Received: September 28, 2019. Revised: November 22, 2019.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Decentralized Control of Cooperative Robotics System Using Neural Networks for the Purpose of an Object Balancing Flat Plate Tasks

Nattapon Jaisumroum[1] **Pholchai Chotiprayanakul**[1] **Wipoo Sriseubsai**[1*]

[1]*Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Thailand*
* Corresponding author's Email: wipoo.sr@kmitl.ac.th

Abstract: In this article, the manipulation to handle the object on a plate using neural networks, design of 1-DOF robot arm under cooperative control, will be explained. The robot's system specifies the object position and velocity, an assignment oriented components for cooperative control. The novelty of this experiment is that under the decentralized control, the robot estimates the position and speed of the object for control end-effector of robot arm using the camera to track position and speed of the object according to training and assigned collaborative tasks which differ from other experiments that use sensors. The experiment includes three robot manipulators which were capable balancing the objects on flat plate with dataset to training and control servo motor assigned to the corresponding position and the end-effector in decentralized to control robotics. Overall, neural network method can be a training scheme using a cooperative robotics in a decentralized control.

Keywords: Decentralized control, Neural networks, Cooperative robotics, Dynamic object manipulation.

1. Introduction

The robotics system can be contained in a single and multiple manipulators, and the complicated task accepting multi-robot has exceptional accommodation than a single robot to essentially handle and assembly heavy materials. A number of researchers have studied cooperative control in multiple robots for various tasks. Learning object handling practice with robotics is a challenging obstacle since the trajectories to propose appropriate handling practice perhaps varied with respect to different situations. Even when it comes to handling the duplicated objects, the evolution of motor in time would be a little different provided to the robotics, and the object movement is located in workspace. This paper shows that an artificial neural network method is adequate for the training on the practice for robotics collaborative balancing of the object. There are many articles of the previous research on the robotic learning for object handling [1]. The improvements on controlling a DC servo motors are obvious positioned and speed in flexible control. DC servo motors are generally used for a collection in industry, robotic systems. The control specifies the motion of the object being grasped and handling the object by the robot's arm, the latter of which has no effect on the object movement. The coordinates depend on the impedance control [2, 3] and hybrid control in force or position [4, 5] and the parameter of manipulator where the objects are placed. This is the recognition on the accuracy to determine the parameter between manipulator and the object position. In order to control multi-robot dynamic parameters uncertainty task, neural networks method [6], learning and training methods [7] and fuzzy sliding mode method [8, 9] in control are suggested. These techniques involve the measurement of times and forces at an end-effector and the adoption of a centralized controller. The sensor is comparatively costly and readily damaged. Centralized control methods are implementation in previous researches of the robotics collaboration. Decentralized control methods [10, 11] in manipulator is governed independently with the controller, are therefore desirable methods due to their simple model. The neural network control is a very successful to solve in nonlinear problems. Thus, neural networks method develops to be an influential tool for learning in linear and extremely nonlinear problems on dynamic systems. The neural network method, rapid variation and elemental approximation proficiency, have appeal to considerable in a researches of robotic engineering, identification and control. In the literature, the use of neural network architecture to regulate complicated and uncertainty in systems has become a subject of significant considerable [12-14]. Most of the research in this field has created and developed the new techniques of object manipulation in a decade. Non-prehensile dynamic of object in manipulation approaches in fields of robotic these new methods. The manipulation of object implies the robotics with no grasp, i.e. the robotic can easily balance object, even with no other robotics during the cooperative assignment [15]. However, these decentralized controllers are adaptive and do not require a sensor to evaluate the relationships between the robotics interaction and the target object. This article proposes a decentralization in neural network control technique without a sensor for multi-robot can be detect the object using web camera, robotic eyes to collaborate balancing the object with other robot interactions. In this study, the parameters of the robotic arms and the object movement are approximated and the motions of robotic cooperating by referring to the neural network model will be explained. For example, the object handling task, the lessons learned by the robots on choosing between the predefined behaviors aboriginal, approaching releasing and carrying an object for particular step. However, this method can be applied to a dynamic object handling behavior [16]. In this study, decentralized control implementing neural network method in cooperative tasks, rather than using sensors the same as other experiments, for balancing the object position, velocity and manipulations utilizing a robot arm are experimented.

2. Methodology

2.1 Dynamics system modeling

The time pivot is included to the position in coordinate and robotics are done in dynamics of object handling system and in rotating processes in specific. The dynamic of object handling methods in manipulation techniques is rolling on flat-plate where balancing with a robot arm, robotics cooperative in balancing the object in rolling operation as shown in Fig. 1.

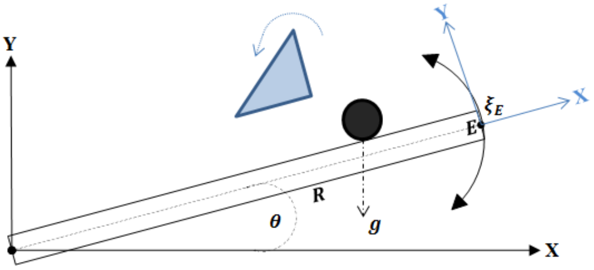


Figure.1 Rolling operation, object of robotics arm, 1-DOF

Transformations of the equation as:

$$x = R\,cos\theta\,, y = R\,sin\theta\,;\ \mathrm{E} = \mathrm{Rot}(\theta)\mathrm{T}_x(R)$$

$$\mathrm{E} = \begin{pmatrix} \cos\theta & -\sin\theta & R\cos\theta \\ \sin\theta & \cos\theta & R\sin\theta \\ 0 & 0 & 1 \end{pmatrix} \qquad (1)$$

The dynamic object manipulation in general and rolling system is particular. The robot is therefore used to create the dynamics scheme rather determined against [17, 18]. Thus, the dynamic manipulation of object, the coordinate of object and robotics are both essential and the manipulators must be built to be ensure achievement of a modeling assignment. This is essential to determine collection of the manipulator object schemes that can be controlled [19].

