Position Control of a Ball on a 2D Surface

Usama Anwar, Hamza Anwar, Abubakr Muhammad
Department of Electrical Engineering
SBA School of Science & Engineering, LUMS
Lahore, Pakistan
{14100194, 14100048, abubakr}@lums.edu.pk

***Abstract* — Ball balancing, like many others is a fundamental problem in control systems. Digital controller implementation in this regard has been formalized in our approach to solve the problem of ball position control. In this paper two dimensional control of plate for balancing of ball is discussed and an innovative control algorithm, involving novel visual sensing and perception has been presented.**

Keywords—digital control systems; position control; ball balancing in 2D

I. Introduction

Balancing of objects is considered to be a central application of control systems. Controlling of objects finds many applications in systems where one needs precise control over body position. For example in physics experiments one needs platforms that are vibration free and resilient to small vibrations of earth. In such cases one needs to devise systems that are able to maintain certain state of system under changing environment and external disturbances. In this perspective we have considered controlling of ball in two dimensions. The problem is explored for research purposes in order to consider different techniques that have been presented to solve this problem and suggest further improvements that can be made in effectively solving this control problem. Furthermore our control problem is presented as an extension to previously well-known control problems.

Controlling the position of a ball on a beam is a 1D control problem. We aim to extend this to 2D by balancing a ball on a flat plate. This plate's orientation will be actuated by two motors along orthogonal spatial dimensions, which increases motors required to actuate the process. Due to this the number of required sensors also increase and the system becomes MIMO from MISO. So a MIMO digital controller would be designed to address this control problem.

In order to capture position of ball on a surface, camera has been used. Several techniques can be used for extracting position of ball from the image obtained from camera. TLD [2], image segmentation or global flow are some of the standard techniques that could be used for this purpose. Among the techniques available, we have used image segmentation as the most appropriate one for our experiment.

After determining the ball position, one then needs to run a control algorithm on this and keep the ball to its center position. In order to design efficient controller for the system proper theoretical model of the system is developed. System involves modelling of dc motor and ball position on inclined plane. Voltage across the dc motor is applied which controls orientation of plate in the space. The orientation of plate then determines ball position and velocity in both x and y dimensions. So for consideration of system model one needs to consider a cascaded system of dc motor and inclined plane. In our paper separate state space analysis of dc motor and inclined plane have been presented and then a cascaded transfer function is presented.

Once the system model has been determined, next step is to design a controller for the system. In this paper we have presented new approach to controlling the system. Our approach involves separate control of dc motor which is then embedded in overall system and a controller is designed that then controls the overall system. Controller design for the system is presented later in the paper.

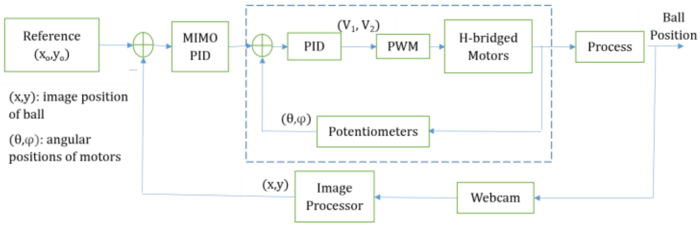

Figure 1(a). System Diagram

II. Methodology

The problem at hand is of regulating a ball's velocity at the center of the plate, and simultaneously tracking a reference position on the 2D plate. Our methodology in tackling this control problem is based on a generic feedback approach – an angular position control of motor using potentiometers.

The setup will consist of three sensors and two motors. Three sensors will compromise of two potentiometers to sense orthogonal angular positions of plate (θ, φ) and one webcam to determine position of ball on the plate (x, y). Camera will be mounted on top facing down onto the plate. To determine ball position, image processing algorithm will be implemented in MATLAB. Among the Image processing algorithms either object tracking algorithm (TLD [1]) or Image Segmentation (black ball against white background) will be employed.

