TEL AVIV UNIVERSITY אוניברסיטת תל אביב

Faculty of Engineering

Lab Brief – Ball & Plate Stabilizer

1. Lab Purpose:

- Modeling of a high-order mechanical system.
- Design of a controller using classical methods.
- Implementation of the controller on the actual system.
- Comparison of a modeled frequency response against a measured one.

2. System Description:

2.1. Mechanical system:

1. System base.
2. Camera stand.
3. 640x480 pixels camera.
4. White plastic plate.
5. SRV-02 motor module.
6. Orange table-tennis ball.

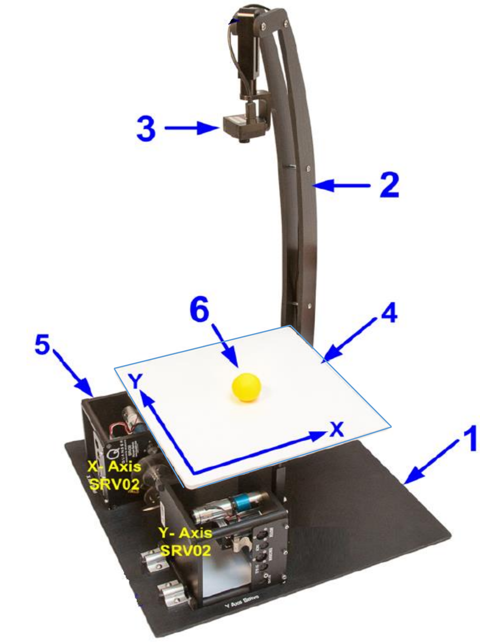

Figure 1: Mechanical system

2.2. Data acquisition:

7. PC connectors.
8. Encoder signal inputs.
9. Analog amplifier outputs.
10. ENABLE digital outputs.

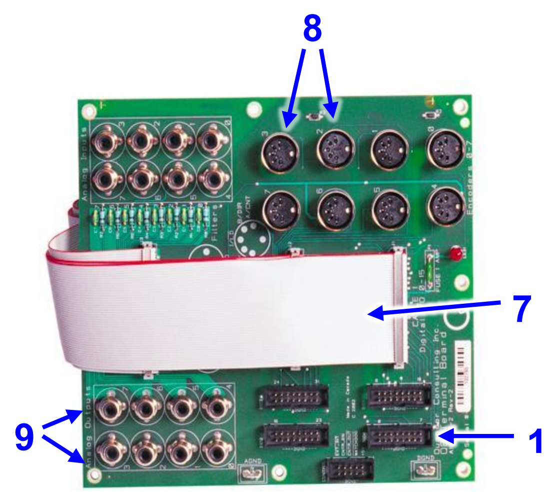

Figure 2: Quanser Q4 DAQ board

2.3. Current amplifier:

11. Analog inputs from the DAQ card.
12. Motor voltage outputs.
13. ENABLE digital inputs.
14. Emergency power cut-off connector.

Figure 3: Quanser VoltpaQ-X2 amplifier

2.4. Explanations:

The system consists of a heavy metal base, which is connected to an arc-like camera stand. There you would find the camera, which serves as the feedback sensor for our orange ball. Thus, the camera is facing downward towards the plate.

The plastic plate is connected to the base by a ball-joint, generally allowing for 3 Degrees-of-freedom, the yaw angle being fixed by the motor arms.

Beneath the tray are two SRV-02 modules, each containing a motor, an encoder, and a gear transmission. Changing the angles of these motors raises/lowers parts of the plate, allowing for pitch and roll angle movement.

Each module has two cable connections – one is used to read the encoders (and thus the angle state), and the other one supplies power to the motors.

The DAQ card reads the angles, transfers them to the PC which calculates the control signals, and sends them back, through the DAQ, to the amplifier. The amplifier supplies the power required to actuate the motors.

The ENABLE cord is strictly functional and does not play a part in the control loop.

3. Theoretical Background.

3.1. mathematical modeling.

During the modeling process, we will treat the system as 2 non-conjugated systems, as each one controls the ball movement along 1 axis. We assume that movement in the X axis does not affect the movement of the Y axis.

The diagram in figure (4) depicts the balance of forces upon which the system's dynamic equations are based. These depend on the ball movement, X, and the plate angle, α.

