RENSSELAER MECHATRONICS

DC Motor Speed and Direction - Step Response

Part 1: Motor Step Response

Objectives:

- Use the DRV8833 or SN754410 motor driver and obtain the motor step response
- Understand the logic needed to control the magnitude and direction of a motor
- Observe the different effects of different logic schemes to drive a motor
- Observe the effect of sampling rate on a discrete sensor

Background Information:

A motor driver chip converts a lower power signal (from the microcontroller: 5v, 40mA) to a high power signal (9v, 1.5A) to drive the motor. It can be thought of as a switch or relay to the driver supply voltage (9v batteries or 5v for USB in this case). Since the supply voltage is fixed this would result in a fixed motor speed. To regulate the speed a PWM signal is used to quickly switch the output on and off so that the average output voltage can be controlled.

The term motor "driver" is also commonly called "amplifier" or "chopper". The DRV8833 driver is rated at providing 2.7-10.8v at 1.5A RMS on each channel.

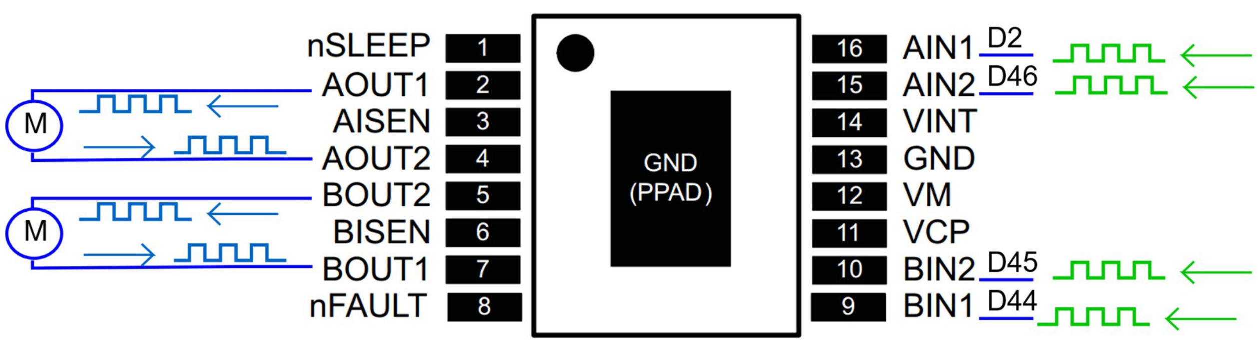


Figure 1: DRV8833pinout/wiring diagram

The SN754410 driver is capable of providing 3.5-36v at 1A on each channel.

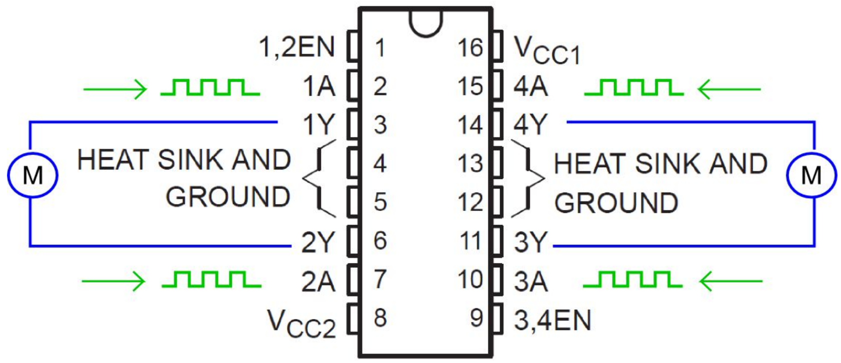


Figure 2: SN754410 pinout/wiring diagram

The green lines show the lower power PWM inputs, that get amplified then sent to the output pins to the motors.

Simulink Model

Build and run the following Simulink diagram (recommended to use demo file for your hardware as starting point). Use the corresponding encoder block for **your system**, and the **digital pins that correspond to your motor driver**. Use the encoder blocks from RASPlib.

It is highly recommended you remove the wheels from your motor before performing motor experiments. It is common to have the motor tires pinch the motor and rub affecting the response. To ensure this is not an issue remove the wheels until they are needed.