To control the process, sensor outputs will be digitized using NI-DAQ cards and will be fed to a computer. MATLAB running on it will provide the environment to implement control algorithm. Before a digital controller could be used, relationship between ball position and orientation of plate needs to be established (2D inclined plane) which would be used to formulate input for the digital PID controller. PID controller will determine duty cycles needed for PWM speed control of motors. A DAQ card interface will connect it to the two H-bridges of the motors governing plate orientation and ball position.

The mechanical setup consists of a flat rectangular plate attached with two motors having perpendicular axes of rotation (Figure 1b). Both these motors have one free running potentiometer, to determine their angular positions. This arrangement makes our control problem simpler – reducing it to two independent motor position control problems (like two simultaneous ball-on-beam setups). The visual sensor for tracking ball position is a USB webcam. This leads to a complicated sensing algorithm – requires image processing. For this matter, we employ image segmentation as this is a rather faster strategy compared to tracking ball position frame-after-frame, or global flow estimation approach. Once we have the ball's current position (Data Acquisition Card) our digital controller can decide on how to rotate the motor to balance the ball at the reference position.

A. *Mathematical Model*

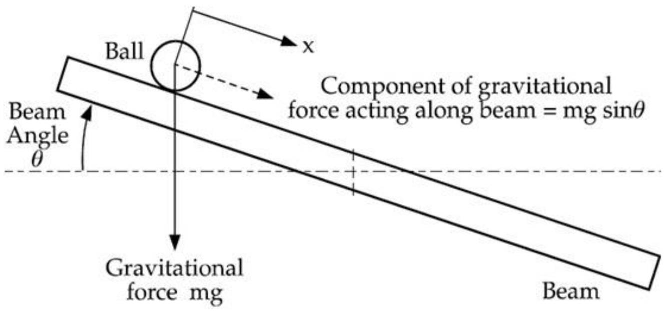

Figure 2. Relation between angle of beam and ball's position for a general ball-on-beam system

Consider a 1D ball on a beam problem (See Figure 2). The Newtonian equations of motion relate the angle of the inclined plane with the ball's position (ignoring friction). We have the following differential equation that describes the *process* (2D plate) in our plant:

$$m\ddot{x} \approx mg\theta - \frac{2}{5}mr^2 \times \frac{\ddot{x}}{r^2}$$

Approximating the beam angle to very small values makes our system linear and further analysis is easier to be done on it. Note that our specific problem is similar to balls on two independent beams, and so this mathematical model of the *process* is usable for the two degrees of motor angular position *(θ,φ)* and, consequently, for the two dimensions of ball's position *(x,y)* in the image captured.

Another major aspect in our system is that of motor position control. This can be regarded as a subsystem which is linearly appended to the *process*. So our combined system is plain concatenation of the two systems. The state space analysis of the process is given below, where x is the 1D position of the ball, y is the sensed position of the ball and θ is the motor's angular position:

$$\begin{bmatrix}\dot{x}\\ \ddot{x}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\end{bmatrix} + \begin{bmatrix}0\\ 5g/7\end{bmatrix}\theta$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\end{bmatrix}$$

For the D.C. Motor, a free body diagram (Figure 3) is used to model it in our simulations. The position control of a D.C. Motor is modeled to have a pole at origin, and another first order pole. The motor's Simulink model was incorporated directly from analysis made in [1].

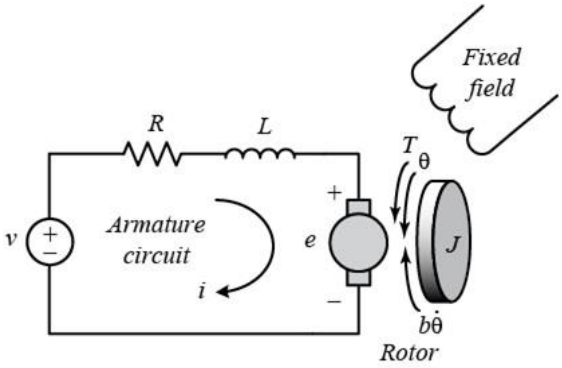

Figure 3. Free body diagram of a DC Motor

B. *Block Diagram*

The system level schematic in Simulink is shown in Figure 1b.

