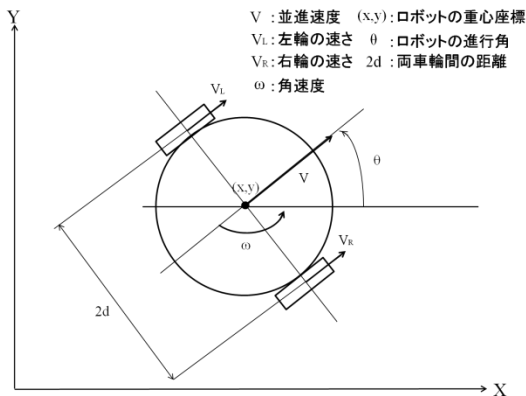


Fig.3 移動型ロボット

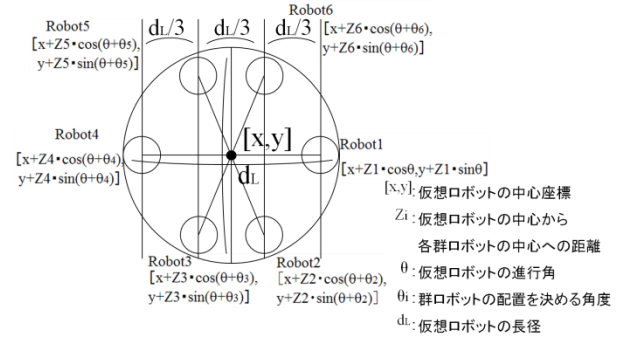


Fig.4 群ロボットの配置

群ロボットの各ロボットは, エージェントから指示される仮想ロボットの外形線に包含される目標位置に自律移動する。各ロボットの目標位置の決定は以下のとおりに行う。

エージェントは自律操作や手动操作指令をもとに, 仮想ロボットの移動及び形状変更を指示する。移動した仮想ロボットの重心点から Fig.4 のようにロボット群の各ロボットの目標位置を決定し, それぞれ目標位置に追従するように指令を出す。Fig.4 において,  $Z_i$ は, 仮想ロボットの重心座標から各群ロボットの中心座標の距離を導出したものである。

仮想ロボットは楕円形に形状変更が可能であり,  $\theta_1 = 0$ ,  $\theta_4 = \pi$ で, それ以外の  $\theta_i$  は, 仮想ロボットの径のうち, 長径  $d_L$ を3分割するような, 群ロボットの配置になるように導出した角度である。実際の  $\theta_i$  や  $Z_i$  は, 以下の式で求める。但し,  $d_L$ は仮想ロボットの長径,  $d_2$ は仮想ロボットの短径を示している。

$$\theta_2 = \tan^{-1} \left( \frac{-d_L/3}{(d_2 \cdot \sqrt{1-(1/9)})} \right) \quad (4)$$

$$\theta_3 = \tan^{-1} \left( \frac{-d_L/3}{-(d_2 \cdot \sqrt{1-(1/9)})} \right) \quad (5)$$

$$\theta_5 = \tan^{-1} \left( \frac{d_L/3}{-d_2 \cdot \sqrt{1-(1/9)}} \right) \quad (6)$$

$$\theta_6 = \tan^{-1} \left( \frac{d_L/3}{d_2 \cdot \sqrt{1-(1/9)}} \right) \quad (7)$$

$$Z_i = \sqrt{\left( d_1^2 / \left( 1 + \frac{(d_1 \cdot \tan \theta_i)}{d_2} \right) \right)^2 + \left( d_2^2 / \left( 1 + \frac{(d_2 \cdot \tan \theta_i)}{d_1} \right) \right)^2} \quad (8)$$

### 4. 2 群ロボットの隊形変更

通常, ロボット群は円形の隊形で移動を行う。しかし, 障害物を回避する際や狭路を進行する際などには, 隊形を変更した方が効率的に目標位置を目指す場合があると考えられる。本研究では, 様々な状況に応じてオペレータが隊形変更を行い, 効率的に目標位置の追従を行えるようなシステムを構築した。具体的には, オペレータがジョイスティックの4つのボタン, (ボタン1~ボタン4)を押すことで, 仮想ロボットの形状を変化させ, その形状を構成するための各ロボットの位置をエージェントが割り出して, その位置に各ロボットを移動させることにより隊形変更を行う。ボタン1と2が, 仮想ロボットの縦の径を大きくするボタンと小さくするボタン, ボタン2と3が, 横の径を大きくするボタンと小さくするボタンになっている。さらに, Fig.5に示すように, 仮

