

# Vision-Based Online Trajectory Generation and Its Application to Catching

Akio Namiki and Masatoshi Ishikawa

University of Tokyo, Tokyo, Japan, {namiki,ishikawa}@k2.t.u-tokyo.ac.jp

## Abstract

*In this paper a method for sensor-based online trajectory generation is proposed. This method is based on a nonlinear mapping from sensor information to a desired trajectory, and a nonlinear mapping is decided by online learning based on constraints of dynamics and kinematics. This method is applied to a catching task, and responsive and flexible motion is realized based on realtime high-speed sensory information. Experimental results about catching of a moving object is shown.*

## 1 Introduction

Human processing architecture for motor control is regarded as a hierarchical processing system composed of several layers such as reflex, control, trajectory generation, and so on [1]. Each layer has realtime afferent signal inputs from the sensor system, and not only in a low-level layer such as reflex but also in a high-level layer such as trajectory generation, sensor information strongly effects its processing result.

This means that high-level processing is also processed dynamically by sensor information. In human reaching motion, for example, it is observed that visual information is used for feedforward trajectory generation in addition to feedback control based on a position error between an end effector and a target [2]. As a result a human can catch the target well even if it moves during reaching motion. Such a realtime sensor-based processing in a high-level layer realize flexible and responsive action to the changes of an environment.

On the other hand, in most conventional manipulation researches, realtime sensor information is used mainly for servo control, and it is not so used for high-level processing such as trajectory generation. Even if sensor information effects trajectory generation, it is static or quasi-static, and a processing rate is not high. In most cases a trajectory is given as a time-based function which is not effected by sensor

information. But a trajectory should be generated based on sensory information in realtime to realize responsive and flexible motion such as human.

In this paper a method for sensor-based online trajectory generation is proposed. This method is based on a nonlinear mapping from sensor information to desired trajectory, and a nonlinear mapping is decided by online learning based on constraints of dynamics and kinematics. This method is applied to a grasping task. Experimental results are shown.

## 2 Related Works

A number of research efforts have focused on the problem of using vision information in the execution of manipulation tasks, several of which addressed problems that are related to trajectory generation for robotic grasping.

In several researches visual information is used for prediction of target trajectory[3, 4, 5]. Because an image processing requires a lot of processing power, a delay of processing occurs. Prediction is useful to compensate for such a delay. A human being also solves this problem using high level prediction because a throughput of human visual processing is low. But there are several problems that it takes much time to learn an optimal predictor, and he can not act in an non-predictable condition.

On the other hand, in several researches a trajectory is directly generated using visual information. Xi et al. proposed a method based on the status equation expression[6]. In this method the time variable is replaced by a scaler which expresses sensor information. This approach is useful because conventional control approaches can be applied, but it is difficult to express multilateral sensor information. Buehler et al. proposed a method in which the end-effector position is mapped to the sensory variable one by one, and realized juggling [7]. This approach has an advantage that it is easy to express constraints of position and velocity which is very important to execute dynamic manipulation. Several researches adopt a similar approaches [8, 9]. But it is purely based on

the geometric relation, dynamics of the manipulator is not considered. As a result it is not enough to control a multi-axis manipulator which has heavy dynamics.

Such a direct online generator was difficult to be applied to realtime manipulation tasks because visual processing rate was too slow. But recently several types of a high-speed vision system have been developed. Our group has been developed massively-parallel digital vision chip systems [10, 11, 12]. In our vision chip architecture the photo detectors and the parallel processing elements are integrated in a single system without the I/O bottleneck. As a result a sampling rate more than 1[KHz] is realized. This technique was applied to a sensory-motor fusion system [13]. Based on this system high-speed grasping and manipulation was realized [14, 15, 16].

This shows that the speed of robotic visual recognition will become higher than a human soon. In such a situation the direct online generator become more important than the predictor, and a robotic system become able to act in an non-predictable condition.