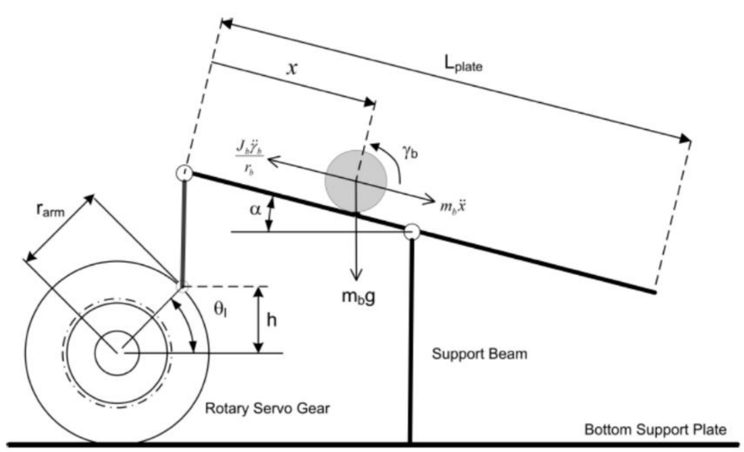

Figure 4: Free-body diagram of the system, surrounding the ball

Signs agreement:

- Positive motor voltage will cause the motor angle (θ) to rotate in the positive direction, that is, Counter-clockwise. The result will be the movement of the tray angle, α, also in the positive direction, meaning that the tray's left-side will move upwards.
  That will cause the ball to move to the right.
  In other words: V_m>0 → $\dot{\theta}$>0 → $\dot{\alpha}$>0 → $\dot{x}$>0 .
- The ball's position will be considered as zero when it is in middle of the plate.
- The θ angle will be considered as zero when the motor arm is parallel to the ground (α=0).

According to Newton's laws, we could derive a force-equilibrium on the ball in the following manner:

$$(1)\ m_b\ddot{x} = \sum F = F_{x,g} - F_{x,i}$$

While m_b is the ball mass, $F_{x,g}$ are forces generated by gravity, and $F_{x,I}$ are forces that originate from the ball's inertia. For the purpose of modeling, we will neglect the natural dampening forces that exist in every system. In order for the ball to stay in place, the sum of these forces must be equal to zero. We can derive from figure (4) that the gravitational force applied over the center of the ball is equal to: $F_{x,g} = m_b \cdot g \cdot \sin(\alpha)$.

In contrast, the inertial force which acts upon the ball is: $F_{x,i} = \frac{\tau_b}{r_b}$, that is, the torque bestowed upon the center of the ball as a result of friction with the plate, divided by the ball radius. We could express the torque to be: $\tau_b = J_b \cdot \ddot{\gamma}_b$, while J_b if the ball's moment of inertia and $\ddot{\gamma}_b$ is its rotational angle. We could further simplify the expression using the linear equivalent: $x = r_b \cdot \gamma_b$. Substituting this equivalence into equation (1) will produce: $m_b\ddot{x} = m_b \cdot g \cdot \sin(\alpha) - \frac{J_b \cdot \ddot{x}}{r_b^2}$ (check this!). isolating $\ddot{x}$ will produce:

$$(2)\ \ddot{x} = \frac{m_b \cdot g \cdot r_b^2}{m_b \cdot r_b^2 + J_b}\sin(\alpha)$$

Now we would want to find the connection between θ and α. This is because we are only able to control the voltage entering the motor, yet a relation exists only between V_m and θ. By observing figure (4), we would find that it is possible to express the relation between the two angle in the following manner: $\sin(\alpha) = \frac{2 \cdot r_{arm}}{L_{plate}} \cdot \sin(\theta)$, which will allow us to substitute that into equation (2) and get, after a linearization of $\sin(\theta)$ around zero:

$$(3) \quad \ddot{x}_{(t)} = \frac{2 \cdot m_b \cdot g \cdot r_b^2 \cdot r_{arm}}{L_{plate}(m_b \cdot r_b^2 + J_b)} \cdot \theta_{(t)}$$

*The rational expression prefixing θ is made of constant parameters, all of which we can combine under 1 name: K_{bb}.

3.2. Deriving the Transfer Function.

Here we will as well make calculations for a 1-axis system, while control will be utilized for each axis individually, and under the assumption that movement in one axis does not affect the other one. In addition, we will assume that both motors are identical (which is not the case in real life).