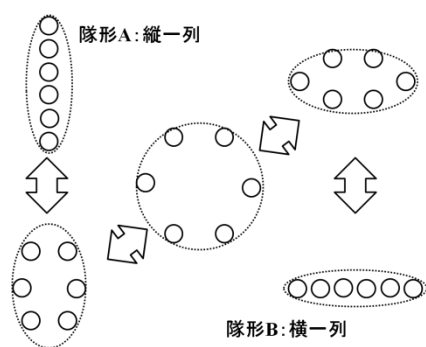


Fig. 5 群ロボットの隊形変更

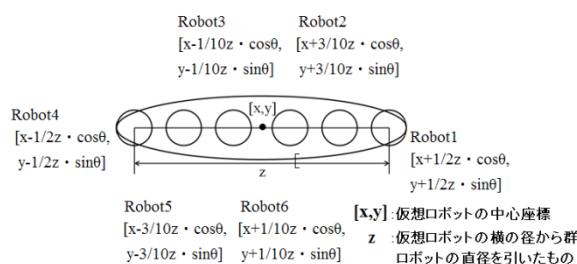


Fig. 6 横一列の場合の配置

想ロボットの各径が一定の値 $\epsilon$ 以下になった場合、縦一列の隊形 A や横一列の隊形 B に変更するようになっている。その際の各群ロボット目標位置は、等間隔な横一列もしくは縦一列の配置になるようにエージェントが決定する。横一列の配置の場合、各群ロボット目標位置は Fig.6 のように等間隔な横一列の配置になるように決定する。縦一列の場合も同様に仮想ロボットの中心座標から各群ロボットの目標位置を決定する。

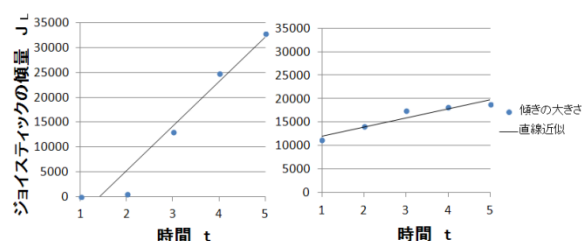
障害物が多く存在する場所を通過したい場合は、仮想ロボットの径を小さくすることで容易に通過することが可能になる。狭路を通過したい場合は、横一列の隊形に変更することで通過が容易になる。また、縦一列の隊形は、群ロボットが物を押す際などに有効な隊形である。

## 5. 自律操作と手動操作の合成

### 5. 1 自律操作と手動操作の割合決定

本研究では、通常、群ロボットは自律操作で目標追従を行い、動作環境など作業情報にはない状況変化が発生し、自律だけでは対応できない場合などに、オペレータが介入操作を行い、様々な状況に柔軟に対応できるシステムの構築を目指している。本システムでは、自律操作によって決定された速度  $\vec{V}_a$  と、オペレータが必要に応じて自律動作に介入し加えた手動操作によって決定された速度  $\vec{V}_m$  を合成して、実際のロボットの速度  $\vec{V}$  を決定する。

オペレータが操作介入する場合、効率的な操作を実現するには、操作介入をどの程度反映させるかということを考慮する必要がある。例えば、急激な状況変化が発生し、進行方向を大きく変化させたい場合などは、自律操作による速度  $\vec{V}_a$  を小さく、手動操作による速度  $\vec{V}_m$  を大きくして、操作介入を大きく反映させる必要がある。逆に、前方にセンサで感知できない小さな障害物がある場合などは、大きく迂回させる必要



