Journal of Engineering Technology

Vol. 6, Special Issue (Emerging Trends in Engineering Technology) Mar. 2018, PP. 82-96

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

Wireless stereo vision for an omnidirectional robot

Han-Wei Zhang[1], Zhi-Ren Tsai [2#], Chin-Teng Lin[3,4,5]

[1] Institute of Electrical Control Engineering, Department of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan.
[2]Department of Computer Science & Information Engineering, Asia University, Taiwan.
[2]Department of Medical Research, China Medical University Hospital, China Medical University, Taichung, Taiwan.
[3]Brain Research Center, National Chiao Tung University, Hsinchu, Taiwan.
[4]Institute of Electrical Control Engineering, Department of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan.
[5] Centre for Artificial Intelligence School of Software, Faculty of Engineering & IT, University of Technology Sydney Broadway 2007, New South Wales, Australia.

Abstract

Introduction: Image processing has been implemented for numerous applications but is yet to be robustly and practically applied for the control of omnidirectional robots. This application uniquely requires a visual servo function and a model-free design; stereo vision, which involves extracting three-dimensional information from images from multiple viewpoints, can be used to meet these requirements. **Methods:** Here, we propose a prototype of a two-camera system to enable wireless stereo vision. This system uses a controller to locate light-emitting diode markers placed on an omnidirectional robot as a robust target to guide the robot through various obstacles. This wireless system includes a control system that leverages artificial intelligence to recognize itself via remote stereo vision and adapt to its environment accordingly. However, this scenario introduces many control problems including time-varying delay, nonlinearity, and uncertainty. Hence, given the sampling period of this control system, a model-free method is proposed. A visual-delay servo is used to guide the robot and an adaptive prediction-control law is implemented via an island-type neural network. Moreover, the sampling time is accounted for in the continuous-time system to design the digital controller. **Results:** The proposed system is tested for guiding a robot into a garage. The experimental results show that the proposed autonomous navigation, stereo vision servo, and CIE-Lab color transformation enable a rapid and accurate pose acquisition for this robot. Moreover, the proposed technique provides better performance compared to traditional methods. **Conclusion:** These results demonstrate the potential of this system, which does not rely on physical models, to be used for a wide range of applications in the future.

Keywords: Wireless stereo vision; Visual-delay servo; LED-based marker; CIE-Lab; Island-type neural network.

[#] Author to whom all correspondence should be addressed (E-mail: ren@asia.edu.tw ; zhirentsai@gmail.com).

Journal of Engineering Technology

Volume 6, Special Issue (Emerging Trends in Engineering Technology) March. 2018, PP. 71-96

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

1. Introduction

Image-processing techniques [1-5] have been applied in various fields including traffic safety, robot vision, and eHealth. However, the traditional method based on the ideal kinematics model [6, 7] to control the shaking of robots on rocky roads does not consider the modeling errors caused by, for example, unknown payloads, consumption of battery power, obstacle avoidance, and friction on the road. Moreover, the digital controller for this model faces a continuous-time mechatronics system but robustness has not been considered in its design. One method [8] has been proposed to improve its robustness, but it is too complex to be implemented in an omnidirectional robot without the use of a physical model. Hence, a visual servo function and a model-free design are needed. One approach to meet these requirements is the use of wireless stereo vision, which can be practically implemented using light-emitting diodes (LEDs) as markers.

Various techniques for image segmentation [9] have been presented in the literature. However, most image-recognition systems, such as machine vision [10], focus on processing images without light disturbances and are not effective when the input images contain dynamic light disturbances. The processing becomes more difficult when the color marker is disturbed by light. In this study, the CIE-Lab descriptor [5] of the colored markers, which are similar to the two brake lights of a car [11], is used to overcome this disturbance. This simplifies the marker detection and allows the system to operate in the dark. The markers are considered the region of interest (ROI) [12] and their positions relative to the image area are identified via a segmentation process in order to accurately calculate the pose of the robot. First, the system is trained to recognize the marker colors and the marker is classified based on the pre-trained template color. Next, Lab transformation is applied to each frame image and the two LEDs are located in each image. The marker is customized to reduce the ROI. Furthermore, a neural network is used to predict the stereo-vision-based information for the future pose of the robot and this information is input to the adaptive predictive controller. The initial parameters of the neural controller are optimized using an island-type neural network (INN). Its weights and biases are predicted by online back-propagation (BP) [13] to predict the pose of the robot and applied to guide the robot, such as moving into a garage. This digital controller is a Nonlinear Auto Regressive moving average eXogenous (NARX) neural network [13] and includes tape delays to learn the continuous-time behavior of the robot despite the limited digital precision. The island genetic algorithm or parallel genetic algorithm (PGA) used in the INN has many islands that share genes or populations of other islands through a common pool of chromosomes, θ_N, by the migration of each island. Chromosome θ_N represents the initial weights and biases of the neural network and is converged by BP. These solutions get off the local optimum by either the big-bang method or re-initialization.

Herein, we describe the design of the wireless stereo vision system and its implementation in an omnidirectional robot. We further test the system for a specific application of navigating the robot into a garage. Finally, the proposed method is compared with other methods to demonstrate its unique advantages for this application.

2. Methods

2.1 Problem formulation and controller design

There is an eight-bit, zero-order-hold (ZOH) microcontroller in the large-scale system, P, composed through the ZOH method of units of P_i: $\dot{\bar{x}}_i(t) = f_i(\bar{x}_i(t), u_i(t)) + \sum_{j=1,\ j\neq i}^{J} f_{ij}(\bar{x}_j(t)) + \Delta_i(t) + \overline{w}_i(t)$, where $u_i(t)$ is the continuous-time version of the i-th sub-controller, $u_i(k \cdot T_s)$, in which T_s is the sampling

Journal of Engineering Technology

Volume 6, Special Issue (Emerging Trends in Engineering Technology) March. 2018, PP. 71-96

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

time. This ZOH microcontroller controls the three motors of the robot's wheels. The weights (W_1, W_2) and biases (B_1, B_2) of the neural network controllers shown in Figs. 1 and 2 need to be trained to control P. The Lyapunov method of considering the modeling error is applied to provide stability to guide the robot. The decentralized controller, C, of P is composed of J digital neural sub-controllers, $u(t) = u(k \cdot T_s) = [u_1, u_2, ..., u_i, ..., u_J] = u(k) = u_k$. The discrete-time versions of the continuous-time states, $\bar{x}_1(t)$, $\bar{x}_2(t)$, and $\bar{x}_3(t)$, of the sub-systems are $x(k \cdot T_s)$, $y(k \cdot T_s)$, and $\theta(k \cdot T_s)$, respectively; furthermore, $\bar{x}(t) = [\bar{x}_1, \bar{x}_2, \bar{x}_3]^T$. The uncertainties of P are $\{\Delta_1(t), \Delta_2(t), ..., \Delta_J(t)\} \in \Delta(t)$ and its unconstrained external disturbances are $\{\overline{w}_1(t), \overline{w}_2(t), ..., \overline{w}_J(t)\} \in \overline{w}(t)$. In addition, f_{ij} represents the interconnection between the i-th and j-th sub-systems. The collected input/output data of P_i will be trained to its neural sub-models, $\{\hat{P}_1, \hat{P}_2, ..., \hat{P}_J\} \in \hat{P}$. The proposed large-scale control strategy will switch one of the three sub-controllers into P based on three conditions that deal with the modeling problem of the unknown system. The proposed two-stage (offline and online stages) strategy for training the neural sub-model and its sub-controller provides two advantages: (1) the need to build a complex physical model is avoided and (2) the prediction ability of the neural network to compensate for the delay in controlling the motions of the robot is enhanced. The motions of this robot are highly simplified into three actions: (1) rotation, (2) side shifting, and (3) forward movement along a straight line. Hence, the interconnected effect, f_{ij}, of P is eliminated using the switching conditions shown in Fig. 2 in order to guarantee good control performance. If the proposed image processing is robust, then the guidance of the robot will be stable. Fig. 2 shows a prototype of the omnidirectional robot, which includes three motors driving its three wheels and a dynamically changing payload; it can be considered a visual-delay, unknown, and highly nonlinear mechatronics system. This robot faces variable friction on the road. Wireless stereo vision is used to sample the pose of the robot. The time-varying delay is caused by the computing time of the image processing and the wireless communication. Thus, this physical model cannot be thoroughly described mathematically. Hence, the NARX controller in Fig. 2, which is composed of three sub-controllers, is proposed as a large-scale controller to achieve a model-free constraint.