The distance from the center of the object regarding the jth in pivot, radius. The d^i is defined the distance during the object's center and the upper of robot arm, the coordinates of the center of the object devise set $\left(R,\ s + d^i\right)$ in frame $\mathcal{F}_j$. The arm of robotic starts with the angle of the manipulator is set $(\theta = \theta_i)$ and the object position on the plate. Assume; the robot arm moves down from $\left(\theta = \theta_i,\ \dot{\theta} = 0\right)$ to $\left(\theta = -\theta_0 < 0,\ \dot{\theta} = 0\right)$ in first time,t_1 seconds. Next; the robotic moves to $\left(\theta = \theta_r,\ \dot{\theta} = \omega_r\right)$ in the second time, t_2 seconds. Where,θ_r is the angle of plate along x axis. Last phase, the robot arm decelerates fast into the object will move in high speed. The object is captured and taking angle θ_c see in Fig. 2. As is normal with manipulator motion planning, the trajectories desired in two phases can be considered as in Eqs. (2), (3), (4) and (5) respectively:

$$\theta(t) = \frac{2(\theta_0+\theta_i)}{t_1^3}t^3 - \frac{3(\theta_0+\theta_i)}{t_1^2}t^2 + \theta_i \qquad (2)$$

$$\omega(t) = \dot{\theta}(t) = \frac{6(\theta_0+\theta_i)}{t_1^3}(t^2 - t_1 t) \qquad (3)$$

$$\theta(t) = \frac{t_2\omega_r - 2(\theta_0+\theta_r)}{t_2^3}t^3 - \frac{t_2\omega_r - 3(\theta_0+\theta_r)}{t_2^2}t^2 - \theta \tag{4}$$

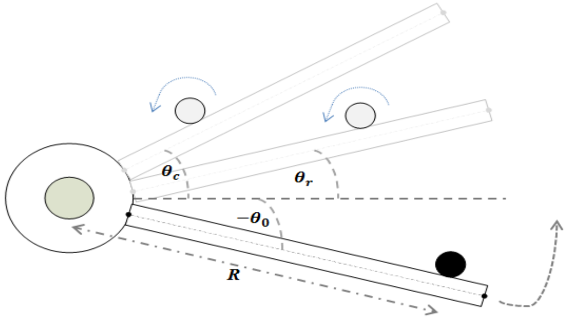


Figure.2 The robot arm handling an object balancing task

$$\omega(t) = \dot{\theta}(t) = \frac{3(t_2\omega_r - 2(\theta_0+\theta_r))}{t_2^3}t^2 - \frac{2(t_2\omega_r - 3(\theta_0+\theta_r))}{t_2^2}t \tag{5}$$

After an impulsive throw, $t = 0$ and $\theta = \theta_r$ movement of the frame $\mathcal{F}^i$ defines itself as:

$$\begin{aligned} x(t) &= R\cos(\theta_r) - (s+d)\sin(\theta_r) \\ &\quad - (R\sin(\theta_r) \\ &\quad + (s+d)\cos(\theta_r))(\omega_r t + \theta_r) \\ y(t) &= R\sin(\theta_r) \\ &\quad + (s+d)\cos(\theta_r) \\ &\quad + (R\cos(\theta_r) \\ &\quad - (s+d)\sin(\theta_r))\,(\omega_r t + \theta_r) \\ &\quad + \frac{1}{2}gt^2 \\ \theta(t) &= \theta_r + \omega_r t \end{aligned} \tag{6}$$

Where $\emptyset(t)$, the angle of p^i axis of $\mathcal{F}^i$ in $\mathcal{F}$ and g, the gravity acceleration, obvious into $x\ direction$. The velocity of the object is $-(R\sin(\theta_r) + (s+d)\cos(\theta_r))\omega_r$ which is negative for $\theta_r < -\tan^{-1}\big((s+d)/R\big)$ and is positive for $\theta_r > -\tan^{-1}\big((s+d)/R\big)$, those correspond to backward and forward throw respectively.

The perceptive motion in vertices of the object is concern in V^ivertex, its originally situated into $p_{V^i,}\, q_{V^i,}$ in $\mathcal{F}^i$, its motion with respect to $\mathcal{F}$ can be written as:

$$\begin{bmatrix} x_{V^i}(t) \\ y_{V^i}(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} \cos(\omega_r t + \theta_r) & -\sin(\omega_r t + \theta_r) \\ \sin(\omega_r t + \theta_r) & \cos(\omega_r t + \theta_r) \end{bmatrix} \begin{bmatrix} p_{V^i} \\ q_{V^i} \end{bmatrix} \tag{7}$$

2.2 Robotic handling of circular object on plate

The dynamic of handling object situation holds circular items on flat-plate in balancing tasks. Despite, begin to investigate the primary problems under this research including circular object show in Fig. 3, which can describe our strategy to be easier to understand in Eq. (6), is accurate on the motion of object.

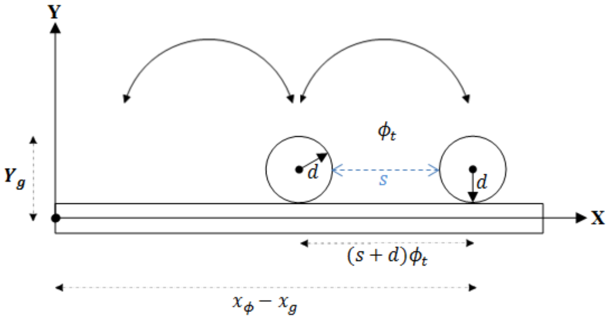


Figure.3 Handling of a circular object

Where; $\emptyset(t) = \omega_r t + \theta_r$, the rotation moment of the object situated at time, $x(t)$ and $y(t)$ can be rewritten as:

$$x(t) = R\cos(\theta_r) - (s+d)\sin(\theta_r) - (R\sin(\theta_r) + (s+d)\cos(\theta_r))\emptyset(t) \tag{8}$$

$$y(t) = \frac{g(\emptyset(t) - \theta_r)^2}{2\omega_r^2} + R\sin(\theta_r) + (s + d)\cos(\theta_r)(R\cos(\theta_r)\ (s+d)\sin(\theta_r))\,\emptyset(t) \tag{9}$$

Where d is the radius of an object, as for the reality that at the end of each throw, $y(t) = y_c$ (9) results to be distinctive ω_r for each rotation required $\emptyset(t)$ that is, in other words, ω_r and $\emptyset(t)$ have a one-to-one correspondence in each spin. It's worth noting that y_c this is the trigonometric function of the angle of catch. According to (8) horizontal displacement of the object during the fth throw (l_f) is:

$$l_f = R_f - x(t_{\text{flight}}) = R_f - R_f\cos(\theta_{rf}) + (s+d)\sin(\theta_{rf}) + \big(R_f\sin(\theta_{rf}) + (s+d)\cos(\theta_{rf})\big)\emptyset_f(t_{\text{flight}}), f = 1,..,F1 \tag{10}$$

Where $\emptyset_f$ and R_f represent the rotations of the object and radius during the fth throw respectively, t_{flight} is time of flight and $F1$ is number of throws. The total horizon distance covered after $F1$ throws is:

$$l_{\text{tot}} = \sum_{f=1}^{F1}\big(R_f - R_f\cos(\theta_{rf}) + (s + d)\sin(\theta_{rf})\big) + \big(R_f\sin(\theta_{rf}) + (s + d)\cos(\theta_{rf})\big)\emptyset_f(t_{\text{flight}}) \tag{11}$$

