A robot arm is neurally controlled using monocular feedback

P. van der Smagt
DLR, German Aerospace Research Establishment
Department of Robotics and System Dynamics
P. O. Box 1116, D–82330 Weßling, GERMANY
email patrick.van.der.smagt@dlr.de

January 18, 1996

Living organisms, in contrast with static robot control systems, 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. While the information that is used in a static system only takes into account the *position* of points of interest, in a moving system such visual observations are interpreted in spatio-temporal domain. **Optic flow**, which is defined as the motion of the observer's environment projected on its image plane, is much more commonly used by living organisms than information obtained from static retinal patterns. The fact that optic flow is fundamental to vision has only been realised since the pioneering work of Gibson [1].

Figure 1: The gannet.

A classical example of the use of optic flow in biological systems is the gannet. When hunting for fish, this bird drops itself from the sky from a great height. Since the fish is moving, the bird needs its wings to correct its path while falling down; however, at the moment of contact with the seawater its wings must be folded to prevent them from breaking. It has been shown that the time remaining between the moment that the bird folds its wings, and that it hits the water, is always the same for one particular bird. It is not controlled by its height or velocity separately, but the quotient of the two. This remaining time is known as the **time-to-contact** and indicated by the symbol τ. When the system is not accelerating, τ is given by the quotient of the distance from an approaching surface and the velocity towards it. Exactly the same information can be obtained from the divergence of the approaching surface [2, 3]—a feature that can be observed monocularly.

Since this bird cannot measure its velocity, it measures the time until it hits the water from time derivatives of the visual observation. It is this mechanism that underlies the method presented in this paper for controlling a monocular robot arm such that it 'lands' on an object. An ordinary grasping task can be described as: at some time, the distance between the object and the hand-held camera must be zero. We go one step beyond that requirement: the distance must be zero at *some time* (which, we now know, is related to the time-to-contact), while the system must be at rest: the velocity, acceleration, and higher derivatives must also be zero. We will call this the **goal state**. But this can be seen as the endpoint of a **trajectory** towards that point in time. In the case of the bird, the decision to fold its wings can be made with the available visually measured quantities. In the sequel it will be shown that by extending the above example to higher-order time derivatives of the visual data, criteria can be developed which specify a trajectory which

ends in a rest state (i.e., zero velocity, acceleration, etc.) at the end point. These criteria will lead to **visual setpoints** along the followed trajectory, and are used as inputs to a neural controller which must generate robot joint accelerations in order to follow the setpoints in visual domain. Thus it is possible that the eye-in-hand robot arm exactly stops on an observed object by use of optic flow. By using time derivatives of the visual field we obtain to an important advantage: the model of the object is not needed to obtain depth information. The system need not be calibrated for each object that has to be grasped, but can approach objects with unknown dimensions.

In this paper we generalise previous time-to-contact based robot arm guidance methods to generation of 3D motion trajectories from optic flow, and use this to construct a model-free self-learning robot arm controller (in fact, not the optic flow field will be used but rather the visual position, velocity, acceleration, etc. of a single white object to be grasped against a black background). Of importance is the fact that no model of the robot arm or the observed world is necessary to obtain depth information with monocular vision.

1 Theory

We describe the motion of the camera which is placed in the robot's manipulator by $\mathbf{d}(t) = \{d_x(t), d_y(t), d_z(t)\}$. The $d_x(t)$, $d_y(t)$, and $d_z(t)$ indicate the Cartesian position of the object with respect to the camera. The task is to bring the system to the state where $\mathbf{d}(t) = \mathbf{0}$. If we write the components of $\mathbf{d}(t)$ by their Taylor approximation

$$d(t) = a_0 + a_1 t + \ldots + a_n t^n \tag{1}$$

and ignore the rest terms, then stopping at the goal state means that we require, at some given time τ_d (the desired time-to-contact),

$$\forall k, 0 \leq k < n: \quad d^{(k)}(\tau_d) = 0. \tag{2}$$

It can be proven [4] that this situation is reached when

$$\frac{a_0[i]}{a_1[i]} = -\frac{(\tau_d - t[i])}{n}, \qquad \forall i: 0 \leq i < \nu, \tag{3}$$

where $\nu \geq n$ and $(\tau_d - t[\nu]) \geq 0$.

