Brushed DC Motor
Microcontroller PWM Speed Control with
Optical Encoder and H-Bridge

L298 Full H-Bridge
HEF4071B OR Gate
Arduino Microcontroller for
Encoder Decoding & Velocity Output

Brushed DC Motor with
Optical Encoder & Load Inertia
Flyback Diodes
Arduino Microcontroller for
Speed Control Implementation
Digital Inputs and Outputs
and PWM Outputs

Pins 3, 5, 6, 9, 10, and 11 work as analog outputs (PWM)

USB Programming Port

Power Supply

Analog Inputs

Brushed DC Motor / Encoder System
# Pittman DC Servo Motor 8322S001

## Assembly Data

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Units</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>V</td>
<td>12</td>
</tr>
<tr>
<td>$S_{NL}$</td>
<td>rpm (rad/s)</td>
<td>7,847 (822)</td>
</tr>
<tr>
<td>$T_C$</td>
<td>oz-in (N-m)</td>
<td>1.6 (1.1E-02)</td>
</tr>
<tr>
<td>$T_{PK}$</td>
<td>oz-in (N-m)</td>
<td>7.4 (5.2E-02)</td>
</tr>
<tr>
<td>$W_M$</td>
<td>oz (g)</td>
<td>7.7 (218)</td>
</tr>
</tbody>
</table>

## Motor Data

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Units</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_T$</td>
<td>oz-in/A (N-m/A)</td>
<td>1.94 (1.37E-02)</td>
</tr>
<tr>
<td>$K_E$</td>
<td>V/krpm (V/rad/s)</td>
<td>1.43 (1.37E-02)</td>
</tr>
<tr>
<td>$R_T$</td>
<td>$\Omega$</td>
<td>3.10</td>
</tr>
<tr>
<td>$L$</td>
<td>mH</td>
<td>1.57</td>
</tr>
<tr>
<td>$I_{NL}$</td>
<td>A</td>
<td>0.25</td>
</tr>
<tr>
<td>$I_F$</td>
<td>A</td>
<td>3.88</td>
</tr>
<tr>
<td>$K_M$</td>
<td>oz-in/\W (N-m/\W)</td>
<td>1.12 (7.91E-03)</td>
</tr>
<tr>
<td>$T_F$</td>
<td>oz-in (N-m)</td>
<td>0.35 (2.5E-03)</td>
</tr>
<tr>
<td>$J_M$</td>
<td>oz-in-s(^2) (kg-m(^2))</td>
<td>1.4E-04 (9.9E-07)</td>
</tr>
<tr>
<td>$\tau_E$</td>
<td>ms</td>
<td>0.52</td>
</tr>
<tr>
<td>$\tau_M$</td>
<td>ms</td>
<td>15.6</td>
</tr>
<tr>
<td>$D$</td>
<td>oz-in/krpm (N-m-s)</td>
<td>0.015 (1.0E-06)</td>
</tr>
<tr>
<td>$K_D$</td>
<td>oz-in/krpm (N-m-s)</td>
<td>0.92 (6.2E-05)</td>
</tr>
<tr>
<td>$\theta_{MAX}$</td>
<td>°F (°C)</td>
<td>311 (155)</td>
</tr>
<tr>
<td>$R_{TH}$</td>
<td>°F/watt (°C/watt)</td>
<td>75.9 (24.4)</td>
</tr>
<tr>
<td>$\tau_{TH}$</td>
<td>min</td>
<td>7.8</td>
</tr>
</tbody>
</table>

## Encoder Data

<table>
<thead>
<tr>
<th>Channels</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CPR</td>
</tr>
<tr>
<td></td>
<td>500</td>
</tr>
</tbody>
</table>
Pittman DC Servo Motor 8322S001

Encoder
500 counts/rev

<table>
<thead>
<tr>
<th>Wire</th>
<th>Function</th>
<th>Color</th>
<th>Pins</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND</td>
<td>Black</td>
<td>GND</td>
</tr>
<tr>
<td>2</td>
<td>Index</td>
<td>Green</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>CH A</td>
<td>Yellow</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Vcc</td>
<td>Red</td>
<td>5V</td>
</tr>
<tr>
<td>5</td>
<td>CH B</td>
<td>Blue</td>
<td></td>
</tr>
</tbody>
</table>

