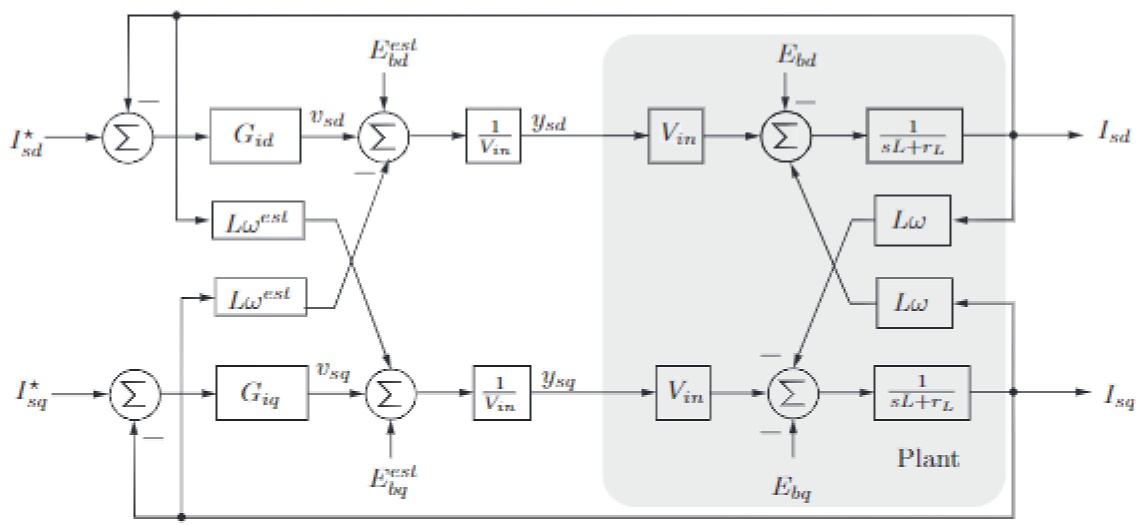
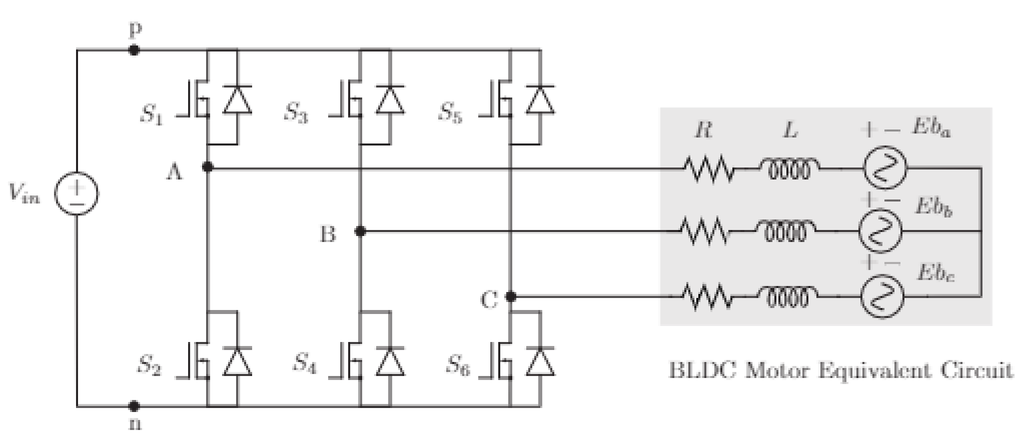
EE 458-533
Experiment 3: BLDC Motor Control

David Babin, Kevin Egedy
Winter 2022





1. Brushless DC Motors

1.3 E-Bike BLDC PLECS Model

PLECS Simulation Parameters	Description
$\boldsymbol{PP} = 24$	Number of pole pairs
$\boldsymbol{p} = \mathbf{2} \cdot \boldsymbol{PP} = 48$	Number of poles
$\boldsymbol{R} = 292.5 \times 10^{-3}\ \Omega$	Stator resistance (Ohms)
$\boldsymbol{L} = 429.7 \times 10^{-6}\ \mathrm{H}$	Stator inductance (H)
$\boldsymbol{\lambda_m} = 0.045$	Back EMF Flux Constant
$\boldsymbol{KE} = \boldsymbol{\lambda_m} \cdot PP$	Amplitude of back EMF (Vs/rad) * pole pairs

2. Realtime simulation of BLDC in Open Loop

2.1 Three phase PWM signal

Start rotating the BLDC motor at a slow speed of $\omega_s = 2\pi\, 5$ rad/s , and then slowly increase your speed. As you increase the ω_s, you will observe that the oscillations in the actual rotor speed also increases. This effect is more pronounced at lower modulation indices when the power pumped in to the motor is also small.

Inside the PLECS environment, use a periodic average filter to filter out the PWM signals that you obtain from the CCS. Verify the operation at a speed of $2\pi 5$ elec-rad/s with modulation index of 0.1,0.5 and 1, and submit waveforms for Back EMF and Speed for each modulation index. Attach the filtered PWM waveforms (all 6 switching signals) for one modulation index from the scope in PLECS. **[4X4 pts]**

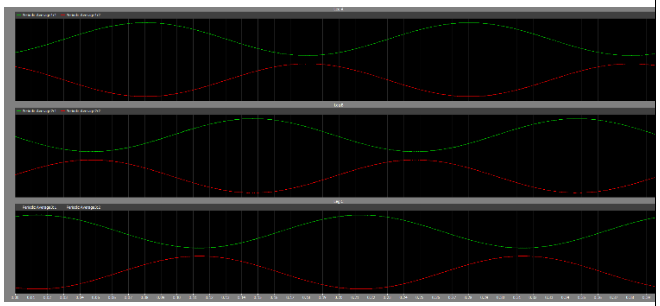
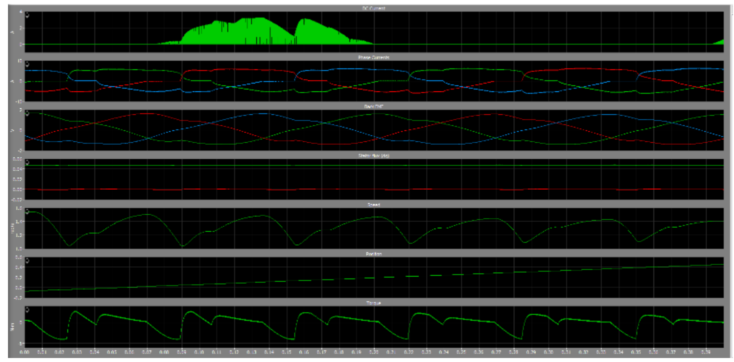
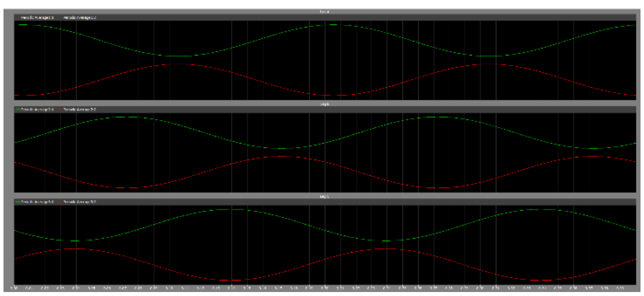
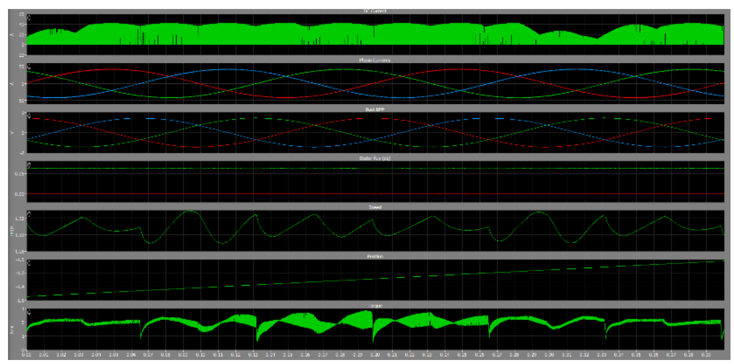
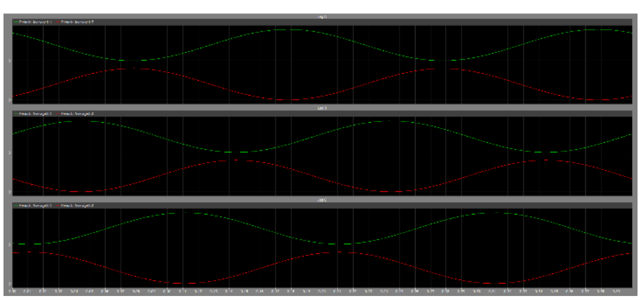
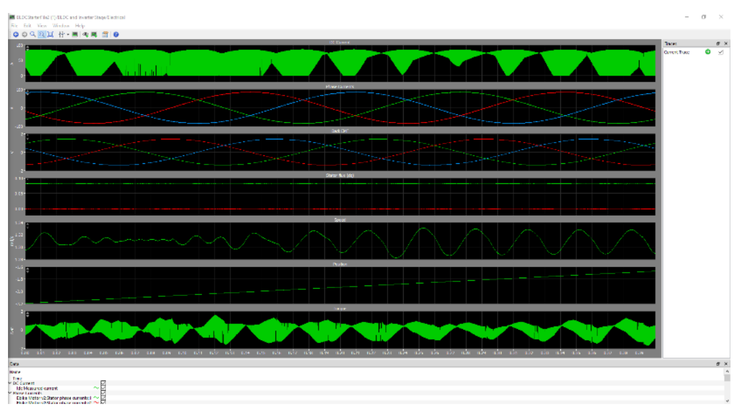
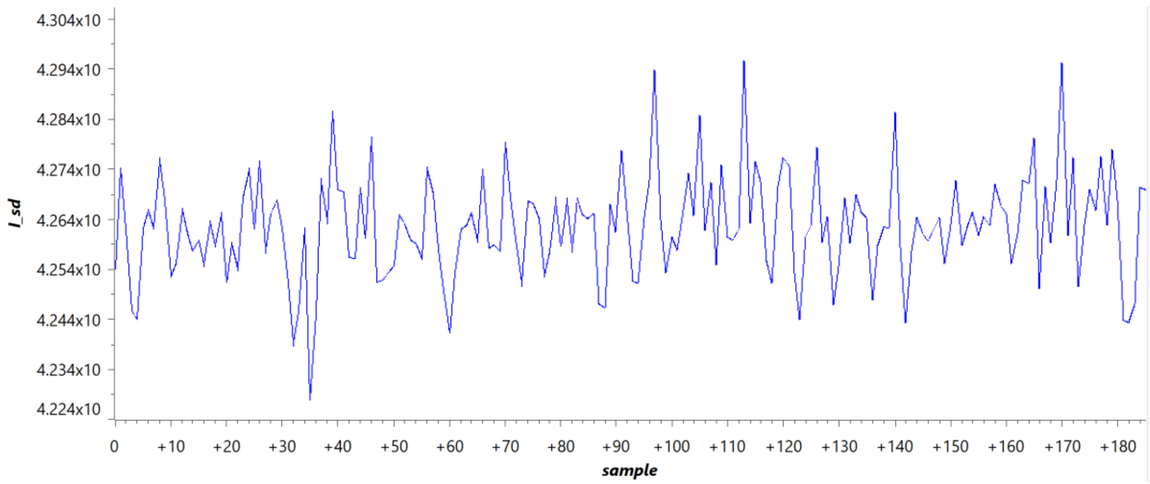
M. I.	PWM	Back EMF & Speed
0.1		
0.5		
1		