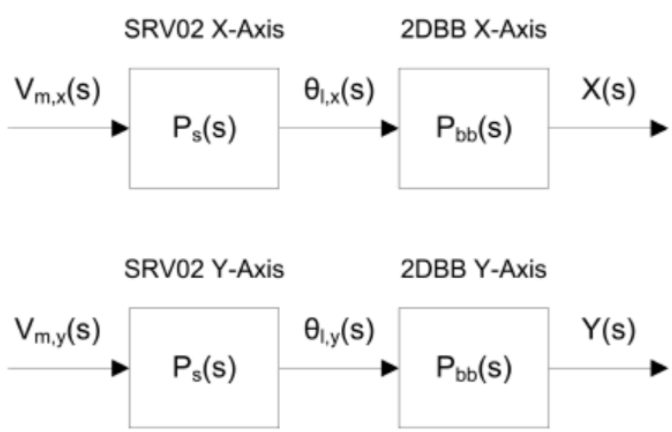

Figure 5: Input/output transfer for 2 individual axes

We could say that our controlled process (Plant) looks like this:

$$(4) \quad P_{(s)} = P_{bb(s)} \cdot P_{s(s)} = \frac{X_{(s)}}{\theta_{(s)}} \cdot \frac{\theta_{(s)}}{V_{m(s)}} = \frac{X_{(s)}}{V_{m(s)}}$$

We know from other lab experiments that: $P_{s(s)} = \frac{K}{s(\tau \cdot s+1)}$, while parameters K, τ are known constants. Taking equation (3) and applying the Laplace transform, will give the following result: $P_{bb(s)} = \frac{K_{bb}}{s^2}$. Therefore, the overall process can be written as:

$$(5)\quad P_{(s)} = \frac{X_{(s)}}{V_{m(s)}} = \frac{K_{bb} \cdot K}{s^3(\tau \cdot s+1)}$$

And this is the transfer function between the position of the ball to the voltage applied to the motor.

3.3. Feedback Signal Coordinate System.

The location of the ball in Cartesian coordinates is measured using the camera which is located above the plate. The camera is mounted in such a way that it is parallel to the plate's hinges (More or less. To enable free movement of the plate, it is not tightly fastened to its suspensions).

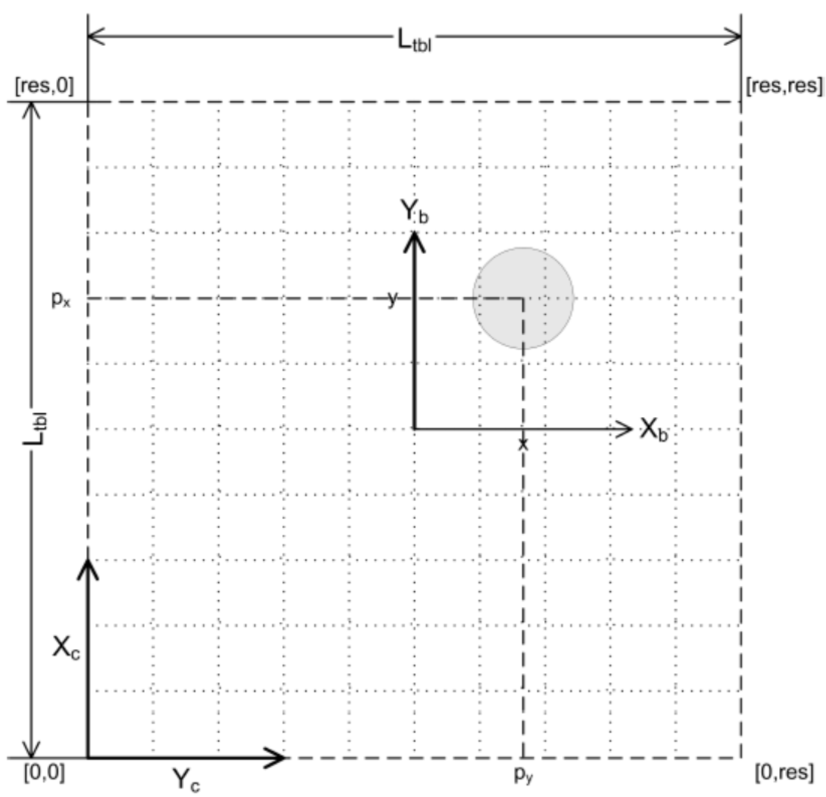

Figure 6: Transition between the camera and model coordinate systems

As depicted in figure 6, the camera calculates the position of the ball throught a pixel-based system, as the data output is the pixel index (p_x, p_y). That will translate to metric position using the camera resolution, res (that is the amount of pixels in every axis). We would want to work in a square-based resolution (i.e. the same amount of pixels for x and y axes).
NOTE – the camera's pixel index is not intuitive, meaning that the x and y axes have changed positions (see figure 6).

