21

Robotics Arm Visual Servo: Estimation of Arm-Space Kinematics Relations with Epipolar Geometry

Ebrahim Mattar
Intelligent Control & Robotics, Department of Electrical and Electronics Engineering, University of Bahrain
Kingdom of Bahrain

1. Introduction

1.1 Visual servoing for robotics applications

Numerous advances in robotics have been inspired by reliable concepts of biological systems. Necessity for improvements has been recognized due to lack of sensory capabilities in robotic systems which make them unable to cope with challenges such as anonymous and changing workspace, undefined location, calibration errors, and different alternating concepts. Visual servoing aims to control a robotics system through artificial vision, in a way as to manipulate an environment, in a similar way to humans actions. It has always been found that, it is not a straightforward task to combine "Visual Information" with a "Arm Dynamic" controllers. This is due to different natures of descriptions which defines "Physical Parameters" within an arm controller loop. Studies have also revealed an option of using a trainable system for learning some complicated kinematics relating object features to robotics arm joint space. To achieve visual tracking, visual servoing and control, for accurate manipulation objectives without losing it from a robotics system, it is essential to relate a number of an object's geometrical features (object space) into a robotics system joint space (arm joint space). An object visual data, an play important role in such sense. Most robotics visual servo systems rely on object "features Jacobian", in addition to its inverse Jacobian. Object visual features inverse Jacobian is not easily put together and computed, hence to use such relation in a visual loops. A neural system have been used to approximate such relations, hence avoiding computing object's feature inverse Jacobian, even at singular Jacobian postures. Within this chapter, we shall be discussing and presenting an integration approach that combines "Visual Feedback" sensory data with a "6-DOF robotics Arm Controller". Visual servo is considered as a methodology to control movements of a robotics system using certain visual information to achieve a task. Visionary data is acquired from a camera that is mounted directly on a robot manipulator or on a mobile robot, in which case, motion of the robot induces camera motion. Differently, the camera can be fixed, so that can observe the robot motion. In this sense, visual servo control relies on techniques from image processing, computer vision control theory, kinematics, dynamic and real time computing.

Robotics visual servoing has been recently introduced by robotics, AI and control communities. This is due to the significant number of advantages over blind robotic systems. Researchers have also demonstrated that, VISUAL SERVOING is an effective and a robust framework to control robotics systems while relying on visual information as feedback. An image-based scheme task is said to be completely performed if a desired image is acquired by a robotic system. Numerous advances in robotics have been inspired by the biological systems. In reference to Fig. (1), visual servoing aims to control a robotics system through an artificial vision in a way as to manipulate an environment, comparable to humans actions. Intelligence-based visual control has also been introduced by research community as a way to supply robotics system even with more cognitive capabilities. A number of research on the field of intelligent visual robotics arm control have been introduced. Visual servoing has been classified as using visual data within a control loop, enabling visual-motor (hand-eye) coordination. There have been different structures of visual servo systems. However, the main two classes are; **P**osition-**B**ased **V**isual **S**ervo systems (**PBVS**), and the **I**mage-**B**ased **V**isual **S**ervo systems (**IBVS**). In this chapter, we shall concentrate on the second class, which is the Image-based visual servo systems.

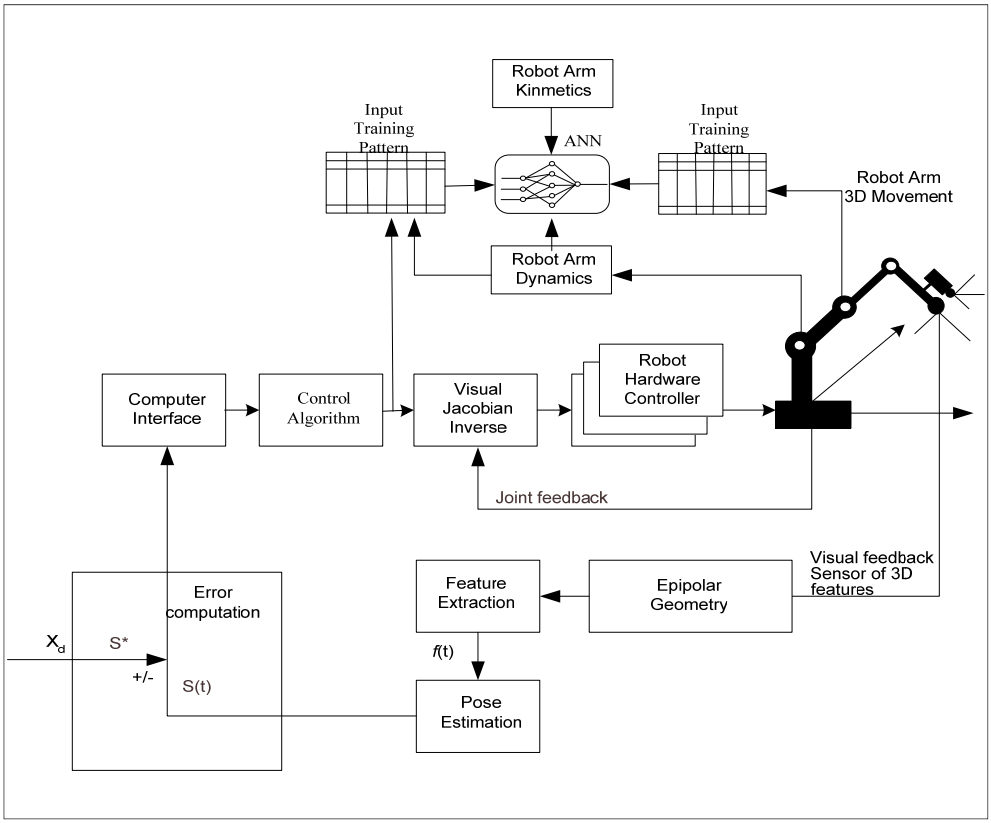


Fig. 1. Overall structure of an ANN based visual servoing

1.2 Literature surveys

EGT, **E**pipolar **G**eometry **T**oolbox, (Eleonora et al., 2004), was also built to grant MATLAB users with a broaden outline for a creation and visualization of multi-camera scenarios. Additionally, to be used for the manipulation of visual information, and the visual geometry. Functions provided, for both classes of vision sensing, the PINHOLE and PANORAMIC, include camera assignment and visualization, computation, and estimation of epipolar geometry entities. Visual servoing has been classified as using visual data within a control loop (Cisneros, 2004), thus enabling visual-motor (Hand-Eye) coordination.

Image **B**ased **V**isual **S**ervoing (**IBVS**) Using Takagi-Sugeno Fuzzy Neural Network Controller has been proposed by (Miao et. al, 2007). In their study, a T-S fuzzy neural controller based IBVS method was proposed. Eigenspace based image compression method is firstly explored which is chosen as the global feature transformation method. Inner structure, performance and training method of T-S neural network controller are discussed respectively. Besides that, the whole architecture of TS-FNNC was investigated.

An Image Based Visual Servoing using Takagi-Sugeno fuzzy neural network controller has been proposed by Miao, (Miao et. al, 2007). In this paper, a TAKAGI-SUGENO Fuzzy Neural Network Controller (TSFNNC) based Image Based Visual Servoing (IBVS) method was proposed. Firstly, an eigenspace based image compression method is explored, which is chosen as the global feature transformation method. After that, the inner structure, performance and training method of T-S neural network controller are discussed respectively. Besides, the whole architecture of the TS-FNNC is investigated.

Panwar and Sukavanam in (Panwar & Sukavanam 2007) have introduced Neural Network Based Controller for Visual Servoing of Robotic Hand Eye System. For Panwar and Sukavanam, in their paper a neural network based controller for robot positioning and tracking using direct monocular visual feedback is proposed. Visual information is provided using a camera mounted on the end-effector of a robotics manipulator. A PI kinematics controller is proposed to achieve motion control objective in an image plane. A Feed forward Neural Network (FFNN) is used to compensate for the robot dynamics. The FFNN computes the required torques to drive the robot manipulator to achieve desired tracking. The stability of combined PI kinematics and FFNN computed torque is proved by Lyapunov theory. Gracia and Perez-Vidal in (Gracia & Perez-Vidal 2009), have presented a research framework through which a new control scheme for visual servoing that takes into account the delay introduced by image acquisition and image processing. The capabilities (steady-state errors, stability margins, step time response, etc.) of the proposed control scheme and of previous ones are analytically analyzed and compared. Several simulations and experimental results were provided to validate the analytical results and to illustrate the benefits of the proposed control scheme. In particular, it was shown that this novel control scheme clearly improves the performance of the previous ones.

Alessandro and researchers, as in (Alessandro et. al. 2007), in their research framework, they proposed an image-based visual servoing framework. Error signals are directly computed from image feature parameters, thus obtaining control schemes which do not need neither a 3-D model of the scene, nor a perfect knowledge of the camera calibration matrix. Value of the depth "Z" for each considered feature must be known. Thus they proposed a method to estimate on-line the value of Z for point features while the camera is moving through the scene, by using tools from nonlinear observer theory. By interpreting "Z" as a continuous unknown state with known dynamics, they build an estimator which asymptotically recovers the actual depth value for the selected feature.

In (Chen et. al. 2008), an adaptive visual servo regulation control for camera-in-hand configuration with a fixed camera extension was presented by Chen. An image-based regulation control of a robot manipulator with an uncalibrated vision system is discussed. To compensate for the unknown camera calibration parameters, a novel prediction error formulation is presented. To achieve the control objectives, a Lyapunov-based adaptive control strategy is employed. The control development for the camera-in-hand problem is presented in detail and a fixed-camera problem is included as an extension. Epipolar Geometry Toolbox as in (Gian et. al. 2004), was also created to grant MATLAB users with a broaden outline for a creation and visualization of multi-camera scenarios. In addition, to be used for the manipulation of visual information, and the visual geometry. Functions provided, for both class of vision sensing, the pinhole and panoramic, include camera assignment and visualization, computation, and estimation of epipolar geometry entities. Visual servoing has been classified as using visual data within a control loop, enabling visual-motor (hand-eye) coordination.
Image Based Visual Servoing Using Takagi-Sugeno Fuzzy Neural Network Controller has been proposed by (Miao et. al. 2007). In their study, a T-S fuzzy neural controller based IBVS method was proposed. Eigenspace based image compression method is firstly explored which is chosen as the global feature transformation method. Inner structure, performance and training method of T-S neural network controller are discussed respectively. Besides that, the whole architecture of TS-FNNC is investigated. For robotics arm visual servo, this issue has been formulated as a function of object feature Jacobian. Feature Jacobian is a complicated matrix to compute for real-time applications. For more feature points in space, the issue of computing inverse of such matrix is even more hard to achieve.