(a) 素早く倒した場合 (b) 緩やかに倒した場合

Fig. 7 過去5回の操作入力の直線近似

Table. 1 操作割合決定のためのファジイルール

		傾き $J_L$		
		ZO	PS	PB
傾きの大きさ	ZO	PB	PB	PB
	PS	PS	NS	NS
	PB	ZO	NS	NB

はないため、自律操作による速度  $\vec{V}_a$  は大きく、手動操作による速度  $\vec{V}_m$  は小さくて良い。実際の速度決定は以下の式で行う。ここで、 $\alpha$  は人間と計算機の操作割合の調整変数である。

$$\vec{V} = \alpha \vec{V}_a + (1 - \alpha) \vec{V}_m \quad (9)$$

本システムでは、ジョイスティックの傾き量  $J_L$  とジョイスティックの傾き量の変化速度  $J_C$  を入力として、簡略ファジイ推論を用い、自律操作を反映させる割合を決定する変数  $\alpha$  を求めた。ジョイスティックを大きく傾けた場合、自律操作の割合を小さくし、急な状況変化が発生しオペレータが素早くジョイスティックを操作した場合も、自律操作の割合を小さくするようなルールを考えた。

まず、ジョイスティックの傾き量の変化速度  $J_C$  について説明する。オペレータの加える過去 5 時点のジョイスティックの入力の値をとり、Fig.7 のように縦軸をジョイスティックの傾き量  $J_L$ 、横軸を時間  $t$  とするグラフを考える。過去 5 回分の入力に対して、最小二乗法を用いて直線近似し、その傾きの大きさを  $J_C$  とした。Fig.7 の(a)が素早くジョイスティックを倒した場合で、(b)がゆるやかにジョイスティックを倒した場合の図である。素早く倒せば倒すほど、近似した直線の傾きは大きくなるため、 $J_C$  も大きくなるのがわかる。

自律操作の割合を決定する関数  $\alpha$  の修正量  $d\alpha$  を求めるファジイルールを以下に示す。

If  $J_L$  is  $A_i$  AND  $J_C$  is  $B_i$  Then  $d\alpha$  is  $C_i$

Table.1 にファジイルールを示す。以上より導出した  $d\alpha$  により  $\alpha$  を更新することで、自動で自律操作と手動操作の割合を決定できるようにした。

### 5. 2 オペレータによる割合調整補助

自律操作の割合を決定することができるようにしたが、先のシステムでは、例えば、多くの障害物が存在し細かな回避行動をとりたい時などのように、自律操作の割合は小さくしたいが、介入操作も大きくしたくない場合に対応することができない。そこで、本システムでは、手動操作の大きさをさらに調整することを可能にするため、式(9)に調整変数  $\beta$  を加えて、実際の仮想ロボットの速度の融合ベクトルを以下の式により決定する。

$$\vec{V} = \alpha \vec{V}_a + (1 - \alpha) \beta \cdot \vec{V}_m \quad (10)$$

ここで  $\beta$  は、オペレータがジョイスティックの操作レバーの横にある、スライド式つまみを動かすことにより調整できるようにした。

## 6. シミュレーション

Fig.3で示す2輪駆動型移動ロボットをコンピュータ上に作成し、動作環境を設定し、オペレータと計算機によりこれらの6台のロボット群の操作実験を行った。

### 6.1 自律操作と手動操作の割合調整

#### 6.1.1 大きく障害物を迂回する際の有効性

自律操作と手動操作の融合において、自律操作と手動操作の割合を自動で決めるシステムの有効性を検証するために、割合調整機能を導入していないシステムとの操作性の比較を行った。まず、割合調整機能を有していないシステムとして、操作介入が無い場合、式(9)の $\alpha = 1$ 、介入があった場合 $\alpha = 0.5$ とするものを用意した。 $\alpha = 0.5$ としたのは、自律操作による速度ベクトルと、手動操作による速度ベクトルを単純に足し算したのでは速度の最高値を超えることがあるので、それぞれのベクトルを半分の大きさにしてそれを防いだためである。このシステムと、 $\alpha$ を自動で調整できるシステムとで比較を行う。

Fig.8のような、塗りつぶされた円で示された、ロボットに認識できない大きな障害物が存在し、人間の介入操作により障害物を大きく迂回することで回避しなければならない環境を設定し比較を行った。Fig.8(a)は割合調整機能を有していないロボットの移動平面、Fig.8(b)は割合調整機能を実装したロボットの移動平面の図である。複数の円は仮想ロボットの

