International Journal of Humanoid Robotics
Vol. 14, No. 3 (2017) 1750023 (24 pages)

DOI: 10.1142/S0219843617500232

Visual Servoing of Humanoid Dual-Arm Robot with Neural Learning Enhanced Skill Transferring Control

Chenguang Yang*,† and Junshen Chen†

Zienkiewicz Centre for Computational Engineering
Swansea University, SA1 8EN, UK
**cyang@theiet.org*

Zhaojie Ju

School of Computing
University of Portsmouth, PO1 2UP, UK

Andy SK Annamalai

Moray College
University of Highlands and Islands, IV30 1JJ, UK

Received 30 April 2017
Revised
Accepted 17 July 2017
Published

This paper presents a novel combination of visual servoing (VS) control and neural network (NN) learning on humanoid dual-arm robot. A VS control system is built by using stereo vision to obtain the 3D point cloud of a target object. A least square-based method is proposed to reduce the stochastic error in workspace calibration. An NN controller is designed to compensate for the effect of uncertain payload and other internal and external uncertainties during the tracking control. In contrast to the conventional NN controller, a deterministic learning technique is utilized in this work, to enable the learned neural knowledge to be reused before current dynamics changes. A skill transfer mechanism is also developed to apply the neural learned knowledge from one arm to the other, to increase the neural learning efficiency. Tracked trajectory of object is used to provide target position to the coordinated dual arms of a Baxter robot in the experimental study. Robotic implementations has demonstrated the efficiency of the developed VS control system and has verified the effectiveness of the proposed NN controller with knowledge-reuse and skill transfer features.

Keywords: Neural networks; deterministic learning; visual servoing; stereo vision.

1. Introduction

The issues pertaining to robot control have gained increasing research attention, recently. Visual servoing (VS) is a technique of control using computer vision

† Corresponding authors.

information to control the motion of a robot. It mainly depends on techniques of computer vision, image processing and control theory.[2] It is of great importance in improving the flexibility of robot control systems[18] and has been widely applied. There are two central setups of the camera and the robot end-effector: Eye-in-hand, or end-point open-loop control, which the position of the object is watched by the camera appended to the robot hand; Eye-to-hand, or end-point closed-loop control, which the movement of the end-effector and the object are both be watched by a camera settled on the world frame.[3] In this paper, the control of a Baxter robot arm end-effector using a stereo visual camera ZED as the eye-to-hand camera is addressed. Because of a narrower field of view that eye-in-hand VS provides, as the sensors are attached in the hand. A least squares-based method is proposed to reduce stochastic errors during camera calibration process.

To improve robot arm's control performance, an adaptive controller was developed for robot manipulators.[22] It employed a barrier Lyapunov function-based synthesis to design controller for the manipulator to operate in an ellipsoidal constrained region. An adaptive neural network (ANN) control for the robot system in the presence of full-state constraints is designed.[16] The NN enables the system to deal with uncertainties and disturbances effectively. Among these work, we see that NN technique has been extensively used for robot control system due to its universal approximation ability and its capability to cope with unmodeled dynamics of the robot systems. The highly nonlinear nature of the robot dynamics makes it challenging to obtain an accurate model under practical operational conditions.[24] However, conventional NN control was focused on internal uncertainties. To overcome the uncertainties bring from unknown payload, a novel NN-based intelligent controller is designed in this paper and obtains an enhanced performance of VS control.

Furthermore, the learning ability of conventional NN controllers is limited, since even repeating same task, the parameters of controller need recalculation every time. Therefore, a *deterministic learning* technique has been developed as, not only be able to obtain control dynamic knowledge from closed-loop control process, but also be reuse the obtained knowledge for another similar control task without readapting to the uncertainties of the environments.[7] Deterministic learning is proposed by using deterministic calculations that began from adaptive control, rather than utilizing syntactical standards. The deterministic learning approach tackles the issue of learning in a dynamic situation and is valuable in numerous applications, for example, dynamic pattern recognition,[8] learning and control of robotics,[9] and oscillation faults diagnosis.[10] In addition to the designed NN controller, deterministic learning feature is added in this paper to efficiently reuse the learned knowledge. After the initial learning of the environmental uncertainties, the proposed NN controller do not need to re-learn until dynamics changes. It can greatly reduce the computational load.

With the aim of improving the "intelligence" of robot, a robot-to-robot skill transfer mechanism is proposed in this paper. Unlike the conventional approach of transferring human skills to robot, the learned knowledge from NN controller is transferred from arm to arm with dual-arm robot in this paper. With guaranteed performance, NN controller only need to learn once of system uncertainties on one side of dual-arm. The other arm can perform the same task without readapting the same uncertainties. It can help to increase the neural learning efficiency and also to further reduce the computational load.

In this context, this paper presents an neural learning enhanced VS control system with knowledge reuse and skill transfer features. The system was successfully implemented on a Baxter humanoid robot and test results are demonstrated, which show the potential of the novel learning controller.

2. Preliminaries

Lemma (Ref. 5). Consider a parameterized linear time-varying (LTV) multivariable systems in the following form:

$$\begin{bmatrix} \dot{e} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} A(e,\lambda) & B(e,\lambda)^{\mathrm{T}} \\ -C(t,\lambda) & 0 \end{bmatrix} \begin{bmatrix} e \\ \theta \end{bmatrix}, \quad z := \begin{bmatrix} e \\ \theta \end{bmatrix}, \tag{1}$$

where $e \in \mathbb{R}^n, \theta \in \mathbb{R}^m$, $A(e,\lambda) \in R^{n\times n}$, $B(e,\lambda) \in R^{m\times n}$, $C(e,\lambda) \in R^{m\times n}$, $\lambda \in D \subset R^l$.

There exists a constant $\phi_M > 0$ such that for all $t \geqslant 0$ and for all $\lambda \in D$,

$$\max\left\{ \|B(t,\lambda)\|, \left\| \frac{\partial B(t,\lambda)}{\partial t} \right\| \right\} \leqslant \phi_M. \tag{2}$$

and there exist symmetric matrices $P(t,\lambda)$ and $Q(t,\lambda)$ such that $P(t,\lambda)B(t,\lambda)^{\mathrm{T}} = C(t,\lambda)^{\mathrm{T}}$ and $-Q(t,\lambda) := A(t,\lambda)^{\mathrm{T}}P(t,\lambda) + P(t,\lambda)A(t,\lambda) + \dot{(t,\lambda)}$. Furthermore, $\exists p_m, q_m, p_M$ and $q_M > 0$ such that, for all $(t,\lambda) \in \mathbb{R}_{\geqslant 0} \times D$, $p_m I \leqslant P(t,\lambda) \leqslant p_M I$ and $q_m I \leqslant Q(t,\lambda) \leqslant q_M I$.

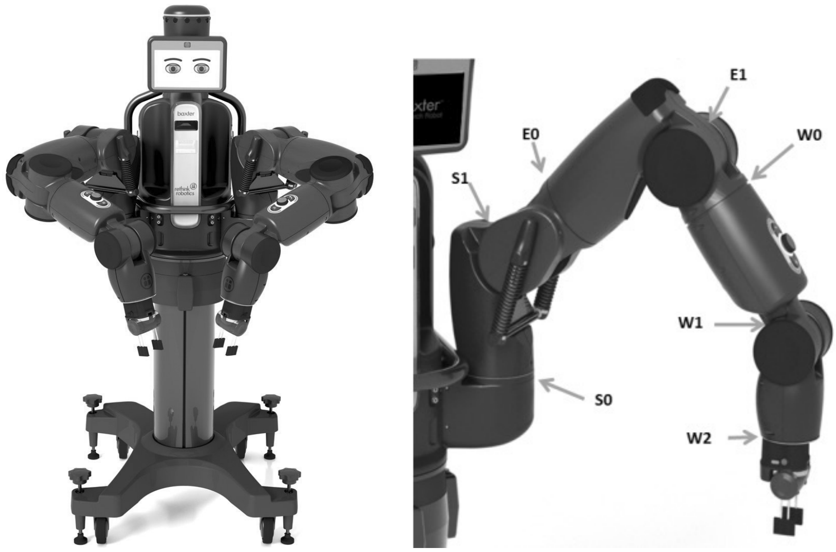
Then, the system is λ-uniformly globally exponentially stable (λ-UGES) if and only if $B(\dot{s},\dot{s})$ is λ-uniformly persistency of excitation (λ-uPE), and the in-bound constants are independent of the initial conditions λ.

