Trajectory tracking with ANFIS and uncalibrated vision system for uncertain robotic manipulators

Aung Myat San[*], Wut Yi Win and Saint Saint Pyone

Department of Mechatronic Engineering, Mandalay Technological University, Myanmar

*Email: agmyatsanmc@googlemail.com

Abstract. This paper describes a new approach for tracking the trajectory of uncertain robotic manipulators using ANFIS (Artificial Neuro-Fuzzy Inference System) and uncalibrated vision system. The main emphasis of this work is on the ability to estimate the positioning accuracy and repeatability of a low-cost robotic arm with unknown parameters under uncalibrated vision system. In this study, captured image data are collected from two fixed-cameras vision system; installed on the top and lateral sides of the robot, respectively. For training and validating purposes, the robot is manoeuvring within its workspace using forward kinematics. The tracking system is trained using ANFIS with subtractive clustering method in MATLAB. Extensive simulations were carried out to illustrate the effectiveness of the proposed visual tracking method for LabVolt R5150 manipulator in our laboratory. Observing the simulation results, the performance of the proposed approach is efficient for using the vision-based learning system as visual feedback in uncertain robotic manipulator.

1. Introduction

In practical usage, a robot is a mechanical device which performs automated physical tasks, either according to direct human supervision, a pre-defined program, or a set of general guidelines using artificial intelligence techniques. The positioning problem of robot manipulators using visual information has been an area of research over the last 40 years. Attention to this subject has drastically grown in recent years. The visual feedback loop can solve many problems that limit applications of current robots: automatic driving, long range exploration, medical robotics, aerial robots, etc.

This paper focuses about mapping visual sensory information for robot control. Recently, the three-dimension (3-D) vision systems for robot applications have been popularly studied. Okada et al. [1] used multi-sensors for the 3-D position measurement. Winkelbach et al. [2] combined one camera with one range sensor to find the 3-D coordinate position of the target. Huang [3] addressed a 3-D position control for a robot arm utilizing two-CCD vision geometry and inverse kinematics. Zhou et al. [4] used position sensitive detector (PSD) for high-precision parallel kinematic mechanisms (PKMs) in order to allow them to accurately achieve their desired poses. Dallej et al. [5] developed 3D pose visual servoing for cable driven parallel robots.

In this paper, the positioning problem of articulated robot manipulators is addressed under two fixed cameras configurations. The main contribution is the development of a new pose-independent learning method for the robotic end-effector positioning using two uncalibrated fixed cameras and forward kinematics. The objective concerning the control is defined in terms of cartesian coordinates which are deduced from visual information.

The paper is organized by four sections. In section 2, the system approach is presented along with the forward kinematic analysis of the R5150 robotic, the learning of ANFIS technique and Kalman approach of motion estimation. In section 3, implementation of ANFIS training system, simulation tests and results are presented. Finally, the paper is concluded with the observed results in section 4.

2. System approach

In this section, the system approach is introduced including forward kinematic analysis for the robotic simulation, ANFIS learning for the Cartesian coordinates, and Kalman filtering for estimating the motion of the robot in occlusion cases. As shown in figure 1, the overall system approach is implemented along with two fixed imaging devices of view limits [0, 1024, 0, 1024].

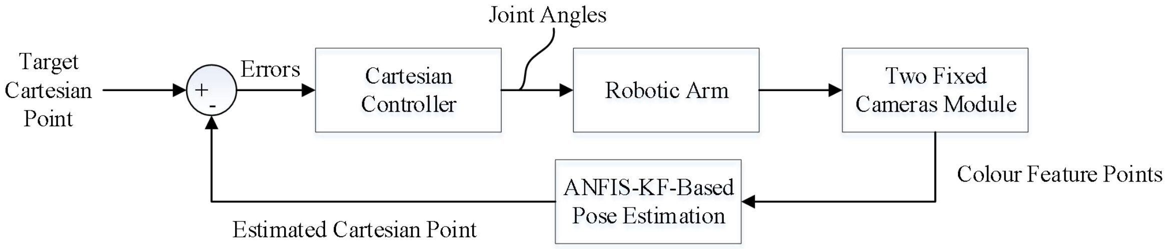


Figure 1. Overall system approach.

2.1. Forward kinematic analysis of the robot

The first step in modelling robot LabVolt R5150 is determining D-H parameters. It is used in robotics, where a robot can be modelled as a number of related solids (segments) where the D-H parameters are used to define the relationship between the two adjacent segments.

