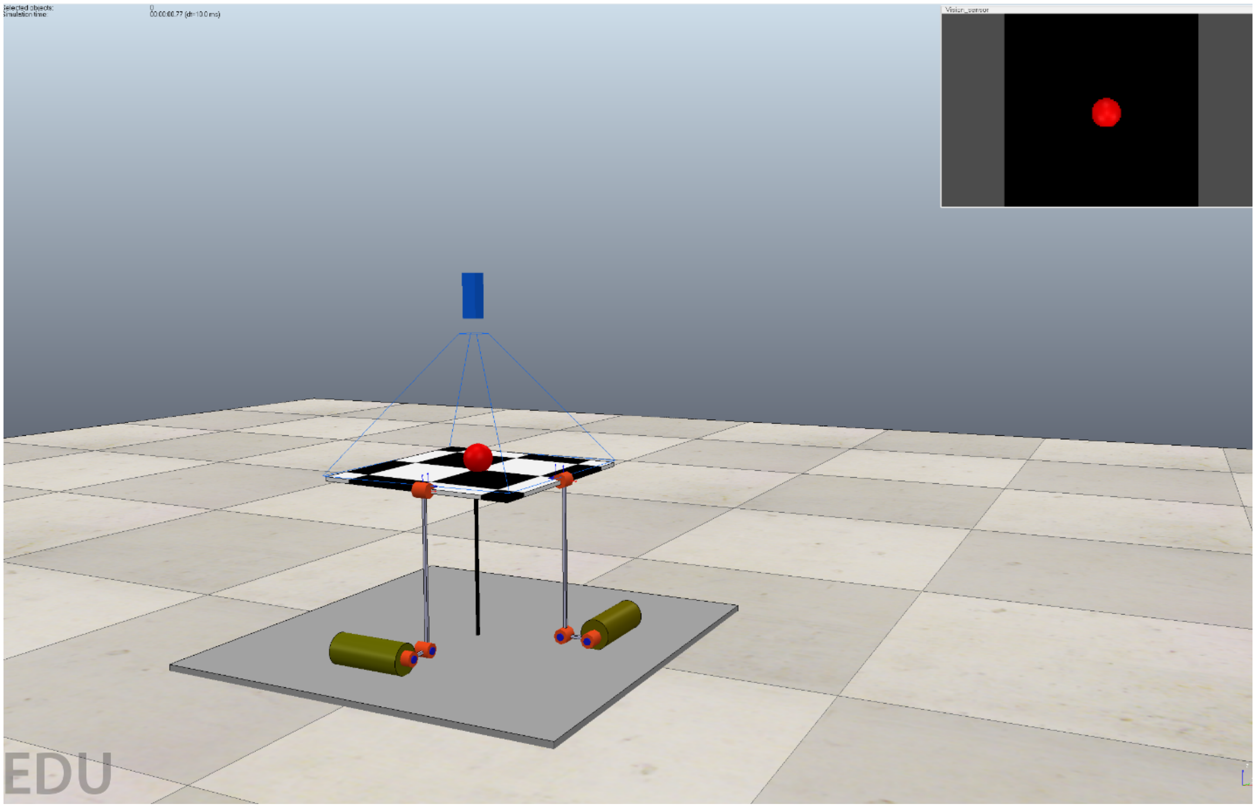
MECA 482 – Control System Design

Ball & Plate Project

May 18, 2020



John Danley

Jonathan Gomez

Marco Machuca

Angel Mendoza

Chen Hung Yang

Department of Mechanical and Mechatronic Engineering and Sustainable Manufacturing
California State University, Chico

Chico, CA 95929-0789

Introduction:

This project will describe a control system for a two-dimensional electromechanical ball and plate system. The system is simulated in Coppelia Sim Edu utilizing a plate that is supported by two revolute joints on the individual sides that control the tilting mechanism in order to balance the ball. The control system described should hold a freely rolling ball positioned on the plate without falling by using feedback from a vision senor to maintain the position of the ball. The goal is to develop the Simulink model that controls the X and Y axis of the plate system in Coppelia Sim Edu.

