Robot Control Using 2D Visual Information Via Database

Yutaka Maeda and Hidetaka Ito
Faculty of Engineering Science
Kansai University
Suita, Japan
maedayut@kansai-u.ac.jp

***Abstract*—This paper proposes a control scheme for a robot system via a database using two-dimensional (2D) visual images. Generally speaking, when we use visual images by cameras to control a robot, three-dimensional (3D) calibration for the position information acquisition is necessary. Moreover, we calibrate the 3D camera coordinate and the robot coordinate. However, these processes are complicated. In addition, especially for hand-eye systems, the calibration is required for every movement of the robot. Thus, the situation is more difficult. In this work, we propose a calibration-free robot system via a database which gives 3D quantity of deviation for the robot arm from 2D image data by two cameras. We also explain a method to establish and update the database effectively using the simultaneous perturbation optimization method. As a result, the hand-eye robot arm system with the proposed control scheme based on visual information can work well without the calibration.**

***Index Terms*—2D Visual Information; Calibration-free; Hand-eye system; Database; Simultaneous Perturbation; Tracking Control;**

I. Introduction

Recently, system control is very important in many industrial fields. There are many methods to control actual systems using certain sensors. It is crucial to establish robust and flexible systems from many aspects [1]. Control using visual information is one of promising approaches in the sense that the system can recognize and adapt to changing or new environment [2], [3], [4]. Robot systems are typical. Visual information has an important role, and thus, research on robot control systems based on the visual information is getting important.

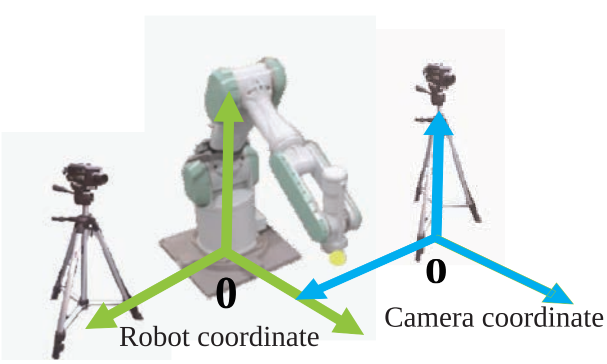


Figure 1. Calibration for robot system.

Now we can assume a robot system controlled by camera vision (see Fig. 1). When we think about control of a robot by the information given by cameras, we generally require three-dimensional (3D) recognition for the position information acquisition.

Using image data from the two CCD cameras like human eyes, the robot can establish 3D coordinates by a camera calibration. However, the calibration is complicated process, because it requires initial setting such as a direct measurement of outside and internal parameters of camera or geometry of the robot [5]. On the other hand, a robot system has its own coordinate. We have to adjust the robot coordinate and the camera one.

Moreover, in case of hand-eye systems in which the robot carries cameras, the calibration is required for every movement, since relation of the robot coordinates and the camera coordinates changes for every movement of the camera in such a system. Therefore the robot control system becomes more complicated and difficult to realize.

In this paper, we solved this problem with a database using two-dimensional (2D) visual image information. That is, we built a calibration-free robot system using the database based on the visual information. The database is established by distance optimization control using the the simultaneous perturbation method. The database converts the image data of the tip of the robot arm and the objective from two CCD cameras into moving quantity.

There are many various complicated tracking control methods by using visual feedback [6]. However, many of them are basically using geometry of robots, cameras or environment of the system. In this work, we do not require such a prior information.

An uncalibrated visual hand-eye system is reported [7]. However, the system contains complicated calculations for Gauss-Newton method with partitioned Broyden's method. On the other hand, our proposed method does not contain such a complex procedure. Instead of the calculation, the database can tell us a moving direction of the robot.

Graefe [8], [9] proposed a system without calibration using a database as well. However, the database does not memorize 3D moving quantities directly.

Section II describes control scheme based on a database. Next section explains meaning of the database. Section IV

 ISBN: 978-1-61208-269-1

shows details of the overall system and how to establish the database using the simultaneous perturbation method. Concrete task using the proposed scheme is demonstrated in Section V. Section VI is conclusion.

