EE 458/559 – Power Electronics Controls
Experiment 3 Lab Procedure: CLosed Loop Control of BLDC Motor

Introduction

The goal of this lab is to develop your understanding of brushless DC (BLDC) motors, drives, and controls. By the end of this lab you will have developed an open loop PWM controller for your E-Bike.

In this lab you will:

1. Learn how a BLDC motor can be driven dq-based current control.
2. Implement a modulation scheme for your BLDC Motor for a given modulation index.
3. Confirm your controller works in real-time.

1 Brushless DC Motors (Background Reading Only)

1.1 Motor Construction

BLDC motors consist of a permanent magnet rotor and a set of field windings on a stator. One common construction technique for E-Bike motors are commonly referred to as "hub" motors, where the electric motor is integrated into the wheel hub. The stator windings are contained within the hub and are mounted to the bicycle frame. The permanent magnets are arranged on the inner rim of the rotor that also connects to the bicycle spokes and wheel. The rotor will spin while the stator and field windings remain stationary.

Figure 1 shows an example of an E-Bike brushless DC hub motor. The rotor is on the left hand side of the figure, with the permanent magnets mounted on the inside of the rotor rim. The permanent magnets have an alternating north-south sequence. Each north-south combination forms a pole-pair. You can also see the connections for the wheel spokes that will connect to the outer rim. On the right hand side of the figure is the stator showing the stator field windings. The wheel axle connects through the stator and remains fixed. The stator is wound so there are several stator windings, but only three phase connections.

With the stator windings open, spinning the rotor at a fixed frequency will cause a changing magnetic field to cut the stator windings, inducing a voltage on the stator windings. The BEMF varies as a function of rotor angular position and angular velocity. The faster you spin the motor, the higher the BEMF. Based on how the stator windings are wound, the BEMF will either be sinusoidal (permanent magnet synchronous machines or PMSM) or trapezoidal (BLDC) function of the rotor position. Look at a the PLECS component help documentation for the BLDC model for more information and [2] for a comparison between PMSM and BLDC motors.

Figure 1: BLDC motor construction [1]

1.2 Theory of Operation

In order to rotate the motor, a rotating magnetic field must be induced by the stator. The motor windings are spatially arranged so there is 120° of electrical angle between phase A, B, and C. A simplified stator arrangement with one pole-pair per phase is shown in Figure 2. The 0° position is aligned with phase A and the angle increases in the counter clockwise direction.

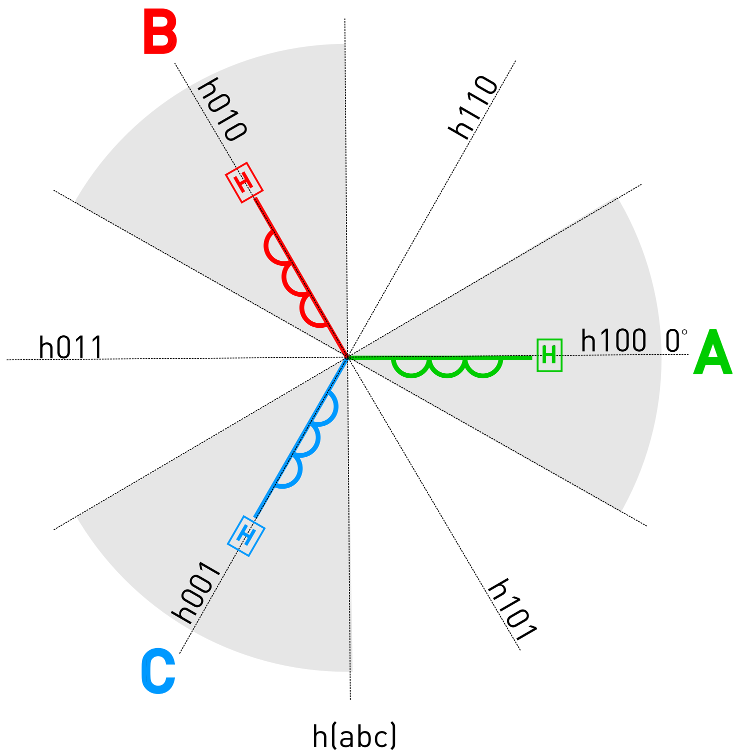

Figure 2: Motor winding and hall sensor arrangement

By controlling the stator phase currents using a three-phase DC to AC inverter circuit, a rotating magnetic field can be generated. The magnetic field generated by the stator should lead the rotor by as close to 90° as possible for maximum torque. Figure 3 shows the required stator field (cyan) to lead the rotor field (purple) by 90°. The stator field is generated by passing a current (red) into phase B and out of phase C of the stator. Phase B current generates a magnetic field vector opposite the direction of current, per the right hand rule, oriented at 120°. Phase C current generates a vector opposite the direction of current at 60°. The vectoral sum of these two fields is oriented at 90°.