Modeling:

In order to simplify the system, it is modeled as two independent ball and beam systems. This is possible due to the fact that the angle of the x and y-axis servo's will only affect the movement of the ball in their respective directions. The equations below were used to develop a mathematical model of a single ball and beam system in order to obtain the relationship between the movement of the ball with respect to the angle of the beam (α) and servo arm (theta). With the goal of obtaining the transfer function. Using a free body diagram (**Figure 1**) the forces from the ball's inertia (F_r) as well as the forces due to gravity on the ball (F_g) can be used to develop the equations of motion.

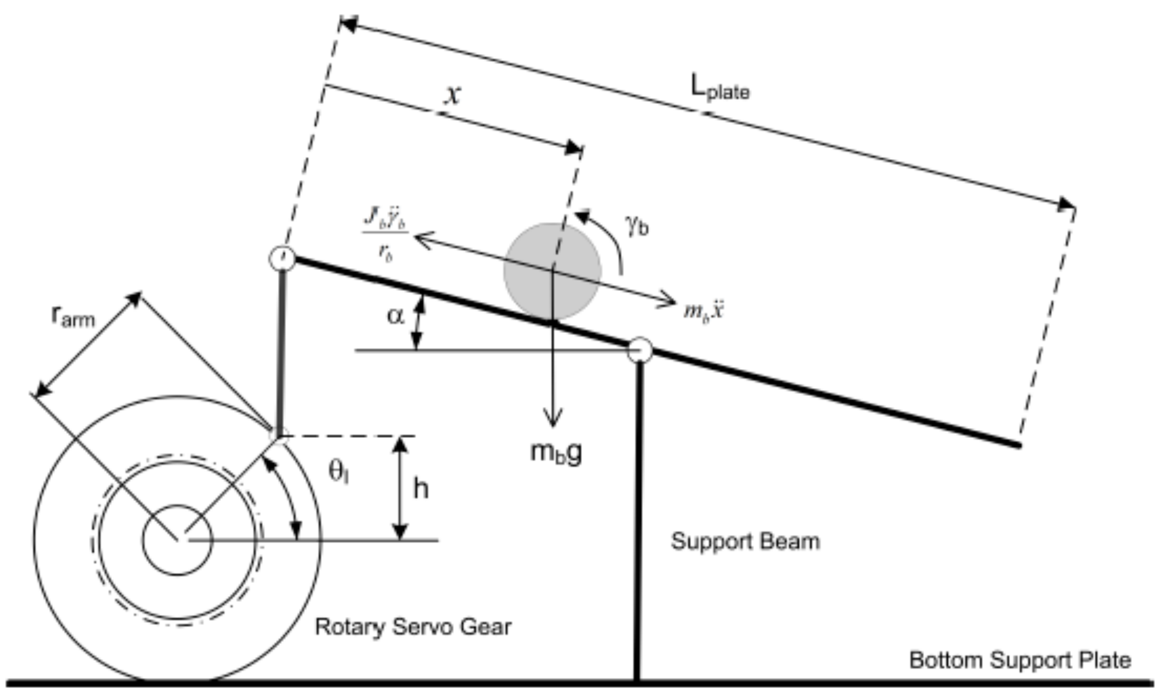


Figure 1: The physical relations of each part of the system

From the sum of forces the following equation can be written.

$$\sum F = m_b a$$

$$a = \ddot{x}(t)$$

$$\sum F = F_r - F_g$$

$$F_r - F_g = m_b \ddot{x}(t)$$

Relating the angle of tilt of the plate (α) to the force on the ball by gravity (F_g) gives the following equation:

$$F_g = m_b g \, sin(\alpha(t))$$

Using the concept of torque, an equation can be formed relating the force of inertia to the radius of the ball (r_b) with the proper torque equation for the rolling ball:

$$F_r = \frac{J_b \ddot{x}(t)}{r_b^2}$$

Given the simulation is for a hollow ball, the equation for the ball's inertia is

$$J_b = \frac{2}{3} m_b r_b^2$$

Therefore, by plugging the ball's inertia into the equation for inertial force (F_r) the new equation is

$$F_r = \frac{5}{3} m_b \ddot{x}$$

Now that F_r and F_g are mapped out, by substitution, the new force equation can be displayed for ball displacement vs lift angle of plate.