II. Robot Control Using Database

Instead of the calibration, we utilize a database. The database realizes a relation of 2D coordinates of two camera images and 3D coordinates of the robot arm. It memorizes proper derivation of movement against the target position.

When we use a single camera for visual control system, the camera converts 3D world into 2D image. Then some information of 3D world disappears. In other words, there is an ambiguity by loss of information in 2D image when using a single camera [10]. The system cannot determine a position of an object in 3D space by this ambiguity of sight if we use a single camera.

Of course, we can remove the ambiguity by using two cameras. Then the system can determine a position of the object in 3D space by 3D calibration. In this work, we realize the relation as the database without the calibration.

Fig. 2 and Table 1 show two 2D images and the database of this study. For camera images, we divided them into 13×20 area(see Fig. 2). This resolution depends on capability of the image processing equipment and size of goal object. The database consists of two camera' 2D coordinates and 3D quantity of movement of the robot arm. The 3D quantity of movement means a deviation between the tip of th robot and the goal position. The database memorizes the moving quantity for every combinations of two 2D coordinates (see Table 1). For examples, we consider the point "a" shown in Fig. 1. In camera A and B, the point "a" is located in (1, 8) and (-3, 6) respectively. Then in order to move the robot arm to the goal point, the 3D deviation ΔX, ΔY and ΔZ is (-10, 20, 55) (see Table 1). Similar to this, when we have the point "b", then locations in camera A and B are (-5,6) and (3,8) respectively, corresponding 3D moving quantity is (30, 35, 45). Using this method, we can obtain the 3D deviation via the database.

If the cameras are equipped with the robot arm and move with the robot arm, the system is called hand-eye system. The cameras have to always catch the tip of the robot arm. We set up the camera position to satisfy this condition. As a result, the robot arm can move without loss of operability of the hand-eye system [10].

The 2D coordinate (0, 0) in the two cameras denotes the robot arm tip, and it is the original point on the display, and this relation of the cameras and the robot arm tip does not change, even if the robot moves.

III. Database for Robot Control

First of all, we have to establish the proper database. At the initial stage, we do not know 3D moving quantities in the database. Fig. 3 shows two cameras images. $d1$ and $d2$ is distance on a straight line between the tip of the robot and its object on two camera images. The sum of the distance D is a clue if the tip arrives at the target.

TABLE I. DATABASE

	Camera A		Camera B		Deviation
	x	y	x	y	$(\Delta X,\Delta Y,\Delta Z)$
	···	···	···	···	···
a point	1	8	-3	6	(-10, 20, 55)
b point	-5	6	3	8	(30, 35, 45)
	···	···	···	···	···

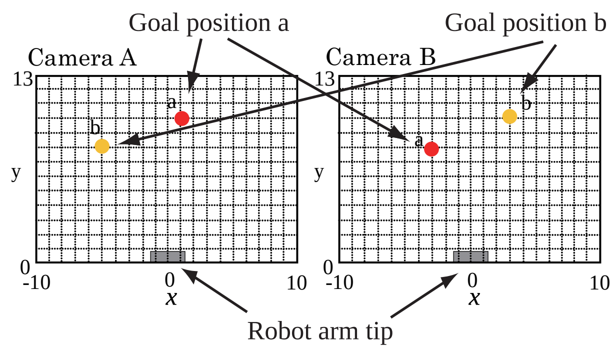


Figure 2. 2D images from two cameras.

$$J(\boldsymbol{w}_t) = D = d1 + d2 \quad (1)$$

If the evaluation D is 0, the tip of the robot arm has arrived at the goal point. Then we can find out how the tip has to move. Memorizing the moving deviation for different positions of the goal, we can establish the database.

We can find the moving quantity for a target position by trial and error. That is, moving the tip of the arm randomly may give the optimal moving quantity by chance. However, this method is extremely ineffective and actually difficult to realize. Then we adopt a distance optimization using the simultaneous perturbation method.

The simultaneous perturbation optimization method was introduced by J. C. Spall [11], [12]. J. Alespector et al. [13] and G. Cauwenberghs [14] describe the same method. Y. Maeda also independently proposed a learning rule using the simultaneous perturbation and reported a feasibility of the learning rule for neural networks [15], [16], [17]. At the same time, the merit of the learning rule was demonstrated in the implementation of neural networks [18], [19], [20], [21]. Convergence conditions of the method in framework of the stochastic approximation is also described [22].