### 3 Vision-based trajectory generator

In this section we propose a trajectory generator for catching of a moving object. Our generator has the similar expression as [7, 8], but it is updated on the constraints of dynamics and kinematics.

#### 3.1 Problem description

Suppose that the desired trajectory is generated based on a nonlinear mapping as

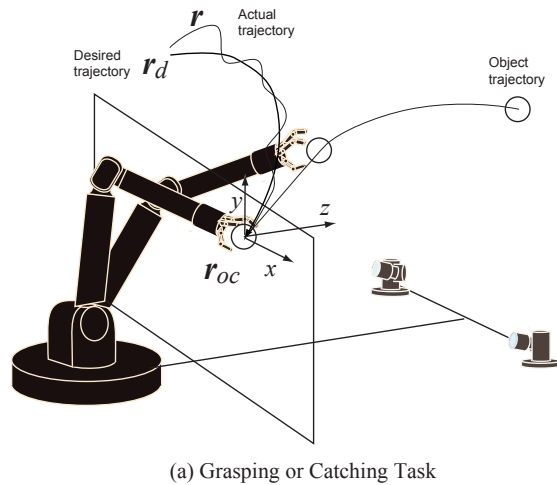
$$\mathbf{r}_d = \mathbf{f}(\mathbf{r}_o; \mathbf{a}), \quad (1)$$

where  $\mathbf{r}_d \in \mathbb{R}^3$  is the desired trajectory of the effector, and  $\mathbf{r}_o \in \mathbb{R}^3$  is the position of the target,  $\mathbf{a}$  is a parameter vector. And suppose that the position of the effector  $\mathbf{r} \in \mathbb{R}^3$  is controlled so as to track the desired position  $\mathbf{r}_d$ . Fig.1 (a) shows the problem description, in which the desired trajectory is generated so as to approach the object trajectory, and the effector is controlled to track the desired trajectory. Fig.1(b) shows the block diagram.

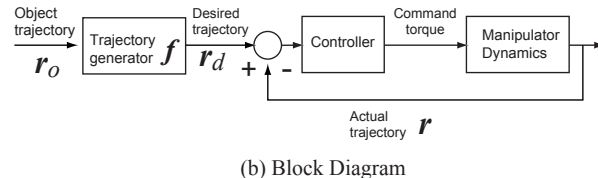
The problem is to decide the parameter  $\mathbf{a}$ . Normally the manipulator has several constraints which is related with dynamics and kinematics. For this reason the parameter  $\mathbf{a}$  should be optimized on the constraints.

#### 3.2 Geometric Constraints

Suppose a surface on which the target is caught just as the target is traversing it [8]. To simplify we set the coordinate frame on the surface so that the z-



(a) Grasping or Catching Task



(b) Block Diagram

Figure 1: Problem description

axis of the frame points in the direction of the surface normal. The positional error between the desired trajectory and the target trajectory is defined as  $\mathbf{e}_o = \mathbf{r}_d - \mathbf{r}_o$ . And the control error between the desired trajectory and the actual trajectory is defined as  $\mathbf{e}_r = \mathbf{r}_d - \mathbf{r}$ .

The following conditions should be satisfied.

1. Just as the target is grasped, a match between the position, velocity, and acceleration of the target and the desired trajectory should be satisfied as

$$\mathbf{e}_o(t_c) = \dot{\mathbf{e}}_o(t_c) = \ddot{\mathbf{e}}_o(t_c) = \mathbf{0}, \quad (2)$$

where  $t_c$  represents the grasped time.

2. At the beginning a match between the position, velocity, and acceleration of the the desired trajectory and the actual effector should be satisfied as

$$\mathbf{e}_r(t_0) = \dot{\mathbf{e}}_r(t_0) = \ddot{\mathbf{e}}_r(t_0) = \mathbf{0}, \quad (3)$$

where  $t_0$  represents the initial time.

