Scientific Research Publishing

Journal of Computer and Communications, 2017, 5, 8-18
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Visual Feedback Balance Control of a Robot Manipulator and Ball-Beam System

Ching-Long Shih*, Jung-Hsien Hsu, Chi-Jen Chang

Department of Electrical Engineering, National Taiwan University of Science and Technology, Taiwan
Email: *shihcl@mail.ntust.edu.tw, M10307427@mail.ntust.edu.tw, M10307429@mail.ntust.edu.tw

How to cite this paper: Shih, C.-L., Hsu, J.-H. and Chang, C.-J. (2017) Visual Feedback Balance Control of a Robot Manipulator and Ball-Beam System. *Journal of Computer and Communications*, **5**, 8-18.
https://doi.org/10.4236/jcc.2017.59002

Received: May 21, 2017
Accepted: June 30, 2017
Published: July 3, 2017

Abstract

In this paper, we present a vision guided robotic ball-beam balancing control system, consisting of a robot manipulator (actuator), a ball-beam system (plant) and a machine vision system (feedback). The machine vision system feedbacks real-time beam angle and ball position data at a speed of 50 frames per second. Based on feedback data, the end-effector of a robot manipulator is driven to control the ball position by maneuvering of the inclination angle of the ball-beam system. The overall control system is implemented with two FPGA chips, one for machine vision processing, and one for robot joints servo PID controllers as well as ball position PD controller. Experiments are performed on a 5-axes robot manipulator to validate the proposed ball beam balancing control system.

Keywords

Ball-Beam System, Robot Manipulator, Vision Feedback Control, PID/PD Control, FPGA

1. Introduction

Machine vision system can be applied in real-time to precisely measure visual properties such as color, length, angle, position, orientation, etc. One advantage of machine vision system is the ability to perform contact-free measurement, which is important especially when contact is difficult. However, one of the difficulties in designing a machine vision system is how to be robust to noises present in an image, to scene disturbances caused by unwanted background and foreground objects, as well as to different illumination conditions. Various machine vision approaches had been applied to visual servo control [1]. Visual servo system must be capable of tracking image features in a sequence of images. Visual object tracking is used to identify the trajectory of moving objects in video frame sequences. Object tracking involves intensive computation in order to extract real-time information from high-volume video data with low power consumption. Real-time image processing tasks require not only high computing power but also high data bandwidth. Thus, prototypical FPGA based real-time image processing systems for computer vision and visual servo applications have been proposed [2] [3] [4]. An application can usually be divided into a pre-processing part located on a dedicated subsystem and a high-level part located on a host system. For instance, the pre-processing system is built on FPGA chips and the high-level part for robot vision applications runs on a PCI-based PC system [2]. In particular, an FPGA implementation of a multi-microrobot visual tracking system using CMOS vision sensors had been shown to achieve a 0.01 mm accuracy [3]. Their system was capable of tracking a single robot at a frequency of over 200 Hz. The motivation of this work is to develop a real-time machine vision feedback based robot motion ball-beam balance control system on FPGA chips.

Ball-and-beam control systems are well-known apparatus designed for demonstrating position control capabilities of unstable open-loop systems. Many nonlinear/linear control methods as well as system stability analysis of ball-beam systems have been presented in literatures [5] [6] [7]. In particular, for a nonlinear ball-beam system model, a simple PD controller with asymptotic stability has been theoretically proven and experimentally tested [7]. However, accurately determining the beam angle and ball position in real-time is an important task. There are many approaches to applying beam inclination and ball position sensors; visual feedback is a common solution [8] [9] [10]. However, there are only few works for applying visual feedback to both beam inclination and ball position measurements. Moreover, there are also few works on the integration of a full-scale robot manipulator and vision feedback system in a ball-beam or a ball-plate balancing control system.

Other related works include ball-beam balance systems with feedbacks from a tablet [11] and from a smartphone [12], a robotic soccer beam balance system [13], and a ball-and-plate balance system [14]. A ball-beam control system with visual feedback sensor from a smart tablet had been designed in [11]. Additionally, augmented reality was integrated into an interactive touchscreen interface. A ball-beam system with wireless sensors on a smartphone had been demonstrated in [12]. A smartphone's inertial and camera sensors were used to measure the angular orientation/velocity of the beam and translational ball position, and a local feedback linearizing controller was used to stabilize the ball-beam control system. Ryu and Oh had experimented with a soccer-and-beam balance system installed on the top of the end-effector of a redundant manipulator [13]. A force-torque sensor was attached to the end-effector to estimate the ball position through differentiation. Cheng and Tsai had designed a ball-and-plate balance system, which was attached as the end effector and maneuver by a two-DOF robotic wrist [14]. The ball's position was captured by a PC-based video camera system and the orientation of the plate was controlled by a LQR algorithm.