Brushed DC Motor / Encoder System

A. da Silva & K. Craig
L298
Dual Full Bridge Driver

Operating supply voltage up to 46 V
Total DC current up to 4 A
Low saturation voltage
Overtemperature protection
Logical "0" input voltage up to 1.5 V
(High noise immunity)
Brushed DC Motor / Encoder System

A. da Silva & K. Craig
Topics

• **Brushed DC Motor**
  – Physical & Mathematical Models, Hardware Parameters
• **H-Bridge Operation**
• **Feedback Control Design**
  – MatLab / Simulink Design and Auto-Code Generation
Brushed DC Motor
For a permanent-magnet DC motor $i_f = \text{constant}$.

**Physical Modeling**
• **Physical Modeling Assumptions**

  – The copper armature windings in the motor are treated as a resistance and inductance in series. The distributed inductance and resistance is lumped into two characteristic quantities, $L$ and $R$.

  – The commutation of the motor is neglected. The system is treated as a single electrical network which is continuously energized.

  – The compliance of the shaft connecting the load to the motor is negligible. The shaft is treated as a rigid member.

  – The total inertia $J$ is a single lumped inertia, equal to the sum of the inertias of the rotor and the driven load.
- There exists motion only about the axis of rotation of the motor, i.e., a one-degree-of-freedom system.
- The parameters of the system are constant, i.e., they do not change over time.
- The damping in the mechanical system is modeled as viscous damping \( B \), i.e., all stiction and dry friction are initially neglected.
- The optical encoder output is decoded in software. Position and velocity are calculated and made available as analog signals for control calculations. The motor is driven with a PWM control signal to a H-Bridge. The time delay associated with this, as well as computation for control, is lumped into a single system time delay.
• **Mathematical Modeling Steps**
  - Define System, System Boundary, System Inputs and Outputs
  - Define Through and Across Variables
  - Write Physical Relations for Each Element
  - Write System Relations of Equilibrium and/or Compatibility
  - Combine System Relations and Physical Relations to Generate the Mathematical Model for the System
Physical Relations

\[ V_L = L \frac{di_L}{dt} \]
\[ V_R = R i_R \]
\[ T_B = B \omega \]
\[ T_J = J \alpha = J \dot{\omega} \]
\[ J \equiv J_{\text{motor}} + J_{\text{load}} \]
\[ T_m = K_t i_m \]
\[ V_b = K_b \omega \]
\[ P_{\text{out}} = T_m \omega = K_t i_m \omega \]
\[ P_{\text{in}} = V_b i_m = K_b \omega i_m \]
\[ \frac{P_{\text{out}}}{P_{\text{in}}} = \frac{K_t}{K_b} \]
\[ P_{\text{out}} = P_{\text{in}} \]
\[ K_t = K_b \equiv K_m \]

\[ K_t (\text{oz} - \text{in} / \text{A}) = 1.3524 K_b (\text{V} / \text{krpm}) \]
\[ K_t (\text{Nm} / \text{A}) = 9.5493 \times 10^{-3} K_b (\text{V} / \text{krpm}) \]
\[ K_t (\text{Nm} / \text{A}) = K_b (\text{V} - \text{s} / \text{rad}) \]

Brushed DC Motor / Encoder System

A. da Silva & K. Craig
System Relations + Equations of Motion

\[ V_{in} - V_R - V_L - V_b = 0 \]

\[ T_m - T_B - T_J = 0 \]

\[ i_R = i_L = i_m \equiv i \]

