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 utilised 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 has gained increasing research attention, recently. VS is a technique of control using computer vision information to control the motion of a robot. It mainly depends on techniques of computer vision, image processing and control theory. It is of great importance in improving the flexibility of robot control systems 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. 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.

*Corresponding author: Chenguang Yang (cyang@theiet.org)  
This work was supported in part by UK EPSRC Grants EP/L026856/2 and Royal Society Newton Mobility Grant IE150858.
To improve robot arm's control performance, an adaptive controller was developed for robot manipulators. 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. 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.

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. Deterministic learning is proposed by using deterministic calculations that began from adaptive control, rather than utilising 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, learning and control of robotics, and oscillation faults diagnosis. 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 visual servoing 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 5: Consider a parameterized linear time-varying (LTV) multivariable systems in the following form:
\[
\begin{bmatrix}
\dot{e} \\
\dot{\theta}
\end{bmatrix} =
\begin{bmatrix}
A(e, \lambda) & B(e, \lambda)^T \\
-C(t, \lambda) & 0
\end{bmatrix}
\begin{bmatrix}
e \\
\theta
\end{bmatrix},
z := \begin{bmatrix}
e \\
\theta
\end{bmatrix}
\]

(1)

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

There exists a constant \( \phi_M > 0 \) such that for all \( t \geq 0 \) and for all \( \lambda \in D \),
\[
\max\left\{ \|B(t, \lambda)\|, \left\| \frac{\partial B(t, \lambda)}{\partial t} \right\| \right\} \leq \phi_M.
\]

(2)

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

Then, the system is \( \lambda \)-uniformly globally exponentially stable (\( \lambda \)-UGES) if and only if \( B(\cdot, \cdot) \) is \( \lambda \)-uniformly persistency of excitation (\( \lambda \)-uPE), and the in-bound constants are independent of the initial conditions \( \lambda \).

3. Kinematics Modelling 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 7 degree of freedom (DOF) manipulators installed. Each manipulator has 7 rotational joints and 8 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 optimised robotic design and algorithm. In this paper, the previous method used on a single arm\(^{19} \) is extended to both arms to calculated the reachable workspace. 6000 randomly chosen points in the joint space for each arm 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 arm independently in order to let them co-operate more efficiently while control.

4. Setup of Stereo Vision Sensor

4.1. System Structure Overview

The robot control system is shown in the 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 coloured depth maps for estimating the positions of objects. In experiments, ZED keeps capturing videos of objects by its 2 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 3-D 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.

![Communication Network](image)

Fig. 3. Communication Network

### 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 equation (3).

\[
T \begin{bmatrix} X_1 & X_2 & \ldots & X_i \\ Y_1 & Y_2 & \ldots & Y_i \\ Z_1 & Z_2 & \ldots & Z_i \\ 1 & 1 & \ldots & 1 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \ldots & x_i \\ y_1 & y_2 & \ldots & y_i \\ z_1 & z_2 & \ldots & z_i \\ 1 & 1 & \ldots & 1 \end{bmatrix}
\]

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 the equation (4).

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

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

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’s 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 equation (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 minimises the sum of the square errors in given data. In order to employ this method in the calibration, we must transform equation (3) into the form of equation (6). The transform can be done as below:

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

where \(I_4 \in \mathbb{R}^{4 \times 4}\) means identity matrix. \(T_{ci} \in \mathbb{R}^{4 \times 1}\) means the column vector in the transform matrix \(T\).

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

\[
AX = B
\] (6)

while \(A\) is a known matrix with dimension of \(4n \times 16\). \(X\) represents the transformation matrix \(T\) with dimension of \(16 \times 1\). \(B\) 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, equation (6) is transformed as below:

\[
A^TAX = A^TB
\] (7)
If $A^T A$ is nonsingular, the transformation matrix can be calculated as below:

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

According to the equation (8), the solution of equation (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 colour and green colour 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 10cm 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 equation (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 were shown in Fig. 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 utilising 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 utilising Epipolar geometry. There are 7 parameters in the Fundamental matrix representing two images’ pixel relations, three for two image planes’ homography and two for each epipole. The Essential matrix has 5 parameters in a $3 \times 3$ 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. Fig. 6(a) delineates the results of stereo correction with row adjusted pixels.

