Proceedings of the 12th
KACC, October 1997

Pose Control of Robotic Manipulator for Grasping the Moving Object
- Visual Servoing Based on Neuro-Fuzzy Model -

° Hyo-Byung Jun and Kwee-Bo Sim

Robotics and Intelligent Information System Laboratory
Dept. of Control and Instrumentation Engineering, Chung-Ang University
221, Huksuk-Dong, Dongjak-Ku, Seoul 156-756, Korea
kbsim@cau.ac.kr, http://rics.cie.cau.ac.kr

Abstracts In image jacobian based visual servoing, generally, inverse jacobian should be calculated by complicated coordinate transformations. These are required excessive computation and the singularity of the image jacobian should be considered. This paper presents a visual servoing to control the pose of the robotic manipulator for tracking and grasping 3-D moving object whose pose and motion parameters are unknown. Because the object is in motion tracking and grasping must be done on-line and the controller must have continuous learning ability. In order to estimate parameters of a moving object we use the kalman filter. And for tracking and grasping a moving object we use a fuzzy inference based reinforcement learning algorithm of dynamic recurrent neural networks. Computer simulation results are presented to demonstrate the performance of this visual servoing

Keywords visual servoing, kalman filter, neuro-fuzzy controller, reinforcement learning

1. Introduction

In the field of robotics, to control the pose of the manipulator's end-effector a lot of visual servoing technique have been studied.. Generally visual servoing is classified into two categories; position-based and feature based visual servoing. Sanderson and Weiss introduced an important distinction between these two. In position-based visual servoing 3 dimensional informations from the geometric model of the object is refered to the desired positions. In this case, because actual position data should be estimated from the 2 dimensional image data there are many difficulties such as noise and parameter variation in real time implementation[1].

In image-based visual servoing only image features are used to determine the pose of the robotic manipulator, so the pose estimation process does not need. This enables the controller to be robust to the noise, but remains a problem to determine the relative pose of the end-effector.

It is known that there exists a nonlinear mapping between the image plane and manipulator's 3 dimensional space. For this nonlinear mapping the image jacobian which transform the features and its variations to the desired end-effector motion should be derived. In image jacobian based visual servoing, generally, inverse jacobian should be calculated by complicated coordinate transformations and these are required excessive computation and the singularity of the image jacobian should be considered. When a moving object is concerned, these coordinate transformation can hardly be calculated and very sensitive to noise and system parameter variations.

Therefore instead of analytically deriving the closed form of such a nonlinear mapping, several researchers have discussed possibilities for the application of artificial neural networks. Miller proposed a learning control approach that utilizes a CMACS neural network model[2]. And Hashimoto et al. proposed a self-organizing visual servo system based on the feedforward type of the two neural networks to learn the feature jacobian[3]. But in these studies it is assumed that the object is static and known because the neural network should be trained off-line.

With regard to the moving object, Houshangi[4] developed a system to grasp targets exhibiting planar, translational motion utilizing a static camera and pre-computed manipulator poses for the start position and the grasp position. And Kimura et al. proposed a system capable of catching a free flying ball. And some researches are based on kalman filter and nonlinear observer[5][6][7].

In this paper, we propose on-line visual servoing for tracking and grasping a moving object using a artificial neural network and fuzzy inference. To implement on-line learning we adopt fuzzy inference based reinforcement learning(FIRL) of dynamic recurrent neural networks(DRNN)[8], and we use the kalman filter to estimate the moving object's parameters in successive image frames. The architecture of the proposed visual servoing is illustrated in Fig. 1.

2. Prediction of moving object features

When the moving object is concerned in visual servoing, the system is required to be able to adapt the changing environment

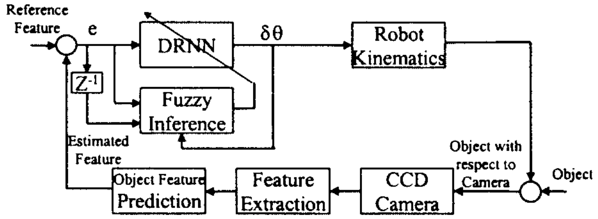


Fig. 1 Control system architecture

and especially the object's motion parameters on the image plane should be estimated. So we use kalman filter for the estimation of moving object parameters. The kalman filter as an optimal stochastic filter is used to estimate the motion parameters, the state vector of the moving object with the unknown dynamics in successive image frames.

