ASEIL Quanser Controls Lab, Bu-212E

Running 2D Ball Balancer Experiment

Contents

Purpose

The 2D Ball Balancer is a two degree of freedom motion platform and a camera. The overhead camera tracks the position of the orange ball on a white background. Using this track, the platform is tilted in the X and Y-axis to attempt to maintain the ball in the commanded position.

The goal of this experiment is to design, implement and test a Proportional-Integral-Derivative (PID) controller that stabilizes the ball to a desired position on the balance plate.

This document explains the setup process for the 2D Ball Balancer.

Physical Setup

The 2D Ball Balancer (2DBB) should be setup, but if it is not, the experimenter will require the following components and they should be assembled as depicted.

ID #	*Component*	*ID #*	*Component*
1	Calibration Base	10	Camera stand
2	SRV02 for X-Axis	11	Camera height adjuster
3	SRV02 for Y-Axis	12	Camera
4	Coupling screw	13	Camera FireWire connector
5	Lever arm	14	Camera Holder
6	Lever arm gimbal	15	Camera Holder Extension
7	Plate	16	Camera Holder Screw
8	Support arm	17	Ball
9	Support arm gimbal		

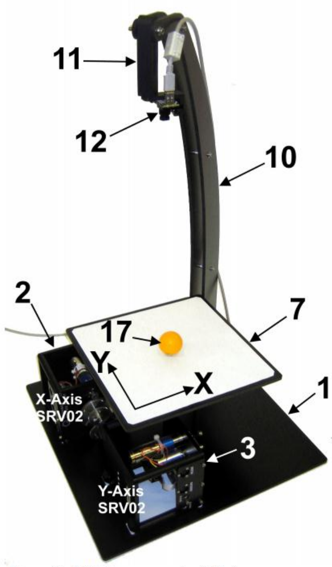


Figure 2: 2DBB components: full view.

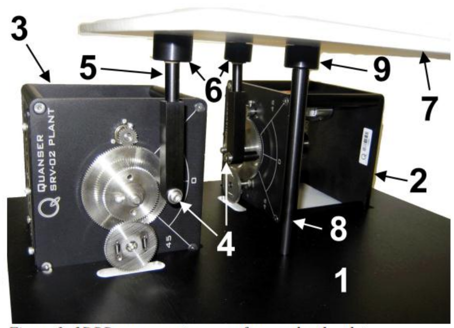


Figure 3: 2DBB components: servo from under the plate.

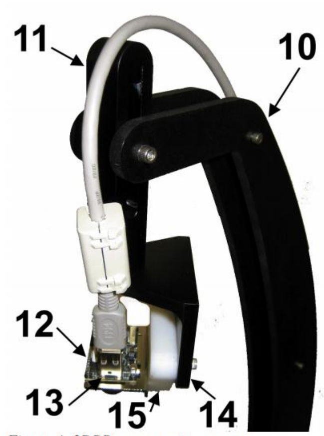


Figure 4: 2DBB components: camera.

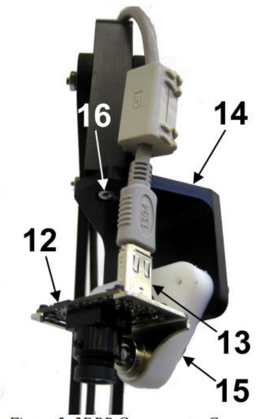
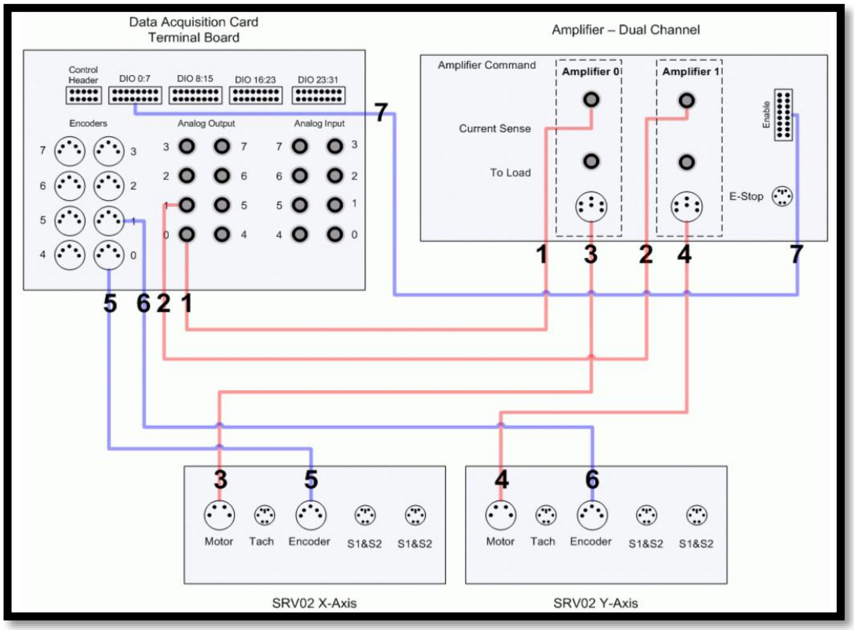


Figure 5: 2DBB Components: Camera (downward view).