Up to now, we have not observed that an automatic ball-beam balance system by a robot manipulator with more than 3-axes. This paper attempts to design an automatic robotic ball-beam balance system using an integrated machine vision system. The main goal of this machine vision based robot motion control system is to measure the ball position and beam angle in real-time, and to balance the ball at any desired point on the beam through the motion of a 5-axes robotic end-effector. This works also focuses on an entirely FPGA based control implementation with input from a CMOS image sensor.

2. Robotic Ball-Beam Control System

The robotic ball-beam control system setup is shown in Figure 1. The end-effector of a 5-axes robot manipulator is connected to the left end of the beam which can be freely rotated about its center pivot. A blue ball is allowed to roll freely along the beam. Both the ball's position and beam angle are measured by a real-time image processing system with input from a CMOS image sensor at a rate of 50 frame-per-second. The overall control system is implemented on two FPGA development boards, one for real-time image processing and the other one for all required control functions, including ball balance control and end-effector position control.

The ball-beam system is a two-dimensional system on the x-z plane and hence only three axes of the 5-axis robot manipulator, q_2, q_3 and q_4, are under independent-joint set-point PID control, the other two axes q_1 and q_5 are stationary and set to zero angle position. The robot's end-effector is attached to the left-end of the beam, as shown in Figure 2. The ball-beam system itself is a sin-

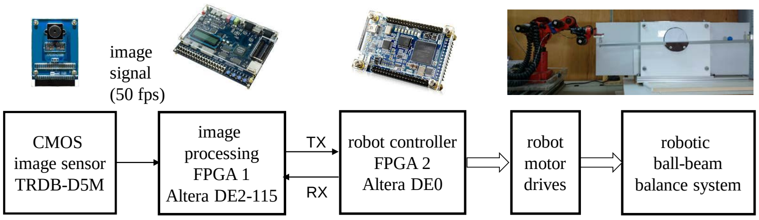


Figure 1. The control system set-up of the vision feedback robotic ball-beam balance system.

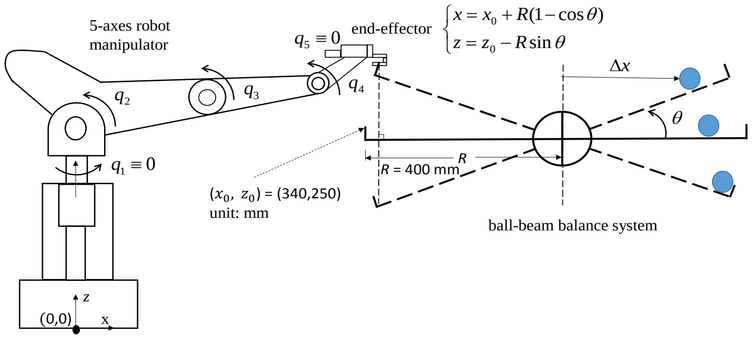


Figure 2. The robot end-effector automatically balances the ball at any desired position on the x-z plane, where (x_0, z_0) is the reference position.

gle-input (beam angle) and single-output (ball position) system.

To have the robot's end-effector automatically drive and balance the ball at any position, a ball position PD controller and independent robot joint PID controllers play major parts in the overall control system as shown in Figure 3. Let Δx and Δx_g be the ball current position and target goal position, respectively. A simple discrete PD control law,

$$\theta_g = k_p e + k_d \Delta e, \tag{1}$$

is applied to guide the robotic end-effector's motion, where $e = \Delta x - \Delta x_g$ is the ball balance position error, and the control sampling time is T_c = 0.02 sec. The robotic end-effector's motion command position and orientation (x_g, z_g, α_g) are then set to

$$\begin{cases} x_g = x_0 + R(1 - \cos\theta_g) \\ z_g = z_0 - R\sin\theta_g \\ \alpha_g = 0 \end{cases}, \tag{2}$$

where R is half of the beam length as shown in Figure 2. Solving the inverse kinematic Equation (2), robot joint goal position commands q_{2g}, q_{3g} and q_{4g} are then obtained. Incremental digital PID control law,

$$\Delta u_i = K_{Pi}\Delta e_i + K_{Di}\Delta^2 e_i + K_{Ii} e_i, \quad e_i = q_{ig} - q_i, \quad i = 2,3,4 \tag{3}$$

are used to independently control the position of each robot joints. The servo control sampling time is T_s = 0.001 seconds.

