Robotica (2001) volume 19, pp. 53–58. Printed in the United Kingdom © 2001 Cambridge University Press

Complementary data fusion in vision-guide and control of robotic tracking

*Y.M. Chen, **C.S. Hsueh

(Received in Final Form: March 24, 2000)

SUMMARY

We present a data fusion control scheme for the hand-held camera of the SCORBOT-ER VII robot arm for learning visual tracking and interception. The control scheme consists of two modules: The first one generates candidate actions to drive the end-effector as accurate as possible directly above a moving target, so that the second module can handily take over to intercept it. The desired camera-joint coordinate mappings are generalized by Elman neural networks for a tracking module. The intercept module then determines a suitable intercept trajectory for the robot within the required conditions. The simulation results support the claim that it could be successfully applied to track and intercept a moving target.

KEYWORDS: Data fusion; Visual control; Neural network.

1. INTRODUCTION

Data fusion is an estimation technique, which optimally extracts desired information from redundant measurement data sources.[1–3] Many different algorithms have been used to combine information from vision, ultrasonic force and other tactile sensors to provide the robot with a complete picture of its unknown and changing environment. The increasing use of multiple sensors in robotics provides robots with growing "intelligence" in the sense that the robots can compensate for changes in the environment and uncertainties in models without an explicit intervention or re-programming. In this paper, the proposed neural network integrates both vision and joint information in an appropriate way to obtain better information. The vision system is used to monitor the difference between the end-effector and the moving target. The generated control signals drive the manipulators for tracking the target. To accomplish a task of this kind, both of two disparate sensors (i.e. joint and visual sensors) are used for complementary observed data rather than redundant ones as in the usual way. In the recent past, visual servoing problems have shown that vision can greatly improve the process of image interpretation and vision-based control.[4–6] Although significant advances have been made in the visual servo control, much of its potential is still unrealized in robot control.[7,8] In general, vision-based control systems require explicit knowledge of the kinematics and dynamics of a robot and a careful calibration of the sensor system. The disadvantages of the vision-based approach are that the control goal is hard to specify because of the systems nonlinear dynamic behavior and changing camera configuration. We present a data fusion control scheme that will allow a robot to respond to multisensor input and catch a moving target. The design of the scheme decomposes the task into two modules: Learning to generate hand-eye coordinate tracking actions and generating a kinematically safe trajectory to intercept the moving target. The Elman neural network[9] has been adopted as a tracking module, which maps the multisensor information directly into the motor commands. The data fusion control scheme is used to control the SCORBOT-ER VII robot arm to track and then to intercept a moving target.

The rest of the paper is organized as follows: Section 2 discusses the proposed tracking module for using the neural network based approach, followed by the interception module development in Section 3. The conclusions of the preceding sections lead to a data fusion control scheme, described in Section 4. Results for both modules in the proposed control scheme applied to a robot are given in Section 5.

2. TRACKING MODULE

In this paper a neural network approach tries to track a moving target on a table. The target was assumed to be moving with a constant velocity. In the tracking case, the task is restricted to 2D movements where the camera is always looking down from a constant altitude. The joint angle (Figure 1) is not independent but expressed as a

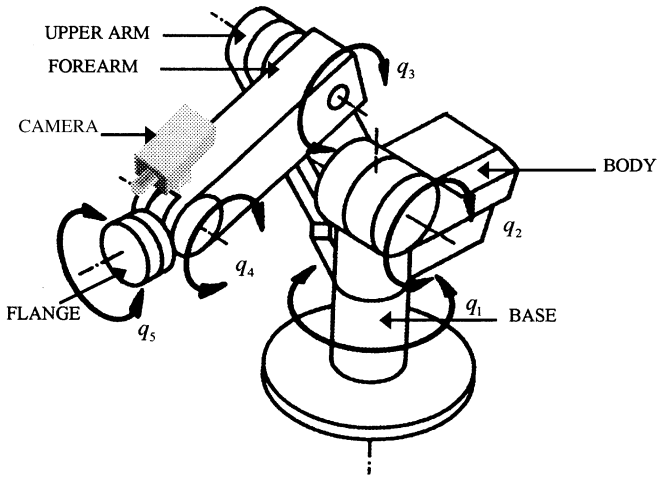


Fig. 1. SORBOT_ER VII Robot.

