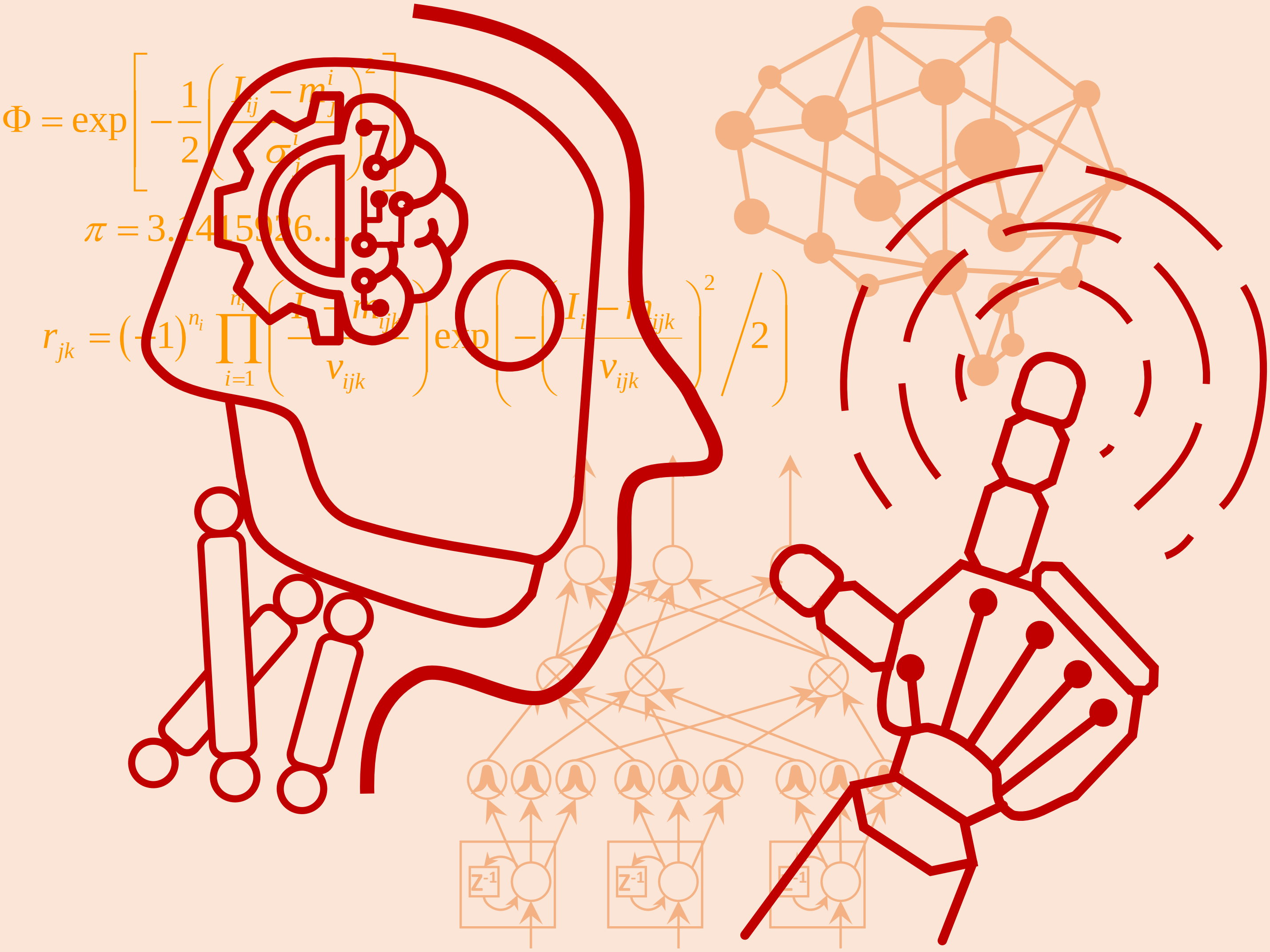
ISSN 2616-8170

RST
Robotics Society of
Taiwan

iRobotics

VOLUME 3
NUMBER 1
MARCH 2020


$\Phi = \exp\left[-\frac{1}{2}\left(\frac{I_{ij}-m_j^i}{\sigma_j^i}\right)^2\right]$
$\pi = 3.1415926...$
$r_{jk} = (-1)^{n_i}\prod_{i=1}^{n_i}\left(\frac{I_i - m_{ijk}}{v_{ijk}}\right)\exp\left(-\left(\frac{I_i - m_{ijk}}{v_{ijk}}\right)^2/2\right)$
Z^{-1}
Z^{-1}
Z^{-1}

PUBLISHED BY THE ROBOTICS SOCIETY OF TAIWAN

*i*Robotics

EDITORIAL BOARD

PUBLISHER

Robotics Society of TAIWAN (RST)
Society President: Chung-Hsien Kuo

Department of Electrical Engineering, National Taiwan University of Science and Technology
Taipei, Taiwan

Tel: +886-2-2736-2446
URL: http://www.rst.org.tw

The ***i*Robotics** is published quarterly each year by the Robotics Society of Taiwan (RST). Institutional rate: US$140 annually; individual annual subscription rate: US$50 for nonmembers, US$25 for members (including postage). Note that another US$100 is needed if the express is required.

Parameter Identification of Delta Robot Dynamics

Meng-Kun Liu, Chien Wang, and Chen-Yang Lan

***Abstract*—This research introduces a novel approach to estimate the dynamic model parameters of the delta robot. The simplified delta robot model is developed by using virtual work principle. This method is efficient and can be computed in real time for the control propose. To estimate the dynamic model parameters, an excitation trajectory is generated based on Fourier series, and the unknown parameters of the dynamic model is estimated by using least mean square algorithm. The simulation result indicates that this method can estimate the dynamic model parameters of delta robot accurately in real time.**

***Index Terms*— parameter identification, delta robot, Fourier series, least mean square algorithm**

I. Introduction

THE application of robot manipulator has become more and more popular in the industry. The robot manipulators can be divided into two categories: series and parallel. The series manipulator performs an open movement, that is, its mechanism is not closed, while the parallel manipulator is the opposite. Compared with the series manipulator, the parallel manipulator has the advantages of high rigidity, low inertia, and simple structure, but its working space is limited. The parallel manipulator is usually used in situations where fast movement is required and the workspace is not large. Failures of the robot manipulator could cause loss of precision, mechanism interference, production shutdown and human injury. Therefore, fault detection of robot manipulator attracts a wide attention. The current fault detection methods can be divided into three types: sensor redundancy, signal-based method, and model-based method.

The sensor redundancy is the simplest method for the fault detection. It installs multiple sensors to the system, and the diagnosis is performed by comparing the measurement from identical sensors [1-3]. This method can effectively detect the faults, but the installation of sensors is expensive and it reduces the workspace of robot manipulator. Therefore, in recent years, people have started to use mathematical methods to replace a large number of sensors [4].

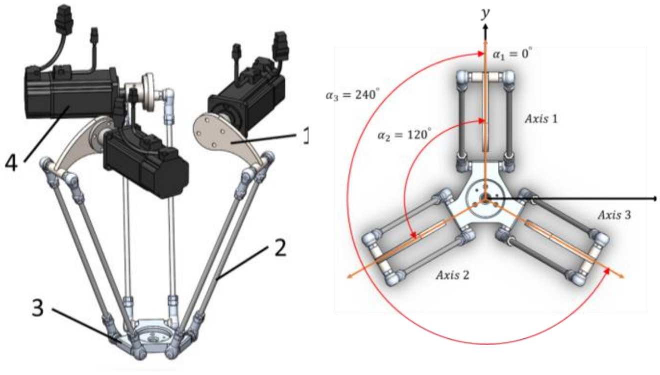
The model-based method establishes a system dynamic model, and compares the model output and the actual output. When the difference between the two exceeds a preset threshold, it indicates that a fault occurs. The model-based method requires an accurate dynamic model and parameters [5]. Therefore, this research proposes a novel method to estimate the parameters of the Delta robot dynamic model.

II. Dynamic Model of Delta Robot

The Delta robot discussed in this research consists of three upper arms (1), three sets of forearms (2), a travelling plate (3), and three sets of actuators (4), as shown in Fig. 1(a). The global coordinate of the Delta robot is defined in Fig. 1(b). Owing to the robot's triple symmetry, one can calculate the movement of one arm and apply the transformation matrix to retrieve the movement of the other arms. The transformation matrix is given by

$$R_i = \begin{bmatrix} \cos(\alpha_\mathrm{i}) & -\sin(\alpha_\mathrm{i}) & 0 \\ \sin(\alpha_\mathrm{i}) & \cos(\alpha_\mathrm{i}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Where i = 1, 2, 3 and $\alpha_1 = 0^\circ, \alpha_2 = 120^\circ, \alpha_3 = 240^\circ$.



(a) (b)

Fig. 1 (a) configuration and (b) global coordinate of the Delta robot

As shown in Fig.2, the origin *O* locates at the center of the delta robot, and the center of the travelling plate locates at the point *P*. In order to simplify the dynamic model of the delta robot, the end of the forearm C'_i has been shifted to C_i. It makes three sets of the forearm intersects at the point P. As shown in Fig. 2, $L_3 = R - r, L_2 = \overline{B_i C_i}$. The dimensions of the delta robot is: R = 100 mm, r = 45 mm, L_1 = 85 mm, and L_2 = 377 mm.

The dynamic model of the delta robot is derived based on the virtual work principle proposed by Codourey [6]. By using this

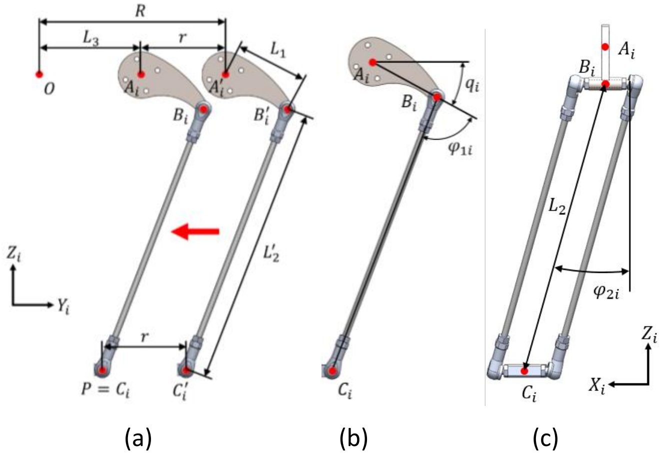
This work was financially supported by the "Center for Cyber-Physical System Innovation" from The Featured Area Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan. Part of the funding also came from the Ministry of Science and Technology (MOST) in Taiwan under Grant no. MOST 108-2221-E-011-160-.

Meng-Kun Liu is with the Department of Mechanical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. (mkliu@mail.ntust.edu.tw)

Chien Wang got his master degree of mechanical engineering in National Taiwan University of Science and Technology. He is now with Taiwan Semiconductor Manufacturing Company. (sam40207@hotmail.com.tw)

Chen-Yang Lan is with the Department of Mechanical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. (jimmylan@mail.ntust.edu.tw)

method, the mass matrix of the robot can be computed separately based on kinetic energy consideration. The model thus obtained is efficient and can be computed in real time for control proposes.



(a) (b) (c)

Fig. 2. The dimension of the forearm

The delta robot consists of one traveling plate, three upper arms, and three forearms. The mass matrix of the robot is the sum of the contributions of traveling plate, upper arm, and forearm:

$$A = A_n + A_u + A_f \qquad (2)$$

The mass matrix of the traveling plate can be represented as:

$$A_n = m_n J^T J \qquad (3)$$

Where J is the Jacobian matrix of the Delta robot and m_n is the mass of the traveling plate and the payload. The contribution of the upper arms can be compacted into one matrix as:

$$A_u = I_b = \begin{bmatrix} I_{b1} & 0 & 0 \\ 0 & I_{b2} & 0 \\ 0 & 0 & I_{b3} \end{bmatrix} \qquad (4)$$

Where $I_{b1} = I_{b2} = I_{b3} = I_{bi}$ and

$$I_{bi} = I_m + L_A^2\left(\frac{m_b}{3} + m_c\right) \qquad (5)$$

Where I_m is the inertia of the motor, m_b is the mass of the upper arm and m_c is the mass of the elbow. The velocity at the lower end of the forearm is the velocity of the traveling plate, and the velocity at the upper end of the forearm can be defined as:

$$v_{u,i} = -R_i \begin{bmatrix} L_A \sin q_i \\ 0 \\ L_A \cos q_i \end{bmatrix} \dot{q}_i \qquad (6)$$

Written as a function of the joint-velocity q_i for each forearm, it becomes:

$$v_{u,1} = \begin{bmatrix} -R_1 \begin{bmatrix} L_A \sin q_1 \\ 0 \\ L_A \cos q_1 \end{bmatrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix} \end{bmatrix} \dot{q} = J_{u,1}\dot{q} \qquad (7)$$

$$v_{u,2} = \begin{bmatrix} \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} & -R_2 \begin{bmatrix} L_A \sin q_2 \\ 0 \\ L_A \cos q_2 \end{bmatrix} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \end{bmatrix} \dot{q} = J_{u,2}\dot{q} \qquad (8)$$

$$v_{u,3} = \begin{bmatrix} \begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix} & -R_3 \begin{bmatrix} L_A \sin q_3 \\ 0 \\ L_A \cos q_3 \end{bmatrix} \end{bmatrix} \dot{q} = J_{u,3}\dot{q} \qquad (9)$$

Thus the contribution of each forearm to the mass matrix of the robot can be calculated as:

$$A_{b,i} = \frac{1}{3} m_{ab}\left(J^T J + J_{u,i}^T J_{u,i} + J_{u,i}^T J\right) \qquad (10)$$

Where m_{ab} is the mass of the forearm. Hence the mass matrix of the Delta robot can be represented as:

$$A = I_b + m_n J^T J + \sum_{i=1}^{3} \frac{1}{3} m_{ab}\left(J^T J + J_{u,i}^T J_{u,i} + J_{u,i}^T J\right) \qquad (11)$$

However, the inertia matrix of the three forearms (the third term of the above equation) would vary depending on the motor angle and the position of the traveling plate. To simplify eqn(11), we put one-third of the forearm mass into the traveling plate, and the remaining two-third to the upper arm. Hence the third term in the above equation can be erased. The simplified inertia matrix can be shown as:

$$A = I_{bt} + m_{nt} J^T J \qquad (12)$$

Where

$$I_{bt} = \begin{bmatrix} I_{bt1} & 0 & 0 \\ 0 & I_{bt2} & 0 \\ 0 & 0 & I_{bt3} \end{bmatrix},\ I_{bti} = I_m + L_1^2\left(\frac{1}{3} m_b + m_c + \frac{2}{3} m_{ab}\right) \qquad (13)$$

The torque of the motor τ_m can be represented as:

$$\tau_m = \tau_n + \sum_{i=1}^{3} \tau_{b,i} + \sum_{i=1}^{3} \tau_{ab,i} \qquad (14)$$

Where τ_n is the contribution of the force acting on the travelling plate, $\tau_{b,i}$ is the force/torque contribution of the upper arm i, and $\tau_{ab,i}$ is the force/torque contribution of the forearm i. Because the traveling plate only performs translation, its torque equation becomes:

$$\tau_n = J^T m_n \left(\ddot{P} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}\right) \qquad (15)$$

Where g is the gravitational acceleration. On the other hand, the upper-arm only performs rotation, and its torque equation is:

$$\tau_b = I_b \ddot{q} + \left(\frac{m_b}{2} + m_c\right) g L_1 \begin{bmatrix} \cos(q_1) \\ \cos(q_2) \\ \cos(q_3) \end{bmatrix} \qquad (16)$$

The gravitational force of each forearm can be placed by one half of the mass at the upper-arm and one half of the mass at the traveling plate. Combining eqn(14), (15) and (16), the dynamic model of the Delta robot becomes as follows:

$$\tau_m = I_{bt}\ddot{q} + m_{nt}J^T\ddot{P} + m_{ng}J^T\begin{bmatrix}0\\0\\g\end{bmatrix} + m_{na}\begin{bmatrix}\cos(q1)\\ \cos(q2)\\ \cos(q3)\end{bmatrix} + F(\dot{q}) \qquad (17)$$

Where

$$m_{nt} = m_n + m_{ab} + 3m_c \qquad (18)$$

$$m_{ng} = m_n + \frac{3}{2}m_{ab}$$

$$m_{na} = \left(\frac{1}{2}m_b + m_c + \frac{1}{2}m_{ab}\right)gL_1$$

And the friction term $F(\dot{q})$ can be shown as:

$$F(\dot{q}) = f_c sng(\dot{q}) + f_v\dot{q} \qquad (19)$$

Where f_c is the Coulomb friction coefficient and f_v is the viscous friction. Their matrixes can be represented as:

$$f_c = \begin{bmatrix} f_{c1} & 0 & 0\\ 0 & f_{c2} & 0\\ 0 & 0 & f_{c3}\end{bmatrix},\ f_v = \begin{bmatrix} f_{v1} & 0 & 0\\ 0 & f_{v2} & 0\\ 0 & 0 & f_{v3}\end{bmatrix} \qquad (20)$$

$$\operatorname{sgn}(\dot{q}) = \begin{bmatrix}\operatorname{sgn}(\dot{q}_1)\\ \operatorname{sgn}(\dot{q}_2)\\ \operatorname{sgn}(\dot{q}_3)\end{bmatrix}\quad \operatorname{sgn}(\dot{q}_i) = \begin{cases}1, & \text{if } \dot{q}_i > 0\\ 0, & \text{if } \dot{q}_i = 0\\ 0, & \text{if } \dot{q}_i < 0\end{cases} \qquad (21)$$

III. Estimation of parameters

A. Least Mean Square Algorithm

The dynamic model is necessary for the controller design. However, because several crucial parameters such as friction are hard to measure, system identification method is applied to estimate the value of these parameters. Least mean square algorithm is applied in this research. Assume the unknown parameters of the Delta robot can be formulated as:

$$x = \big[I_{bt1} \quad I_{bt2} \quad I_{bt3} \quad m_{nt} \quad m_{ng} \quad m_{l1} \quad m_{l2} \quad m_{l3} \quad f_{c1} \quad f_{c2} \quad f_{c3} \quad f_{v1} \quad f_{v2} \quad f_{v3}\big] \qquad (22)$$

The measured torque of the three motors are:

$$y = \big[\tau_{m1} \quad \tau_{m2} \quad \tau_{m3}\big]^T \qquad (23)$$

Where m = 14. The basis function is

The basis matrix is defined as

$$h = \left[\begin{matrix}\ddot{q}_1 & 0 & 0 & & & \cos(q_1) & 0 & 0\\ 0 & \ddot{q}_2 & 0 & J^T\ddot{P} & J^T\begin{bmatrix}0\\0\\g\end{bmatrix} & 0 & \cos(q_1) & 0\\ 0 & 0 & \ddot{q}_3 & & & 0 & 0 & \cos(q_1)\end{matrix}\right. \qquad (24)$$

$$\left.\begin{matrix}\operatorname{sgn}(\dot{q}_1) & 0 & 0 & \dot{q}_1 & 0 & 0\\ 0 & \operatorname{sgn}(\dot{q}_2) & 0 & 0 & \dot{q}_2 & 0\\ 0 & 0 & \operatorname{sgn}(\dot{q}_3) & 0 & 0 & \dot{q}_3\end{matrix}\right]$$

and

$$H = \big[h_1 \quad h_2 \quad \dots \quad h_m\big]^T$$

$$Y = \big[y_1 \quad y_2 \quad \dots \quad y_m\big]^T$$

Where m = 14 to make H full rank. The parameters of the Delta robot can be estimated by using

$$\hat{x} = (\mathrm{H}^T\mathrm{H})^{-1}H^TY \qquad (25)$$

B. Excitation Trajectory

The excitation trajectory is generated by using 3rd-order Fourier series to form the motor rotation angle. They are

$$\phi_1(\mathrm{t}) = \phi_{10} + \sum_{i=1}^{3} a_{1i}\sin(2\pi f_{1i}\,\mathrm{t}) + b_{1i}\cos(2\pi f_{1i}\,\mathrm{t})$$

$$\phi_2(\mathrm{t}) = \phi_{20} + \sum_{i=1}^{3} a_{2i}\sin(2\pi f_{2i}\,\mathrm{t}) + b_{2i}\cos(2\pi f_{2i}\,\mathrm{t}) \quad (26)$$

$$\phi_3(\mathrm{t}) = \phi_{30} + \sum_{i=1}^{3} a_{3i}\sin(2\pi f_{3i}\,\mathrm{t}) + b_{3i}\cos(2\pi f_{3i}\,\mathrm{t})$$

The rotation frequency is intentionally set to $f_1 = 0.41Hz$, $f_2 = 0.45Hz$, and $f_3 = 0.43Hz$. The constraints of the Delta is listed in Table I.

TABLE I
Constraints of the delta robot

Parameters	Operating rage
Motor rotation angle ϕ_i	$-60^{\circ} \le \phi_i \le -10^{\circ}$
rotation speed $\dot{\phi}_i$ (rad/s)	$-1.5 \le \dot{\phi}_i \le 1.5$
rotation acceleration $\ddot{\phi}_i$ (rad/s²)	$-15 \le \ddot{\phi}_i \le 15$
platform position x (m)	$-0.06 \le x \le 0.06$
platform position y (m)	$-0.06 \le x \le 0.06$
platform position z (m)	$-0.38 \le z \le -0.35$

The optimization method proposed in [7] is applied to calculate $a_{1i}, a_{2i,}\ a_{3i}, b_{1i}, b_{2i}, b_{3i}$ of the Fourier series trajectory. Their values are shown in Table II.

TABLE II
COEFFICIENTS OF THE FOURIER SERIES TRAJECTORY

	ϕ_{i0}	a_{i1}	a_{i2}	a_{i3}	b_{i1}	b_{i2}	b_{i3}
$i = 0$	0.5	-0.028	-0.003	0.059	-0.084	0.02	-0.018
$i = 1$	0.5	-0.094	0.001	0.044	0.111	-0.361	0.025
$i = 2$	0.5	-0.062	0.012	0.014	0.082	-0.012	-0.13

The unit of $\phi_i(\mathrm{t})$ is in radius.

The trajectory of the center of the traveling plate is shown in Fig. 3. By applying torque control of the three servo motors to follow the Fourier series trajectory, the unknown parameters of the Delta robot can be estimated by using the least mean square algorithm. The estimation error of each parameter is shown in Table III. All the parameters of the Delta robot dynamic, including Coulomb friction and viscous friction coefficients, can be estimated accurately.

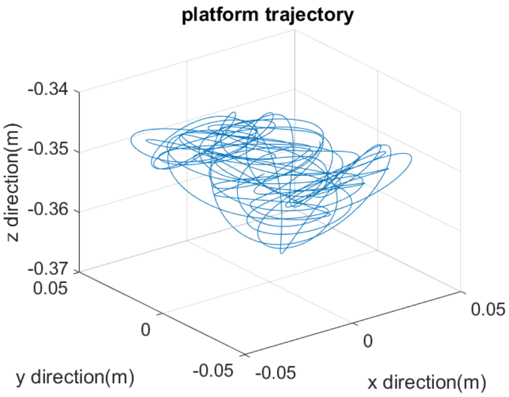


Fig. 3. The trajectory of the traveling plate

TABLE III
ESTIMATION ERROR OF DYNAMIC MODEL PARAMETERS

Parameter	Estimation error
I_{bt1}	0%
I_{bt2}	0%
I_{bt3}	0%
m_{nt}	1.09%
m_{ng}	0.33%
m_{l1}	0.4%
m_{l2}	0.4%
m_{l3}	0.4%
f_{c1}	4%
f_{c2}	0%
f_{c3}	1.3%
f_{v1}	2.7%
f_{v2}	0.6%
f_{v3}	1%

C. Application to Delta Robot

The proposed identification method is applied to the Delta robot developed in NTUST, as shown in Fig. 4, and the estimation error is indicated in Table IV. It can be seen that although the errors of I_{bt1}, m_{nt}, and m_{ng} are within 10%, most parameters deviate from their theoretical values. Unbalance is observed on the inertia and the mass parameters of the forearms. The large error could be caused by the misalignment of the Delta robot, model uncertainty, and un-modeled dynamics. However, each parameter's coefficient of variation remains small for five repeated experiments. It means that the proposed method is able to provide a stable estimation, and the estimated value is very sensitive to the robot dynamics in the real-world scenario. Hence the proposed method could be used to monitor the status of the Delta robot.

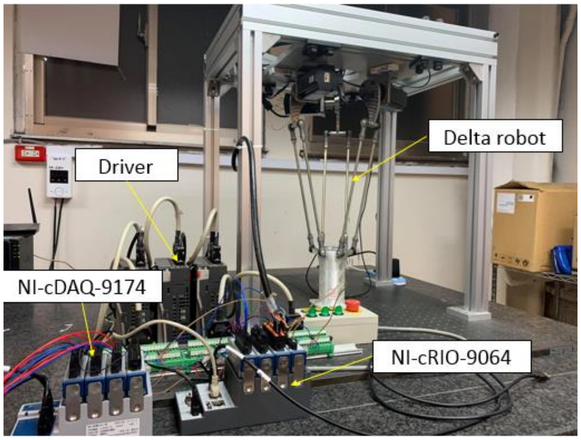


Fig. 4. The Delta robot developed in NTUST

TABLE IV
ESTIMATION ERROR OF DELTA ROBOT DYNAMICS

Parameter	Estimation error	Coefficients of variation
I_{bt1}	4.3%	1.2%
I_{bt2}	21.7%	2.1%
I_{bt3}	30.4%	3.1%
m_{nt}	4%	0.2%
m_{ng}	8.5%	0.2%
m_{l1}	13.4%	1.0%
m_{l2}	25.9%	0.4%
m_{l3}	27.9%	0.4%
f_{c1}	N/A	2.0%
f_{c2}	N/A	3.9%
f_{c3}	N/A	3.2%
f_{v1}	N/A	2.3%
f_{v2}	N/A	6.3%
f_{v3}	N/A	4.5%

IV. CONCLUSION

A novel method to identify the dynamic parameters of the Delta robot dynamics is proposed in this research. The dynamic model of Delta robot is developed by using the virtual work principle, and the frictional force equation is integrated into the model. The excitation trajectory is generated for the torque control of the Delta robot, and the least mean square algorithm is used to estimate the value of the parameters. The simulation result indicates that the proposed method can estimate the dynamic parameters accurately in real time. Although large errors are observed in certain estimated values in the real-world experiment, the proposed method still provide a stable estimation and it could be used to monitor the status of the Delta robot.