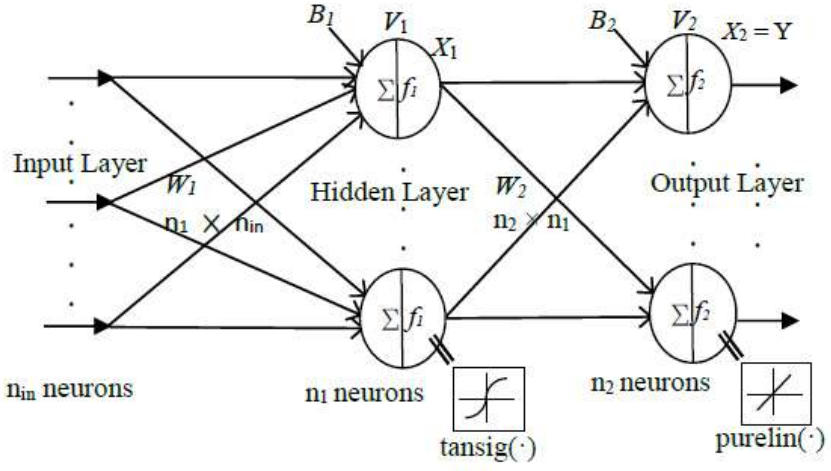


Figure 1. Structure of the adaptive predictive neural controller.

Journal of Engineering Technology

Volume 6, Special Issue (Emerging Trends in Engineering Technology) March. 2018, PP. 71-96

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

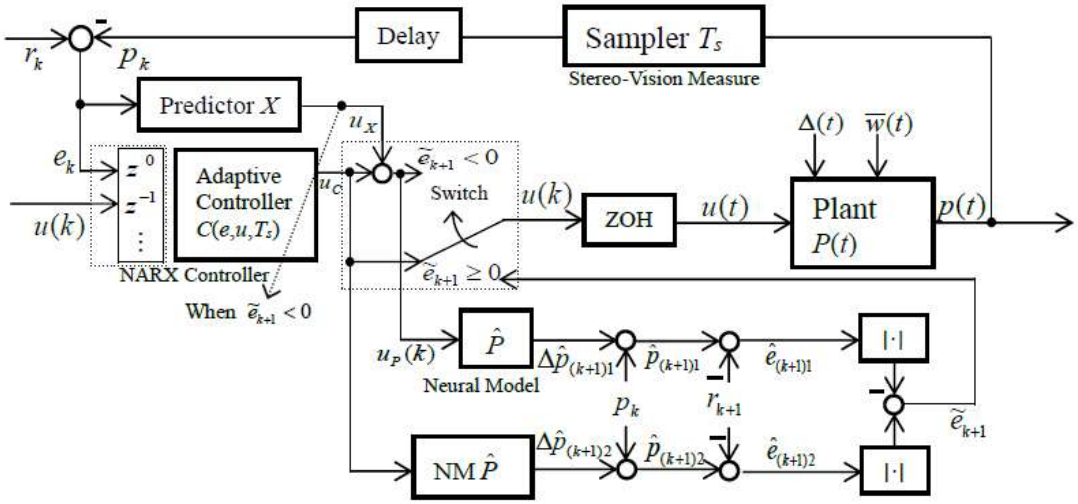


Figure 2. Digital control structure for the delayed continuous-time nonlinear plant with uncertainty.

2.2 Stereo vision measurements

Based on the approach described in [5], first, the pixels within the color range of the proposed marker are identified as shown in Fig. 3. The noise is filtered by searching for the largest area of pixels in order to reduce the ROI in the image. Then, fuzzy C-means [14] clustering segmentation is used to detect the relative position of the center of the circle corresponding to each LED on a grayscale image compensating the image distortion based on the intrinsic parameters of the cameras. Next, the stereocenter of each LED is calculated in the world coordinate system by stereo vision [15] in order to compute the robot's pose for the servo. Finally, the next ROI is predicted by the Kalman predictor [16].

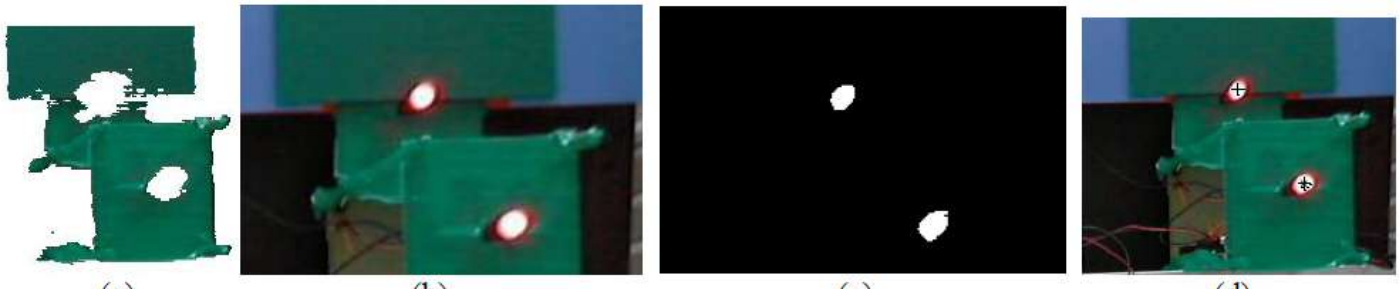


Figure 3. Experimental steps of image recognition of the LEDs: (a) color recognition, (b) ROI identification, (c) segmentation of circles, and (d) detection of the centers of the two circles (denoted by + and *).

Algorithm 1: The center of each circle is detected as follows:

Step 1: Process the two LED bulbs in the RGB images from the left and right wireless cameras to two similar circles of grayscale and binary images; this is referred to as the marker region.

Step 2: Calculate all gradient vectors, V_1,V_2, V_3,…, in this marker region.

Step 3: Find the vector pairs, V_1 and V_2, that satisfy the following conditions:

Condition 1: The angle between V_1 and V_2 is 180° ($\alpha \approx 180^\circ$).