We consider a simple optimization problem. We would like to find a minimum point of a function $J(\boldsymbol{w})$ with the parameter $\boldsymbol{w}$. The simultaneous perturbation recursion using sign vector is described as follows [17], [18], [20], [21];

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \boldsymbol{\Delta w}_t \quad (2)$$

$$\boldsymbol{\Delta w}_t = -\alpha \frac{J(\boldsymbol{w}_t + c\boldsymbol{s}_t) - J(\boldsymbol{w}_t)}{c} \boldsymbol{s}_t \quad (3)$$

Where, $\boldsymbol{w}_t (\in \Re^n)$ and $\boldsymbol{\Delta w}_t$ are an adjustable parameter vector and its modifying quantity at the t-th iteration. $c(>0)$

is a perturbation. $\boldsymbol{s}_t$ is a sign vector whose elements are +1 or -1.

The simultaneous perturbation estimates gradient of the function J using a kind of finite difference, effectively. The most important advantage of the simultaneous perturbation method is its simplicity. The simultaneous perturbation can estimate the gradient of a function using only the two values of the function. Therefore, it is relatively easy to implement for many optimization problems. Moreover even if the function is not differentiable partly, we can apply the method. In our research, we applied the simultaneous perturbation method to obtain optimal value of moving quantity of the robot arm.

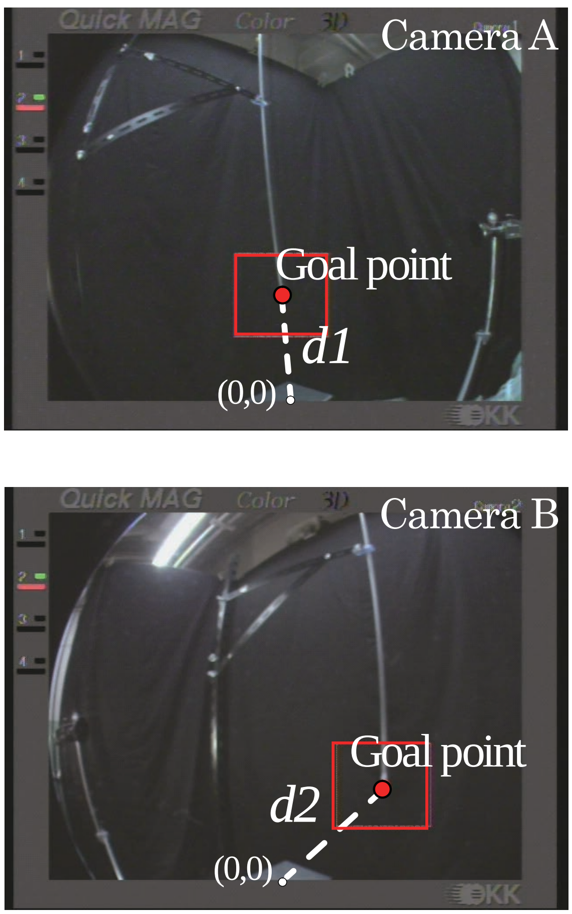


Figure 3. Two camera images.

Now, we explain details of the distance optimization control via the simultaneous perturbation. Fig. 4 shows the flowchart of the distance optimization control via the simultaneous perturbation.

At first, we set a value of learning coefficient α, perturbation c and the other as an initial setting. After the setting, we obtain image data from the two cameras and have the evaluation $J(\boldsymbol{w}_t)$. Where $\boldsymbol{w}_t$=(x_t, y_t, z_t) denotes 3D position of the tip in the robot coordinate at the t-th time. $J(\boldsymbol{w}_t)$ is the evaluation function defined by (1).

After that, we add the perturbation to the tip of the robot arm. That is, the tip moves by the perturbation $c\,\boldsymbol{s}_t$. $\boldsymbol{s}_t$ denotes the 3D sign vector whose elements are +1 or -1, and c is a scalar perturbation. Therefore the perturbation vector $c\,\boldsymbol{s}_t$ takes value of $+c$ or $-c$ and the sign is decided randomly.