Assume that the objective position and orientation of the object after $F1$ throws are X_g and $\emptyset_0$ radians respectively, the aim of the arranging is to discover a set of $F1$ throws such that the object reaches its target for configuration. The accepting drive and catching angles as specification allows space for further supplied motion planning of the robotic, despite the complexity on the algorithm of planning. Furthermore, the established balance at non-zero angles is concentrate. In addition, it is simpler to implement a deceleration system for fixed angles at set corners. Therefore, as in other researchers, we assume catching and delivering at angle zero. In this study, hypothesis define scheme to forward and backward deliver. On the other hand, we still have a set of parameters to achieve for motion of robotics planning. The parameter will be implemented in the successive section. This article, beyond casualty of principle to consider backward throws. In forward throws, scheduling technique submitted could be presented [20, 21] which is reformulated and analyzed in a following section. If $\theta_r = 0$ Eq. (8) and (9) are simplified to:

$$x(t) = R - (s+d)\emptyset(t) \tag{12}$$

$$y(t) = \frac{g\big(\emptyset(t)\big)^2}{2\omega_r^2} + (s+d) + R\emptyset(t) \tag{13}$$

With,$\theta_c = 0$, we have catching at $y_c = s + d$ for all but the last throw in which $y_c = Y_g$. Using (12) horizontal displacement during fth throws is given by:

$$l_f = R_f - x\big(t_{\text{flight}}\big) = (s+d)\emptyset_f\big(t_{\text{flight}}\big),\\ f = 1, \ldots, F1 \tag{14}$$

Now, the total covered horizontal distance is:

$$l_{\text{tot}} = \sum_{f=1}^{F1} l_f = (s+d)\sum_{f=1}^{F1}\emptyset_f\\ = (s+d)\emptyset_{\text{tot}} \tag{15}$$

The desired total orientation changed after $F1$ throws ($\boldsymbol{\emptyset_0}$) resulting in the covered horizontal distance can be written as:

$$\emptyset_{\text{tot}} = \emptyset_0 + 2k\pi,\ \ k \in Z\ ,\\ l_{\text{tot}} = (s+d)(\emptyset_0 + 2k\pi) \tag{16}$$

Now, starting from position X_0 , if the goal position be $X_g < 0$ in the frame $\mathcal{F}$, according to Fig.3, we need to have: $l_{\text{tot}} = X_0 - X_g$ and Eq. (16) we're obtaining as:

$$\frac{1}{2\pi}\left(\frac{X_0 - X_g}{s+d} - \emptyset_0\right) = k = \text{integer} \tag{17}$$

Given $\emptyset_0$ and X_0, the goal condition $\left(\emptyset_g, X_g\right)$ is reachable only if X_g satisfies (18). So it is impossible to move a circular object into all desired configurations.

Corollary1. Suppose the circular object hold on the robot arms at position $\mathrm{X_0}$ in the $\mathcal{F}$ frame. If required, an object orientation will be changed $\emptyset_\mathrm{g} = \emptyset_0$ radians after F1 sequential throws and then it can only be shifted to positions $\mathrm{X} = \mathrm{X_g}$ that fulfill condition (17).

2.3 Kinematics of robotic

The kinematics of the cooperative robots is two Jacobian matrices relate velocity of the object and the position of the end-effector. In Fig. 4 shows the control structure proposed in this study considering the coordinated cooperative control of manipulators. The kinematics of robot arm taking the connection between joint's velocities $\dot{q}$, interrelated the end-effector with angular velocity w_ω and translational velocity w_v [22]. They are related through the geometric Jacobian $J_g(q)$ [23]:

$$\begin{bmatrix} w_v \\ w_\omega \end{bmatrix} = J_g(q)\dot{q} \tag{18}$$

The end-effector (orientation and position) is indicated via regarding an illustration in the robotics work space; it is available to estimate the Jacobian method through differentiation of the handle kinematics complying to joint positions. The Jacobian method, describe analytical of Jacobian, $J_g(q)$, is related to the commutative Jacobian as:

$$J_g(q) = \begin{bmatrix} I & 0 \\ 0 & T_{(q)} \end{bmatrix} J_g(q) \tag{19}$$

Where $T_{(q)}$ is a transformation matrix that depends on the parameter of the end-effector orientation.

2.4 Camera visions for control of robotics

In this study, we utilized facilitates of the intrigued focuses on the plane of image to represent a vector. The definition of parameters established the DC servo control scheme. The image based in servo control design is recognizable the vector *s* comprises a set of optical appearance acquired from the image plane [24]. The parameter m performs the image evaluation while a is refers to intrigued focuses on parameters of camera. The relation of frame, image and camera is shown in Fig. 5. The point of 3D, P cans projection toward the image plane as point in 2D adopting the proposed projection as:

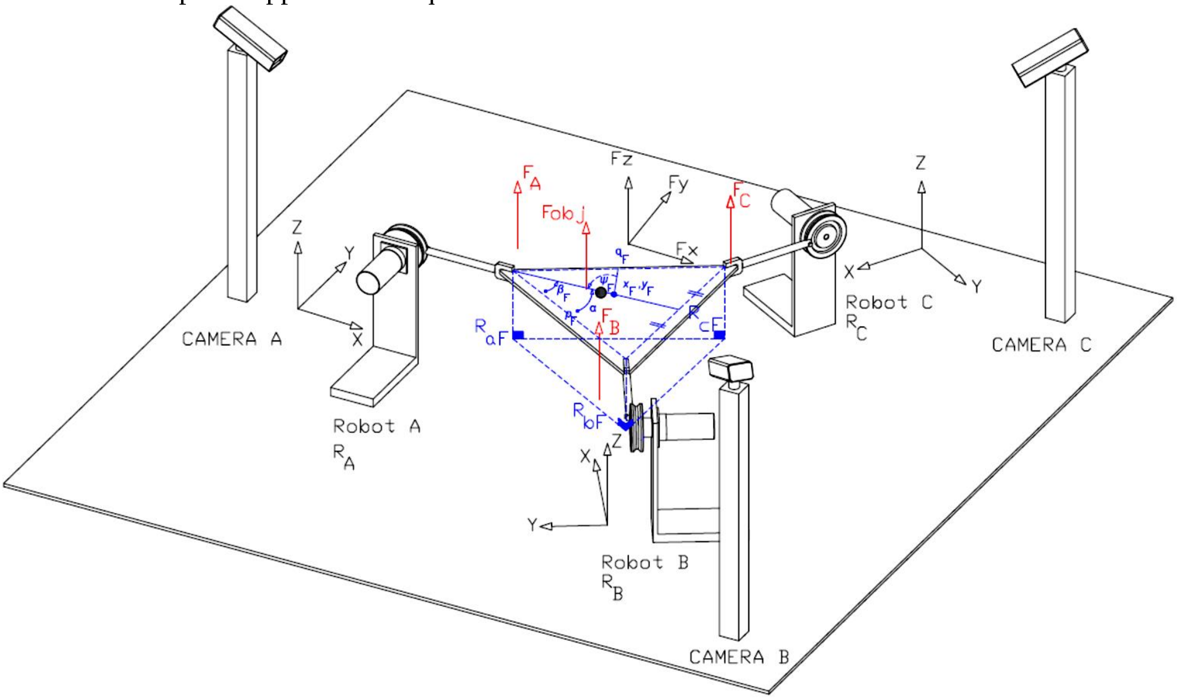


Figure.4 Structure variable of robot's arm, cooperation task for experiment

$$x = f\frac{X}{Z} = \frac{(u-u_0)}{f}\,;\; y = f\frac{Y}{Z} = \frac{(v-v_0)}{f} \tag{20}$$

Where $m = (u, v)$ represents the image plane coordinates of the 2D point of passion in a pixel and $a = (u_0, v_0, f)$ is the camera elemental parameters vector, u_0 and v_0 are coordinates of the main point, while f denoted focal lengths. For velocity of the 3D point regarding the frame of camera as:

$$\dot{P} = -v_c - w_c \times P \tag{21}$$

While in scalar form, considering that $P = (X, Y, Z)$,

$v_c = (v_x, v_y, v_z)$ and $w_c = (w_x, w_y, w_z)$ as:

$$\begin{aligned}\dot{X} &= -v_x - w_y \times Z + w_z Y,\\ \dot{Y} &= -v_y - w_z \times X + w_z Z,\\ \dot{Z} &= -v_z - w_x \times Y + w_y X\end{aligned} \tag{22}$$

While in scalar form,

$$\begin{aligned}\dot{x} &= -\frac{v_x}{Z} + \frac{xv_z}{Z} + xyw_z - (1+x^2)w_y\\ &\quad + yw_z,\\ \dot{y} &= -\frac{v_{yx}}{Z} + \frac{yv_z}{Z} + xyw_y - (1+x^2)w_x\\ &\quad + xw_z\end{aligned} \tag{23}$$

The point p_i^c is expressed in the camera coordinate system Σ_c as followed by the dependent position p_0^c and orientation R_0^c of the object according to the camera~~as~~:

$$p_i^c = R_0^c p_i^0 + p_0^c \tag{24}$$

Whereas:

$$p_i^c = \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}, p_0^c = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix},$$

$$R_0^c = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{25}$$

In the current implementation, the directions of dimensions to move the robotics were retrieved from the in direction of imagery captured with on flat-plate of the camera vision. The advantages of the camera visions in this application is to manage a single adequate mark of projection. That system, subsequently a relevant calibration, whole point of the image directional can be correlated with a particular ray over the focal of camera system.

In Fig. 6, the unit directions $\hat{u}_{ij} \in R^3$ stand for the direction between manipulator i and j indicated in the coordinate frame of manipulator i. Give ${}^{i}T_j \in R^3$ and ${}^{i}R_j \in SO$ perform respectively the rotation and translation of manipulator j with respective the frame reference of manipulator i.

Externally failure generalization we can determine the mention frame of manipulator 1 as our base frame of mention and restore the composition of the robotic cooperation with the improvement on the orientations and positions of other robotic with consideration in the frame. The particular frame, internal corner between the position vectors to alternative other manipulators (ψ_i) can be resolved scalar product; for occasion $(\psi_2) = \cos^{-1}(\hat{u}_{21} \cdot \hat{u}_{23})$. Including this angle data, the translation between the frames can promptly be determined ~~to~~ a scale of the factor by implementing the rule in the triangle shown as Fig. 6.

Promptly framework the dimension of L_{23}, we have the relative situation for particular of the operator as:

$$ {}^{1}T_2 = L_{12}\hat{u}_{12} = \frac{\sin(\cos^{-1}(\hat{u}_{21} \cdot \hat{u}_{23}))}{\sin(\cos^{-1}(\hat{u}_{13} \cdot \hat{u}_{12}))}\hat{u}_{12} $$

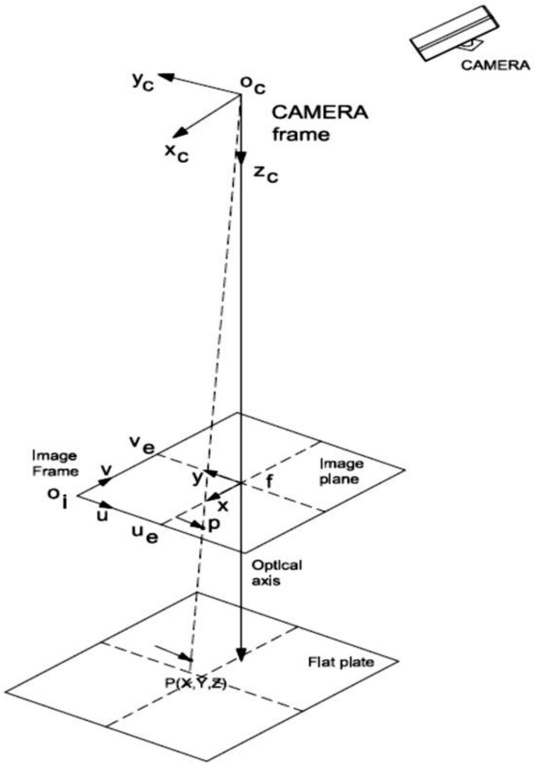


Figure.5 The relationship of camera and the image frame

$$ {}^{1}T_3 = L_{13}\hat{u}_{13} = \frac{\sin(\cos^{-1}(\hat{u}_{32} \cdot \hat{u}_{31}))}{\sin(\cos^{-1}(\hat{u}_{13} \cdot \hat{u}_{12}))}\hat{u}_{13} \quad (26) $$

The vectors of position corresponding to further frames can be still received through proving the correlative of unit vectors. Including the situation of representative established expect the relative direction of the framework to accomplish the location operation.