3. Kinematics Modeling of Humanoid Baxter® Robot Arms

3.1. *Dual arms workspace identification for humanoid Baxter® robot*

Baxter® robot is a humanoid robot with an identical pair of seven degree of freedom (DOF) manipulators installed. Each manipulator has seven rotational joints and eight links as shown in Fig. 1(a). The joint naming of arm was displayed in Fig. 1(b).

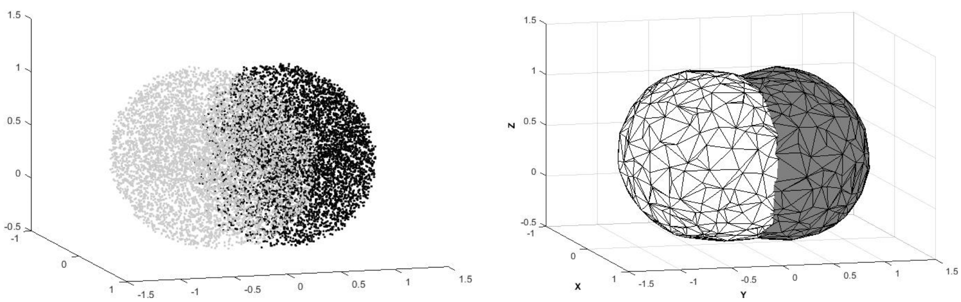
Baxter robot's kinematic model together with DH parameters and joint rotation limits were discussed from our previous work.[19] It is essential to estimate the robot manipulator workspace for optimized robotic design and algorithm. In this paper, the previous method used on a single arm[19] is extended to both arms to calculate the reachable workspace. 6000 randomly chosen points in the joint space for each arm



(a) Baxter robot arm (b) Baxter robot arm joint naming

Fig. 1. Baxter humanoid robot and its joint naming. S0 - Shoulder Roll, S1 - Shoulder Pitch, E0 - Elbow Roll, E1 - Elbow Pitch, W0 - Wrist Roll, W1 - Wrist Pitch, W2 - Wrist Roll.

were generated by using homogenous radial distribution. Then, point clouds of the reachable workspace for both manipulators were generated based on the end-effector positions calculated with forward kinematics, as illustrated in Fig. 2(a). Furthermore, Delaunay triangulation is applied to the point cloud to generated a convex hull of the joint space, as illustrated in Fig. 2(b). These are used to constrain the individual workspace for left and right arms independently in order to let them co-operate more efficiently while control.



(a) The point cloud of reachable workspace of Baxter robot arms

(b) The convex hull of reachable workspace of Baxter robot arms

Fig. 2. The identification of Baxter's workspace.

4. Setup of Stereo Vision Sensor

4.1. *System structure overview*

The robot control system is shown in Fig. 3. The ZED stereo camera is a passive depth camera consists of two RGB-cameras with fixed alignment. It is used as the visual sensors in the robotic control system. It captures videos in 30 fps under 1280×720 resolution to produce dense colored depth maps for estimating the positions of objects. In experiments, ZED keeps capturing videos of objects by its two sensors and sends them to a client computer via an USB 3.0 cable. Based on the difference between two videos, client computer constructs disparity maps where the 3D position information of objects can be read. Then, the target object's position information will be sent to the Sever Computer via UDP packets. Sever computer will receive and decode them and then command Baxter to follow the target object along a reference trajectory.

4.2. *Stereo camera calibration*

Raw pictures captured by ZED are distorted because lenses in ZED introduce non-linear lens distortion deviating from the simple pin-hole model. To solve this problem, camera parameters calibration is necessary. The aim is to find out the camera parameters such as the intrinsic, extrinsic and distortion. Usually researchers used a 2D checker-board pattern to evaluated them, avoiding complexity of 3D reference models and high cost of precise calibration objects. In our work, these parameters are provided by the manufacturer, we can employ them directly.

After we completed the camera parameters calibration, undistorted pictures can be captured from ZED. Then, we can get object's co-ordinates in ZED coordinate system. However, in practice, the position of objects is presented in Baxter coordinate system rather than ZED. Therefore, we need to transform the ZED coordinates into the Baxter coordinates, i.e., the position calibration is necessary. The transform equation is shown as

$$T\begin{bmatrix} X_1 & X_2 & \dots & X_i \\ Y_1 & Y_2 & \dots & Y_i \\ Z_1 & Z_2 & \dots & Z_i \\ 1 & 1 & \dots & 1 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \dots & x_i \\ y_1 & y_2 & \dots & y_i \\ z_1 & z_2 & \dots & z_i \\ 1 & 1 & \dots & 1 \end{bmatrix}, \tag{3}$$

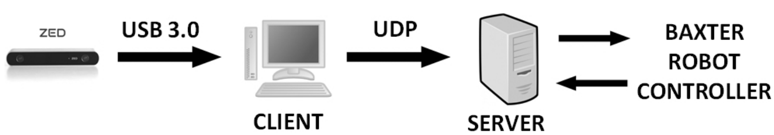


Fig. 3. Communication network.

where T is the transform matrix. (X_i, Y_i, Z_i) means coordinates in ZED and (x_i, y_i, z_i) means coordinates in Baxter. The aim of position calibration is to form the co-ordinate transform matrix T. T can be achieved by

$$T = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} X_1 & X_2 & X_3 & X_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \\ Z_1 & Z_2 & Z_3 & Z_4 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \in R^{4\times4}, \tag{4}$$

where (x_i, y_i, z_i) and (X_i, Y_i, Z_i), $i = 1, 2, 3, 4$, are four non-coplanar point coordinates in the robot coordinate system and the ZED coordinate system, respectively.

To measure coordinates in Baxter coordinate system, the most simple way is to use rulers. However, it is very coarse because the origin of the Baxter coordinate system is inside Baxter's body which is unavailable. Furthermore, it is also hard to ensure the horizontality and verticality of the ruler. Another way to measure coordinates is to use the kinematics of Baxter. At first some established reference coordinates are given and then we command Baxter's end-effector to move to these positions by using kinematics. In this way, we can get the end-effector's coordinates without direct measurement. Then, we use ZED to measure the end-effector's coordinates in ZED's coordinate system, which will be introduced in the next section. In this way, the points' coordinates in both Baxter coordinate system and ZED in Eq. (4) are easily achieved.

However, when using kinematics, stochastic errors always exist. In order to reduce these errors, least squares method is employed. The aim of this algorithm is to calculate an overall solution which minimizes the sum of the square errors in given data. In order to employ this method in the calibration, we must transform Eq. (3) into the form of Eq. (6). The transform can be done as below:

$$\begin{bmatrix} X_1 I_4 & Y_1 I_4 & Z_1 I_4 & I_4 \\ X_2 I_4 & Y_2 I_4 & Z_2 I_4 & I_4 \\ \vdots & \vdots & \vdots & \vdots \\ X_n I_4 & Y_n I_4 & Z_n I_4 & I_4 \end{bmatrix} \begin{bmatrix} T_{c1} \\ T_{c2} \\ T_{c3} \\ T_{c4} \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \\ \vdots \\ x_n \\ y_n \\ z_n \\ 1 \end{bmatrix}, \tag{5}$$

where $I_4 \in R^{4\times4}$ means identity matrix. $T_{ci} \in R^{4\times1}$ means the column vector in the transform matrix T.

Let $A = \begin{bmatrix} X_1 I_4 & Y_1 I_4 & Z_1 I_4 & I_4 \\ X_2 I_4 & Y_2 I_4 & Z_2 I_4 & I_4 \\ \vdots & \vdots & \vdots & \vdots \\ X_n I_4 & Y_n I_4 & Z_n I_4 & I_4 \end{bmatrix}$, $\mathrm{X} = \begin{bmatrix} T_{c1} \\ T_{c2} \\ T_{c3} \\ T_{c4} \end{bmatrix}$ and $B = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \\ \vdots \\ x_n \\ y_n \\ z_n \\ 1 \end{bmatrix}$, we can rewrite (5) into

$$AX = B \tag{6}$$

while A is a known matrix with dimension of $4n \times 16$. X represents the transformation matrix T with dimension of 16×1. B is a column vector with dimension of $4n \times 1$. In most cases, this equation has no solution. However, we can compute the least square solution of it by the following approach. Initially, Eq. (6) is transformed as below:

$$A^T A X = A^T B \tag{7}$$

If $A^T A$ is nonsingular, the transformation matrix can be calculated as below:

$$X = (A^T A)^{-1} A^T B \tag{8}$$

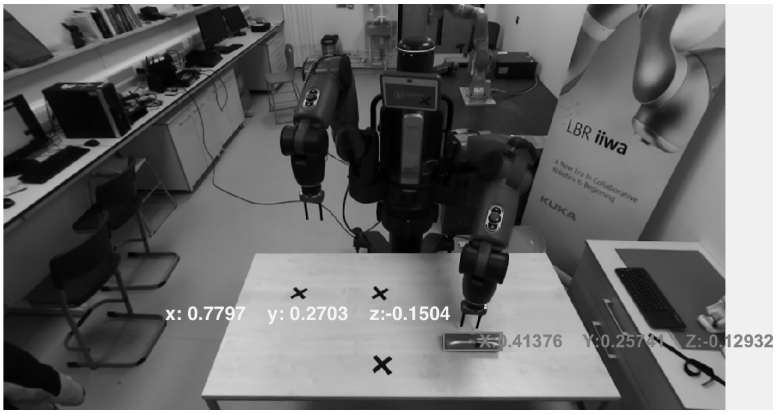
According to Eq. (8), the solution of Eq. (5) can be achieved, i.e., the transform matrix T can be solved by the method of least squares. We can get a more precise solution by completing more coordinates measurement in ZED and Baxter.