Condition 2: If the positions of V_1 and V_2 are defined as P_1 and P_2, respectively, the angle between

Journal of Engineering Technology

Volume 6, Special Issue (Emerging Trends in Engineering Technology) March. 2018, PP. 71-96

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

$\vec{P_1P_2}$ and V_1 is zero ($\beta \approx 0^\circ$).

Step 4: The midpoint between P_1 and P_2 is a candidate for the circle center and $|\vec{P_1P_2}|/2$ is the radius, r, of this circle. Compute all vector pairs from $V_1, V_2, V_3, \ldots$ in this marker region.

Step 5: Find all candidates from all vector pairs. The mean of all candidates is then taken as the circle center, C(x,y).

Algorithm 2: The pose angle, θ, and its error, $\Delta\theta_k$, of the robot are calculated as follows:

Step 1: The desired pose vector of the robot is $v_r = (\Delta x_r, \Delta y_r)$. Measure the positions of the upper and lower LED bulbs, $(x, y) = (x_1, y_1)$ and $(x, y) = (x_2, y_2)$, using stereo vision [15].

Step 2: Calculate the current pose vector of the robot as $v_p = (x_2 - x_1, y_2 - y_1)$ and calculate $d_p = v_p \cdot v_r$.

Step 3: Calculate the lengths of the two vectors, v_p and v_r, as $l_1 = \|v_p\|$ and $l_2 = \|v_r\|$, respectively. The larger one that computes the results of these two vectors using the cross operator is the angle $\theta_M = 180 - \cos^{-1}(\frac{d_p}{l_1 l_2})\frac{180}{\pi}$. Then, calculate $\Delta x_r = x_2 - x_1 - \Delta x_r$.

Step 4: If $\theta_M > 90$ and $\Delta x_r \geq 0$, then $\theta = 180 - \theta_M$; if $\theta_M > 90$ and $\Delta x_r < 0$, then $\theta = \theta_M - 180$.

Algorithm 3: A large-scale predictive control based on NARX compensates for the visual delay. This neural controller includes a recursive structure but is trained by the feedforward-structure (BP) method to allow its weights and biases to converge. The desired signal or path of the robot is $r(k) = r_k = [x_d(k), y_d(k), \theta_d(k)]^T$ and the real path is $p(k) = p_k = [x(k), y(k), \theta(k)]^T$, where p_k is the actual output and r_k is the desired output of the system. Hence, the training data, $\{e_k, u_k\}$, of $C(\cdot)$ in Fig. 2 are used for $u_k = C(u_{k-1}, u_{k-2}, ..., u_{k-m}, e_k, e_{k-1}, ..., e_{k-c}, W_C)$, where m, c are the amounts of the tap delays (u, e, respectively) and the tracking error is $e_k = r_k - p_k$. The W_C matrix is composed of the weights and biases of C. These offline training data are rearranged to avoid the local optimal ranges. A training-data-shuffle method is applied to obtain the global optimal neural controller by using the other data (testing data) to verify the local optimal ranges. The absolute values , Ξ_1 and Ξ_2, of the training and testing errors, respectively, are obtained by BP to obtain the fitness function, $100/(\Xi+1)$, and cost function, $\Xi = \rho \cdot \Xi_1 + (1-\rho) \cdot \Xi_2$, where $\rho \in [0,1]$. Finally, the three neural predictive sub-controllers are obtained as follows:

$$u_{1k} = C_1(S_{u1}u_1(k-1), ..., S_{u1}u_1(k-c_{u1}), S_{e1}e_1(k), ..., S_{e1}e_1(k-c_{e1}))(1/S_{u1}),$$
$$u_{2k} = C_2(S_{u2}u_2(k-1), ..., S_{u2}u_2(k-c_{u2}), S_{e2}e_2(k), ..., S_{e2}e_2(k-c_{e2}))(1/S_{u2}), \text{ and}$$
$$u_{3k} = C_3(S_{u3}u_3(k-1), ..., S_{u3}u_3(k-c_{u3}), S_{e3}e_3(k), ..., S_{e3}e_3(k-c_{e3}))(1/S_{u3}),$$

where the scale factors, $\{S_{u1}, S_{e1}\}, \{S_{u2}, S_{e2}\}, \{S_{u3}, S_{e3}\}$, limit the input ranges of the neural controller to [-10, 10] and $c_{u1}, c_{u2}, c_{u3}, c_{e1}+1, c_{e2}+1, c_{e3}+1$ are the amounts of the tap delays of u, e for the three sub-controllers, respectively. The two stages of this process are as follows:

First stage: First, $\{\Delta\hat{p}_{k+1}, u_k\}$ are collected for the offline training of the neural sub-model: $\Delta\hat{p}_{1(k+1)} = \hat{P}_1(\hat{S}_{u1}u_{1k})S_1$, $\Delta\hat{p}_{2(k+1)} = \hat{P}_2(\hat{S}_{u2}u_{2k})S_2$, and $\Delta\hat{p}_{3(k+1)} = \hat{P}_3(\hat{S}_{u3}u_{3k})S_3$, where

Journal of Engineering Technology

Volume 6, Special Issue (Emerging Trends in Engineering Technology) March. 2018, PP. 71-96

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

$\Delta\hat{p}_{k+1} = [\Delta\hat{p}_{1(k+1)}, \Delta\hat{p}_{2(k+1)}, \Delta\hat{p}_{3(k+1)}]^T$. Then, based on the training data, $\{e_k, u_k\}$, the offline controller, $u_k = [u_{1k}, u_{2k}, u_{3k}]^T$, can be obtained using the INN, where $e_k = [e_{1k}, e_{2k}, e_{3k}]^T$ and k is the index of the time sequence. The fitness function for each adaptive sub-controller is designed as follows:

$$\text{First sub-controller: } 100/(\frac{1}{N_1}\sum_{k=1}^{N_1}| C_1(e_{1k}, e_{1(k-1)},..., W_{C1}) - u_{1k} | +1);$$

$$\text{Second sub-controller: } 100/(\frac{1}{N_2}\sum_{k=1}^{N_2}| C_2(e_{2k}, e_{2(k-1)},..., W_{C2}) - u_{2k} | +1); \text{ and}$$

$$\text{Third sub-controller: } 100/(\frac{1}{N_3}\sum_{k=1}^{N_3}| C_3(e_{3k}, e_{3(k-1)},..., W_{C3}) - u_{3k} | +1).$$

The training input data are multiplied by $\{S_{u1}, S_{e1}\}, \{S_{u2}, S_{e2}\}, \{S_{u3}, S_{e3}\}$ to generate convergence.

Second stage: According to Theorems 1–2 (defined as follows), the weights and biases of the sub-models and sub-controllers are updated online based on the initial weights and biases determined offline in the first stage.

Theorem 1: If the number of recursive network neurons is sufficient and T_s is appropriate to make the modeling error $\|\bar{p}_k - p_k\| \le \bar{\varepsilon}$ acceptable, then the condition $0 < \eta_P(k) < \dfrac{2}{\|\partial\hat{p}_k / \partial(W_P)_k\|^2} \le \bar{\eta}_P$, where $\bar{p}_k$ is the perfect output of the neural model, is satisfied. Thus, the trajectory of $\hat{p}_k \to p_k$ is uniformly ultimately bounded (UUB); in order words, $\hat{p}_k - p_k$ has converged and the global optimal solution of the neural model is obtained. A detailed proof is shown in the Appendix. There is a sensitivity $S_k = \dfrac{\partial\hat{p}_k}{\partial u_k} \approx \dfrac{\Delta\hat{p}_{k+1}}{\Delta u_k} = \dfrac{\Delta\hat{p}_{k+1}}{u_X}$, which is related to Theorem 2 as follows.