3. Ball-Beam System Image Processing

Let (c_0, r_0) be the image coordinate of the beam's pivot joint, (c_1, r_1) be the image coordinate of the blue ball, and (c_2, r_2) be the image coordinate of the point where r_2 is the lowest pixel position of the beam along the vertical line $c = c_2$, as shown in Figure 4. A blue ball is used here in order to simplify ball detection and to improve measurement accuracy. Let $\Delta c = c_1 - c_0$ and $\Delta r = r_2 - r_0$ be the image feedback output variables. When beam angle θ is small, we have $\Delta z = L\tan\theta \cong L\theta$, where L is a constant length; therefore, the ball position and beam angle can be expressed as

$$\Delta x = s\Delta c \tag{4}$$

and

$$\theta \cong \frac{\Delta z}{L} = \frac{s\Delta r}{L} = k\Delta r, \tag{5}$$

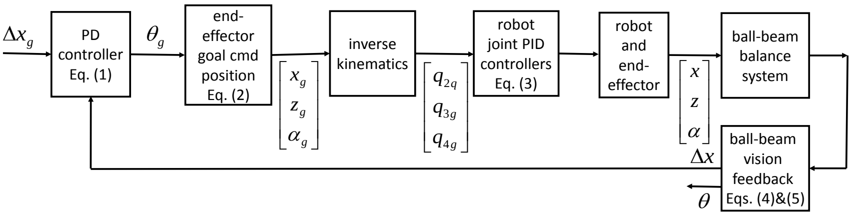


Figure 3. The system block diagram of the robotic ball-beam control system.

where scaling constant $s = 1.5$ mm/pixel and $k = 0.24$ deg/pixel.

The proposed ball-beam system visual feedback is built on the Altera DE2 FPGA development board with input from a CMOS image sensor camera (TRDB-D5M). The image processing module captures real-time images with a 640 × 480 resolution and at 50 fps. Figure 5 shows the functional block diagram of image processing module, in which G is the G-component of the RGB color image. The processing pipeline includes color space transformation, Sobel edge detection, binary threshold, erosion/dilation, and target object position detection.

Real-time image input is fed to the FPGA chip row by row. Up to 5 rows of an image are stored in FIFO line-buffers in a pipelined fashion. The image processing unit is clocked at 96 MHz. The raw image data is captured by a color

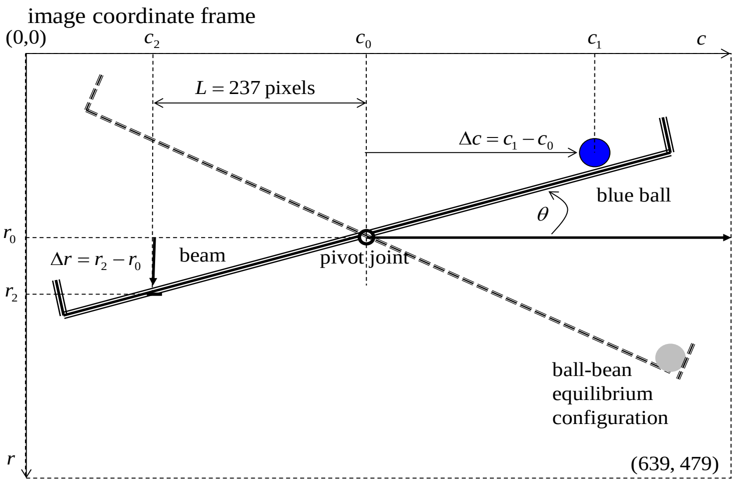


Figure 4. Ball-beam visual sensor system, where L depends on the distance between the camera and the beam.