References

[1] Martin L. Leuschen, et al.(2002).”Robotic Fault Detection Using Nonlinear Analytical Redundancy” IEEE International Conference on Robotics & Automation.

[2] Fabrizio Caccavale, et al.(2013). “Discrete-Time Framework For Fault Diagnosis In Robotic Manipulators” IEEE Transections on Control Systems Technology, Vol 21, No 5

[3] Yilin Mi, et al.(2017). ”Fault-Tolerant Control of a 2-Dof Robot Manipulator Using Muti-Sensor Swithing Strategy”

[4] Inseok Hwang, et al.(2010). “A Survey of Fault Detection, Isolation, and Reconfiguration Methods” IEEE Transections on Control Systems Technology, Vol 18, No.3

[5] Jun Wu, Jinsong Wang, Zheng You, “An overview of dynamic parameter identification of robots” Robotics and Computer-Integrated Manufacturing 26(2010) 414-419

[6] Codourey, Alain. "Dynamic modeling of parallel robots for computed-torque control implementation." The International Journal of Robotics Research 17.12

[7] 张铁, et al. "基于牛顿欧拉法的 SCARA 机器人动力学参数辨识." 华南理工大学学报 (自然科学版) 45.10 (2017).

Meng-Kun Liu received his PhD in Mechanical Engi-neering from Texas A&M University in 2012. Since 2014, He has joined the department of mechanical en-gineering in National Taiwan University of Science and Technology as an assistant professor. His research interests include time-frequency analysis of manufacturing process, sensorless fault diagnosis of induction motors, robot force control and chaos control.

Chien Wang received his master degree of mechanical engineering in National Taiwan University of Science and Technology in 2019. His reserch interstes are robot dynamics and system identification. He is now with Taiwan Semiconductor Manufacturing Company.

Chen-yang Lan received the B.S. degree in mechanical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1998, the M.S. in mechanical engineering from Texas A&M University, in 2004, and the Ph.D. degree in mechanical engineering from Texas A&M University, in 2008.

He is currently an assistant professor at the Department of Mechanical Engineering, National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan. His research interests include intelligent control systems, quantitative feedback theory, system identification, condition monitoring, and fault detecting and isolation. His research works appear in International Journal of Robust and Nonlinear Control, IEEE Conference on Decision and Control, American Control Conference and ASME Dynamic Systems and Control Conference.

Diagnosing Alzheimer's disease and Mild Cognitive Impairment Based on Relative EEG Powers

Thanh-Tung Trinh, Yu-Tsung Hsiao, Chieh-Hung Chang, Chia-Fen Tsai, Chien-Te Wu, Yi-Hung Liu

***Abstract*—Alzheimer's is a disease of the brain relating to problems of memory, thinking, attention, and behavior. There are no disease-modifying drugs available for Alzheimer's but early diagnosing may reduce its symptoms and help improve quality of life. In this study, the relative power (RP) of electroencephalography (EEG) is proposed as the feature to determine medical conditions, including patients with Alzheimer's disease (AD), patients with mild cognitive impairment (MCI), and healthy control adults (HC). We also applied Fisher's class separating criterion to determine the best RPs and the best frequency band (FB). Our results showed that using only 8 channels and EEG RP features could achieve a high accuracy of binary (AD vs HC > 90%) or even three classification tasks (AD vs MCI vs HC = 57.90%).**

Index Terms*—*Alzheimer's disease (AD), EEG, fisher criterion, LDA, mild cognitive impairment (MCI), relative power (RP).

I. Introduction

IMPROVEMENTS in health care in the past century have contributed to people living longer and healthier lives. This has also resulted in an increase in the number of people with non-communicable diseases, including dementia. Alzheimer disease is the most common form of dementia. It has become a major public health concern as the world's population ages. Because of increasing in the occurrence of the disease, the diagnosis has attracted the attention of researchers in the field. The higher of age, the higher of probability of disease. This prevalence is just 1% for people between 60 and 64 years, but it raises to 38% for people over 85 years [1]. Mild cognitive impairment (MCI) is the stage between the expected cognitive decline of normal aging and the more serious decline of dementia [2]. In this regard, further research is crucial to identify incipient AD, because patients with MCI get high risk of later developing to AD [3]. Recent studies estimated that the conversion rate from MCI to AD is approximately 15% per year [4]. Despite many years of intensive and effective research, no available treatments that stop or reverse the progression of the disease, which worsens as it progresses, and eventually leads to death. It has become increasingly clear that, if the disease is to be treated successfully, it must be detected as early as possible, perhaps even before symptoms are evident. Thus, there is a great need for reliable diagnostic methods to slow or prevent the disease in proper way [5].

To identify abnormalities in the brain, various techniques have been applied: functional magnetic resonance imaging (fMRI), positron emission tomography (PET), magnetic resonance spectroscopy electroencephalography (EEG), and magnetoencephalography (MEG), and others [6]. While PET was and is better in some situations than fMRI, it has many disadvantages overall. With higher cost, lower temporal resolution, and need for isotopes, the disadvantages of PET seem to outweigh the advantages. However, EEG and MEG are non-invasive techniques with high temporal resolution, allowing for studying the complex dynamical processes in brain systems [7]. EEG equipment is relatively inexpensive compared with other devices and simple to operate. Moreover, it is functionally fast and safe way of checking different areas of brain. It is interesting that EEG has already revealed its advantages to characterize brain dynamics in AD and MCI [7]. EEG is really a promising method in diagnosis Alzheimer's disease as well as other neurological disorders.

Known as spectral analysis, frequency domain analysis is the most conventional yet one of the most powerful and standard methods for EEG analysis. Among all the spectral methods, band power (BP) spectral analysis is commonly used [8]. Besides, the relative power (RP), the power difference between a pair of electrodes, is a noticeable feature. Up-to-date findings showed that the EEG-based three-class classification problem is very challenging [9], which calls to the need to search for more effective EEG markers. The EEG RP features have been successfully applied in depression detection [10], but have never been used in AD detection.

The present study thus aims to using EEG RP as features and Fisher's criterion as feature selection strategy. Linear discriminant analysis (LDA) with leave-one-participant-out cross validation (LOPU-CV) is adopted to confirm classification accuracy in both binary and three-classification tasks. The obtained results allowed us compare and determine the best FB and 8 channels as well as corresponding RPs, which can help us obtain high classification accuracy. This is a reliable suggestion for an 8 channels Quick-cap.

II. Material

A. Subjects

Patients with AD and MCI were diagnosed by a board-certified psychiatrist, whereas HC were elderly subjects without a cognitive impairment and with no history of neurological or psychiatric disorder. There were totally 87 participants took part in this study: 31 AD patients (17 males and 14 females, mean age: 75.1±6.9 y/o, mean education years: 10.9±4.5), 24 MCI patients (10 males and 14 females, mean age: 71.4±7.9 y/o, mean education years: 12.5±3.4), 32 HC subjects (8 males and 24 females, mean age: 66.4±6.6 y/o, mean education years: 13.9±3.5).

This study was conducted at Taipei Veterans General Hospital, Taipei, Taiwan. All participants were fully informed of the aims and methods of the study, and written consent was obtained from all participants.

B. *Diagnostic criteria*

All participants underwent a complete medical interview and neuropsychiatric evaluation. This study used Mini-Mental State Examination (MMSE) and Montreal Cognitive Assessment (MoCA).

C. *EEG recording*

Three minutes of spontaneous EEG activity were recorded using a 32-channel Quick-Cap connected to a 40-channel NuAmps (NeuroScan Inc.). The arrangement of channels was followed the international 10-20 system. Impedance was kept below 10 kOhm and the band-pass filter was set at 0.5-100 Hz with a sampling rate of 500 Hz.

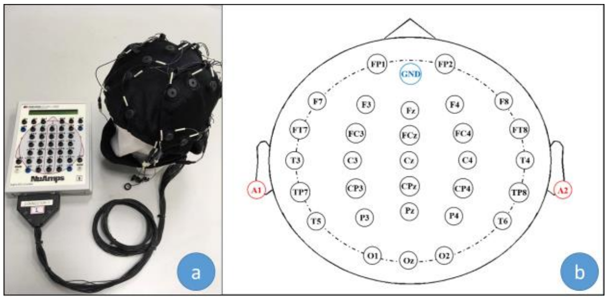


Fig. 1. (a) Quick-Cap; (b) International 10-20 system

Subjects were asked to stay in a relaxed state, awake, and look at a central fixation cross on computer screen during EEG acquisition. During the recording procedure, EEG traces were visually monitored in real time, and muscle activity was identified to avoid high-frequency noise. Ocular artifacts coming from blinking, vertical eye movement or horizontal eye movement were removed through artifact removal software of NeuroScan (Scan4.5). Afterwards, EEG signals were digitally filtered using a Finite Impulse Response (FIR) filter designed with a window between 0.5 and 50 Hz. Additionally, Independent Component Analysis (ICA) also was performed to minimize the presence of noise. Finally, the 20 3-sec epochs were taken out from 180-sec EEG signal.

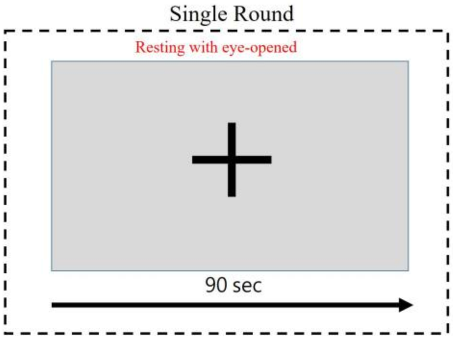


Fig. 2. EEG data recording process

III. METHODS

A diagram of the proposed methodology to discriminate between AD, MCI and HC subjects is presented in Figure 3. After EEG-signal recording and data pre-processing, relative power was computed. Then, fisher criterion was applied and removed some features following the original purpose. Finally, LDA was adopted to check classification accuracy for all task.

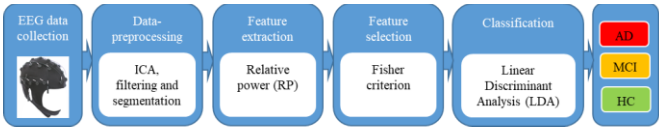


Fig. 3. Schematic diagram of the methodology

A. *Feature Extraction*

In this study, the frequency bands of interest were delta (1–4 Hz), theta (4–8 Hz), alpha (8–13 Hz), beta (13–30 Hz), and gamma (30–45Hz). But each band was divided into sub-bands with smallest length of 2 Hz. Thus, we had totally 260 sub-bands. The spectral power density of each electrode's signal was extracted using Fast Fourier Transform (FFT), and the band power (BP) of each frequency sub-band was then calculated. For each participant and each electrode, we averaged power in each sub-band across 20 EEG epochs. For each participant, a total of 30 BP feature values (from 30 electrodes) in each frequency sub-band were obtained. The BPs were subsequently used to calculate relative power (RP) features.

There are many different types of relative powers used in various EEG studies. In this study, we extracted one type of RP following below calculation:

$$RP = \frac{W(A) - W(B)}{W(A) + W(B)} \qquad (1)$$

where W is the power within a specific sub-band of interest divided by the total power within the entire sub-band. W(A) and W(B) denote the W of two different electrodes A and B in the same frequency sub-band.

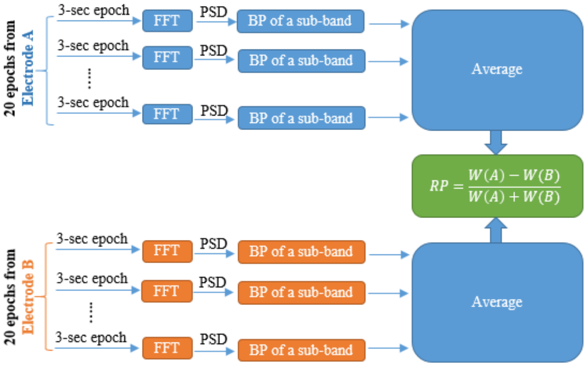


Fig. 4. Illustration of the relative power calculation in a specific frequency band between a pair of electrodes A and B. (FFT = fast Fourier transform; PSD = power spectrum density, and BP = band power).

For each participant, a total of 435 (30x29/2) RP values were extracted for each frequency sub-band. We obtained one D-dimensional feature vector (D=435) from each participant for each frequency sub-band. A feature vector is called a data point in this paper.

B. *Feature Selection*

High-dimensional data in the input space is usually not good for classification due to the curse of dimensionality. It significantly increases the time and space complexity for processing the data. A common way to resolve this problem is feature selection, which reduces the dimensionality by selecting a subset of features from the input feature set. It is often used to reduce the computational cost and remove irrelevant and redundant features for problems with high dimensional data. Fisher score is one of the most widely used supervised feature selection methods. In particular, supposing:

$\mathbf{x}_{\mathrm{ij}} \in R^D$ (D features) is j^{th} data point of i^{th} class (i=1, 2, ...C).

1) *Sample covariance matrix for class i:*

$$\mathrm{S}_i = \frac{1}{n_i}\sum_{i=1}^{n_i}\left(\mathbf{x}_{\mathrm{ij}} - \mathbf{m}_{\mathrm{i}}\right)\left(\mathbf{x}_{\mathrm{ij}} - \mathbf{m}_{\mathrm{i}}\right)^{\mathrm{T}} \quad (2)$$

2) *Within-class scatter:*

$$S_W = \sum_{i=1}^{C} P_i S_i \,, where\ P_i = \frac{n_i}{\sum_{i=1}^{C} n_i} \quad (3)$$

3) *Between-class scatter:*

$$\mathrm{S}_B = \sum_{i=1}^{C} P_i(\mathbf{m}_{\mathrm{i}} - \mathbf{m})\,(\mathbf{m}_{\mathrm{i}} - \mathbf{m})^{\mathrm{T}} \quad (4)$$

4) *For the k^{th} feature:*

$$\text{Fisher Score(k)} = \frac{S_B^{(k)}}{S_W^{(k)}} \quad (5)$$

where $\mathbf{m}_{\mathrm{i}}$ is mean and n_i is number of data in i^{th} class.

After computing the Fisher score for each feature, it selects the top-d ranked features with large scores (d < D).

For purpose of finding 8 best channels, we considered top 28 (d from 1 to 28) features from 435 features because 8 channels can make maximum of 28 (8×7/2) features.

C. Classification

In this study, the LDA classifier was employed to implement binary (AD-HC, AD-MCI, MCI-HC) and three (AD-MCI-HC) classification tasks.

For binary classification task, LDA finds a hyperplane as the decision boundary in the original space of patterns. For a test data point $\mathbf{x} \in R^d$, where d is the dimension of $\mathbf{x}$, the LDA decision function for x is given by the following:

$$D_{LDA}(x) = (\mu_1 - \mu_2)^T \Sigma^{-1} x - \frac{1}{2}(\mu_1 - \mu_2)^T \Sigma^{-1}(\mu_1 + \mu_2) - \ln\left(\frac{C_{12}\pi_2}{C_{21}\pi_1}\right) \quad (6)$$

where$\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are the mean vectors of the training data of the first (positive) and the second (negative) classes, respectively, Σ is the d×d covariance matrix of the training data of the two classes, C_{12} is the penalty weight for the positive class's training error, C_{21} is the penalty for the negative class's training error, andπ_1 and π_2 are the a priori probabilities of the positive and the negative classes, respectively. Here, we set $\mathrm{C}_{12} = \mathrm{C}_{21} = 1$. The test data point x is classified as positive if $\mathrm{D}_{\mathrm{LDA}}(\mathbf{x}) > 0$; otherwise, it belongs to the negative group.

For three-classification task, this study utilized a common method: one-against-one. Solving the three-class problem is considered to handle three binary-class problems.

D. Performance Evaluation

The present study performed the add-one-feature-in strategy [10] and leave-one-participant-out cross validation (LOPO-CV) to assess the classification performance and find the best RP subsets (built by 8 channels) from the 435 RP candidates. LOPO-CV is technique for evaluating how well the results of a method will generalize to unseen data. In each fold of the LOPO-CV, data from participants were used to train the classifier, and then, the d-dimensional data from the one remaining participant were used as the test data. This step was repeated until every participant's data had served as the test data once.

We also analyzed the sensitivity of various EEG epoch lengths to the 3-class classification accuracy.

IV. Results and Discussions

TABLE I
LOPO-CV RESULTS IN 4 CLASSIFICATION TASKS

	AD vs HC	AD vs MCI	MCI vs HC	3-Class
Acc.	90.42%	79.91%	75.52%	57.90%
Selected 8 Channels	CP4; TP7; P4; T5; CZ; CPZ; PZ; OZ	F3; FC3; FC4; C3; CP4; TP8; T5; FZ	F3; FC3; FT8; C3; P3; FZ; FCZ; CZ	CP4; P3; P4; O2; FCZ; CZ; PZ; OZ
Best RP subset	CPZ-PZ; CZ-PZ; P4-CZ; TP7-P4; P4-T5; CP4-CZ; P4-OZ	F3-T5; FC3-C3; T5-FZ; FC3-T5; FC4-FZ; F3-TP8; FC3-CP4	C3-CZ; F3-FZ; F3-FCZ; FC3-FCZ; FT8-P3	CP4-O2; P3-CZ; P3-FCZ; P4-OZ; O2-PZ; CP4-OZ
Best FB	18-29 Hz	13-20 Hz	19-22Hz	15-30 z

TABLE II
ACCURACY IN DIFFERENT EEG EPOCH LENGTHS

Length	2 sec	3 sec	5 sec	8 sec
Acc.	58.03%	57.90%	56.86%	56.89%

Table I shows that the selected 8-channels achieved higher 75.52% accuracies for all binary classification tasks. Particularly, the best AD-HC classification accuracy of 90.42% was obtained by using 8 channels: CP4, TP7, P4, T5, CZ, CPZ, PZ and OZ; the AD-MCI classification accuracy of 79.91% was reached by selecting 8 channels: F3, FC3, FC4, C3, CP4, TP8, T5 and FZ; the MCI-HC classification accuracy of 75.52% was get by choosing 8 channels: F3, FC3, FT8, C3, P3, FZ, FCZ and CZ. A much higher than chance level (33%) accuracy of 57.90% was also obtained for the three-class classification task (AD vs MCI vs HC) by using 8 channels: CP4, P3, P4, O2, FCZ, CZ, PZ, OZ. Interestingly, most of discovered electrodes are in the central and posterior regions.

It should be emphasized that the number of electrodes affects to equipment cost and experimental setting time as well as computational cost. The obtaining results with such small quantities of electrodes will be motivation for thinking about the future application in wearable devices.

On the other hand, the table I also indicates that all of 4 classification tasks (AD-HC, AD-MCI, MCI-HC, AD-MCI-HC) show the best FB belongs to Beta band (18-29 Hz, 13-20 Hz, 19-22 Hz and 15-30 Hz, respectively). In other words, Beta band (13-30 Hz) is the good choice for classifying. This finding is suitable to early studies. Literature shows that changing at this band is considered the first EEG finding in AD [11], [12].

Table II reveals that the RP features were not sensitive to the change of EEG epoch length (2 sec, 3 sec, 5 sec and 8 sec), providing a stable estimate of accuracy (58.03%, 57.90%, 56.86% and 56.89%, respectively). These acceptable results are achieved with employing a sample classifier as LDA. This also prove that RP features are selected based on Fisher's score are promising and trustworthy.

V. Conclusions

This study demonstrated that using only 8 channels and EEG RP features can applied to diagnose Alzheimer's disease with high accuracy. However, the result of 57.9% for 3-class classification task is not really as expected. To overcome this issue, we tried to combine and choose features found from all binary classification tasks. In this way, overfitting may be unavoidable. In future studies, we suggest verifying in a larger sample to validate the findings found here, and more detailed work needs to be done in collecting more comprehensive data about participants.

Acknowledgment

This research was supported by Taipei Veterans General Hospital. We also thank our colleagues from NESS Lab, National Taipei University of Technology who provided insight and expertise that greatly assisted the research.

References

[1] Alzheimer's Association, "Alzheimer's disease facts and figures," *The Journal of the Alzheimer's Association*, vol. 13, pp. 325–373, Apr. 2017.

[2] R. C. Petersen, "Alzheimer's disease: Progress in prediction," *The Lancet. Neurology*, vol. 9, pp. 4-5, Jan. 2010.

[3] E. J. Mufson, L. Binder, S. E. Counts, S. T. DeKosky, L. DeToledo-Morrell, S. D. Ginsberg, M. D. Ikonomovic, S. E. Perez and S. W. Scheff, "Mild cognitive impairment: Pathology and mechanisms," *Acta neuropathologica*, vol. 123, pp. 13-30, Jan. 2012.

[4] C. Davatzikos, P. Bhatt, L. M. Shaw, K. N. Batmanghelich and J. Q. Trojanowski, "Prediction of MCI to AD conversion, via MRI, CSF biomarkers, and pattern classification," *Neurobiology of aging*, vol. 32, pp. 19-27, Dec. 2011.

[5] P. J. Lin and P. J. Neumann, "The economics of mild cognitive impairment," *Alzheimer's & Dementia*, vol. 9, pp. 58–62, Jan. 2013.

[6] M. Ewers, R. A. Sperling, W. E. Klunk, M. W. Weiner and H. Hampel, "Neuroimaging markers for the prediction and early diagnosis of Alzheimer's disease dementia," *Trends in neurosciences*, vol. 34, pp. 430-442, Aug. 2011.

[7] J. Poza, C. Gómez, M. García, R. Corralejo, A. Fernández and R. Hornero, "Analysis of neural dynamics in mild cognitive impairment and Alzheimer's disease using wavelet turbulence," *Journal of neural engineering*, vol. 11, Apr. 2014.

[8] M. Baker, K. Akrofi, R. Schiffer and M. W. O. Boyle, "EEG Patterns in Mild Cognitive Impairment (MCI) Patients," *The open neuroimaging journal*, vol. 2, pp. 52-55, Aug. 2008.

[9] S. Ruiz-Gomez, C. Gomez, J. Poza, G. Gutirrez-Tobal, M. Tola-Arribas, M. Cano and R. Hornero, "Automated multiclass classification of spontaneous EEG activity in Alzheimer's disease and mild cognitive impairment," *Entropy*, vol. 20, pp. 35, Jan. 2018.

[10] C. T. Wu, D. G. Dillon, H. C. Hsu, S. Huang, E. Barrick and Y. H. Liu, "Depression detection using relative EEG power induced by emotionally positive images and a conformal kernel support vector machine," *Applied Sciences*, vol. 8, pp. 1244, Jul. 2018.

[11] R. Ihl, T. Dierks, L. Froelich, E. M. Martin, K. Maurer, "Segmentation of the spontaneous EEG in dementia of the Alzheimer type", *Neuropsychobiology*, vol. 27, pp. 231-236, Dec. 1993.

[12] J. Jeong, "EEG dynamics in patients with Alzheimer's disease", *Clinical neurophysiology*, vol. 115, pp. 1490-1505, Dec. 2004.

Design and Control of an Upper-Limb Exoskeleton Robot with Visual Sensing

Chun-Hsu Ko, Shu-Ling Cheng, Kuu-young Young, Jian-Bin Huang, I-Yu Lin, and Shang-Yuan Young

***Abstract*—As the aging society is approaching, people with limb immobility are increasing. Motivated by it, we develop a wearable two-DOF upper-limb exoskeleton robot, named as HAMEXO. It is designed to be fitting and comfortable for the user. Meanwhile, to attain the goal of providing motion assistance in daily life, we equip HAMEXO with a visual system to detect the locations and movements of the objects in the environment, and also a motion controller for its governing. For dealing with the coupling between the two joints during movements and adapting to different users, we propose using the learning mechanism for controller design. The experimental results demonstrate that the proposed system can effectively assist the user to accomplish the task in daily life, such as object- fetching.**

***Index Terms*—upper-limb exoskeleton robot; mechanical design; visual sensing; motion assistance; learning mechanism.**

I. Introduction

As the number of elderly population and stroke patients increases, medical staffs are highly demanded for providing assistance in their daily lives. It solicits the use of robots to alleviate the workloads from the caregivers. Among them, the exoskeleton robot has been of a focus for its closeness with the user [1–12]. Compared to the traditional robot that distances itself from the person in use, the exoskeleton robot is designed to be worn on human body for acting with the user. As the exoskeleton robot is highly interacting with the human, its design [1-6] and control [7] [8] that allow proper coordination between them is very imperative.

The exoskeleton robots can be mainly classified into upper-body, lower-body, and full-body types [1–6]. In previous research, they have been applied for rehabilitation, daily activities, and others, including BLEEX [1], HAL [2], ULERD [3], NTUH-ARM [4], ReWalk [5], TTL-Exo [6], etc. According to the given applications, they may be fixed to a base or wheelchair [9], heavy for stability or light for portability, etc. In their use for assistance, the exoskeleton robot can be operated mainly in three styles: passive, active-assisted, or active-resistive [4,10]. The passive style allows the robot to dictate the entire motion without any involvement from the user. For the active-assisted or active-resisted style, the robot usually takes a supporting role in assisting the user for task execution. Consequently, it is crucial for the robot to realize the intention of the user for providing proper assistive force. For this purpose, electromyography (EMG) and electroencephalography (EEG) measured from the user, among others [11], are frequently used for detecting user's motion intention.

Continuing from previous research [12][13], our laboratory has developed a wearable 2-DOF upper-limb exoskeleton robot, HAMEXO. As it is intended for daily activity, we equip it with a visual system to detect the object in the working environment for those tasks, such as object picking and drinking. With the object identified, a motion controller is then developed for task execution. Due to the coupling present between the two joints during motion and the demand for adapting to various users, we adopted the adaptive network-based fuzzy inference system (ANFIS) for controller design due to its excellence at adaptation [14]. In summary, the contribution of this paper lies on:

- Design of a two-DOF upper-limb exoskeleton robot that is comfort and fitting for the user;
- Development of a visual sensing system for detecting objects in the working environment;
- Development of an intelligent controller that can be adapted to various users.

In this stage, we mainly focus on the passive style of assistance. For demonstration, the proposed system is applied for experiments based on the task of object fetching.

