INTECH
open science | open minds

OPEN ACCESS
ARTICLE

International Journal of Advanced Robotic Systems

Visual Servoing Tracking Control of a Ball and Plate System: Design, Implementation and Experimental Validation

Regular Paper

Ming-Tzu Ho[1,*], Yusie Rizal[1] and Li-Ming Chu[1]

1 Engineering Science Department, National Cheng Kung University, Tainan, Taiwan
* Corresponding author E-mail: bruceho@mail.ncku.edu.tw

Received 28 Jun 2012; Accepted 9 Apr 2013

DOI: 10.5772/56525

Abstract This paper presents the design, implementation and validation of real-time visual servoing tracking control for a ball and plate system. The position of the ball is measured with a machine vision system. The image processing algorithms of the machine vision system are pipelined and implemented on a field programmable gate array (FPGA) device to meet real-time constraints. A detailed dynamic model of the system is derived for the simulation study.By neglecting the high-order coupling terms, the ball and plate system model is simplified into two decoupled ball and beam systems, and an approximate input-output feedback linearization approach is then used to design the controller for trajectory tracking. The designed control law is implemented on a digital signal processor (DSP). The validity of the performance of the developed control system is investigated through simulation and experimental studies. Experimental results show that the designed system functions well with reasonable agreement with simulations.

Keywords Visual Servoing, Ball And Plate System, Feedback Linearization

1. Introduction

A ball and plate system is a common experimental setup in control laboratories. It consists of a horizontal plate, which is tilted along each of two horizontal axes such that a ball can be controlled to roll to any position on the plate. It serves to visually demonstrate and reinforce the underlying principles of nonlinear dynamics and control theory. Moreover, because of its inherent nonlinearity, instability, and underactuation, this system is widely used as a test bed for verifying the performance and effectiveness of new control algorithms or technology. Several aspects of mechatronic design, implementation, and control of a ball and plate system have been investigated in the literature [1-10]. In existing studies, the control design has usually been based on the

simplified model of the ball and plate system, and derivation of a detailed dynamic model has received far less attention. Based on the model linearized with respect to the equilibrium, [1] designed a PID controller and lead controller to stabilize the system. In [3, 7], a nonlinear simplified model was given, and a sliding mode control was used to achieve trajectory tracking. Based on a linear simplified model, [10] designed a PID controller and state observer feedback controller to achieved set-point regulation and trajectory tracking.

For a ball and plate system, sensors for sensing the position of the ball can be divided into two main types: the touch screen [1, 10] and the overhead camera [2-6, 8, 9]. Both these sensing devices suffer from certain shortcomings. The ball can easily lose contact with the touch screen plate during the motion, and this causes discontinuity in the position measurement. Since the sampling rate of a common camera (30 Hz) is limited, use of a common camera for sensing the position of the ball imposes real-time constraints on the system. Visual servoing [11] is a control framework that incorporates visual information in feedback control loops. Real-time image data acquisition and processing is a critical issue in visual servoing applications. A high frame rate and low processing latency are essential since the visual servoing system must make quick decisions based on the information extracted from a scene. Real-time image processing requires a high pixel processing rate, massive and parallel computation, and efficient hardware utilization. General-purpose processors cannot always provide enough computational power to fulfil real-time requirements due to their sequential nature. Due to their inherent architectural parallelism and configurable flexibility, field programmable gate arrays (FPGAs) have become increasingly popular implementation platforms for real-time image processing. In [12], an FPGA/DSP (digital signal processor) architecture was proposed for real-time image processing. This architecture was designed to deal with parallel/pipelined procedures to handle multiple input images. A flexible FPGA-based systolic architecture for real-time window-based image processing was proposed in [13]. Its computational core was based on a configurable two-dimensional systolic array of processing elements. In [14, 15], FPGAs with a pipelined architecture were used to implement optical flow estimation from image sequences in real time.

The idea behind feedback linearization [16-18] is to find a diffeomorphism and a state feedback control law that transform the nonlinear system into a linear time-invariant system. The design is then carried out on the linear system using linear control design techniques. Both full-state feedback linearization and partial-state feedback linearization can be considered. In full-state feedback linearization, also called input-state feedback linearization, the state equations are fully linearized. In partial-state feedback linearization, also called input-output feedback linearization, the input-output map of the system is linearized, while the map from the input to the state can be only partially linearized. However, the applicability of feedback linearization is limited to special classes of nonlinear systems satisfying the constraints of controllability, involutivity, and the existence of a relative degree or minimum phase property. One approach to relaxing one or more of these applicability constraints is to use the techniques of approximate feedback linearization. Approximate feedback linearization was first proposed in [19], where necessary and sufficient conditions for approximate full-state feedback linearization of nonlinear systems are given. This method is intended to find a nonlinear coordinate change and state feedback such that a linear approximation is obtained for an equilibrium point accurate to second or higher order. In [20], it was shown that one can approximately input-output linearize a nonlinear system by neglecting the nonlinearities responsible for the failure of the involutivity conditions. An approximate tracking controller was then designed for a ball and beam system.

This paper presents the design, implementation, and validation of tracking control of a ball and plate system using visual feedback. The position of the ball is measured with a machine vision system that uses an image sensor with a frame rate of 150 frames per second to guarantee a certain level of real-time performance. The pipelined image processing of the machine vision system is implemented on an FPGA device to meet real-time constraints. The detailed dynamic model of the ball and plate system is derived. This detailed dynamic model is adopted to build a reliable and accurate simulation model for the purpose of validating the closed-loop performance of the control system, once a controller design is available. For control design purposes, the detailed dynamic model is far too complex. By neglecting the high-order coupling terms, the ball and plate system is simplified into two decoupled ball and beam systems, and an approximate input-output feedback linearization approach [20] is then used to design a controller for trajectory tracking. A DSP is used for the implementation of the control algorithm and complex arithmetic operations. The effectiveness of the controller and machine vision system is validated through simulation and experimental results. Some studies [8, 9] have reported the control of a vision-based ball and plate system via an FPGA implementation. The processing frame rate of the machine vision system developed in this paper (150 frames/sec) is much higher than that of [8] and [9] (16 frames/sec).Unlike the present work, the experimental results given in [8] and [9] are the results of set-point regulation, which is less challenging.

The rest of this paper is organized as follows. In Section 2, a brief description of the experimental setup is given. Section 3 describes the relationship between the world coordinates and the image coordinates. The image processing algorithms for determining the position of the ball from the image data are introduced, and the implementation and architecture of the image processing algorithms on an FPGA device are presented. In Section 4, the dynamic model of a ball and plate system is derived. In Section 5, a review of approximate input-output feedback linearization is presented. Based on approximate input-output feedback linearization, the design of a controller for trajectory tracking is given. In Section 6, the simulation and experimental results are given. Finally, Section 7 contains some concluding remarks.

2. Experimental Apparatus

The experimental setup of a ball and plate system is shown in Figure 1, and the schematic overview of the experimental setup is shown in Figure 2. The system consists of two dc motors with gear heads. Each motor shaft is fixed to an L-shaped linkage that is attached to the plate with a universal joint. This L-shaped linkage consists of two linkages connected with a bearing joint. The plate is pivoted at its centre with a universal joint. When the L-shaped linkage is driven by the motor to move up and down, the plate inclines and changes its rotational angles; thus the ball can roll on the plate. This actuation mechanism is similar to the one designed and implemented in [1]. An optical encoder with a resolution of 1250 pulses/rev is attached to each motor shaft to measure the angular position of the motor. The position of the ball is measured with an overhead machine vision system, which consists of a CMOS camera sensor, static random access memory (SRAM), and an FPGA board. The camera is a Micron CMOS image sensor, MT9M011, which can capture 352×288 pixels quantized to 1024 RGB-colour-levels at 160 frames per second. In this study, the image sensor is set to operate in the grey-level mode at 150 frames per second. The SRAM (13M bits) acts as a frame buffer, which is used for storing an image during camera calibration. An FPGA board is used to implement the image processing algorithms written in very-high-speed integrated circuit hardware description language (VHDL). The FPGA board is an Altera Cyclone EP1C12Q240C8 FPGA. The controller is implemented on a DSP board, which is the Texas Instruments TMS320F240 digital signal processor (150-MHz/32-bit) equipped with two 12-bit D/A converters and two quadrature encoder pulse (QEP) units. The QEP unit provides the angular position and rotational direction of the motor from the quadrature encoder signals. The velocity of the ball and the angular velocity of the motors are estimated from the displacement travelled per unit time, and then the estimated velocities are passed through digital low-pass filters to attenuate the high-frequency noise. Voltage signals are generated according to the designed control law and supplied to the power amplifiers that drive the dc motors. The power amplifiers are operational amplifiers that can operate at up to ±20 V and 2.75 A.

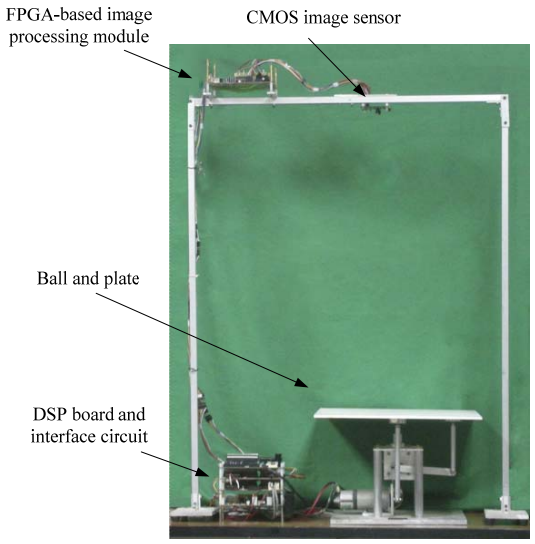

Figure 1. Experimental setup of the ball and plate system.