To apply the kalman filter method to the estimation of the moving object's parameters, the dynamic model of the motion and the proper output equations must be defined first. We assumed that the object was moving with constant acceleration. in robotic manipulator's workspace. Then we can describe the discrete dynamic equation and measurement equation as follows

$$X(k+1) = A \cdot X(k) + B \cdot w(k) \tag{1}$$

$$y(k) = C \cdot X(k) + v(k) \tag{2}$$

where $X = [x \;\; \dot{x} \;\; \ddot{x}]^T$, $y(k)$ is a measurement vector, and $w(k)$ and $v(k)$ are assumed to be zero mean gaussian noises with covariance W, V respectively. And by the constant acceleration assumption

$$A = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} \frac{T_s^2}{2} \\ T_s \\ 1 \end{bmatrix}, C = [1 \;\; 0 \;\; 0]$$

where T_s is sampling time.

Then the standard recursive equations for the discrete kalman filter are stated by

$$\hat{X}(k) = A \cdot \hat{X}(k-1) + G \cdot (y(k) - C \cdot A \cdot \hat{X}(k-a)) \tag{3}$$

$$G = P \cdot C^T \cdot V^{-1} \tag{4}$$

where $\hat{X}(k)$ is updated state vector and G is kalman gain. And this kalman gain is obtained recursively by the Riccati equation.

$$AP + PA^T + BWB^T - PC^T V^{-1} CP = 0 \tag{5}$$

From these equations we can estimate simply the moving object position, x_p, on the image plane as follows

$$x_p = \left[1 \;\; T_s \;\; \frac{T_s^2}{2}\right]^T \cdot \hat{X}(k) \tag{6}$$

This predicted value is refered to the desired features and then image error is fed to neuro-fuzzy controller. Next section describes the learning algorithm of a artificial neural network using this image error to ouput the manipulator's joint angles.

3. Manipulator control by artificial neural network

Because the neural network can cope with nonlinear mapping without analytically deriving the closed form, neural network based robot control is widely studied.

However, to implement such supervised learning controllers, a lot of training exercise should be used to train the network to approximate the desired transformation. In Miller's CMACS neural network[2] and a self-organizing visual servo system proposed by Hashimoto et al.[3] neural networks should be trained over the whole work space to learn the feature Jacobian between the image features of the object and the joint angles for the desired position and orientation of the manipulator end-effector. And in the case of the moving object the training data can not be obtained simply and the convergency of the back propagation networks to global minimum is not generally guaranteed. Thus, the method seems to be very difficult to practically apply to a real task. So we introduce reinforcement learning of recurrent neural networks to visual servoing as on-line learning algorithm

3.1 *Reinforcement learning*

The supervised learning needs teaching signal from the exact modeling of the environment, but the reinforcement learning is generally unsupervised learning algorithm, finding the state-action rule or action generating strategy maximizing reward for the controller or agent's action under dynamically changing environment.

But as is often the case with real world, there is no immediate reinforcement until a goal state is reached. This requires improving long-term consequences of an action or of a strategy for performing actions, in addition to short-term consequences.

This problem is known as temporal credit assignment problem. A widely studied approach to this problem is to learn an internal evaluation function that is more informative than the evaluation function implemented by the external critic. The representative methods to this problem are actor-critic architecture by Sutton's temporal difference(TD) method[9] and Watkin's Q-learning[10].

3.2 *FIRL of DRNN*

The procedure of this algorithm is that a robot takes an action generated by neural networks, and this action is evaluated by the fuzzy inference engine. Then using the evaluated value, internal reinforcement in Fig. 2, the neural networks' weights are updated so that the robot learn and adapt the environment.

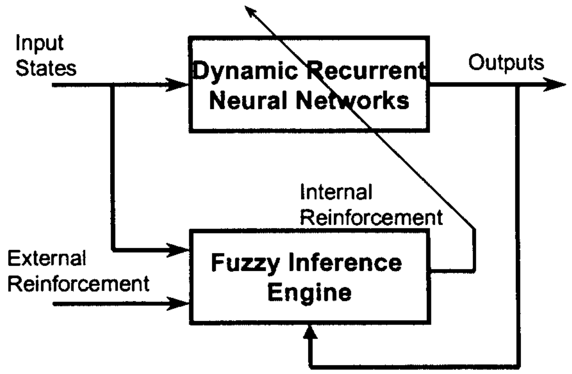


Fig. 2 Block diagram of FIRL

Generally actor-critic architecture used feedforward neural networks for generating the internal reinforcement. But we introduce the fuzzy inference engine as critic, whose input variables are composed state variables, the output of the action network, and external reinforcement, and internal reinforcement is its output.

In fuzzy logic, there are three types of fuzzy reasoning, the first is Mamdani's minimum fuzzy implication rule, the second is Tsukamoto's method with linguistic terms as monotonic membership functions, and the third is that the consequent of a rule is a function of input linguistic variables.

In this paper, we use Mamdani's fuzzy implication rule, max-min compositional rule of inference. The rules are expressed qualitatively and linguistically by fuzzy IF-THEN rules. And center of area defuzzification method is used.

