The Balancing Ball

Meca 482- Section 3323
Fall 2019
California State University, Chico

Janelle Bettencourt
Emmily Griner
Isabelle Schicht
Jason Arias Torres

Project Description:

This project models a balancing ball on a plate using two rotary servo base units in conjunction with two degrees of freedom gimbals to keep the ball on the plate. A camera will be used to record the location of the ball and send information to the controller in order to adjust the plate. The inspiration for this project came from a project by Quanser.

This project is important because a system like this would be extremely useful for people with disabilities like Parkinson's Disease. If a smaller version of this system could be built into a bracelet, or something similar, the system could help keep that person's hand still. With their hands still, the person can use silverware and get food to their mouth before it falls off or other everyday activities they could not do before.

Modeling:

Firstly, some assumptions are made about the system to develop a simpler free body diagram. These assumptions include the symmetry of the table which leads to being able to represent both rotary servo motors with one motor present. The next assumption is that the x-axis of the servo angle contributes to only the x-direction movement of the ball. This is the same for the y direction. From these assumptions the free body diagram below was generated.

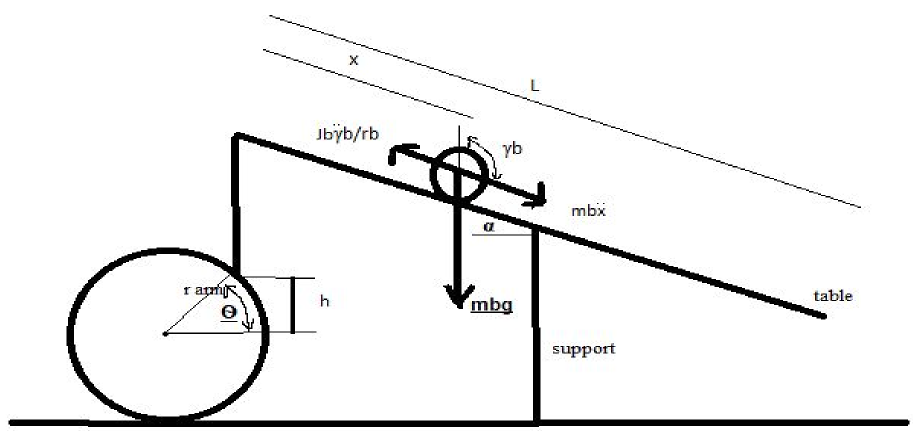

Figure 1: Free-body diagram

Next, an equation of motion is derived from this diagram. The forces for this system are summed and used to derive the equation of motion. Due to the nonlinearity of the system, a nonlinear equation of motion is derived to represent the ball and beam.

$$x''(t) = \frac{m_b g \sin\alpha(t) r^2_b}{m_b r^2_b + J_b}$$

Where:	m_b=	mass of ball	g=	gravity
	α=	angle between 0° and the table	r_b=	radius of the ball
	J_b=	inertia of the ball		

In order to linearize the system, an equation of motion for the position of the ball in reference to the angle of the servo load gear was found.

$$\sin\alpha(t) = \frac{2r_{arm}\sin\theta(t)}{L}$$

Where: r_{arm}= radius of the motor θ= angle between 0° and r_{arm}

This equation and the previous equation are combined to create a linear equation of the motion of the ball shown.

$$x''(t) = \frac{2m_b g r_{arm} r^2{}_b}{L(m_b r^2{}_b + J_b)}\theta(t)$$

Following this, a block diagram is generated for the system. Since the x and y axis are evaluated separately, one block diagram is created for each.