*Department of Electrical Engineering, Lee Ming Institute of Technology, Taipei 243, Taiwan (R.O.C.).
Email: ymchen@ccit.cc04.edu.tw
**Chung Shan Institute of Science & Technology.
Email: g881707@cc04.ccit.edu.tw

https://doi.org/10.1017/S0263574700002897 Published online by Cambridge University Press

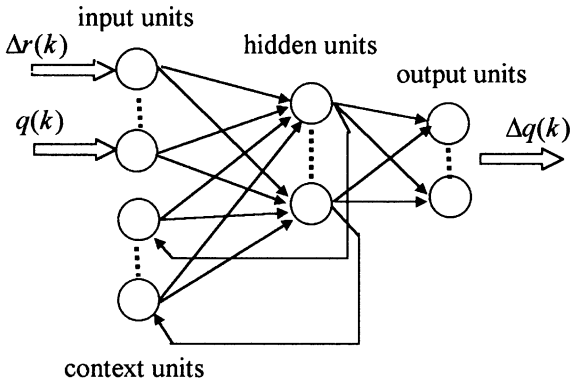


Fig. 2. Elman network for tracking module.

function of q_1, q_2 and q_3. The joint angles q_4 and q_5 are kept constant, resulting in q_1, q_2, and q_3 as the three degrees of freedom of the manipulator. The proposed neural network integrates the current servomotor status q_1, q_2, and q_3 with the visual information, which is presented in the next sections.

2.1 Neural controller for tracking

The task of the neural controller is to minimize the observed position r = (x, y) of the target relative to the center of the camera image. The neural controller is an Elman network (Figure 2) with inputs consisting of the visual information Δr (Δx and Δy of the target), which is compared with the center of the image, and of the current servomotor status q (q_1, q_2, and q_3). The inputs are connected via 10 hidden neurons to the three outputs, which give the displacement vector $\Delta q(k)$ (Δq_1, Δq_2, Δq_3) of the manipulator joints.

$$\Delta q(k) = NN_t(\Delta r(k), q(k)) \tag{1}$$

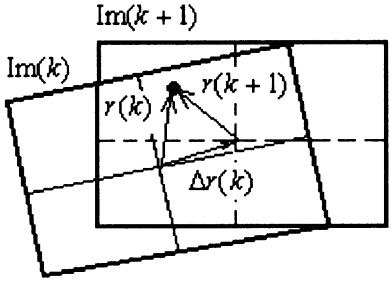


Fig. 3. Two successive image frames before and after the movement of the end-effector.

The desired x-y coordinate values of the target in the camera picture are obtained in advance by moving the end-effector to the desired point. From the target coordinate before the movement r (k) and the coordinate after the movement r ($k+1$) the translation component $\Delta r(k)$ of the coordinate transformation matrix can be obtained. Figure 3 shows two successive camera images.

The position of the target in the coordinate frame of image Im (k + 1) can be expressed in the coordinate frame of image Im (k):

$$r(k) = R_{k+1}^{k} r(k+1) + \Delta r(k) \tag{2}$$

where R_{k+1}^{k} denotes the rotation matrix of frame $k+1$ with respect to frame k, and $\Delta r(k)$ denotes the translation vector of frame $k+1$ with respect to frame k. It is exactly this translation which is needed to bring a target located at $\Delta r(k)$ to the center of the camera image. Therefore, the neural controller's aim is to drive the joint displacement Δq which results in r ($k+1$) = 0. Since the rotation is only about the z-axis, the R_{k+1}^{k} can be calculated from q_1 (the rotation about the robot base). Therefore, $\Delta r(k)$ can be derived. Both $\Delta r(k)$ and $q(k)$ as inputs together with the outputs $\Delta q(k)$ yield a learning pattern. These learning patterns are used in the dynamic backpropagation algorithm[10] to minimize the sum square error of the overall learning samples:

$$E = \frac{1}{2} \sum_{i} [(\Delta q_d - \Delta q)\,(\Delta q_d - \Delta q)^T]^2 \tag{3}$$

where Δq_d is the i-th sampling of the desired reference to the manipulator, and Δq is the i-th sampling generated reference by the neural network.