Table 1. D-H parameters.

	Link	d (m)	a (m)	α	Joint limits (min, max)
1	Base	0.2555	0	90˚	338˚ (-185˚, 153˚)
2	Shoulder	0	0.19	0	181˚ (-32˚, 149˚)
3	Elbow	0	0.19	0	198˚ (-147˚, 51˚)
4	Wrist Pitch	0	0	90˚	185˚ (-5˚, 180˚)
5.	Wrist Roll	0.115	0	0	360˚ (-360˚, 360˚)

The observed robot has 3-DOF three links with a 2-DOF wrist mechanism. In this work, the location of the end of the first three links is tracked using one colour feature point. For specifying D-H parameters of the robot, conventional D-H method is used. Using D-H parameters defined in the previous steps in table 1, robot model was created in MATLAB software using the Robotic and Vision Toolbox [6]. Robot model in addition to previously determined D-H parameters contains physical parameters which is using in the calculation of the dynamics movement.

The link transformation matrix between coordinate frames {i-1} and {i} has equation (1) [7];

$$ {}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i.\cos\alpha_i & \sin\theta_i.\sin\alpha_i & a_i.\cos\theta_i \\ \sin\theta_i & \cos\theta_i.\cos\alpha_i & -\cos\theta_i.\sin\alpha_i & a_i.\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

Since a coloured feature point is attached to the end of the first three links of the robot, we need to calculated the first three basic transformation matrices to get the location of the feature point. The final homogeneous transformation matrix for locating the feature point is got from the product of three basic transformation matrices as equation (2):

$$ {}^{0}T_3 = {}^{0}T_1.{}^{1}T_2.{}^{2}T_3 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} & Q^{3\times3} & & P^{3\times1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

where,

$$ Q = \begin{bmatrix} \cos(\theta_1)\cos(\theta_2+\theta_3) & -\cos(\theta_1)\sin(\theta_2+\theta_3) & \sin(\theta_1) \\ \sin(\theta_1)\cos(\theta_2+\theta_3) & -\sin(\theta_1)\sin(\theta_2+\theta_3) & -\cos(\theta_1) \\ \sin(\theta_2+\theta_3) & \cos(\theta_2+\theta_3) & 0 \end{bmatrix} $$

$$ P = \begin{bmatrix} \cos(\theta_1)\left(a_3\cos(\theta_2+\theta_3)+a_2\cos(\theta_2)\right) \\ \sin(\theta_1)\left(a_3\cos(\theta_2+\theta_3)+a_2\cos(\theta_2)\right) \\ d_1 + a_3\sin(\theta_2+\theta_3)+a_2\sin(\theta_2) \end{bmatrix} $$

From the position matrix *P*, we get the location of the feature point as the robotic motion. In this work, the forward kinematics of the robot is used to simulate and drive the robot for learning ANFIS networks and driving the robot to a specified trajectory or location.

2.2. ANFIS training

In this work, the visual positioning is trained using ANFIS as well as robotic forward kinematics and multi-view geometry. The first step was to test the working space of the robotic arm in vision. In order to do so, a m-file have been created in MATLAB based on the direct kinematics of the robotic arm and the perspective geometries of two cameras. In this work, two pin-hole cameras are installed on the top and lateral sides of the robot, respectively. The cameras' focal length and view limits are identical as 0.002 and [0, 1024, 0, 1024], respectively. In this research, mapping the visual sensory information into the Cartesian coordinates is performed using ANFIS method and MATLAB Fuzzy Logic toolbox.

In MATLAB, "*genfis2*" generates a Sugeno-type FIS structure using subtractive clustering. *genfis2* is generally used where there is only one output; hence here it has been used to generate initial FIS for training the ANFIS. On the other hand, "*genfis2*" achieves this by extracting a set of rules that simulates the data values. In order to determine the number of rules and antecedent membership functions, "*subclust*" function has been used by the rule extraction methods. Further it uses the linear least squares estimation to determine each rule's consequent equations.

However, ANFIS itself is only suitable for single output system. For a system with multiple outputs, ANFIS will be placed side by side to produce a Multiple ANFIS (MANFIS) [8]. The number of ANFIS required depends on the number of required output. In this research, three cartesian coordinate (XYZ) points have to be outputted as ANFIS outputs (i.e., X, Y and Z).