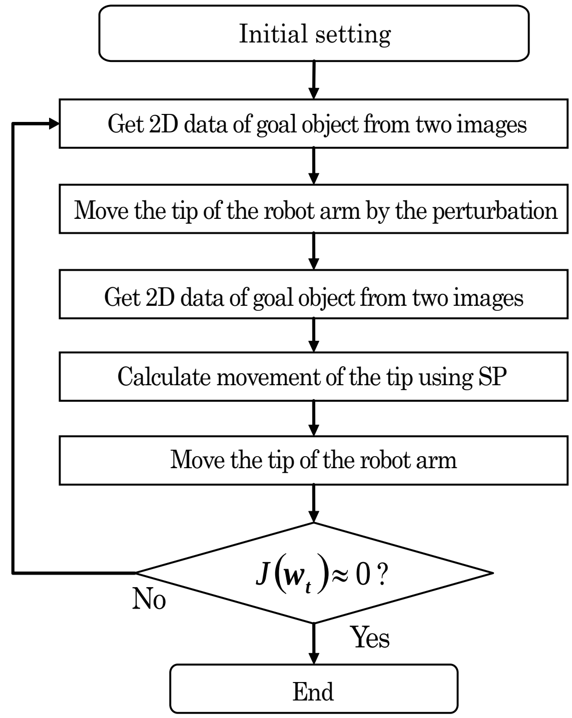


Figure 4. Distance optimization control via simultaneous perturbation.

Afterward we obtain image data again and have the evaluation$J(\boldsymbol{w}_t + c\boldsymbol{s}_t)$. Then we can estimate a gradient of the evaluation function based on two values of the evaluation functions with the perturbation and without the perturbation. This corresponds to the first term of the right hand side of (4). Then we calculate a quantity of movement $\boldsymbol{q}_t$ of the robot arm as follows;

$$\boldsymbol{q}_t = -\alpha \frac{J(\boldsymbol{w}_t + c\boldsymbol{s}_t) - J(\boldsymbol{w}_t)}{c}\boldsymbol{s}_t - c\boldsymbol{s}_t \qquad (4)$$

The robot arm moved by $c\boldsymbol{s}_t$ from the position $\boldsymbol{w}_t$ in order to obtain the perturbed evaluation $J(\boldsymbol{w}_t + c\boldsymbol{s}_t)$. The second term of the right hand side of the above equation $-c\boldsymbol{s}_t$ is for compensation of the perturbed procedure.

The tip of the robot arm moves according to the quantity calculated by (4). At last, we observe evaluation function after the movement.

When the evaluation function $J(\boldsymbol{w}_t)$ is minimized in this observation, in other words, when $J(\boldsymbol{w}_t)$ is in the neighborhood of 0, the tip of the robot arm arrived at goal point. If the evaluation function is still large, we repeat this process until the evaluation function converges to 0.

Obtained moving deviation is memorized in the database. Repeating the series of procedure for different positions of the target establishes the proper database.

 ISBN: 978-1-61208-269-1

IV. Experimental System

A. Experiment equipments

Fig. 5 shows the experimental system in this study. The robot arm PA-10 (Mitsubishi Heavy Industry) is a control object in this system. The tip of the robot arm has a metal plate in Fig. 6. The controller consists of a servo driver, a personal computer (PC) and an image processing equipment (Quick MAG). Two CCD cameras that are equipped on the robot arm take images of color marker of the target. Quick MAG produces 2D coordinate data of the color marker of two image data. The data is sent to PC, and it calculates quantity of movement of the robot arm based on the proposed scheme. Servo driver receives the calculated value and converts it into an actuating signal of the robot arm. The robot arm tip moves to a certain direction of the robot coordinate by the signal.