#### 3.3 Dynamics constraints

Generally the dynamics of a manipulator in terms of the effector position  $\mathbf{r}$  is described as

$$M_1(\mathbf{q}) \ddot{\mathbf{r}} + H_1(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{r}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}, \quad (4)$$

where  $\mathbf{r} \in \mathbb{R}^3$  is the position of the end effector,  $\mathbf{q} \in \mathbb{R}^{m_q}$  is the vector of joint angles ( $m_q$ : the number of joints),  $\boldsymbol{\tau} \in \mathbb{R}^{m_q}$  is the vector of joint torques,  $\mathbf{g} \in \mathbb{R}^{m_q}$  is the gravity term. And  $M_1, H_1$  is described as

$$M_1(\mathbf{q}) \triangleq M(\mathbf{q}) J(\mathbf{q})^{-1}, \quad (5)$$

$$H_1(\mathbf{q}, \dot{\mathbf{q}}) \triangleq \left( H(\mathbf{q}, \dot{\mathbf{q}}) - M(\mathbf{q}) J(\mathbf{q})^{-1} \dot{J}(\mathbf{q}) \right) J(\mathbf{q})^{-1} \quad (6)$$

where  $M \in \mathbb{R}^{m_q \times m_q}$  is the inertia matrix, and  $H(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{m_q}$  gives the coriolis and centrifugal force terms, and  $J \triangleq \mathbb{R}^{3 \times m_q}$  is the Jacobian matrix.

The computed torque control law of Eqn.(4) becomes

$$\boldsymbol{\tau}_d = M_1(\mathbf{q}) \ddot{\mathbf{r}}_d + H_1(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{r}} + \mathbf{g}(\mathbf{q}). \quad (7)$$

On the other hand the 2nd order differentiation of Eqn.(1) becomes

$$\ddot{\mathbf{r}}_d = F(\mathbf{r}_o; \mathbf{a}) \ddot{\mathbf{r}}_o + \dot{F}(\mathbf{r}_o; \mathbf{a}) \dot{\mathbf{r}}_o, \quad (8)$$

where  $F \triangleq \frac{\partial \mathbf{f}}{\partial \mathbf{r}_o}$ . Substituting Eqn.(8) into Eqn.(7),

$$\begin{aligned} \boldsymbol{\tau}_d = & M_1(\mathbf{q}) F(\mathbf{r}_o; \mathbf{a}) \ddot{\mathbf{r}}_o + M_1(\mathbf{q}) \dot{F}(\mathbf{r}_o; \mathbf{a}) \dot{\mathbf{r}}_o \\ & + H_1(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{r}} + \mathbf{g}(\mathbf{q}). \end{aligned} \quad (9)$$

Supposing  $\mathbf{r}_d = \mathbf{r}$ ,  $\mathbf{q}$  is computed as

$$\mathbf{q} \simeq \mathbf{l}^{-1}(\mathbf{r}_d) = \mathbf{l}^{-1} \circ \mathbf{f}(\mathbf{r}_o; \mathbf{a}), \quad (10)$$

where a function  $\mathbf{l}$  represents the direct kinematics. As a result Eqn. (9) is rewritten as

$$\begin{aligned} \tilde{\boldsymbol{\tau}}_d(\mathbf{r}_o, \dot{\mathbf{r}}_o, \ddot{\mathbf{r}}_o; \mathbf{a}) & \\ = & \tilde{M}_1(\mathbf{r}_o; \mathbf{a}) F(\mathbf{r}_o; \mathbf{a}) \ddot{\mathbf{r}}_o \\ & + \left( \tilde{M}_1(\mathbf{r}_o; \mathbf{a}) \dot{F}(\mathbf{r}_o; \mathbf{a}) + \tilde{H}_1(\mathbf{r}_o, \dot{\mathbf{r}}_o; \mathbf{a}) F(\mathbf{r}_o; \mathbf{a}) \right) \dot{\mathbf{r}}_o \\ & + \tilde{\mathbf{g}}(\mathbf{r}_o; \mathbf{a}), \end{aligned} \quad (11)$$