4. Controller Design

The control model's block diagram is made of an inner and outer loop:

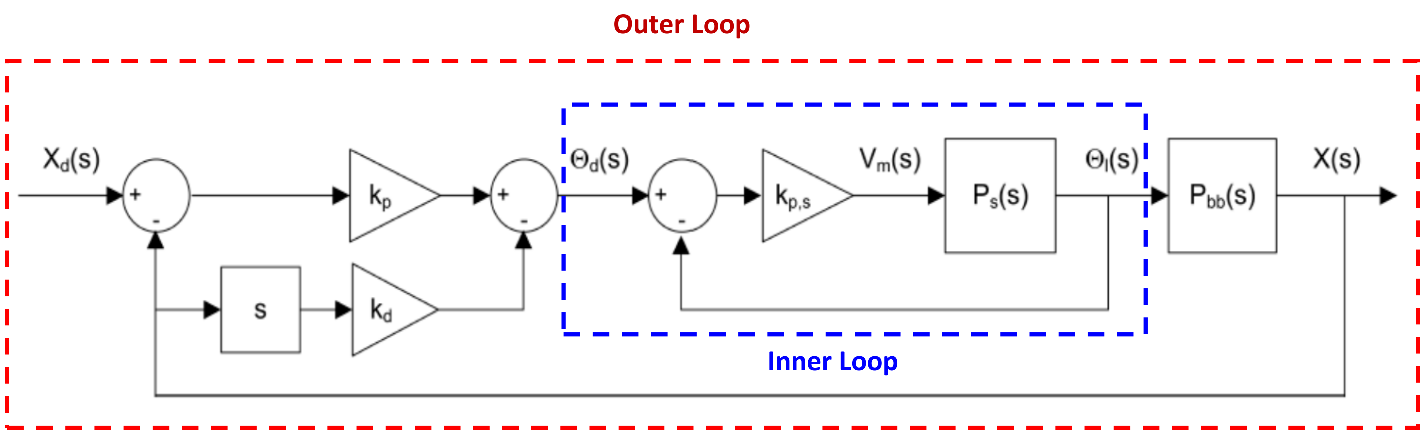

Figure 7: Block diagram of a single system axis

The inner loop has to do with the motor transfer function. It only applies proportional control, and its sole purpose is to bring the motor shaft to the desired angle. If it converges to the desired value fast enough, then we can ignore it and assume that θ_d (desired angle for controlling the ball) is equal to θ_i (the actual angle of the motor). Thus we shall name that variable simply as θ. The outer loop applies PD control, without the integral term.
As we can see from the diagram, variable θ is given by the expression:

$$(6)\quad \theta_{(s)} = k_p\left(X_{d(s)} - X_{(s)}\right) - k_d \cdot S \cdot X_{(s)}$$

Design specifications:

- Settling time (4% corridor): $t_s \leq 3.0 \quad [\text{sec}]$
- Overshoot response: $OS \leq 10\%$
- Steady-state error: $|e_{ss}| \leq 5 \quad [mm]$