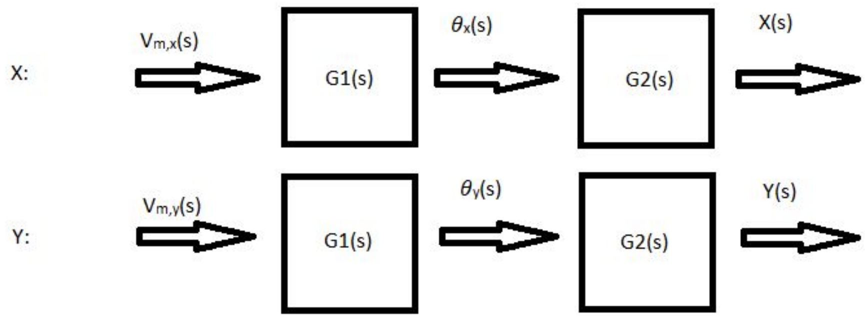

Figure 2: Block diagram for each axis of system

The system has two internal transfer functions that need to be found first. One represents the servo input motor voltage and resulting load angle, $G_1(s)$, and the other represents the angle of the servo load gear to the position of the ball output, $G_2(s)$.

$$G_1(s) = \frac{\theta(s)}{Vm(s)};\ G_2(s) = \frac{X(s)}{\theta(s)}$$

It is found using Laplace transforms that give

$$G_1(s) = \frac{k}{s(\tau s+1)}\ ; G_2(s) = \frac{k_2}{s^2}\ .$$

Since the transfer functions are in series they can be multiplied to get the transfer function for the whole system that represents the servo voltage input and ball displacement output.

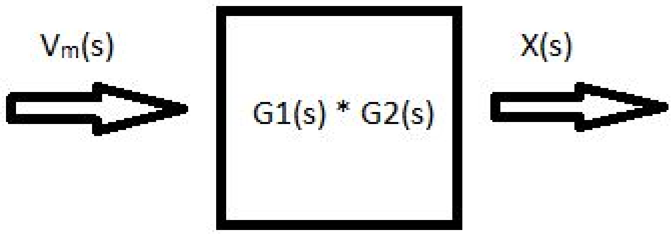

Figure 3: Simplified block diagram (method from Control systems Engineering)

$$G(s) = \frac{X(s)}{V(s)} = \frac{kk_2}{s^3(\tau s+1)}$$

Calibration:

Due to the use of a camera in this system, calibration is required. The camera and ball both observe different coordinate systems. The table is treated as a grid where the resolution (res) of the camera gives the height and width of the square table. Each step in this grid is a pixel given as p_x and p_y . The image below gives this set-up.

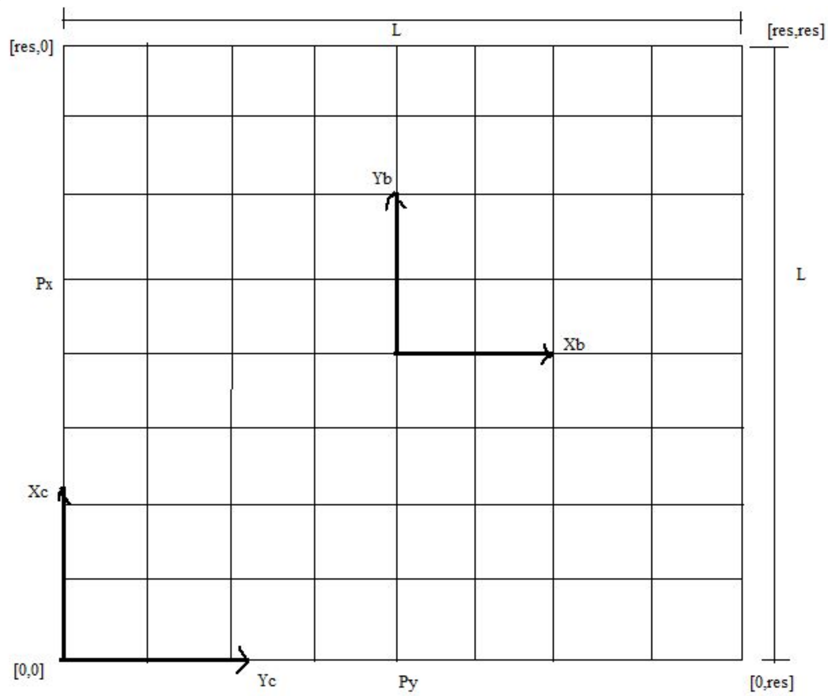

Figure 4: Camera calibration grid system

In order to convert the camera position to the ball position coordinate system in meters, the below equations are used where L is the length of the table, p_x and p_y are the camera coordinate system position, and res is the resolution of the camera.