Accomplishing the present, the vectors ${}^{j}T_i$ and ${}^{i}T_j$ should have the equal magnitude, but the opposite direction when related by the corresponding rotation matrix ${}^{j}R_i$. We note a similar relationship between the vectors ${}^{j}T_i - {}^{j}T_k$ and ${}^{i}T_k$. From these, we obtain the following pairs of equation:

$$ \begin{aligned} -{}^{1}T_2 &= {}^{1}R_2 {}^{2}T_1 \,;\; {}^{1}T_3 - {}^{1}T_2 = {}^{1}R_2 \, {}^{2}T_3 \\ -{}^{1}T_3 &= {}^{1}R_3 {}^{3}T_1 \,;\; {}^{1}T_2 - {}^{1}T_3 = {}^{1}R_3 \, {}^{2}T_2 \quad (27) \end{aligned} $$

2.5 Decentralized control with neural network

Back Propagation artificial neural networks (BPANNs) have two conditions which ~~is~~ are the directional information that flows from input to output and feed-back for compute the error and adjusted weight. The number of layers is limitless and the simplest BPANN is one perceptron that has one function, i.e. a single function of learning problems. The multi-layer BPANNs shown in Fig.

Received: September 28, 2019. Revised: November 22, 2019.

7 are corresponding analytical description with the sets in equations driven to generally mathematical method. The hardware, computer and microcontroller, Arduino Uno and specialized software are Matlab and C++ for programming use~~d~~ which is appropriate to reconstruct, illustrate and optimize standards of BPANNs.

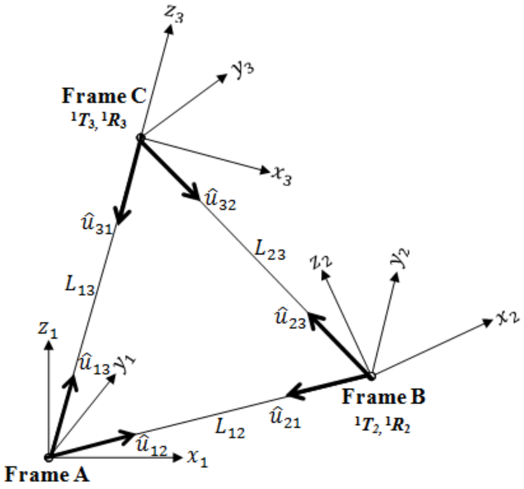


Figure.6 Dimensional of robotics workspace

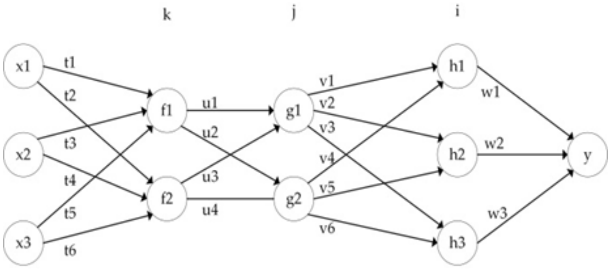


Figure.7 Artificial Intelligent Neural Networks

Table 1. Notation of back-propagation neural network

Definition	Neural Network System
$x1, x2$ *and* $x3$	Set of input vector receive new observation
$f1$ *and* $f2$	Hidden layer 1st, bias vector activation
$g1$ *and* $g2$	Hidden layer 2nd, bias vector activation
$h1, h2$ *and* $h3$	Hidden layer 3rd, bias vector activation
$t1 - t6$	Weight from neuron input
$u1 - u4$	Weight from neuron in the hidden layer 1st
$v1 - v6$	Weight from neuron in the hidden layer 2nd
$w1, w2$ *and* $w3$	Weight from neuron in the hidden layer 3rd
$\hat{y}$	Output vector in prediction
y	Output vector (PWM)

Definition: Historical data are input x_1: position of the object (x,y), x_2: speed of the object (camera frame reference before and after position during time), x_3: robots interaction and u,g and h (weights and biases) are coefficient. The output, y is target of pulses width modulation, PWM value for control servo motor, forward and backward.

In this study NN have two steps as follows:

Feed-forward (receive observation of data x)

- Initialization of weight
- Utilized weigh to some random value
- While stopping condition, do next step
- Get prediction of the output, y and error

Back-Propagation

- Compute the error
- Update weight value

Feed-forward as:

$$\begin{aligned} u1, u2 &= f1(t1x1 + t3x2 + t5x3) \\ u3, u4 &= f2(t2x1 + t4x2 + t6x3) \\ v1, v2, v3 &= g1(f1u1 + f2u3) \\ v4, v5, v6 &= g2(f1u2 + f2u4) \\ w1 &= h1(v1g1 + v4g2) \\ w2 &= h2(v2g1 + v5g2) \\ w3 &= h3(v3g1 + v6g2) \\ y &= w1 + w2 + w3 \end{aligned} \tag{28}$$

While, replace w in the Eq. (28) as:

$$y = [h1(v1g1 + v4g2)] + [h2(v2g1 + v5g2)] + [h3(v3g1 + v6g2)] \tag{29}$$

While, replace v in the Eq. (29) as:

$$y = \left[h1\left((g1)^2(f1u1 + f2u3) + (g2)^2(f1u2 + f2u4)\right)\right] + \left[h2\left(g1)^2(f1u1 + f2u3) + (g2)^2(f1u2 + f2u4)\right)\right] + \left[h3\left((g1)^2(f1u1 + f2u3) + (g2)^2(f1u2 + f2u4)\right)\right] \tag{30}$$

While, replace u in the Eq. (30) as:

$$y = [h1((g1)^2((f1)^2(t1x1 + t3x2 + t5x3) + (f2)^2(t2x1 + t4x2 + t6x3)) + (g2)^2((f1)^2(t1x1 + t3x2 + t5x3) + (f2)^2(t2x1 + t4x2 + t6x3)))] + \left[h2\left(g1)^2\left((f1)^2(t1x1 + t3x2 + t5x3) + (f2)^2(t2x1 + t4x2 + t6x3)\right) + (g2)^2\left((f1)^2(t1x1 + t3x2 + t5x3) + (f2)^2(t2x1 + t4x2 + t6x3)\right)\right)\right] +$$

$$\Big[h3\Big((g1)^2\big((f1)^2(t1x1+t3x2+t5x3)+(f2)^2(t2x1+t4x2+t6x3)\big)+(g2)^2\big((f1)^2(t1x1+t3x2+t5x3)+(f2)^2(t2x1+t4x2+t6x3)\big)\Big)\Big] \quad (31)$$