2.2 Intercept module for pregrasping

The intercept module allows the manipulator to reach the intercept location within the work envelope. The task of the module is to generate a robot joints displacement which forces the robot manipulator to follow a trajectory leading to the goal state in which the end-effector is placed on the target ready for grasping. The trajectory generation strategy in this research is described by a fifth-order polynomial function (f).[11] Suppose that a catching decision is made by the switching mechanism at time t_1 when the target is at location r_1. Assume that reaction time required by the robot

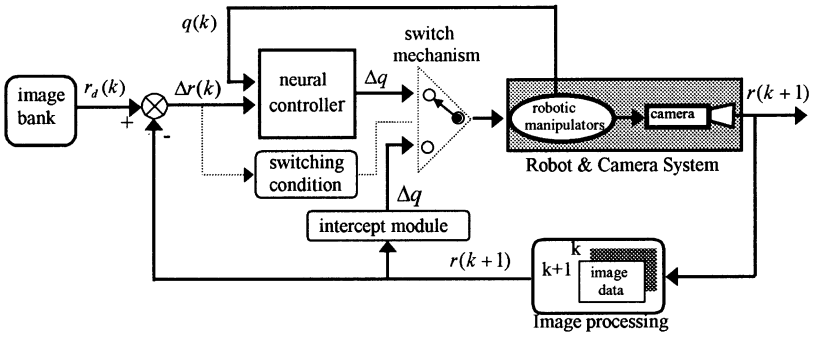


Fig. 4. The data fusion control scheme for tracking and interception.

https://doi.org/10.1017/S0263574700002897 Published online by Cambridge University Press

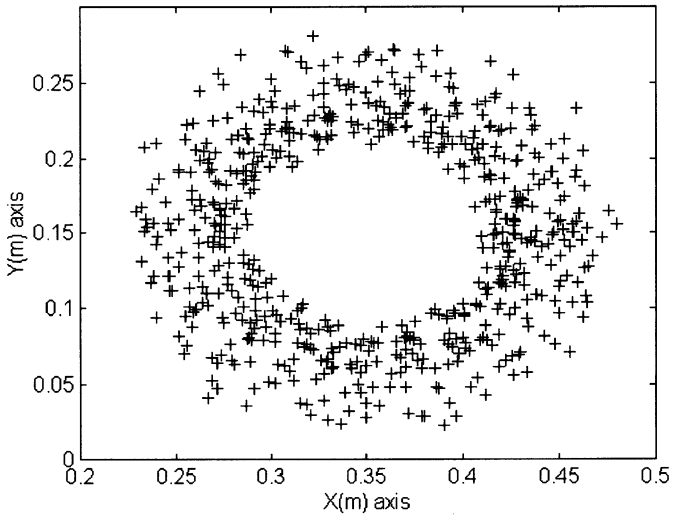


Fig. 5. The training patterns for neural network.

is Δt, and t_2 the time when the target must be intercepted at location r_f. The r_1 and r_f must be converted from Cartesian space to the joint space, q_f and q_i, respectively. Then, we have

$$t_2 = t_1 + \Delta t \quad \Delta q = f(q_f, q_i, \Delta t) \tag{4}$$

The reaction time Δt is made up of the communication times among the robot processing times. In addition, the velocity must become zero at the moment of pregrasping. Therefore, the interception criterion $\dot{q}_f = 0$ will be taken into consideration for Δt.

3. THE DATA FUSION CONTROL SCHEME FOR TRACKING AND INTERCEPTION MODULES

To accomplish the desired task, a data fusion control scheme is developed that consists of two modules, which are described in the previous sections. In this scheme, the tracking module tracks the target motion. The learning of this module is realized on-line to further update the weights of the Elman network. The adaptation phase ensures the necessary coherence between the location of the visual position in the present camera picture and those in the previous picture (desired picture). Therefore the inputs of the Elman network are $\Delta r (= r_d - r)$ and q. The on-line adaptation of the neural controller could lead to excellent results. Once the intercept location is met, the intercept module gives the proper trajectory parameters to generate a safe trajectory within the required switch conditions. The switch mechanism depends on the switch conditions of the two modules, as shown in Figure 4.