2 Visual measurement of the stopping criteria

The deceleration algorithms described above is expressed in parameters a_k. We know that the $d(t)$ and thus the a_k cannot be measured with one camera when there is no model at hand of the observed object(s).

Instead of looking at $d_x(t)$, $d_y(t)$, and $d_z(t)$, we have to consider the quantities that are actually measured: $\xi_x(t)$, $\xi_y(t)$, and $\xi_z(t)$. These indicate the *observed* position of the object with respect of the centre of the camera image:

$$\xi_x(t) = \frac{d_x(t)}{\sqrt{A}}, \qquad \xi_y(t) = \frac{d_y(t)}{\sqrt{A}}, \qquad \xi_z(t) = \frac{d_z(t)}{f\sqrt{A}}$$

where f is the focal distance of the lens, and A is the 'real' area of the object. Thus the ξ's can all be measured by the camera, and they can just as well be fitted by polynomials

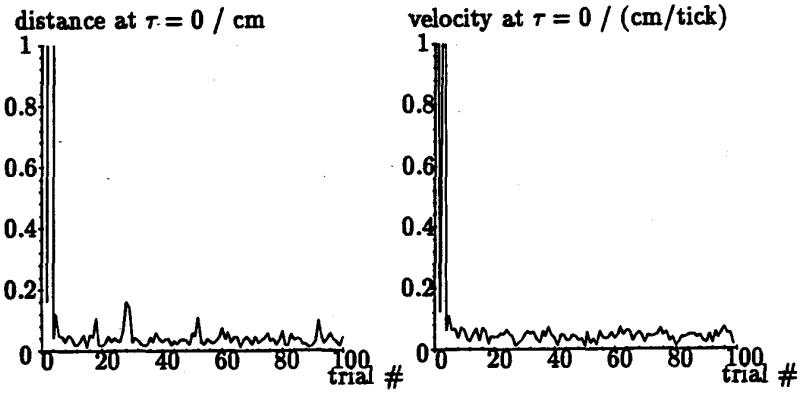

Figure 2: Distance and velocity at $\tau = 0$.

as their Cartesian counterparts; we find that

$$\xi(t) = \sum_{i=0}^{n} v_i t^i + o(t^n). \tag{4}$$

Now, the deceleration constraints can still be expressed in visual parameters, and we know that $v_i = ca_i$ where c is c_x, c_y, or c_z for the x, y, and z components of $\mathbf{d}$, given by

$$c_x = A^{-1/2}, \qquad c_y = A^{-1/2}, \qquad c_z = (f^2 A)^{-1/2}. \tag{5}$$

This means that the time-dependent constraint from can be written as

$$\frac{v_0[i]}{v_1[i]} = -\frac{(\tau_d - t[i])}{n}. \tag{6}$$

3 Results

A neural network takes as input the measured v_0 and v_1 as well as the current robot position, velocity, and acceleration (joint values). The output of the network consist of joint accelerations. Learning samples are generated on-line, and the training of the network takes place during operation of the robot arm. Details of the implementation can be found in [4]. The results show that after only a few trials (in this case, 4), the positional error at $(\tau_d - t[0]) = 0$ is below one millimeter, while the velocity is below 0.1cm per simulator time unit (typical end-effector velocities during deceleration are 0.5–2.0 cm per simulator time unit).

References

[1] J. J. Gibson. *The perception of the visual world.* Houghton Mifflin, 1950.

[2] J. J. Koenderink and A. J. van Doorn. Local structure of the motion parallax. *Optica Acta*, 22(9), 1975.

[3] D. N. Lee. The optic flow field: the foundation of vision. *Phil. Trans. R. Soc. Lond. B*, 290:169–179, 1980.

[4] Patrick van der Smagt. *Visual Robot Arm Guidance using Neural Networks.* PhD thesis, Dept of Computer Systems, University of Amsterdam, March 1995.