Back-Propagation as:

$$\begin{aligned}
&E = \tfrac{1}{2}(y-\hat{y})^2\\
&\frac{\partial E}{\partial h_i} = (y-\hat{y})\,\frac{\partial y}{\partial h_i} = (y-\hat{y})\,y(1-y)w_i\\
&\frac{\partial E}{\partial g_i} = (y-\hat{y})\,\frac{\partial y}{\partial g_i}\\
&\frac{\partial E}{\partial g_i} = (y-\hat{y})\,y(1-y)\textstyle\sum_i w_i\frac{\partial h_i}{\partial g_j}\\
&\frac{\partial E}{\partial g_i} = (y-\hat{y})\,y(1-y)\textstyle\sum_i w_i h_i\,(1-h_i)v_{ij}\\
&\frac{\partial E}{\partial u} = (y-\hat{y})\,\frac{\partial y}{\partial u}\\
&\frac{\partial E}{\partial u} = (y-\hat{y})\,y(1-y)\,\textstyle\sum_i w_i\frac{\partial h_i}{\partial u}\\
&\frac{\partial E}{\partial u} = (y-\hat{y})\,y(1-y)\,\textstyle\sum_i w_i h_i(1-h_i)\sum_j v_{ij}\frac{\partial g_i}{\partial u}\\
&\frac{\partial E}{\partial u} = (y-\hat{y})\,y(1-y)\,\textstyle\sum_i w_i h_i(1-h_i)v_{ij}g_j(1-g_j)f_k
\end{aligned} \quad (32)$$

a) Compute the error as:
For each,

$$\frac{\partial E}{\partial g_i} = \textstyle\sum_i \sigma(h_i)v_{ij}\,\frac{\partial E}{\partial h_i} \quad (33)$$

b) Update weight value as:
For each,

$$\frac{\partial E}{\partial u_{jk}} = \frac{\partial E}{\partial g_j}\,\sigma\,(g_j)f_k$$

Update weight

$$u_{jk} \leftarrow u_{jk} - \eta\frac{\partial E}{\partial u_{jk}} \quad (34)$$

Where,

$$\begin{aligned}
y &= \sigma(w_0 + \textstyle\sum_i w_i h_i)\\
h_i &= \sigma(v_{i0} + \textstyle\sum_j v_{ij} g_j)\\
g_j &= \sigma(u_{i0} + \textstyle\sum_k u_{jk} f_k)\\
f_k &= \sigma(t_{k0} + \textstyle\sum_m t_{km} x_m)
\end{aligned} \quad (35)$$

3. Results

The experiments in this study demonstrate that there is a possibility in terms of problem-solving on time systems applying neural network in learning and controlling to develop control of robotics interaction. The system of decentralized control, including neural network, is achieved. The experimentation is operated with a close investigation. The experimentation involved information learning to orient and position of the robotics interaction in cooperative task and to track trajectories relative into an object moving for balancing task on flat-plate at arbitrary orientations, position, speed and interaction signal of the object using feedback from web camera and forward only a priori information of the robotic kinematics and camera characteristics. In the initial position, the end-effector of the manipulators are set in the intermediate of workspace. In spite of learning and teaching, the object movement is uncertainty direction on flat plate in the middle of the end-effector of manipulators. Some experiment of simulation is shown in Figs. 8 and 9 is shown initial boundary of work space area.

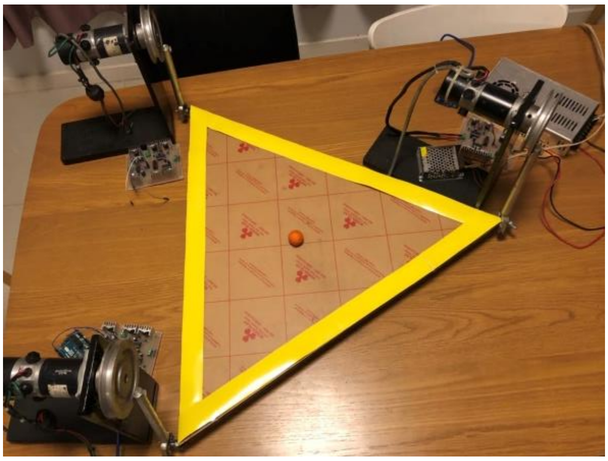


Figure.8 Experimental system consisting of robot's arm

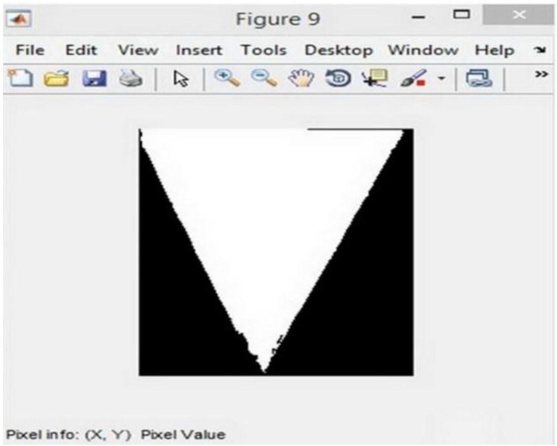


Figure 9
File Edit View Insert Tools Desktop Window Help
Pixel info: (X, Y) Pixel Value

Figure.9 The area of an object can balance on flat plate

Neural Network (NN) method is adaptive parameters for tracking strategy in control DC servo motors, uncertainty of value was advanced,

Received: September 28, 2019. Revised: November 22, 2019.

including the strategy control in position and speed of the servo motors are forced to follow used neural network method and forecasted the PWM variable of the position and motion of the object, speed and signals for robotics interaction. The principles of scheduled controller compose in relevant for complete types of DC servo, operation handle several job or tasks. The results of the simulation are contributed to corroborate the efficiency of the neural network method; hence accomplishing it can be applicable for high achievement DC servo motor tracking applications, in decentralized control for manipulators cooperative task.

The robotics in the experiment has demonstrated the interactive generation and dynamic for multiple robots to learned handling the object using the BPANNs. As the results have shown and discussed, the experimented approach provides the evidence that the robots can interact and seem to have eye and flexibility in such behavior generation by utilizing the fast and the slow dynamics in the BPANNs.

In the robotic interaction system using circle signal and detect image of signal shape from web camera and matching in data set, signal shape in Fig. 10. The signal shape characteristic can be 15 levels, step in to 5 degree of freedom of flat plate angle, upper bound is 35 degree and lower bound is -35 degree along x-axis. It means positive and negative angle is forward and backward respectively. The signal shapes are robotics acknowledge for cooperative interaction task.

The simulation results are shown in Figs. 11–15. Fig. 11 shows the object tracking position in system workspace and Fig. 12 shows image feature trajectories position of an object on the image plane in adaptive control. Fig. 13 represents the trajectory of end-effector of the robotic, again for the adaptive cases in robotics workspace.