KVL

\[ V_{in} - Ri - L \frac{di}{dt} - K_b \omega = 0 \]

\[ J \frac{d\omega}{dt} + B \omega - K_t i = 0 \]

\[
\begin{bmatrix}
  \frac{d\omega}{dt} \\
  \frac{di}{dt} \\
\end{bmatrix}
= \begin{bmatrix}
  -B & K_t \\
  -K_b & -R \\
\end{bmatrix}
\begin{bmatrix}
  \omega \\
  i \\
\end{bmatrix}
+ \begin{bmatrix}
  0 \\
  \frac{1}{L} \\
\end{bmatrix} V_{in}
\]
Steady-State Conditions

\[ V_{in} - Ri - L \frac{di}{dt} - K_b \omega = 0 \]

\[ V_{in} - R \left( \frac{T}{K_t} \right) - K_b \omega = 0 \]

\[ T = \frac{K_t}{R} V_{in} - \frac{K_t K_b}{R} \omega \]

\[ T_s = \frac{K_t}{R} V_{in} \quad \text{Stall Torque} \]

\[ \omega_0 = \frac{V_{in}}{K_b} \quad \text{No-Load Speed} \]
Transfer Functions

\[ V_{in} - Ri - L \frac{di}{dt} - K_b \omega = 0 \]
\[ J \frac{d\omega}{dt} + B\omega - K_t i = 0 \]
\[ V_{in}(s) - (Ls + R)I(s) - K_b \Omega(s) = 0 \]
\[ (Js + B)\Omega(s) - K_t I(s) = 0 \]
\[ \frac{\Omega(s)}{V_{in}(s)} = \frac{K_t}{(Js + B)(Ls + R) + K_t K_b} = \frac{K_t}{JLs^2 + (BL + JR)s + (BR + K_t K_b)} \]
\[ = \frac{K_t}{JL} \]
\[ s^2 + \left( \frac{B}{J} + \frac{R}{L} \right)s + \left( \frac{BR}{JL} + \frac{K_t K_b}{JL} \right) \]
Block Diagram

Block Diagram for Brushed DC Motor / Encoder System

\[ V_{in} \rightarrow \Sigma \rightarrow \frac{1}{Ls+R} \rightarrow i \rightarrow K_t \rightarrow T_m \rightarrow \frac{1}{Js+B} \rightarrow \omega \]

\[ K_b \]

\[ \sum \]

A. da Silva & K. Craig
Simplification

\[ \tau_m = \frac{J}{B} \quad \Rightarrow \quad \tau_e = \frac{L}{R} \]

\[ V_{in} - Ri - K_b \omega = 0 \]

\[ J \frac{d\omega}{dt} + B\omega - K_t i = 0 \]

\[ J \frac{d\omega}{dt} + B\omega = K_t i = K_t \left( \frac{1}{R} (V_{in} - K_b \omega) \right) = \frac{K_t}{R} (V_{in} - K_b \omega) \]

\[ \frac{d\omega}{dt} + \left( \frac{K_t K_b}{RJ} + \frac{B}{J} \right) \omega = \frac{K_t}{RJ} V_{in} \]

\[ \frac{d\omega}{dt} + \left( \frac{1}{\tau_{motor}} + \frac{1}{\tau_m} \right) \omega = \frac{K_t}{RJ} V_{in} \]

\[ \frac{d\omega}{dt} + \left( \frac{1}{\tau_{motor}} \right) \omega = \frac{K_t}{RJ} V_{in} \quad \text{since} \quad \tau_m \gg \tau_{motor} \]
Brushed DC Motor / Encoder System

A. da Silva & K. Craig

Pitmann Brushed DC Motor Modeling

Equations of Motion

\[ E_{\text{in}} = L \frac{d\omega}{dt} + Ra + K_b \omega \]

\[ J \ddot{\omega} + B \omega = K_t \tau \]

Coulomb friction neglected

Open-Loop Transfer Function \( \frac{\omega}{E_{\text{in}}} (s) : \)

\[ \frac{\omega}{E_{\text{in}}} = \frac{K_t \sqrt{JL}}{D^2 + \left( \frac{B}{J} + \frac{R}{L} \right) D + \left( \frac{BR}{JL} + \frac{K_t K_b}{JL} \right)} \]

\[ \tau_m = \frac{J}{B} \}

\[ \tau_m \gg \tau_e \]

\[ \tau_e = \frac{L}{R} \]

\[ \ddot{\omega} + B \omega = K_t \tau = K_t \left[ \frac{1}{R} (E_{\text{in}} - K_b \omega) \right] = \frac{K_t}{R} (E_{\text{in}} - K_b \omega) \]

