Improvement of an LUT-based Intelligent Motion Controller by Underestimation of Reachable Sets

Shota Sekine, Shigeki Matsumoto and Katsutoshi Yoshida

***Abstract*— We develop a preprocessing method to improve the performance of an intelligent motion controller. Our method addresses the competitive problems that have been observed in the coupled inverted pendula model, in which a controller outputs impulsive forces to produce the desired final states based on a look-up table (LUT) that stores dynamical correspondences from the initial to the final states. However, the degradation in performance due to misclassifications in the LUT occurs near the boundary points of the reachable sets of the model. The proposed algorithm removes the boundary points from the LUT. The resulting controller successfully reduces misclassification errors and improves overall control performance.**

I. INTRODUCTION

Multiple agents can exhibit both competitive and cooperative dynamics when sharing common resources and environments. Examples of such dynamics can be found in interspecific competition [1], multi-robot systems solving cooperative problems [2–8] and competitive problems [9,10], and so on.

In a previous study [11], we proposed a coupled inverted pendula (CIP) framework in which the tips of two inverted pendula are linked by a connecting rod, and each pendulum is primarily stabilized by a proportional-derivative (PD) controller. Using the CIP model, we examined a strategy in which one pendulum is stabilized while the other pendulum is reversed [12]. We then developed an intelligent controller comprising three components: a classifier, a selector, and an impulse generator [13]. The controller outputs impulsive forces to produce the desired final state based on a look-up table (LUT) that stores dynamical correspondences between the initial and final states (basin of attraction). It was demonstrated that the performance depends not only on the quantization resolution of the LUT but also on the delay time of the introduced elements.

Although similar use of the concept of basin of attraction can be found in the studies by Liu [14] and also by Sprott [15], they focused on low dimensional nonlinear dynamical systems. For example, Liu [14] proposed a method for controlling a second-order ordinary differential equation. In contrast, in our studies [11–13], we delt with the eight-dimensional nonlinear dynamical system.

In our previous study [13], the LUT-based classifier that we proposed displayed a problem of misclassification: the classifier sometimes predict a final state that disagrees with the final state to which the CIP model actually converges.

S. Sekine, S. Matsumoto, and K. Yoshida are with the Department of Mechanical and Intelligent Engineering, Utsunomiya University, Yoto 7-1-2, Utsunomiya, Tochigi 321-8585, Japan sekine@katzlab.jp

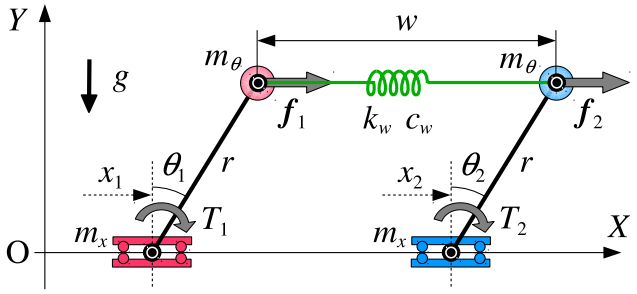

Fig. 1. Coupled inverted pendula (CIP) model with viscoelastic connection.

TABLE I

PARAMETER SETTING OF THE CIP MODEL.

x_i [m]	horizontal displacement of the ith cart	
θ_i [rad]	slant angle of the ith pendulum	
f_i [N]	reaction force acting on the tip of ith pendulum	
T_i [N]	input torque on θ_i	
m_x [kg]	mass of cart	0.68
m_θ [kg]	mass of pendulum	0.067
w [m]	length of connection rod	1
r [m]	length of pendulum	0.3
k_w [N/m]	spring coefficient of connection rod	5000
c_w [Ns/m]	viscous coefficient of connection rod	50
g [m/s^2]	acceleration of gravity	9.8
c_x [Ns/m]	viscous coefficient along x	0.01
c_θ [Ns/rad]	viscous coefficient about θ	0.01
k_f [N/m]	spring coefficient of floor	500
c_f [Ns/m]	viscous coefficient of floor	10
K	proportional gain of standing control	1
L	derivative gain of standing control	0.01

In this study, we address this problem by developing a preprocessing step that removes the boundary points from the LUT. The resulting classifier successfully reduces the rate of misclassification and improves control performance.

II. THE COUPLED INVERTED PENDULA FRAMEWORK

A. Coupled Inverted Pendula

In order to create mechanical agents that maintain their balance when mechanically coupled to each other, we apply the CIP model, as shown in Fig. 1. Each inverted pendulum is attached to a cart moving along a horizontal floor ($Y = 0$); the simple pendulum rotates about a point on the cart. For simplicity, a common physical specification is given to both pendula. The physical parameter values are listed in Table I.