<Switching conditions>

$$\| r_f - r_i \| \le \| p_f \| \tag{5}$$

If the condition (5) is satisfied then the intercept module is used, otherwise NN_t is used. p_f is the desired final position difference between the robot and the moving target. Moreover, the manipulator must intercept the target within a specified time, namely:

$$\sum_{i=1}^{n} t_i \le \Delta t \tag{6}$$

where n denotes the number of pursuing steps, t_1 the time the pursuing step i takes, and Δt the upper bound on the time spent on the intercept process.

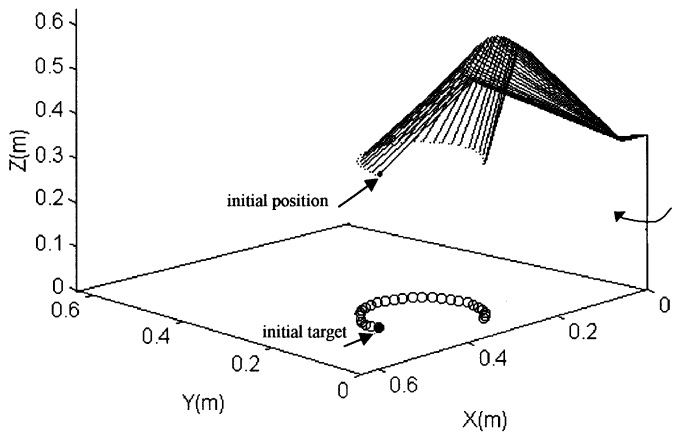


Fig. 6. The tracking of circular trajectory of a first three-links robot.

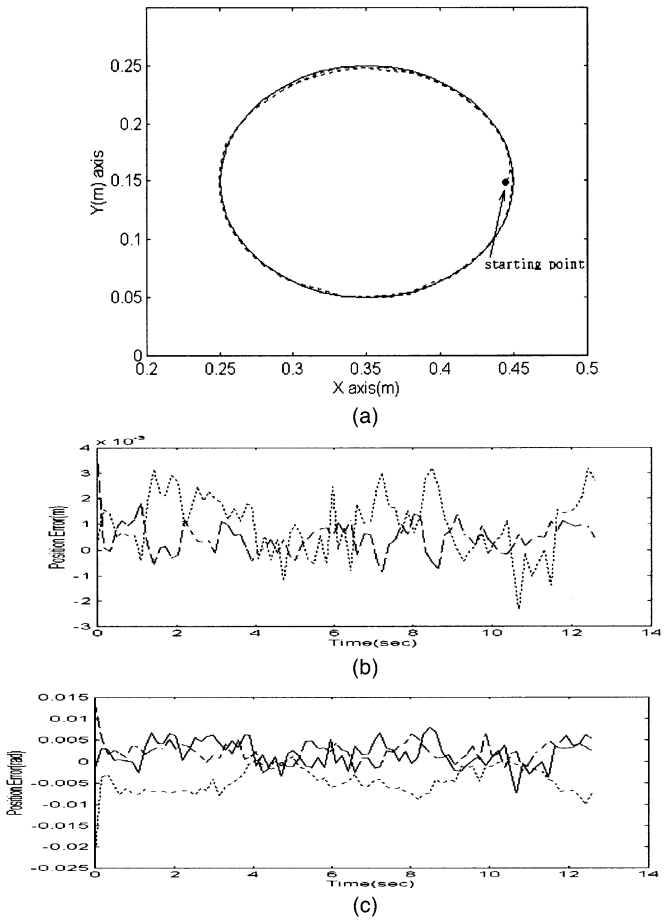


Fig. 7. Tracking path and tracking error plots of the robotic tracking
(a) Tracking path
(b) End-effector tracking position errors (... x axis error; --- y axis error)
(c) Joints tracking position errores (___ joint 1; --- joint 2; ... joint 3)

https://doi.org/10.1017/S0263574700002897 Published online by Cambridge University Press

After this change has been executed by the manipulator, the camera image will be checked again and the above procedure will be repeated if necessary.