$$\ddot{x}(t) = \frac{3}{5} g \, sin(\alpha(t))$$

Through trigonometric relations, the angle displacements of the plate and the motor can be written as so

Plate: $$sin(\alpha(t)) = 2 \frac{h}{L_{plate}}$$

Motor: $$sin(\theta(t)) = \frac{h}{r_{arm}}$$

$$\ddot{x}(t) = \frac{6 g r_{arm} \, sin(\theta(t))}{5 L_{plate}}$$

With small angle approximation and by way of Laplace transform, the transfer function comes out to

$$\frac{X(s)}{\theta(s)} = \frac{6}{5} * \frac{g r_{arm}}{L_{plate}} * \frac{1}{s^2}$$

For a better representation of the open-loop system, the block diagram below (**Figure 2**) illustrates the input of $V_m(s)$ through the servo plant (SRVO2), outputting the servo angle ($\theta(s)$) which in response, adjusts the position of the ball (X(S)) after going through the plant of the balance plate ($P_{bb}(S)$).

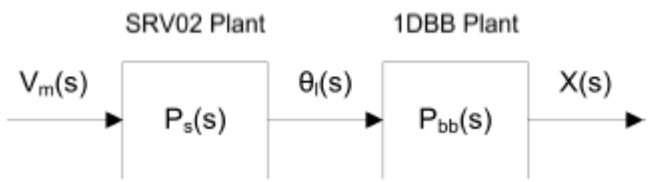


Figure 2: Block diagram of simplified 1 DOF system

Using a SRV02 motor, after proper motor analysis while using the motor's specifications, the transfer function for said motor is stated as so

$$P_s(S) = \frac{K}{s(\tau s + 1)}$$

Where;

$$K = 1.53\left[\frac{rad}{s}\right] \quad \& \quad \tau = 0.0248[s]$$

Using **Figure 2** an even more simplified transfer function can be made to relate the input voltage ($V_m(S)$) to the outputted change in location of the moving ball (X(S)).

$$P(s) = \frac{X(s)}{V_m(s)} = \frac{6}{5} * \frac{g r_{arm} K}{L_{Plate}} * \frac{1}{s^3(\tau s + 1)}$$

Controller Design & Simulations

For this design, the type of controller designed for is a PD controller to be employed to the camera sensor. As the ball moves, the movement will be captured by the sensor, which gives position information to the controller, ultimately sending information out to the servo motor to position the plate accordingly.

With said controller, the following parameters were determined to find the appropriate gains for the controller.

Table 1: Chosen targets for controller parameters

Settling time; T_s (s)	≤ 3.0
Percent overshoot; %OS (%)	≤ 10

The specifications help determine the necessary values for the natural frequency (ω_n) and damping ratio (ξ).

$$\xi = -\ln\left(\frac{{}^{0}\!/_{0}\,OS}{100}\right) * \sqrt{\frac{1}{\ln^2\left(\frac{{}^{0}\!/_{0}\,OS}{100}\right) + \pi^2}}$$