3. Visual Measurement and Image Processing Algorithm

A machine vision system is used to determine the displacement of the ball on the plate. A black ball and white plate are used to simplify visual sensing and to increase the accuracy of visual measurements. Determination of the position of the ball is based on the perspective pinhole camera model [21-22], which is given by

$$\lambda \mathrm{p} = \mathrm{K_I}[\mathrm{R_E} \mid \mathrm{t}]\mathrm{P_O}, \tag{1}$$

where $\mathrm{p} = [\mathrm{x_{im}}, \mathrm{y_{im}}, 1]^{\mathrm{T}}$ represents the two-dimensional homogeneous coordinates of the image point in the image coordinate system; $\mathrm{P_O} = [\mathrm{X}, \mathrm{Y}, \mathrm{Z}, 1]^{\mathrm{T}}$ represents the three-dimensional homogeneous coordinates of the object point in the world coordinate system, and λ is a scalar factor. The 3×3 matrix $\mathrm{R_E}$ and three-dimensional vector t are extrinsic camera parameters that describe the rotation and translation between the world frame and the camera frame, respectively. $\mathrm{K_I}$ is the intrinsic camera parameter matrix given by

$$\mathrm{K_I} = \begin{bmatrix} \mathrm{f_{sx}} & \mathrm{f_{s\theta}} & \mathrm{o_x} \\ 0 & \mathrm{f_{sy}} & \mathrm{o_y} \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

Here, (o_x, o_y) is the principal point on the image coordinate system in pixels; f_{sx} is the size of the unit length in horizontal pixels; f_{sy} is the size of the unit length in vertical pixels, and $f_{s\theta}$ is the skew of the pixels. The world frame $oXYZ$ is chosen with the Z-axis pointing

vertically upwards and the origin o located at the centre of the plate. The camera is mounted right above the plate with its optical axis passing through the origin of the world frame. The camera frame is denoted by $o_c x_{cm} y_{cm} z_{cm}$. The x_{cm}-axis of the camera frame is chosen to be parallel and aligned to the X-axis of the world frame. Thus, the y_{cm}-axis points vertically downwards. The extrinsic camera parameters are then given by:

$$[R_E \mid t] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & d_c \end{bmatrix}, \quad (3)$$

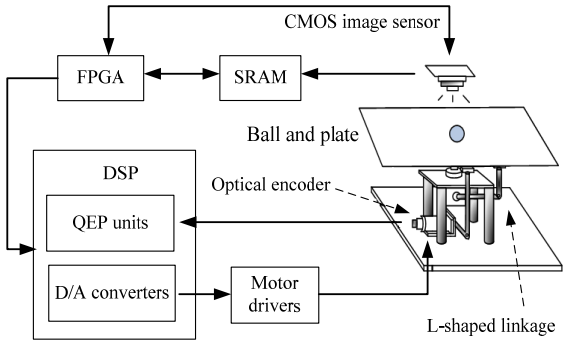

Figure 2. Schematic diagram of the experimental setup.

Parameter	Value
λ	0.9 (m)
f_{sx}	153440 (pixels/m)
f_{sy}	-146050.47 (pixels/m)
$f_{s\theta}$	-4768.43 (pixels/m)
o_x	182.25 (pixels)
o_y	154.27 (pixels)

Table 1. Intrinsic camera parameters.

where $d_c = 0.91$ (m) is the distance from the origin of the world frame to the origin of the camera frame. The intrinsic camera parameters are obtained using the camera calibration procedure proposed in [21]. In this study, the parameter $\lambda = d_c - R$ is obtained, where R is the radius of the ball. The intrinsic camera parameters are given in Table 1.

To determine the position of the ball on the image plane, the captured images are processed using a series of image processing algorithms. A sample of a raw image captured by the camera is shown in Figure 3.

The image processing algorithm steps [23] are described below:

Step 1. Background subtraction

Background subtraction is an approach to differentiate foreground objects from their background. This step is used here to remove brightness fluctuations due to uneven background illumination. Background subtraction was applied to the image in Figure 3 and the resulting image is shown in Figure 4.

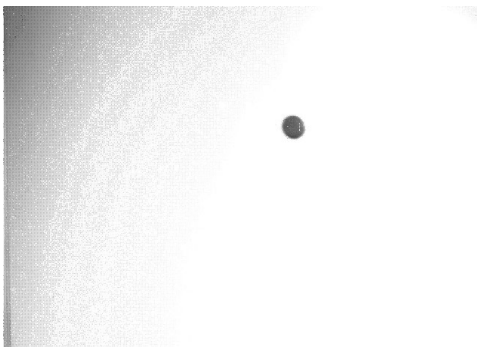

Figure 3. Image captured by the camera.

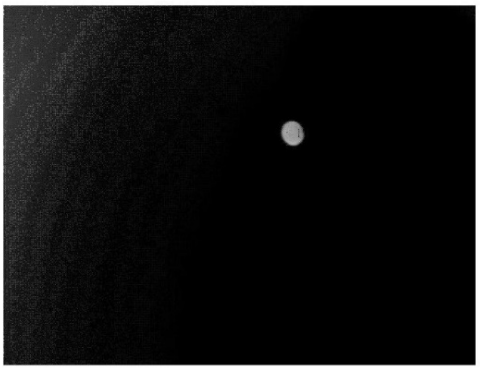

Figure 4. Background subtraction applied to the image in Fig. 3.

Step 2.Thresholding

Thresholding is a simple and efficient method applied to grey-level intensity images to differentiate between the object and the background. Moreover, thresholding converts a grey-scale image into a bi-level image which contains all of the essential information concerning the shape, position and number of objects. This process is described as follows:

$$f(x_{im}, y_{im}) = \begin{cases} 255 & \text{if } f(x_{im}, y_{im}) \geq T_{th} \\ 0 & \text{if } f(x_{im}, y_{im}) < T_{th} \end{cases}, \quad (4)$$

where $f(x_{im},y_{im})$ is the grey-scale level of the pixel at point (x_{im},y_{im}) in the image coordinate system, and T_{th} is the threshold value. Applying thresholding with a threshold value of 100 to the image in Figure 4 produces the image in Figure 5. The threshold value is determined experimentally to obtain the best system performance.

Step 3. Median filter

As seen in Figure 5, there is noise in the image. Further filtering processing is needed to remove noise. Noise removal is performed by a median filter, which requires the use of a 3×3window to slide along the image, and the median intensity value of the pixels within the window

becomes the output intensity of the centre pixel. Applying the median filtering process to the image in Figure 5 produces the image in Figure 6.

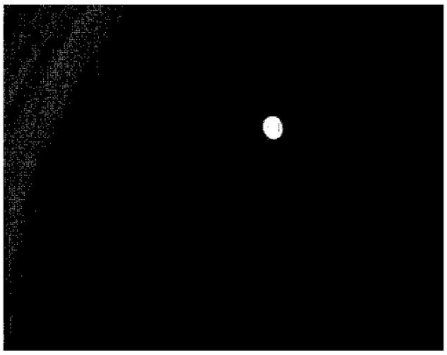

Figure 5. Thresholding applied to the image in Fig. 4.

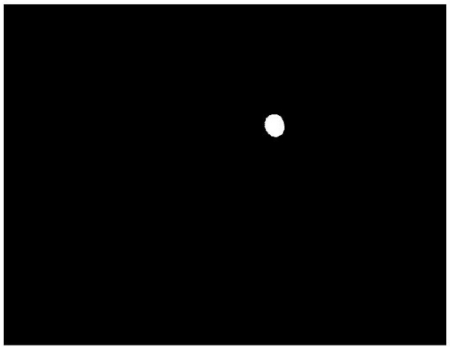

Figure 6. Median filtering process applied to the image in Fig. 5.

Step 4. Centroid determination

As shown in Figure 6, the white pixels correspond to the ball, while the plate is shown by the black pixels. Since the ball is symmetrical, the centroid of the image of the ball is taken as its location on the plate. The centroid (x_c,y_c) is given by

$$x_c = \frac{\sum_{x_{im}}\sum_{y_{im}} x_{im} \cdot f(x_{im}, y_{im})}{\sum_{x_{im}}\sum_{y_{im}} f(x_{im}, y_{im})}, \quad (5)$$

$$y_c = \frac{\sum_{x_{im}}\sum_{y_{im}} y_{im} \cdot f(x_{im}, y_{im})}{\sum_{x_{im}}\sum_{y_{im}} f(x_{im}, y_{im})}. \quad (6)$$

In order to process the image data in real time, the image processing algorithms are implemented on an FPGA using VHDL. The FPGA technology is chosen because it is able to provide massive amounts of reprogrammable hardware acceleration and parallel/pipelined processing ability. In this study, the image processing is divided into the modules as shown in Figure 7.

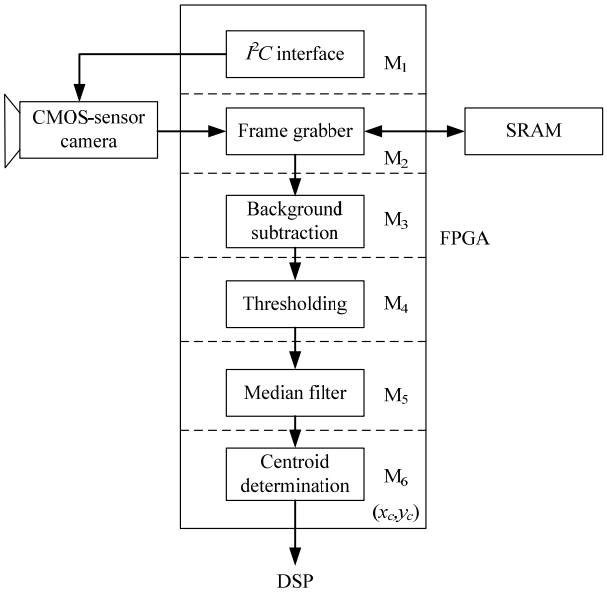

Figure 7. FPGA architecture pipelined.

The functions of the modules are:

- **M1**:The Inter-Integrated Circuit(I^2C) serial interface is used to configure the frame rate, gain control, exposure time, and integration time of the camera.
- **M2**:The frame grabber module acquires the pixels at a rate of 25 Mbytes/sec from the camera. The image data can be directly passed to the next module or stored in external SRAM for camera calibration.
- **M3**: This module is designed to realize background subtraction. The grey-scale level of the background image pixel is subtracted from the incoming pixel.
- **M4**: This module applies thresholding to the output image of the background subtraction stage. Since after background subtraction the ball has higher intensity than the plate in the image, the thresholding method is used to differentiate the ball from the background.
- **M5**: Median filtering is performed on the output image of the thresholding stage to reduce noise.
- **M6**: The centroid of the ball (Equations (5) and (6)) is calculated in this stage.