Pre-Lab Questions

1. The loop at figure 7, and the content of equation (6), depict a situation where the derivative term applies only on the feedback signal and not the whole error signal. Such a way to close the loop is called "Derivative on Measurement". Explain what it is good for, and what might happen in some occasions where we derive the entire error signal.
2. Find the expression for the closed outer-loop transfer function, $\left[\frac{X_{(s)}}{X_{d(s)}}\right]$. Ignore the process of P_s and only take into account the P_{bb} process.
3. The ping-pong ball can be modeled as a hollow sphere with mass m. Derive an expression for a hollow-sphere moment of inertia (can use the Wikipedia), substitute it into equation (2) instead of J_b and simplify the expression. How would the system dynamics be affected if the ball mass was doubled? And what if its radius was doubled? What important detail can we learn about the system from these facts?
4. The prefix of θ in equation (3), along with the simplification done in the previous question, is actually the parameter K_{bb}. Calculate its numerical value using the table found in appendix 1 at the end of the brief.
5. Use the transfer function found in question 2 and the value of K_{bb} in order to design a PD controller which satisfies the demands given at the end of section 4. Present the entire calculation process and not the final answer only.
6. Show a Bode diagram (using Matlab) of the closed loop along with the controller. Show the cutoff frequency and stability margins.
7. Look at the explanation in section 3.3. and at figure 6. Derive expressions for two functions: $y = f(p_x, L_{tbl}, res)$, $x = f(p_y, L_{tbl}, res)$, in such a way that given the arguments, we could convert the ball position from discrete camera pixels (measured from the plate's corner) into the model coordinate system (in cm, measured from the middle of the plate).
8. Assume that each axis has 440 pixels, and that the plate length is 27.5 [cm]. check your function – what output value is received for $(p_x, p_y) = (0,0)$? And for (160,220)?

5. Lab Process

First thing you do is go to the documents\MATLAB folder, make a copy of the folder “2DOF Ball Balancer”, and rename it to your names.

You should be gentle with the system components, especially with the camera. If you are unsure of anything, call the instructor.

5.1. Camera Calibration.

Gently remove the lens cover. On the desktop, click the icon “Point Grey FlyCap2”.
In the upcoming window, select “Firefly MV FMVU-03MTC” and click OK.
At the window corresponding to figure 8, at the top menu choose
Settings → Toggle Camera Control Dialog.

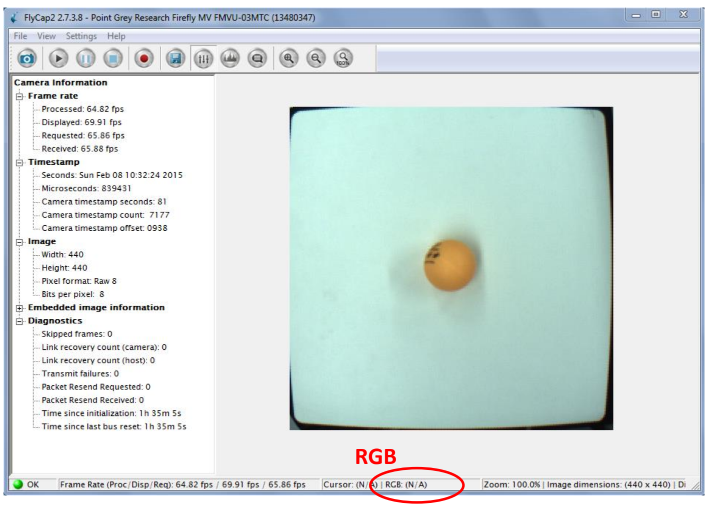

Figure 8: Camera preview screen

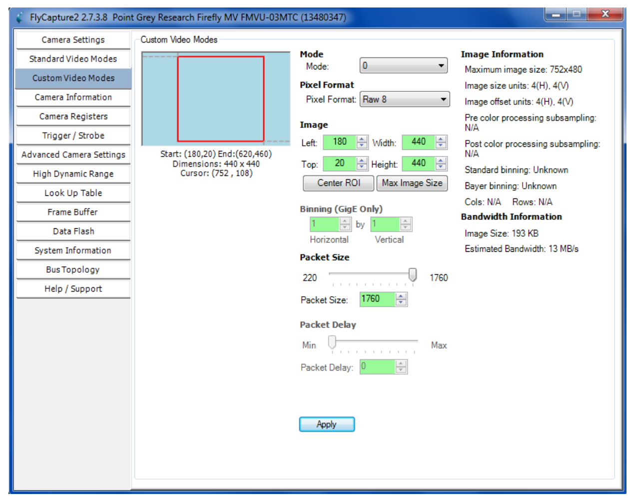

Figure 9: Camera settings window

In the upcoming window, disable the ✓ sign from the Shutter and Gain rows.
Change the shutter value to ?? and the gain value to ?? (it is fine If the values are not exact). If the image still flickers, call the instructor for help. If not, on the left menu click “Custom Video Modes”. In the upcoming window, insert the following parameters: Width=Height=440, Left=120, Top=20, and click “Apply”.
If the camera field-of-view does not (about) match the outline of the plate, similar to figure 8, then you should change the numbers until you reach that point. In any case, the Width and Height values much remain identical (square region).
Ask the instructor for permission to proceed before your final choice.
Write down and keep the values you ended up with.

