The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

# Implementation of a time-delayed controller on an FPGA for ROBOKER by estimating acceleration with different filters

Hyo-Won Jeon and Seul Jung

*Intelligent Systems and Emotional Engineering Lab. Chungnam National University, Daejeon, Korea*
*(Tel : 82-42-821-6876; Fax : 82-42-823-4919)*
*(Email address: atlantis44@naver.com, jungs@cnu.ac.kr)*

*Abstract*: This paper presents the hardware design and implementation of the time-delayed controller for controlling position of the ROBOKER. The ROBOKER has two arms with 12 degrees-of-freedom. The time-delayed control method is very simple and robust to outer disturbances if two conditions are met. Two conditions are good estimation of inertia and acceleration with an appropriate sampling time. Those estimated values are used to cancel out uncertainties in the system by subtracting with one sample-delayed information. However, estimating acceleration is not easy and often using the finite difference method which is the 1st filter adds noises to the system so that control performances are degraded. Thus, to improve the performance of the time-delayed controller, the Kalman filter is designed and tested to estimate acceleration terms better. Performances of the designed filters are tested by controlling the ROBOKER arms.

***Keywords***: Time-delayed controller, ROBOKER, Kalman filter.

## I. INTRODUCTION

Recently, robots are getting smarter and smaller to satisfy the high level standard of human desires. To make robots smarter and smaller, the brain-like inside controllers are required to be smaller as well so that it is embedded for conducting specific tasks. This leads to a birth of the area of embedded robotics.

One typical embedded controller implementation is the controller-on-chip (COC) that has controllers implemented on a single FPGA (Field Programmable Gate Array) chip. The well known PID motion controllers can be designed and embedded on a single chip[1,2]. This leads to the advantages of having a stand alone-type controller instead of using PCs.

Since the FPGA chip has the remarkable expandability to pan out many functions such as PWM generation, encoder counters, and data collection, the FPGA can provide solutions for defects of microprocessors or DSPs.

Practical applications of using FPGA chips are a multi-axes control of the humanoid robot arms system and a tele-operation control system [3]. In a tele-operated system, the slave robot arms can be controlled by a single chip controller through collected motion data from the master arm [4]. For the robot arms, there are many joints to be controlled. The ROBOKER has 12 joints. Thus, a single hardware is preferred to control all the joints.

For the accurate position control of the robot arms, the robust control algorithms are required other than PID controllers. Although PID controllers are efficient in implementation and cost-effectiveness, they suffer from nonlinearities of the robot system. This results in large tracking errors. This requires the nonlinear controller for the nonlinear system.

The time-delayed controller is one of robust control algorithms that use the previous information of the system to cancel out uncertainties [5-8]. Although the time-delayed controller is simple and practical such that it does not require the dynamic model of the system, there are two requirements to be satisfied for the better results. One is the estimation of the inertial information of the system. The inertial information is usually approximated by constants for simplicity. The other requirement is to estimate the acceleration term. The inaccurate estimation of aforementioned two terms leads to the tracking errors of the robot arms.

Therefore, in this paper, the acceleration term required in the time-delayed controller is estimated by the Kalman filter. Estimating acceleration by the Finite Difference Method (FDM) and the Kalman Filter is applied to robot arm control, and then performances are compared. The inertial term is also estimated experimentally by trial and error procedure.

Experimental studies of position control of the ROBOKER arm are conducted. Through experimental results, we found that the performance of the Kalman filter is better than that of the FDM.

## II. TIME-DELAYED CONTROL

The robot dynamic equation is given by

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \qquad (1)$$

where $q$ is an n×1 joint vector, $\tau$ is an n×1 torque vector, $D(q)$ is an n×n inertia matrix, $C(q,\dot{q})\dot{q}$ is

an n×1 Coriolis and centrifugal force vector, $G(q)$ is an n×1 gravity force vector.
The control law is

$$\tau = \hat{D}u + \hat{C} + \hat{G} \tag{2}$$

where u is given by

$$u \cong \ddot{q} = \ddot{q}_d + k_D\dot{e} + k_P e \tag{3}$$

In the equation (1), the Corioils and centrifugal forces with other uncertain terms can be estimated with one sample delayed information as

$$h(t) = \tau(t-\lambda) - \hat{D}(t-\lambda)\ddot{q}(t-\lambda) \tag{4}$$

where, $h(t) = C(q,\dot{q})\dot{q} + G(q)$. Combining (2), (3), and (4) yields the time-delayed control law as

$$\tau = \hat{D}(t)u(t) + \tau(t-\lambda) - \hat{D}(t-\lambda)\ddot{q}(t-\lambda) \tag{5}$$

Thus, the overall control law with the constant $\hat{D}$ becomes

$$\tau = \hat{D}(\ddot{q}_d + k_D\dot{e} + k_P e) + \tau(t-\lambda) - \hat{D}\ddot{q}(t-\lambda) \tag{6}$$

To have the better performance, the acceleration term $\ddot{q}(t-\lambda)$ should be accurate. In a practical situation, however, it is very difficult to have exact values. The estimated values are used [5].