The definition of variables utilised 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 characterised as $d = x_l - x_r$.

Block matching is actualised 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 low as $[0 \ 40]$ to match the low texture difference of the experiment environment.
In order to get a more complete outcome, Semi Global method is used to drive the disparity values to the neighbouring pixels\textsuperscript{17}. The output of disparity map is
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.

illustrated in Fig. 6(b). Disparity can be calculated by the Triangulation equation $D = B_d f$. It is inversely proportional to the depth of the pixel. Bougues’s algorithm is used to obtain the Cartesian co-ordinates from the reconstruction of the image, and the equation is shown below (9).

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

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

5. Detection and Localization of Target Object

5.1. Colour object detection

Colour based segmentation is utilised in order to isolated a single colour 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 colour you want to detect. By using this method, you may be able to detect almost every single distinguishable colours in a frame. However, implementing this approach in live video is challenging because of ambient light. An alternative ap-
proach was used in this paper in view of our previous work\textsuperscript{6}, to convert the captured image into L*a*b* colour space where the value of ‘a’ and ‘b’ is related to the colour information of a point.

During the experiments, all images are converted into L*a*b* colour space and the variance between every point’s colour and the standard colour 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 section 4.2, the object’s coordinates in Baxter’s coordinate system can be calculated. Fig. 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 colour in images keeps changing as the object moves. Sometimes the value of ‘a’ and ‘b’ change 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’ colour and the colour 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 colour marks with the new value. Then start next detection based on these new colour 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\textsuperscript{26}, an adaptive NN based controller is designed to achieve the following control of the joint space trajectory. The dynamic equation of the manipulator is shown in (10).

\[
M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) + \tau_{ext} = \tau, \tag{10}
\]

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

Define \(s = \dot{e}_\theta + \Lambda e_\theta, v = \dot{\theta}_d - \Lambda e_\theta\), where \(e_\theta = \theta - \theta_d, \Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_n)\). Then, the dynamic equation (10) can be rewritten as (11).

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

where \(F \in \mathbb{R}^n\), is defined as
\[ F = M(\theta)\dot{\theta} + C(\theta, \dot{\theta})v + G(\theta) + \tau_{ext} \] (12)

Design the adaptive controller as (13).
\[ \tau = \hat{F} - Ks \] (13)

where \( \hat{F} \) is the estimate of \( F \), and \( K = \text{diag}\{k_i\}, i = 1, 2, \cdots, n \) 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).
\[ M(\theta)s + C(\theta, \dot{\theta})s = \tilde{W}^T S(z) - \epsilon(z) - Ks \] (14)

The following function approximation method is used.
\[ F = W^*^T S(z) + \epsilon(z) \]
\[ \hat{F} = \hat{W}^T S(z) \]
\[ \tilde{F} = \hat{F} - F = \tilde{W}^T S(z) - \epsilon(z) \]
\[ \hat{W} = \tilde{W} - W^* \] (15)

where \( W^* = [W_1^*, W_2^*, \cdots, W_n^*] \in R^{N \times n} \) is the weight matrix, \( S(z) \) is the basis function vector, \( z \in \Omega_z \subset R^n \) is the input vector with \( \Omega_z \subset R^q \) being a compact set, \( N \) is the number of NN node, and \( \epsilon(z) \) is the approximation error. \( s(z) = [s_1(\|z - \mu_1\|), \cdots, s_N(\|z - \mu_N\|)]^T \), is the regressor vector, with \( s_i(\cdot) \) being a radial basis function, and \( \mu_i \) \((i = 1, \cdots, N)\) being the center. The Gaussian functions choose as
\[ s_i(\|z - \mu_i\|) = \exp \left[ \frac{-(z - \mu_i)^T(z - \mu_i)}{\varsigma^2} \right] \] (16)

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

Choose the following Lyapunov function.
\[ V = \frac{1}{2}s^T M(\theta)s + \frac{1}{2}\text{tr}(\hat{W}^T Q \hat{W}) \] (17)

where \( Q \) 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
\[ \dot{V} = -s^T Ks + s^T \epsilon(z) + \text{tr}\left[ \hat{W}^T \left( S(z)s^T + Q \hat{W} \right) \right] \] (18)

The update law is designed as follow.
\[ \dot{\hat{W}} = -Q^{-1}(S(z)s^T + \sigma \hat{W}) \] (19)

where \( \sigma \) is a pre-designed positive constant.

