



ELSEVIER

Microprocessors and Microsystems 21 (1998) 571–580

MICROPROCESSORS AND
MICROSYSTEMS

A vision-guided fuzzy logic control system for dynamic pursuit of a moving target

Kok-Meng Lee^{a,*}, Yifei Qian^b

^aThe George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0405, USA

^bLucent Technologies, Inc., 2000 Northeast Expressway, Suite G010, Norcross, GA 30071, USA

Received 6 June 1997; received in revised form 5 November 1997; accepted 16 December 1997

Abstract

We investigate the use of linguistic rules based on ‘rules-of-thumb’ experiences and engineering judgments to guide an industrial robot to follow a moving target using visual information. The problem has been formulated in the context of Prey Capture with the robot as a ‘pursuer’ and a moving object as a passive ‘prey’. Such a formulation mimics the function and capability of a natural being to pursue its prey. The feasibility of the fuzzy logic control strategy was verified experimentally. The experiments, built upon the framework of a low-cost flexible integrated vision system developed at Georgia Institute of Technology, were performed on a vibratory feeder with robotic hand–eye coordination. Experiments demonstrated that this approach does not require an accurate description of the dynamics of the pursuit process or that of the target’s motion, nor does it require the goals and constraints of the system to be quantified as single numerical values. These attractive features of the control strategy mean it has significant potential in industrial applications where models of the controlled process are not available, but the operator’s experience may be used as a guide to formulate the control rules. © 1998 Elsevier Science B.V.

Keywords: Part feeding; Intelligent control; Fuzzy logic control; Machine vision; Robotics

1. Introduction

This paper addresses the problem of guiding an industrial robot to follow a moving target with machine vision in an unattended and less structured environment. Such an application shares many of the characteristics of a common phenomenon; namely, the pursuit of prey. Other potential applications include automatic ship berthing, picking up moving objects, interception of flying attackers, and transferring live broilers in poultry processing.

In the application of machine vision for tracking, a camera is primarily used to detect the relative motion between a target and the tracker in which the camera resides. Two common approaches in tracking are the optical flow method [1] and the feature-matching method [2]. The optical flow method determines a field of instantaneous velocities from the gray level images of a target. It has been used in Refs. [3–5] among others in studies of visual feedback for robot controls. The feature-matching method involves registering certain features of moving targets from its consecutive images and uses the relationships between features to compute the parameters describing the motion. This method was

adopted in Refs. [6–9]. Many of these research efforts have treated the problem in the context of classical feedback servo, with visual feedback to close the loop of a robot control itself; where the dynamics of both a robot and a target are modeled analytically. Hence, much attention has been paid to system modeling and the accommodation of vision feedback in the robot control design in order to enhance the dynamics of the overall system.

For industrial applications such as picking up moving targets from a vibratory feeder, past research solutions are rather limited. The challenges are as follows.

1. The dynamics of the moving targets are highly nonlinear and are often impractical to model analytically.
2. The goals and constraints of the problem are not always quantifiable by single numerical values.
3. It is often impractical for users to customize off-the-shelf robots, feeders, and vision systems (which typically have their own stand-alone controllers) for their specific applications, especially when no model of the controlled process is available.

However, an experienced person is often able to estimate a target’s motion, and can take appropriate action to grasp it based on his/her knowledge gained over long-term observation

* Corresponding author. E-mail: kokmeng.lee@me.gatech.edu



Fig. 1. Dynamic pursuit of a sea lion by a white shark.

for particular circumstances. For example, a human being can usually catch a fly successfully with a fly swatter. This inspires us to apply the fuzzy logic theory introduced by [10] to tackle such problems.

We investigate the use of linguistic rules based on ‘rules-of-thumb’ experiences and engineering judgments to formulate control rules, and apply experimental or heuristic knowledge as a basis for logical inference. Such a formulation mimics the function and capability of a natural being to pursue its prey as illustrated in Fig. 1 [11], where a white shark dynamically tracks and captures its prey, typically a seal or a sea lion. Strong findings suggest that white sharks select their prey on the basis of shape with vision since the prey, illuminated from above by the sun, would appear as a black silhouette to the shark looking upward from below. Like many popular ideas, the predatory behavior of the stealthy white shark, which offers some interesting scientific implications, is employed in the design of an intelligent vision-guided control system for dynamic pursuit of moving target.

The remainder of this article is organized as follows: Section 2 presents the design concept of the vision-guided dynamic pursuit system, followed by the discussion of the fuzzy logic control rules in Section 3. The experimental prototype is briefly outlined in Section 4 and results are given in Section 5. Finally, the conclusions are summarized in Section 6.

2. Design concept

The system for target pursuit consists of at least three components or subsystems; namely, a robotic system, a

vision system, and a target. The problem is formulated here in the context of prey capture, where the robot behaves as a ‘pursuer’ and a moving target as a ‘prey’ or ‘evader’. Mathematically, the problem can be expressed as follows:

$$\|x_t - x_r\| \leq \|x_f\|, \tag{1}$$

where x_t and x_r are the state vectors of the moving target and the robot, respectively; x_f is the desired final state difference between the robot and the moving target; and the notation $\|*\|$ denotes the norm of vectors: $\|x\| = (x_1^2 + x_2^2 + \dots + x_n^2)^{1/2}$. In Eq. (1), the state vector is defined as $x_{(*)} = [p_{(*)}, v_{(*)}, \theta_{(*)}]^T$ where $p_{(*)}$, $v_{(*)}$, $\theta_{(*)}$ are the position, velocity and orientation vectors of $(*)$, respectively, where $x \in R^n$, and $x = (x_1, x_2, \dots, x_n)^T$.

Eq. (1) represents the control objective of the pursuit process subjected to the following constraints imposed by the robot, the object, and the environment.