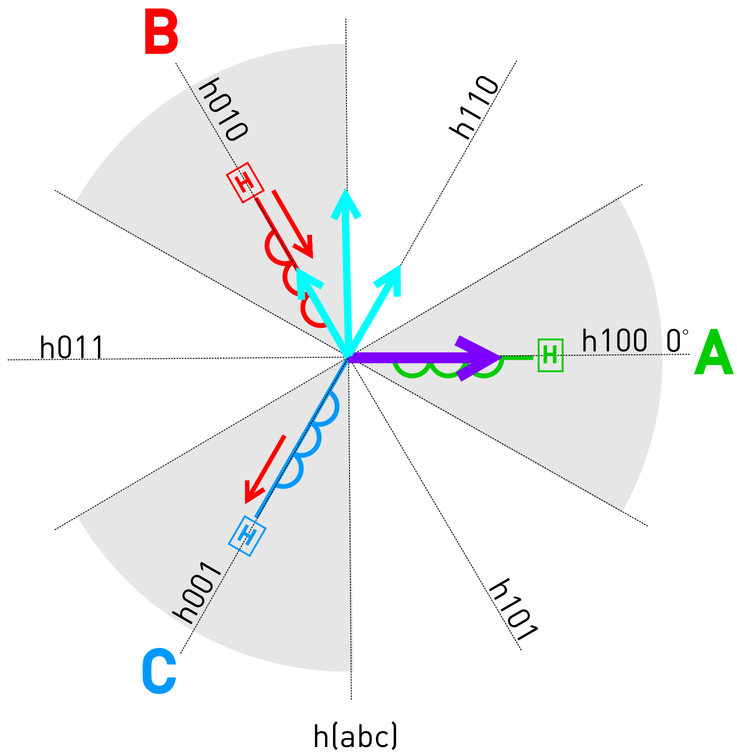

Figure 3: Current (red), rotor position (purple), and stator field (cyan) in sector 1

Current control can be achieved with a three phase DC to AC inverter, as shown in 4. The status of each switch is used to set the direction of current flow. The duty cycle of the switches controls the average voltage at the motor stator windings. PWM techniques can be used to increase or decrease the current flowing into the winding, and in turn change the stator field magnitude and motor torque.

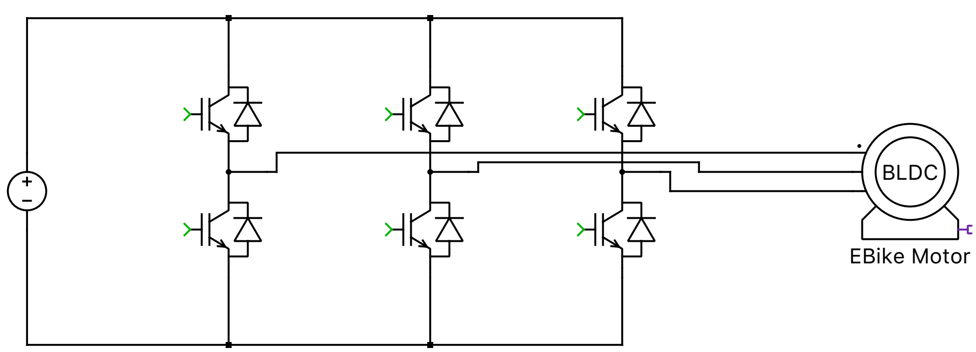

Figure 4: BLDC Drive Circuit

Also located with a separation of a 120° electrical degrees are a set of hall sensors, noted by the “H” icon associated with each phase in Figures 2 and 3. The hall sensors are used in sensored applications as a low cost means to detect the rotor position relative to the stator. The hall sensors detect a magnetic field. The sensor will produce a value of 1 if there is a positive magnetic field detected and a value of 0 if there is not a positive magnetic field. Here we are assuming the hall sensor will be active if the rotor north pole is within $\pm 90°$ of a sensor. For example, consider the rotor north pole is aligned with phase A at 0°. Only the phase A hall sensor will detect the field. If the rotor was moved so the north pole is alined just beyond the 30° position, then both the A and B hall sensors will be active. Just as the rotor spins past 90° the phase A hall sensor will turn off. Figure 2 shows how the entire 360° of the circle can be subdivided into a set of six different hall sensor states, or sectors, where each set of hall sensor readings is shown as h(abc).

Table 1 summarizes how the rotor position, hall sensor reading, and current flow relate in each sector. The commutation sequence shown in the table below generates a counter-clockwise rotating magnetic field, resulting in positive electrical torque and speed.

Table 1: BLDC Commutation Table

Rotor Electrical Angle (θ_e)	sector	Hall A	Hall B	Hall C
$-30° \leq \theta_e < 30°$	1	1	0	0
$30° \leq \theta_e < 90°$	2	1	1	0
$90° \leq \theta_e < 150°$	3	0	1	0
$150° \leq \theta_e < 210°$	4	0	1	1
$210° \leq \theta_e < 270°$	5	0	0	1
$270° \leq \theta_e < 330°$	6	1	0	1

1.3 E-Bike BLDC PLECS Model

The E-Bike motor parameters should be filled up from the previous quarter lab. If you did not do this lab, talk with your TA.**[1X4 pts]**

- Number of poles: p
- Combined stator and mutual inductance: L
- Stator resistance (r_{L}):
- Back EMF Flux Constant (λ_m) (recall: $K_E = \frac{p}{2}\,\lambda_m$):

It is recommended you review what these parameters mean in the PLECS motor model and set these values correctly in the machine model. Refer to [2] for a more in depth treatment of the subject.