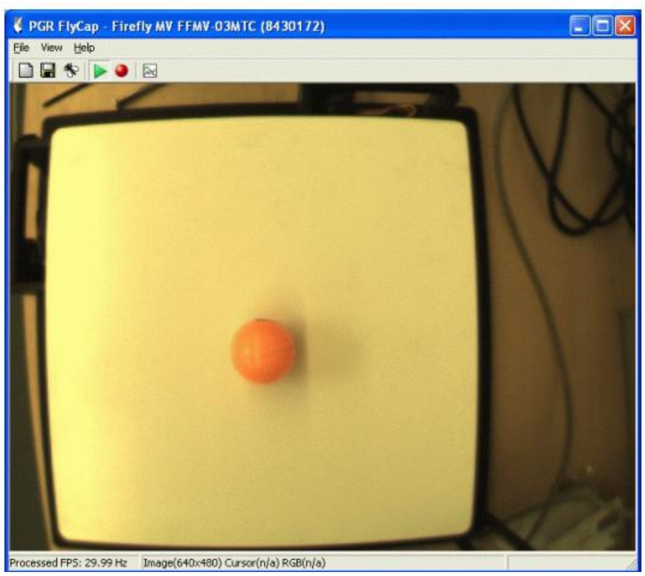
Further assembly instructions can be found in the 2D Ball Balancer - User Manual.pdf on Sakai. Finally, ensure that the controller and 2DBB are wired appropriately



Procedures

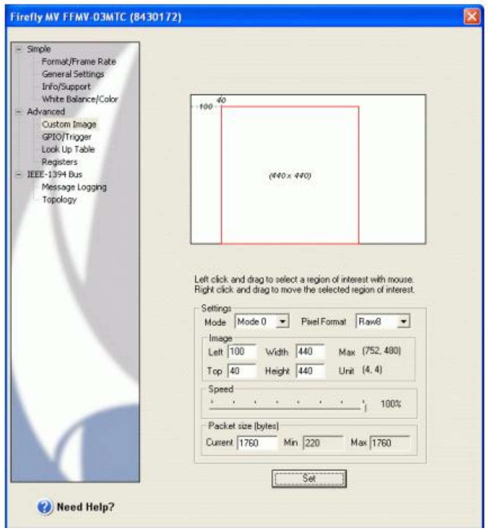
Step 1: Calibration

With the computer turned on and MATLAB closed, go to the search bar (immediately under the start button) and type "`flycam`". Double click Flycam.exe which will open the camera calibration software.



If you see no image, close the software and make sure the camera is on (Green light on camera). If it is not, contact the lab administrator for assistance as the camera is the most difficult component of the lab setup. Next, ensure that there is a live feed from the camera. Take you hand and place it into view. Make sure that the image on the screen updates and you can see your hand. Again, if you cannot, talk with the lab administrator to get the camera working before proceeding.

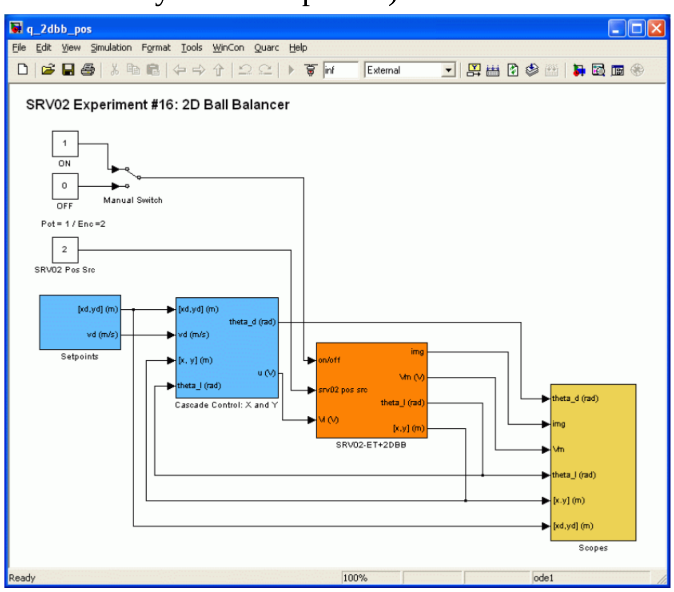
The goal of the following procedure is to crop and orient the image. Start by making sure that the camera is square. To adjust the angle, simply grab the camera and twist it gently until the view is square. Next, to crop the image, click on the fly icon, right next to the save icon. This will open a dialog box with lots of options. In the tree view (Left expandable list) select Custom Image under the Advanced node.



