Design and Implementation of DSP and FPGA-Based Robust Visual Servoing Control of an Inverted Pendulum

Yi-Wei Tu and Ming-Tzu Ho*, Member, IEEE

Abstract—This paper presents the design and implementation of robust real-time visual servoing control with an FPGA-based image co-processor for a rotary inverted pendulum. The position of the pendulum is measured with a machine vision system whose image processing algorithms are pipelined and implemented on a field programmable gate array (FPGA) device to meet real-time constraints. To enforce robustness to model uncertainty, and to attenuate disturbance and sensor noise, the design of the stabilizing controller is formulated as a problem of the mixed $H_2/H_\infty$ control, which is then solved using the linear matrix inequality (LMI) approach. The designed control law is implemented on a digital signal processor (DSP). The effectiveness of the controller and the FPGA-based image co-processor is verified through experimental studies. The experimental results show that the designed system is able to robustly control an inverted pendulum in real-time.

I. INTRODUCTION

Visual servoing [1] is a control framework that incorporates visual information in feedback control loops. Visual servoing has played a key role in many fields, such as robotics [2, 3], industrial automation [4], and automated vehicle guidance [5]. Visual servoing involves results from many areas, including image processing, kinematics, dynamics, control theory, and real-time computation. Real-time image data acquisition and processing is a critical issue in visual servoing applications. A high frame rate and low processing latency is essential since the visual servoing system must make quick decisions based on the information extracted from a scene. To meet real-time constraints, the development of visual servoing systems usually requires high-cost specialized hardware and software. This presents serious obstacles to the design of real-time visual servoing systems. Real-time image processing imposes serious demands on a high pixel processing rate, massive and parallel computation, and hardware utilization. General-purpose processors cannot always provide enough computational power to fulfill 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 as implementation platforms for real-time image processing [6, 7]. Measurements made from images are noisy in nature due to changes in the environment, background, and illumination, which deteriorates the performance of visual servoing systems. To overcome this difficulty, a filter is commonly used to smooth noisy data before it is fed to the controller. Various filtering approaches have been proposed in the literature for visual servoing, such as Kalman filters [8].

The $H_2$ and $H_\infty$ norms are the most important performance and robustness measures for the analysis and synthesis of control systems. Combining the merits of $H_2$ optimal control and $H_\infty$ robust control, the mixed $H_2/H_\infty$ control design allows one to minimize a nominal performance measurement subject to robust stability constraints. The mixed $H_2/H_\infty$ control was first formulated in [9] as an LQG control-design problem with a constraint on $H_\infty$ disturbance attenuation. In [10], the state-feedback and output-feedback problems of the mixed $H_2/H_\infty$ control were considered. It was shown that the mixed $H_2/H_\infty$ control problem can be reduced to a convex optimization problem, which is computationally tractable. In recent years, linear matrix inequalities (LMIs) [11, 12] have emerged as a powerful tool for numerically solving multiobjective control synthesis problems. Once a problem has been expressed in the form of LMIs, efficient numerical algorithms and software packages [13] can be used to solve the LMIs in a fast and user-friendly manner. An LMI-based solution to the mixed $H_2/H_\infty$ control problem was presented in [14]. A complete overview of the LMI-based approach applied to multiobjective control synthesis was given in [15], which showed that LMIs offer great flexibility for combining various design constraints on a closed-loop system in a numerically tractable manner. A number of studies have used the mixed $H_2/H_\infty$ control to design control systems for real-world problems [16, 17].