1.3 Chapter contribution

For robotics arm visual servo, this issue has been formulated as a function of object feature Jacobian. Feature Jacobian Matrix entries are complicated differential relations to be computed for real-time applications. For more feature points in space, the issue of computing inverse of such matrix is even more hard to achieve. In this respect, this chapter concentrates on approximating differential visual information relations relating an object movement in space to the object motion in camera space (which usually complicated relation), hence to joint space. This is known as the (object feature points). The proposed methodology will also discuss how a trained learning system can be used to achieve the needed approximation. The proposed methodology is entirely based on utilizing and merging of three MatLab Tool Boxes. Robotics Toolbox developed by Peter Corke (Corke, 2002), secondly is the Epipolar Geometry Toolbox (EGT) developed by Eleonora Alunno (Eleonora et. al. 2004), whereas the third is the ANN MatLab Tool Box.
This chapter is presenting a research framework which was oriented to develop a robotics visual servo system that relies on approximating complicated nonlinear visual servo kinematics. It concentrates on approximating differential visual changes (object features) relations relating objects movement in a world space to object motion in a camera space (usually time-consuming relations as expressed by time-varying Jacobian matrix), hence to a robotics arm joint space.
The research is also presenting how a trained Neural Network can be utilized to learn the needed approximation and inter-related mappings. The research whole concept is based on

utilizing and mergence three fundamentals. The first is a robotics arm-object visual kinematics, the second is the Epipolar Geometry relating different object scenes during motion, and a learning artificial neural system. To validate the concept, the visual control loop algorithm developed by RIVES has been thus adopted to include a learning neural system. Results have indicated that, the proposed visual servoing methodology was able to produce a considerable accurate results.

1.4 Chapter organization

The chapter has been sub-divided into six main sections. In this respect, in section (1) we introduce the concept of robotics visual servo and related background, as related to visual servo. In section (2), we present a background and some related literatures for single scene via signal camera system. Double scene, as known as Epipolar Geometry is also presented in depth in section (3). Artificial Neural Net based **IBVS** is also presented in Section (4), whereas simulated of learning and training of an Artificial Neural Net is presented in section (5). Section (5) presents a case study and a simulation result of the proposed method, as compared to RIVES algorithm. Finally, Section (6) presents the chapter conclusions.

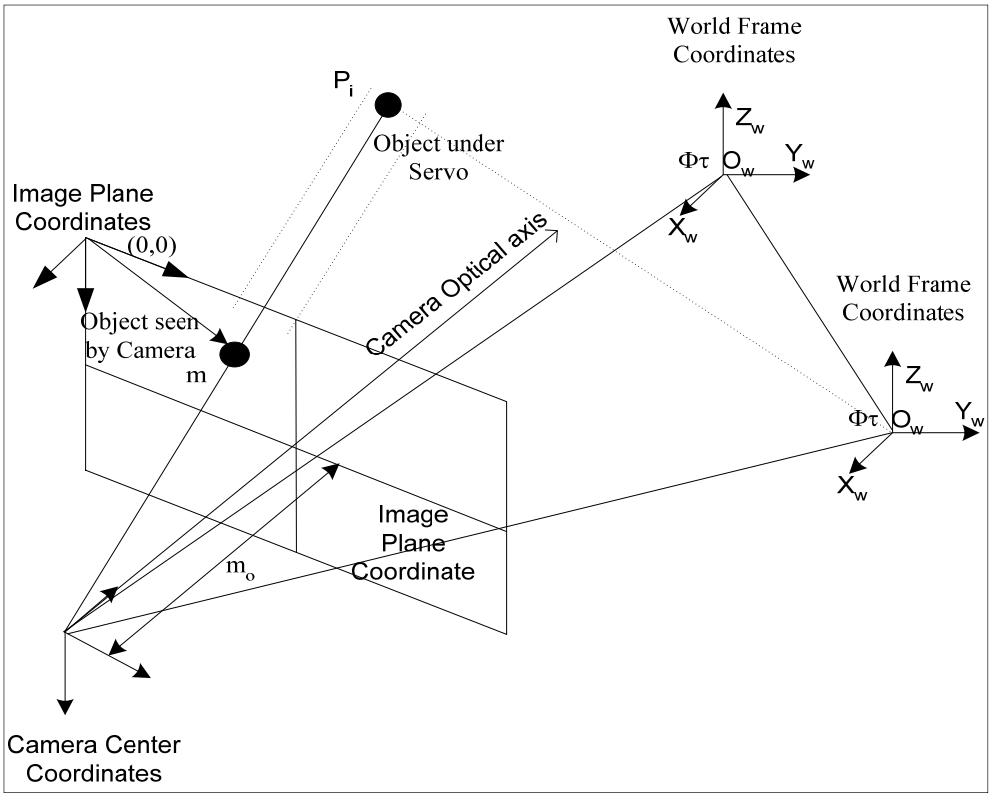


Fig. 2. Camera geometrical representation in a 3-D space

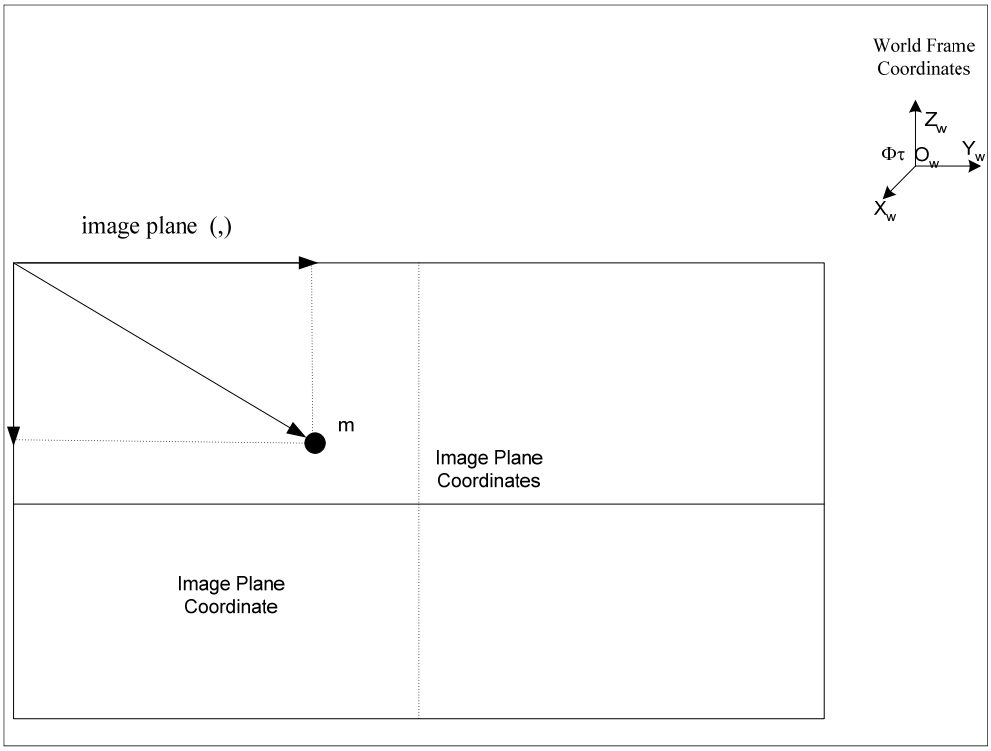


Fig. 3. Camera image plane and geometry

2. Single camera model: {object to camera plane kinmeatics}

Fig. (2) shows a typical camera geometrical representation in a space. Hence, to assemble a closed loop visual servo system, a loop is to be closed around the robotics arm system. In this study, this is a PUMA-560 robotics arm, with a Pinhole camera system. The camera image plane and associated geometry is shown in Fig. (3). For analyzing closed loop visual kinematics, we shall employ a Pinhole Camera Model for capturing object features. For whole representation, details of a Pinhole camera model in terms of image plane (ξ^a, ψ^a) location are expressed in terms (ξ, ψ, and ζ), as in Equ. (1). In reference to Fig. (2), we can express $\left(\xi^a, \psi^a\right)$ locations as expressed in terms (ξ, Ψ, ξ):

$$\begin{cases} \xi^a = \phi^a\left(\dfrac{\xi}{\zeta}\right) \\ \psi^a = \phi^a\left(\dfrac{\psi}{\zeta}\right) \end{cases} \tag{1}$$

Both $\left(\xi^a, \psi^a\right)$ location over an image plane is thus calculated by Equ. (1). For thin lenses (as the Pinhole camera model), camera geometry are thus represented by, (Gian et. al. 2004) :

$$\left(\frac{1}{\phi}\right)=\left(\frac{1}{\zeta^{a}}-\frac{1}{\zeta}\right) \tag{2}$$

In reference to (Gian et. al. 2004), using Craig Notation, denotes coordinate of point P in frame B. For translation case :

$$^{B}P = {}^{A}P + {}^{B}O_{A} \tag{3}$$

$^{B}O_{A}$ is a coordinate of the origin O_{A} of frame "A" in a new coordinate system "B". Rotations are thus expressed :

$$^{B}_{A}R=\left({}^{B}i_{A} \quad {}^{B}j_{A} \quad {}^{B}k_{A}\right)=\begin{pmatrix} {}^{A}i_{B}^{T} \\ {}^{A}j_{B}^{T} \\ {}^{A}k_{B}^{T} \end{pmatrix} \tag{4}$$