There is no iterative process in this design. Thus, this processing can be fully pipelined in hardware implementation to improve efficiency and provide a high data throughput rate. The design is implemented on Altera Cyclone EP1C12Q240C8 FPGA. The FPGA configuration data is stored in an electrically-erasable programmable read-only memory (EEPROM) and can be re-programmed by the host computer. A programmable clock generator is used to generate three global clocks for system clocks and the pixel clock. The I^2C serial interface is set to operate at 62.5 kHz; the operation frequency for carrying out image processing is 100 MHz, and the pixel clock is also 100 MHz. In order for the DSP board to

communicate with the FPGA, a 16-bit bus connected to the DSP board data bus is used for data exchange. The read/write commands are specified with different addresses. The proposed design can process 150 grey-level images per second with a resolution of 352×288 pixels.

4. System Modelling

In this section, we derive the mathematical model of the ball and plate system by using the Euler-Lagrange method [24]. Figure 8 illustrates the basic features of the system. Assuming that the ball rolls on the plate without slipping, the ball is always in contact with the plate; there is no rotational motion of the ball with respect to its vertical axis, and all friction forces and torques are neglected. There are two coordinate frames used, as shown in Figure 8.

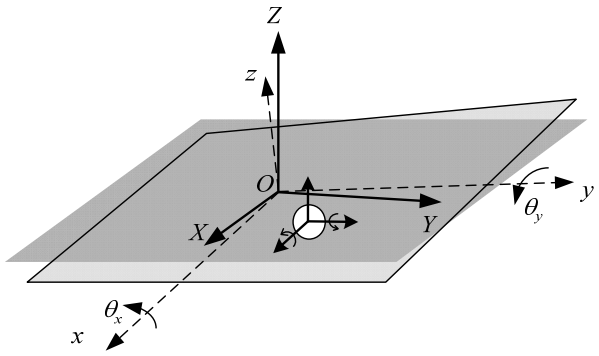

Figure 8. Coordinate frames for the ball and plate system.

The space-fixed coordinate system (world frame) *oXYZ* has the *Z*-axis pointing vertically upwards and the origin located at the centre of the plate. Unit vectors along the *oXYZ*-coordinate system are denoted by $\hat{\mathrm{I}}$, $\hat{\mathrm{J}}$, and $\hat{\mathrm{K}}$. The coordinate system fixed on the plate with the *z*-axis perpendicular to the plate and the origin located at the centre of the plate is denoted by *oxyz*. Unit vectors along the *oxyz*-coordinate system are denoted by $\hat{\mathrm{i}}$, $\hat{\mathrm{j}}$, and $\hat{\mathrm{k}}$. The *oxyz*-coordinate system is, with respect to the *oXYZ*-coordinate system, rotated about the *x*-axis by an angle θ_{x} followed by a rotation about the *y*-axis by an angle θ_{y}. The Euler-Lagrangian equation is

$$\frac{\mathrm{d}}{\mathrm{dt}}\left[\frac{\partial \mathrm{L}}{\partial \dot{\mathrm{q}}}\right]-\frac{\partial \mathrm{L}}{\partial \mathrm{q}}=\mathrm{Q}, \tag{7}$$

where *L* is the Lagrangian function, and *Q* and *q* are the generalized forces and generalized coordinates, respectively. The Lagrangian function *L* is defined as

$$\mathrm{L}=\mathrm{T}-\mathrm{V}, \tag{8}$$

where *T* is the kinetic energy, and *V* is the potential energy. For this system, *q* is selected as

$$\mathrm{q}=\begin{bmatrix}\mathrm{x}\\ \mathrm{y}\\ \theta_{\mathrm{x}}\\ \theta_{\mathrm{y}}\end{bmatrix}, \tag{9}$$

Q is given by

$$\mathrm{Q}=\begin{bmatrix}0\\ 0\\ \tau_{\mathrm{x}}\\ \tau_{\mathrm{y}}\end{bmatrix}, \tag{10}$$

where τ_{x} and τ_{y} are the torques exerted on the plate in the *x*-axis and *y*-axis, respectively.

If the inclination of the plate changes, the ball moves on the plate. The position of the ball relative to the plate-fixed coordinate can be mapped into the space-fixed coordinate. The transformation between these coordinates is written as follows:

$$\begin{bmatrix}\mathrm{x}\\ \mathrm{y}\\ \mathrm{z}\end{bmatrix}={}^{\mathrm{P}}\mathrm{R}_{\mathrm{F}}\begin{bmatrix}\mathrm{X}\\ \mathrm{Y}\\ \mathrm{Z}\end{bmatrix}, \tag{11}$$

where ${}^{\mathrm{P}}\mathrm{R}_{\mathrm{F}}$ is the rotation matrix and

$${}^{\mathrm{P}}\mathrm{R}_{\mathrm{F}}=\begin{bmatrix}\cos\theta_{\mathrm{y}} & \sin\theta_{\mathrm{x}}\sin\theta_{\mathrm{y}} & -\cos\theta_{\mathrm{x}}\sin\theta_{\mathrm{y}}\\ 0 & \cos\theta_{\mathrm{x}} & \sin\theta_{\mathrm{x}}\\ \sin\theta_{\mathrm{y}} & -\sin\theta_{\mathrm{x}}\cos\theta_{\mathrm{y}} & \cos\theta_{\mathrm{x}}\cos\theta_{\mathrm{y}}\end{bmatrix}. \tag{12}$$

The position vector of the ball in the oxyz-coordinate system is given by

$$\hat{\mathrm{r}}=\mathrm{x}\hat{\mathrm{i}}+\mathrm{y}\hat{\mathrm{j}}+\mathrm{R}\hat{\mathrm{k}}, \tag{13}$$

where *R* is the radius of the ball. From (11), we have

$$\begin{aligned}\mathrm{X}&=\mathrm{x}\cos\theta_{\mathrm{y}}+\mathrm{R}\sin\theta_{\mathrm{y}},\\ \mathrm{Y}&=\mathrm{x}\sin\theta_{\mathrm{x}}\sin\theta_{\mathrm{y}}+\mathrm{y}\cos\theta_{\mathrm{x}}-\mathrm{R}\sin\theta_{\mathrm{x}}\cos\theta_{\mathrm{y}},\\ \mathrm{Z}&=-\mathrm{x}\cos\theta_{\mathrm{x}}\sin\theta_{\mathrm{y}}+\mathrm{y}\sin\theta_{\mathrm{x}}+\mathrm{R}\cos\theta_{\mathrm{x}}\cos\theta_{\mathrm{y}}.\end{aligned} \tag{14}$$

By taking derivative of (14) with respect to time, the velocity vector of ball in the oXYZ-coordinate system is then given by

$$\hat{\mathrm{v}}=\dot{\mathrm{X}}\hat{\mathrm{I}}+\dot{\mathrm{Y}}\hat{\mathrm{J}}+\dot{\mathrm{Z}}\hat{\mathrm{K}}, \tag{15}$$

where

$$\dot{\mathrm{X}}=\dot{\mathrm{x}}\cos\theta_{\mathrm{y}}-\mathrm{x}\dot{\theta}_{\mathrm{y}}\sin\theta_{\mathrm{y}}+\mathrm{R}\dot{\theta}_{\mathrm{y}}\cos\theta_{\mathrm{y}}, \tag{16}$$

$$\begin{aligned}\dot{\mathrm{Y}}=&\dot{\mathrm{x}}\sin\theta_{\mathrm{x}}\sin\theta_{\mathrm{y}}+\dot{\mathrm{y}}\cos\theta_{\mathrm{x}}-\mathrm{y}\dot{\theta}_{\mathrm{x}}\sin\theta_{\mathrm{x}}\\ &+\mathrm{x}(\dot{\theta}_{\mathrm{x}}\cos\theta_{\mathrm{x}}\sin\theta_{\mathrm{y}}+\dot{\theta}_{\mathrm{y}}\sin\theta_{\mathrm{x}}\cos\theta_{\mathrm{y}})\\ &-\mathrm{R}(\dot{\theta}_{\mathrm{x}}\cos\theta_{\mathrm{x}}\cos\theta_{\mathrm{y}}-\dot{\theta}_{\mathrm{y}}\sin\theta_{\mathrm{x}}\sin\theta_{\mathrm{y}}),\end{aligned} \tag{17}$$

$$\begin{aligned}\dot{Z} &= -\dot{x}\cos\theta_x \sin\theta_y + \dot{y}\sin\theta_x + y\dot{\theta}_x\cos\theta_x \\ &\quad - x(-\dot{\theta}_x \sin\theta_x \sin\theta_y + \dot{\theta}_y \cos\theta_x \cos\theta_y) \\ &\quad + R(-\dot{\theta}_x \sin\theta_x \cos\theta_y - \dot{\theta}_y \cos\theta_x \sin\theta_y).\end{aligned} \tag{18}$$

The angular velocity vector of the plate due to rotation is denoted by

$$\hat{\omega}_p = \omega_{px}\hat{i} + \omega_{py}\hat{j} + \omega_{pz}\hat{k}\,. \tag{19}$$

The relation between the rotation matrix and angular velocity [24] is given as

$${}^{P}R_F({}^{P}\dot{R}_F)^T = \begin{bmatrix} 0 & -\omega_{pz} & \omega_{py} \\ \omega_{pz} & 0 & -\omega_{px} \\ -\omega_{py} & \omega_{px} & 0 \end{bmatrix}. \tag{20}$$

From (20) and (12), it is given that

$$\hat{\omega}_p = \dot{\theta}_x \cos\theta_y \hat{i} + \dot{\theta}_y \hat{j} + \dot{\theta}_x \sin\theta_y \hat{k}. \tag{21}$$