$$\omega_n = \frac{4}{T_s \xi}$$

$$k\_bb = \frac{6}{5} * \frac{g r_{arm}}{L_{plate}}$$

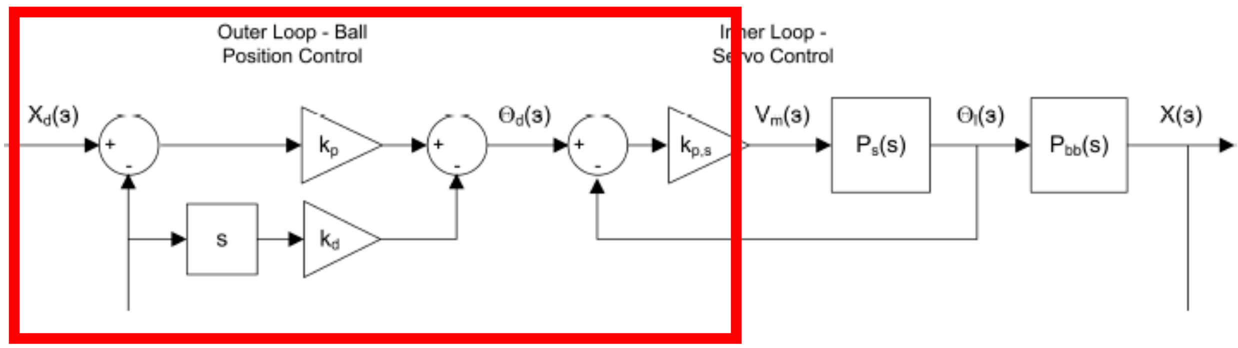


Figure 3: Block diagram used to determine the needed gains for the PV controller

A mathematical representation can be developed to calculate the effects of the flow of information and how the proportional and derivative gains, k_p and k_d, provide the necessary output to properly control the servo motor. Comparing the input position (X(S)) to the desired position ($X_d(S)$), for a desired angle ($\theta_d(S)$) the equation comes out to

$$\theta(S) = \theta_d(S) = k_p\big(X_d(S) - X(S)\big) - k_d s X(S)$$

By substitution into the P(S) transfer function,

$$\frac{X(S)}{X_d(s)} = \frac{k\_bb * k_p}{s^2 + k\_bb * k_d * s + k\_bb * k_p}$$

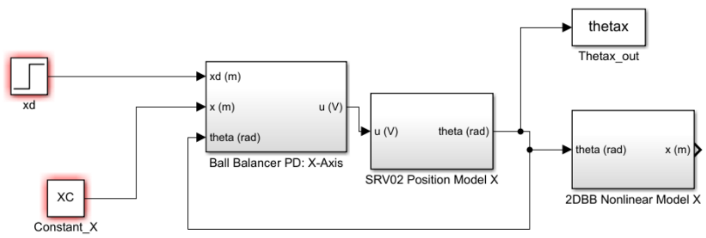
Now that the transfer function has been created to represent the controller's effects on the input, the equation can now be analyzed. From the general 2nd order differential equation, the natural frequency and damping ratio is used to determine the proportional and derivative gains.

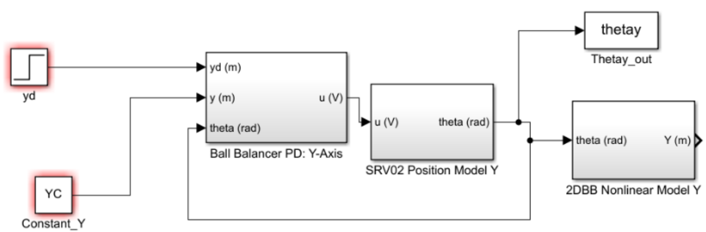
$$k_p = \omega_n{}^2 / k\_bb$$

$$k_d = 2\,\xi * \omega_n / k\_bb$$

In order to establish communication between the vision sensor in Coppelia and Matlab, a child script was written. It serves the purpose of collecting the coordinate position of the ball and returning it to Matlab to perform the calculations necessary to control the position of the motors accordingly.

Appendix A: Simulation Coded





References

Messner, Bill; Tilbury, Dawn. "Control Tutorials for Matlab and Simulink." ctms.engin.umich.edu, University of Michigan, 2017. Web. May 2020

http://ctms.engin.umich.edu/CTMS/index.php?aux=About_Tutorials

Retrieved from Internet, March 13, 2020, https://www.quanser.com/products/2-dof-ball-balancer/

https://www.coppeliarobotics.com/helpFiles/en/b0RemoteApi-matlab.htm