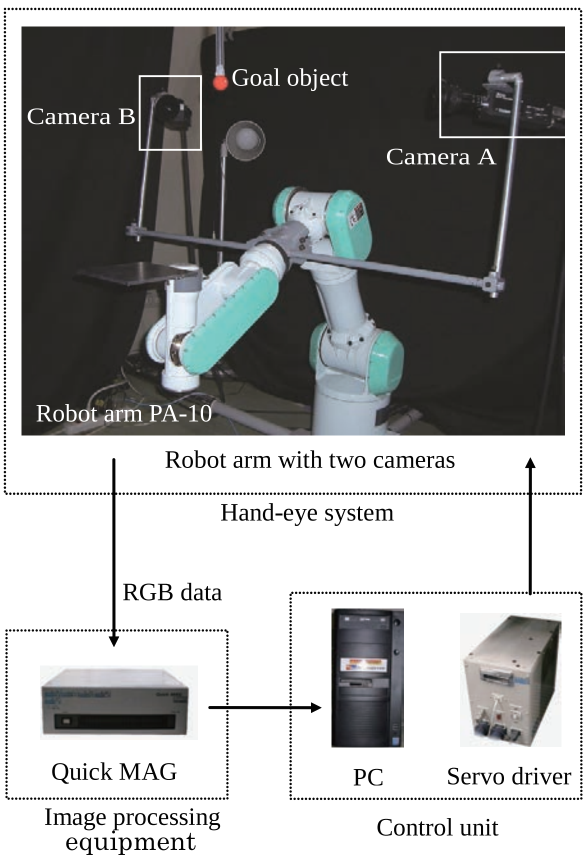


Figure 5. Calibration-free robot system.

This system uses only 2D image data to calculate quantity of movement of the robot arm without 3D calibration. In this sense, this system is a calibration free robot system.

B. Establishment of database via simultaneous perturbation

Fig. 7 shows the flowchart of a process to establish the database. By this process, the robot arm moves and the database is also redeemed for every movement.

In this operation, there are three movement patterns. If there is a datum for a goal position in the database, then the robot arm obtains 3D quantity of deviation from the database. The tip of the robot arm can move quickly to the goal point. Then the database does not modified.

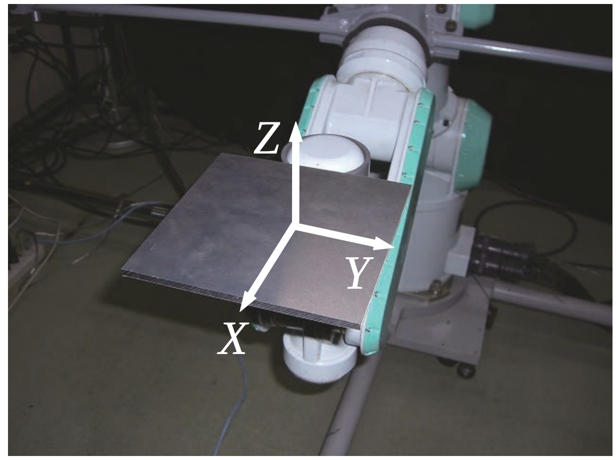


Figure 6. Tip of the robot arm.

If the database does not contain a datum for the goal position, the system searches data in the search area, that is, the neighbor area(see Fig. 8). If there is a datum in the search area, then the robot arm can obtain 3D quantity of deviation for a neighborhood position and moves to the neighborhood point.