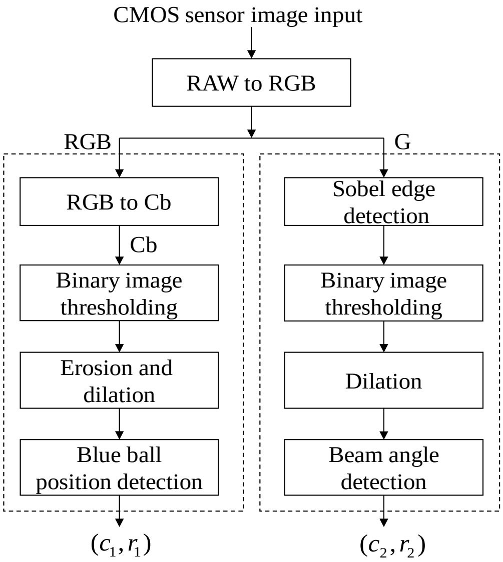


Figure 5. The image processing pipeline.

camera with 1280×960 resolution at 50 fps (frames per second). The raw image is converted to a 640×480 24-bit RGB image (by down sampling of every 2×2 scanned pixels). The center position of the blue ball is obtained as follows. First, RGB to C_b color space transformation is performed : $C_b = -0.169R - 0.331G + 0.5B + 128$, where $0 \le R, G, B \le 255$ and $0 \le C_b \le 255$. Color space transformation is followed by binary segmentation for the C_b component with a threshold value of 130. An erosion operation in a 3×3 window is performed next, followed by a dilation operation in a 3×3 window. True pixels indicate potential ball position. Finally, the center of blue ball (c_1, r_1) is calculated from the center of the true pixels. Beam angle detection is processed in a similar way. Since the G-component of the RGB color image is similar to the gray image. First, Sobel edge detection is applied to the G-component of the original RGB image with a threshold value of 384, followed by a dilation operation in 3×3 window. For all pixels, r_2 is the lowest pixel point of the beam in column c_2 ($c_2 = 71$) such that $r_2 = \max\left\{r : P_{c_2,r} = 1, 180 \le r \le 340\right\}$. Figure 6 shows the ball-beam visual feedback processing experimental results.

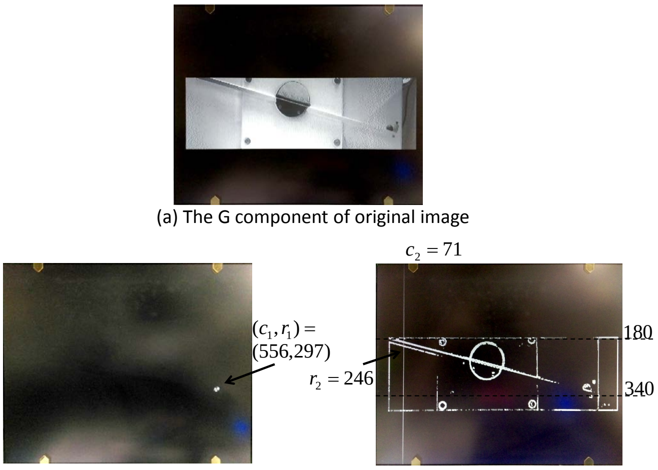
4. Experimental Results

4.1. Robot Joints Set-Point Control Experiments

The robot manipulator is running in set-point position control mode, in which each joint is under independent joint PID servo control. The digital PID control is implemented in relative form

$$\begin{aligned} u_i = u_i(n-1) + K_{Ii} e_i(n) + K_{Pi}\left(e_i(n) - e_i(n-1)\right) \\ + K_{Di}\left(e_i(n) - 2e_i(n-1) + e_i(n-2)\right) \end{aligned}$$

$$u_i(n) = sat(u_i) = \begin{cases} u_{\max} & u > u_{\max} \\ u_i & u_{\min} \le u \le u_{\max} \\ u_{\min} & u < u_{\min} \end{cases},$$



(a) The G component of original image

(b) binary image for ball position detection

(c) binary image for beam angle detection

Figure 6. Experimental results of the ball-beam visual feedback processing.

where $e_i(n) = q_{ig}(n) - q_i(n)$, $i = 2,3,4$, is the current position error of joint *i*. PID gains for joint 2, 3 and 4 are $K_{P2} = 60$, $K_{I2} = 1$, $K_{D2} = 10$ (q_2-axis), $K_{P3} = 150$, $K_{I3} = 1$, $K_{D3} = 10$ (q_3-axis) and $K_{P4} = 95$, $K_{I4} = 1$, $K_{D4} = 2$ (q_4-axis). **Figure 7** shows robot manipulator set-point control experiments for joints q_2, q_3 and q_4.