In Equ (4), $^{B}i_{A}$ is a frame axis coordinate of "A" in another coordinate "B". In this respect, for rigid transformation we have:

$$^{B}P = {}^{B}_{A}R\,{}^{A}P$$

$$^{B}P = {}^{B}_{A}R\,{}^{A}P + {}^{B}O_{A} \tag{5}$$

For more than single consecutive rigid transformations, (for example to frame "C"), i.e. form frames $A \rightarrow B \rightarrow C$, coordinate of point P in frame "C" can hence be expressed by:

$$^{B}P = {}^{B}_{A}R\left({}^{B}_{A}R\,{}^{A}P + {}^{B}O_{A}\right) + {}^{C}O_{B}$$

$$^{B}P = \left({}^{C}_{B}R\,{}^{B}_{A}R\,{}^{A}P\right) + \left({}^{C}_{B}R\,{}^{A}O_{A} + {}^{C}O_{A}\right) \tag{6}$$

For homogeneous coordinates, it looks very concise to express ^{B}P as :

$$\begin{pmatrix} {}^{B}P \\ 1 \end{pmatrix}=\begin{pmatrix} {}^{B}_{A}R & {}^{B}O_{A} \\ O^{T} & 1 \end{pmatrix}\begin{pmatrix} {}^{A}P \\ 1 \end{pmatrix} \tag{7}$$

$$\begin{pmatrix} {}^{C}P \\ 1 \end{pmatrix}=\begin{pmatrix} {}^{C}_{B}R & {}^{C}O_{B} \\ O^{T} & 1 \end{pmatrix}\begin{pmatrix} {}^{B}P \\ 1 \end{pmatrix} \tag{8}$$

$$\begin{pmatrix} {}^{C}P \\ 1 \end{pmatrix}=\begin{pmatrix} {}^{C}_{B}R & {}^{C}O_{B} \\ O^{T} & 1 \end{pmatrix}\begin{pmatrix} {}^{B}_{A}R & {}^{B}O_{A} \\ O^{T} & 1 \end{pmatrix}\begin{pmatrix} {}^{A}P \\ 1 \end{pmatrix} \tag{9}$$

We could express elements of a transformation (Γ) by writing :

$$\Gamma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \sigma_{34} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_{44} \end{pmatrix} \quad \Gamma = \begin{pmatrix} A & \Omega \\ O^T & 1 \end{pmatrix} \tag{10}$$

as becoming offline transformation. If $(A = R)$, i.e., a rotation matrix (Γ), once $R^T R = I$, then :

$$\Gamma = \begin{pmatrix} R & \Omega \\ O^T & 1 \end{pmatrix} \tag{11}$$

Euclidean transformation preserves parallel lines and angles, on the contrary, affine preserves parallel lines but not angles, Introducing a normalized image plane located at focal length $\phi = 1$. For this normalized image plane, pinhole (C) is mapped into the origin of an image plane using:

$$\hat{P} = \begin{pmatrix} \hat{u} & \hat{v} \end{pmatrix}^T \tag{12}$$

$\hat{\mathrm{C}}$ and P are mapped to : $$\hat{P} = \begin{pmatrix} \hat{u} \\ \hat{u} \\ 1 \end{pmatrix} = \frac{1}{\zeta^C} \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} P^C \\ 1 \end{pmatrix} \tag{13}$$

$$\hat{P} = \frac{I}{\zeta^C} \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} \xi^c \\ \psi^c \\ \zeta^c \\ I \end{pmatrix} \tag{14}$$

we also have : $$\begin{pmatrix} \mathrm{u} \\ \mathrm{v} \\ 1 \end{pmatrix} = \frac{1}{\zeta^{\mathrm{C}}} \begin{pmatrix} \mathrm{k}\phi & 0 & \mathrm{a}_0 \\ 0 & \mathrm{k}\phi & \mathrm{v}_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \xi^{\mathrm{c}} \\ \psi^{\mathrm{c}} \\ \zeta^{\mathrm{c}} \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \left(\frac{1}{\zeta^C} \right) \begin{pmatrix} k\phi & 0 & u_0 \\ 0 & k\phi & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} \xi^c \\ \psi^c \\ \zeta^c \\ 1 \end{pmatrix} \tag{15}$$

Now once $\kappa = k\phi$ and $\beta = L\phi$, then we identify these parameters $\kappa, \beta, \mathrm{u_o}, \mathrm{v_o}$ as intrinsic camera parameters. In fact, they do present an inner camera imaging parameters. In matrix notation, this can be expressed as :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \left(\frac{1}{\zeta^C}\right) \begin{pmatrix} k\phi & 0 & u_0 \\ 0 & k\phi & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \xi^c \\ \psi^c \\ \zeta^c \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \left(\frac{1}{\zeta^C}\right) \begin{pmatrix} \kappa & 0 & u_0 & 0 \\ 0 & \beta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & \Omega \\ O^T & 0 \end{pmatrix} \begin{pmatrix} \xi^c \\ \psi^c \\ \zeta^c \\ 1 \end{pmatrix} \tag{16}$$

Both (R) and (Ω) extrinsic camera parameters, do represent coordinate transformation between camera coordinate system and world coordinate system. Hence, any (u,v) point in camera image plan is evaluated via the following relation:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \left(\frac{1}{\zeta^C}\right) M_1 M_2 p^w \qquad \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{\zeta^C} M p^w \tag{17}$$

Here (M) in Equ (17) is referred to as a Camera Projection Matrix. We are given (1) a calibration rig, i.e., a reference object, to provide the world coordinate system, and (2) an image of the reference object. The problem is to solve (a) the projection matrix, and (b) the intrinsic and extrinsic parameters.

2.1 Computing a projection matrix

In a mathematical sense, we are given $\begin{pmatrix} \xi_i^w & \psi_i^w & \zeta_i^w \end{pmatrix}$ and $(u_i \quad v_i)^T$ for i = (1 …… n), we want to solve for M_1 and M_2:

$$\begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = \frac{1}{\zeta^C} M_1 M_2 \begin{pmatrix} \xi_i^w \\ \psi_i^w \\ \zeta_i^w \\ 1 \end{pmatrix} \tag{18}$$

$$\begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} = \frac{1}{\zeta^C} M \begin{pmatrix} \xi_i^w \\ \psi_i^w \\ \zeta_i^w \\ 1 \end{pmatrix}$$

$$\zeta_i^c = \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} \xi_i^w \\ \psi_i^w \\ \zeta_i^w \\ 1 \end{bmatrix} \tag{19}$$

$$\begin{aligned} \zeta_i^c u_i &= m_{11}\xi_i^w + m_{12}\psi_i^w + m_{13}\zeta_i^w + m_{14} \\ \zeta_i^c u_i &= m_{21}\xi_i^w + m_{22}\psi_i^w + m_{23}\zeta_i^w + m_{24} \\ \zeta_i^c u_i &= m_{31}\xi_i^w + m_{32}\psi_i^w + m_{33}\zeta_i^w + m_{34} \end{aligned} \tag{20}$$

$$\zeta_i^c = \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \sigma_{34} \end{bmatrix} \begin{bmatrix} \xi_i^w \\ \psi_i^w \\ \zeta_i^w \\ 1 \end{bmatrix} \tag{21}$$

$$\begin{aligned} \zeta_i^c u_i &= \sigma_{21}\xi_i^w + \sigma_{22}\psi_i^w + \sigma_{23}\zeta_i^w + \sigma_{24} \\ \zeta_i^c u_i &= \sigma_{31}\xi_i^w + \sigma_{32}\psi_i^w + \sigma_{33}\zeta_i^w + \sigma_{34} \end{aligned} \tag{22}$$

Obviously, we can let $\sigma_{34} = 1$. This will result in the projection matrix is scaled by σ_{34} . Once $KM = U$, $K \in \Re^{2n \times 11}$ is a matrix, a 11-D vector, and $U \in \Re^{2n-D}$ vector. A least square solution of equation $KM = U$ is thus expressed by:

$$M = K^{+}U$$

$$M = K^{T}K^{-1}K^{T}U \tag{23}$$

K^{+} is the pseudo inverse of matrix K , and m and m_{34} consist of the projection matrix M . We now turn to double scene analysis.

3. Double camera scene {epipolar geometry analysis}

In this section, we shall consider an image resulting from two camera views. For two perspective views of the same scene taken from two separate viewpoints O_1 and O_2 , this is illustrated in Fig. (3). Also we shall assume that (m_1 and m_2) are representing two separate points in two the views. In other words, perspective projection through O_1 and O_2 , of the same point X_w , in both image planes Λ_1 and Λ_2 . In addition, by letting (c_1) and (c_2) be the optical centers of two scene, the projection E_1 (E_2) of one camera center O_1 (O_2) onto the image plane of the other camera frame Λ_2 (Λ_1) is the epipole geometry.

It is also possible to express such an epipole geometry in homogeneous coordinates in terms $\tilde{E}_1$ and $\tilde{E}_2$:

$$\tilde{E}_1 = \begin{pmatrix} E_{1x} & E_{1y} & 1 \end{pmatrix}^T \text{ and } \tilde{E}_2 = \begin{pmatrix} E_{2x} & E_{2y} & 1 \end{pmatrix}^T \tag{24}$$

One of the main parameters of an epipolar geometry is the fundamental Matrix H (which is $\Re \in 3\times 3$). In reference to the H matrix, it conveys most of the information about the relative position and orientation (t, R) between the two different views. Moreover, the fundamental matrix H algebraically relates corresponding points in the two images through the Epipolar Constraint. For instant, let the case of two views of the same 3-D point X_w, both characterized by their relative position and orientation (t, R) and the internal, hence H is evaluated in terms of K_1 and K_2, representing extrinsic camera parameters, (Gian et al., 2004) :

$$H = K_2^{-T} (t)_x R K_1^{-1} \tag{25}$$

In such a similar case, a 3-D point (X_w) is projected onto two image planes, to points (m_2) and (m_1), as to constitute a conjugate pair. Given a point (m_1) in left image plane, its conjugate point in the right image is constrained to lie on the epipolar line of (m_1). The line is considered as the projection through C_2 of optical ray of m_1. All epipolar lines in one image plane pass through an epipole point.