2 Realtime simulation of BLDC in Open Loop

In this section you will implement a modulation approach to drive the motor. You will test out the different sections of the pwm and feedback which will set the stage for the closed loop control in the upcoming lab. **For this part, use the BLDCOpenLoopLab.plecs starter file from Canvas.**

Following are the components that you need to implement in CCS code.

2.1 Three phase PWM signal

1. From the previous lab, you might remember that you have configured EPwm1 to switch the boost converter devices. In this lab, you will configure EPwm 2,3,4 to switch each of the three legs as shown in 4. EPwm2A drives the top switch of A phase and the EPwm2B output drives the bottom switch of phase A. Similarly the EPwm3 and EPwm 4 are reserved for the B and C phases respectively. You have to implement sinusoidal PWM. The following lines of code will be helpful. Henceforth we shall refer to the frequency of this sinusoid as the stator frequency, ω_s.

```
    // The following generates the frequency of the PWM
    // The PWM frequency is the frequency at which the
    // stator field rotates, and hence the rotor too
    //In this code, w_s is the stator frequency
    float w_s = 2*pi*10;
    float f_div = 2*pi*fsamp/w_s;
    if(wt>2*pi)
         {
             wt=0;
         }
         wt = wt  + (2*pi/f_div);
         //f_div = 20000/10  This means we want 10 Hz sinusoid.

    // EPwm 2,3,4 is set up for A,B and C phase respectively

    EPwm2Regs.CMPA.bit.CMPA = (0.5+ 0.5*mod_index*sin(wt))*N;
    EPwm3Regs.CMPA.bit.CMPA = (0.5+ 0.5*mod_index*sin(wt-al))*N;
    EPwm4Regs.CMPA.bit.CMPA = (0.5+ 0.5*mod_index*sin(wt+al))*N;
```

   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**]

2.2 Estimation of angle and angular velocity [Skip 2.2 for now due to time constraints; if you have time you may do this for extra credit]

In this section you will estimate the angular velocity, θ from the GPIO signals. For this, we first need to define the three GPIO. The GPIO 33,34,35 (see Appendix) are the pins which collect the information from A, B and C phase hall effect sensor respectively. Now, create a map between the hall effect sensors and sector information. You could use switch statement, if-else statements or other relatively simpler options to obtain the information of the sector from the Hall effect sensor outputs. So your sector would start incrementing sector one to six and then roll over to one. Use the Table 1 to find out the sector information from the Hall effect sensor outputs. Store this Sector information in a variable (array) which you can then plot in realtime using the *Graph* feature of the CCS. Also, using the graphs, plot the three GPIO inputs of the HALL. Capture these four channels/graphs in a single screen and attach in your reports. [**0 pts**]

2.2.1 Angular velocity estimation

If all the hall sensors have the same reading, it indicates damaged sensor. Your code should take care of this condition. Now in your code, define a new integer variable which will keep a counter that keeps incrementing everytime an ADC interrupt occurs and you enter the subroutine interrupt void adcA1ISR(void). Keep incrementing the counter till you have not obtained two sector changes. We use two sector changes to reduce noise in our angle estimator. Once you have obtained two sector changes, reset the counter. When you obtain two sector changes, that means one third of an electrical cycle has elapsed, since there are six sectors in total. Let us say the value of counter is N_{counter}. This means that the time taken for one electrical cycle is, $N_{\text{counter}}T_{\text{samp}}$, where T_{samp} is the sampling time or $50\mu sec$ (when implementig 10 kHz PWM with peak-valley sampling). You can obtain the angular frequency of the motor as follows

$$\omega = \frac{2\pi}{3(N_{\text{counter}}T_{\text{samp}})} \tag{1}$$

Once you have estimated this angular frequency, record a screenshot of this (from CCS watch window) and show that this matches well with the actual stator frequency, ω_s. As you increase the ω_s you will notice that the estimated and the actual speed might differ. One cause of this is the limited time-resolution of the $50\mu sec$ sampling of the hall sensor inputs and the integer nature of N_{counter}. There are other ways to overcome this that are not discussed in the lab. [**0 pts**]

2.2.2 Angle estimation

Let us now define two new variables, θ_c and θ_f. We obtain the crude information of angle from θ_c, which can be set to a value as follows,

$$\theta_c = (sector - 1)\frac{2\pi}{6} - \frac{\pi}{6} \tag{2}$$

Let us say we have just transitioned from sector 6 to sector 1. From Fig. 2, this can be identified by a sudden transition of hall effect sensor of C phase from 1 to 0. When traveling in the counter-clockwise direction (positive speed) we know that the electrical angle at the moment of transition is $-\pi/6$ and so for this entire sector, we define $\theta_c = -\pi/6$. Similarly, at the moment of transition from sector 1 to sector 2, we know the angle is $+\pi/6$ and so on.