\[ \ddot{\omega} + \left( \frac{K_t K_b}{RJ} + \frac{B}{J} \right) \omega = K_t \frac{1}{RJ} E_{\text{in}} \]

\[ \dot{\omega} + \left( \frac{1}{\tau_{\text{motor}}} + \frac{1}{\tau_m} \right) \omega = \frac{K_t}{RJ} E_{\text{in}} \Rightarrow \dot{\omega} + \left( \frac{1}{\tau_{\text{motor}}} \right) \omega = \frac{K_t}{RJ} E_{\text{in}} \]
Analysis: Pittman 83225001 Brushed DC Servo Motor

\[
\begin{align*}
J_{\text{motor}} &= 9.9 \times 10^{-7} \text{ kg-m}^2 \\
J_{\text{load}} &= \frac{1}{2} \text{m} R^2 \\
&= 9.93 \times 10^{-6} \text{ kg-m}^2
\end{align*}
\]

\[
\frac{J_L}{J_m} = 10.03 \quad J_{\text{total}} = 10.92 \times 10^{-6} \text{ kg-m}^2
\]

\[
B = 1.0 \times 10^{-6} \quad \text{N-m-s} \\
K_T = 1.37 \times 10^{-2} \quad \text{(N-m)/A} \\
K_b = 1.37 \times 10^{-2} \quad \text{V/rad/s} \\
R = \frac{3.10}{A} \\
L = 1.57 \times 10^{-3} \text{ H}
\]

\[
\begin{align*}
T_m &= \frac{J}{B} = 10.92 \\
T_e &= \frac{J}{K_T} = 5.06 \times 10^{-4} \\
\frac{T_m}{T_e} &= 2.16 \times 10^4
\end{align*}
\]

\[
\begin{align*}
T_{\text{motor}} &= \frac{R T}{K_T K_b} = 0.180 \\
\frac{T_m}{T_{\text{motor}}} &= 60.5
\end{align*}
\]
\[ T_{motor} \dot{\omega} + \omega = \frac{1}{K_b} e_{in} \]

\[(0.180) \dot{\omega} + \omega = (73.0) e_{in} \]

\[ T \dot{\omega} + \omega = K e_{in} \]

\[ \frac{\omega}{e_{in}} = \frac{K}{T_D + 1} \]

1st Order ODE

\[ \begin{cases} T = 0.180 \text{ sec} \\ K = 73.0 \end{cases} \]

From Pittman Data Sheet

Motor Constant \( K_m = 7.91 \times 10^{-3} = \frac{K_t}{\sqrt{R}} \)

\( T_f = 2.5 \times 10^{-3} \text{ N-m} \) Friction Torque (Coulomb friction dynamic)

\( T_E = 0.52 \times 10^{-3} \text{ sec} = \frac{L}{R} \) same as \( T_E \)

\( T_m = 15.6 \times 10^{-3} \text{ sec} = \frac{RJ}{K_t K_b} \) same as \( T_{motor} \)

\( K_b = \text{damping constant} = \frac{K_t K_b}{R} = 6.2 \times 10^{-5} \text{ N-m-s} \) (same units as \( b \))
Steady-State Analysis

\[ E_{in} = L \frac{dL}{dt} + R_\ell + K_b \omega \implies E_{in} = R_\ell + K_b \omega \]

\[ J \dot{\omega} + B\omega = K_t \ell \implies B\omega = K_t \ell \]

Define \( T = B\omega \) (includes all load torque)

\[ E_{in} - R_\ell - K_b \omega = 0 \]

\[ E_{in} - R \left( \frac{T}{K_t} \right) - K_b \omega = 0 \implies T = \frac{K_t}{R} E_{in} - \frac{K_t K_b}{R} \omega \]

\[ T = \frac{K_t}{R} E_{in} - \frac{K_t K_b}{R} \omega \]

Linear Torque-Speed Relation

\( \omega = 0 \implies T_r = \frac{K_t}{R} E_{in} \) (stall torque)

\( T = 0 \implies \omega_0 = \frac{E_{in}}{K_b} \) (no-load speed)
\[ E_{in} = 12 \text{ V} \]

\[ T_s = \frac{K_t}{R} E_{in} = 0.053 \text{ N-m} \]

\[ \omega_0 = \frac{E_{in}}{K_b} = 876 \text{ rad/s} \]