2.3. Generating train data and check data

For a feature point, a red-coloured feature point is placed at the end of the three links. Therefore, the feature point will be moved as long as the arm is moved. In this research, Corke's RVC v9.8 MATLAB toolbox [6] is used for simulation of robot and two cameras.

The data for ANFIS training process is generated by manoeuvring the robot in its operational workspace and acquiring image data from two cameras. This is so called as motor babbling phase in which the robot is driven in joint space. The displacement between two consecutive joint elements was limited to 30 degrees among their maximum and minimum ranges. To get the required data, the manipulator is manoeuvred within its workspace using forward kinematics, and the end-effector image coordinates are acquired from two cameras. For the neuro-fuzzy model in this work, data set are 588 cartesian points analytically obtained using forward kinematics, and feature points captured by two cameras in motor babbling phase. They are halved for training and validation of the ANFIS models; 294 and 294, respectively. The image coordinates of these cameras are [u_1, v_1] and [u_2, v_2]. And position matrix using forward kinematics is [p_x, p_y, p_z]. So, there are four inputs and one Cartesian point output for individual ANFIS; i.e. [u_1, v_1, u_2, v_2, p_x], [u_1, v_1, u_2, v_2, p_y] and [u_1, v_1, u_2, v_2, p_z].

2.4. Tracking an object on cameras' image plane using discrete Kalman filter

Kalman filter is a recursive solution to the discrete-data linear filtering problem. The Kalman filter has been used extensively for tracking in interactive computer graphics. This research focuses on tracking an object on camera's image plane to compensate the losses of image feature data due to lighting or occlusion. Therefore, a simple discrete Kalman filter is implemented in the system.

The state model of discrete Kalman filter is defined as equation (3) and (4).

$$\text{Estimate: } x_{k+1} = Ax_k + w_k \tag{3}$$

$$\text{Measurement: } z_k = Hx_k + v_k \tag{4}$$

where, w_k and v_k are white noises denoting the process and measurement noises, respectively. Parameters of the proposed discrete Kalman filter using equation (3) and (4) are as follows:

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},\ Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ R = 0.01\begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix},\ \hat{x}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \text{ and } P_0 = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}.$$

3. Simulation tests and results for trajectory tracking

For trajectory tracking purpose, MATLAB Simulink model is built as shown in figure 2. In which, the robot is driven in joint trajectory control from one joint coordinate to another (e.g., from [π/6, π/6, 0, 0, 0] to [π/4, π/2, π/4, 0, 0] as shown in figure 3). The comparative results of forward-kinematic-based and ANFIS-estimated trajectories are shown in figure 4. RMSE errors are 0.0014903, 0.0032865 and 0.0032303 for X, Y and Z, respectively. And Kalman filtering results for estimation the feature points in cameras' image planes are shown in figure 5.

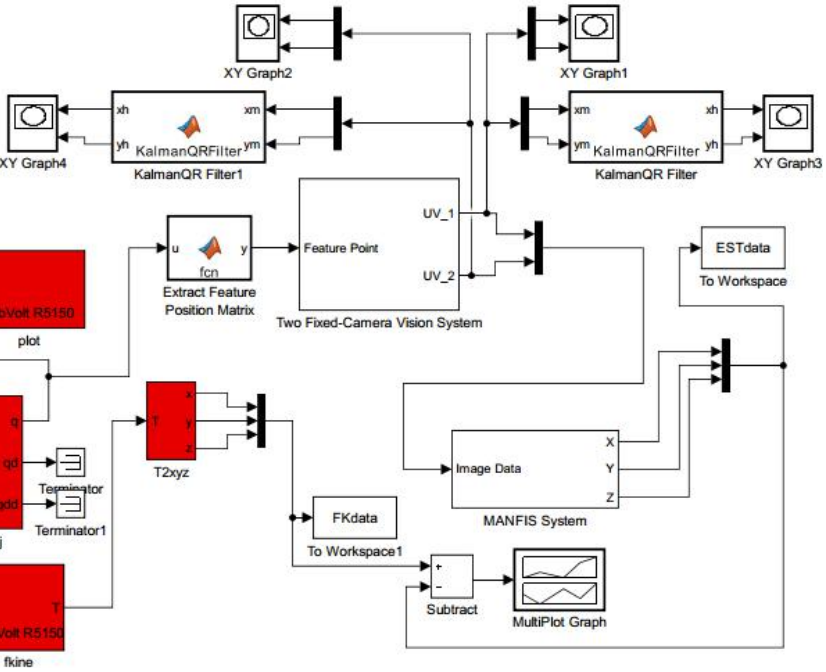


Figure 2. Simulink of the trajectory tracking for the robotic arm.

