

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

Akio Namiki<sup>1</sup> and Masatoshi Ishikawa<sup>2</sup>

University of Tokyo, 7-3-1 Bunkyo-ku, Hongo, Tokyo 113-8656, Japan

**Abstract.** In this paper, a method for vision-based online trajectory generation is proposed. The proposed method is based on a nonlinear mapping of visual information to the desired trajectory, and this nonlinear mapping is defined by learning based on constraints of dynamics and kinematics. This method is applied to a catching task, and a reactive and flexible motion is obtained owing to real-time high-speed visual information. Experimental results on catching a moving object using a high-speed vision chip system are presented.

## 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 vision system, and visual information affects the processing result strongly not only in low-level layers such as that of reflex but also in high level layers such as that corresponding to trajectory generation. This means that high-level processing is also processed dynamically by visual information.

In human reaching motion, for example, it is observed that visual information is used for feed-forward trajectory generation in addition to feedback control based on a position error between an end effector and target [3]. As a result, a human can catch a target well, even if it moves during the reaching motion. Such a realtime vision-based processing in a high-level layer produces a flexible action, reactive to changes of the environment.

On the other hand, in most conventional manipulation researches, realtime visual information is used mainly for servo control, and it is not directly used as much for high-level processing such as trajectory generation. Even if visual information affects trajectory generation, it is static or quasi-static, and the processing rate is not high. In most cases a trajectory is given as a time-based function which is not affected directly by visual information. However a vision-based online trajectory generator has more advantages to produce a reactive and flexible motion in the case that it is difficult to predict target motion because of disturbance and uncertainty.

In this paper, a method for vision-based online trajectory generation is proposed and applied to a catching task of a ball. This method is based on a nonlinear mapping of visual information to the desired trajectory, and this nonlinear mapping is defined by learning based on constraints of dynamics

and kinematics. In section 2 previous related works on manipulation using realtime visual information are presented. In section 3 we propose an algorithm for online vision-based trajectory generator, and in section 4 an experimental result using a high-speed vision system and a manipulator is presented.

## 2 Related Works

A number of research efforts have focused on the problem of using visual 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 predicting a target trajectory [9,2,6]. Image processing requires a lot of processing power, which produces a processing delay. Human beings solve this problem using high level prediction because the throughput of human visual processing is low. But there are several problems requiring a long learning time to produce an optimal predictor, and they can not act in 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 expression of the status equation [4]. In this method the time variable is replaced by a scalar that expresses sensor information. This approach is useful because conventional control approaches can be applied, but multilateral sensor information is difficult to express. Büehler et al. proposed a method in which the end-effector position is mapped to the sensory variables one by one, and with which they realized a juggling motion [5]. This approach has advantage that it is easy to express constraints of position and velocity, which is very important to execute dynamic manipulation. Several researches adopt similar approaches [16,15]. But they are purely based on geometric considerations and the dynamics of the manipulator is not considered. As a result they are not powerful enough to control a multi-axis manipulator which has heavy dynamics.

Such a direct online generator could not be applied easily to realtime manipulation tasks because the visual processing rate was too slow. But recently several types of high-speed vision systems have been developed. Our group has developed massively- parallel digital vision chip systems [7,8,10]. In our vision chip architecture the photo detectors and the parallel processing elements are integrated in a single system without the I/O bottleneck which results in a sampling rate higher than 1 KHz. This technique was applied to a sensory-motor fusion system [14] used to produce high- speed grasping and manipulation [13,12,11].

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

### 3 Vision-Based Online Trajectory Generator

In this section we propose a trajectory generator for catching a flying ball one-handed. This generator is based on a nonlinear mapping of visual information to the desired trajectory, and this nonlinear mapping is defined by learning based on constraints of geometry, dynamics and kinematics.

#### 3.1 Problem Description

Suppose that the system can recognize the 3-dimensional position  $\mathbf{r}_o \in \mathbb{R}^3$  of a ball by vision, and a desired trajectory of joints angle  $\mathbf{q}_d \in \mathbb{R}^{m_q}$  ( $m_q$ : the number of joints) is generated from a nonlinear mapping of it as

$$\mathbf{q}_d = \mathbf{f}(\mathbf{r}_o). \quad (1)$$

And suppose that the joint angles  $\mathbf{q} \in \mathbb{R}^{m_q}$  is controlled so as to track the desired position  $\mathbf{q}_d$  using an appropriate controller. Fig.1 shows the block diagram.

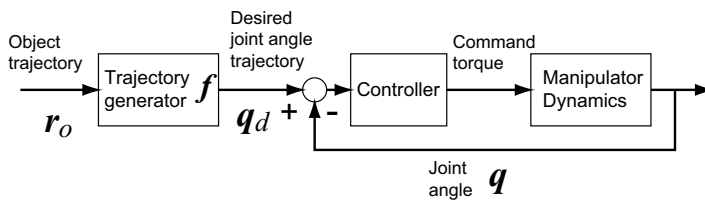
The desired trajectory  $\mathbf{q}_d$  should be planned so that the trajectory of the end effector  $\mathbf{r} \in \mathbb{R}^3$  intersects the trajectory of the target  $\mathbf{r}_o$  at a certain catching point. Fig.2 shows a catching task, in which the desired trajectory of an end effector  $\mathbf{r}_d \in \mathbb{R}^3$  is generated so as to approach the object trajectory  $\mathbf{r}_o$ , and  $\mathbf{r}$  is controlled to track  $\mathbf{r}_d$ . The trajectory and desired trajectory of the end effector are computed as

$$\mathbf{r}_d = \mathbf{l}(\mathbf{q}_d), \quad (2a)$$

$$\mathbf{r} = \mathbf{l}(\mathbf{q}), \quad (2b)$$

where the function  $\mathbf{l}$  represents direct kinematics.