(822 in data sheet)

\[ P_{\text{Power}} = T \omega = \omega \left[ \frac{K_t}{R} E_{in} - \frac{K_t K_b}{R} \omega \right] \]

\[ = \omega \left[ T_s - \frac{T_s}{\omega_0} \omega \right] = \omega T_s \left[ 1 - \frac{\omega}{\omega_0} \right] \]

\[ P = T_s \left[ \omega - \frac{\omega^2}{\omega_0} \right] \]

\[ \frac{dP}{d\omega} = T_s - \frac{2 T_s}{\omega_0} \omega = 0 \quad \Rightarrow \quad \omega_{\text{max}} = \frac{\omega_0}{2} \]

\[ \text{Peak Current (Stall)} = \frac{E_{in}}{R} = 3.87 \text{ A} \]

\[ \text{No-Load Current} = 0.25 \text{ A} \]
Pittman Data Sheet:

no-load speed = 822 rad/s
no-load current = 0.25 A

Interpret this information:

No-load current is a measure of friction loss.

\[ E_m = L \frac{dI}{dt} + R I + K_b \omega \]
\[ \frac{d\omega}{dt} = 0 \]

No-load speed:
\[ \omega = \frac{E_m - R I}{K_b} = \frac{12 - (3.10)(0.25)}{(1.77 \times 10^{-2})} = 819.3 \text{ rad/s} \]

At this no-load speed:
\[ T_m = K_t \omega = (1.77 \times 10^{-2})(819.3) = 3.43 \times 10^{-3} \text{ N-m} \]

At steady state:
\[ T_m = T_{viscous} + T_{coulomb} \]
\[ = B \omega + T_f \]
\[ = (1.0 \times 10^{-6})(819.3) + (2.5 \times 10^{-5}) = 3.32 \times 10^{-3} \text{ N-m} \]
Load Torque vs. Efficiency

Rated Armature Voltage = 12 V

Load Torque $T_L = 1.5 \times 10^{-3}$ in = \( (1.50 \times 10^{-3}) \times (3.60) \times (39.79) \times (106) \times 10^{-2} \) in

\[ T_L = 1.0 \times 10^{-6} \text{ Nm} \]

\[ K_T = 1.37 \times 10^{-2} \frac{\text{Nm}}{A} \]

\[ T_F = 2.5 \times 10^{-3} \text{ Nm} \]

During steady state:

\[ K_T i = B \omega + T_F + T_L \]

\[ \omega = \frac{1}{B} \left( K_T i - T_F - T_L \right) \]

Also during steady state:

\[ e = R i + K_b \omega \]

\[ i = \frac{e}{R} - \frac{K_b \omega}{R} = \frac{e}{R} - \frac{K_b \omega}{R} \left( K_T i - T_F - T_L \right) \]

\[ \text{Substitute numbers:} \]

\[ \omega = \frac{1}{1.0 \times 10^{-6}} \left[ (1.37 \times 10^{-2})(654) - (2.5 \times 10^{-3}) - T_L \right] = 654 \text{ rad/s} \]
Power Input: \( P_{in} = e \times L = (12)(1.004) = 12.05 \text{ W} \)

Power Output: \( P_{out} = T \omega = (1.06 \times 10^{-2})(654.8) = 6.94 \text{ W} \)

Efficiency \( \% = \frac{P_{out}}{P_{in}} \times 100 = \frac{6.94}{12.05} \times 100 = 57.6 \% \)

Losses: \( P_{loss} = R I^2 = (2.10)(1004)^2 \)

\[ = 2.125 \text{ W} \]

\( P_{friction} = (B\omega + T_f)\omega = [(1.0 \times 10^{-6})(654.8) + (2.5 \times 10^{-3})] \times (654.8) \]

\[ = 2.066 \text{ W} \]

Total Losses: \( 2.066 + 2.125 = 5.191 \text{ W} \)

\( P_{in} = P_{out} + P_{loss} \)

\[ 12.05 = 6.94 + 5.19 \]

\[ 12.05 = 12.13 \checkmark \]
% Pittman Motor 8322S001 Load Torque vs. Efficiency Plot