4.2. Ball-Beam PD Controller Tuning Experiment

To estimate the system transfer function and design the controller of the ball-beam system, a manual balance control test is performed. A human attaches his hand to the left-end of the beam and balances the ball at the center pivot joint position. **Figure 8** shows the ball position $\Delta x(n)$ and beam angle $\theta(n)$ trajectories sampled at an interval 0.02 seconds. One can apply MATLAB's *ident* system identification function with input/output data as shown in **Figure 7**, the system transfer function of the ball-beam system is estimated to be

$$\hat{G}(s) = \frac{-(7.511s + 10.57)}{s^3 + 1.138s^2 + 0.694s}. \tag{6}$$

Having the ball-beam system transfer function $\hat{G}(s)$, one can then use MATLAB's *pidtool* as shown in **Figure 9** to tune a continuous-time PD controller based on Eq. (1), and obtains

$$C_1(s) = -0.0595s - 0.577. \tag{7}$$

Then a digital PD controller is implemented as shown below

$$u = k_p e(n) + k_d\left(e(n) - e(n-1)\right),$$

$$\theta_g(n) = sat(u) = \begin{cases} \theta_{\max} & u > \theta_{\max} \\ u & \theta_{\min} \le u \le \theta_{\max} \\ \theta_{\min} & u < \theta_{\min} \end{cases}.$$

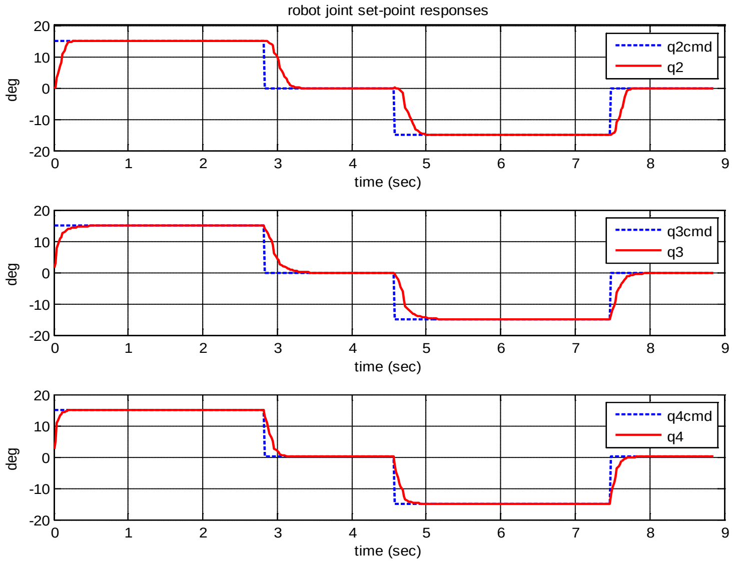


Figure 7. Set-point control experimental results for robot joints q_2, q_3 and q_4.

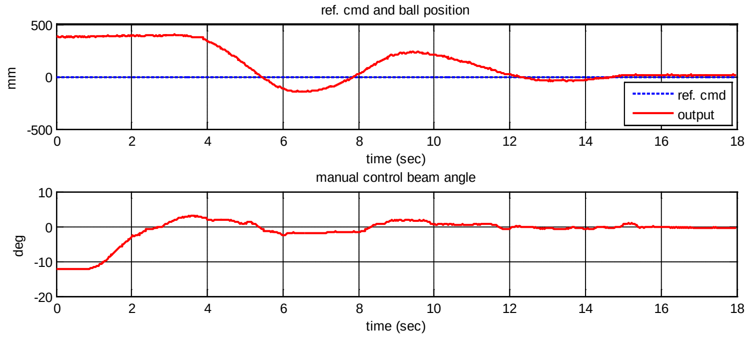
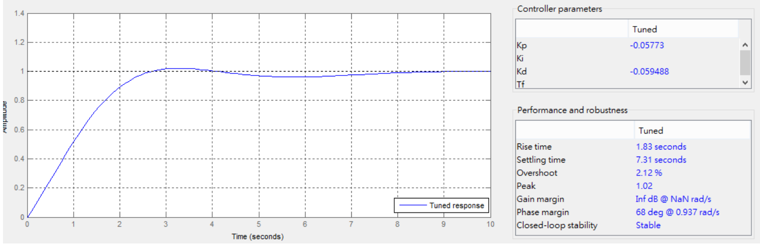


Figure 8. Manual ball-beam balance control test trajectories.