This also means that using the hall sensors alone to estimate the angle, the value of the angle remains constant within each sector and only changes in discrete jumps of $\pi/6$ at the beginning of every sector transition. This estimate of the rotor angle is poor and is the correct rotor angle only at the sector transitions. To rectify this, we add a correction factor called, θ_f which signifies the finer resolution of the angle within each sector. θ_f is an estimated angular offset that is obtained by integrating the electrical speed of the rotor which has already been estimated by (1). Make sure to limit this integrator output to a value of $\pi/6$ since that is the largest value θ_f can take within each sector (otherwise a hall sensor transition should have occurred, or will occur shortly).

Finally obtain the actual rotor angle by adding them as follows,

$$\theta = \theta_f + \theta_c \tag{3}$$

In CCS, plot this variable, θ using the graph feature. **[0 pts]**

Now use another ADC to sense in the actual electrical angle of the rotor from the PLECS. Plot this variable in a CCS graph as well. Show that these two match exactly. **[0 pts]** There will be a certain delay between the estimation of the angle and its actual value. What is the theoretical delay between them? **[0 pts]**

2.3 ABC to DQ conversion for stator currents

1. In this section, we use the information of the rotor angle, θ. We know that the d and the q axis are rotating axes, while the abc or the $\alpha - \beta$ are the stationary axes. But, we need to align the rotating axis with the stationary axis at the beginning of the machine's rotation when $\theta = 0$. Therafter, as the machine rotates, the angle of the d (or q) axis changes with respect to that stationary axis in a speed synchronous with the rotor speed. For our case, we will align the d axis with the a axis. This requires that the angle with which we do the frame transformation, needs to be aligned with the A-axis as shown in 3. In other words, the theta needs to start at zero, everytime the rotor passes through the A-axis. From our previous discussion, we notice that the $-\pi/6$ factor in θ_c is used to align the A-axis with zero electrical degrees.

2. Use three ADC channels (recommend A3, A5, A6) to sense the stator ac currents. This is different than all the previous sensing exercises since we are sensing ac quantities now. The **offset** parameter in the PLECS "Analog Out" will now need to be set to 3.3/2. Let us take for example, we want to measure the A phase stator current through the ADC. Let the actual circuit current be $i_{s,a}^{\text{PLECS}}$. Let the ADC scaling factor has to be $3.3/(2I_{\max})$ and the offset has to be 3.3/2. Thus, the ADC result register would read a digital value as follows:

$$i_{s,a}^{adc} = (4096/3.3)(i_{s,a}^{\text{PLECS}} \frac{3.3}{2I_{\max}} + \frac{3.3}{2}) \tag{4}$$

   You may use an Imax = 100A. Once you obtain the digital word, $i_{s,a}^{adc}$ obtain the actual value of the stator current in CCS by some mathematical manipulation, and let us call it $i_{s,a}$. You have to repeat similar operations for the other two phases. Obtain the other two currents as $i_{s,b}$ and $i_{s,c}$. In the next few lines, we will obtain the d and q axis components of the stator current which are denoted as, $i_{s,d}$ and $i_{s,q}$ respectively.

   The "angle" used for this DQ transformation is usually estimated by the hall sensors. For this lab, you may read in the angle from PLECS on a 4th ADC (recommend ACD A9) using a scaling factor of $ceil(3.3/2\pi) = 3.3/7$.

```
    #define al 2.094395102393195f // This is 2pi/3
      // ABC to DQ
Isd =  0.66667*(Isa*cos(angle)+Isb*cos(angle-al)+Isc*cos(angle+al));
Isq = -0.66667*(Isa*sin(angle)+Isb*sin(angle-al)+Isc*sin(angle+al));
```

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**]

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**]
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**]
4. [**Optional for everyone**]Repeat the first question for a generating mode of the BLDC machine. You can simulate this case in PLECS with the BLDC motor driven by a torque source and the electrical stator terminated by some resistances.[**6 pts**]
5. [**Optional for everyone**]Repeat the first question with a new frame transformation wherein you now lock the q-axis to the A-axis at the start of operation in stead of the d-axis. Use the same notation that the d-axis lags the q-axis by 90 deg.[**6 pts**]

3 Realtime simulation of Closed Loop Control of BLDC Motor

The BLDC motor drive is a three phase inverter. The inverter is fed through a boost converter which boosts a 25 V input source to a 48-50 V intermediate DC bus voltage. The Fig 5 nomenclature convention follows prior lab manuals. The power rating of the BLDC motor is 250 W.