Theorem 2: If Theorem 1 is satisfied and the above S_k is calculated to satisfy the condition $0 < \eta_C(k) < \dfrac{2}{\|S_k\|^2 \|\partial u_k / \partial(W_C)_k\|^2} \le \bar{\eta}_C$, then according to the proposed switching method, each sub-system is UUB and it can be inferred that the large-scale system is also UUB. Hence, C causes the tracking errors $e(k) = r_k - p_k$ to also be UUB, where S_k is related to $\{\Delta\hat{p}_{(k+1)1}, \Delta\hat{p}_{(k+1)2}\} \in \Delta\hat{p}_{k+1}$, $\{\hat{p}_{(k+1)1}, \hat{p}_{(k+1)2}\} \in \hat{p}_{k+1}$, $\{\hat{e}_{(k+1)1}, \hat{e}_{(k+1)2}\} \in \hat{e}_{k+1}$, and u_X. In other words, C requires $\hat{P}$ to run through S_k to update its control parameters and S_k to be related to $\tilde{e}_{k+1}$ in Fig. 2. In this way, u_X can be estimated to obtain a short-time prediction. The identified solution of the INN can reach the global optimum systematically in two ways: (1) the training data are rich enough or (2) the two-stage training strategy is used.

The proposed switching strategy is introduced to reduce the design complexity of the control system as follows:

Condition 1 for P_3 and C_3: The angle of the robot pose is zero ($\theta_k \to \theta_d = 0$).

Condition 2 for P_1 and C_1: The position of the robot shifts to the target ($x_k \to x_d$).

Condition 3 for P_2 and C_2: The distance of the robot reaches the desired value ($y_k \to y_d$).

2.3 Application

The priorities of the conditions of the sub-systems are as follows: Condition 1 > Condition 2 > Condition 3.

Journal of Engineering Technology

Volume 6, Special Issue (Emerging Trends in Engineering Technology) March. 2018, PP. 71-96

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

The strategy is as follows:

Step 1: As Condition 1 is the top priority, a tolerable angle error, c_θ, a tolerable shift error, c_x, and a tolerable distance error, c_y, are set. If $|\theta-\theta_d|=|\Delta\theta_k|\le c_\theta$ (in degrees), then go to Step 2; otherwise, switch u_{1k} into the robot in Fig. 4a by sub-controller 1, C_1, until $|\Delta\theta_k|\le c_\theta$.

Step 2: If $|x-x_d|=|\Delta x_k|>c_x$(in mm) and $|\Delta\theta_k|\le c_\theta$, then switch u_{2k} into the robot by sub-controller 2, C_2, until $|\Delta x_k|\le c_x$ and $|\Delta\theta_k|\le c_\theta$. If $|\Delta\theta_k|>c_\theta$, then return to Step 1. If $|\Delta x_k|\le c_x$ and $|\Delta\theta_k|\le c_\theta$, then continue to Step 3.

Step 3: If $|y-y_d|=|\Delta y_k|>c_y$(in mm), $|\Delta x_k|\le c_x$, and $|\Delta\theta_k|\le c_\theta$, then switch u_{3k} into the robot by sub-controller 3, C_3, until $|\Delta y_k|\le c_y$, $|\Delta x_k|\le c_x$, and $|\Delta\theta_k|\le c_\theta$, and stop the robot; otherwise, go to Step 1.

According to the switching strategy, if each sub-system is controlled well by its sub-controller, then the closed-loop sub-system is bounded and stable and the large-scale system is also bounded and stable because $|e(k)|=|r_k-p_k|$ is bounded. The visual feedback acts in the bandwidth response of the lower-frequency motion of the robot. Thus, the higher-frequency training data are neglected to increase the robustness of the system. The NARX controller, $u(k)$, is composed of three sub-controllers, $u_1(k)=u_{1k}$, $u_2(k)=u_{2k}$, $u_3(k)=u_{3k}$, and the parameters c_x, c_y, c_θ and Algorithm 3 are used to decide which one is switched into this nonlinear system to guide the robot. First, the reference or desired vector, $r(k)=r_k=[x_d(k),y_d(k),\theta_d(k)]^T$, and the real state vector, $p(k)=p_k=[x(k),y(k),\theta(k)]^T$, can be applied to construct a neural sub-model of each sub-system, where x, y, and θ are the states of sub-systems 1, 2, and 3, respectively. Furthermore, $p_{k+1}-p_k=\Delta p_{k+1}\approx\Delta\hat{p}_{k+1}=\hat{P}(\Delta\hat{p}_k,\Delta\hat{p}_{k-1},...,\Delta\hat{p}_{k-n},u_k,u_{k-1},...,u_{k-p},W_P)$, where $\Delta\hat{p}_{k+1}=\hat{p}_{k+1}-p_k$; the matrix W_P includes the weight and bias of $\hat{P}(\cdot)$; $n+1$ and $p+1$ are the tape delays of $\Delta\hat{p}$ and u, respectively; $\hat{P}=\{\hat{P}_1(W_{P1}),\hat{P}_2(W_{P2}),\hat{P}_3(W_{P3})\}$; $W_P=\{W_{P1},W_{P2},W_{P3}\}$; $W_{P1}=\{W_{P1x},B_{P1x},W_{P1y},B_{P1y},W_{P1\theta},B_{P1\theta}\}$; $W_{P2}=\{W_{P2x},B_{P2x},W_{P2y},B_{P2y},W_{P2\theta},B_{P2\theta}\}$; and $W_{P3}=\{W_{P3x},B_{P3x},W_{P3y},B_{P3y},W_{P3\theta},B_{P3\theta}\}$. The tracking error is defined as $e(k)=e_k=r_k-p_k=[e_{1k},e_{2k},e_{3k}]^T$, where $e_{1k}=x_d(k)-x(k)$, $e_{2k}=y_d(k)-y(k)$, $e_{3k}=\theta_d(k)-\theta(k)$, and $v(t)=[v_1(t),v_2(t),v_3(t)]^T$. The three motors in Fig. 4d, which have different properties, are controlled by a Parallax BS2 microcontroller and their rotational velocities are $v_1(t),v_2(t),v_3(t)$, where $[v_1(u_1(k)),v_2(u_1(k)),v_3(u_1(k))]^T$, $[v_1(u_2(k)),v_2(u_2(k)),v_3(u_2(k))]^T$, and $[v_1(u_3(k)),v_2(u_3(k)),v_3(u_3(k))]^T$ are the rotational commands of sub-controllers 1, 2, and 3, respectively. However, the relation between $\{u_1(k),u_2(k),u_3(k)\}=\{u_{1k},u_{2k},u_{3k}\}\in u(k)=u_k$ and $v(t)$ is unknown. The weights and biases, $W_C=\{W_{C1},W_{C2},W_{C3}\}$, of the NARX controller, where $W_{C1}=\{W_{C1x},B_{C1x},W_{C2x},B_{C2x}\}$, $W_{C2}=\{W_{C1y},B_{C1y},W_{C2y},B_{C2y}\}$, $W_{C3}=\{W_{C1\theta},B_{C1\theta},W_{C2\theta},B_{C2\theta}\}$, and $C=\{C_1(W_{C1}),C_2(W_{C2}),C_3(W_{C3})\}=u_k=C(u_{k-1},u_{k-2},...,u_{k-m},e_k,e_{k-1},...,e_{k-c},W_C)$, the matrix W_C represents the weight and bias of $C(\cdot)$ and m and c are the tape delays of u and e, respectively. The prediction error is $\hat{e}_{k+1}=r_{k+1}-\hat{p}_{k+1}$ from