where

$$\tilde{M}_1(\mathbf{r}_o; \mathbf{a}) \triangleq M_1(\mathbf{l}^{-1} \circ \mathbf{f}(\mathbf{r}_o; \mathbf{a})), \quad (12)$$

$$\tilde{H}_1(\mathbf{r}_o, \dot{\mathbf{r}}_o; \mathbf{a}) \triangleq H_1(\mathbf{l}^{-1} \circ \mathbf{f}(\mathbf{r}_o; \mathbf{a}), F(\mathbf{r}_o; \mathbf{a}) \dot{\mathbf{r}}_o), \quad (13)$$

$$\tilde{\mathbf{g}}(\mathbf{r}_o; \mathbf{a}) \triangleq \mathbf{g}(\mathbf{l}^{-1} \circ \mathbf{f}(\mathbf{r}_o; \mathbf{a})). \quad (14)$$

As a result, the following constraints should be satisfied:

$$\boldsymbol{\tau}_{\min} \leq \tilde{\boldsymbol{\tau}}_d(\mathbf{r}_o, \dot{\mathbf{r}}_o, \ddot{\mathbf{r}}_o; \mathbf{a}) \leq \boldsymbol{\tau}_{\max}, \quad (15)$$

where  $\boldsymbol{\tau}_{\min}$  is the minimum torque, and  $\boldsymbol{\tau}_{\max}$  is the maximum torque.

### 3.4 Kinematics constraints

Supposing  $\mathbf{r}_d = \mathbf{r}$ , the joint velocity  $\boldsymbol{\omega} \in \mathbb{R}^{m_q}$  is computed as,

$$\begin{aligned} \boldsymbol{\omega} &= J(\mathbf{q})^{-1} \dot{\mathbf{r}} \\ &= J(\mathbf{l}^{-1} \circ \mathbf{f}(\mathbf{r}_o; \mathbf{a}))^{-1} F(\mathbf{r}_o; \mathbf{a}) \dot{\mathbf{r}}_o, \end{aligned} \quad (16)$$

As a result the following constraint should be satisfied:

$$\boldsymbol{\omega}_{\min} \leq \boldsymbol{\omega}(\mathbf{r}_o, \dot{\mathbf{r}}_o; \mathbf{a}) \leq \boldsymbol{\omega}_{\max}, \quad (17)$$

where  $\boldsymbol{\omega}_{\min}$  is the minimum angular velocity, and  $\boldsymbol{\omega}_{\max}$  is the maximum angular velocity.

### 3.5 Approximation of trajectory generator

The function  $\mathbf{f}(\mathbf{r}_o; \mathbf{a})$  is a nonlinear function, and it is necessary to approximate it to analytical functions. In this paper we adopt the polynomial expression because geometric constraints are easily expressed with it.

Using Taler expansion the trajectory generator is approximated as

$$\mathbf{f}(\mathbf{r}_o) \simeq \mathbf{f}(\mathbf{0}) + \sum_{n=1}^N \frac{1}{n!} (\mathbf{r}_o \cdot \nabla)^n \mathbf{f}(\mathbf{0}), \quad (18)$$

where  $\nabla = [\frac{\partial}{\partial r_{ox}}, \frac{\partial}{\partial r_{oy}}, \frac{\partial}{\partial r_{oz}}]^T$ .

To satisfy Eqn.(2) and Eqn.(3), it is rewritten as,

$$\mathbf{f}(\mathbf{r}_o) \simeq \mathbf{r}_o + \sum_{\substack{0 \leq i, 0 \leq j, 3 \leq k \\ i+j+k \leq N}}^N \mathbf{a}_{ijk} r_{ox}^i r_{oy}^j r_{oz}^k, \quad (19)$$

where  $\mathbf{a}_{ijk} \triangleq \frac{\partial^{i+j+k} \mathbf{f}}{\partial r_{ox}^i \partial r_{oy}^j \partial r_{oz}^k}$ .