Shield or Board	Motor Number	Driver Pins	Encoder Pins
M1V4	1	5, 4	2, 3
	2	9, 11	6* or 0*, 10*
M2V3	1	6, 8	15, A8
	2	2, 5	18, 19
MinSegMega	1	2, 46	PE6, PE7
	2	44, 45	18,19
MinSegNano V1,V2	1	9, 10	2, 3
	2**	11, 12	A3, A2
MinSegNano V3	1	9, 10	2, 3
	2	5, 6	A2, A3

* Requires Jumpers on J4, J5, J6

** located on side pins

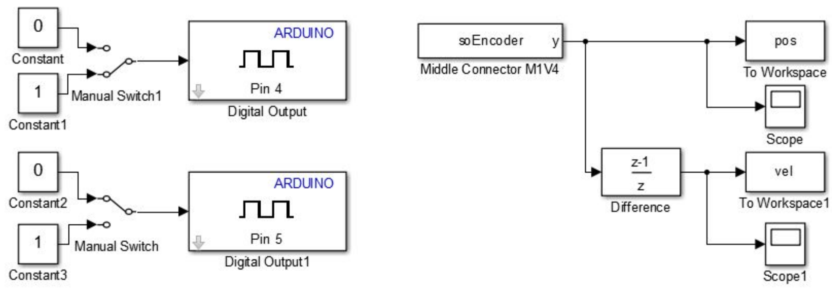


Figure 3: for M1V4 with encoder block

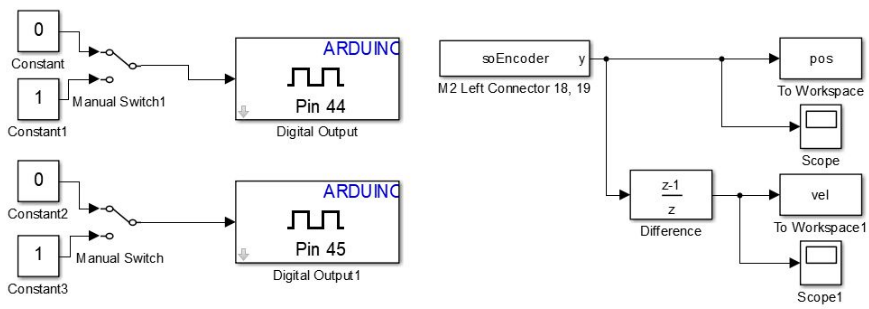


Figure 4: MinSegMega with encoder block

- Run the model in external mode to log the speed and position into variables 'vel' and 'pos'
  - Sampling rate of .03 seconds
- Observe the response of the system
  - toggle the manual switches in the Simulink diagram. Make sure the
    - battery switch for your board is set to USB power
    - driver voltage is set to USB
    - If you have a driver enable it is set to ON
- Stop the model then plot the results
  - The commented line below stores the variables 'pos' and 'vel' into the data file 'ext_dat.mat'. Copy and paste the commented command in the command line to save the data. **You do not want this uncommented in an m-file since it might cause you to accidentally overwrite previously stored data file when you run the m-file.**

```
%save ext_dat pos vel    % save the data
load ext_dat             % load the data
figure, plot(pos)
figure, plot(vel)
```

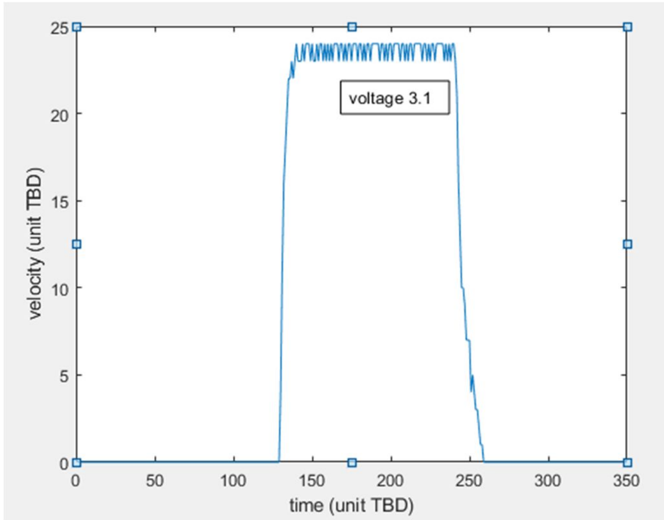


Figure 4: Velocity Step Response Graph

Questions:

- Provide a plot that contains the velocity step response. Plot the velocity in RPM and time in seconds. You will need to determine the unit conversions. An example plot is shown above for an unknown motor with unknown units. Provide labels on both axes with appropriate unites and an appropriate title.

