

IAS 4

A monocular robot arm can be neurally positioned

P. van der Smagt and F. C. A. Groen and B. J. A. Kröse

Department of Computer Systems, University of Amsterdam

Kruislaan 403, 1098 SJ Amsterdam, THE NETHERLANDS

email: smagt@fwi.uva.nl

phone: +31 20 5257524 fax: +31 20 5257490

July 25, 1994

1 Introduction

Robot systems mostly depend on static visual information, taking into account only the *positions* of points of interest. On the other hand, living organisms use continuous feedback from their eyes in order to interact with their dynamically changing environment. In the meantime, there is sensory activity due to ego-motion which is taken care of.

In this paper it is investigated how visual feedback can be incorporated in a monocular robot arm control system, in order to position the arm directly above an object while the system must be at rest: the velocity, acceleration, and higher derivatives must also be zero. It will be shown that by using time derivatives of the visual data, criteria can be developed which specify a trajectory which ends in a rest state (i.e., zero velocity and higher derivatives) at the end point. These criteria will be the visual setpoints along the followed trajectory. Thus it is possible that the eye-in-hand robot arm exactly stops on an observed object by use of optic flow, without having visual depth information.

The use of optic flow for robot navigation is no novelty. Cipolla and Blake [1], for example, describe a system for estimating time-to-contact from the flow-field, and use this for obstacle avoidance of a robot arm with known kinematics. Sharma [2] uses time-to-contact derived directly from the optic flow as a basis for mobile robot trajectory planning. Vernon and Tistarelli [4] use visual velocity vectors to estimate depth in a complex visual scene, but again for a robot with known kinematics. In this paper, a method is

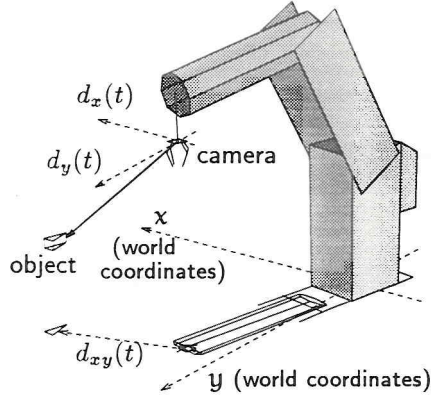


Figure 1: The robot tries to grasp an object. The object is observed by the camera relative to the coordinate system of the end-effector. Subsequent measurements of the object occur at subsequent time steps. The figure also shows the projections of the vectors and objects on the $z = 0$ plane.

presented which does not have any knowledge of the robot at the kinematic level; instead, a self-learning neural controller learns the camera–robot mapping directly from the behaviour of the robot, without any prior knowledge.

2 System and problem description

The robot system used is a 6 DoF anthropomorphic robot arm, as depicted in figure 1. The length of the lower and upper arm of the robot are 50 and 51 cm, respectively. The camera, as depicted in the figure, is situated in the end-effector. By controlling the wrist joints 4, 5, and 6, the camera is always kept looking down: only joints 1, 2, and 3 are used for positioning.

The task of the controller is to move the robot arm to a state where the distance $d(t)$ between the camera and the object is zero, and the arm is at rest. This can be formally specified as,

$$\forall k \geq 0 : \quad d^{(k)}(\tau_d) = 0, \quad (1)$$

i.e., the $d(t)$ and its time derivatives must be zero at some user-specified τ_d , which we call the desired *time-to-contact*. No detailed model of the robot, camera, or object is available; the robot–camera mapping must be autonomously learned by the neural controller from the behaviour of the system.

3 The solution

In order to solve the posed problem, we decompose the distance $d(t)$ in finite Taylor series for its x , y , and z components, such that

$$d(t) = a_0[i] + a_1[i]t + \dots + a_n[i]t^n \quad (2)$$

where the a_j 's are different for the x , y , and z . The index i indicates a feedback step, i.e., $a_0[i]$ is the trajectory parameter a_0 measured in time slot i . Then the stopping criterion (1) for finite n

$$\forall k, 0 \leq k < n: \quad d^{(k)}(\tau_d[i]) = 0 \quad (3)$$

leads to the following constraint on the parameters of the trajectory:

$$\frac{a_0[i]}{a_1[i]} = -\frac{\tau_d[i]}{n}. \quad (4)$$

A proof of this assumption is available.

Equation (4) shows that, by putting restrictions on the parameters of the motion, the deceleration profile can be controlled. By repeatedly, in subsequent time steps, adhering to (4), deceleration leading to the stopping criterion can be guaranteed. Note that the value of a_0 , which is the position of the camera relative to the object at $t = 0$, and a_1 , which is the camera's velocity, need not be known, but rather their ratio. Without knowledge of the dimensions of the object, a_0 and a_1 can indeed not be measured with a single camera. However, from the change of the observed area of the object, a_0/a_1 is a measurable quantity, and the desired values in (4) can be used as visual setpoints for a_0/a_1 along the trajectory.