1. The speed and acceleration of the robot are limited:

$$\|v_r\| \leq v_{rmax}, \tag{2a}$$

$$\|a_r\| \leq a_{rmax}, \tag{2b}$$

where v_{rmax} and a_{rmax} denote the maximum speed and acceleration of the robot respectively.

2. The robot can only move without hitting any obstacle within a certain workspace, whereas moving targets will not venture into certain areas. In other words,

$$P_r \subseteq P_{r0} \tag{3a}$$

$$P_o \subseteq P_{o0} \tag{3b}$$

where P_{r0} denotes the range of P_r , and P_{o0} the range of P_o .

The objective of the fuzzy logic controller (FLC), as shown in Fig. 2, is to find an appropriate command to direct the robot gripper to the vicinity of the moving target based on the visual information from the vision system. Here, the ‘pursuer’ (robot) determines its action based on how far a ‘prey’ (moving target) is away from it, how fast the ‘prey’ is fleeing, and in which direction the ‘prey’ is going to flee.

As shown in Fig. 2, the displacement and velocity differences are chosen as the primary motion parameters

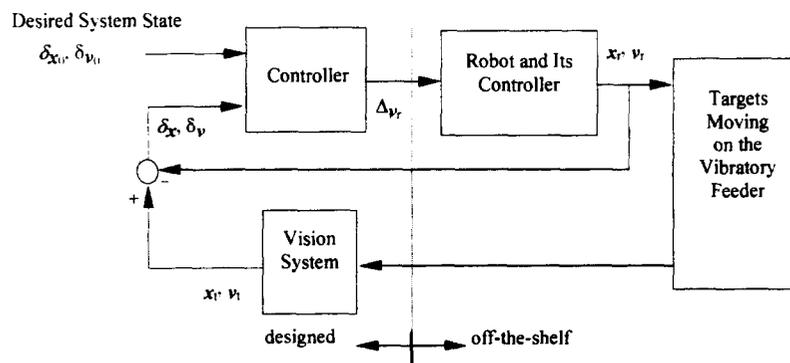


Fig. 2. System schematic of dynamic pursuit.

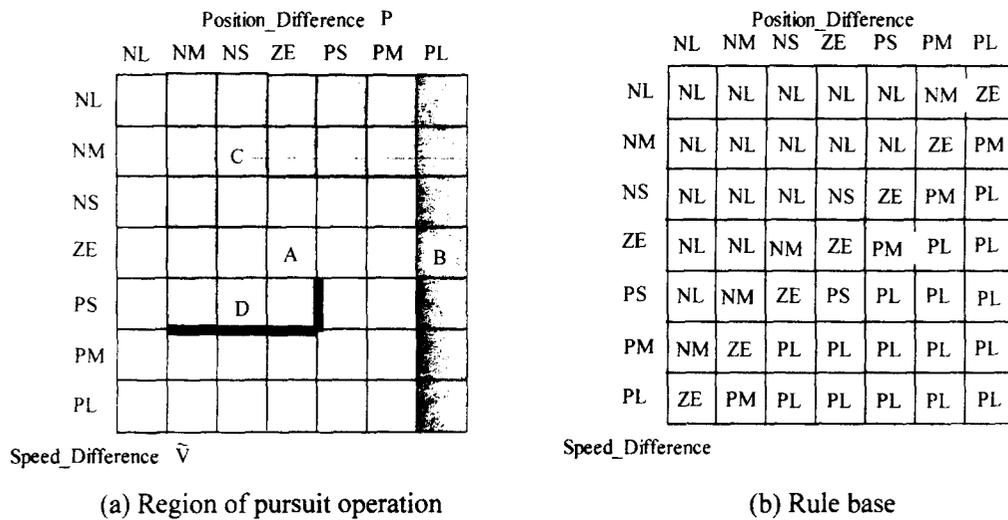


Fig. 3. Desired control actions for target pursuit: (a) region of pursuit operation; (b) rule base.

to describe the relationship between the robot and the target. Thus, these variables are defined as inputs to the controller as follows:

$$\delta \mathbf{x} = \mathbf{x}_t - \mathbf{x}_r, \tag{4a}$$

$$\delta \mathbf{v} = \mathbf{v}_t - \mathbf{v}_r, \tag{4b}$$

where \mathbf{x}_t and \mathbf{x}_r are the positions of the target and the robot gripper, respectively, and \mathbf{v}_t and \mathbf{v}_r are the velocities of the target and the robot gripper. Likewise, the change to the velocity is chosen as the output variable in order to accelerate or decelerate the motion of the 'pursuer'. That is, the robot velocity command is computed by

$$\mathbf{v}_r(t+1) = \mathbf{v}_r(t) + \delta \mathbf{v}_r, \tag{5}$$

where t denotes the current time interval, $t+1$ the next interval, and $\delta \mathbf{v}_r$ is the output of the fuzzy logic control—the speed change to the robot gripper. Both input and output variables are characterized by a fuzzy set of seven linguistic values. These values are

- | | |
|-----------------------|-----------------------|
| NL: negatively large | PS: positively small |
| NM: negatively medium | PM: positively medium |
| NS: negatively small | PL: positively large |
| ZE: zero | |

where N denotes negative, P positive, S small, M medium, and L large. The controller output is determined using the linguistic rules in the following form:

$$\text{IF } \tilde{P} \text{ is } A_i \text{ AND } \tilde{V} \text{ is } B_i, \text{ THEN } \tilde{R} \text{ is } C_i \tag{6}$$

where \tilde{P} and \tilde{V} are the input fuzzy variables (position and speed differences); \tilde{R} is the output fuzzy variable (change to robot speed); A_i , B_i , and C_i are the linguistic values corresponding to the i th rule for the fuzzy variables \tilde{P} , \tilde{V} and \tilde{R} , respectively. For a two-input system with seven fuzzy values for each input, a fully populated rule base will have $7 \times 7 = 49$ possible input combinations of rules.