II. Design and Development for HAMEXO

Based on the discussions above, proper design of the exoskeleton robot is very crucial, and should fit its application. As it is mainly intended for providing the freedoms for object-picking types of tasks, HAMEXO (abbreviated for Human and Machine Exoskeleton) is developed to be a 2-DOF upper-body exoskeleton robot, with the two DOFs used for the flexion and extension of the shoulder (θ_1) and elbow (θ_2). According to the motion range of human upper-body anatomy [15] [16], the ranges of θ_1 and θ_2 are arranged to be

$$0 \le \theta_1 \le \pi/2,\quad 0 \le \theta_2 \le 3\pi/4 \tag{1}$$

Chun-Hsu Ko is with the Department of Electrical Engineering, I-Shou University, Kaohsiung, Taiwan (corresponding author, e-mail: chko@isu.edu.tw).

Shu-Ling Cheng is with the Department of Multimedia and Game Developing Management, Far East University, Tainan, Taiwan.

Kuu-young Young, Jian-Bin Huang, I-Yu Lin, and Shang-Yuan Young are with the Department of Electrical Engineering, National Chiao Tung University, Hsinchu, Taiwan.

This work was supported in part by the Ministry of Science and Technology, Taiwan under grant MOST 102-2221-E-009-138-MY3.

Fig. 1 shows the CAD modeling for HAMEXO. The polylactide 3D printed platform is attached with a strap belt for each of the two links to secure user's arm to the exoskeleton. Sliding parts are equipped on the upper-arm, forearm, shoulder, and backpack, respectively, to accommodate for different human body sizes. The brushless DC motors, along with reduction gears and incremental encoders for position feedback, were adopted as the actuators [17]. In addition, the hard foam was used as padding between the user and exoskeleton for comfort and the power-kill switch for safety [18]. For illustration, Fig. 2 shows the photo with the user wearing HAMEXO.

In its use, we found that HAMXO might lean to one side, as it is worn only on one arm of the user. This is a major reason that we have included a backpack for it, as shown in Fig. 3, which is modeled after human body and closely attached to the back of the user [19]. This backpack provides the support in sharing the weight of HAMEXO. As the effort for weight reduction, we managed to make the weight of HAMEXO be about 4.9 kg, and that of the backpack be 2.2 kg.

To evaluate whether HAMEXO is fitting and comfortable for use in daily activity, we invited 10 university students, 8 males and 2 females, for testing, with their ages ranging between 22~26 years old, heights between 162~176 cm, and weights between 50~70 kg, respectively. They were asked to wear HAMEXO to move their arms to a number of locations. After the trial, we administered a questionnaire to obtain subjects' evaluations on the aspects of fixity, comfort, mobility, and lightness, with the scores shown in Fig. 4 (1~5, with 5 as the highest). From the evaluation, the subjects highly agreed that HAMEXO was closely fixed on his/her arm via the assistance of the backpack, as indicated by a score of 4.1. The score of 3.7 for comfort implicates that the wearing of HAMEXO was not so disturbing, while some subjects reported that they still felt an unbalancing weight distribution. Although there were only 2 DOFs available for HAMEXO, its motion feasibility in the given direction was acceptable, as indicated by a score of 3.7 for mobility. Meanwhile, the score of 2.3 for lightness shows that it was still too heavy for a long-time use. Further interview on the subjects also discloses that it was easy to use, as most of them felt the system was quite responding and its operation was intuitive. Some subjects suggested to use HAMEXO for rehabilitation.

III. MOTION GOVERNING SYSTEM

The proposed motion governing system mainly consists of a visual system and an ANFIS- PID controller. Since the main purpose of the system is to assist the user to accomplish the task in daily life, like object-fetching, the visual system is first used to detect the location of the object. Based on the visual information, a smooth trajectory for the task is generated via a motion planner previously developed and the ANFIS PID

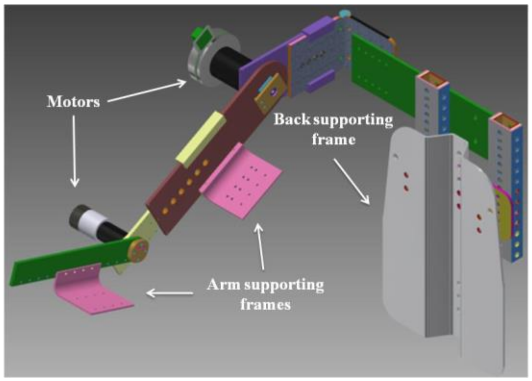


Figure 1. CAD modeling of HAMEXO.

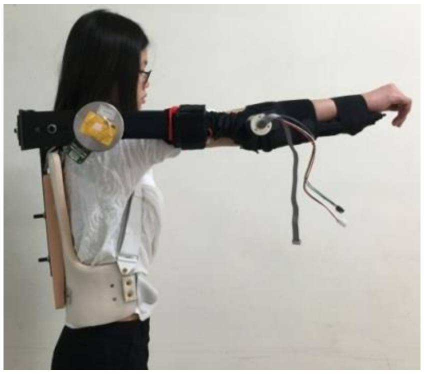


Figure 2. The photo with the user wearing HAMEXO.

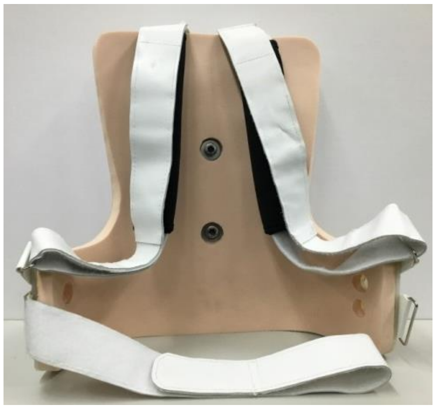


Figure 3. The backpack for holding HAMEXO.

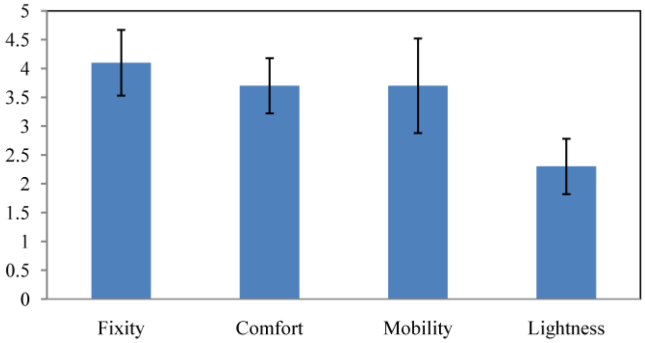


Figure 4. User evaluation on HAMEXO.

controller is then applied to govern the motion of HAMEXO. The proposed visual system and ANFIS PID controller are described below.

A. Visual System

The proposed visual system uses two cameras to locate the objects in the workspace, as shown in Fig. 5 [13]. The calibration procedure has been conducted for the two cameras to derive accurate parameters. With the visual system, the 3D object location can then be deduced with the 2D images captured from the cameras.

In the pinhole camera model commonly used [20], the relation between a 3D point $[x_c \quad y_c \quad z_c]^T$ in camera coordinates and its corresponding 2D projection $[u \quad v]^T$ on the image plane can be described as

$$sm = AX_C \tag{2}$$

with

$$m = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, A = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \text{and } X_C = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \tag{3}$$

where s is a scaling factor and A is the camera intrinsic matrix. In A, u_0 and v_0 are the coordinates of the principal point, and f_x and f_y the scaling factors in u and v directions, respectively. These four parameters can be easily obtained by applying the calibration procedure provided by the camera calibration toolbox [21]. From X_C in 3D camera coordinates, the corresponding $X_W = [x_w \quad y_w \quad z_w]^T$ in world coordinates can be obtained via rotation and translation:

$$X_W = RX_C + T \tag{4}$$

where R is the rotation matrix and T the translation vector. From Eqs. (2) and (4), we can obtain

$$X_W = sRA^{-1}m + T \tag{5}$$

with

$$A^{-1} = \begin{bmatrix} 1/f_x & 0 & -u_0/f_x \\ 0 & 1/f_y & -v_0/f_y \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

We use the two cameras shown in Figure 4 to find X_W for an object. Letting the rotational matrix and translational vector for cameras 1 and 2 be $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ and T_1 and T_2, respectively, the two corresponding X_{W1} and X_{W2} in world coordinates for them can then be derived as

$$X_{W1} = s_1\vec{v}_1 + T_1 \tag{7}$$

$$X_{W2} = s_2\vec{v}_2 + T_2 \tag{8}$$

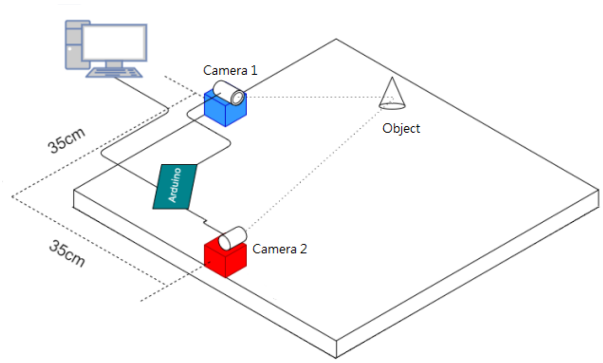


Figure 5. The visual system for object detection

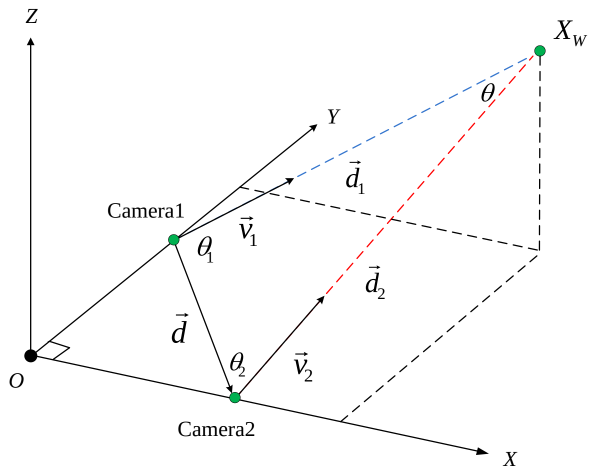


Figure 6. Deployment of the two cameras for detecting the object in the workspace.

where

$$\vec{v}_1 = R_1A_1^{-1}m_1 \tag{9}$$

$$\vec{v}_2 = R_2A_2^{-1}m_2 \tag{10}$$

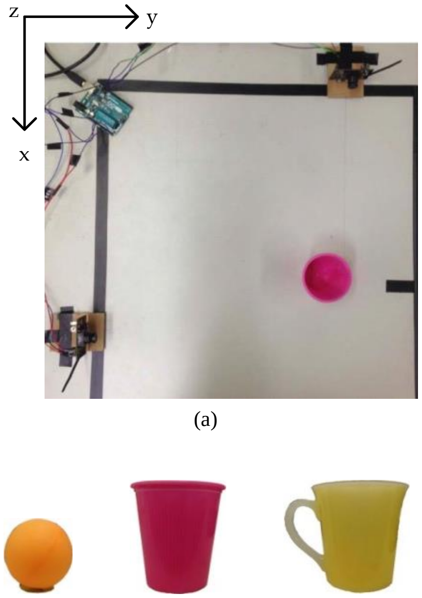
To find the scaling factors s_1 and s_2 in Eqs. (7)-(8), we first calculate lengths $\|\vec{d}_1\|$ and $\|\vec{d}_2\|$ for the triangle in Figure 5 as

$$\|\vec{d}_1\| = \frac{\|\vec{d}\|\sin\theta_2}{\sin\theta}, \quad \|\vec{d}_2\| = \frac{\|\vec{d}\|\sin\theta_1}{\sin\theta} \tag{11}$$

With

$$\theta_1 = \cos^{-1}\frac{\vec{v}_1 \cdot \vec{d}}{\|\vec{v}_1\|\|\vec{d}\|}, \quad \theta_2 = \cos^{-1}\frac{\vec{v}_2 \cdot (-\vec{d})}{\|\vec{v}_2\|\|\vec{d}\|}, \quad \theta = \pi - \theta_1 - \theta_2 \tag{12}$$

where $\vec{d}$ in Figure 5 is the vector from the location of camera 1 to that of camera 2. s_1 and s_2 can be calculated as



(a)

(b)

Figure 7. (a) The setup for accuracy evaluation for the visual system and (b) the three objects.

TABLE I
AVERAGE ERRORS FOR THE THREE OBJECTS

Object	*x*(cm)	*y*(cm)	*z*(cm)
Ping-pong ball	0.40	0.45	0.30
Cup	0.35	0.37	0.32
Cup with a handle	0.27	0.33	0.34

$$s_1 = \frac{\left\|\vec{d}_1\right\|}{\left\|\vec{v}_1\right\|},\ s_2 = \frac{\left\|\vec{d}_2\right\|}{\left\|\vec{v}_2\right\|} \tag{13}$$

Consequently, X_{W1} and X_{W2} can be calculated by using Eqs. (7)-(8), and X_W is then taken as their average.

We conducted a series of experiments to evaluate the accuracy of the visual system. As the setup shown in Fig. 7(a), the visual system was used to locate the positions of the three objects randomly placed in the workspace, which include a ping-pong ball, a cup, and a cup with a handle (shown in Fig. 7(b)), with their width and height as (1.9 cm, 2.2 cm), (2.5 cm, 3.5 cm), and (3.5 cm, 4.7 cm), respectively. The process was repeatedly 15 times for each object, with their detected and actual positions recorded. Table 1 lists the average errors of the three objects in *x*, *y*, and *z* directions, and they are all within 0.5 cm, indicating this visual system is accurate enough for our targeted tasks.

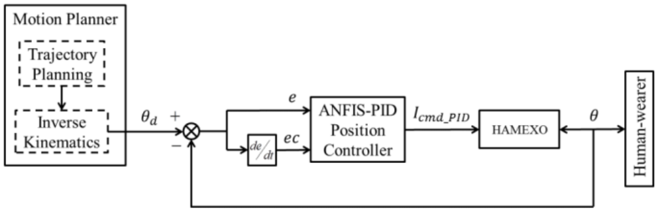


Figure 8. The proposed motion controller based on the ANFIS.

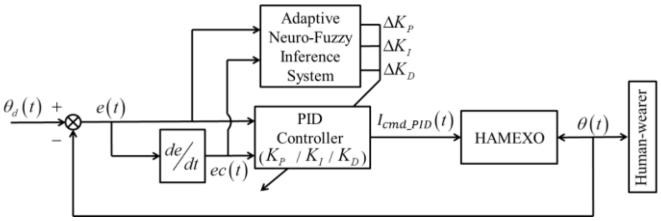


Figure 9. The proposed ANFIS-PID position controller.

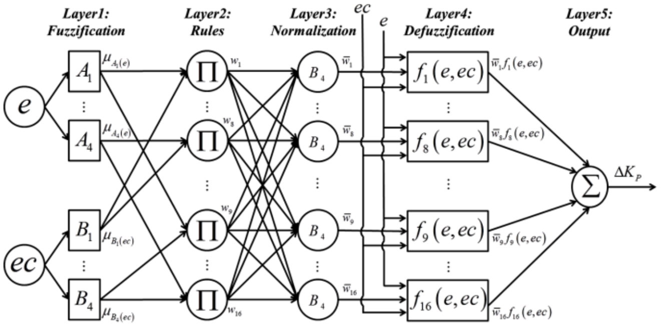


Figure 10. The ANFIS architecture for deriving ΔK_p.

B. ANFIS-PID Controller

With the relative distance between the object and HAMEXO identified from the visual system, it will be forwarded to the proposed ANFIS-based motion controller, shown in Fig. 8, to move HAMEXO. In Fig. 8, the motion planner first generates a path (θ_d). The B-spline method is employed to make the path smooth. The planned path (θ_d) is then sent to the ANFIS-PID position controller, which determines proper control commands in current (I_{cmd_PID}) based on position error (e) and position error rate (ec), formulated as

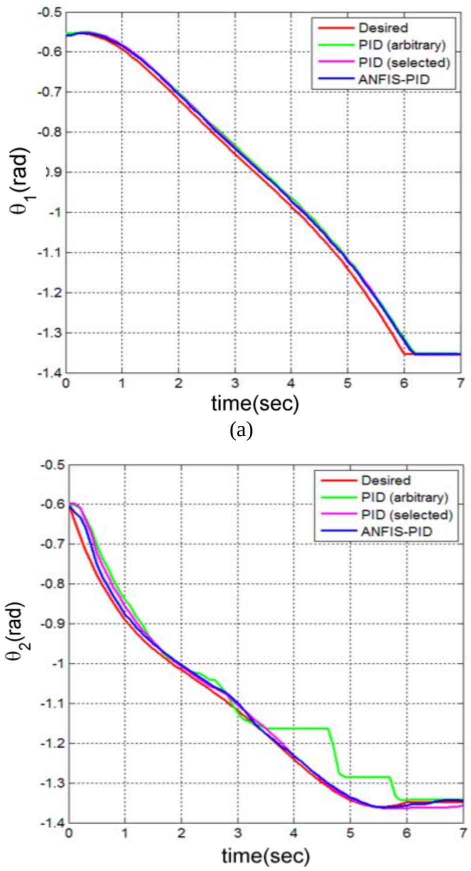
$$I_{cmd\_PID}(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \dot{e}(t) \tag{14}$$

The commands shall then drive the motors for moving HAMEXO to follow the desired path (θ_d).

Fig. 9 shows how the proposed ANFIS-PID position controller operates, which is equipped on each link of HAMEXO. It was designed as a PID controller with adjustable gains K_P, K_I, K_D to be tuned by the ANFIS. Fig. 10 shows the ANFIS architecture for deriving ΔK_P as an illustration. Those for ΔK_I, and ΔK_D can be derived similarly. This ANFIS is designed to consist of five layers, including fuzzification, rule, normalization, defuzzification, and output layers, with the nodes in layers 1 and 4 adaptive. In searching for proper parameters, the cost function was designed to minimize the tracking error and the back-propagation gradient descent method employed for learning. Details for its implementation can be referred to [12].

IV. EXPERIMENT

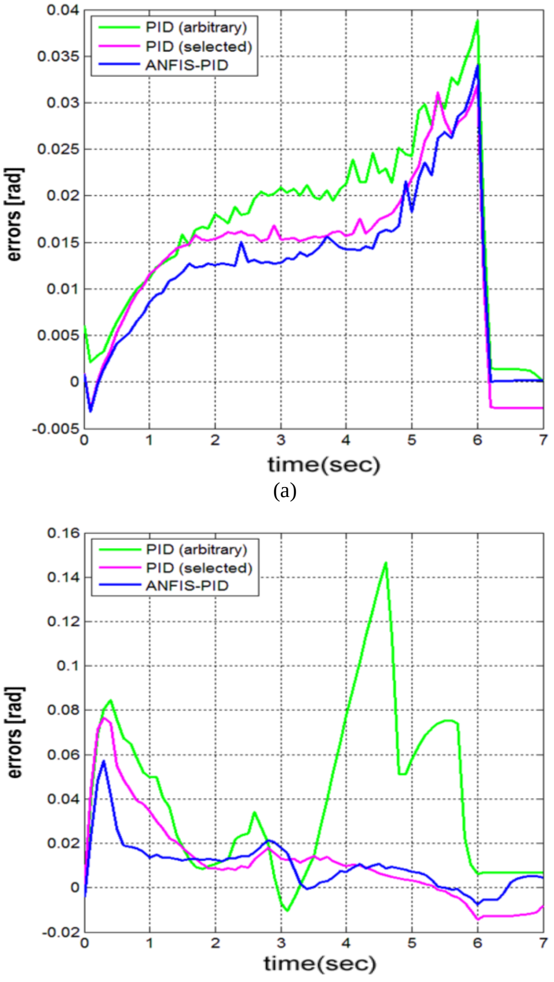
We first conducted experiments for the performance of the ANFIS-PID controller in the presence of disturbances, such as friction, gravity, nonlinearity in system dynamics, and uncertainty in parameter identification. During the experiments, the subject was asked to wear HAMEXO, and let it carry their arm to follow a straight or curved path. For comparison, we also conducted the experiments by using the PID controller of fixed gains. Two sets of fixed gains were used, one arbitrarily chosen and the other one selected via a trial-and-error process for achieving acceptable performance. Figs. 11(a)–(b) shows the resultant joint trajectories for the shoulder and elbow, and Figs. 12(a)–(b) shows the errors during path following. The errors along the path were computed in the root mean square (RMS) manner, as listed in Table II. From the results, both the proposed ANFIS-PID controller and PID controller with selected gains outperformed the PID controller with arbitrary gains, while the proposed one led to smaller errors. Meanwhile, the proposed ANFIS-PID controller was deemed to be more efficient, as the trial-and-error process in searching proper gains for the PID controller was very time consuming.



(a)

(b)

Figure 11. Experimental results for evaluating the performance of the ANFIS-PID controller: (a) joint trajectories for the shoulder and (b) joint trajectories for the elbow.



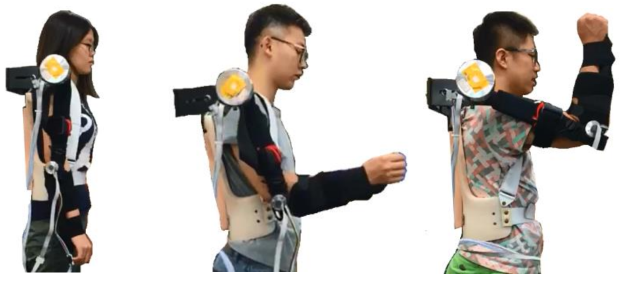
(a)

(b)

Figure 12. Experimental results for evaluating the performance of the ANFIS-PID controller: (a) errors for the shoulder and (b) errors for the elbow.

TABLE II
ERRORS IN RMS DURING PATH FOLLOWING

	Shoulder (rad)	Elbow (rad)
ANFIS-PID	0.0148	0.0151
PID (selected)	0.0163	0.0234
PID (arbitrary)	0.0194	0.0552

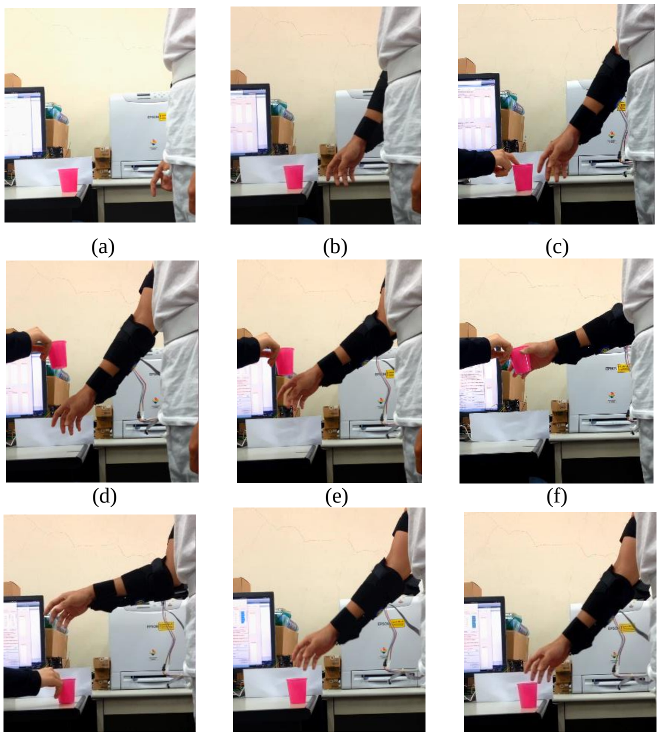


(a) (b) (c)

Figure 13. Photos of the subjects invited for the experiments. (a) subject A, (b) subject B, and (c) subject C.

With the effectiveness of the proposed ANFIS-PID controller verified, we went on to evaluate how the system performed when it was combined with the visual system for the task of object fetching. We invited three subjects to conduct the experiment, with their photos shown in Fig. 13. The maximum motor speeds for the shoulder and elbow were set to be 300 and 250 rpm, respectively, for safety consideration. During the experiments, the subjects needed to fetch a cup, which would appear in an arbitrary manner, i.e., the subjects did not know where it was located in advance.

Figs. 14(a)-(i) show the experimental setup, in which the cup on the desk was first reached by the subject at the first location (shown in Fig. 14(c)), raised up to the air to the second location (shown in Fig. 14(f)), and then put down to the desk at the third location (shown in Fig. 14(i)). Table III lists the joint angles of



(a) (b) (c)

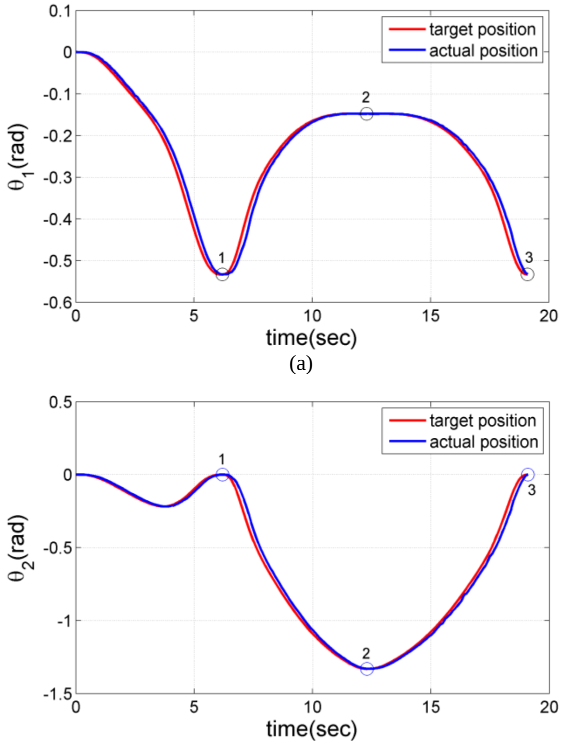
(d) (e) (f)

(g) (h) (i)

Figure 14. The setup for the experiment of object fetching: (a)-(c) the actions for reaching the cup, (d)-(f) the actions for raising the cup, and (g)-(i) the actions for putting down the cup.