Since the robot arms contain red color and green color, they are easily impacted by illumination, a blue object were used for detection. We firstly extracted, the (X_i, Y_i, Z_i), $i = 1, 2, 3, 4$ of the object's centroid from four different positions, out of ZED camera, as the black XYZ shown in Fig. 4(a). The end-effector's position (x_i, y_i, z_i), $i = 1, 2, 3, 4$ were recorded simultaneously. The end-effector were posed 10 cm behind the object's centroid, in order to follow the object while not block the object from camera view, as the white xyz shown in Fig. 4(a).

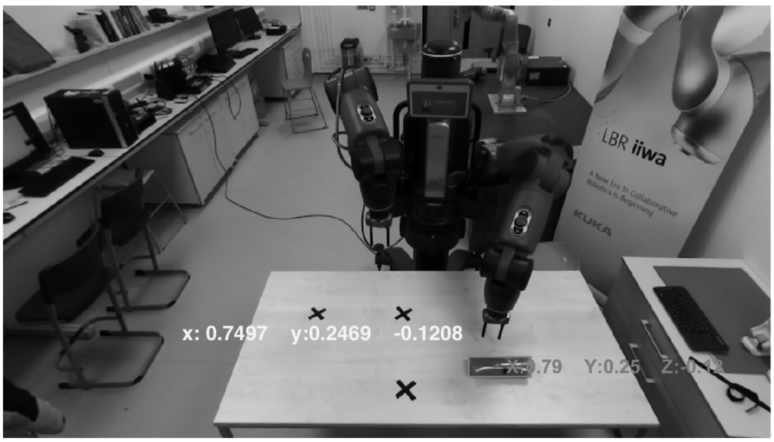
Then we substituted (x_i, y_i, z_i) and (X_i, Y_i, Z_i), $i = 1, 2, 3, 4$ into Eq. (5) to get the transformation matrix T. T will be applied to the object's centroid position, and the data will be send to robot as reference coordinates for following the object. The result was shown in Figs. 4(b) and 5, black XYZ stands for object's reference coordinates and white xyz stands for the coordinates that robot end-effector actually followed.

4.3. *Theory of depth measurement in ZED*

Both pictures captured under active ambient lighting by the ZED stereo camera are aligned utilizing the camera intrinsics and are amended for distortion. In this way, the undistorted images will be stereo rectified to adjust both the projection planes' epipolar lines and guarantee comparable pixels' presence in a predetermined row of the image. The pictures acquired are then frontal paralleled and are estimated correspondingly. The fundamental and the essential frameworks are figured by utilizing Epipolar geometry. There are seven parameters in the fundamental matrix representing



(a) Before calibration



(b) After calibration

Fig. 4. Positions of the object and the end-effector, left image used for displaying and monitoring. Black XYZ: object's coordinates under camera's frame of reference. White xyz: end-effector's coordinates under robot's frame of reference.

two images' pixel relations, three for two image planes' homography and two for each epipole. The essential matrix has five parameters in a 3×3 matrix, three of them are the rotation values between the camera projection planes and two for translation. Then, the epipolar lines were adjusted and the epipoles was moved to infinity. Figure 6(a) delineates the results of stereo correction with row adjusted pixels.

The definition of variables utilized underneath is given in Table 1. Stereo correspondence is a technique for coordinating pixels with comparative surface texture over two co-planar picture planes. The separation between the columns of these splendidly coordinated pixels is characterized as $d = x_l - x_r$.

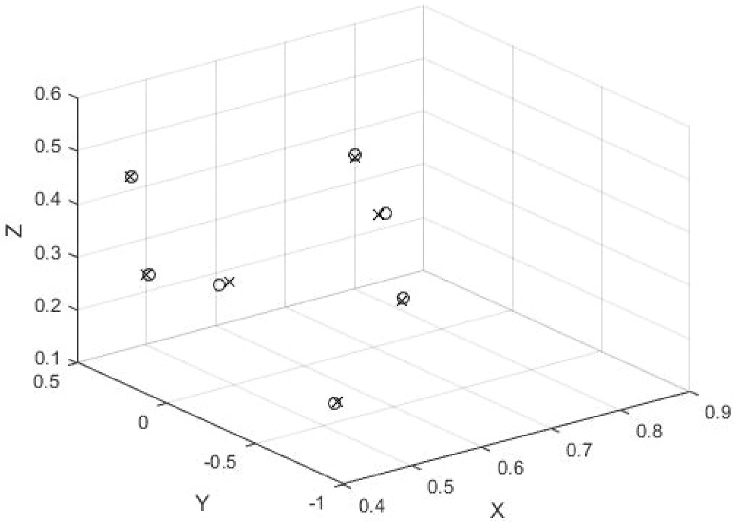


Fig. 5. Precision of calibration. Cross mark: Object's position. Circle mark: End-effector's position.

(a) Rectified stereo images

(b) Disparity map

Fig. 6. Stereo images and 3D reconstruction.

Block matching is actualized for assessing the image correspondence. With the use of sum of absolute differences (SAD), a 15-pixel window block is used to discover the matching results. Considering computational load, the disparity range is selected as low as [0 40] to match the low texture difference of the experiment environment.

Table 1. Definition of variables.

1	x_l	Column value of left image pixel
2	x_r	Column value of right image pixel
3	D	Depth (mm)
4	B	Baseline (mm)
5	f	Focal length (mm)
6	d	Disparity
7	P	Projection matrix
8	X/ω, Y/ω, Z/ω	3D World co-ordinates

In order to get a more complete outcome, Semi Global method is used to drive the disparity values to the neighboring pixels.[17] The output of disparity map is illustrated in Fig. 6(b). Disparity can be calculated by the Triangulation equation $D = B\frac{f}{d}$. It is inversely proportional to the depth of the pixel. Bouguets algorithm is used to obtain the Cartesian co-ordinates from the reconstruction of the image, and the equation is

$$P[x, y, d, 1]^T = [X, Y, Z, \omega]^T \tag{9}$$

where $\omega \neq 1$ is the homogeneous component.

5. Detection and Localization of Target Object

5.1. *Color object detection*

Color-based segmentation is utilized in order to isolate a single color object from the captured image. One approach is to convert the entire RGB frame into corresponding Hue-Saturation-Value (HSV) plane and concentrate the pixel values of the color you want to detect. By using this method, you may be able to detect almost every single distinguishable colors in a frame. However, implementing this approach in live video is challenging because of ambient light. An alternative approach was used in this paper in view of our previous work,[6] to convert the captured image into $L^*a^*b^*$ color space where the value of "a" and "b" is related to the color information of a point.

During the experiments, all images are converted into $L^*a^*b^*$ color space and the variance between every point's color and the standard color marks will be calculated. The estimations are selected based on the minimum variance value of each images. Furthermore, intersection of the diagonals was used to calculate the centroid and Harris corner detector was used to calculate the corners of the object. According to the centroid point in the image, the object's coordinates in ZED is then extracted from the images. By applying the transformation matrix in Sec. 4.2, the object's coordinates in Baxter's coordinate system can be calculated. Figure 4(b) demonstrates the calculated centroid of the object after co-ordinate transformation in robot co-ordinates.

5.2. *Object detection regulation*

In experiments, we find that because of the nonuniform distribution of light in space, object's color in images keeps changing as the object moves. Sometimes the value of "a" and "b" changes a lot that it affects the stability of object detection. To solve this problem, we employed a regulation algorithm in object detection. The algorithm is described below. (i) Calculate the variance between the image points' color and the color marks. (ii) If the value of the variance of the object is not so large, go back to (i) and continue next detection. Conversely, go to (iii). (iii) Calculate the average value of "a" and "b" around the centroid points, and update the older color marks with the new value. Then, start next detection based on these new color marks.