Journal of Engineering Technology

Volume 6, Special Issue (Emerging Trends in Engineering Technology) March. 2018, PP. 71-96

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

$\hat{e}_k = r_k - \hat{p}_k$, the predictor $u_\chi = X(e_k)$, where $\{K_x e_{1k}, K_y e_{2k}, K_\theta e_{3k}\} \in X$, and K_x, K_y, K_θ are related to the predicted constant values of sub-systems 1, 2, and 3, respectively. If $\hat{p}_{k+1}$ is from the converged neural model and controller, then $\hat{e}_{k+1}$ is bounded and stable.

3. Results

To demonstrate the capabilities of the vision-control strategy, the following three cases are designed for comparison:

Case 1: A physical model design for control without using any camera, as described in [17]

Case 2: A genetic algorithm control design using a proposed vision technique, as described in [18]

Case 3: The proposed method shown in Fig. 2 with $K_x = 0.005, K_y = 0.003, K_\theta = 0.012$.

In Case 3, as shown in Figs. 1 and 5, the training results are obtained for sub-controllers 1–3 based on the trained input/output data related to the control commands for the motors from 400 poses of the robot. In the first neural sub-model, $W_{P1x} = \begin{bmatrix} -10.3090 \\ 2.1949 \end{bmatrix}$, $B_{P1x} = \begin{bmatrix} 11.4133 \\ -0.8483 \end{bmatrix}$ are for the hidden layer and $W_{P2x} = [-0.6145, \quad 0.3684]$, $B_{P2x} = 0.8654$ are for the output layer. In the second neural sub-model, $W_{P1y} = \begin{bmatrix} 12.3865 \\ -13.6988 \end{bmatrix}$, $B_{P1y} = \begin{bmatrix} -7.5354 \\ 4.9952 \end{bmatrix}$ are for the hidden layer and $W_{P2y} = [0.1412, \quad -0.2068]$, $B_{P2y} = 0.6129$ are for the output layer. In the third neural sub-model, $W_{P1\theta} = \begin{bmatrix} -4.273 \\ 2.5514 \end{bmatrix}$, $B_{P1\theta} = \begin{bmatrix} 5.1975 \\ -0.9716 \end{bmatrix}$ are for the hidden layer and $W_{P2\theta} = [-0.2133, \quad 0.2786]$, $B_{P2\theta} = 0.3697$ are for the output layer.

Journal of Engineering Technology

Volume 6, Special Issue (Emerging Trends in Engineering Technology) March. 2018, PP. 71-96

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

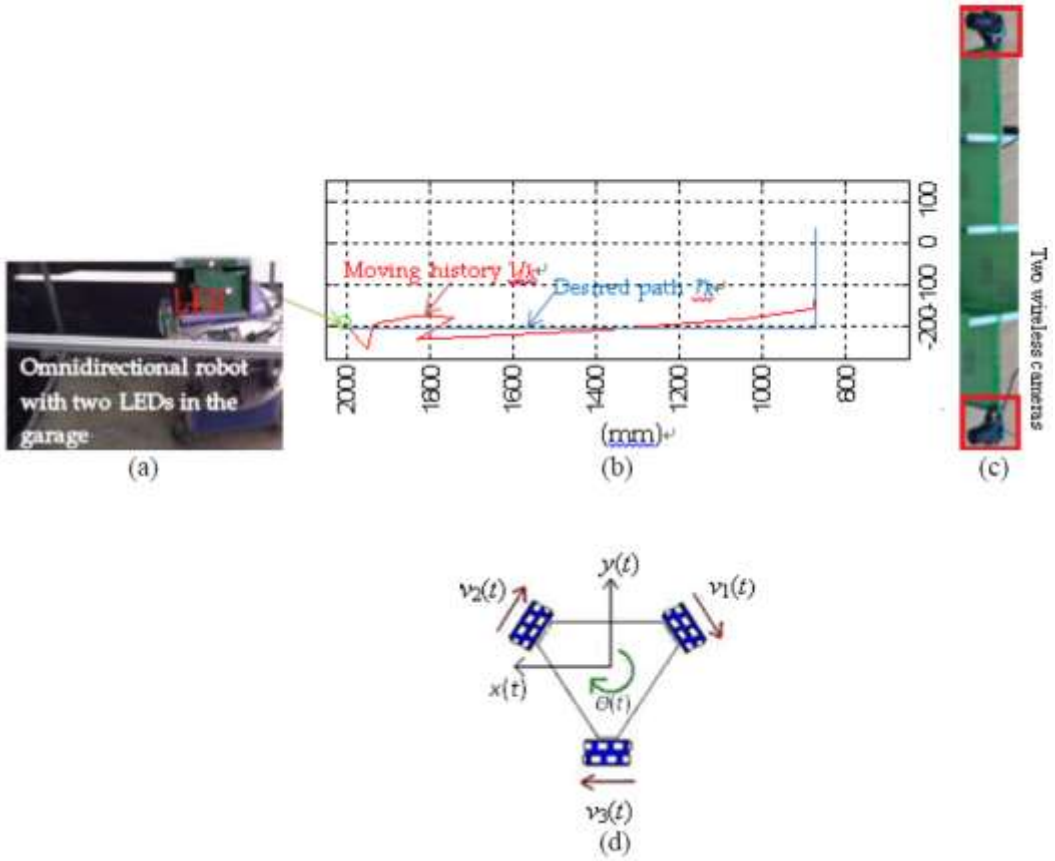


Figure 4. (a) Two LEDs fixed on the omnidirectional robot. (b) This robot is far away from the two wireless cameras (c) that are used for the prototype of the stereo-vision system. (d) The three wheels of the robot.

In this experiment, it is assumed that $x_d = -188$ mm, $y_d = 2007$ mm, and $\theta_d = 0$ and the following initial neural control parameters are used. The weights, W_{C1x}, and biases, B_{C1x}, of the hidden layer of sub-controller 1 are obtained using the proposed genetic algorithm such that

$$W_{C1x} = \begin{bmatrix} 1.9871, & -0.0003, & 0.0001, & -0.0005, & 0 \\ 1.8675, & 2.1766, & 2.3628, & -1.576, & 1.1309 \\ 0.0389, & 0, & 0, & 0, & 0 \\ -5.8699, & 3.8169, & -6.1318, & -4.8098, & -7.1897 \\ -1.042, & 1.4338, & -1.0524, & 1.2076, & 0.4349 \\ -2.5389, & 2.3157, & -3.9693, & 3.3156, & 0.2336 \end{bmatrix} \text{ and } B_{C1x} = \begin{bmatrix} -2.3309 \\ -2.5444 \\ -0.0029 \\ 11.2999 \\ -1.6944 \\ -2.7055 \end{bmatrix}.$$

For the output layer of sub-controller 1, $W_{C2x} = [0.0011,\ 0,\ 38.543,\ 0,\ 0,\ 0]$ and $B_{C2x} = 0.1117$.