4. SIMULATION STUDIES

The scheme is tested on a simulated 3 DOF SCORBOT-ER VII robot. Two target motions on the table serve as test cases: One follows the circular trajectory with different radii and the other follows a new trajectory-square trace. To accomplish visual control, we created synthetic images that emulate real-world images. Each pixel's intensity is corrupted by the white noise in the synthetic images. The test cases are implemented with 10% white noise. A three-link robot manipulator, whose parameters are taken from the first three links of SCORBOT-ER VII robot, and a 1 cm focal length lens of the camera are used for simulation studies. Without loss of generality, we assume the sampling time being 1 sec. The network was trained using 615 random patterns on a circle centered at (0.35, 0.15) m and the radii were varied in the range of 0.066 m to 0.125 (Figure 5). The target moved along a circle of the radius of 0.1 m at 0.1 m/sec. The sequence of tracking action is shown in Figure 6. The end-effector tracking results are plotted in the x-y plane of the world coordinate as depicted in Figure 7(a) which shows the nearly circular trajectory achieved. The corresponding end-effector tracking position errors and joint position errors of the joint 1, 2 and 3 for 2 cycles are shown in Figure 7(b), (c). It is clear that the learning process has the ability of convergence and tracking performance.

Once the system's performance meets our minimal requirements, further tracking simulations were tested using the various trajectory. Testing proceeded with reference to a target, which traced either an alternative radius circle or a square. In the first tracking, we allowed the simulations to continue for the circle of various radius ratios in order to determine the repeatability of the system. The second traced was a square; it demonstrated robust tracking in spite of the target exhibiting discontinuities in the velocity curves (at the corners of the square). The results of the manipulator tracking are shown in Figure 8. The simulation tests demonstrate the fast adaptive rate of the neural controller for a meaningful on-line application. These were confirmed again by the tracking performance of the neural controller.

If the switching conditions are met, the robot execution process changes from a tracking module to an interception module. The generated intercept trajectory corresponding to the joint angles, speeds and accelerations for the joints 1, 2 and 3 are shown in Figure 9. The animated manipulator intercept action follows the joints motor commands to successfully intercept the target illustrated in the 3D space, as shown in Figure 10.

5. CONCLUSIONS

The basis for developing an intelligent robot system has been discussed. The problem has been formulated in the