By employing the algorithm above, object detection becomes more stable and more adapted to the environment.

6. Neural Network Controller Design

6.1. *Adaptive neural controller*

According to our previous work,[26] an adaptive NN-based controller is designed to achieve the following control of the joint space trajectory. The dynamic equation of the manipulator shows

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \boldsymbol{G}(\boldsymbol{\theta}) + \boldsymbol{\tau}_{\mathrm{ext}} = \boldsymbol{\tau}, \tag{10}$$

where $\boldsymbol{M}(\boldsymbol{\theta})$ is the manipulator inertia matrix, $\boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ is the Coriolis matrix for the manipulator, $\boldsymbol{G}(\boldsymbol{\theta})$ is the gravity terms and $\boldsymbol{\tau}_{\mathrm{ext}}$ denotes the external torque including the payload gravity applied at the end-effector.

Define $\boldsymbol{s} = \dot{e}_{\boldsymbol{\theta}} + \boldsymbol{\Lambda}\boldsymbol{e}_{\boldsymbol{\theta}}$, $\boldsymbol{v} = \dot{\boldsymbol{\theta}}_{\boldsymbol{d}} - \boldsymbol{\Lambda}\boldsymbol{e}_{\boldsymbol{\theta}}$, where $\boldsymbol{e}_{\boldsymbol{\theta}} = \boldsymbol{\theta} - \boldsymbol{\theta}_{\boldsymbol{d}}$, $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$. Then, the dynamic equation (10) can be rewritten as

$$\boldsymbol{M}(\boldsymbol{\theta})\dot{\boldsymbol{s}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\boldsymbol{s} + \boldsymbol{F} = \boldsymbol{\tau}, \tag{11}$$

where $F \in R^n$ is defined as

$$\boldsymbol{F} = \boldsymbol{M}(\boldsymbol{\theta})\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\boldsymbol{v} + \boldsymbol{G}(\boldsymbol{\theta}) + \boldsymbol{\tau}_{\mathrm{ext}}. \tag{12}$$

Design the adaptive controller as

$$\boldsymbol{\tau} = \hat{\boldsymbol{F}} - \boldsymbol{K}\boldsymbol{s}, \tag{13}$$

where $\hat{\boldsymbol{F}}$ is the estimate of $\boldsymbol{F}$, and $\boldsymbol{K} = \mathrm{diag}\{k_i\}, i = 1, 2, \ldots, n$ is a diagonal matrix and $\min\{k_i\} > 0.5$.

Then, by substituting (13) into (11), the closed-loop dynamics of the robot system can be written as (14).

$$\boldsymbol{M}(\boldsymbol{\theta})\dot{\boldsymbol{s}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\boldsymbol{s} = \tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{S}(\boldsymbol{z}) - \epsilon(\boldsymbol{z}) - \boldsymbol{K}\boldsymbol{s}. \tag{14}$$

The following function approximation method is used.

$$\begin{aligned} \boldsymbol{F} &= \boldsymbol{W}^{*\mathrm{T}}\boldsymbol{S}(\boldsymbol{z}) + \boldsymbol{\epsilon}(\boldsymbol{z}), \\ \hat{\boldsymbol{F}} &= \hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{S}(\boldsymbol{z}), \\ \tilde{\boldsymbol{F}} &= \hat{\boldsymbol{F}} - \boldsymbol{F} = \tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{S}(\boldsymbol{z}) - \epsilon(\boldsymbol{z}), \\ \tilde{\boldsymbol{W}} &= \hat{\boldsymbol{W}} - \boldsymbol{W}^*, \end{aligned} \tag{15}$$

where $\boldsymbol{W}^* = [W_1^*, W_2^*, \ldots, W_n^*] \in R^{N\times n}$ is the weight matrix, $\boldsymbol{S}(\boldsymbol{z})$ is the basis function vector, $\boldsymbol{z} \in \Omega_z \subset R^q$ is the input vector with $\Omega_z \subset R^q$ being a compact set, N is the number of NN node, and $\boldsymbol{\epsilon}(\boldsymbol{z})$ is the approximation error.

$\boldsymbol{s}(\boldsymbol{z}) = [s_1(\|z-\mu_1\|), \ldots, s_N(\|z-\mu_N\|)]^T$, is the regressor vector, with $s_i(\dot{s})$ being a radial basis function, and μ_i $(i = 1, \ldots, N)$ being the center. The Gaussian functions choose as

$$s_i(\|z-\mu_i\|) = \exp\left[\frac{-(z-\mu_i)^T(z-\mu_i)}{\varsigma^2}\right], \tag{16}$$

where $\mu_i = [\mu_{i1}, \mu_{i2}, \ldots, \mu_{iq}]^T \in R^q$ represents the center of each receptive field and ς is the variance.

Choose the following Lyapunov function

$$\boldsymbol{V} = \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{M}(\theta)\boldsymbol{s} + \frac{1}{2}\mathrm{tr}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{Q}\tilde{\boldsymbol{W}}), \tag{17}$$

where $\boldsymbol{Q}$ is a positive definite weight matrix. And using the skew symmetry[1] of the matrix $\dot{M} - 2C$, the first derivative of V can be calculated as

$$\dot{V} = -\boldsymbol{s}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{s} - \boldsymbol{s}^{\mathrm{T}}\epsilon(\boldsymbol{z}) + \boldsymbol{tr}[\tilde{\boldsymbol{W}}^{\mathrm{T}}(\boldsymbol{S}(\boldsymbol{z})\boldsymbol{s}^{\mathrm{T}} + \boldsymbol{Q}\dot{\hat{\boldsymbol{W}}})]. \tag{18}$$

The update law is designed as

$$\dot{\hat{\boldsymbol{W}}} = -\boldsymbol{Q}^{-1}(\boldsymbol{S}(\boldsymbol{z})\boldsymbol{s}^{\mathrm{T}} + \boldsymbol{\sigma}\hat{\boldsymbol{W}}), \tag{19}$$

where $\boldsymbol{\sigma}$ is a pre-designed positive constant.

Substituting (19) into (18), we have

$$\dot{\boldsymbol{V}} = -\boldsymbol{s}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{s} - \boldsymbol{s}^{\mathrm{T}}\epsilon(\boldsymbol{z}) - \sigma\boldsymbol{tr}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\hat{\boldsymbol{W}}). \tag{20}$$

Based on Young's inequality, from (20) we can have

$$\dot{\boldsymbol{V}} \leq -\left(\boldsymbol{\lambda}_{\min}(\boldsymbol{K}) - \frac{\mathbf{1}}{\mathbf{2}}\right)\|\boldsymbol{s}\|^2 - \frac{\sigma}{2}\|\tilde{\boldsymbol{W}}\|^2 + \rho, \tag{21}$$

where $\boldsymbol{\rho} = \frac{1}{2}\boldsymbol{\varepsilon}^{\mathbf{2}} + \frac{\sigma}{2}\|\boldsymbol{W}^*\|^2$, with $\boldsymbol{\varepsilon}$ is the upper limit of $\|\boldsymbol{\epsilon}\|$ over Ω. If $\tilde{\boldsymbol{W}}$ and $\boldsymbol{s}$ satisfy the following inequality

$$\left(\boldsymbol{\lambda}_{\min}(\boldsymbol{K}) - \frac{\mathbf{1}}{\mathbf{2}}\right)\|\boldsymbol{s}\|^{\mathbf{2}} + \frac{\boldsymbol{\sigma}}{\mathbf{2}}\|\tilde{\boldsymbol{W}}\|^{\mathbf{2}} \geq \boldsymbol{\rho} \tag{22}$$

where I is the unit matrix, then we can have $\dot{\boldsymbol{V}} \leq \mathbf{0}$.