The problem is to define a formula of the function  $\mathbf{f}$ . Normally a manipulator has several constraints which are related to geometry, dynamics and kinematics. For this reason the function  $\mathbf{f}$  should be optimized with respect to these constraints.



**Fig. 1.** The block diagram of a catching control. The desired trajectory of joint angles  $\mathbf{q}_d$  is directly generated from the position of the target  $\mathbf{r}_o$ .

### 3.2 Terminal Geometric Constraints

Just as the target is caught, a match between the position and velocity of the target and the desired trajectory should be satisfied by the condition:

$$\mathbf{r}_e(t_c) = \dot{\mathbf{r}}_e(t_c) = \mathbf{0}, \quad (3)$$

where  $t_c$  represents the time just as the target is caught, and

$$\mathbf{r}_e \triangleq \mathbf{l} \circ \mathbf{f}(\mathbf{r}_o) - \mathbf{r}_o. \quad (4)$$

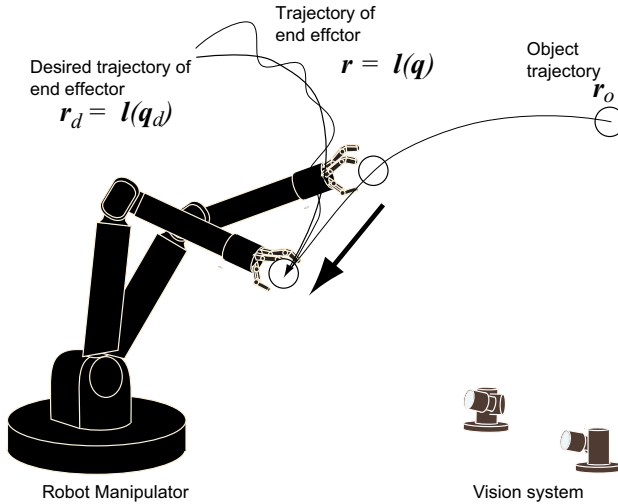
The vector  $\mathbf{r}_e$  represents the error between the desired trajectory of the end effector and the target trajectory.

One solution of Eqn.(3) is the inverse kinematics function  $\mathbf{f}(\mathbf{r}_o) = \mathbf{l}^{-1}(\mathbf{r}_o)$ . But it is inconvenient for a trajectory generator since the domain of  $\mathbf{l}^{-1}(\mathbf{r}_o)$  is limited within the area the end effector reaches,

In human catching motion, the 3-axis shoulder joints controls a rough position of the hand, and the elbow and wrist joints are mainly used for a match between the position and velocity of the target and the hand. It gives us a hint to set the formula of  $\mathbf{f}$ .

We select one appropriate elbow or wrist joint  $i$ , and suppose that the target is caught at the point  $q_i = q_\alpha$ , where  $q_i \in \mathbb{R}$  is the  $i$ -th joint angle, and  $q_\alpha \in \mathbb{R}$  is a constant value. We define a function  $\mathbf{l}_c$  as

$$\mathbf{r}_o = \mathbf{l}_c(\mathbf{q}) \triangleq \mathbf{l}(\mathbf{q})|_{q_i=q_\alpha} + \frac{\partial \mathbf{l}}{\partial q_i} \Big|_{q_i=q_\alpha} (q_i - q_\alpha). \quad (5)$$



**Fig. 2.** Catching: The desired trajectory of an end effector  $\mathbf{r}_d \in \mathbb{R}^3$  is generated so as to approach the object trajectory  $\mathbf{r}_o$ , and the position of the end effector  $\mathbf{r}$  is controlled to track the desired trajectory  $\mathbf{r}_d$ .

This is the first order Taylor expansion of the direct kinematics  $\mathbf{l}(\mathbf{q})$ . Fig.2 describes the mapping  $\mathbf{l}_c$ . The first term  $\mathbf{l}(\mathbf{q})|_{q_i=q_\alpha}$  represents the catching point, and  $\frac{\partial \mathbf{l}}{\partial q_i}|_{q_i=q_\alpha}$  represents a tangent vector of  $i$ -th joint motion at the catching point. In this function the distance between the target and the catching point is mapped to the  $i$ -th joint angle, and the target is caught at the point  $q_i = q_\alpha$ .