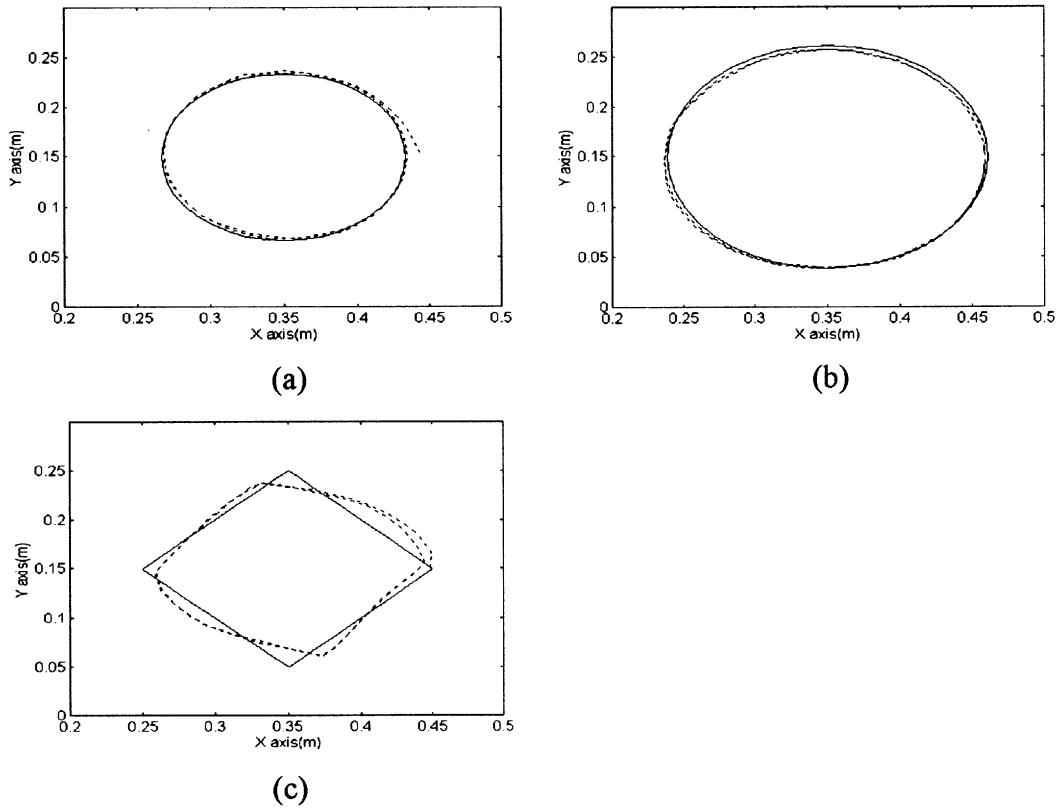
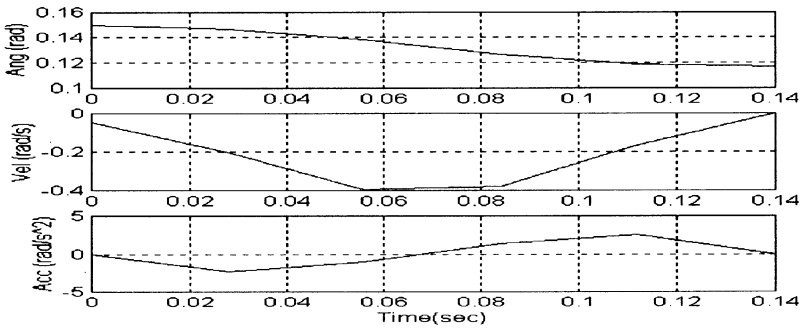
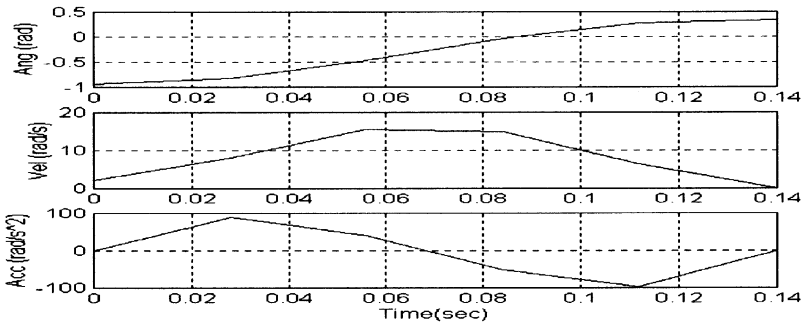


Fig. 8. Tests for different paths
(a) $r_d / r_t = 0.83$
(b) $r_d / r_t = 1.11$
(c) $A_{square} / A_{circle(r_t = 0.1)} = 0.73$
[r_d = the radius of the new desired circle; r_t = 0.1 m (the radius of the trained circle in the previous example)].

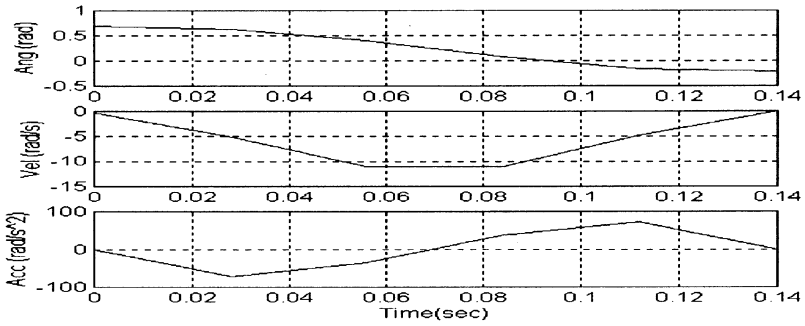
https://doi.org/10.1017/S0263574700002897 Published online by Cambridge University Press



(a) Joint1



(b) Joint 2



(c) Joint 3

Fig. 9. Interception path of first three joint angle, speed, and acceleration versus time.