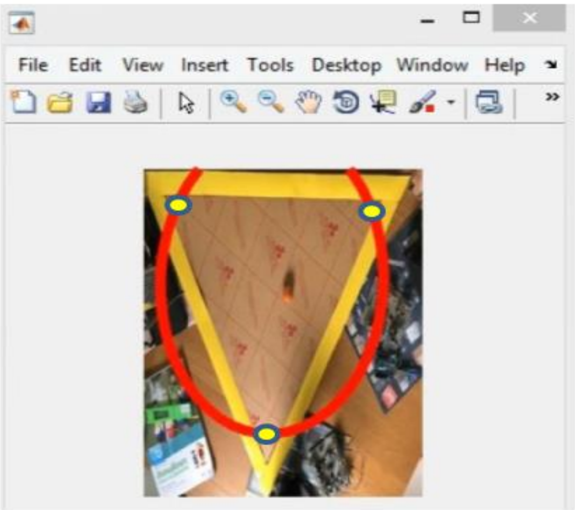


Figure.10 The signals for robotics interaction of system

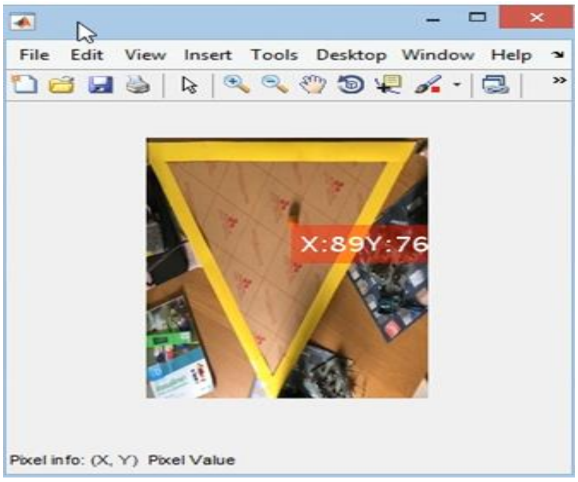


Figure.11 Experimental of object tracking position

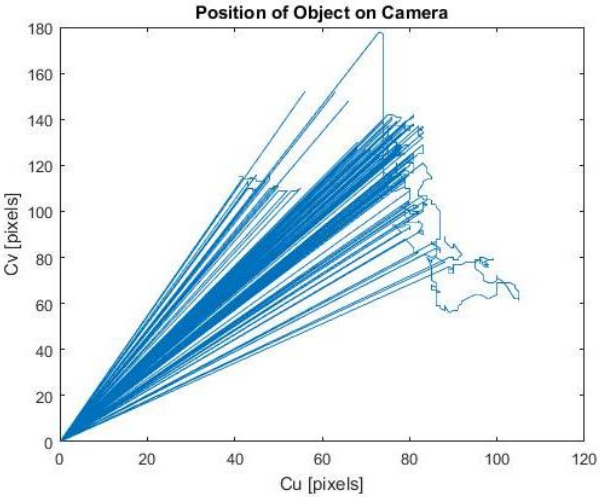


Figure.12 Trajectories position in image plane

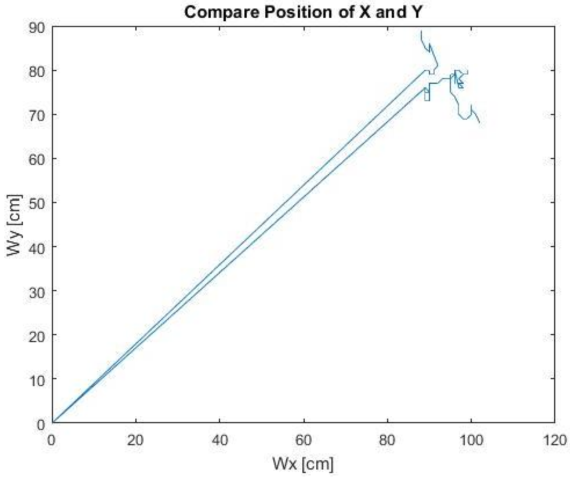
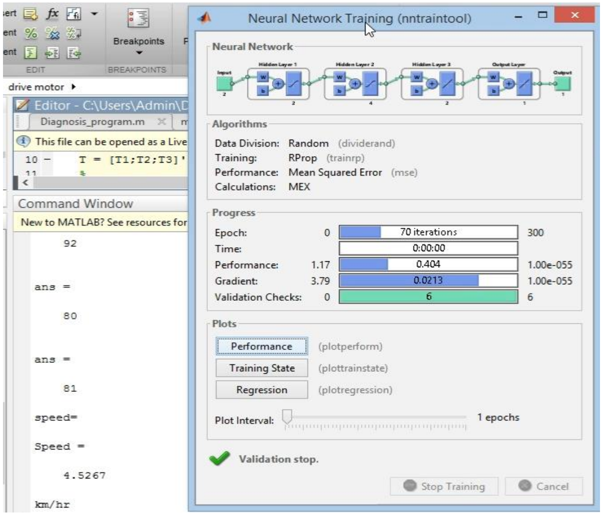


Figure.13 Trajectories in robotics workspace

Received: September 28, 2019. Revised: November 22, 2019.



This file can be opened as a Live

```
10 -      T = [T1;T2;T3]'
```

New to MATLAB? See resources for

```
    92


ans =

    80


ans =

    81

speed=

Speed =

    4.5267

km/hr
```

Algorithms

Data Division: Random (dividerand)
Training: RProp (trainrp)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

Epoch:	0	70 iterations	300
Time:		0:00:00	
Performance:	1.17	0.404	1.00e-055
Gradient:	3.79	0.0213	1.00e-055
Validation Checks:	0	6	6

Plots

Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)

Plot Interval: 1 epochs

✓ Validation stop.

Figure.14 Tracking object speed and training in NN

Fig. 14 presents the progression of the control and training the speed of an object movement, and Fig. 15 shows the object speed tracking trajectories on the image plane. It can be obviously seen from the above figures that the adaptive control in neural network achieves a decentralized cooperative control of the robotics systems using NN for propose in handling task an object on flat plate in best performance.

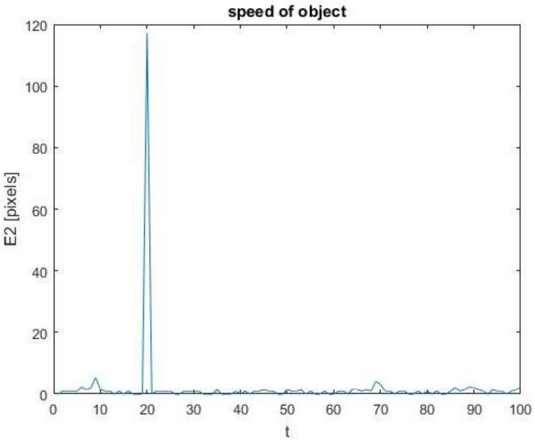


Figure.15 The object speed tracking trajectories

4. Discussion

In this study, a practical adaptive parameter with neural networks in control strategy for the multiple-robot cooperative of manipulators is proposed. Despite, there have some aberration and ambiguous when the object movement was fast. This study on the application of a robotic manipulator confirms that the neural network method can be used control in robot-robot interaction and learning in cooperative with an uncertainty environment. The whole picture of this study clearly shows that the robotics interaction can be learned through cooperation task, and this decentralized control approach is different from many researches in intelligent engineering schemes that the objective is to determine according to manipulate the cooperative robotics rather than how to perform a specific movement for a robotics cooperation task for balancing an object on flat plate.