Since no model of the controlled process is available, the operator's experience is used as a guide to formulate the control rules. To illustrate the rule-based pursuit operation, the phase plane of \tilde{P} and \tilde{V} is divided into regions as shown in Fig. 3(a) where the unshaded region is a mirror image of the shaded cells. The corresponding rules are given in Fig. 3(b). The control action is determined from a specified pair of input fuzzy variables, and the desired final state is at the center of the rule base table, denoted cell A in Fig. 3(a). Typical control action is illustrated as follows. If there is a large positive difference in position (PL), as indicated in shaded region B in Fig. 3(a), the robot should be commanded to accelerate to reduce the difference. Thus, the Robot_Velocity_Change is assigned to be Positively Large (PL). As the corresponding speed difference between the target and the robot becomes increasingly negative and finally reaches NL, the control action will gradually be lowered to ZE, as shown on the upper right cell of the rule base matrix in Fig. 3(b). As the position difference reduces, the control action reverts to NL, lowering the robot speed as the system enters region C. However, a reduction in robot speed will tend to increase the speed difference. The system then moves through region C designed to drive the speed difference from NL to ZE and finally to region D. The rules for the three cells in region D are

IF \tilde{P} is NS AND \tilde{V} is ZE, THEN \tilde{R} is NM

IF \tilde{P} is NS AND \tilde{V} is PS, THEN \tilde{R} is ZE

IF \tilde{P} is ZE AND \tilde{V} is PS, THEN \tilde{R} is PS.

Finally, the system is dampened in toward cell A, which is the desired system final state. Note that the unshaded cells in Fig. 3(a) is the mirror image of the shaded cells that correspond to a negative large position difference. The

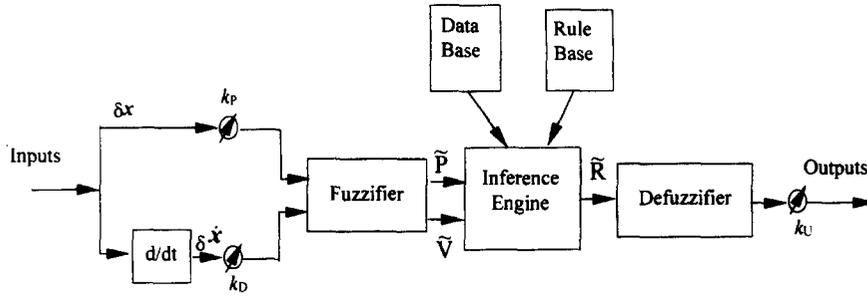


Fig. 4. Schematic of a fuzzy logic controller.

control actions of the unshaped region thus have identical amplitude but opposite polarity actions to that of the shaded region.

3. Fuzzy logic control (FLC) algorithm

The schematic of the fuzzy logic controller is shown in Fig. 4. It consists of a fuzzifier, an inference engine made up of a data base and a rule base, and a defuzzifier. The fuzzifier transforms the inputs in real crisp numbers into fuzzy values or fuzzy sets. The inference engine performs the fuzzy reasoning upon the linguistic or other control rules stored in the rule base. The data base provides the inference engine with the membership functions of the fuzzy sets which describe the input and output variables and are used in the rule base. The defuzzifier transforms the outputs of the fuzzy inference engine, a fuzzy set itself, into the outputs in real crisp number to provide single-valued control signals for the plant.

As discussed in the previous section, the measured position and the velocity differences are chosen as input variables. The change to the robot speed is chosen as the controller output. In the fuzzification process, the inputs and outputs are usually normalized to a certain range. In general, for a variable defined in range $[a, b]$, it can be normalized to range $[-c, c]$ as follows:

$$x = 2c \left[\frac{y - (a + b)/2}{a - b} \right] \tag{7}$$

where c can be any integer whose value is dependent on specific applications.

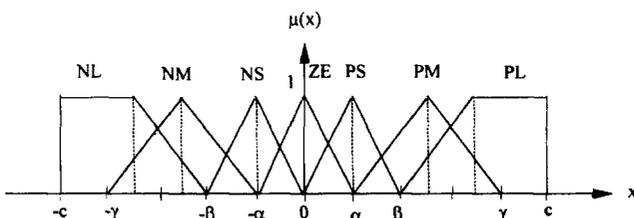


Fig. 5. Definition of the membership functions.

3.1. Fuzzification

Each of the input and output physical variables are characterized by a fuzzy set of linguistic values, namely, NL, NM, NS, ZE, PS, PM, and PL. Each linguistic value is defined by its membership function, as shown in Fig. 5. Membership functions are defined to have an overlap with each other in order to provide a smooth output transition between regions. The degree of overlap is characterized by the truth value of the intersecting point between the two adjacent membership functions and can be tuned. The system tends to be more robust as the overlap increases but the control is more sensitive as the overlap decreases. The output from the controller can then be obtained from the inputs by using the compositional rule of inference.

As stated in Section 2, the controller attempts to find an appropriate action to reduce the position and speed difference between the gripper and the target to a desired level. Thus, when the position difference is very large, it causes the pursuer to accelerate with a speed higher than its prey, so it can catch up with its prey quickly. This initial pursuit operation can be regarded as a coarse motion control. However, once the pursuer gets closer to the prey, emphasis is shifted to a finer motion control. The fine control action reduces the speed difference between pursuer and prey to within a specified range of reach. To achieve this effect, the FLC is designed with a self-tuning capability that automatically shifts from a coarser to a finer motion. Narrower membership functions are chosen for the fuzzy variables close to zero where fine motion control is desired, whereas broader spanned membership functions are used for fuzzy variables far from zero where coarser motion control will be accepted. In other words, the parameters of the universe are set as

$$(c - \beta) \geq (\gamma - \alpha) \geq \beta \tag{8}$$

where $(c - \beta)$, $(\gamma - \alpha)$, and β are the base lengths of the polygons characterized the membership functions defined in Fig. 5.