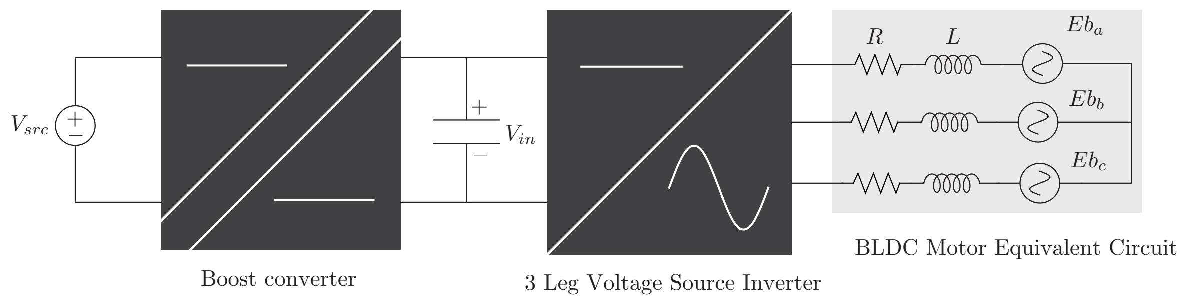

Figure 5: The figure shows the integrated system. The boost converter has a low voltage dc source at its input that is boosted at a voltage suitable for the motor operation. The boost converter controls the output voltage which then ensures a stiff dc link voltage for the inverter. The three phase inverter is the drive stage for the motor. The dc link capacitor provides the decoupling of the two systems.

Any machine can be operated in the four quadrants of operation wherein the axes are speed and torque. We fix the sign convention by the power transfer direction. When the product of the speed and torque (power) is positive, we assign the machine to be in the motoring mode. Motoring mode corresponds to the machine operating as an electric load. The load consumes power when the voltage across its terminal and current into the positive voltage terminal have the same polarity. When either the direction of speed or torque (not both) changes, the machine is said to be working in the regenerative mode. In such a condition, the power consumed by the machine is negative or in other words it generates a power.

A typical example is a machine driven by a mechanical power source. When a mechanical power source rotates the machine shaft in the positive direction and a negative electrical torque is applied, the mechanical power from the rotor of the machine is converted to electrical power in the stator, that is then transmitted from the stator to a storage through an inter-stage converter.

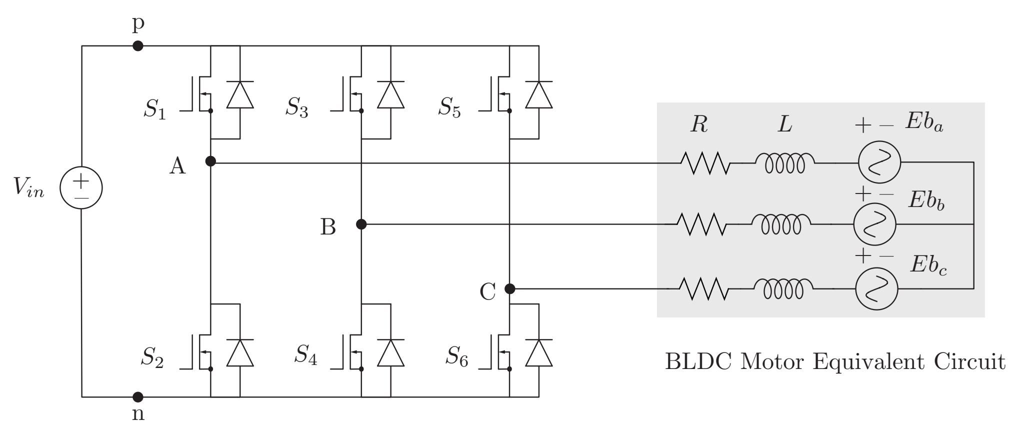

Figure 6: The figure shows the three phase inverter as the BLDC motor drive. The motor winding lumped inductance, L, and resistance, R (also referred to as r_{L} later in the document) and the back emf for three phases is shown.

For a BLDC motor, the back emf generated due to the permanent magnet rotor is given as,

$$E_q = \lambda_m \omega_m \frac{p}{2} = \lambda_m \omega_e, \tag{5}$$

where λ_m is the motor back EMF constant in volt-seconds, p is the number of poles, and ω_m is the mechanical speed in mechanical-radians/sec, and ω_e is the electrical speed which is numerically expressed as, $\omega_e = \omega_m p/2$. We equate the electrical power $P_e = 1.5(E_d I_d + E_q I_q)$ to the rotational power, $T_e \omega_m$, to obtain the electrical torque, T_e as follows,

$$T_e = \lambda_m \frac{3}{2} \frac{p}{2} I_q. \tag{6}$$

Thus, the back EMF is directly proportional to the rotor speed and the torque generated is directly proportional to the current in the stator windings. Equations (5)–(6) give a link between the electrical quantities like back-emf and currents to their mechanical counterparts of speed and torque. This means that the quadrant operation using mechanical parameters like speed and torque can be easily supplanted with electrical parameters like back EMF and the armature current.

This is useful since once we retrieve the information about the current and back EMF, or more particularly their signs, we can easily restrict the motor operation into a limited set of quadrants. This greatly simplifies the motor drive strategy. In our simplified controller, the generating modes are not of our interest, hence we will generally operate in the quadrant 1.