Substituting (19) into (18), we have
\[ \dot{V} = -s^T Ks - s^T \epsilon(z) - \sigma \text{tr}(\hat{W}^T \hat{W}) \] (20)
Based on Young’s inequality, from (20) we can have
\[ \dot{V} \leq -\left( \lambda_{\text{min}}(K) - \frac{1}{2} \right) \|s\|^2 - \frac{\sigma}{2} \|\hat{W}\|^2 + \rho \]  
where \( \rho = \frac{1}{2} \varepsilon^2 + \frac{\sigma}{2} \|W^*\|^2 \), with \( \varepsilon \) is the upper limit of \( \|\epsilon\| \) over \( \Omega \). If \( \hat{W} \) and \( s \) satisfy the following inequality
\[ (\lambda_{\text{min}}(K) - \frac{1}{2}) \|s\|^2 + \frac{\sigma}{2} \|\hat{W}\|^2 \geq \rho \]  
where \( I \) is the unit matrix, then we can have \( \dot{V} \leq 0 \).

By using LaSalle’s theorem, we see that \( \|\hat{W}\| \) and \( \|s\| \) will converge to an invariant set \( \Omega_s \subseteq \Omega \), on which \( \dot{V}(t) = 0 \), where \( \Omega \) is the bounding set that is defined as
\[ \Omega = \left\{ \left( \|\hat{W}\|, \|s\| \right) \mid \frac{\sigma}{2} \|\hat{W}\|^2 + \frac{(2K - I)}{2\rho} \|s\|^2 \leq 1 \right\} . \]

6.2. Analysis of NN Learning Convergence

By denoting a new subscript \( \zeta \), it represents the region which is close to the tracking trajectory, and \( \tilde{\zeta} \) 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 \( \zeta \), and \( \hat{W}_{\zeta} \) is the associated weight matrix of NN. From (19) we can have
\[ \dot{\hat{W}}_{\zeta} = -Q_{\zeta}^{-1}(S_{\zeta}(z)s^T + \sigma_{\zeta}\hat{W}_{\zeta}) \]  
and from (15) we have that the NN approximation error \( \epsilon_{\zeta}(z) \) is close to \( \epsilon(z) \).

\( \tilde{S}_{\zeta} \) and \( \tilde{W}_{\zeta} \) are defined as below:
\[ \tilde{S}_{\zeta} = \begin{bmatrix} S_{\zeta} & 0_{[N_{\zeta} \times 1]} & \cdots & 0_{[N_{\zeta} \times 1]} \\ 0_{[N_{\zeta} \times 1]} & S_{\zeta} & \cdots & 0_{[N_{\zeta} \times 1]} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{[N_{\zeta} \times 1]} & \cdots & 0_{[N_{\zeta} \times 1]} & S_{\zeta} \end{bmatrix} \in \mathbb{R}^{N_{\zeta} \times n} \]  
and
\[ \tilde{W}_{\zeta} = [W_{\zeta}^T, W_{2\zeta}^T, \ldots, W_{N_{\zeta}}^T]^T \in \mathbb{R}^{n_{\zeta}} \]  
Subsequently, we define an augmented matrix of the diagonal matrix \( \sigma_{\zeta} \) as \( \tilde{\sigma}_{\zeta} = [\sigma_{\zeta}, \sigma_{\zeta}, \ldots, \sigma_{\zeta}] \in \mathbb{R}^{N_{\zeta} \times N_{\zeta}} \). From this, we could rewrite (24) into:
\[ \dot{\hat{W}}_{\zeta} = -\tilde{S}_{\zeta}(z)Q\zeta^{-1}s^T - Q\zeta^{-1}\tilde{\sigma}_{\zeta}\hat{W}_{\zeta} \]  
Using the spatially localized approximation ability of RBF NN, the closed-loop system from (14) can be expressed as below:
\[ \dot{s} = M^{-1}(\theta)[-Ks + \tilde{S}_{\zeta}(z)\tilde{W}_{\zeta}^T - \epsilon_{\zeta}(z) - C(\theta, \dot{\theta})s] \]
Then, a LTV system can be created from the system of (28) and (27) as follow