Using a function  $\mathbf{l}_c$  we define a formula of a trajectory generator which satisfies the terminal geometry constraints as

$$\mathbf{q}_d = \mathbf{f}(\mathbf{r}_o) \triangleq \mathbf{g} \circ \mathbf{l}_c^{-1}(\mathbf{r}_o), \quad (6)$$

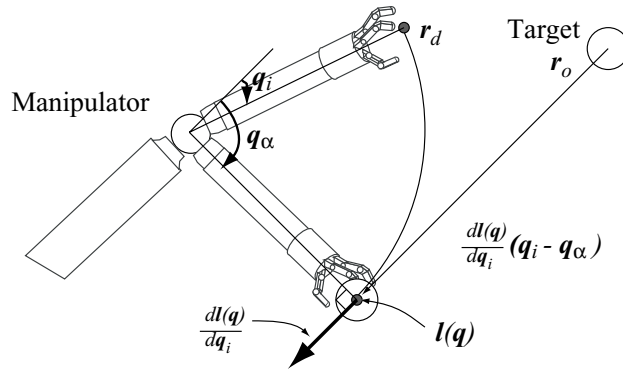
where

$$\mathbf{g}(\mathbf{q}) \triangleq \mathbf{q} + \mathbf{b}(\mathbf{q}) \quad (7)$$

and  $\mathbf{b}(\mathbf{q})$  is an arbitrary function of joint angles  $\mathbf{q}$ , provided that  $\dot{\mathbf{b}}|_{q_i=q_\alpha} = \mathbf{0}$ .

In Eqn.(6) the function  $\mathbf{l}_c^{-1}$  is regarded as a nonlinear mapping from the target position coordinates to the joint angle coordinates. On the other hand, the function  $\mathbf{g}$  is regarded as a nonlinear mapping from the joint angle coordinates to the joint angle coordinates, and it is expected that joint  $i$  is mainly used for a match between the target and the end effector.

It is clear that Eqn.(6) satisfies the terminal constraints described in Eqn.(3). But it is necessary to select the joint  $i$  and its catching angle  $q_\alpha$  appropriately, so that the inverse mapping  $\mathbf{l}_c^{-1}$  has a solution even if the target position  $\mathbf{r}_o$  takes any value.



**Fig. 3.** A mapping from a target position  $\mathbf{r}_o$  to a vector of joint angles  $\mathbf{q}$ . The distance between the target and the catching point is mapped to the  $i$ -th joint angle  $q_i$ .

### 3.3 Initial Geometric Constraints

At the beginning, a match between the position and velocity of the desired trajectory and of the actual effector should be satisfied by the condition:

$$\mathbf{q}_e(t_0) = \dot{\mathbf{q}}_e(t_0) = \mathbf{0}, \quad (8)$$

where  $t_0 \in \mathbb{R}$  represents the initial time, and

$$\mathbf{q}_e \triangleq \mathbf{f}(\mathbf{r}_o) - \mathbf{q}, \quad (9)$$

where  $\mathbf{q}_e \in \mathbb{R}^{m_q}$  represents the control error between the desired trajectory and the actual trajectory of the joint angles.

From Eqn.(7),  $\mathbf{q}_e$  is rewritten as

$$\mathbf{q}_e = \mathbf{q}_o - \mathbf{q} + \mathbf{b}(\mathbf{q}_o), \quad (10)$$

where  $\mathbf{q}_o = \mathbf{l}_c^{-1}(\mathbf{r}_o)$ . To simplify the problem, we approximate the function  $\mathbf{b}$  with analytical functions. In this paper we adopt a polynomial expression because initial geometric constraint are easily expressed in this way. Using  $N$ th-order Taylor expansion at the point  $q_i = q_\alpha$  and the terminal constraints, the function  $\mathbf{b}$  is approximated as follows:

$$\mathbf{b}(\mathbf{q}) \simeq \sum_{\substack{0 \leq j_1, 0 \leq j_2, \dots, 0 \leq j_{m_q} \\ j_1 + j_2 + \dots + j_{m_q} \leq N-2}}^N \alpha_{j_1 j_2 \dots j_{m_q}} q_1^{j_1} q_2^{j_2} \dots (q_i - q_\alpha)^{j_i+2} \dots q_{m_q}^{j_{m_q}}, \quad (11)$$

where  $\alpha_{j_1 j_2 \dots j_{m_q}} \in \mathbb{R}^{m_q}$  is a constant parameter.

Based on Eqn.(11), the initial geometric constraints is written as,

$$C \boldsymbol{\alpha} = \begin{bmatrix} \mathbf{q}(t_0) - \mathbf{q}_o(t_0) \\ \dot{\mathbf{q}}(t_0) - \dot{\mathbf{q}}_o(t_0) \end{bmatrix}, \quad (12)$$

where  $\boldsymbol{\alpha}$  is a vector which is defined by juxtaposing all  $\alpha_{j_1 j_2 \dots j_{m_q}}$ , and

$$C \triangleq \begin{bmatrix} \frac{\partial \mathbf{b}(t_0)}{\partial \boldsymbol{\alpha}} \\ \frac{\partial \dot{\mathbf{b}}(t_0)}{\partial \boldsymbol{\alpha}} \end{bmatrix} \quad (13)$$

is a constant matrix.

### 3.4 Dynamics constraints

Generally the dynamics of a manipulator in terms of the joint angle  $\mathbf{q}$  is described by

$$M(\mathbf{q}) \ddot{\mathbf{q}} + H(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}, \quad (14)$$

where  $\boldsymbol{\tau} \in \mathbb{R}^{m_q}$  is the vector of joint torques,  $M \in \mathbb{R}^{m_q \times m_q}$  is the inertia matrix,  $H(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} \in \mathbb{R}^{m_q}$  gives the coriolis and centrifugal force terms, and  $\mathbf{g} \in \mathbb{R}^{m_q}$  is the gravity term.

On the other hand, the 1st order differentiation of Eqn.(1) becomes

$$\dot{\mathbf{q}}_d = F(\mathbf{r}_o) \dot{\mathbf{r}}_o \quad (15)$$

where  $F \triangleq \frac{\partial \mathbf{f}}{\partial \mathbf{r}_o}$ .