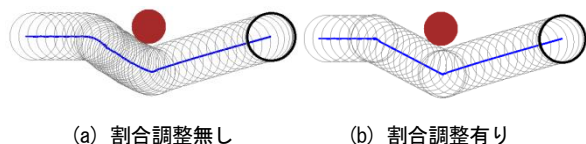


Fig. 8 大きな障害物の回避

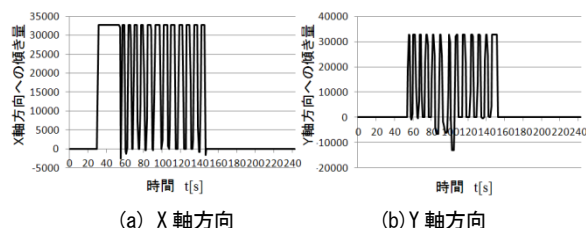


Fig. 9 割合調整が無い場合の操作入力

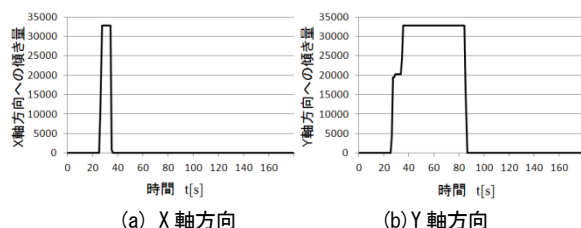


Fig. 10 割合調整が有る場合の操作入力

移動の軌跡を表わしており、実線は仮想ロボットの重心の軌跡を表わしている。また、各群ロボットの軌跡は図を見やすくするために省略している。ロボット群は自律操作指令のもとに太い線で描かれた円で示された目標点を目指し直進するが、障害物を回避するために、割合調整有りの場合も無しの場合も、どちらもほぼ同じ軌道を描くようにオペレータが操作介入を行い障害物の回避を行った。

Fig.9は、オペレータが割合調整機能を有していないシステムに加えたオペレータのジョイスティックへの入力、Fig.10は、割合調整機能を実装したシステムに加えたジョイスティックへの入力である。どちらもほぼ同じ軌跡を移動したにも関わらず、調整機能を実装していない方は、ジョイスティックをx,y軸方向に何度も傾けて操作しているのに比べて、自動調整機能を実装している方は、あまり複雑な操作をしていないことがわかる。目標点に到達するまでにかかった時間は、調整機能無しのシステムが242[s]にかかったのに対して、調整機能有りのシステムは169[s]であった。

以上より大きく障害物を迂回する際に自動割合調整機能が有効であることがわかる。

#### 6.1.2. 小さな介入操作を加える際の有効性

次に、割合調整機能を有していないシステムとして、操作介入が無い場合、式(9)の $\alpha = 1$ 、介入があった場合 $\alpha = 0.5$ とするものを用意した。このシステムと自動での割合調整機能を有しているシステムとで比較を行った。Fig.11のような、計算機に認識できない小さな障害物が存在し、それを回避するためにオペレータが介入し、少しだけロボット群の進行方向を変えなければならない環境を設定し比較を行った。6.1.1と同様 Fig.11(a), (b)に示すように、割合調整無しの場合と有りの場合がほぼ同じ軌道を描くように操作を加えて障害物回避を行った。

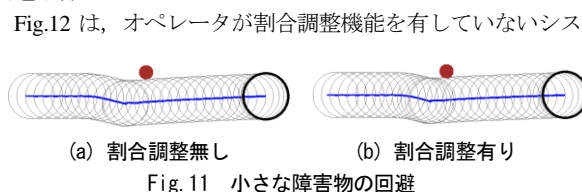


Fig. 11 小さな障害物の回避

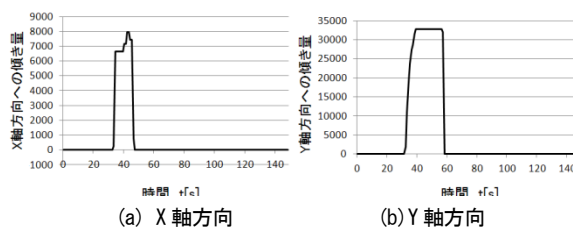


Fig. 12 割合調整が無い場合の操作入力

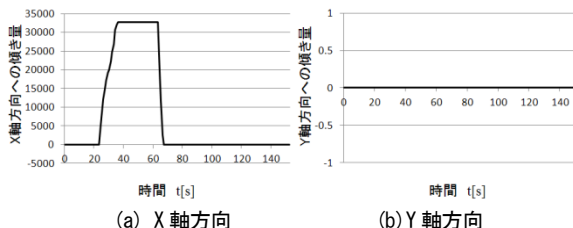


Fig. 13 割合調整が有る場合の操作入力



テムに加えたジョイスティックの入力, Fig.13 は割合調整機能を実装したシステムに加えたジョイスティックの入力である. Fig.13 を見てわかるように, 割合調整機能を有する場合は, ゆっくりとジョイスティックを x 軸方向に倒すだけで良い. それに対して割合調整が無い場合は, Fig.12(a)を見るとオペレータが x 軸方向に細かな操作を加えており, さらに Fig.12(b)をみると y 軸方向にも操作を加えていることがわかる. どちらもほぼ同じ軌道を描くように操作しているにも関わらず, 割合調整機能が有る方が単純な介入操作で, 障害物の回避を行えていることがわかる. これらから小さく障害物を迂回する際にも割合調整機能が有効であることがわかる.

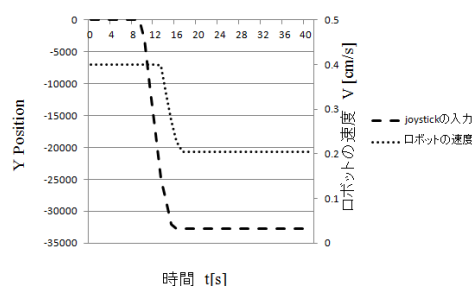
以上より, 自律操作と手動操作の操作分担割合を自動で調整する支援機能を有する操作系が有効であることが分かる.

## 6.2 手動操作の大きさの調整

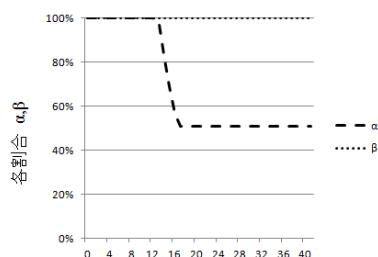
本研究では, 手動操作の大きさを決める変数 $\beta$ をスライド式つまみで調整できる. このシステムの有効性を検証するために, 手動操作の大きさ調整機能が無いシステムとして, 式(10)の  $\beta = 1$  とするものを用意し比較を行う.

ロボットが狭路や障害物が多く点在する場所等を走行する場合, ロボットが障害物等に衝突することを防ぐために徐行する必要がある. このシステムのロボットの速度を半分の速度に落とす場合に, 手動操作の大きさ調整機能が有るロボットと無いロボットとでは操作にどのような差があるか比較を行った.

Fig.14 及び Fig.15 が比較を行った結果である. それぞれ, オペレータがジョイスティックに加えた操作力, ロボットの速度変化, 変数  $\alpha$  と  $\beta$  の変化を示している. 大きさの調整機能が無い場合は, Fig.14 (a)をみてわかるように, 速度を落とすためにジョイスティックを後ろ向きにゆっくりと倒している. ジョイスティックを傾ける速さによって  $\alpha$  の値が変わって, それによりロボットの速度も変化するので, 速度を 0.2 に調整するのは難しい. さらに, 進行方向を変更する際は, ジョイスティックを後ろ向きに傾けた状態のまま行



(a) 操作入力と速度



(b) 変数  $\alpha$  と  $\beta$

Fig. 14 大きさ調整機能が無い場合

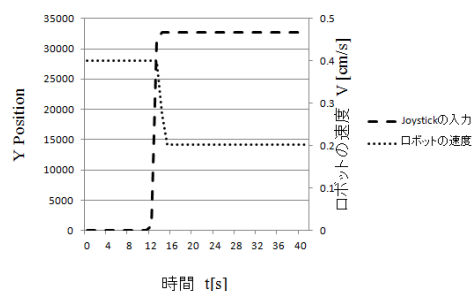
なければならない. それに対して Fig.15 の大きさ調整機能がある場合は, (b)を見て解るように,  $\beta$ の値をスライド式つまみで 0.5 に調整し, (a)のようにジョイスティックを前に素早く倒すだけで, 速度を 0.2 にできる. 以上より, 手動操作の大きさ調整機能を付加することにより, 介入操作による速度調整が行いやすくなり, 本調整機能が有効性であることが分かる.