$$\begin{bmatrix} \dot{s}_j \\ \dot{W}_{\xi i} \end{bmatrix} = \begin{bmatrix} -M^{-1}(\theta)N(t) & M^{-1}(\theta)S^T_{\xi i}(z) \\ -Q^{-1}_{i}S_{\xi i}(z) & 0_{[\mathcal{N_i} \times \mathcal{N_i}]} \end{bmatrix} \begin{bmatrix} s_j \\ \dot{W}_{\xi i} \end{bmatrix} + \begin{bmatrix} -M^{-1}(\theta)\epsilon_i(z) \\ -Q^{-1}_{i}\sigma_i\dot{W}_{\xi i} \end{bmatrix}$$

(29)

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

$$\mathcal{A}^T P + P\mathcal{A} + \dot{P} = Q^{-1}_{i}\left(\dot{M}(\theta) - 2C(\theta, \dot{\theta}) - 2K\right) := U$$

(30)

Since $\min k_i > 0.5$, $Q_i$ is positive, and using the skew symmetry 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 $\sigma$, the parameter error $\dot{W}_\xi$ will converge exponentially to a small neighbourhood (determined by $|\epsilon(z)|$) and $|\sigma_{ \dot{W}_\xi}|$ of zero for all $t > T_1$. Thus, $\dot{W}_\xi$ can converge exponentially to a small neighbourhood of the desired weight value $\dot{W}_\xi^*$ for all $t > T_1$.

### 6.3. Knowledge Reusing and Skill Transfer

Now, we can accurately approximated the dynamical system $F(z)$ by using the localisation feature of RBFNN, with the convergence of $\dot{W}_\xi$ such as

$$F(z) = \dot{W}_\xi^T S_{\xi}(z) + \bar{\epsilon}(z)$$

(31)

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

$$\dot{W}_\xi = \text{mean}_{\epsilon \in [t_{a_i}, t_{b_i}]} \dot{W}_\xi(t) = \frac{1}{t_{b_i} - t_{a_i}} \int_{t_{a_i}}^{t_{b_i}} \dot{W}_\xi(s)ds$$

(32)

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

Let us define

$$\bar{W} = \text{mean}_{\epsilon \in [t_{a_i}, t_{b_i}]} \dot{W}(t) = \frac{1}{t_{b_i} - t_{a_i}} \int_{t_{a_i}}^{t_{b_i}} \dot{W}(s)ds$$

(33)

we will have

$$\dot{W}_\xi^T S_{\xi}(z) \approx \bar{W}_\xi^T S_{\xi}(z)$$

(34)

Therefore, we could use $\dot{W}_\xi^T S_{\xi}(z)$ to replace $\dot{W}_i^T S_i(z)$ 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 $\bar{W}$ is able to converge into a small neighbourhood 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 \((P_1 : [0.7, -0.2, -0.2])\) to the end point \((P_2 : [0.7, 0.2, -0.2])\) 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 initialised as \(\bar{W}(0) = 0 \in R^{15309 \times 7}\). The design parameters \(K\) of the controller are specified as \(K = \text{diag} \{9, 9, 8, 4.5, 1.8, 1.2, 0.8\}\).

The object reference trajectories which has been recored 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 colour object was held by the operator and was moved along a predefined trajectory as introduced earlier. From 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 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 8(b), the robot was able to restore to normal tracking position. The control torque inputs of right and left arm are shown in Fig. 10(a) and 10(b).
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 arm are shown in Fig. 10(c) and 10(d). The performance of tracking is illustrated in 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 colour 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 realise optimal performance of the designed VS control.

References

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, Dr. Annamalai 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 au-
topilots for marine robots (for the Royal Navy and US Navy). Guidance, navigation and control systems traditionally used 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 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.
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.
(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.

(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. NN learning weights for each single joint.
(a) The compensation torque of right arm obtained by NN while learning.

(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.

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

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