TABLE III
HAMEXO JOINT ANGLES CORRESPONDING TO THE THREE LOCATIONS

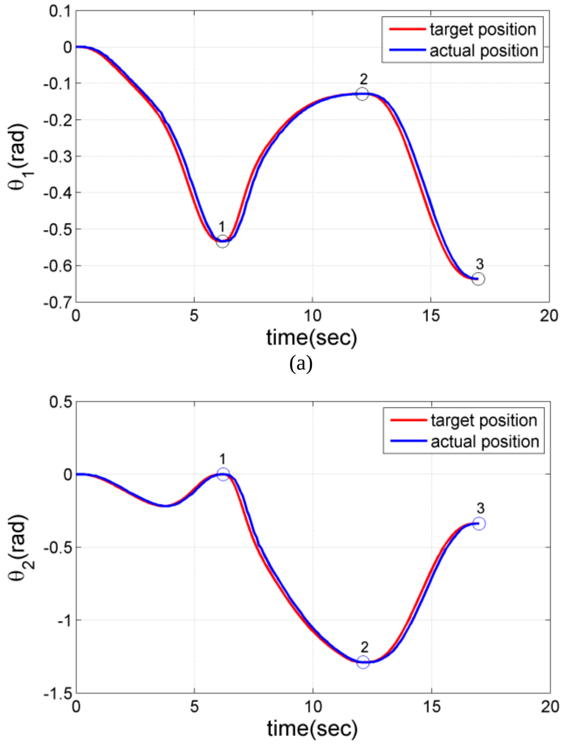
Subject	Location	Shoulder (rad)	Elbow (rad)
A	1st	-0.5331	0.0003
	2nd	-0.1480	-1.3302
	3rd	-0.5327	-0.0001
B	1st	-0.5331	0.0007
	2nd	-0.1286	-1.2891
	3rd	-0.6362	-0.3369
C	1st	-0.5332	-0.0003
	2nd	-0.5430	-0.3707
	3rd	-0.3592	-1.0365



(a)

(b)

Figure 15. Experimental results for evaluating the performance of the motion governing system (subject A): (a) joint trajectory for the shoulder r (θ_1) and (b) joint trajectory for the elbow(θ_2).



(a)

(b)

Figure 16. Experimental results for evaluating the performance of the motion governing system (subject B): (a) joint trajectory for the shoulder (θ_1) and (b) joint trajectory for the elbow (θ_2).

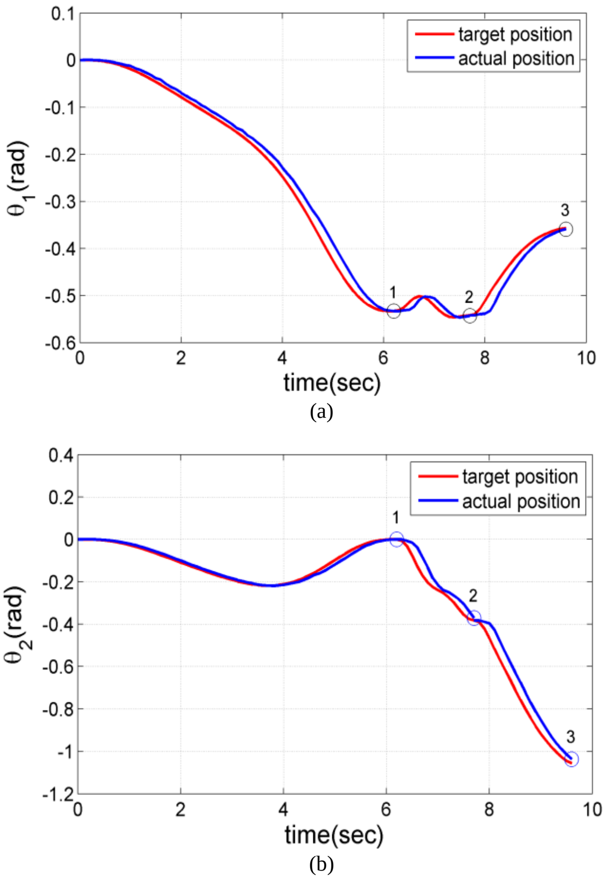


Figure 17. Experimental results for evaluating the performance of the motion governing system (subject C): (a) joint trajectory for the shoulder (θ_1) and (b) joint trajectory for the elbow (θ_2).

the shoulder and elbow of HAMEXO corresponding to the three locations for subjects A, B, and C, respectively. As a demonstration, Figs. 15-17 show the angles of both the shoulder and elbow joints during the motion to fetch a cup for the three subjects, in which the round circles 1, 2, and 3 represent the three sequential change locations, the red line the trajectory designed by the motion planner, and the blue line the actual trajectory executed by HAMEXO. In these figures, the actual joint trajectories followed the planned ones quite well, and all three target locations were reached, indicating the effectiveness of the proposed system.

V. Conclusions

In this paper, we have developed a wearable two-DOF upper-limb exoskeleton robot, HAMEXO, which is designed to be comfort and fitting for the user when wearing. For its motion governing, we have developed a visual and control system. The system aims to accurately detect and track the targeted object and be adapted to various users. The experimental results demonstrate its effectiveness in successfully assisting the user to accomplish the object-fetching task. As one of our future works, we intend to conduct a large-scale user's experience evaluation to assess its performance via the means of the interview and survey as the feedback. The results will be analyzed for system enhancement. We will also upgrade the ability of the visual system and motion controller, so that HAMEXO can deal with more different types of daily tasks.

References

[1] A. Chu, H. Kazerooni, and A. Zoss, "On the biomimetic design of the Berkeley lower extremity exoskeleton (BLEEX)," *IEEE International Conference on Robotics and Automation*, pp. 4345-4352, 2005.

[2] H. Satoh, T. Kawabata, and Y. Sankai, "Bathing care assistance with robot suit HAL," *IEEE International Conference on Robotics and Biomimetic*, pp. 498-503, 2009.

[3] W. Wei, W. Zhang, S. Guo, X. Zhao, and Y. Wang, "Development of an upper limb rehabilitation robot system for bilateral training," *IEEE International Conference on Mechatronics and Automation*, pp. 930-935, 2014.

[4] W. W. Wang, B. C. Tsai, L. C. Hsu, L. C. Fu, and, J. S. Lai, "Guidance-control-based exoskeleton rehabilitation robot for upper limbs: application to circle drawing for physiotherapy and training," *Journal of Medical and Biological Engineering*, vol. 34, pp. 284–292, 2014.

[5] M. Lancini, M. Serpelloni, and S. Pasinetti, "Instrumented crutches to measure the internal forces acting on upper limbs in powered exoskeleton users," *IEEE International Workshop on Advances in Sensors and Interfaces*, pp. 175-180, 2015.

[6] B. Ugurlu, M. Nishimura, K. Hyodo, M. Kawanishi, and T. Narikiyo, "Proof of concept for robot-aided upper limb rehabilitation using disturbance observers," *IEEE Transactions on Human-Machine Systems*, vol. 45, pp. 110–118, 2015.

[7] H. Seo and S. Lee, "Design of general-purpose assistive exoskeleton robot controller for upper limbs," *Journal of Mechanical Science and Technology*, vol. 55, no. 7, pp.3509-3519, 2019.

[8] M. Deng, Z. Li, Y. Kang, C. L. P. Chen, and X. Chu, "A learning-based hierarchical control scheme for an exoskeleton robot in human-robot cooperative manipulation," *IEEE Transactions on Cybernetics*, vol. 50, no. 1, pp.112-125, 2020.

[9] A. J. Huete, J.G. Victores, S. Martinez, A. Gimenez, and C. Balaguer, "Personal autonomy rehabilitation in home environments by a portable assistive robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, pp. 561–570, 2012.

[10] Z. Song, Z. Wang, S. Guo, and B. Gao, "Study on resistance training for upper-limb rehabilitation using an exoskeleton device," *IEEE International Conference on Mechatronics and Automation*, pp. 932-938, 2013.

[11] H. Liu, J. Tao, P. Lyu, and F. Tian, "Human-robot cooperative control based on sEMG for the upper limb exoskeleton robot," *Robotics and Autonomous Systems*, vol. 125, 2020.

[12] J. B. Huang, K.Y. Young, and C. H. Ko, "Effective control for an upper-extremity exoskeleton robot using ANFIS," *IEEE International Conference on System Science and Engineering*, 2016.

[13] J. B. Huang, I Y. Lin, K.Y. Young, and C. H. Ko, "Motion and visual control for an upper-limb exoskeleton robot via learning," *International Symposium on Neural Networks*, 2017.

[14] J.-S. R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, pp. 665-685, 1993.

[15] J. C. Perry, J. Rosen, and S. Burns, "Upper-limb powered exoskeleton design," *IEEE/ASME Transactions on Mechatronics*, vol. 12, pp. 408-417, 2007.

[16] J. Karner, W. Reichenfelser, and M. Gfoehler, "Kinematic and kinetic analysis of human motion as design input for an upper extremity bracing system," *IASTED International Conference on Biomedical Engineering*, pp. 376–383, 2012.

[17] J. Choo, D. H. Jeong, S. Jeong, G. Chu, and J. H. Park, "Oscillation reduction scheme for wearable robots employing linear actuators and sensors," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2160-2165, 2014.

[18] I. Koo, C. Yun, M. V. Costa, and J. V. Scognamiglio, "Development of a meal assistive exoskeleton made of soft materials for polymyositis patients," *IEEE International Conference on Intelligent Robots and Systems*, pp. 542-547, 2014.

[19] Y. Muramatsu, H. Umehara, and H. Kobayashi, "Improvement and quantitative performance estimation of the back support muscle suit," *International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2844-2849, 2013.

[20] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd ed. New York: Cambridge University Press , 2003.

[21] Bouguet's Camera Calibration Toolbox for Matlab. Available online: http://www.vision.caltech.edu/bouguetj/calib_doc/

Implementation of Smart Vision-Based Handling Gripper

Kuo-Ho Su, *Member, RST,* and Chung-Hsien Kuo, *Member, RST*

***Abstract*—The front end of the robot arm has been one focus of technology development. Once the gripping objects are random patterns, the damage may be caused due to the fixed traditional jaw gripping force. In this research, a smart neural network model is developed via backpropagation algorithm to recognize the object. The whole identification procedure can be divided into four steps. Step 1: Use the camera module of the Raspberry Pi microcontroller to capture the image of all possible objects. Then the invariant moments are calculated and the neural network model is established. Finally, the model is embedded into the memory of microcontroller. Step 2: When the gripper is working, the Raspberry Pi microcontroller trigs the camera to capture the image at fixed distance according to the ultrasonic sensor. Step 3: The Raspberry Pi and the Open CV are used to pre-process and calculate the characteristic values. Step 4: The Raspberry Pi identifies the image via the embedded neural network model. To verify its feasibility, some simulation and experiments are carried out and a gripper prototype with vehicle robot platform is implemented in this research. From the results and the prototype performance, the proposed hardware architecture and neural network identification method possess good recognition.**

***Index Terms*—Backpropagation algorithm, Handling gripper, Image recognition, Invariant moments, Neural network model, Raspberry Pi microcontroller**

I. Introduction

THE combination of image recognition and artificial intelligence has always been the goal of researchers [1]. One of the most common applications is the identification of license plate in the traffic violation, flow guidance, residential vehicle management, electronic toll collection management, etc. In 2012, a license plate location method was proposed by Song et al. [2], in which a mathematical morphology method is based and a modified radial base function neural network algorithm is used to complete the identification. To conform to the standard of ISO/TC104 in identification formats and styles, a fast method to extract the recognition from freight containers was proposed by Fahn et al. [3]. Further to improve the accuracy of pattern recognition under low illumination environment, a remarkable texture feature from license plate area was employed to find the approximate area of the license plate [4]. On the other hand, because the collection of extracted features from the image can define the character uniquely, the selection of relevant feature extraction method was the most important factor in achieving high identification performance [5]. To deal with the movement of vehicle license extraction from complex environment, a template matching was used to process the license plate image, location of the license plate area, character segmentation and identification in [6]. The neural network with a lot of neurons, its structure is similar to the human brain with a deduced mathematical leaning algorithm. A robust end-to-end convolutional neural network was proposed to simultaneously detect and classify the traffic-signs [7]. In 2017, a deep neural network was proposed for structural prediction and lane detection in traffic scene by Li et al. [8]. Nowadays, the spatial features of hyperspectral imagery (HSI) have gained attention, an HSI reconstruction model based on deep convolutional neural network was proposed to enhance spatial features [9]. In 2019, a deep neural network with polynomial activations [10] was proposed to improve the classification accuracy and training time.

Grasping task in the robot industry is common and important. However, in a simple environment, as well as the parameters of the object to grasp for known circumstances, you can construct a mathematical model for its features to reach grasping task. However, after leaving the pure environment faced many unknown objects, we will need to obtain a contact or non-contact of the parameters. Non-contact objects commonly used its image [11] to measure the length, width, area, center of mass, the degree of bend, even three-dimensional model can be reconstructed, and the data is provided to grasping task a suitable location for crawling [12-14] , but the vision is still developing stage, in a complex environment image will be all kinds of interference by environmental factors, such as shadow, reflection, background color interference … , so the image will not be able to provide a very effective and accurate data. The contact object is to use pressure sensors, tactile sensors for the main, the former is composed of a piezoresistive material, and the latter is composed of a piezoelectric material. Piezoresistive material reaches the steady state is slower, but is applicable to stationary power moments, and the piezoelectric material to reach steady-state response is faster, but the principle will not apply force static timing. Both can be used to surveillance whether to touch the items [15-17]. Through proper detection can discernible its body, force feedback, and other useful information to crawl task. Due to the non-contact and contact-type has its own advantages and disadvantages, so there have been a mixed type of architecture, or the multi-sensor architecture, trying to crawl task could be more perfect [16, 18]. Whether contact or non-contact sensors are useful feedback information to the controller then referred to the actuator, the controller generally divided into three categories PID control, fuzzy control, and hybrid control is the most taken method [19-26].

This research supported by the Ministry of Science and Technology of Taiwan, R.O.C. through its grant MOST 107-2622-E-034-002 -CC3.

Kuo-Ho Su is with the Graduate Institute of Digital Mechatronic Technology, Chinese Culture University, Taipei, Taiwan (e-mail: sgh@faculty.pccu.edu.tw)

Chung-Hsien Kuo is with the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan (e-mail: chkuo@gmail.ntust.edu.tw)

II. Neural-Network-Based Image Recognition

The pixel values of the sample patterns are found and set as the P-value of the input layer, and then the represented digital values of the sample patterns are set as the T-value of the output layer. Then a 3-layer simple back-propagation neural network architecture is used. It contains input, hidden and output layers. The error function is defined as

$$E = \frac{1}{2}\sum_{j}(T_j - O_j)^2 \tag{1}$$

where T_j and O_j are the target output and the actual output of neuron *j*, respectively. The adjusting algorithm is derived as:

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial O_j^n}\frac{\partial O_j^n}{\partial net_j^n}\frac{\partial net_j^n}{\partial W_{ij}} \tag{2}$$

where net_j^n is the net input of the j^{th} neuron at n^{th} layer.

From the settings of training function, adaptive learning function, performance function, input layer and output layer, momentum coefficient, number of layers, learning rate α, learning rate increasing factor, learning rate decreasing factor, maximum performance increasing value and number of neurons, a heuristic neural network is developed. The proposed adaptive momentum gradient descent (AMGD) adjusting algorithms for weighting and bias values are designed as

$$\Delta W^m(k) = \gamma\Delta W^m(k-1) - (1-\gamma)\alpha s^m(a^{m-1})^T \tag{3}$$

$$\Delta b^m(k) = \gamma\Delta b^m(k-1) - (1-\gamma)\alpha s^m \tag{4}$$

where *m* is the number of layer, *s* is the sensitivity, γ is the momentum coefficient, α is the learning rate and *k* is the number of iteration. In order to achieve rapid convergence and increase the identification capability, the learning rate α of the neural network is modulated according to the classification error of previous iterations are calculated and the NNTOOL of Matlab is used to establish the neural network model, which is trained via backpropagation algorithm. Finally, the model is embedded into the Raspberry Pi microcontroller.

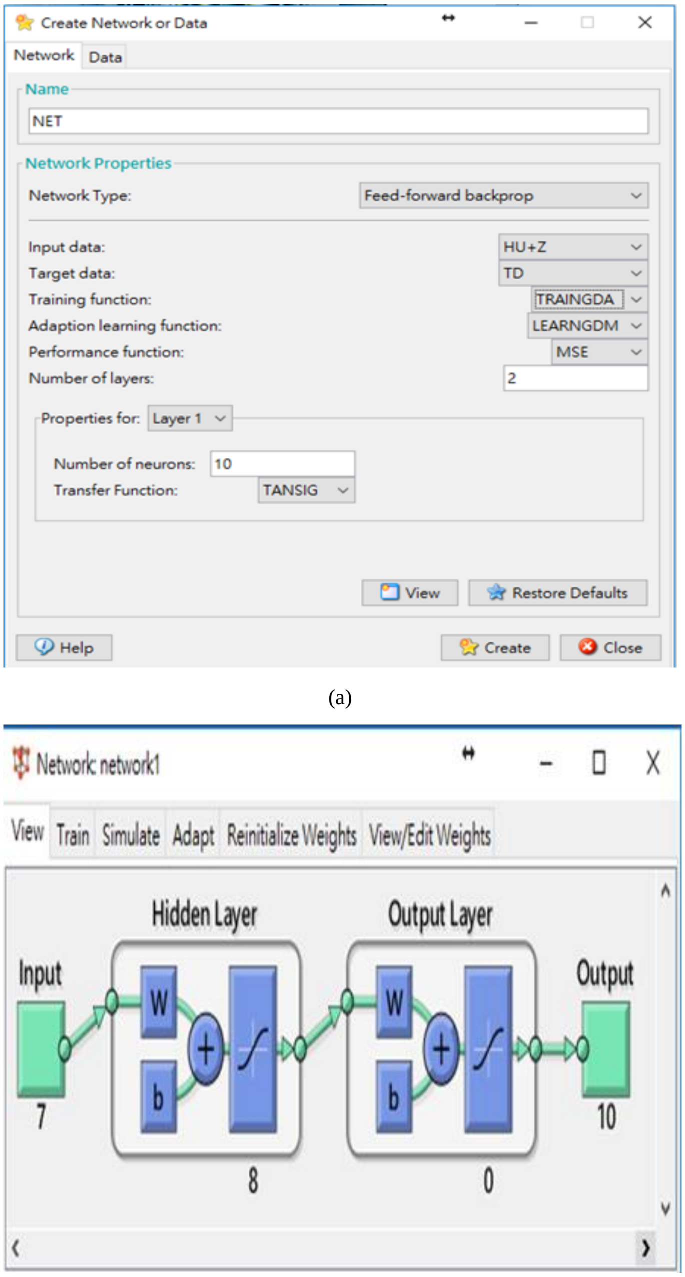
In general, the geometrical properties of a position, size, and direction of an image are related to the matrix. Seven invariant moments are obtained by the second and third moments. The seven invariant moments are not sensitive to the rotation, shift, zoom in and zoom out. Although the calculation of the error will cause a little difference, but in general the feature of the moment is quite obvious. In this project, ten category vectors are used as training samples, and the invariant moments of different sizes, positions and rotation angles are calculated for each sample. The average vector $\mathbf{m}_j$ of each category is calculated as the eigenvalues of each sample and the eigenvalues of new captured image is

$$\mathbf{x} = [\varphi_1 \quad \varphi_2 \quad \varphi_3 \quad \varphi_4 \quad \varphi_5 \quad \varphi_6 \quad \varphi_7] \tag{5}$$

And the classifier is defined as

$$D_j(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_j - \tfrac{1}{2}\mathbf{m}_j^T\mathbf{m}_j \tag{6}$$

According to the maximum value of D_j (x), the *j*th category can be found. The backpropagation algorithm is adopted to train and the NNTOOL of Matlab is used to establish the neural network model. After establishing the model, the trained data is loaded and embedded into the memory of microcontroller. As the eigenvalues of new captured image is entered, the correct category of the object can be decided. Fig. 1(a) is the NNTOOL interface and Fig. 1(b) is the neural network model.



Create Network or Data

Network | Data

Name: NET

Network Properties

Network Type: Feed-forward backprop

Input data: HU+Z

Target data: TD

Training function: TRAINGDA

Adaption learning function: LEARNGDM

Performance function: MSE

Number of layers: 2

Properties for: Layer 1

Number of neurons: 10

Transfer Function: TANSIG

(a)

(b)

Fig. 1. (a) NNTOOL interface; (b) Neural network model.

Further to describe the object shape, the Zernike invariant moments are used. When calculating the Zernike moments of the image, the primitive coordinates are mapped into the unit circle with the center of gravity of the image as the origin. The Zernike moments are complex moments, and the Zernike moments are often characterized by the shape of the object. Firstly, the Cnm and Snm are calculated then the absolute value of Znm can be obtained. In this project, the Zernike invariant moments, containing 25 values, are calculated and three category vectors are used as training samples. The NNTOOL of Matlab is also used to establish the neural network model, which is trained via backpropagation algorithm. Same as previous procedure, the trained model is also embedded into the microcontroller. As the eigenvalues of new captured image is entered, the correct category of the object can be decided.

III. Implementation Of Gripper And Its Controller

The Cero sketch of the proposed handling gripper is shown in Fig. 2 and the implemented gripper prototype is shown in Fig. 3. The control problem is to find a control law so that the gripper can find and grasp the desired object. First, a TSK fuzzy controller is established to control the base motors of the robot platform to near the grasped object. Then the controlled signals are sent to the motors of the gripper to trig the grasping action. The hardware structure and control block diagram are shown in Fig. 4 and Fig. 5, respectively. Fig. 6 is the program flowchart of Raspberry Pi microcontroller.

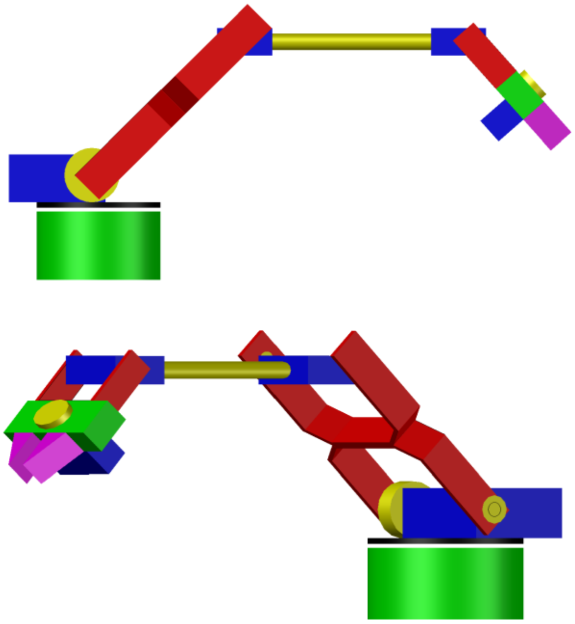


Fig. 2. The Cero sketch of the proposed handling gripper.

The detailed descriptions are exhibited as follows. The action to grasp object can be decomposed into ten distinct steps in the proposed handling gripper prototype with vehicle robot platform:

Step 1: Search the object.
Step 2: Preprocess and identify the captured image.
Step 3: Test whether the object is in line with the gripper by identifying the position of the pasted mark. The mark is pasted in the middle of object. If the mark is in the middle of the image then the object is in line. If not, rotate the pet robot until in line.
Step 4: Trig the ultrasound sensor and approach the object.
Step 5: Keep the distance in 15 cm between the object and pet robot by reading the returned value of ultrasound sensor.
Step 6: Test again whether the object is in line with the gripper. If not, rotate the pet robot until in line.

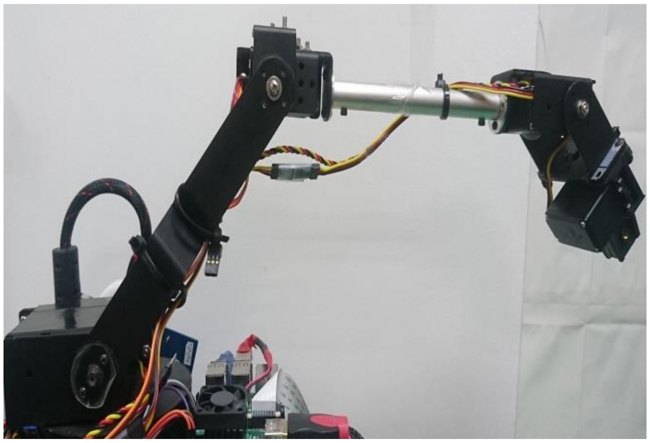
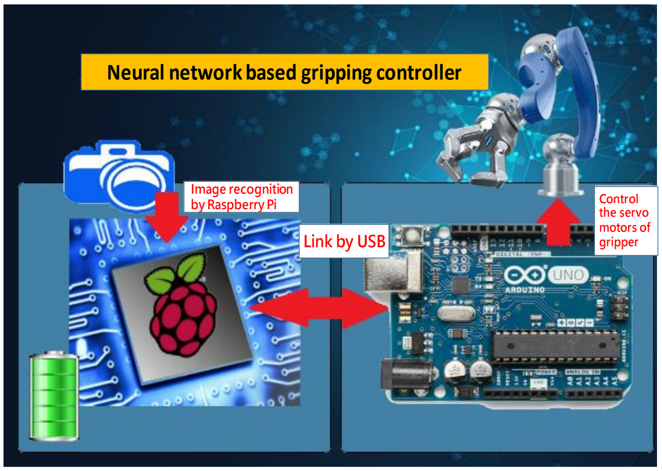


Fig. 3. Implemented gripper prototype.



Neural network based gripping controller
Image recognition by Raspberry Pi
Link by USB
Control the servo motors of gripper

Fig. 4. Proposed architecture of the controller hardware.

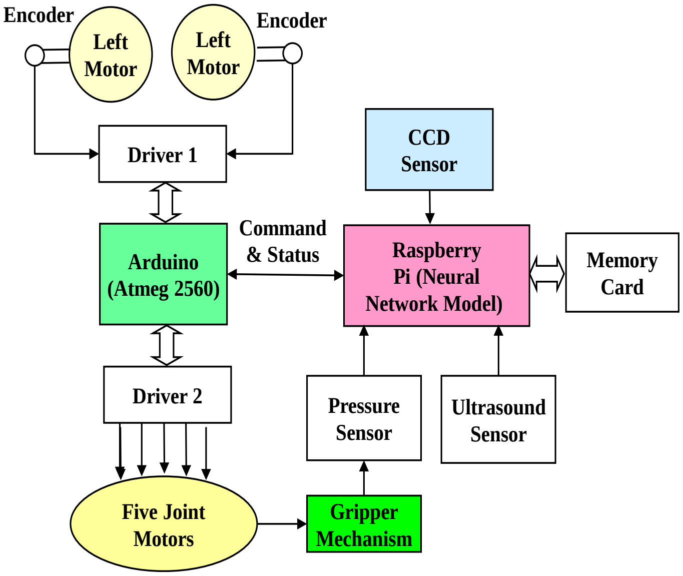


Fig. 5. Control block diagram.