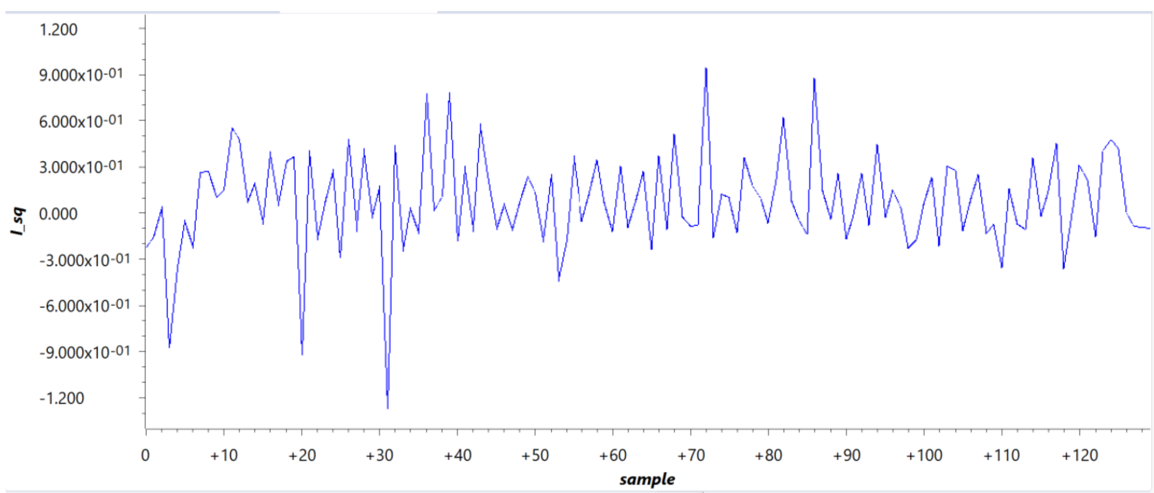
Figure: PLECS Back EMF and Speed Waveforms (ωs = 2π 5 rad/s)

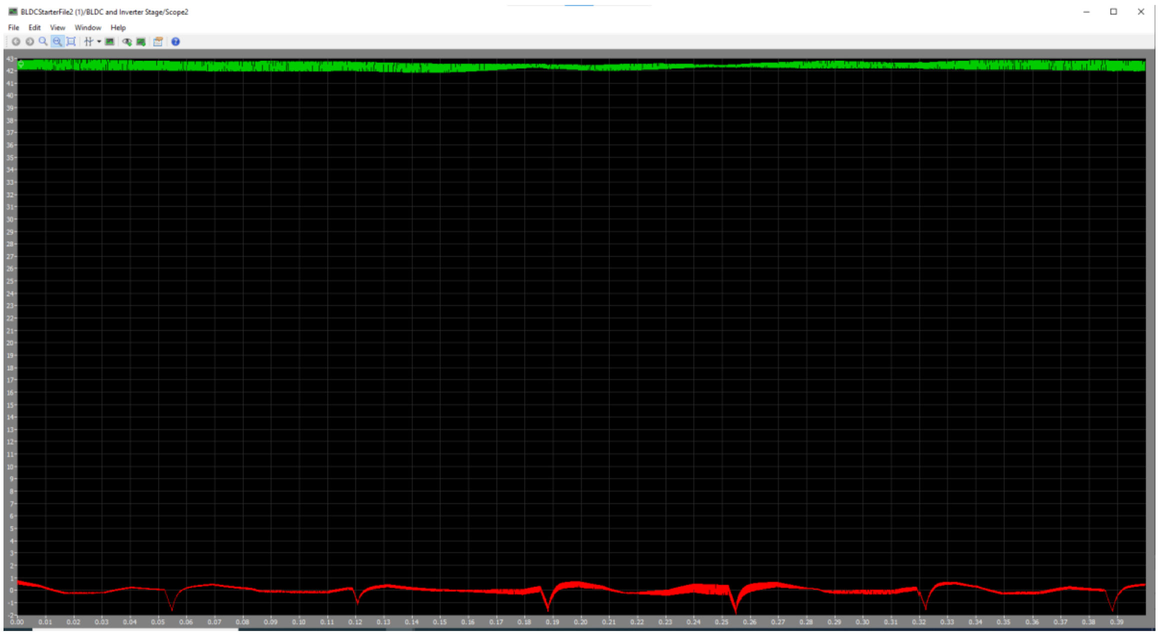
3. If your frame transformation and the angle estimation is correct, you will now observe that the d and q component of the currents will read dc values and not ac signals.

Their magnitudes will be fairly constant when observed in the CCS watch window. The magnitude of the currents i_{sd}, i_{sq} is related to the peak of the three phase stator current. Capture a snapshot of the PLECS to show the peak of the stator current and the CCS watch window expression showing $i_{s,d}$ and $i_{s,q}$. **[15 pts]**

Expression	Type	Value
Isd	float	42.5862961
Isq	float	0.491215944



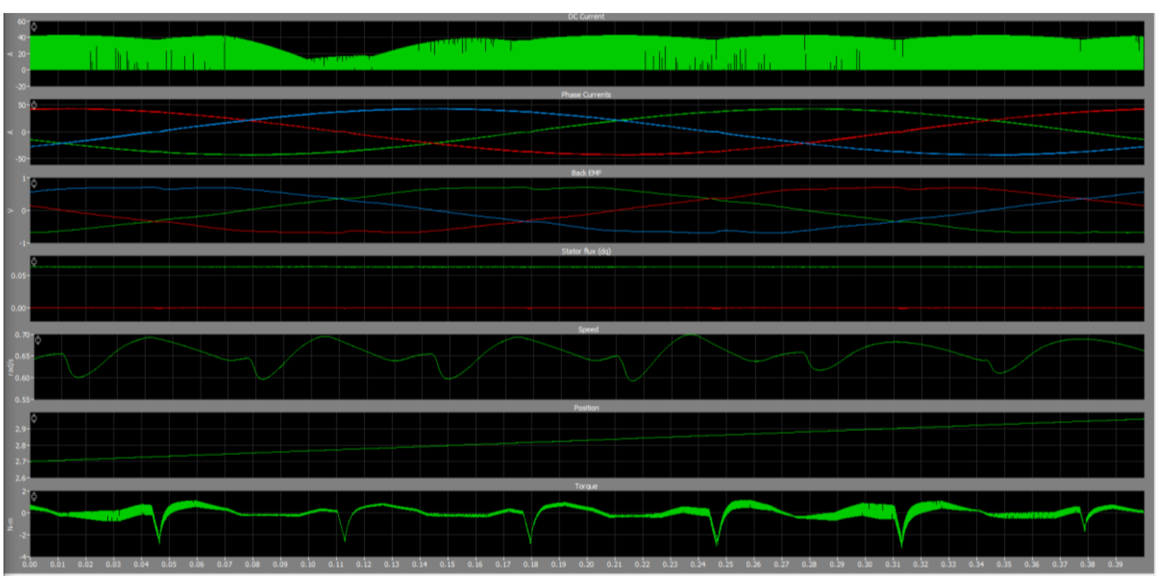


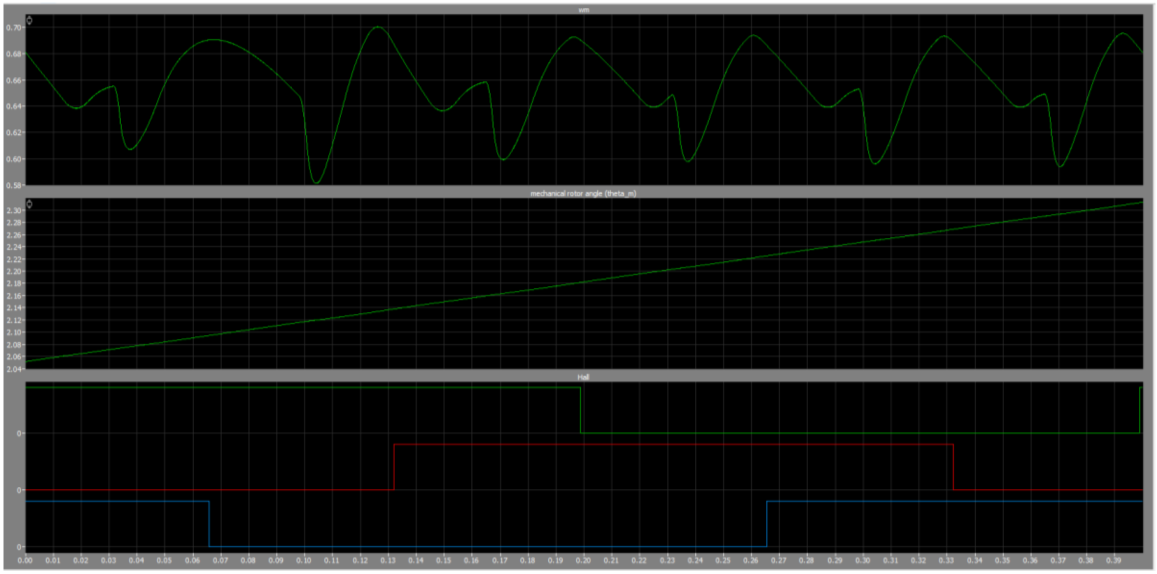


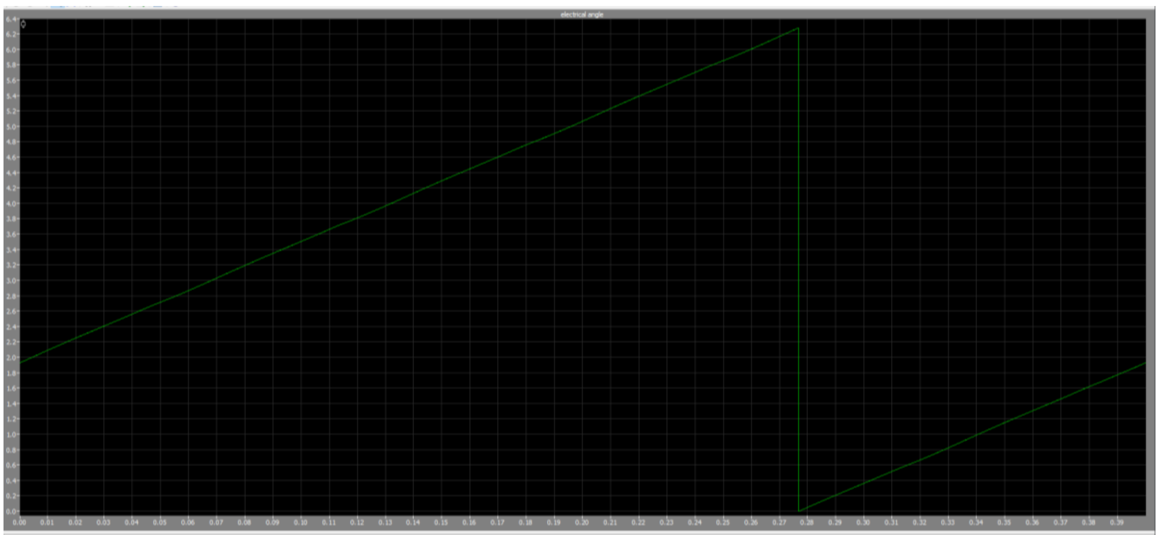
2.3 ABC to DQ conversion for stator currents