This paper presents the design and implementation of a robust real-time visual servoing system that uses off-the-shelf technology. The proposed design is applied to the problem of balancing an inverted pendulum using visual feedback. This is similar to the problem of balancing a broomstick vertically on the palm of a human hand using hand-eye coordination. The control of an inverted pendulum is a well-known challenging problem that is widely used as a benchmark for verifying the performance and effectiveness of new control algorithms or technology due to its structural simplicity. Understanding how to use visual feedback to balance an inverted pendulum allows us to solve other visual servoing problems. Some studies [18, 19] have reported the use of visual information for the balance control of the inverted pendulum system using specialized vision systems. Visual servoing control requires a machine vision system with the ability of real-time image processing. In comparison with the
digital signal processor (DSP). FPGA implementation provides higher performance in repetitive and massive computations due to inherent architectural parallelism. Thus, in this paper, the position of the pendulum is measured with a machine vision system whose pipelined image processing is implemented on an FPGA device to meet real-time constraints. To enforce robustness to model uncertainty, and to attenuate disturbance and noise due to visual measurements, the mixed $H_2/H_\infty$ control is applied to design the stabilizing controller.

The rest of this paper is organized as follows. Section 2 describes the relationship between the world coordinates and the image coordinates. The image processing algorithms for determining the displacement of the pendulum from the image data are introduced. Section 3 presents the hardware implementation and architecture of the image processing algorithms on an FPGA device. In Section 4, the mathematical model of the inverted pendulum system is presented. The design of a mixed $H_2/H_\infty$ controller is given in Section 5. The experimental results are given in Section 6. Finally, Section 7 contains some concluding remarks.

II. VISUAL MEASUREMENT AND IMAGE PROCESSING ALGORITHMS

The machine vision system is used to determine the angular displacement of the pendulum. A circular-shaped marker is placed on the top of the pendulum to simplify visual sensing and to increase the accuracy of visual measurements. Determination of the position of the marker is based on the perspective pinhole camera model [20, 21], which is given by

$$\lambda p = K[R | t] P, \quad (1)$$

where $p = [x, y, 1]^T$ represents the two-dimensional homogeneous coordinates of the image point in the image coordinate system, $P = [X, Y, Z, 1]^T$ represents the three-dimensional homogeneous coordinates of the object point in the world coordinate system, and $\lambda$ is a scalar factor. The 3×3 matrix $R$ and three-dimensional vector $t$ are the extrinsic camera parameters that describe rotation and translation between the world frame and the camera frame, respectively. $K$ is the intrinsic camera parameter matrix given by

$$K = \begin{bmatrix} f_{sx} & f_{so} & o_x \\ 0 & f_{sy} & o_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

Here, $(o_x, o_y)$ is the principal point on the image coordinate system in pixels. $f_{sx}$ is the size of unit length in horizontal pixels, $f_{sy}$ is the size of unit length in vertical pixels, and $f_{so}$ is the skew of the pixels. The world frame is chosen to be coincident with the camera frame. Thus, the extrinsic camera parameters are

$$[R | t] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (3)$$

The intrinsic camera parameters are obtained using the camera calibration procedure proposed in [20]. $\lambda$ is chosen to be the distance from the origin of the camera frame to the pendulum. The intrinsic camera parameters are given in Table I.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>0.16 m</td>
</tr>
<tr>
<td>$f_{sx}$</td>
<td>20.3591×10³ pixels/m</td>
</tr>
<tr>
<td>$f_{sy}$</td>
<td>21.2285×10³ pixels/m</td>
</tr>
<tr>
<td>$f_{so}$</td>
<td>-0.2207×10³ pixels/m</td>
</tr>
<tr>
<td>$o_x$</td>
<td>-0.4186 pixels</td>
</tr>
<tr>
<td>$o_y$</td>
<td>-0.2136 pixels</td>
</tr>
</tbody>
</table>

To determine the position of the marker on the image plane, the captured images are processed by a series of image processing algorithms. A sample of a raw image with a uniform background captured by the camera is shown in Fig. 1. The essential image processing algorithms [22] are described below.

A. Edge Detection

Edges are the pixels at or around which a large change in gray-level intensity occurs. In most images, edges characterize object boundaries and are therefore useful for the segmentation [22] and identification of objects in a scene. Edge detection is the process of locating edges in an image by detecting significant intensity changes. Most edge detection techniques are based on the idea of computing image gradients. Let $f(x, y)$ denote the grayscale level of the pixel at point $(x, y)$ in the image coordinate system.