And the structure of fully connected recurrent neural networks is shown in Fig. 3, made up of asymmetrically interconnected stochastic neurons.

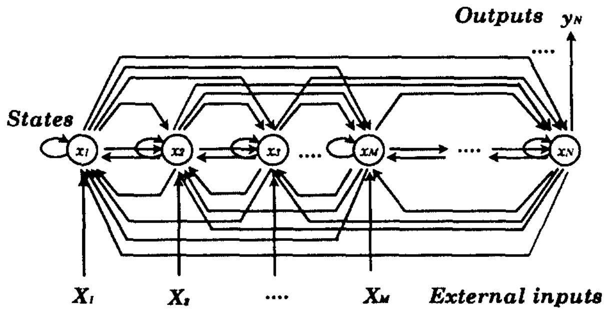


Fig. 3 Dynamic Recurrent Neural Networks

The output of the i-th neuron is,

$$y_i(t) = f(h_i(t-1)) + \Lambda_i(\sigma)$$
$$h_i(t) = (\sum_j w_{ij} y_j(t) + x_i(t)) \tag{7}$$

where $h_i(t-1)$ is net-input to th i-th node at time t-1, $x_i(t)$ is external input at time t, $f(\cdot)$ is nonlinear derivative activation function. And $\Lambda(\sigma)$ is gaussian random number with 0 mean and σ as standard deviation. This standard deviation is set properly as a function of reinforcement signal, r, as expressed in equation (8).

$$\sigma = \begin{cases} \dfrac{1}{\sum r} & r > 0 \\ 1 & r = 0 \\ \sum |r| & r < 0 \end{cases} \tag{8}$$

This gaussian random variable plays an important role in escaping local minimum and convergency to the global minimum.

Then the total cost function to be minimized is

$$E(t) = \frac{1}{2} \sum_k (E_k(t))^2 \tag{9}$$

$$E_k(t) = |(1 - r(t))| \cdot y_k(t) \tag{10}$$

where $r(t)$ is internal reinforcement signal at time t.

Equation (10) is an appropriate error measure for the output node k. By the gradient descent method, the change in weights is,

$$\Delta w_{pq}(t) = -\eta \frac{\partial E(t)}{\partial w_{pq}} = \eta \sum_k E_k(t) \frac{\partial y_k(t)}{\partial w_{pq}} \tag{11}$$

Here let $\frac{\partial y_i(t)}{\partial w_{pq}} \equiv z^i_{pq}$, then z^i_{pq} is given by

$$z^i_{pq}(t) = f'(h_i(t-1)) \left[\delta_{ip} y_q(t-1) + \sum_j w_{ij} z^j_{pq}(t-1) \right] \tag{12}$$

where δ_{ip} is a kronecker delta function

Consequently, from the above equations the weight changes to be applied to each weight w_{pq} in the networks is

$$\Delta w_{pq}(t) = \eta \cdot r(t) \cdot \sum_k E_k(t) \cdot z^k_{pq} \tag{13}$$

where η is a learning constant.

This DRNN composed stochastic unit is used as action networks and it's weights are updated by the equation (13) using reinforcement generated by fuzzy inference engine. And this neural networks learns the nonlinear mapping between the image error and desired robotic manipulator motion.

4. Simulation

We applied the proposed visual servoing to RV-M2 robotic manipulator simulations. It is assumed that the object's initial position and features are known and all the features are within the image plane. The object is given as a $50 \times 50 \times 30$mm cubic and has four feature points. And it is assumed to move in XY plane with 10mm/sec. The range and maximum speed of RV-M2's 5 joint angles are illustrated in Table 1.

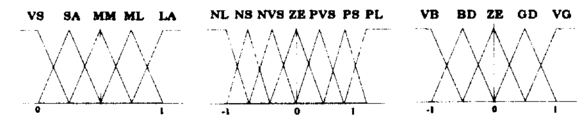
Table 1 Spec. of RV-M2 manipulator's joint angle

	θ_1	θ_2	θ_3	θ_4	θ_5
Range	-150° ~150°	-30° ~100°	-120° ~0°	-200° ~20°	-180° ~180°
Max. speed	140° /sec	79° /sec	140° /sec	163° /sec	223° /sec

The image error is calculated at each step by

$$e(t) = \frac{1}{N} \sum_{i=1}^{N} \sqrt{(x_i^r - x_i^p(t))^2 + (y_i^r - y_i^p(t))^2} \tag{14}$$

where N is the number of feature points, $(x_i^r \ , y_i^r)$ is the reference features, and $(x_i^p(t) \ , y_i^p(t))$ is the predicted feature points on the image plane. Using this image error and error changes, $\Delta e(t)$, as precondition and reinforcement as consequent,, we constructed fuzzy rules. Used fuzzy rules and fuzzy labels are ; VS(Very Small), SA(SmAll), MM(MediuM), ML(Medium Large), LA(LArge), NL(Negative Large), NS(Negative Small), NVS(Negative Very Small), ZE(ZEro), PVS(Positive Very Small), PS(Positive Small), PL(Positive Large), VB(Very Bad), BD(BaD), GD(GooD), VG(Very Good), as shown in Fig. 4 and Fig. 5.