3.2. Inference engine

Since the membership functions of the fuzzy variables overlap, the inference engine must determine a control

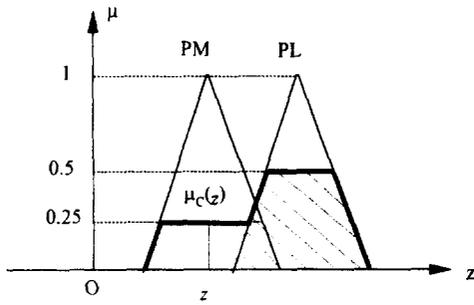


Fig. 6. Schematic of defuzzification process.

output r , the actual crisp change to the robot speed, from any two control rules for the fuzzy variable \tilde{R} given the position difference p for the fuzzy variable \tilde{P} and the speed difference v . The decision process of the inference engine is as follows. Recall that the rules are in the form represented by Eq. (6). Consider two rules as follows:

Rule 1 : IF \tilde{P} is A_1 and \tilde{V} is B_1 , THEN \tilde{R} is C_1

Rule 2 : IF \tilde{P} is A_2 and \tilde{V} is B_2 , THEN \tilde{R} is C_2 .

For Rule 1, the truth value of p_0 is $\mu_{A_1}(p_0)$, the truth value of v_0 , $\mu_{B_1}(v_0)$, where $\mu_{A_1}(p)$ and $\mu_{B_1}(v)$ are the membership functions for A_1 and B_1 , respectively. Then the strength of Rule 1 can be calculated by

$$\mu_1 = \mu_{A_1}(p_0) \wedge \mu_{B_1}(v_0). \quad (9)$$

The control output of Rule 1 is calculated by applying the matching strength of its conditions on its conclusion:

$$\mu_{C_1}'(r) = \mu_1 \wedge \mu_{C_1}(r) \quad (10)$$

where z ranges over the values the rule conclusion can take.

Similarly, for Rule 2, the truth value of p_0 is $\mu_{A_2}(p_0)$, the truth value of v_0 is $\mu_{B_2}(v_0)$, where $\mu_{A_2}(p)$ and $\mu_{B_2}(p)$ are the membership functions for A_2 and B_2 , respectively. The corresponding strength of Rule 2 will be

$$\mu_2 = \mu_{A_2}(p_0) \wedge \mu_{B_2}(v_0) \quad (11)$$

and the control output of Rule 2 is

$$\mu_{C_2}'(r) = \mu_2 \wedge \mu_{C_2}(r). \quad (12)$$

Thus, as a result of reading the feedback p_0 and v_0 from the vision system, two independent control actions are recommended. Rule 1 is recommending a control action with membership function $\mu_{C_1}'(r)$ and Rule 2 $\mu_{C_2}'(r)$. The inference engine or the inference process then produces

$$\mu_C(r) = \mu_{C_1}'(r) \vee \mu_{C_2}'(r) = [\mu_1 \wedge \mu_{C_1}(r)] \vee [\mu_2 \wedge \mu_{C_2}(r)], \quad (13)$$

where $\mu_C(r)$ is a piece-wise membership function for the combined conclusion of Rule 1 and Rule 2. From this membership function, a deterministic control action r_0 to apply to the plant is found by the following defuzzification process.

3.3. Defuzzification

The defuzzification process consists of two steps. The first step deduces a single-value control action from all the fuzzy actions. Each rule action specifies an output fuzzy value such as 'Robot_Speed_Change \tilde{R} is Positive_Medium (PM)', and a strength which indicates the truth value, for example, say $\mu(\text{PM}) = 0.25$. Here the membership function representing each fuzzy value is topped at the value of the truth value, μ , with which the rule fired. With all the control action combined, the resulting piece-wise membership function is produced as shown in Fig. 6 with the darkened line. The crisp value is determined in defuzzification as follows:

$$r_0 = \frac{\int r \mu_C(r) dr}{\int \mu_C(r) dr}, \quad (14)$$

where $\mu_C(r)$ is the piece-wise membership function of the combined control action at the time instant being considered; and r_0 is the defuzzified normalized crisp value of the control output. Integration is done over the entire universe of discourse. The second step transforms the resulting single fuzzy action into a crisp, executable system output. The actual control, the robot speed change, is given by

$$\delta v_r = k_U \times r_0 \quad (15)$$

where δv_r is the actual robot speed change in real crisp numbers, and k_U is a scaling gain.

4. Vision-guided controller system architecture

The vision-guided controller was built upon a flexible integrated vision system (FIVS) [12] developed at Georgia Institute of Technology. As shown in Fig. 7, the vision system has five basic functional modules.

1. An on-board computer consists of a microprocessor and its associated EEPROM, scratch RAM, and communication hardware.
2. A video head includes the imaging sensor—a charge coupled device (CCD), a high-bandwidth signal-conditioning amplifier, an analog-to-digital converter (ADC), and video RAM.
3. An optic system houses the lens (or simply a pinhole) and associated illumination.
4. An off-line host-interface allows the user to carry out off-line calibration, perform image analysis, and implement application-specific software through a host computer.
5. A real-time video record/playback allows failure-modes to be analyzed off-line. The DSP-based control board is designed to communicate with several option boards in parallel to tailor the system for a number of applications. Each of these option boards is controlled independently

