**PLECS/LabView Code and TI Hardware Documentation**

*Overview and Preliminaries*

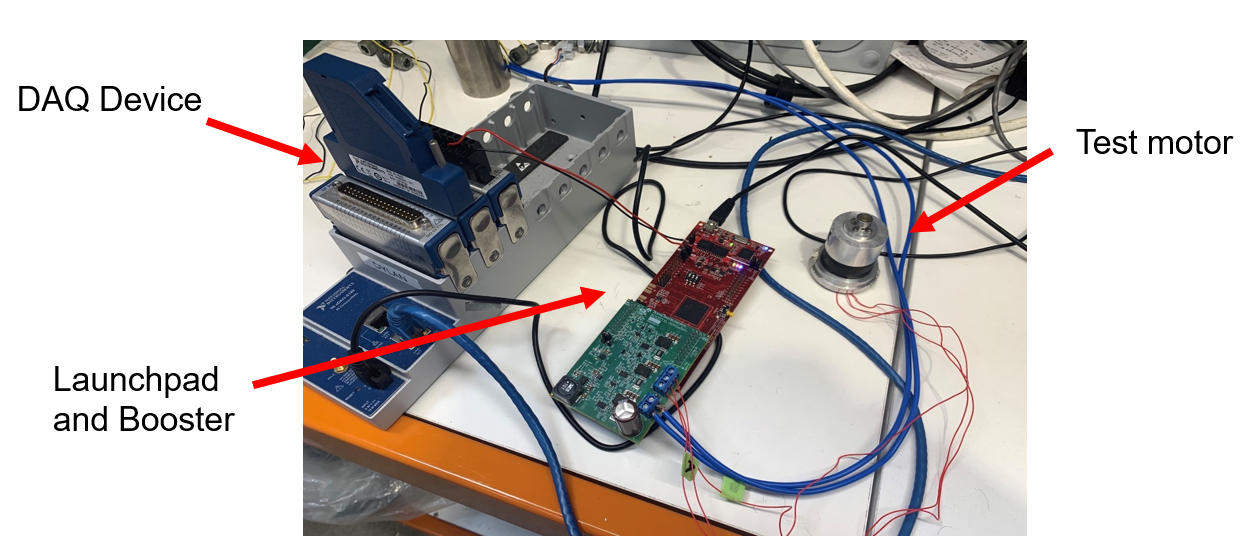
The PLECS program located, as of 08/02/2023, at R:\01 HTS Large Scale\12 Fastest Machine\13 Student Work\Andrew Moore\PLEXIM\Andrew Motor Control, was built to control a high-speed superconducting motor, smoothly, to high speeds. This is achieved by using the program itself alongside LabView code (located as of 08/02/2023 at R:\01 HTS Large Scale\12 Fastest Machine\13 Student Work\Andrew Moore\Code\Horizontal Bearing Control) and hardware provided by Texas Instruments (TI). Below is an in-depth description of each component and step of the motor control programs and setup which could be used to interpret or recreate it.

The TI hardware consists of a LAUNCHXL-F28379D C2000 Delfino Launchpad Development Kit (<https://www.ti.com/tool/LAUNCHXL-F28379D>) and a BOOSTXL-3PhGaNInv Three-Phase Inverter (<https://www.ti.com/tool/BOOSTXL-3PHGANINV>). The pinouts and other useful information about these boards can be found on their datasheets[[1]](#footnote-1),[[2]](#footnote-2). The former (henceforth referred to as the launchpad) is an extended TI launchpad which is the initial connection between the programs provided by a computer/desktop. Signal generating PLECS code is embedded onto the launchpad whereas the real-time motor speed controlling LabView code is connected via a digital signal from a DAQ device to a launchpad pin. The latter (henceforth referred to as the booster) is a 48V/10A 3-phase GaN inverter with precision in-line shunt-based phase current sensing for accurate control of precision drivers such as servo drivers. During testing, the hardware was run with a 12V power supply which was limited to 5A and ran far below this number. This setup is shown in Figure 1.

Preliminary procedures carried out to ensure complete connectivity between the hardware and your computer. Firstly, after connecting the USB port of the launchpad to a USB port on your computer, the connection can be checked by inspecting device manager. Under ‘Ports (COM & LPT)’, the device should appear as ‘XDS100 Class USB Serial Port’. The ‘Texas Instruments Debug Probes’ device should also appear under the main device manager menu, containing both the ‘XDS100 Class Auxiliary’ and ‘XDS100 Class Debug’ ports.