In this paper, the computation of the image gradient is based on the Sobel operator [22], which approximates the partial derivatives by a mask operation. A general 3×3 mask structure is defined as

$$\begin{bmatrix} w(-1,-1) & w(-1,0) & w(-1,1) \\ w(0,-1) & w(0,0) & w(0,1) \\ w(1,-1) & w(1,0) & w(1,1) \end{bmatrix}. \quad (4)$$

The corresponding mask operation is given by

$$f_x(x, y) = \sum_{j=-1}^{1} \sum_{i=-1}^{1} f(x+i, y+j) w_x(i, j) \quad (5)$$

$$f_y(x, y) = \sum_{j=-1}^{1} \sum_{i=-1}^{1} f(x+i, y+j) w_y(i, j) \quad (6)$$

where $w_x(i, j)$ is the coefficient of the horizontal mask, and $w_y(i, j)$ is the coefficient of the vertical mask. Since the background of the scene is uniform and the objects in the scene are simple, the Sobel edge detector is adequate to extract the edges of the objects. Sobel edge detection was
applied to the image in Fig. 1 with the resulting image shown in Fig. 2.

![Fig. 2. Edge detection using Sobel masks for the image in Fig. 1.](image)

**B. Thresholding**

Thresholding is a simple and efficient method applied to gray-level intensity images to differentiate between objects and the background in the image. Thresholding converts a gray-level image into a binary image that contains all of the essential information concerning the shape, position, and number of objects. Since we are interested in the lighter marker and the background is uniformly darker, global thresholding is used. The thresholding process is given by

$$f_T(x, y) = \begin{cases} 
255 & \text{if } f(x, y) \geq T \\
0 & \text{if } f(x, y) < T 
\end{cases}$$

where \(T\) is the threshold value. Applying thresholding with a threshold value \(T = 150\) to the image of Fig. 2 produces the image in Fig. 3.

![Fig. 3. Thresholding applied to the image in Fig. 2.](image)

**C. Centroid and Displacement Determination**

Ideally, the thresholding operation separates the edge of the marker from the background and the pendulum rod of the scene. As shown in Fig. 3, the white pixels correspond to the edge of the marker. Because the marker is symmetrical, the centroid of the edge is taken as its location. The centroid \((x_c, y_c)\) is given by

$$x_c = \frac{\sum x \cdot f_T(x, y)}{\sum f_T(x, y)},$$

$$y_c = \frac{\sum y \cdot f_T(x, y)}{\sum f_T(x, y)},$$

where \(f_T(x, y)\) represents the binary level of the pixel at point \((x, y)\) in the image coordinate system.

**III. IMPLEMENTATION OF IMAGE PROCESSING ALGORITHMS ON AN FPGA**

In order to process the image data in real-time, the image processing algorithms introduced in the previous section are implemented on an FPGA using the hardware description language VHDL. In this study, the image processing is divided into the modules shown in Fig. 4.

The functions of the modules are:

- **M1**: The Inter-Integrated Circuit (I2C) 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 10 Mbytes/s from the camera. The image data can be directly passed to the next module.
- **M3**: This module consists of two parallel operations that realize Sobel edge detection using the horizontal and vertical mask operations (Equations (2) and (3)) to obtain the derivatives. Vertical and horizontal pixel counters are used to position the masks.
- **M4**: The sum of the absolute values of image derivatives is calculated to obtain the magnitude of the image gradient. The circuit synthesis requires an adder in this stage.
- **M5**: This module applies thresholding (Equation (4)) to the output image of the edge detection stage.
- **M6**: The centroid of the marker (Equation (5) and (6)) is calculated in this stage. The circuit synthesis requires two multipliers, three adders, two dividers, and three sum registers.

![Fig. 4. Parallel/pipelined architecture of FPGA.](image)

**IV. MODELING OF THE ROTARY INVERTED PENDULUM SYSTEM**

Consider the basic features of the rotary inverted pendulum system shown in Fig. 5.