The configuration of this linkage is uniquely determined by four variables: the horizontal displacement of the two carts x_1 and x_2, and the slant angles of the pendula θ_1 and

θ_2. Applying Lagrangian mechanics and assuming viscous frictional forces $c_x\dot{x}_i$ and $c_\theta\dot{\theta}_i$ on x_i and θ_i, respectively, we obtain the equations of motion (EOM) for the CIP model (Fig. 1) as follows:

$$\begin{cases} (m_x + m_\theta)\ddot{x}_i + (m_\theta r\cos\theta_i)\ddot{\theta}_i - m_\theta r\dot{\theta}_i^2\sin\theta_i \\ \qquad = -c_x\dot{x}_i + (1,0)\boldsymbol{f}_i, \\ (m_\theta r\cos\theta_i)\ddot{x}_i + (m_\theta r^2)\ddot{\theta}_i - m_\theta gr\sin\theta_i \\ \qquad = -c_\theta\dot{\theta}_i + r(\cos\theta_i, -\sin\theta_i)\boldsymbol{f}_i + T_i \quad (i=1,2). \end{cases} \tag{1}$$

B. Reaction Force from the Connection Rod

We calculate the reaction force $\boldsymbol{p}$ from the connection rod. The displacement vector $\boldsymbol{w}$ from the left-hand tip $\boldsymbol{X}_1$ to the right-hand tip $\boldsymbol{X}_2$ of the pendula is given by

$$\boldsymbol{w} = (w_X, w_Y)^T := \boldsymbol{X}_2 - \boldsymbol{X}_1, \tag{2}$$

where

$$\boldsymbol{X}_i = (X_i, Y_i)^T = \begin{bmatrix} x_i + r\sin\theta_i \\ r\cos\theta_i \end{bmatrix} \quad (i=1,2). \tag{3}$$

We then model the viscoelasticity of the connection rod to derive the reaction force:

$$\boldsymbol{p} = (-k_w(w - w_0) - c_w\dot{w})\boldsymbol{w}/w, \quad w = \|\boldsymbol{w}\|. \tag{4}$$

Here, w_0 is a natural length of the connection rod, and $\|\cdot\|$ denotes a norm of a vector. Substituting $\boldsymbol{p}$ into the EOM (1) via $\boldsymbol{f}_i = (-1)^i\boldsymbol{p}$, we obtain an analytic expression of the CIP model with the viscoelastic connection.

C. Modeling Floor

We add the floor to the CIP model in the following manner. Using penalty methods, we first model a normal force R_i from the floor ($Y = 0$) acting on the tip of ith pendulum as

$$R_i = U(-Y_i)\{-k_f Y_i - c_f\dot{Y}_i\} \quad (i=1,2), \tag{5}$$

where Y_i is the height of the ith tip from the floor, $U(\cdot)$ is a unit step function, and k_f and c_f are viscoelastic parameters representing reaction properties. In practice, to avoid numerical errors we approximate the step function with a sigmoid function, defined by

$$U_\sigma(s) := \{1 + \exp(-\sigma s)\}^{-1}, \tag{6}$$

where $\lim_{\sigma\to\infty} U_\sigma(s) = U(s)$ is applicable.

A Coulomb friction force F_i acting on the ith tip from the floor can be expressed as, $F_i = -\mu R_i\ \mathrm{sgn}(\dot{X}_i)$, where μ is a friction coefficient, $\dot{X}_i$ is the relative velocity of the ith tip from the floor, and $\mathrm{sgn}(\cdot)$ is a unit signum function whose smooth approximation can be given by $\mathrm{sgn}(s) \approx \mathrm{sgn}_\sigma(s) := 2U_\sigma(s) - 1$. The CIP model with the viscoelastic connection on the floor is therefore obtained by substituting

$$\boldsymbol{f}_i = (-1)^i\boldsymbol{p} + (F_i, R_i)^T \quad (i=1,2) \tag{7}$$

into the EOM in (1).

The CIP model can be expressed as an eight-dimensional dynamical system:

$$\begin{cases} \boldsymbol{x}_i := \left(x_i, \dot{x}_i, \theta_i, \dot{\theta}_i\right)^T \in R^4 \quad (i=1,2), \\ \dot{\boldsymbol{x}}_i = \boldsymbol{f}_i(\boldsymbol{x}, T_i), \quad \boldsymbol{x} := (\boldsymbol{x}_1^T, \boldsymbol{x}_2^T)^T, \\ \boldsymbol{T} = (T_1, T_2)^T = \boldsymbol{u}^{\mathrm{pd}} + \boldsymbol{u}^{\mathrm{ic}} + \boldsymbol{v}, \\ \boldsymbol{u}^{\mathrm{pd}} = (u_1^{\mathrm{pd}}, u_2^{\mathrm{pd}})^T, \quad \boldsymbol{u}^{\mathrm{ic}} = (u_1^{\mathrm{ic}}, u_2^{\mathrm{ic}})^T. \end{cases} \tag{8}$$

D. Standing Control with Falling

We begin by developing a feedback controller in which each inverted pendulum on the floor forms three stable equilibriums. This is done by defining a feedback controller of the following form:

$$u_i^{\mathrm{pd}} := \mathrm{trap}_\alpha(\theta_i; \Delta\theta)\{-K_p\theta_i - K_d\dot{\theta}_i\} \quad (i=1,2), \tag{9}$$

where $\mathrm{trap}_\alpha(\theta_i; \Delta\theta)$ is a smooth trapezoidal function of unit height centered at $\theta = 0$ as a product of the sigmoid function in (6), and $\Delta\theta > 0$ is the half-width of the trapezoidal shape, which becomes steeper as α increases. It follows from the deadband characteristics in (9) that u_i^{pd} simply acts as a PD controller inside the limit $|\theta_i| < \Delta\theta$ while rapidly cutting off the output on the outside of the limit. Therefore, an appropriate setting of the gains K_p and K_d makes it possible for the ith pendulum to be stabilized about the standing position $\theta_i = 0$ while falling towards the floor when $|\theta_i|$ exceeds the given limit $\Delta\theta$.

