# Horizontal Bearing rig

This documentation will attempt to provide a complete overview of information relevant to the Horizontal bearing setup and resources to access extra information. Blue text indicates less critical information, which is hopefully unnecessary.

Check other documentation if certain information is lacking ('How to' Guide - Test Motor, LabView & PLECS and PLECS Code and TI Hardware Documentation). However, the project has changed a reasonable amount since these were written.

## Operation

Setup:

Check the booster and Launchpad boards are connected.

Check the J5 jumper is disconnected.

Check the connection of the computer to TP-Link and of the Link to NI chassis and Temperature Controller.

Connect the computer and Launchpad and build the PLECS code onto the Launchpad.

Activate external mode and auto triggering.

Connect the booster board to the motor.

Open the LabView code and check the TCP/IP connection socket is correct along with the chassis connections.

Check appropriate temperatures are reached.

Set the servo control to release the shaft.

Collecting data:

Run the LabView code. –

Ensure the PLECS speed is set to 0, and the rate limiter is set to a low increase.

Connect the power supply to the booster board and turn it on.

Activate the relay in LabView.

Enter the desired speed. Critical speed should be in the ballpark of 50Hz. Try and move through the critical speed quickly.

Increase the rate limiter.

Finishing collection:

Decrease the speed.

Reduce the rate limiter (only in the positive direction so it is low the next time the motor is started).

Save the LabView file.

Disconnect relevant parts (probably turn off the power supply).

Set the servo to hold the shaft again.

Testing and debugging may be done by connecting the power supply, a small motor, the computer and the boards, then running the PLECS code.

**DO NOT:**

Adjust the voltage or current provided to the booster board quickly (jumps of 20V are too fast).

Reverse the positive and negative terminals.

Allow a connected motor to start and stop or change speed rapidly (dependent on the motor). This can include running the motor test code.

Reduce the carrier frequency to that used with commercial motors (10k) without understanding what you're doing. Low inductance motors can require much higher carrier frequency.

I’m sure you can guess where this advice comes from.

## Design

One overview is that a signal is sent from the computer to define a frequency on the Launchpad, creating sine waves to modulate a PWM signal sent to the motor. Data is recorded with a DAQ and LabView.

### History

Throughout its use, the method of transferring a signal to the Launchpad has been adjusted.

The initial design sent a signal from the computer to an analog output device connected to analog pins on the Launchpad, which would read the voltage, then be converted to a PWM and sent to the motor.

This design eventually ran into signal noise problems. While the produced signal was very clean, the analog pins receiving the signal observed fluctuations around 0.1mV. While this is not huge, it must be cleaned to drive the motor smoothly. There was also a distinct offset in the average value of the voltage read by the pins, which was found to vary significantly over time (up to 100%), voltage and pin. Several solutions were explored in conjunction with each other. Very rough rounding helped make the signal clean enough but reduced the resolution at which the motor speed could be controlled. Fast changes in speed **did** damage the booster board multiple times. If readdressed, this problem could be alleviated using the rate limiter block in PLECS. Furthermore, no matter how harsh the rounding, there would always be a line which, if the motor were to oscillate around, would cause unpleasant speed changes. For example, a voltage of 0.5, when rounded to the nearest integer, can oscillate between 0 and 1. Very long periodic averages slowed the response time of the motor with much less benefit than was expected. Calibration was also addressed. Passing both 1V and 0V to pins on the Launchpad to correct for scale and offset was found to be almost useless due to the noisy readings being specific to each pin, only exacerbating the problem by adding more sources of noise.

A possible solution to avoid this noise was to send a pulse train to GPIO pins on the Launchpad instead. While this would be perfectly possible, it would require some non-trivial interpretation on the Launchpad, so it was not implemented due to there being a more accessible option.

The Launchpad is currently controlled through external mode with parameter tunability. The computer is directly connected to the Launchpad with a USB cable, allowing for real-time motor speed adjustment and visualisation of the response. The noise in this connection is considered negligible. The booster board power supply must be isolated from the Launchpad while connected to the computer to prevent overloading the Launchpad. More details will be in the operation section of the documentation.

## PLECS code

Adjusting frequency problems

Rate limiter

External mode

Preferences and software

Protection

PWM enable

## LabView code

This reads and saves temperature and rpm data to a file, activating the relay connecting the driver to the motor. Most relevant information can be inferred from the front panel. Further specifics should be inferred from the code itself, as this documentation will not be able to keep up with changes.

In the past, when this was used to output an analog signal, it was essential to reset that signal to 0 before aborting the program.

### Links

Launchpad

Increasing length. All are separately relevant.

[TMS320F28379D LaunchPad Quick Start Guide (Rev. A) (digikey.com)](https://www.digikey.com/htmldatasheets/production/2368159/0/0/1/launchxl-f28379d.pdf)

[LAUNCHXL-F28379D Overview User's Guide (Rev. C) (ti.com)](https://www.ti.com/lit/ug/sprui77c/sprui77c.pdf?ts=1705459313333&ref_url=https%253A%252F%252Fwww.bing.com%252F)

[TMS320F2837xS Real-Time Microcontrollers Technical Reference Manual (Rev. H)](https://www.ti.com/lit/ug/spruhx5h/spruhx5h.pdf?ts=1704766915996&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTMS320F28377S)

Booster schematic

[BOOSTXL-3PhGaNInv Schematic (Rev. A)](https://www.ti.com/lit/df/sluray0a/sluray0a.pdf?ts=1704752569661)

Inverter stats

[LMG5200 80-V, 10-A GaN Half-Bridge Power Stage datasheet (Rev. E) (ti.com)](https://www.ti.com/lit/ds/symlink/lmg5200.pdf?ts=1705465988619&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FLMG5200)

More information on the hardware used does exist online, but I found these to be the most useful.

### Further work

If this project is readdressed, one way to increase the stability and likely the speed is to address rotor dynamics. A brief outline of what I thought may be relevant can be found in the MagMapper documentation.

Tim Notes

Voltage readings 80V scaled to 3.3V

Offsets on both

5V

GND

ADCIN0?5 -> vdc

ADCINC5 -> va

ADCINB5 ->Vb

ADCINA5 -> Vc

ADCINC4 -> Ia

ADCINB4 -> Ib

ADCINA4 -> Ic

ADCINA0 -> Vref