This is a projection of conjugate optical centre: $\tilde{E}_1 = \tilde{P}_2 \begin{pmatrix} c_1 \\ 1 \end{pmatrix}$. Parametric equation of epipolar line of $\tilde{m}_1$ gives $\tilde{m}_2{}^T = \tilde{E}_2 + \lambda P_2 P_1{}^{-1} \tilde{m}_1$. In image coordinates this can be expressed as:

$$(u) = (m_2)_1 = \left(\frac{(\tilde{e}_2)_1 + \lambda (\tilde{v})_1}{(\tilde{e}_2)_3 + \lambda (\tilde{v})_3} \right) \tag{26}$$

$$(v) = (m_2)_2 = \left(\frac{(\tilde{e}_2)_2 + \lambda (\tilde{v})_2}{(\tilde{e}_2)_3 + \lambda (\tilde{v})_3} \right) \tag{27}$$

here $\tilde{v} = P_2 P_2{}^{-1} \tilde{m}_1$ is a projection operator extracting the (i^{th}) component from a vector.
When (C_1) is in the focal plane of right camera, the right epipole is an infinity, and the epipolar lines form a bundle of parallel lines in the right image. Direction of each epipolar line is evaluated by derivative of parametric equations listed above with respect to (λ) :

$$\left(\frac{du}{d\lambda} \right) = \left(\frac{[\tilde{v}]_1 [\tilde{e}_2]_3 - [\tilde{v}]_3 [\tilde{e}_2]_1}{([\tilde{e}_2]_3 + \lambda [\tilde{v}]_3)^2} \right) \tag{28}$$

$$\left(\frac{dv}{d\lambda} \right) = \left(\frac{[\tilde{v}]_2 [\tilde{e}_2]_3 - [\tilde{v}]_3 [\tilde{e}_2]_2}{([\tilde{e}_2]_3 + \lambda [\tilde{v}]_3)^2} \right) \tag{29}$$

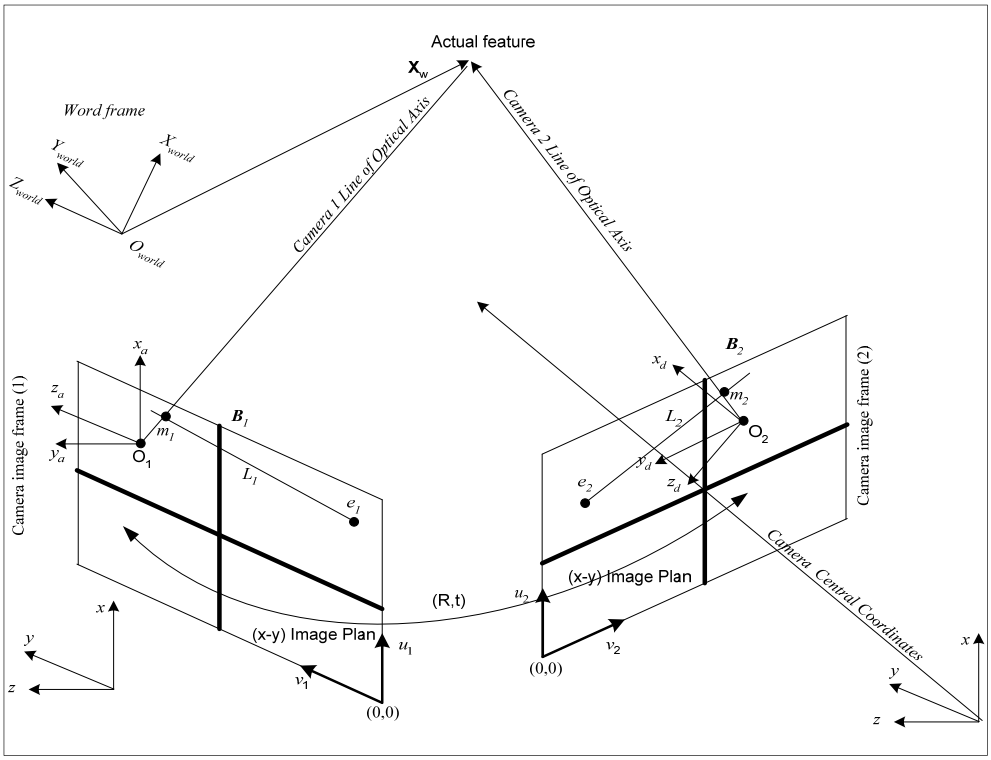


Fig 4. Camera image frame (Epipolar Geometry)

The epipole is projected to infinity once $\left(\tilde{E}_2\right)_3 = 0$. In such a case, direction of the epipolar lines in right image doesn't depend on any more. All epipolar lines becomes parallel to vector $\left(\left[\tilde{E}_2\right]_1 \quad \left[\tilde{E}_2\right]_2\right)^T$. A very special occurrence is once both epipoles are at infinity. This happens once a line containing (C_1) and (C_2), the baseline, is contained in both focal planes, or the retinal planes are parallel and horizontal in each image as in Fig. (4). The right pictures plot the epipolar lines corresponding to the point marked in the left pictures. This procedure is called rectification. If cameras share the same focal plane the common retinal plane is constrained to be parallel to the baseline and epipolar lines are parallel.

4. Neural net based Image - Based Visual Servo control (ANN-IBVS)

Over the last section we have focused more in single and double camera scenes, i.e. representing the robot arm visual sensory input. In this section, we shall focus on "Image-Based Visual Servo" (IBVS) which uses locations of object features on image planes (epipolar) for direct visual feedback. For instant, while reconsidering Fig. (1), it is desired to move a robotics arm in such away that camera's view changes from (an initial) to (final) view, and feature vector from (ϕ_0) to (ϕ_d). Here (ϕ_0) may comprise coordinates of vertices, or areas of the object to be tracked. Implicit in (ϕ_d) is the robot is normal to, and centered

over features of an object, at a desired distance. Elements of the task are thus specified in image space. For a robotics system with an end-effector mounted camera, viewpoint and features are functions of relative pose of the camera to the target, (${}^{c}\xi_{t}$). Such function is usually nonlinear and cross-coupled. A motion of end-effectors DOF results in complex motion of many features. For instant, a camera rotation can cause features to translate horizontally and vertically on the same image plane, as related via the following relationship :

$$\phi = f\left({}^{c}\xi_{t}\right) \tag{30}$$

Equ (30) is to be linearized. This is to be achieved around an operating point:

$$\delta\phi = {}^{f}J_{c}\left({}^{c}x_{t}\right)\delta\,{}^{c}x_{t} \tag{31}$$

$${}^{f}J_{c}\left({}^{c}x_{t}\right) = \left(\frac{\partial\phi}{\partial\,{}^{c}x_{t}}\right) \tag{32}$$

In Equ (32), ${}^{f}J_{c}\left({}^{c}x_{t}\right)$ is the Jacobian matrix, relating rate of change in robot arm pose to rate of change in feature space. Variously, this Jacobian is referred to as the feature Jacobian, image Jacobian, feature sensitivity matrix, or interaction matrix. Assume that the Jacobian is square and non-singular, then:

$${}^{c}\dot{x}_{t} = {}^{f}J_{c}\left({}^{c}x_{t}\right)^{-1}\dot{f} \tag{33}$$

from which a control law can be expressed by :

$${}^{c}\dot{x}_{t} = (K)\,{}^{f}J_{c}\left({}^{c}x_{t}\right)^{-1}\left(f_{d} - f(t)\right) \tag{34}$$

will tend to move the robotics arm towards desired feature vector. In Equ (34), K^{f} is a diagonal gain matrix, and (t) indicates a time varying quantity. Object posture rates ${}^{c}\dot{x}_{t}$ is converted to robot end-effector rates. A Jacobian , ${}^{f}J_{c}({}^{c}x_{t})$ as derived from relative pose between the end-effecter and camera, $\left({}^{c}x_{t}\right)$ is used for that purpose. In this respect, a technique to determine a transformation between a robot's end-effector and the camera frame is given by Lenz and Tsai, as in (Lenz & Tsai. 1988). In a similar approach, an end-effector rates may be converted to manipulator joint rates using the manipulator's Jacobian (Croke, 1994), as follows:

$$\dot{\theta}_{t} = {}^{t6}J_{\theta}^{-1}(\theta)\,{}^{t6}\dot{x}_{c} \tag{35}$$

$\dot{\theta}_{t}$ represents the robot joint space rate. A complete closed loop equation can then be given by:

$$\dot{\theta}_{t} = K\,{}^{t6}J_{\theta}^{-1}(\theta)\,{}^{t6}J_{\theta}^{f}J_{c}^{-1}\,{}^{c}x_{t}\left(f_{d} - f(t)\right) \tag{36}$$

For achieving this task, an analytical expression of the error function is given by :

$$\phi = Z^{+}\phi_1 + \gamma\left(I_6 - Z^{+}Z\right)\frac{\partial \phi_2}{\partial X} \tag{37}$$

Here, $\gamma \in \Re^{+}$ and Z^{+} is pseudo inverse of the matrix Z, $Z \in \Re^{m \times n} = \Re(Z^{T}) = \Re(J_1^{T})$ and J is the Jacobian matrix of task function as $J = \left(\frac{\partial \phi}{\partial X}\right)$. Due to modeling errors, such a closed-loop system is relatively robust in a possible presence of image distortions and kinematics parameter variations of the Puma 560 kinematics. A number of researchers also have demonstrated good results in using this image-based approach for visual servoing. It is always reported that, the significant problem is computing or estimating the feature Jacobian, where a variety of approaches have been used (Croke, 1994). The proposed IBVS structure of Weiss (Weiss et. al., 1987 and Craig, 2004), controls robot joint angles directly using measured image features. Non-linearities include manipulator kinematics and dynamics as well as the perspective imaging model. Adaptive control was also proposed, since ${}^{f}J_{\theta}^{-1}({}^{c}\theta)$, is pose dependent, (Craig, 2004). In this study, changing relationship between robot posture and image feature change is learned during a motion via a learning neural system. The learning neural system accepts a weighted set of inputs (stimulus) and responds.

4.1 Visual mapping: Nonlinear function approximation ANN mapping

A layered feed-forward network consists of a certain number of layers, and each layer contains a certain number of units. There is an input layer, an output layer, and one or more hidden layers between the input and the output layer. Each unit receives its inputs directly from the previous layer (except for input units) and sends its output directly to units in the next layer. Unlike the Recurrent network, which contains feedback information, there are no connections from any of the units to the inputs of the previous layers nor to other units in the same layer, nor to units more than one layer ahead. Every unit only acts as an input to the immediate next layer. Obviously, this class of networks is easier to analyze theoretically than other general topologies because their outputs can be represented with explicit functions of the inputs and the weights.