Let $\omega_1\hat{i} + \omega_2\hat{j} + \omega_3\hat{k}$ denote the angular velocity vector of the ball with respect to the *oxyz*-coordinate system. Under the assumption of rolling without slipping and twisting we have the following conditions:

$$\begin{aligned}\omega_1 &= -\frac{\dot{y}}{R}, \\ \omega_2 &= \frac{\dot{x}}{R}, \\ \omega_3 &= 0.\end{aligned} \tag{22}$$

Let $\hat{\omega}_b$ be the angular velocity vector of the ball relative to the *oXYZ*-coordinate system. Since both the plate and ball are rotating, $\hat{\omega}_b$ is the sum of the angular velocity of the ball relative to the plate and the angular velocity of the plate. Thus, from (21) and (22), $\hat{\omega}_b$ is given by

$$\begin{aligned}\hat{\omega}_b &= \omega_1\hat{i} + \omega_2\hat{j} + \omega_3\hat{k} + \hat{\omega}_p \\ &= (-\frac{\dot{y}}{R} + \dot{\theta}_x\cos\theta_y)\hat{i} + (\frac{\dot{x}}{R} + \dot{\theta}_y)\hat{j} + (\dot{\theta}_x\sin\theta_y)\hat{k}.\end{aligned} \tag{23}$$

The kinetic energy of the ball is given by

$$\begin{aligned}T_{ball} &= \frac{1}{2}m\hat{v}\cdot\hat{v} + \frac{1}{2}J_b\hat{\omega}_b\cdot\hat{\omega}_b \\ &= \frac{1}{2}m(\dot{X}^2 + \dot{Y}^2 + \dot{Z}^2) \\ &\quad + \frac{1}{2}J_b[(-\frac{\dot{y}}{R} + \dot{\theta}_x\cos\theta_y)^2 \\ &\quad + (\frac{\dot{x}}{R} + \dot{\theta}_y)^2 + (\dot{\theta}_x\sin\theta_y)^2],\end{aligned} \tag{24}$$

where m and J_b are the mass and moment of inertia of the ball, respectively. Let the inertia matrix of the plate be

$$J_p = \begin{bmatrix} J_{px} & 0 & 0 \\ 0 & J_{py} & 0 \\ 0 & 0 & J_{pz} \end{bmatrix},$$

where J_{px}, J_{py}, and J_{pz} are the moment of inertia about the *x*, *y*, and *z*-axes, respectively. The kinetic energy of the plate is then given by

$$\begin{aligned}T_{plate} &= \frac{1}{2}[\omega_{px}\ \omega_{py}\ \omega_{pz}]J_p\begin{bmatrix}\omega_{px}\\ \omega_{py}\\ \omega_{pz}\end{bmatrix} \\ &= \frac{1}{2}[J_{px}(\dot{\theta}_x\cos\theta_y)^2 + J_{py}(\dot{\theta}_y)^2 + J_{pz}(\dot{\theta}_x\sin\theta_y)^2].\end{aligned} \tag{25}$$

Then, the total kinetic energy of the system is

$$T = T_{ball} + T_{plate}\,. \tag{26}$$

From (14), the potential energy of system is given by

$$\begin{aligned}V &= mgZ \\ &= mg[-x\cos\theta_x\sin\theta_y + y\sin\theta_x \\ &\quad + R\cos\theta_x\cos\theta_y],\end{aligned} \tag{27}$$

where g is the gravitational acceleration. By using (8), (26) and (27), the Lagrangian function is given by

$$\begin{aligned}L &= \left(\frac{mR^2 + J_b}{2R^2}\right)\dot{x}^2 + \left(\frac{mR^2 + J_b}{2R^2}\right)\dot{y}^2 \\ &\quad + \frac{1}{2}\left(J_b + J_{py} + mR^2 + mx^2\right)\dot{\theta}_y{}^2 - m\dot{x}y\dot{\theta}_x\sin\theta_y \\ &\quad + \frac{1}{4}\left(2J_b + 2J_{px}\cos^2\theta_y + 2J_{pz}\sin^2\theta_y + mR^2 + mx^2\right. \\ &\quad \left. + 2my^2 + (mR^2 - mx^2)\cos 2\theta_y - 2mRx\sin 2\theta_y\right)\dot{\theta}_x{}^2 \\ &\quad + \left(\frac{mRx\sin\theta_y - (J_b + mR^2)\cos\theta_y}{R}\right)\dot{\theta}_x\dot{y} \\ &\quad - my\left(x\cos\theta_y + R\sin\theta_y\right)\dot{\theta}_x\dot{\theta}_y + \left(\frac{J_b + mR^2}{R}\right)\dot{\theta}_y\dot{x} \\ &\quad + mg\cos\theta_x(x\sin\theta_y - R\cos\theta_y) - mgy\sin\theta_x.\end{aligned} \tag{28}$$

Then, from (7) the dynamic equations of the ball and plate system are given by

$$\begin{aligned}&\frac{1}{R^2}(J_b + mR^2)\ddot{x} + \frac{1}{R}(J_b + mR^2)\ddot{\theta}_y \\ &-mx\dot{\theta}_y{}^2 - mg\cos\theta_x\sin\theta_y \\ &-2m\dot{y}\dot{\theta}_x\sin\theta_y + mR\dot{\theta}_x{}^2\cos\theta_y\sin\theta_y \\ &-my\ddot{\theta}_x\sin\theta_y - mx\dot{\theta}_x{}^2\sin^2\theta_y = 0,\end{aligned} \tag{29}$$

$$\begin{aligned}
&\frac{1}{R^2}(J_b + mR^2)\ddot{y} - my\dot{\theta}_x^2 \\
&-\frac{1}{R}[(J_b + mR^2)\cos\theta_y - mRx\sin\theta_y]\ddot{\theta}_x \\
&+\frac{1}{R}[2mRx\cos\theta_y + (2mR^2 + J_b)\sin\theta_y]\dot{\theta}_x\dot{\theta}_y \\
&+2m\dot{x}\dot{\theta}_x\sin\theta_y + mg\sin\theta_x = 0,
\end{aligned} \tag{30}$$

$$\begin{aligned}
&\frac{1}{R}\left[mRx\sin\theta_y - (J_b + mR^2)\cos\theta_y\right]\ddot{y} \\
&-\left(my\sin\theta_y\right)\ddot{x} - my\left(x\cos\theta_y + R\sin\theta_y\right)\ddot{\theta}_y \\
&+\left[J_b + J_{px}\cos^2\theta_y + \frac{1}{2}m\left(R^2 + x^2 + 2y^2\right)\right. \\
&+\frac{1}{2}m\cos 2\theta_y\left(R^2 - x^2\right) + J_{pz}\sin^2\theta_y - mRx\sin 2\theta_y\left.\right]\ddot{\theta}_x \\
&+\left(mx - mx\cos 2\theta_y - mR\sin 2\theta_y\right)\dot{\theta}_x\dot{x} - 2my\cos\theta_y\dot{\theta}_y\dot{x} \\
&+\frac{J_b}{R}\dot{y}\dot{\theta}_y\sin\theta_y + 2my\dot{y}\dot{\theta}_x - my\left(R\cos\theta_y - x\sin\theta_y\right)\dot{\theta}_y^2 \\
&+\left(2J_{pz}\cos\theta_y\sin\theta_y - 2mRx\cos 2\theta_y - 2J_{px}\cos\theta_y\sin\theta_y\right. \\
&\left.-mR^2\sin 2\theta_y + mx^2\sin 2\theta_y\right)\dot{\theta}_x\dot{\theta}_y \\
&-mg\left(R\sin\theta_x\cos\theta_y - y\cos\theta_x - x\sin\theta_x\sin\theta_y\right) = \tau_x,
\end{aligned} \tag{31}$$

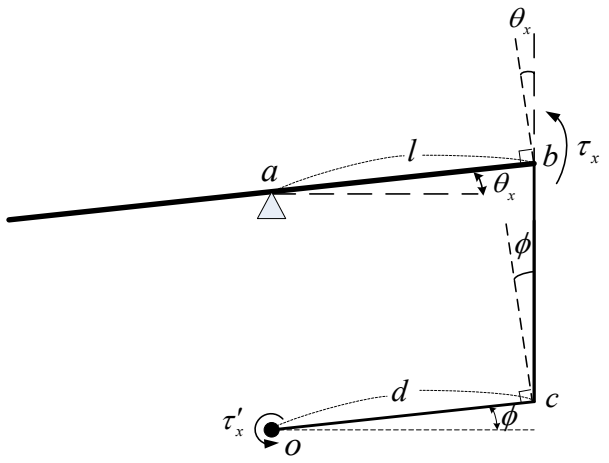

Figure 9. Connections of the linkages and plate.

$$\begin{aligned}
&\frac{1}{R}\left(J_b + mR^2\right)\ddot{x} - my\left(x\cos\theta_y + R\sin\theta_y\right)\ddot{\theta}_x \\
&+\left[J_b + J_{py} + m\left(R^2 + x^2\right)\right]\ddot{\theta}_y + 2mx\dot{x}\dot{\theta}_y \\
&-\frac{1}{R}\left(J_b\sin\theta_y + 2mRx\cos\theta_y + 2mR^2\sin\theta_y\right)\dot{y}\dot{\theta}_x \\
&+\frac{1}{2}\left[2mRx\cos 2\theta_y + 2\left(J_{px} - J_{pz}\right)\cos\theta_y\sin\theta_y\right. \\
&\left.+m\left(R^2 - x^2\right)\sin 2\theta_y\right]\dot{\theta}_x^2 \\
&-mg\left(x\cos\theta_y + R\sin\theta_y\right)\cos\theta_x = \tau_y.
\end{aligned} \tag{32}$$