Questions:

1. Capture the following waveforms for the BLDC machine operating as a motor driven by the inverter (use $\omega_s = 5Hz$, modulation index of 0.5): stator currents in abc frame, back emf in abc frame, stator flux in dq frame, hall sensor outputs, rotor angle. [**5 pts**]







2. **[Required for Grad Students]** Graph the stator currents in the DQ frame in PLECS and compare them to the values you are seeing in CCS, and to the peak of the phase currents. Do this for modulation indices [0.1, 0.5, 0.9]. **[5 pts]**

Modulation Index 0.1

Expression	Type	Value	Address
(x)= Isd	float	7.9685545	0x0000A9CC@...
(x)= Isq	float	-0.705110908	0x0000A9CE@...

*Figure: CCS Stator DQ Currents (*Modulation Index = 0.1)

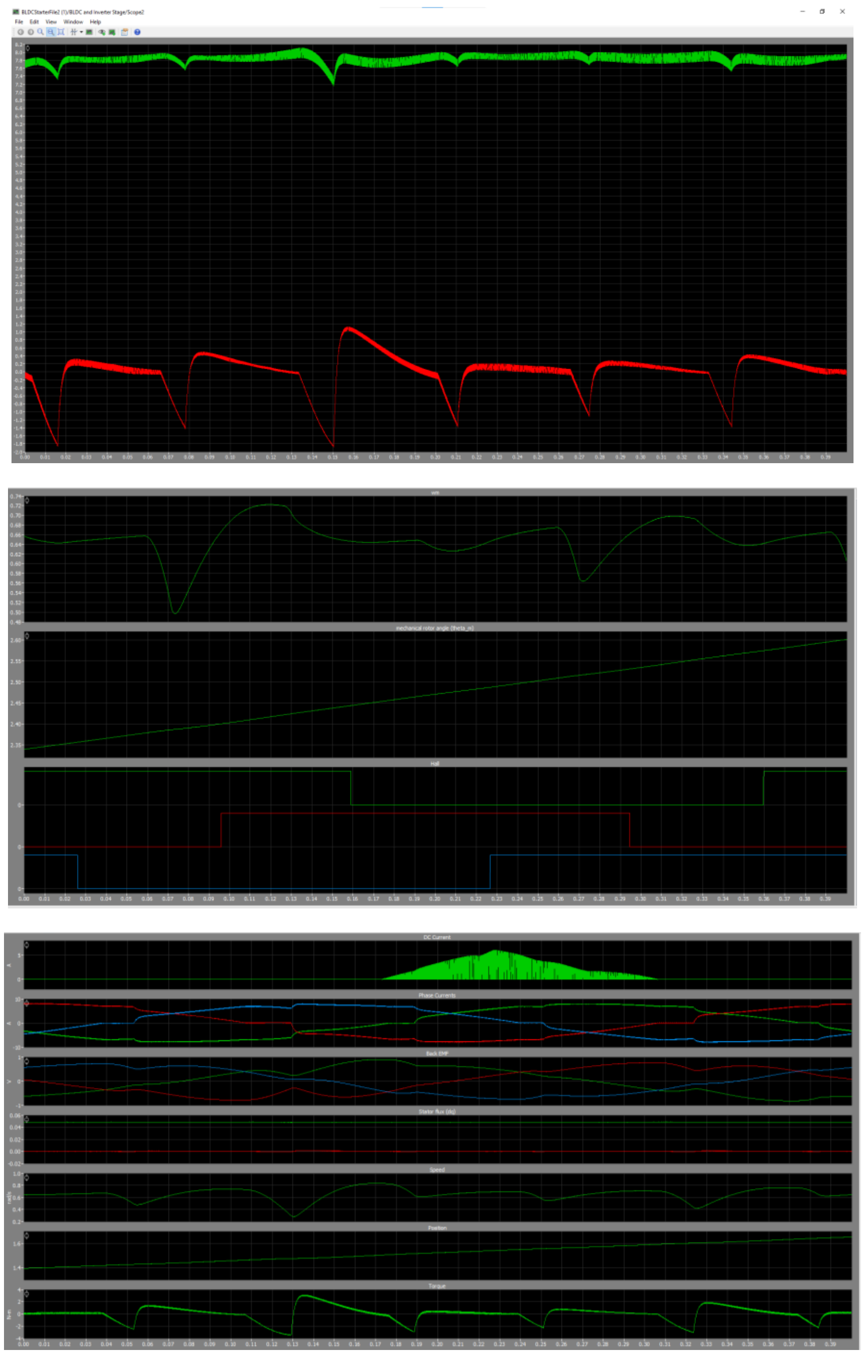


*Figure: Real-Time Stator DQ Currents (*Modulation Index = 0.1)

Modulation Index 0.5

Expression	Type	Value
Isd	float	42.5862961
Isq	float	0.491215944

*Figure: CCS Stator DQ Currents (*Modulation Index = 0.5)

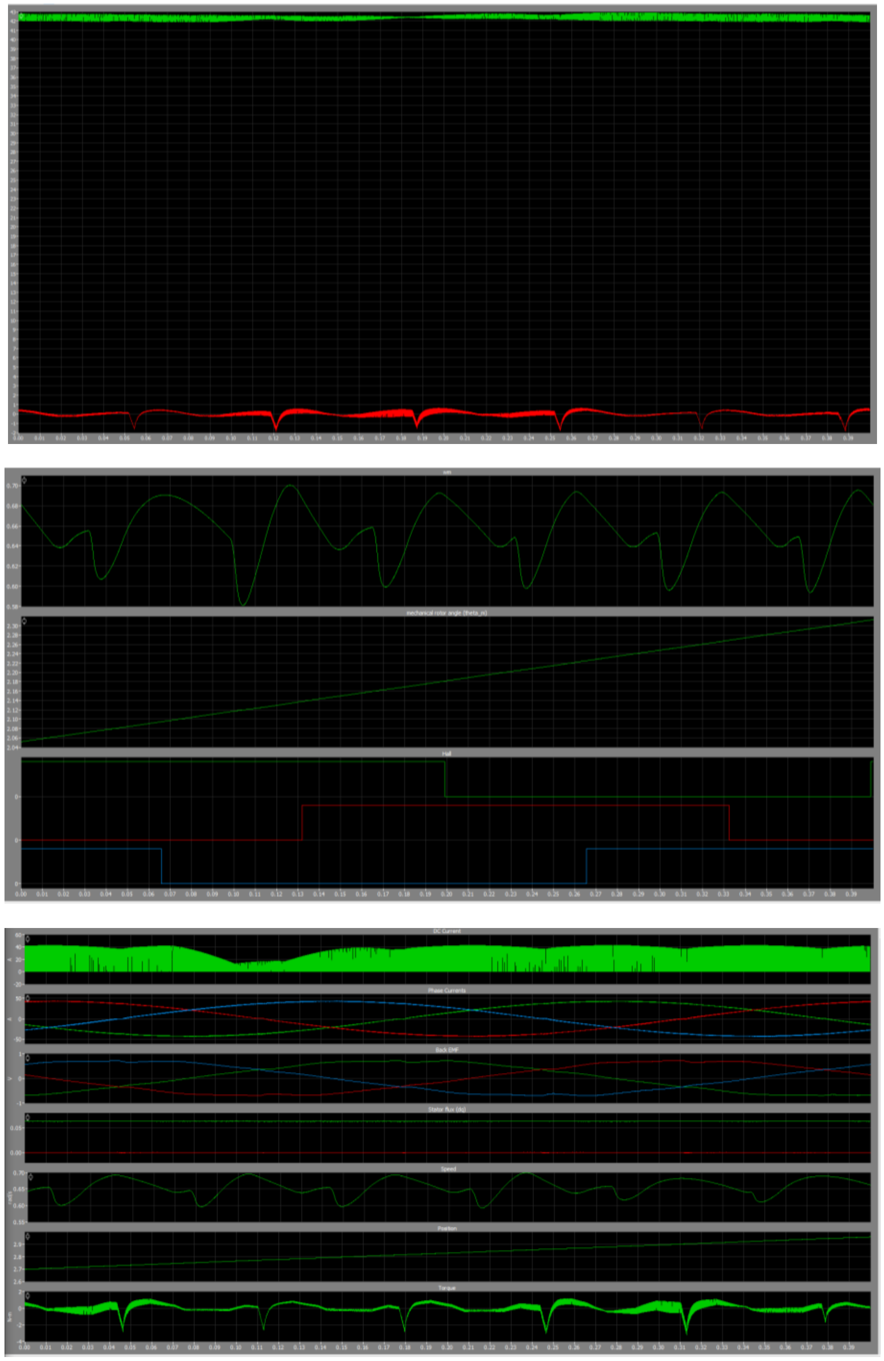
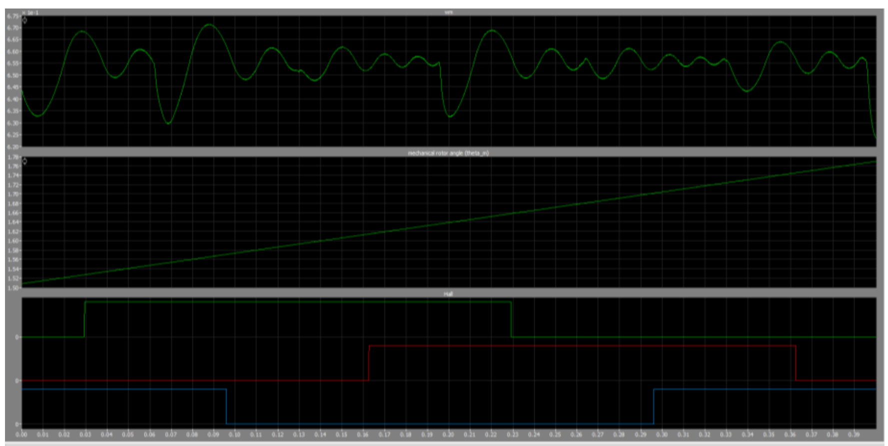


*Figure: Real-Time Stator DQ Currents (*Modulation Index = 0.5)

Modulation Index 0.9

(x)= Isd	float	76.4021606	0x0000A9CC@...
(x)= Isq	float	0.326694697	0x0000A9CE@...