Journal of Engineering Technology

Volume 6, Special Issue (Emerging Trends in Engineering Technology) March. 2018, PP. 71-96

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

For the hidden layer of sub-controller 2, $W_{C1y} = \begin{bmatrix} 3.0449, & -5.3559, & 7.3066, & -0.1302, & 4.8863 \\ -4.3606, & -0.0161, & -0.0881, & -0.0319, & -0.0126 \\ -0.1994, & 0.0009, & 0.013, & 0.0042, & 0.001 \\ -1.2721, & -1.0959, & -2.3448, & -2.8852, & 4.6081 \\ -0.2205, & -0.0009, & -0.0137, & -0.0044, & -0.0011 \\ -4.4739, & -1.3168, & 1.2751, & -8.373, & -5.4394 \end{bmatrix}$ and

$B_{C1y} = \begin{bmatrix} -2.9948 \\ 2.2703 \\ 0.0905 \\ 2.8676 \\ 0.121 \\ 5.3092 \end{bmatrix}$.

For the output layer of sub-controller 2, $W_{C2y} = [0, 0.0109, -14.7521, 0, -14.0596, 0]$ and $B_{C2y} = 3.0154$.

For the hidden layer of sub-controller 3, $W_{C1\theta} = \begin{bmatrix} -5.5395, & 0.2384, & 0.0296, & 1.2137, & 0.9881 \\ -0.4269, & -0.2136, & -1.2461, & 1.1948, & -0.246 \\ -0.4118, & -0.0002, & 0.0016, & 0.0004, & 0.0013 \\ -0.8855, & 0.2679, & -0.7551, & 0.0573, & -0.4525 \\ 1.6958, & 0.9569, & -0.5721, & -2.6424, & -0.618 \\ -5.4698, & -0.4023, & 0.4952, & -0.8294, & 0.024 \end{bmatrix}$ and

$B_{C1\theta} = \begin{bmatrix} 2.8287 \\ 1.1326 \\ 0.2641 \\ 0.6014 \\ 1.553 \\ 0.8677 \end{bmatrix}$. For the output layer of sub-controller 3,

$W_{C2\theta} = [0.0009, -0.0001, -1.1692, -0.0033, 0.0003, -0.0029]$ and $B_{C2\theta} = 0.3053$.

Further, $S_{e1} = 1/300,\ S_{u1} = 1,\ S_{e2} = 1/2000,\ S_{u2} = 1,\ S_{e3} = 1/40,\ S_{u3} = 1$, $\hat{S}_{u1} = \hat{S}_{u2} = \hat{S}_{u3} = 1,$ $S_1 = 300,\ S_2 = 1200,\ S_3 = 40$. The switching parameters are $c_\theta = 5$, $c_x = 35$, $c_y = 35$.

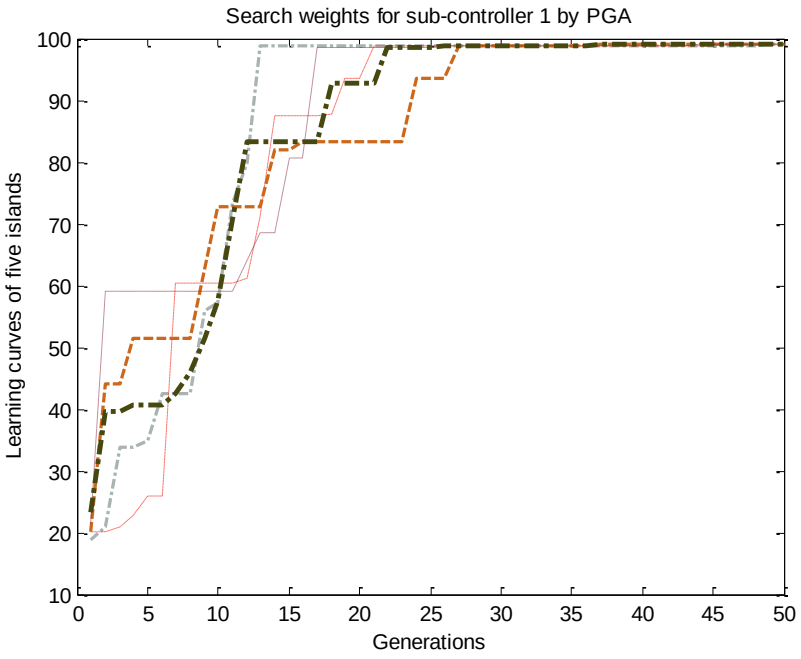
The results shown in Fig. 6a demonstrate that the methods of Cases 1 and 2 do not guide this robot well on a rocky road. From Figs. 4 and 6, the proposed control process for Case 3 designs a variable index that is defined as follows:

Index = 0: The motion of the robot is an approximate rotation to the left or right.
Index = 1: The motion of the robot is an approximate shift to the right.
Index = 2: The motion of the robot is an approximate shift to the left.
Index = 3: The motion of the robot is an approximate straight reverse.
Index = 4: The motion of the robot is an approximate straight advance.

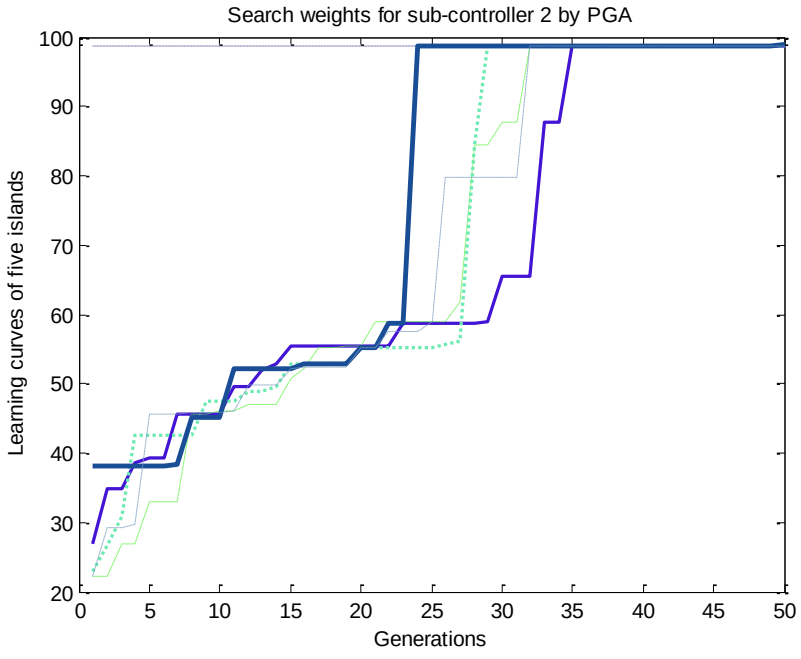
Journal of Engineering Technology

Volume 6, Special Issue (Emerging Trends in Engineering Technology) March. 2018, PP. 71-96

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY



(a)