This detailed dynamic model is adopted to build a reliable and accurate simulation model for the later simulation studies. The dynamics given in (29)-(32) are highly coupled and nonlinear. They are far too complex for control design purposes, so model simplification is necessary. The ball and plate system is a two-dimensional extension of the ball and beam system. Assuming that the operating ranges of θ_x and θ_y are small, the high-order coupling terms are, therefore, small and negligible. The ball and plate system can be treated as two decoupled ball and beam systems as follows:

$$(m + \frac{J_b}{R^2})\ddot{x} - mx\dot{\theta}_y^2 - mg\sin\theta_y = 0, \tag{33}$$

$$\begin{aligned}
&(J_{px} + J_b + my^2)\ddot{\theta}_x + 2my\dot{y}\dot{\theta}_x \\
&+mgy\cos\theta_x - mgR\sin\theta_x = \tau_x,
\end{aligned} \tag{34}$$

$$(m + \frac{J_b}{R^2})\ddot{y} - my\dot{\theta}_x^2 + mg\sin\theta_x = 0, \tag{35}$$

$$\begin{aligned}
&(J_{py} + J_b + mx^2)\ddot{\theta}_y + 2mx\dot{x}\dot{\theta}_y \\
&-mgx\cos\theta_y - mgR\sin\theta_y = \tau_y.
\end{aligned} \tag{36}$$

From (33)-(36), we know that the dynamic equations in the x and y-axes are identical. Also, the actuation mechanisms driving the tilts of the plate in the x and y-axes are identical. In the following, only the ball and beam system and the actuation mechanism in the x-axis will be discussed. The inclination of the plate is driven by a four-bar linkage as shown in Figure 9, where τ_x' is the torque generated by the motor, and τ_x is the torque exerted on the plate. Based on the force analysis, we have

$$\tau_x = \frac{l}{d}\tau_x'\cos\theta_x\cos\phi. \tag{37}$$

We assume that the angles of θ_x and ϕ are small enough that $l\sin\theta_x = d\sin\phi$ can be simplified as

$$l\theta_x \cong d\phi\,.$$

Since $l = d$, we have

$$\theta_x \cong \phi\,. \tag{38}$$

Note that θ_x is not directly measurable. However, for feedback control, θ_x can be approximated by the measurement of the angular displacement of the motor shaft. Moreover, (37) can be rewritten as

$$\tau_x \cong \tau_x'\cos^2\theta_x\,. \tag{39}$$

A voltage signal, which is generated according to a control law as designed below, is supplied to an amplifier, which drives the permanent-magnet dc motor to control the L-shaped linkage. Since the electrical time constant of a dc motor is usually much smaller than the mechanical time constant, and the value of the viscous friction coefficient is negligible, the following reduced-order model [24] of the dc motor is adopted:

$$\tau_x' = n\frac{K_{mx}}{R_{ax}}v_x - n^2\frac{K_{mx}^{\ 2}}{R_{ax}}\dot{\theta}_x, \tag{40}$$

where v_x is the control voltage, K_{mx} is the motor constant, R_{ax} is the armature resistance, and n is the gear ratio. In this study, the value of n is chosen as 8.

The state vector is defined as

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^T = \begin{bmatrix} y & \dot{y} & \theta_x & \dot{\theta}_x \end{bmatrix}^T. \tag{41}$$

From (34), (35), (39), and (40), the state space equation of the system on the basis of (41) can be written as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ E(x_1 x_4^2 - g\sin x_3) \\ x_4 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_x, \tag{42}$$

where u_x is the new control input, which is given by

$$u_x = \frac{1}{J_{px} + J_b + m{x_1}^2}\Big[\left(8\frac{K_{mx}}{R_{ax}}v_x - 64\frac{K_{mx}^2}{R_{ax}}x_4\right)\cos^2 x_3 - 2mx_1x_2x_4 + mgx_1\cos x_3 + mgR\sin x_3 \Big] \tag{43}$$

and

$$E = \frac{m}{m + \frac{J_b}{R^2}}.$$

The state space equation (42) is used for later control design. The physical parameters of the system are listed in Table 2.

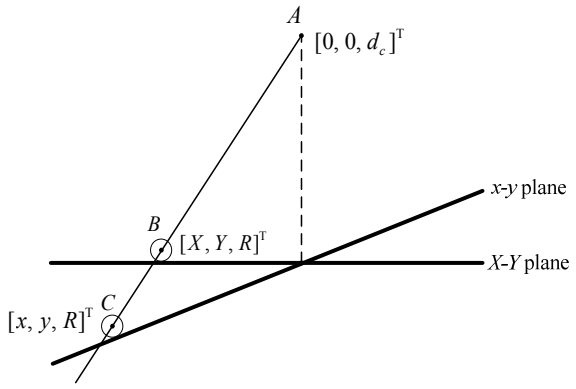

Figure 10. Coordinates of the centre of the ball measured by the machine vision system and the actual coordinates relative to the plate-fixed coordinate.

Based on the perspective projection (1), the machine vision system measures the position of the ball relative to the space-fixed coordinate *oXYZ*. To determine the actual position of the ball relative to the plate-fixed coordinate *oxyz*, we consider the coordinates of the centre of the ball measured by the machine vision system and the actual coordinates relative to the plate-fixed coordinate as shown in Figure 10.

Here, the point ***A*** with coordinates $[0, 0, d_c]^T$ relative to the space-fixed coordinate is the origin of the camera frame. The point ***B*** with coordinates $[X, Y, R]^T$ relative to the space-fixed coordinate is the centre of the ball measured by the machine vision system. The point ***C*** with coordinates $[x, y, R]^T$ relative to the plate-fixed coordinate is the actual centre of the ball. The coordinates of the point ***A*** relative to the plate-fixed coordinate are denote by $[a_x, a_y, a_z]^T$, which can be obtained using (11):

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = {}^P R_F \begin{bmatrix} 0 \\ 0 \\ d_c \end{bmatrix}. \tag{44}$$

The coordinates of the point ***B*** relative to the plate-fixed coordinate are denoted by $[b_x, b_y, b_z]^T$, which can be obtained by

$$\begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = {}^P R_F \begin{bmatrix} X \\ Y \\ R \end{bmatrix}. \tag{45}$$

Based on the perspective projection, the points ***A***, ***B***, ***C*** must be collinear. Thus, we have

$$\frac{x - a_x}{b_x} = \frac{y - a_y}{b_y} = \frac{R - a_z}{b_z}. \tag{46}$$

From (46), the centre of the ball on the *x-y* plane of the plate-fixed coordinate is given by

$$x = a_x + \frac{R - a_z}{b_z}b_x, \tag{47}$$

and

$$y = a_y + \frac{R - a_z}{b_z}b_y. \tag{48}$$

Description	Symbol	Value	Unit
Mass of the plate	m	0.56	kg
Radius of the ball	R	0.011	m
Moment of inertia of the ball	J_b	2.42×10^{-6}	kg.m^2
Moment of inertia of the plate about the *x*-axis	J_{px}	7.466×10^{-3}	kg.m^2
Moment of inertia of the plate about the *y*-axis	J_{py}	7.466×10^{-3}	kg.m^2
Moment of inertia of the plate about the *z*-axis	J_{pz}	14.933×10^{-3}	kg.m^2
Gravitational acceleration	g	9.8	m.sec^{-2}
Motor constant	K_{mx}	0.045	N.m/A
Motor armature resistance	R_{ax}	8.95	Ω

Table 2. Physical Parameters.

5. Tracking Control Design Based on Approximate Input-Output Feedback Linearization

This far, we have discussed the ball and plate system simplified into two decoupled ball and beam systems. In this section, for each decoupled ball and beam system we use an approximate input-output feedback linearization technique to design a tracking control law.

Consider a single-input single-output system of the form

$$\dot{x} = f(x) + g(x)u, \tag{49}$$

$$y_o = h(x), \tag{50}$$

where $x \in R^n$, u and $y_o \in R$, f and g are smooth vector fields on R^n, and $h : R^n \to R$ is a smooth scalar function. We assume that $x = 0$ is an equilibrium point of the system. The derivative of $h(x)$ along vector field $f(x)$ is expressed by the *Lie derivative* $L_f h(x) = \frac{\partial h}{\partial x} f(x)$, while $L_f^2 h(x)$ stands for $L_f(L_f h)(x)$, and $L_f^k h(x)$ denotes $L_f(L_f^{k-1}h)(x)$. Note that $L_g L_f h(x) = \frac{\partial L_f h(x)}{\partial x} g(x)$, while $L_g L_f^k h(x)$ stands for $L_g L_f(L_f^{k-1}h)(x)$, and so on. We take subsequent derivatives of the output until, say at γ, the control appears, and we obtain

$$\begin{aligned} \dot{y}_o &= L_f h(x), \\ \ddot{y}_o &= L_f^2 h(x), \\ &\vdots \\ y_o^{(\gamma)} &= L_f^\gamma h(x) + L_g L_f^{\gamma-1} h(x) u. \end{aligned} \tag{51}$$

The system of (49) and (50) is said to have the relative degree γ, which is well-defined in an open neighbourhood U of the origin if for $x \in U$, $L_g L_f^i h(x) = 0$ with $i = 0, 1, ..., \gamma - 2$, and $L_g L_f^{\gamma-1} h(x) \neq 0$. Then, the feedback control law $u = \frac{-L_f^\gamma h(x) + v}{L_g L_f^{\gamma-1} h(x)}$ achieves the input-output linearization from v to y_o.