### 3.6 Trajectory generation algorithm

Based on Eqn.(15) and Eqn.(17) the optimal trajectory generator is decided as the solution of

$$\min_{\mathbf{a}} E(\mathbf{a}) = \sum_{\mathbf{r}_o, \dot{\mathbf{r}}_o, \ddot{\mathbf{r}}_o} (\tilde{\boldsymbol{\tau}}_d^T K_{\tau} \tilde{\boldsymbol{\tau}}_d + \boldsymbol{\omega}_d^T K_{\omega} \boldsymbol{\omega}_d) \quad (20)$$

Subject to

$$\boldsymbol{\tau}_{\min} \leq \tilde{\boldsymbol{\tau}}_d(\mathbf{r}_o, \dot{\mathbf{r}}_o, \ddot{\mathbf{r}}_o; \mathbf{a}) \leq \boldsymbol{\tau}_{\max}$$

$$\boldsymbol{\omega}_{\min} \leq \boldsymbol{\omega}_d(\mathbf{r}_o, \dot{\mathbf{r}}_o; \mathbf{a}) \leq \boldsymbol{\omega}_{\max}$$

where  $K_{\tau}$  and  $K_{\omega}$  are a positive diagonal matrix.

To compute Eqn.(20) various target trajectories are needed. But it is difficult to input every possible target trajectory, if there are large numbers of possible target trajectories. And if impossible trajectories are inputted, the system performance go down. To solve

these problems it is necessary to optimize parameters using an online method.

The total control diagram is described in Fig.2. The parameters of the trajectory generator is changed depending on the dynamics and the kinematics of the manipulator and the object. The adaptation can be executed in both online and offline, and it should be executed in online in dynamic changing environments.

If adaptation of the trajectory generator is not enough, the error is mainly compensated in the controller. As the trajectory generator approaches the optimal configuration, the load of the controller reduces.

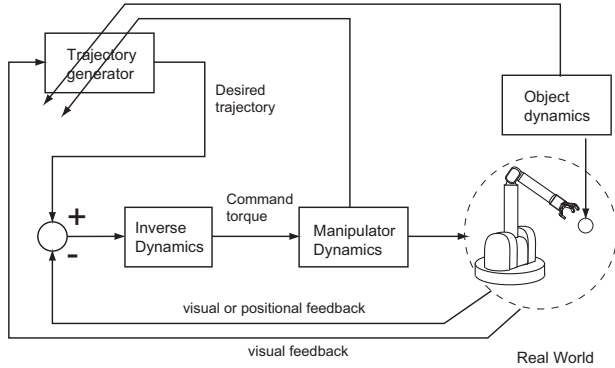


Figure 2: Control diagram

## 4 Experimental system

In Fig.3 the experimental system is shown.

The processing part consists of many floating-point DSP module(TMS320C6701), and I/O ports are distributed on several DSPs so that sensor signals are input in parallel.

The vision is a massively parallel vision system called CPV (column-parallel high speed vision system) [12]. The CPV has  $128 \times 128$  photo detectors and all pixel parallel processing array based on a vision chip architecture and exclusive summation circuit for calculating moment values. Because the visual processing is executed in parallel in the processing array, high-speed visual processing (moment detection, segmentation, and etc) is realized within 1[ms].

The arm is a 4-axis manipulator (Barrett Technology Inc.). The maximum speed of an end-effector is 6m/s, and the maximum acceleration of an end-effector is  $58\text{m/s}^2$ . Because it is said that a velocity of a human arm is about 6 – 10[m/s], a motion of the arm is as high-speed and responsive as a human motion.