Suppose  $\mathbf{q} \simeq \mathbf{q}_d = \mathbf{f}(\mathbf{r}_o)$ , Substituting Eqn.(15) into Eqn.(14),

$$\boldsymbol{\tau} = \tilde{M}(\mathbf{r}_o)F(\mathbf{r}_o)\ddot{\mathbf{r}}_o + \left( \tilde{M}(\mathbf{r}_o)\dot{F}(\mathbf{r}_o) + \tilde{H}(\mathbf{r}_o, \dot{\mathbf{r}}_o)F(\mathbf{r}_o) \right) \dot{\mathbf{r}}_o + \tilde{\mathbf{g}}(\mathbf{r}_o), \quad (16)$$

where

$$\tilde{M}(\mathbf{r}_o) \triangleq M(\mathbf{f}(\mathbf{r}_o)), \quad (17a)$$

$$\tilde{H}(\mathbf{r}_o, \dot{\mathbf{r}}_o) \triangleq H(\mathbf{f}(\mathbf{r}_o), F(\mathbf{r}_o) \dot{\mathbf{r}}_o), \quad (17b)$$

$$\tilde{\mathbf{g}}(\mathbf{r}_o) \triangleq \mathbf{g}(\mathbf{f}(\mathbf{r}_o)). \quad (17c)$$

As a result, the following constraints should be satisfied:

$$\boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau}(\mathbf{r}_o, \dot{\mathbf{r}}_o, \ddot{\mathbf{r}}_o) \leq \boldsymbol{\tau}_{\max}, \quad (18)$$

where  $\boldsymbol{\tau}_{\min} \in \mathbb{R}^{m_q}$  and  $\boldsymbol{\tau}_{\max} \in \mathbb{R}^{m_q}$  are the minimum and maximum torque respectively.

### 3.5 Kinematics constraints

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

$$\boldsymbol{\omega} = \dot{\mathbf{q}} = F(\mathbf{r}_o) \dot{\mathbf{r}}_o. \quad (19)$$

As a result the following constraint should be satisfied:

$$\boldsymbol{\omega}_{\min} \leq \boldsymbol{\omega}(\mathbf{r}_o, \dot{\mathbf{r}}_o) \leq \boldsymbol{\omega}_{\max}, \quad (20)$$

where  $\boldsymbol{\omega}_{\min} \in \mathbb{R}^{m_q}$  is the minimum angular velocity, and  $\boldsymbol{\omega}_{\max} \in \mathbb{R}^{m_q}$  is the maximum angular velocity.

### 3.6 Trajectory generation algorithm

From Eqn.(5)~(7),(11), a direct mapping from the target position to the desired trajectory is defined as

$$\mathbf{q}_d = \mathbf{f}(\mathbf{r}_o) \triangleq \mathbf{g} \circ \mathbf{l}_c^{-1}(\mathbf{r}_o),$$

where

$$\mathbf{g}(\mathbf{q}) \triangleq \mathbf{q} + \sum_{\substack{0 \leq j_1, 0 \leq j_2, \dots, 0 \leq j_{m_q} \\ j_1 + j_2 + \dots + j_{m_q} \leq N-2}}^N \boldsymbol{\alpha}_{j_1 j_2 \dots j_{m_q}} q_1^{j_1} q_2^{j_2} \dots (q_i - q_\alpha)^{j_i+2} \dots q_{m_q}^{j_{m_q}},$$

$$\mathbf{l}_c(\mathbf{q}) \triangleq \mathbf{l}(\mathbf{q})|_{q_i=q_\alpha} + \left. \frac{\partial \mathbf{l}}{\partial q_i} \right|_{q_i=q_\alpha} (q_i - q_\alpha).$$

From Eqn.(12),(18),(20) the optimal trajectory generator is found as the solution of

$$\min_{\boldsymbol{\alpha}} E(\boldsymbol{\alpha}) = \sum_{\mathbf{r}_o, \dot{\mathbf{r}}_o, \ddot{\mathbf{r}}_o} (\boldsymbol{\tau}^T K_\tau \boldsymbol{\tau} + \boldsymbol{\omega}^T K_\omega \boldsymbol{\omega}), \quad (22)$$

subject to

$$C\boldsymbol{\alpha} = \begin{bmatrix} \mathbf{q}(t_0) - \mathbf{q}_o(t_0) \\ \dot{\mathbf{q}}(t_0) - \dot{\mathbf{q}}_o(t_0) \end{bmatrix},$$

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

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

where  $K_\tau$  and  $K_\omega$  are positive definite diagonal matrices.

To compute the minimum in Eqn.(22) various target trajectories are needed. But if the number of possible trajectories is large, it is difficult to consider every one of them. And if impossible trajectories are considered, the system performance goes down. To solve these problems it is necessary to optimize parameters using an online method.

The total control diagram is described in Fig.4. The parameters of the trajectory generator are changed depending on the dynamics and the kinematics of the manipulator and of the object. The adaptation can be executed in both online and offline modes. However, in dynamically changing environments it should be executed online.