In this research we focused on the use of Back-Propagation Algorithm as a learning method, where all associated mathematical used formulae are in reference to Fig. (5). The figure depicts a multi-layer artificial neural net (a four layer) being connected to form the entire network which learns using the Back-propagation learning algorithm. To train the network and measure how well it performs, an objective function must be defined to provide an unambiguous numerical rating of system performance. Selection of the objective function is very important because the function represents the design goals and decides what training algorithm can be taken. For this research frame work, a few basic cost functions have been investigated, where the sum of squares error function was used as defined by Equ. (38):

$$E = \frac{1}{NP}\sum_{p=1}^{P}\sum_{i=1}^{N}\left(t_{pi} - y_{pi}\right)^2 \tag{38}$$

where p indexes the patterns in the training set, i indexes the output nodes, and t_{pi} and y_{pi} are, respectively, the target hand joint space position and actual network output for the i^{th} output unit on the p^{th} pattern. An illustration of the layered network with an input layer, two hidden layers, and an output layer is shown in Fig. (5). In this network there are i

inputs, (m) hidden units, and (n) output units. The output of the j^{th} hidden unit is obtained by first forming a weighted linear combination of the (i) input values, then adding a bias:

$$a_j = \left(\sum_{i=1}^{l} w^1_{ji} x_i + w^1{}_{j0} \right) \tag{39}$$

where $w^{(1)}_{ji}$ is a weight from input (i) to hidden unit (j) in the first layer. $w^{(1)}_{j0}$ is a bias for hidden unit j. If we are considering a bias term as being weights from an extra input $x_0 = 1$, Equ. (39) can be rewritten to the form of:

$$a_j = \left(\sum_{i=0}^{l} w^1_{ji} x_i \right) \tag{40}$$

The activation of hidden unit j then can be obtained by transforming the linear sum using a *nonlinear activation function* $g(x)$:

$$h_j = g\left(a_j\right) \tag{41}$$

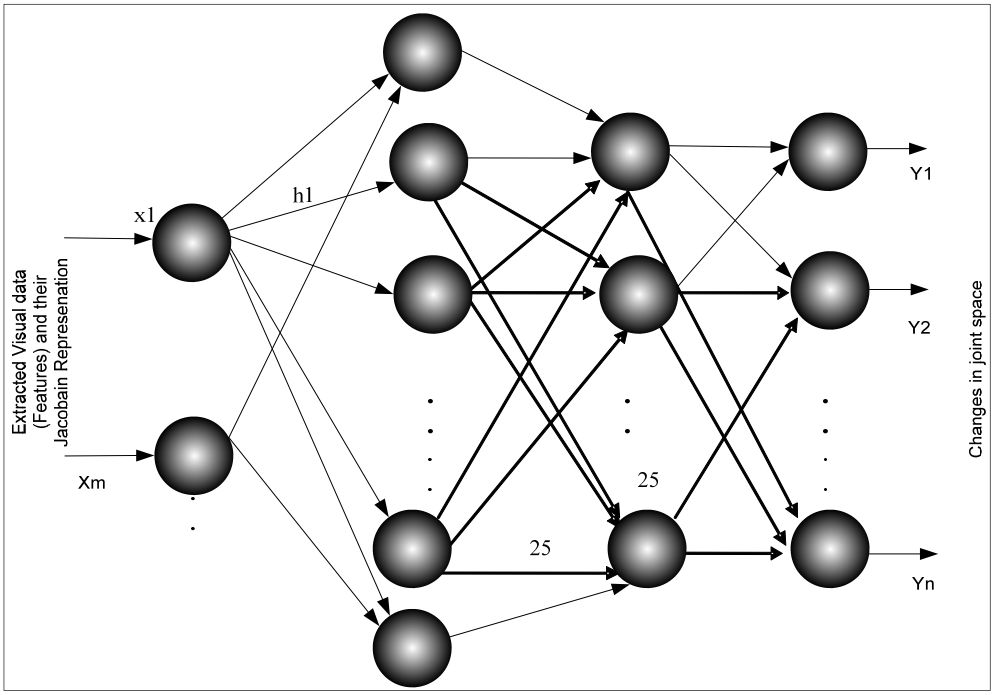


Fig. 5. Employed four layers artificial neural system

Outputs of the neural net is obtained by transforming the activation of the hidden units using a second layer of processing units. For each output unit k, first we get the linear combination of the output of the hidden units, as in Equ. (42):

$$a_k = \left(\sum_{j=1}^{m} w_{kj}^{2} h_j + w_{k0}^{2} \right) \tag{42}$$

absorbing the bias and expressing the above equation to:

$$a_k = \left(\sum_{j=0}^{m} w_{kj}^{2} h_j \right) \tag{43}$$

Applying the activation function $g_2(x)$ to Equ. (43), we can therefore get the k^{th} output :

$$y_k = g_2(a_k) \tag{44}$$

Combining Equ. (40), Equ. (41), Equ. (43) and Equ. (44), we get a complete representation of the network as:

$$y_k = g_2\left(\sum_{j=0}^{m} w_{kj}^{2} g\left(\sum_{i=0}^{l} w_{ji}^{2} x_i \right) \right) \tag{45}$$

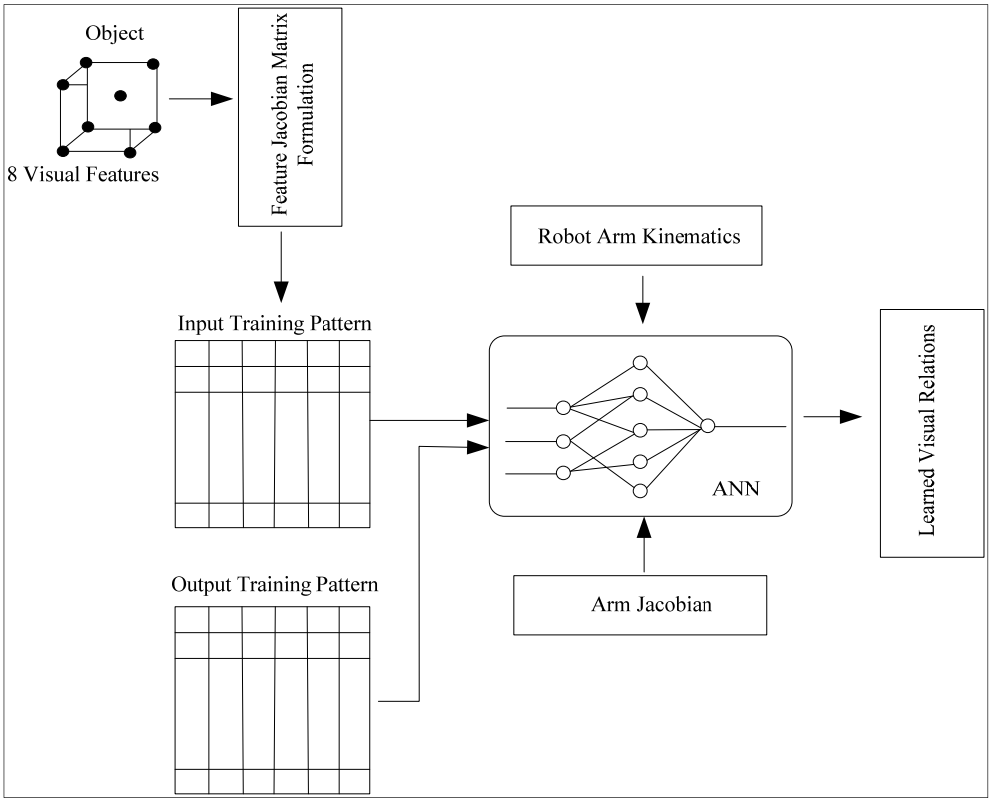


Fig. 6. Features based data gathering: Training patterns generations

The network of Fig. (5) is a synthesized ANN network with two hidden layers, which can be extended to have extra hidden layers easily, as long as we make the above transformation further. Input units do transform neural network signals to the next processing nodes. They are hypothetical units that produce outputs equal to their supposed inputs, hence no processing is done by these input units. Through this approach, the error of the network is propagated backward recursively through the entire network and all of the weights are adjusted so as to minimize the overall network error. The block diagram of the used learning neural network is illustrated in Fig. (6). The network learns the relationship between the previous changes in the joint angles $\Delta\Theta_{k-1}$, changes in the object posture Δu_a^c, and changes joint angles $\Delta\Theta_k$. This is done by executing some random displacements from the desired object position and orientation. The hand fingers is set up in the desired position and orientation to the object. Different Cartesian based trajectories are then defined and the inverse Jacobian were used to compute the associated joints displacement $\Theta_h(k)$. Different object postures with joint positions and differential changes in joint positions are the input-output patterns for training the employed neural network. During the learning epoch, weights of connections of neurons and biases are updated and changed, in such away that errors decrease to a value close to zero, which resulted in the learning curve that minimizes the defined objective function shown as will be further discussed later. It should be mentioned at this stage that the training process has indeed consumed nearly up to three hours, this is due to the large mount of training patterns to be presented to the neural network.

4.2 Artificial neural networks mapping: A biological inspiration

Animals are able to respond adaptively to changes in their external and internal environment and surroundings, and they use their nervous system to perform these behaviours. An appropriate model/simulation of a nervous system should be able to produce similar responses and behaviours in artificial systems. A nervous system is built by relatively simple; units, the neurons, so copying their behaviour and functionality should be the solution, (Pellionisz, 1989). In reality, human brain is a part of the central nervous system, it contains of the order of (10^{+10}) neurons. Each can activate in approximately 5ms and connects to the order of (10^{+4}) other neurons giving (10^{+14}) connections, (Shields & Casey, 2008). In reality, a typical neural net (with neurons) is shown in Fig. (5), it does resemble actual biological neuron, as they are made of:

- *Synapses*: Gap between adjacent neurons across which chemical signals are transmitted: (known as the input)
- *Dendrites*: Receive synaptic contacts from other neurons
- *Cell body /soma*: Metabolic centre of the neuron: processing
- *Axon*: Long narrow process that extends from body: (known as the output)

By emulation, ANN information transmission happens at the synapses, as shown in Fig. (5). Spikes travelling along the axon of the pre-synaptic neuron trigger the release of neurotransmitter substances at the synapse. The neurotransmitters cause excitation or inhibition in the dendrite of the post-synaptic neuron. The integration of the excitatory and inhibitory signals may produce spikes in the post-synaptic neuron. The contribution of the signals depends on the strength of the synaptic connection (Pellionisz, 1989). An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way

biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANN system, like people, learn by example.