Step 7: According to the identifying result, read the depth and width of the object from gathered data base.
Step 8: Grasp test.
Step 9: Read the pressure sensor. If the value is not match then release and go to Step 8, otherwise go to next step.
Step 10: Lift, move and release object.

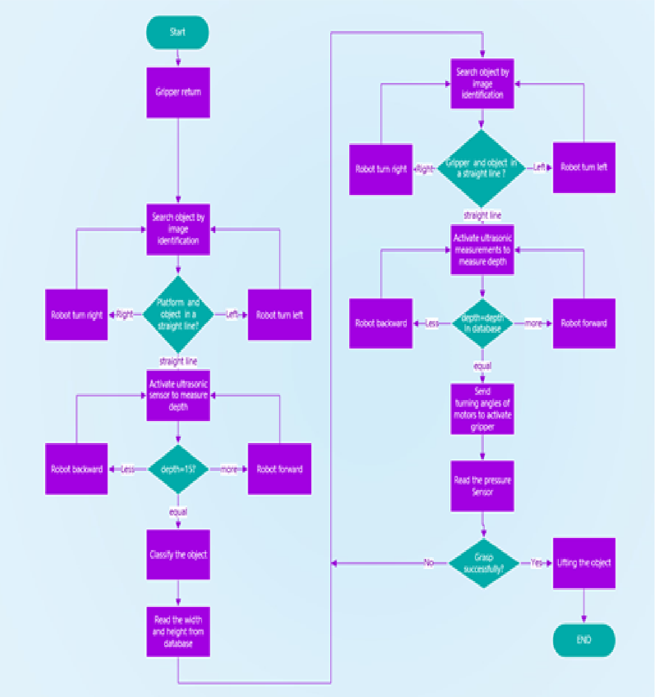

Start
Gripper return
Search object by image identification
Platform and object in a straight line?
Right
Left
Robot turn right
Robot turn left
straight line
Activate ultrasonic sensor to measure depth
depth=15?
Less
more
Robot backward
Robot forward
equal
Classify the object
Read the width and height from database
Search object by image identification
Gripper and object in a straight line ?
Right
Left
Robot turn right
Robot turn left
straight line
Activate ultrasonic measurements to measure depth
depth=depth in database
Less
more
Robot backward
Robot forward
equal
Send turning angles of motors to activate gripper
Read the pressure Sensor
Grasp successfully?
No
Yes
Lifting the object
END

Fig. 6. Program flowchart of Raspberry Pi microcontroller.

IV. Simulation and Experimental Results

The image identifications for mobile phone, kettle and porter bottle are carried out. First, the simulation results of the Hu and Zernike invariant moments are summarized in Tables 1and 2. Then the 7 Hu values in Table 1 are used to train the NN model. The trained results of the NN model are depicted in Fig. 7 and the identification results are shown in Table 3. For comparison, pure gradient descent (GD) method and adaptive gradient descent (AGD) method are also carried out and the identification results are shown in Tables 4 and 5, respectively.

Table 1
Hu invariant moments

Hu values	mobile phone	kettle	porter bottle
Hu1	1.7364928e-1	1.7365876e-1	1.7366317e-1
Hu2	2.3782500e-3	2.3798600e-3	2.3806800e-3
Hu3	7.8040664e-10	4.1849030e-10	4.7620623e-10
Hu4	2.6962462e-11	1.0359705e-11	2.2913269e-11
Hu5	-1.7041451e-21	-5.7996288e-22	-2.2493076e-21
Hu6	-1.0921835e-12	-3.3195986e-13	-1.1046331e-12
Hu7	-3.5203202e-21	3.5907817e-22	-8.1811779e-22

Table 2
Zernike invariant moments

Zernike values	mobile phone	kettle	porter bottle
Z1	3.1830989e-1	3.1830989e-1	3.1830989e-1
Z2	2.6242820e-2	2.8166740e-2	1.7629200e-2
Z3	1.3308085e-1	9.6083600e-2	2.4739000e-2
Z4	2.1953600e-2	5.4299740e-2	1.9100140e-2
Z5	6.7308870e-2	1.9075780e-2	7.9891400e-3
Z6	1.6762430e-2	2.2346870e-2	1.1916820e-2
Z7	1.0340753e-1	3.9908610e-2	1.6976670e-2
Z8	4.6742910e-2	6.0618170e-2	1.7276290e-2
Z9	8.8914000e-3	3.1120380e-2	8.2173400e-3
Z10	4.0629900e-2	2.3996830e-2	9.2487900e-3
Z11	2.4107390e-2	2.9212090e-2	4.6141800e-3
Z12	7.1298700e-3	2.7251110e-2	3.5044400e-3
…	…	…	…

Table 3
Identification results of proposed AMGD method

I/P O/P	mobile phone	kettle	porter bottle
O1	0.9981130	0.0016280	0.0015990
O2	0.0012360	0.9982910	0.0018460
O3	0.0016060	0.0016120	0.9983830

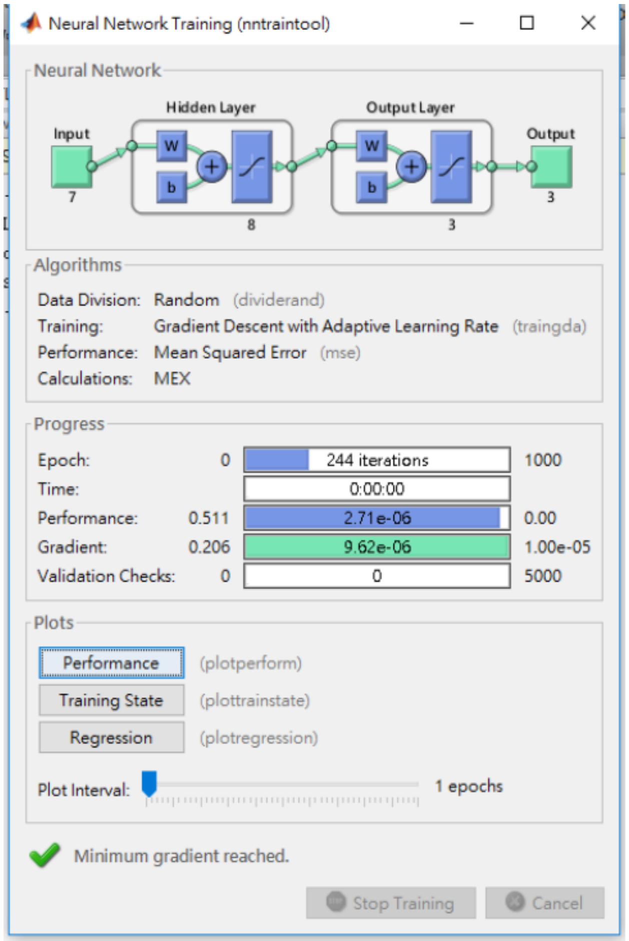
Then the trained data of the NN model is loaded and embedded into the memory of Raspberry Pi microcontroller. Some experiments are carried out with the implemented gripper prototype. The process of line correction is shown in Fig. 8, in which the grasped object is in left and the center of line is drawn as white line. The PWM signals of left and right motors under large forward are shown in Fig. 9.

Table 4
Identification results of pure GD method

I/P O/P	mobile phone	kettle	porter bottle
O1	0.9768667	0.0192989	0.0159469
O2	0.0110754	0.9750179	0.0141126
O3	0.0055968	0.0172089	0.9859714

Table 5
Identification results of AGD method

I/P O/P	mobile phone	kettle	porter bottle
O1	0.9971988	0.0022747	0.0015335
O2	0.0023882	0.9973780	0.0021015
O3	0.0026306	0.0021622	0.9973924



Algorithms

Data Division: Random (dividerand)
Training: Gradient Descent with Adaptive Learning Rate (traingda)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

Epoch:	0	244 iterations	1000
Time:		0:00:00	
Performance:	0.511	2.71e-06	0.00
Gradient:	0.206	9.62e-06	1.00e-05
Validation Checks:	0	0	5000

Plots

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

Plot Interval: 1 epochs

✓ Minimum gradient reached.

Fig. 7. The trained results of the NN mode.

V. Conclusion

To continue the research of the intelligent handling gripper in the previous project, a smart image identification and neural network model based handling gripper is proposed in this paper. In the proposed gripper, the camera module, pressure sensor and the backpropagation neural network model are utilized to increase its sensitivity. Three functions have been accomplished in this project: (1) Mechanism design and prototype making of a simple smart handling gripper. (2) Image recognition of the position and type of the grasped object. (3) The integration of smart handling gripper prototype with vehicle robot platform. From the simulated, experimental results and the prototype performance, the proposed hardware architecture and neural network identification method possess good recognition.

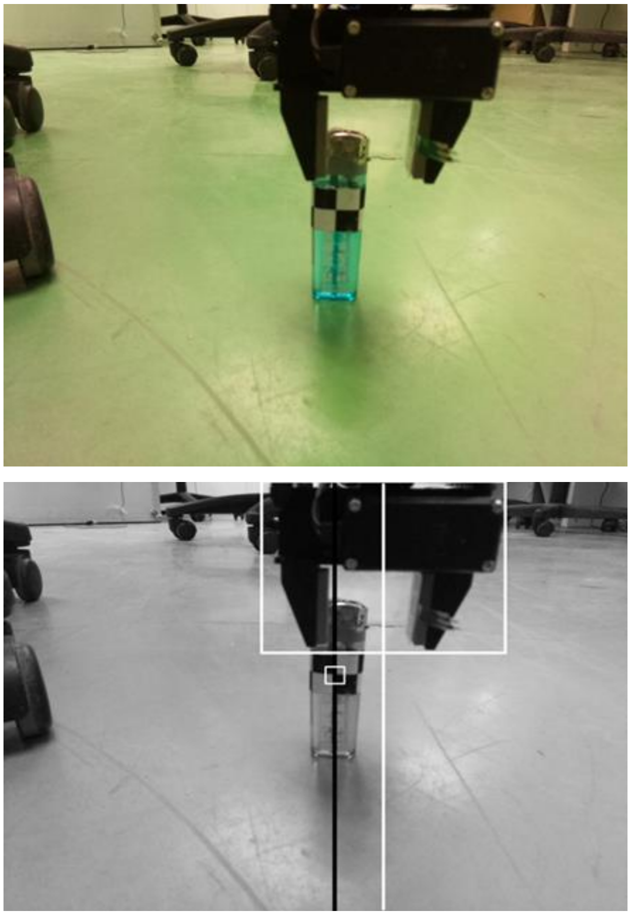


Fig. 8. The process of line correction (object in left).

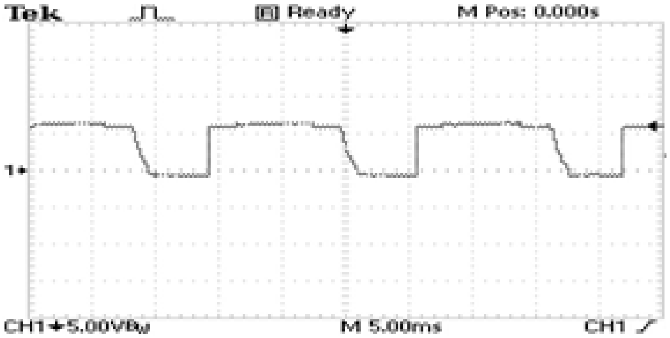


Fig. 9. PWM signals of left and right motors under large forward.

Acknowledgment

The authors gratefully acknowledge the financial support by MOST of Taiwan, R.O.C. through its grant: MOST 107-2622-E-034-002 -CC3.

References

[1] F.W. Ning, Y. Shi, M.L. Cai and W.Q. Xu, "Various realization methods of machine-part classification based on deep learning," Journal of Intelligent Manufacturing, DOI: 10.1007/s10845-020-01550-9, 2020.

[2] Q.K. Song, H.J. Yuan and T. Zhou, "License plate recognition based on mathematical morphology method and RBF neural network," in Proc. of 2012 Int. Conf. on Measurement, Information and Control, Harbin, China, 18-20 May 2012, pp. 782-786.

[3] C.S. Fahn, B.Y. Su and M.L. Wu, "Vision-based identification extraction for freight containers," in 2015 Int. Conf. on Wavelet Analysis and Pattern Recognition, Guangzhou, China, 12-15 July 2015, pp. 51-57.

[4] A. Shemarry, M. Salman, Y. Li and S. Abdulla, "An efficient texture descriptor for the detection of license plates from vehicle images in difficult conditions," IEEE Trans. on Intelligent Transportation Systems, vol. 21, no. 2, pp. 553-564, 2020.

[5] R.S. Hussien, A.A. Elkhidir and M.G. Elnourani, "Optical character recognition of arabic handwritten characters using neural network," in 2015 Int. Conf. on Computing, Control, Networking, Electronics and Embedded Systems Engineering, Khartoum, Sudan, 7-9 Sept. 2015, pp. 456-461.

[6] J. Xing, J. Li, Z. Xie, X. Liao and W. Zeng, "Research and implementation of an improved radon transform for license plate recognition," in 2016 8th Int. Conf. on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, Zhejiang, China, 27-28 Aug. 2016, pp. 42-45.

[7] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li and S. Hu, "Traffic-sign detection and classification in the wild," in 2016 IEEE Conf. on computer vision and pattern recognition, Las Vegas, USA, 27-30 June 2016, pp. 2110-2118.

[8] J. Li, X. Mei, D. Prokhorov and D. Tao, "Deep neural network for structural prediction and lane detection in traffic scene," IEEE Trans. on Neural Networks and Learning Systems, vol. 28, no. 3, pp. 690-703, 2017.

[9] Y.S. Li, W.Y. Xie and H.Q. Li, "Hyperspectral image reconstruction by deep convolutional neural network for classification," Pattern Recognition, vol. 63, pp. 371-383, 2017.

[10] J. Kileel, M. Trager and J. Bruna, "On the expressive power of deep polynomial neural networks," in 33rd Conf. on Neural Information Processing Systems, Vancouver, Canada, 13-14 Dec. 2019, pp. 1-10.

[11] L. Chuankai and Q. Hong, "Vision-based 3-D grasping of 3-D objects with a simple 2-D gripper," IEEE Trans. on Systems, Man, and Cybernetics, vol.44, no.5, pp.605-620, 2014.

[12] L. Bodenhagen and A.R. Fugl, "An adaptable robot vision system performing manipulation actions with flexible objects," IEEE Trans. on Automation Science and Engineering, vol.11, no.3, pp.749-765, 2014.

[13] V. Lippiello and F. Ruggiero, "Visual grasp planning for unknown objects using a multifingered robotic hand," IEEE/ASME Trans. on Mechatronics, vol.18, no.3, pp.1050-1059, 2013.

[14] X. Wang and L. Chen, "A vision-based coordinated motion scheme for dual-arm robots," Journal of Intelligent & Robotic Systems, vol.97, no.1, pp.67-79, 2020.

[15] J.M. Romano and K. Hsiao, "Human-inspired robotic grasp control with tactile sensing," IEEE Trans. on Robotics, vol.27, no.6, pp.1067- 1079, 2011.

[16] Z.Y. Chu, L. Su, G. Chen, J. Cui and F.C. Sun, "A miniaturized five-axis isotropic tactile sensor for robotic manipulation," IEEE Sensors Journal, vol.19, no.22, pp. 10243-10252, 2019.

[17] S. Nisar, M.O. Martinez, T. Endo, F. Matsuno and A.M. Okamura, "Effects of different hand-grounding locations on haptic performance with a wearable kinesthetic haptic device," IEEE Robotics and Automation Letters, vol.4, no.2, pp. 351-358, 2019.

[18] T. Endo, K. Umemoto and F. Matsuno, "Exponential stability of dual flexible arms for grasping and orientation control," IET Control Theory and Applications, vol.13, no.16, pp. 2546-2555, 2019.

[19] J.M. Kang, S.H. Choi, J.W. Park, K.S. Park, "Position error prediction using hybrid recurrent neural network algorithm for improvement of pose accuracy of cable driven parallel robots," Microsystem Technologies– Micro and Nano Systems- Information Storage and Processing Systems, vol. 26, no. 1, pp. 209-218, 2020..

[20] R.J. Wai and Y.C. Huang, “Adaptive fuzzy-neural-network velocity sensorless control for robot manipulator position tracking,” IET Trans. on Control Theory & Applications, vol.4, no.6, pp.1079-1093, 2010.

[21] K.H. Su, “Robust tracking control design and its application to balance a two-wheeled robot steering on a bumpy road,” Journal of Systems and Control Engineering, Proceedings of the Institution of Mechanical Engineers, Part I, vol. 226, no.7, pp. 887-903, 2012.

[22] J.J. Xu, Y.F. Li, L.S. Xu, C. Peng, S.Q. Chen, J.F. Liu, C.C. Xu, G.X. Cheng, H. Xu and Y. Liu, “A multi-mode rehabilitation robot with magnetorheological actuators based on human motion intention estimation,” IEEE Trans. on Neural Systems and Rehabilitation Engineering, vol. 27, no. 10, pp. 2216-2228, 2019.

[23] R.J. Wai and R. Muthusamy, “Fuzzy-neural-network inherited sliding-mode control for robot manipulator including actuator dynamics,” IEEE Trans. on Neural Networks and Learning Systems, vol.24, no.2, pp.274-287, 2013.

[24] J. Cashbaugh and C. Kitts, “Vision-based object tracking using an optimally positioned cluster of mobile tracking stations,” IEEE System Journal, vol.12, no.2, pp. 1423-1434, 2018.

[25] N. Adhikary and C. Mahanta, “Sliding mode control of position commanded robot manipulators,” Control Engineering Practice, vol. 81, pp. 183-198, 2018.

[26] G. Cho, J. Kim and H. Oh, “Vision-based obstacle avoidance strategies for MAVs using optical flows in 3-D textured environments,” Sensors, vol. 19, no. 11, DOI: 10.3390/s19112523, 2019.

Kuo-Ho Su was born in Yunlin, Taiwan, R.O.C., in 1959. He received the B.S., M.S., and Ph.D. degrees in Electrical Engineering from Tatung University, Taipei, Taiwan, R.O.C., in 1983, 1985, and 2005, respectively. Currently, he is a Professor with the Graduate Institute of Digital Mechatronic Technology, Chinese Culture University, Taipei, Taiwan, R.O.C. His research interests include the design and application of intelligent controller (fuzzy, neural network, genetic algorithm), robot system and embedded MCU system.

Chung-Hsien Kuo received the M.S. and Ph.D. degrees in Mechanical Engineering from National Taiwan University in 1995 and 1999, respectively. Currently, he is the Chairman and Professor of Department of Electrical Engineering, National Taiwan University of Science and Technology. He also is the Director of Taiwan TECH Industry 4.0 Center. His research areas include autonomous mobile robots, autonomous driving techniques, artificial intelligence, deep image learning, interactive cognitive learning, intelligent systems for Industry 4.0.

Research and Development of a Disability Care Assistance Management System for Fall Detection

Chien-Wu Lan, *Member*, *RST*, and Sung-Chun Chen

***Abstract*—In the research field of fall detection, the method of proposing a universal, simple, and accurate fall detection method has been a popular topic of discussion. In this study, signal vector magnitude (SVM) and threshold sliding window are combined to extract the feature information of the postures of falls and activities of daily living (ADLs) from an accelerometer as datasets. Thereafter, the back propagation neural network (BPNN) algorithm is used to perform posture classification to determine if a fall has occurred. Using the proposed method, the model trained at the offline training stage can accurately identify a single motion among various types of falls and ADLs. Furthermore, in the experiment on ADLs involving different age groups, the system obtained a specificity of 99.6%, thereby proving that the system has excellent generalizability. In the real-time verification, the system obtained a sensitivity of 100% for fall detection and a false alarm rate of 0.8%. Finally, we compare the proposed method with the descent gradient and k-nearest neighbor algorithms. The results of this comparison reveal that the proposed method has a more robust sensitivity and false alarm rate, as compared to the other methods.**

Index Terms*—disability care, fall detection, LoRa, neural network, three-axis accelerometer*

I. Introduction

The elderly may respond more poorly to the environment as compared to the youth due to their aging characteristics. Consequently, their autonomy in activities of daily living (ADLs) and their ability to integrate ADLs are significantly reduced [1]. Due to disabilities or pre-existing fall experiences, they are significantly more likely to fall again, as compared to average people [2-3]. Therefore, detecting falls in the elderly for the purpose of providing immediate medical assistance has become an important issue for caregiving facilities.

Fall detection methods can be classified by the hardware type of the device into two types, i.e., wearable system and non-wearable system, while fall detection algorithms can be divided into analytical methods (ANM) and machine learning methods (MLM) [4-5].

For non-wearable fall detection systems, floor vibration detection, infrared light, acoustics, and imaging technologies are often widely discussed. The most direct way of fall detection is to deploy a pressure sensor on the ground to detect the vibration of the floor [6], or to use an accelerometer under the tiles to detect the change in the acceleration state caused by the impact pressure during the fall [7], both of which are contact sensing methods. There are also non-contact sensing methods, e.g., behaviors of the elderly can be recognized by sensing changes in infrared flux [8], or the sound wave technology can be used to mark an event as a fall or an ADL based on sound characteristics [9]. In recent years, imaging technologies mainly employ 3D cameras to identify ADLs and falls and extract contours from depth images to estimate human postures and head motion trajectories [10-13]. The advantage of non-wearable systems is that users do not need to wear them, and therefore will not have foreign sensation. However, such solutions require a large number of sensors to be deployed in the environment, resulting in high installation costs, difficulty in deployment, lack of adjustment flexibility, and requirement for regular adjustments. Therefore, wearable systems are mainly used in practice.

Generally, in wearable systems, one or more sensors are fixed on a user to determine if a fall occurs. Common sensors for fall detection include accelerometers (ACC), gyroscopes (Gyro), cardiotachometers, and barometers. Herein, we provide several examples of studies related to accelerometers. Based on an assumption of accelerometer-based fall detection, two MPU6050 six-axis sensors are donned on the chest and the thigh. Subsequently, the threshold (TH) method is used to identify a stand-up, sit-down, lie-down, and fall [14]. However, this wearing method evidently affects user activities in practical applications; therefore it is impractical. In [15] and [16], a wearable device integrating an accelerometer and a barometer is used to check for various states such as falls, sit-downs, and stand-ups, based on the acceleration and inclination and the maximum speed and height difference indicated on the barometer. In [15], the threshold judgment method was employed. Although its detection rate is 100%, this method only relies on a binary judgment, i.e., the judgment is either a “yes” or a “no,” which limits its generalizability for practical applications. In [16], only sit-downs and stand-ups were used to classify fall states, which is insufficient to meet the practical requirements of ADL judgment.

From the aforementioned literature, it is evident that an accelerometer is the primary method to detect a fall, and a cardiotachometer and barometer are subsequently used to help determine the falling state. Accordingly, accelerometers have been widely employed in studies on human postures and possess the advantages of ease of use, portability, and low cost [17-25]. Therefore, several researchers and research institutions have been actively developing this type of fall detection system.

After explaining the advantages and disadvantages of wearing methods and sensors in fall detection, we introduce fall detection algorithms. Fall detection algorithms are the core technology in related research, which can be roughly divided into ANM and MLM [4]. Table I summarizes some of the literature on fall detection from the past five years. The table

The authors deeply acknowledge finance support from the Ministry of Science and Technology (MOST), Taiwan, ROC, under contract: MOST 108-2221-E-606-015-.

C. W. Lan is with the Department of Electrical and Electronic Engineering, Chung Cheng Institute of Technology, National Defense University, Taoyuan City 335, Taiwan (e-mail: chienwulan@gmail.com).

S. C. Chen was with the Department of Electrical and Electronic Engineering, Chung Cheng Institute of Technology, National Defense University, Taoyuan City 335, Taiwan (e-mail: g061984@gmail.com).

presents a comparison of classification, analysis method, wearing location, used sensors, and research results. Each research is evaluated by the sensitivity (true positive ratio, TPR) and specificity (true negative ratio, TNR) of confusion matrix to represent its performance. In the following table of this paper, these two indicators will be represented by this abbreviation. Based on the research, both ANM and MLM have high and low sensitivity. However, the overall sensitivity of MLM is better than that of ANM. These findings also add weight to the need for discussion on determining a universal, simple, and accurate fall detection method that has the aforementioned advantages.

TABLE I
COMPARISON OF FALL DETECTION ALGORITHMS

Ref.	Class.	Method	Location	Sensors	Best Results
[18]	ANM	HIDDEN MARKOV	Neck, Waist, Chest, and in pockets	ACC	TPR:100% TNR: 99.8%
[23]	ANM	TH	Chest	Phone	TPR: 93.33% TNR: 98%
[24]	ANM	TH	Wrist	ACC	TPR: 94.44% TNR:100%
[17]	MLM	BPNN	Waist	ACC, Gyro	TPR: 98% TNR: 96.5%
[19]	MLM	*k*-NN	Wrist	ACC, Gyro, Magneto	TPR:100% TNR: *97.9%*
[21]	MLM	*k*-NN	Back of the vest	ACC, Gyro	TPR: 95.8% TNR: 99.2%
[22]	ANM/ MLM	FUZZY-NN		ACC	TPR: 97.29% TNR: 98.7%
[29]	ANM/ MLM	TH-SVM	Pocket	Phone	TPR: 92% TNR: 99.75%

Based on the results of the aforementioned literature, we propose a left-wrist-mounted fall detection system. Although the accuracy of wrist-mounted devices is not optimal compared to the devices mounted to other parts of the body [25], the wrist is considered to be the most comfortable part for wearing a wearable device. The device uses a three-axis accelerometer, which is commonly built in a commercially available wearable device, to capture users' postures, providing more space for imagination on future commercial applications. The system uses MLM to assess whether a fall occurs and collects training data of different age groups to build a more general classification model. The experiment results show satisfactory accuracy.

II. System Overview

This study proposes a disability care assistance management system in which the fall detection algorithm detects whether a fall occurs. This section explains the system architecture and system flow.

A. System Architecture

To make the designed system have better applicability, we used the Internet of Things (IoT) concept to develop the architecture of the disability care system, which consists of the client, server, and monitoring side, as shown in Fig. 1.

The client is a wearable device that consists of an inertial measurement unit (IMU) integrated with a microcontroller (MCU), and a LoRa wireless communication module. The server is the core of the system, which consists of a disability care assistance management system (equipped with a fall detection algorithm, a database, and an alarm function) and a LoRa Dongle. The monitoring side includes the information broadcasting function and a device (computer or handheld device) that receives information.