E. CIP Framework

Since each agent with (9) has three stable equilibriums $\bar{\theta}_i = 0, \pm\pi/2$, a pair of agents being coupled together under suitable conditions can produce 9 ($= 3 \times 3$) stable equilibriums by

$$\boldsymbol{\omega}_i := \lim_{t\to\infty} \boldsymbol{x}(t) = \left(\bar{x}_1^i, 0, \bar{\theta}_1^i, 0, \bar{x}_2^i, 0, \bar{\theta}_2^i, 0\right)^T \quad (i = 1, \cdots, 9), \tag{10}$$

as shown in Fig. 2, when equating horizontal translations of final position $\bar{x}_1^i$ and $\bar{x}_2^i$ without loss of generality. The components $x_1(t)$ and $x_2(t)$ of (8) are neutrally and not asymptotically stable because no restoring forces on $x_1(t)$ and $x_2(t)$ are assumed, by definition. We then attach competitive meanings to the nine equilibriums, as listed in Fig. 2. The agent that remains standing is regarded as the winner. Eventually, we arrive at the CIP framework composed of a combination of (A) and (B) as bellow.

(A) The CIP model: the system of equations defined in (8).
(B) The win-loss matrix: the competitive interpretation of the nine equilibriums defined in Fig. 2.

III. INTELLIGENT CONTROLLER

A. Problem Setting and Requirements

Based on the CIP framework, we develop an intelligent controller (IC) which receives a measurement vector $\boldsymbol{y}$ given

$\theta_1 \backslash \theta_2$	0	$-\pi/2$	$\pi/2$
0	$\boldsymbol{\omega}_1$: draw	$\boldsymbol{\omega}_2$: left win pulling	$\boldsymbol{\omega}_3$: left win pushing
$\pi/2$	$\boldsymbol{\omega}_4$: right win pulling	$\boldsymbol{\omega}_5$: draw	$\boldsymbol{\omega}_6$: draw
$-\pi/2$	$\boldsymbol{\omega}_7$: right win pushing	$\boldsymbol{\omega}_8$: draw	$\boldsymbol{\omega}_9$: draw

Fig. 2. Competitive interpretation of the equilibriums.

by the following linear measurement equation:

$$\begin{aligned}
&\boldsymbol{y} = H\boldsymbol{x}, \quad \boldsymbol{y} \in R^4, \quad \boldsymbol{x} \in R^8, \\
&H = \left[\boldsymbol{o}^{(4)}, \boldsymbol{o}^{(4)}, \boldsymbol{e}_1^{(4)}, \boldsymbol{e}_2^{(4)}, \boldsymbol{o}^{(4)}, \boldsymbol{o}^{(4)}, \boldsymbol{e}_3^{(4)}, \boldsymbol{e}_4^{(4)}\right], \\
&h(\boldsymbol{y}) := H^+\boldsymbol{y} + x_1\boldsymbol{e}_1^{(8)} + \dot{x}_1\boldsymbol{e}_2^{(8)} + x_2(x_1, \theta_1, \theta_2)\boldsymbol{e}_5^{(8)} \\
&\qquad + \dot{x}_2(\dot{x}_1, \theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2)\boldsymbol{e}_6^{(8)},
\end{aligned} \tag{11}$$

where $\boldsymbol{e}_i^{(d)}$ and $\boldsymbol{o}^{(d)}$ denote the ith standard basis vector and the zero vector in Euclidean space R^d, respectively, and $h(\boldsymbol{y})$ is an inverse of $\boldsymbol{y} = H\boldsymbol{x}$, where H^+ is the Moore-Penrose pseudoinverse of H.

Then, we assume that the IC outputs a series of impulsive forces in the following form:

$$u_i^{\mathrm{ic}}(t) := \sum_{j=1}^{N} P_i\, I_{\Delta\tau}(t - t_i^j), \tag{12}$$

where

$$I_{\Delta\tau}(t) = \begin{cases} (\Delta\tau)^{-1} & (0 \le t < \Delta\tau), \\ 0 & (\text{otherwise}) \end{cases} \tag{13}$$

is a rectangular function of unit area of width $\Delta\tau \ll 1$, P_i is an angular impulse of the input torque $u_i^{\mathrm{pd}}(t)$, and $\{t_i^1, \cdots, t_i^m\}$ is a series of rise time that satisfies

$$t_i^1 < t_i^2 < \cdots < t_i^N, \quad \max_{j,k} |t_i^j - t_i^k| \ge \tau_G \ge \Delta\tau, \tag{14}$$

where τ_G is a relaxation time to avoid overlapping outputs.

In practical implementation, the rise times $t_i^1, \cdots, t_i^m$ should be sequentially determined by the real-time architecture described in Fig. 3, comprising three components: a classifier C, a selector S, and an impulse generator G.

B. Classifier C

We define the classifier C of the first IC u_1^{ic} as a function from a measured state $\boldsymbol{y} = H\boldsymbol{x}$ to an index number ν of an equilibrium $\boldsymbol{\omega}_\nu$, in which a solution of (8) for $u_1^{\mathrm{ic}} := P_i\, I_{\Delta\tau}(t - t_0)$ and $u_2^{\mathrm{ic}} = 0$ converges.

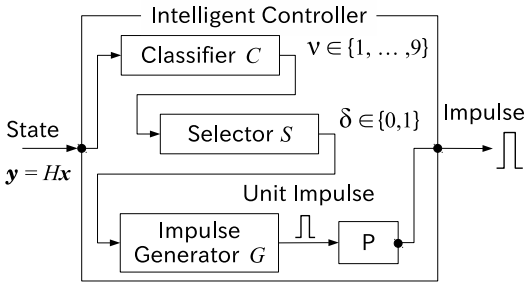

Fig. 3. Intelligent controller (IC).