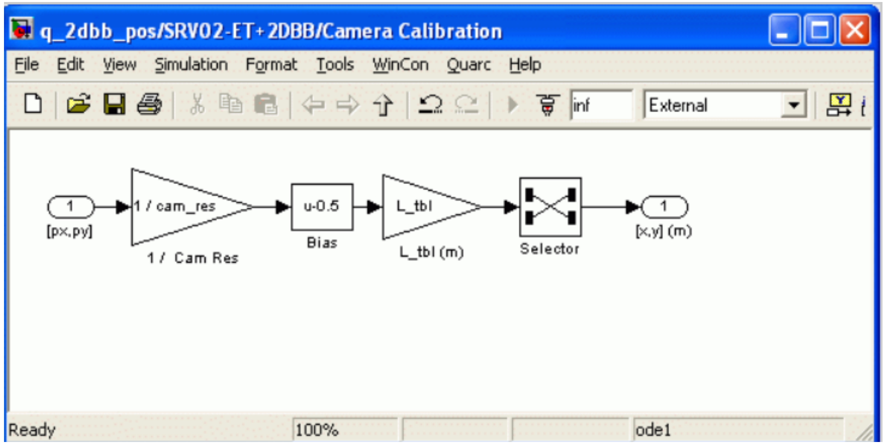
Now adjust the Left, Top, Width and Height values. It is recommended that you adjust these values by intervals of 20. With each new set of values press the “Set” button to apply the new settings. Observe the image and make any adjustments needed to fully crop the image so that only the white surface and ball are showing. Write the resulting values down for later use. Back in the flycam window move the mouse around. The RGB (Color values) of the pixel under the mouse will show up in the bottom toolbar. Place the mouse over the orange ball and record these values.

Step 2: MATLAB and the Environment

Find and open the “q_2dbb_pos_sol” Simulink file (The solution file differs from the normal file in that the camera calibration block has already been completed.).



If instead the user opens the “q_2dbb_pos” file the camera calibration block will need to be constructed. It would be good measure to open the camera calibration block and ensure it is setup anyways:



Back in the main Simulink model window ensure that the Manual Switch is in the on position. The state of this switch can be changed by clicking on the image. If this is left off, the system will show the encoder positions but will not give any commands to the motors. Next ensure that the correct controller is selected. Double click on the orange box. The subsequent screen will have an image of the controller in the upper right hand corner. Double click this icon. This will bring up a dialog box. Ensure that the controller selected is the q8-usb. Accept these changes and close the dialog box.

In the q-2dbb_pos Simulink window select tools and options. Then select the compiler and ensure that the win64 compiler is selected (If you are having trouble with this, it is an advanced step and the lab administrator will be able to help).

Step 3: Setup File

Now, open the "setup_srv02_exp17_2dbb" file. This is the setup file for the experiment. Find the Camera Configuration section and enter the values you found during the camera calibration section:

```
% Resolution (pixels): determines width and height of image
cam_res = 440;
% Image Pos Left (pixels 0-752)
img_pos_left = 100;
% Image Pos Left (pixels 0-752)
img_pos_top = 40;
% RGB Values of ball
img_R = 250;
img_G = 125;
img_B = 50;
% Threshold (%)
img_threshold = 50;
% Minimum object size
min_object_size = 500;
```

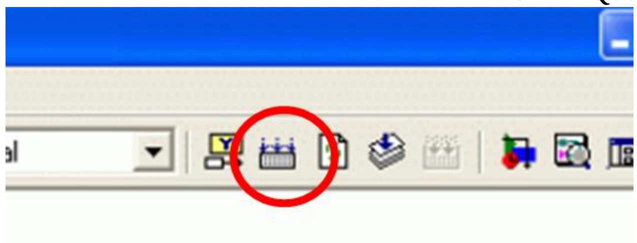
Text 2: "Camera Configuration" section in the setup_srv02_exp17_2dbb.m script.

Save and run the setup script.

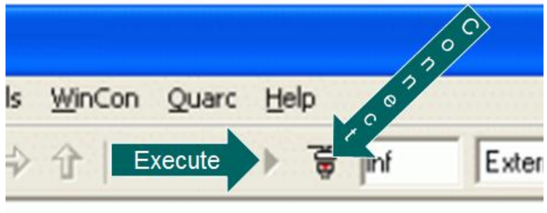
Returning to the qu_2dbb_pos Simulink model, select the Setpoints subsystem and change the inputs to the desired type (Recommend a step input for the first trials).

Step 4: Compile and Run

In the qu_2dbb_pos Simulink model, compile the code:



Now connect to the controller and execute the code:



Allow the code to run and observe the response. When complete, press the stop button (square) which replaced the execute button. From here the user can change values and manipulate the experiment as desired.

References

[1] Quanser Inc., *DAQ User Manual*.
[2] Quanser Inc., *QUARC User Manual* (type `doc quarc` in MATLAB to access).
[3] Quanser Inc., *QUARC Installation Guide*, 2009.
[4] Quanser Inc., *Power Amplifier User Manual*.
[5] Quanser Inc., *SRV02 User Manual*, 2009.
[6] Quanser Inc., *SRV02 QUARC Integration*, 2008.
[7] Quanser Inc., *Rotary Experiment #1: SRV02 Modeling*.
[8] Quanser Inc., *Rotary Experiment #2: SRV02 Position Control*.
[9] Quanser Inc., *Rotary Experiment #4: Ball and Beam Position Control*.
[10] Quanser Inc., *2D Ball Balancer User Manual*.