4 Controller structure

The neural controller for this experiment consists of an 15-input feed-forward neural network with 35 hidden units in a single layer. The neural controller \mathcal{N} performs the mapping

$$\mathcal{N}\left(\theta[i], \dot{\theta}[i], \ddot{\theta}[i], a_{0x}[i]/a_{1x}[i], \tau_d[i]/n, a_{0y}[i]/a_{1y}[i], \tau_d[i]/n, a_{0z}[i]/a_{1z}[i], \tau_d[i]/n\right) = \ddot{\theta}[i+1] \quad (5)$$

where $\theta[i]$ are the joint values for joints 1, 2, and 3, and ditto $\dot{\theta}$ and $\ddot{\theta}$. The visual input to the neural network consists of the measured a_0/a_1 in the x , y , and z direction, and their desired values $\tau_d[i]/n$. Once

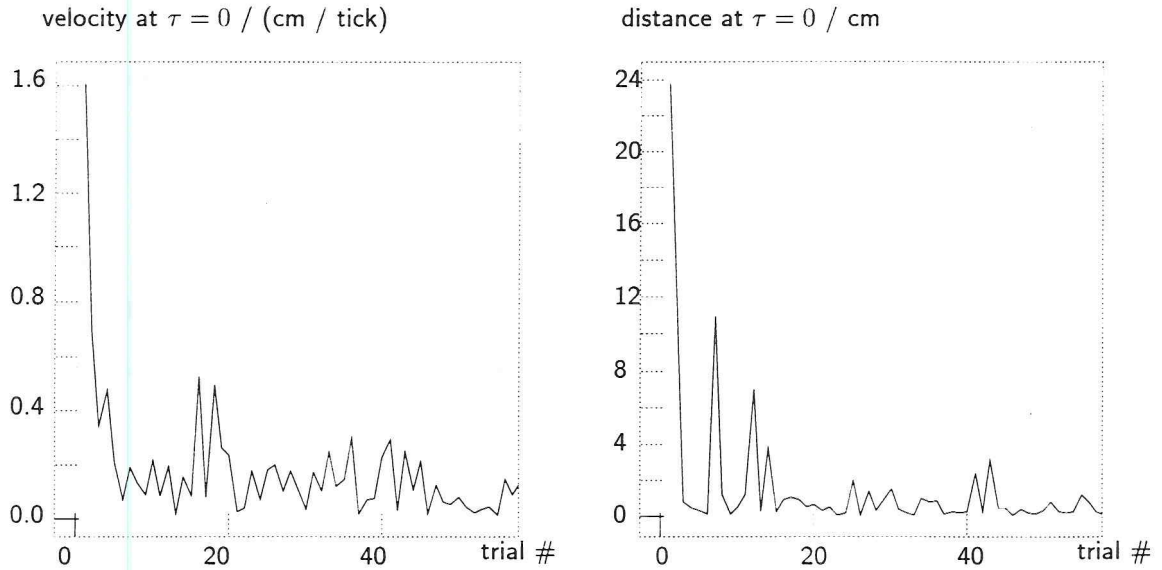


Figure 2: The left figure shows the velocity of the end-effector in cm/tick at $\tau = 0$. Typical end-effector velocities are between 0.5 and 2.0. The right figure shows the distance between the end-effector and the approached object at $\tau = 0$. The horizontal axis indicates the trial number. A typical deceleration path, in which the whole reach space of the robot can be transversed, takes 30 ticks.

the transition $i \rightarrow i + 1$ has been made, the actual visual transition $a_0[i]/a_1[i] \rightarrow a_0[i + 1]/a_1[i + 1]$ is available and can be used as a learning sample. The learning method used is a conjugate gradient optimisation algorithm [3].

5 Results

The method is applied to a simulated robot as described above. The results in figure 2 show that after only a few trials, the neural network is able to generate joint accelerations which lead to the optimal path towards the object. Implementation on a real robot arm is currently being realised. Due to the fact that only the first derivative of the visual data needs to be measured, the system is robust against noise.

References

- [1] R. Cipolla and A. Blake. Surface orientation and time to contact from image divergence and deformation. In G. Sandini, editor, *Computer Vision—ECCV '92*, pages 187–202. Springer-Verlag, 1992.
- [2] R. Sharma. Active vision in robot navigation: Monitoring time-to-collision while tracking. In *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2203–2208. IEEE, June 1992.
- [3] P. van der Smagt. Minimisation methods for training feed-forward networks. *Neural Networks*, 7(1):1–11, 1994.
- [4] D. Vernon and M. Tistarelli. Using camera motion to estimate range for robotic parts manipulation. *IEEE Transactions on Robotics and Automation*, 7(5):509–521, October 1990.