For this purpose, we introduce the transition operator of (8) from an initial state $\boldsymbol{x}(t_0) = \boldsymbol{\xi}_0$ as $\boldsymbol{x}(t) := \boldsymbol{\phi}_t(\boldsymbol{\xi}_0, \boldsymbol{T})$ and define a set of the initial measured states $\boldsymbol{\eta}_0 \in R^4$ in which the state $\boldsymbol{x}(t)$ starting from the inverse $\boldsymbol{\xi}_0 = h(\boldsymbol{\eta}_0)$ converges to $\boldsymbol{\omega}_\nu$:

$$\Psi_\nu := \left\{\boldsymbol{\eta}_0 \in R^4 \;\middle|\; \lim_{t\to\infty} \boldsymbol{\phi}_t\big(h(\boldsymbol{\eta}_0), \boldsymbol{T}\big) = \boldsymbol{\omega}_\nu\right\}. \tag{15}$$

The set Ψ_ν is generally called a reachable set [16] or a basin of attraction [17]. The classifier C can be obtained as a single valued function of the measured state $\boldsymbol{y}$ in the following form:

$$C(\boldsymbol{y}) := \nu \quad \text{if} \quad \boldsymbol{y} \in \Psi_\nu. \tag{16}$$

C. Selector S

In the competitive problem in Fig. 2, some of the equilibriums are selected depending upon the strategies of the agent being considered. Such a selection process can be modeled by a selector $S_{\mathbb{J}}$ given by

$$\delta = S_{\mathbb{J}}(\nu) := \begin{cases} 1 & (\nu \in \mathbb{J} \subset \{1, \cdots, 9\}), \\ 0 & (\text{otherwise}), \end{cases} \tag{17}$$

where $\nu = C(\boldsymbol{y})$ is an output of the classifier, and $\mathbb{J}$ is a given subset of indices of the equilibriums $\boldsymbol{\omega}_1, \cdots, \boldsymbol{\omega}_9$.

D. Impulse Generator G

The impulse generator G is designed to receive the binary signal from the selector $\delta(t) = S_{\mathbb{J}}(\nu(t))$ and output the unit impulse (see [13] for details). This comprises two timer functions, T_I and T_G, and a two-input AND gate. The timer T_I produces a unit impulse as

$$G(t) = T_I(t) := I_{\Delta\tau}(t - t_r), \tag{18}$$

and the timer T_G cuts off the binary signal $\delta(t) = S_{\mathbb{J}}(\nu(t))$ for a given relaxation time τ_G in (14) by

$$T_G(t) := \begin{cases} 0 & (t_r < t < t_r + \tau_G), \\ 1 & (\text{otherwise}), \end{cases} \tag{19}$$

where t_r is the rise time from 0 to 1 of the Boolean product $\hat{\delta}(t) = S_{\mathbb{J}}(\nu(t)) \wedge T_G(t)$. As stated below (14), $T_G(t)$ is required to avoid overlapping outputs.

E. Numerical Approximation of the Classifier C

We numerically construct an LUT $C : \boldsymbol{y} \mapsto \nu$. First, we take a 4-dimensional cubic region D of measuring range within a direct sum of the reachable sets:

$$\bigoplus_{k=1}^{9} \Psi_k \supset D := [a_1, b_1] \times \cdots \times [a_4, b_4], \tag{20}$$

where $\bigoplus$ represents a direct sum (disjoint union) and $[a, b]$ denotes an interval. We divide it into a direct sum of uniform subcubes $D^{\boldsymbol{i}}$ as

$$D = \bigoplus_{\boldsymbol{i} \in \mathbb{I}} D^{\boldsymbol{i}}, \tag{21}$$

where $\boldsymbol{i} := (i_1, i_2, i_3, i_4)^T$ is a 4-dimensional integer vector that moves in a 4-dimensional integer lattice, defined by

$$\mathbb{I} = [1, 2, \cdots, m_1] \times \cdots \times [1, 2, \cdots, m_4]. \tag{22}$$

We then introduce center points of the subcubes $\boldsymbol{y}^{\boldsymbol{i}} \in D^{\boldsymbol{i}}$ ($\boldsymbol{i} \in \mathbb{I}$), whose jth component is given by,

$$(\boldsymbol{y}^{\boldsymbol{i}})_j := a_j + \left((\boldsymbol{i})_j - \frac{1}{2}\right) \frac{b_j - a_j}{m_j}, \tag{23}$$

where $(\boldsymbol{v})_j$ denotes the jth component of a vector $\boldsymbol{v}$.

The abovementioned formula allows us to build a numerical method as follows:

1) As an offline learning procedure, the mapping $\bar{C}$:

$$\bar{C}(\boldsymbol{i}) := \nu \quad \text{if} \quad \lim_{t \to \infty} \boldsymbol{\phi}_t\big(h(\boldsymbol{y}^{\boldsymbol{i}}), \boldsymbol{T}\big) = \boldsymbol{\omega}_\nu \tag{24}$$

is numerically stored by solving (8) from $\boldsymbol{\xi}_0 = h(\boldsymbol{y}^{\boldsymbol{i}})$.