5. Conclusion

This article presented the decentralized control using neural network can be applied to robot handling the object balancing task of a robotic cooperation manipulator .The image processing for object tracking and neural network method are written on MATLAB® and controller using with Arduino UNO are written on C++, PWM for control DC servo motor. The simulation results are that robotics can be learned to handling an object balancing on a flat plate. As well clearly decentralized control is not restricted to be done only with sensor method, as the neural network method can be implemented with the dataset to acknowledge the movement control of their servo motor for robot-robot interaction with decentralized control effectively. Other research experiments usually use force or torque sensor for acknowledge to control the robotics. This study operates without the sensor which differs from other experiments, a robotics seem have eye with a camera used serial control in acknowledge in environments. The robotics may perform in cooperative with particular robot-robot interaction used decentralized control of robotics to balance the object on flat pate with dataset training in NN method. In addition, this research can be applied to industrial sectors that use robots to work together in the same working style as the article.

References

[1] J. De Jesús Rubio, “Adaptive least square control in discrete time of robotic arms”, *International Journal of Soft Computing*, Vol.19, No.12, pp.3665-3676, 2015.

[2] H. Farivarnejad, and SAA. Moosavian, “Multiple impedance control for object manipulation by a dual arm underwater vehicle–manipulator system”, *International Journal of Ocean Engineering*, Vol.89, pp.82-98, 2014.

[3] SS. Ge, Y. Li, and C. Wang, “Impedance adaptation for optimal robot–environment

interaction", *International Journal of Control,* Vol.87, No.2, pp.249-263, 2014.

[4] V. Vladareanu, P. Schiopu, and L. Vladareanu, "Theory and application of extension hybrid force-position control in robotics", *International Journal of UPB Sci. Bull., Series A,* Vol.76, No.3, pp.43-54, 2014.

[5] P. Gierlak, "Hybrid position/force control in robotised machining", *Solid State Phenomena,* Vol. 210, pp. 192-199, 2014.

[6] S. Li, J. He, Y. Li, and MU. Rafique, "Distributed recurrent neural networks for cooperative control of manipulators: A game-theoretic perspective", *IEEE Transactions on Neural Networks and Learning Systems,* Vol.28, No.2, pp. 415-426, 2016.

[7] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "Qt-Opt:Scalable deep reinforcement learning for vision-based robotic manipulation", In: *Proc. of International Conf. on Robot Learning,* pp. 1-23, 2018.

[8] R. Ul Islam, J. Iqbal, and Q. Khan, "Design and comparison of two control strategies for multi-DOF articulated robotic arm manipulator", *International Journal of Control Engineering and Applied Informatics,* Vol.16, No.2, pp. 28-39, 2014.

[9] J. De Jesús Rubio, "Sliding mode control of robotic arms with deadzone", *International Journal of IET Control Theory and Applications,* Vol.11, No.8, pp. 1214-1221, 2017.

[10] G. Antonelli, F. Arrichiello, F. Caccavale, and A. Marino, "Decentralized time-varying formation control for multi-robot systems", *The International Journal of Robotics Research,* Vol.33, No.7, pp. 1029-1043, 2014

[11] S. Omidshafiei, A. Akbar. A. Mohammadi, C. Amato, SY. Liu, JP. How, and J. Vian, "Decentralized control of multi-robot partially observable Markov decision processes using belief space macro-action", *The International Journal of Robotics Research,* Vol.36, No.2, pp. 231-258, 2017.

[12] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints", *IEEE Transactions on Cybernetics,* Vol.46, No.3, pp. 620-629, 2015.

[13] W. He, and Y. Dong, "Adaptive fuzzy neural network control for a constrained robot using impedance learning", *IEEE Transactions on Neural Networks and Learning Systems,* Vol.29, No.4, pp. 1174-1186, 2017.

[14] W. He, AO. David, Z. Yin, and C. Sun, "Neural network control of a robotic manipulator with input dead zone and output constraint" *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* Vol.46, No.6, pp. 759-770, 2015.

[15] N. Chavan-Dafle, and A. Rodriguez, "Prehensile pushing: In-hand manipulation with push-primitives", In: *Proc. of International Conf. on Intelligent Robots and Systems*, pp.6215-6222, 2015.

[16] F. Šuligoj, B. Šekoranja, M. Švaco, and B. Jerbić, "Object tracking with a multiagent robot system and a stereo vision camera", In: *Proc. of International Symp. on Intelligent Manufacturing and Automation*, pp. 968-973, 2013.

[17] M. Svinin, Y. Bai, and M. Yamamoto, "Dynamic model and motion planning for a pendulum-actuated spherical rolling robot", In: *Proc. of International Conf. on Robotics and Automation*, 2015.

[18] A. Morinaga, M. Svinin, and M. Yamamoto, "A motion planning strategy for a spherical rolling robot driven by two internal rotors", *IEEE Transactions on Robotics,* Vol.30, No.4, pp.993-1002, 2014.

[19] V. H. Kirill, Y. Fujimoto, and A. Kawamura, "A combination of particle swarm optimization and model predictive control on graphics hardware for real-time trajectory planning of the under-actuated nonlinear Acrobot", In: *Proc. of IEEE 13th International Workshop on Advanced Motion Control*, pp.464-469, 2014.

[20] S. Tayebi-Haghighi, F. Piltan, and JM. Kim, "Robust composite high-order super-twisting sliding mode control of robot manipulators", *International Journal of Robotics,* Vol.7.1, No.13, pp.1-18, 2018.

[21] MS. Kim, WJ. Chung, and SW. Jeong, "Application of Gain Scheduling Technique to a 6-Axis Articulated Robot using LabVIEW®", In: *Proc. of International Conf. on Computer Science, Computer Engineering, and Applied Computing*, pp.1-6, 2014.

[22] N. Jaisumroum, P. Chotiprayanakul, and S. Limnararat, "Decentralized control with neural network of cooperative robot manipulator for object balancing task on flat plate", In: *Proc. of International Conf. on Control, Automation and Systems*, pp.819-824, 2017.

[23] N. Jaisumroum, P. Chotiprayanakul, and S. Limnararat, “A conceptual framework of decentralized learning neural network control approach for multi-robot cooperation in an object balancing task”, In: *Proc. of International Conf. on Industrial Engineering and Engineering Management*, pp. 434-437, 2016.

[24] P. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised*, Vol. 118. Springer International Publishing, Verlag Berlin Heidelberg, New York, N.Y.2017.