*Hardware Setup*

The hardware used in this setup consists of the launchpad and booster mentioned above, a National Instruments (NI) DAQ device labelled NI9263 held in a NI cDAQ-9189, a Flycat 2805/140KV brushless test motor, and an external power supply. This setup is shown in Figure 1 below. The booster is attached to the launchpad through pins on rows J5//J6/J7/J8, and to the power supply. The test motor is connected to the booster through the three driver ports labelled VA/VB/VC. A description of pin connections to wires and embedded code is in the ‘PLECS Program’ section.



**Figure 1**: Experimental hardware setup for the motor driver.

*LabView Program*

The LabView code used in this motor driving system is an adjustment from the pre-existing vi located (as of 08/02/2023) at R:\01 HTS Large Scale\12 Fastest Machine\13 Student Work\Andrew Moore\From Mohammad\HorizontalBearing. The adjustments made were to the while loop that controlled the signal sent to the launchpad. The original vi used a PWM signal to control the motor but this new system will provide simply provide a voltage output to the launchpad which will determine the frequency of the 3-phase AC signal created by the GaN inverters on the booster (as explained in the PLECS section below).

The purpose of the added code is to provide a signal to the launchpad which corresponds to a particular frequency for the AC signal sent to the motor. As shown on the block diagram, an analogue output voltage is created using the ‘DAQmx Create Channel’ vi and ‘DAQmx Write’ vi. The signal created, determined by the ‘Voltage Output’ control on the front panel, is (as of 08/02/2023) written to port 7 on the DAQ device labelled NI9263. The output is restricted to a value between 0V and 3V because, once sent to the DAQ device, the signal will be connected to an ADC pin on the launchpad which can read an input of up to 3V. This output is read to the DAQ every 10ms by the while loop on the block diagram and this can be adjusted using ‘Voltage Output’ at a controlled rate by adjusting the value of ‘Increment’ on the front panel.

The voltage reading provided to the DAQ, and subsequently to the launchpad, will be slightly lower than that intended on the LabView program due to inherent resistance. Hence, in order to provide the intended signal to the launchpad, a second digital output signal is provided with this LabView program. This second, ‘Normalisation’, signal is created through the same process as the first and can account for the loss in the ‘Voltage Output’ signal by normalising the voltage read by the pin using PLECS (as explained below). For example, if the ‘Voltage Output’ is set to the maximum 3V and the pin is only reading 2.9V, the ’Normalisation’ signal can be slowly adjusted until the PLECS code reads a value of 3V. This is important because the voltage input received by PLECS is what determines the frequency of the 3-phase AC supply sent to the motor so if it is not exactly what you expect then the rotational speed provided will not be close to that expected.

The rest of the LabView program was already developed prior to this motor driver development. No other adjustments were made to the original vi other than those mentioned above. However, the layout of the front panel is slightly different so it may be worth inspecting everything on the front panel first.

*PLECS Program*

Overview and Connection

PLECS is a software tool for system simulations of electrical circuits and software embedding for hardware implementation. The program utilises both inbuilt Plexim mathematical and computational blocks as well as external software compatible blocks associated with the F28379D launchpad. Description of the motor control PLECS program refers to the block diagram mentioned earlier.

Initially, a connection tasks was built to verify that the connections mentioned above are working properly. The ‘Digital Out’ block is sending a constant value of 0 to GPIO124 (pin 13) on the launchpad which is also connected to pin 13 on the booster which is required to enable PWM signal transfer between the launchpad and booster. If you want further confirmation of connectivity between the launchpad and your computer, use the PLECS program ‘LED Pulse’ (located as of 08/02/2023 at R:\01 HTS Large Scale\12 Fastest Machine\13 Student Work\Andrew Moore\PLEXIM \LED Pulse) which sends pulsing signals to the blue and red LEDs on the launchpad (GPIO31 and GPIO34 respectively) allowing for visual connection confirmation.

Block Diagram Description

The external speed control information required by the PLECS program is supplied by the ADC blocks on the left of the bottom section of the block diagram. From top to bottom, these blocks correspond to ADC channels C, B and A, respectively. Real-time inspection of the waveforms of the 3-phase current/voltage supplied to the three GaN inverters on the booster are provided by ADC C4/B4/A4 and ADC C5/B5/A5, respectively. These ports correspond to pins 67/68/66 and 64/65/63 on the launchpad and pins 28/29/27 and 24/25/23 on the booster. The individual voltage and current waveforms can also be inspected by connecting oscilloscope probes to the correct booster pins and the ground connection (GND) on booster pin 22. The ADC blocks also provide digital input values controllable from the LabView front diagram detailed above.