Part 2: Motor Logic: Direction and Magnitude

To regulate the speed a PWM signal is used to quickly switch the output on and off so that the average output voltage can be controlled. In addition, the correct switching logic needs to be implemented so that the motor can change direction based on the sign of the input.

The subsystem block below is used to correctly output the correct magnitude and logic of the PWM if the only input is a scalar input voltage (which could be negative)**). You do not have to make this Simulink diagram – it is shown for reference only.**

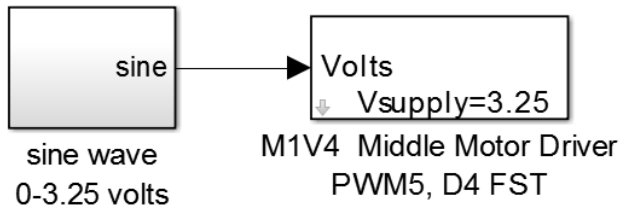


Figure 5: Motor driver block to control magnitude and direction

To correctly use this block, you must specify Vsupply. Vsupply is the maximum high power voltage the driver (and your computer) is capable of providing. The voltage it can supply is usually the high power voltage input supplied to the driver minus the losses associated with the driver. It is also depended on the computer power supply and USB type (2.0, 3.0, type C), or battery supply.

For example, if the SN754410 driver is supplied a high power voltage of 4.5 volts from the USB, the max voltage observed at the output pins might be 3.3 volts (ideally this would be 4.5 volts). Due to driver losses the max voltage the application will receive is 3.3 volts.

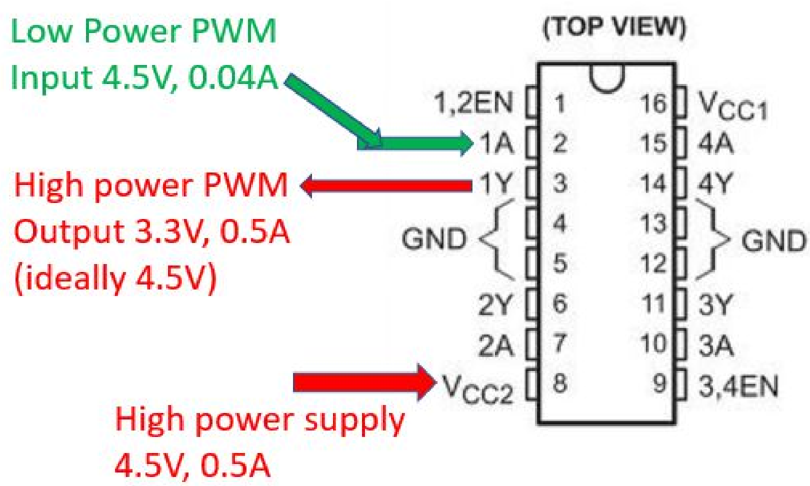


Figure 6: Vsupply for this example is 3.3 volts - the maximum voltage the driver is capable of supplying. Even though the input high power supply voltage is 4.5volts, due to driver losses the max available at the output is 3.3 volts.

To translate your desired voltage in Simulink to the PWM block you have to scale your voltage by 255/3.3.

For the SN754410 driver this is approximately the voltage supply minus 1.2 volts which are the losses from the driver. For the DRV8833 driver the loss is about 0.2 volts.