2) When the IC is in process, the classifier C is quantized by C^* as

$$C(\boldsymbol{y}) \approx C^*(\boldsymbol{y}) := \bar{C}(\boldsymbol{i}) \quad \text{for } \boldsymbol{i} \text{ of } \boldsymbol{y} \in D^{\boldsymbol{i}}. \tag{25}$$

In this method, the accuracy of the classifier C^* depends on structure of the measurement H, the size and placement of the measuring range $[a_j, b_j]$, and the resolution of quantization of the reachable set m_j ($j = 1, \cdots, 4$).

In summary, we obtain the IC for the left-hand agent as a composed function of the quantized classifier C^*, the selector S, and the impulse generator G in a closed-loop form as

$$\begin{aligned} u_1^{\mathrm{ic}} = u_1^{\mathrm{ic}}\big(\boldsymbol{y}(t); \mathbb{J}\big) &:= P_1 \cdot (G \circ S_{\mathbb{J}} \circ C^*)\big(\boldsymbol{y}(t)\big) \\ &= P_1 \cdot \big(G \circ S_{\mathbb{J}} \circ C^* \circ H\big)\big(\boldsymbol{x}(t)\big). \end{aligned} \tag{26}$$

If τ_G is sufficiently large, it is implied that the solution $\boldsymbol{x}(t)$ in (8), starting from certain initial states in D, undergoes an impulsive force at the time $t = t_0$ decided autonomously by u_1^{ic} and that the solution converges to the stable equilibriums specified by $\mathbb{J}$, under the resolution limit $\min_j(m_j) \to \infty$.

IV. Low Performance of the Proposed IC

A. Performance Functions

Evaluation of the individual performance of the proposed IC is based on an impulse response of (8) with the IC, given by

$$\begin{aligned} &\boldsymbol{v} = (v(t), 0)^T,\ v(t) := QI_{\Delta\tau}(t),\ \mathbb{J}_1 := \{2, 3\}, \\ &\boldsymbol{u}^{\mathrm{ic}} := \big(u_1^{\mathrm{ic}}(\boldsymbol{y}(t); \mathbb{J}), 0\big)^T, \end{aligned} \tag{27}$$

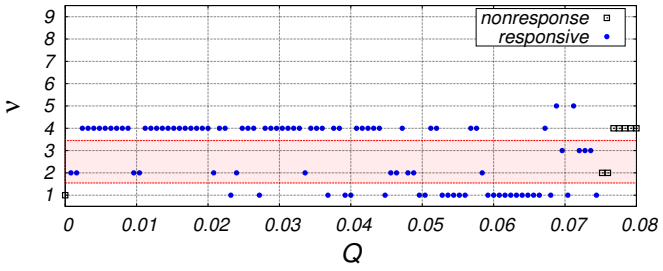

Fig. 4. The final state index ω_ν as a function of the initial disturbance strength Q for $m = 100$.

where $v(t)$ is a initial disturbance and Q is a strength of impulse. Solving (8) with (27) numerically from a trivial initial state $\boldsymbol{x}(0) = \boldsymbol{o}^{(8)} + w_0 \boldsymbol{e}_5^{(8)}$ for a given Q, we obtain the final position $\boldsymbol{\omega}_\nu$ and the correspondence from Q to ν. Here, Q is taken at $N_Q = 100$ uniform grid points over the interval $[0, Q_{\max}]$ ($Q_{\max} = 0.08$). For each Q, we restrict u_1^{ic} to output only a single impulse. We represent a result of the impulse response as an element of the following Cartesian product:

$$\Omega = \{\mathrm{T}, \mathrm{F}\} \times \{1, \cdots, 9\}, \tag{28}$$

where T and F represent presence and absence of the output of the IC, respectively. We then classify the set (28) as

$$\begin{cases} \Omega_{\mathrm{T}}^j := \{\mathrm{T}\} \times \mathbb{J}_j \subset \Omega, \quad (j = 1, 2, 3), \\ \Omega_{\mathrm{F}} := \{\mathrm{F}\} \times \{1, \cdots, 9\} \subset \Omega, \end{cases} \tag{29}$$

and introduce the following performance functions:

$$\begin{cases} E_1 := N(\Omega_{\mathrm{T}}^1)/N_Q \times 100(\%), & \mathbb{J}_1 := \{2, 3\}, \\ E_2 := N(\Omega_{\mathrm{T}}^2)/N_Q \times 100(\%), & \mathbb{J}_2 := \{4, 7\}, \\ E_3 := N(\Omega_{\mathrm{T}}^3)/N_Q \times 100(\%), & \mathbb{J}_3 := \overline{\mathbb{J}_1 \cup \mathbb{J}_2}, \\ E_4 := N(\Omega_{\mathrm{F}})/N_Q \times 100(\%), \end{cases} \tag{30}$$

where $N(A)$ denotes the number of outcomes that belong to $A \subset \Omega$. We call E_1 the winning rate, E_2 the losing rate, E_3 the draw rate, and E_4 the nonresponse rate.

B. Low Performance Problem

The model parameters are set to the values listed in Table I. The quantization resolution m_j is set to $m = 100$ for all j. The impulse strength of the IC is taken as $P_1 = 0.0744$, below the threshold at which the impulse stops producing switching motions, from the trivial initial position $\boldsymbol{\omega}_1$ to the other positions. The measurement region D is taken as

$$\begin{aligned} D := &[-2.39, 0.57] \times [-3.63, 12.57] \\ &\times [-0.56, 0.39] \times [-4.38, 6.86], \end{aligned} \tag{31}$$

which circumscribes at least all of the trajectories $\boldsymbol{y} = (\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2)^T$ for the disturbance of $0 \le Q \le 0.08$. For the numerical integration, a fourth-order Runge-Kutta-Gill method is employed with a time step $\Delta t = 5 \times 10^{-4}$s.