By using LaSalle's theorem, we see that $\|\tilde{\boldsymbol{W}}\|$ and $\|\boldsymbol{s}\|$ will converge to an invariant set $\Omega_s \subseteq \Omega$, on which $\dot{V}(t) = 0$, where Ω is the bounding set that is defined as

$$\Omega = \left\{(\|\tilde{\boldsymbol{W}}\|, \|\boldsymbol{s}\|)\middle|\frac{\sigma}{2\rho}\|\tilde{\boldsymbol{W}}\|^2 + \frac{(2K-I)}{2\rho}\|\boldsymbol{s}\|^2 \leq 1\right\}. \tag{23}$$

6.2. *Analysis of NN learning convergence*

By denoting a new subscript ζ, it represents the region which is close to the tracking trajectory, and $\bar{\zeta}$ represents the region which is far away from the tracking trajectory. Let $S_\zeta(z)$ be the element that the neurons located in the region of ζ, and $\hat{W}_\zeta$ is the associated weight matrix of NN. From (19), we can have

$$\dot{\hat{W}}_{\zeta} = -Q_{\zeta}^{-1}(S_{\zeta}(z)s^{\mathrm{T}} + \sigma_{\zeta}\,\hat{W}_{\zeta}) \tag{24}$$

and from (15) we have that the NN approximation error $\epsilon_{i\zeta}(z)$ is close to $\epsilon(z)$.

$\bar{S}_{\zeta}$ and $\bar{\hat{W}}_{\zeta}$ are defined as

$$\bar{S}_{\zeta} = \begin{bmatrix} S_{\zeta} & 0_{[N_{\zeta}\times 1]} & \cdots & 0_{[N_{\zeta}\times 1]} \\ 0_{[N_{\zeta}\times 1]} & S_{\zeta} & \cdots & 0_{[N_{\zeta}\times 1]} \\ \vdots & \vdots & \vdots & \vdots \\ 0_{[N_{\zeta}\times 1]} & \cdots & 0_{[N_{\zeta}\times 1]} & S_{\zeta} \end{bmatrix} \in R^{nN_{\zeta}\times n} \tag{25}$$

and

$$\bar{W}_{\zeta} = [\,W_{1\zeta}^{T}, W_{2\zeta}^{T}, \ldots, W_{n\zeta}^{T},]^{\,T} \in R^{nN_{\zeta}}. \tag{26}$$

Subsequently, we define an augmented matrix of the diagonal matrix σ_{ζ} as $\bar{\sigma}_{\zeta} = [\sigma_{\zeta}, \sigma_{\zeta}, \ldots, \sigma_{\zeta}] \in R^{N_{\zeta}\times N_{\zeta}}$. From this, we could rewrite (24) into:

$$\dot{\bar{\hat{W}}}_{\zeta} = -\bar{S}_{\zeta}(z)Q_{\zeta}^{-1}s^{\mathrm{T}} - Q_{\zeta}^{-1}\bar{\sigma}_{\zeta}\,\hat{W}_{\zeta}. \tag{27}$$

Using the spatially localized approximation ability of RBF NN, the closed-loop system from (14) can be expressed as

$$\dot{s} = M^{-1}(\theta)[-Ks + \bar{S}_{\zeta}(z)\bar{\tilde{W}}_{\zeta}^{\mathrm{T}} - \epsilon_{\zeta}(z) - C(\theta,\dot{\theta})s]. \tag{28}$$

Then, a LTV system can be created from the system of (28) and (27) as

$$\begin{bmatrix} i \\ \dot{\tilde{W}}_{\zeta i} \end{bmatrix} = \begin{bmatrix} -M^{-1}(\theta)N(t) & M^{-1}(\theta)\bar{S}_{\zeta i}^{\,T}(z) \\ -Q_i^{-1}\bar{S}_{\zeta i}(z) & 0_{[N_{\zeta}\times N_{\zeta}]} \end{bmatrix} \begin{bmatrix} s_i \\ \bar{\tilde{W}}_i \end{bmatrix} + \begin{bmatrix} -M^{-1}(\theta)\epsilon_i(z) \\ -Q_i^{-1}\sigma_i\,\hat{W}_i \end{bmatrix}, \tag{29}$$

where $N(t) = k_i + C(\theta,\dot{\theta}),\ i = 1,2,\ldots,n$. Let $P = Q_i^{-1}M(\theta)$, which is symmetric, and let $\mathcal{A} = -M^{-1}(\theta)N(t)$, $\mathcal{B} = M^{-1}(\theta)\bar{S}_{\zeta i}^{\,T}(z)$, and $\mathcal{C} = Q_i^{-1}\bar{S}_{\zeta i}(z)$, then we have

$$\mathcal{A}^{T}P + P\mathcal{A} + \dot{P} = Q_i^{-1}(\dot{M}(\theta) - 2C(\theta,\dot{\theta}) - 2K) := U. \tag{30}$$

Since $\min k_i > 0.5$, Q_i is positive, and using the skew symmetry[1] of the matrix $\dot{M} - 2C$, such that we can have $U < 0$. This guarantees the exponential stability of the nominal part of the system (29). Then on the premise of small enough σ, the parameter error $\tilde{W}_{\zeta}$ will converge exponentially to a small neighborhood (determined by $|\epsilon_{\zeta}(z)|$ and $\|-\sigma_{\zeta}\,\hat{W}_{\zeta}\|$) of zero for all $t > T_1$. Thus, $\hat{W}_{\zeta}$ can converge exponentially to a small neighborhood of the desired weight value W_{ζ}^{*} for all $t > T_1$.

6.3. *Knowledge reusing and skill transfer*

Now, we can accurately approximate the dynamical system $F(z)$ by using the localization feature of RBFNN, with the convergence of $\hat{W}_{\zeta}$ such as

$$F(z) = \bar{W}_{\zeta}^{\,T}S_{\zeta}(z) + \bar{\epsilon}_{\zeta}(z), \tag{31}$$

where $\bar{\epsilon}(z)$ is close to $\epsilon(z)$ in the steady-state process, and

$$\bar{W}_{\zeta} = \text{mean}_{t\in[t_{ai},t_{bi}]}\, \hat{W}_{\zeta}(t) = \frac{1}{t_{bi} - t_{ai}} \int_{t_{ai}}^{t_{bi}} \hat{W}_{\zeta}(s)\,ds \tag{32}$$

with $[t_{ai}, t_{bi}]$, $t_{bi} > t_{ai} > T_1$ representing a time segment after the transient process.

Let us define

$$\bar{W} = \text{mean}_{t\in[t_{ai},t_{bi}]}\, \hat{W}(t) = \frac{1}{t_{bi} - t_{ai}} \int_{t_{ai}}^{t_{bi}} \hat{W}(s)\,ds \tag{33}$$

we will have

$$\hat{W}^{T} S_{\bar{\zeta}}(z) \approx \bar{W}_{\zeta}^{T} S_{\zeta}(z). \tag{34}$$

Therefore, we could use $\bar{W}_{\zeta}^{T} S_{\zeta}(z)$ to replace $\bar{W}_{i}^{T} S_{i}(z)$ for approximating the uncertainties of system dynamics $F(z)$.

Since the learnt knowledge will not keep in the memory, the control parameters have to be recalculated even when reproduce the similar control tasks. However, since the estimate $\hat{W}$ is able to converge into a small neighborhoods of the optimal W^*, the $F(z)$ which is the accurate approximation of the system dynamics can be still achieved. The above learning method can be considered as approximate the system dynamics using constant NN weights.

Based on our previous work,[4] the following control law is proposed to reuse the learnt knowledge instead of using the original NN based controller (13) and the updated law of RBFNN's weight (19)

$$\tau = -Ks + \bar{F}(z), \tag{35}$$

where $K = \text{diag}\{k_i\}, i = 1, 2, \ldots, n, \min\{k_i\} > 0.5$ and $\bar{F}(z) = \bar{W}^{T} S(z)$.

With the property of dual-arm, once one side of arm learnt the uncertainties of environment, i.e., payload, the learned knowledge can also be transferred and reused on another arm, without readapting the uncertainties. This feature can also be extended to robot to robot skill transfer. While performing same tasks, this mechanism can greatly help to reduce computational load with guaranteed performance.

7. Experiment Studies

A visual tracking task was performed to test the proposed VS method, with neural learning and without neural learning for comparison. The experiment setup is shown in Fig. 7. In each set of tests, the blue object was moved by operator from the starting point ($\boldsymbol{P_1} : [0.7, -0.2, -0.2]$) to the end point ($\boldsymbol{P_2} : [0.7,\ 0.2, -0.2]$) in a rectangle trajectory. The object was lifted up after leaving the starting point and generally put down on the operating table level at the end.