In [20], it was shown that the relative degree of the ball and beam system is not well defined at certain locations, and this system violates the involutivity conditions of feedback linearization. Therefore, neither exact input-output feedback linearization nor full-state feedback linearization is applicable to this particular system. Alternatively, the approximate input-output feedback linearization proposed in [20] provides a method to find a set of coordinate transforms $\xi_i = \phi_i(x)$, $i = 1, ..., \gamma$, that approximate the output and its subsequent derivatives. The idea of approximate input-output feedback linearization is to construct the coordinate transform in the usual fashion as in the exact input-output feedback linearization by neglecting the second or higher order terms in the approximate output and its subsequent derivatives. We consider the ball and beam system of (42) with the output $y_o = h(x) = x_1$. To proceed with approximate input-output feedback linearization, a smooth function $\phi_1(x)$ is chosen to approximate the output

$$\xi_1 = \phi_1(x) = y_o = x_1. \tag{52}$$

Then, we obtain

$$\begin{aligned} \dot{\xi}_1 &= \dot{y}_o = \underbrace{x_2}_{\phi_2(x)}, \\ \dot{\xi}_2 &= \ddot{y}_o = \underbrace{Ex_1x_4^2 - Eg\sin x_3}_{\phi_3(x)}, \\ \dot{\xi}_3 &= \dddot{y}_o = \underbrace{Ex_2x_4^2 - Egx_4\cos x_3}_{\phi_4(x)} + \underbrace{2Ex_1x_4u_x}_{\substack{\text{disgard this} \\ \text{high order term}}}, \\ \dot{\xi}_4 &= y_o^{(4)} = E^2x_1x_4^4 + E(1-E)x_4^2\sin x_3 \\ &\quad + (-Eg\cos x_3 + 2Ex_2x_4)u_x. \end{aligned} \tag{53}$$

The term of $2Ex_1x_4u_x$ in $\dddot{y}_o$ is discarded, and then $y_o^{(4)}$ is obtained by taking the derivative of $\phi_4(x)$. By neglecting the term $2Ex_1x_4u_x$ and using the coordinate transformation

$$\begin{aligned} \xi_1 &= \phi_1(x) = x_1, \\ \xi_2 &= \phi_2(x) = x_2, \\ \xi_3 &= \phi_3(x) = Ex_1x_4^2 - Eg\sin x_3, \\ \xi_4 &= \phi_4(x) = -Egx_4\cos x_3 + Ex_2x_4^2, \end{aligned} \tag{54}$$

we obtain a feedback linearizable system given by

$$\begin{aligned} \dot{\xi}_1 &= \xi_2, \\ \dot{\xi}_2 &= \xi_3, \\ \dot{\xi}_3 &= \xi_4, \\ \dot{\xi}_4 &= \alpha(x) + \beta(x)u_x, \end{aligned} \tag{55}$$

where $\alpha(x) = E^2x_1x_4^4 + E(1-E)x_4^2\sin x_3$, and $\beta(x) = -Eg\cos x_3 + 2Ex_2x_4$. Thus, the feedback control law

$$u_x = \frac{1}{\beta(x)}[v_x - \alpha(x)] \tag{56}$$

approximately linearizes the ball and beam system of (42) with the output $y_o = h(x) = x_1$ from v to y_o up to the terms of $O(x, u_x)^2$. Here, a scalar function $\delta(x)$ is said to be $O(x)^n$ if $\lim_{\|x\| \to 0} \frac{|\delta(x)|}{\|x\|^n}$ exists and is not zero. Note that there exists a neighbourhood of the equilibrium, in which $\beta(x) \neq 0$. Since the control objective is to make the output

y_o track a desired trajectory y_d, approximate tracking can be obtained by choosing $u_x = \frac{1}{\beta(x)}[v_x - \alpha(x)]$ with

$$\begin{aligned} v_x = y_{dx}^{(4)} + K_3(\dddot{y}_{dx} - \xi_4) + K_2(\ddot{y}_{dx} - \xi_3) \\ + K_1(\dot{y}_{dx} - \xi_2) + K_0(y_{dx} - \xi_1) \end{aligned}, \quad (57)$$

where K_i are chosen so that $s^4 + K_3 s^3 + K_2 s^2 + K_1 s + K_0$ is a Hurwitz polynomial.

To analyse the tracking error and stability of the closed-loop system, we define the state vector and control input in the *x* and *y*-axes thus:

$$\begin{aligned} x_p^T &= \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \end{bmatrix}^T \\ &= \begin{bmatrix} y & \dot{y} & \theta_x & \dot{\theta}_x & x & \dot{x} & \theta_y & \dot{\theta}_y \end{bmatrix}^T, \\ u^T &= \begin{bmatrix} u_x & u_y \end{bmatrix}^T \end{aligned} \quad (58)$$

Then, the ball and plate system can be written as two decoupled ball and beam systems with higher order coupling terms as follows:

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} &= \begin{bmatrix} x_2 \\ E(x_1 x_4^2 - g\sin x_3) \\ x_4 \\ 0 \\ x_6 \\ E(x_5 x_8^2 - g\sin x_7) \\ x_8 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} u + \Delta f(x_p, u), \\ \begin{bmatrix} y_{ox} \\ y_{oy} \end{bmatrix} &= \begin{bmatrix} x_1 \\ x_5 \end{bmatrix}, \end{aligned} \quad (59)$$

where $\Delta f(x_p, u)$ are the higher order coupling terms. y_{ox} and y_{oy} are the outputs in the *x* and *y*-axes, respectively. Using the coordinate transforms (54) and feedback control law (56), the approximate input-output feedback linearization for the system (59) is given by

$$\begin{aligned} \begin{bmatrix} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \dot{\xi}_3 \\ \dot{\xi}_4 \\ \dot{\xi}_5 \\ \dot{\xi}_6 \\ \dot{\xi}_7 \\ \dot{\xi}_8 \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \xi_7 \\ \xi_8 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} + \psi(x_p, u), \\ \begin{bmatrix} y_{ox} \\ y_{oy} \end{bmatrix} &= \begin{bmatrix} \xi_1 \\ \xi_5 \end{bmatrix}, \end{aligned} \quad (60)$$

where $\psi(x_p, u)$ are the summation of the higher order coupling terms $\Delta f(x_p, u)$ and the approximation errors. We define the tracking error vector *e* as

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \end{bmatrix} = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \xi_7 \\ \xi_8 \end{bmatrix} - \begin{bmatrix} y_{dx} \\ \dot{y}_{dx} \\ \ddot{y}_{dx} \\ \dddot{y}_{dx} \\ y_{dy} \\ \dot{y}_{dy} \\ \ddot{y}_{dy} \\ \dddot{y}_{dy} \end{bmatrix}. \quad (61)$$

With the stable tracking control law (57), the closed-loop system can be then written as

$$\dot{e} = Ae + \psi\left(x_p, u(x_p, y_d)\right), \quad (62)$$

where $y_d^T = \begin{bmatrix} y_{dx} & \dot{y}_{dx} & \ddot{y}_{dx} & \dddot{y}_{dx} & y_{dy} & \dot{y}_{dy} & \ddot{y}_{dy} & \dddot{y}_{dy} \end{bmatrix}$ and *A* is an 8×8 matrix with all its eigenvalues in the open left-half complex plane. Since $\psi\left(x_p, u(x_p, y_d)\right)$ is a so-called uniformly higher order function [20], there exist $\varepsilon > 0$, $d > 0$, and a function K_ε, which is a monotonically increasing function of ε, such that

$$\left\| \psi\left(x_p, u(x_p, y_d)\right) \right\| \le \varepsilon K_\varepsilon \left(\|e\| + d \right) \quad (63)$$

for $x_p \in U_\varepsilon \subset R^8$ and $\|y_d\| \le d$. Since *A* is a Hurwitz matrix, there exists a positive definite matrix *P* such that

$$A^T P + PA = -I. \quad (64)$$

Consider the Lyapunov candidate function for the error system (62):

$$V = e^T P e. \quad (65)$$

The derivative of *V* along the trajectories of (62) is given by

$$\begin{aligned} \dot{V} &= -\|e\|^2 + 2e^T P \psi\left(x_p, u(x_p, y_d)\right) \\ &\le -\|e\|^2 + \varepsilon K_\varepsilon \mu \|e\| \left(\|e\| + d \right) \quad \text{(for some } \mu > 0) \\ &\le -\left(\frac{1}{2} - \varepsilon K_\varepsilon \mu \right) \|e\|^2 - \left(\frac{\|e\|}{\sqrt{2}} - \frac{1}{\sqrt{2}} \varepsilon k_\varepsilon \mu d \right)^2 + \frac{1}{2} (\varepsilon k_\varepsilon \mu d)^2 \\ &\le -\left(\frac{1}{2} - \varepsilon K_\varepsilon \mu \right) \|e\|^2 + \frac{1}{2} (\varepsilon k_\varepsilon \mu d)^2 \end{aligned} \quad (66)$$

Then, for all $\varepsilon \le \frac{1}{4K_\varepsilon \mu}$, it follows that

$$\dot{V} \le -\frac{\|e\|^2}{4} + \frac{(\varepsilon k_\varepsilon \mu d)^2}{2}. \quad (67)$$

Therefore, $\dot{V} < 0$ whenever $\|e\|$ is large, which implies that $\|e\|$ and, hence, the states of the closed-loop system are bounded. Moreover, for a sufficiently small reference command $\|y_d\|$ and appropriate initial conditions, we can conclude that the tracking error is of order $O(\varepsilon)$.

6. Simulation and Experimental Results

To measure the performance of the designed control law, the control system was simulated in MATLAB/Simulink using the parameters in Table 2. The control gains of (57) were chosen to be [K_3, K_2, K_1, K_0]=[8.7788, 145.6124, 799.0362, 1998.3245]. The simulation of circular trajectory tracking was conducted. The ball was commanded to follow a circular trajectory of radius 0.08 m, centred at the origin with a constant angular speed of 0.75 rad/sec. The corresponding trajectory references of each axis are $y_{dx} = 0.08\sin(0.75t)$ and $y_{dy} = 0.08\cos(0.75t)$. The initial conditions were set to $x = 0$ m, $y = 0$ m, $\theta_x = 0$ rad, $\theta_y = 0$ rad, $\dot{x} = 0$ m/sec, $\dot{y} = 0$ m/sec, $\dot{\theta}_x = 0$ rad/sec, and $\dot{\theta}_y = 0$ rad/sec. The simulation results of the position responses of the ball, the angular position responses of the inclining plate, the control voltages, position errors, and the trajectory of the ball are shown in Figure 11(a), (b), (c), (d) and (e), respectively. As shown in Figure 11(a) and (e), the ball converges towards the desired trajectory in 2 seconds with a starting point at the origin. The position errors are shown in Figure 11(d). This confirms that the static error in tracking a commanded trajectory is very small, as mentioned in Section 5. The operating range of the control voltage, which is ±20 V, is constrained by the motor driver.