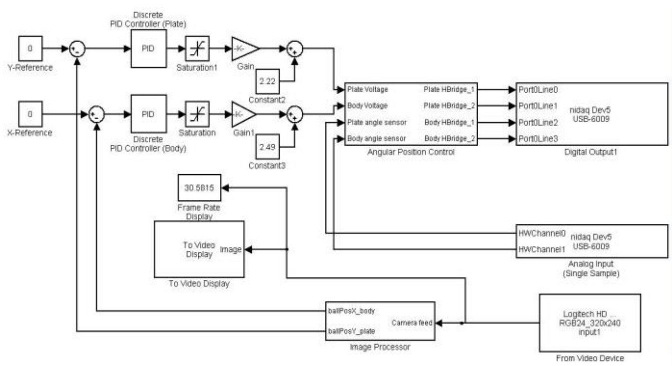

Figure 1(b). System Simulink Diagram

The sensing part of ball's position was done using live image segmentation. For this, we made our plate and all that was visible in the background white-colored, whereas the ball was black. So, with a frame rate of over 30 per second, and a 1 kHz sampling rate of the NI-DAQ card, our webcam produced nicely available data. The overall block diagram of the Image Processor sub-system is shown in Figure 4.

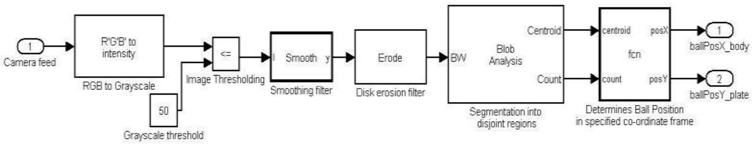

Figure 4. Image processor

This system has the following tasks in order. As soon as RGB image from the camera is digitized, the first filter to apply is to convert it to gray-scale, apply a smoothing filter and erode the image based on a *round* mask of radius 5 pixels. This was based on the resolution of the acquired image that we took to be 320 x 240 (Figure 5).

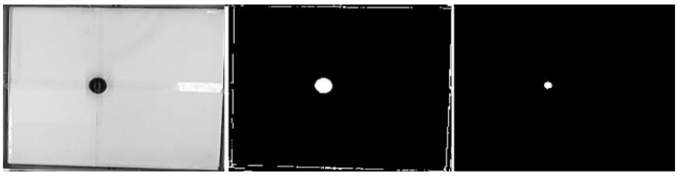

Figure 5. Different steps of image processor

Once we have the black and white image after erosion, a simple blob detection MATLAB script finds the centroid of the white spot i.e. the ball, in the image. This strategy of image segmentation gives nice and accurate results, although it becomes extremely slow when Simulink is running the motors and sensors simultaneously in real-time.

Some important things in this part are positioning the camera exactly front-o-parallel to the plate held levelled. Also, initial calibration of the camera for cropping out the Region of Interest i.e. encompassing only the area which covers the plate is required.

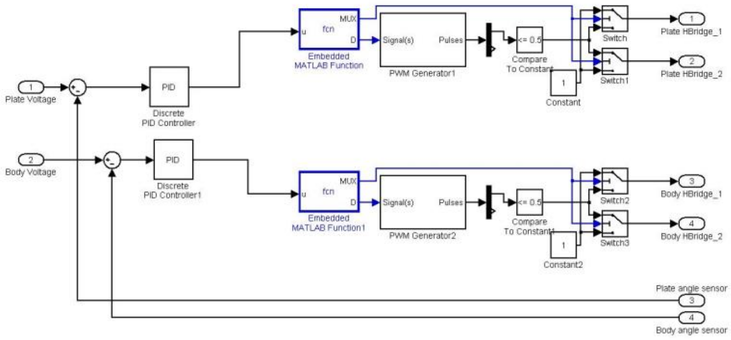

Figure 6. Subsystem showing motor control schematic

For the motor's control, a subsystem (shown in Figure 6) was created. It has an embedded feedback loop that takes input from the potentiometers and adjusts the motor's position based on a PID controller. Two independently driven motor control systems are embodied in this subsystem for the motor's plate and body control (i.e. two degrees of freedom).