The frequency control voltage output to the DAQ is connected to the launchpad through pin 24, which corresponds to ADC channel C3. Additionally, the signal normalisation constant is supplied, via the DAQ, to the launchpad through pin 26, which corresponds to ADC channel A3. These ADC pins can read a maximum voltage of 3V which corresponds to the range of the voltage output dial in the LabView program. The values and waveforms of these ADC inputs can be viewed with the scopes and digital readings wired to the ADC blocks. The frequency control voltage is divided by the normalisation signal via a ‘product’ block (illustrated by the ‘’ symbols) and then scaled by another ‘product’ block (illustrated by ‘’ symbol). This transforms the range of the signal to lie between 0 and the maximum signal frequency eventually supplied to the synchronous motor. To account for the discrepancy between the ‘Voltage Output’ provided by LabView and that received by ADC pin 24, you can adjust the normalisation constant until you read the intended value on the ‘Normalisation Value’ indicator.

To ensure the motor accelerates smoothly, the frequency controlling waveform is rounded out using the ‘Periodic Average’ block. This block uses the running average value of the previous (adjustable) 1 second to prevent fluctuations in the signal due to inherent noise from causing the value from decreasing during acceleration or increasing during deceleration or from changing when nobody is adjusting the signal. When incorporated with the ‘rounding’ blocks, this allowed the test motor to run smoothly at rotational frequency increments of 1Hz. If this value needs to be lowered for the synchronous motor this can be done by scaling the signal with another ‘product’ block after the second ‘rounding’ block.

The digital ‘Frequency (Hz)’ reading provides a real-time reading of the frequency of the PWM signals sent to the launchpad. These waveforms are created by firstly generating sinusoidal waveforms using the ‘function’ blocks labelled ‘Fcn5’, ‘Fcn6’ and ‘Fcn7’. These blocks output sine waves, at a frequency equal to the signal provided by the earlier code, via an oscillation provided by the ‘clock’ block labelled ‘Clock3’. The output can be seen through the ’Signal Waveforms’ scope. Finally, the PWM block converts these waveforms into PWM signals and sends them, at a carrier frequency of 10e3Hz, to the launchpad via PWM channels 4/5/6 A and B (for the signals and their inverses) through pins 75-80, which are connected to the booster through pins 35-50. These pins were chosen as they are specifically assigned to interpret PWM signals and provide them to the GaN inverters to create a 3-phase AC signal. **To connect and enable auto triggering for the ADC tasks, go to ‘Coder’ on the top toolbar and select ‘Coder Options’, then ‘Externa Mode’. Ensure the target device matches that from device manager then select ‘Build’, ‘Connect’, then ‘Activate auto triggering’.**

The middle section of the PLECS code connected to the ‘PWM Signals’ scope creates a visual interpretation of the PWM signals provided to the launchpad for a particular frequency. This section is not required to run the code and can also be achieved by connecting probe pins to the correct PWM pins mentioned above and the ground connection (GND) through booster pin 20.

The top section of the PLECS code is designed to drive the motor through a combination of pre-set accelerations and decelerations. The first stage is acceleration at a fixed rate then leaving the motor to run at a constant speed. The second stage consists of decelerating the motor to 0Hz before continuing to accelerate the motor in the opposite direction more quickly than the initial acceleration in the first stage. This method effectively speeds the motor up to a maximum speed from an initial negative speed in the opposite direction which could provide greater acceleration capabilities. It is not fully developed yet but if completed it could also provide a way of smoothly testing the acceleration capabilities of a motor at a slower and faster rate.

*Motor Operation*

To begin the motor driver program, complete the embedding and connection steps on PLECS mentioned above. While waiting, make sure the ‘Voltage Output’ control is set to 0V on the LabView front panel and the power supply is set to an appropriate setting (for the test motor 12V and a 5A limit was used). Once the code has been built into the launchpad and the device has been set to externally run with auto triggering, turn on the power supply. It will be useful to have the PLECS block diagram and the LabView front panel visible on screen because these will be where you adjust the motor speed and check the expected/true rotational speeds, etc.

Set the ‘Increment’ values on the LabView front panel to a value below 0.1V. It will be safest to use the arrows on the smaller control to begin the speed up/slow down processes before using the slider to continue these processes. Next, start the LabView program and use the controls to speed up the motor. The code should allow for a smooth, continuous acceleration but will likely cause unwanted results if the slider voltage is increased too rapidly. Use the ‘DAQ Device Voltage’ scope on PLECS to monitor any discrepancy to the slider and adjust accordingly with the Normalisation value. At higher speeds, it may be better to return to the small grey arrow voltage output control.

1. Launchpad Datasheet <https://www.ti.com/lit/ug/sprui77c/sprui77c.pdf?ts=1666679327772&ref_url=https%253A%252F%252Fwww.google.com%252F> , [↑](#footnote-ref-1)
2. 3-Phase Inverter Datasheet <https://www.ti.com/lit/df/sluray0a/sluray0a.pdf?ts=1666731376657&ref_url=https%253A%252F%252Fwww.google.com%252F> [↑](#footnote-ref-2)