![Fig. 5. Rotary inverted pendulum system.](image)

The variables and parameters of the system are defined as follows:

- \(m_1\): Mass of the rotating arm (kg)
- \(m_2\): Mass of the pendulum (kg)
\( l_1 \): Length of the rotating arm (m)
\( l_2 \): Length of the pendulum (m)
\( C_1 \): Distance from the pivot point to the center of gravity of the rotating arm (m)
\( C_2 \): Distance from the pivot point to the center of gravity of the pendulum (m)
\( J_1 \): Moment of inertia of the rotating arm about the center of gravity (kg-m\(^2\))
\( J_2 \): Moment of inertia of the pendulum about the center of gravity (kg-m\(^2\))
\( \theta_1 \): Angular displacement of the rotating arm (degrees)
\( \theta_2 \): Angular displacement of the pendulum with respect to the vertical (degrees)
\( g \): Gravitational acceleration (m/s\(^2\))
\( \tau_r \): Torque exerted on the rotating arm (N-m)

The Euler-Lagrange formulation [23] was used to obtain the mathematical model of the system. It describes the dynamic behavior of the rotary inverted pendulum system.

\[
M(q) \ddot{q} + V_w(q, \dot{q}) \dot{q} + G(q) = \tau + d, \tag{7}
\]

where \( d \) is the disturbance torque, and

\[
q = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_r \\ 0 \end{bmatrix}, \quad d = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}. \tag{8}
\]

\[
M(q) = \begin{bmatrix} P_1 + P_2 \sin^2 \theta_1 & P_3 \cos \theta_2 \\ P_3 \cos \theta_2 & P_4 \end{bmatrix}, \tag{9}
\]

\[
V_w(q, \dot{q}) = \begin{bmatrix} \alpha \ddot{\theta}_2 \\ -P_2 \dot{\theta}_2 \sin \theta_2 + a \dot{\theta}_1 \end{bmatrix}, \tag{10}
\]

\[
G(q) = \begin{bmatrix} 0 \\ -P_5 \dot{\theta}_1 \sin \theta_2 \end{bmatrix}, \tag{11}
\]

\[
\alpha = \frac{1}{2} P_4 \sin (2 \theta_1).
\]

The parameters \( P_i \), \( i = 1, \ldots, 5 \) are defined as

\[
P_1 = m_1 l_1^2 + J_1, \quad P_2 = m_2 C_2^2, \quad P_3 = m_2 l_2 C_2, \quad P_4 = m_2 C_2^2 + J_2, \quad P_5 = m_2 C_2 g.
\]

The relation between the force and control voltage is given by

\[
\tau_1 = \frac{K_m}{R_o} u - \frac{K_m^2}{R_o} \dot{\theta}, \tag{12}
\]

where \( \tau_1 \) is the control torque, \( u \) is the control voltage, \( K_m \) is the motor constant and \( R_o \) is the armature resistance. TABLE II lists the physical parameters of the rotary inverted pendulum system.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_1 )</td>
<td>0.065 kg</td>
<td>( m_2 )</td>
<td>0.022 kg</td>
</tr>
<tr>
<td>( l_1 )</td>
<td>0.16 m</td>
<td>( l_2 )</td>
<td>0.156 m</td>
</tr>
<tr>
<td>( C_1 )</td>
<td>0.08 m</td>
<td>( C_2 )</td>
<td>0.078 m</td>
</tr>
<tr>
<td>( J_1 )</td>
<td>0.00056 kg-m(^2)</td>
<td>( J_2 )</td>
<td>0.00017 kg-m(^2)</td>
</tr>
<tr>
<td>( K_m )</td>
<td>0.01376 N-m/A</td>
<td>( R_o )</td>
<td>1.08265 ( \Omega )</td>
</tr>
<tr>
<td>( g )</td>
<td>9.8 m/s(^2)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