$$x = L\left(\frac{p_y}{res} - \frac{1}{2}\right);\; y = L\left(\frac{p_x}{res} - \frac{1}{2}\right)$$

Controller Design and Simulation:

In order to design the controller, the time domain requirements were developed: steady state error is =< 5mm, percent overshoot is =<10%, and 4% settling time for =< 3 seconds. A block diagram was created to represent each axis of the system.

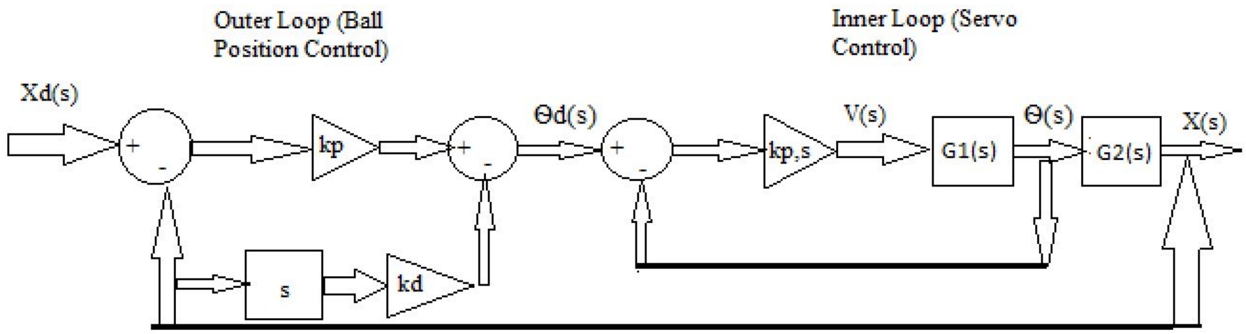

Figure 5: Controller block diagram for one axis

Because this project has 2 DOF, this block diagram is repeated to create a system with 2 axes. The full system can be seen in Figure 6 below. This model was modified from a previous version in order to fit the needs of this project.

X(s) is the measured ball position, $\theta_d(s)$ is the servo load shaft angle which is computed by the outer loop to get the desired ball position $X_d(s)$, and the inner loop maintains the servo position using the proportion gain, $k_{p,s}$. Based on the block diagram, the outer loop is described as $\theta_d(s) = k_p(X_d(s) - X(s)) - k_d sX(s)$. It can also be assumed that the inner loop is ideal. Therefore the closed loop transfer function is found as $\frac{X(s)}{X_d(s)} = \frac{K_2 k_p}{s^2 + K_2 k_d s + K_2 k_p}$, where k_p and k_d are the proportional and derivative gains. Using the percent overshoot and settling time formulas with this transfer function, the natural frequency and damping ratio can be found. They are found to be $\omega_n = 1.94\ rad/s\ and\ \xi = 0.591$. In addition, k_p and k_d are found to be 3.44 rad/m and 2.11 rad/m/s, respectively.