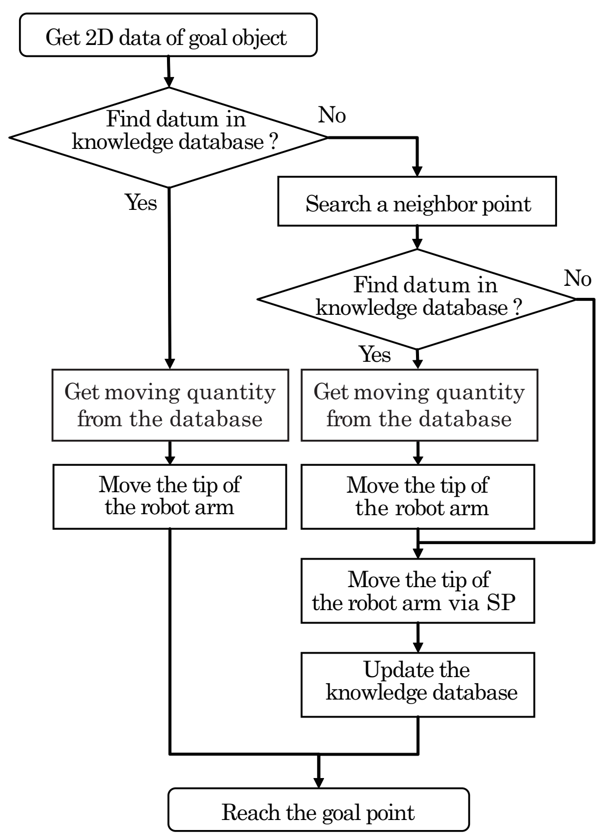


Figure 7. Flowchart of establishing database.

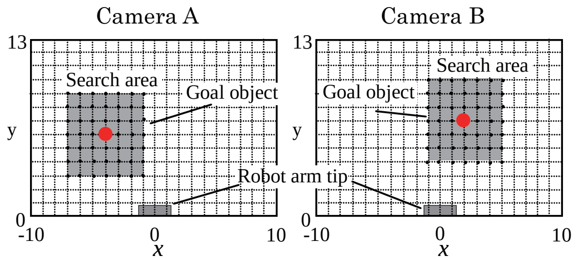
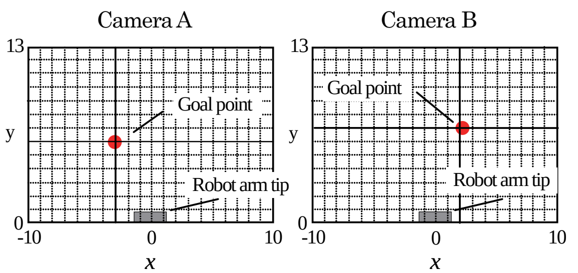
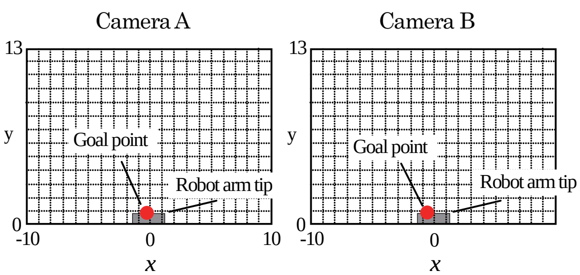


Figure 8. Search of the neighbor area.



Camera A		Camera B		Robot arm
x	y	x	y	$(\Delta X, \Delta Y, \Delta Z)$
-3	6	2	7	No data

(a) Before update



Camera A		Camera B		Robot arm
x	y	x	y	$(\Delta X, \Delta Y, \Delta Z)$
-3	6	2	7	(-25,30,75)

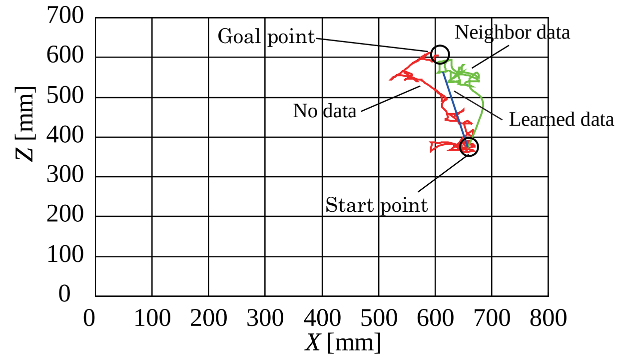
(b) After update

Figure 9. Update of the database.

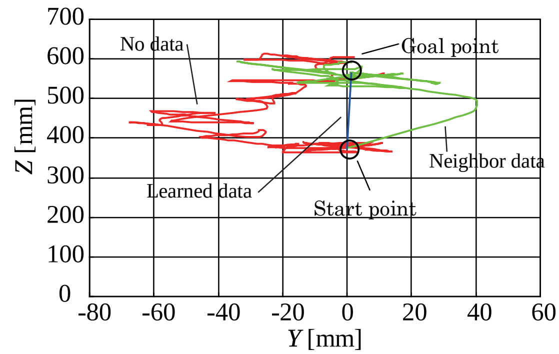
After the movement, the robot arm searches the goal object by the distance optimization control via the simultaneous perturbation. As the robot arm can obtain 3D quantity of the deviation from the initial point to the goal point, the database is updated(see Fig. 9).