a) Error b) Error changes c) Reinforcement

Fig. 4 Membership functions

		Δe						
		NL	NS	NVS	ZE	PVS	PS	PL
	VS	BD		VG		ZE	VB	
	SA	ZE	VG	GD		BD		
e	MM		GD		BD		VB	
	ML	VG				BD		VB
	LA	GD		BD		VB		

Fig. 5 Fuzzy rules for learning DRNN

And we used total 30 units recurrent neural networks with 8 inputs and 5 outputs. The error variations of four feature points in direction x and y are the 8 inputs and the outputs are RV-M2's 5 joint angles respectively.

The simulation result with 0.02(sec) sampling time plotted image error versus learning steps is shown in Fig. 6. And Fig. 7 shows the initial and final state of cubic on the image plane.

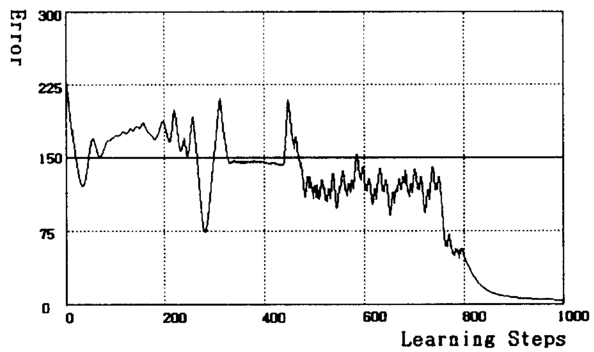
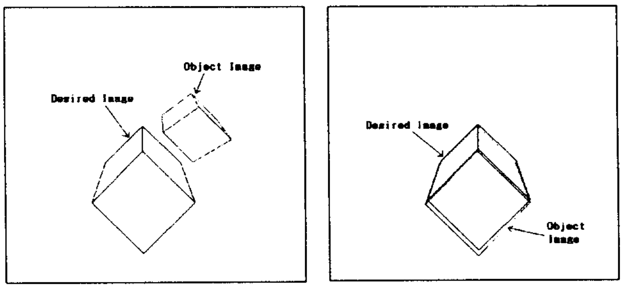


Fig. 6 Variance of image error



a) Initial image b) Final image

Fig. 7 Initial and final state

5. Conclusions

Neuro-fuzzy controller based visual servoing has been proposed in this paper. The main components are object parameter prediction module and fuzzy inference based reinforcement learning of dynamic recurrent neural networks.

In the simulation results we showed that recurrent neural networks can be trained on-line using the simple fuzzy rules with respect to error and error changes.

References

[1] T.Oya, H.Hashimoto, F.Harashima, "Predictive Filtering for Visual Tracking," *IECON*, pp. 1718-1723, 1993

[2] W.T.Miller, "Real-Time Applications of Neural Networks for Sensor Based Control of Robots with Vision," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol.19, No.4, 1989

[3] H.Hashimoto, T.Kubota, M.Kudo, F.Harashima, "Self-organizing visual servo system based on neural networks," *IEEE Control Systems Magazine*, Vol.12, No.4, pp.31-36, 1992

[4] N. Houshange, "Control of a robotic manipulator to grasp a moving target using vision," *Proc. of the IEEE International Conference on Robotics and Automation*, 604-609, 1990

[5] L.E.Weiss, A.C.Sanderson, C.F.Neuman, "Dyanmic sensor based control of robot with visual feedback", *IEEE Jounal of Robotics and Automation*, RA-3(50, pp.404-417, 1987

[6] K.Hashimoto, H.Kimura, "Visual Servoing with Nonlinear Observer," *Proc. of the IEEE International Conference on R&A*, pp.484-489, 1995

[7] K.B.Sim,"Visual Servoing Control of Robotic Manipulator for Moving Objects," *Jounal of the KITE*, Vol.4, No.2, pp15-24, 1996

[8] H.B.Jun, K.B.Sim, "Fuzzy Inference-based Reinforcement Learning of Dynamic Recurrent Neural Networks," *Proc. of the 36th SICE Annual Conference*, pp.1083-1088, 1997

[9] R.S. Sutton, "Learning to Predict by the Methods of Temporal Differences," *Machine Learning*, vol. 8, pp. 9-44, 1992.

[10] C.J.C.H. Watkins and P. Dayan " Technical Note : Q-Learning," *Machine Learning*, vol. 8, pp. 279-292, 1992.