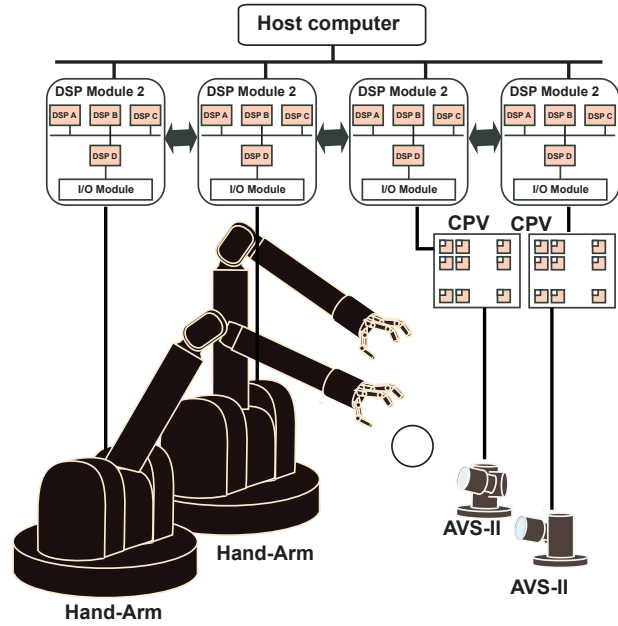


Figure 3: Experimental system

The hand is a 3-fingered and 3-axis dextrous hand. By reducing a weight of each finger and using high power actuator, as a result fingers can close a 90-degree angle in 20[ms]. Because it is said that a human hand has an ability to tap a table at about 10[Hz], our hand is as high-speed as a human.

## 5 Simulation Result

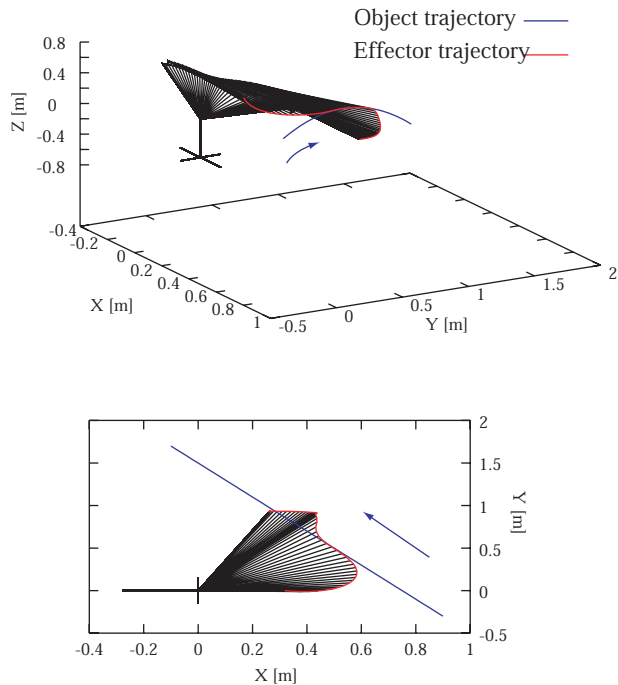
We used the 5-th order polynomial expression in Eqn.(19). We gave all possible target trajectories in the neighborhood of the catching point, and using these data parameters of the polynomial expression were optimized.

Fig.4 shows the arm trajectory based on the proposed method, and Fig.5 shows the arm trajectory based on a simple PD feedback method.

In the proposed method the trajectory was generated so that the end effector can catch the target at one point, and a match between the position, velocity, and acceleration of the target and the end effector was satisfied. In the feedback method the effector kept touching the target in some period, as a result there was loss of power of the manipulator.

Fig.6 shows the time response of the command torque during catching. In the feedback method the big command torque was needed just after starting

the motion. In the proposed method small command torque was used except the moment of catching.