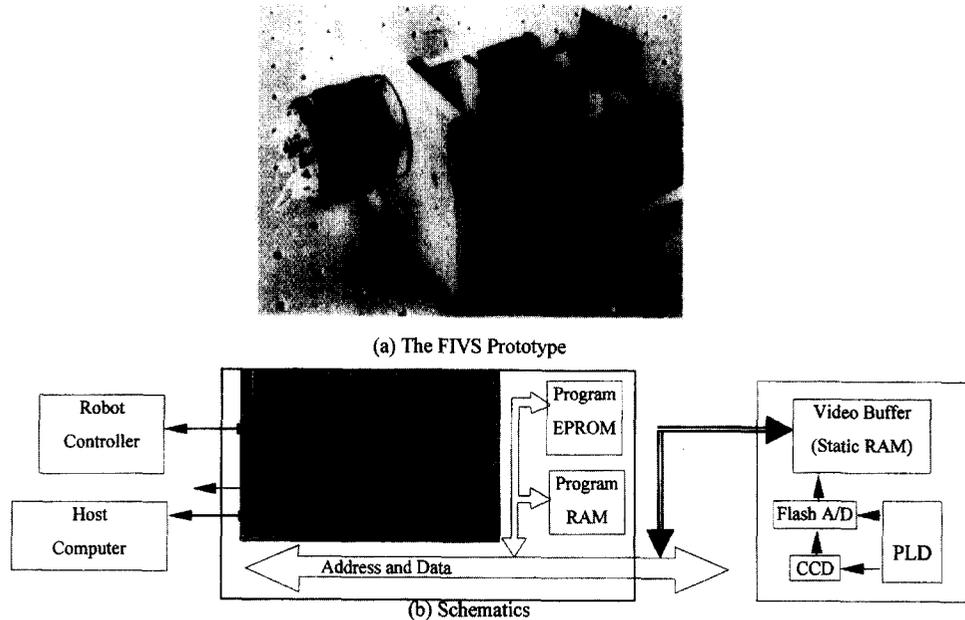


Fig. 7. The FIVS vision system: (a) the FIVS prototype; (b) schematics.

by a programmable logic device (PLD) which receives a peripheral select signal, a read/write signal, and an address signal from the microprocessor control board. Typical examples of the option boards for the FIVS are the digital video head, a real-time video record/display/playback board, and an expandable memory board.

The video head consists of a $m \times n$ CCD array, the output of which is conditioned by a high bandwidth amplification circuitry and sampled by a 'flash' ADC. The DSP-based control board provides direct software control of CCD array scanning and integration time, the intensity of the collocated illumination, and the real-time execution of a user-selectable vision algorithm imbedded in the EEPROM. In operation, the PLD decodes the control signals to initiate row shifts and column shifts in response to commands from the DSP-based control board. Particular row shifts and

column shifts enable the retrieval of a relevant area from an image. The PLD also provides control signals to ADC for performing the analog-to-digital conversion synchronized with row shifts, and enables the video buffer when the DSP reads or writes data to the VRAM.

Unlike conventional RS170-based systems or the early version of the LTS/IVS [13] which require pixel data to be stored in a video buffer before the processing of pixel data can commence, the FIVS design provides an option to completely by-pass the video buffer. Thus, the FIVS offers a means to process and/or to store the digitized pixel data by directly transferring the ADC output to the DSP. For real-time vision-based object tracking and motion control system applications, the scheme represents a significant saving in time and video buffer size required for processing an image. As an illustration, consider an image array of $m \times n$ pixels. The time needed to store the entire image (with no computation) in a memory at K MHz is $(m \times n)/K \mu\text{s}$ and requires $(m \times n)$ bytes of memory. Typical array size of a CCD ranges from 200×160 to 4096×4096 pixels. The corresponding video buffer and time required simply to store the entire image at a clock rate of 10 MHz would range from 32 kbytes to 16 Mbytes and 3.2-1600 ms, respectively! Clearly, the option to completely by-pass the video buffer offers a potentially useful solution to eliminate the frame storage prerequisite often required in conventional vision systems. Furthermore, this scheme completely eliminates the special hardware needed in acquiring the digitized pixel data for storage.

The main kernel of FIVS provides a user interface whereby the user can reprogram the EEPROM of the processor board. This allows the user to customize the image processing for a particular task, from a library of algorithms. Based on the hardware design, the software is

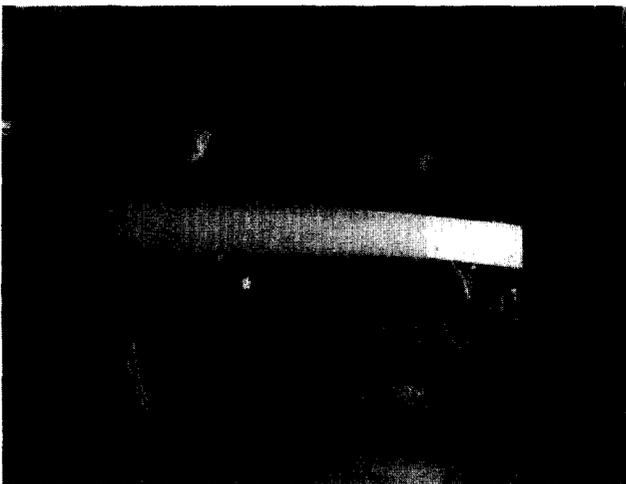


Fig. 8. Dyna-Slide vibratory feeder.