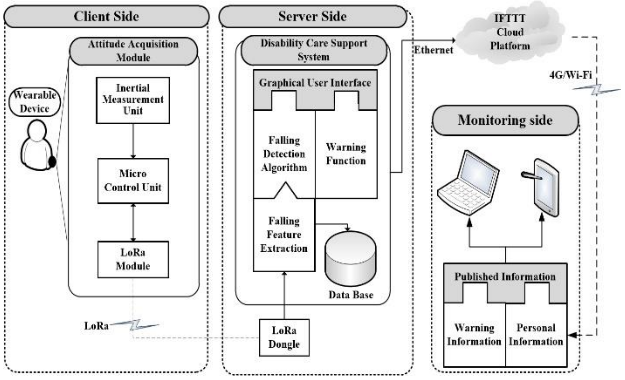


Fig. 1. System architecture.

B. System Flow

The system performs the following steps to detect whether the user falls. First, the three-axis acceleration values of the user's posture are sensed using the IMU in the wearable device on the client side, and returned to the disability care assistance management system on the server side in real-time through LoRa. The system then fuses the data and extracts fall eigenvalues from posture information and uses the fall detection algorithm to identify a fall. After the fall, the system broadcasts the fall information to the monitoring side. Finally, the caregivers on the monitoring side can receive the fall alarm notification through the mobile device and strive for decreased rescue time of the person who fell and reduce the injuries caused by the fall.

In the system flow, real-time and accurate processing of continuous posture sensing information is the key to fall detection. To address this demand, we propose four main methods: 1) fall detection modeling, which fuses the accelerometer data; 2) threshold sliding window, which extracts the most likely fall posture from the continuous sensing signals; 3) learning model, which judges the fall; 4) flow of a fall alarm, which explains how the system provides a fall alarm. These methods are described in detail in the next section.

III. Methodology

After describing the system overview and the functions between the system components, we explain the methods required by the system to detect falls in this section.

A. Fault Detection Modelling

The device reads the three-axis accelerometer values at a sampling interval of 100 ms. It uses the signal vector magnitude (SVM) method to fuse the eigenvalues of the motion postures as a source for fall feature extraction and subsequent processing. The SVM for extracting eigenvalues can be given by Eq. (1):

$$\mathrm{SVM}[i] = \sqrt{a_x[i]^2 + a_y[i]^2 + a_z[i]^2} \qquad (1)$$

Where $a_x[i]$, $a_y[i]$, and $a_z[i]$ represent the accelerometer readings obtained in the ith sampling, and SVM[i] represents the SVM calculated based on the information obtained in the ith sampling. Fig. 2 shows the eigenvalues of the SVM during the fall process. We can conclude from [26-28] that the fall process is divided into three intervals, i.e., descent interval, impact interval, and short static interval, taking about 3 seconds in total. The intervals in the fall process are also used as the time intervals for measuring an ADL.

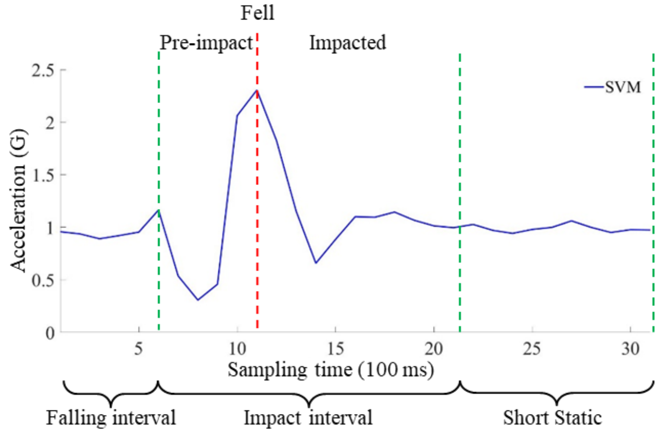


Fig. 2. SVM during the fall.

We can see from Fig. 2 that SVM reaches the peak value upon the fall impact, and the values around the peak are the important information about the posture change before and after the fall impact. Therefore, this characteristic can be further used as a basis for screening effective information.

B. Threshold Sliding Window

Through fall detection modeling, we obtain the SVM peak value, which is the most representative of a fall event, and important information displayed before and after the peak. We can observe from Fig. 3 and Fig. 4 that when the body descends and hits the plane, the SVM peak value in a fall is significantly higher than that in the cases of sit-down and going upstairs. Therefore, the most obvious difference between a fall and an ADL is whether an impact occurs, and only the feature set obtained based on a high enough SVM value may be the feature of a fall event. Hence, we designed a threshold judgment algorithm in reference to [29]. Before the posture data are inputted into the learning model for fall judgment, the threshold judgment algorithm first identifies highly recognizable data with an SVM threshold of 1.8 G (17.64 m/s^2) to improve the fall detection accuracy of the system [29]. When the SVM is smaller than 1.8 G, the system only senses users' postures in real-time. When the SVM is greater than 1.8 G, the system performs fall judgment, and the obtained dataset is judged by the learning model.

Further, the sampling intervals should be defined to detect a fall from the continuous data. To address this problem, we proposed the sliding window method, as shown in Fig. 5.

This method, referring to the fall interval and features after modeling, takes 10 sampling points (T_{-1} to T_{-10}) before and 20 sampling points (T_{+1} to T_{+20}) after the SVM peak sampling time T. 31 parameters are extracted at a sampling period of 100 ms on each axis. Hence, the feature set DataSet(S) including a total of 93 parameters on three axes can be expressed as follows:

$$\text{DataSet}(s) = \left\{ a_s^x, a_s^y, a_s^z \mid s = T_{-10}, \cdots, T, \cdots, T_{+20} \right\} \quad (2)$$

where s represents the sampling time of the fall interval, which is 3.1 s in total. T is the time point when the SVM peak occurs. a_s^x, a_s^y, and a_s^z represent the accelerometer readings at time s.

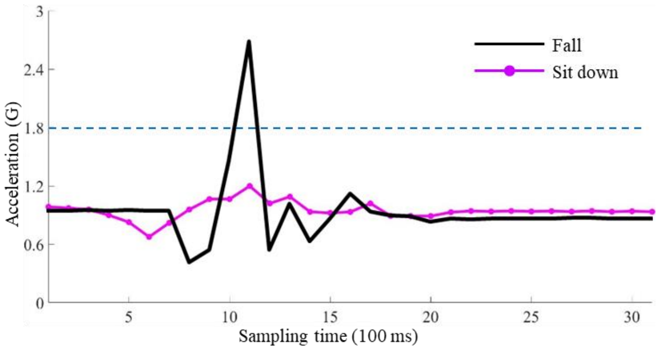


Fig. 3. SVM feature map of sit-downs and falls.

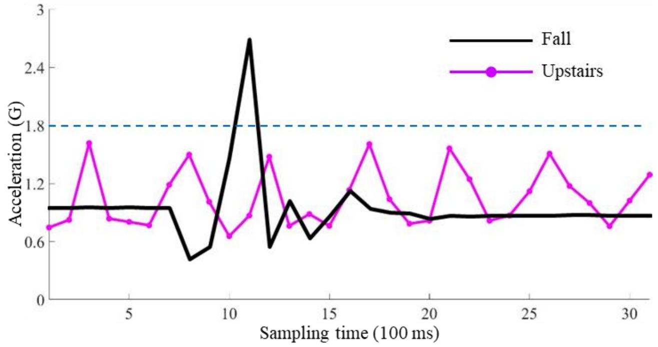


Fig. 4. SVM feature map of going upstairs and falls.

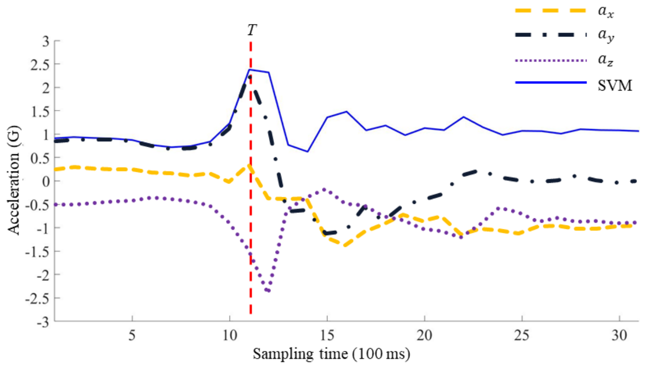


Fig. 5. SVM eigenvalues.

After the design of the threshold sliding window method was completed, we extracted eigenvalues from nine motions, including four fall motions (i.e., forward fall, backward fall, left fall, and right fall) and five ADL motions (i.e., walking, sitting down, standing up, going upstairs, and going downstairs). Judging falls in such continuous and complex sampling signals is a non-linear issue. Therefore, a method capable of effectively judging this type of signal is required to judge falls.

C. *Learning Model*

In this study, we used Levenberg Marquardt's back propagation neural network (BPNN) [30] algorithm in error back propagation (EBP) as the core method for systematically judging falls and took 93 pieces of data extracted from the sliding window as the input. The output of fall judgment was obtained through the BPNN. The idea behind the BPNN is to propagate information forward and errors backward during the learning process. As shown in Fig. 6, a neural network has an input layer (i), an output layer (o), and a (or multiple) hidden layer (h) between the two. The input layer in the network has n inputs (x), the output layer has m outputs (y), and the hidden layer has z neurons (h). The weight from the input layer to the hidden layer is w_{ih}, and the weight from the hidden layer to the output layer is w_{ho}. The transfer functions (f) of the hidden layer and output layer are f_h and f_o, respectively. The partial weights (b) of the hidden layer and output layer are b_h and b_o, respectively. The target output value of training is y_t. Here, the three-layer BPNN is taken as an example for description. The parameters used in this study are explained in the experiment design section.

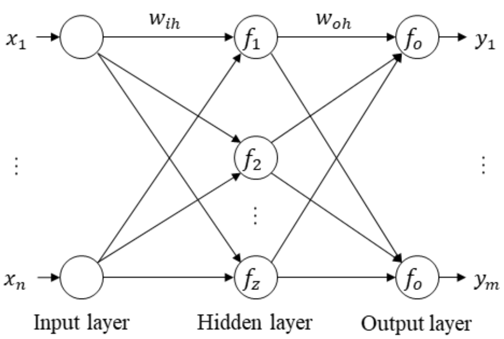


Fig. 6. Architecture of the neural network.

The input layer in the network has a total of n inputs of data from the hidden layer. The output of the hidden layer, α_h, can be given by Eq. (3):

$$\alpha_h = f_h\left(\sum_{i=1}^{n} w_{ih}\cdot x_i - b_h\right),\ h=1,2,\cdots,z \qquad (3)$$

Where $\sum_{i=1}^{n} w_{ih}\cdot x_i - b_h$ is the sum of the product of the weight from the input layer to the hidden layer. We then perform a non-linear transformation through f_h to obtain the output value inferred at the hidden layer, i.e., α_h. We use α_h as the input data of the output layer. In this way, we can obtain the output of the output layer, i.e., y_o, as shown in Eq. (4).

$$y_o = f_o\left(\sum_{h=1}^{z} w_{ho}\cdot \alpha_h - b_o\right),\ \mathrm{o}=1,2,\cdots,m \qquad (4)$$

Eq. (4) can also be rewritten as Eq. (5):

$$y_o = f_o\left(\sum_{h=1}^{z} w_{hz}\cdot f_h\left(\sum_{i=1}^{n} w_{ih}\cdot x_i - b_h\right) - b_o\right) \qquad (5)$$

BPNN is a supervised learning network that aims to reduce the difference between the network output value and the target output value. Therefore, we can use the error function E to test the learning quality after obtaining the network output:

$$E = \frac{1}{2}\sum_{o=1}^{m}\left(y_t - y_o\right)^2 \qquad (6)$$

The back-propagation process is to minimize the error function to meet the expected answer. The descent gradient method is used to find the optimal solution of the error function such that the weight can be adjusted quickly to optimize the network. The corrected magnitude of the weight from the output layer back to the hidden layer can be given by Eq. (7):

$$\Delta w_{oh} = -\eta\frac{\partial E}{\partial w_{oh}} \qquad (7)$$

Where η is the magnitude of the weight corrected by the learning rate, which generally ranges between 0.1 and 0.9. The updated weight w_{oh} can be given as follows:

$$w_{oh}\left(n\right) = w_{oh}\left(n-1\right) + \Delta w_{oh} \qquad (8)$$

We can also calculate the corrected weight of the back propagation from the hidden layer to the input layer by referring to Eqs. (7) and (8). We then iteratively calculate and correct the weight until the network output matches the expected answer.

D. *Flow of Falling Alarm*

The system integrates various information based on the aforementioned methods to provide fall alarms. Fig. 7 is a flowchart of the system's fall detection alarm, which is divided into two stages: offline training and online real-time detection.

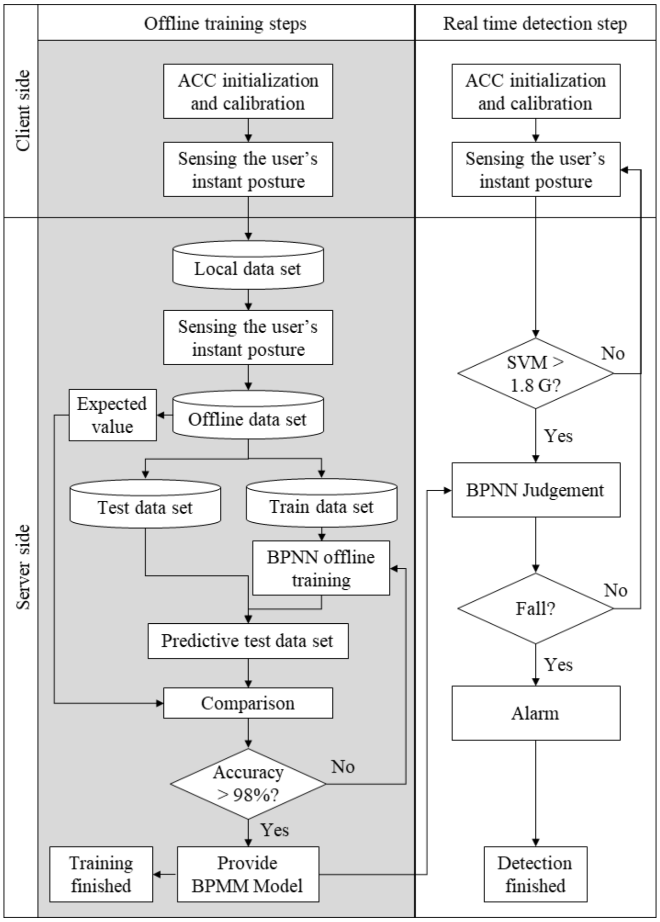


Fig. 7. Flowchart of the fall detection algorithm.

At these two stages, the client first takes the average of the first 100 readings of the accelerometer on each axis to initialize and correct the device. The stages of a fall alarm are described as follows.

1) *Offline Training Stage*

The purpose of this stage is to train an optimal fall judgment algorithm. At this stage, a user collects accelerometer information through the client and sets up training and test datasets for the neural network on the server. For reliable fall judgment of the system, the trained model cannot exit the training and be considered as an optimal model until its fall judgment accuracy reaches above 98%. The trained model can then be used as a fall judgment algorithm at the online real-time detection stage.

2) *Online Real-Time Detection*

After the training of the offline fall judgment algorithm is completed, online real-time detection is performed. At this stage, the system initially filters the accelerometer data infused by the client based on the designed SVM threshold (1.8 G). The fall posture data with a stronger SVM are inputted into the fall judgment algorithm to judge a fall. The performance of this method is explained in the experiment section.

IV. Experiments and Results

After designing a fall detection method, we tested and verified the system and obtained the performance of the system from the experimental results, which are explained in order below.

A. *Experimental Setup*

As shown in Fig. 8, the device developed in this study consists of the IMU, LoRa wireless communication module, MCU, lithium battery, and other parts. We used 3D printed shells and simple hardware to realize the detection of fall postures.

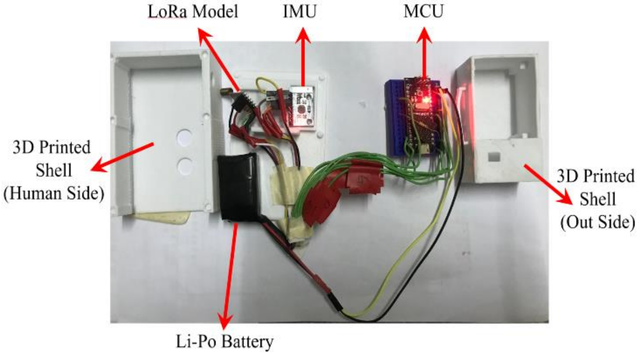


Fig. 8. Device components.

We used the accelerometer in the IMU (MPU6050) to obtain a user's postures on a scale of ± 2 g. We then used the LoRa wireless communication module (iFrog SX1272) to perform data transmission. This module is a long-distance transmission technology with low power consumption, anti-interference capability, and strong penetrability. It is suitable for wearable devices with limited power and transmission distance and can also complete the transmission for the judgment of fall postures. Finally, we used the compact Arduino Nano, the MCU of the wearable device, to process the accelerometer readings and transmit the data to the backend through the LoRa wireless module to judge falls. The Fig. 9 shows the appearance of the wearable device on a user's left wrist and the coordinates of the system.

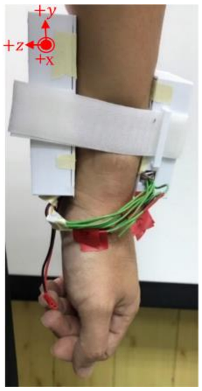


Fig. 9. Appearance and coordinates of the wearable device.

The fall experiment was set up on a cushion with a thickness of 23 cm on the ground. Among ADLs, walking was sampled in a straight line of 10 m on a plane with no height difference. Sitting and standing were sampled on a hard-faced chair with a height of 45 cm. Finally, going upstairs and going downstairs were tested on a total of 12 steps (with a height of 18 cm each).

To verify the performance of the system, we divided the state outputs of the neural network into Positive and Negative to represent falls and ADLs. The training results of the model can be evaluated from the state judgment matrix in Table II.

TABLE II
State Judgment Matrix

		True Class	
		P	N
BPNN Class	Y	True Positives	False Positives
	N	False Negatives	True Negatives

In True Class, P indicates that a fall occurs, and N indicates that a fall does not occur. In BPNN judgment, Y indicates that a motion is judged as a fall, and N indicates that a motion is judged not as a fall. True Positives (TP) indicate that a fall is correctly classified, False Negatives (FN) indicate that a fall is not detected by the BPNN, True Negatives (TN) indicate that an ADL is correctly classified, and False Positives (FP) indicate that an ADL is incorrectly classified by the BPNN.

The training results were evaluated based on indicators such as sensitivity, specificity, accuracy, and false alarm rate (FAR). These indicators can be calculated as follows:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{PN}} \tag{9}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{10}$$

$$\text{Accurancy} = \frac{\text{TN} + \text{TP}}{\text{P} + \text{N}} \tag{11}$$

$$\text{False Alarm Rate} = \frac{\text{FP}}{\text{P} + \text{N}} \tag{12}$$

Sensitivity measures the proportion of correct judgments of a fall when a fall occurs. Specificity measures the proportion of judgments of non-falls when there is no fall. Accuracy measures the proportion of falls and ADLs being correctly judged. The false alarm rate measures the proportion of ADLs being judged as falls. The higher the sensitivity and specificity, the higher the system reliability and accuracy. On the other hand, the lower the false alarm rate, the better.

After the experimental setup, we verified the effect of the neural model at the offline training stage on single motion classification and the ADL specificity of the system for different age groups. We then performed a real-time fall sensitivity verification in continuous motion at the online real-time detection stage for the post-learning neural network, to understand the sensitivity of the system in practical applications. Finally, we compared the performance of the proposed BPNN-TH method combining ANM and MLM with that of *K*-NN and TH, which are introduced in order below.t

B. *Single Motion Classification*

This experiment was an offline training conducted in the laboratory. It was mainly used to train and judge whether the BPNN model can correctly discriminate between fall and ADL motions. After obtaining a dataset of the accelerometer readings on users' postures through the wearable device, we generated a 1 × 93 array according to the proposed method and inputted it into an input layer containing 93 neurons. Using prior experimental experience, two hidden layers, each with seven neurons, were used, and the learning rate was 0.001. Finally, the values between 0 and 1 in a 1 × 2 array were outputted at the output layer. We performed model training on this network architecture. The actual performance can be verified by experimental results.

Various motions in this experiment are explained in Table III, and they are also used as the standard motions to follow in each experiment. Nine motions were designed in total, including four fall motions (i.e., forward fall, backward fall, left fall, and right fall) and five ADL motions (i.e., walking, sitting down, standing up, going upstairs, and going downstairs).

In the experiment, we collected data of all motions with the aid of eight participants. Among them, seven participants with normal mobility repeatedly performed each motion 10 times, and the other one only repeated each motion five times due to physical health. Therefore, 75 datasets were obtained for each designed motion. Among the 675 datasets obtained for nine motions, we selected an average of 72 datasets as the test datasets to reduce the specificity between the training data and used the remaining 603 datasets as training datasets.

After completing the training of the neural network, we inputted 72 test sets of fall and ADL motions into the system to judge the motions (with a motion recognition probability between 0 and 1). If the probability converges to 1, it indicates that the motion is classified correctly. In the test results, one dataset of the backward fall was judged to be a fall with a recognition probability of 0.8 and a sit-down with a recognition probability of 0.2. Although this single result did not fully converge, the motion recognition probability was above 0.5, and this motion was therefore still judged as a correct classification. Because the sensitivity, specificity, and accuracy of the test sets were all 100%, the trained result can be used as the optimal neural model. Therefore, the model can complete the verification of various types of single motions. On this basis, we then performed the next experiment.

TABLE III
DEFINITION OF THE MOTIONS

Motion No.	Class	Motion name	Motion description
F_f	FALL	FORWARD	The knees hit the ground before the face points downwards.
F_b		BACKWARD	The bottom hits the ground before the face points upwards.
F_l		LATERAL-LEFT	The left knee hits the ground before the left shoulder, with the face pointing to the left.
F_r		LATERAL-RIGHT	The right knee hits the ground before the right shoulder, with the face pointing to the right.
A_w	ADL	WALK	The left foot steps forward first, and the wrists naturally swing back and forth.
A_s		SIT-DOWN	In a standing posture, the two arms hang down naturally with the palms facing inward. In a sitting posture, the palms lie down flat on the thighs.
A_u		STAND-UP	In a sitting posture, the palms lie down flat on the thighs. The user stands up with hands on thighs and returns to a standing posture with two arms hanging down naturally and palms inward.
A_{us}		UP-STAIRS	The left foot moves upwards, and the wrists naturally swing back and forth, without lying on the escalator.
A_{ds}		DOWN-STAIRS	The left foot moves downwards, and the rest of the movement and environment are the same as those of going upstairs.

C. *ADL Specificity Verification in Different Ages*

M. Kangas et al. have found that the elderly and the middle-aged participants have similar descent intervals and impact intervals during the fall [31]. According to their conclusion, we can judge that the trained BPNN model for falls is also suitable for all age groups, but whether the trained BPNN for ADL motions is suitable for all age groups is uncertain. Therefore, the purpose of this experiment is to verify and detect the ADL specificity, i.e., the proportion of an ADL motion being misjudged as a fall, based on the optimal neural model trained at the previous stage, and to verify whether the ADL motions of non-modelers increase the false alarm rate of the system.

This experiment involved 67 volunteers who were divided into three groups: 21-40 years old, 41-64 years old, and above 65 years old. Table Ⅳ presents the physiological parameters of the volunteers. The volunteers met the following three activity indicators: 1) being able to walk over 10 m without using a walker; 2) being able to stand up from the chair on his or her own; 3) being able to participate in motion detection with spoken and written words. Among them, 11 had suffered foot injuries or structural degradation, but they all met the test conditions.

The experimental environment at this stage is similar to the original experimental setup. We performed motion detection based on the current situation of the volunteers' home and public space environment. The straight line distance was more than 10 m in an open field, and there is no obstruction to affect the inertial behavior of walking forward; the chair height was between 42 cm and 50 cm, and the seat cushion had hard/soft surfaces or were sofas; eight to 10 steps of the stairs were used, with the height of each step ranging being 14 cm to 18 cm.

TABLE IV
PHYSIOLOGICAL PARAMETERS OF THE VOLUNTEERS

Group (years)	Age (years)	Participants		Height (cm)	Weight (kg)
		Male	Female		
21-40	30 ± 6	11	7	169.5 ± 9.5	69.5 ± 20.5
41-64	59 ± 5	6	9	166.0 ± 13.0	69.0 ± 21.0
65 UP	78.5 ± 13.5	19	15	159.5 ± 13.5	63.5 ± 18.5
Total		67			

Volunteers performed five ADL motions each time and repeated them three times. For the 18 volunteers aged 21-40, a total of 270 ADL motions were tested, of which three sit-downs were misjudged as falls. For the 15 volunteers aged 41-64, a total of 225 ADL motions were tested, and no ADL motion was misjudged as falls. For the 34 elderly aged over 65, a total of 510 ADL motions were tested, of which one ADL motion was misjudged as a fall.

We can conclude from Table V that a total of four ADL motions were misjudged as falls in 1,005 tests, with a specificity of 99.6% and a false alarm rate of only 0.4%. This conclusion shows that the system has high reliability in detecting ADLs of different age groups and non-modelers. Next, we performed real-time falling sensitivity verification in continuous motion to understand the performance of the system in actual operation.

TABLE V
STATISTICS OF FALL DETECTION AT THE SECOND STAGE

Category	ADL Activities		
Detect Fall	21-40 Years	41-64 Years	65 Up Years
Y	3	0	1
N	267	225	509
Total		1005	
TPR		99.6%	
FAR		0.4%	

D. Real-Time Falling Sensitivity Verification in Continuous Motion

The verification results at the offline training stage show that the system has a universal and significant effect on fall detection. Next, we conducted an experiment on online real-time detection. At this stage, five males aged 21-35 were tested in the laboratory. First, we conducted a static experiment of single motion detection to confirm the system's stability in detecting a single motion of participants. Participants performed the defined fall and ADL motions two times. The recognition rate of a single motion was then tested. By testing the dataset of 40 fall motions and 50 ADL motions, we obtained a sensitivity of 100% and no false positives. Therefore, we conducted a dynamic experiment of real-time fall sensitivity verification in continuous motion.

Table VI explains the definitions of continuous motions (Act$_1$ to Act$_4$) in the dynamic experiment. To make the experiment close to the real ADLs of the participants, we added four single motions, i.e., opening the door (A_o), pulling the chair (A_{pc}), picking up an item from the ground (A_{pu}), and washing hands (A_{wh}). By testing these continuous motions, we verified the robustness and applicability of the system in real-time and continuously.