B = 1.0E-6;
Kt = 1.37E-2;
Tf = 2.5E-3;
Kb = 1.37E-2;
R = 3.10;
TL = 0:.1:7.5;
TL = TL/3.6/39.37;
e = 12;
i = (1/(R + Kb*Kt/B))*(e + (Kb/B)*(Tf + TL));
omega = (1/B)*((Kt*i) - Tf - TL);
Pin = e^i;
Pout = TL.*omega;
E = (Pout./Pin)^100;
plot(TL*3.6*39.37,E)
H-Bridge Operation

• For DC electric motors, a power device configuration called H-Bridge is used to control the direction and magnitude of the voltage applied to the load. The H-Bridge consists of four electronic power components arranged in an H-shape in which two or none of the power devices are turned on simultaneously.

• A typical technique to control the power components is via PWM (Pulse Width Modulation) signal. A PWM signal has a constant frequency called *carrier frequency*. Although the frequency of a PWM signal is constant, the width of the pulses (the duty cycle) varies to obtain the desired voltage to be applied to the load.
The H-Bridge can be in one of the four states: **coasting**, **moving forward**, **moving backward**, or **braking**, as shown on the next slide.

- In the **coasting mode**, all four devices are turned off and since no energy is applied to the motor, it will coast.
- In the **forward direction**, two power components are turned on, one connected to the power supply and one connected to ground.
- In **reverse direction**, only the opposite power components are turned on supplying voltage in the opposite direction and allowing the motor to reverse direction.
- In **braking**, only the two devices connected to ground are tuned on. This allows the energy of the motor to quickly dissipate, which will take the motor to a stop.
• The four diodes shown in anti-parallel to the transistors are for back-EMF current decay when all transistors are turned off.
• These diodes protect the transistors from the voltage spike on the motor leads due the back-EMF when all four transistors are turned off. This could yield excessive voltage on the transistor terminals and potentially damage them.
• They must be sized to a current higher than the motor current and for the lowest forward voltage to reduce junction temperature and the time to dissipate the motor energy.
Diodes for back-EMF protection are shown. The solid line is the current flow when the transistors on the upper left corner and on the right lower corner are turned on. The dashed line shows the motor current when all transistors are turned off.
• The approach shown next to generate the PWM command for an H-Bridge was developed for the Dual Full Bridge Driver L298 from STMicroelectronics.

• Each bridge contains two inputs (IN1 and IN2 for the bridge A and IN3 and IN4 for bridge B) and an enable for each bridge (ENA for one bridge and ENB for the other bridge).

• The operation of this bridge is shown in the table below for bridge A. The operation for bridge B is identical.

<table>
<thead>
<tr>
<th>Enable</th>
<th>Inputs</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN = 1</td>
<td>IN1 = 1, IN2 = 0</td>
<td>Forward Move</td>
</tr>
<tr>
<td></td>
<td>IN1 = 0, IN2 = 1</td>
<td>Reverse Move</td>
</tr>
<tr>
<td></td>
<td>IN1 = IN2</td>
<td>Motor Fast Stop</td>
</tr>
<tr>
<td></td>
<td>IN1 = X, IN2 = X</td>
<td>Motor Coast</td>
</tr>
</tbody>
</table>

1 = High, 0 = Low, X = Don’t care
Block diagram of L298 (Dual Full Bridge Driver)
• This approach to generate the PWM command for the L298 consists of **three steps**:
  – Split the analog torque command to the motor into two PWM signals (one for each input of one of the bridges of the L298)
  – Logics to control inputs and enable the L298
  – Motor connection and protection of the bridge
• **Step A**
  - The torque command from the control system can be split into two PWM signals for an Arduino board as shown below.

  ![Diagram](image)

  - A dead-band control is used to avoid short circuits on the bridge with inductive loads while switching direction, as the transistor that is commanded to turn off stays conducting for a short period of time due the motor back-EMF when the other transistor on the same branch may be commanded to turn on for the switching in direction.
– Thus, if the torque command is within the dead-band, all four transistors are turned off. If the torque command is positive and higher than the dead-band threshold, a signal is applied to the “PWM FWD Direction” output as shown. Similarly, if the torque command is negative and lower than the dead-band threshold, signal is applied to the “PWM REV Direction” output.