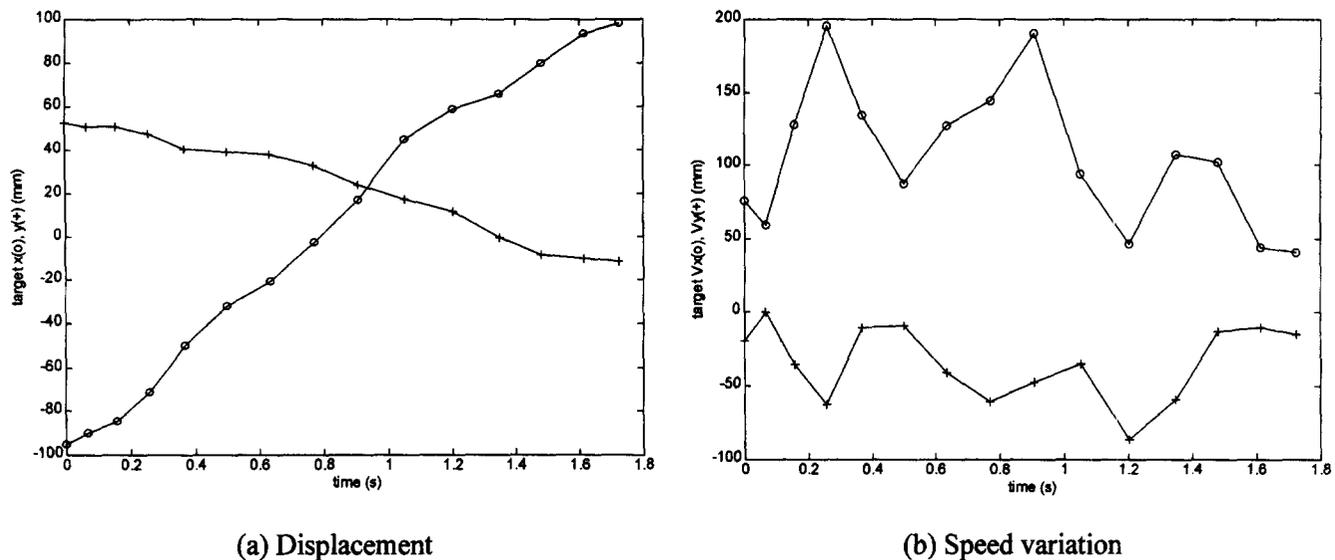


Fig. 9. The trajectories of a moving target in x and y directions: (a) displacement; (b) speed variation.

able to control the CCD array scanning and integration time, and the intensity of the collocated illumination. With the CCD under software control, partial frames can be 'captured' and processed instead of the customary full frame, reducing the cycle time required to capture and process an image. The ability to shift out partial frames is ideal for high speed tracking applications where the approximate location is known from a prior image.

5. Case study experimental results

Vibratory feeders are commonly used to separate, feed, or kit parts by industrial robots. As shown in Fig. 8, the feeder has resilient brushed surface to cushion and direct parts to move forward. Often, a dedicated fixture must be designed to adapt to a specific part shape to locate parts at the end of the feeder for pick-up by a pre-programmed industrial robot. More recently, vision systems are used to locate isolated objects on the vibratory surface for pick-up by robots. In most cases, the feeder must stop to allow the robot to pick up the objects. A vision-guided FLC system provides an effective alternative to pick up moving objects from vibratory surface. The vision-guided FLC, while eliminating the need of a dedicated fixture for each part family, does not require to stop the feeder for picking up the parts. The following case study illustrates the feasibility of applying the vision-guided controller to direct the robot (gripper) to the vicinity of the target for pick up.

The experimental setup consists of a vision system FIVS, an industrial robot, a vibratory feeder, and a host computer. The vision system was mounted on a six degree-of-freedom Cincinnati Milacron T³ 786 industrial robot. An Intel 486 PC serves as a host to coordinate the operation between the vision system and the robot via a RS232 serial communication. A Dina-Slide Vibratory feeder was used to circulate

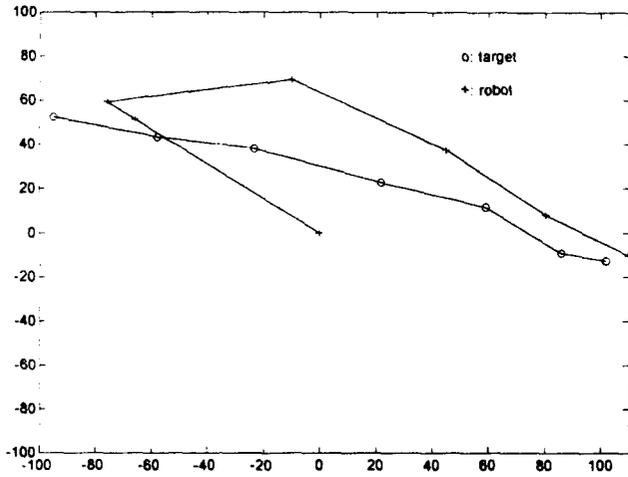
parts on a vibratory surface continuously. The maximum speed of parts on the feeder is approximately 300 mm s^{-1} .

A typical cycle of the dynamic pursuit operation is as follows. The vision system is positioned at a pre-specified location $(0, 0, 1\text{m})$ above the vibratory feeder such that the optical axis of the camera is perpendicular to the vibratory surface. The vision system repeatedly scans for a moving object to appear in its field of view (FOV), approximately $\pm 100 \text{ mm}$ in both x and y directions. Once it detects a moving target which is any object with an area within a specified range ($A_{\min} < A < A_{\max}$), the vision system computes the initial position and velocity of the moving object from two or more consecutive images. The target speed in x and y directions was smoothed out before being compared with the speed of the robot and sent to the FLC with the difference. That is:

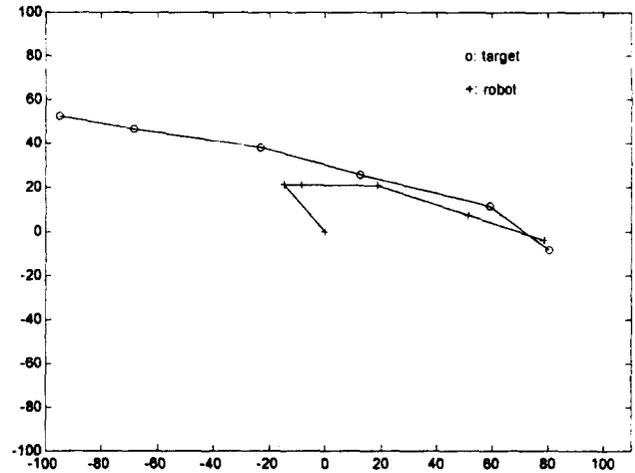
$$v_i'(i) = \frac{1}{2}[v_i(i) + v_i(i-1)] \quad (16)$$

where v_i is the target speed in either the x or the y direction; i represents the current sampling time instant, and $i-1$ the preceding sampling time instant. The computed data are then sent to the host computer that commands the robot to 'pursue' the object of interest via a DDCMP communication protocol. Unlike off-the-shelf video-based camera, FIVS performs on the board image processing and outputs the computed position of the target to the host computer. Typical motion of the moving target, recorded by the FIVS, is shown in Fig. 9.