TABLE VI
DEFINITIONS OF THE MOTIONS

Motion	Behavioral procedure	Data set	
		Fall	ADL
Act$_1$	$A_{ds} \rightarrow A_{us} \rightarrow A_w \rightarrow F_f$	5	15
Act$_2$	$A_{ds} \rightarrow A_{us} \rightarrow A_w \rightarrow A_o \rightarrow A_w \rightarrow A_{pc} \rightarrow F_b$	5	30
Act$_3$	$A_{ds} \rightarrow A_{us} \rightarrow A_w \rightarrow A_o \rightarrow A_w \rightarrow A_s \rightarrow A_u \rightarrow F_l$	5	35
Act$_4$	$A_{ds} \rightarrow A_{us} \rightarrow A_w \rightarrow A_o \rightarrow A_w \rightarrow A_{pu} \rightarrow A_w \rightarrow A_{wh}$	None	55

Fig. 10 to Fig. 13 show the feature maps of Act$_1$ to Act$_4$ obtained in this experiment, respectively. We can clearly see from Fig. 10 to Fig. 12 that the SVM waveform caused by a fall has a quite high peak, and the signal features can be clearly identified from the motion, showing that setting an SVM threshold can make a preliminary and obvious discrimination between falls and ADLs. On the other hand, if the fierce or irregular signal vibrations caused by a rapid descent and rotation or vibration of hand movements (such as picking up items from the ground, opening doors, washing hands) are higher than the SVM threshold, the BPNN can be used for further classification.

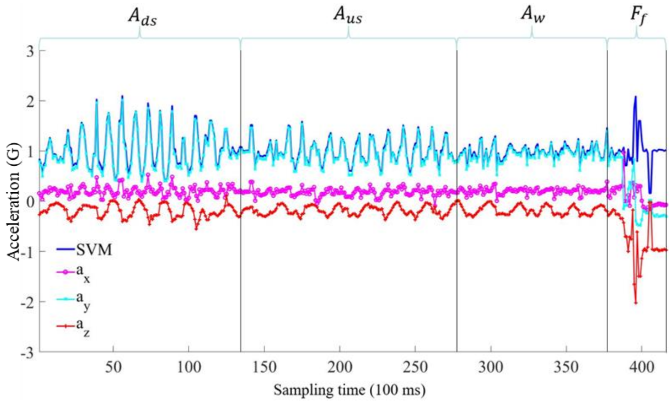


Fig. 10. Feature map of Motion 1.

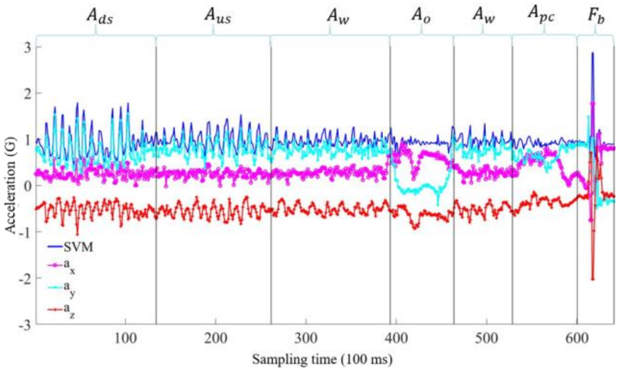


Fig. 11. Feature map of Motion 2.

Table Ⅶ shows statistics of all experiments. The results show that the system has a sensitivity of 100%, a specificity of 98.9%, and an accuracy of 99.2% overall in judging static and dynamic falls. Although there were two false alarms during hand washing in Act$_4$, the overall false alarm rate was only 0.8%, indicating that the system has stable performance.

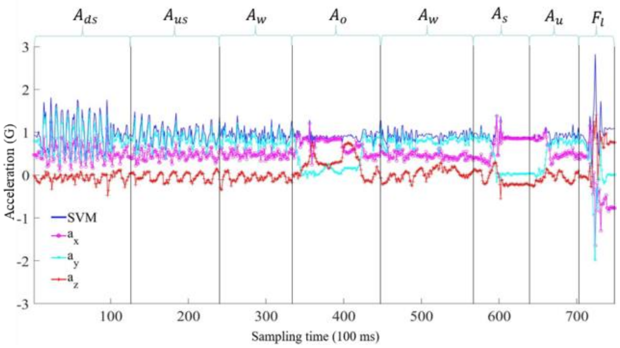


Fig. 12. Feature map of Motion 3.

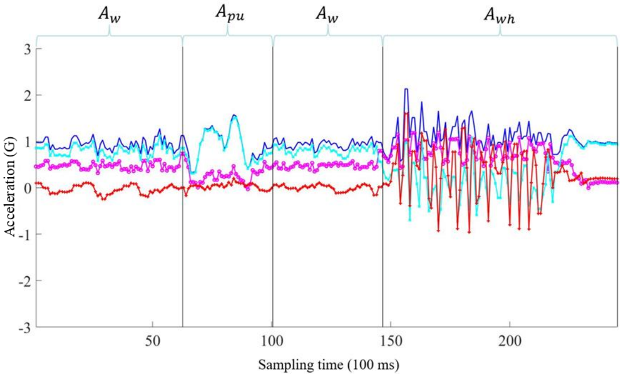


Fig. 13. Feature map of Motion 4 (showing only the movement after standing up).

TABLE VII
ACCURACY OF FALL JUDGMENT AT THE REAL-TIME DETECTION STAGE

Activities	Free-Living										Static Activities		Total
	Act_1		Act_2		Act_3		Act_4		Subtotal				
Detect fall	Y	N	Y	N	Y	N	Y	N	Y	N	Y	N	
Fall(P)	5	0	5	0	5	0	-	-	15	0	40	0	55
ADL(N)	0	15	0	30	0	35	2	53	0	135	0	50	185
TPR	100%		100%		100%		-		100%		100%		100%
TNR	100%		100%		100%		96.3%		98.5%		100%		98.9%
Accuracy	100%		100%		100%		96.3%		98.7%		100%		99.2%
FAR	None		None		None		3.6%		1.3%		None		0.8%

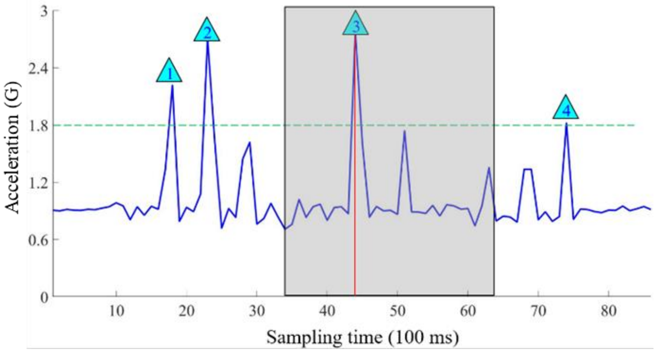


Fig. 14. Feature set of false alarms in the case of hand washing.

We further analyzed one false alarm event of Act4. We found from Fig. 14 that in the interval of false alarm events, four events exceeded the SVM threshold, among which the third event complied with the fall features described in Fig. 5. Although the system has achieved expected results through the SVM threshold and BPNN training methods, more types of ADL datasets can be added to neural training in the future to effectively reduce the false alarm rate.

E. *Performance of Different Fall Recognition Methods*

According to the comparison of the fall judgment methods discussed in the existing literature, the TH judgment and K-NN algorithm are commonly used among ANM and MLM, respectively. For the wrist-mounted fall detection device as well, we combined the K-NN algorithm [19] and descent gradient TH method [24], and tested the sensitivity and false alarm rate using 72 test datasets (including 32 datasets of falls and 40 datasets of ADLs) in offline experiments and 150 datasets in the online real-time experiment (including 15 falls and 135 ADLs). The experimental results are as follows:

1) *K-NN Algorithm*

Based on angle changes, researchers have performed machine learning training on six fall motions (i.e., forward fall, backward fall, left fall, right fall, fall after clockwise rotation, and fall after counterclockwise rotation) and six ADL motions (i.e., walking, clapping, opening and closing doors, moving objects, tying shoestrings, and sitting down), and achieved a 99% accuracy in the offline test [19]. We trained the *K*-NN model using the training dataset used in the offline experiment and achieved a false alarm rate of 1% in classification during the training process. The verification using test datasets shows that the trained model has an accuracy of 100%, which is consistent with the accuracy at the offline training stage.

We then used *K*-NN parameters to perform dynamic detection at the online real-time stage and achieved a sensitivity of 100% and a false alarm rate of 15.3%. False alarms occurred during untrained motions of door opening and hand washing. We concluded that the false alarm rate was very high for untrained motions in this method. Since it is impossible to train all real-life ADLs, the scope of application of *K*-NN is limited.

2) *TH Judgement Method*

In [24], the descent gradient threshold method achieved a sensitivity of 94.44% and a false alarm rate of 2% in the cases of pre-planned falls, walking, and stillness, as well as real-life jogging. Next, we applied the training dataset in the offline experiment to the descent gradient threshold method. In the experimental results, three falls were judged as ADL motions, rendering a sensitivity of 90.6%. Because the accuracy of the TH method was too low during the offline experiment, we no longer performed online real-time dynamic detection. Further, 15 ADL motions were misjudged as falls, rendering a false alarm rate of 20.8%. Further analysis found that the false alarms mainly occurred in motions such as sitting down, standing up, and going downstairs. These three motions are not the motions planned in [24], but this also highlights the use of the TH method alone to implement motion judgment, which clearly shows the limitations of its use.

Based on the above results and data, we presented the performance comparisons in Table Ⅷ. We can clearly see from the table that the BPNN-TH method combining ANM and MLM can achieve better sensitivity and lower false alarm rate than the single method of K-NN and TH judgment.

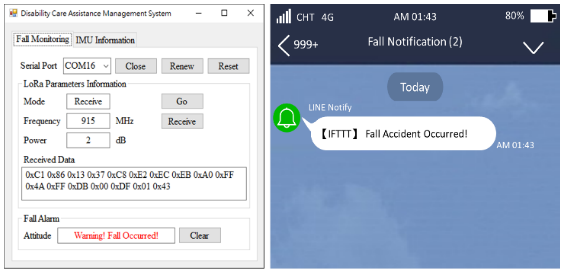
TABLE VIII
COMPARISON OF FALL DETECTION PERFORMANCE OF *K*-NN, TH, AND BPNN-TH

Method	*K*-NN	TH	BPNN-TH
Reference	[19]	[24]	Proposed method
Location	Wrist	Wrist	Wrist
Off-line	TPR：100% FAR：0%	TPR：90.6% FAR：20.8%	TPR：100% FAR：0%
Free-living	TPR：100% FAR：15.3%	None	TPR：100% FAR：1.5%

It is worth noting that the method proposed in this study has quite excellent judgment results in the recognition of non-planned motions in free-living.

V. CONCLUSIONS

In this study, we proposed a fall detection system that only employs an accelerometer for posture measurements. We combined SVM and data from a three-axis accelerometer and thus employed a sliding window method to extract feature information of falls and ADLs from continuous signals. Thereafter, we used a BPNN algorithm to classify the motions for fall judgment. During the offline training stage, the trained model can correctly judge various types of single motions. In the experiment on ADL motions of different age groups, the system can obtain a specificity of 99.6%, thereby yielding excellent generalizability. In the real-time verification, the system could achieve a sensitivity of 100% and a false alarm rate of 0.8% for fall judgment. Finally, the BPNN-TH method exhibited better sensitivity in different motion states than the K-NN and TH methods. In addition, the system uses LoRa wireless transmission and provides fall alarms. Fig. 15 shows that the server sends a fall alarm to the monitoring side through the IFTTT cloud platform on the Internet. We believe that the results of this study can provide a reference for future fall detection systems.



Serial Port COM16

LoRa Parameters Information

Mode Receive

Frequency 915 MHz

Power 2 dB

Received Data

0xC1 0x86 0x13 0x37 0xC8 0xE2 0xEC 0xEB 0xA0 0xFF 0x4A 0xFF 0xDB 0x00 0xDF 0x01 0x43

Fall Alarm

Attitude Warning! Fall Occurred!

(a)

Fall Notification (2)

【IFTTT】 Fall Accident Occurred! AM 01:43

(b)

Fig. 15. (a) Alarm of GUI on PC; (b) Alarm Message of Mobile Device.

REFERENCES

[1] E. M. John, “The top 10 hot topics in aging,” *J. Gerontol*, vol. 59, no. 1, pp. M24–M33, 2004.

[2] S. Jiraporn, P. T. Gregory, F. B. Jonathan, A. L. Lewis, and S. W. Liu, “Revisit, subsequent hospitalization, recurrent fall, and death within 6 months after a fall among elderly emergency department patients,” *Ann. Emergency Med.*, vol. 70, no. 4, pp. 516–521.e2, 2017.

[3] L. Z. Rubenstein, “Falls in older people: Epidemiology, risk factors and strategies for prevention,” *Age Ageing*, vol. 35, no. Suppl_2, pp. ii37–ii41, 2006.

[4] K. Chaccour, R. Darazi, A. H. E. Hassani, and E. Andrès, “From fall detection to fall prevention: A generic classification of fall-related systems,” *IEEE Sens. J.*, vol. 17, no. 3, pp. 812–822, 2017.

[5] Y. S. Delahoz and M. A. Labrador, “Survey on fall detection and fall prevention using wearable and external sensors,” *Sensors*, vol. 14, no. 10, pp. 19806–19842, 2014.

[6] M. Alwan, P. J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder, “A smart and passive floor-vibration based fall detector for elderly,” in IEEE 2nd International Conference on Information & Communication Technologies, 2006, pp. 1003–1007.

[7] M. Daher, A. Diab, M. E. B. E. Najjar, M. Khalil, F. Charpillet, “Auto-matic fall detection system using sensing floors,” *Int. J. Comput. Inf. Sci.*, vol. 12, no. 1, pp. 75–82, 2016.

[8] A. Sixsmith, N. Johnson, and R. Whatmore, “Pyroelectric IR sensor arrays for fall detection in the older population,” *J. Phys. IV France*, vol. 128, pp. 153–160, 2005.

[9] M. Popescu and A. Mahnot, “Acoustic fall detection using one-class classifiers,” in 31st Annual International Conference of the IEEE EMBS, 2009, pp. 3505–3508.

[10] E. E. Stone and M. Skubic, “Fall detection in homes of older adults using the Microsoft Kinect,” *IEEE J. Biomed. Health Inf.*, vol. 19, no. 1, pp. 290–301, 2015.

[11] L. Yang, Y. Ren, and W. Zhang, “3D depth image analysis for indoor fall detection of elderly people,” *IEEE J. Biomed. Health Inf.*, vol. 2, no. 1, pp. 24–34, 2016.

[12] N. Lu, Y. Wu, L. Feng, and J. Song, “Deep Learning for Fall Detection: Three-dimensional CNN combined with LSTM on video kinematic data,” *IEEE J. Biomed. Health Inf.*, vol. 23, no. 1, pp. 314–323, 2019.

[13] E. Akagündüz, M. Aslan, A. Şengür, H. Wang, and M. C. İnce, “Sil-houette orientation volumes for efficient fall detection in depth videos,” *IEEE J. Biomed. Health Inf.*, vol. 21, no. 3, pp. 756–763, 2017.

[14] L. Malheiros, G. D. A. Nze, and L. X. Cardoso, “Body and fall detection system with heart rate monitoring,” *IEEE Latin Amer. Trans.*, vol. 15, no. 6, pp. 1021–1026, 2017.

[15] A. M. Sabatini, G. Ligorio, A. Mannini, V. Genovese, and L. Pinna, “Prior-to- and post-impact fall detection using inertial and barometric altimeter measurements,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 7, pp. 774–783, 2016.

[16] A. Ejupi, M. Brodie, S. R. Lord, J. Annegarn, S. J. Redmond, and K. Delbaere, “Wavelet-based sit-to-stand detection and assessment of fall risk in older people using a wearable pendant device,” *IEEE Trans. Biomed. Eng.*, vol. 64, no. 7, pp. 1602–1607, 2017.

[17] B.T. Nukala, N. Shibuya, A. I. Rodriguez, J. Tsay, T. Q. Nguyen, S. Zupancic, and D. Y. C. Lie, “A real-time robust fall detection system using a wireless gait analysis sensor and an artificial neural network,” in 2014 IEEE Healthcare Innovation Conference (HIC), 2014, pp. 219–222.

[18] S. Yu, H. Chen, and R. A. Brown, “Hidden Markov model-based fall detection with motion sensor orientation calibration: A case for real-life home monitoring,” *IEEE J. Biomed. Health Inf.*, vol. 22, no. 6, pp. 1847–1853, 2018.

[19] T. de Quadros, A. E. Lazzaretti, and F. K. Schneider, “A movement decomposition and machine learning-based fall detection system using wrist wearable device,” *IEEE Sens. J.*, vol. 18, no. 12, pp. 5082–5089, 2018.

[20] R. Liu, C. Yuen, T. Do, D. Jiao, X. Liu, and U. Tan, “Cooperative relative positioning of mobile users by fusing IMU inertial and UWB ranging information,” in 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017 pp. 5623–5629.

[21] J. He, M. Zhou, X. Wang, and Y. Han, “Application of Kalman filter and k-NN classifier in wearable fall detection device,” in 2017 IEEE SmartWorld, 2017.

[22] A. Jahanjoo, M. N. Tahan, and M. J. Rashti, “Accurate fall detection using 3-axis accelerometer sensor and MLF algorithm,” in 2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA), 2017, pp. 90–95.

[23] T. Nakano, B. T. Nukala, S. Zupancic, A. Rodriguez, D. Y. C. Lie, J. Lopez, and T.Q. Nguyen, “Gaits classification of normal vs. patients by wireless gait sensor and Support Vector Machine (SVM) classifier,” in Proceedings of the 15th IEEE/ACIS International Conference on Computer and Information Science, 2016, pp. 99–104.

[24] S. Q. Zhou, J. Chen, X. Wang, L. Zhou, B. Zhen, and J. Cui, “Incli-nation gradient-based fall detection algorithm for wrist-worn device,” in Consumer Electronics – Taiwan (ICCE-TW) 2015 IEEE International Conference., 2015, pp. 148–149.

[25] N. Pannurat, S. Thiemjarus, and E. Nantajeewarawat, "A hybrid tem-poral reasoning framework for fall monitoring," *IEEE Sens. J.*, vol 17, no. 6, pp. 1749–1759, 2017.

[26] S. Abbate, M. Avvenuti, F. Bonatesta, G. Cola, P. Corsini, and A. Vecchio, "A smartphone-based fall detection system," *Pervas. Mob. Comput.*, vol. 8, no. 6, pp. 883–899, 2012.

[27] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti, "A high reliability wearable device for elderly fall detection," *IEEE Sens. J.*, vol. 15, no. 8, pp. 4544–4553, 2015.

[28] M. Kangas, I. Vikman, J. Wiklander, P. Lindgren, L. Nyberg, and T. Jämsä, "Sensitivity and specificity of fall detection in people aged 40 years and over," *Gait Posture*, vol. 29, no. 4, pp. 571–574, 2009.

[29] L. J. Kau and C. S. Chen, "A smart phone-based pocket fall accident detection, positioning, and rescue system," *IEEE J. Biomed. Health Inf.*, vol. 19, no. 1, pp. 44–56, 2015.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning repre-senttations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[31] M. Kangas, I. Vikman, L. Nyberg, R. Korpelainen, J. Lindblom, and T. Jämsä, "Comparison of real-life accidental falls in older people with experimental falls in middle-aged test subjects," *Gait Posture*, vol. 35, no. 3, pp. 500–505, 2012.

Chien-Wu Lan was born in Tainan, Taiwan, 1981. He received the *B.S.*, *M.S.* and *Ph.D.* degrees from the *Department of Electrical and Electronic Engineering*, and the *School of Defense Science*, *Chung Cheng Institute of Technology*, *National Defense University*, Taoyuan, Taiwan, in 2003, 2006, and 2013, respectively. Since 2013, he is an *Assistant Professor* of the *Department of Electrical and Electronic Engineering* of *Chung Cheng Institute of Technology*, *National Defense University*. His current research interests include *humanoid robot*, *computer vision*, and *remote control*.

Sung-Chun Chen received the *M.S. degrees* in *Chung Cheng Institute of Technology*, *National Defense University*, Taoyuan, Taiwan, in 2019. He has experience in automatic control programming and digital control technique. His main interests are *Internet of Things*, *neural network algorithm* and *LoRa communication*.

New Design of Robot Manipulator Controller based on Figure of Demerit Objective Function for Tracking Linear and Nonlinear Trajectories

M. Elsisi*

Abstract— **The main challenge to design the controller of the robotic manipulator is the tracking of a predefined trajectory. Furthermore, most of robotic manipulators consist of more arms and each arm must be track different trajectory with some specifications. This paper suggests a novel design to tune the parameters of the robot manipulator controller to track linear and nonlinear trajectories. The optimization operation is created by utilizing a new tuning algorithm named butterfly optimization algorithm (BOA) instead of the trial and error of designer's expertise. This algorithm is devoted to search for the optimal gains of the proportional integral derivative (PID) controller based on a new figure of demerit objective function (FOD). The suggested technique is evaluated with the multi-objective genetic algorithm (GA) and cuckoo search algorithm (CSA) in Literature. Different tests are created to confirm the superiority of the proposed method.**

Index Terms— Robot Manipulator, Objective Function, Trajectories, Optimization.

I. Introduction

RECENTLY, the robotic controller design represents many challenges in the applications of robot because of the changes in trajectories types and the robotic system nonlinearities [1]. Different controllers are utilized to control the links of the robot manipulator such as proportional integral derivative (PID) controller [2-4], fuzzy logic [5, 6], and artificial neural network [7, 8]. Among these control methods, the PID controller is the most popular one and it is utilized in different industrial applications because it is simple in the implementation [9, 10]. The main challenge that retards the utilization of the PID controller is the selection of the controller parameters. This controller requires a proper tuning for its gains to ensure a good performance for the system response. In literature, different tuning methods are applied to select the controller's gains such as Ziegler Nichols (ZN) method [11] and graphical methods [12]. These methods are conventional and it doesn't prove a good performance because the ZN method is depending on constant basics for any system [13] and the graphical methods are built based on the linearized model of the system. Furthermore, it is complicated mathematically especially in the large systems, and consume a long time [14]. Artificial intelligence (AI) techniques can solve the tuning problem of the controller and it proved good performance in many engineering applications [15]. In [16], the genetic algorithm (GA) is utilized for the optimal tuning of the robotic torque controller. A new particle swarm optimization (PSO) is introduced for the tuning of a fractional PID controller in [17]. In [18], the ant colony optimization (ACO) is applied to select the optimal parameters of the fuzzy controller for a mobile robot. The cuckoo search algorithm (CSA) is introduced for the path planning for a mobile robot in [19]. In [20], the CSA is utilized to optimize the parameters of a sliding mode controller for multi- degrees of freedom robot. The main problem of these optimization algorithms is the trapping in a local optimum solution.

This paper suggests a novel optimization method to tune the robot manipulator controller based on a new figure of demerit objective function (FOD) and the butterfly optimization algorithm (BOA). This algorithm utilizes the cooperative movement between the butterflies to get the optimal global solution and overcome the enclosing in a local optimum solution. The proposed method is compared with the multi- objective GA-PID controller due to [21] and the multi-objective cuckoo search algorithm (CSA)- PID controller due to [22]. The performance of the suggested method is confirmed by carrying out different test scenarios.

II. Butterfly optimization algorithm Overview

Butterfly optimization algorithm is a novel AI technique inspired by the foraging behavior of the butterflies [23, 24]. The cooperation between the butterflies awards this algorithm the manner of global search. The BOA divided into three phases named the initial, iterative, and final stage respectively. In the initial stage, the parameters of the algorithm and the objective function are defined then the initial population is started randomly and the initial solutions are determined. After the initial population, the iterative stage starts by

This work was financially supported by the "Center for Cyber-physical System Innovation" from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

M. Elsisi, [1]Center for Cyber-Physical System Innovation, National Taiwan University of Science and Technology, Taipei, Taiwan., [2]Department of Electrical Engineering, Faculty of Engineering (Shoubra), Benha University, Cairo, Egypt (corresponding author e-mail: mahmoud.elsisi@mail.ntust.edu.tw).

calculating the fitness function of all butterflies. Then, the butterflies generate the fragrance according to stimulus intensity as follows,

$$f = cI^{a} \tag{1}$$

where
f The value of fragrance
c The sensory modality
I The stimulus intensity
a The absorption indicator

Each butterfly in the algorithm produces different fragrance in the intensity according to the fitness function. In the algorithm, there are two steps of search named global and local search respectively with switching probability (*P*). In the global search, the butterfly moves to the best butterfly or solution as follows,

$$x_i(t+1) = x_i(t) + (r^2 * g^* - x_i(t)) \times f_i \tag{2}$$

where
x_i(t) The current solution vector
t The current iteration
i The butterfly index
g^* The current best solution
f_i and r Random number within [0, 1]

In the local search, the butterfly moves randomly to the neighbor butterfly or solution as follows,

$$x_i(t+1) = x_i(t) + (r^2 * x_j(t) - x_k(t)) \times f_i \tag{3}$$

where x_j(t) and x_k(t) are the neighbor butterflies of the current solution. This local moving between the butterflies increases the exploration behavior of the algorithm and prevents it from the trapping in a local optimum solution. The following flowchart in Fig. 1 summarizes the steps of BOA.