*Figure: CCS Stator DQ Currents (*Modulation Index = 0.9)



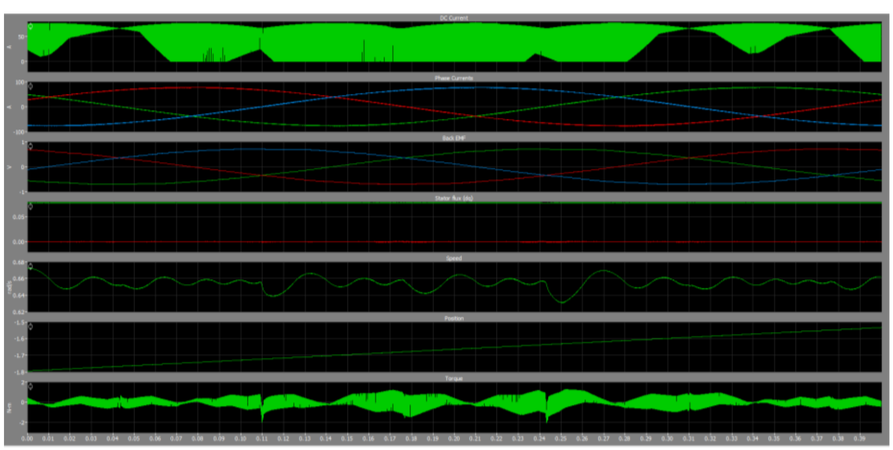
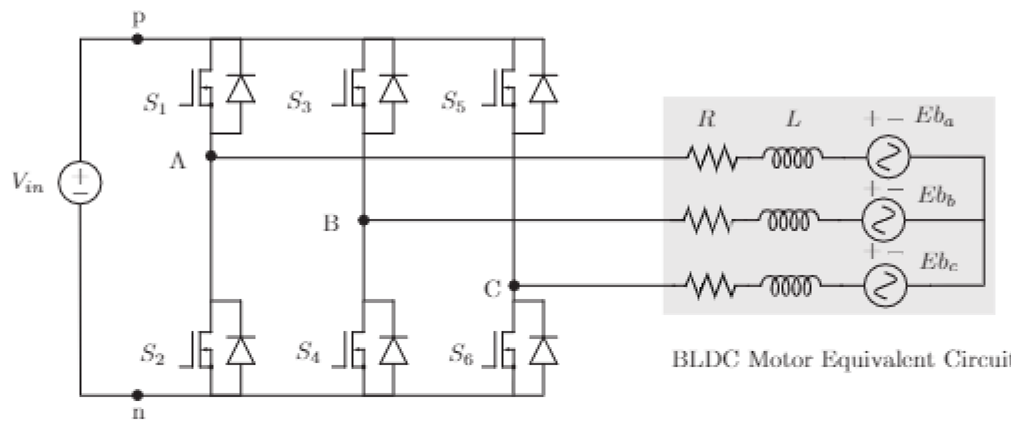


*Figure: Real-Time Stator DQ Currents (*Modulation Index = 0.9)

3. How does the back EMF magnitude change with the angular speed of the motor? Imagine you are on the bicycle zooming down a hill at top speed. What would happen in your drive circuit if the motor was spinning too fast? Describe an approach you can take to protect your electric circuit under this condition. Your approach may make use of additional hardware including switches, resistors, capacitors, etc.[**2 pts**]

$$E = \lambda_{max}\omega_e \cos(\theta_e)$$



As angular speed increases, the magnitude of the back EMF also increases. If the motor is spinning too fast, then the back EMF will eventually equal the supply voltage Vin, at which point the current and torque will both equal zero. If the speed continues past this point, then the back EMF can be greater than the supply voltage and reverse current through the inverter circuit. To protect the circuit, diodes can be implemented along wires A, B, and C. Another solution is to implement supercapacitors in between the supply voltage and the inverter circuit so that this energy can be captured and protect any circuitry that would be damaged by the back EMF.

3. Realtime Simulation of Closed Loop Control of BLDC Motor

3.2 Controller Design

1. Design a current controller based on the material covered in Section 3. The closed loop response of your controller should meet the following conditions. Use L, R_L from your lab in EE 452. If you did not take EE 452 or have some concerns, talk to the TA.

   - The steady state error for DQ current tracking should be less than 5%
   - The rise time should be less than 0.7ms.

   Tabulate the k_p, k_i values and the bandwidth and that you are targeting for the d and q axis controller. Also report the phase margin you achieve. [**4 pts**].

Bandwidth of the DQ axis current control loop:

$$\omega_{BW,dq} \geq \left\{\frac{2.5}{t_{r,max}} = \frac{2.5}{0.7 \times 10^{-3}} = \mathbf{3.5714 \times 10^3}\right\}\frac{\text{rad}}{\text{s}}$$

$$\omega_{BW,dq} \approx \omega_{cg,dq}$$

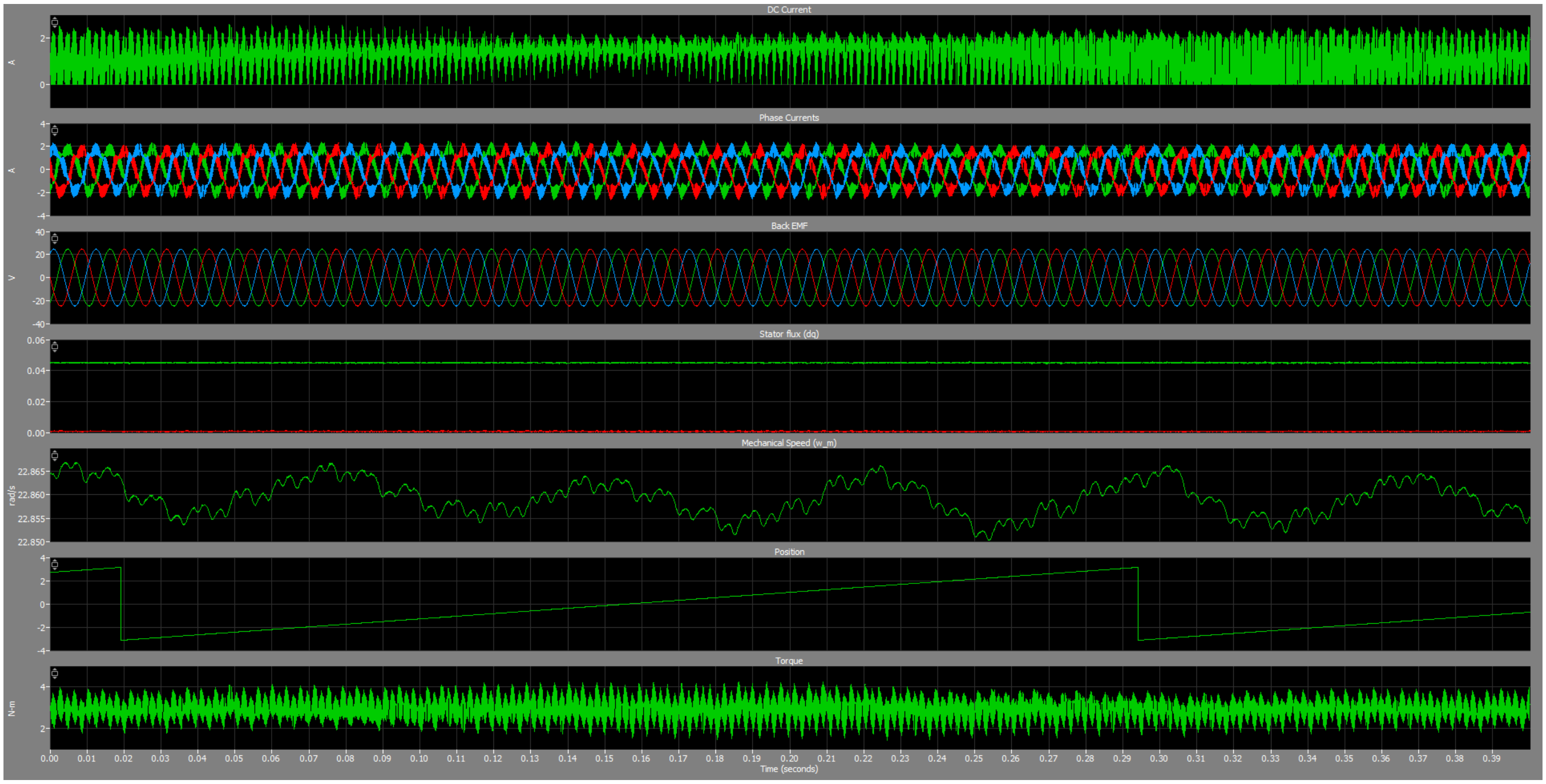
Open loop gain of the DQ axis current control loop:

$$\ell_{dq}(s) = G_{dq}(s) \cdot P_{dq}(s) = \frac{\omega_{cg,dq}}{s}$$

$$G_{dq} = K_{p,dq} + \frac{K_{i,dq}}{s} = \left\{\frac{\ell_{dq}(s)}{P_{dq}(s)} = \frac{\omega_{cg,dq}/_{s}}{\frac{1}{sL+R_L}} = (sL + R_L)\frac{\omega_{cg,dq}}{s} = \omega_{cg,dq}L + \frac{\omega_{cg,dq}R_L}{s}\right\} \Rightarrow \begin{array}{l} K_{p,dq} = \omega_{cg,dq}L \\ K_{i,dq} = \omega_{cg,dq}R_L \end{array}$$