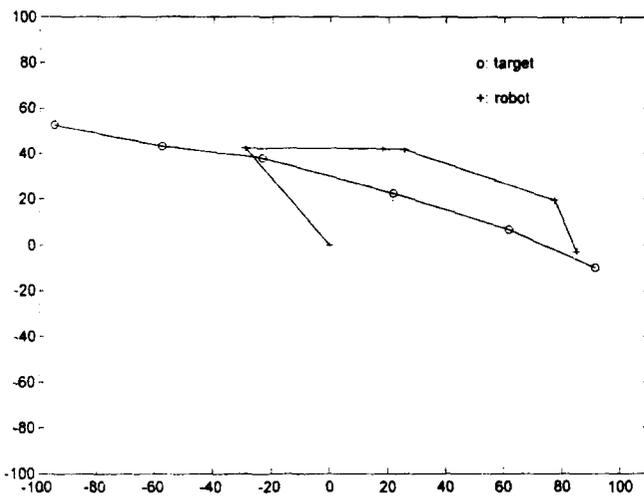
Fig. 10(a) to Fig. 10(d) display experimental results of different gain settings, where the cycle-time between adjacent data points is 0.3 s. The average velocity in each time step can be inferred from the spacing between adjacent data points in the plot. The scaling gain settings used in the experiments are summarized in Table 1. In these tests, the parameters of the membership functions, α , β , and γ , were



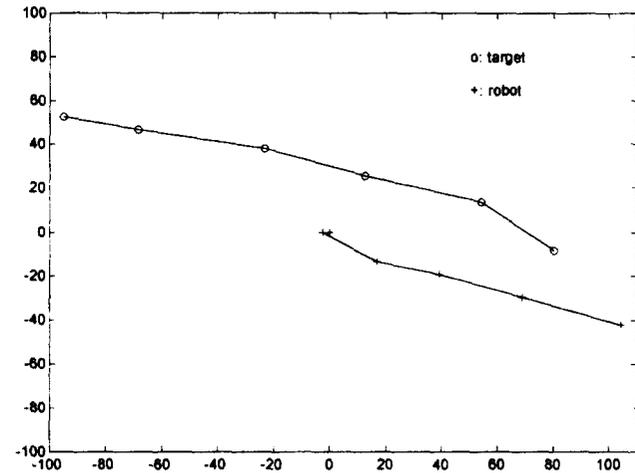
(a) Gain Setting 1



(b) Gain Setting 2



(c) Gain Setting 3



(d) Gain Setting 4

Fig. 10. Robot and target trajectory: (a) gain setting 1; (b) gain Setting 2; (c) gain setting 3; (d) gain setting 4.

chosen to be 2, 4 and 8, respectively. The maximum speeds of the robot, in x and y directions, were set to 200 mm s^{-1} .

As shown in Fig. 10(a) to Fig. 10(c), the robot has successfully approached its specified vicinity defined as

Table 1
Gain settings used in experiments

	K_p		K_D		K_u	
	X	Y	x	y	X	Y
Gain setting 1, Fig. 10(a)	20	20	80	40	24	12
Gain setting 2, Fig. 10(b)	20	20	40	20	24	12
Gain setting 3, Fig. 10(c)	20	20	40	20	40	20
Gain setting 4, Fig. 10(d)	40	40	40	20	24	12

K_p , K_D and K_u represent scaling gains; X , x , Y and y refer to the direction of motion.

$\pm 5 \text{ mm}$ in both x and y directions. In most cases, the robot managed to reach the specified vicinity of the target in less than 2 s. As shown in Fig. 10(a), the robot was initially at rest when the target appeared in its FOV. The FIVS determined the initial position and velocity of the target based on a sequence of two images. The FLC predicted the next target position based the information fed back from FIVS and commanded the robot to accelerate from rest to the specified maximum speed. The initial coarse motion control resulted in a rapid reduction of position and velocity differences as desired. However, the high velocity scaling gain K_D results in a significant momentum and thus a significant overshoot in position and exhibits a relatively large velocity difference at the end of its 1.8 s pursuit operation. The effect appears to be similar to the classical proportional control of a velocity servo. Hence, the velocity scaling gain K_D was scaled to half of the original value while other gains remained unchanged (see Gain Setting 2 in Table 1) and the corresponding performance is illustrated

in Fig. 10(b). The reduction in K_D essentially eliminated the overshoot during the initial coarse motion control, and the robot was able to complete its pursuing task in 1.5 s with a velocity difference of $\pm 5 \text{ mm s}^{-1}$. The scaling gain K_U provides an incremental speed change to the robot. Fig. 10(c) shows the trajectory where the gain K_U was increased by a factor of 1.67, the system exhibits similar position and velocity overshoot as the FLC with a high velocity scaling gains. This result has demonstrated a similar effect of a classical proportional controller of a velocity servo. Fig. 10(d) shows an attempt to double the position scaling gain K_P from that of gain setting 2. It was found that K_P plays a role similar to the integral gain in a typical velocity servo. It was observed that near-zero velocity difference was achieved in about 1.5 s in both x and y directions. However, since end-point position control of the robot gripper was not a control objective, a significant offset in position was found throughout the chase.