Controller parameters

	Tuned
Kp	-0.05773
Ki	
Kd	-0.059488
Tf	

Performance and robustness

	Tuned
Rise time	1.83 seconds
Settling time	7.31 seconds
Overshoot	2.12 %
Peak	1.02
Gain margin	Inf dB @ NaN rad/s
Phase margin	68 deg @ 0.937 rad/s
Closed-loop stability	Stable

Figure 9. Tuning the ball-beam PD controller $C_1(s)$ with MATLAB's pidtool.

where $e(n) = \Delta x(n) - \Delta x_g(n)$, $k_p = 0.0577$ and $k_d = \dfrac{0.0595}{T_s} = 2.975.$

4.3. Robotic Ball-Beam Balance Control Experiments

Several robotic ball-beam balance control experiments are performed for different ball target positions: center, right-side, left-side and multi-positions, as shown in **Figures 10-13**, respectively. The control performance for the center target position as shown in **Figure 10** can be summarized as follows: rise time is 4.0 sec., settling time 8.0 sec., overshot 25% and steady-state error 2%. When compared with **Figure 8**, one can see that the control performance of PD controller is much better than that of a human in terms of less rise/settle, smaller overshoot, and less steady-state error. The control performance for the right-side target position as shown in **Figure 11** can be summarized as follows: rise time 4.0 sec., settling time 7.0 sec., overshot 2% and steady-state error 1%. The control performance for the left-side target position as shown in **Figure 12** can be summarized as follows: rise time 4.0 sec., settling time 8.0 sec., overshot 20% and steady-state error 1%. In summary, the robotic ball-beam balance control has less overshoot especially for smaller displacement and when the ball is rolling from left to right as shown from **Figure 13**.

5. Conclusion

In this work, we have implemented a visual feedback based robotic motion control system on two FPGA chips with efficient parallel architecture. We have experimentally verified that a robotic motion control system with an integrated

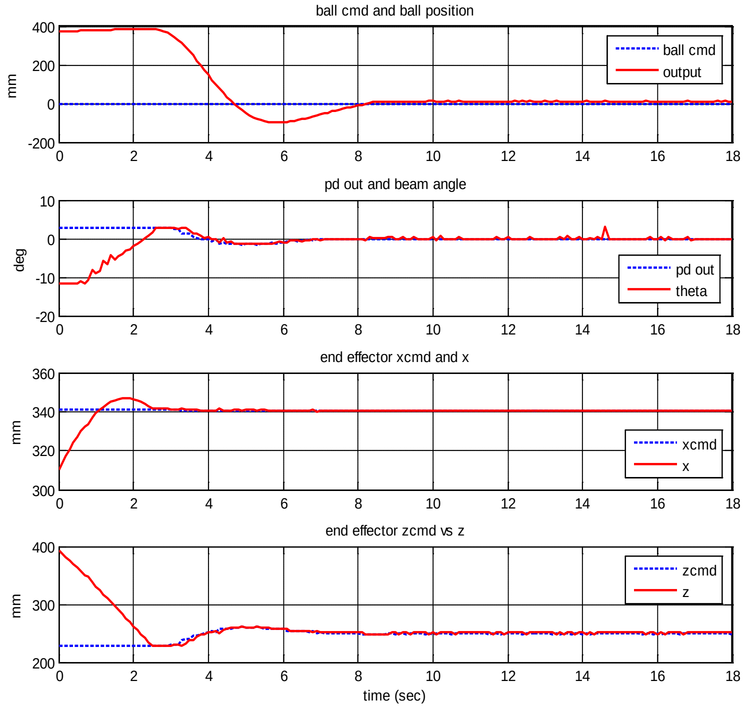


Figure 10. Ball-beam balance control experiment 1: center position.