An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANN system as well (Aleksander & Morton, 1995). The four-layer feed-forward neural network with (*n*) input units, (*m*) output units and N units in the hidden layer, already shown in the Fig. (5), and as will be further discussed later. In reality, Fig. (5). exposes only one possible neural network architecture that will serve the purpose. In reference to the Fig. (5), every node is designed in such away to mimic its biological counterpart, the neuron. Interconnection of different neurons forms an entire grid of the used ANN that have the ability to learn and approximate the nonlinear visual kinematics relations. The used learning neural system composes of four layers. The {input}, {output} layers, and two {hidden layers}. If we denote (${}^{w}v_{c}$) and (${}^{w}\omega_{c}$) as the camera's linear and angular velocities with respect to the robot frame respectively, motion of the image feature point as a function of the camera velocity is obtained through the following matrix relation:

$$\dot{\gamma} = -\left(\frac{\alpha\lambda}{{}^{c}p_{c}}\right)\begin{pmatrix} 0 & 0 & \frac{{}^{c}p_{x}}{{}^{c}p_{z}} & \frac{{}^{c}p_{x}\,{}^{c}p_{x}}{{}^{c}p_{z}} & -\frac{{}^{c}p_{x}\,{}^{c}p_{x}}{{}^{c}p_{z}} & {}^{c}p_{x} \\ 1 & -1 & \frac{{}^{c}p_{y}}{{}^{c}p_{z}} & \frac{{}^{c}p_{x}\,{}^{c}p_{x}}{{}^{c}p_{z}} & -\frac{{}^{c}p_{x}\,{}^{c}p_{x}}{{}^{c}p_{z}} & -{}^{c}p_{x} \end{pmatrix}\begin{pmatrix} {}^{c}R_{w} & 0 \\ 0 & {}^{c}R_{w} \end{pmatrix}\begin{pmatrix} {}^{w}v_{c} \\ {}^{w}\omega_{c} \end{pmatrix} \tag{46}$$

Instead of using coordinates (${}^{x}P_{c}$) and (${}^{y}P_{c}$) for the object feature described in camera coordinate frame, which are a priori unknown, it is usual to replace them by coordinates (*u*) and (*v*) of the projection of such a feature point onto the image frame, as shown in Fig. (7).

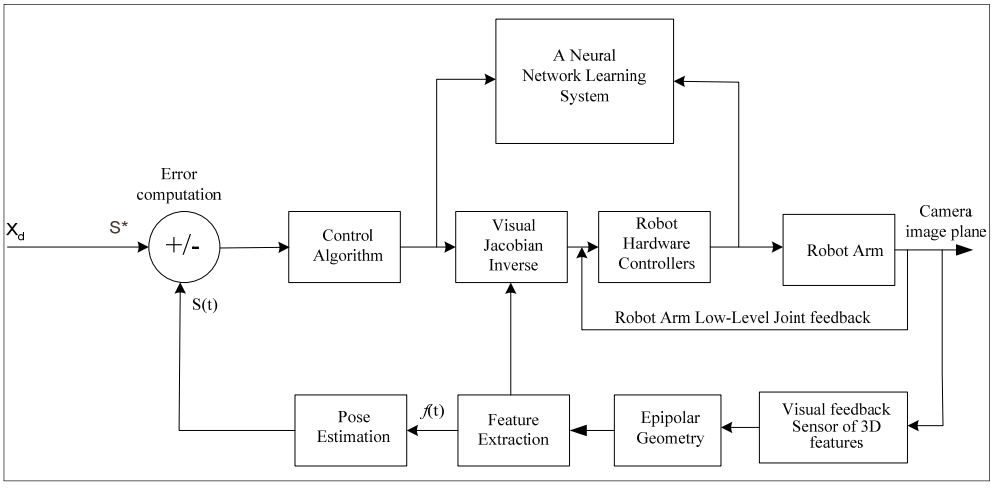


Fig. 7. Neural net based visual servo system

5. Simulation case study: Visual servoing with pin-hole camera for 6-DOF PUMA robotics arm

Visual servoing using a "pin-hole" camera for a 6-DOF robotics arm is simulated here. The system under study is a (PUMA) and integrated with a camera and ANN. Simulation block is shown Fig. (7). Over simulation, the task has been performed using 6-DOF-PUMA manipulator with 6 revolute joints and a camera that can provide position information of the robot gripper tip and a target (object) in robot workplace. The robot dynamics and direct kinematics are expressed by a set of equations of PUMA-560 robotics system, as documented by Craig, (Craig, 2004). Kinematics and dynamics equations are already well known in the literature, therefore. For a purpose of comparison, the used example is based on visual servoing system developed by RIVES, as in (Eleonora, 2004). The robotics arm system are has been servoing to follow an object that is moving in a 3-D working space. Object has been characterized by some like (8-features) marks, this has resulted in 24, $\Re \in^{8\times 3}$ size, feature Jacobian matrix. This is visually shown in Fig. (7). An object 8-features will be mapped to the movement of the object in the camera image plane through defined geometries. Changes in features points, and the differentional changes in robot arm, constitute the data that will be used for training the ANN. The employed ANN architecture has already been discussed and presented in Fig. (5).

5.1 Training phase: visual training patterns generation

The foremost ambition of this visual servoing is to drive a 6-DOF robot arm, as simulated with Robot Toolbox (Corke , 2002), and equipped with a pin-hole camera, as simulated with Epipolar Geometry Toolbox, EGT (Gian et al., 2004), from a starting configuration toward a desired one using only image data provided during the robot motion. For the purpose of setting up the proposed method, RIVES algorithm has been run a number of time before hand. In each case, the arm was servoing with different object posture and a desired location in the working space. The EGT function to estimate the fundamental matrix H, given U_1 and U_2, for both scenes in which U_1 and U_2 are defined in terms of eight (ξ, ψ, ζ) feature points:

$$U_1 = \begin{pmatrix} \xi_1^1 & \xi_1^2 & \cdots & \xi_1^8 \\ \psi_1^1 & \psi_1^2 & \cdots & \xi_1^8 \\ \zeta_1^1 & \zeta_1^2 & \cdots & \zeta_1^8 \end{pmatrix}$$

and (47)

$$U_2 = \begin{pmatrix} \xi_2^1 & \xi_2^2 & \cdots & \xi_2^8 \\ \psi_2^1 & \psi_2^2 & \cdots & \xi_2^8 \\ \zeta_2^1 & \zeta_2^2 & \cdots & \zeta_2^8 \end{pmatrix}$$

Large training patterns have been gathered and classified, therefore. Gathered patterns at various loop locations gave an inspiration to a feasible size of learning neural system. Four layers artificial neural system has been found a feasible architecture for that purpose. The

net maps 24 (3×8 feature points) inputs characterizing object cartesian feature position and arm joint positions into the (six) differential changes in arm joints positions. The network is presented with some arm motion in various directions. Once the neural system has learned with presented patterns and required mapping, it is ready to be employed in the visual servo controller. Trained neural net was able to map nonlinear relations relating object movement to differentional changes in arm joint space. Object path of motion was defined and simulated via RIVES Algorithm, as given in (Gian et al., 2004), after such large number of running and patterns, it was apparent that the learning neural system was able to capture such nonlinear relations.

5.2 The execution phase

Execution starts primary while employing learned neural system within the robotics dynamic controller (which is mainly dependent on visual feature Jacobian). In reference to Fig. (7), visual servoing dictates the visual features extraction block. That was achieved by the use of the Epipolar Toolbox. For assessing the proposed visual servo algorithm, simulation of full arm dynamics has been achieved using kinematics and dynamic models for the Puma 560 arm. Robot Toolbox has been used for that purpose. In this respect, also Fig. (8) shows an "aerial view" of actual object "initial" posture and the "desired" posture. This is prior to visual servoing to take place. The figure also indicates some scene features. Over simulation, Fig. (9) shows an "aerial view" of the Robot arm-camera servoing, as

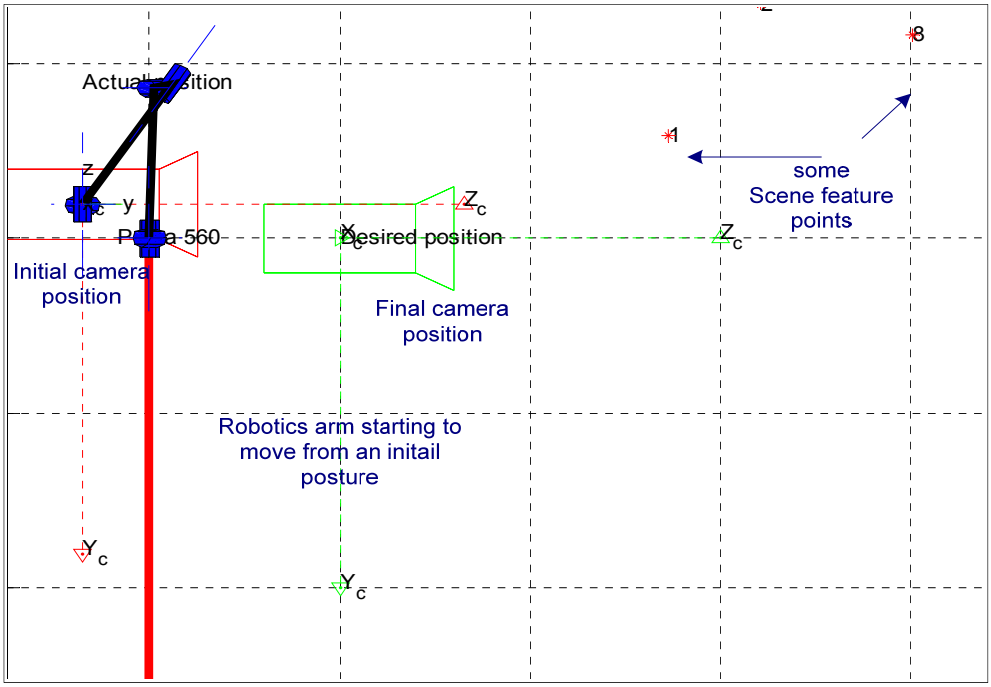


Fig. 8. Top view. Actual object position and desired position before the servoing

approaching towards a desired object posture. ANN was fed with defined patterns during arm movement. Epipolars have been used to evaluate visual features and the update during arm movement.