The image processing algorithm being used in the system is based on following a particular color, that would be the color of our ball. Theoretically, the computer is able to follow any color which isn't white or black. We should supply the desired (R,G,B) value so that the computer would know which color to track. In order to do so, you should open the window that streams the video from the camera (figure 8).
One of the lab partners should hold the ball in place from its side using 2 fingers, while the other one should put the mouse cursor over the ball center and write down the RGB values. Note that these values fluctuate so you need to take an estimated average. This should be done in each corner and in the middle of the plate, due to different lighting conditions across the plate. In the end, calculate an average of each measurement, and write down your result.

Now exit the software and restart the computer!

This is important so that Matlab could read the needed values without conflicting the the 3rd party software. After restarting, enter Matlab and go to the folder you created when you first started the lab. Open the Simulink model: "q_2dbb_camera_calib.mdl".

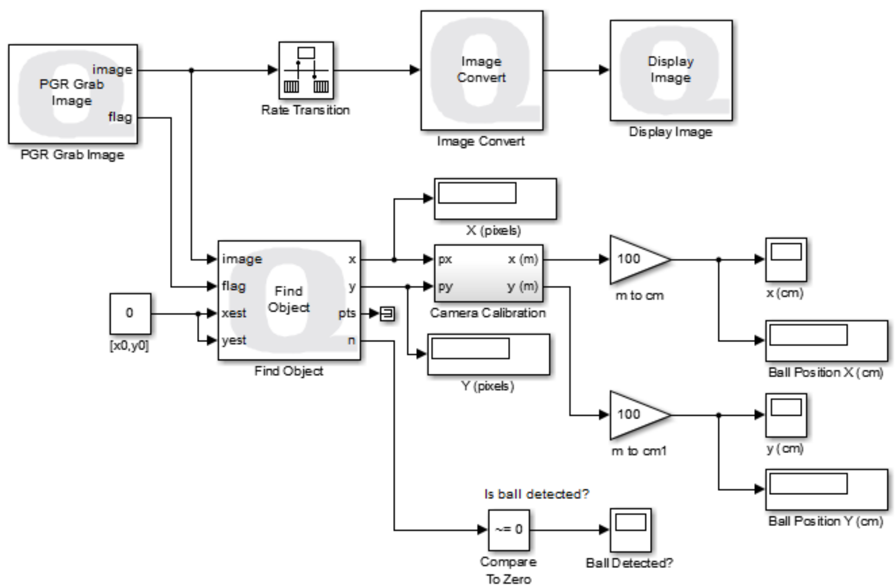

Figure 10: Camera calibration Simulink model

Open the block "PGR Grab Image" and insert the parameters taken about the image size and position. **Do not change any other parameters.** Now open the block "Find Object" and insert the RGB values taken about the ball's color. **Again, do not change any other parameters.**

At the top menu, click: Quarc → Build. After the compilation is finished, click on: Connect to Target, and then: Run. Open the “Display Image” block and make sure that the received image roughly covers the plate area. Now move the ball along the plate (using 2 fingers, as before) and make sure that the ball position input is shown at the scopes, without too much noise. Check which location on the plate translates to (0,0). If there is any problem (i.e. if the position hangs at [-13,-13]), call the instructor.
If everything runs smoothly click: Stop, exit the model and move on.

5.2. Simultaion.
In Matlab, run the script: “setup_2dbb.m”, and then use the commend line to insert the following variables: kp, kd, K_bb (case-sensitive!).
Now open the Simulink model: “s_2dbb.mdl” and you should see the following diagram:

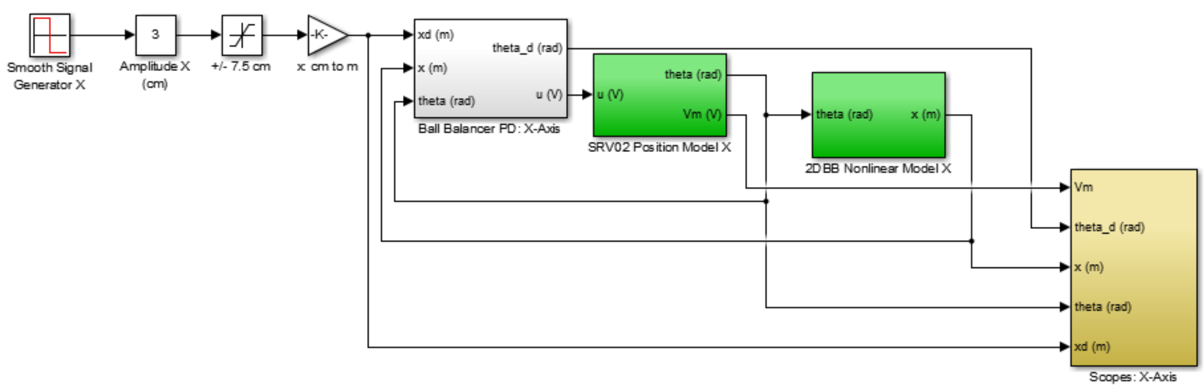

Figure 11: Process simulation model

Run the simulation and check the scopes for the results. Does the output meet the design requirements? If not, change kp and kd until it does. **In addition,** the motor output voltage, V_m, should not go over the value of 10 [V]. There is a dedicated scope for monitoring that voltage.
After meeting all requirements, save a plot of the position, angle and voltage versus time. You may use the following commands:

```
plot(data_vm_x(:,1),data_vm_x(:,2));
plot(data_x(:,1),data_x(:,2), data_x(:,1),data_x(:,3));
plot(data_theta_x(:,1), data_theta_x (:,2), data_theta_x (:,1), data_theta_x (:,3));
```

It is recommended to save the plot both as a JPG file and a Matlab FIG file so that one could modify the plots at home, should the need occur. You will need to include these graphs and control coefficients for the post-lab report.

5.3. Real Experiment, Testing the Controller.

Close all the models opened so far and open the Simulink model: “q_2dbb.mdl”.

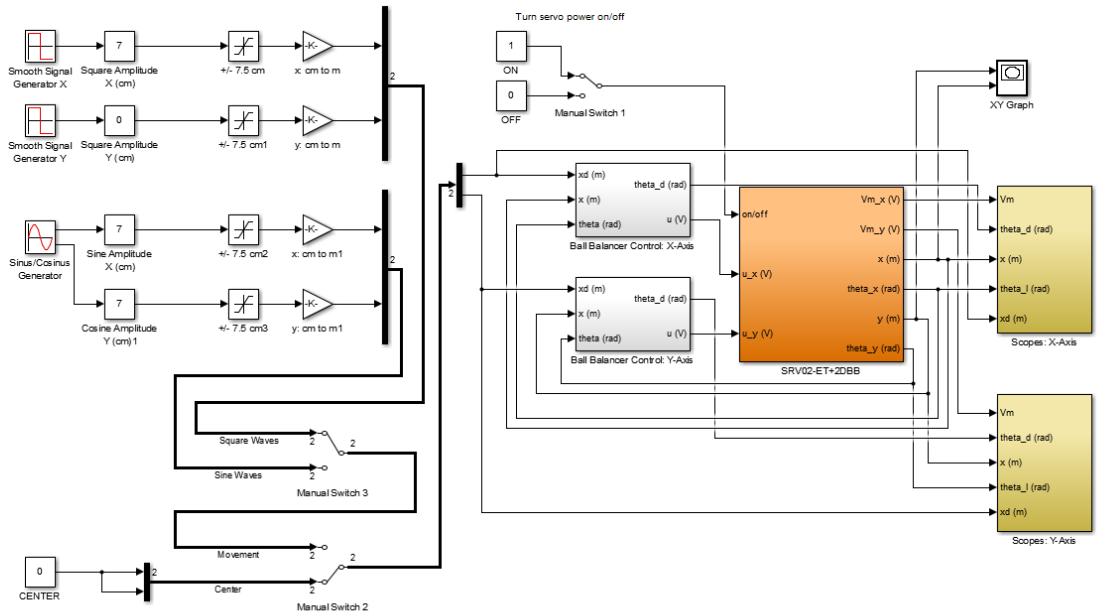

Figure 12: Real experiment Simulink model

The model isn’t as complicated as it may seem. On the left side we have different input signals, routed through switches. The grey blocks at the center contain the control loop. The orange blocks is responsible for collecting the feedback and sending control signals back, and the blocks on the right contain all the scopes.