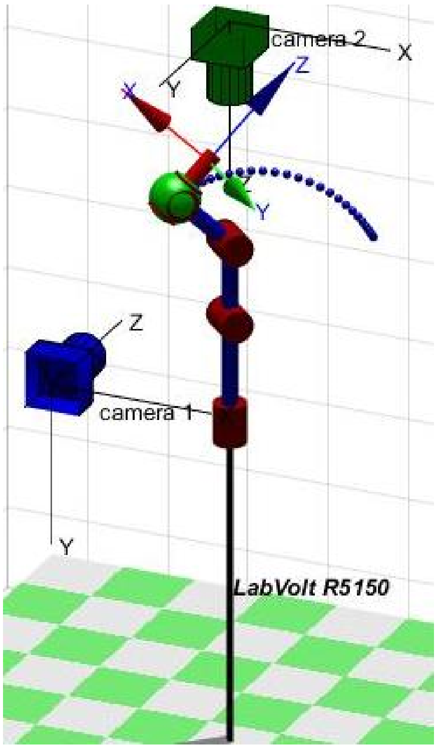


Figure 3. Simulation of the trajectory.

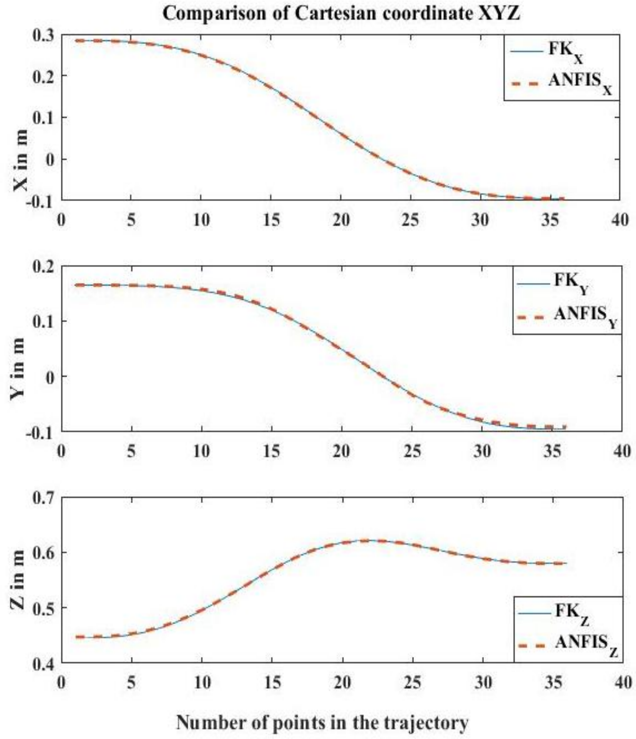
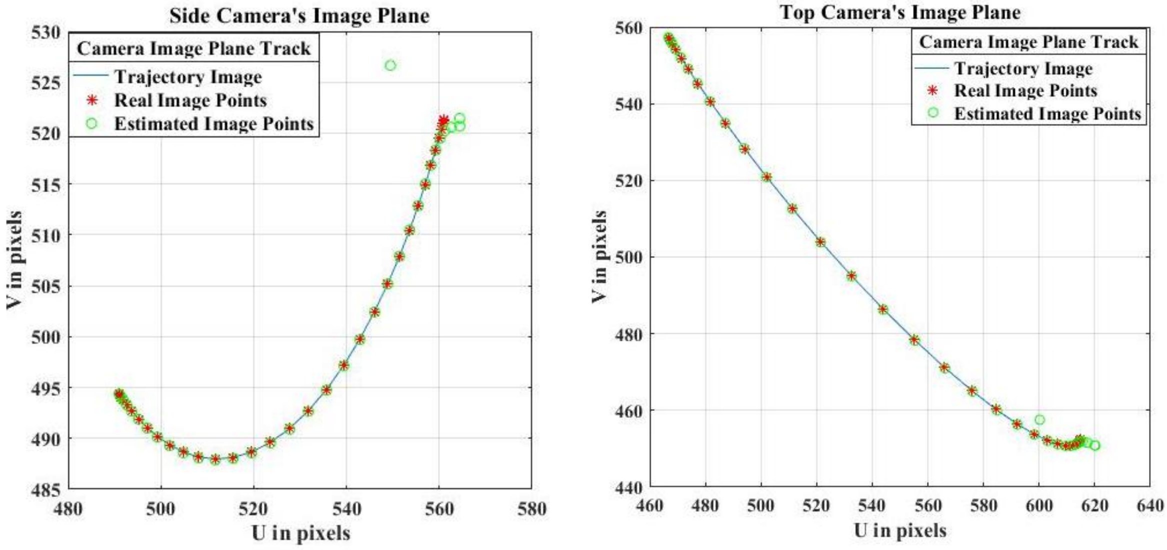