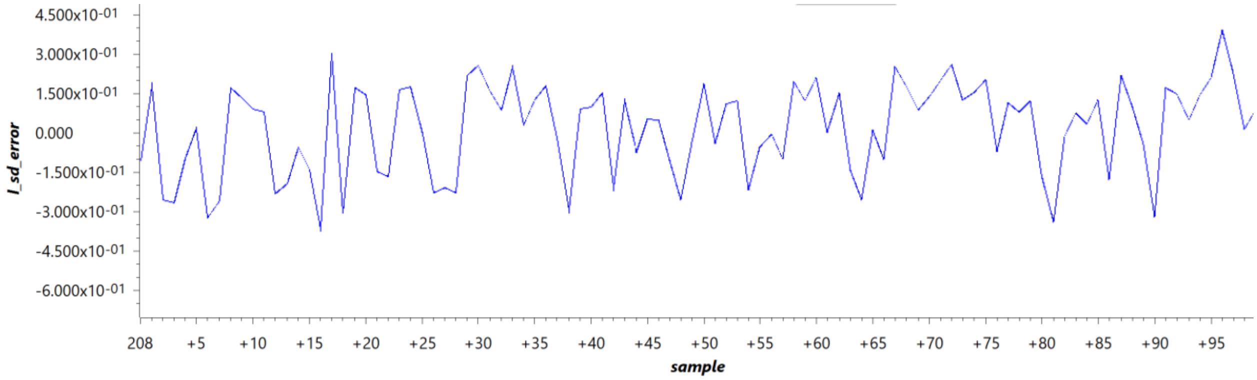
r_L stator resistance	$r_L = 292.5\text{e}-3$
L stator inductance	$L = 429.7\text{e}-6$
$\boldsymbol{\omega_{BW,dq}}$	$\mathbf{3.5714 \times 10^3}$ **rad/s**
$\boldsymbol{K_{p,dq} = \omega_{cg,dq}L}$	$\mathbf{1.5346}$
$\boldsymbol{K_{i,dq} = \omega_{cg,dq}R_L}$	$\mathbf{1.0446 \times 10^3}$
$\boldsymbol{PM_{dq}}$	$\mathbf{90^\circ}$

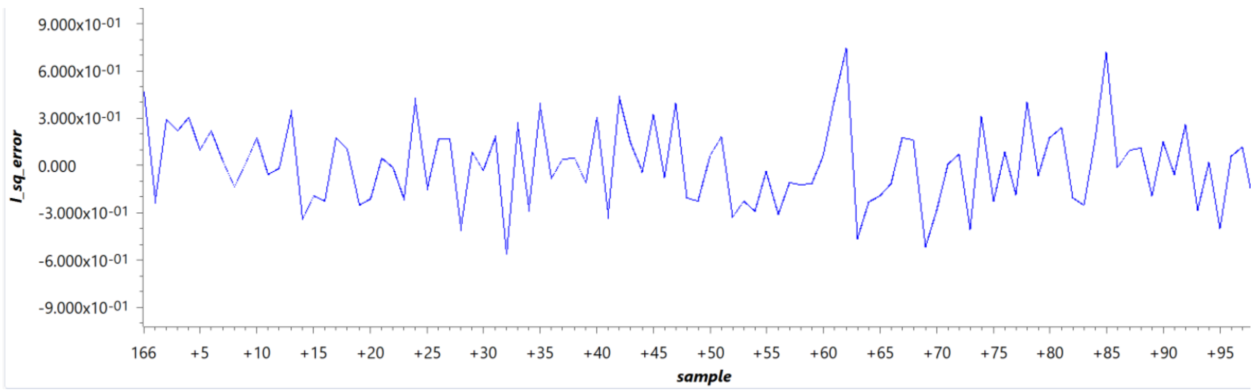
Generate plots in the PLECS to show that your current, back emf waveforms are sinusoidal, mechanical speed is almost constant and hall effect sensor error is small (if you start dumping in more power and your motor starts spinning very fast the time-accuracy of sector transitions is limited by your ISR execution at $50\mu sec$). **[10 pts]**



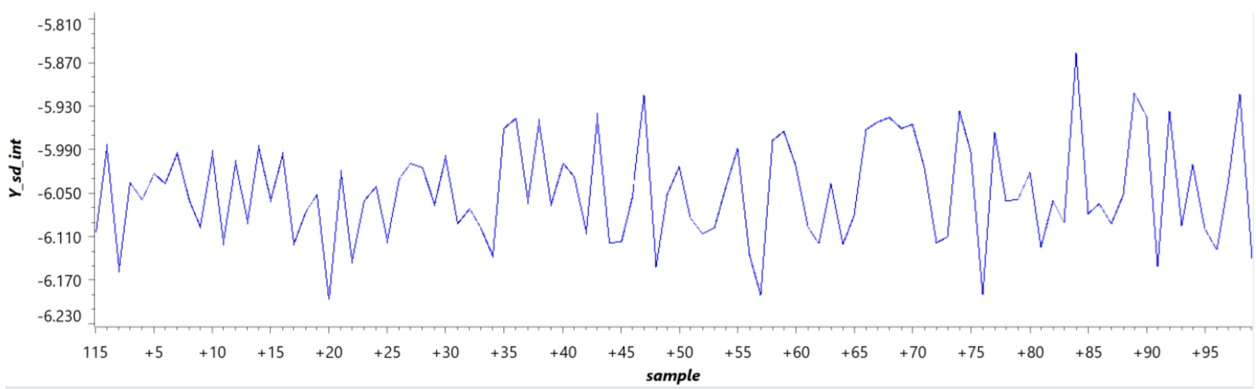
Verify in CCS that the error in d and q axis currents is very close to zero. Also show that your integrators are working fine and are not hitting limits of saturation. Finally, observe the three phase modulation indices. Plot them in different CCS graphs.**[10 pts]**

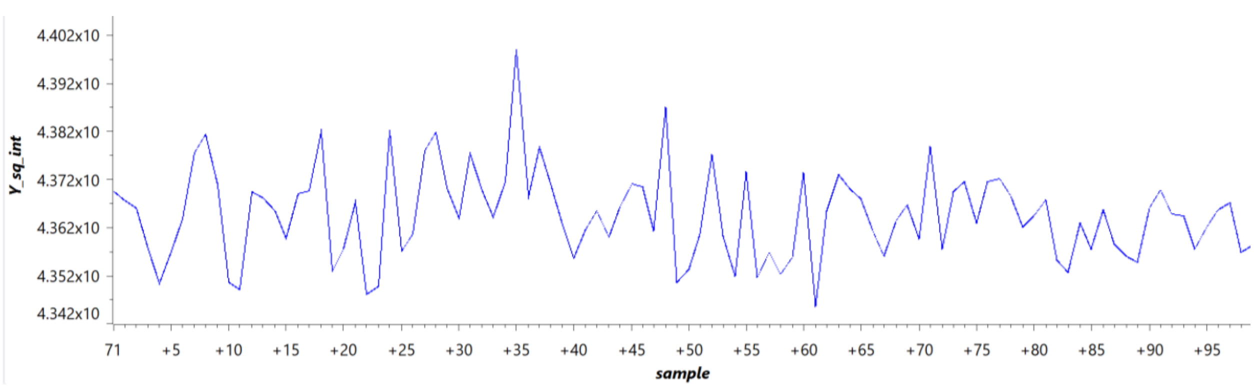
D & Q axis currents are close to 0:



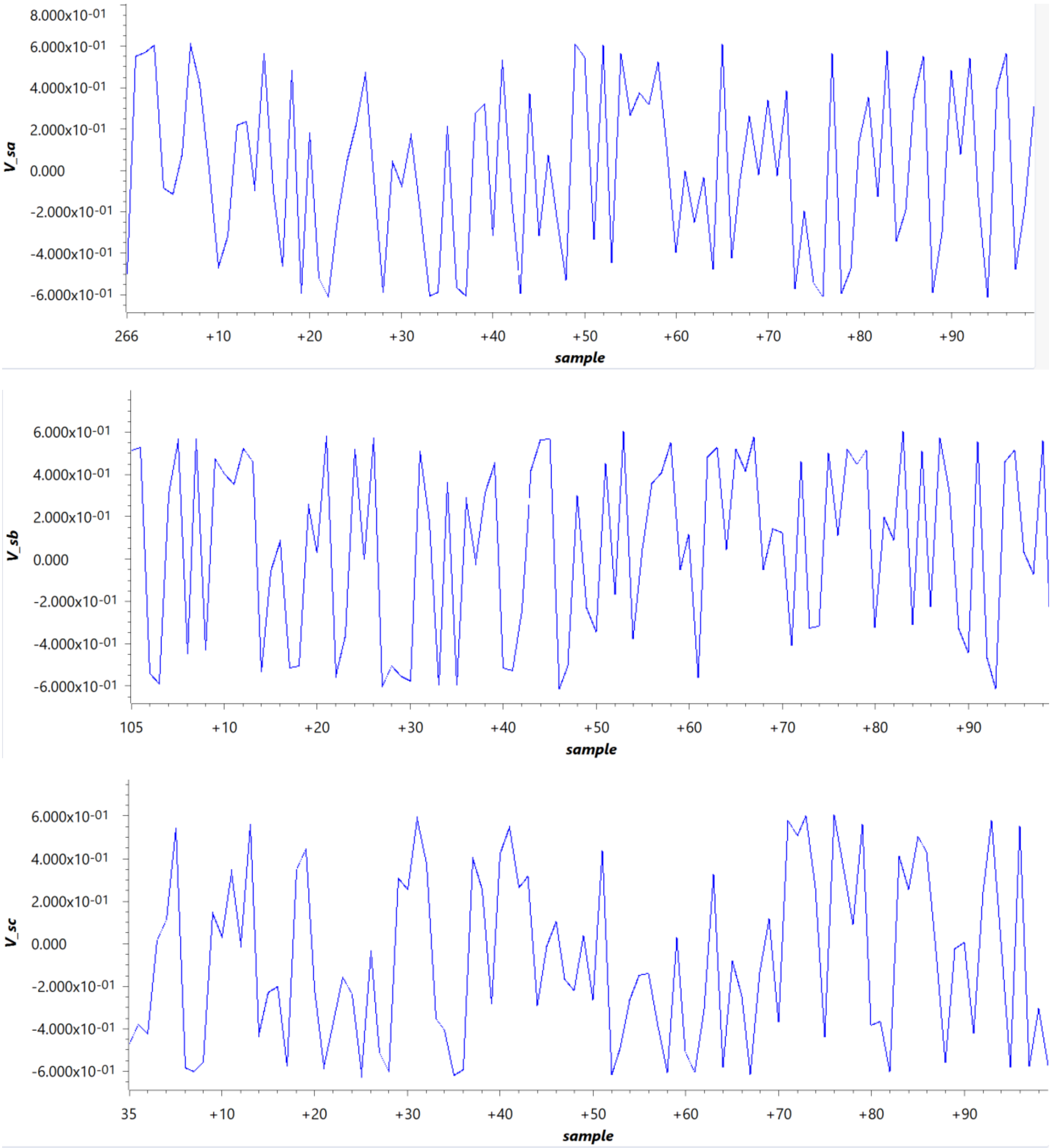


None of the integrators are hitting their saturation limits:

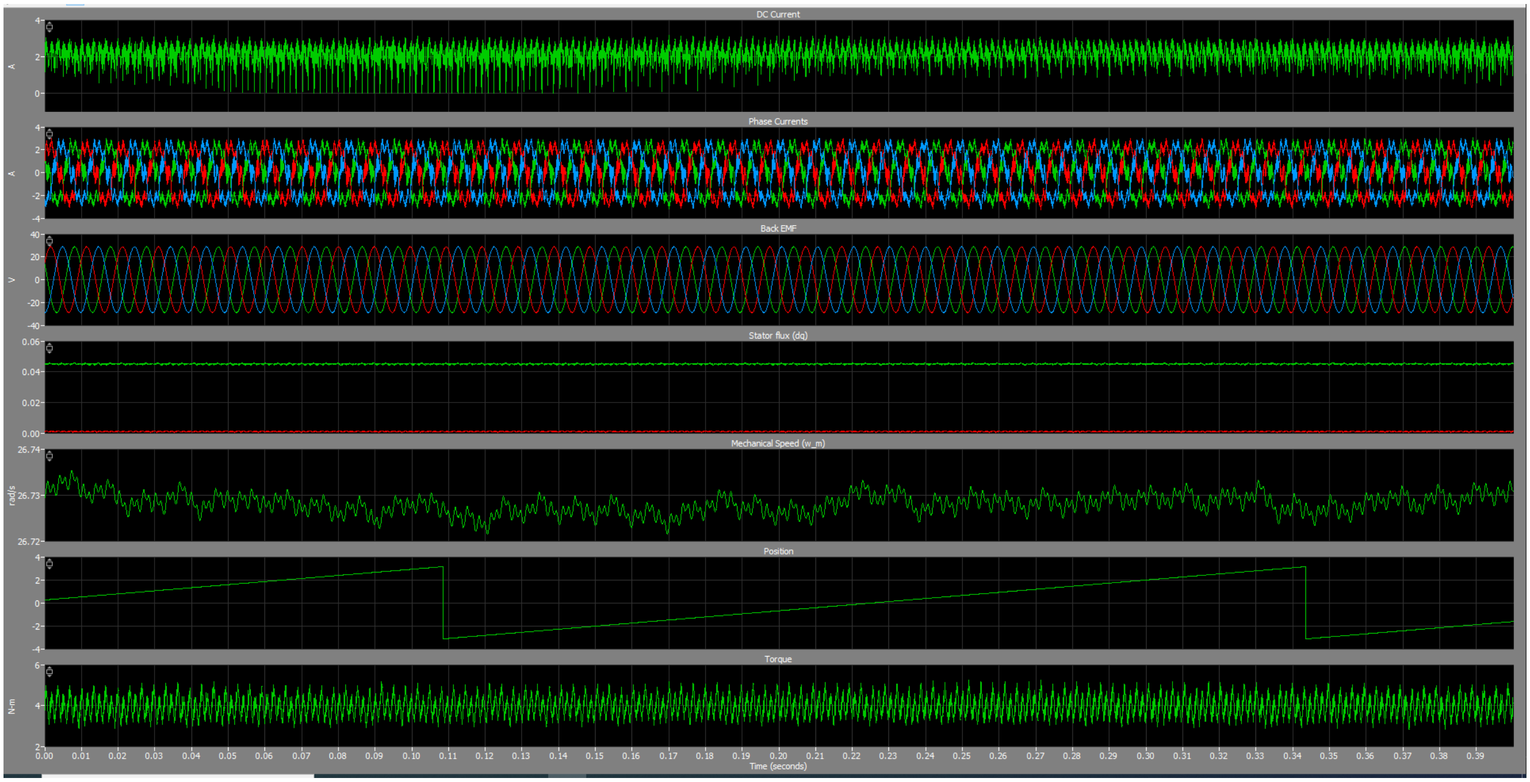




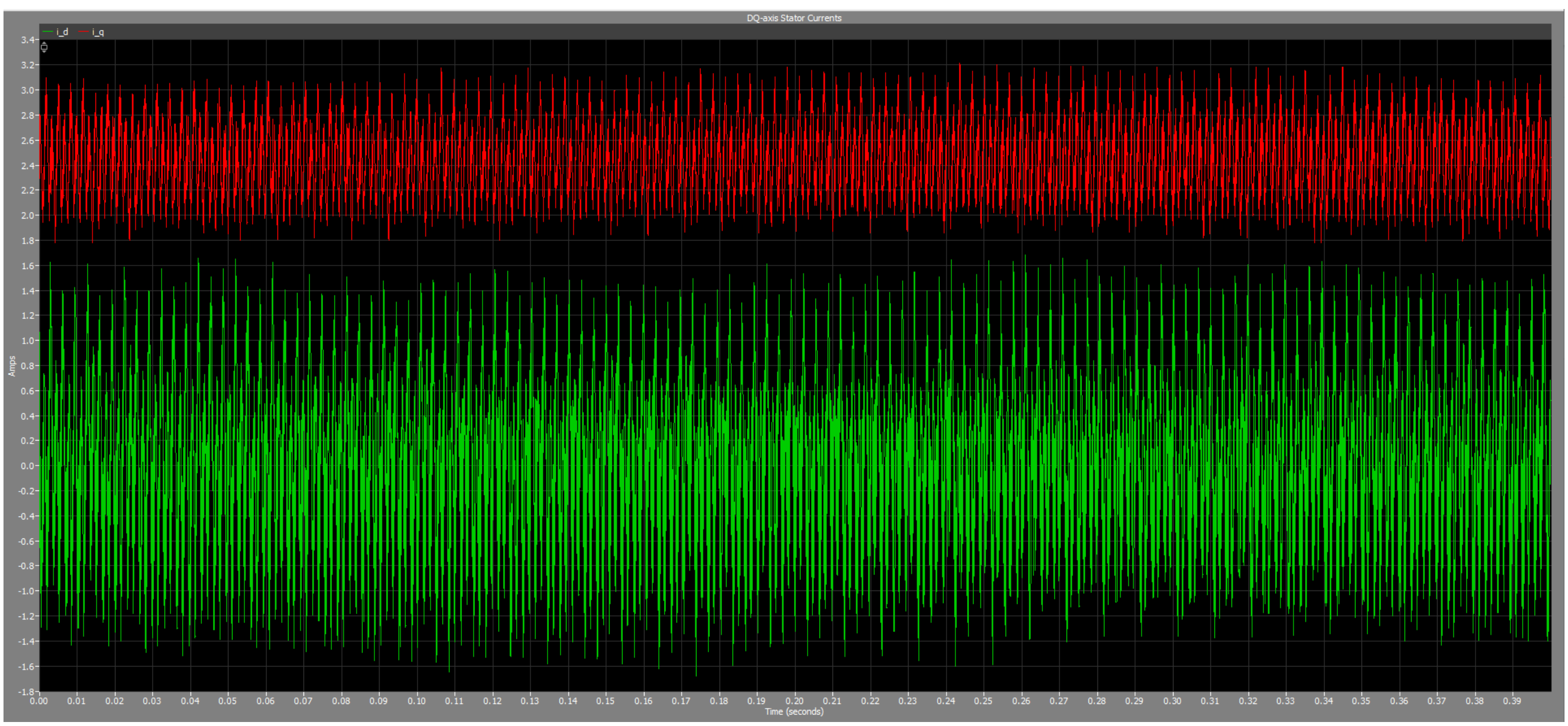
The 3 phase modulation indices are between -1 and 1:



3. Implement the decoupling and feed-forward to reject the disturbances from the controller. Show that the dynamic performance has changed from the one without feed-forward with waveforms which you think are suitable, such as by injecting a disturbance. **[5 pts]**.

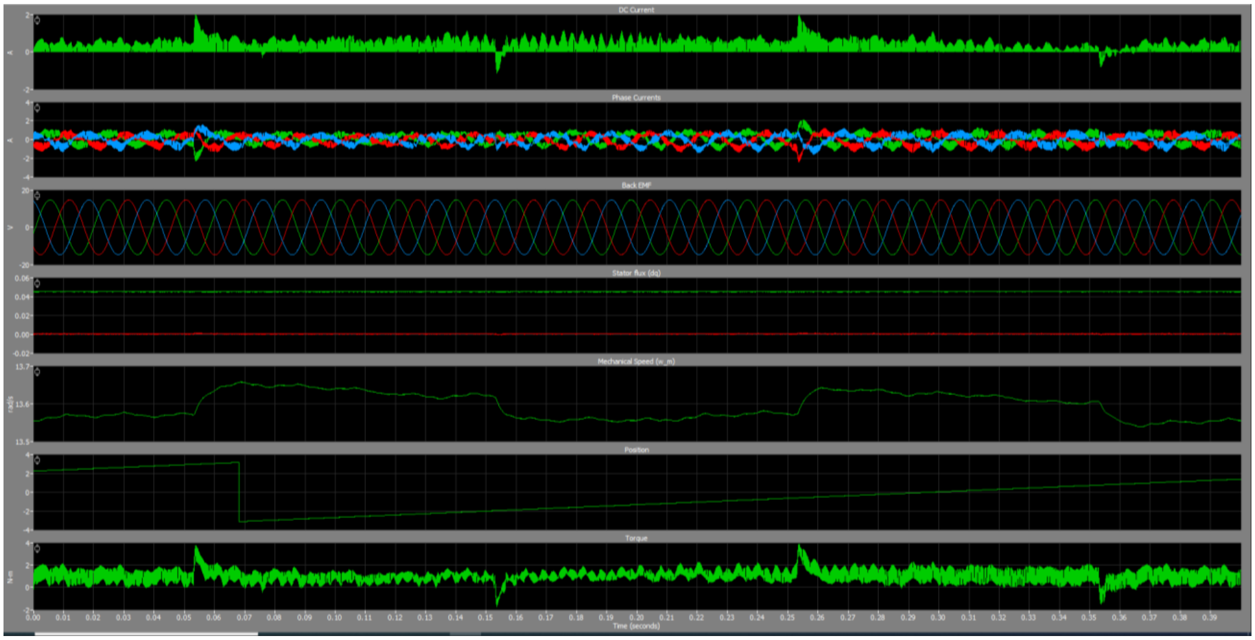


Seen below is a simple illustration of the operation of the current controller when we set $I_{d,ref} = 0$ Amps and $I_{q,ref} = 2.5$ Amps:

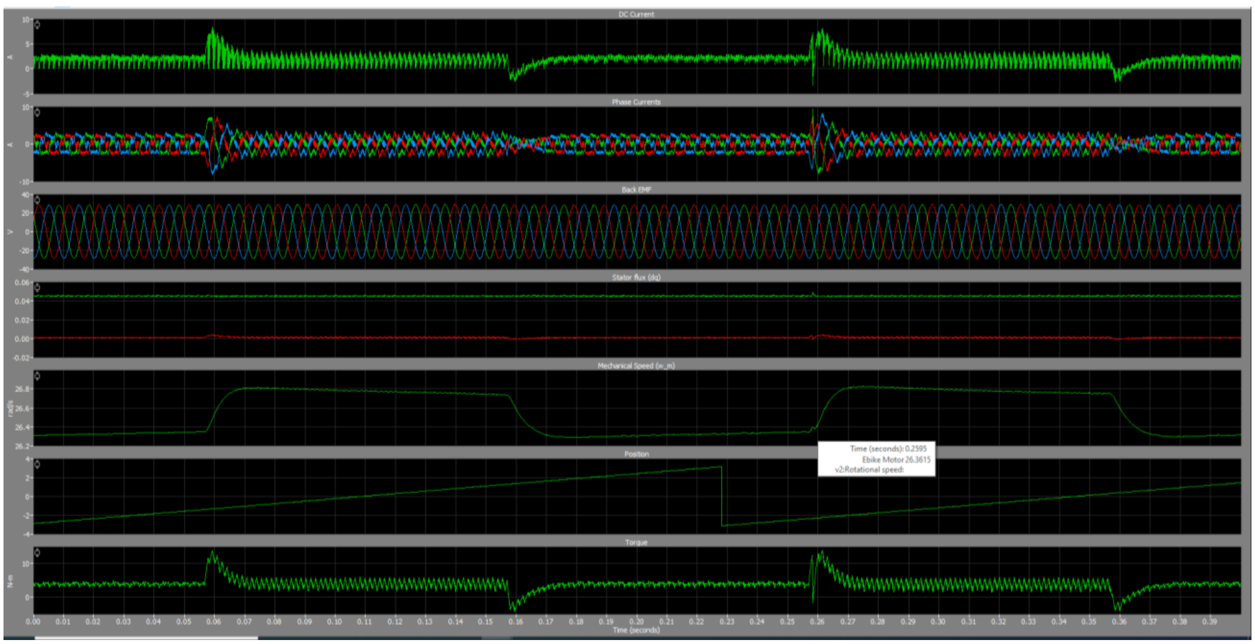


To test the efficiency of this system, we inject a disturbance. Specifically, we alternate the input voltage V_{in} between 50 Volts and 55 Volts at a rate of 5 Hz.

- Without feed-forward



- With feed-forward



As seen from the plots above, implementing feed-forward with decoupling improves our performance as we are able to improve our disturbance rejection.

5. Saturate and reset any integrators in your controller (follow last lab). The limits of your saturator blocks would follow from logical constraints like duty ratio cannot be greater than one or practical limitations of current and voltages. Describe what saturators have you used, including any integrator or PI controllers that you might have saturated. Also, explain what were the numerical values they were saturated to. **[5 pts]**

The saturation blocks we have used are as follows:

- D-axis stator current **PI** controller

```
#define Y_sd_min 0
#define Y_sd_max 50
```

- D-axis stator current **integrator** controller

```
#define Y_sd_int_min -50
#define Y_sd_int_max  50
```

- Q-axis stator current **PI** controller

```
#define Y_sq_min 0
#define Y_sq_max 50
```

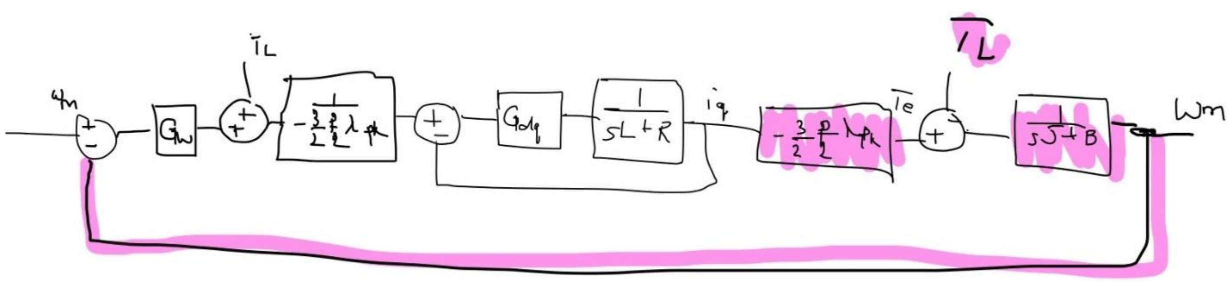
- Q-axis stator current **integrator** controller

```
#define Y_sq_int_min -50
#define Y_sq_int_max  50
```

The $\mathrm{q}-$axis saturation limits for both the integral were set to ± 50 and the overall PI controller saturation limits were set to $\{0,\ 50\}$ because 50 Amps is the maximum allowable current which we expect to flow in the $\mathrm{q}-$axis coil (based on prior calculations). The same saturation limits were set for the $\mathrm{d}-$axis integral and PI controller because they gave us better performance (although these limits should actually be scaled by half → i.e., 25.)

Questions:

2. The two currents, I_{sd} and I_{sq} references are set by you. What is the physical implication of each of the current. At the end of the day, you would be interested in changing the speed of the motor. Draw clearly a block diagram which related the speed controller with the present current controller, showing the output of the speed controller to be a "certain" current reference. You should have an inner-outer control loop architecture like the one in previous lab. [**10 pts**]



Question 2: Answer

I_{sd}^*	0
I_{sq}^*	2.5

Table: Stator Current Command References

4. Speed Control

1. Design a speed controller to track a desired speed with zero steady state error. Ensure the speed control loop is at least 10 times slower than the current control loop. Use J and B from the specification that the BLDC simulation model provides. Tabulate the $k_{p,\omega}$, $k_{i,\omega}$ values and the bandwidth and that you are targeting for the speed controller. Also report the phase margin you achieve. [**4 pts**].

Bandwidth of the speed control loop:

$$\{\Omega_\omega = \omega_{BW,\omega}\} \geq \left\{\frac{\omega_{BW,dq}}{10} = \frac{3.5714 \times 10^3}{10} = \mathbf{357.14}\right\}\frac{\text{rad}}{\text{s}}$$

$$\{\Omega_\omega = \omega_{BW,\omega}\} \approx \omega_{cg,\omega}$$

Open loop gain of the speed control loop:

$$\ell_\omega(s) = G_\omega(s) \cdot P_\omega(s) = \frac{\Omega_\omega}{s}$$

$$G_\omega = K_{p,\omega} + \frac{K_{i,\omega}}{s} = \left\{\frac{\ell_\omega(s)}{P_\omega(s)} = \frac{\Omega_\omega/_s}{\frac{1}{sJ+B}} = (sJ+B)\frac{\Omega_\omega}{s} = \Omega_\omega J + \frac{\Omega_\omega B}{s}\right\} \Rightarrow \begin{matrix} K_{p,\omega} = \Omega_\omega J \\ K_{i,\omega} = \Omega_\omega B \end{matrix}$$

J net moment of inertia of the bike and the rider	$J = 0.1127$
B cumulative friction coefficient of the shaft and road	$B = 0.001$
$\boldsymbol{\Omega_\omega = \omega_{BW,\omega}}$	$\mathbf{357.14\ rad/s}$
$\boldsymbol{K_{p,\omega} = \Omega_\omega J}$	$\mathbf{40.25}$
$\boldsymbol{K_{i,\omega} = \Omega_\omega B}$	$\mathbf{0.3571}$
$\boldsymbol{PM_\omega}$	$\mathbf{90°}$

2. For the closed loop control, you will implement a PI controller. Make sure your integrator terms in the PI controllers are limited appropriately. Also implement proper saturator and anti-windup for the speed controller. Note the controller limits and the anti-windup structure you have used. [**4 pts**]

Speed Controller Implementation:

```
/// Speed Controller
//////////////////////////////////////////////////////////////////////////////////

// D-axis Current Error
w_e_error = w_mph_ref - w_e;

// PI Controller
    // Error propagated through the proportional controller
Y_w_prop = w_e_error * Kp_w;
    // Error propagated through the integral controller
Y_w_int = 0.5 * T_samp * (w_e_error + w_e_error_prev) * Ki_w + Y_w_int_prev;

// Saturate Integral Controller Output
if (Y_w_int < Y_w_int_min) Y_w_int = Y_w_int_min;
else if (Y_w_int > Y_w_int_max) Y_w_int = Y_w_int_max;
else if (Y_w_int == Y_w_int_max && enable_anti_windup) Y_w_int = 0; // anti-windup

// Update stator d-axis values
    // Update the integrator
Y_w_int_prev = Y_w_int;
    // update the current error
w_e_error_prev = w_e_error;

// D-axis stator current PI Controller Output
Y_w = Y_w_prop + Y_w_int;

// Saturate PI Controller Output
if (Y_w < Y_w_min) Y_w = Y_w_min;
else if ( Y_w > Y_w_max ) Y_w = Y_w_max;

T_e = Y_w;

I_sq_ref = 4 / (3 * lambda_m * poles) * T_e;
```

Saturation for speed controller:

```
// Speed Controller Saturation Limits (all limits in mph)
    // Integrator Saturation Limits
#define Y_w_int_min -15 // 487.383 electrical rad/s = 20.3076 mechanical rad/s = 15 mph
#define Y_w_int_max  15
    // Net Saturation Limits
#define Y_w_min 0
#define Y_w_max 15 // 487.383 electrical rad/s = 20.3076 mechanical rad/s = 15 mph
```

Anti-windup structure:

```
else if (Y_w_int == Y_w_int_max && enable_anti_windup) Y_w_int = 0; // anti-windup
```

Note, w_e (electrical angular speed) is converted from electrical rad/s to mph by the conversion:

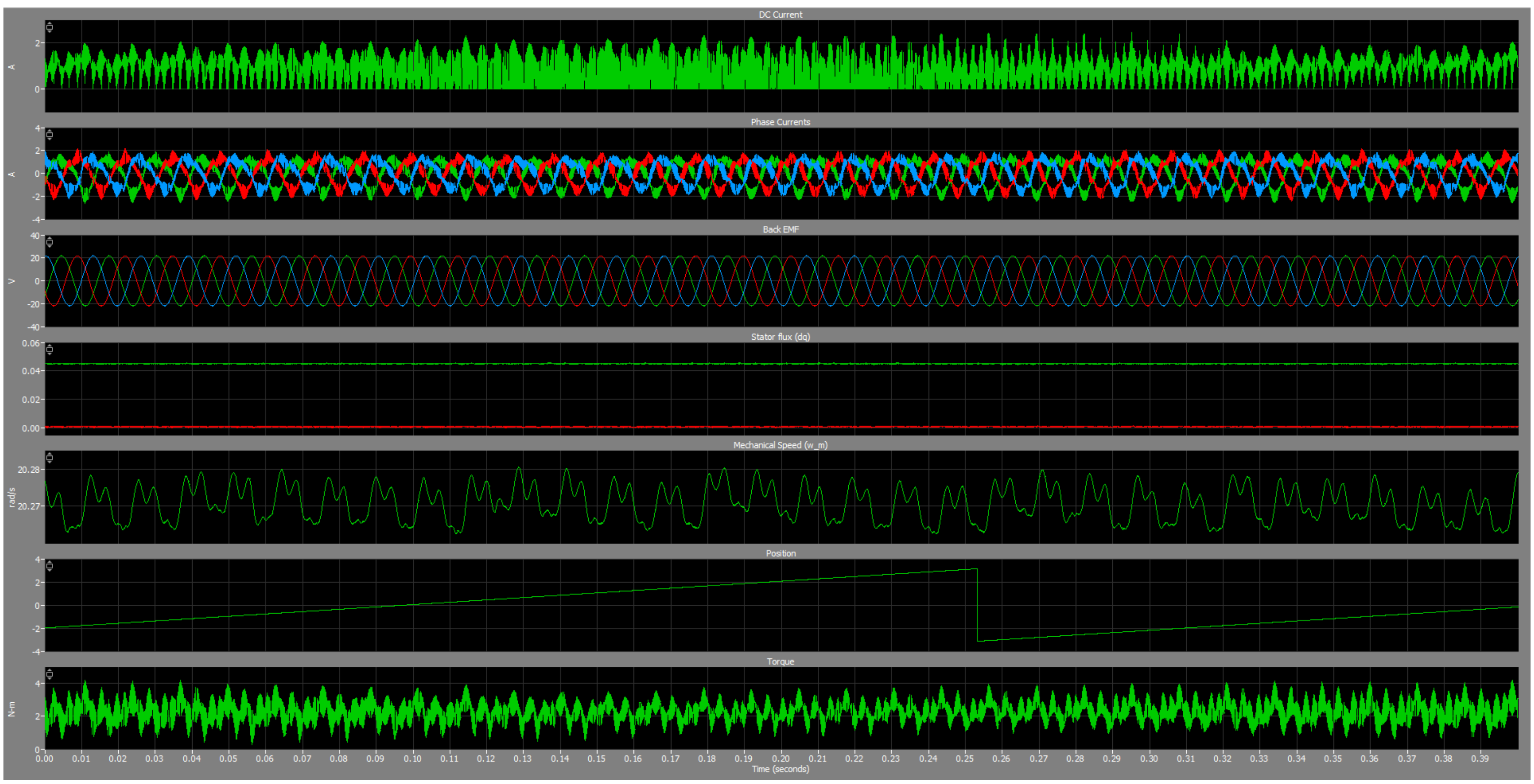
```
w_e =   (float)AdcaResultRegs.ADCRESULT4*(3.3/4096)*1000/3.3;

// Convert from electrical rad/s to mechanical mph
w_m = w_e * 2 / poles;
w_mph = w_m * 2.23694 * (13.0 / 39.37);

// Express w_e in mph
w_e = w_mph;
```

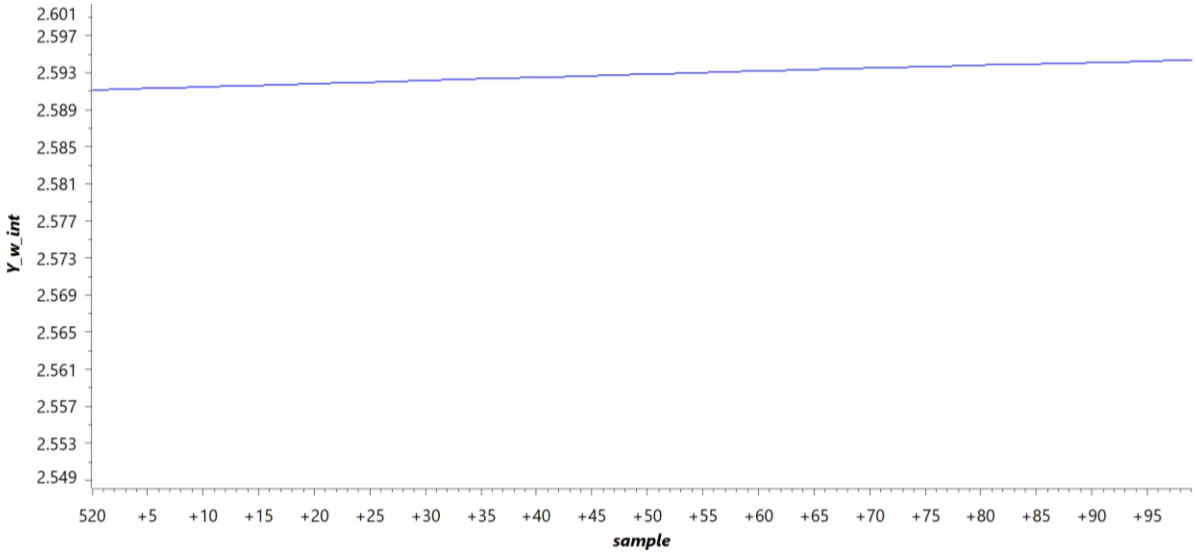
Note, because of the conversion of w_e from electrical rad/s to mph, all specified saturation limits are also in mph.

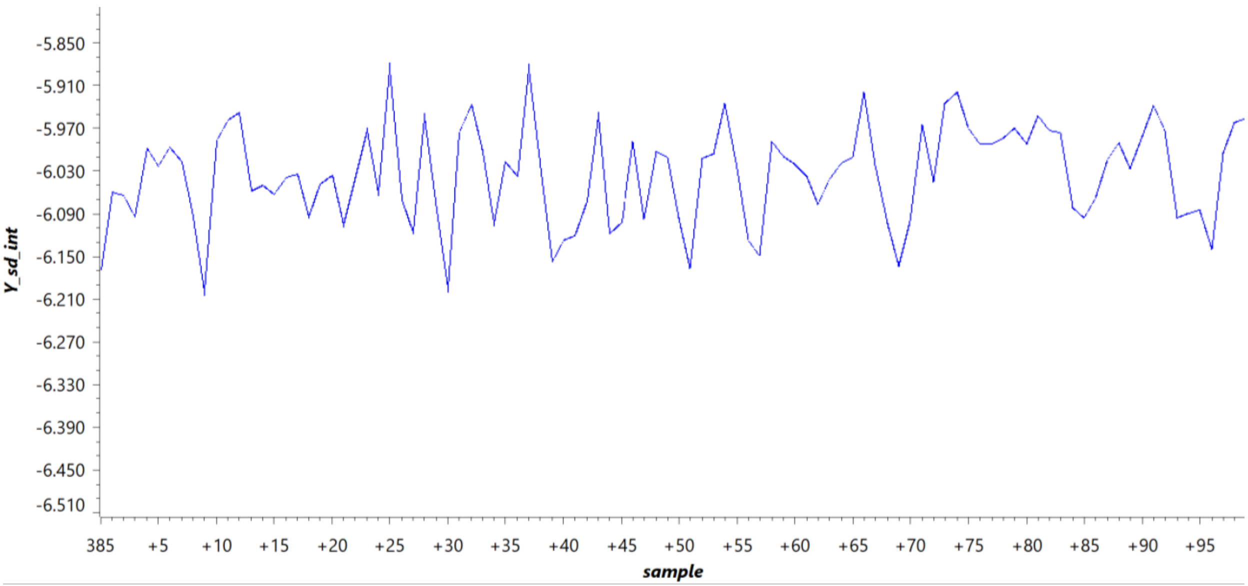
3. Generate plots in the PLECS to show that your current, back emf waveforms are sinusoidal, mechanical speed is almost constant and hall effect sensor error is small (if you start dumping in more power and your motor starts spinning very fast the time-accuracy of sector transitions is limited by your ISR execution at 50μsec). **[10 pts]**

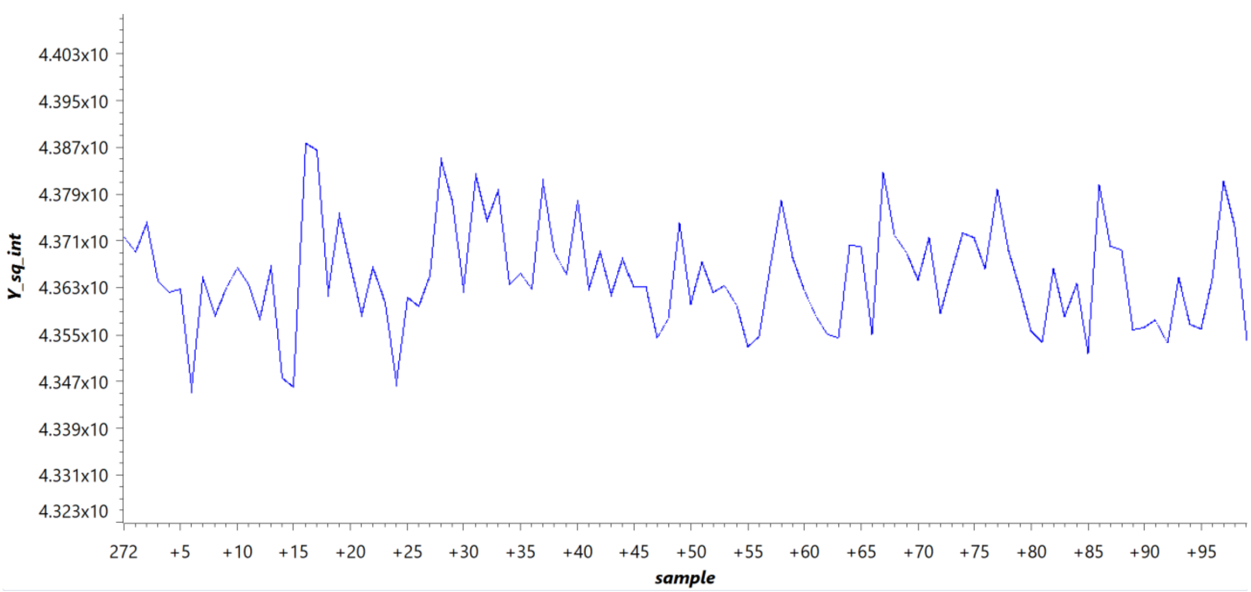


Verify in CCS that the error in speed is very close to zero. Also show that your integrators are working fine and are not hitting limits of saturation. Plot them in different CCS graphs.**[10 pts]**

None of the integrators are hitting their saturation limits:







Error in speed is very small: