

# Multi-Robot Coordination Using Switching of Methods for Deriving Equilibrium in Game Theory

Wataru Inujima<sup>1</sup>, Kazushi Nakano<sup>2</sup>, and Shu Hosokawa<sup>3</sup>, Non-members

## ABSTRACT

The study of a Multi-Agent System using multiple autonomous robots has recently attracted much attention. With the problem of target tracking as a typical case study, multiple autonomous robots decide their own actions to achieve the whole task which is tracking target. Each autonomous robot's action influences each other. So, an action decision in coordination with other robots and the environment is needed to achieve the whole task effectively.

The game theory is a major mathematical tool for realizing a coordinated action decision. The game theory mathematically deals with a multi-agent environment influencing each other as a game situation. The conventional methods model one of the target tracking as a  $n$ -person general-sum game, and the use of the non-cooperative Nash equilibrium theory in non-cooperative games and the semi-cooperative Stackelberg equilibrium. The semi-cooperative Stackelberg equilibrium may obtain better control performance than the non-cooperative Nash equilibrium, but requires the communication among robots.

In this study, we propose switching of methods in the equilibrium derivation both from the non-cooperative Nash equilibrium and the semi-cooperative Stackelberg equilibrium in a coordination algorithm for the target tracking. In the simulation, our proposed method achieves coordination with less connections than the method using the semi-cooperative Stackelberg equilibrium at all times. Furthermore, the proposed method shows better control performance than the non-cooperative Nash equilibrium.

**Keywords:** Multi-Agent, Coordination Control, Robot, Game Theory

## 1. INTRODUCTION

Recently, robots following people to support carrying baggage has been studied. Using multi-robot, it is expected that an entire performance of achieving the whole task is improved with cooperating individual robot whose performance and costs are low. For instance, robots in cooperation can carry heavy baggage which a single robot cannot carry.

Thus, the study of a Multi-Agent System (MAS) using multiple autonomous robots is actively being increased. With the problem of target tracking as a typical case study, the competitive situation to prevent achieving the whole task by the robot collision with a tracked target robot, obstacles, and other robots is assumed. Therefore, a coordination control to solve these is needed.

The game theory is the method realizing a coordinated action decision of each robot, and deals with a multi-agent environment in which multiple autonomous agents decide their own actions to achieve the whole task and these action decisions influence each other [1].

In the game theory, the situation assuming a non binding agreement among agents is called "non-cooperative game". In contrast, the situation assuming binding an agreement among agents is called "cooperative game" [2].

Skrzypczyk [3][4][5] introduces the cost value as a method using the game theory for the target tracking problem, and models the target tracking problem as "n-person general sum game". The cost value indicates a burden to achieve the task based on the position of each tracking robot, a tracked target robot and obstacles.

For deciding an action of the robot in coordination for the target tracking, Skrzypczyk [3] presents a method using "Nash equilibrium" based on the non-cooperative game theory. On the other hands, Harmati [4] proposes "Stackelberg equilibrium" based on a type of cooperative game.

In the coordination using the Nash equilibrium, tracking robots as agents do not have communication mechanism in a tracking team, and robots decide their own action by using only cost values. In the coordination using the Stackelberg equilibrium, tracking robots have communication mechanism and exchange information in a tracking robot team. Robots obtain the agreement of an action to guarantee an ac-

Manuscript received on July 20, 2013 ; revised on January 6, 2014.

Final manuscript received May 3, 2014.

<sup>1,2</sup> The authors are with Dept. of Mechanical Eng. and Intelligent Systems, The University of Electro-Communications 1-5-1, Chofugaoka, Chofu, Tokyo, 182-8585, Japan, E-mail: inujima@francis.ee.uec.ac.jp and nakano@ee.uec.ac.jp

<sup>3</sup> The author is with Dept. of Electronic Eng., The University of Electro-Communications 1-5-1, Chofugaoka, Chofu, Tokyo, 182-8585, Japan, E-mail: hosokawa@francis.ee.uec.ac.jp

tion decision of a robot whose cost value is relatively large. By the action decision under the assumption of this agreement, the Stackelberg equilibrium derives a better balance among cost values than the Nash equilibrium [5].

Therefore, it is expected that the appearance of the remarkably poor performance robot is suppressed. The Stackelberg equilibrium earns a lower cost value than the Nash equilibrium, and it is expected to obtain better performance to achieve the task than the Nash equilibrium.

For these reasons, we propose a switching method for equilibrium derivation.

Our proposed method aims to coordinate the Nash equilibrium with the Stackelberg equilibrium that needs communication in the situation that only the Nash equilibrium that needs no communication is difficult to achieve the task.

## 2. FORMULATION OF PROBLEM

This chapter defines various information and estimated values that a control algorithm deals with. To deal with the target tracking problem for the game theory, we present a proposal for the target tracking by cost values.

### 2.1 Definition of Target Tracking and State Estimation

With the target tracking in this study, there are a single tracked target robot, a robot team constructed by  $N$  autonomous robots following a tracked target robot, and  $M$  static obstacles. The positions of each robot and obstacle are shown in Fig. 1. Each robot is cylinder in shape and has radius  $D_r$ .

The task of target tracking is that each autonomous robot in a tracking robot team achieves collision avoidance with the other tracking robots, a tracked target robot, and obstacles, while keeping a desired formation. For the target tracking, rotational angular velocity and translational velocity are controlled by a control algorithm. All tracking robots can obtain the positions of all robots and obstacles by a camera mounted in an environment.

The position  $\mathbf{p}_i$  of an autonomous robot  $i$  ( $i = 1, 2, \dots, N$ ) in the tracking robot team, the position  $\mathbf{g}$  of a tracked target robot, and the position  $\mathbf{o}_j$  of an obstacle  $j$  ( $j = 1, 2, \dots, M$ ) at discrete time  $t_n$  obtained from a camera are defined in Eqs. (1)-(3).

$$\mathbf{p}_i(t_n) = \begin{bmatrix} x_i(t_n) & y_i(t_n) \end{bmatrix}^T \quad (1)$$

$$\mathbf{g}(t_n) = \begin{bmatrix} x_g(t_n) & y_g(t_n) \end{bmatrix}^T \quad (2)$$

$$\mathbf{o}_j(t_n) = \begin{bmatrix} x_j & y_j \end{bmatrix}^T \quad (3)$$

The positions of obstacles are constant.

The position  $\mathbf{c}(t_n)$  shown the center of a tracking

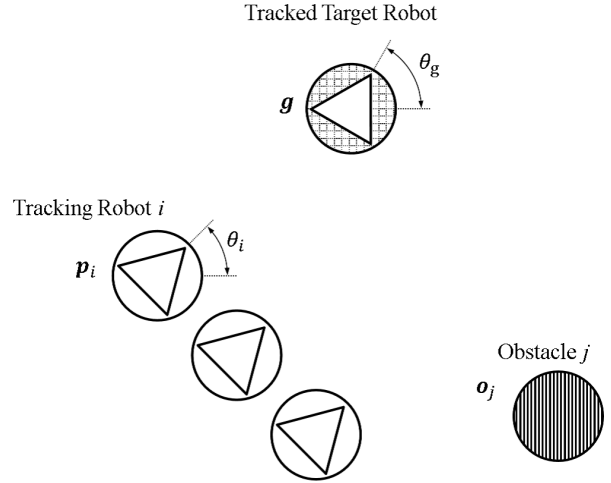


Fig.1: Position Relation

robot team is defined by

$$\mathbf{c}(t_n) = \left[ \frac{1}{N} \sum_{i=1}^N x_i(t_n) \quad \frac{1}{N} \sum_{i=1}^N y_i(t_n) \right]^T. \quad (4)$$

The estimation is needed in order to obtain a route direction, translational velocity, angular velocity and various information of robots that will be observed at  $t_{n+1}$ .

At discrete time  $t_n$ , a robot  $i$  estimates the translational velocity  $\hat{v}_g(t_n)$  of a tracked target robot from Eq. (5) using the position obtained by a camera.

$$\begin{aligned} \hat{v}_{g,x}(t_n) &= \frac{x_g(t_n) - x_g(t_{n-1})}{t_n - t_{n-1}} \\ &= \frac{x_g(t_n) - x_g(t_{n-1})}{\Delta t} \\ \hat{v}_{g,y}(t_n) &= \frac{y_g(t_n) - y_g(t_{n-1})}{t_n - t_{n-1}} \\ &= \frac{y_g(t_n) - y_g(t_{n-1})}{\Delta t} \\ \hat{v}_g(t_n) &= \sqrt{(\hat{v}_{g,x}(t_n))^2 + (\hat{v}_{g,y}(t_n))^2} \quad (5) \end{aligned}$$

The estimated position  $\hat{\mathbf{g}}(t_{n+1}) = [\hat{x}_g(t_{n+1}) \quad \hat{y}_g(t_{n+1})]^T$  after one control period is obtained by

$$\begin{aligned} \hat{x}_g(t_{n+1}) &= x_g(t_n) + [\hat{v}_{g,x}(t_n) \Delta t + x_g(t_n)] \\ &= \hat{v}_{g,x}(t_n) \Delta t + x_g(t_n) \\ \hat{y}_g(t_{n+1}) &= y_g(t_n) + [\hat{v}_{g,y}(t_n) \Delta t + y_g(t_n)] \\ &= \hat{v}_{g,y}(t_n) \Delta t + y_g(t_n). \quad (6) \end{aligned}$$

The route direction of a tracking robot  $i$  is estimated as

$$\hat{\theta}_i(t_n) = \arctan \left( \frac{y_i(t_n) - y_i(t_{n-1})}{x_i(t_n) - x_i(t_{n-1})} \right). \quad (7)$$

The positions of tracking robots after one control

period depend on the control inputs applied to robots. The control algorithm shown in Eq. (8) is applied to a tracking robot  $i$ .

$$\mathbf{u}_i = \begin{bmatrix} \omega_{i,d_i}(t_n) & v_i(t_n) \end{bmatrix}^T$$

$$\omega_{i,d_i}(t_n) \in \left[-\frac{\pi}{6}, 0, \frac{\pi}{6}\right], \quad 0 \leq v_i(t_n) \leq v_{\max} \quad (8)$$

$\omega_{i,d_i}(t_n)$  and  $v_i(t_n)$  denote the the angular velocity and translational velocity applied to a robot.  $d_i$  denotes an action decision of a robot  $i$ , which decides the angular velocity  $\omega_{i,d_i}(t_n)$  from  $(-\frac{\pi}{6}, 0, \frac{\pi}{6})$ .  $v_{\max}$  denotes the maximum value of the translational velocity in a control input.  $v_i(t_n)$  is calculated by

$$v_i^r(t_n) = k_f \hat{v}_g(t_n) + \frac{l_{i,g}(t_n) - D_r}{k_m D_r} k_p v_{\max}$$

$$v_i(t_n) = \begin{cases} v_i^r(t_n) & (v_i^r(t_n) < v_{\max}) \\ v_{\max} & (\text{otherwise}) \end{cases} \quad (9)$$

Here,  $k_f C k_m C k_p$  are factors, and  $l_{i,g}$  is the distance between estimated position  $\mathbf{g}(t_{n+1})$  after one control period and the position  $\mathbf{p}_i(t_n)$  of a tracking robot  $i$  is shown as:

$$\hat{x}_{g,\text{dist}} = \hat{x}_g(t_{n+1}) - x_i(t_n)$$

$$\hat{y}_{g,\text{dist}} = \hat{y}_g(t_{n+1}) - y_i(t_n)$$

$$l_{i,g}(t_n) = \sqrt{\hat{x}_{g,\text{dist}}^2 + \hat{y}_{g,\text{dist}}^2} \quad (10)$$

Thus, the translational velocity is deterministically derived at discrete time  $t_n$ , so the role of the game theory in a control algorithm is derivation of an appropriate  $\omega_{i,d_i}(t_n)$ .

Each autonomous robot control period is denoted by  $\Delta t$ , and the time delay in the position detection and the calculation of a control algorithm denotes  $T_0$ ,  $\Delta t_{-T_0} = \Delta t - T_0$  denotes between the time when the calculation of a control algorithm is completed and the time when the next control period starts. From these, the estimated robot position  $\hat{\mathbf{p}}_{i,d_i}(t_{n+1}) = [\hat{x}_{i,d_i}(t_{n+1}) \quad \hat{y}_{i,d_i}(t_{n+1})]^T$  after one control period acquiring an action decision  $d_i(t_n)$  are

calculated from Eq. (11) at discrete time  $t_n$

$$\begin{aligned} \hat{x}_{i,d_i}(t_{n+1}) &= x_i(t_n) \\ &+ v_i(t_{n-1}) T_0 \cos(\hat{\theta}_i(t_n) + \omega_{i,d_i}(t_{n-1}) T_0) \\ &+ v_i(t_n) \Delta t_{-T_0} \\ &\cdot \cos(\hat{\theta}_i(t_n) + \omega_{i,d_i}(t_{n-1}) T_0 + \omega_{i,d_i}(t_n) \Delta t_{-T_0}) \\ \hat{y}_{i,d_i}(t_{n+1}) &= y_i(t_n) \\ &+ v_i(t_{n-1}) T_0 \sin(\hat{\theta}_i(t_n) + \omega_{i,d_i}(t_{n-1}) T_0) \\ &+ v_i(t_n) \Delta t_{-T_0} \\ &\cdot \sin(\hat{\theta}_i(t_n) + \omega_{i,d_i}(t_{n-1}) T_0 + \omega_{i,d_i}(t_n) \Delta t_{-T_0}) \end{aligned} \quad (11)$$

To estimate the center position of a tracking robot team after one control period, when an action decision set  $(d_1, d_2, \dots, d_N)$  at discrete time  $t_n$  is acquired, the position  $\hat{\mathbf{c}}(d_1, \dots, d_N)$  is calculated from

$$\hat{\mathbf{c}}(d_1, \dots, d_N) = \left[ \frac{1}{N} \sum_{i=1}^N \hat{x}_{i,d_i}, \frac{1}{N} \sum_{i=1}^N \hat{y}_{i,d_i} \right] \quad (12)$$

## 2.2 Cost Value

To achieve the target tracking, the control algorithm makes an action decision by the game theory using the cost value indicating the situation of each robot in a tracking team. The cost value is designed to take a larger value in the action decision of the tracking robot preventing the achievement of the target tracking.

In the assumption that the tracking robot acquired decision set  $(d_1, d_2, \dots, d_N)$ , the cost value  $I_i$  of the tracking robot  $i$  is given by Eq. (13).

$$I_i(d_1, d_2, \dots, d_N)$$

$$= f_i(d_1, d_2, \dots, d_N, \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N, \mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_M, \mathbf{g}) \quad (13)$$

The cost value is constructed of four elements depending on the positions of other tracking robots, a tracked target robot and obstacles. These are defined by

$$I_i(d_1, d_2, \dots, d_N)$$

$$= K_{1,i} I_{1,i} + K_{2,i} I_{2,i} + K_{3,i} I_{3,i} + K_{4,i} I_{4,i} \quad (14)$$

$K_{1,i} C \quad K_{2,i} C \quad K_{3,i} C K_{4,i}$  are the weights of corresponding cost elements. Cost elements  $I_{1,i} C I_{2,i} C I_{3,i} C I_{4,i}$  are defined by

$$I_{1,i}(d_1, d_2, \dots, d_N) = \min_{\substack{j=1,2,\dots,N \\ k=1,2,\dots,M \\ j \neq i}} \left( \hat{l}_{r,i,j}, \hat{l}_{o,i,k}, \hat{l}_{i,g} \right)^{-1}. \quad (15)$$

The cost element of Eq. (15) acts as a collision avoidance term.

$\hat{l}_{r,i,j}$  denotes the distance between the estimated positions of robots  $i$  ( $i = 1, 2, \dots, N$ ) and  $j$  ( $j = 1, 2, \dots, N, i \neq j$ ) in a tracking robot team. Similarly,  $\hat{l}_{o,i,k}$  does the distance between the estimated position  $\hat{\mathbf{p}}_{i,d_i}$  of a tracking robot  $i$  and the position  $\mathbf{o}_k$  ( $k = 1, 2, \dots, M$ ) of an obstacle  $k$ ,  $\hat{l}_{i,g}$  does the distance between the estimated position of a robot  $i$  and the estimated position  $\hat{\mathbf{g}}$  of a tracked target robot. A value of  $I_{1,i}$  is increased when the robot  $i$  approaching other robots, a tracked target robot and obstacles.

Therefore, to avoid a situation in which a collision almost close, a decision should be made to reduce the value of  $I_{1,i}$ .

The cost element  $I_{2,i}$  itself is just the distance  $\hat{l}_{c,i}$  between the estimated position  $\hat{\mathbf{p}}_{i,d_i}$  of a robot  $i$  and the estimated center position  $\hat{\mathbf{c}}(d_1, d_2, \dots, d_N)$  of a tracking robot team. So,  $I_{2,i}$  is defined as

$$I_{2,i}(d_1, d_2, \dots, d_N) = \hat{l}_{c,i}. \quad (16)$$

$I_{2,i}$  is the term for keeping a constant distance among tracking robots. So,  $I_{2,i}$  take a larger value as a robot leaving from  $\hat{\mathbf{c}}$ ,  $I_{2,i}$ .

The third cost element  $I_{3,i}$  means to achieve one of the task of target tracking where a tracking robot team follows a tracked target robot while keeping a certain distance.  $I_{3,i}$  is defined using a distance  $\hat{l}_{c,g}$  between the estimated position  $\hat{\mathbf{c}}(d_1, \dots, d_N)$  of a tracked target robot  $\hat{\mathbf{g}}$  and the estimated position of robot team center and a desired value  $I_{\text{ref},i}^3$  for  $\hat{l}_{c,g}$  to keep a desired formation as follows:

$$I_{3,i}(d_1, d_2, \dots, d_N) = \left| \hat{l}_{c,g} - I_{\text{ref},i}^3 \right|. \quad (17)$$

Thus, increasing in the distance between an entire tracking robot team and a tracked target robot from a desired value makes  $I_{3,i}$  large.

The fourth cost element  $I_{4,i}$  means to keep a desired formation for a tracking robot team and is calculated from

$$I_{4,i}(d_1, d_2, \dots, d_N) = \hat{l}_{p_i, p'_i}. \quad (18)$$

where  $\hat{\mathbf{p}}'_i$  denotes the estimated position of a tracking robot  $i$  when each tracking robot forms a desired formation including the estimated position  $\hat{\mathbf{g}}$  of

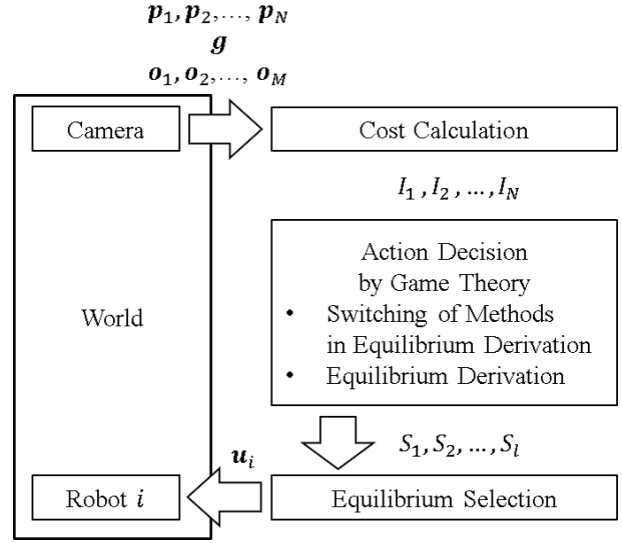


Fig.2: Control Algorithm

a tracked target.  $\hat{l}_{p_i, p'_i}$  does the estimated position  $\hat{\mathbf{p}}_i$  by an action decision of a robot  $i$  and the estimated position  $\hat{\mathbf{p}}'_i$ .

Therefore, the more a robot  $i$  leaves far from the placement to meet a desired formation, the larger  $I_{4,i}$  is.

### 3. CONTROL ALGORITHM FOR COORDINATION

In this chapter, the control algorithm given to each tracking robot in order to achieve the target tracking is described.

#### 3.1 Control Algorithm

Figure 2 shows the control algorithm of each tracking robot to be used in this study.

In the control algorithm, the target tracking is modeled from  $\mathbf{g}$  as the position of a tracked target robot,  $\mathbf{p}_i$  ( $i = 1, 2, \dots, N$ ) as the position of each robot in a tracking team and  $\mathbf{o}_j$  ( $j = 1, 2, \dots, M$ ) as the position of each obstacle by a camera in a real environment, and the cost values are calculated.

Equilibriums  $S_1, S_2, \dots, S_l$  are derived using the game theory from the calculated cost values. When there are more than one equilibrium ( $l \neq 1$ ), one equilibrium is selected in order to obtain a more even position control. If one equilibrium is selected, a tracking robot obtains a  $d_i$  as the action decision. The control input  $\mathbf{u}_i$  is given to a tracking robot  $i$  by the derived decision.

#### 3.2 Action Decision by Game Theory

In this study, two methods in the equilibrium derivation in the game theory are used to derive a rational decision for each robot.

### 3.2.1 Non-cooperative Nash Equilibrium

We now introduce the non-cooperative Nash equilibrium in [5]. In this equilibrium, the target tracking problem is regarded as a non-cooperative game. The Nash equilibrium which has self-enforcing [6] is obtained. The tracking robot acquires the action minimizing the cost value by this equilibrium and eventually achieve the task.

Here, the Nash equilibrium is a set of decisions  $(d_1^{NN}, d_2^{NN}, \dots, d_N^{NN})$  so that the cost values  $I_1, I_2, \dots, I_N$  of all tracking robots satisfy Eq. (19).

$$\begin{aligned} I_1(d_1^{NN}, d_2^{NN}, \dots, d_N^{NN}) \\ &\leq I_1(d_1, d_2^{NN}, \dots, d_N^{NN}) \\ I_2(d_1^{NN}, d_2^{NN}, \dots, d_N^{NN}) \\ &\leq I_2(d_1^{NN}, d_2, \dots, d_N^{NN}) \\ &\vdots \\ I_N(d_1^{NN}, d_2^{NN}, \dots, d_N^{NN}) \\ &\leq I_N(d_1^{NN}, d_2^{NN}, \dots, d_N) \end{aligned} \quad (19)$$

Therefore, the cost value  $I_i^{NN}$  of a tracking robot acquired by the Nash equilibrium is represented by

$$I_i^{NN} = I_i(d_1^{NN}, d_2^{NN}, \dots, d_N^{NN}) . \quad (20)$$

### 3.2.2 Semi-Cooperative Stackelberg Equilibrium

In the non-cooperative Nash equilibrium, the equilibrium cannot always makes evenly cost values of entire tracking robots. Therefore, when there is a robot whose cost value is different from cost values of other robots, an inconvenient situation for the robot may continue.

And so, we next introduce the semi-cooperative Stackelberg equilibrium in [5] to stimulate the evenness of the cost value by a decision which focuses on reducing the maximum cost value in a tracking robot team. In the semi-cooperative Stackelberg equilibrium, the communication mechanism is set to each tracking robot, and tracking robots exchange their own cost values each other in a robot team.

The leader as a robot with the maximum cost value and the followers as other robots are classified by these cost values. The agreement to guarantee the decision of the leader is obtained. Next, assuming the guaranteed decision of the leader, decisions of followers are derived by the Nash equilibrium in the non-cooperative game constructed only with followers (the followers game).

Thus, The target tracking problem is dealt with as a semi-cooperative situation because decisions are obtained among followers in the non-cooperative game theory. On the other hand, the decision of the leader is obtained in cooperative game theory.

The robot  $i$  as the leader derives a Nash equilibrium  $d_{-i} = (d_1, d_2, \dots, d_{i-1}, d_{i+1}, \dots, d_N)$  assuming

the followers game when the decision  $d_i$  of leader is locked.

The leader repeats this process for their own all decisions, and derives all Nash equilibriums of followers for each decision of the leader. Their own decision  $d_i^{SS}$  and followers' decision  $d_{-i}^{SS} = (d_1^{SS}, d_2^{SS}, \dots, d_{i-1}^{SS}, d_{i+1}^{SS}, \dots, d_N^{SS})$  of followers which minimize the cost value  $I_i$ . After that, the leader communicates  $d_i^{SS}$  to each follower, and each follower derives the Nash equilibrium of the followers game assuming communicated  $d_i^{SS}$ .

From these, a Stackelberg equilibrium  $(d_1^{SS}, d_2^{SS}, \dots, d_N^{SS})$  is acquired in the semi-cooperative game.

Here,  $I_i^{SS}$  as the cost value obtained by a tracking robot  $i$  in the Stackelberg equilibrium of the semi-cooperative game is represented as

$$\begin{aligned} I_1^{SS} &= I_1(d_1^{SS}, d_2^{SS}, \dots, d_N^{SS}) \\ I_2^{SS} &= I_2(d_1^{SS}, d_2^{SS}, \dots, d_N^{SS}) \\ &\vdots \\ I_N^{SS} &= I_N(d_1^{SS}, d_2^{SS}, \dots, d_N^{SS}) . \end{aligned} \quad (21)$$

The leader is represented as the robot  $L \in 1, 2, \dots, N$ , and Eq. (22) is established.

$$I_L^{SS} \leq I_L(d_1^{NN}, d_2^{NN}, \dots, d_N^{NN}) \quad (22)$$

### 3.2.3 Equilibrium Selection

Multiple Nash equilibriums may exist. To select the equilibrium  $S'$  which attains the minimum deviation of cost values among multiple equilibriums, Eqs. (23), (24) and used in [3].

$$\begin{aligned} C(S_k) &= \sum_{i=1}^N (I_i(S_k) + I_i^{dev}(S_k)) \\ S' &= \arg \min_{k=1,2,\dots,l} C(S_k) \end{aligned} \quad (23)$$

$$I_i^{dev} = \left| I_i - \frac{1}{N} \sum_{j=1}^N I_j \right| \quad (24)$$

## 3.3 Switching of Methods in Equilibrium Derivation

In the traditional control algorithm, any one equilibrium derivation is decided to either non-cooperative Nash equilibrium or semi-cooperative Stackelberg equilibrium in advance. In a non-cooperative Nash equilibrium, when these remains an inconvenient situation for a tracking robot significantly leaving from the task, it is effective to encourage the evenness of cost values and to focus on making the maximum cost value in a tracking robot team low

for achieving the task. But, a semi-cooperative Stackelberg equilibrium requires communications among robots. So, a non-cooperative Nash equilibrium is favorable from the point of view of burden in communication.

By properly switching these two methods in the equilibrium derivation, the improvement of communicative efficiency and the ability to achieve the task are expected very much. Therefore, in this paper, we calculate equilibriums of two methods in the equilibrium derivation in parallel. And switch these two equilibriums depending on the  $I_{MAX}$  of each equilibriums. In order to switch from a non-cooperative Nash equilibrium to a semi-cooperative Stackelberg equilibrium in an appropriate timing, a cost threshold  $I_{th}$  is introduced. Only when  $I_{MAX}$  as the maximum cost value in a tracking robot team becomes  $I_{MAX} \geq I_{th}$ , the tracking robot team uses a semi-cooperative Stackelberg equilibrium. Otherwise the tracking robot team uses a non-cooperative equilibrium.

#### 4. VALIDATION BY THE SIMULATION

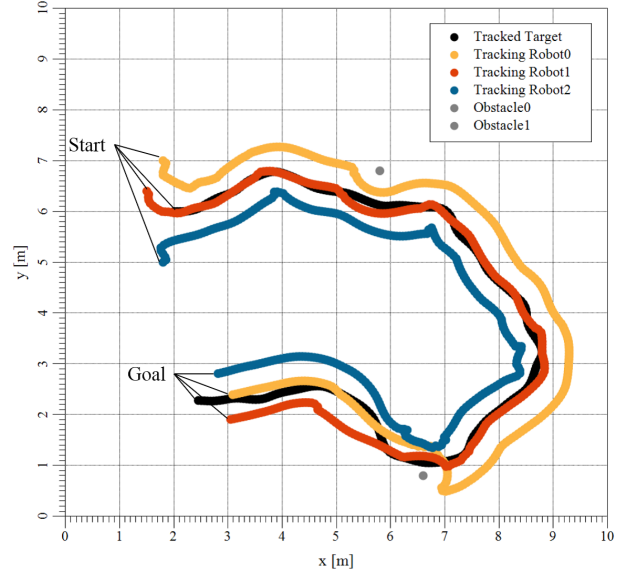
We validate the effectiveness of a decision using a switching method for the equilibrium derivation through a simulation. It is simulated that three tracking robots follow one tracked target robot while making a formation. All the parameters in the simulation is shown in Table 1.

**Table 1:** Parameters used by Simulation

Parameters	Values
Translational Velocity of Tracked Target $v_g$	0.017 [m/s]
Max Translational Velocity of Tracking Robot $v_{max}$	0.023 [m/s]
Radius $D_r$	0.09 [m]
$k_f$	0.3
$k_p$	0.8
$k_m$	10
Computation Time $T_0$	0.05 [s]
Control Period $\Delta t$	0.1 [s]
Cost Weight $K_{1,i}$ ( $i = 1, 2, 3$ )	1.8
Cost Weight $K_{2,i}$ ( $i = 1, 2, 3$ )	7
Cost Weight $K_{3,i}$ ( $i = 1, 2, 3$ )	11
Cost Weight $K_{4,i}$ ( $i = 1, 2, 3$ )	2
$I_{ref,3,1}$	0 [m]
$I_{ref,3,2}$	0.04 [m]
$I_{ref,3,3}$	0.04 [m]
Cost Threshold $I_{th}$	7.96

Each parameter is determined by trial and error. The orbits of these robots when only a non-cooperative Nash equilibrium is used for their own decision are shown in Fig. 3.

From Figure 3, each tracking robot starts moving from the initial position, and it is confirmed that each



**Fig.3:** Orbits of Robots by Non-Cooperative Nash Equilibrium

tracking robot moves with formation keeping, collision avoidance and tracked target following robot.

A relatively large disturbance in the formation of a tracking team occurs after the position  $(x, y) = (6.8[m], 3.8[m])$  in which a tracked target changes in the route direction.

Nevertheless, target tracking and collision avoidance have been achieved finally. But alignments of the tracking robot 0 and the robot 1 are reversed at the scene of avoiding the second obstacle. Eventually a precise formation has never been achieved.

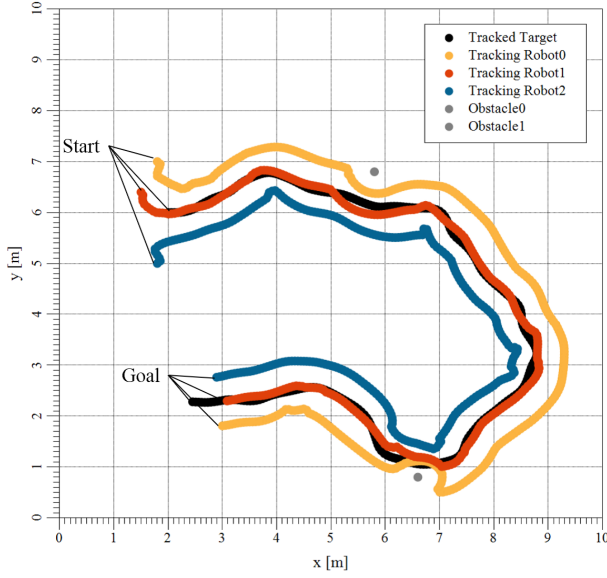
When a relatively large disturbance in the formation occurs in the obstacle avoidance, it is considered that the tracking robot whose cost value rises significantly compared to the other tracking robots appears.

But because a non-cooperative Nash equilibrium considers not only their own cost value but also reducing the cost values of the other tracking robots, even if the robot keeps away significantly from achieving the task, it is difficult to make a decision to focus on reducing the cost value of a specific robot.

Therefore, the action decision which keeps a different formation from desired has been obtained.

In this situation, in order to keep the desired formation, it is effective to acquire a decision to focus on reducing the maximum cost value in a tracking robot team.

Next, the orbit of each robot when using the proposed switching method for the equilibrium derivation for decision is shown in Fig. 4. Because each robot uses a semi-cooperative Stackelberg equilibrium only if the maximum cost value in a tracking robot team is larger than  $I_{th}$ , the number of communication in a tracking robot team is suppressed to about 77.1% in the case of using a semi-cooperative Stack-

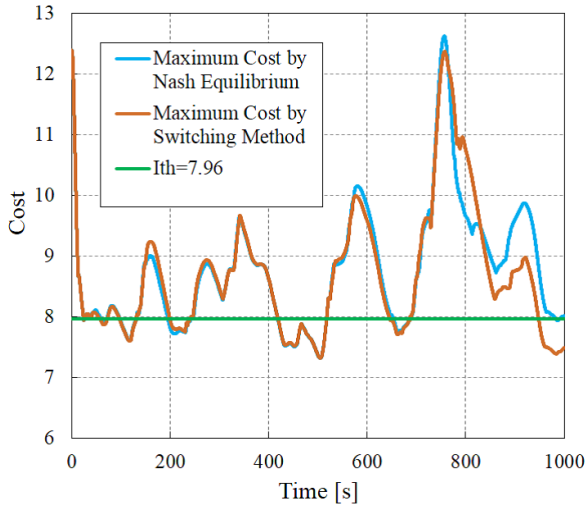


**Fig.4:** Orbits of Robots by Switching Methods

elberg equilibrium at all times.

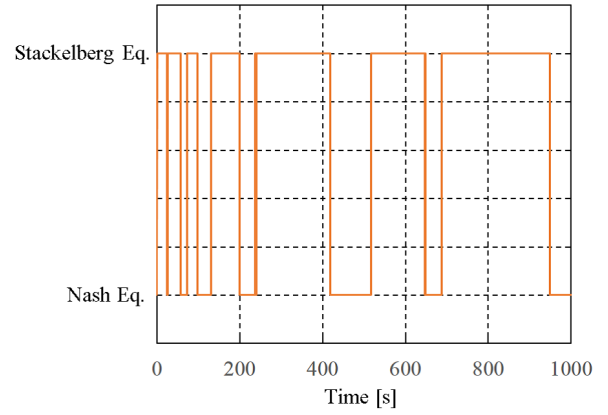
Similarly to Fig. 3, each tracking robot starts moving from the initial position in Fig. 4, and it is confirmed that each tracking robot moves with formation keeping, collision avoidance and tracked target following. But in the scene of avoiding the second obstacle, the behavior of tracking robots differ from Fig. 3.

Finally, the reverse placement of tracking robots observed in Fig. 3 has never occurred. As a result, a precise formation keeping has been achieved.



**Fig.5:** Maximum Cost Values in Team of Tracking Robot

The maximum cost value in a tracking robot team is shown in Fig. 5 in the cases of using only a non-cooperative Nash equilibrium and using switching of two methods in the equilibrium derivation, and the transition of the switching is shown in Fig. 6 in the cases of using switching of two methods in the equilibrium derivation.



**Fig.6:** Transition of Switching in Switching Methods

librium derivation.

The variation of the maximum cost value hardly occurs until the halfway point from the start in the simulation. But in the avoidance of a tracking robot team from the second obstacle, variations in the orbits of tracking robots are observed.

These are due to the following: The decision is made to focus on reducing the maximum cost value in a tracking robot team because a semi-cooperative Stackelberg equilibrium is used as a method in the equilibrium derivation caused by the maximum cost value that is larger than or equal to  $I_{th}$  as the cost threshold in use of the switching method in the equilibrium derivation.

Owing to this effect, it is expected that the appearance of a tracking robot whose cost value is larger than those of other robots is suppressed. Therefore, the reverse placement does not occur.

Finally it is considered that a precise formation keeping has been achieved.

## 5. CONCLUSION

In this study, we proposed a switching method for the equilibrium derivation in the game theory for the control algorithm to target tracking, and the effectiveness of our proposed method was demonstrated through a simulation.

In a non-cooperative Nash equilibrium, the situation that the task has not been achieved occurred in the long term. Especially, in the situation where an increasing risk of collision by a rapid unplanned change in the route direction of a tracked target robot while approaching obstacles, it was confirmed that effective measures cannot be implemented for the appearance of the robots remarkably deviating away from the achievement task.

In the switching method for the equilibrium derivation, only when the maximum cost value in a tracking robot team that was larger than or equal to the cost threshold, we switched from a non-cooperative Nash

equilibrium to a semi-cooperative Stackelberg equilibrium.

Thus, it was possible to suppress the communication among robots in comparison with the case of using a semi-cooperative Stackelberg equilibrium at all times. In addition, the effect of equalizing the cost values of tracking robots in a semi-cooperative Stackelberg equilibrium worked well by focusing on the robot which obtained the maximum cost value.

As a result, it was confirmed that the placement of desired formation that has never been achieved in a non-cooperative Nash equilibrium has been achieved.

## References

- [1] Azuma Ohuchi, Masahito Ymamoto, Hidenori Kawamura, "Theory and application of multi-agent systems : computing paradigm from complex systems engineering", *Corona Publishing Co., Ltd.*, 2002 (in Japanese).
- [2] Shigeo Mutou, "Game Theory", *Ohmsha, Ltd.*, 2011 (in Japanese).
- [3] K. Skrzypczyk, "Game theory based target following by a team of robots", *Proceedings of Forth International Workshop on Robot Motion and Control*, pp.91-96, 2004.
- [4] I. Harmati, "Multiple robot coordination for target tracking using semi-cooperative stackelberg equilibrium", *IEEE International Control Conference*, 2006.
- [5] I. Harmati, K. Skrzypczyk, "Robot team coordination for target tracking using fuzzy logic controller in game theoretic framework", *Robotics and Autonomous Systems*, Volume 57, Issue 1, pp.75-86, 2009.
- [6] Akira Okada, "Game Theory", *Yuhikaku Publishing Co., Ltd.*, 1996 (in Japanese).



**Shu Hosokawa** received his B.Eng. from The University of Electro-Communications, Tokyo, Japan in 2007. In 2009, he received M.Eng. from the graduate school of The University of Electro-Communications. His research interests include Control Engineering and Robotics. Now he is working at Control System Security Center.



**Wataru Inujima** is a master's course student in the second year of The University of Electro-Communications, Tokyo, Japan. He received B. Eng. from The University of Electro-Communications in 2012. His research interests include Control Engineering and Robotics.



**Kazushi Nakano** received his Dr. Eng. degree from Kyushu University, Fukuoka, Japan, in 1982. From 1980, he was a Research Associate at Kyushu University. From 1986, he was an Associate Professor at Fukuoka Institute of Technology. He is currently a Professor in the Department of Informatics and Engineering, the University of Electro-Communications, Tokyo, Japan. His interests include system identification/control and their applications. He is a member of IEEE, ISCIE and IEEJ.

tion/control and their applications. He is a member of IEEE, ISCIE and IEEJ.