In the course of design and simulation, membership functions, control rules, defuzzification methods, etc. were defined or determined based on the understanding of the controlled system or process. It is expected that adjusting the range and shape of each membership function can further fine-tune the system. Furthermore, it is also expected that a derivative term can be added to provide an additional freedom in tuning the control performance.

6. Conclusions

A vision-guided fuzzy logic controller for robotic dynamic pursuit of a moving object in Cartesian space has been developed. Built upon a non-conventional vision system developed at Georgia Institute of Technology, experiments on parts moving on an industrial vibratory feeder have shown that the vision-based fuzzy logic controller can effectively guide an industrial robot to follow highly nonlinear motion of a moving target and approach its vicinity. A case study based on an industrial part kitting system has demonstrated a potentially useful solution to kitting moving parts on vibratory feeder without relying on object-dependent fixtures. Since the control effort requires only the change in robot end-point velocity in Cartesian space, the technique can be readily applied to off-the-shelf industrial robots without having to change the system hardware. As demonstrated experimentally, the control

parameters of the vision-guided FLC system can be adjusted following similar guidelines for classical PID controller.

Acknowledgements

This research was partly supported by the National Science Foundation (NSF) through the NSF Presidential Young Investigator Grant No. 8958383 and the Woodruff Faculty Fellow. The donation of the Small Parts Kitting Cell by General Motor Company in supporting this work is greatly appreciated.

References

- [1] Singh, A., *Optic Flow Computation: A Unified Perspective*, IEEE Computer Society Press, Los Alamitos, CA, 1991.
- [2] J.K. Aggarwal, N. Nandhakumar, On the computation of motion from sequences of images—a review, *Proceedings of the IEEE* 76 (8) (1988) 917–935.
- [3] Ruo, R.C., Mullen, R.E. and Wessell, D.E., An adaptive robotic tracking system using optical flow, in: *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, Philadelphia, Pennsylvania, 1988.
- [4] Allen, P.K., Yoshimi, B. and Timcenko, A., Real-time visual servoing, in: *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA, April 1991, pp. 851–856.
- [5] Papanikolopoulos, N.P. and Khosla, P.K., Robotic visual servoing around a static target: An example of controlled active vision, in: *1992 American Control Conference*, Chicago, IL, 1992.
- [6] Weiss et al., 1987. Author, please provide details.
- [7] J.T. Feddema, O.R. Mitchell, Vision-guided servoing with feature-based trajectory generation, *IEEE Transactions on Robotics and Automation* 5 (5) (1989) 691–700.
- [8] Zhang, D.B., Gool, L.V. and Oosterlink, A., Stochastic predictive control of robot tracking systems with dynamic visual feedback, in: *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, Cincinnati, OH, May 1990.
- [9] Koivo, A.J. and Houshang, N., Real-time vision feedback for servoing robotic manipulator with self-tuning controller, in: *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 1, 1991, pp. 134–142.
- [10] L.A. Zadeh, Fuzzy sets, *Information Control* 8 (3) (1965) 338–353.
- [11] A.P. Klimley, The predatory behavior of the White Shark, *Am. Sci.* 82 (1994) 122–133.
- [12] Lee, K.-M. and Blenis, R., Design and implementation of a flexible integrated vision system, in: *Machine Vision Conference 92*, Atlanta, GA, June 1992.
- [13] Dickerson, S.L. and Lee, K.-M., Image reading system, U.S. Patent No. 5,146,340, 8 September 1992.



Dr. Kok-Meng Lee received his BS degree in mechanical engineering from the State University of New Buffalo in 1980 and earned his MS and Ph.D. degrees in mechanical engineering from the Massachusetts Institute of Technology in 1982 and 1985, respectively. He is an Associate Professor and a Woodruff Faculty Fellow in the George W. Woodruff School of Mechanical Engineering at Georgia Institute of Technology. Dr. Lee's research interests include system dynamics and control, robotics and automation, optomechatronics, and multimedia in manufacturing education.

Dr. Lee has been an active member of the ASME Dynamic System and Control Division and of the IEEE Robotics and Automation Society. He has served as an Associate Editor for the IEEE Transaction on Robotics and Automation and its Society Magazine and as a Local Arrangement Chairperson in the 1993 IEEE International Conference on Robotics and Automation. He currently serves as an Editor for the IEEE/ASME Transaction on Mechatronics and as a General Co-Chair for the 1997 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. He is a senior member of SME. Other recognition of his contributions includes the National Science Foundation (NSF) Presidential Young Investigator Award, Sigma Xi Junior Faculty Research Award, International Hall of Fame New Technology Award. He holds five patents in machine vision, three degree-of-freedom variable reluctance spherical motor, and optical orientation encoder.



Yifei Qian received his Ph.D. in mechanical engineering in 1995 from the Georgia Institute of Technology, Atlanta, USA, and his MS in mechanical engineering in 1986 and his BS in aeronautical engineering in 1983 from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, respectively. From 1986 to 1990, he worked as an instructor in the Department of Mechanical Engineering at the Nanjing University of Aeronautics and Astronautics, partly teaching and partly doing research in the field of microprocessor based numeric controls of electro-hydro-mechanical systems.

From 1995 to 1997, he worked in Motorola Manufacturing Systems as an application engineer to develop automated manufacturing, measuring, and testing systems with applications of robotics and machine vision. He joined Lucent Technologies, Inc. as a Member of Technical Staff of Bell Lab in late August 1997. His current interests of research and development include statistical process and quality monitoring and controls, robotics and machine vision applications in optic fiber preform manufacturing. He has been a member of ASME since 1991.