V. Stabilizing Controller Design Using Mixed \( H_2 \)/\( H_\infty \) Control Synthesis

The rotary inverted pendulum system has two equilibrium points. The downward pendulum is a stable equilibrium point. The upward pendulum is an unstable equilibrium point. In order to design the stabilizing controller, the Jacobian linearized model is needed. First, the state variables are defined as

\[
x = [x_1, x_2, x_3, x_4]^T = [\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]^T. \tag{13}
\]

Using the Jacobian linearization method [24], the dynamic equations (7) and (12) are linearized with respect to the stable equilibrium \([0, 0, 0, 0]^T\). The state-space equations of the linearized system are given by

\[
\begin{align*}
x_1 &= x_2, \\
x_2 &= x_3, \\
x_3 &= \frac{1}{\Delta \ell}(-P_4 P_5 x_1 - P_2 P_3 x_2 + P_4 P_5 u) + P_4 d_1 - P_3 d_2, \\
x_4 &= \frac{1}{\Delta \ell}(P_5 P_5 x_1 + P_5 P_3 x_2 - P_5 P_3 u) - P_4 d_1 + P_3 d_2,
\end{align*}
\]

where

\[
\Delta \ell = P_4 P_5 - P_2^2, \quad P_6 = \frac{K_m^2}{R_o}, \quad P_7 = \frac{K_m}{R_o}.
\]

With the parameters from TABLE II, we obtain the state-space equations. The control objective is to stabilize the pendulum in the upright position. In this study, the measurement inaccuracy, quantization errors, and variance in illumination of visual measurements are regarded as white noise. The synthesis of the controller is formulated as a problem of the mixed \( H_2/H_\infty \) control. The \( H_\infty \) norm constraint is used to enforce robustness against dynamic uncertainty, while the \( H_2 \) performance is used to ensure control effort, disturbance attenuation, and sensor noise attenuation. We introduce the vector of the measured output

\[
y = C_y x + D_y w_0,
\]

where \( w_0 = [d \quad n]^T \) is the exogenous input, \( d \in \mathbb{R}^2 \) is the torque disturbance as previously defined, and \( n \in \mathbb{R} \) is the visual sensor noise. The regulated output for \( H_2 \) performance is chosen to be

\[
z_2 = C_z x + D_z u.
\]

Thus, the open loop inverted pendulum model \( P \) is given by

\[
\dot{x} = Ax + Bu + B_\nu w_0,
\]

\[
z_2 = C_z x + D_z u,
\]

\[
y = C_y x + D_y w_0.
\]

The control is given by an output-feedback controller \( K \)

\[
\dot{x} = Ax + Bu + B_\nu y,
\]

\[
u = C_z x + D_z y,
\]

where \( x \in \mathbb{R}^n \) and the matrices \( A, B, C_z, \) and \( D_z \) are of appropriate dimensions. A block diagram of the closed-loop system, which includes the linearized inverted pendulum
model, the feedback controller, and the performance objectives, is shown in Fig. 6.

![Fig. 6. Block diagram of the closed-loop system.](image)

Here, we assume that the torque disturbance \( \tau \) is a unit impulse, which can represent the impulsive torque disturbance, such as a sudden tapping on the pendulum. The sensor noise \( n = W_s(s) \eta_n \), where \( \eta_n \) is the white noise of zero mean and unit variance, and \( W_s(s) \) is the noise model.

To determine the noise model \( W_s(s) \), the visual sensor and an optical encoder are used to measure the angular displacement of the pendulum simultaneously. We assume that the measurement of the optical encoder is noise-free. The sensor noise of the visual measurement is obtained by subtracting the measurement of the optical encoder from that of the visual sensor. Using FFT, we then obtain the frequency content of the noise, which is shown in Fig. 7. The frequency spectrum shown in Fig. 7 indicates that the noise from the visual sensor is colored. The noise model \( W_s(s) \) is selected to envelope the magnitude of the spectral content of the noise. It is given by

\[
W_s(s) = \frac{5.2}{s + 4.5}.
\]

![Fig. 7. Frequency spectrum of the noise and frequency response of the noise model.](image)

Let \( w = [d \quad \eta_n]^T \), and denote the sensitivity function of the closed-loop system to be

\[
S = (I + PK)^{-1}.
\]

The objective of the control design is to design a feedback controller \( K \) such that the closed-loop system is asymptotically stable. Moreover, the closed-loop system satisfies the following design specifications:

1. The closed-loop system guarantees robust stability against the additive plant uncertainty;
2. The \( H_2 \) norm of the closed-loop transfer function \( T_{w2} \) from \( w \) to \( z_2 \) is minimized to achieve the optimal \( H_2 \) performance.