Figure 4. Comparative result for the trajectory.



(a) Tracking on side camera's image plane. **(b)** Tracking on top camera's image plane.

Figure 5. KF-based tracking estimations on cameras' image planes for the trajectory.

The proposed ANFIS-KF tracking method gives good estimation of the position of the robotic end-effector. The learning ability of ANFIS makes the kinematic and dynamic knowledge of the robot flexible under uncalibrated vision system. Since the learning is based upon Cartesian coordinates, it is easy to implement the controller designs and human-machine interfaces better. Moreover, the implementation of Kalman filter makes the system reliable in tracking concept. Therefore, the proposed ANFIS-KF approach for the trajectory tracking has better in flexibility, user-friendly manner and computational concepts over conventional techniques.

4. Conclusion

An ANFIS-based visual trajectory tracking approach using two uncalibrated cameras is proposed in this paper. The idea of using forward kinematic equations and two cameras for generating training data for ANFIS led to a nearly accurate training of the ANFIS network. The simulation experiments show that the location of the robotic arm can be trained in ANFIS using two uncalibrated cameras, and the tracking approach makes the robotic vision control systems more flexibly in control designs and more user-friendly in learning the robotic motion in Artificial Intelligence than conventional techniques. The proposed ANFIS-based approach of tracking the pose of the robot is very useful in obtaining the position of the robotic arm as it can work as a control algorithm. The Cartesian-coordinate-based learning can be used in robotic calibration, visual servoing and Cartesian controller. The authors are planning to use the pose tracking using MANFIS and uncalibrated cameras for the visual servoing of the robot in future.

Acknowledgments

The author would like to thank our teachers at the department for many interesting and informative discussions. The author would also like to thank the anonymous reviewers of the original version of this paper who provided comments and remarks that have been invaluable in helping us understand what we are trying to do as well as significantly improving the paper. And he also would like to thank his family and friends who have helped him throughout these years.

References

[1] K Okada, M Kojima, S Tokutsu, T Maki, Y Mori and M Inaba 2007 *Multi-cue 3D object recognition in knowledge-based vision-guided humanoid robot system* (*IEEE/RSJ Int. Conf. Intell. Robot Syst.*) (CA, USA) pp 3217-22

[2] S Winkelbach, S Molkenstruck and F M Wahl *2006 Low-cost laser range scanner and fast surface registration approach* (DAGM Symp. Pattern Recognit.) (Berlin, Germany)

[3] C H Huang, C S Hsu, P C Tsai, R J Wang and W J Wang 2011 *Vision Based 3-D Position Control for a Robot Arm* (*IEEE Control Systems*) pp 1699-1703

[4] E Zhou, M Zhu, A Hong, G Nejat and B Benhabib 2015 *Line-Of-Sight Based 3D Localization of Parallel Kinematic Mechanisms* (*Internal Journal on Smart Sensing and Intelligent System* vol 8 no 2) pp 1172-83

[5] T Dallej, M Gouttefarde, N Andreff, M Michelin and P Martinet 2011 *Towards vision-based control of cable driven parallel robots* (*IEEE Int. Conf. Intell. Robots and Systems IROSS'11*) (San Francisco, United States) pp 2855-60

[6] Corke P 2011 *Robotic, Vision and Control*, ed Bruno S, Oussama K and Frans G (Berlin: Spring Verlag)

[7] J J Denavit and R S Hartenberg 1955 *A Kinematics Notation for Lower-Pair Mechanisms Based on Artices* (*ASME Journal of Applied Mechanics* vol 22) pp 215-21

[8] Jang S R, Sun C T and Mizutani E 1997 *Neurl-Fuzzy and Soft Computing* (Prentice Hall)