The contents of this subsystem and the scaling to PWM are shown below:

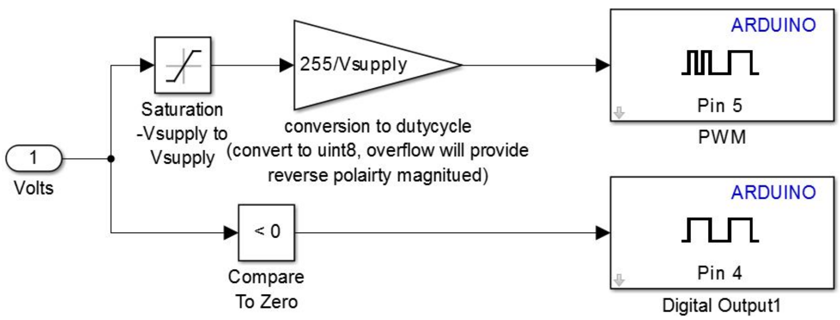
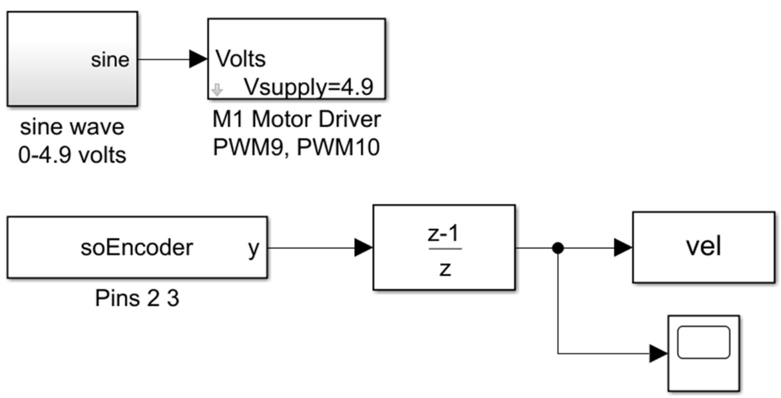


Figure 7: Example Logic for controlling magnitude and direction of motor voltage (for reference only - do not build this diagram, the contents of your driver block may look different)

It performs the 3 following basic functions (you do not have to make the block diagram in Figure 8):

- It scales the input by the voltage that is being supplied to the driver (Vsupply), this appropriately scales the voltage to PWM (a number from 0-255).
- Based on the sign of the input – it will switch one of the motor control lines to positive voltage, or ground (controlling the direction of the motor)
- Based on the magnitude of the input it controls the magnitude of the resulting PWM

Use this subsystem block to build the following Simulink diagram in the figure below to generate a sinusoidal motor response (**use the blocks from *your* hardware demo file**).



- Set the parameters of the sine wave so it has a magnitude of the same volts as Vsupply and a period of around 3 seconds. **Use the sine wave block from the demo file (You have saved it in Lab 1a)**. The default sine wave block in Simulink appears to use slow the serial port data transmission in external mode for older versions of MATLAB (might be using timers that interferes with serial timers). Note that the diagram above is for reference only – you will may to appropriately modify the values.

Questions:

- Provide a plot of the sinusoidal velocity in external mode at .03 at .003 seconds. Explain why these two look different (look at the values of the magnitude of the response).

Part 3: Using the Discrete Derivative Block

In the previous sections the Difference block was used to capture velocity. The difference block ensured the resulting number would be integers (the encoder block outputs number of counts) this could allow the data to be sent as single byte (unsigned integer). Using the derivative block will automatically divide by the sample time and the data must be represented as a floating point number. A unit conversion is then needed to convert form counts to radians/second. (**you will need to use the correct unit conversion for *you* motor and the correct blocks**)

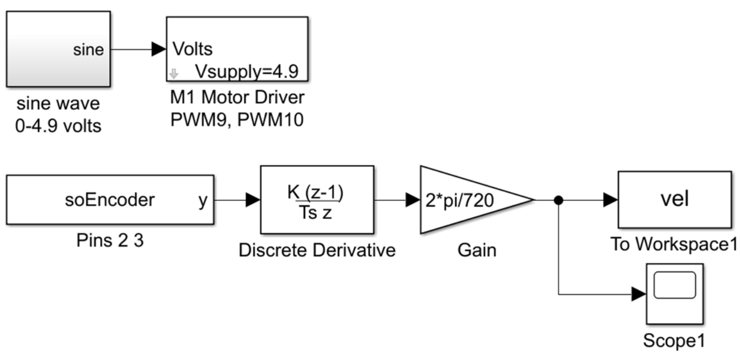


Figure 8: Use the correct blocks for your system

Questions:

- Provide a plot of the sinusoidal velocity in external mode at .03 at .003 seconds. Using the derivative block. Label the axes and title appropriately with correct units.