Results:

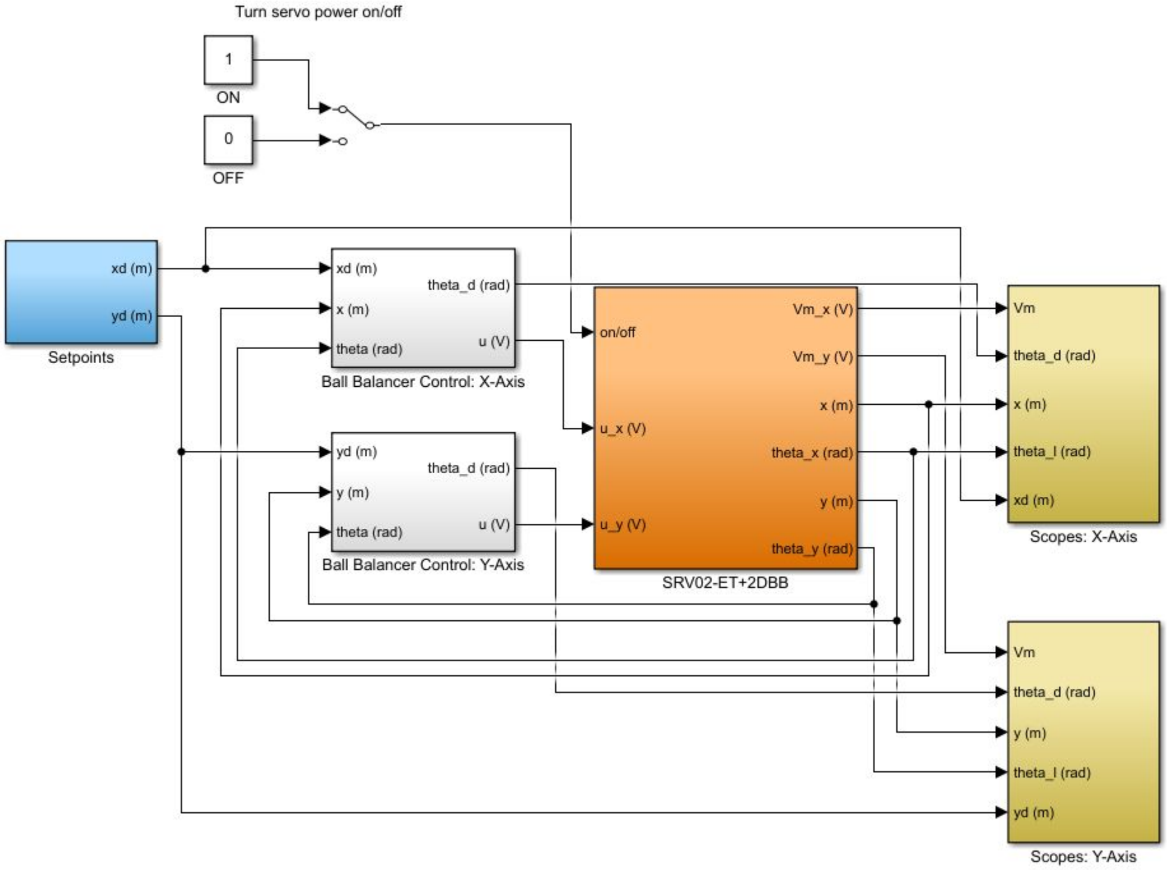

Figure 6: Simulink Model

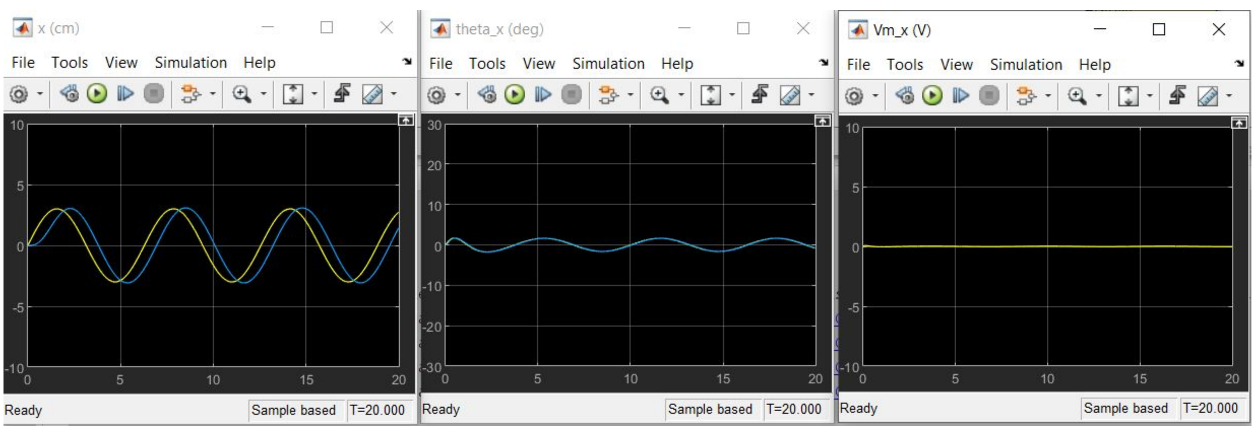

Figure 7: Simulink Output

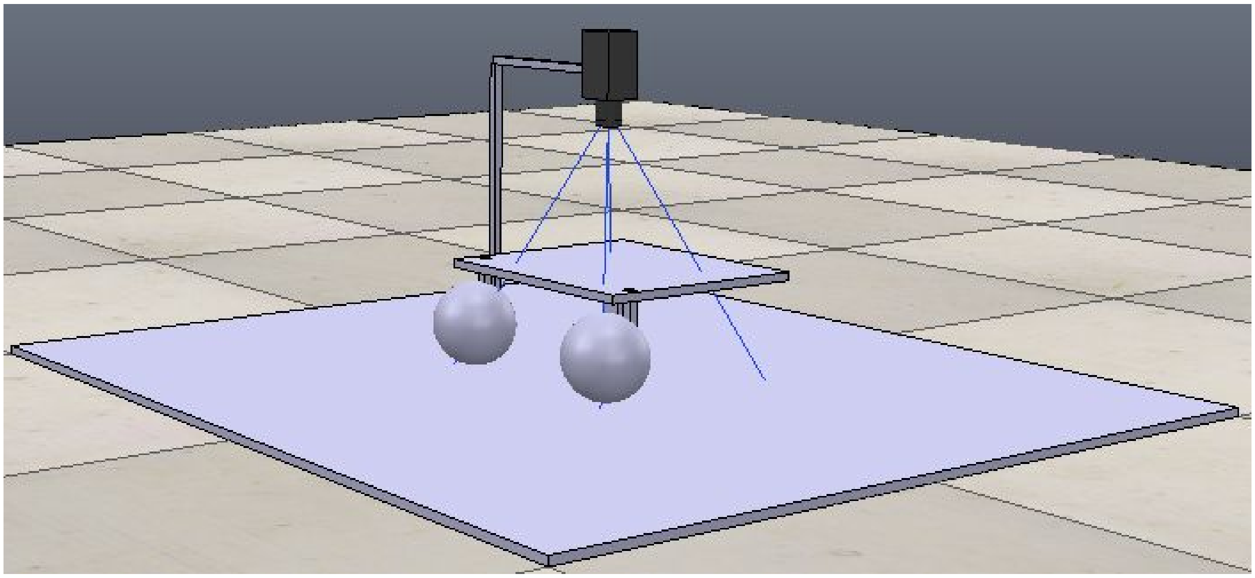

Figure 8: V-Rep Model

This model was created in V-Rep using newly learned skills. The model and the Simulink should be connected allowing for a simulation to occur. Doing this not only verifies the files used to control the system, but it also shows a proof of concept before the physical system is built.