5.3 Object visual features migration

Fig. (10) shows the error between the RIVES Algorithm and the proposed ANN based visual servo for the first (60) iterations. Results suggest high accuracy of identical results, indicating that a learned neural system was able to servo the arm to desired posture. Difference in error was recorded within the range of (4×10^{-6}) for specific joint angles. Fig. (11) shows migration of the eight visual features as seen over the camera image plan. Just the once the Puma robot arm was moving, concentration of features are located towards an end within camera image plane. In Fig. (12), it is shown the object six dimensional movements. They indicate that they are approaching the zero reference. As an validation of the neural network ability to servo the robotics arm toward a defined object posture, Fig. (13) show that the trained ANN visual servo controller does approach zero level of movement. This is for different training patterns and for different arm postures in the 6-dimenstional space. Finally, Fig. (14) shows the error between RIVES computed joint space values and the proposed ANN controller computed joint space values. Results indicate excellent degree of accuracy while the visual servo controller approaching the target posture with nearly zero level of erroe for different training visual servo target postures.

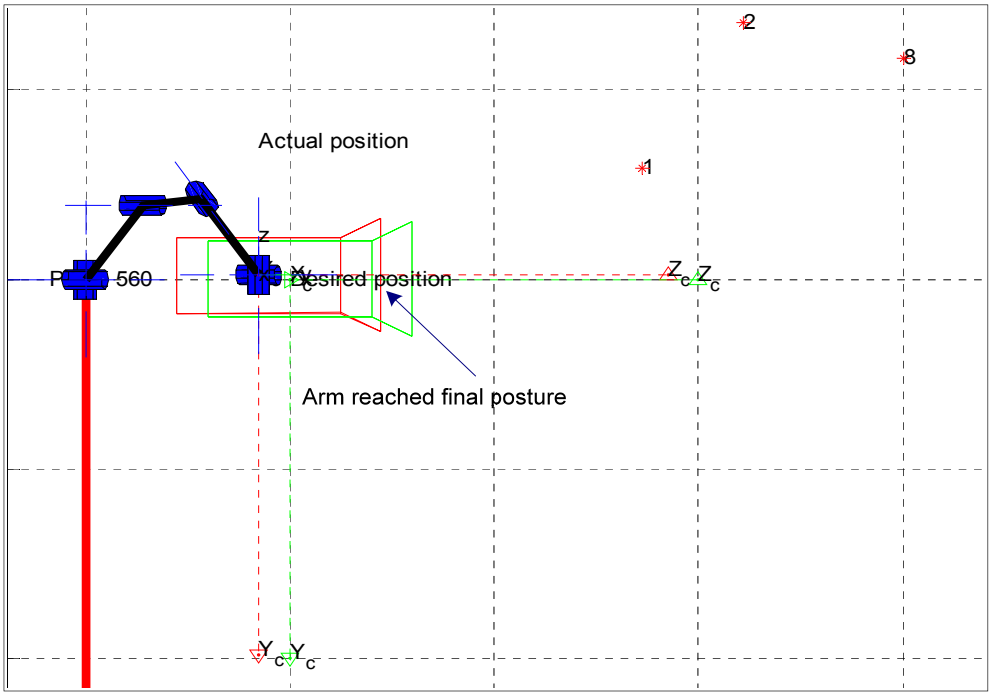


Fig. 9. Robot arm-camera system: Clear servoing towards a desired object posture

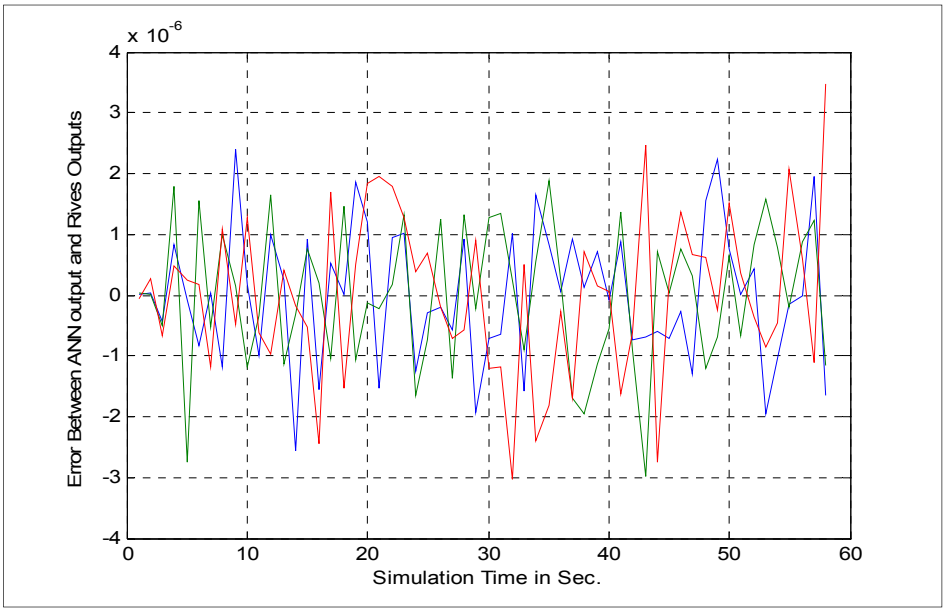


Fig. 10. Resulting errors. Use of proposed ANN based visual servo

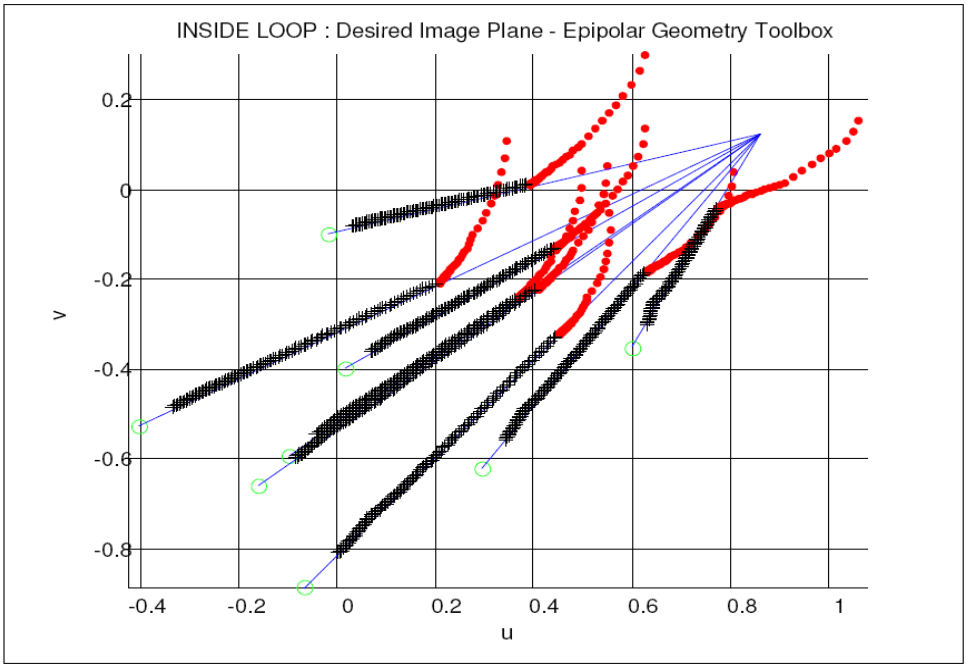


Fig. 11. Migration of eight visual features (as observed over the camera image plan)