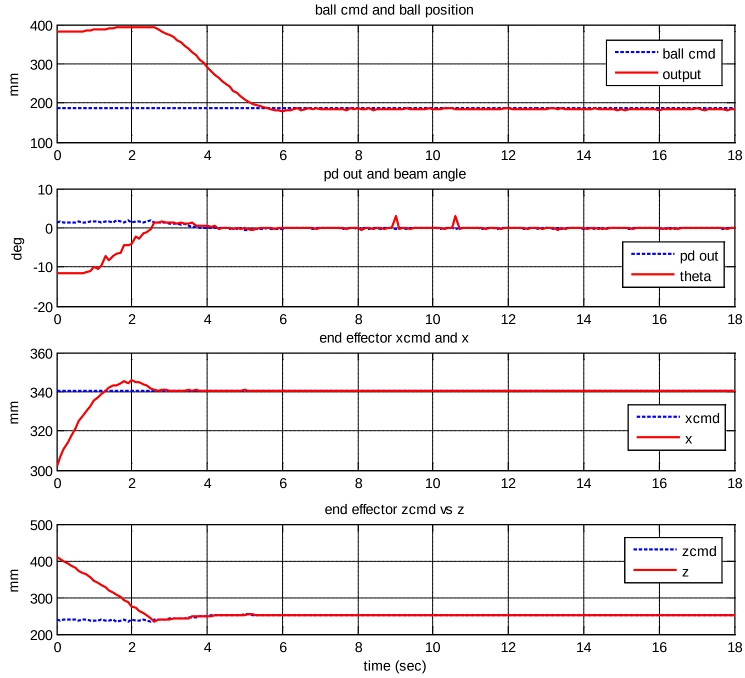


Figure 11. Ball-beam balance control experiment 2: right-side position.

visual feedback system can automatically balance a ball-beam system with the end-effector of a 5-axis manipulator. This machine vision based robotic ball-beam set-point control system is able to balance the ball at any desired position on the beam efficiently in real-time. Although the proposed system is almost as good as a naive user manually balancing the ball to stay at any desired position,

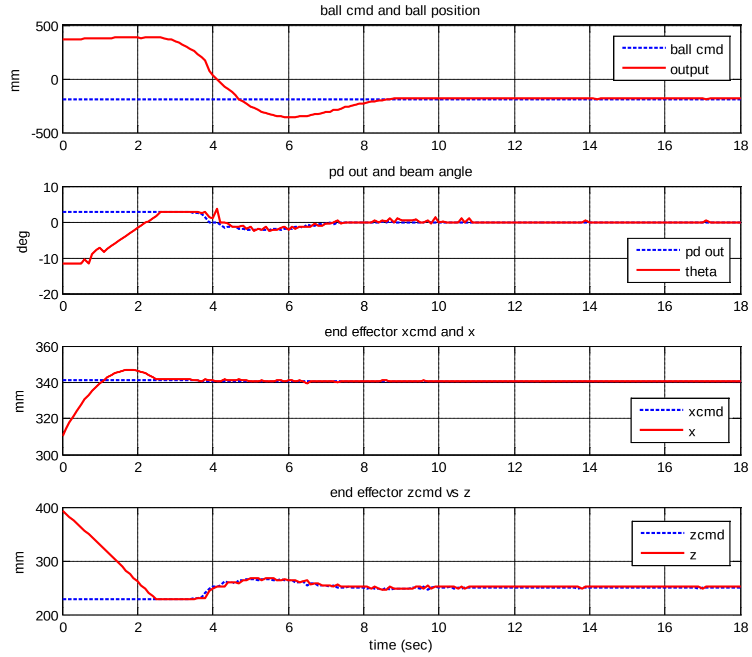


Figure 12. Ball-beam balance control experiment 3: left-side position.

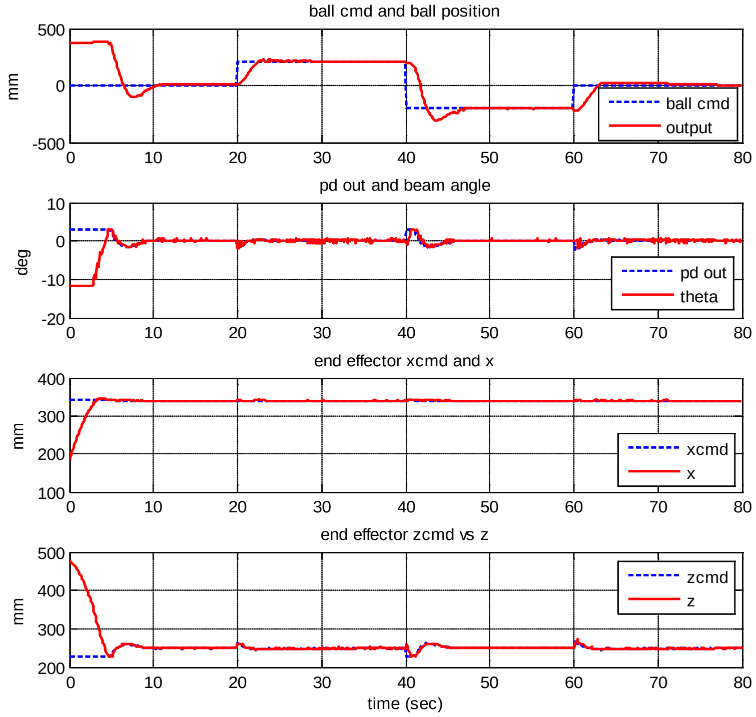


Figure 13. Ball-beam balance control experiment 3: left-side position.

the limitations of the current system are lower sampling time in visual control feedback and smaller angle range of the bean angle.