3.1 Current Controller

The plant transfer function needs to be derived first. For this, we follow through the lecture notes and your Assignment 5, and notice that under the operation of circuit in Fig 5,

$$\begin{bmatrix} V_{\mathrm{inv,a}} \\ V_{\mathrm{inv,b}} \\ V_{\mathrm{inv,c}} \end{bmatrix} = \begin{bmatrix} sL + r_{\mathrm{L}} & 0 & 0 \\ 0 & sL + r_{\mathrm{L}} & 0 \\ 0 & 0 & sL + r_{\mathrm{L}} \end{bmatrix} \begin{bmatrix} I_{\mathrm{s,a}} \\ I_{\mathrm{s,b}} \\ I_{\mathrm{s,c}} \end{bmatrix} + \begin{bmatrix} E_{\mathrm{b,a}} \\ E_{\mathrm{b,b}} \\ E_{\mathrm{b,c}} \end{bmatrix},$$

where $V_{\mathrm{inv,a}}$ is the voltage that appears at the inverter terminal a, $I_{\mathrm{s,a}}$ is the stator current through a phase and the back emf for a phase is denoted as, E_{ba}. The L and r_{L} are effective motor inductance and resistance respectively.

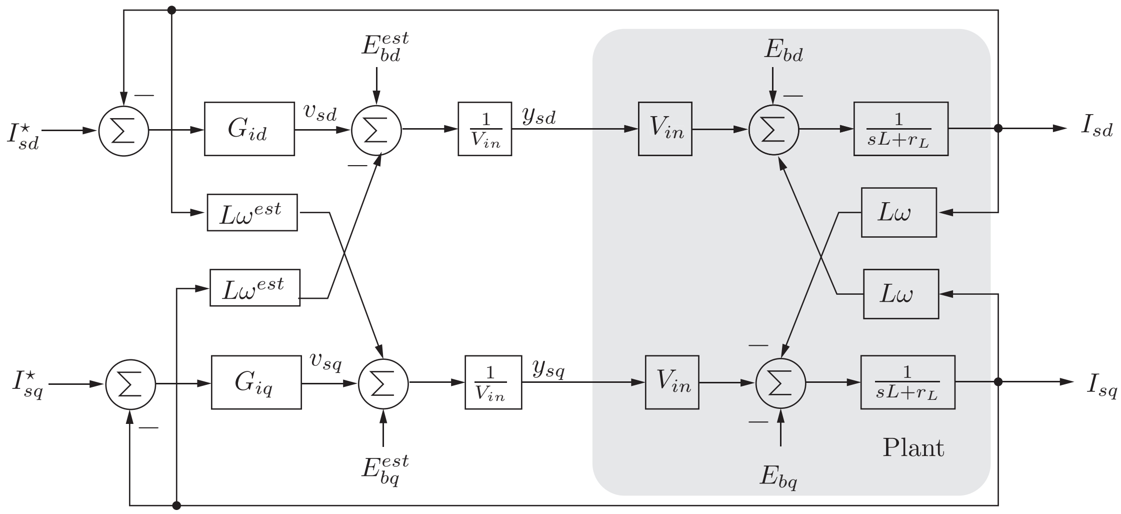

Figure 7: The figure shows the control implementation of the BLDC motor drive in the dq domain. The shaded portion represents the circuit laws that the hardware or real time system implements. Since we want to obtain a first order plant of the form $1/(sL + r_\mathrm{L})$, we need to implement appropriate feedforward. The back emf and the speed have been estimated for this and those quantities are super-scripted with "est". Depending on the frame transformation convention, one of the estimated back emf, $E_{bd}^{est}, E_{bq}^{est}$ would be zero. In actual control implementation, the feedback variable I_{sd}, I_{sq} are obtained from the actual three phase stator currents by abc-dq transform, whereas the two duty ratios, y_{sd}, y_{sq} are used to generate the three phase PWM reference signals after a dq-abc transform. The feedforward quantities are usually very slow varying and popular approximate control solutions do use only v_{sd}, v_{sq} with $1/V_{in}$ scaling to obtain the d-q domain reference signals too.

Let us do a abc-to-dq transform and obtain the following,

$$V_\mathrm{inv,d} = (sL + r_\mathrm{L})I_{sd} - L\omega I_{sq} + E_\mathrm{bd}$$
$$V_\mathrm{inv,q} = (sL + r_\mathrm{L})I_{sq} + L\omega I_{sd} + E_\mathrm{bq}$$

By proper choice of alignment we can set either $E_\mathrm{bd} = 0$ or $E_\mathrm{bq} = 0$.

Hence the plant transfer function can be written as follows,

$$I_{sd}(s) = \frac{y_{sd}V_\mathrm{in} - E_\mathrm{bd} + L\omega I_{sq}}{sL + r_\mathrm{L}}$$
$$I_{sq}(s) = \frac{y_{sq}V_\mathrm{in} - E_\mathrm{bq} - L\omega I_{sq}}{sL + r_\mathrm{L}},$$

where we acknowledge that $V_\mathrm{inv,d} = y_{sd}V_\mathrm{in}$ and $V_\mathrm{inv,q} = y_{sq}V_\mathrm{in}$; y_d, y_q being the output of the d and q axis controller respectively. This convention of symbols is consistent with the Fig. 7.