III. Robot Manipulator Modeling

Set of differential equations are utilized to describe the robot dynamics. These equations consist of different terms named inertia, torque, load, and gravity terms. The movements of the links in a defined trajectory with a certain speed require appropriate torque to be applied in the actuator of the links. The modeling of the manipulator which represents the robot dynamics of n-links is governed as follows [21],

$$\tau = M(\theta)\ddot{\theta} + C(\theta,\dot{\theta}) + G(\theta) \tag{4}$$

where
τ Vector of links torques
$M(\theta)$ Positive matrix
$C(\theta,\dot{\theta})$ Vector of Coriolis torques
$G(\theta)$ Vector of gravity torques
θ Angular position of links
$\dot{\theta}$ Velocity of links
$\ddot{\theta}$ Acceleration of links
n Number of links

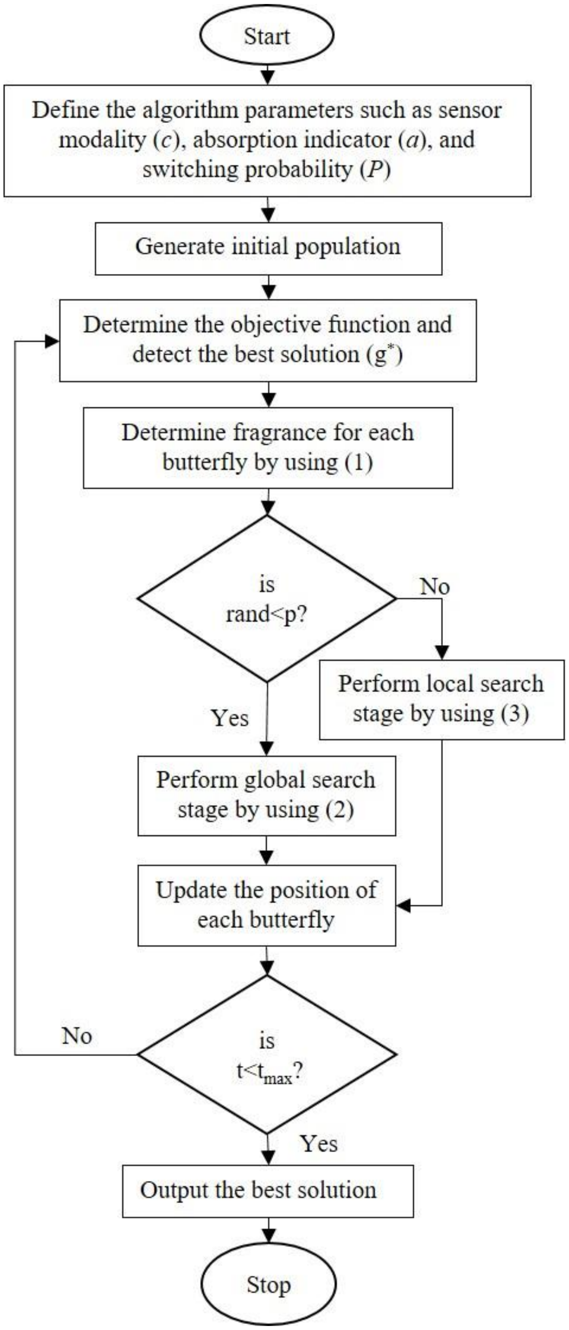

Start
Define the algorithm parameters such as sensor modality (c), absorption indicator (a), and switching probability (P)
Generate initial population
Determine the objective function and detect the best solution (g*)
Determine fragrance for each butterfly by using (1)
is rand<p?
No
Perform local search stage by using (3)
Yes
Perform global search stage by using (2)
Update the position of each butterfly
No
is t<t_max?
Yes
Output the best solution
Stop

Fig. 1. The flowchart of the BOA.

In this paper, two degrees of freedom robotic manipulator is used with ‘n=2’ of links. The dynamics equations that represent this manipulator links which shown in Fig. 2 are described as follows [25],

$$\begin{aligned}\tau_1 &= m_2 l_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) \\ &+ (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 s_2 \dot{\theta}_2^2 \\ &- 2 m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1\end{aligned} \tag{5}$$

$$\tau_2 = m_2 l_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_1 g c_{12} \qquad (6)$$

where $c_1 = \cos(\theta_1)$, $c_2 = \cos(\theta_2)$, $c_{12} = \cos(\theta_1 + \theta_2)$, $s_1 = \sin(\theta_1)$, and $s_2 = \sin(\theta_2)$. In this paper, the control signal of the PID controller represents the torque of each link as follows,

$$\tau_i = K_{P,i} \times e_i + K_{I,i} \int e_i . dt + K_{D,i} \times \frac{de_i}{dt} \qquad (7)$$

$$e_i = \theta_{d,i} - \theta_i \qquad (8)$$

where $i = 1,2$, e_i is the error signal, $\theta_{d,i}$ is the desired trajectory, and θ_i is output angular position.

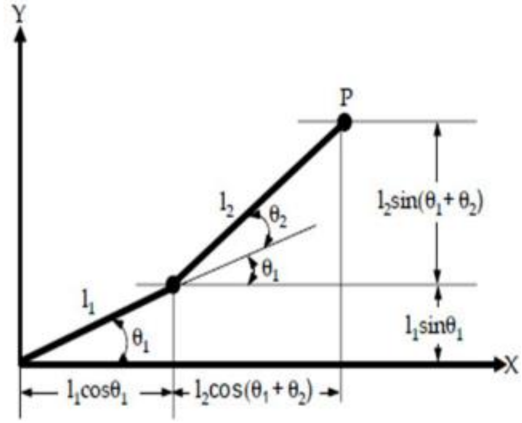


Fig. 2. The geometric representation of two degrees of freedom robotic manipulator.

IV. Results and Discussions

In this section, the BOA is consecrated to get the optimal parameters of the links controllers in order to track any trajectory with less error, settling time, and maximum overshoot. A new time-domain objective function named figure of demerit (FOD) is developed to ensure the minimization of steady-state error, settling time, and maximum overshoot simultaneously. This function is represented as follows,

$$J = \sum_{i=1}^{2} (1 - e^{-\beta})(M_{P,i} + E_{SS,i}) + e^{-\beta}(t_{s,i} - t_{r,i}) \qquad (9)$$

where

$M_{P,i}$	The maximum overshoot
$E_{SS,i}$	The steady-state error
$t_{s,i}$	The settling time
$t_{r,i}$	The rise time
β	A weighting factor
i	The index of each robot link

The value of weighting factor 'β' controls the minimization of settling time and maximum overshoot. When β <0.7, it can minimize the settling time. In contrast, When the β>0.7, it can minimize the maximum overshoot. In this paper, the selected value of β =0.7 to ensure a fair minimization of the settling time, and maximum overshoot. The optimization operation is created at the system nominal parameters and a unit step reference for the position of each link. The system parameters are: m_1=m_2=0.1 kg, l_1=0.8 m, l_2=0.4 m, and g=9.81m/s^2 [21]. The proposed BOA-PID controller is evaluated with the GA-PID controller due to [21] and the CSA-PID controller of [22]. The controller parameters and the corresponding value of the objective function (*J*) are listed in Table I. The steps of the BOA to tune the controller parameters are concluded in the following pseudo-code.

The pseudo-code of the BOA to tune the controller parameters

```
Start BOA
Run the robot manipulator model with the controller
Determine the fitness function in (9)
Select the best solution ($g^*$)
while (t < iterations max ($t_{max}$))
Determine fragrance for each butterfly by using (1)
If  rand<p
Perform global search stage by using (2)
else
Perform local search stage by using (3)
End
Update the position of each butterfly
Run the robot manipulator model with the controller
Determine the fitness function in (9)
Select the best fitness ($g^*$)
end while
Stop
```

It is clear from Table I that the proposed method has less performance index value compared with other methods. The effectiveness of the proposed technique is confirmed by applying the following test scenarios,

Scenario 1: *Step reference trajectory*

In this Scenario, the effectiveness of the proposed method is confirmed by applying unit step input as a reference trajectory for each link. Figures 3 and 4 show the output responses of the system due to the different methods. The maximum overshoots and the settling time of the output responses of the system are listed in Table II. It is clear from Figs. 3, 4 and Table II that the proposed BOA-PID controller has high damped performance and less settling time and overshoot compared with the GA-PID controller and the CSA-PID controller.

Table I The controller parameters of each technique with the corresponding objective function (*j*) value

		GA-PID	CSA-PID	Proposed BOA-PID
Controller	Link	$K_{P,1}$ =	$K_{P,1}$ =	$K_{P,1}$ =

Parameters	1	184.76, $K_{I,1}$ = 49.68, $K_{D,1}$=8.94	782.417, $K_{I,1}$ = 225.2123, $K_{D,1}$=35.1995	249.388, $K_{I,1}$ = 0.4896, $K_{D,1}$=11.9204
	Link 2	$K_{P,2}$ = 11.46, $K_{I,2}$ = 16.54, $K_{D,2}$=0.2	$K_{P,2}$ = 324.523, $K_{I,2}$ = 119.245, $K_{D,2}$=20.1025	$K_{P,2}$ = 192.4835, $K_{I,2}$ = 0.3178, $K_{D,2}$=4.3558
J		1.1758	0.3292	0.0443

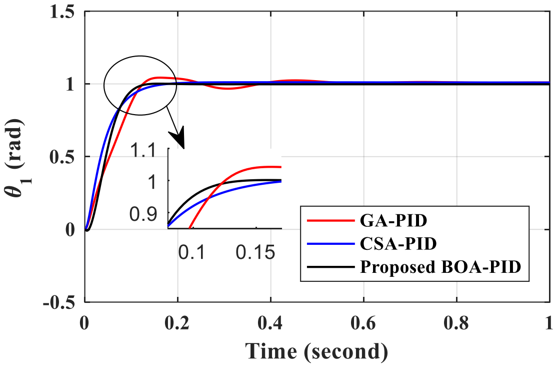


Fig. 3. The output response of link$_1$ in case of Scenario1.

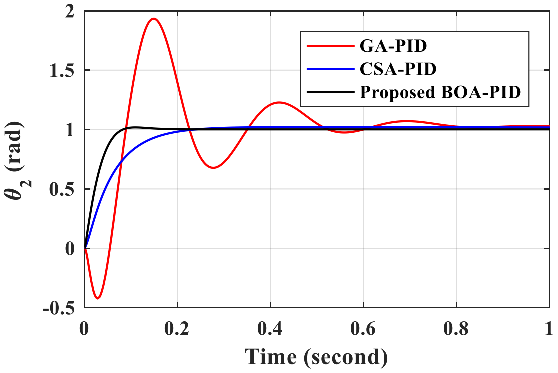


Fig. 4. The output response of link$_2$ in case of Scenario1.

Table II The maximum overshoot and the settling time of the system responses in the case of Scenario 1 due to the different techniques

		GA-PID	CSA-PID	Proposed BOA-PID
Maximum overshoot	Link$_1$	4.301%	1.1421%	0.1791%
	Link$_2$	93.3058%	2.1193%	1.8101%
Settling time	Link$_1$	0.4899	0.1404	0.1138
	Link$_2$	1	0.694	0.0733

Scenario 2: *Nonlinear reference trajectory*

This scenario is created by applying a cubic reference trajectory for each robot link. The cubic reference is generated from the following equation [25],

$$\theta_{d,i} = c_{0,i} + c_{1,i} \times t + c_{2,i} \times t^2 + c_{3,i} \times t^3 \qquad (10)$$

where, $c_{0,1} = c_{0,2} = 0$, $c_{1,1} = c_{1,2} = 0$, $c_{2,1} = 0.09375$, $c_{2,2} = 0.75$, $c_{3,1} = -0.015625$, and $c_{3,2} = -0.125$ at assumed desired final positions of each link are ‘ $\theta_{f,1} = 0.5$ rad and $\theta_{f,2} = 4$ rad’ at final time ‘t_f=4 sec’. The initial position and velocity are equal to zero.

The generated cubic references for each link are shown in Fig. 5. The output response of the system due to this scenario is shown in Figs. 6 and 7. It is concluded from these Figs that the proposed method can track the nonlinear cubic reference trajectory with negligible error.

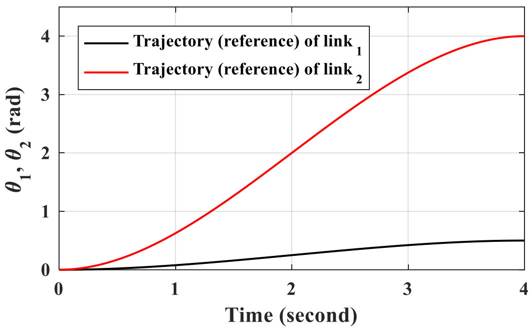


Fig. 5. The cubic references for each robot link.

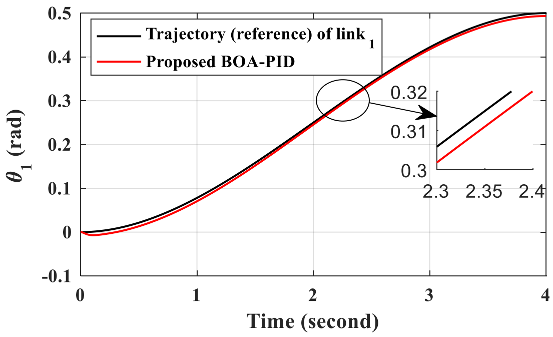


Fig. 6. The output response of link$_1$ in case of Scenario2.

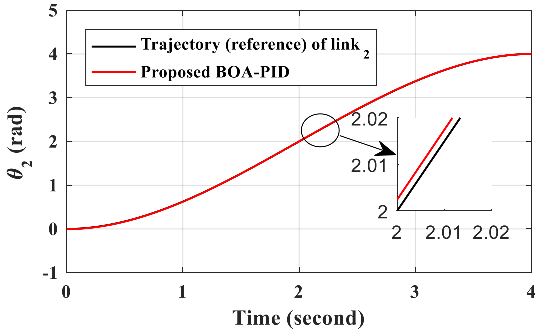


Fig. 7. The output response of link$_2$ in case of Scenario2.

Scenario 3: The effectivity test of the proposed method against parameters uncertainties

This test is created to confirm the effectiveness of the proposed method against the change of system parameters. The test is carried out by changes the masses and the lengths of the robot links with ±20% from the nominal values. Figures 8 and 9 clear that the proposed method can overcome the uncertainties of the parameters with a negligible change in the system response.

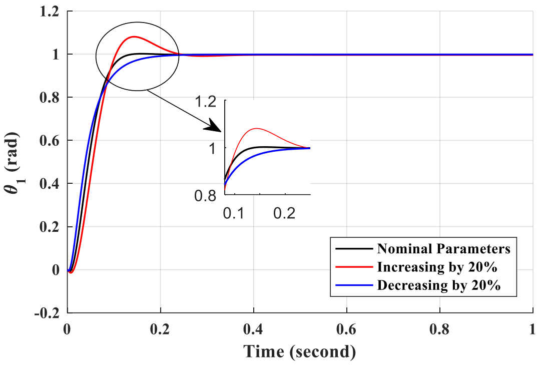


Fig. 8 The output response of link$_1$ in case of parameters uncertainty.

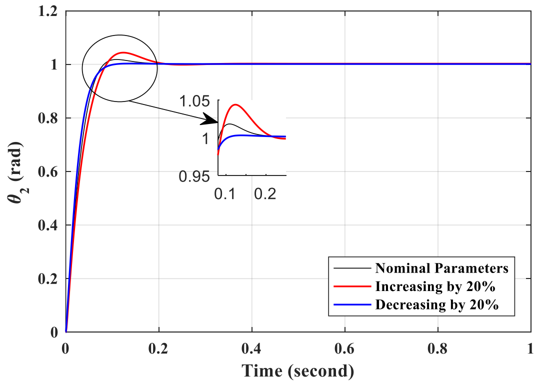


Fig. 9 The output response of link2 in case of parameters uncertainty test.

V. Conclusion and Future Works

This paper introduces a novel global optimization algorithm named BOA for the optimal design of the robotic manipulator controller. The proposed BOA is consecrated to find the optimal parameters of the controller in order to minimize a new developed time-domain objective function. The new objective function can ensure the minimization of the settling time and the maximum overshoot simultaneously. The proposed technique is compared with two previous techniques named GA-PID controller and the CSA-PID controller. The simulation results emphases that the proposed method has high damped and effective performance compared with other methods in case of tracking linear and nonlinear trajectories.

In the future, this work can be extended to use adaptive PID controller based on the proposed method for different robotic manipulator problems with experimental implementation.

References

[1] R. M. Murray, “A mathematical introduction to robotic manipulation,” CRC press. 2017. [Online]. Available: https://www.cds.caltech.edu/~murray/books/MLS/pdf/mls94-complete.pdf.

[2] M. Khosravi and H. Taghirad, “Robust PID control of fully-constrained cable driven parallel robots,” *Mechatronics*, vol. 24, no. 2, pp. 87-97, 2014.

[3] W. Yu and J. Rosen, “A novel linear PID controller for an upper limb exoskeleton,” *In 49th IEEE Conference on Decision and Control (CDC)*, 2010.

[4] J. Li and Y. Li, “Dynamic analysis and PID control for a quadrotor,” In *2011 IEEE International Conference on Mechatronics and Automation*, 2011.

[5] L. Ardila, E. Coronado, H. Hendra, J. Phan, Z. Zainalkefli and G. Venture, “Adaptive Fuzzy and Predictive Controllers for Expressive Robot Arm Movement during Human and Environment Interaction,” *International Journal of Mechanical Engineering and Robotics Research*, pp. 207-219, 2019.

[6] E. Camci, D. Kripalani, L. Ma, E. Kayacan and M. Khanesar, “An aerial robot for rice farm quality inspection with type-2 fuzzy neural networks tuned by particle swarm optimization-sliding mode control hybrid algorithm,” *Swarm and Evolutionary Computation*, vol. 41, pp. 1-8, 2018.

[7] L. Jin, et al., “Robot manipulator control using neural networks: A survey,” *Neurocomputing*, vol. 285, pp. 23-34, 2018.

[8] L. Jin, et al., “Neural dynamics for cooperative control of redundant robot manipulators,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 3812-3821, 2018.

[9] A. D. O. D. S. Dantas, et al., “PID control for electric vehicles subject to control and speed signal constraints,” *Journal of Control Science and Engineering*, pp.1-11, 2018.

[10] J. Moreno-Valenzuela, et al., “Nonlinear PID-type controller for quadrotor trajectory tracking,” *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 5, pp. 2436-2447, 2018.

[11] K. J. Åström and T. Hägglund, “Revisiting the Ziegler–Nichols step response method for PID control,” *Journal of process control*, vol. 14, pp. 635-650, 2004.

[12] S. Srivastava and V. S. Pandit, “A PI/PID controller for time delay systems with desired closed loop time response and guaranteed gain and phase margins,” *Journal of Process Control*, vol. 37, pp. 70-77, 2016.

[13] S. Tavakoli and M. Tavakoli, “Optimal tuning of PID controllers for first order plus time delay models using dimensional analysis,” *In 2003 4th International Conference on Control and Automation Proceedings*, 2003.

[14] H. Wu, et al., “PID controllers: Design and tuning methods,” *In 2014 9th IEEE Conference on Industrial Electronics and Applications*, 2014.

[15] B. K. Patle, et al., “A review: On path planning strategies for navigation of mobile robot,” *Defence Technology*, pp. 1-25, 2019.

[16] M. H. Sangdani, et al., "Genetic algorithm-based optimal computed torque control of a vision-based tracker robot: Simulation and experiment," *Engineering Applications of Artificial Intelligence*, vol. 67, pp. 24-38, 2018.
[17] M. P. Aghababa, "Optimal design of fractional-order PID controller for five bar linkage robot using a new particle swarm optimization algorithm," *soft Computing*, vol. 20, pp. 4055-4067, 2016.
[18] C. F. Juang, et al., "An interpretable fuzzy system learned through online rule generation and multiobjective ACO with a mobile robot control application," *IEEE transactions on cybernetics*, vol. 46, no. 12, pp. 2706-2718, 2015.
[19] P. K. Mohanty and D. R. Parhi, "Optimal path planning for a mobile robot using cuckoo search algorithm," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 28, pp. 35-52, 2016.
[20] S. A. Moezi, et al., "Sliding mode control of 3-RPR parallel robot on the optimal path using cuckoo optimization algorithm," *Modares Mechanical Engineering*, vol. 15, pp. 147-158, 2015.
[21] 5a4bb247-1430-4e46-942c-d692dead831f&groupId=14040.
[22] H. V. H. Ayala and L. dos Santos Coelho, "Tuning of PID controller based on a multiobjective genetic algorithm applied to a robotic manipulator," *Expert Systems with Applications*, vol. 39, pp. 8968-8974, 2012.
[23] H. Chhabra, et al., "Multi-objective Cuckoo Search Algorithm-Based 2-DOF FOPD Controller for Robotic Manipulator," *In Advances in Signal Processing and Communication*, Springer, Singapore, 2019.
[24] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Computing*, vol. 23, pp. 715-734, 2019.
[25] M. Ghanbari and H. Arian, "Forecasting Stock Market with Support Vector Regression and Butterfly Optimization Algorithm," *arXiv preprint arXiv:1905.11462*, pp. 1-11, 2019.
[26] J. J. Craig, "Introduction to robotics: mechanics and control," Pearson Education India, 2009. [Online]. Available: http://www.mech.sharif.ir/c/document_library/get_file?uuid=

M. Elsisi was born in Cairo, Egypt, on 1989. He received his B.Sc., M.Sc., and PhD degrees in 2011, 2014, and 2017 respectively from the Electrical Engineering department, Faculty of Engineering at Shoubra, Beneha University, Cairo, Egypt. From 2017, he worked as an project assistant professor at Electrical Engineering department, Faculty of Engineering at Shoubra, Benha University, Cairo, Egypt. Now, He is project assistant professor at Industry 4.0 Implementation Center, National Taiwan University of Science and Technology for postdoctoral study. His research activity includes studying the power system dynamics, stability, control, and artificial intelligence techniques.

Information for Authors

Aim and Scope

The *i***Robotics** is an official journal of Robotics Society of Taiwan (RST) and is published quarterly. The *i***Robotics** will consider high quality papers that deal with the theory, design, and application of intelligent robotic system, intelligent artificial system, and extension theory systems ranging from hardware to software. Survey and expository submissions are also welcome. Submission of a manuscript should indicate that it has been neither published, nor copyrighted, submitted, accepted for publication elsewhere (except that the short version may have been presented at the conferences). Submitted manuscripts must be typewritten in English and as concise as possible.

Process for Submission of a Manuscript

The *i***Robotics** publishes two types of articles: regular papers and technical notes. All contributions are handled in the same procedure, where each submission is reviewed by an associate editor who makes an initial decision to send the manuscript out for peer review or to reject without external review. Articles can be rejected at this stage for a variety of reasons, such as lack of novelty, topics outside the scope of the Journal, flaws in the scientific validity, or unprofessional presentation. We are therefore not normally able to provide authors with feedback on rejected manuscripts. If the associate editor believes the article may be of interest to our readers, it is then sent out for external peer review by at least two external reviewers. According the recommendation of the associate editor, the Editor-in-Chief makes the final decision. All manuscripts should be submitted electronically in Portable Document Format (PDF) through the manuscript submission system at [http://www.rst.org.tw]. The corresponding author will be responsible for making page proof and signing off for printing on behalf of other co-authors. Upon acceptance of a paper, authors will be requested to supply their biographies (100 to 200 words) and two copies of the final version of their manuscript (in DOC format and in PDF format).

Style for Manuscript

Papers should be arranged in the following order of presentation:

1) First page must contain: a) Title of Paper (without Symbols), b) Author(s) and affiliation(s), c) Abstract (not exceeding 150 words for Papers or 75 words for Technical Note, and without equations, references, or footnotes), d) 4-6 suggested keywords, e) Complete mailing address, email address, and if available, facsimile (fax) number of each author, f) Preferred address for correspondence and return of proofs, and g) Footnotes (if desired).
2) The text: Submitted manuscripts must be typewritten double-spaced. All submitted manuscripts should be as concise as possible. Regular papers are normally limited to 26 double-spaced, typed pages, and technical notes are 12 double-spaced, typed pages. Please see the Page charge for those who want to submit long papers.
3) Acknowledgements of financial or other support (if any).
4) References: References should be numbered and appear in a separate bibliography at the end of the paper. Use numerals in square brackets to cite references, e.g., [15]. References should be complete and in the style as follows.
   [1] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller - Part I," *IEEE Trans. Syst. Man Cybern.*, vol. 20, no. 2, pp. 404-418, 1990.
   [2] C. Golaszewski and P. Ramadge, "Control of discrete event processes with forced events," in *Proc. of 26th IEEE Conf. Decision and Control*, Los Angeles, CA, pp. 247-251, Dec. 1987.
   [3] P. E. Wellstead and M. B. Zarrop, *Self-Tuning Systems*, New York: Wiley, 1991.
   [4] Project Rezero, available at http://rezero.ethz.ch/project_en.html (last visited: 2017-07).
5) Tables
6) Captions of figures (on separate sheet of paper)

Style for Illustrations

1) It is in the author's interest to submit professional quality illustrations. Drafting or art service cannot be provided by the Publisher.
2) Original drawings should be in black ink on white background. Maximum size is restricted to 17.4 by 24.7 cm. Glossy prints of illustrations are also acceptable.
3) All lettering should be large enough to permit legible reduction of the figure to column width, sometimes as small as one quarter of the original size. Typed lettering is usually not acceptable on figures.
4) Provide a separate sheet listing all figure captions, in proper style for the typesetter, e.g., "Fig. 5. The error for the proposed controller."
5) Illustrations should not be sent until requested, but authors should be ready to submit these immediately upon acceptance for publication.

Page Charges

After a manuscript has been accepted for publication, the author's company or institution will be approached with a request to pay a page charge to cover part of the cost of publication. The charges include:

1) NT$ 5000 for the 10 printed pages of a full paper or for the 6 printed pages of a short paper, and the excess page charge of NT$ 1500 per extra printed page for both full and short papers.
2) For color figures or tables, an additional cost will be charged. The cost, depending on the number of color figures/tables and the final editing result, will be given upon the acceptance of this paper for publication.

Copyright

It is the policy of the RST to own the copyright of the technical contributions. It publishes on behalf of the interests of the RST, its authors, and their employers, and to facilitate the appropriate reuse of this material by others. Authors are required to sign a RST Copyright Form before publication.

Manuscripts (in PDF Format) Submission Website: http://www.rst.org.tw

Editor-in-Chief: Prof. Ching-Chih Tsai, Department of Electrical Engineering, National Chung Hsing University, Taiwan
Email: cctsai@nchu.edu.tw
Prof. Tzuu-Hseng S. Li, Department of Electrical Engineering, National Cheng Kung University, Taiwan
Email: thsli@mail.ncku.edu.tw

Managing Editor: Jian-Wen Chen, Department of Electrical Engineering, NTUST, Taiwan. Email: vito9580work@gmail.com
Noorman Rinanto, Department of Electrical Engineering, NTUST, Taiwan. Email: noorman.rinanto@gmail.com

iRobotics

VOLUME 3, NUMBER 1 MARCH, 2020

CONTENTS

REGULAR PAPERS

TECHNICAL NOTE

iRobotics

Volume 3, Number 1, MARCH, 2020