## III. KALMAN FILTER

### 1. Kalman Filter

To estimate the acceleration term, the Kalman filter is used twice, one for the velocity and the other for the acceleration. The estimation has been done with encoder measurements.

Since the velocity and the acceleration term are estimated from the position measurement, the system for the velocity estimation is given by

$$x_{k+1} = \begin{bmatrix} 1 & Ts \\ 0 & 1 \end{bmatrix} x_k \quad , \quad y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k \tag{7}$$

The state can be estimated by the process shown in Fig. 1.

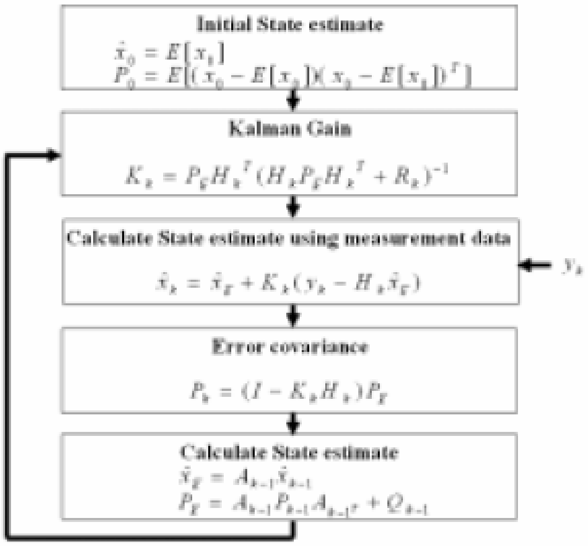

Fig.1. Procedure for estimating states

### 2. Kalman Filter Design on FPGA

To implement on the FPGA, each calculation has been done sequentially. The following equation is actually implemented on the FPGA. Block A calculates equations (8) and (9). Equations (10)-(13) are calculated in the block B. Block C calculates equations (14)- (16) and block D calculates (17)-(20).

$$\hat{x}_{\bar{k}}(1,1) = \hat{x}_{k-1}(1,1) + Ts \cdot \hat{x}_{k-1}(2,1) \tag{8}$$

$$\hat{x}_{\bar{k}}(2,1) = \hat{x}_{k-1}(2,1) \tag{9}$$

$$\begin{aligned} P_{\bar{k}}(1,1) = {} & P_{k-1}(1,1) + Ts \cdot \{P_{k-1}(1,2) \\ & + P_{k-1}(2,1)\} + Ts^2 P_{k-1}(2,2) \end{aligned} \tag{10}$$

$$P_{\bar{k}}(1,2) = P_{k-1}(1,2) + Ts \cdot P_{k-1}(2,2) \tag{11}$$

$$P_{\bar{k}}(2,1) = P_{k-1}(2,1) + Ts \cdot P_{k-1}(2,2) \tag{12}$$

$$P_{\bar{k}}(2,2) = P_{k-1}(2,2) + Q \times 2^{14} \tag{13}$$

$$Den = P_{\bar{k}}(1,1) + R \times 2^{14} \tag{14}$$

$$\hat{x}_k(1,1) = \hat{x}_{\bar{k}}(1,1) + \frac{(y_k - \hat{x}_{\bar{k}}(1,1))}{Den} P_{\bar{k}}(1,1) \tag{15}$$

$$\hat{x}_k(2,1) = \hat{x}_{\bar{k}}(2,1) + \frac{(y_k - \hat{x}_{\bar{k}}(1,1))}{Den} P_{\bar{k}}(2,1) \tag{16}$$

$$P_k(1,1) = P_{\bar{k}}(1,1) - \frac{P_{\bar{k}}(1,1)^2}{Den} \tag{17}$$

$$P_k(1,2) = P_{\bar{k}}(1,2) - \frac{P_{\bar{k}}(1,1) \cdot P_{\bar{k}}(1,2)}{Den} \tag{18}$$

$$P_k(2,1) = P_{\bar{k}}(2,1) - \frac{P_{\bar{k}}(1,1) \cdot P_{\bar{k}}(2,1)}{Den} \tag{19}$$

$$P_k(2,2) = P_{\bar{k}}(2,2) - \frac{P_{\bar{k}}(1,2)\cdot P_{\bar{k}}(2,1)}{Den} \quad (20)$$

The schematic design of the filter is shown in Fig. 2.