**Figure 4:** Catching using vision-based trajectory generator

## 6 Experimental result

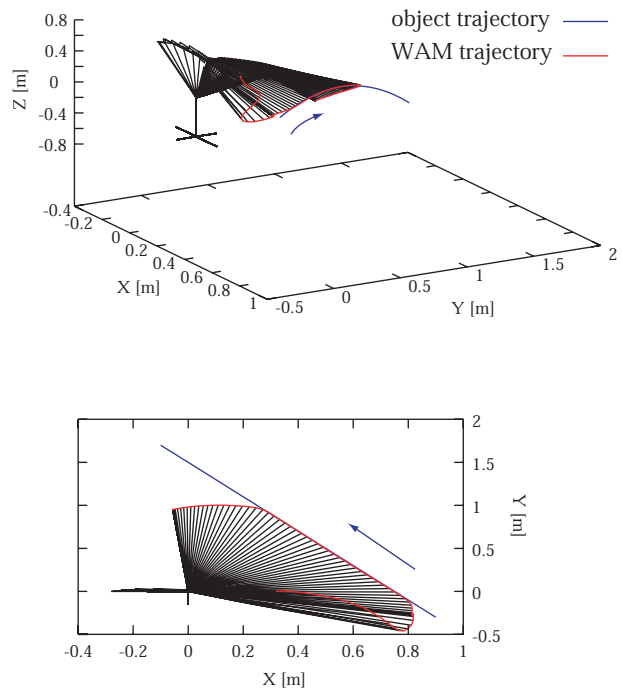
Using the proposed method we realized the robotic catching of a moving object. In Fig.7 a catching task of a sphere is shown as a continuous sequence of pictures every 0.1[s].

## 7 Conclusion

In this paper we have proposed a realtime online trajectory generation algorithm using high-speed vision. Now several applications of dynamic manipulation have been developed based on the proposed algorithm.

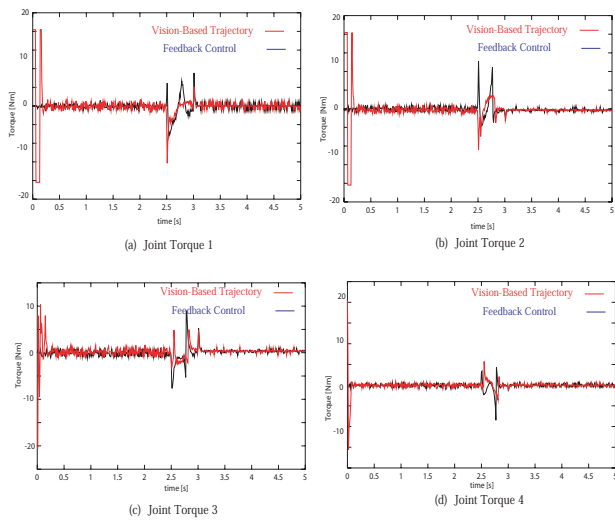
## References

- [1] J.S. Albus, "Outline for a theory of intelligence," *IEEE Trans. on Syst., Man, and Cybern.*, vol. 21, no. 3, pp. 473–509, 1991.
- [2] C. Bard, Y. Turrell, M. Fleury, N. Teasdale, Y. Lamarre, and O. Martin, "Deafferentation and pointing with visual double-step perturbations," *Exp. Brain Res.*, vol. 125, pp. 410–416, 1999.
- [3] A.J. Kovic and N. Houshangi, "Real-time vision feedback for servoing robotic manipulator with self-tuning controller," *IEEE Trans. of Syst., Man, and Cybern.*, vol. 21, no. 1, pp. 134–142, 1992.