Due to the 7-DOF robot dynamics, $N = 3^7 \times 7$ nodes are employed for the NN to complete a high precision of approximation. While the NN's weight matrix is initialized

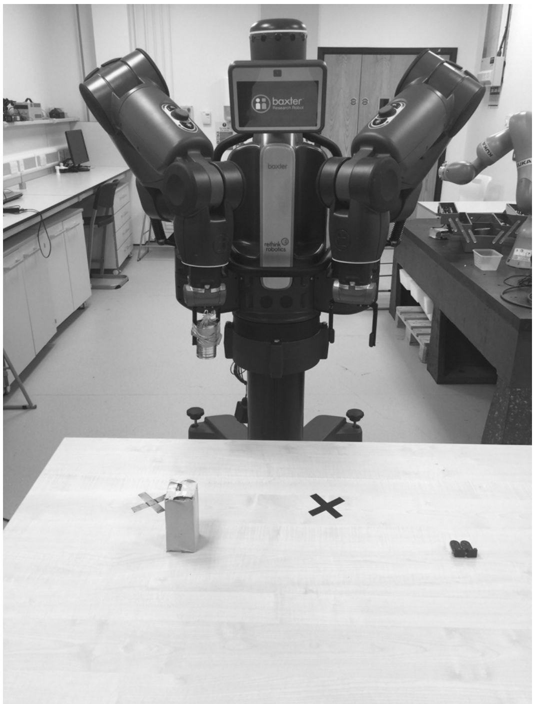


Fig. 7. The experiment setup. Left cross: start point. Right cross: end point. Two different payload was held in both grippers on the manipulator. The right and left one each weighs 1.3 kg and 0.7 kg, respectively.

as $\hat{W}(0) = 0 \in R^{15309\times 7}$. The design parameters K of the controller are specified as $K = \mathrm{diag}\{9, 9, 8, 4.5, 1.8, 1.2, 0.8\}$.

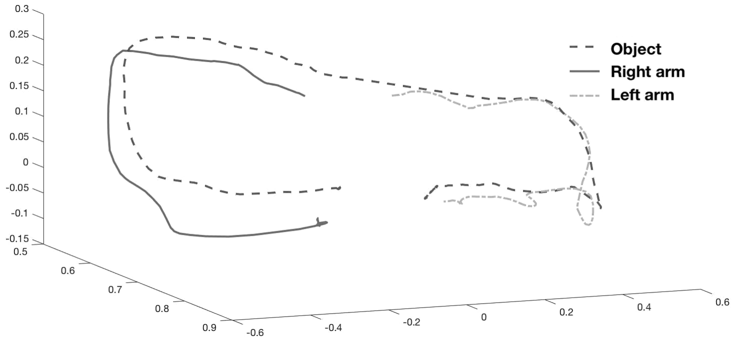
The object reference trajectories which has been recoded using MATLAB and the end-effector trajectories of this set of comparative experiments are demonstrated in Fig. 8. The NN learning weights of individual joints are demonstrated in Fig. 9. The compensation torques obtained by NN of each joint are shown in Fig. 10.

7.1. *Control without NN learning*

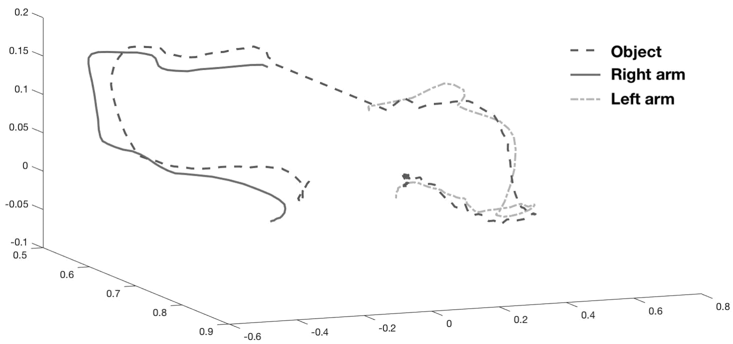
During this initial set of experiments, the performance of the control method without NN learning is tested to establish baseline performance. The color object was held by the operator and was moved along a predefined trajectory as introduced earlier. From Fig. 8(a), we can see the actual position trajectory is below the reference trajectory because of the heavy payload.

7.2. *Control with NN learning*

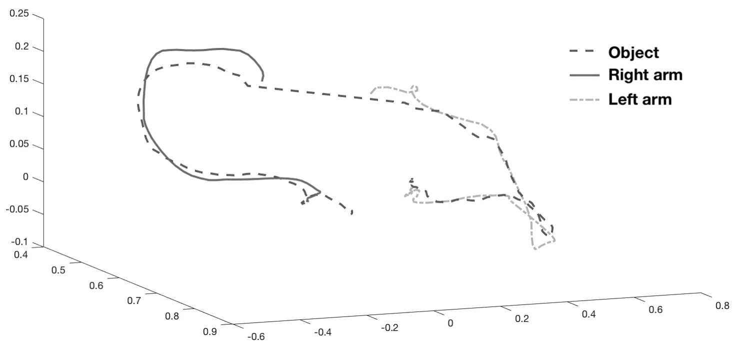
During this set of experiments, the same task as the first experiment was performed. In this set, the NN learning was added to the controller and the performance of the



(a) Without NN

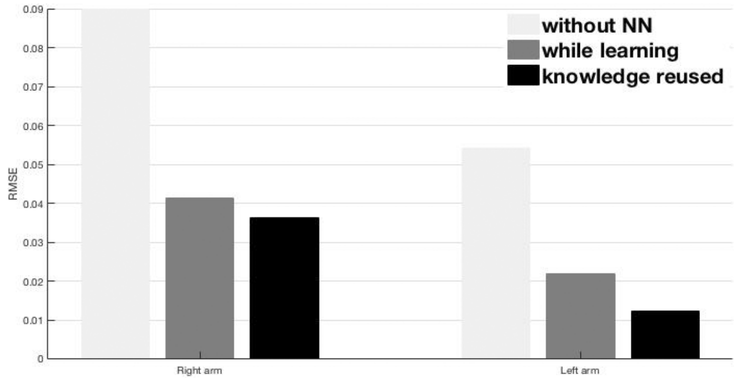


(b) Trajectory while learning



(c) Trajectory after learning reused

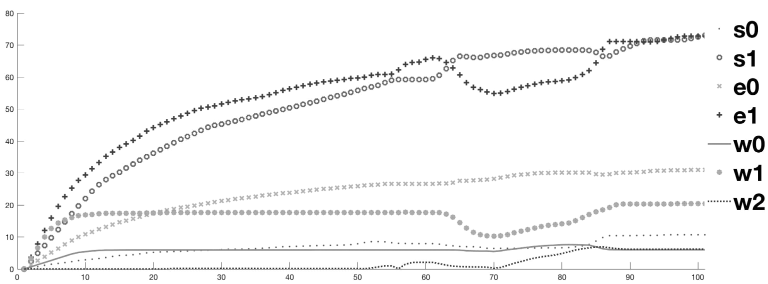
Fig. 8. Tracking trajectory and root-mean-square error (RMSE). (a)–(c) Dashed line: reference trajectories generated by object tracking. Solid and Dash–dot lines: actual position trajectories of both robot right and left manipulators, respectively. (d) Left: RMSE of right arm under three different conditions. Right: RMSE of left arm under three different conditions.



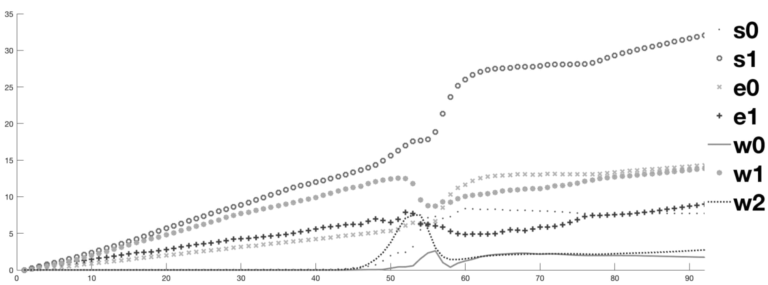
(d) RMSE

Fig. 8. (*Continued*)

telerobot manipulator was recorded. Compared with the first test, NN is learning the payload's weight during teleoperation, and affects the control inputs. As can be seen from Fig. 8(b), the robot was able to restore to normal tracking position. The control torque inputs of right and left arms are shown in Figs. 10(a) and 10(b).