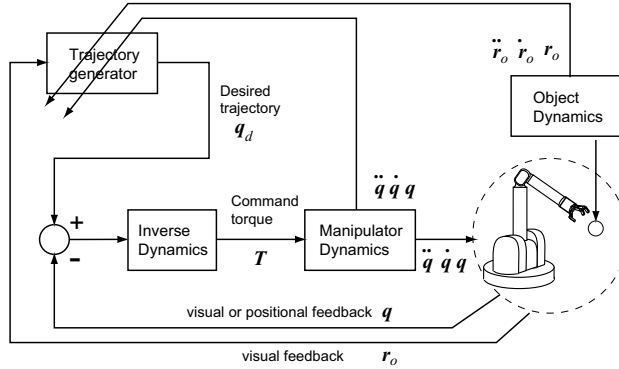


Fig. 4. Trajectory generation algorithm.

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

## 4 Experiment

### 4.1 Experimental system

The experimental system is shown in Fig.6 (a), and the specification of the system is described in table 1.

The vision is a massively parallel vision system called CPV (column-parallel high speed vision system) [10]. The CPV has  $128 \times 128$  photo detectors and pixel parallel processing array based on vision chip architecture and exclusive summation circuit for calculating moment values. Because visual processing is executed in parallel in the processing array, high-speed visual processing (moment detection, segmentation, etc) is realized within 1ms.

The arm is a 4-axis manipulator (Barrett Technology Inc.). The maximum speed of the end-effector is 6m/s, and its maximum acceleration is  $30\text{m/s}^2$ . Since the velocity of the human arm is about 6 – 10m/s, the motion of the arm is as fast and reactive as a human motion. The hand is a 3-fingered and 3-axis hand. By reducing the weight of each finger and using high power actuators, fingers can close a 90-degree angle in 20ms.

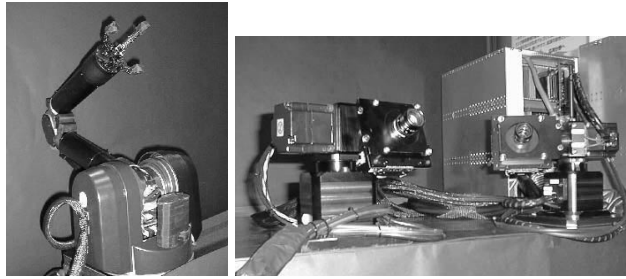
In Fig.6(b) kinematics of the arm is drawn. The arm has 4-axis joints, but, we used only the axis 1, the axis 2, and the axis 4 to simplify the problem. The axis 3 was fixed at the zero point. The initial position of the arm is  $\mathbf{q} = [0, \frac{\pi}{4}, 0, 0]$ , and  $q_4 = \frac{\pi}{2}$ .

The target was thrown in the field of two active vision system (AVS-II). Early image processing in CPV are achieved in the following order: segmentation of the image, extraction of the target area, and computation of the image moments. From these data, the position  $\mathbf{r}^o$  of the target is computed. At last a Kalman filter is used on the observed target position to remove noise.

### 4.2 Experimental Result

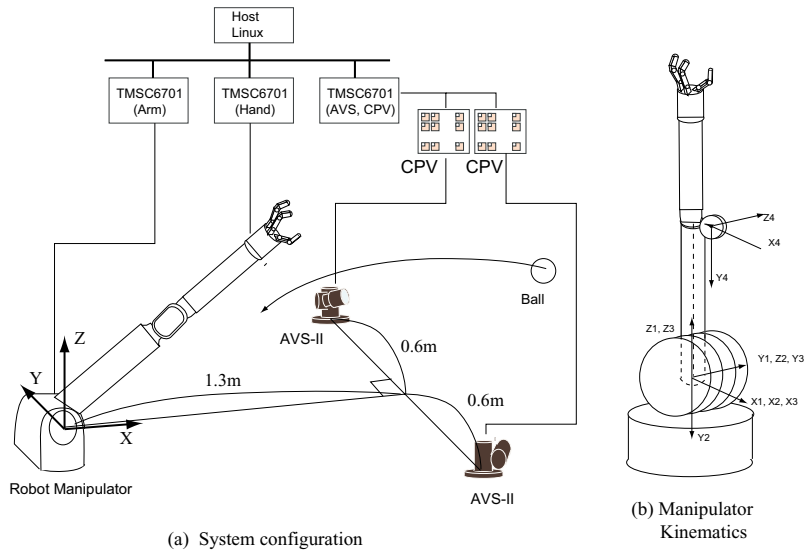
We used a 4-th order polynomial expression in Eqn.(11). Parameters of this polynomial expression were optimized using all possible target trajectories in the neighborhood of the catching point. To simplify the problem we used an offline optimization method in this experiment. Fig.7 shows changes of the cost functions during optimization process. The cost of dynamics constraints  $\sum \boldsymbol{\tau}^T K_{\boldsymbol{\tau}} \boldsymbol{\tau}$  and the cost of kinematics constraints  $\sum \boldsymbol{\omega}^T K_{\boldsymbol{\omega}} \boldsymbol{\omega}$  were reduced at each optimization step.

Fig.8 shows the catching trajectory with optimization, and Fig.9 shows the time responses of the observed target and the end effector. These shows that smooth catching motion is achieved.



(a) Manipulator (b) Active Vision System (AVS-II)

Fig. 5. System components.



(a) System configuration

(b) Manipulator Kinematics

Fig. 6. Experimental system.

Table 1. System specification

High-speed active vision (AVS-II)	degrees of freedom	2DOF × 2set
	maximum velocity (joint angle)	525[rad/s]
	resolution	128 × 128
	density	8[bit]
High-speed arm	degrees of freedom	4DOF
	maximum velocity (end-effector)	6 m/s
	maximum acceleration (end-effector)	30 m/s <sup>2</sup>
	maximum force (end-effector)	26N
	total length	0.9m
High-speed hand	degrees of freedom	3fingers, 3DOF
	closing time (90 degree)	20[ms]
	finger length	0.1[m]

Fig.10 shows the time response of the joint angles, and Fig.11 shows the time response of the command torque during catching. In each figure the time response is shown both with optimization and without optimization ( $\alpha = \mathbf{0}$ ). In each case the desired trajectory of the end effector approached the target position finally. But, in the unoptimized method, the change of joint angle and command torque were bigger than that of the optimized method. This means that there were useless movement in the unoptimized method. As a result a catching task could not be achieved. On the other hand the optimized trajectory was smoother and there seemed to be not useless movement during catching. As a result stable motion could be achieved.

In Fig.12 the catching task of a sphere is shown as a continuous sequence of pictures taken every frame (33ms). The target speed was fast (about 6m/s), and it is difficult to catch it only using simple feedback control. In our proposed algorithm an optimized mapping from visual information to motor command was computed. As a result there was very little delay in the arm motion, and the catching task was achieved.

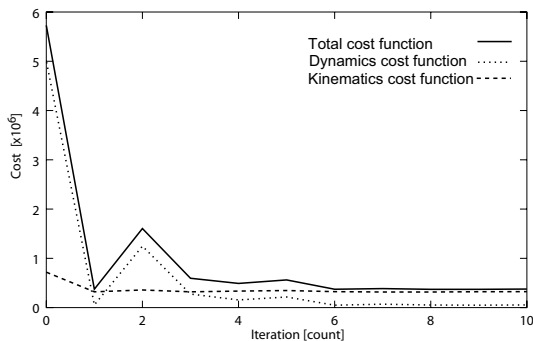
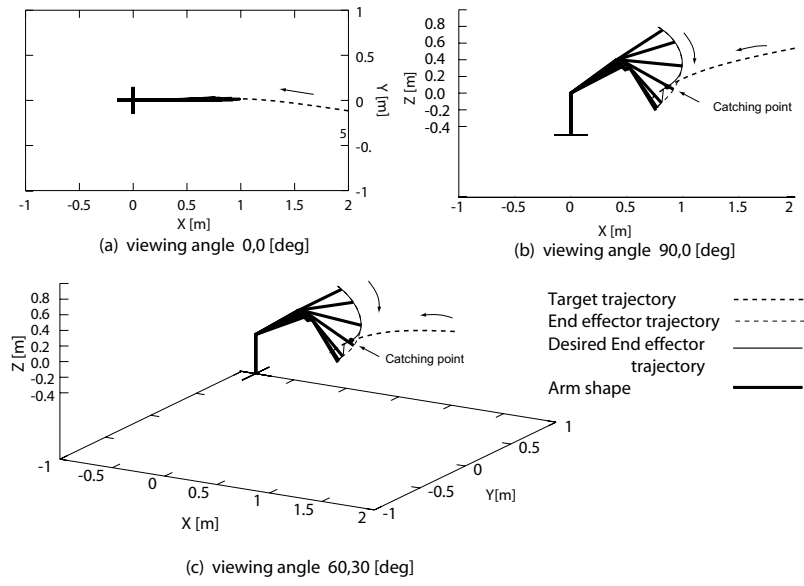


Fig. 7. Optimization of cost functions

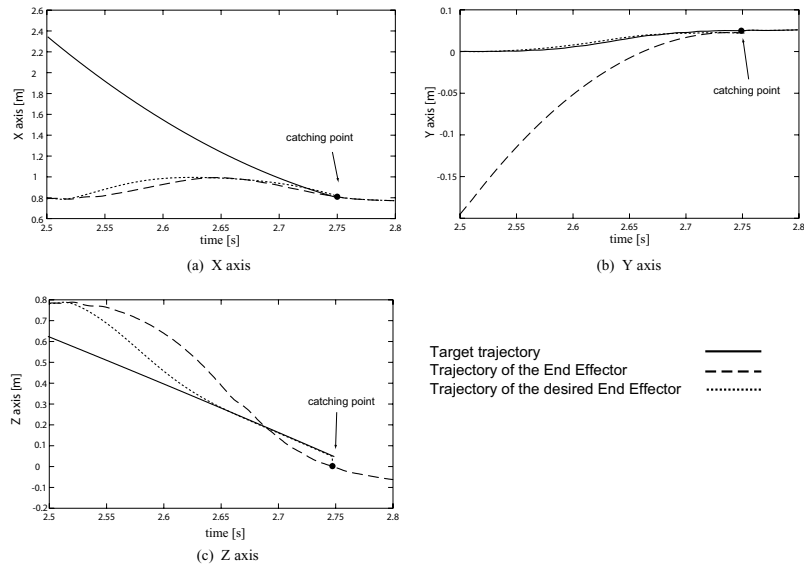
## 5 Conclusion

In this paper we have proposed a realtime online trajectory generation algorithm using high-speed vision. This is based on a nonlinear mapping of visual information to the desired trajectory, and this nonlinear mapping is defined by learning based on constraints of dynamics and kinematics. As a result this method has more advantages to produce a reactive and flexible motion in the case that it is difficult to predict target motion because of disturbance and uncertainty. The proposed method is applied to a catching task.

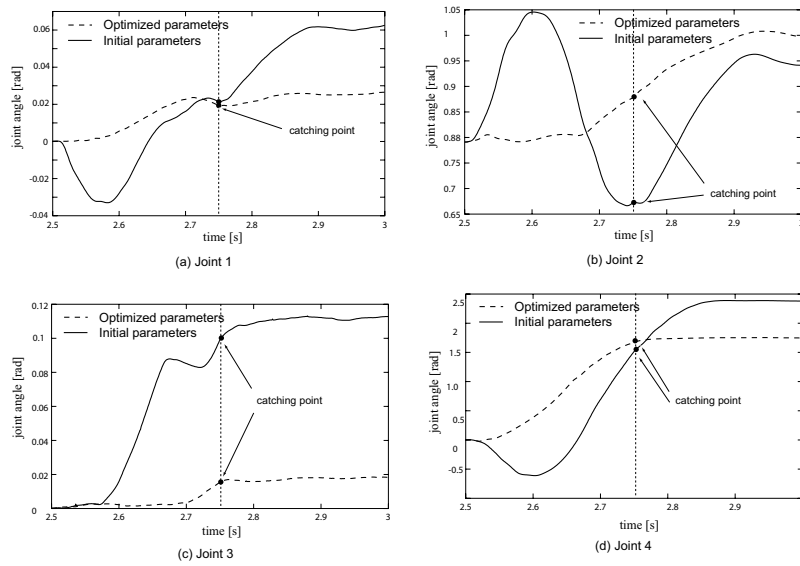
Future works are written as follows: quantitative estimation of effectiveness of mapping, a modular system using several types of mapping, application to other tasks except catching.



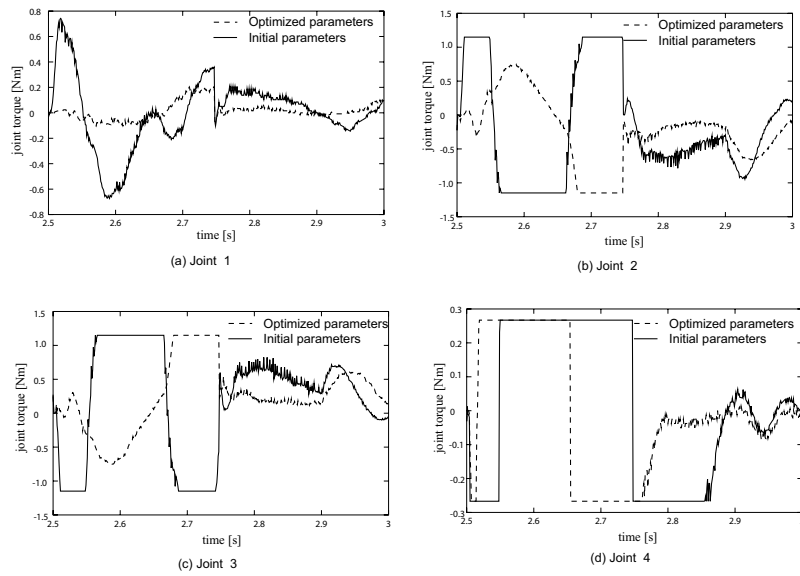
**Fig. 8.** Catching trajectory. Each trajectory is plotted every 1ms, while the arm shape is plotted every 50ms.



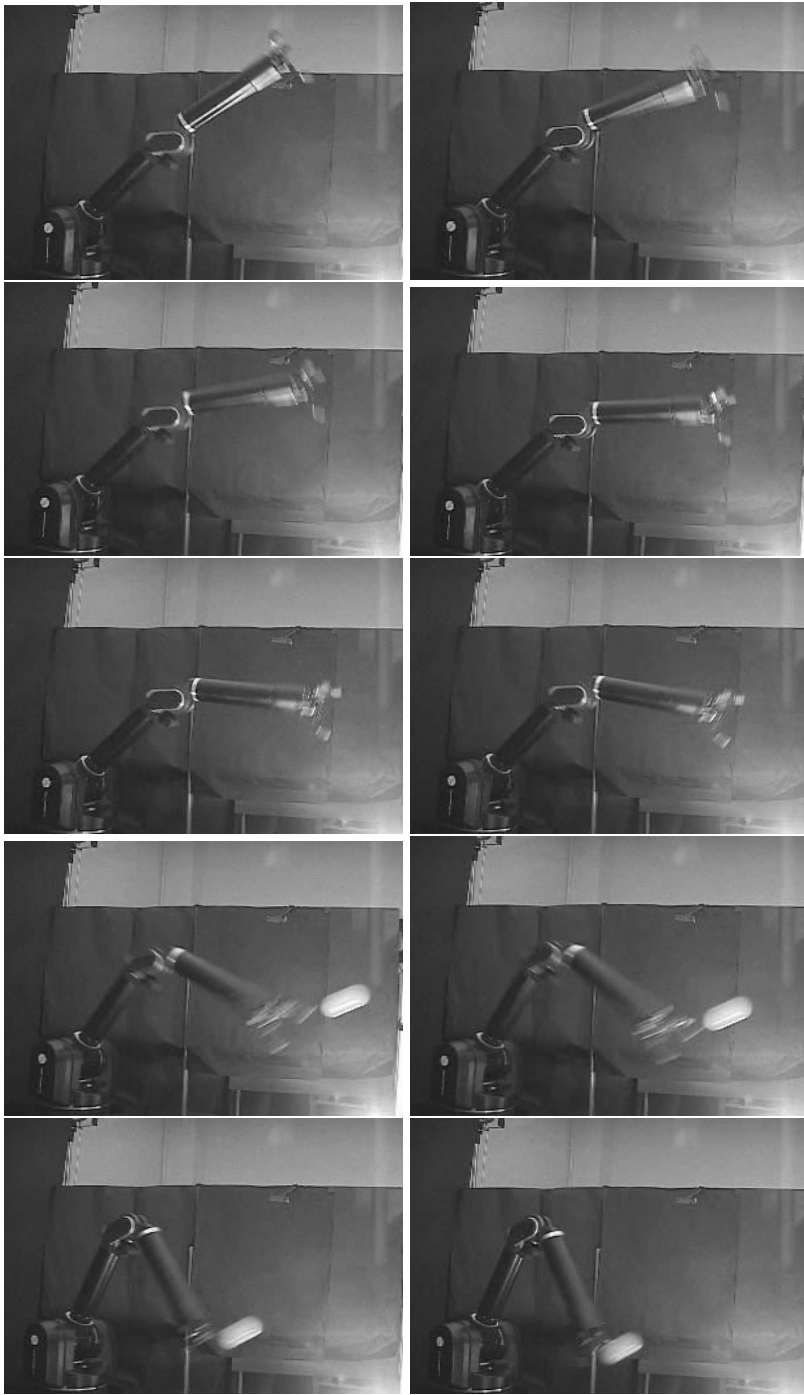
**Fig. 9.** Time response of the target and the end effector.



**Fig. 10.** Time response of joint angles. The dot lines represent the optimized trajectory. The solid lines represent the unoptimized trajectory.



**Fig. 11.** Time response of joint command torques. The dot lines represent the optimized trajectory. The solid lines represent the unoptimized trajectory.



**Fig. 12.** Continuous sequence of pictures taken every frame (33ms).

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

## References

1. Albus, J. (1991). Outline for a theory of intelligence. *IEEE Trans. on Syst., Man, and Cybern.*, 21(3):473–509.
2. Allen, P., Yoshimi, B., and Timucenko, A. (1991). Real-time visual servoing. *Proc. IEEE Int. Conf. Robot. and Automat.*, pages 2376–2384.
3. Bard, C., Turrell, Y., Fleury, M., Teasdale, N., Lamarre, Y., and Martin, O. (1999). Deafferentation and pointing with visual double-step perturbations. *Exp. Brain Res.*, 125:410–416.
4. B.Ghosh, N.Xi, and T.J.Tarn (1999). *Control in robotics and automation: sensor-based integration*. ACADEMIC Press.
5. Buehler, M., Koditshek, D., and Kindlmann, P. (1994). Planning and control of robotic juggling and catching tasks. *Int. J. of Robot. Res.*, 13(2):101–118.
6. Hong, W. and Slotine, J. (1995). Experiments in hand-eye coordination using active vision. *Proc. 4th Int. Symp. on Experimental Robot.*
7. Ishikawa, M., Morita, A., and Takayanagi, N. (1992). High speed vision system using massively parallel processing. *Proc. IEEE Int. Conf. Intelligent Robot. and Systems*, pages 373–377.
8. Komuro, T., Ishii, I., and Ishikawa, M. (1997). Vision chip architecture using general-purpose processing elements for 1ms vision system. *Proc. IEEE Int. Workshop on Computer Architecture for Machine Perception*, pages 276–279.
9. Kovio, A. and Houshangi, N. (1992). Real-time vision feedback for servoing robotic manipulator with self-tuning controller. *IEEE Trans. of Syst., Man, and Cybern.*, 21(1):134–142.
10. Nakabo, Y., Ishikawa, M., Toyoda, H., and Mizuno, S. (2000). 1ms column parallel vision system and its application of high speed target tracking. *Proc. IEEE Int. Conf. Robot. and Automat.*, pages 650–655.
11. Namiki, A. and Ishikawa, M. (2001). Sensory-motor fusion architecture based on high-speed sensory feedback and its application to grasping and manipulation. *Proc. Int. Symp. Robotics*, pages 784–789.
12. Namiki, A., Nakabo, Y., Ishii, I., and Ishikawa, M. (1999a). 1ms grasping system using visual and force feedback. *Video Proc. IEEE Int. Conf. Robot. Automat.*
13. Namiki, A., Nakabo, Y., Ishii, I., and Ishikawa, M. (1999b). High speed grasping using visual and force feedback. *Proc. IEEE Int. Conf. Robot. Automat.*, pages 3195–3200.
14. Namiki, A., Nakabo, Y., Ishii, I., and Ishikawa, M. (2000). 1ms sensory-motor fusion system. *IEEE/ASME Trans. Mechatron.*, 5(3):244–252.
15. Sakaguchi, T., Fujita, M., Watanabe, H., and Miyazaki, F. (1993). Motion planning and control for a robot performer. *Proc. IEEE Int. Conf. Robot. and Automat.*, 3:925–931.
16. Zhang, M. and Buehler, M. (1994). Sensor-based online trajectory generation for smoothly grasping moving objects. *Proc. IEEE Int. Symp. Intelligent Control*, pages 141–146.