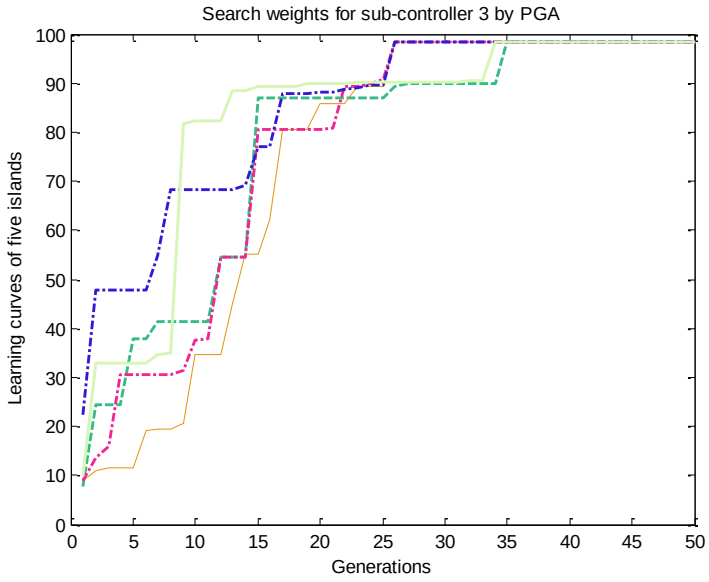


(b)

Journal of Engineering Technology

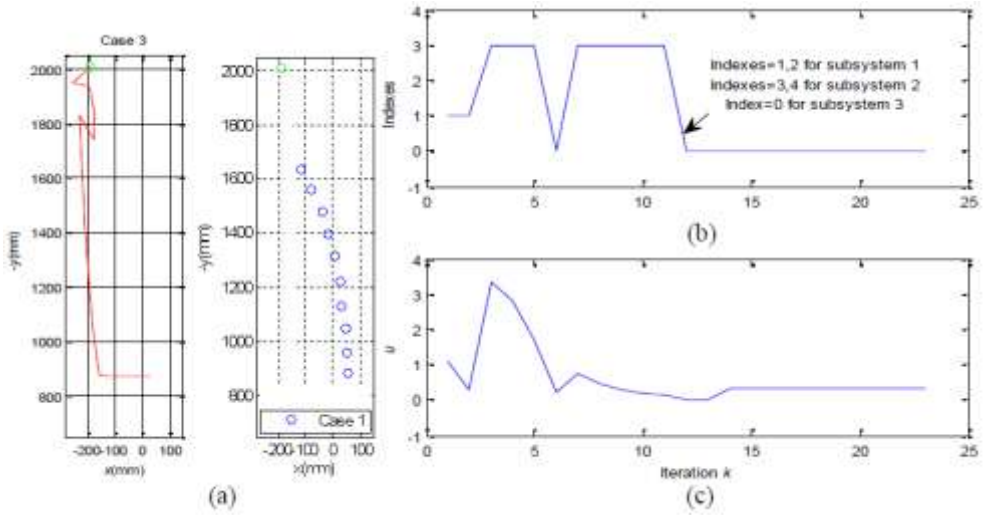
Volume 6, Special Issue (Emerging Trends in Engineering Technology) March. 2018, PP. 71-96

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY



(c)

Figure 5. Learning curves of the three neural sub-controllers ((a),(b), and (c)) by PGA

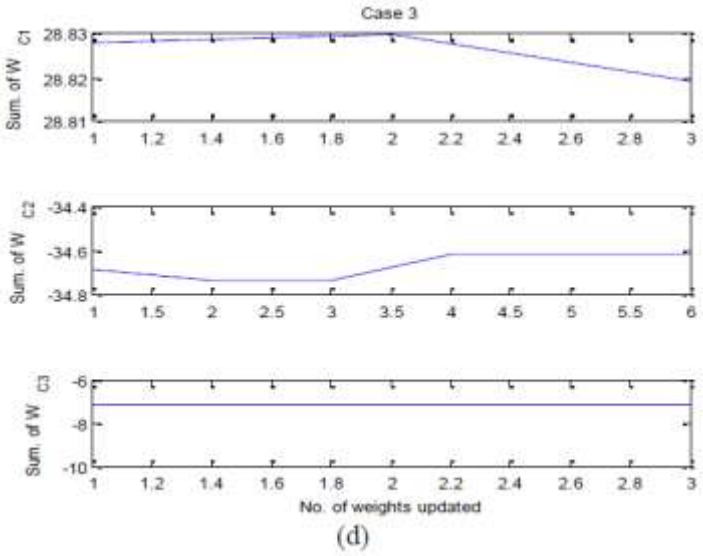


(a) (b) (c)

Journal of Engineering Technology

Volume 6, Special Issue (Emerging Trends in Engineering Technology) March. 2018, PP. 71-96

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY



(d)

Figure 6. Experimental results obtained by the method of Case 3: (a) The real motion path of the robot is compared with that obtained by the method of Case 1. (b) The proposed control process switches to one of the three sub-controllers into one of three sub-systems to guide the robot by using the method of Case 3. (c) The control command of the large-scale controller u at k in the method of Case 3. (d) The summations of the adaptive parameters, Wc1, Wc2, and Wc3, represent the convergence of the neural controller, C.

4. Conclusion

This study combines two techniques: control and image processing. A flexible large-scale control strategy and a stereo-vision process are used to address the control problem of the modeling error. In this approach, the use of two LED as markers for position detection provides more stability than the other marker designs to guide the robot more accurately. The positions of the front and back LEDs define the direction of the centerline of the robot, thus acting as a compass function but with higher precision. Moreover, this robot does not require encoders, laser sensors, ultrasound sensors, or a compass. Finally, the two LEDs guide the omnidirectional robot to successfully follow the desired path, while the other cases fail in this task.

Acknowledgments

The authors thank the Ministry of Science and Technology, R.O.C., for the support under contracts MOST 106-2632-B-468-001 and 106-2221-E-468-023. Our gratitude also goes to Michael Burton, editor, Asia University.

References

[1] P. Babu, V. Rajamani, K. Balasubramanian, Multipeak mean based optimized histogram modification framework using swarm intelligence for image contrast enhancement, Mathematical Problems in Engineering 2015 (2015) 1-14.

[2] C.P. Jose, M.T. Antonio, R.R. Juan, G.T. Jorge, T. Victor, T.P. Jose, Bilateral image subtraction and multivariate models for the automated triaging of screening mammograms, BioMed Research International 2015 (2015) 1-12.

[3] X. Fan, P. Shi, J. Ni, M. Li, A thermal infrared and visible images fusion based approach for multitarget detection under complex environment, Mathematical Problems in Engineering 2015 (2015) 1-11.

[4] J. Wang, A. Sumalee, Automatic freeway incident detection for free flow conditions: a vehicle reidentification based approach using image data from sparsely distributed video cameras, Mathematical Problems in Engineering 2015 (2015) 1-13.

[5] Z.R. Tsai, Robust kinect-based guidance and positioning of a multidirectional robot by Log-ab recognition, Expert Systems with Applications 41 (2014) 1271-1282.

[6] T.K. Tanev, Geometric algebra based kinematics model and singularity of a hybrid surgical robot, Advances in Robot Kinematics 4 (2016) 431-440.