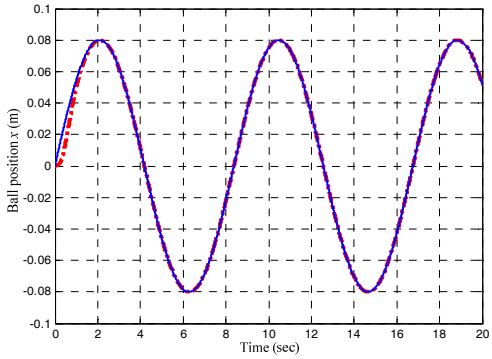

Figure 11 (a-1). Comparison of simulation results (dash-dot line) of the position responses of the ball and tracking command (solid line) in the *x*-axis.

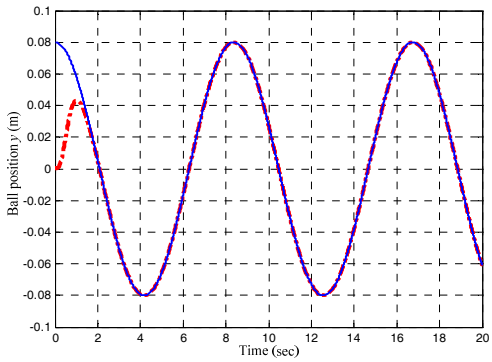

Figure 11 (a-2). Comparison of simulation results (dash-dot line) of the position responses of the ball and tracking command (solid line) in the *y*-axis.

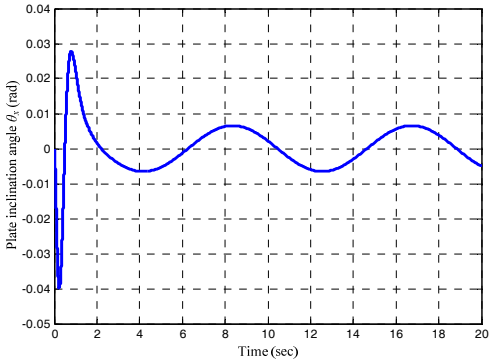

Figure11 (b-1). Simulation result of the angular position response of the inclining plate in the *x*-axis.

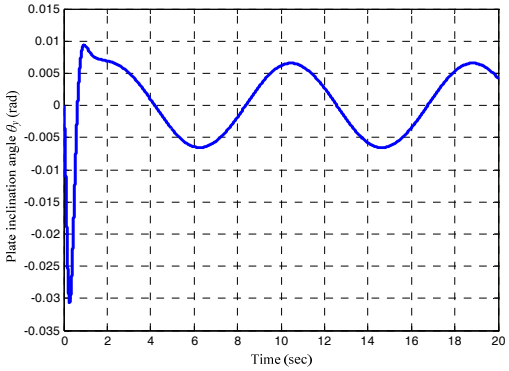

Figure 11 (b-2). Simulation result of the angular position response of the inclining plate in the *y*-axis.

Figure 11(c) shows that the control voltages did not saturate in the simulation for the chosen initial conditions and trajectory references.

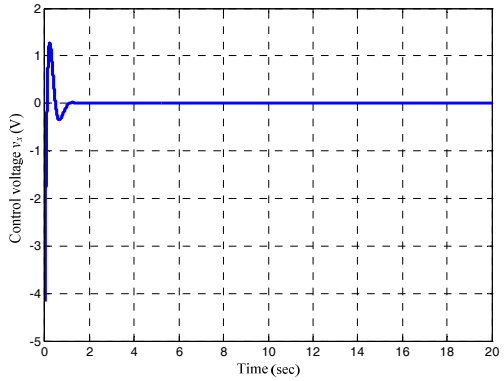

Figure 11 (c-1). Simulation result of the control voltage v_x.

For further validation, the designed control law and image processing algorithms were implemented and tested on the experimental setup, as shown in Figure 1. The image processing algorithms were implemented on the FPGA board in VHDL. The designed controller was implemented on the DSP system in C. The camera in the proposed system operated at 150 frames/sec. Thus, the sampling frequency of the system was set to 150 Hz.

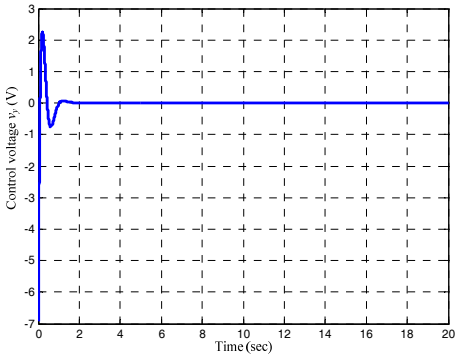

Figure 11 (c-2). Simulation result of the control voltage v_y.

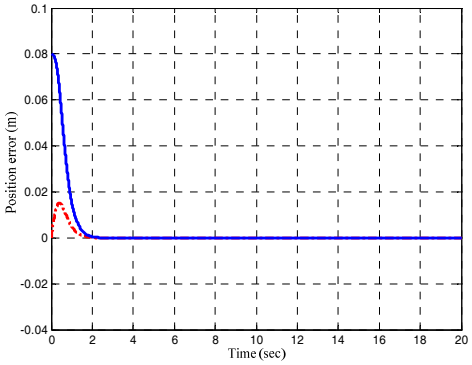

Figure 11 (d). Simulation results of the position error of the x-axis (dash-dot line) and position error of the y-axis (solid line).

The trajectory tracking experiment was conducted with the same initial conditions and controller parameters as in the simulation. The experimental results of the position responses of the ball, the angular position responses of the inclining plate, the control voltages, position errors, and the trajectory of the ball are shown in Figure 12(a), (b), (c), (d), and (e), respectively. As shown in Figure 12(a) and (e), the ball converged towards the desired trajectory in 1.9 seconds with a starting point at the origin. In Figure 12(d) it can be observed that the static position error in the x-axis is within a range from -0.0094 m to 0.0049 m, while the static position error in the y-axis is within a range from -0.0080 m to 0.0091 m. It also shows a good agreement between simulation results and experimental results. The mismatch and amplitude oscillations between simulation and experimental results are possibly caused by approximation errors in control design, disturbances, and unmodelled hardware effects such as friction, ball slipping, plate unevenness, illumination variation, low resolution of the camera (352×288 pixels), etc. As shown in the analysis of tracking error and stability, one can choose a small reference command and initial conditions with a small initial tracking error to improve tracking performance.

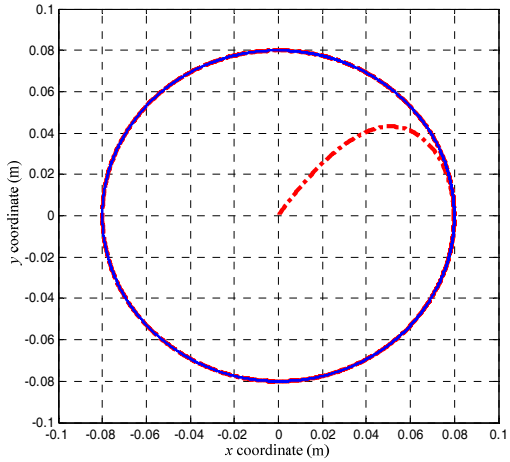

Figure 11 (e). Comparison of the simulation result (dash-dot line)of the trajectory of the ball and commanded trajectory (solid line).

For a camera with a low frame rate, visual servoing can result in performance degradation or even closed-loop instability. The obtained controller was discretized by the Tustin's method. The control system was then simulated in MATLAB/Simulink. From the simulation shown in Figure 13, we found that the closed-loop system becomes unstable for trajectory tracking when the sampling time is greater than 43 ms (23.26 Hz). The tested system cannot be stabilized if the frame rate is below 31 frames/sec. This is the reason this system requires a fast camera. A video clip demonstrating the performance of the developed system is available at http://lab.es.ncku.edu.tw/csplab/ball&plate-wmv.htm. The video shows that the system performs well even with different initial conditions and disturbance.

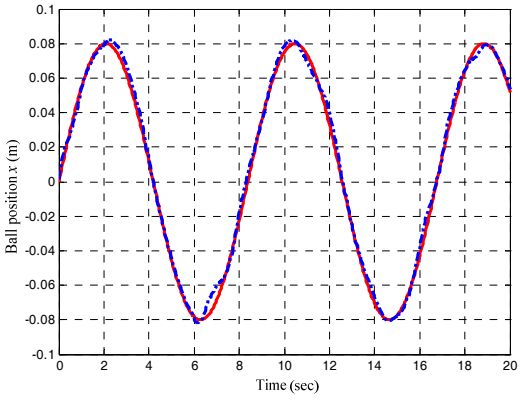

Figure 12 (a-1). Comparison of experimental result (dash-dot line) of the position response of the ball and tracking commands (solid line) in the$x$-axis.

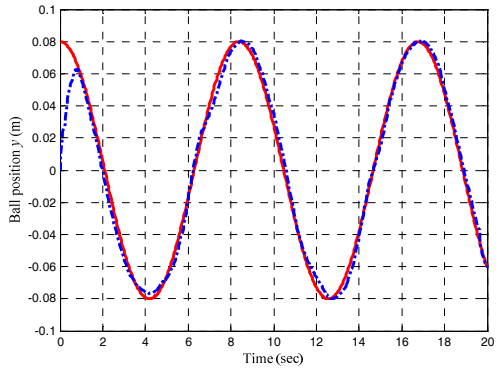

Figure 12 (a-2). Comparison of experimental result (dash-dot line) of the position response of the ball and tracking commands (solid line) in the y-axis.

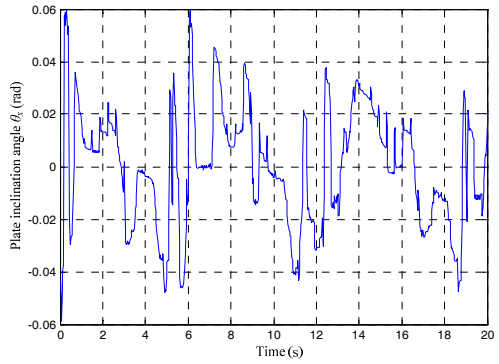

Figure 12 (b-1). Experimental result of theangular position response of the inclining plate in the x-axis.