Figure 4 shows the resulting final state index ν as a function of the initial disturbance strength Q. The small circles and the small squares represent $\nu(Q)$ in the presence

and absence of the IC output, respectively. The range of the target positions, defined by $\mathbb{J}_1$, is hatched in red. The winning rate is evaluated as $E_1 = 15.9\%$, the losing rate as $E_2 = 48.5\%$, the draw rate as $E_3 = 27.7\%$, and the nonresponse rate as $E_4 = 7.9\%$. It can be thus seen that the IC developed in Section III does not perform well.

In our previous study [13], it was shown that this low performance can be caused by the proposed classifier that sometimes predict a final state that disagrees with the final state to which the CIP model actually converges, and that this occurs near the boundary points of the reachable sets of the model [13]. Although, in theory, a sufficiently large resolution m provides reachable sets that are nearly exact, which greatly increases the computational load.

V. PERFORMANCE IMPROVEMENT BY REMOVING BOUNDARY POINTS

We propose a preprocessing method that can solve the low performance discussed in Section IV by removing the boundary points from the LUT.

A. Removing Boundary Points from the LUT

We consider a reachable set limited in D and its quantization as

$$\hat{\Psi}_\nu := D \cap \Psi_\nu, \quad \Lambda_k := \{\boldsymbol{i} \in \mathbb{I} \mid \boldsymbol{y}^{\boldsymbol{i}} \in \hat{\Psi}_k\} \subset \mathbb{I}, \tag{32}$$

and introduce direct sums of (32) with respect to $\mathbb{J}$ as

$$\hat{\Psi}_{\mathbb{J}} := \bigoplus_{k\in\mathbb{J}} \hat{\Psi}_k \subset D, \quad \Lambda_{\mathbb{J}} := \bigoplus_{k\in\mathbb{J}} \Lambda_k \subset \mathbb{I}, \tag{33}$$

respectively. Table II summarizes the correspondence between quantities in the measurement space and those in the quantized space (lattice).

TABLE II

CORRESPONDENCE BETWEEN VARIOUS QUANTITIES IN THE MEASUREMENT SPACE AND THOSE IN THE QUANTIZED SPACE.

	Measurement space	Lattice
Range	D	$\mathbb{I}$
Element	$\boldsymbol{y} \in D$	$\boldsymbol{i} \in \mathbb{I}$
Reachable set limited in D	$\hat{\Psi}_\nu \subset D$	$\Lambda_\nu \subset \mathbb{I}$
Reachable set limited in D (w.r.t. $\mathbb{J}$)	$\hat{\Psi}_{\mathbb{J}} := \bigoplus_{k\in\mathbb{J}} \hat{\Psi}_k \subset D$	$\Lambda_{\mathbb{J}} := \bigoplus_{k\in\mathbb{J}} \Lambda_k \subset \mathbb{I}$
Neighborhood	undefined	$\mathcal{N}_1(n;\boldsymbol{c})$, $\mathcal{N}_\infty(n;\boldsymbol{c})$
Boundary	$\partial\hat{\Psi}_{\mathbb{J}}(n)$	$\partial\Lambda_{\mathbb{J}}(n)$

In order to define a boundary of the reachable sets, we consider the complement $\bar{\Lambda}_{\mathbb{J}} := \mathbb{I} \setminus \Lambda_{\mathbb{J}}$ and a lattice point $\boldsymbol{c} \in \mathbb{I}$. Introducing a neighborhood $\mathcal{N}_s(n;\boldsymbol{c})$ of $\boldsymbol{c} \in \mathbb{I}$, specified below, we define a boundary as

$$\partial\lambda_{\mathbb{J}} := \bigcup \left\{ \Lambda_{\mathbb{J}} \cap \mathcal{N}_s(n;\boldsymbol{c}), \;\middle|\; \boldsymbol{c} \in \bar{\Lambda}_{\mathbb{J}} \right\}. \tag{34}$$

Hereafter, we define $\Lambda^\circ_{\mathbb{J}} := \Lambda_{\mathbb{J}} \setminus \partial\Lambda_{\mathbb{J}} \subset \mathbb{I}$ as the interior of $\Lambda_{\mathbb{J}}$.

TABLE III

ELEMENTS SELECTED BY PROJECTION $\mathcal{P}$.

Index k	Projection Operator	Selected elements θ_1	$\dot{\theta}_1$	θ_2	$\dot{\theta}_2$
1	$\mathcal{P}_{(1)} := id$	○	○	○	○
2	$\mathcal{P}_{(2)} := \mathcal{P}_{34}$	○	○		
3	$\mathcal{P}_{(3)} := \mathcal{P}_{24}$	○		○	
4	$\mathcal{P}_{(4)} := \mathcal{P}_{23}$	○			○
5	$\mathcal{P}_{(5)} := \mathcal{P}_{14}$		○	○	
6	$\mathcal{P}_{(6)} := \mathcal{P}_{13}$		○		○
7	$\mathcal{P}_{(7)} := \mathcal{P}_{12}$			○	○
8	$\mathcal{P}_{(8)} := \mathcal{P}_4$	○	○	○	
9	$\mathcal{P}_{(9)} := \mathcal{P}_3$	○	○		○
10	$\mathcal{P}_{(10)} := \mathcal{P}_2$	○		○	○
11	$\mathcal{P}_{(11)} := \mathcal{P}_1$		○	○	○
0	$\bar{C}(\boldsymbol{i}) = \bar{C}'(n;\mathcal{P};\boldsymbol{i})$				