– The gains “255/8” converts the torque command of a maximum of 8 Nm in this example into a digital signal of 8 bits ($2^8 = 256$) which is the resolution of an analog output on Arduino boards.

– The analog output on the Arduino is actually a PWM signal of approximately 490 Hz. Thus, there is no need for generating PWM signal from the analog torque command using Arduino because the analog output is PWM.
– If the torque command is a symmetric sinusoid with amplitude of 8 Nm, the outputs “PWM FWD Direction” and “PWM REV Direction” would be as shown below.
• **Step B**

  - The logic to control inputs and enable the L298 consists of a single OR gate that allows disabling all transistors of the H-Bridge when the command PWM signals are at zero. This logic located between the Arduino board and the H-Bridge is shown below.
• **Step C**
  - The motor is connected to the outputs of the bridge. Depending on the type of H-Bridge used, internal protection to the transistor of the bridge may not exist. In this case, external protection circuitry needs to be provided. This protection consists of diodes connected in anti-parallel to the transistors. Schottky diodes are preferred for inductive loads. The motor rated voltage needs to be supplied to the bridge in order to allow the motor to develop rated torque. If the bridge is supplied with voltage higher than the motor rated voltage, damage may occur to the motor. A sensing resistor ($R_s$) can be used to monitor the motor current and shutdown the transistors if the motor rated current or the bridge maximum current is exceeded. The connection of the motor to the bridge and the diodes to protect the transistors are shown on the next slide.
Feedback speed control of the DC motor can be accomplished using several approaches.

- The Single-Output, Single-Input (SISO) MatLab tool is typically used to design classical feedback control systems. A combination of the root locus approach and the frequency response approach is most effective.

- Once a controller, e.g., PI or PID, is designed, the block diagram shown can then be used in Simulink with the Code Generation to create and download to the Arduino microcontroller the control code for the motor.
DC-Motor - Closed Loop Velocity Control

SampleTime = 0.020 seconds
for data monitoring

Velocity Feedback (rev/sec)

PID Controller

Saturation
12V to -12V

H-Bridge Control

10-bit A/D

5V = 1023 = 50 rev/sec
2.5V = 512 = 0 rev/sec
0V = 0 = -50 rev/sec

Control Implementation with Arduino and Simulink Code Generation

Brushed DC Motor / Encoder System

A. da Silva & K. Craig 45
**H-Bridge Control Subsystem**

H-Bridge Control Block

It contains a “dead band” control to avoid two transistors on the same side of the H-bridge turning on at the same time and damaging the H-bridge. It also contain the logic to split the signal from the PID controller into forward and reverse direction for the transistors of the H-bridge.
These blocks show how you can send multiple bytes by combining different signals into a vector using the Mux block. In this case we are adding a Header and Terminator character to our message.

We are also using the Convert block to take the int16 value we get from the Analog Input and convert it to a 2-element uint8 vector.

Since the serial port only reads integers, this gain adds a decimal point to the readings. But the signal will be 10 times the actual value.

Use the "Serial Configuration" block to make sure that the used serial port is the one to which the arduino is connected.

A value coming from analog "In1" will also be received and displayed.
Steps to Use the Plot Data Subsystem

- **Plot Data**: This block reads any data from the Simulink code and plots it on a Scope. The steps to use this feature are as follows:

1. Connect the Arduino board to the USB port of your computer
2. Run the command in the MatLab Command Window:
   
   ```matlab
   comPorts=arduino.Prefs.searchForComPort
   ```
3. This command will show the number of the COM port that the Arduino board is connected.
4. Open the block called “Launch host side”
5. Open the “Serial Configuration” block and set the “Communication Port” to the port number identified in Step 2.
6. Open the block “Serial Receive” and set the “Communication Port” to the port number identified in Step 2.

7. Run the following command in the Command Window: SampleTime = 0.02.

8. Download the code to Arduino

9. Hit “Play” on the “demo_arduino_serial_communication_host”
Double click on the Scope to start monitoring the data.

The time to capture data on the Scope can be changed as shown.