## 6.3 障害物回避および狭路の通過

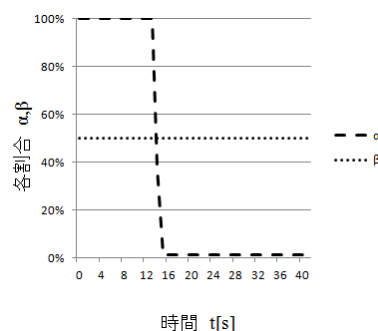
群ロボットの操作性を検証するために以下の実験を行う.

Fig.16 のような, 塗りつぶされた円で示される小さな障害物がロボットの進行方向にあり, さらにその先に狭路が存在し, その中を通過しなければならない作業条件を設定した. ロボット群は目標位置である太い線で描かれた円を目指して自律移動するが, 障害物及び狭路は, 計算機に認識できないものと設定した. そのため自律操作では対応できなくなり, オペレータの操作介入が必要になる. この環境で, 隊形変更支援機能のみを実装しているシステムと, 人間と計算機の操作割合調整支援機能も実装しているシステムで操作実験を行った. Fig.16(a)は, 隊形変更の支援機能のみを実装している場合, (b)は割合調整支援機能も実装している場合の移動平面である. 小さな複数の円が, 各群ロボットの移動の軌跡を表わしている. どちらも障害物を回避し, 狭路を通過して目標位置に到達していることがわかる. また, (a)の場合は, オペレータが介入し障害物を大きく迂回させて回避しているが, (b)は障害物を大きく迂回することなくスムーズに回避させていることがわかる.

Fig.17 は隊形変更支援機能のみを有する場合, Fig.18 は隊形変更と割合調整支援機能を実装しているシステムでの操作によるロボットの位置, 速度, オペレータの介入時における操作入力の状況を示している. それぞれの図の(a), (b)を比較すると, どちらも仮想ロボットを基準として各ロボットが



(a) 操作入力と速度



(b) 変数  $\alpha$  と  $\beta$

Fig. 15 大きさ調整機能がある場合

移動出来ていることがわかる。

Fig.17, Fig.18 の(c)を比較すると、全機能を有する支援システムの場合は狭路に入る前の80~160[s]付近で、速度を落として徐行が出来ていること、160~210[s]付近の、狭路を通過時に速度が安定していることがわかるが、隊形変更支援機能のみの場合は、徐行ができておらず、狭路でも速度が安定していない。さらに、各図の(d)を見比べると、計算機と人間の操作割合調整機能が含まれないシステムではオペレータが細かな介入操作を行っており、割合調整機能を有する支援システムの場合はオペレータの操作が単純であり、オペレータの介入操作が簡単に行われていることが分かる。

以上から、提案する隊形変更と両操作の操作割合調整の支援機能を有する人間と計算機の協調操作システムにより効率的なロボット群の操作を行えることがわかる。

## 7. おわりに

本研究では、6台の移動型ロボットを対象として、人間と計算機によるロボット群の協調操作システムを構築した。

本システムでは、計算機による操作に人間が介入し操作を加える場合、ロボットの操作を簡略化するために6台のロボットを1台の仮想ロボットとしてオペレータに提示し、ファジィ制御で目標位置追従を行う。また、オペレータが仮想ロボットの形を変化させれば、支援システムにより各ロボットに指令が分配され、各ロボットが移動してロボット群の隊形を変更させることができる。さらに、オペレータが介入して計算機と協調して操作を行う場合の操作分担の割合を移動環境・状況に応じて調整する支援機能を付加した。