The above two equations would be useful to understand the feed-forward structure. Under proper feed-forward, the cross-coupling terms and the back emf terms will vanish leaving behind a simple plant of the form $1/(sL + r_\mathrm{L})$. For the feed-forward of the back EMF we have to estimate speed using the outputs of hall effect sensors. For all practical applications, considering fixed and well-regulated DC bus voltage V_{in}.

3.2 Controller Design

We will try to motivate in this section how to select a controller. We propose a bold hypothesis, for many practical plants, we should design a controller which will ensure that the open loop plant transfer function has the following form $\ell(s) = \frac{\Omega_c}{s}$. It should be noted that every control system design in terms of tracking, noise, and disturbance rejection is unique and no such simplifying control assumption will work. But we will see that with this approach we will still comfortably control a large number of plants.

The reason it works well is:

1. We will almost always control a DC quantity. Even when we would want to control an AC quantity, we can do a frame transformation to obtain an equivalent dc quantity to control.
2. The first order nature of the open loop TF $\frac{\Omega_c}{s}$ ensures precise control over the bandwidth (Ω_c) and phase margin (always 90°)

For example, after employing all feedforward and scaling, we observe that for the current control, the equivalent plant transfer function is $\frac{1}{(sL+r_\mathrm{L})}$. The way we would define a controller is as follows.
We understand that since the plant has a first order denominator, we need a first order numerator to exactly do a pole zero cancellation.

$$\begin{aligned}
\ell(s) = \frac{\Omega_c}{s} \equiv & (k_p + \frac{k_i}{s})\frac{1}{(sL + r_\mathrm{L})} \\
= & \frac{k_i}{sr_\mathrm{L}}\frac{sk_p/k_i + 1}{sL/r_\mathrm{L} + 1} \\
\therefore & \, k_i = \Omega_c \, r_\mathrm{L}, \quad k_p = \Omega_c \, L
\end{aligned}$$

The last two equations give us the design steps to choose the controller parameters. A similar design procedure can be adopted for the outer speed control. Depending on the frame transformation, one of the axis current will result in effective torque whereas the other axis current will result in no change in torque. The output of the speed controller will be the reference of the current that controls torque.

Tasks:

For this lab, you do not need to generate any results from "PLECS only" simulation. All results have to be pulled from either the CCS watch window and graph or the real-time scope waveforms in PLECS. However, it will be helpful to use a PLECS only simulation to validate your controller before implementing it in code; you may also use this simulation for partial credit if you code does not work). The following are the tasks that should be covered in the lab:

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**].

2. For the closed loop control, you will implement two PI controllers, one for the d axis current and the other for the q axis current. Follow the steps as described,

   - Let us say the output of the PI controllers from the d and q axis controller be v_{sd} and v_{sq} respectively.
   - You need to do a frame transformation by the following lines of codes,

```
    #define al 2.094395102393195f // This is 2pi/3
      // DQ to ABC
             vsa = (vsd*cos(angle)-vsq*sin(angle))/Vin;
             vsb = (vsd*cos(angle-al)-vsq*sin(angle-al))/Vin;
             vsc = (vsd*cos(angle+al)-vsq*sin(angle+al))/Vin;
```

   - You could use your saturator at a lot of places. You could saturate v_{sd}, v_{sq} or you could saturate v_{sa}, v_{sb}, v_{sc}. But it makes little sense to saturate both. Just be sure your integrator terms in the PI controllers are limited appropriately.
   - Finally, generate the PWM modulation signals as follows.

```
    EPwm2Regs.CMPA.bit.CMPA = (0.5+ 0.5*vsa)*N;
    EPwm3Regs.CMPA.bit.CMPA = (0.5+ 0.5*vsb)*N;
    EPwm4Regs.CMPA.bit.CMPA = (0.5+ 0.5*vsc)*N;
```

   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**]

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**].

   The feedforwards have to follow as below where v_{sd} and v_{sq} are outputs of the d and q axis controller respectively.

$$y_{sd} = v_{sd} - L\omega I_{sq} + E_{bd}$$
$$y_{sq} = v_{sq} + L\omega I_{sd} + E_{bq},$$

   Depending on alignment, we can have (E_{bd}, E_{bq}) as $(-\lambda_m\omega, 0)$ or $(0, \lambda_m\omega)$. Where ω is the electrical rotational frequency.

4. Determine if the additional phase delay from your digital control loop is significant. Adjust your controller to compensate for this. Note that if you select a significantly low bandwidth, effects of digital delay will be negligible. But if you have a higher bandwidth to meet the design specs, the effects of digital delay will be clear. You can correlate the delay to a loss in phase margin and hence, higher overshoots. The key is to design your controllers effectively. [**Optional: 0 pts**]

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**]

6. A helpful suggestion, start without feedforward but the same controller. Start without integrator saturation or limits, see if it works. Also, in the beginning, rather than controlling (for doing frame transformations, feedforward) using the estimated rotor angle or rotor speed, take the rotor angle and speed in through the ADC and use those quantities directly. Lastly, if you feel for the closed loop control, your speed estimation or angle estimation is giving poor results, one of the reasons could be your motor is spinning too fast to accurately estimate the angle. In such case you should increase the load torque by doubling the coefficient, "M.Drag".