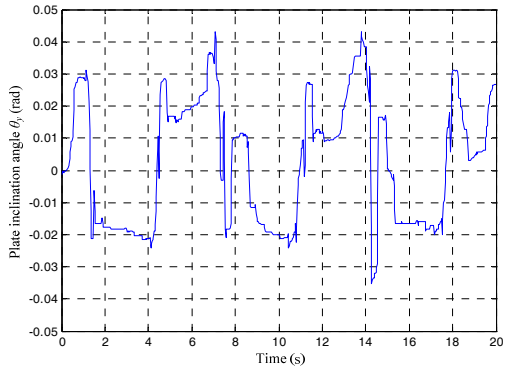

Figure 12 (b-2). Experimental result of the angular position response of the inclining plate in the$y$-axis.

7. Conclusions

Visual servoing tracking control for a ball and plate system was designed, implemented, and validated. The experimental setup, system modelling and implementation of the machine vision system were described in detail. The main contribution of this paper is twofold. First, the complete system model was derived for system simulation and analysis. Approximate input-output feedback linearization was used to design the control law for trajectory tracking. Second, an FPGA implementation was used to carry out image processing algorithms to achieve real-time constraints.

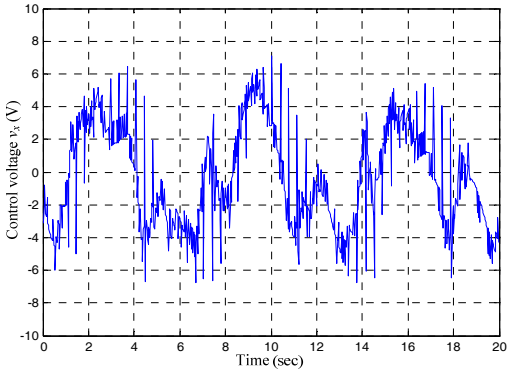

Figure 12 (c-1). Experimental result of the control voltage v_x.

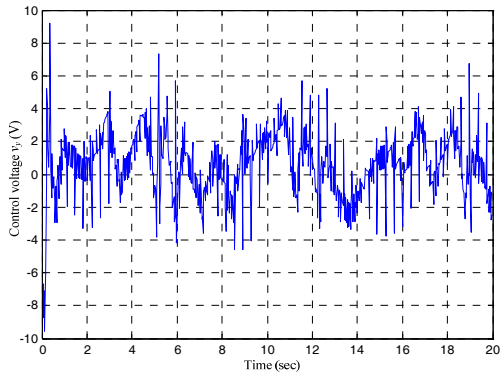

Figure 12 (c-2). Experimental result of the control voltage v_y.

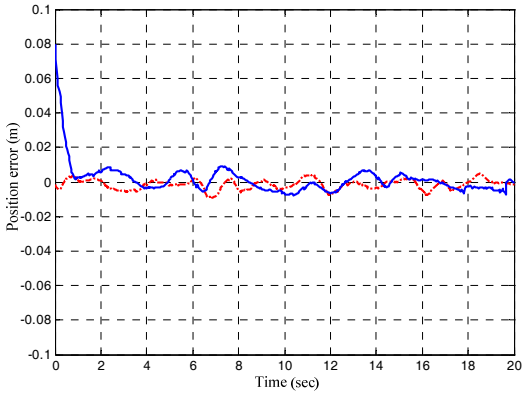

Figure 12 (d). Experimental results of the position error of the x-axis (dash-dot line) and position error of the y-axis (solid line).

The performance of the designed control system was investigated via simulated and experimental responses providing comparisons and the validation of controller design and system modelling. Both simulation and experimental results showed that the designed system had good performance. This experimental setup can serve as a test bed for nonlinear control schemes as well as supporting laboratory practice.

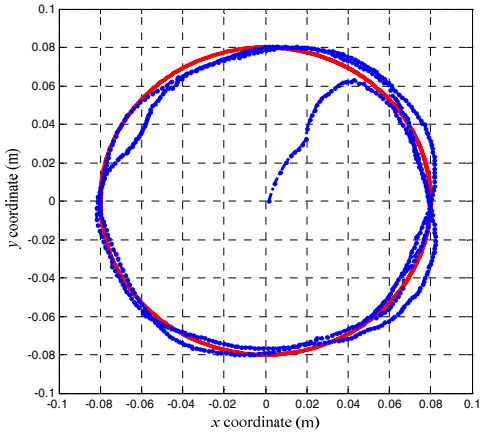

Figure 12 (e). Comparison of the experimental result (dot line)of the trajectory of the ball and commanded trajectory (solid line).

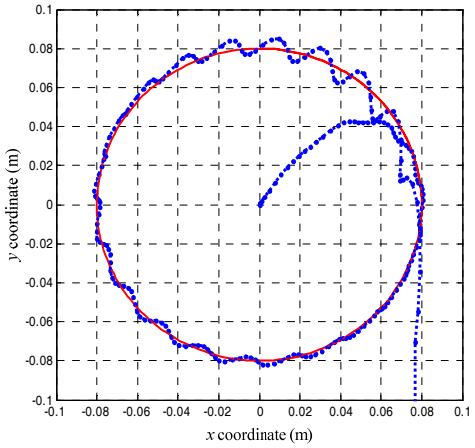

Figure 13. System becomes unstable when system performs trajectory tracking with a sampling time of 43 ms.

8. Acknowledgements

This work was supported by the Chieftek Precision Co., Ltd. and by the National Science Council, Taiwan, through grants NSC-100-2627-E-009-001 and NSC-101-2627-E-009-00.

9. References

[1] Awtar S, Bernard C, Boklund N, Master A, Ueda D, Craig K (2002) Mechatronic Design of a Ball on Plate Balancing System. Mechatronics. Vol. 12, No. 2, pp. 217-228.

[2] Fan X, Zhang N, Teng S (2003) Trajectory Planning and Tracking of Ball and Plate System Using Hierarchical Fuzzy Control Scheme. Fuzzy Sets and Systems. Vol. 144, No. 2, pp. 297-312.

[3] Park J H, Lee Y J (2003)Robust Visual Servoing for Motion Control of the Ball on a Plate. Mechatronics. Vol. 13, No. 7, pp. 723-738.

[4] Rad A B, Chan P T, Lo W L, Mok C K (2003) An Online Learning Fuzzy Controller. IEEE Transactions on Industrial Electronics. Vol. 50, No. 5, pp. 1016-1021.

[5] Yip P T (2004) Symbol-based Control of a Ball on Plate Mechanical System. Master's Thesis.University of Maryland.

[6] Moarref M, Saadat M, Vossoughi G (2008) Mechatronic Design and Position Control of a Novel Ball and Plate System. Proceedings of 16th Mediterranean Conference on Control and Automation, pp. 1071-1076.

[7] Liu D, Tian Y, Duan H (2009)Ball and Plate Control System Based on Sliding Mode Control with Uncertain Items Observe Compensation. Proceedings of IEEE International Conference onIntelligent Computing and Intelligent Systems, pp. 216-221.

[8] Moreno-Armendáriz M A,Pérez-Olvera C A,Rodriguez F O, Rubio E (2010)Indirect Hierarchical FCMAC Control for the Ball and Plate System. Neurocomputing, Vol. 73, No. 13-15, pp. 2454-2463.

[9] Moreno-Armendáriz M A, Rubio E, Pérez-Olvera C A (2010) Design and Implementation of a Visual Fuzzy Control in FPGA for the Ball and Plate System. International Conference on Reconfigurable Computing, pp. 85-90.

[10] Yuan D, Zhang Z (2010)Modeling and Control Scheme of the Ball-Plate Trajectory-Tracking Pneumatic System with a Touch Screen and a Rotary Cylinder. IET Control Theory and Applications, Vol. 4, No. 4, pp. 573-589.

[11] Hutchinson S, Hager GD, Corke PI (1996) A Tutorial on Visual Servo Control.IEEE Transactions on Robotics and Automation, Vol. 12, No. 5, pp. 651-670.

[12] Batlle J, Marti J, Ridao P, Amat J (2002) A New FPGA/DSP-based Parallel Architecture for Real-Time Image Processing. Real-Time Imaging, Vol. 8, No. 5, pp. 345-356.

[13] Torres-Huitzil C, Arias-Estrada M (2004) Real-Time Image Processing with a Compact FPGA-based Systolic Architecture. Real-Time Imaging, Vol. 10, No. 3, pp. 177-187.

[14] Martin J L, Zuloaga A, Cuadrado C, Lazaro J, Bidarte U (2005) Hardware Implementation of Optical Flow Constraint Equation Using FPGAs. Computer Vision and Image Understanding, Vol. 98, No. 3, pp. 462-490.

[15] Diaz J, Ros E, Pelayo F, Ortigosa E M, Mota S (2006) FPGA-based Real-Time Optical-Flow System. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, No. 2, pp. 274-279.

[16] Isidori A (1995) Nonlinear Control Systems. London: Springer.

[17] Nijmeijer H, Van der Schaft A J (1990) Nonlinear Dynamical Control Systems. New York: Springer.

[18] Marino R, Tomei P (1995) Nonlinear Control Design: Geometric, Adaptive and Robust. London: Prentice Hall.

[19] Krener A J (1984) Approximate Linearization by State Feedback and Coordinate Change. Systems and Control Letters. Vol. 5, No. 3, pp. 181-185.

[20] Hauser J, Sastry S, Kokotovic P V (1992) Nonlinear Control Via Approximation Input-Output Linearization: The Ball and Beam Example. IEEE Trans. on Automatic Control, Vol. 37, No. 3, pp. 392-398.
[21] Faugeras O (1993) Three Dimensional Computer Vision.Massachusetts: MIT Press. Cambridge.
[22] Ma Y, Soatto S, Kosecka J, Sastry S (2003) An Invitation to 3-D Vision: From Images to Geometric Models. New York: Springer.
[23] Gonzalez R C, Woods R E (2002) Digital Image Processing. New Jersey: Prentice Hall.
[24] Greenwood D T (2006) Advanced Dynamics.New York: Cambridge University Press.
[25] Utkin V, Guldner J, Shi J (1999) Sliding Mode Control in Electromechanical Systems. Florida: CRC Press. Boca Raton.