これらの支援機能を有した人間と計算機の協調操作システムの総合的な操作実験を行った結果、提案する操作支援機能を加えることにより、人間と計算機の協調操作がより効率的に行えることを確認した。

以上から、本システムにより様々な動作環境・状況の変化にも柔軟に対応してより効率的にロボット群の操作を行うことが可能となる。

## 参考文献

- 1) 根, 福島, 松野: 予測時刻間の衝突回避を考慮した複数移動体のモデル予測編隊制御, 計測自動制御学会論文集, Vol.46 No.7, 383/390 (2010)
- 2) Thomas B. Sheridan: Telerobotics, Automation, and Human Supervisory Control, The MIT Press (1992)
- 3) 平井: Shared Autonomy の理論, 日本ロボット学会誌, Vol.11 No.6, 788/793 (1993)
- 4) M. Soeda, T. Furuya: Adaptive Master-Slave Arm Control with Computer Auxiliary Operation by RBFN, Proceedings of The 7th IEEE International Conference on Methods and Models in Automation and Robotics, 437/442 (2001)
- 5) 添田他: IT 技術を用いた計算機・人間協調型防犯防災用警備ロボットの開発, 環境資源工学会誌, 第52巻第4号, 215/218 (2005)

(2013年11月11日 受理)

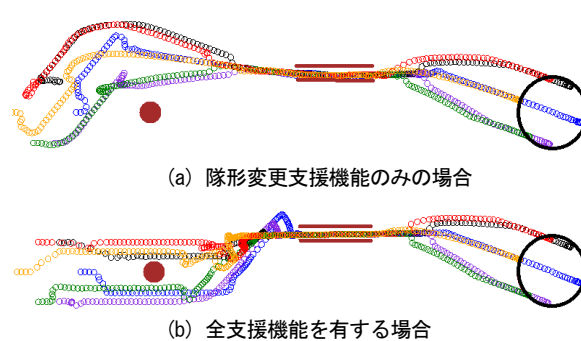


Fig. 16 実験環境と移動平面

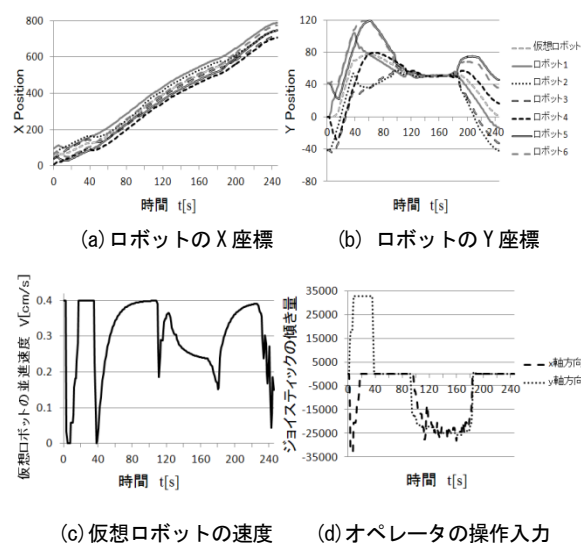


Fig. 17 隊形変更支援機能のみの場合

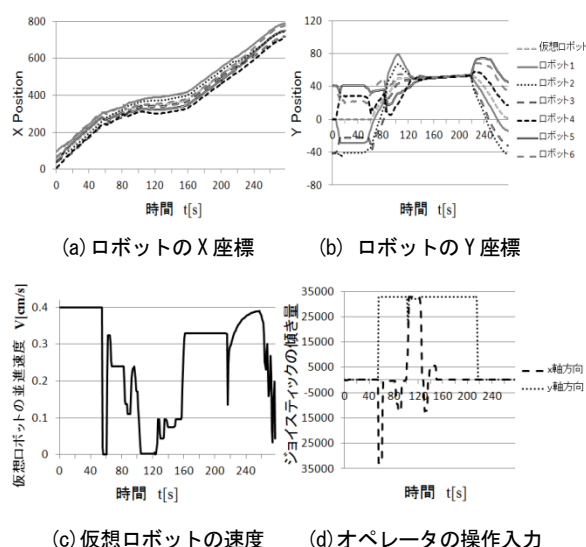


Fig. 18 全支援機能を有する場合