Questions:

1. At your designed bandwidth, Ω_c, what is the additional phase loss due to digital delay? If you had no delay, what was the phase margin that you would have achieved? Compute the delay and come up with at least two methods to compensate the loss of phase margin due to the delay. [**Optional: 0 pts**]

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**]

4 Speed Control

For the speed controller, the dynamics of the BLDC motor mounted on a bike with a rider is given as follows:

$$J\frac{d\omega_m}{dt} + B\omega_m = T_e - T_m, \tag{7}$$

where J represent the net moment of inertia of the bike and the rider, B represents the cumulative friction coefficient of the shaft and road, T_e represents the input electrical torque as in (6).

Taking Laplace transform of (7), we obtain the following transfer function,

$$\omega_m(s) = \underbrace{\frac{1}{sJ+B} T_e(s)}_{\text{Plant}} \quad + \quad \underbrace{\frac{1}{sJ+B} T_m(s)}_{\text{Disturbance}} \tag{8}$$

Since it is hard to measure the mechanical torque, T_m, we do not implement any feed-forward to cancel it out, instead we hope that the speed controller having large dc gain will cancel any effect of the mechanical (load) torque on the speed ω_m. The plant transfer function in (8) is a first order transfer function. We implement a PI controller to exactly cancel the first order dynamics and ensure the resulting loop gain (equivalently open loop system) has the following form :

$$\begin{aligned}
\ell_\omega(s) = \frac{\Omega_\omega}{s} &\equiv (k_{p,\omega} + \frac{k_{i,\omega}}{s}) \frac{1}{(sJ+B)} \\
&= \frac{k_{i,\omega}}{sB} \frac{sk_{p,\omega}/k_{i,\omega} + 1}{sJ/B + 1} \\
&\therefore k_{i,\omega} = \Omega_\omega \, B, \quad k_{p,\omega} = \Omega_\omega \, J
\end{aligned}$$

The last two equations give us the design steps to choose the controller parameters for the speed controller. The speed controller generates the electrical torque reference as follows:

$$T_e^\star = \left(k_{p,\omega} + \frac{k_{i,\omega}}{s} \right) (\omega_e^\star - \omega_e),$$

where $\omega_e^\star$ is the mechanical speed that the bike-rider desires. To get an intuition as to why the speed controller would generate a torque reference, try to think in terms of a boost converter how an output voltage controller generate a current reference because the current would charge the capacitor voltage, so controlling the current would in turn regulate the capacitor voltage. The speed control loop is effectively a voltage control loop and the torque loop is an effective current control loop- a similarity that we have already introduced in (5)–(6).

To inter-connect the speed control loop and the current control loop, you need to obtain the q-axis current reference, $I_{\mathrm{q}}^\star$ from the output of the current controller, $T_e^\star$ by inverting (6) as follows:

$$I_{\mathrm{q}}^\star = \frac{4}{3\lambda_{\mathrm{m}} p} T_e^\star$$

Tasks:

For this section, you do not need to generate any results from "PLECS only" simulation. All results have to be pulled from either the CCS watch window and graph or the real-time scope waveforms in PLECS. However, it will be helpful to use a PLECS only simulation to validate your controller before implementing it in code; you may also use this simulation for partial credit if you code does not work). The following are the tasks that should be covered in the lab:

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**].

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**]

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 $50\mu sec$). [**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**]

5 System Integration

Integrate the BLDC motor with the boost converter. You need to appropriately define the new PWM and ADC channels. Also keep on mind that with such heavy coding overflow, you should sample only once per switching cycle, rather than doing a peak-valley sampling. Consequently your sampling frequency changes to 10 kHz, the same as your switching frequency. Remove the load resistor from the boost converter before integrating. Good luck! [**No credits as of now**]

Appendix A Signal Mapping

Feature	RT Box Signal	LaunchPad Pin	TI280049C Signal
$i_{s,a}$	AO14	J7 69	ADCINA3
$i_{s,b}$	AO0	J3 23	ADCINA5
$i_{s,c}$	AO8	J7 63	ADCINA6
Drive PWM A U	DI20	J8 76	ePWM2A
Drive PWM A L	DI21	J8 75	ePWM2B
Drive PWM B U	DI4	J4 36	ePWM3A
Drive PWM B L	DI5	J4 35	ePWM3B
Drive PWM C U	DI18	J8 78	ePWM4A
Drive PWM C L	DI19	J8 77	ePWM4B
Hall Sensor A	DO22	J1 04	GPIO33
Hall Sensor B	DO23	J1 04	GPIO34
Hall Sensor C	DO27	J1 04	GPIO35

References

[1] https://www.ebikes.ca/learn/hub-motors.html

[2] P. Pillay, R. Krishnan, "Modeling, simulation, and analysis of permanent-magnet motor drives, Part II: The brushless DC motor drive", IEEE Trans. on Ind. App., Vol. 25, No. 2, Mar./Apr. 1989.

[3] A. Muetze and Y. C. Tan, "Electric bicycles - A performance evaluation," in IEEE Industry Applications Magazine, vol. 13, no. 4, pp. 12-21, July-Aug. 2007.