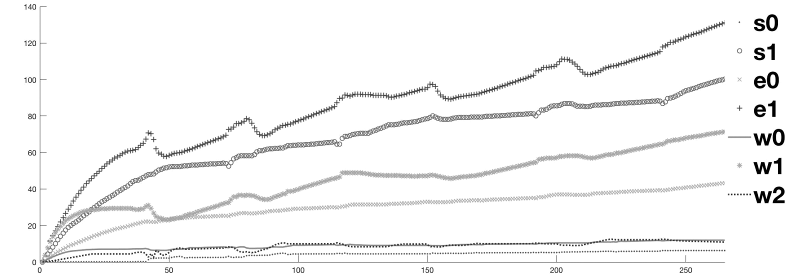


(a) NN learning weights for each single joint of right arm while learning

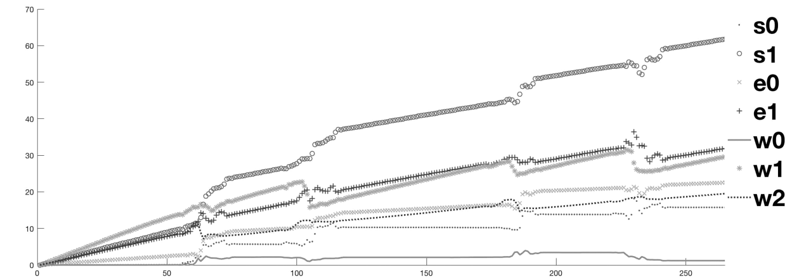


(b) NN learning weights for each single joint of left arm while learning

Fig. 9. NN learning weights for each single joint.

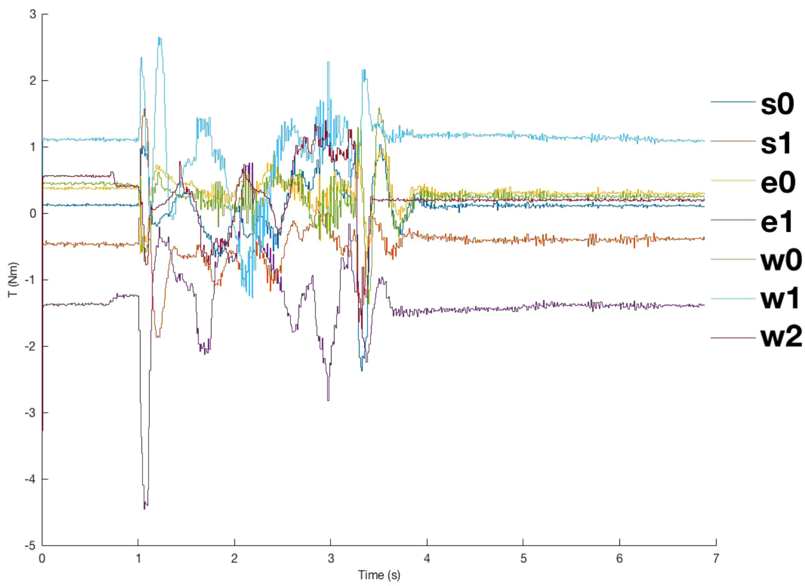


(c) NN learning weights for each single joint of right arm while learning reused



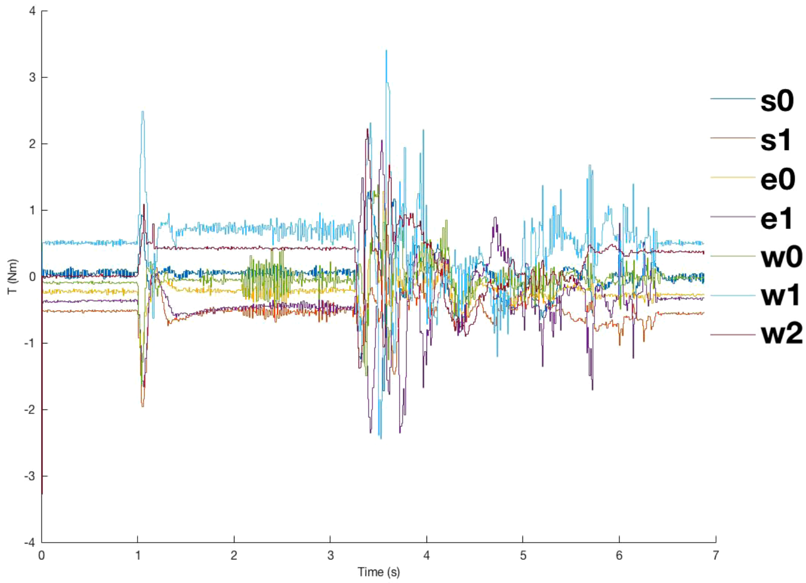
(d) NN learning weights for each single joint of left arm while learning reused

Fig. 9. (*Continued*)

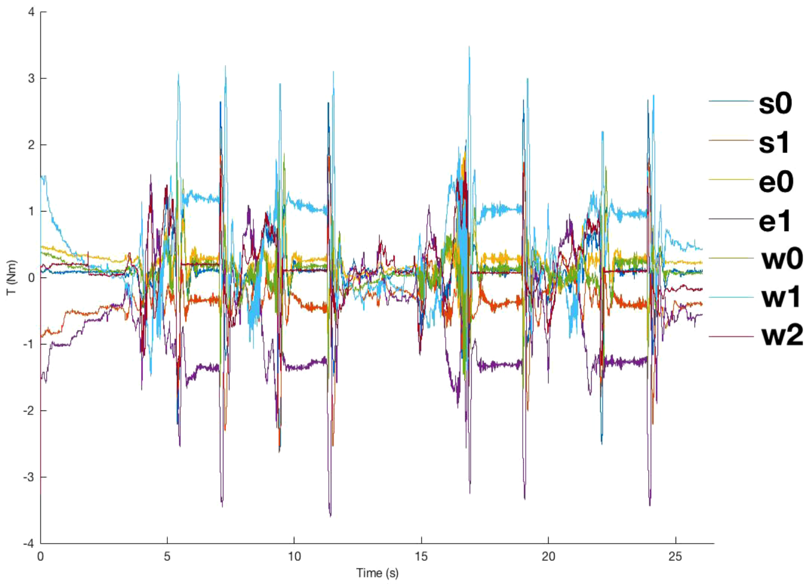


(a) The compensation torque of right arm obtained by NN while learning

Fig. 10. NN control torques for each single joint of both arms.

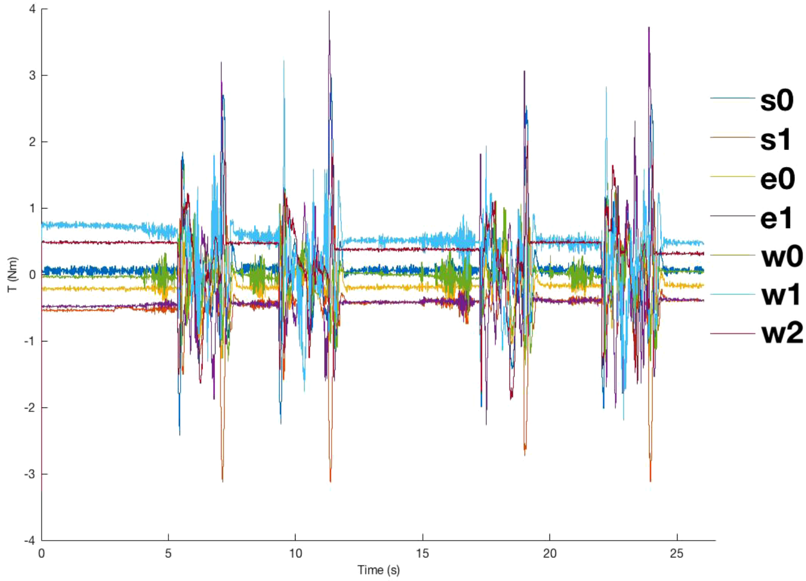


(b) The compensation torque of left arm obtained by NN while learning



(c) The compensation torque of right arm obtained by the NN after training

Fig. 10. (*Continued*)



(d) The compensation torque of left arm obtained by the NN after training

Fig. 10. (*Continued*)

7.3. *Control after NN learning*

During the last set of experiments, the NN will first learn the dynamics while both manipulators tracking the object along a repeated trajectory, same as previous two. After four cycles, the NN was adapted with the external dynamics (attached payload). So that the trained NN will be reused for the further teleoperation. The control torque inputs of right and left arms are shown in Figs. 10(c) and 10(d). The performance of tracking is illustrated in Fig. 8(c).

From Fig. 8(d), it can be seen that the designed adaptive controller can help system compensate tracking error from both internal and external dynamics. The trained NN has a steady performance with reusing the trained knowledge to increase tracking performance.

8. Conclusion

An NN learning enhanced VS control method was developed in this paper and implemented on a humanoid dual-arm Baxter robot. The color object was detected by a stereo camera and an regulation algorithm was applied to ensure the effectiveness of detection. The calibration between camera and robot's coordinates was done with the proposed least squared-based method to reduce stochastic errors.