Let \( W_{z_2}(s) \) denote the transfer function matrix describing the frequency-domain characteristics for which the robustness is required. The \( H_\infty \) norm of \( \| W_{z_2} KS \|_\infty < 1 \) guarantees robust stability against the additive plant uncertainty. Thus, the above design specifications can be converted into a mixed \( H_2 / H_\infty \) control design problem. That is to design a stabilizing controller \( K \) such that:

\[
\min_\gamma \| W_{z_2} KS \|_\infty < 1 \text{ and } \| T_{z_2} \|_2 < \gamma.
\]

The weighting function matrix

\[
W_{z_2}(s) = 0.6 \cdot \frac{s + 100}{0.01s + 100} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\]

is chosen to have high-pass characteristics to ensure robustness against plant uncertainty. The above mixed \( H_2 / H_\infty \) control design can be cast into an LMI optimization problem [15] that can be solved in a numerically tractable manner. We use the MATLAB function \textit{hinfdmix} in the LMI Control Toolbox [28] to obtain a ninth-order controller. Note that the closed-loop performance and robustness of the system depend strongly on the selection of the weights \( W_{z_2}(s) \) and \( W_{z_2}(s) \).

VI. EXPERIMENTAL RESULTS

The designed control law and image processing algorithms were implemented and tested on the experimental setup shown in Fig. 8. The image processing algorithms were implemented on the FPGA board in VHDL. The designed robust controller was implemented on the DSP system in C. In this work, the camera can operate up to 580 frames/sec. However, the higher the frame rate, the stronger the intensity of light sources needed to ensure good quality of the acquired image. To overcome this difficulty, we lowered the frame rate to 250 frames/sec and installed three light emitting diodes near the top of the camera to provide the necessary intensity of light sources. The sampling frequency of the system was set to 250 Hz. The angular position responses of the pendulum and the rotating arm are shown in Fig. 9 and Fig. 10, respectively. The experimental results show that the designed visual servoing control robustly stabilizes the inverted pendulum system. To examine the robustness of the control system, an impulsive disturbance was manually added by tapping the pendulum at time 14 s (impulsive disturbance is indicated with \( D \) in Fig. 9). As shown in Fig. 9 and Fig. 10, the control system recovers from the impulsive disturbance.
In this paper, we reported the design and implementation of robust visual servoing control for balancing a rotary inverted pendulum. The image processing of the machine vision system was implemented on an FPGA device to meet real-time constraints. The robust controller was implemented through a DSP. The main contribution of this paper is twofold. First, the mixed $H_2/H_\infty$ control was used to enhance the ability against plant uncertainty and noise due to image measurement. Secondly, the FPGA implementation was used to carry out image processing algorithms to achieve real-time constraints. The controller and the machine vision system were implemented and tested. Even with disturbance and sensor noise, the designed controller balanced the pendulum.

VII. CONCLUSIONS

In this paper, we reported the design and implementation of robust visual servoing control for balancing a rotary inverted pendulum. The image processing of the machine vision system was implemented on an FPGA device to meet real-time constraints. The robust controller was implemented through a DSP. The main contribution of this paper is twofold. First, the mixed $H_2/H_\infty$ control was used to enhance the ability against plant uncertainty and noise due to image measurement. Secondly, the FPGA implementation was used to carry out image processing algorithms to achieve real-time constraints. The controller and the machine vision system were implemented and tested. Even with disturbance and sensor noise, the designed controller balanced the pendulum.

REFERENCES