Appendix A: Matlab Code

SETUP_2DBB

Sets the necessary parameters to run the Quanser 2 DOF Ball Balancer experiment.

clear all;
%

SRV02 Configuration

External Gear Configuration: set to 'HIGH' or 'LOW'

```
EXT_GEAR_CONFIG = 'HIGH';
% Encoder Type: set to 'E' or 'EHR'
ENCODER_TYPE = 'E';
% Is SRV02 equipped with Tachometer? (i.e. option T): set to 'YES' or 'NO'
TACH_OPTION = 'YES';
% Type of Load: set to 'NONE', 'DISC', or 'BAR'
LOAD_TYPE = 'NONE';
% Amplifier Gain used: set to 3 when using VoltPAQ-X2
% (or to 1 with VoltPAQ-X1)
K_AMP = 1;
% Amplifier Type: set to 'VoltPAQ'
AMP_TYPE = 'VoltPAQ';
% Digital-to-Analog Maximum Voltage (V)
VMAX_DAC = 10;
%
```

Lab Configuration

Type of controller: set it to 'AUTO', 'MANUAL'

```
CONTROL_TYPE = 'AUTO';
% CONTROL_TYPE = 'MANUAL';
%
```

Control specifications

2DBB Position Control Specifications Settling time percentage

```
c_ts = 0.04;
% Settling time (s)
ts = 3.0; % 2.5 s
```

% Percentage overshoot (%)
PO = 10;
%

System Parameters

Sets model variables according to the user-defined system configuration

[Rm, kt, km, Kg, eta_g, Beq, Jm, Jeq, eta_m, K_POT, K_TACH, K_ENC, VMAX_AMP, IMAX_AMP] = config_srv02(EXT_GEAR_CONFIG, ENCODER_TYPE, TACH_OPTION, AMP_TYPE, LOAD_TYPE);
% Load 2DBB model parameters.
[L_tbl, r_arm, r_b, m_b, J_b, g, THETA_MIN, THETA_MAX] = config_2dbb( );
% Load model parameters based on SRV02 configuration.
[K, tau] = d_model_param(Rm, kt, km, Kg, eta_g, Beq, Jeq, eta_m, AMP_TYPE);
%

Filter Parameters

2DBB High-pass filter in PD control used to compute velocity Cutoff frequency (rad/s)

wf = 2 * pi * 2.5;
%

Calculate Control Parameters

```
if strcmp ( CONTROL_TYPE , 'MANUAL' )
    % Calculate Balance Table model gain.
    K_bb = 0;
    % Design Balance Table PV Gains
    kp = 0;
    kd = 0;
    %
elseif strcmp ( CONTROL_TYPE , 'AUTO' )
    % Calculate Balance Table model gain.
    [ K_bb ] = d_2dbb_model_param(r_arm, L_tbl, r_b, m_b, J_b, g);
    % Design Balance Table PD Gains
    [ kp, kd ] = d_2dbb_pd( K_bb, PO, ts, c_ts );
end
%
```

Display

disp(' ');
disp('Balance Table model parameter: ');

```
disp( [ '   K_bb = ' num2str( K_bb, 3 ) ' m/s^2/rad' ] );
disp( 'Balance Table Specifications: ' );
disp( [ '   ts = ' num2str( ts, 3 ) ' s' ] );
disp( [ '   PO = ' num2str( PO, 3 ) ' %' ] );
disp( 'Balance Table PID Gains: ' );
disp( [ '   kp_bb = ' num2str( kp, 3 ) ' rad/m' ] );
disp( [ '   kd_bb = ' num2str( kd, 3 ) ' rad.s/m' ] );
%
```

```
Balance Table model parameter:
   K_bb = 1.09 m/s^2/rad
Balance Table Specifications:
   ts = 3 s
   PO = 10 %
Balance Table PID Gains:
   kp_bb = 3.45 rad/m
   kd_bb = 2.11 rad.s/m
```

Published with MATLAB® R2018b

Appendix B: References

Bayar, Gokhan. “Theoretical and experimental investigation of backlash effects on a 2-DOF robotic balancing table.” *Dynamics of Mechanical Systems*,vol. 23, no. 1, 2017

Nise, Norman S. *Control Systems Engineering*. Wiley, 2019.

Quanser. *2 DOF Ball Balancer*. Quanser Inc., 2013.