We define a neighborhood lattice of $\boldsymbol{c}$ of radius n as a 4-dimensional sphere with the radius n centered at $\boldsymbol{c}$:

$$\mathcal{N}_s(\boldsymbol{c};n) := \left\{ \boldsymbol{i} \in \mathbb{I} \;\middle|\; \| \boldsymbol{i} - \boldsymbol{c} \|_s \leq n \right\} \setminus \{\boldsymbol{c}\}, \quad (s = 1, \infty). \tag{35}$$

In particular, the neighborhood lattice using infinity-norm $\| \boldsymbol{i} \|_\infty$ is called the ∞-neighborhood lattice, written as $\mathcal{N}_\infty$. Correspondingly, the neighborhood lattice using 1-norm $\| \boldsymbol{i} \|_1$ is called the 1-neighborhood lattice, written as $\mathcal{N}_1$.

We expand the neighborhood lattice in (35) by introducing a projection operator $\mathcal{P}$ as

$$\mathcal{N}_s(\boldsymbol{c};n,\mathcal{P}) := \left\{ \boldsymbol{i} \in \mathbb{I} \;\middle|\; \| \mathcal{P}(\boldsymbol{i}) - \mathcal{P}(\boldsymbol{c}) \|_s \leq n \right\} \setminus \{\boldsymbol{c}\} \tag{36}$$

for $s = 1, \infty$. Specifically, we introduce projections $\mathcal{P}_i$ and $\mathcal{P}_{ij}$ as follows. For an d-dimensional vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_d)^T$, we define

$$\mathcal{P}_i(\boldsymbol{x}) := (\cdots, x_{i-1}, x_{i+1}, \cdots)^T \in R^{d-1}, \tag{37}$$

where $\mathcal{P}_i(\boldsymbol{x})$ represents the $(d-1)$-dimensional vector obtained by removing the ith element from $\boldsymbol{x}$. Similarly, we define

$$\mathcal{P}_{ij}(\boldsymbol{x}) := (\cdots, x_{i-1}, x_{i+1}, \cdots, x_{j-1}, x_{j+1}, \cdots)^T \in R^{d-2}, \tag{38}$$

where $\mathcal{P}_{ij}(\boldsymbol{x})$ represents the $(d-2)$-dimensional vector obtained by removing the ith and jth elements from $\boldsymbol{x}$.

The boundary can therefore be defined as

$$\partial\Lambda^s_{\mathbb{J}}(n,\mathcal{P}) := \bigcup \left\{ \Lambda_{\mathbb{J}} \cap \mathcal{N}_s(\boldsymbol{c};n,\mathcal{P}) \;\middle|\; \boldsymbol{c} \in \bar{\Lambda}_{\mathbb{J}} \right\}. \tag{39}$$

The elements selected by projection operators $\mathcal{P}_i$ and $\mathcal{P}_{ij}$ are listed in Table III where id denotes identity operator and $\bar{C}'(\boldsymbol{i};n,\mathcal{P})$ is as specified in the next section. Note that $k = 0$ produce the IC developed in Section III.

B. Modification of the IC

Considering the boundary derived above, we can rewrite the mapping $\bar{C}(\boldsymbol{i})$ in (24) into $\bar{C}'(\boldsymbol{i};n,\mathcal{P})$, using the following algorithm:

function $\bar{C}' =$ *RewriteBorder*$(\bar{C})$
$\bar{C}'(\boldsymbol{i};n,\mathcal{P}) = \bar{C}(\boldsymbol{i})$
for $\boldsymbol{c} \in \bar{\Lambda}_{\mathbb{J}}$

for $\boldsymbol{i} \in \mathcal{N}_s(\boldsymbol{c}; n, \mathcal{P})$
if $\boldsymbol{i} \in \Lambda_{\mathbb{J}}$ *then* $\bar{C}'(\boldsymbol{i}; n, \mathcal{P}) = -n$ *endif*
endfor
endfor
endfunction

This algorithm produces

$$\bar{C}'(\boldsymbol{i}; n, \mathcal{P}) := -n \quad \text{if} \quad \boldsymbol{i} \in \partial\Lambda^s_{\mathbb{J}}(n, \mathcal{P}), \tag{40}$$

providing a modified classifier as

$$C^*(\boldsymbol{y}; n, \mathcal{P}) := \bar{C}'(\boldsymbol{i}; n, \mathcal{P}) \quad \text{for } \boldsymbol{i} \text{ s.t. } H\boldsymbol{x} \in D^{\boldsymbol{i}}. \tag{41}$$

Similar to (26), we derive the following modified IC.

$$\begin{aligned} u_1^{\text{ic}} &= u_1^{\text{ic}}\big(\boldsymbol{y}(t); \mathbb{J}, n, k\big) \\ &:= P_1 \cdot G \circ S_{\mathbb{J}} \circ C^*(\bullet; n, \mathcal{P}_{(k)})\big(\boldsymbol{y}(t)\big). \end{aligned} \tag{42}$$

As the modified IC in (42) does not generate the impulsive force for $\boldsymbol{x}(t)$ crossing the boundary, the misclassifications addressed in Section IV are expected to reduce.