C. Controller Design

For Controller Design, we employed simple PID Tuning – for both the inner and outer PID controllers.

III. Simulations & Experiments

A. Camera frame rate

Camera frame rate becomes the bottle neck in our control problem. If the sensor sampling and processing rate is quite large compared to the overall running rate of the control system, it becomes almost impossible to control the system. For these reasons it is very important to run an image processing algorithm that takes minimum time in extracting the position of ball. Although minimum time taking algorithm might not be most efficient and might give many erroneous positions of other objects as well, but considering the need for high speed control of the system, it becomes necessary to extract position of ball in minimum time possible.

Figure 7 represents simulations that have been performed in MATLAB with different sensor sampling times:

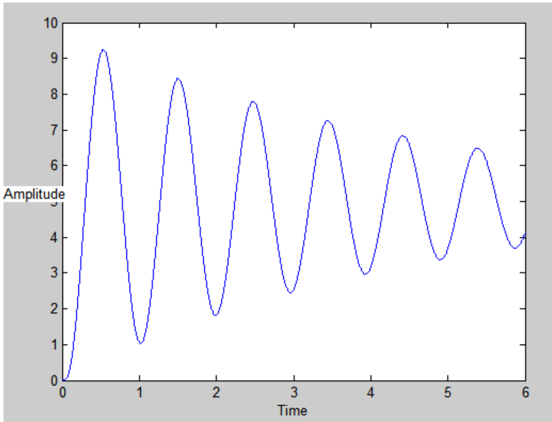

Figure 7a. Step response with frame rate of 0.01s

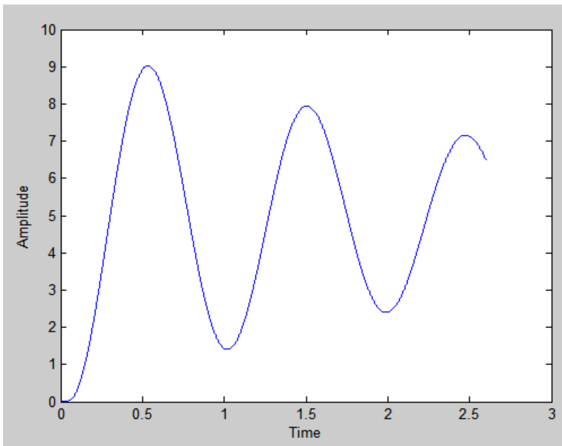

Figure 7b. Step response with frame rate of 0.0001s

From the simulations it can be seen that it deceasing frame rate increases the system time for settling and significantly degrades performance of the system.

A. *Motor Dead Zones*

The position of plane is rotated slightly around the reference position in order not to disturb the system greatly. In small position control of plane, dead zones of motor play significant role. These dead zones limit the amount of control one has in position control of plane. Therefore to control the position of ball precisely on the plane, dead zones must either be accounted for in software or more precise servos should be used for controlling the plane.

B. *Current Amplifiers*

Output of the DAQ card for the motor H bridges control was insufficient to drive enough current for the motors. For this we designed *single BJT Current Amplifiers* to provide enough current to the H Bridges.

IV. Conclusions & Further Work

We have achieved the control of plate orientation (angular motor position). A big jump from the common was to use Image processing algorithms in our sensing approach. Ball position sensing using camera feed, was done by a novel image segmentation strategy. MIMO controller design was another major contribution. For future work, work needs to be done on better frame rate acquisition for faster image processing.

References

[1] Muhammad Hasan Jafry, Haseeb Tariq, Abubakr Muhammad, “Ball Balancing on a Beam”. Technical Report *Digital Control Systems* Fall 2014, SBASSE LUMS.

[2] Zdenek Kalal, Krystian Mikolajczyk, Jiri Matas, "Tracking-Learning-Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409-1422, July 2012.