Provided that there is no datum in the neighborhood area for the goal position, the robot arm searches the goal object by the distance optimization control via the simultaneous perturbation from the beginning. When the arm arrived at the goal, we updates the database(see Fig. 9).

Totally, the database is effectively established. That is, we can construct the precise database through this procedure. The database contains about 1000 data. Especially, main area is covered by these data.



(a) X-Z plane



(b) Y-Z plane

Figure 10. Locus of the robot arm.

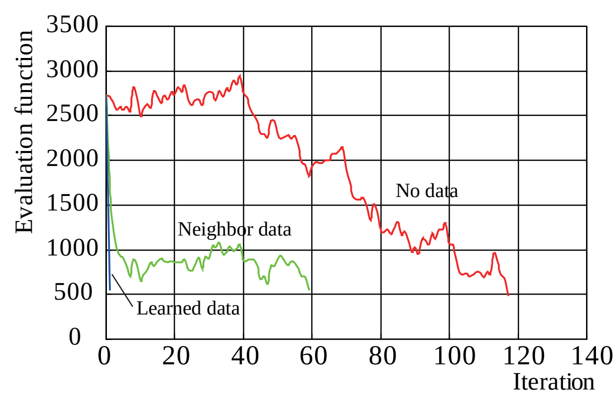


Figure 11. Change of evaluation function.

Fig. 10 shows loci of the arm tip on Robot X-Y plane and Y-Z plane for this process. Notice that the locus of the robot arm tip moves to the goal object quickly, provided that there is a datum in the database(see blue line in Fig. 10). If the database does not have corresponding datum and has data for neighbor area, the arm moves to the nearest point first and moves to the goal point from the point by the simultaneous perturbation method. Therefore, the robot arm moves to a certain position first as straight as a line, then the distance optimization process is carried out(see green line in Fig. 10). If the database has no datum, the arm tip moves to the goal point from start point using the distance optimization(see red line in Fig. 10).

The more the system moves, the more the accuracy of the database increases. The robot arm becomes to move quickly based on the database(see Fig. 11).

Fig. 11 shows typical changes of the evaluation function. If there exists a datum in the database, the robot can move to the objective position immediately, that is, the evaluation is zero at this moment. If there exists a datum in neighbor area, value of the evaluation started a certain value and decreases gradually. If there is no datum, the evaluation function decreases via the distance optimization gradually. We set that end condition for the evaluation function is 500.

V. Moving Tasks for Real-time Control

In order to confirm feasibility of the proposed control scheme, we tried real time robot control task. The task is that the tip of the robot arm attached a white marker pursue a target red ball. The system controls the robot with $c = 30$, $\alpha = 2.0$.

The system is monitoring the target object constantly. If the database contains a datum for the present position of the target, the robot arm moves to the target position using deviation in the database. Since we have enough data around assumed space, the system works well.

It is interesting that real-time robot control is carried out using the database approach.

VI. Conclusion

Calibration is crucial but troublesome for robot control only using visual information. In this paper, we propose a calibration free robot control system using 2D camera images via a database. Moreover, we described a method to construct a proper database effectively. This scheme is unsuitable for precise robot control. However, it is relatively easy for rough control. We confirmed feasibility and effectiveness of the proposed real-time control scheme for the hand-eye robot system through moving task.

Acknowledgment

This work is financially supported by Grant-in-Aid for Scientific Research (No.23500290) of the Ministry of Education, Culture, Sports, Science and Technology(MEXT) of Japan, and Nonverbal Communication Technology and Media Information Communication Technology Research group of ORDIST, Kansai University.

References

[1] E. Kiyak and F. Caliskan, "Design of Fault Tolerant Flight Control System," WSEAS Transactions on Systems and Control, 5, 2010, pp. 454 – 463.

[2] K. Hashimoto, T. Ebine, and H. Kimura, "Visual servoing with hand-eye manipulator-optimal control approach," IEEE Transactions on Robotics and Automation, 12, 1996, pp.766 – 774.