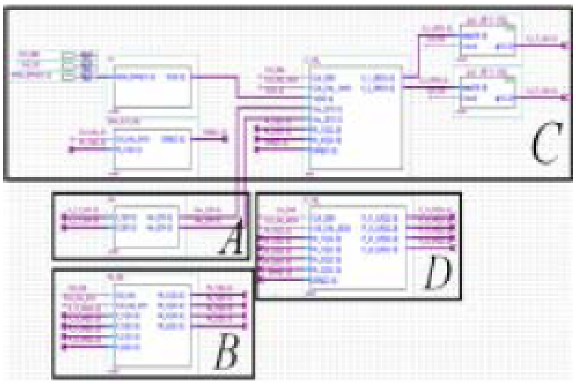

Fig.2. Kalman filter implementation

In case of hardware programming, synchronization of the timing for calculation of each process is very critical. Therefore, the calculation time of each block should be synchronized to have stable data.

Fig.3 shows the cascaded structure of the Kalman filter to estimate the acceleration.

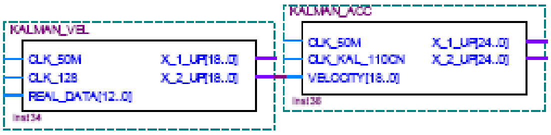

Fig. 3 Kalman Filter Acceleration module

## IV. FPGA IMPLEMENTATION of TIME DELAY CONTROLLER

Fig.4 shows the time-delayed control module designed by hardware description language (HDL). The program is downloaded on the FPGA chip. The module contains a serial communication, PWM generator, and encoder counters.

## V. EXPERIMENT

The ROBOKER arm is required to follow the specified trajectory for testing. Fig. 5 shows the humanoid Robot Arm, ROBOKER. The controller gains P, D gains are selected as 15, 1 respectively. The inertia values are selected as $\hat{D} = \frac{1}{128} = 0.0078125$ by trial and error experiments. The sampling frequency is 128Hz.

For the Kalman parameters, Q and R are selected as 1, 0.0001(in velocity) and 0.005, 0.001(in acceleration), respectively.

Fig. 5 The ROBOKER

Figs 6 and 7 show the tracking result by the FDM and the Kalman filter. We see the performance with the Kalman filter shown in Fig. 7 is better than that with the FDM of Fig. 6.

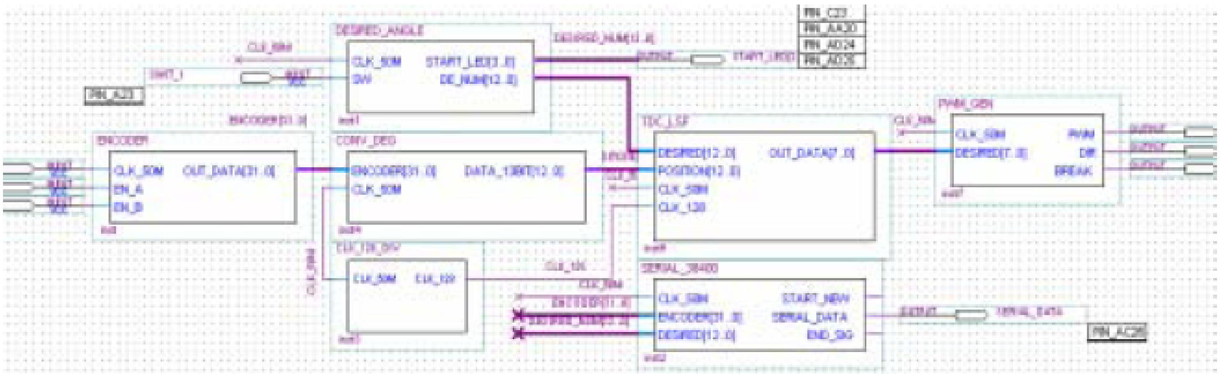

Fig. 4 Time-delayed controller design

The robot follows the desired trajectory, but shows the small tracking errors. This error seems to occur due to the modeling error of the Kalman parameters such as Q and R. The other possibility is to take the system dynamics into no account. Here the estimation of the acceleration is totally dependant upon the encoder measurements. We also tested the tracking task with the finite difference method. The result is worse than that of the Kalman filter.