[7] H. Jin, Q. Chen, Z. Chen, Y. Hu, J. Zhang, Multi-LeapMotion sensor based demonstration for robotic refine tabletop object manipulation task, CAAI Transactions on Intelligence Technology 1 (2016) 104-113.

[8] P. Serra, R. Cunha, T. Hamel, D. Cabecinhas, C. Silvestre, Landing of a quadrotor on a moving target using dynamic image-based visual servo control, IEEE Transactions on Robotics 32 (2016) 1524-1535.

[9] J. Li, X. Li, B. Yang, X. Sun, Segmentation-based image copy-move forgery detection scheme, IEEE Transactions on Information Forensics and Security 10 (2015) 507-518.

[10] S. Cubero, W.S. Lee, N. Aleixos, F. Albert, J. Blasco, Automated systems based on machine vision for inspecting citrus fruits from the field to postharvest—a review, Food and Bioprocess Technology 9 (2016) 1623-1639.

[11] Z.R. Tsai, Multipurpose design of an LED dimming system, Journal of Engineering Technology 6 (2017) 754-763.

[12] H. Bickel, K. Pinker, S. Polanec, H. Magometschnigg, G. Wengert, C. Spick, W. Bogner, B.H. Zsuzsanna, T.H. Helbich, P. Baltzer, Diffusion-weighted imaging of breast lesions: Region-of-interest placement and different ADC parameters influence apparent diffusion coefficient values, European Radiology 27 (2017) 1883-1892.

[13] Z.R. Tsai, Y.Z. Chang, Two-stage digital redesign of a cooling system with time delay by dither, Journal of Engineering Technology 6 (2017) 764-775.

[14] M. Gong, Y. Liang, J. Shi, W. Ma, J. Ma, Fuzzy c-means clustering with local information and kernel metric for image segmentation, IEEE Transactions on Image Processing 22 (2013) 573-584.

[15] Z. Zhang, Camera calibration: a personal retrospective, Machine Vision and Applications 27 (2016) 963-965.

[16] G. Ligorio, A.M. Sabatini, A novel Kalman filter for human motion tracking with an inertial-based dynamic inclinometer, IEEE Transactions on Biomedical Engineering 62 (2015) 2033-2043.

[17] L.C. Lin, H.Y. Shih, Modeling and adaptive control of an omni-mecanum-wheeled robot, Intelligent Control and Automation 4 (2013) 166-179.

[18] R. Koker, A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization, Information Sciences 222 (2013) 528-543.

Journal of Engineering Technology

Volume 6, Special Issue (Emerging Trends in Engineering Technology) March. 2018, PP. 71-96

RESEARCH ARTICLE -ENGINEERNG TECHNOLOGY

APPENDIX

The Lyapunov candidate for minimizing the modeling error is designed as follows:

$$V_{1k} = \frac{1}{2}(\hat{p}_k - p_k)^T(\hat{p}_k - p_k) = \frac{1}{2}\left\|\hat{p}_k - \bar{p}_k + \bar{p}_k - p_k\right\|^2 = \frac{1}{2}\left\|\hat{p}_k - \bar{p}_k\right\|^2 + \varepsilon_k = V_{2k} + \varepsilon_k,$$

where $\hat{p}_k - p_k$ is the learning error between the model and the plant; $V_{2k} = \frac{1}{2}\left\|\hat{p}_k - \bar{p}_k\right\|^2$ includes the reality, $\bar{p}_k$; $\varepsilon_k = \frac{1}{2}\left\|\bar{p}_k - p_k\right\|^2 + (\hat{p}_k - \bar{p}_k)^T(\bar{p}_k - p_k)$ is the modeling error; and $\bar{p}_k \approx p_k$. Next, V_{2k} needs to be minimized by the following updating law for training the neural model:

$$\frac{(\Delta W_P)_k}{\eta_P(k)} = -\frac{\partial V_{2k}}{\partial W_P} = -(\hat{p}_k - \bar{p}_k)\frac{\partial(\hat{p}_k - \bar{p}_k)}{\partial W_P} = -(\hat{p}_k - \bar{p}_k)\frac{\partial \hat{p}_k}{\partial W_P},$$

where $(\Delta W_P)_k = W_P(k+1) - W_P(k)$. Then, the Lyapunov candidate for training the neural controller is designed as $V_3(k) = V_{3k} = \frac{1}{2}(\hat{p}_k - r_k)^T(\hat{p}_k - r_k) = \frac{1}{2}\left\|\hat{p}_k - r_k\right\|^2$. Furthermore, the change in the Lyapunov candidate, V_3, can be obtained as $V_3(k+1) - V_3(k) = \frac{1}{2}(\left\|\hat{p}(k+1) - r(k+1)\right\|^2 - \left\|\hat{p}(k) - r(k)\right\|^2)$. Finally, the adaptation law is obtained to train the parameters of the controller: $\frac{(\Delta W_C)_k^T}{\eta_C(k)} \approx -\frac{\partial V_3(k)}{\partial (W_C)_k} = -(\hat{p}_k - r_k)S_k\frac{\partial u_k}{\partial (W_C)_k}$.

The stable learning rates of the convergence theorems can help in stabilizing the neural network. First, the change in $e_P(k) = \hat{p}_k - \bar{p}_k$ is calculated using the following equations:

$$e_P(k+1) = e_P(k) - \left[\frac{\partial e_P(k)}{\partial (W_P)_k}\right]^T\left[\eta_P(k)e_P(k)\frac{\partial e_P(k)}{\partial (W_P)_k}\right] = e_P(k)\left(1 - \left[\frac{\partial e_P(k)}{\partial (W_P)_k}\right]^T\eta_P(k)\frac{\partial e_P(k)}{\partial (W_P)_k}\right)$$

$$= e_P(k)\left(1 - \left[\frac{\partial \hat{p}_k}{\partial (W_P)_k}\right]^T\eta_P(k)\frac{\partial \hat{p}_k}{\partial (W_P)_k}\right)$$

and $V_2(k+1) - V_2(k) = \frac{1}{2}(\left\|e_P(k+1)\right\|^2 - \left\|e_P(k)\right\|^2) = \frac{1}{2}\left(\left\|e_P(k)\left(1 - \left[\frac{\partial \hat{p}_k}{\partial (W_P)_k}\right]^T\eta_P(k)\frac{\partial \hat{p}_k}{\partial (W_P)_k}\right)\right\|^2 - \left\|e_P(k)\right\|^2\right)$

$$= \frac{1}{2}\left\|e_P(k)\right\|^2\left[\left(1 - \left[\frac{\partial \hat{p}_k}{\partial (W_P)_k}\right]^T\eta_P(k)\frac{\partial \hat{p}_k}{\partial (W_P)_k}\right)^2 - 1\right].$$

Hence, if $-1 < \left(1 - \left[\frac{\partial \hat{p}_k}{\partial (W_P)_k}\right]^T\eta_P(k)\frac{\partial \hat{p}_k}{\partial (W_P)_k}\right) < 1$ and $\left\|\bar{p}_k - p_k\right\| \le \bar{\varepsilon}$, then $V_2(k+1) < V_2(k)$. That is, $V_2(k) \to 0$ or $\hat{p}_k \to \bar{p}_k$ is UUB.

Theorem 2 can also be proved using a similar approach.