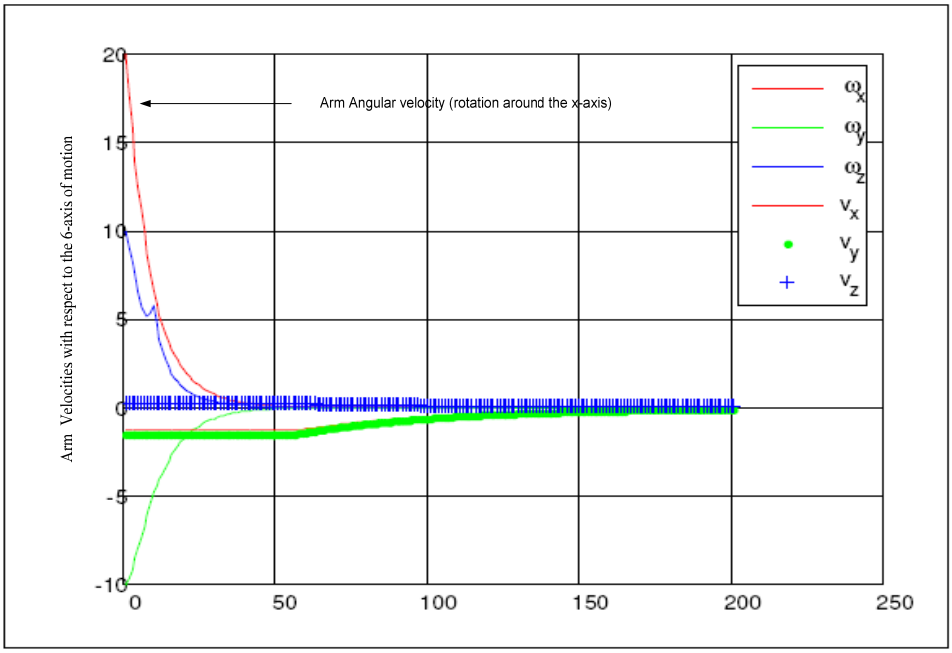


Fig. 12. Puma arm six dimensional movements

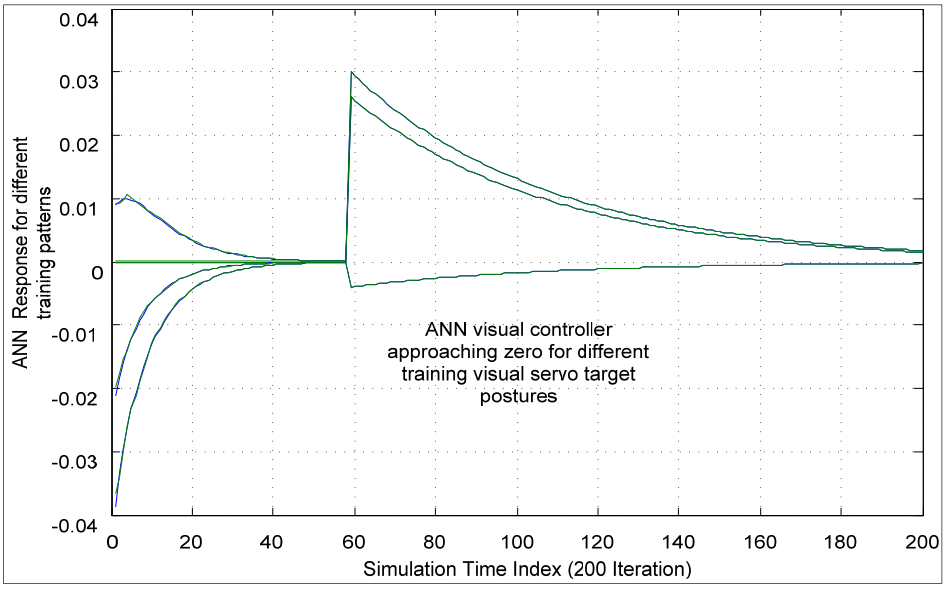


Fig. 13. ANN visual servo controller approaching zero value for different training visual servo target postures

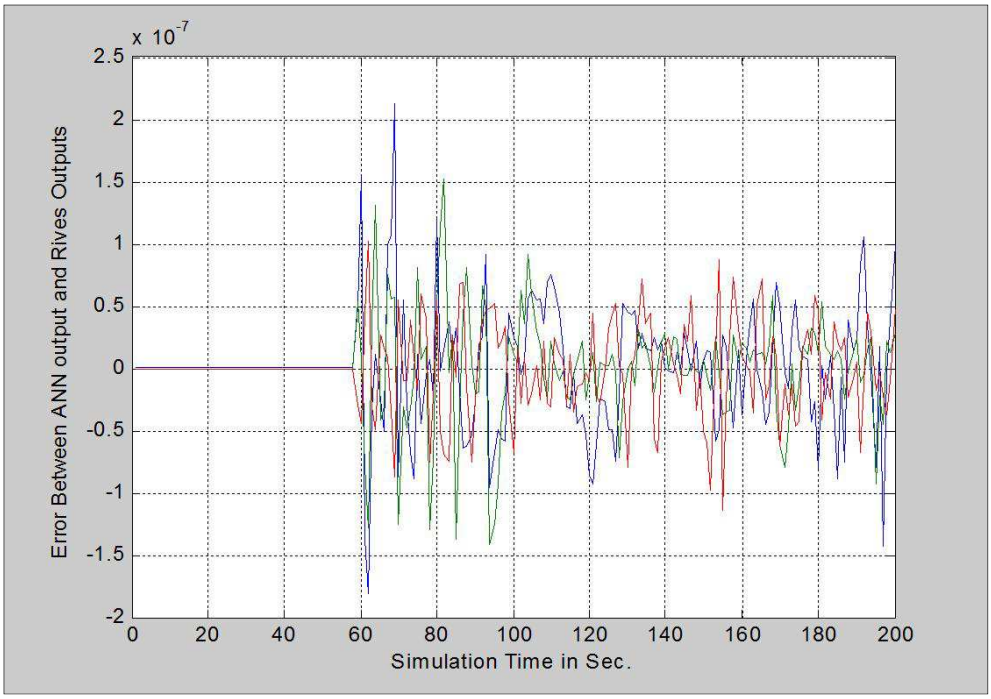


Fig. 14. Error of arm movements: ANN based controller and RIVES output difference ANN visual Servo

6. Conclusions

Servoing a robotics arm towards a moving object movement using visual information, is a research topic that has been presented and discussed by a number of researchers for the last twenty years. In this sense, the chapter has discussed a mechanism to learn kinematics and feature-based Jacobian relations, that are used for robotics arm visual servo system. In this respect, the concept introduced within this chapter was based on an employment and utilization of an artificial neural network system. The ANN was trained in such away to learn a mapping relating the " complicated kinematics" as relating changes in visual loop into arm joint space. Changes in a loop visual Jocobain depends heavily on a robotics arm 3-D posture, in addition, it depends on features associated with an object under visual servo (to be tracked). Results have shown that, trained neural network can be used to learn such complicated visual relations relating an object movement to an arm joint space movement. The proposed methodology has also resulted in a great deal of accuracy. The proposed methodology was applied to the well-know Image Based Visual Servoing, already discussed and presented by RIVES as documented in (Gian et al., 2004). Results have indicated a close degree of accuracy between the already published "RIVES Algorithm" results and the newly proposed "ANN Visual Servo Algorithm". This indicates ANN Visual Servo, as been based on some space learning mechanisms, can reduce the computation time.

7. References

Aleksander, I. & Morton, H. (1995), An Introduction to Neural Computing, *Textbook, International Edition 2nd Edition*.

Alessandro, D.; Giuseppe, L.; Paolo, O. & Giordano, R. (2007), On-Line Estimation of Feature Depth for Image-Based Visual Servoing Schemes, *IEEE International Conference on Robotics and Automation*, Italy, 1014

Chen, J.; Dawson, M.; Dixon, E. & Aman, B. (2008), Adaptive Visual Servo Regulation Control for Camera-in-Hand Configuration With a Fixed-Camera Extension, *Proceedings of the 46th IEEE Conference on Decision and Control*, CDC, 2008, pp. 2339-2344, New Orleans, LA, United States

Cisneros P. (2004), Intelligent Model Structures in Visual Servoing, *Ph.D. Thesis*, University of Manchester, Institute for Science and Technology

Corke, P. (2002), Robotics Toolbox for MatLab, *For Use with MATLAB, User Guide*, Vo1. 1.

Craig, J. (2004), Introduction to Robotics: Mechanics and Control, Textbook, *International Edition, Prentice Hall*

Croke, P. (1994), High-Performance Visual Closed-Loop Robot Control, Thesis submitted in total fulfillment of the Requirements for the Degree of Doctor of Philosophy.

Eleonora, A.; Gian, M. & Domenico, P. (2004), Epipolar Geometry Toolbox, *For Use with MATLAB, User Guide*, Vo1. 1.

Gian, M.; Eleonora, A. & Domenico, P. (2004), The Epipolar Geometry Toolbox (EGT) for MATLAB, *Technical Report*, 07-21-3-DII University of Siena, Siena, Italy

Gracia, L. & Perez-Vidal, C. (2009), A New Control Scheme for Visual Servoing, *International Journal of Control, Automation,and Systems*, Vol. 7, No. 5, pp. 764-776, DOI 10.1007/s12555-009-0509-9

Lenz, K. & Tsai, Y. (1988), Calibrating a Cartesian Robot with Eye-on-Hand Configuration Independent of Eye-To-Hand Relationship, *Proceedings Computer Vision and Pattern Recognition*, 1988 CVPR apos., Computer Society Conference, Vol.5, No. 9, pp. 67 – 75

Martinet, P. (2004), Applications In Visual Servoing, *IEEE-RSJ IROS'04 International Conference*, September 2004, Japan.

Miao, H.; Zengqi, S. & Fujii, M. (2007), Image Based Visual Servoing Using Takagi-Sugeno Fuzzy Neural Network Controller, *IEEE 22nd International Symposium on Intelligent Control, ISIC 2007*, Singapore, Vol. 1, No. 3, pp. 53 – 58

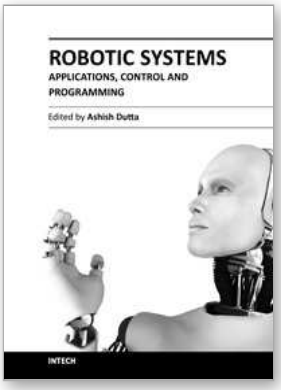
Miao, H.; Zengqi, S. & Masakazu, F. (2007), Image Based Visual Servoing Using Takagi-Sugeno Fuzzy Neural Network Controller, *22nd IEEE International Symposium on Intelligent Control*, Part of IEEE Multi-conference on Systems and Control Singapore, pp. 1-3

Panwar, V. & Sukavanam, N. (2007), Neural Network Based Controller for Visual Servoing of Robotic Hand Eye System, *Engineering Letters*, Vol. 14, No.1, EL_14_1_26, Advance Online Publication

Pellionisz, A. (1989), About the Geometry Intrinsic to Neural Nets, *International Joint Conference on Neural Nets*, Washington, D.C., Vol. 1, pp. 711-715

Shields, M. & Casey, C. (2008), A Theoretical Framework for Multiple Neural Network Systems, *Journal of Neurocomputing*, Vol. 71, No.7-9, pp. 1462-1476

Weiss, L.; Sanderson, A & Neuman, A. (1987), Dynamic Visual Servo Control of Robots: An adaptive Image-Based Approach , *IEEE Journal on Robotics and Automation*, Vol. 3, No.5, pp. 404–417.



Robotic Systems - Applications, Control and Programming
Edited by Dr. Ashish Dutta

ISBN 978-953-307-941-7
Hard cover, 628 pages
Publisher InTech
Published online 03, February, 2012
Published in print edition February, 2012

This book brings together some of the latest research in robot applications, control, modeling, sensors and algorithms. Consisting of three main sections, the first section of the book has a focus on robotic surgery, rehabilitation, self-assembly, while the second section offers an insight into the area of control with discussions on exoskeleton control and robot learning among others. The third section is on vision and ultrasonic sensors which is followed by a series of chapters which include a focus on the programming of intelligent service robots and systems adaptations.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821