Before we begin, make sure that:

- Manual Switch 1 at the top of the model is switched to “ON”.
- Manual Switch 2 on the left is switched to “Center”.
- Manual Switch 3 is switch to “Square Waves”.
- The value inside the “Square Amplitude X” block is set to 7 [cm].
- The value inside the “Square Amplitude Y” block is set to 0 [cm].
- The values of both X and Y Sine/Cosine amplitudes are set to 7 [cm].
- The plate is fully horizontal, in such a way that the ball does not roll off of it.
- The yellow/red emergency stop button is at an easily reachable place.
- Inside the orange block, insert the correct parameters you found in the previous section into the "PGR Grab Image" and "Find Object" blocks.

One should use caution while activating the system. If by any chance the system runs out of control, you should immediately press the emergency button.

Build (compile), connect to target and run the model. The ball should center itself on the plate. Double-click Manual Switch 2 to change the input into a square wave.
The ball should be moving from side to side along the X axis (forward/backwards on the tray coordinate system). After a few runs back and forth, double-click Manual Switch 2 again to return the ball to the center of the tray, and then press "Stop". Note, the scope resets itself after the signal reaches the end of its range, so try to do this entire process within the range limit (before the signal reaches the end of the scope).
If the output looks reasonable (not too slow or too fast), plot a graph of the ball position versus time. Save it for the post-lab report.
Repeat the above process, after changing the "Square amplitude Y" value to 7.
Do not forget to change Switch 2 to "Center" each time before stopping the model.
The ball should be moving in a diagonal line (simultaneous X and Y axes movement).
Save a plot of the said process (only for X).

5.4. Real Experiment, Sinusoidal Signal.
Assuming Switch 2 points to "Center", move switch 3 from "Square Waves" into "Sine Waves". Make sure that inside the block "Sinus / Cosinus Generator", the frequency is 0.2 [Hz] and that the amplitude is 1 (remember! It is multiplies by 7 along the way).
As usual, click "Run" and move Switch 2 to "Movement". Wait a few cycles, return Switch 2 to "Center" and stop the model. Plot and save the results. From the graph, measure the amplitude (peak-to-peak) of the input and output, and the time difference between two consecutive peaks, in both axes. Calculate the phase lag and gain between the input and the output.
Repeat this section numerous times, raising the frequency by 0.2 [Hz] in every run, up until (and including) 1.4 [Hz]. Put your results in the following table (you may use Excel):

Frequency [Hz}	Frequency [rad/sec]	Amplitude [cm]	Gain [dB]	Time delay [sec]	Phase lag [deg]
0.2					
…					
1.4					

Create such a table for both X and Y axes.

6. Post-Lab Report Requirements

Report format:

1) Lab purpose (in your own words).
2) Short theoretical background summary: model, calculations, parameters.
3) Course of the experiment summary: what did you do at the lab and why.
4) Answers to the post-lab questions.
5) Conclusions: what did you understand or learn.

Post-Lab Questions:

a. Look at the Theta plot from section 5.2. : What does each graph represent? What are the differences between them? If we take the inner-loop requirements into account, is it safe to say that we made the right assumption (ignoring the inner loop)? Explain.

b. Show (on different figures) the results of running the simulation, running the experiment with X axis only, and running the experiment with both axes. Answer:
   - What are the differences between the simulation and real experiment? What could be the reasons for these differences?
   - Is the graph resulting from running the system along the x axis alone identical to the graph from running the system along both axes together? Is the assumption that the axes are non-conjugate correct? What are the advantages / disadvantages of that approach?

c. Use the table made in section 5.4 to build an approximation of a Bode plot of the system, in the measured interval, for each axis. Find the cut-off frequency and the stability margins (hint: you may use polynomial approximation using Excel).

d. Present the resulting plots on a logarithmic scale. Are there similarities between these plots and the plots generated for the model transfer-function in the pre-lab questions?

Appendix 1: System Constants

The following table contains the needed values in order to model the system:

Notation	Value	Units
r_{arm}	2.54	[cm]
L_{table}	27.5	[cm]
r_b	4	[cm]
m_b	0.003	[kg]