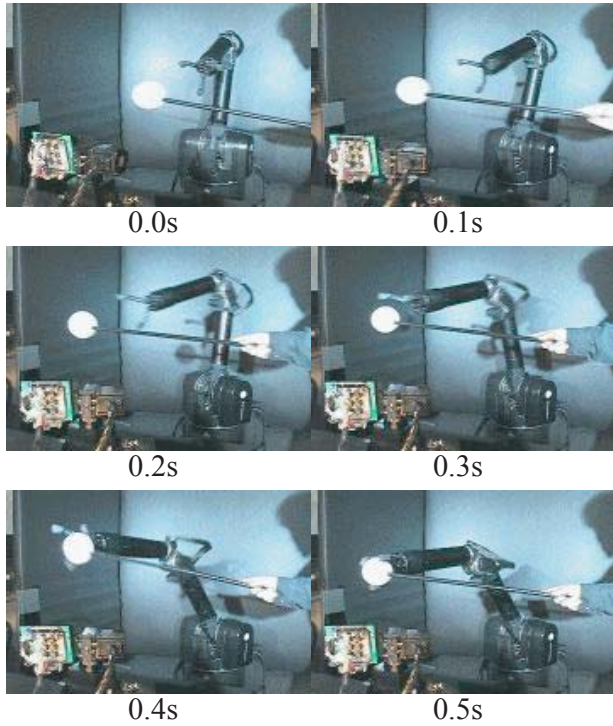
**Figure 5:** Catching using simple PD feedback control

- [4] P.K. Allen, B. Yoshimi, and A. Timucenko, "Real-time visual servoing," *Proc. IEEE Int. Conf. Robot. and Automat.*, pp. 2376–2384, 1991.
- [5] W. Hong and J.E. Slotine, "Experiments in hand-eye coordination using active vision," *Proc. 4th Int. Symp. on Experimental Robot.*, 1995.
- [6] B.Ghosh, N.Xi, and T.J.Tarn, *CONTROL IN ROBOTICS AND AUTOMATION: Sensor-Based Integration*, ACADEMIC, 1999.
- [7] M. Buehler, D.E. Koditshek, and P.J. Kindlmann, "Planning and control of robotic juggling and catching tasks," *Int. J. of Robot. Res.*, vol. 13, no. 2, pp. 101–118, 1994.
- [8] M. Zhang and M. Buehler, "Sensor-based online trajectory generation for smoothly grasping moving objects," *Proc. IEEE Int. Symp. Intelligent Control*, pp. 141–146, 1994.
- [9] T. Sakaguchi, M. Fujita, H. Watanabe, and F. Miyazaki, "Motion planning and control for a robot performer," *Proc. IEEE Int. Conf. Robot. and Automat.*, vol. 3, pp. 925–931, 1993.
- [10] M. Ishikawa, A. Morita, and N. Takayanagi, "High speed vision system using massively parallel processing," *Proc. IEEE Int. Conf. Intelligent Robot. and Systems*, pp. 373–377, 1992.
- [11] T. Komuro, I. Ishii, and M. Ishikawa, "Vision chip architecture using general-purpose processing elements for lms vision system," *Proc. IEEE Int. Workshop on Computer Architecture for Machine Perception*, pp. 276–279, 1997.



**Figure 6:** Command torque

- [12] Y. Nakabo, M. Ishikawa, H. Toyoda, and S. Mizuno, "1ms column parallel vision system and its application of high speed target tracking," *Proc. IEEE Int. Conf. Robot. and Automat.*, pp. 650–655, 2000.
- [13] A. Namiki, Y. Nakabo, I. Ishii, and M. Ishikawa, "1ms sensory-motor fusion system," *IEEE/ASME Trans. Mechatron.*, vol. 5, no. 3, pp. 244–252, 2000.
- [14] A. Namiki, Y. Nakabo, I. Ishii, and M. Ishikawa, "High speed grasping using visual and force feedback," *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 3195–3200, 1999.
- [15] A. Namiki, Y. Nakabo, I. Ishii, and M. Ishikawa, "1ms grasping system using visual and force feedback," *Video Proc. IEEE Int. Conf. Robot. Automat.*, 1999.
- [16] A. Namiki and M. Ishikawa, "Sensory-motor fusion architecture based on high-speed sensory feedback and its application to grasping and manipulation," *Proc. Int. Symp. Robotics*, pp. 784–789, 2001.



**Figure 7:** Trajectory of the end effector