C. Performance Evaluation of the Modified IC

We evaluate the performance of the modified IC in (42) for $\boldsymbol{u}^{\text{ic}} := (u_1^{\text{ic}}\big(\boldsymbol{y}(t); \mathbb{J}, n, k\big), 0)^T$. We calculate the performance functions in (30) for all combination of the neighborhood lattices $\mathcal{N}_s(\boldsymbol{c}; n, \mathcal{P})$ $(s = 1, \infty)$ in (36) and the projection operators $\mathcal{P} = \mathcal{P}_{(k)}$ $(k = 1, \cdots, 9)$ in Table III.

The results of the best performance are obtained for $\mathcal{N}_1$ and $k = 4$, 7, and 10. Note that these results of the best performance take the same values of the performance functions. Specifically, we obtain a winning rate of $E_1 = 61.4\%$, a losing rate of $E_2 = 1.0\%$, a draw rate of $E_3 = 19.8\%$ and a nonresponse rate of $E_4 = 17.8\%$.

Therefore, the modified IC yields the following performance:

- The winning rate E_1 increases from 15.9% to 61.4%.
- The losing rate E_2 decreases from 48.5% to 1.0%.
- The draw rate E_3 decreases from 27.7% to 19.8%.
- The nonresponse rate E_4 increases from 7.9% to 17.8%.

The results show that the method proposed in this study improves the IC performances except the nonresponse rate.

VI. CONCLUSION

We developed a preprocessing method to improve the performance of an IC, in an attempt to solve the competitive motion control problems of the CIP model. In the proposed method, the boundary points were removed from a LUT-based classifier that stores dynamical correspondence from the initial to the final states. The modified LUT in our proposed method was benchmarked with the modified IC. The performance of the resulting controller was numerically evaluated.

The results of the performance evaluation suggest that the proposed method improves the overall performance of the IC by increasing the winning rate E_1 and decreasing the losing rate E_2 and the draw rate E_3. However, a negative result was that the nonresponse rate E_4 also increased.

In future work, we plan to develop a physical CIP system equipped with the proposed IC to experimentally demonstrate the competitive behavior.

REFERENCES

[1] K. Sigmund and J. Hofbauer, *Evolutionary Games and Population Dynamics*, Cambridge University Press, 1998.

[2] P. Hsu, "Coordinated control of multiple manipulator systems," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 4, 1993, pp. 400-410.

[3] Y. Nakamura, K. Nagai and T. Yoshikawa, "Mechanics of coordinative manipulation by multiple robotic mechanisms," *Proceedings of 1987 IEEE International Conference on Robotics and Automation*, vol. 4, 1987, pp. 991-998.

[4] V. Panwar, N. Kumar, N. Sukavanam and J. Borm, "Adaptive neural controller for cooperative multiple robot manipulator system manipulating a single rigid object," *Applied Soft Computing*, vol. 12, no. 1, 2012, pp. 216-227.

[5] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Autonomous Robots*, vol. 8, no. 3, 2000, pp. 345-383.

[6] E. Pagello, A. D'Angelo, F. Montesello, F. Garelli and C. Ferrari, "Cooperative behaviors in multi-robot systems through implicit communication," *Robotics and Autonomous Systems*, vol. 29, no. 1, 1999, pp. 65-77.

[7] J. Zhang, D. Gong and Y. Zhang, "A niching PSO-based multi-robot cooperation method for localizing odor sources," *Neurocomputing*, vol. 123, 2014, pp. 308-317.

[8] A. Hernansanz, A. Casals and J. Amat, "A multi-robot cooperation strategy for dexterous task oriented teleoperation," *Robotics and Autonomous Systems*, vol. 68, 2015, pp. 156-172.

[9] A. Nelson, E. Grant and T. Henderson, "Evolution of neural controllers for competitive game playing with teams of mobile robots," *Robotics and Autonomous Systems*, vol. 46, no. 3, 2004, pp. 135-150.

[10] J. Wu, V. Snášel, E. Ochodková, J. Martinovič, V. Svatoň and A. Abraham, "Analysis of strategy in robot soccer game," *Neurocomputing*, vol. 109, 2013, pp. 66-75.

[11] K. Yoshida and H. Ohta, "Coupled Inverted Pendula Model of Competition and Cooperation," *Journal of System Design and Dynamics*, vol. 2, no. 3, 2008, pp. 727-737.

[12] J. Higeta, Y. Otaka and K. Yoshida, "Switching Equilibriums of Nonlinear Dynamical Systems by Human Manipulation," *Trans. JSME, Series C (in Japanese)*, 77-784, 2011, pp. 232-243.

[13] K. Yoshida, S. Matsumoto and Y. Matsue, "Artificial Wrestling: A Dynamical Formulation of Autonomous Agents Fighting in a Coupled Inverted Pendula Framework," *The 12th International Conference on Motion and Vibration Control (MOVIC 2014)*, 2014, No. 2C11.

[14] Y. Liu, M. Wiercigroch, J. Ing and E. Pavlovskaia, "Intermittent control of coexisting attractors," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, p. 20120428.

[15] J.C. Sprott, X. Wang and G. Chen, "Coexistence of point, periodic and strange attractors," *International Journal of Bifurcation and Chaos*, vol. 23, no. 5, 2013, p. 1350093.

[16] R. Bayadi, R.N. Banavar and D.E. Chang, "Characterizing the reachable set for a spacecraft with two rotors," *Systems & Control Letters*, vol. 62, no. 6, 2013, pp. 453-460.

[17] M. Lhommeau, L. Jaulin and L. Hardouin, "Capture basin approximation using interval analysis," *International Journal of Adaptive Control and Signal Processing*, Vol. 25, no. 3, 2011, pp. 264-272.