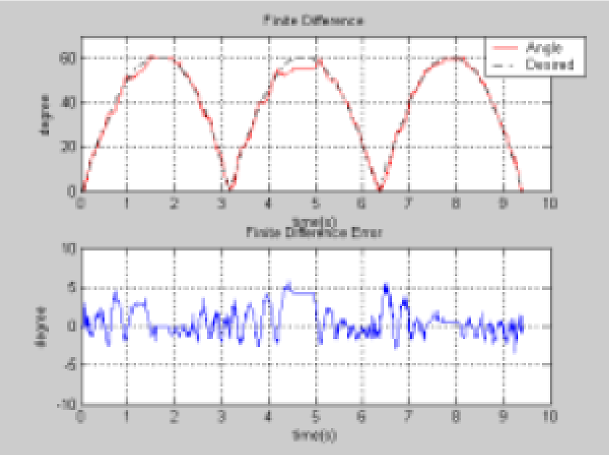

Fig.6. Position control of ROBOKER arm with the finite difference method

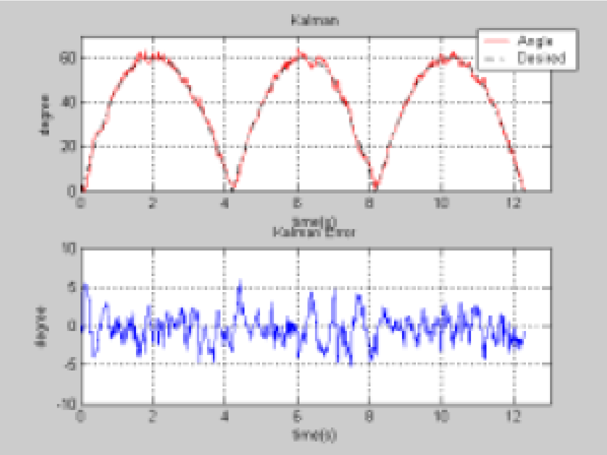

Fig.7. Position control of ROBOKER arm with the Kalman filter

## VI. CONCLUSION

The Kalman filter has been embedded on the FPGA to estimate the acceleration of the system from the encoder measurements. The time-delayed controller is also embedded to control the robot arm. The tracking task of the ROBOKER arm to follow the trajectory has been performed. We have observed the slight tracking error. To improve the performance, several issues are raised. The dynamic of the whole system should be taken into account so that the acceleration can be more accurately estimated.

## Acknowledgement

This research has been jointly supported by the 2006 CNU research foundation and the Korea research foundation in 2007.

## REFERENCES

[1] Sungsu Kim and Seul Jung(2006), "Hardware implementation of a neural network controller with an MCU and am FPGA for nonlinear systems", International Journal of Control, Automation, and Systems Engineering, 4,(5): 567-574

[2] Seul Jung and Sungsu Kim(2007), "Hardware implementation of a real-time neural network controller with a DSP and an FPGA for nonlinear systems, IEEE Transaction on Industrial Electronics, 4(5):265-271

[3] W. K. Lee, D. H. Song, and Seul Jung(2004), "Control and interface between an exoskeleton master robot and a human like slave robot with two arms", IEEE AIM, pp. 319-324

[4] WoonKyu Lee and Seul Jung(2006) ,"FPGA design for controlling humanoid robot arms by exoskeleton motion capture system", IEEE ROBIO, pp. 1378-1383

[5] T. C. Hsia, L.S. Gao(1990), "Robot Manipulator Control Using Decentralized Linear Time-Invariant Time-Delayed Joint Controllers", IEEE pp.2070-2075

[6] K. Yousef-Toumi and O. Ito(1990), "A Time-delay controller for systems with unknown dynamics", Journal of Dynamic Systems, measurement, and Control", vol. 112:133-142

[7] S. Jung, T. C. Hsia, and R. G. Bonitz(2001), "Force Tracking Impedance control for robot manipulators with an unknown environment : Theory, Simulation , and Experiment", Journal of Robotics Research, 29(9):765-774

[8] Woon Kyu Lee and Seul Jung(2006), "Implementation of Embedded Time-delayed Controller Using FPGA for Controlling Humanoid Robot Arm", International Symposium on Humanized Systems, pp. 46-50