Acknowledgements

This work is supported by Taiwan Ministry of Science and Technology Grant

MOST 105-2221-E-011-047.

References

[1] Croke, P. (2011) Robotics, Vision and Control. Springer-Verlag, Berlin Heidelberg. https://doi.org/10.1007/978-3-642-20144-8

[2] Jorg, S., Langwald, J. and Nickl, M. (2004) FPGA Based Real-Time Visual Sevoing. *Proceedings of the* 17*th International Conference on Pattern Recognition*, Cambridge, 26-26 August 2004, 749-753. https://doi.org/10.1109/ICPR.2004.1334300

[3] Diederichs, C. (2011) Fast Visual Servoing of Multiple Microrobots Using an FPGA-Based Smart Camera System. *IFAC Proceedings Volumes*, **44**, 14636-14641. https://doi.org/10.3182/20110828-6-it-1002.02465

[4] Garc, G., Jara, C., Pomares, J., Alabda, A., Poggi L. and Torres, F. (2014) A Survey on FPGA-Based Sensor Systems: Towards Intelligent and Reconfigurable Low-Power Sensors for Computer Vision, Control and Signal Processing. *Sensors*, **14**, 6247-6278. https://doi.org/10.3390/s140406247

[5] Hauser, J., Sastry, S. and Kokotovic, P. (1992) Nonlinear Control via Approximate Input-Output Linearization: The Ball and Beam Example. *IEEE Transactions on Automatic Control*, **37**, 392-398. https://doi.org/10.1109/9.119645

[6] Hirschorn, R. (2002) Incremental Sliding Mode Control of the Ball and Beam. *IEEE Transactions on Automatic Control*, **47**, 1696-1700. https://doi.org/10.1109/TAC.2002.803538

[7] Yu, W. and Ortiz, F. (2005) Stability Analysis of PD Regulation for Ball and Beam System. *Proceedings of the* 2005 *IEEE Conference on Control Applications*, Toronto, Canada, 517-522.

[8] Petrovic, I., Brezak, M. and Cupec, R. (2002) Machine Vision Based Control of the Ball and Beam. *IEEE* 7*th International Workshop on Advanced Motion Control*, Maribor, 3-5 July 2002, 573-577. https://doi.org/10.1109/amc.2002.1026984

[9] Ho, C. and Shih, C. (2008) Machine Vision Based Tracking Control of Ball Beam System. *Key Engineering Materials*, **381-382**, 301-304. https://doi.org/10.4028/www.scientific.net/KEM.381-382.301

[10] Xiao, H.L., Yong, X.L. and Hai, Y.L. (2011) Design of Ball and Beam Control System Based on Machine Vision. *Applied Mechanics and Materials*, **71-78**, 4219-4225.

[11] Frank, J., Gomez, J. and Vikram, K.V. (2015) Using Tablets in the Vision-Based Control of Ball and Beam Test-Bed. *Proceedings of* 12*th International Conference on Informatics in Control, Automation and Robotics*, Alsace, 21-23 July 2015, 92-102. https://doi.org/10.5220/0005544600920102

[12] Brill, A., Frank, J. and Kapola, V. (2016) Using Inertial and Visual Sensing from a Mounted Smartphone to Stabilize a Ball and Beam Test-Bed. *Proceedings of American Control Conference*, Boston, 6-8 July 2016, 1335-1340. https://doi.org/10.1109/acc.2016.7525103

[13] Ryu, K. and Oh, Y. (2011) Balance Control Of Ball-Beam System Using Redundant Manipulator. *Proceedings of the* 2011 *IEEE International Conference on Mechatronics*, Beijing, 7-10 August 2011, 403-408. https://doi.org/10.1109/ICMECH.2011.5971319

[14] Cheng, C. and Tsai, C. (2016) Visual Servo Control for Balancing a Ball-Plate System. *International Journal of Mechanical Engineering and Robotics Research*, **5**, 28-32.