The dynamic parameters of the manipulator are estimated by the radial basis function NN and an improved adaptive control method is designed for compensating the effect of uncertain payload and other uncertainties during the dynamic control of the robot. Specifically, a knowledge reuse method with skill transfer feature has been created to increase the neural learning efficiency. So that the learned NN knowledge can be easily reused for finishing repetitive tasks and also can be transferred to another arm for performing the same task. The proposed NN controller was validated with tests on a Baxter humanoid robot, and can realize optimal performance of the designed VS control.

Acknowledgment

This work was supported in part by UK EPSRC Grants EP/L026856/2 and Royal Society Newton Mobility Grant IE150858.

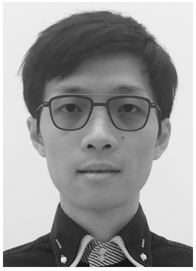
References

1. B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, *Robotics: Modelling, Planning and Control*, Springer Science Business Media (Springer, 2010).
2. S. Hutchinson and F. Chaumette, Visual servo control, part i: Basic approaches, *IEEE Robot. Automat. Mag.* **13**(4) (2006) 82–90.
3. S. Hutchinson and F. Chaumette, Visual servo control, part ii: Advanced approaches, *IEEE Robot. Automat. Mag.* **14**(1) (2007) 109–118.
4. S. L. Dai, M. Wang and C. Wang, Neural learning control of marine surface vessels with guaranteed transient tracking performance, *IEEE Trans. Indust. Electron.* **63**(3) (2016) 1717–1727.
5. A. Loria and E. Panteley, Uniform exponential stability of linear time-varying systems: Revisited, *Syst. Control Lett.* **47**(1) (2002) 13–24.
6. C. Yang, S. Amarjyoti, X. Wang, Z. Li, H. Ma and C. Y. Su, Visual servoing control of baxter robot arms with obstacle avoidance using kinematic redundancy, in *Int. Conf. Intelligent Robotics and Applications* (Springer International Publishing, 2015), pp. 568–580.
7. C. Wang and D. J. Hill, *Deterministic Learning Theory for Identification, Recognition, and Control*, Vol. 32 (CRC Press, 2009).
8. C. Wang and D. J. Hill, Deterministic learning and rapid dynamical pattern recognition, *IEEE Trans. Neural Networks* **18**(3) (2007) 617–630.
9. Z. Xue, C. Wang and T. Liu, December. Deterministic learning and robot manipulator control, in *IEEE Int. Conf. Robotics and Biomimetics (ROBIO)* (2007), pp. 1989–1994.
10. C. Wang and T. Chen, Rapid detection of small oscillation faults via deterministic learning, *IEEE Trans. Neural Netw.* **22**(8) (2011) 1284–1296.
11. P. Liang, C. Yang, Z. Li and R. Li, Writing skills transfer from human to robot using stiffness extracted from semg, in *IEEE Int. Conf. Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)* (2015), pp. 19–24.
12. M. Ralph and M. A. Moussa, Toward a natural language interface for transferring grasping skills to robots, *IEEE Trans. Robot.* **24**(2) (2008) 468–475.
13. F. Chaumette and S. Hutchinson, *Visual Servoing and Visual Tracking* (Springer, Berlin, 2008).

14. C. Dune, E. Marchand and C. Leroux, One click focus with eye-in-hand/eye-to-hand cooperation, in *IEEE Int. Conf. Robotics and Automation* (2007), pp. 2471–2476.
15. E. Guizzo and E. Ackerman, *How Rethink Robotics Built its New Baxter Robot Worker* (IEEE Spectrum, 2012).
16. W. He, Y. Chen and Z. Yin, Adaptive neural network control of an uncertain robot with full-state constraints, *IEEE Trans. Cybernetics* **46**(3) (2015) 1.
17. H. Hirschmuller, Accurate and efficient stereo processing by semi-global matching and mutual information, in *IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR)*, Vol. 2 (2005), pp. 807–814.
18. S. Hutchinson, G. D. Hager and P. I. Corke, A tutorial on visual servo control, *IEEE Trans. Robot. Automat.* **12**(5) (1996) 651–670.
19. Z. Ju, C. Yang and H. Ma, Kinematics modeling and experimental verification of baxter robot, in 33*rd Chinese Control Conf. (CCC)* (2014), pp. 8518–8523.
20. A. A. Maciejewski and C. A. Klein, Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments, *Int. J. Robot. Res.* **4**(3) (1985) 109–117.
21. J. Stavnitzky and D. Capson, Multiple camera model-based 3-d visual servo, *IEEE Trans. Robot. Automat.* **16**(6) (2001) 732–739.
22. K. P. Tee, S. S. Ge, R. Yan and H. Li, Adaptive control for robot manipulators under ellipsoidal task space constraints, in *IEEE/RSJ Int. Conf. on Intelligent Robots Systems* (2012), pp. 1167–1172.
23. X. Wang, C. Yang and H. Ma, Automatic obstacle avoidance using redundancy for shared controlled telerobot manipulator, in *IEEE 5th Annual Int. Conf. Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)* (2015), pp. 1338–1343.
24. B. Xu, C. Yang and Y. Pan, Global neural dynamic surface tracking control of strict-feedback systems with application to hypersonic flight vehicle, *IEEE Trans. Neural Netw. Learn. Syst.* **26**(10) (2015) 2563–2575.
25. L. Zlajpah and B. Nemec, Kinematic control algorithms for on-line obstacle avoidance for redundant manipulators, in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vol. 2 (2002), pp. 1898–1903.
26. C. Yang, J. Chen and L. Cheng, Neural learning enhanced teleoperation control of robots with uncertainties, in *9th Int. Conf. Human System Interactions (HSI)* (2016), pp. 223–228.

Chenguang Yang received the B.Eng. degree in measurement and control from Northwestern Polytechnical University, Xi'an, China, in 2005, and the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010. He received postdoctoral training at Imperial College London, UK. He is a senior lecturer with Zienkiewicz Centre for Computational Engineering, Swansea University, UK.

His research interests lie in robotics, automation and computational intelligence.



Junshen Chen received the B.Eng. degree in Electronic Engineering and Automation from Shanghai University of Electric Power, Shanghai, China, in 2012, and the MSc degree in Robotics from Plymouth University, Plymouth, UK, in 2014. He is currently pursuing the Ph.D. degree with Swansea University, Swansea, UK.

His current research interests include robotics, automation and computational intelligence.

Zhaojie Ju received the B.S. degree in automatic control and the M.S. degree in intelligent robotics both from the Huazhong University of Science and Technology, Hubei, China, in 2005 and 2007, respectively, and the Ph.D. degree in intelligent robotics at the University of Portsmouth, U.K., in 2010. He is currently a Senior Lecturer in the School of Computing, University of Portsmouth.

He previously held research appointments in the Department of Computer Science, University College London and Intelligent Systems and Biomedical Robotics group, University of Portsmouth, U.K. His research interests include machine intelligence, robot learning, pattern recognition and their applications in robotic/prosthetic hand control, and human–robot interaction.

Andy SK Annamalai is currently a Lecturer at the University of Highlands and Islands (UHI), Scotland, UK. He has secured a research grant from Scottish informatics and computer science alliance to pursue research in improving tourism based on geoinformatics systems. He is a member of the Academic titles review board at UHI and actively contributes to the Learning and teaching quality committee.

Prior to his tenure at UHI, He was a researcher at the Marine and Industrial Dynamic Analysis Research Group, Plymouth University, Devon, United Kingdom. His other research concerns the design and development of adaptive autopilots for marine robots (for the Royal Navy and US Navy). Guidance, navigation and control systems traditiused in Missile systems and petrochemical industries were adapted to suit a low cost marine platform. His paper titled "Innovative Adaptive Autopilot Design for Uninhabited Surface Vehicles" was selected for a best paper award at the 25th IET Signals & Systems Conference 2014. Additionally, he won the NASA innovative solutions competition for the UK Southwest region, 2013. His current research interests include system identification and extraction of model parameters and the development of automated decision making systems. He obtained his Masters by research in Communications

August 8, 2017 8:54:46pm WSPC/191-IJHR 1750023 ISSN: 0219-8436 Page Proof

Engineering and Signal Processing at Plymouth, UK. His research focused on designing sonic data acquisition systems where a communication link was designed between a towed array of 10,000 sensors and a submarine (for the Royal Navy). Additionally, he gained significant experience in the industry (Space technology, military technology, electronics, IT) and academia.