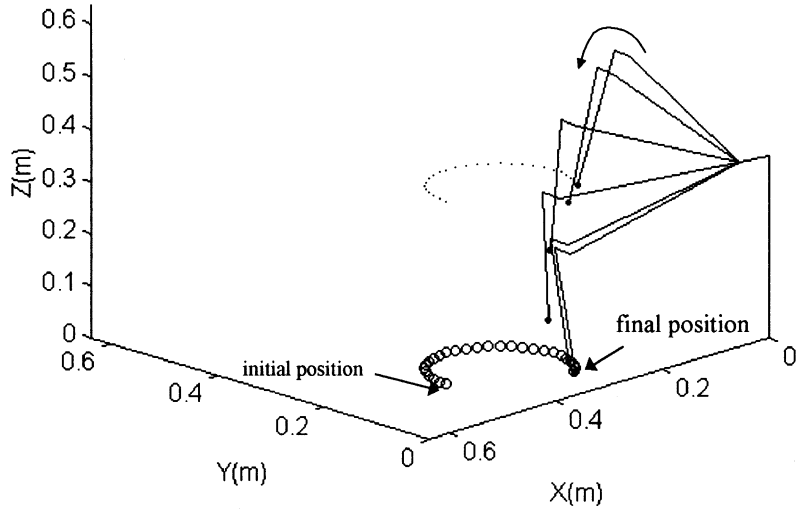


Fig. 10. Intercept animation after a circular tracking motion ("○" indicates target motion; "..." end-effector tracking path; and "●" interception path)

https://doi.org/10.1017/S0263574700002897 Published online by Cambridge University Press

context of prey capture with the robot as a pursuer and the moving target as a passive prey. By adding an "eye" to the robot, the system gains the ability to locate and intercept a target in subsequent processes. Systems that fuse data from multiple sensors are generally based on statistical techniques and are often computationally intensive. However, the neural network that processes the multiple sensors signals performs the required tasks more efficiently. This increase in efficiency arises from the learning capability and its nonlinear mapping function. The work has been presented with a specific neural structure. The modules in the scheme open perspectives to extend it to a more complex architecture.

References

1. H.F. Durrant-Whyte, 'Integrating distributed sensor information, an application to a robot system coordinator', *Proc. IEEE Int. Conf. Syst., Man, and Cyber.* (1985) pp. 415–418.
2. M. Hashimon and R.P. Paul, 'Integration of multisensor manipulator actuator information for robust robot control systems', *Proc. Annual Conf. Japan Robotics Society* (1987) pp. 293–396.
3. R.C. Luo, M.H. Lin and R.S. Scherp, 'The issues and approaches of a robot multisensor integration', *Proc. IEEE Int. Conf. Robotic and Automation* (1987) pp. 1941–1946.
4. H.L. Alexander, K. Eberly and H. Weigl, 'Experiments in Vision-Based Control of a Neutrally-Buoyant Free-Flyer', *Proceedings of SPIE Intelligent Robots and Computer Vision X: Algorithms and Techniques.* (November, 1991) **Vol. 1607**, pp. 689–699.
5 P. Allen, B. Yoshimi, A. Timcenko and P. Michelman, 'Hand-Eye Coordination for Grasping Moving Objects', *Proceedings of SPIE Sensor Fusion III: 3-D Perception and Recognition* (November, 1990) **Vol. 1383**, pp. 176–188.
6. N. Houshangi, 'Control of a Robotic Manipulator to Grasp a Moving Target Using Vision', *Proceedings of the IEEE International Conference on Robotics and Automation* (May, 1990) pp. 604–609.
7. C. L. Boddy, 'A Real-Time Trajectory Planner and End-Effector Collision Control System for a Robotic Arm', *Proceedings of the IEEE International Conference on Control '91* (March, 1991) pp. 69–74.
8. J. Woodfill and R. Zabih, 'A Real-Time Vision System for Robots in Unstructured Domains', *Proceedings of SPIE Sensor Fusion IV: Control Paradigms and Data Structures* (November, 1991) **Vol. 1611**, pp. 346–355.
9. Duc Truong Pham and Liu Xing, *Neural Networks for Identification, Prediction and Control* (Springer-Verlag, London, 1995).
10. D.T. Pham and X. Liu, 'Dynamic system modelling using partially recurrent neural networks'. *J. Systems Engineering*, **2**(2), 90–97 (1992).
11 R.P. Paul, *Robot Manipulators: Mathematics, Programming, and Control* (MIT Press, Cambridge, Massachusetts, 1981).

https://doi.org/10.1017/S0263574700002897 Published online by Cambridge University Press