[3] G.D. Hager, Wen-Chung Chang, and A.S. Morse, "Robot hand-eye coordination based on stereo vision," IEEE Control Systems, 15, 1995, pp.30 – 39.

[4] G.D. Hager, "A modular system for robust positioning using feedback from stereo vision," IEEE Transactions on Robotics and Automation, 13, 1997, pp.582 – 595.

[5] Sang De Ma, "A self-calibration technique for active vision systems," IEEE Transactions on Robotics and Automation, 12, 1996, pp.114 – 120.

[6] J.A. Piepmeier and H. Lipkin, "Uncalibrated Eye-in-Hand Visual Servoing," The International Journal of Robotics Research, 22, 2003, pp.805–819.

[7] Q. Xie, V. Graefe, and K. Vollmann, "Using a Knowledge Base in Manipulator Control by Calibration-Free Stereo Vision," IEEE International Conference on Intelligent Processing System, 1997, pp.1307–1311.

[8] V. Graefe and A. Maryniak, "The Sensor-Control Jacobian as a Basis for Controlling Calibration-Free Robots," IEEE International Symposium on Industrial Electronics, 1998, pp.420–425.

[9] A. Satou, Computer vision - Geometry of sight -, Corona publishing, 1999 (in Japanese).

[10] S. Matsuda and Y. Maeda, "Hand-eye System Via Knowledge Base Using Image Data," Proceedings of 2006 International Symposium on Flexible Automation, 2006, pp.942–945.

[11] J.C. Spall, "A stochastic approximation technique for generating maximum likelihood parameter estimates," Proceedings of the 1987 American Control Conference, 1987, pp. 1161–1167.

[12] J.C. Spall, "Multivariable stochastic approximation using a simultaneous perturbation gradient approximation," IEEE Transactions on Automatic Control, 37, 1992, pp.332–341.

[13] J. Alespector, R. Meir, B. Yuhas, A. Jayakumar, and D. Lippe, "A parallel gradient descent method for learning in analog VLSI neural networks," in: S. Hanson, J. Cowan, C. Lee (Eds.), Advances in neural information processing systems, Vol. 5, Morgan Kaufmann Publisher, Cambridge, MA, 1993, pp. 836–844.

[14] G. Cauwenberghs, "A fast stochastic error-descent algorithm for supervised learning and optimization," in: S.J.Hanson, J.D.Cowan, C. Lee (Eds.), Advances in neural information processing systems, Vol. 5, Morgan Kaufmann Publisher, Cambridge, MA, 1993, pp. 244–251.

[15] Y. Maeda and Y. Kanata, "A learning rule of neural networks for neuro-controller," Proceedings of the 1995 World Congress of Neural Networks, 2, 1995, pp.402–405.

[16] Y. Maeda and R.J.P.de Figueiredo, "Learning rules for neuro-controller via simultaneous perturbation," IEEE Transaction on Neural Networks, 8, 1997, pp.1119–1130.

[17] Y Maeda, Y. Fukuda, and T. Matsuoka, "Pulse Density Recurrent Neural Network Systems with Learning Capability Using FPGA," WSEAS Transactions on Circuits and Systems, 7, 2008, pp.321–330.

[18] Y. Maeda, H. Hirano, and Y. Kanata, "A learning rule of neural networks via simultaneous perturbation and its hardware implementation," Neural Networks, 8, 1995, pp.251–259.

[19] G. Cauwenberghs, "An analog VLSI recurrent neural network learning a continuous-time trajectory," IEEE Transactions on Neural Networks, 7, 1996, pp.346–361.

[20] Y. Maeda, A. Nakazawa, and Y. Kanata, "Hardware implementation of a pulse density neural network using simultaneous perturbation learning rule," Analog Integrated Circuits and Signal Processing, 18, 1999, pp.153–162.

[21] Y. Maeda and T. Tada, "FPGA implementation of a pulse density neural network with learning ability using simultaneous perturbation," IEEE Transactions on Neural Networks, 14, 2003, pp.688–695.

[22] J.C.Spall, Introduction to Stochastic Search and Optimization, John Wiley & Sons, Inc., 2003.