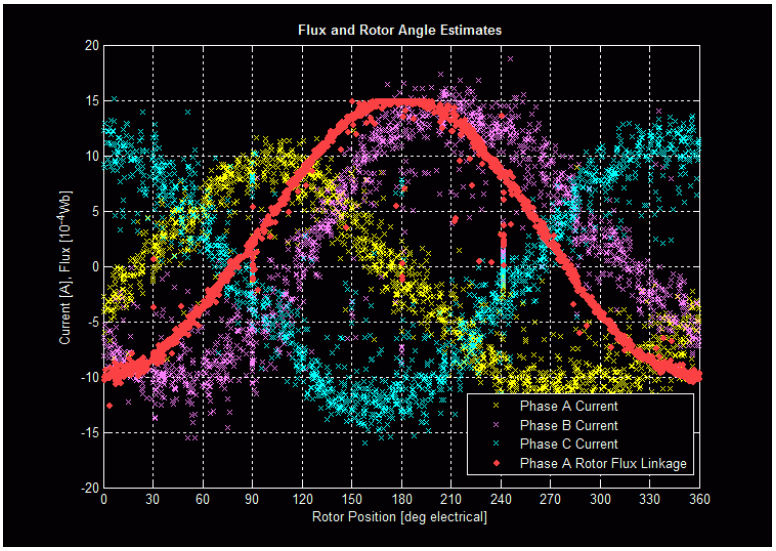
Flux Observer-Based Sensorless Field-Oriented Control of Surface Permanent Magnet Synchronous Motors (Gen. 1)

Shane Colton
<colton.shane@gmail.com>
[scolton.blogspot.com]

Document Rev. 1 | 03/02/2014

Hardware Referenced:
3ph v3.1
FF v1.1
FF v1.2s



Contents

1 Background

Permanent magnet synchronous motors (PMSM) and motor controllers are among the most efficient transformers of electrical and mechanical power in existence. They are routinely operated at 90% system efficiency or higher, allowing for extremely high specific torque and specific power. However, they require more complex electronics and control than other types of motors (e.g. brushed DC motors, asynchronous induction motors). The key word is "synchronous": without the mechanical timing of brushes and commutators, it is the task of the electronics to generate a rotating current and flux to interact with the rotor as it spins.

To accurately generate this synchronous, rotating field, the motor controller must know the position of the rotor. Specifically, it must know the phase offset between the rotor permanent magnets and the stator phase windings. For "brushless DC" motors, a coarse estimate is good enough. More advanced field-oriented control methods require high-resolution position knowledge. This knowledge can be gained by directly measuring the rotor position with a rotary position sensor. However, these sensors add cost, size, and complexity to the system and may be the reliability bottleneck in some cases.

The goal of sensorless field-oriented control is to acquire the necessary high-resolution rotor position estimate using only values that are readily sensed at the motor controller itself, namely voltage and current. In this way, no external sensors are required. In order to indirectly estimate the rotor position, the concept of an observer is put to use. Specifically, this article details the use of a rotor flux observer. Since rotor flux is position-dependent, it can be used to arrive at a high-resolution rotor position estimate for use in field-oriented control.

The observer depends heavily on a model of the PMSM. Thus, a good starting place for discussion is with the physical model of such a motor:

1.1 Three-Phase Permanent Magnet Motors

1.1.1 Motor Model, Back EMF, Motor Constant

Permanent magnet motors are well-modeled as a speed-dependent voltage source in series with an inductor and a resistor. The speed-dependent voltage source is called the **back EMF**, and is a physical consequence of wire loops moving through a varying magnetic field. In the case of a brushed DC motor, the back EMF is a constant DC voltage proportional to speed. The constant of proportionality is called the **motor constant**, often designated K_t. The electromechanical model for a brushed DC motor is shown in Figure 1.

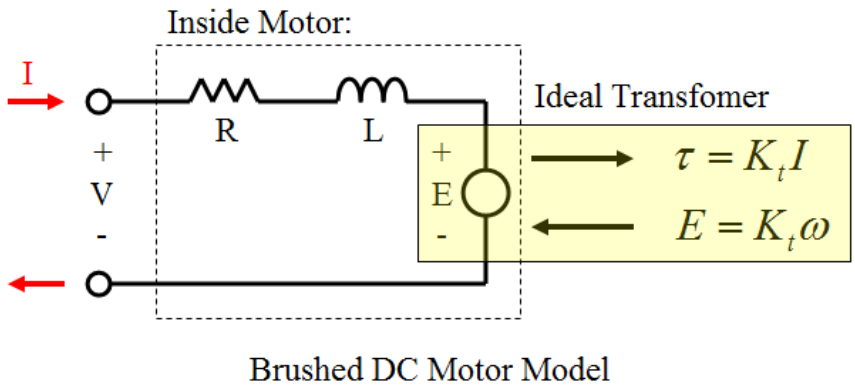


Figure 1: The electromechanical model for a brushed DC motor contains an ideal transformer in series with a resistor and an inductor.

The motor constant, K_t, sets both the torque per unit current and the back EMF voltage per unit angular velocity. Both of these ratios have the same SI base units and the equivalency of these two constants is a consequence of power conservation in the ideal transformer part of the model. (All losses are accounted for by external elements such as electrical resistance or bearing friction.)

This model can be extended for three-phase brushless motors. In a brushed motor, the position of current-carrying windings with respect to the magnets is fixed by the brushes and commutator. In a brushless motor, this phasing is variable and handled by software. One way to capture this physically is to allow the motor constant, K_t, to become a periodic function of *electrical* angle, θ_e, the relative phase between the rotating magnet poles and the fixed stator windings. The model can then be applied independently to each of the three phases, as shown in Figure 2. This model assumes balanced phases with the same number of turns per phase, the same phase resistance, and the same inductance. The angular velocity is left defined as mechanical speed, ω_m, to simplify the analysis.

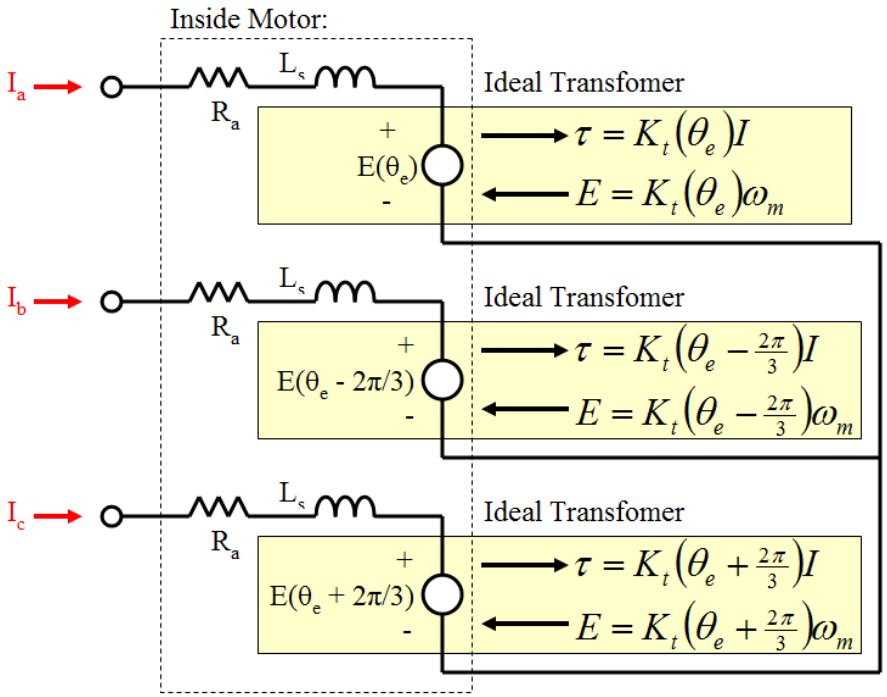


Figure 2: The DC motor model extended to a three-phase brushless motor.

This extension of the simple model to three-phase brushless might seem a bit forced, but in fact it is very powerful in this most general form. So far, nothing at all has been assumed about the shape of $K_t(\theta_e)$, other than that it is periodic in electrical angle. Nothing has been assumed about the driving currents either, other than that they sum to zero. Applying this general model to different motor/drive combinations demystifies the many complexities of brushless motors with sinusoidal, trapezoidal, and arbitrarily-shaped waveforms, driven by either sinusoidal "AC"- or square wave "BLDC"-type controllers. A thorough analysis of these possible combinations is presented in [3].

One way to see the exact shape of $K_t(\theta_e)$ is to spin the motor with no load and measure the periodic back EMF waveform. (Analogous to spinning a brushed DC motor and measuring the DC voltage it produces to determine K_t.) The back EMF per unit mechanical speed *is* $K_t(\theta_e)$, and this is also the function that defines torque per unit current *as a function of electrical angle*. The goal of brushless motor control is to drive each phase with the appropriate current to get maximum net positive torque at a given electrical angle. Thus, some form of angular position measurement or estimation (to determine θ_e) is necessary for proper commutation of a brushless motor.

For the rest of this article, both a sinusoidal back EMF and a sinusoidal drive current are assumed:

$$E_a = A\omega_m \sin(\theta_e)$$

$$E_b = A\omega_m \sin\left(\theta_e - \frac{2\pi}{3}\right)$$

$$E_c = A\omega_m \sin\left(\theta_e + \frac{2\pi}{3}\right)$$

$$I_a = B\sin(\theta_e - \varphi)$$

$$I_b = B\sin\left(\theta_e - \frac{2\pi}{3} - \varphi\right)$$

$$I_c = B\sin\left(\theta_e + \frac{2\pi}{3} - \varphi\right)$$

The method of field-oriented control applies specifically to this case, with only balanced three-phase sinusoidal voltages and currents. Without loss of generality, the electrical angle is defined such that Phase A's back EMF, E_a, is proportional to $\sin(\theta_e)$. This zero angle definition should be considered flexible for the purpose of analysis; the relative offsets of ±120º are more fundamental. A phase shift, φ, is added to the phase of the three currents to emphasize that they do not necessarily align with the back EMF. However, it will be shown in the next section that for this type of motor, maximum torque per amp is achieved when $\varphi = 0$.

1.1.2 Torque Production

Torque production is derived from the ideal motor model, presented in 1.1.1. Torque is produced as a direct consequence of power converted through the back EMF. The power converted by a phase at any instant is the product of the drive current and back EMF at that instant. The average power converted by each phase is the average of that product over one electrical cycle, and generally depends on the shapes of both the back EMF and the drive current. Finally, there are three symmetric phases, so the average power converted by the whole motor is three times the average power converted by each phase. Torque is power divided by motor mechanical angular velocity. This is true instantaneously and on average (assuming the motor speed to be constant over one electrical period). The following equations summarize instantaneous and average power conversion and torque production based on these fundamentals:

$$P(t) = \sum_{a,b,c} I(t)\cdot E(t) = \tau(t)\cdot\omega_m$$

$$P_{avg} = 3\frac{1}{T}\int_0^T I(t)\cdot E(t)\cdot dt = \tau_{avg}\cdot\omega_m$$

In the case of pure sinusoidal back EMF with pure sinusoidal drive current, the product becomes a dot product of two vectors, offset from being parallel by the angle φ. In this special case, the power is not a function of θ_e, since both the back EMF and the current vector will simply rotate by the same amount without changing amplitudes. Based on the constants of proportionality defined in Section 1.1.1, the magnitude of power converted (both instantaneously and on average) is:

$$P(t) = P_{avg} = \frac{3}{2}AB\omega_m\cos\varphi$$

and is maximized when $\varphi = 0$. The factor of 1/2 results from the average value of $\sin^2(\theta_e)$ over one full electrical cycle. The torque produced is the power divided by the mechanical angular velocity:

$$\tau = \frac{P}{\omega_m} = \frac{3}{2}AB\cos\varphi$$

Thus for a given peak amplitude of current (constant B) and a given flux linkage (constant A), maximum torque is achieved when the back EMF and current vectors are aligned, $\varphi = 0$. Note that this is only strictly true for ideal *surface* permanent magnet motors. Different maximum torque per amp relationships exist for other types of brushless motors. The concept of treating sinusoidal back EMF, current, and voltage as vectors is fundamental to field-oriented control and will be detailed in Section 1.2.2.

1.1.3 Resistance, Inductance, Saliency, and Field Weakening

Since they consist of coils of copper wire, motor windings have an electrical resistance. Though the resistance is distributed along the length of the coil, it can be modeled as a simple series resistor on each phase as in Figure 2. The winding resistance per phase, R_a, is easy to measure and to predict based on the resistivity of copper. Since power is dissipated in this resistor, it contributes strongly to motor inefficiency. In fact, it is the only source of loss that is captured by the simple motor model. At many operating points, resistive loss is the dominant loss in a motor and the simple model is sufficient for predicting motor efficiency. One important exception is at or near no-load speed, where currents are small and speed-dependent losses (e.g. friction, eddy current) become dominant. These losses are considered to be external to the simple motor model of Figure 2.

The total power dissipated by the motor resistance depends on the shape of the drive currents. Most generally, it is calculated by:

$$P_r = \frac{3}{2\pi}\int_0^{2\pi}\left[I(\theta_e)\right]^2 R_a d\theta_e = 3I_{rms}^2 R_a .$$

The root mean square current captures the effect of waveform shape. Given a purely sinusoidal back EMF waveform, a purely sinusoidal current will produce the most torque per unit of heat dissipation in the motor resistance.

Motor windings also have inductance. Physically, this means that current flowing in the windings will induce magnetic flux through them, even in the absence of permanent magnet flux from the rotor. It also means that the windings will resist rapid changes in current by generating voltage across this inductor. However, this is not the back EMF. Back EMF is *only* the component of voltage that is generated by the permanent magnet flux. Thus, there is a separate series inductor included in the motor model.

The value of inductance is less straightforward to calculate because the phases are not magnetically independent. That is, current in one phase can induce flux in another. Under sinusoidal drive currents, it is possible to use a lumped inductance, called the synchronous inductance, to accommodate this. The value of the synchronous inductance is:

$$L_s = \frac{3}{2}L_a ,$$

where L_a is the inductance that would be measured independently on one phase, if it could be isolated. This is derived in [1] for certain types of motors.

The winding inductance has many theoretical and practical effects on the motor. It stores energy in the form of a magnetic field any time there is current in the winding. When a winding is switched off, this energy must go somewhere. For this reason, controllers contain flyback diodes that allow this current to

circulate even when all the transistor switches are open. Under high frequency pulse-width modulated (PWM) control, the winding inductance filters out current ripple. However, as a low-pass filter on current it also creates phase lag. This lag is explored in detail in Section 1.2.3 as motivation for the use of field-oriented control.

The winding inductance is a function of motor geometry and the number of turns in the winding. In non-salient (round rotor) motors, the inductance is not a function of electrical angle. This is the case for motors with complete radial symmetry of the rotor's steel backing at any angle. (The magnets themselves don't matter, since they have nearly the same permeability as air.) Motors with magnets mounted to the surface of the rotor steel, called surface permanent magnet (SPM), fall into this category. By contrast, salient motors have an inductance that varies periodically with electrical angle. This is the case if the rotor's steel backing is differently-shaped at the poles than in between them. Motors with magnets embedded in the steel backing, called interior permanent magnet (IPM), fall into this category. This article will focus exclusively on non-salient, SPM motors. Torque production for salient motors requires a more complicated analysis.

Motor inductance also has a large effect on field weakening, a technique usually used to extend the operating speed range of a motor. In field weakening, some current is used to induce a field which partially cancels the permanent magnet field. This results in less torque per unit current, but also decreases the back EMF per unit speed, allowing the motor to be operated to higher speeds with a given voltage. In general, motors with lower inductance have less field-weakening capability.

1.2 Field-Oriented Control

Field-oriented control (FOC) is an advanced control technique used primarily for AC induction motors and permanent magnet synchronous motors. It has the advantage of isolating the torque-producing component of motor current from the field-augmenting or field-weakening component. This allows for a simple and independent torque controller and field controller, as would be the case with a separately-excited DC motor. Field-oriented control is not synonymous with space vector modulation (SVM), sinusoidal commutation, or phase advance, though all or some of these other techniques may be used to achieve field-oriented control [1].

This article will focus on field-oriented control as it applies to permanent magnet synchronous motors. In PMSM, it is very easy to isolate the torque-producing component of motor current by working in the rotating reference frame of the rotor, which is designated the d-q reference frame. Motor quantities can be mapped into the d-q frame by simple trigonometry, and current (torque, field) control can be executed in this frame. This control structure is called a **synchronous current regulator**, and is discussed in more detail in Section 1.2.5. A modified synchronous current regulator with advantages in computational efficiency and transient response is also outlined in Section 1.2.5.

1.2.1 d-q Reference Frame

Essential to field-oriented control in PMSM is the establishment of a frame of reference that is fixed with respect to the spinning rotor. Even simple BLDC controllers accomplish this, to some extent, by using Hall effect sensors or back EMF sensing to estimate rotor position. Field-oriented control goes a step further by using a finer rotor position estimate to map sinusoidal motor currents into the rotating frame.

The rotating frame is defined by two axes, labeled direct (d) and quadrature (q) and fixed to the rotor as illustrated in Figure 3. (A two-pole outside rotor configuration is used to facilitate the illustration.)

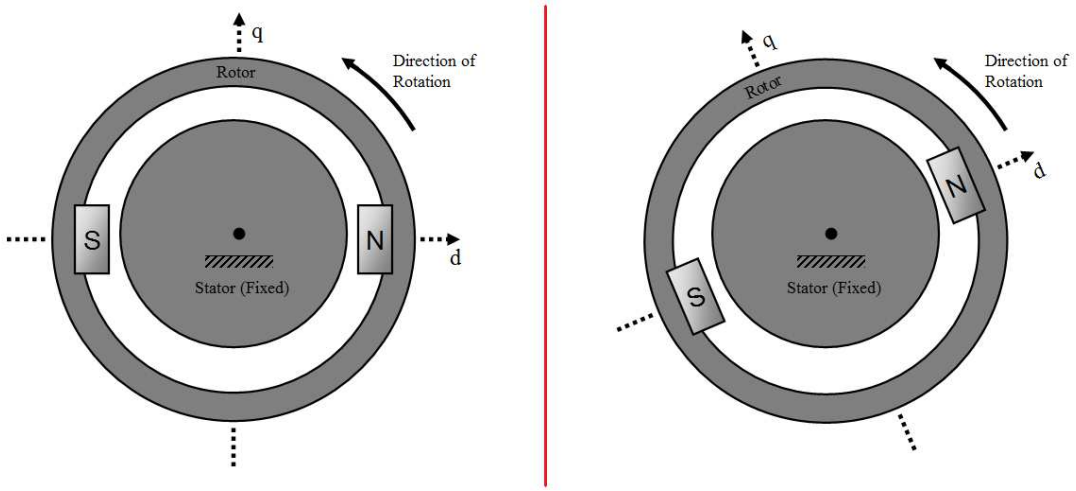


Figure 3: The d-q reference frame is fixed to the rotor such that the d-axis always falls on the magnetic axis and the q-axis always falls in between the magnetic axis, 90º ahead in the direction of rotation.

The direct (d) axis is defined to be on the magnetic axis passing through the center of a set of permanent magnets on the rotor. The quadrature (q) axis is defined to be 90º *electrical* ahead of the q-axis in the direction of rotation. In other words, the q-axis always falls exactly between two magnets. In the case of a two-pole motor, this is identical to 90º *mechanical*, and the axes are physically orthogonal. For higher pole counts, the axes are not physically orthogonal. For example, in a four-pole motor, they are separated by 45º *mechanical*. For simplicity, the two-pole, outside rotor illustration in Figure 3 will be used throughout this section. Since motor quantities will be projected based on electrical angle measurements, the number of poles does not affect the control strategy.

1.2.2 Vector Motor Quantities

With the d- and q-axis defined, any motor quantity that has a direction associated with it can be mapped to a vector in the d-q frame by projection. Many motor quantities, including current, are associated with the motor phase windings, which reside on the stator. These quantities take as their direction the principal axis of that phase winding. The three phases (a, b, c) are always established at intervals of 120º *electrical* to each other. These phase axes are added to the illustration in Figure 4. While the d- and q-axis rotate, the a-, b-, and c-axis stay fixed to the stator.

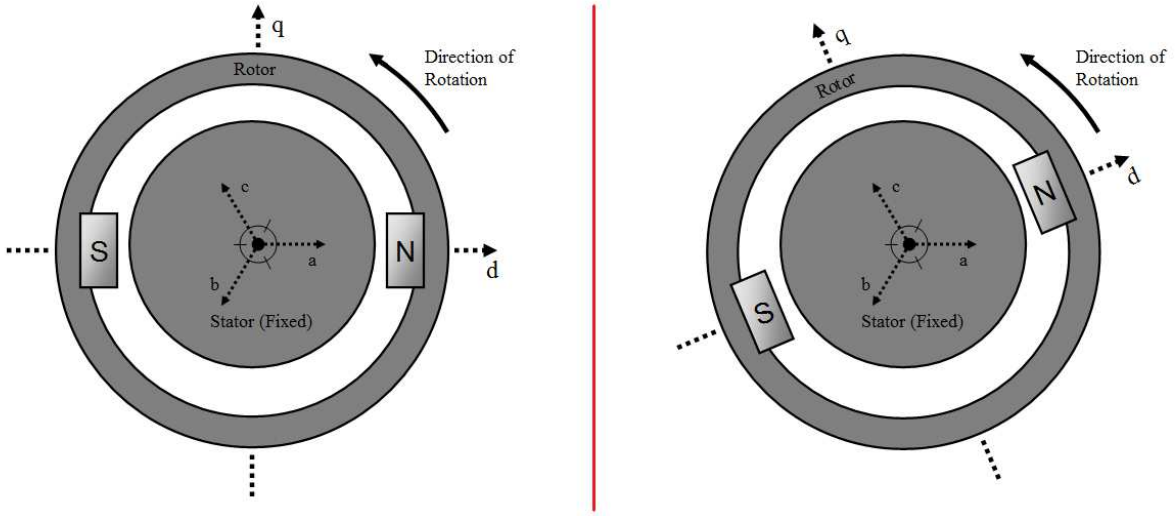


Figure 4: The three phase winding of the motor defines three equally-spaced axes that are fixed to the stator, labeled a, b, and c.

For example, consider the following balanced set of phase currents:

$$I_a = 10A$$
$$I_b = -5A$$
$$I_c = -5A$$

These are illustrated as vectors on the stator in Figure 5. Note that you can also sum the three vectors to get a single resultant current vector. (All the methods of vector geometry apply here.)

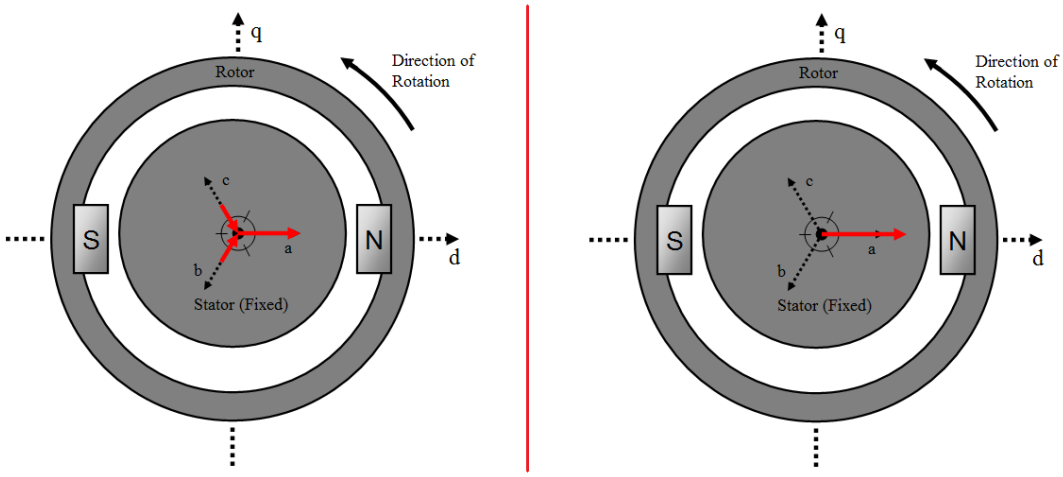


Figure 5: The balanced phase currents can be represented as vectors on the stator (left). They can also be summed into one resultant current vector (right).

The magnitude of the resultant vector is calculated as follows:

$$|I| = |I_a| + |I_b|\cos(60°) + |I_c|\cos(60°) = 15A$$

$$|I| = \frac{3}{2}|I_a|$$

This factor of 3/2, which appears frequently with balanced three-phase quantities, will be important to the analysis. However, different transformations from (a,b,c) to (d,q) may or may not account for this factor. Thus, magnitude is deemphasized for now and the focus will be on the *direction* of the resultant.

In Figure 5, it is easy to see how the current vector would be mapped onto the d- and q-axis. (I_d is positive, I_q is zero.) By considering the current vector as the principal axis of a coil of wire on the stator, the resulting interaction between the rotor and the stator is intuitively clear. The stator becomes like an electromagnet, with its poles along the axis of the resultant current vector. Since the stator electromagnet and the rotor permanent magnet axes are already aligned in Figure 5, there will be no torque produced. (Given the assumption that the d-axis points from south to north, it is the stable point. Otherwise, it would be the anti-stable point. This directional convention is not crucial to the analysis.)

A detailed look at where to place current in the d-q frame for maximum torque production is now presented. First, two other motor quantities are mapped in the d-q frame. These are the flux generated by the permanent magnets on the rotor (λ), and the back EMF that flux creates in the motor coils (E). These two vectors are plotted in Figure 6.

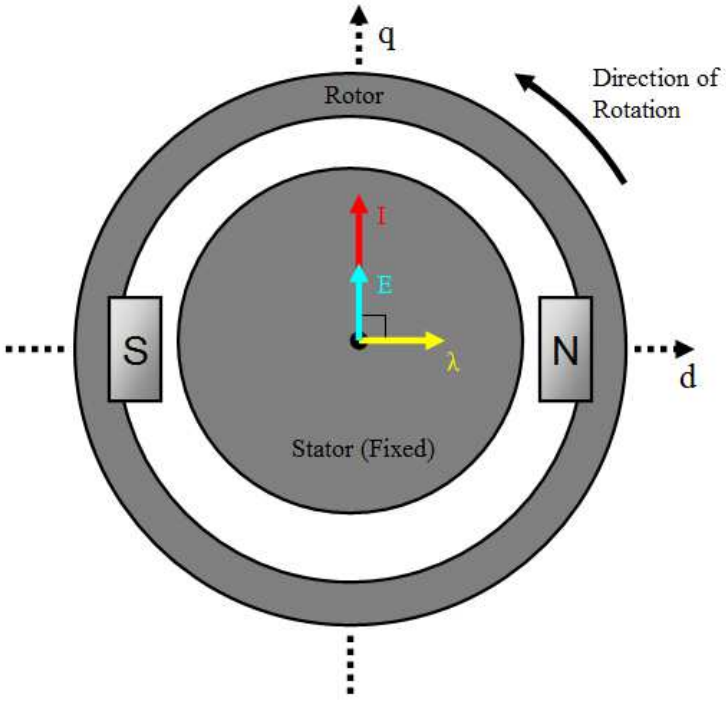


Figure 6: Flux caused by the permanent magnets will always align with the d-axis. Back EMF will always lead this flux by 90º electrical.

The link between permanent magnet flux and back EMF is based on the fundamental formula for back voltage created on a coil of wire in a varying magnetic field:

$$E = \frac{d\lambda}{dt}.$$

In the case of a sinusoidal time-varying flux, λ, the back EMF, E, will also be sinusoidal and will lead the flux by 90º. This is the condition illustrated in Figure 6, with the flux and back EMF vectors representing the instantaneous location of the peak flux and back EMF. These peaks will rotate with the d-q frame such that λ is always on the d-axis and E is always on the q-axis. Since the flux considered is from permanent magnets only, this is true regardless of stator current.

Given a rotor angular velocity, the magnitude of E is fixed by the motor constant, as in Section 1.1.1. To convert as much power as possible with a given current, the dot product of the I and E vectors should be maximized. This occurs when current is on the q-axis exclusively. Maximizing power is the same as maximizing torque, since the speed is given. This analysis works as well in the limit as speed goes to zero. Thus, *peak torque will always occur when current is on the q-axis*.

1.2.3 Why Control is Necessary: Motor Inductance

Most often, motor controllers are created with elements that can be modeled as voltage sources. For example, a set of two switching power transistors creates a time-averaged voltage applied to each motor phase using pulse-width modulation (PWM). This is an open-loop phenomenon: the voltage is exactly set by controlling the duty cycle of the two switching power devices.

Torque production, however, is governed by current, not voltage. If a motor winding were well-modeled as a simple resistor, there would be no challenge to aligning current on the q-axis. Wherever the rotor is, the phase voltages could simply be set to produce a voltage vector on the q-axis. With no inductance, that would also be the direction of the current vector. This unrealistic scenario is shown in vector form in Figure 7. The broken-line vector represents the voltage across the winding resistance, which is the difference between V and E. Note that even in this ideal scenario, knowledge of the rotor position is required in order to correctly place the voltage vector.

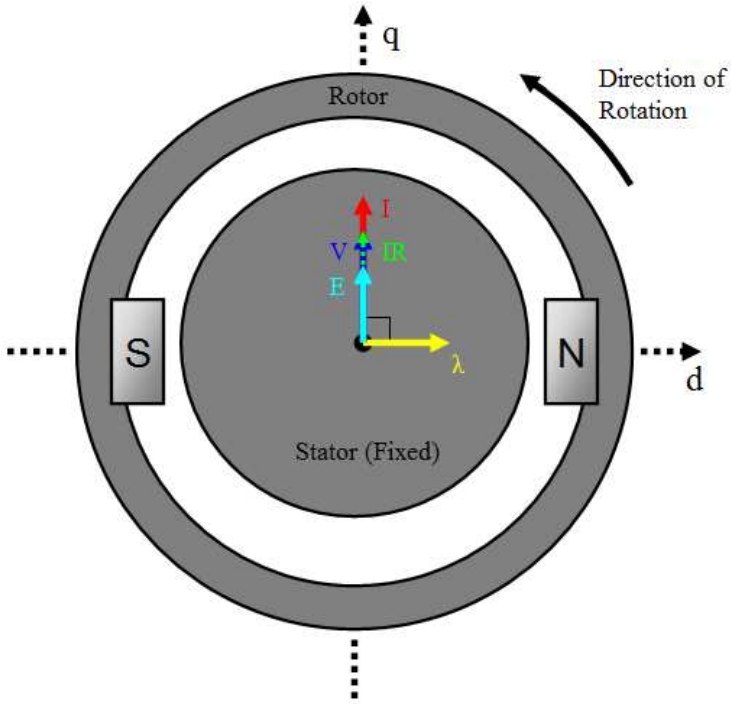


Figure 7: This is what the essential motor quantities would look like in the absence of winding inductance. Voltage, current, and back EMF could all be easily aligned in an open-loop manner for maximum torque at any given rotor position.

A real motor, however, has some inductance. Inductors resist changes in current according to the constitutive equation:

$$V_L = L\frac{dI}{dt}.$$

Thus some voltage will be developed across the winding inductance that resists changes in current. If an inductor is subjected to a sinusoidal time-varying voltage, the current will also be sinusoidal and will lag the voltage by 90º. This lag makes alignment of the current and back EMF vectors more difficult.

Since they are sinusoidal time-varying quantities, all of the motor quantities can be represented as complex variables to capture the mathematical relationships between voltage, current, resistance, inductance, flux, and back EMF. However, the important effects are more intuitively captured in the relationships between vectors in the d-q frame. For example, if the voltage were fixed to be on the q-axis, Figure 8 shows what effect inductance might have on the current vector.

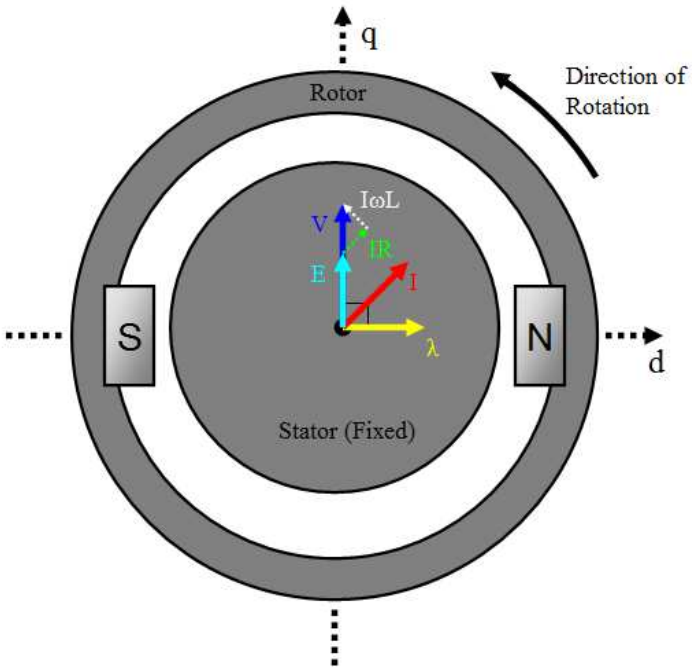


Figure 8: The vector relationships between motor quantities if the applied voltage is fixed to the q-axis, but there is inductance present.

Current now lags applied voltage, due to the inductance. The two broken-line vectors represent components of voltage across the winding resistance (*IR*) and winding inductance (*IωL*). The resistance component is parallel to current and the inductance component leads current by 90º. (In other words, current lags voltage across the inductor by 90º.) The vector sum of voltages is consistent with Kirchoff's Voltage Law (KVL).

In Figure 8, current and back EMF are no longer aligned, so torque per unit current is not maximized. Open-loop phase advance is one possible way to accommodate for the effect of inductance: Simply placing voltage ahead of the q-axis by some angle (which could be a function of other motor parameters, measured or known) could offset most, if not all, of the angle lost to inductance. However, any solution based on open-loop phase advance would be motor-specific. Field-oriented control seeks a more flexible solution based on real-time current measurements and closed-loop control.

1.2.4 Field-Oriented Control Objective

The objective of field-oriented control is to achieve measurement and closed-loop control of the motor current *vector*, essentially placing it on the d-q plane. The controller presented is general enough to place the vector anywhere, including leading the q-axis to achieve field weakening. For the most part, though, this article will focus on placing the current vector on the q-axis, to maximize torque per unit current. Figure 9 shows what the vector motor quantities might look like with the current vector controlled to be on the q-axis.

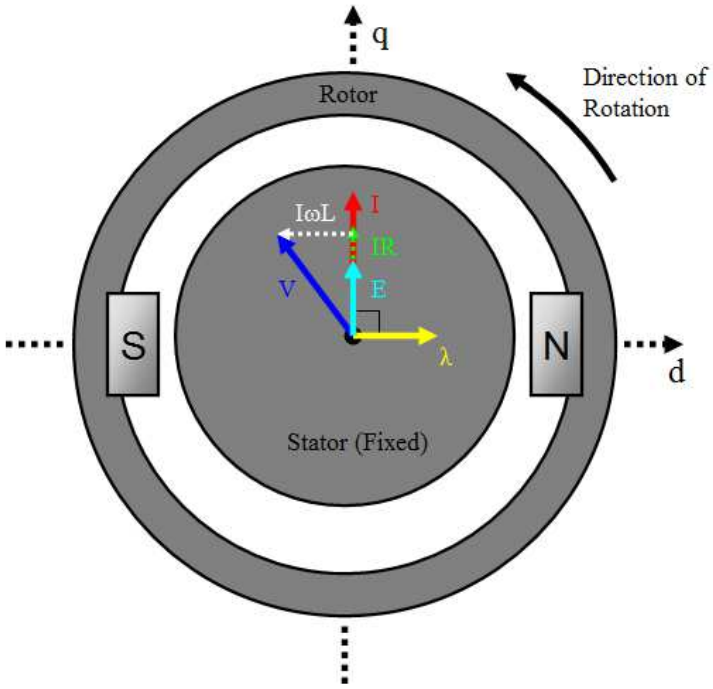


Figure 9: With current controlled to be on the q-axis, the voltage must lead the back EMF to account for inductance.

The voltage vector is advanced ahead of the back EMF. (In other words, V_d, the d-axis voltage, is negative.) This counteracts the lag introduced by motor inductance. Current still lags applied voltage, but it is now in phase with back EMF, producing optimal torque. The broken-line vectors represent the components of voltage across the winding resistance and inductance needed to satisfy KVL. Here it is clear that if the product of current, speed, and inductance ($I\omega L$) is sufficiently small as to be negligible compared to the other voltages, the degenerate case is that of Figure 7. This product (normalized to the applied voltage) can be used as a test for whether field-oriented control is justified for a given motor and operating range.

1.2.5 Synchronous Current Regulator

The synchronous current regulator is a closed-loop current controller operating in the d-q reference frame. It relies on the ability to transform control quantities readily between the stator (a,b,c) frame and rotor (d,q) frame. To do the projections required for this transformation, *the rotor position must be known.* All control is done on quantities in the d-q frame, even though the measurements and outputs are done in the stationary frame. Figure 10 shows the standard synchronous current regulator block diagram.

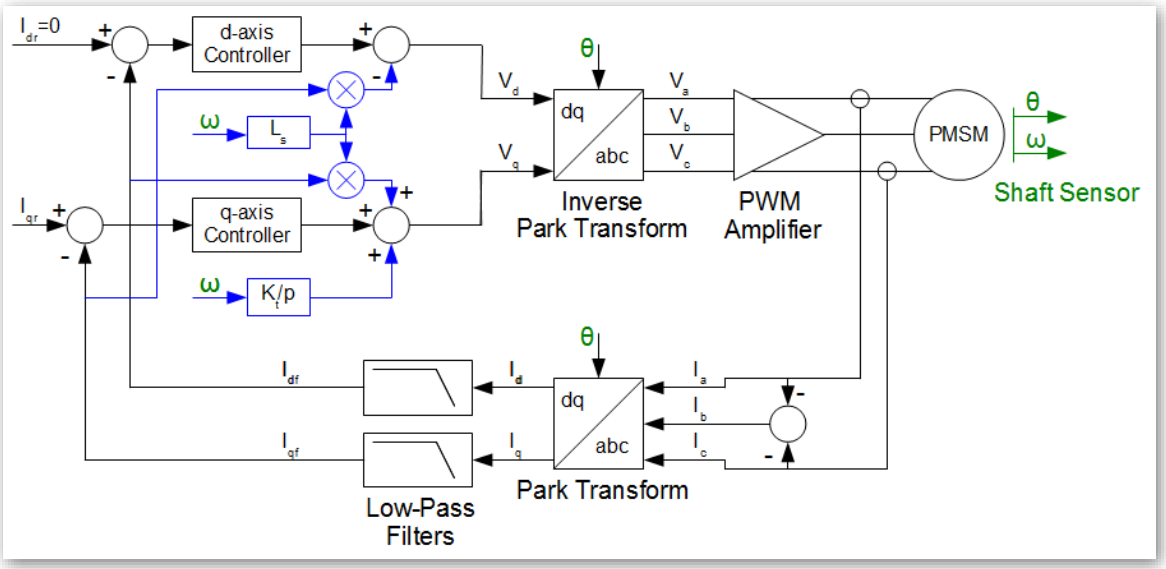


Figure 10: The standard synchronous current regulator block diagram. Using a measured rotor position, the Park and inverse Park Transforms convert between the stationary and rotating frame.

Current is measured on two of three motor phases. (The third must balance the first two.) The phase currents are projected onto the d-q frame as shown in Section 1.2.2 using the Park Transform [1]. I_d and I_q are compared to reference values representing the desired current vector. (In this case, the desired d-axis current is zero.) The errors are used in standard controllers (e.g. proportional+integral) to generate voltage outputs via pulse width modulation (PWM). Feed-forward terms proportional to inductance and motor speed (shown in blue) may also be present to fully decouple the d- and q-axes [2]. The controller outputs, V_d and V_q, are converted back to the stationary frame with the inverse Park Transform and sent to the power stage, where they set voltages at the motor phases.

One of the main advantages of field-oriented control is the DC nature of the current sensor signals, I_d and I_q. They can be low-pass filtered with a time constant longer than the electrical period of the motor, if necessary, to reduce current sensor noise. The low-pass filters on I_d and I_q do not impart any motor speed-dependent phase shift, since the signals are steady-state DC values. The desired torque/field current control bandwidth is the only constraint on these filters.

A modified synchronous current regulator is shown in Figure 11. This form replaces the q-axis and d-axis controllers with ones outputting voltage magnitude and phase, respectively. These outputs drive a simple look-up table-based sine wave generator that controls the three-phase PWMs. The modified synchronous current regulator has a number of practical advantages and is the author's preferred implementation. For a number of reasons, it is computationally efficient and well-suited for implementation on low-power embedded processors:

- No inverse Park transform is required.
- The regulator is readily segmented into a "fast loop" that runs at PWM frequency and a "slow loop" that need only run fast enough to meet filter and controller bandwidth requirements.

- The fast loop contains only position sensor read, look-up table, and scaling operations. No trigonometric, filtering, or control math is required[1]. All of this is pushed into the slow loop.

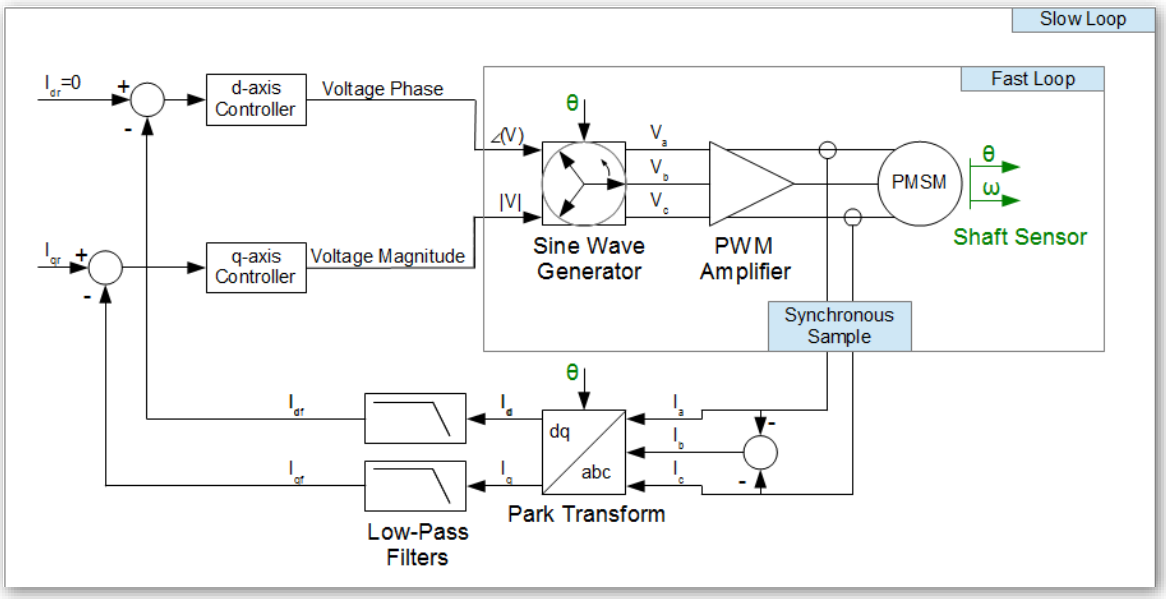


Figure 11: The modified synchronous current regulator.

The benefit of computational efficiency is diminished by ever-improving embedded processors. However, it is still useful to minimize the fast-loop processing requirement of the FOC algorithm in order to leave as much processing power as possible for more advanced sensorless position estimation algorithms. Additionally, the modified synchronous current regulator offers some advantage besides computational efficiency:

- Better transient response to torque/field commands than the standard synchronous current regulator without feedforward decoupling.
- No dependence on motor inductance, speed, or back EMF constant parameter estimates, since no feedforward decoupling is used.

In both synchronous current regulator implementations, knowledge of the rotor position is assumed. The Park transform uses rotor position to project phase currents onto the d-q reference frame. The inverse Park transform, or the sine wave generator in the modified regulator, uses the rotor position to create three-phase voltage drives from the field-oriented control outputs. When the rotor position cannot be directly measured using a sensor (encoder, resolve, Hall effect, etc.), a method of estimating rotor position using only known electrical quantities (voltage, current) must be used. This is the basis for sensorless field-oriented control, the topic of the rest of this article.

[1] This is true for *sensored* field-oriented control. Sensorless control may still require a good amount of fast math (trig, filtering) for position estimation, which must occur in the fast loop. The sensorless estimator presented in this article does not require fast-loop trig, but the filtering burden makes it necessary to have at a minimum fixed-point single-cycle multiply capability.

1.3 Sensorless Field-Oriented Control Hardware

The hardware requirement for sensorless field-oriented control is similar to that of any other three-phase motor control. The motor phases are driven by a three-phase inverter of six power transistors (MOSFETs or IGBTs), which are themselves driven by an intermediate gate drive stage. The transistors switch at high frequency in order to generate an average voltage on each motor phase by pulse width modulation (PWM). In square wave BLDC control, one phase is left open-circuit at any given time, which is also convenient for back-EMF sensing. However, with sinusoidal commutation (as in field-oriented control), all three phases are always being driven. A microcontroller capable of coordinating three PWMs is therefore an additional requirement. An example of a three-phase inverter and gate drive designed for sinusoidal field-oriented control is presented in this section.

While current sensing is often used even in BLDC control, it isn't necessary except for torque control and overcurrent protection. Field-oriented control, by contrast, requires at least two phase current sensors in order to measure and control the current vector. In sensorless algorithms, the current signals are also used to derive rotor position. For this reason, the quality and bandwidth of the current sensors is especially important in sensorless field-oriented control. Two different methods of phase current sensing, as well as ways to ensure the quality of the current sensor signals, are also detailed in this section.

Reference designs for three different controllers on which sensorless field-oriented control has been implemented are also available as schematics and board images in Appendix B and for download as EAGLE projects at [9].

1.3.1 Three Phase Inverter and Gate Drive

A typical three-phase inverter is shown in Figure 12. For low voltages (up to around 150V), MOSFET-based inverters are more common. Above 200V, IGBTs are more common. In either case, six power transistors are required to create a three-phase bridge. In MOSFET-based inverters, it's most common to use six identical N-channel FETs. In addition to the FETs, a large DC-link capacitor (C_{DC}) is placed across the DC bus to source and sink fast-transient current for the inverter. *Since the loop between the DC-link capacitor and the FETs contains most of the high dI/dt transients, it should be kept small and sensitive analog circuitry should be kept out of it.*

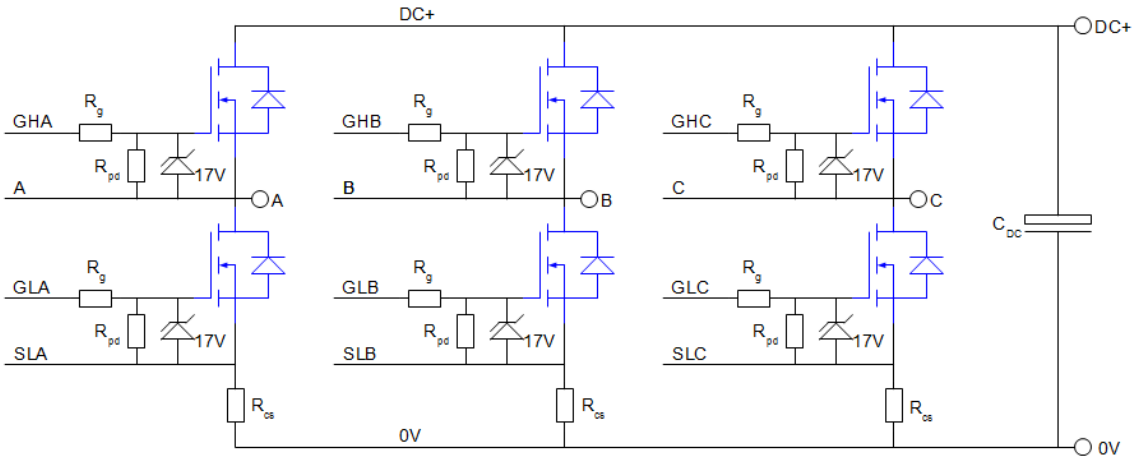


Figure 12: A typical three-phase bridge inverter.

The outputs themselves are labeled A, B, and C, and drive the three phase windings of the motor. The inverter enforces an average voltage on each phase by varying the fraction of time the high-side FET and low-side FETs are on. Note that they are never on at the same time and, unlike in a BLDC driver, they are never both off for extended periods of time. Thus, with the exception of short periods of deadtime (see below), the voltage on each output is either near DC+ or near 0V at any point in time. The average voltage on each phase is given by:

$$V_X = D_X V_{DC} = \frac{t_{HXon}}{T_{PWM}} V_{DC}$$

Where D_x is the duty cycle of Phase X, which is defined as the time the high side FET of phase X is on, t_{HXon}, divided by the total PWM period, T_{PWM}. The voltage is a ratio of DC bus voltage, V_{DC}, and so it also varies with changes in V_{DC} (as in a battery discharging, for example).

It's important to remember that this is an average voltage; the actual instantaneous voltage on each phase output swings from near 0V to near DC+ with very high dV/dt. *Because of this, the output areas should be kept small and away from noise-sensitive analog circuitry.* Because the outputs are connected to the high-side gate driver, this means keeping the gate drive circuitry close to the FETs and away from logic and analog circuits. Additionally, it means sensitive signals (such as encoders, current sense lines, and temperature sense lines) should be kept away or shielded from the outputs, including motor phase wires and gate drive signals.

In the implementation shown, each FET gate has a gate resistor (R_g), a pull-down resistor (R_{pd}), and a gate protection Zener diode. The purpose of the gate resistor is to limit the gate drive current, either to meet the maximum output specifications of the gate driver itself or to slow down the switching in order to reduce dI/dt and dV/dt-induced noise. The pull-down resistor ensures that the FET is turned off even in the event of gate driver failure. The Zener diode should be a Transient Voltage Suppression (TVS) type diode and protects the gate against high-voltage transients that could otherwise punch through its insulation. Both the pull-down resistor and the TVS diode should be placed close to the FET and directly connect gate to source.

Well-designed gate drive is essential to sensorless field-oriented control. As with any type of motor control hardware, the gate drive must ensure efficient switching of the power transistors with minimal switching loss. It must also protect the inverter from shoot-through, the condition where both transistors in the half-bridge momentarily turn on, shorting the DC bus. In sensorless control, good gate drive design also plays an important role in reducing signal bias and noise.

One method of implementing gate drive is shown in Figure 13 [3,6]. Only one phase is shown; the others are identical. This method uses gate drive optocouplers, devices that take as their input an LED drive signal and output a push-pull gate drive signal. The optocouplers provide isolation between the logic and PWM-generating circuitry and the three-phase inverter itself. By connecting the optocoupler input LEDs of each half-bridge in reverse-parallel, it is ensured that only one gate drive output can be on at any given time.

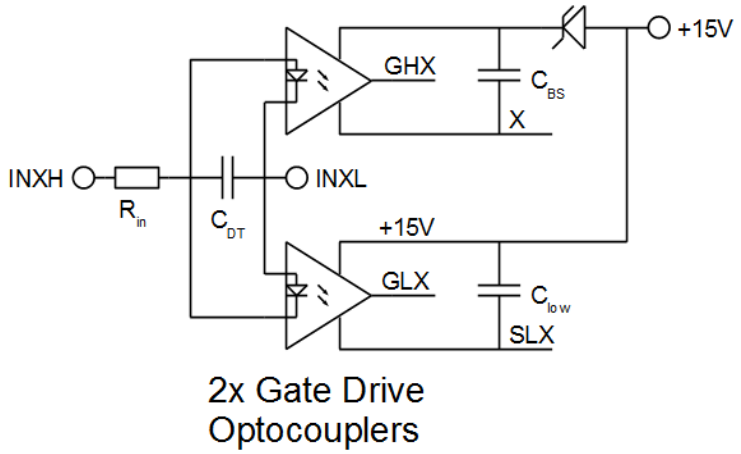


Figure 13: An optocoupled gate driver for one half-bridge of the inverter.

The gate drive is supplied by a DC voltage, typically 15V, which is enough to turn the MOSFETs fully on. This voltage is typically referenced to 0V, but in order for the FETs to turn on they require a gate voltage referenced to their source. For low-side FETs, this is not usually a problem since the low side source is close to 0V. (There is some offset due to current sense resistors but it's typically less than 1V.)

For high-side FETs, the gate voltage must be driven above the phase output voltage, which can be up to DC+. An isolated +15V supply, referenced to the phase output voltage, could be used on each high side gate drive. However, this solution is expensive and large. A technique called bootstapping is often used instead to generate voltages higher than DC+. A bootstrap capacitor (C_{BS} above) is charged from +15V through a diode while the low-side FET is on (phase voltage X is near 0V and the diode can conduct). C_{BS} creates a floating supply referenced to X that is used to drive the high-side gate. Once the high-side FET is on, X is near DC+ and the bootstrapping diode blocks C_{BS} from discharging back to the gate drive supply. Because of the need to recharge the bootstrap capacitor, there is a maximum PWM duty cycle achievable with this type of gate drive. (Typically 90-95% or higher.)

Connection of the low-side gate drive to its FET is an important design consideration. Ideally, the gate drive should be referenced to SLX, directly connected to the low-side source of its FET as near to the source pins as possible. In this way, gate drive current does not pass through current sense resistors (R_{CS} in Figure 12) or through areas of high current where ground transients could be developed. This ensures gate drive currents are not included in current sense measurements (although these measurements should not be taken during switching events anyway, see Section 1.3.2). However some gate drivers specify that this connection be tied to logic ground. If this is the case, connecting to 0V may reduce ground noise. Using isolated gate drivers, even on the low side, eliminates the need for this tradeoff.

An important function of the gate driver is deadtime generation. Deadtime is time between the turn-off of one gate and the turn-on of the other in a half-bridge. The deadtime ensures that, even though it takes some time to charge or discharge the gates, there is no overlap of time where both FETs are on (shoot-through). In Figure 13, capacitor C_{DT}, along with LED current-limiting resistor R_{LED}, determine the deadtime. The deadtime can be set based on the expected gate switching times and confirmed by measuring both gate voltages with an oscilloscope.

Usually, deadtime generation results in a small fraction of time where both FETs are off. In this period of time, depending on the direction of current flow in the phase, one of the FET body diodes will be forced to conduct and will drive the phase voltage to near 0V or near DC+. Because of this, there is some uncertainty in the average voltage equation presented above. The uncertainty can lead to a voltage estimate bias, which negatively effects sensorless position estimation (see Section 2.2.2). The deadtime is especially important at extremely low duty cycles, where it may even eliminate t_{XHon} entirely.

There are several ways to mitigate the influence of deadtime bias. The easiest is to use as little deadtime as necessary to ensure no shoot-through, and to use a lower PWM frequency. If the deadtime is a small fraction of the total PWM period, its influence can often be neglected. The remainder of this article assumes this to be the case. In some cases this isn't possible, so the voltage used in sensorless position estimation algorithms must account for the deadtime. A more detailed method for calculating average voltage, including the discontinuities and nonlinearities of deadtime, is one way to achieve this. A thorough analytical method would also have to account for current-dependent flyback diode conduction. Another way is to directly measure the phase voltage at high sample rate and average it in software.

1.3.2 Current Sensing

Current sensing is already a critical feature of field-oriented control since the goal of this control strategy is to control the current vector, as in Section 1.2.4. However, one of the benefits of field-oriented control is that it allows the measured currents to be filtered in the d/q frame of reference, where they are steady-state DC quantities. This means the filter time constants are only limited by the desired torque- and field-control bandwidth, not by the electrical frequency of the motor. It is possible to use relatively noisy current sensors, heavily filtered, if the torque bandwidth requirement is not too high.

With sensorless field-oriented control, this is not the case; the current sensors must be both fast and clean. The d/q-frame filters can still be used for current control, but high-bandwidth, low-noise, low-bias current measurements are crucial for accurate position estimation in the fast loop. For that reason, more careful design of the phase current sensors and signal processing is necessary in sensorless control. Some of the design considerations are explored in this section.

The two most common ways to measure phase currents are with Hall effect sensors directly on the phase outputs and with current shunt resistors at the base of each half-bridge. Hall effect current sensors have the advantage of complete isolation from the power circuitry, so they can go on the phase outputs directly without coupling noise on to the measurement circuitry. They can also measure current at any point in the PWM cycle, offering more opportunities for sampling. There are many off-the-shelf Hall effect sensors that produce pre-amplified linear outputs proportional to current. They come in a variety of sense ranges and most have bandwidths that are high enough for sensorless field-oriented control (50-100kHz).

Current shunt resistors are a smaller and less expensive current sensing solution, well-suited for space-constrained controllers. They are placed at the base of the half-bridge of each phase for which current is to be sensed. In Figure 12, all three phases have current sense resistors, labeled R_{CS}. More typically, only two of the phase currents are sensed. The third is calculated so that all three sum to zero, as in Figure 11. It is technically possible to use current sense resistors on the phase outputs directly, but since they are high dV/dt nodes, the current sense amplifier would need unrealistically high common-mode rejection ratio in order for the signal to be useful.

Figure 14 shows a typical low-side current sense amplifier. The voltage created across the sense resistor is scaled with an op-amp configured as a differential amplifier. The sense resistor is typically 1% tolerance or better, and is sized to minimize power loss, so its voltage drop is usually 100-1000x smaller than the DC bus voltage at rated current. Amplification is necessary in order to scale this voltage back up to the range where a typical analog-to-digital converter can read it with high resolution (3.3V or 5V full scale).

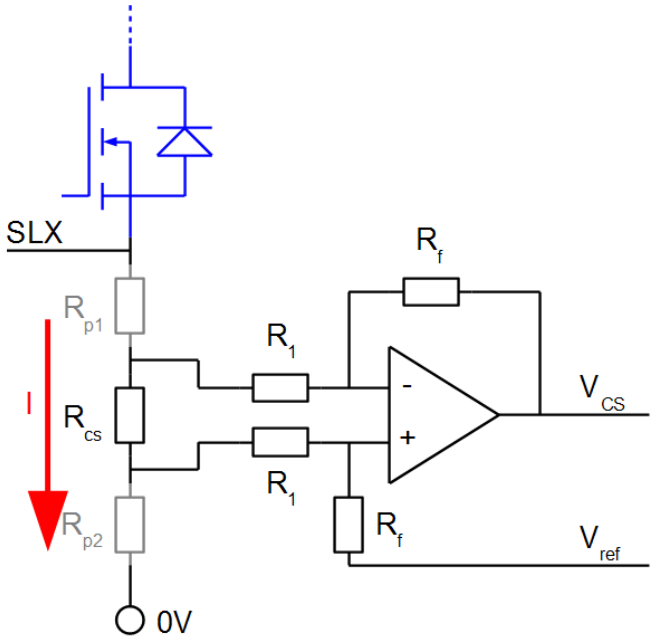


Figure 14: An op-amp in differential configuration for measuring the voltage across a low-side current shunt resistor.

The differential configuration is used in order to sense only the voltage drop across the current sense resistor only. (Referencing the negative input to 0V would include voltage drop across high-current traces between the sense resistor and the analog ground tie to 0V, shown using a parasitic resistor R_{p2} in Figure 14. There is another parasitic resistor, R_{p1}, from the MOSFET source to the current sense resistor terminal that also should not be included in the current sense measurement. For this reason, it's important that *both connections to the current sense amplifier be made as close as possible to the current sense resistor terminals*.

Using the differential op-amp configuration, the output voltage is symmetric about the reference voltage, V_{ref}, and is given by:

$$V_{CS} = V_{ref} + \left(\frac{R_f}{R_1}\right) I R_{CS}$$

Using typical op-amps, amplification ranges of 10-100x are possible. The bandwidth may be limited by the chosen op-amp's gain-bandwidth product. When using low-side current shunts, it's important to have a bandwidth significantly higher than the PWM frequency. The current sense resistor only conducts during the low-side FET's on time, so there may be only a small window in which to sense the current. A higher current amplifier bandwidth is necessary to ensure that the output signal has time to settle before sampling occurs, as shown in Figure 15.

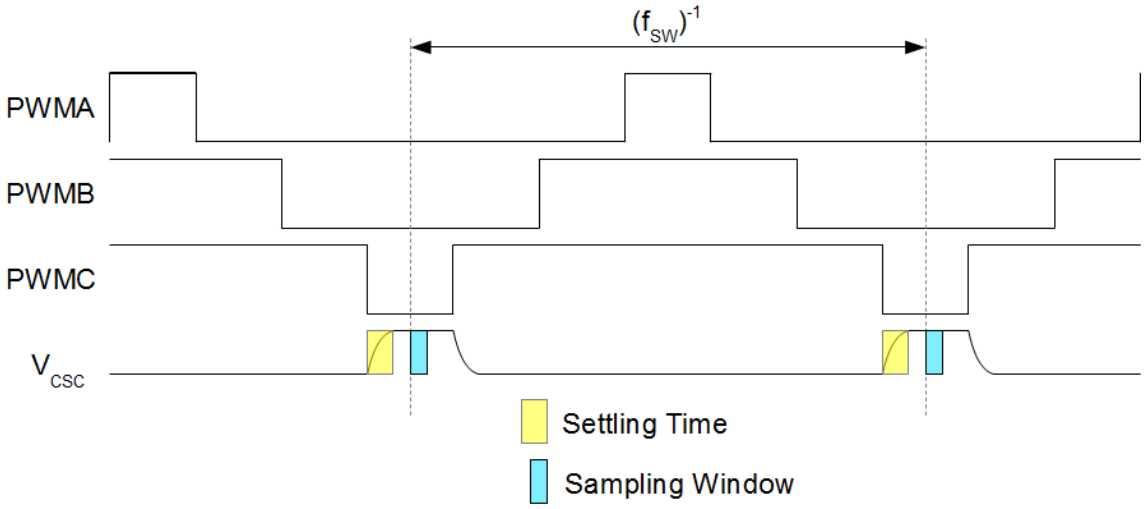


Figure 15: PWM-synchronous sampling of low-side current sense amplifier.

Figure 15 also illustrates the concept of *synchronous sampling*. Phase current is sampled synchronously with the PWM outputs of the controller, during a period where all three PWM outputs are low (all current sense resistors are conducting). This can be ensured to occur once per PWM cycle by using center-aligned PWMs with a minimum low-side on-time. The minimum low-side on-time is equivalent to a maximum PWM duty cycle, which may be required anyway if boostrapped high-side gate drive is used. Synchronous sampling imposes another constraint on the maximum duty cycle: the minimum low-side on-time must be sufficient to allow the current sensor output to settle, and then be sampled by the analog-to-digital converter.

Synchronous sampling is necessary for shunt resistor-based current measurement, but it can also be used with Hall-effect sensors. In both cases, synchronous sampling ensures that current is measured outside of any switching transients, so dV/dt- and dI/dt-induced noise is less likely to effect the current measurement. In order to use it effectively, the microcontroller coordinating the PWMs must be able to trigger its analog-to-digital converter to sample at an exact position in the PWM cycle, as shown in Figure 15. Not all microcontrollers have this capability, so it is a hardware requirement that should be considered when designing for sensorless field-oriented control.

Good board design practices and synchronous sampling can help reduce noise on the current sensor, but another important characteristic, especially for sensorless control, is bias. As will be explored in Section 2.2.2, the basis of the sensorless control method used in this article is a flux observer that relies heavily on sensed current *and its integral*. Even a small amount of current sensor bias can cause integrator drift, so a high-pass filter is also applied. This high-pass filter limits low-speed performance of the flux estimator. So, by minimizing current sensor bias, the high-pass filter can be made less aggressive (passing relatively lower frequencies) and the sensorless algorithm can operate at lower speeds. The quantitative effect of current sensor bias on the flux estimate is derived in Section 2.2.2.

2 Sensorless Control (Gen. 1)

In Section 1.2, the technique of field-oriented control as it can be applied to a permanent magnet synchronous motor was presented. It offers benefits of increased efficiency, reduced torque ripple, and

more accurate torque and field control compared to brushless DC methods. However, it requires high-resolution rotor position information in order to properly project motor quantities into the rotating d-q frame of reference (oriented to the rotor field) and control the *vector* motor current.

This rotor position information can come from a rotational position sensor such as an optical encoder, analog Hall effect sensors, or a resolver. However, these sensors add cost and complexity to a motor control system and can also reduce reliability. The goal of sensorless field-oriented control is to eliminate the need for rotor position sensors by using already-known or already-sensed electrical quantities, primarily voltage and current, to derive a rotor position estimate.

The method discussed in this article applies specifically to surface permanent magnet motors with non-salient rotors. These motors are not good candidates for inductance sensing methods that could be used to detect rotor inductance that is a function of electrical angle. In this type of motor, there may still be position-dependent inductive effects that can be measured at zero speed due to nonlinearities of the stator's B-H curve and the position of the strong rotor magnets. However this is not assumed here and methods that rely on B-H curve nonlinearities are outside the scope of this article. The method discussed here applies equally well to a motor with weaker magnets and little or no stator saturation.

One typical sensorless field-oriented control method involves the use of observers to indirectly obtain the motor's back EMF signal. (In BLDC control, this signal can be measured directly on the undriven phase. However, in field-oriented control, all phases are assumed to be driven at all times.) The observer uses known quantities of driven voltage and measured phase current, as well as the electrical model of the motor (Figure 2), to produce an estimate of the back EMF. Since the back EMF is a function of rotor position, the rotor position can be acquired from the observed back EMF. An example of this method is detailed in [1].

Because back EMF is only present when the motor is moving, the back EMF observer requires the motor to already be spinning in order to converge on a rotor position estimate. The back EMF is also speed-dependent. At low speeds, both analog noise and digital quantization may deteriorate the quality of the back EMF signal. So, the observer has a minimum operating speed and a startup method for getting from zero speed to a speed at which the back EMF observer can converge is needed.

Rotor flux linkage offers a more suitable target for the sensorless observer. Like back EMF, the rotor flux linkage is a function of rotor position. (As will be utilized in 2.2, the back EMF is just the derivative of the rotor flux linkage.) However, the rotor flux linkage is *not* a function of motor speed. The signal level remains constant at all speeds. Although they are mathematically related by just an integral transformation, the practical implication is that the flux observer can operate at lower speeds than a back EMF observer. Analog noise is filtered out by the flux observer's numeric integration of voltage and current signals. And the observed signal amplitude is constant, so digital quantization effects can be avoided at low speeds.

For reasons that are outlined in Section 2.2.2, operation at zero and extremely low speeds is still limited by sensor bias, which causes drift in the numeric integration. However, the limitations can be quantified and used to predict the operating frequency limits, as well as guide the design of a startup routine to get from zero to the target minimum operating speed.

2.1 Overview

Figure 16 shows a block diagram overview of the sensorless control method implemented in this article. The modified synchronous current regulator, discussed in Section 1.2.5, is also included to show how it is integrated with the sensorless position estimation algorithm. The angle θ, which is the estimated electrical angle of the rotor, is the output of the sensorless position estimation algorithm. It drives both the inverse Park transform and the phase of the sine-wave generator used by the modified synchronous current regulator.

For the most part, information flows from left to right on the diagram. The inputs are the three phase voltages and currents ($v_{\{A,B,C\}}$ and $i_{\{A,B,C\}}$). These are not all measured quantities. Known values for the driven voltages can be used instead of direct voltage measurement. It *is* necessary to measure the DC bus voltage in this case, since the voltage outputs will be proportional to that. It is also necessary to account for deadtime, or ensure that it is not a significant fraction of the PWM cycle (see Section 1.3.1). Only two of the three phase currents need to be sensed, since the third must satisfy Kirchoff's Current Law, summing to zero with the other two.

The voltage and current inputs feed three independent flux observers, which use a model of the motor to estimate the rotor flux linkage on each phase. This model relies on estimates of the resistance and synchronous inductance of the motor. The outputs of the flux observers are then fed to software comparators to detect flux zero crossings. These zero-crossings trigger a position estimate update. In between updates, time-based extrapolation of the rotor position provides gap-filling resolution for the position estimate.

This method deviates from the more standard practice, which would be to implement observers in a two-axis stationary frame (via Clarke transform) on α and β flux linkage or back EMF and then use a trigonometric method to calculate the rotor position. Instead, flux observers are implemented on all three axes and position estimation is done in much the same way it would be done with three-phase digital Hall effect switches, using time-based extrapolation [3]. This requires no Clark transform, no inverse tangent implementation, no angle-tracking observer [4]. The computation requirement is thus extremely minimal.

The accuracy of this method depends on the update rate of the speed estimate and on the speed and acceleration of the rotor. It is easily quantified (see Section 2.4) and for a wide range of operating conditions is completely adequate for achieving a position estimate with enough resolution and accuracy for field-oriented control.

Each of the blocks in the overview diagram of Figure 25 will be discussed in more detail in the following sections.

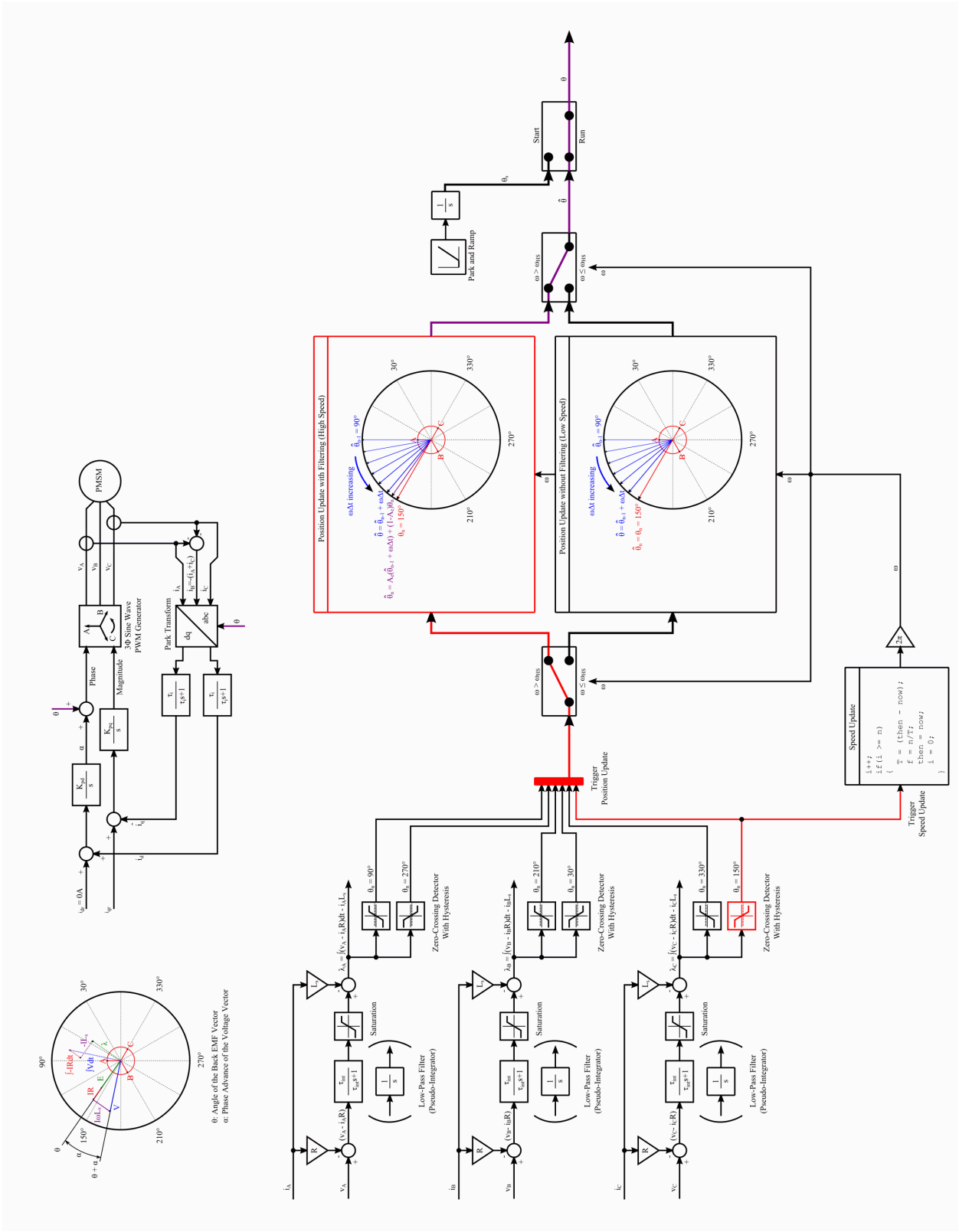


Figure 16: An overview of the sensorless control method discussed in this article.

2.2 Flux Observer

In this implementation of sensorless field-oriented control, each of the three phases of the PMSM has its own flux observer, which estimates the instantaneous *rotor flux linkage* of that phase winding. Like the back EMF, the rotor flux linkage contains information about the position of the rotor. However, unlike back EMF, the flux linkage waveform does not vary in amplitude as a function of speed. This makes the flux linkage a more suitable state to estimate, especially at low speeds.

The flux observer for each phase takes the form show in Figure 17. The observer is based on the state equation of a single phase of a round-rotor PMSM driven by a balanced three-phase voltage source:

$$\frac{di_A}{dt} = -\frac{R}{L_s} i_A + \frac{1}{L_s} v_A - \frac{1}{L_s} e_A$$

which comes directly from the PMSM model detailed in Section 1.1. The state equation can be rewritten in integral form to find an expression for the flux linkage:

$$i_A L_s = \int (v_A - i_A R) dt - \int e_A dt$$

$$\lambda_A = \int e_A dt = \int (v_A - i_A R) dt - i_A L_s$$

The rotor flux linkage on Phase A, λ_A, is obtained by first integrating the applied voltage, v_A, less the resistive voltage drop, $i_A R$, then subtracting the stator-induced flux linkage, $i_A L_s$. The integral formulation is captured in block diagram form in Figure 17 with some modifications for practical numeric integration, discussed in Section 2.2.2.

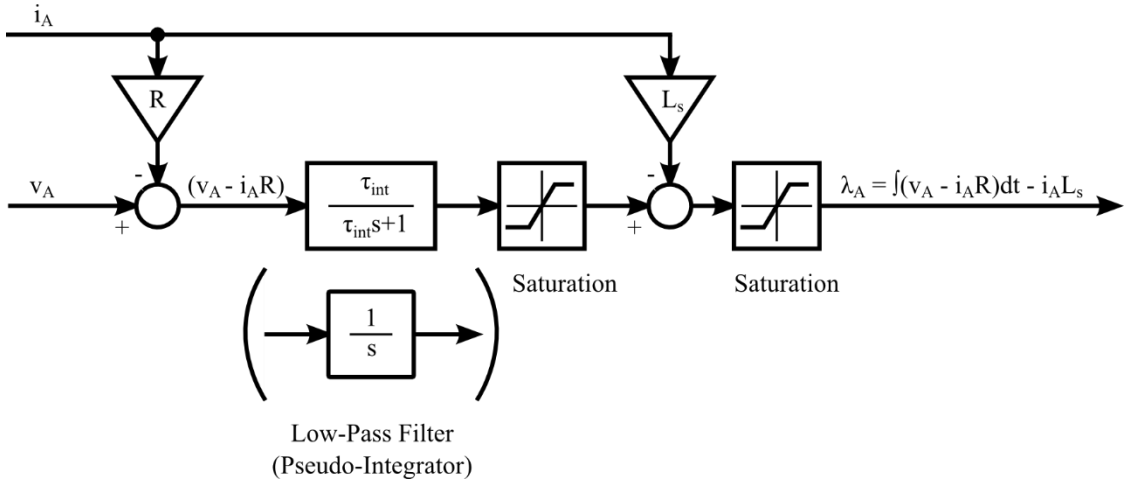


Figure 17: One phase (Phase A) of the flux observer.

2.2.1 Inputs

The inputs to the flux observer are the phase voltage, v_A, and the phase current, i_A. The phase voltage is derived from the measured DC bus voltage and the known PWM duty cycle commanding that phase. The phase current is a measured quantity, sampled from a current sensor on that phase. This implementation assumes that the measured phase current is the true state. By contrast, a Luenberger observer structure

would use the state equations to predict a current, which would then be compared to the measured current. The error would be fed back into the model to improve the current and flux estimates [1].

Derivation of the phase voltage input to the flux observer involves both measured and calculated values. Any time the power stage for a particular phase is switching, the instantaneous phase voltage is either v_{DC} or 0V (neglecting switching transients, transistor voltage drops, and deadtime). However, the state equation is based on the *average* phase voltage, taken over an integer number of switching cycles. The average voltage is a function of the DC voltage and the fraction of time for which the PWM output is high, the duty cycle. Figure 18 shows how the average phase voltage, $v_{\{A,B,C\}}$, is derived.

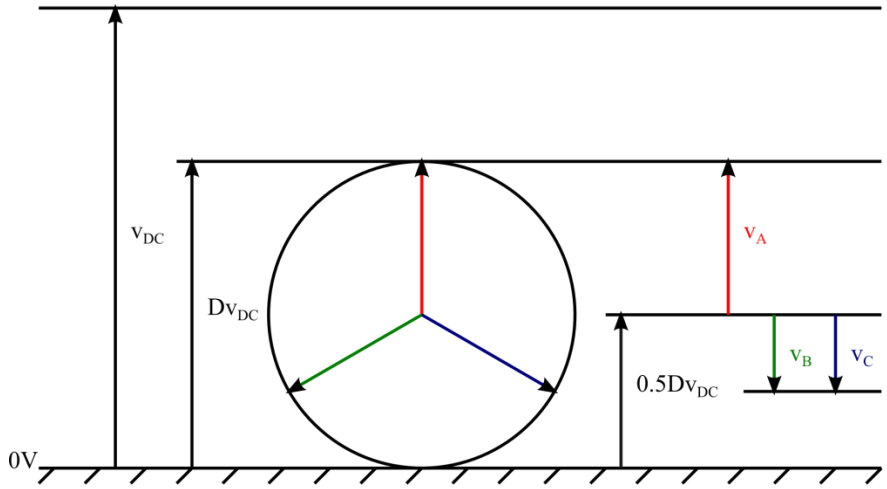


Figure 18: The three phase voltages are a function of the maximum duty cycle, D, the DC bus voltage, V_{DC}, and the angle of the three-phase voltage generator. At this particular angle, $v_A = 0.5Dv_{DC}$ and $v_B = v_C = -0.25Dv_{DC}$.

If the power stage for a particular phase is *not* switching, a direct measurement of the back EMF is possible and this value could be substituted as the input to the flux observer. (No current is flowing, so $v_A = e_A$ and $i_A = 0$A.) This approach can be used for unpowered startup or for restarting a motor after a fault has occurred but before the rotor has stopped spinning.

The phase current input, $i_{\{a,b,c\}}$, is a measured quantity that comes directly from two phase current sensors. (The third phase current is derived from Kirchoff's Current Law, which requires $i_a+i_b+i_c=0$.) Since the phase currents come from real sensors, there is the possibility for bias that can negatively affect the flux estimate, as discussed in the Section 2.2.2. For this reason, accurately zeroing the current sensors at startup is important.

2.2.2 Integrator, Filtering, Bias, and Saturation

Implementing a pure integrator in a real system with sensor bias and numeric rounding errors is not feasible, so a pseudo-integrator that works for the frequencies of interest is needed. One way to deal with biased signals is to add a high-pass filter in series with the integrator. This simplifies to a low-pass filter with a gain equal to its time constant:

$$\left(\frac{\tau_{int}s}{\tau_{int}s+1}\right)\left(\frac{1}{s}\right)=\frac{\tau_{int}}{\tau_{int}s+1}$$

At high frequencies ($s \to \infty$), this filter behaves like an integrator:

$$\lim_{s\to\infty}\left(\frac{\tau_{int}}{\tau_{int}s+1}\right)=\frac{1}{s}$$

For practical purposes, it is useful to define a lower bound for the filter time constant, τ_{int}, based on the minimum desired closed-loop sensorless operating frequency of the motor. For example, to ensure no more than 10° of phase lead compared to a pure integrator:

$$\tan^{-1}(\tau_{int}\omega_{min}) > 80°$$

$$\tau_{int} > \frac{5.671}{\omega_{min}}$$

In other words, the filter's cutoff frequency should be at least 5.7x slower than the lowest electrical frequency of the motor when operated under closed-loop sensorless control. Figure 19 shows the minimum required time constant to stay below 10° of phase lead compared to the pure integrator at different speeds for a 14-pole motor.

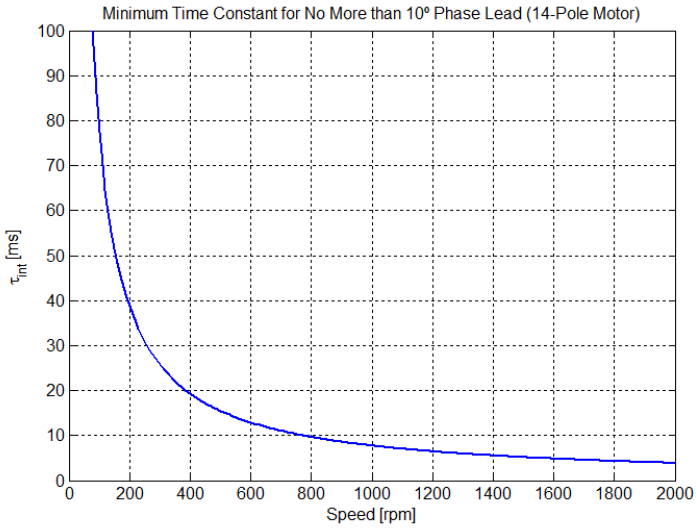


Figure 19: Minimum required LPF time constant for less than 10° of phase lead on a 14-pole motor.

While the minimum operating frequency only sets a lower bound on the filter time constant, there is good reason not to user a higher time constant than necessary. The time constant is also the DC gain of the first-order low-pass filter:

$$\left.\frac{\tau_{int}}{\tau_{int}s+1}\right|_{s=0}=\tau_{int}$$

This DC gain acts on any sensor or rounding bias in the phase voltage and current inputs. Higher values of the filter time constant, τ_{int}, will lead to a larger offset of the flux estimate. The total DC bias of the flux estimate is given by:

$$\lambda_{A,DC} = (v_{A,DC} - i_{A,DC}R)\tau_{int} - i_{A,DC}L_s$$

The voltage bias, $v_{A,DC}$, is only a concern if there are rounding biases in the math used to generate the phase voltage input, or if the gate drive deadtime is a significant fraction of the total PWM period (See Section 1.3.1). The current bias, $i_{A,DC}$, is a reality of the current sensor and must be accounted for. To minimize its effect on the flux estimate, the current sensor should be zeroed at startup and the filter time constant, τ_{int}, should be kept as short as possible given the constraints above. Even after these measures, it is important to check that the worst-case bias is small compared to the magnitude of the flux waveform using the equation above.

Integrator saturation can also be used to mitigate the effects of input bias. If the expected magnitude of the flux waveform is known, a saturation limit on the pseudo-integrator can be enforced. A maximum value for the integrator saturation limit can be calculated by:

$$\left[\int (v_A - i_A R)dt\right]_{limit} = \lambda_{A,max} + i_{A,max}L_s$$

A further limit on the flux estimate itself can be imposed if necessary. If the expected flux magnitude is not known a-priori, it is possible to use adaptive limits based on the measured flux magnitude. In addition to limiting the effects of bias, the saturation limits prevent integrator windup during startup or transient conditions.

Integrator and output saturation should not be used as a substitute for other means of eliminating input bias. When the integrator or the flux hit their limits, the waveforms are clipped and the performance of the position estimate will suffer. It is even possible to hear the clipping when the saturation limits begin to take effect. The limits can help prevent loss of synchronous commutation in extreme transients, but most of the work of dealing with bias should be done with carefully zeroed sensors and a well-chosen filter time constant.

A second-order filter could also be used as a pseudo-integrator that would eliminate DC sensor bias. Adding another first-order high-pass component to the previous filter gives:

$$\left(\frac{\tau_{int}}{\tau_{int}s + 1}\right)\left(\frac{\tau_{HPF}s}{\tau_{HPF}s + 1}\right) = \frac{\tau_{int}\tau_{HPF}s}{\tau_{int}\tau_{HPF}s^2 + (\tau_{int} + \tau_{HPF})s + 1}$$

This also behaves like a pure integrator at high frequencies, but has the advantage of zero DC gain:

$$\lim_{s\to\infty} \frac{\tau_{int}\tau_{HPF}s}{\tau_{int}\tau_{HPF}s^2 + (\tau_{int} + \tau_{HPF})s + 1} = \frac{1}{s}$$

$$\left.\frac{\tau_{int}\tau_{HPF}s}{\tau_{int}\tau_{HPF}s^2 + (\tau_{int} + \tau_{HPF})s + 1}\right|_{s=0} = 0$$

One disadvantage of the second-order filter is that it introduces more phase lead at low frequencies than the simple low-pass filter (see Figure 20), so care must be taken to choose the two time constants based on the motor's minimum electrical frequency for sensorless control. In general, if $\tau_{HPF} >> \tau_{int}$, the second-order filter will not add a significant amount of phase lead compared to the 1st-order filter in the range of motor speeds where the phase lead of the first-order filter is less than 10° (see Figure 21).

Since the phase lead induced by both the first-order and second-order filters is a known function of frequency, it is also possible to counteract the effect of the phase lead on the rotor position estimate [5]. As will be discussed in Section 2.3, the zero-crossings of the three-phase flux estimates are what drive the position estimator. These zero-crossings are shifted at low frequencies due to the phase lead associated with the flux estimate filters. It is entirely possible to subtract the speed-dependent phase lead from the final position estimate using an interpolated look-up table. This would significantly extend the low-speed operating region.

Another disadvantage of a higher-order filter is that it requires more computation, but with high-speed fixed- or floating-point processing, three second-order digital filters (one per phase) could definitely be implemented at PWM frequency.

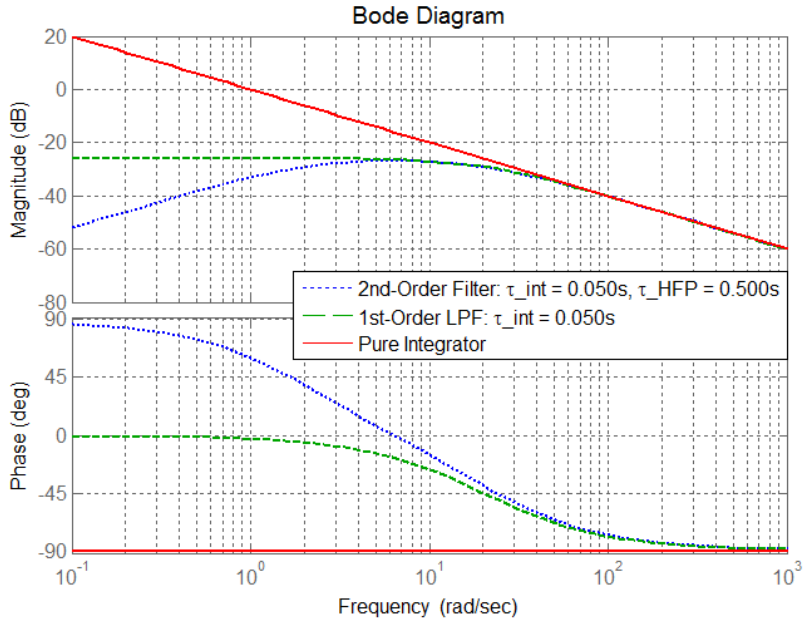


Figure 20: Magnitude and Phase comparison of pure integrator, 1st-order low-pass filter, and 2nd-order filter.

Figure 21: When $\tau_{HPF} >> \tau_{int}$, the 2nd-order filter does not add a significant amount of phase lead for the range of motor speeds where phase lead from the 1st-order filter is less than 10°.

2.3 Flux Comparator

After the three phase flux observers generate flux estimates, the estimates are processed by a software comparator that detects zero-crossing events. Even though the flux estimate and the rotor position estimate are continuous signals (updated at every PWM/sampling loop), the zero-crossing events are used as discrete triggers to update the position estimate. In this way, the flux zero-crossing comparators produce a similar signal to Hall effect sensors. Unlike Hall effect sensor-based commutation, the position estimate is interpolated between each zero-crossing event to produce a continuous, high-resolution position estimate.

Other methods of producing a continuous position estimate from the flux observers include a phase-lock loop, and angle-tracking observer [4], and direct trigonometric calculation [1]. These methods do not use any dead-reckoning, so they offer advantages when rotor speed changes rapidly. The method implemented here was chosen for its robustness, ease of implementation, and ease of integration with existing Hall effect sensor-based commutation. Its limitations in terms of rotor acceleration are discussed in Section 2.4.

2.3.1 Relationship to Hall Effect Sensors

The inspiration for using flux zero-crossings to trigger position update events came from previous work with Hall effect sensor-based field-oriented control. Hall effect sensors are common in brushless DC motor control and are used to generate a six-step commutation waveform as shown in Figure 22. Using the sensor states alone, the commutation resolution is 60°. However, interpolation can be used to create a more accurate position estimate in between sensor events [6], enabling the use of field-oriented control techniques with relatively inexpensive Hall effect sensors.

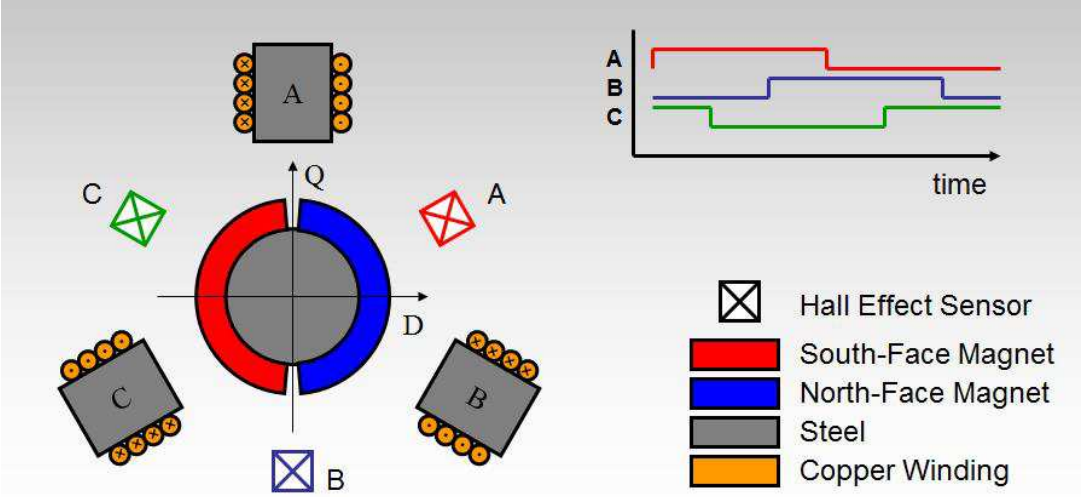


Figure 22: BLDC position sensing using 3-phase Hall effect sensors.

The Hall effect sensors, generally placed inside the motor, detect the polarity of the magnetic field in their vicinity and produce a digital signal to indicate whether they are in North or South fields, with respect to their sensitive axis. They can either detect the rotor magnets themselves, or an identically-arranged set of smaller sense magnets. Separate sense magnets offer the advantage Hall effect sensing without influence of the stator field, giving a more accurate rotor position estimate.

The three-phase flux observers, with their zero-crossing comparators, serve as virtual Hall effect sensors. In a round-rotor permanent magnet motor, the rotor flux linkage is aligned exactly with the field of the permanent magnets on the rotor. Thus, detecting the flux linkage in Phase A determines whether the Phase A winding is primarily distributed under North magnets or South magnets at that instant in time. At the point where the Phase A is exactly at a magnetic transition, the rotor flux linkage is zero and the position update event occurs as it would with a Hall effect sensor.

The structure of the flux zero-crossing comparators and position update events make it very easy to merge sensored and sensorless control. Either the Hall effect sensor events or the flux zero-crossing events can be used to trigger a position update, or they can be used simultaneously. In initial development, having the Hall effect sensors running in parallel with the sensorless algorithm was very useful for debugging purposes, as the Hall effect sensors would provide a reference position against which to compare the sensorless position estimate (see Section 4.1). Beyond debugging, this paralleled sensor/sensorless position estimators could be useful in several ways:

- **Sensored Startup:** Startup without position sensors is difficult, since the back EMF signal isn't available for position estimation. Using sensors for startup and low-speed control can offer performance benefits such as higher starting torque. Above a threshold speed, the control can switch to a more efficient and reliable sensorless mode.

- **Sensor Timing:** One of the drawbacks of sensored control is obtaining an absolute position reference. The sensors may not be exactly aligned with a given phase winding, or their hysteresis may cause them to be differently phased in forward as in reverse. A parallel sensorless position estimate can be used to identify the phasing of the sensors.

- **Sensor Fault Detection / Backup:** The sensors can be used as the primary position measurement method but the parallel sensorless position estimate can be used to detect a sensor failure and provide an instant back-up method for estimating position.

2.3.2 Hysteresis, Hold-Off, Commutation Order

Because the flux estimates are inherently noisy, care must be taken to discount extra zero-crossings. Several methods can be used for this, including hysteresis, hold-off times, and commutation order enforcement. Though they are not mutually exclusive, one or two may be enough to handle all situations where noise can induce erroneous zero-crossings.

Hysteresis involves setting the low-to-high flux transition value slightly higher than zero and the high-to-low flux transition value slightly lower than zero. The space between the two transition values is the *deadband.* In this way, a noisy signal that has just triggered the low-to-high comparator is unlikely to trigger the high-to-low comparator if the noise amplitude is smaller than the deadband. The value of the deadband is adjustable, but should be small relative to the total amplitude of the flux waveform. As the deadband grows larger, a phase delay is introduced. An example of zero-crossing detection with hysteresis is shown in Figure 23a.

Hold-off sets an amount of time for which comparator triggers are ignored after a zero-crossing detection. A noisy signal will trigger the comparator once, then subsequent zero-crossing are ignored for some time, ideally until the signal is far enough away from zero that the noise cannot retrigger the comparator. Hold-offs can be applied to each phase individually or to all three phase flux comparators (global hold-off). Figure 23b shows a hold-off applied to an individual phase flux estimate. For a motor that operates over a wide range of speeds, a speed-dependent hold-off can be used (e.g. 5% of the expected commutation period).

Commutation order enforcement is a technique that is specific to multi-phase motor control, and involves a check to ensure that the zero crossing encountered is the one to be expected based on the last most recent zero crossing. It amounts to a state machine in which only one state transition is allowed from any given state (Figure 24). Other zero crossings are assumed to be caused by noise and are ignored.

The valid state transitions depend on the direction of commutation (forward or reverse), which must be determined independently in a way that does not directly use the states themselves. (Otherwise, noise at a transition could cause the motor to rapidly switch between forward to reverse.) For the most part, motor direction is set before the start of sensorless control. When switching from forward to reverse, the motor must restart from zero speed using some startup method before resuming sensorless control in the opposite direction.

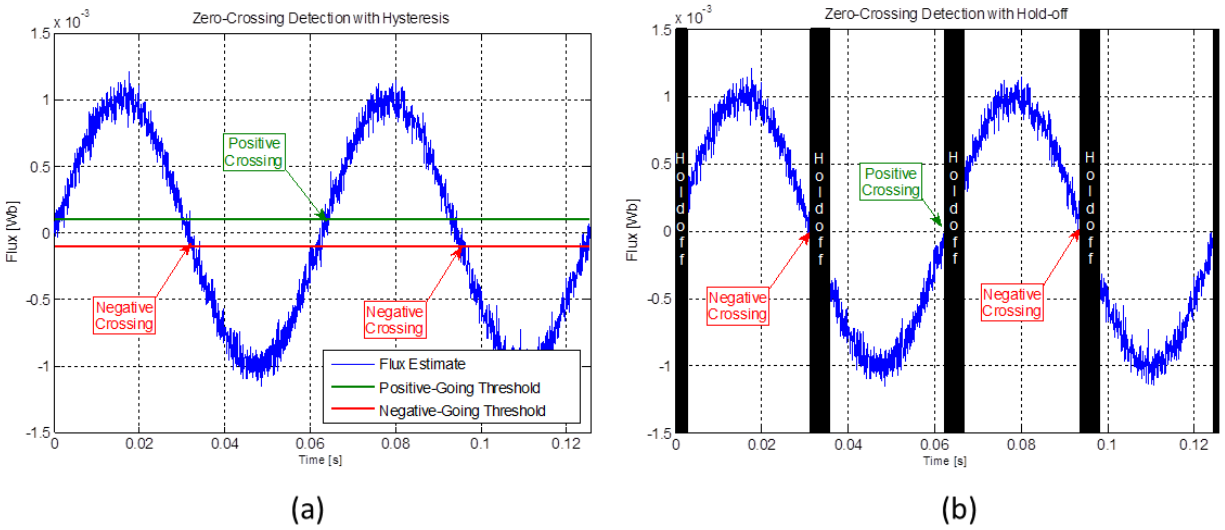


Figure 23: Two methods of do zero-crossing detection a noisy flux signal: using hystersis (a) and hold-off (b).

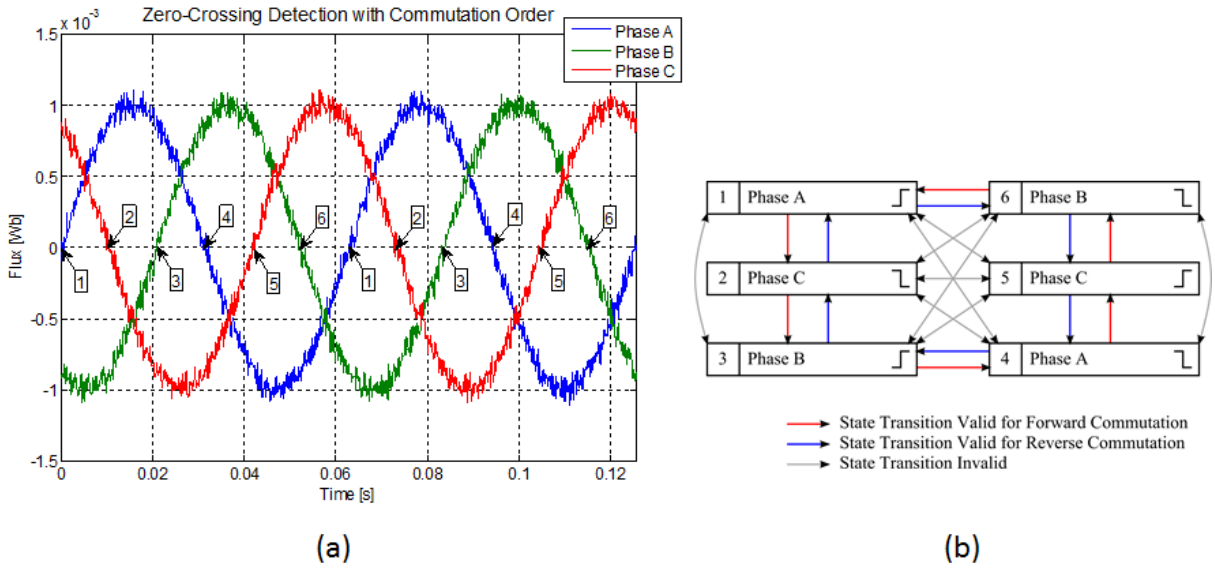


Figure 24: Three phase flux estimate zero-crossing detection showing commutation states (a) and valid state transition for enforcing commutation order (b).

2.4 Position Estimator

Like Hall effect sensors, the flux estimator zero-crossings alone give only six unique states per electrical cycle (a position resolution of 60° electrical). Field-oriented control requires a high resolution position estimate to work properly. Most sensorless position estimators use the entire flux (or back EMF) waveform to achieve higher-resolution position estimates. For example, an estimate of rotor electrical angle can be made by converting the three-phase flux values into a stationary frame (α, β) vector using the Clarke transform, and finding its angle using an arctan function [1].

The implementation presented here uses linear extrapolation to estimate the rotor electrical angle in the interval between flux estimator zero crossings. To do this, an estimate of the rotor speed must be acquired based on the time interval between a number of previous zero-crossings. This method imposes limitations on rotor acceleration, which are discussed later in this section. Two methods of extrapolating position are discussed here: one without filtering (Section 2.4.1) and one with filtering (Section 2.4.2). Figure 25 shows an overview of the position estimation method implemented here, with filtering switched on at higher speeds.

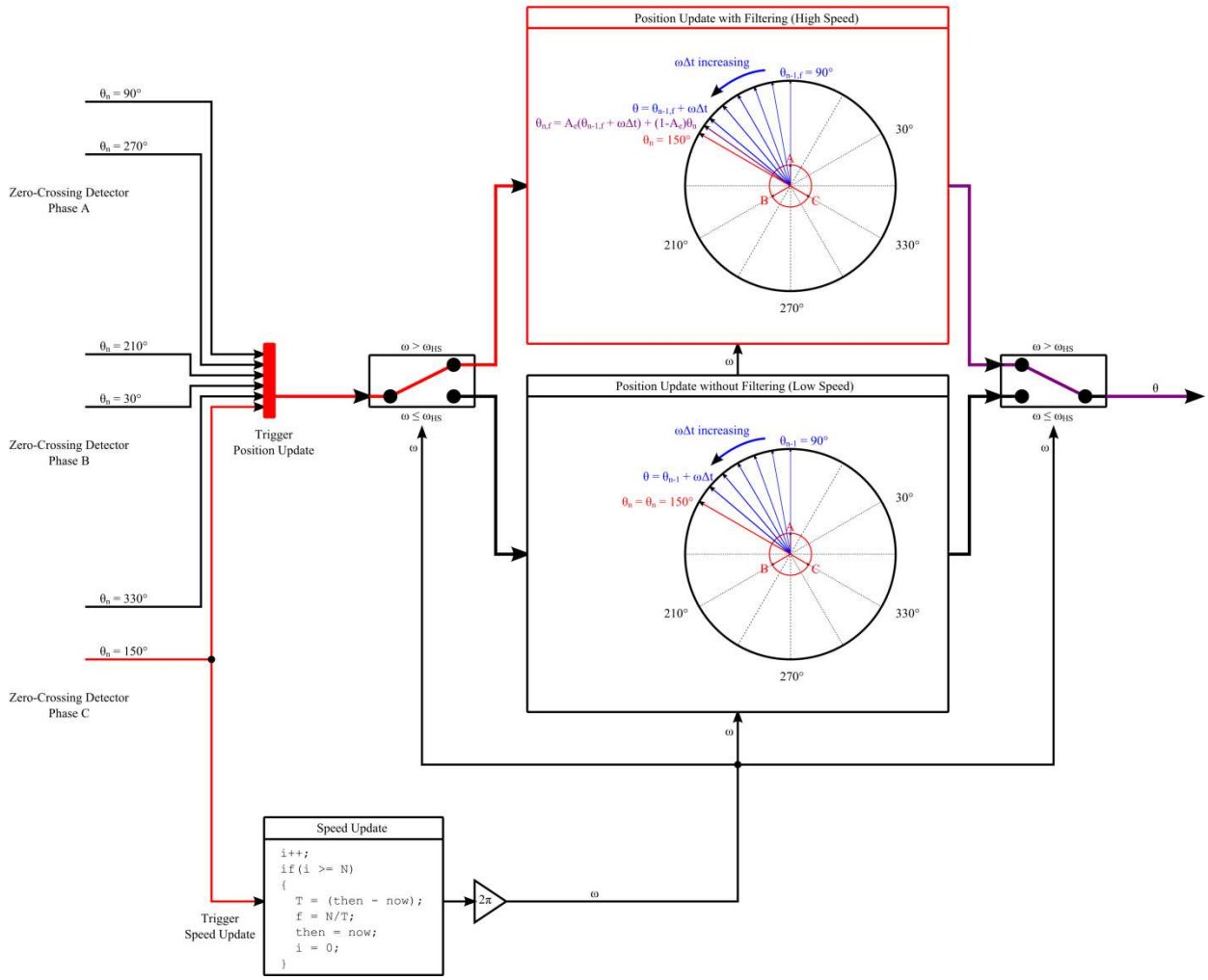


Figure 25: Overview of the position and speed estimation algorithm, triggered by flux estimator zero-crossings.

Some advantages of using linear extrapolation from zero-crossings for position estimation, as compared to methods that use the continuous flux waveform, are:

- **Computation Efficiency:** The method is simple to implement in fixed-point math, not requiring any trigonometric calculations. It requires only one division to determine the rotor speed between

a fixed number of zero-crossings.

- **Flux Waveform Tolerance:** Using zero-crossings with extrapolation allows the method to produce a ripple-free angle estimate even with non-sinusoidal flux waveforms.

- **Integration with Hall Effect Sensors:** The zero-crossing method allows integration with Hall effect sensors, which are essentially physical rotor flux zero-crossing detectors. The Hall effect sensors could be trigger inputs to the same position estimator in a combined method.

Disadvantage of the linear extrapolation method are:

- **Speed Estimate:** The method requires knowledge of the rotor speed, whereas methods using the continuous flux waveforms do not necessarily require the rotor speed as an input. (However, most observer-based sensorless methods still use rotor speed in the prediction step.)

- **Acceleration Error:** Because only a first-order linear extrapolation is used, the method has error during rotor acceleration. This error depends on the rotor speed, the rotor acceleration, and the interval over which rotor speed estimates are taken. It is discussed in detail in Section 2.4.

A higher-order forward predictor and/or the use of a current-based feed-forward acceleration estimate could yield lower angle prediction error in between zero crossings, at the expense of more computation. An observer that combines information from the entire flux waveform with a predictor based on the motor dynamics would improve the position estimate even more, but also introduces more computation and more parameter sensitivity.

Understanding the limitations of this simple position estimator is key to effective use of the sensorless control method implemented in this article. The following sub-sections describe the linear extrapolation-based position estimator and its limitations in more detail.

2.4.1 Linear Position Extrapolation

Linear position extrapolation fills in the gaps between flux estimate zero-crossings (or Hall effect sensor edges) to create a higher-resolution angle estimate. The position estimator runs at the same sampling frequency as the flux observer and PWM (15.625kHz, as implemented). Once per sampling period, the flux observers are updated and checked for zero-crossings. A zero-crossing indicates new sensor information and triggers a position update, which will be discussed below.

If no zero-crossing has occurred in the current sampling period, the angular position estimate is incremented by a small angle:

$$\Delta\theta = \omega T_s$$

where ω is the estimated electrical frequency of the motor and T_s is the sampling period. The total electrical angle extrapolated since the previous zero crossing is:

$$\theta_{inc} = \omega\Delta t = \omega n_s T_s = n_s\Delta\theta$$

where Δt is the total amount of time that has passed since the last position update, and n_s is the total number of elapsed samples since the last position update. This extrapolation continues until a position update is triggered by a new zero crossing, or the total extrapolated angle reaches a certain limit. The next position update is expected at $\theta_{inc} = 60°$, if the rotor speed is constant. Since the rotor speed may change,

the extrapolation limit must be set to a larger value. In this implementation, the limit is 90° such that at all times:

$$\theta_{inc} \leq 90°$$

This limit stops the linear position extrapolation from proceeding even if the rotor has completely stopped.

The incremental extrapolated angle is added to the absolute angle determined by the most recent position update, θ_{n-1}. Thus, the total angle estimate at any sampling period is the sum of the absolute angle from the previous position update and the incremental angle:

$$\theta = \theta_{n-1} + \theta_{inc} = \theta_{n-1} + \omega\Delta t$$

If a flux zero-crossing has occurred in the current sample, the simplest course of action is to treat the new position information with 100% confidence, using the trigger to set a new absolute angular position and reset the incremental extrapolated angle, θ_{inc}, to zero. This is the basis of the unfiltered position estimate block, shown in Figure 26, and is the focus of the rest of this sub-section. The position update may also filter the new position information with the extrapolated position, the topic of Section 2.4.2.

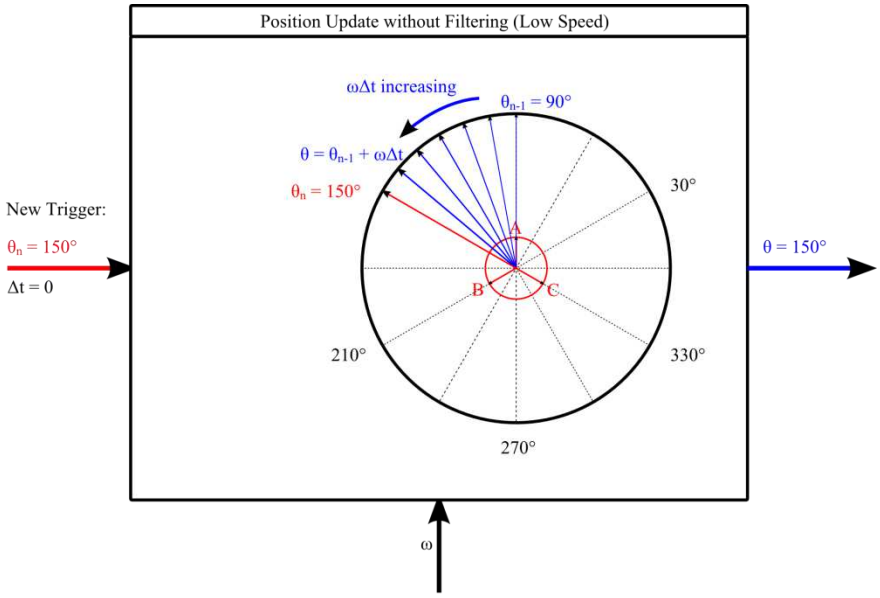


Figure 26: The unfiltered position estimator shown at the instant of a new flux zero-crossing trigger. The unfiltered estimator treats a new position from the flux zero-cossing detector with 100% confidence, setting the estimated angle to exactly this position and restarting the extrapolation timer.

The rotor speed used for the position estimator is itself an estimate based on previous position updates. At a minimum, the time between the two most recent zero-crossings could be used. However, this interval may span an insufficient number of sampling periods at high speeds, leading to numerical errors. The zero-crossings are also noisy, so some averaging is useful to get a relatively clean speed estimate. In this implementation, the speed estimate is taken over an integer number of complete electrical cycles, *N*, as shown in Figure 25. Thus, 6*N* position update intervals are averaged together for each speed update.

The additional filtering achieved by calculating the speed estimate over *N* electrical cycles also causes lag in the speed estimate. If the rotor is accelerating, the lagging speed estimate will result in position estimate error that can negatively affect field-oriented control. The worst-case position error, derived in Appendix A, can be calculated by:

$$\theta_{ex} = \left(1 - \frac{1}{\sqrt{1+K}}\right)A, \quad K = \frac{(18N-1)\alpha A}{\hat{\omega}^2}, \quad A = 60edeg$$

where α is the angular acceleration, $\hat{\omega}$ is the most recent speed estimate, and *N* is the number of electrical cycles over which the speed estimate is taken. The value *K* is a dimensionless acceleration parameter that relates to the error between the previous speed prediction and the actual speed of the motor. The value *A* is the electrical angle over which extrapolation occurs, 60edeg in this case.

For example, a 14-pole motor under a constant acceleration, α, of 10,000rpm/s, and with *N* = 1 has the maximum position error vs. estimated speed relationship shown in Figure 27

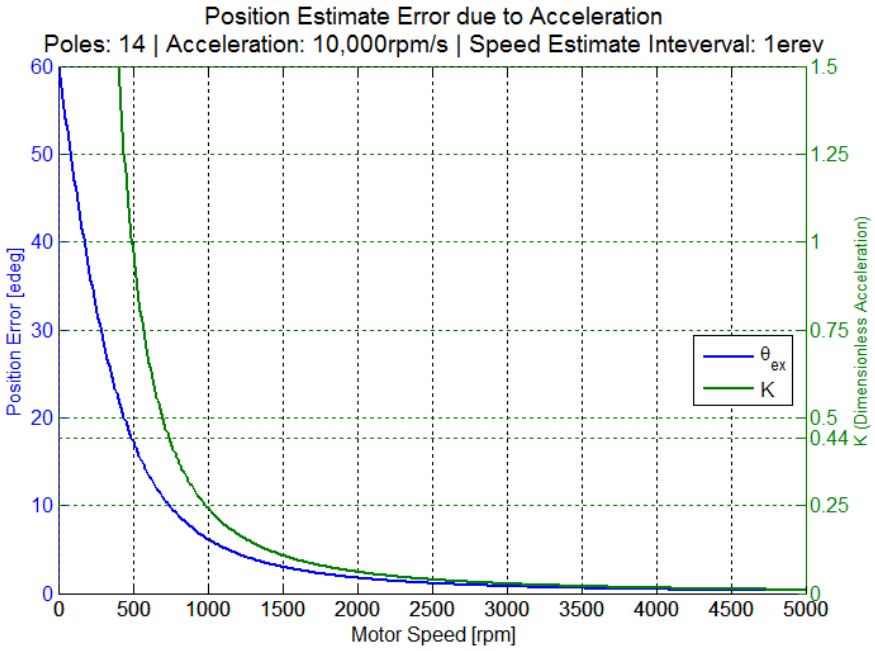


Figure 27: Position estimate error due to acceleration for a 14-pole motor with an acceleration of 10,000rpm/s and speed estimate updated every electrical cycle.

As expected, the maximum position error decreases with increasing speed. The speed changes less per speed update, which is reflected in the dimensionless acceleration parameter, *K*. When *K* is less than 0.44, the maximum position error due to acceleration is less than 10edeg. In Figure 27, this is true at speeds above 750rpm.

Because the position extrapolation is done digitally, it is subject to several other common discrete/numeric limitations. For example, the sampling delay can create lag at high speeds. As much as one sampling period may pass between the actual flux zero crossing and its sampled trigger. The worst-case position error resulting from this is a linear function of electrical frequency:

$$\theta_{es} = \omega T_s$$

The sampling delay creates larger error at high frequency and is always a lagging error. The error caused by rotor acceleration matters more at low frequency, and could be a leading or lagging error depending on the sign of the acceleration. These two bounds set the range over which the discrete linear position extrapolation method can be used, given a certain electrical angle error tolerance.

Figure 28 shows a simulation of the unfiltered position estimator error for a 14-pole motor undergoing 10,000rpm/s acceleration. The sampling frequency is 15.625kHz and the speed estimate interval is one electrical cycle ($N = 1$). The simulation was done from a standing start to 10,000rpm. The total worst-case error, which is the sum of the worst-case error from acceleration (Figure 27) and the worst-case error from digital delay, sets a maximum bound on the simulated position estimate error. The interval over which the worst-case position estimate error is less than 10edeg is quite narrow: about 1,000-3,400rpm. For a maximum error bound of 20edeg, the range is 500-7,300rpm.

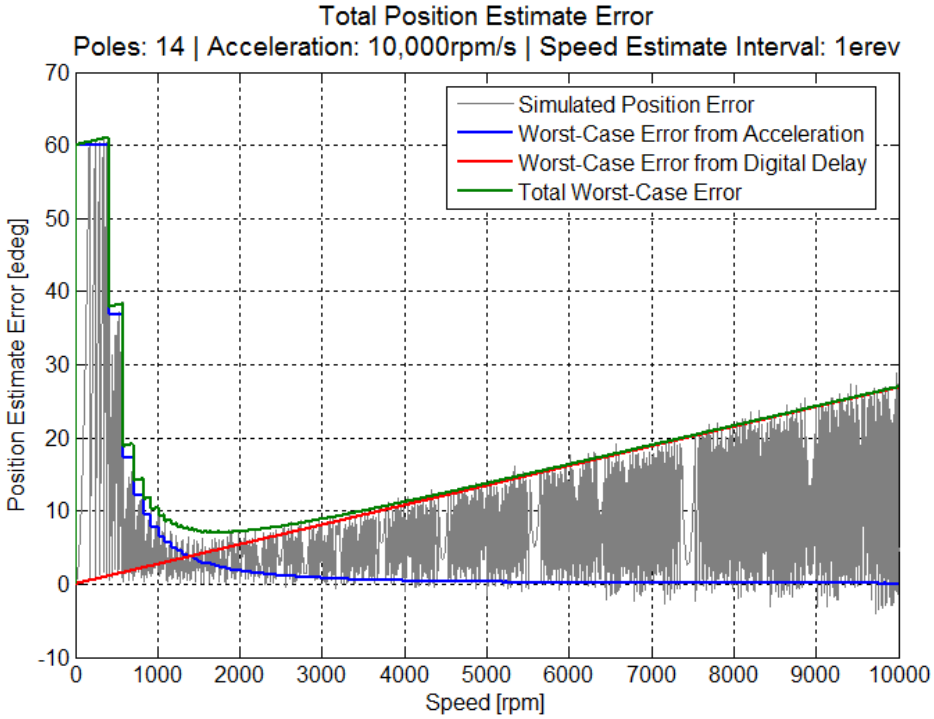


Figure 28: Simulated position estimate error for a 14-pole motor undergoing 10,000rpm/s acceleration. The sampling frequency is 15.625kHz and the speed estimate interval is one electrical cycle.

Although the position error caused by digital delay can be quite large at high speeds, the effect on field-oriented control performance is not as drastic as the position error suggests. The phase currents are also sampled at 15.625kHz and experience the same digital delay as the position estimate. Thus, when the delayed phase currents are projected onto the d/q axes using the delayed rotor angle, the resulting d-Axis and q-Axis currents should be unaffected.

The digital delay is still present between the voltage output and the sampled current measurements, but this is automatically compensated by the normal closed-loop D- and Q-Axis current controllers. This

suggests that the maximum speed of operation is higher than what the worst-case digital delay bound on the position estimate error indicates. However, since the exact discrete-time behavior has not been modeled or analyzed, keeping $\theta_{es} < 20°$ ($f_s > 18f$) is advised.

2.4.2 High-Speed Position Filter

Since the flux estimates are derived from sensor measurements (DC voltage and individual phase currents), they are contaminated by sensor noise. Thus, it is possible for a flux zero-crossing to be triggered too early or too late due to noise on the flux estimate. This can be particularly problematic at high speeds, where each sample by which the trigger leads or lags the true zero-crossing can mean tens of degrees of motor commutation error. For this reason, it is useful to include a position estimate filter at high speeds.

Instead of treating each new zero-crossing position update with 100% confidence, as in the unfiltered estimator of Section 2.4.1, the new position update is merged with the extrapolated position using a tunable weighting factor. This method, shown in Figure 29, is the basis of the filtered position estimator implemented in this article.

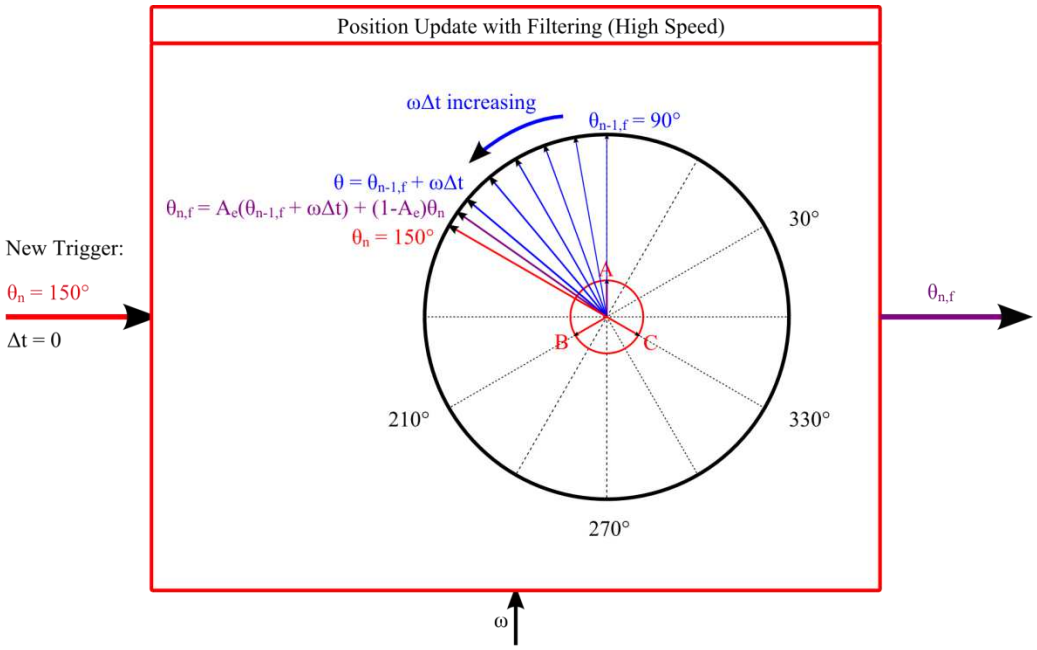


Figure 29: The filtered position estimator shown at the instant of a new flux zero-crossing trigger. The filtered estimator merges the new position from the zero-crossing detector with the extrapolated position, and restarts the extrapolation timer.

When a new zero-crossing trigger comes in, it is merged with the current position extrapolated since the previous position update. The filter constant, A_e, determines the weighting of the new vs. old information. Higher A_e favors the extrapolated position, providing heavier filtering. Lower Ae favors the new position information more. An A_e of zero is identical to the unfiltered position estimate of Section 2.4.1.

If the rotor speed is constant and the zero crossing noise is random, the filtering should not introduce lag to the position estimate. On average, the extrapolated position will be exactly the position indicated by the zero-crossing trigger. However, in the presence of rotor acceleration, the extra filtering comes at the

expense of increased phase lag of the position estimate when compared to the unfiltered version. It is important to quantify this additional phase lag to understand the limits of the extra position filtering.

The amount of position error introduced by the extra position filtering can be approximated using the following equation:

$$\theta_{ef} \approx \left(\frac{\alpha A}{\hat{\omega}^2}\right)\left(\frac{A_e}{1-A_e}\right)6NA, \qquad \frac{\alpha A}{\hat{\omega}^2} \ll \left(\frac{1-A_e}{A_e}\right), \qquad A = 60edeg$$

where α is the angular acceleration, $\hat{\omega}$ is the most recent speed estimate, and N is the number of electrical cycles over which the speed estimate is taken. The value A_e is the filter constant described above. The value A is the electrical angle over which extrapolation occurs, 60edeg in this case.

The error is approximately proportional to the product of a dimensionless acceleration and a term related to the filter time constant. Higher acceleration or a more filtering both cause more lag, but the effect is mitigated at high speeds. Since the filtering is most useful at higher speed anyway, it can be enabled only above a speed at which the extra error introduced due to acceleration is negligible. Or, the filter constant, A_e, can be made a function of speed to keep the angular error less than some maximum bound.

As one example, Figure 30 shows the effect of position filtering on a 14-pole motor undergoing 10,000rpm/s acceleration. The sampling frequency is 15.625kHz and the speed estimate interval is one electrical cycle. The scenario is identical to that of Figure 28, except that a position filter with A_e = 0.9 has been applied at each zero crossing position update.

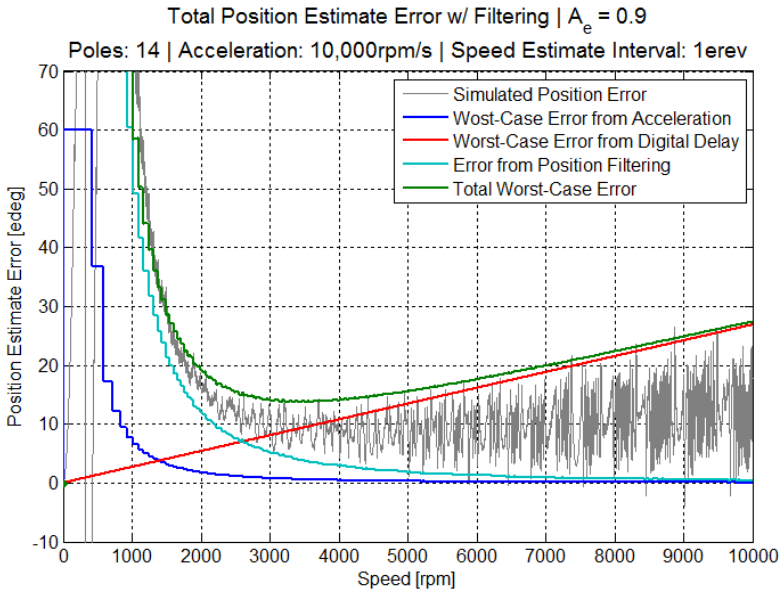


Figure 30: Simulated position estimate error for a 14-pole motor undergoing 10,000rpm/s acceleration with position filtering. The sampling frequency is 15.625kHz and the speed estimate interval is one electrical cycle.

With the position filter enabled, the minimum speed required to ensure no more than 20edeg of position estimate error under maximum acceleration is now 2,000rpm. In fact, the approximation used to calculate the position error due to filtering begins to break down below this speed, as evidenced by the simulated

position error exceeding the worst-case estimate at speeds less than 2,000rpm. However, the approximation presented above is accurate when the position error due to filtering is less than 20edeg, which contains the entire valid design space anyway.

The strategy might be to make the position filter constant a function of speed in order to bound the maximum position error at all speeds valid speeds for sensorless operation. For the sake of simplicity, it is also possible to use a fixed filter constant and only enable the filter at speeds above which its effect on acceleration-induced lag is negligible. Combining the information from Figure 28 and Figure 30, one might choose to operate without filtering from 500-4,000rpm and with filtering at $A_e = 0.9$ from 4,000-7,000rpm for the motor scenario presented.

2.5 Sensitivity and Limitations

2.5.1 Sensitivity to Parameter Variations

The sensorless position estimation routine discussed above relies on knowledge of motor parameters, specifically line-to-neutral resistance, R, and synchronous inductance, L_s (see Figure 17). These parameters are not perfectly known, so it is important to understand the sensitivity of the position estimate to variation in either parameter. Figure 31 shows the phasor representation of the position estimator, normalized by setting ω = 1rad/s.

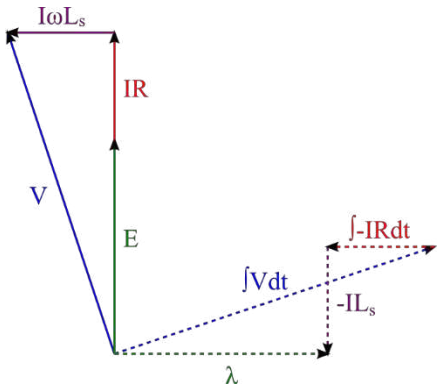


Figure 31: Phasor representation of the flux estimator, normalized by setting ω = 1rad/s.

The rotor flux linkage, λ, is estimated by integrating *V-IR* and then subtracting stator flux linkage IL_s. (Since ω = 1rad/s, integration of each phasor is just clockwise rotation of 90°.) If parameters R and L_s are accurate, the resulting rotor flux linkage will have the correct phase.

Figure 32 shows the same phasor diagram, but with small positive and negative deviations from the true motor parameters.

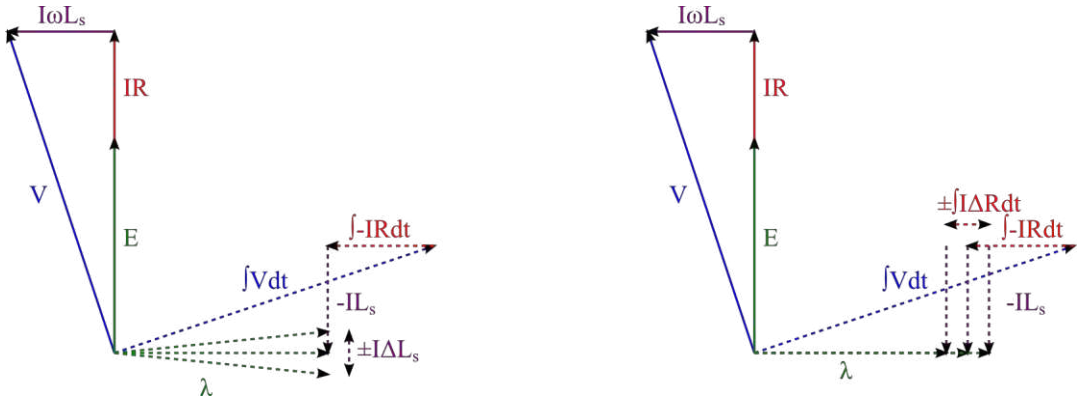


Figure 32: Small positive and negative deviations in motor inductance (left) and resistance (right) parameters.

Small deviations in resistance (ΔR) change the magnitude of the estimated flux vector, but have little effect on its phase, which is the important quantity for position estimation. Small deviations in synchronous inductance (ΔL_s) do affect the phase of the estimated flux vector. If the inductance parameter is too large, the estimated flux vector lags the true flux vector. If the inductance parameter is too small, the estimated flux vector leads the true flux vector.

The current controllers will drive current to be in phase with the estimated back-EMF vector (90edeg leading the estimated flux vector), so a more thorough depiction of the over- and under-estimated inductance cases must account for the action of the current controller causing the current vector to either lag or lead the back EMF, as in Figure 33.

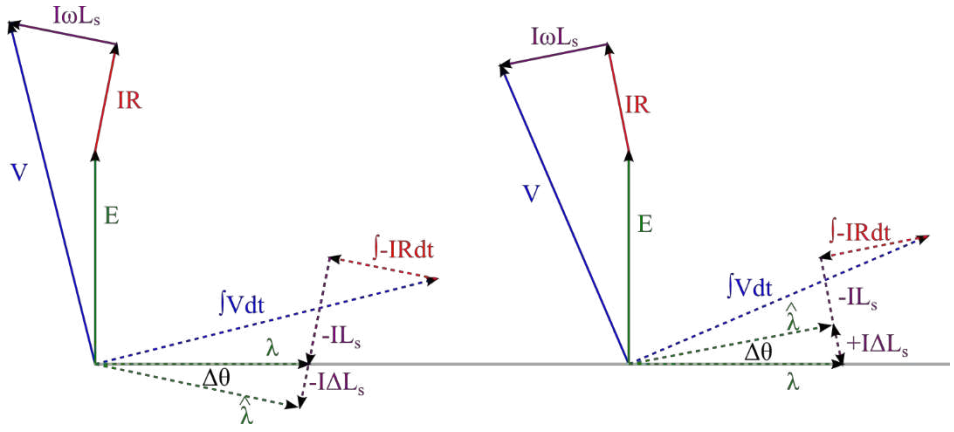


Figure 33: Over- or under-estimating the inductance parameter can cause the flux estimate to lag or lead the true flux.

In the case of an overestimated motor inductance (Figure 33, left), the estimated flux, $\hat{\lambda}$, *lags* the true flux, λ, by angle $\Delta\theta$. The current controller attempts to drive current at an angle 90edeg leading the estimated flux vector (in phase with the estimated back EMF). However, current *lags* the true back EMF vector by the same angle, $\Delta\theta$. In the case of an underestimated motor inductance (Figure 33, right), the estimated flux, $\hat{\lambda}$, *leads* the true flux, λ, by angle $\Delta\theta$. The current controller attempts to drive current at an angle 90edeg leading the estimated flux vector (in phase with the estimated back EMF). However, current *leads* the true back EMF vector by the same angle, $\Delta\theta$.

In both cases, the phase angle error can be approximated by:

$$\Delta\theta = \sin^{-1}\left(\frac{\mathrm{I}\Delta L_s}{\lambda}\right) \approx \frac{\mathrm{I}\Delta L_s}{\lambda}, \quad \frac{\mathrm{I}\Delta L_s}{\lambda} < 0.5$$

2.5.2 Summary of Frequency Limits

Several sources of rotor position estimate error that are frequency-dependent were discussed above. Table 1 summarizes these error sources and their effect on the rotor position estimate.

Table 1: Summary of Frequency-Dependent Position Error Sources

Error Source	**Direction**	**Amount**	**Discussed In**
Low Pass Filter (Pseudo-Integrator)	Lead	$90º - tan^{-1}(\tau_{int}\omega)$	Section 2.2.2
Hold-Off	Lag	$-T_h\omega$	Section 2.3.2
Linear Position Extrapolation	Lead or Lag (Depends on Acceleration)	$-\left(1 - \frac{1}{\sqrt{1+K}}\right)A,$ $K = \frac{(18N-1)\alpha A}{\hat{\omega}^2}$	Section 2.4.1
High-Speed Position Filter	Lead or Lag (Depends on Acceleration)	$-\left(\frac{\alpha A}{\hat{\omega}^2}\right)\left(\frac{A_e}{1-A_e}\right)6NA,$ $\frac{\alpha A}{\hat{\omega}^2} \ll \left(\frac{1-A_e}{A_e}\right)$	Section 2.4.2
Digital Delay	Lag	$-T_s\omega$	Section 2.4.1

Definition of Variables:

τ_{int}: First-Order Low-Pass Filter Time Constant [s]

ω: Electrical Angular Frequency [rad/s]

T_h: Hold-Off Time [s]

K: Dimensionless Acceleration [ ]

A: Electrical Angular Span of Interpolation [rad] (60º or π/3 as-implemented)

N: Number of Electrical Cycles per Speed Estimate

α: Electrical Angular Acceleration [rad/s^2]

$\hat{\omega}$: Electrical Angular Frequency Estimate [rad/s]

A_e: High-Speed Position Filter Constant (Extrapolation Confidence Factor)

T_s: Sampling Time [s]

The total frequency-dependent electrical angle estimation error is the sum of all these contributing factors. To ensure that the sensorless position estimate will be valid over the entire range of operating frequencies, the electrical angular error should be kept within a reasonable bound (suggested: ±20º). Conversely, the minimum and maximum operational frequencies for closed-loop control can be found by setting an upper bound on electrical angular error and solving implicitly for the range of valid angular frequencies, ω. To do this, an assumption must be made that $\hat{\omega} \approx \omega$, that is, the speed estimate is approximately equal to the true speed

The high-speed limit is dominated by digital delay, so increasing the sampling frequency is the most direct way to achieve higher maximum electrical frequencies. This is especially important for motors with high pole counts and high maximum speeds. If the sampling frequency and PWM frequency are both increased together, the added benefit of lower current ripple is also realized. However, higher switching losses will be incurred.

The low-speed limit has more contributing factors and is dependent on the direction of acceleration. The worst-case scenario is a short time-constant low-pass filter with heavy deceleration (both contribute significant phase lead). Though the lower speed limit is more complex, the possible ways to reduce the minimum operational speed are more numerous as well. Several are listed here, each with its own tradeoffs:

1. Lengthen the time constant of the low-pass filter, τ_{int}. This will increase flux estimate bias, as discussed in Section 2.2.2, so it must be combined with higher quality sensors and better calibration and biasing.
2. Use no hold-off on the flux zero-crossing detector. This means relying on other methods (hysteresis, commutation order, see Section 2.3.2) for preventing noise from false-triggering the zero-crossings.
3. Use a smaller number of electrical cycles per speed estimate. At very low speed, fractional cycles could be used, down to one speed estimate per sensor transition (1/6 cycle). The tradeoff is speed estimate accuracy at higher speeds, although a dynamic cycle count that is a function of speed would mitigate this tradeoff.
4. Limit rotor acceleration/deceleration at low-speeds. This can be achieved roughly by limiting positive and negative currents, voltage (PWM duty cycle) slew rates, or using an outer speed-control loop. There is an obvious tradeoff in motor performance and control bandwidth, but reducing the minimum operational speed would mean achieving closed-loop control earlier in startup.
5. Use no position filtering at low speed. The cutoff for high-speed filtering could be set such that the angular error contribution from the high-speed filter is negligible. The position estimate may be more sensitive to noise.

There are also fixed (not frequency-dependent) error sources, namely flux zero-crossing detector hysteresis (Section 2.3.2) and inductance parameter variation (Section 2.5.1). It's important to keep these in mind when setting an upper bound for electrical angular error.

With imperfect sensors and no knowledge of initial rotor position, the sensorless method presented in this article cannot operate at zero speed and in general will always have some minimum speed for closed-loop

field-oriented control. The exception may be traverses from forward to reverse where the flux estimate is able to track through the zero speed crossing. However, this exception is not considered this article. Section 3 covers one method of achieving open-loop startup from zero speed to the minimum operational closed-loop speed.

3 Start-up and Low-Speed Operation (Gen. 1)

3.1 Overview

In order to bring the motor from zero speed up to the minimum closed-loop speed as determined in Section 2.5.2, an open-loop start-up procedure is needed. This is in lieu of any methods for detecting rotor position at zero or very low speed, such as inductance measurement via test signal injection. Instead, the stator current is driven with no knowledge of the true rotor position but in a manner which, given assumptions of the load characteristics, will result in the desired speed ramp.

Although the manner of control is different, the philosophy of the open-loop startup is similar to that of a stepper motor. The rotor is forced to follow the stator voltage and current, if it can. If it can't, the motor will slip by one pole pair at a time, similar to missing steps on an overloaded stepper motor. This is, of course, undesirable and will usually result in a failed startup.

Figure 34 shows a general overview of the states and transitions involved in the open-loop startup routine. The states and transitions are managed by high-level software while the low-level motor control software operates differently depending on the state. The four states (Idle, Park, Ramp, and Run) are discussed in detail in Section 3.2.

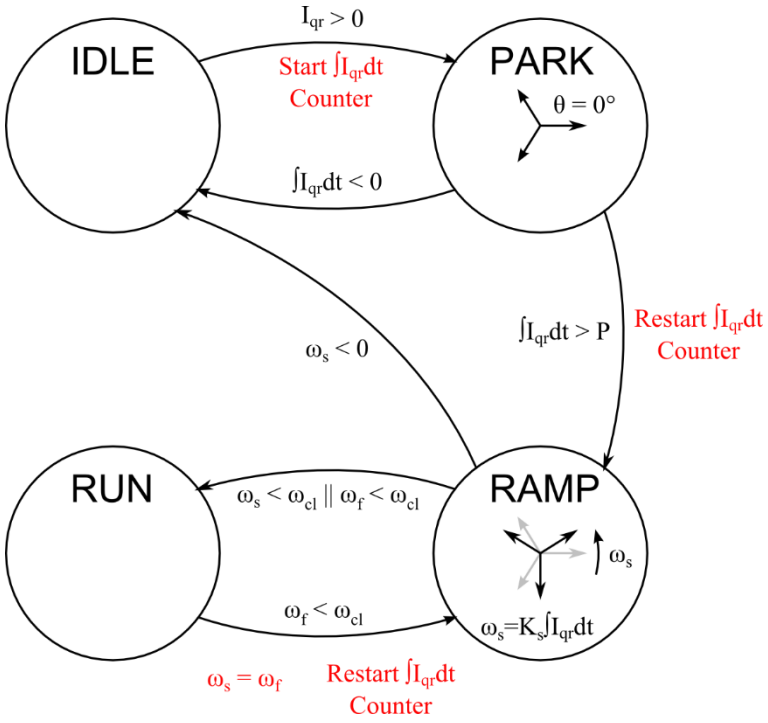


Figure 34: States and transitions involved in open-loop sensorless startup.

This method is limited to one direction of rotation. That's not to say the motor can't be reversed, but the process for reversing is not built-in to the state machine. Instead, the motor would first be brought to the Idle state, then the direction would be changed by software reconfiguration for reverse. (A simple way to do this would be to swap two output PWM channels.) Smooth transitions through reverse would require a continuous method.

Generally, the torque per amp in open-loop startup will be significantly lower than what would be available in closed-loop sensorless FOC. For that reason, suitable safety factors for both the current and the rate (parking time, ramp rate) should be factored in to ensure good startup, even in the presence of extra load such as higher than expected friction or inertia, or gravity during a hill start.

3.2 States and State Transitions

3.2.1 Idle

The Idle state is a zero-speed waiting state. The phase voltage magnitude is set to zero, but the gate drive remains enabled. It is important to reset all state variables and integrating counters in the idle state. This could include variables for voltage magnitude and phase, I_{qr} integral counters (discussed below), PI current control variables, speed estimates, angle estimates, and look-up table pointers.

To transition out of idle state, a positive command (current/torque command, I_{qr}, in this case) must be input.

3.2.2 Park

The Park state is used for initial rotor alignment, to attempt to bring the rotor to a fixed electrical angle (θ_p = 0º) before the start of the open-loop ramp. To do this, a controlled amount of current (set by the current reference input, I_{qr}) is driven in the stator winding at that electrical angle. In the absence of friction or any other constant loading, the rotor will align itself with the stator current, as in Figure 5.

The important parameters for the Park state are how much current to drive the stator with, and for how long. These depend on the type of load connected to the motor. At low speed, the load may be dominated by viscous friction, Coulomb friction, inertia, or gravity (hill start), or it may be a combination of several of these. Table 2 details how the different load types effect the minimum parking state current and time.

Table 2: Several different load types and their effects on rotor parking.

Dominant Load Type	Parking Constraint
Coulomb Friction	Characterized by a constant frictional torque opposing the motion of the rotor. The parking current (I_{qr}) needs to be able to overcome this torque in order to break past static friction. However, since electrical angle of current isn't known, the required current might be much larger than the normal value governed by the motor's torque constant with ideal current vector control. There will be an offset induced by the constant torque, preventing the motor from becoming exactly aligned.
Viscous Friction	Characterized by an overdamped mass-spring-damper response. The settling time depends on the inertia, damping, and motor current (which sets the effective spring constant for small angles). Increasing motor current should decrease settling time *until the response becomes underdamped*

	(inertia dominated). The time integral of motor current provides a good way to account for both the current and the time.
Inertia	Characterized by an underdamped mass-spring-damper response. The settling time depends only on the mass and damping coefficient. Increasing the effective spring constant by increasing motor current *will not* further reduce settling time. *This type of load defines the minimum settling time in most cases*.
Gravity (Hill Start)	Characterized by a constant torque related to mass and incline angle. Similar to Coulomb friction, the parking current must be significantly larger than would be required to overcome the constant torque under optimal current vector control. (Because the electrical angle of the rotor is not known.) As with Coulomb friction, there will be an offset induced by the constant torque, preventing the motor from becoming exactly aligned.

As a way to ensure that both minimum current and minimum settling time constraints are met, a time integral of parking current is used to time the parking state. Note that the parking current is considered to be on the q-axis and designated I_{qr} (reference q-axis current). This is done to allow normal operation of the q-axis control loop of the modified synchronous current regulator, which controls voltage magnitude (see Figure 11). The d-axis control loop is disabled and the voltage phase is fixed to a parking value (0º, in this case). In reality, the location of current in the d-q frame of reference is not yet known. As the rotor aligns, it is more likely to be near the d-axis, as in Figure 5.

The parking current may be constant, or it may be set by an external current command. In either case, the integral of current is tallied in the parking state until a specific threshold (in units of A-s, amp-seconds) is met. This threshold can be calculated if the load is known, or it may be determined experimentally by observing the time it takes for the rotor to align and settle at various parking currents. When the A-s threshold is met, the startup routine moves to the Ramp state. If the integral counter goes negative (for example due to a braking input of negative I_{qr}), the motor reverts to Idle state.

3.2.3 Ramp

The goal of the Ramp state is to bring the motor from zero speed up to a speed high enough for sensorless position estimation to take over. This means the motor must produce net positive torque for a period of time without knowing the rotor position. It does this by driving current at an electrical angle that is incremented gradually, with the assumption that the rotor will continue to align with the stator current as the current is rotated. This is similar in principle to the operation of a stepper motor.

The ramping current may be constant, or it may be set by an external current command. . Note that the ramping current is considered to be on the q-axis and designated I_{qr} (reference q-axis current). This is done to allow normal operation of the q-axis control loop of the modified synchronous current regulator, which controls voltage magnitude (see Figure 11). The d-axis control loop is disabled. Instead, the voltage phase is incremented in an open-loop fashion at a speed (ω_s) which increases as the Ramp state progresses.

The rate at which the voltage phase is increased is a critical parameter of the Ramp state. It is limited by the load, which may be one or more of the load types discussed in Section 3.2.2 With a very slow ramp rate relative to the inertial and damping constraints of the load, each small increment in voltage phase is like a parking transient as the rotor settles to near the zero-torque position each time (Figure 35a). This will eventually get the motor up to closed-loop speeds, but is wasteful and slow. If the ramp rate is too high, the current vector will pass the q-axis and torque will drop, resulting in loss of synchronicity, a "slipped pole" (Figure 35b). This will likely cause the startup routine to fail.

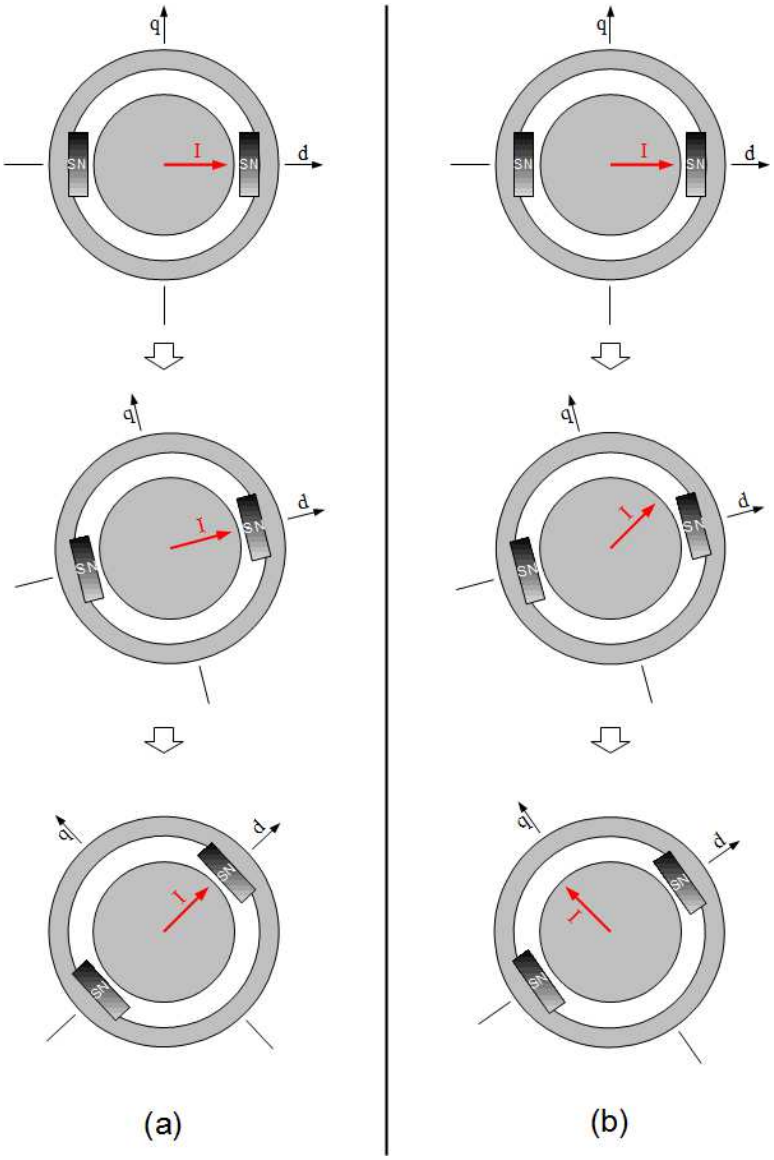


Figure 35: A ramp rate that is too slow (a) is wasteful. One that is too fast (b) will overtake the q-axis and cause a loss of synchronicity (slipped pole).

Choosing a ramp rate which allows the current vector to remain somewhere between the d-axis and q-axis for the entire Ramp state is ideal. The closer the current vector is to the q-axis, the closer the open-loop performance will be to ideal closed-loop torque. But the risk of slipping a pole is also higher. A good safety margin (2-3) should be allowed between the expected load torque and the ideal torque if all the

current were on the q-axis. This corresponds to a current vector angle approximately 20º-30º ahead of the d-axis (Figure 36).

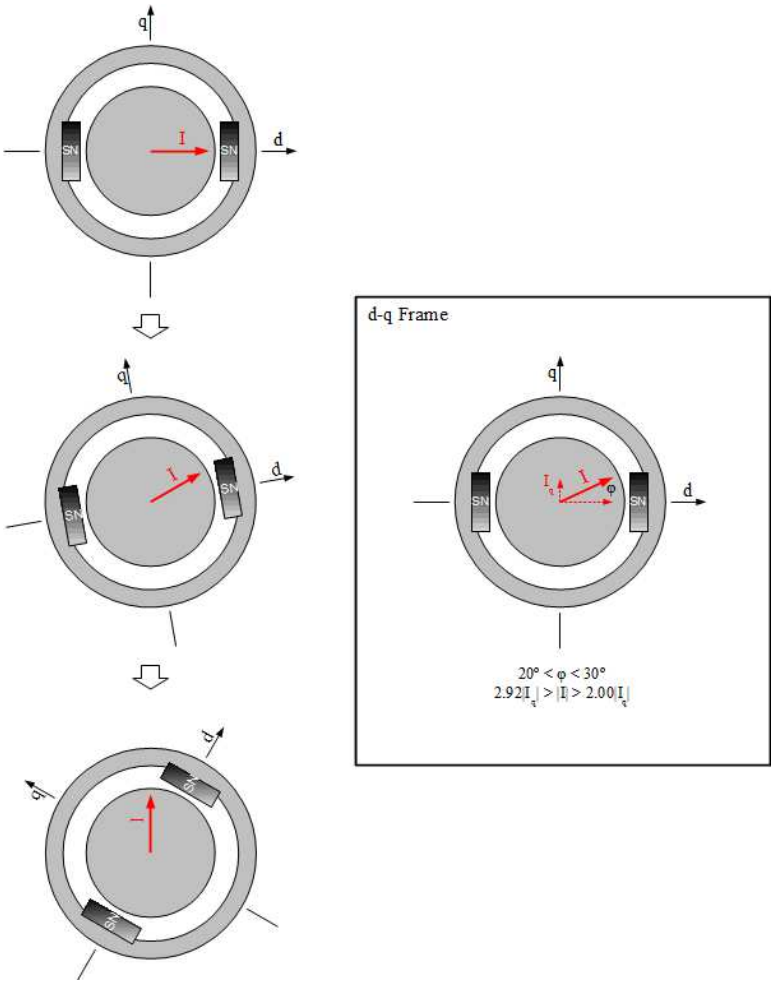


Figure 36: A good ramp rate allows the current vector to lead the d-axis by 20º-30º, producing about 1/3 to 1/2 as much torque as it would if placed on the q-axis.

Since the current vector is still predominantly on the d-axis, the startup is not as efficient as normal closed-loop operation; more current is required to produce the same torque. However, by leaving a margin of 2-3, the risk of losing synchronicity during transients or with higher-than-expected loads is minimized. The exact margin of safety depends on the variability and type of load. Lightly-damped loads may require more margin to account for overshoot and oscillation during the ramp.

If the load is dominated by inertia during the Ramp state, the load torque is proportional to the acceleration, which is defined by the ramp rate:

$$\tau_L = J\alpha = J\frac{\ddot{\theta}_e}{p}$$

where J is the system inertia, reflected to the motor's rotational axis, θ_e is the electrical angle (which changes at the same rate as the voltage phase angle, in a steady-state ramping condition), and p is the number of pole pairs. The torque produced by the motor with ideal current vector placement would be

$$\tau_m = K_t I_{qr}$$

Allowing for a factor of three reduction in torque during open-loop operation, this imposes a constraint on the ramp rate as a function of ramping current:

$$K_t I_{qr} \geq 3J \frac{\ddot{\theta}_e}{p}$$

$$\ddot{\theta}_e \leq \left(\frac{1}{3}\right)\left(\frac{pK_t I_{qr}}{J}\right)$$

Other types of load may impose other constraints on ramp rate, but the following method assumes that achievable ramp rate is proportional to ramping current. A time integral of this current can be used to set the angular rate:

$$\dot{\theta}_e = \omega_s = K_s \int I_{qr} dt$$

where K_s is a single parameter which sets the ramp rate proportional to current. K_s can be calculated using the equation above if the load is predominantly inertial and the motor torque constant is known. It can also be set experimentally to a value which achieves the fast but reliable startup under worst-case load conditions.

Because the current vector is not aligned with the q-axis, startup torque will be limited compared to the motor's normal closed-loop torque. One way to offset this limitation is to allow more current during the startup routine than during normal operation. Since the startup is short, it should not cause any thermal problems for the motor. However, the controller (transistors, current sense, etc.) must be able to tolerate this short period of higher currents.

The transition from Ramp state to Run state occurs when the motor speed exceeds a speed set as the minimum for closed-loop operation (ω_{cl}). This threshold can be assessed on the speed set by the current integral counter (ω_s), or the speed derived by the flux estimator (ω_f), or both. If the current integral becomes negative (for example due to a braking command), the motor can revert back to Idle state.

3.2.4 Run

The Run state is identical to the sensorless position-estimated closed-loop control as discussed in Section 2. Once this state has been reached, the startup routine is finished. If the motor slows down past the minimum closed-loop speed (ω_{cl}), it can be carefully transitioned back to the Ramp state if the last measured speed is used as the starting speed for the ramp down. In practice, it might be more desirable to return to Idle state and/or float the phase outputs to allow a smooth coast down.

Some hysteresis should be used to allow operation at lower measured speeds once the motor has transitioned into the Run state; using a lower speed than ω_{cl} for the transition back out of Run state. This prevents the startup from rapidly jumping between Ramp and Run state. It also allows for lower-speed

operation once the rotor position estimate converges, without having to use a marginal closed-loop transition speed that may result in failed startup.

4 Case Studies / Experimental Validation

The sensorless field-oriented control method presented in this article has been developed and tested on a variety of motors, ranging from a relatively low speed but high-power in-wheel scooter motor to high-speed, high-bandwidth motors used on multirotors. The hardware has evolved with the control method (to meet many of the requirements discussed in Seciton 1.3. Three examples of hardware are available as reference designs [9]. Two software configurations, one targeting traction applications ("ground") and one targeting multirotor motor applications ("air") are also available as reference.

This section discusses the development and testing of the hardware and software used to implement sensorless field-oriented control using two case studies.

4.1 Pneu Scooter / Custom Hub Motor

Pneu Scooter is a compact and lightweight electric kick scooter with a rear wheel hub motor. The scooter frame is an aluminum C-channel, mostly filled with batteries but with space at the rear for a motor controller. The motor is a custom-built PMSM that directly drives the rear wheel with a peak power of approximately 700W. Figure 37 and Figure 38 show the mechanical layout of the scooter, motor controller, and rear wheel motor.

Figure 37: Pneu Scooter, a lightweight electric kick-scooter with a custom in-wheel motor.

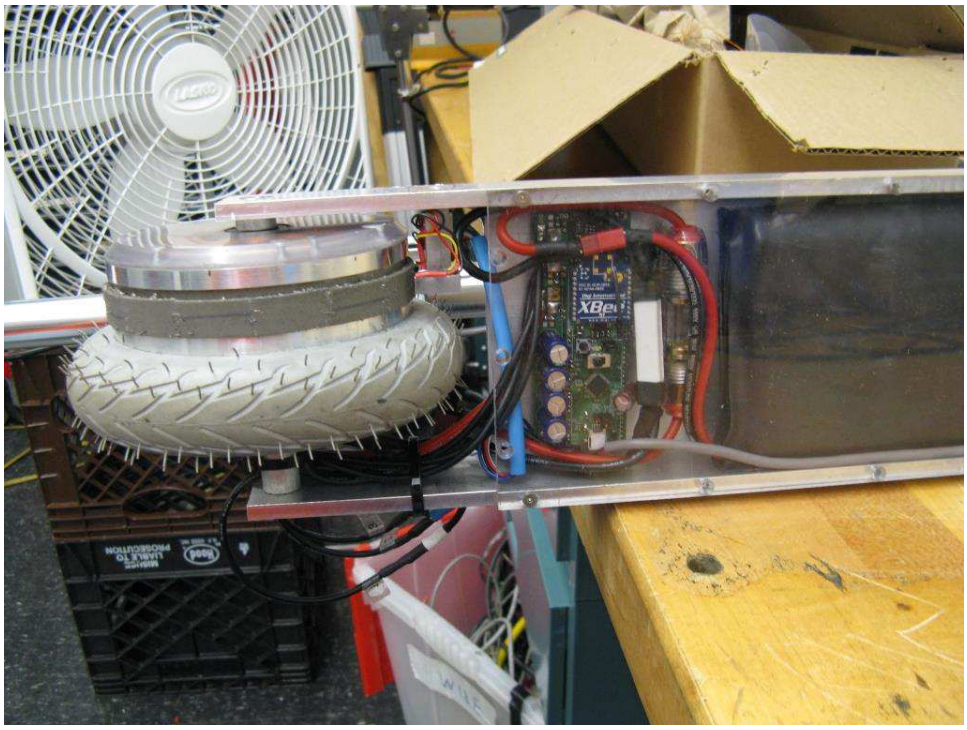


Figure 38: The rear wheel, motor, and motor drive of Pneu Scooter.

Originally a sensored system, Pneu Scooter was converted to sensorless field-oriented control using the 3ph v3.1 hardware [9]. This motor drive uses the IXYS GWM100-01X1 three-phase MOSFET module, which allows clean heat sinking to the aluminum scooter deck. It uses Hall effect current sensors on two of the three phases. A modular microcontroller board based on the STM32F103 control the power stage. The schematic and board layout of the 3ph v3.1 controller power stage are in Appendix B. The top and bottom of the controller are shown in Figure 39 and Figure 40, respectively.

The vector current controller implemented on the scooter is exactly that of Figure 11, with the torque-producing current I_q set by a hand throttle. The rotor electrical angle is obtained using the sensorless position estimation routine discussed in this article, implemented on the STM32F103 microcontroller. Because the STM32F103 does not have hardware floating-point capability, the math used in the flux observer is implemented in fixed-point at 15.625kHz (once per PWM cycle).

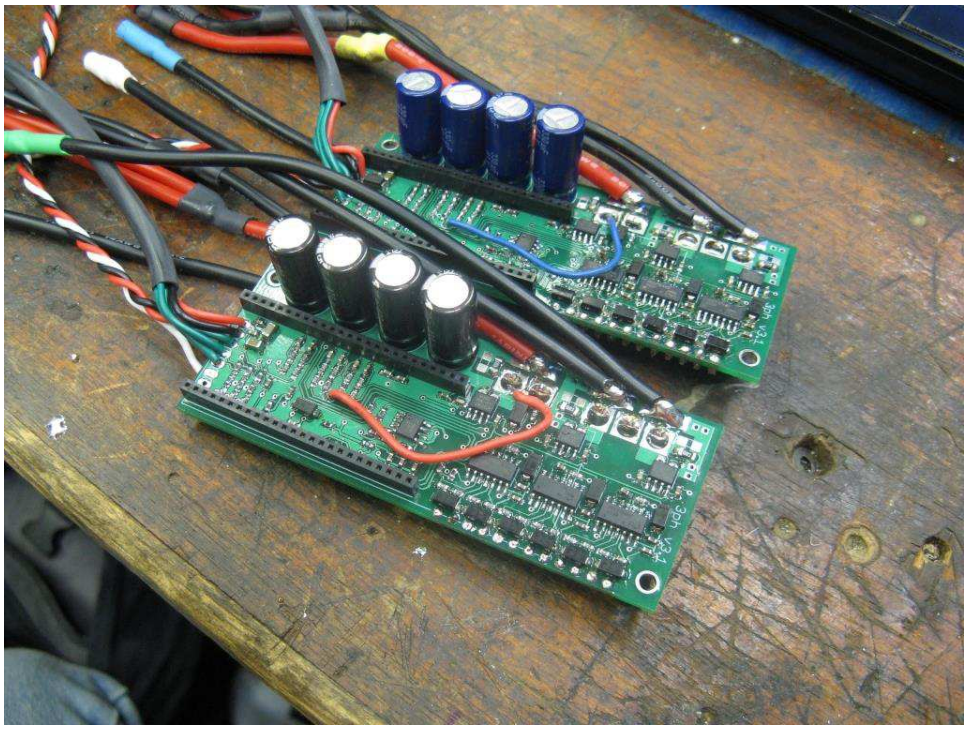


Figure 39: The top of two 3ph v3.1 controllers, showing DC bus capacitors and gate drivers.

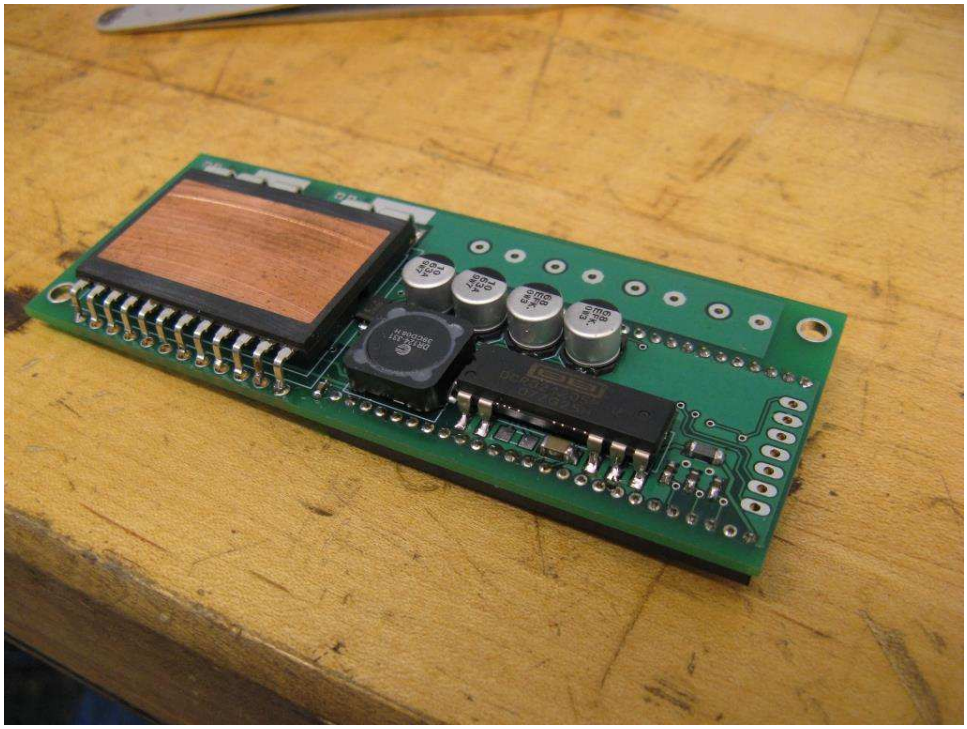


Figure 40: The bottom of a 3ph v3.1 controller, showing the MOSFET module and DC/DC converter for powering gate drive and logic.

Since the motor also had Hall effect sensors for detecting rotor position, this platform provided a useful opportunity for validating the sensorless routine against a known position estimate. Figure 41 shows some

early testing of convergence between the sensored and sensorless speed estimates. Convergence down to very low speeds (50-100rpm/1-2mph for a motor with a top speed of 1200rpm/21mph) was possible. Additionally, regulation of both d-axis and q-axis current is demonstrated. Because the rider can push off to the minimum closed-loop operating speed in one kick, no startup ramp is necessary for the scooter.

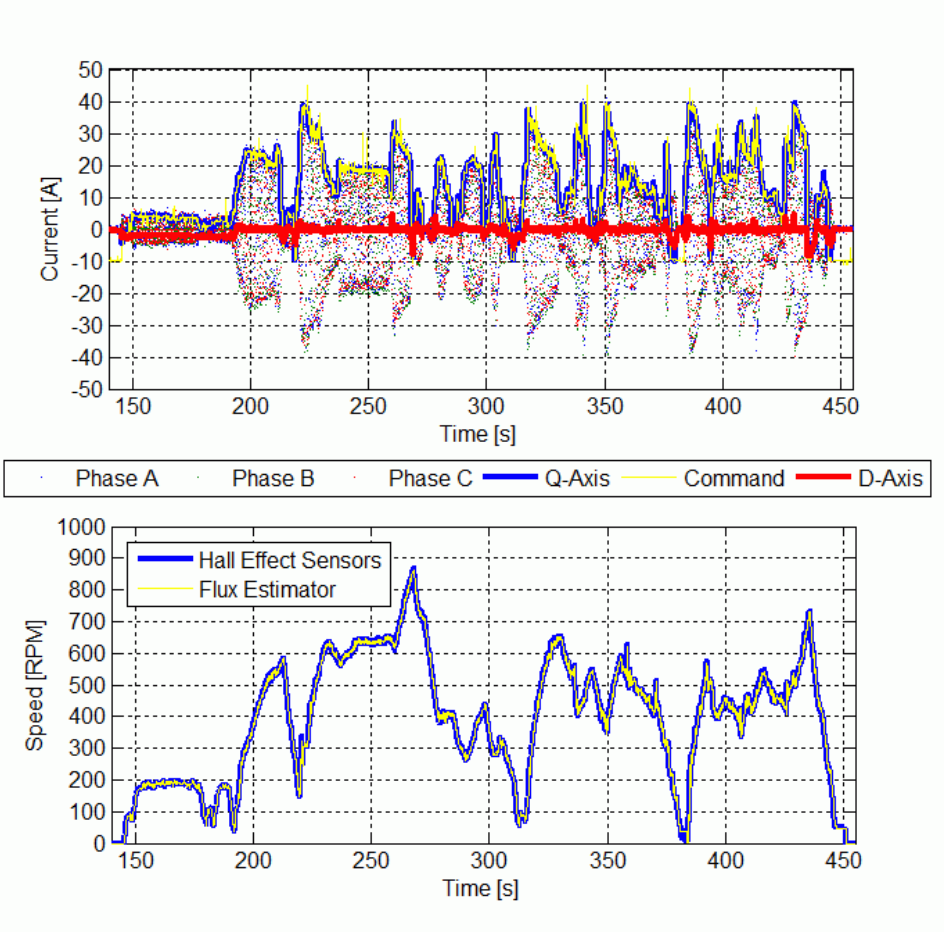


Figure 41: Test driving the sensorless FOC Gen. 1 method on Pneu Scooter.

The offset of the flux observer rotor angle estimate at low speeds can also be seen compared to the reference Hall effect sensor rotor position measurement, as shown in Figure 42. As expected, the position estimate from the flux observer leads the actual position due to high-pass filtering of the V-IR interal (see Section 2.2.2). Figure 42 shows that the amount of phase lead is predictable based on the filter time constant. There is a bulk shift of about 15º electrical between the Hall effect sensor position the flux observer position estimate, but this could also be attributable to the timing of the Hall effect sensors themselves.

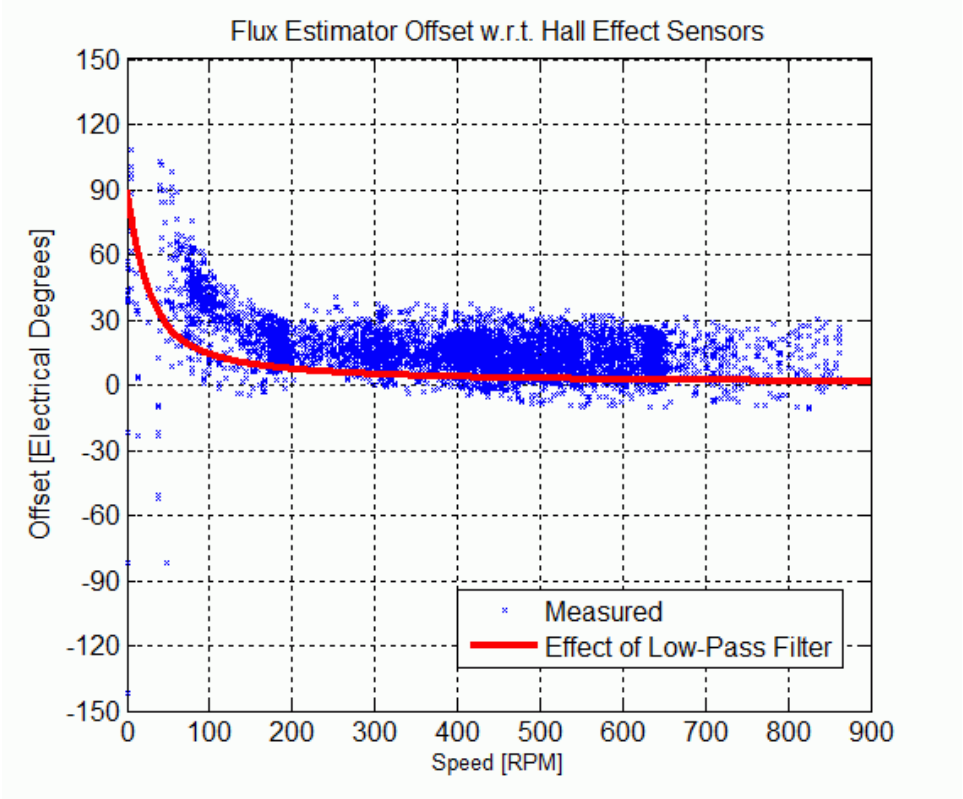


Figure 42: Flux estimator offset at low speeds due to filtering.

Although an individual electrical cycle is too fast to see using the scooter's telemetry and data logging, accurate snapshots of the estimated rotor position, observed flux, and measure motor currents can be used to fill in a full picture of a flux waveform over time, as in Figure 43. Although not a perfectly clean signal, the sinusoidal phase-A flux linkage is captured in relation to the motor currents. As expected, the phase-A current leads the phase-A flux linkage by 90º, for optimum torque production. The magnitude of the flux waveform can be compared to prediction based on the motor's torque constant. Because the observer is estimating rotor flux linkage, it should not depend on motor speed or stator currents, only on the torque constant of the motor.

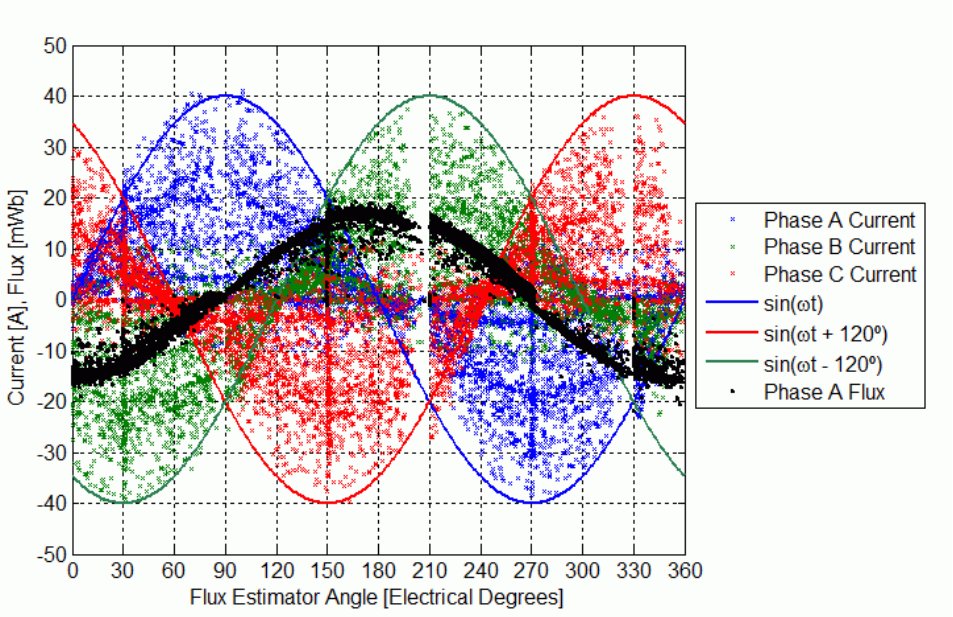


Figure 43: Scooter motor flux and current waveforms, plotted against rotor electrical angle.

A ramping startup, as in Section 3, was implemented on Pneu Scooter to test starting from zero speed. Traction applications require high torque even at low speeds, so in order to satisfy the ramping constraints of 3.2.3, the ramp rate had to be particularly slow. In practice, it was simpler and faster for the rider to provide a push start up to minimum closed-loop speed. This was similar enough to the normal motion of starting off that the actual operation of the scooter was almost indistinguishable from the sensored version. Pneu Scooter is still regularly used and has utilized sensorless field-oriented control for more than a year.

Software for the 3ph v3.1 as implemented on the scooter is not provided here, since it is somewhat out of date. However, the "ground" software [9] for the FFv1.1 controller hardware (discussed in Section 4.2) was also tested on the scooter with good results. This software represents the most up-to-date traction motor implementation of this sensorless field-oriented control method, and its development was heavily based on the work done first with Pneu Scooter and the 3ph v3.1 controller.

4.2 Quadrotor / BLDC Motor

Further work on this sensorless field-oriented control method was done with the intention of implementing it on remote-controlled multirotor aircraft. The majority of RC multirotors use inexpensive BLDC motors to directly spin propellers. These motors are typically 14-pole surface permanent magnet motors with outside rotors ("outrunners"). Although the back EMF shape varies, it is often considered to be trapezoidal and the motors are mostly paired with six-step square wave drive sensorless controllers. While this work on sensorless field-oriented control was being done, and even at the time of writing, almost all commercial RC multirotors use this combination of low-cost hardware and BLDC control.

Even though the motors are considered "brushless DC" and tend to have a more trapezoidal back-EMF waveform, there can still be efficiency and performance advantages to using sinusoidal commutation and field-oriented control:

- Lower current ripple and harmonic content with sinusoidal commutation can reduce I^2R losses compared to six-step square wave drive.

- Excitement of mechanical resonances in the rotor can at 6x the electrical frequency are eliminated, which can reduce audible noise and high-frequency vibration of the rotor.

- While most sensorless BLDC controllers offer programmable "timing" to adjust for inductive lag as in Figure 8, the timing is fixed during operation. Field-oriented current control allows for dynamic and optimized placement of the current vector even as the speed and load change.

- Low-speed performance of the flux observer should be an improvement over back-EMF sensing, allowing lower idle speeds and less chance of losing sync if propeller speed is forced lower by a transient load. Startup is also less of a challenge since the propeller torque increases with speed squared.

Despite these potential benefits, there are both practical and theoretical issues making implementation of sensorless field-oriented control a challenge for multirotor motors and other high-speed BLDC-replacement applications:

- Typical rotor speeds can be up to 10,000rpm, or even higher for small motors and propellers. This is an electrical frequency of 1,167Hz for a 14-pole motor. Digital delay begins to play a large role at these speeds (Figure 28). To counteract this, a higher PWM frequency and current sensor sampling frequency might be needed. This requires higher bandwidth current sensors, faster processing of the fast loop, and more switching losses in the power stage.

- Motor control hardware for RC multirotors is limited in size, cost, and heat sinking compared to traction applications. The cost and size restriction means more integration of signal and power circuitry, which can lead to higher dV/dt and dI/dt noise coupling. It also means planning for higher thermal resistance from the power stage transistors to ambient air temperature, since the weight of large heat sinking isn't practical for flying.

- Multirotor motor controllers are typically "speed" controllers. (They control PWM duty cycle, which sets the average voltage. Ignoring the voltage drop across the motor's resistance and assuming proper timing, this correlates to the back EMF, which is a function of rotor speed.) Field-oriented control is fundamentally a current control strategy. A speed control outer loop must be applied to make it compatible with standard flight controllers.

- In order to facilitate stable and responsive flight control, the input-output bandwidth of the multirotor motor controller typically must be a lot higher than necessary for a traction application (10s of Hz instead of single-digit Hz, roughly). Additionally, this is the bandwidth from commanded speed to actual speed, not commanded torque to actual torque as would be the case in a traction application. So, the outer-loop speed controller and inner-loop field-oriented current controller must still allow high slew rates, and the sensorless position estimation algorithm must be able to tolerate much higher rotor acceleration than would be seen in a traction application.

With these challenges in mind, a new set of hardware was developed with which to implement sensorless field-oriented control for multirotor motor drives and other high-speed / high-bandwidth applications. Two hardware configurations were developed: the FF v1.1, a larger design with a target power of ~1kW and the FFv1.2s, a smaller design with a target power of about 250W. Figure 44 and Figure 45 show the two motor controller designs side-by-side. The schematic and PCB files are available for reference [9].

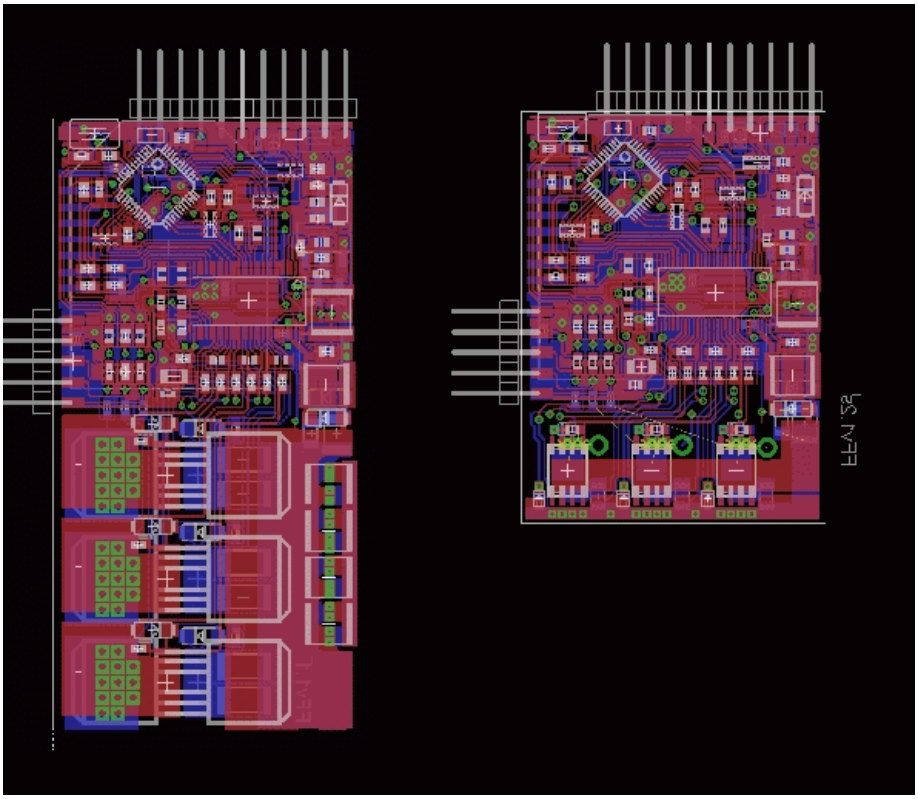


Figure 44: The FFv1.1 (left) and FFv1.2s (right) motor controllers used to implement high-speed and high-bandwidth sensorless field-oriented control.

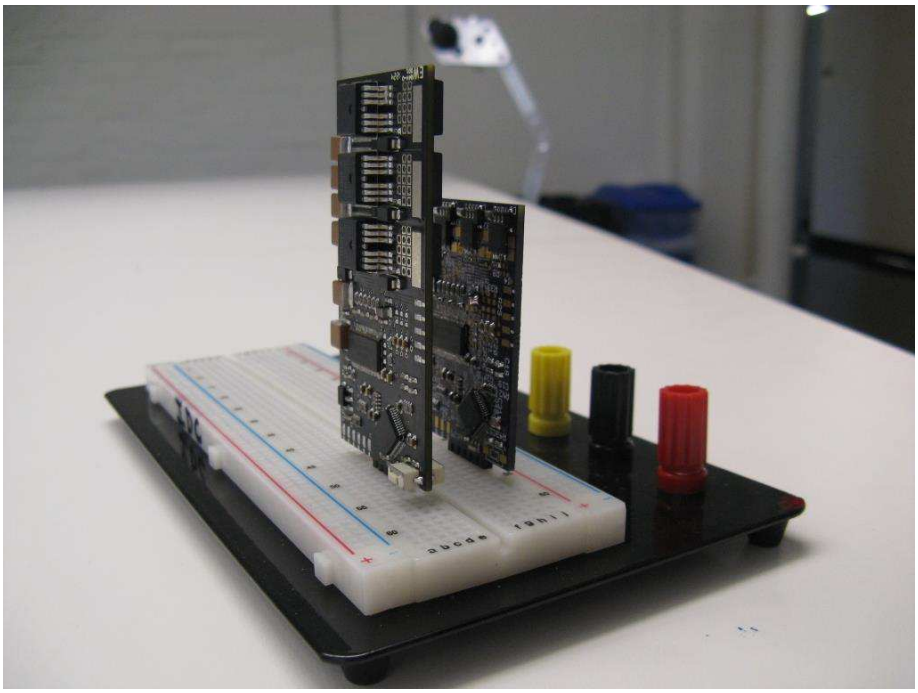


Figure 45: The FFv1.1 (left) and FFv1.2s (right) hardware as assembled.

In order to reduce the cost and size of the controllers enough to make them practical for multirotor use, a single-board design was implemented (rather than the split logic/signal design of the 3ph v3.1 hardware). The signal portion of each board is identical, containing an STM32F103 microcontroller and a TI DRV8301 integrated three-phase gate driver / current sensor / DC/DC converter IC. The consolidation of gate driver, DC/DC converter, and current sensor made possible by the DRV8301 also reduced size and cost significantly.

The power stage implemented differs for the FFv1.1 and the FFv1.2s. The FFv1.1 uses large D2Pak-7 MOSFETs, while the FFv1.2s uses small but high-current Power SO8 package MOSFETs. Many FETs are available in both these packages, allowing for a range of different voltage and current ratings for each controller. The maximum voltage is also limited by the choice of DC bus capacitor and ultimately by the DRV8301's maximum input voltage of 60V (48V is a more conservative working voltage limit).

Current sensing is done using two low-side current sense resistors, amplified by the DRV8301's internal op-amp circuit. This saves the cost and space of Hall effect current sensors. However, it requires synchronous current sampling and limits the sampling frequency to once per PWM cycle. The schematics and board layouts of both the FFv1.1 and FFv1.2s controller are in Appendix B

The larger of the two boards, the FFv1.1, has been tested in traction applications on Pneu Scooter as well as another electric scooter and a small 4WD electric go-kart with hub motors (Figure 46) [7]. The traction implementation is similar to that of Pneu Scooter and the 3ph v3.1 controller, where the torque-producing current I_q is directly controlled by the rider. In the case of the 4WD go-kart, synchronizing the start-up routine of all four motors proved to be the biggest challenge and wasn't adequately implemented. However, for single-motor traction applications the software as-is works well and is provided as a reference [9]. Because of the higher power requirement for traction applications, the "ground" software is mostly intended for the larger FFv1.1 hardware.

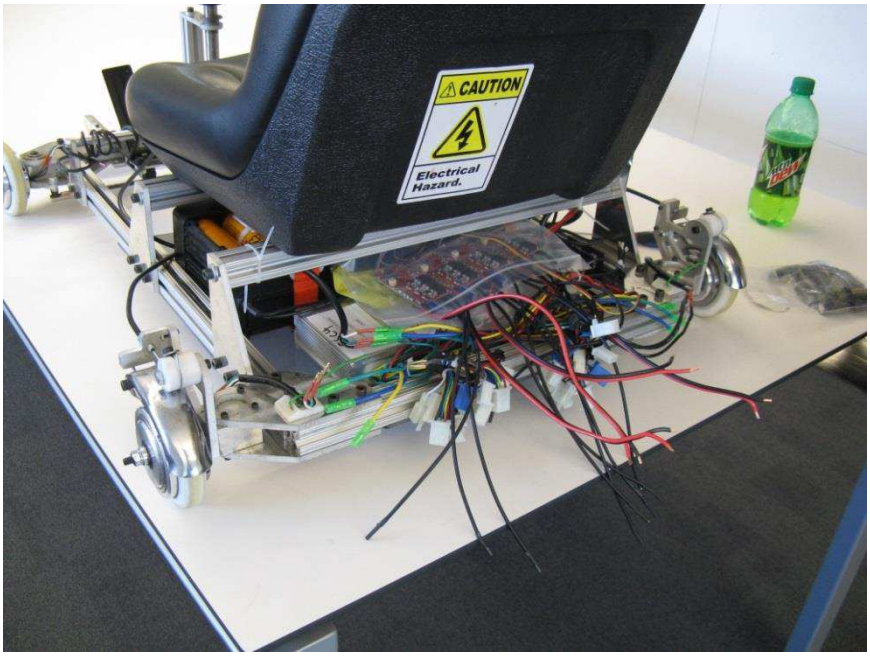


Figure 46: A 4WD electric go-kart with independent wheel motors on which FFv1.1 ground firmware was tested [7].

The sensorless field-oriented control method discussed in this article was augmented with an outer-loop speed controller for use on multirotor applications. The speed controller is a simple proportional+integral (PI) controller that acts on the error between the commanded RPM (from a flight controller) and the measured RPM (from the sensorless position estimator). However, the bandwidth limitations of the speed measurement and current control make it necessary to implement feed-forward channels to achieve the necessary speed control bandwidth for stable and responsive flight control. Figure 47 shows the control structure and feedforward channels used in the "air" software for the FFv1.1 and FFv1.2s controllers.

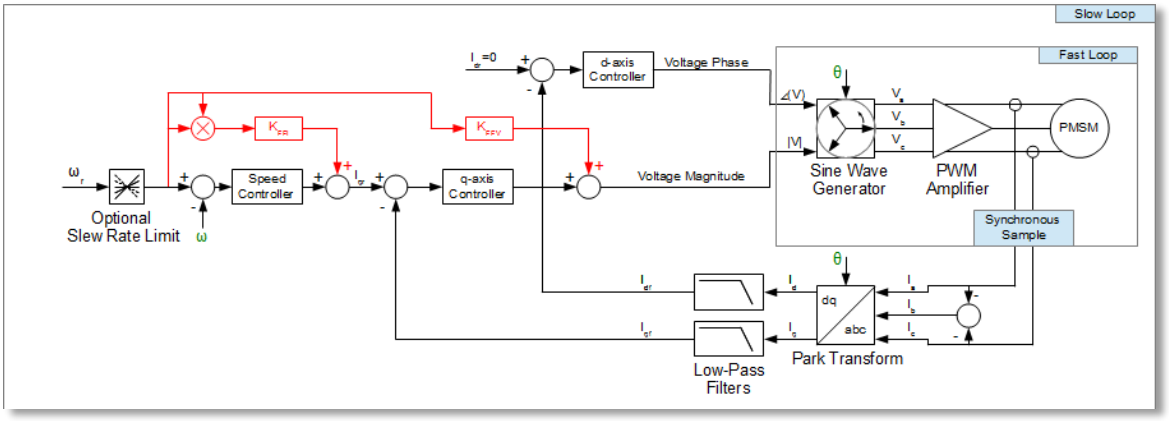


Figure 47: The modified synchronous current regulator, augmented with an outer-loop speed controller and feedforward current and voltage channels.

The inner loop remains the same as in Figure 11 (except that rotor electrical angle, θ, is provided by the sensorless position estimate rather than a shaft sensor). The outer loop speed controller acts on the difference between measured speed, ω, from the sensorless estimate, and commanded speed, ω_r, from the input. Because of the bandwidth limitation which may be imposed by the d-axis and q-axis current low-pass filters and the sensorless speed estimate, two feedforward channels are included (shown in red):

The current feedforward path acts on commanded speed squared, with the assumption that the steady-state q-axis current (torque) will be proportional to speed squared for a motor driving a propeller. The constant of proportionality, K_{FFI}, depends on the motor and propeller together, and should be adjusted experimentally to match the measured current vs. speed curve (so that in steady state, the PI speed controller output is close to zero). However, small errors in K_{FFI} can be accommodated by the PI controller.

The voltage feedforward path acts on commanded speed directly, and adds to the voltage magnitude output of the q-axis current controller, with the assumption that the driving voltage magnitude will be roughly proportional to motor speed. The constant of proportionality, K_{FFV}, depends only on the motor's torque constant and can be set based on that and/or experimental speed vs. voltage data. *Most of the high-frequency control bandwidth comes from this feedforward channel*, since it bypasses the low-pass filtered current sensors. Consequently, *this channel also allows uncontrolled transient currents outside of normal bounds set by the q-axis controller.*

To help limit the overcurrent transients that can be produced by the direct voltage feedforward channel, an optional slew rate limit can be implemented on the speed control input. The slew rate limit, combined with the inertia of the rotor and propeller, determine a maximum transient torque and current that can be

seen under normal operation. (Note that this is *in addition* to the steady-state torque to drive the propeller.) The maximum current is only guaranteed by design – there is no closed-loop measurement and limiting q-axis current at high frequency. So, the slew rate limit should be set conservatively so that the hardware can measure and tolerate transient currents at least 2-3x higher than that predicted. The slew rate limit can and should also be chosen to limit the rotor acceleration to values that the position estimator can tolerate. These limits are detailed in Section 2.4.

The FFv1.2s hardware is more suitable for use on small and medium-sized RC multirotors, since it is compact and lightweight. The Power SO8 MOSFETs also tend to have lower gate charge, which allows for the higher PWM frequency necessary for high-speed sensorless control. Figure 48 shows a FFv1.2s board wired up for multirotor use, with an external DC bus capacitor added for additional input voltage buffering. Figure 49 shows a multirotor with 10” propeller set up with four FFv1.2s motor controllers driving off-the-shelf BLDC motors.

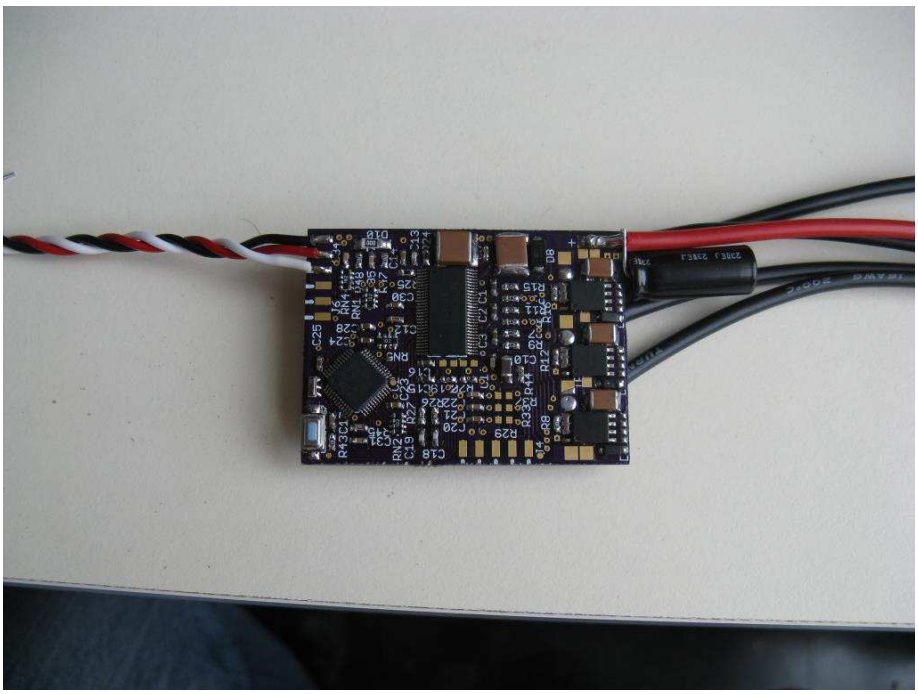


Figure 48: An FFv1.2s board wired up for use on a multirotor.

Figure 49: Four FFv1.2s boards on a multirotor with 10” propellers.

The flight controller used was an off-the-shelf RC multirotor controller that puts out commands at 450Hz in the form of standard RC servo pulse widths. These commands are interpreted by the motor controller as the reference speed and send to the controller of Figure 47. Accurate and high-bandwidth speed control was achieved, as in telemetry data from one of the multirotor’s motor drive shown in Figure 50.

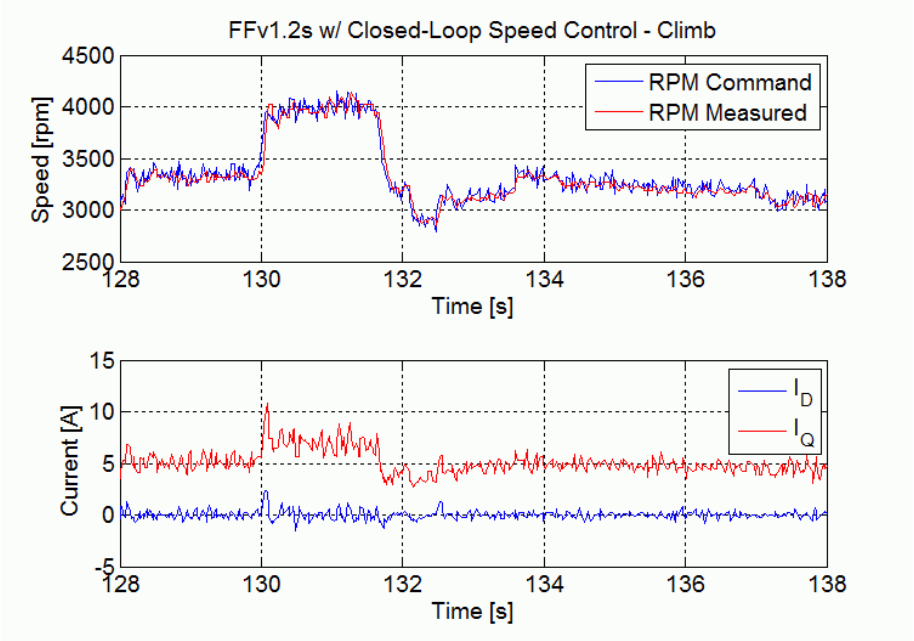


Figure 50: Closed-loop speed control on FFv1.2s multirotor motor drive.

Although the current measurement is noisy, it is still possible to see the transient current that peaks above 10A during the start of a climb as the rotor is accelerating. This transient and the high-frequency bandwidth it achieves are due mostly to the voltage feedforward path. The peak reached depends on the speed slew rate limit. d-axis current is regulated to zero as usual, except during short transients where the voltage feedforward channel may cause offset in the phase of the motor current.

Performance of the flux observer can be verified by overlaying snapshots of the flux estimate, phase currents, and rotor electrical angle estimate over time, as in Figure 51. This data is taken from a larger multirotor using FFv1.1 hardware. A significant amount of bias in the phase current estimates can be seen, and the result is a flux waveform that appears to be shifted upward. This would produce an offset but still useable position estimate for field-oriented control. Better zeroing of the current sensors and management of deadtime in the voltage PWM generation would likely improve the performance.

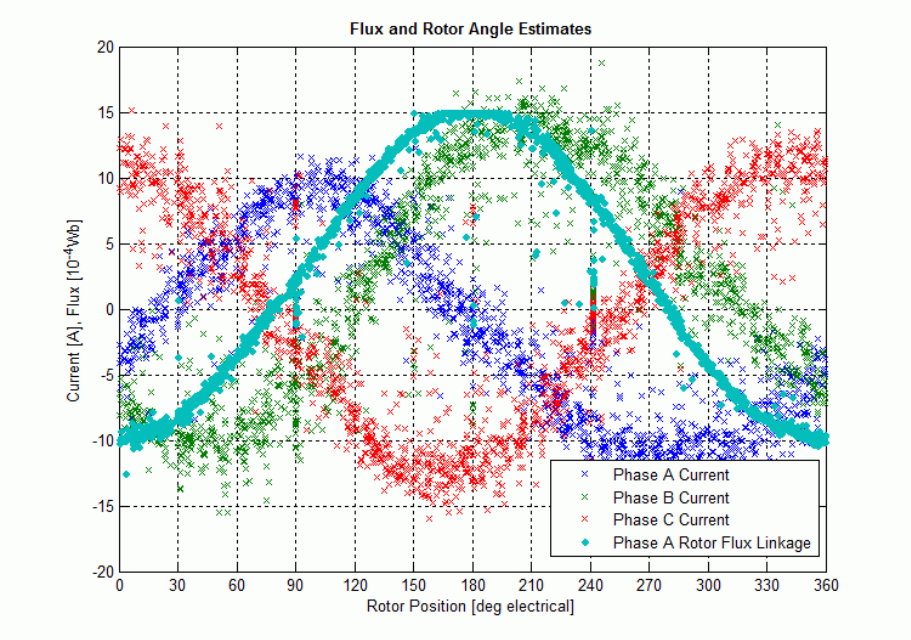


Figure 51: Flux estimate and phase currents plotted against the rotor position estimate on a larger multirotor drive.

In a similar way, the flux estimate vs. rotor electrical angle can be plotted against rotor speed to confirm good flux observer performance over the entire speed range. Figure 52 shows data from the smaller quadrotor and FFv1.2s drive over a range of speeds from about 1,500rpm to 6,000rpm. Some attenuation and phase shift is present below 1,500rpm due to the high-pass filter on *V-IR* (Section 2.2.2). However, in the operating speed range the flux waveform is sinusoidal and clean enough for sensorless control.

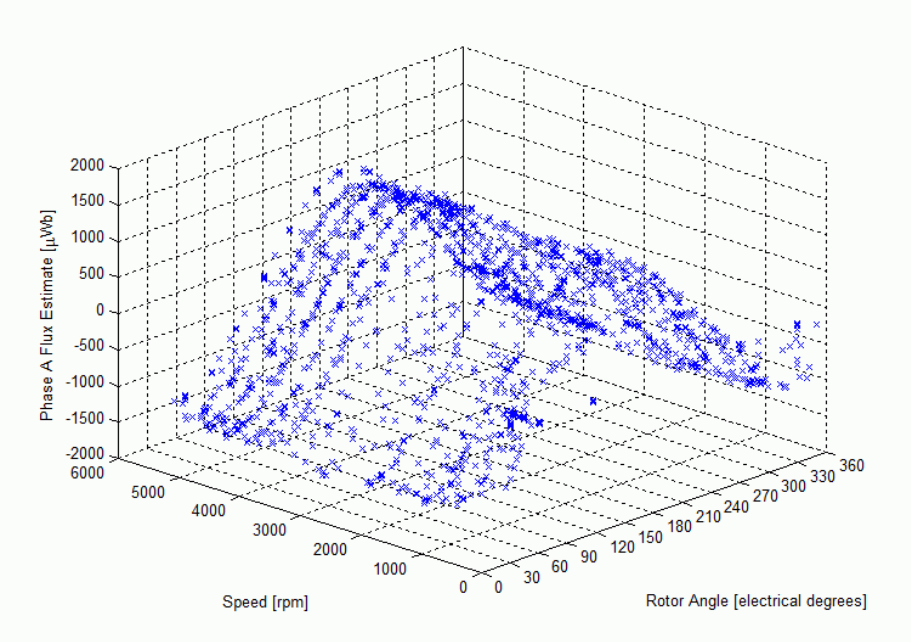


Figure 52: Flux estimate vs. speed and rotor electrical angle for an FFv1.2s mulitorotor drive.

The FFv1.1 and FFv1.2s drive and the "air" software have been flight-tested on a number of airframes successfully. They provide a quieter and, in most cases, more efficient controller for multirotors than normal off-the-shelf BLDC drives. Although rotor acceleration limits may prevent its practical use, they are also capable of regenerative braking to slow the propellers faster than they would normally coast down. This, combined with bandwidth-extending feedforward channels and closed-loop speed control, can offer performance improvements over normal BLDC drives as well.

The "air" software implemented on both the FFv1.1 and FFv1.2s (they are pin-compatible as far as the microcontroller is concerned) is provide for reference [9]. Some video of the FFv1.2s drives on a quadrotor is also available [8].

5 Conclusion

This article details a comparatively simple method of sensorless field-oriented control of permanent magnet synchronous motors based on a rotor flux-observer, and its implementation on several test platforms. Although it has some limitations, the method presented has advantages in simplicity and computation efficiency compared to other observer-based methods. Its flux comparator algorithm is inspired by the more straightforward application of digital Hall-effect sensors and linear position extrapolation to field-oriented control. As such, the majority of the software used to implement this method could be used interchangeably (or simultaneously) with Hall effect sensors in a mixed sensored/sensorless system. However, it has also served well as a stand-alone sensorless position estimation algorithm for systems where a minimum closed-loop speed is defined.

The use of a flux observer offers benefits at low-speeds, where its integral relationship to voltage creates a natural averaging effect of very small voltage signals. A physical model of the motor, presented in Section 1.1, guides the development of the flux observer. The details of the flux observer are not only

determined by theoretical concerns, but also by the practical implementation of sensor and driver hardware, as explored in Section 1.3.

The most important parameter determining the limits of this method is motor speed. At low speed, convergence of the flux observer is limited by the high-pass filtering required to minimize the effect of sensor bias, as well as by angular error accumulated due to rotor acceleration. At high speeds, the effects of digital delay are the most limiting factor. A summary of the frequency limits of this method, along with equations to guide the analysis of these limits, was presented in Section 2.5.2. This summary may be the most useful reference for applying this method to a new system.

The sensorless method presented here has been demonstrated on several test platforms with motors varying in power and speed enough to show scalability. It has been implemented on low-cost microcontrollers with minimal code size requirements. That said, the hardware and software available in [9] are not intended to be a complete solution. Instead, they should be considered a technical reference or a starting point for development. Very little abstraction is applied and it would be impossible implement this method without both a thorough understanding of the theory *and* the ability to independently create the hardware and software required from scratch.

Although this document may be edited for accuracy or clarification, development of the Gen. 1 method is considered complete at the original time of writing. Hopefully, it provides a useful reference for you and your own projects!

-Shane Colton

References

[1] Mevey, James Robert. *Sensorless Field Oriented Control of Brushless Permanent Magnet Synchronous Motors*. S.M. Thesis, Kansas State University. 2009.

[2] S. Chakrabarti, T.M. Jahns, and R.D. Lorenz, "Current Regulation for Surface Permanent Magnet Synchronous Motor Drives Using Integrated Current Sensors in the Low-Side Switches," IEEE Transactions on Industry Applications, vol. 42, no. 4, pp. 1080-1091, Jul./Aug. 2006.

[3] Colton, Shane W. *Design and Prototyping Methods for Brushless Motor and Motor Control*. S.M. Thesis, Massachusetts Institute of Technology. 2010. Available online: https://googledrive.com/host/0B0ZbiLZrqVa6Y2d3UjFVWDhNZms/motordrive/SC_ThG.pdf.

[4] Wilson, Dave. *Field Oriented Control of Permanent Magnet Motors*. Texas Instruments, 2012. Presentation available online: https://www.youtube.com/watch?v=cdiZUszYLiA.

[5] Sang-Soo Lee, Jae-Hoon Kim, Dong-Seok Hyun. *Programmable LPF-based Stator Flux Estimator for Sensorless Induction Motor Control*. IEEE Vehicle Power and Propulsion Conference, Seoul, Korea. Oct. 9-12, 2012.

[6] Colton. Shane W. *3ph Duo: 2 x 1kW Brushless Motor Controller w/ Field-Oriented Control: Design Notes*. 2010. Available online: https://googledrive.com/host/0B0ZbiLZrqVa6Y2d3UjFVWDhNZms/3phduo.pdf

[7] Guan, Charles. *Chibikart*. Project Documentation available online: http://www.etotheipiplusone.net/?page_id=1987.

[8] Colton, Shane. *FFv1.2s Custom ESCs + KK2.0 Angle Control*. Video available online: https://www.youtube.com/watch?v=09bm0NCaHWw.

[9] The documentation (hardware and software) associated with this article is available online:

All documentation in one zip file (28.9MB): https://googledrive.com/host/0B0ZbiLZrqVa6Y2d3UjFVWDhNZms/motordrive/sensorless_gen1_DOC.zip

3ph v3.1 power stage EAGLE schematic and board layout, miscellaneous documentation (9.18MB): https://googledrive.com/host/0B0ZbiLZrqVa6Y2d3UjFVWDhNZms/motordrive/3ph_v3_1_DOC.zip

FF v1.1 EAGLE schematic and board layout, BOM, datasheets, and other documentation (9.84MB): https://googledrive.com/host/0B0ZbiLZrqVa6Y2d3UjFVWDhNZms/motordrive/FF_v1_1_DOC.zip

FF v1.2s EAGLE schematic and board layout, BOM, datasheets, and other documentation (9.39MB): https://googledrive.com/host/0B0ZbiLZrqVa6Y2d3UjFVWDhNZms/motordrive/FF_v1_2s_DOC.zip

FF v1.1 and v1.2s alpha firmware (Air and Ground versions), IAR EWARM Projects (515KB): https://googledrive.com/host/0B0ZbiLZrqVa6Y2d3UjFVWDhNZms/motordrive/FF_FWR_alpha.zip

Appendices

Appendix A: Derivation of Phase Lag during Acceleration

Figure 53 shows a geometric representation of the error under constant acceleration.

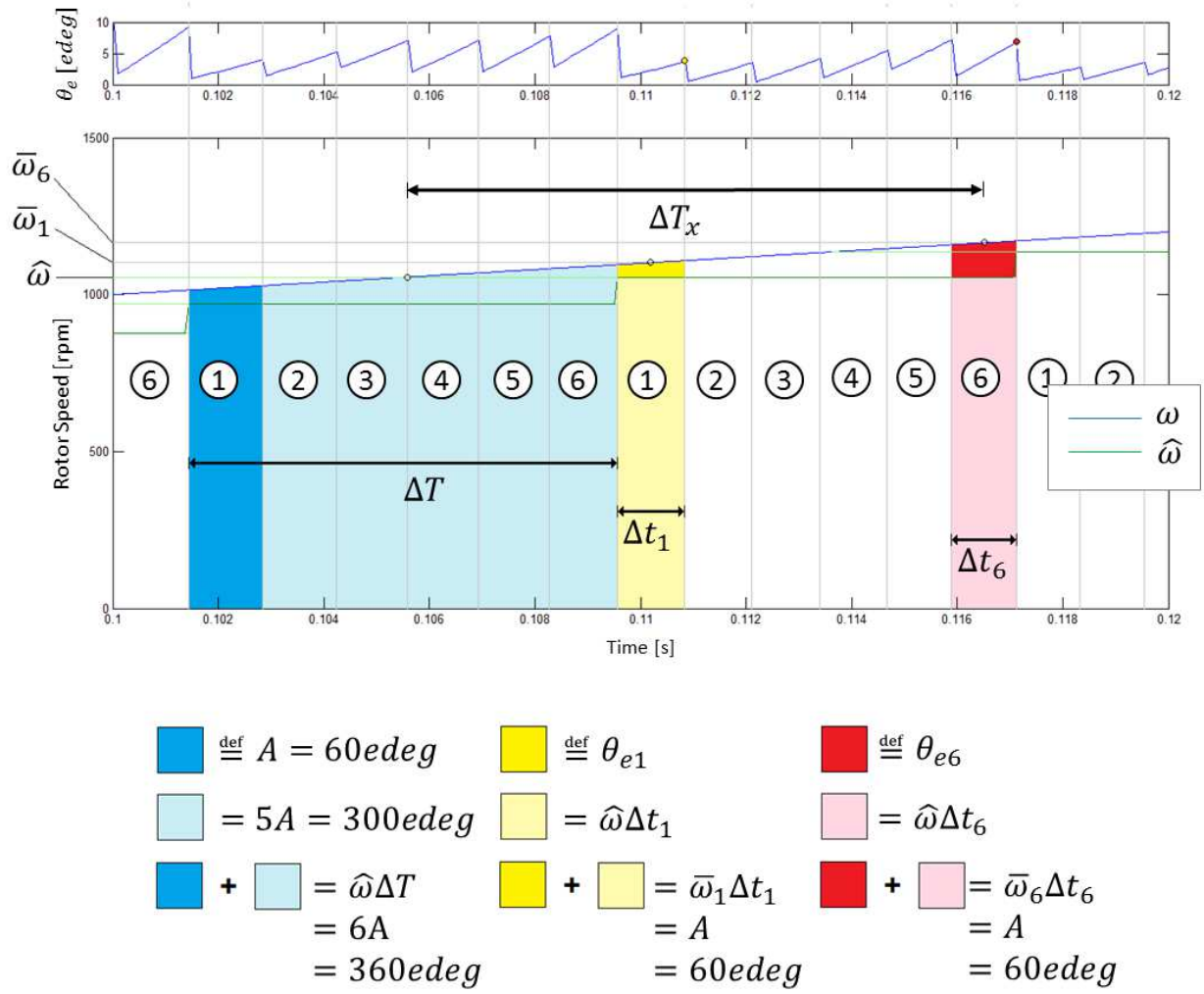


Figure 53: Geometric representation of the position estimate error during acceleration.

The blue line represents the true speed of the motor; the green line is the speed estimate, which is updated once per electrical cycle ($N = 1$). Vertical gray lines show the zero-crossings. Note that the time interval between zero-crossings is decreasing, since speed is increasing. However, the area under the true speed curve between each zero-crossing is constant and equal 60edeg, defined as A. Using the parameters defined in Figure 53, it is possible to derive a general formula for the position error accumulated during each extrapolation interval:

$$\theta_{ek} = A - \hat{\omega}\Delta t_k$$

$$= A - \hat{\omega}\left(\frac{A}{\bar{\omega}_k}\right)$$

$$= \left(1 - \frac{\hat{\omega}}{\bar{\omega}_k}\right)A$$

As expected, the position error accumulated depends on how much the rotor speed has changed since the most recent speed update. The worst-case position error occurs in the extrapolation interval just prior to a new speed estimate. Let the average speed during this final interval be $\bar{\omega}_x$. (With $N = 1$, as in Figure 53, $\bar{\omega}_x = \bar{\omega}_6$.) The average speed during the final interval before a speed update and the previous speed estimate, $\hat{\omega}$, obey a kinematic relationship:

$$\bar{\omega}_x^2 = \hat{\omega}^2 + 2\alpha(9N - 0.5)A$$

where α is the acceleration (in consistent units with $\hat{\omega}$ and $\bar{\omega}_k$) and $(9N - 0.5)A$ is the total angular displacement of the rotor from the center of the previous speed estimate interval to the center of the final extrapolation interval. In Figure 53, where $N = 1$, the total angular displacement covered over this interval, labeled ΔT_x, is:

$$(9N - 0.5)A = 8.5A = 510edeg$$

Substituting the kinematic relationship between estimated and final average speed into the formula for position error, the worst-case position error during the final extrapolation interval before a new speed update is define:

$$\theta_{ex} = \left(1 - \frac{\hat{\omega}}{\bar{\omega}_x}\right)A$$

$$= \left(1 - \frac{\hat{\omega}}{\sqrt{\hat{\omega}^2 + (18N - 1)\alpha A}}\right)A$$

This relationship can be simplified by introducing a dimensionless acceleration parameter, *K*:

$$K = \frac{(18N - 1)\alpha A}{\hat{\omega}^2}$$

Then the maximum position error becomes:

$$\theta_{ex} = \left(1 - \frac{1}{\sqrt{1 + K}}\right)A$$

Using $A = 60$edeg, and the worst-case position error can be plotted as a function of dimensionless acceleration *K* as shown in Figure 54.

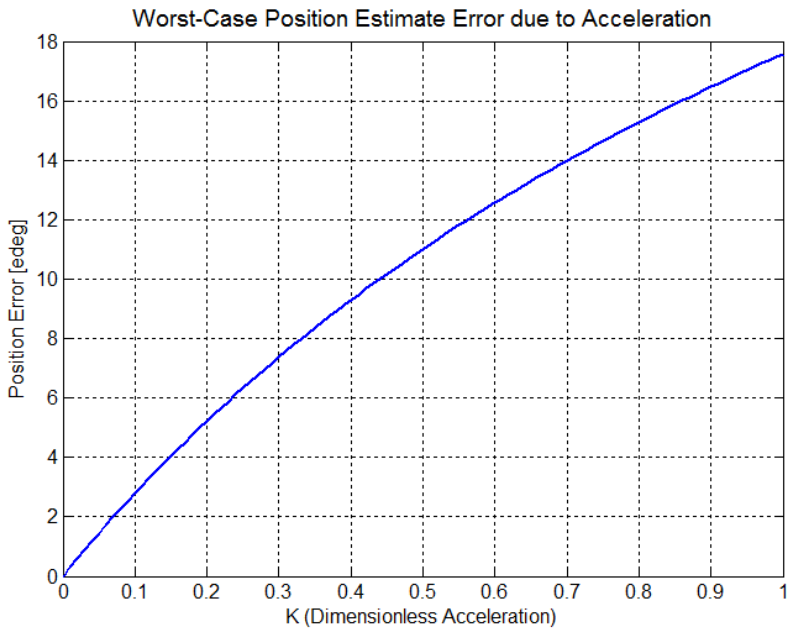


Figure 54: Worst-case position estimate error as a function of dimensionless acceleration.

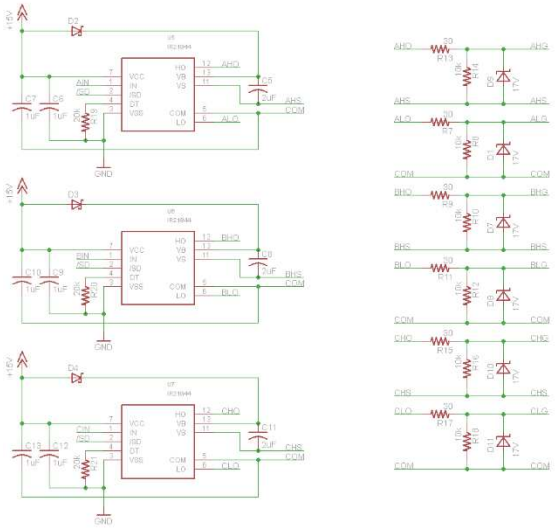
For example, to keep the position error under 10edeg, the maximum value for *K* should be 0.44 over the entire range of possible speeds and accelerations during sensorless control.

Appendix B: Schematics and Board Images

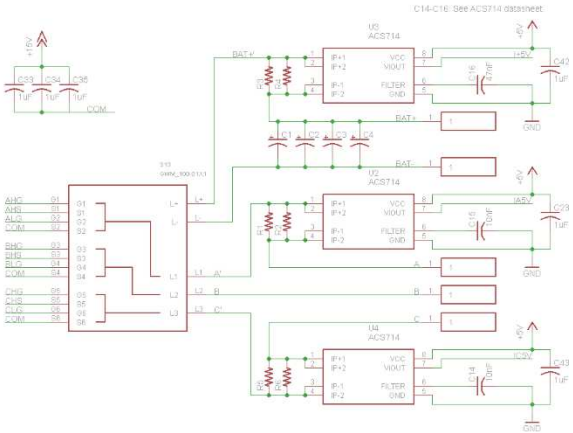
Schematics and printed circuit board images for three motor controllers are provided in this section. The PCB design is done in EAGLE and the EAGLE files are available for reference [9].

3ph v3.1

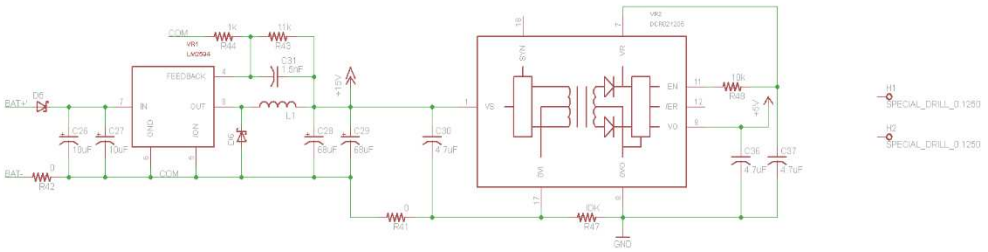
Gate Drive:



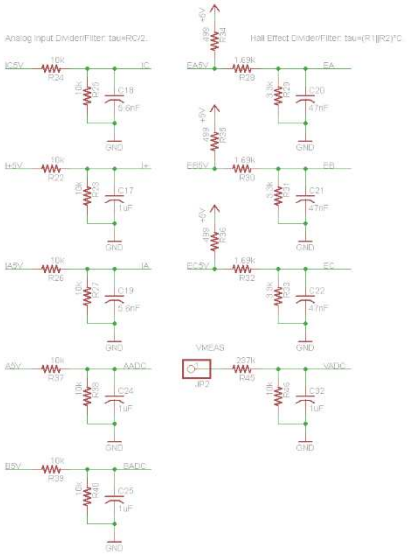
MOSFET Three-Phase Inverter:



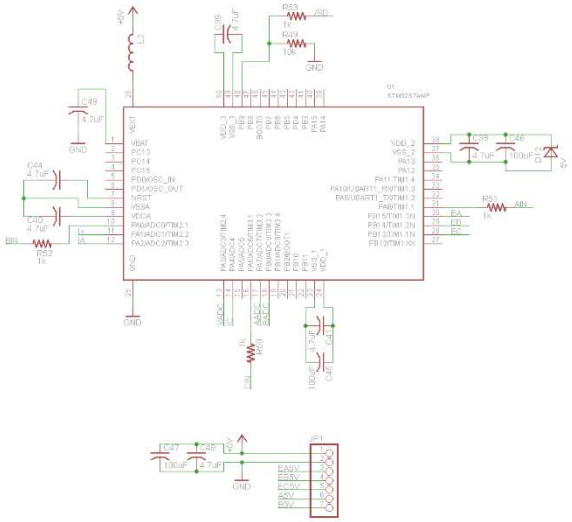
Power Supplies:

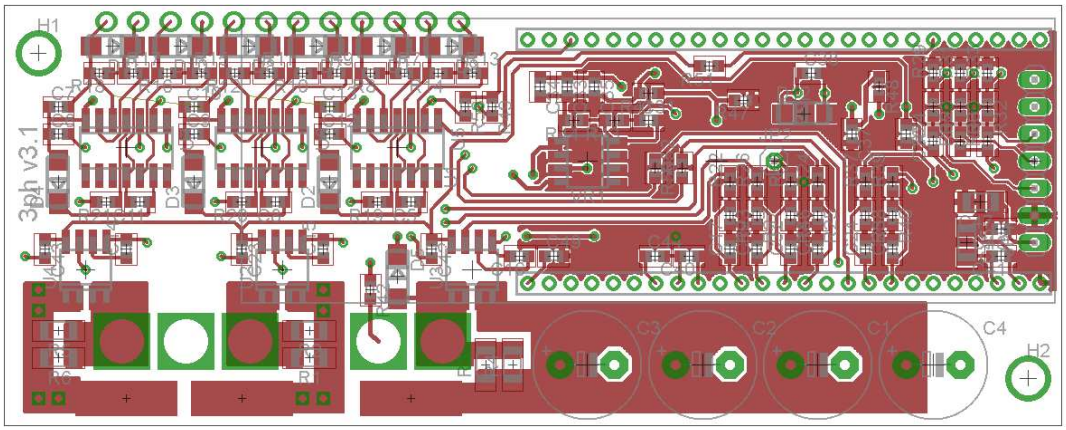


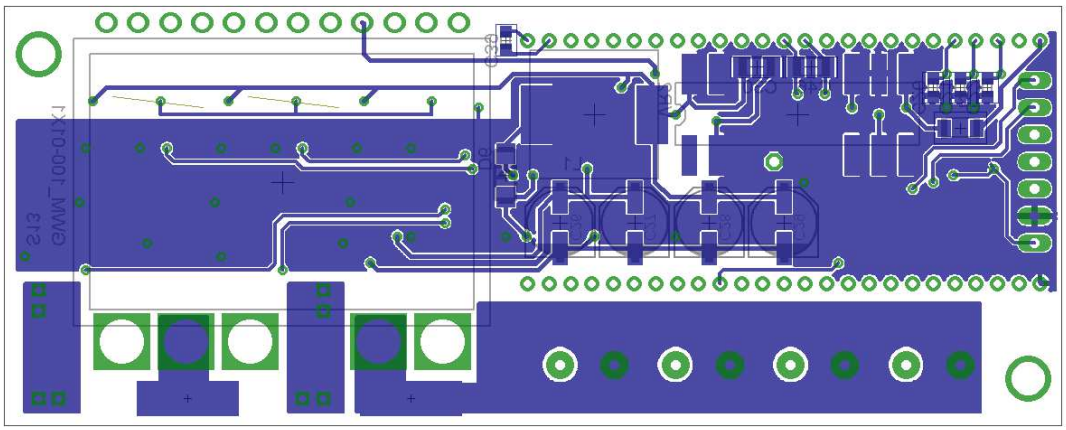
Filters and Dividers:



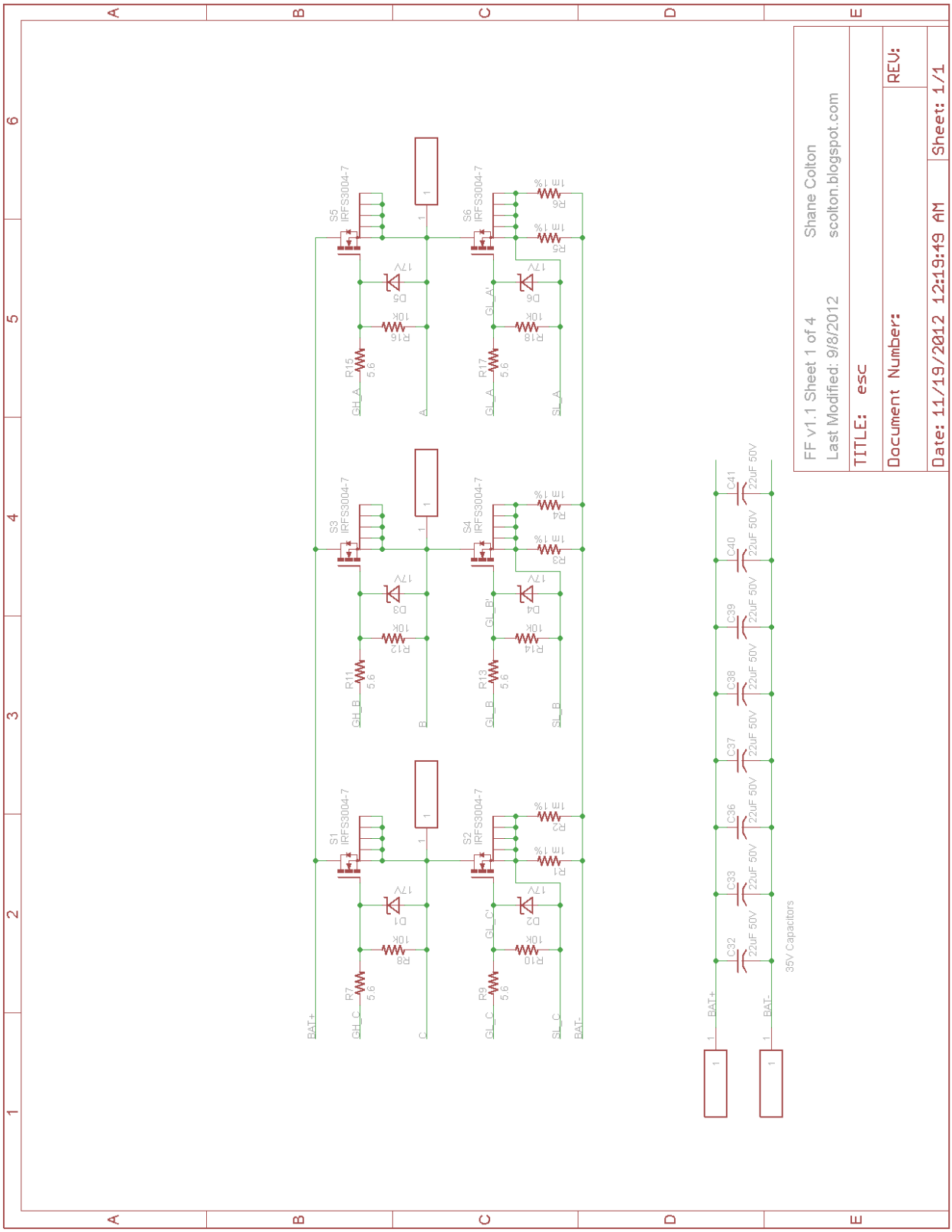
Microcontroller:

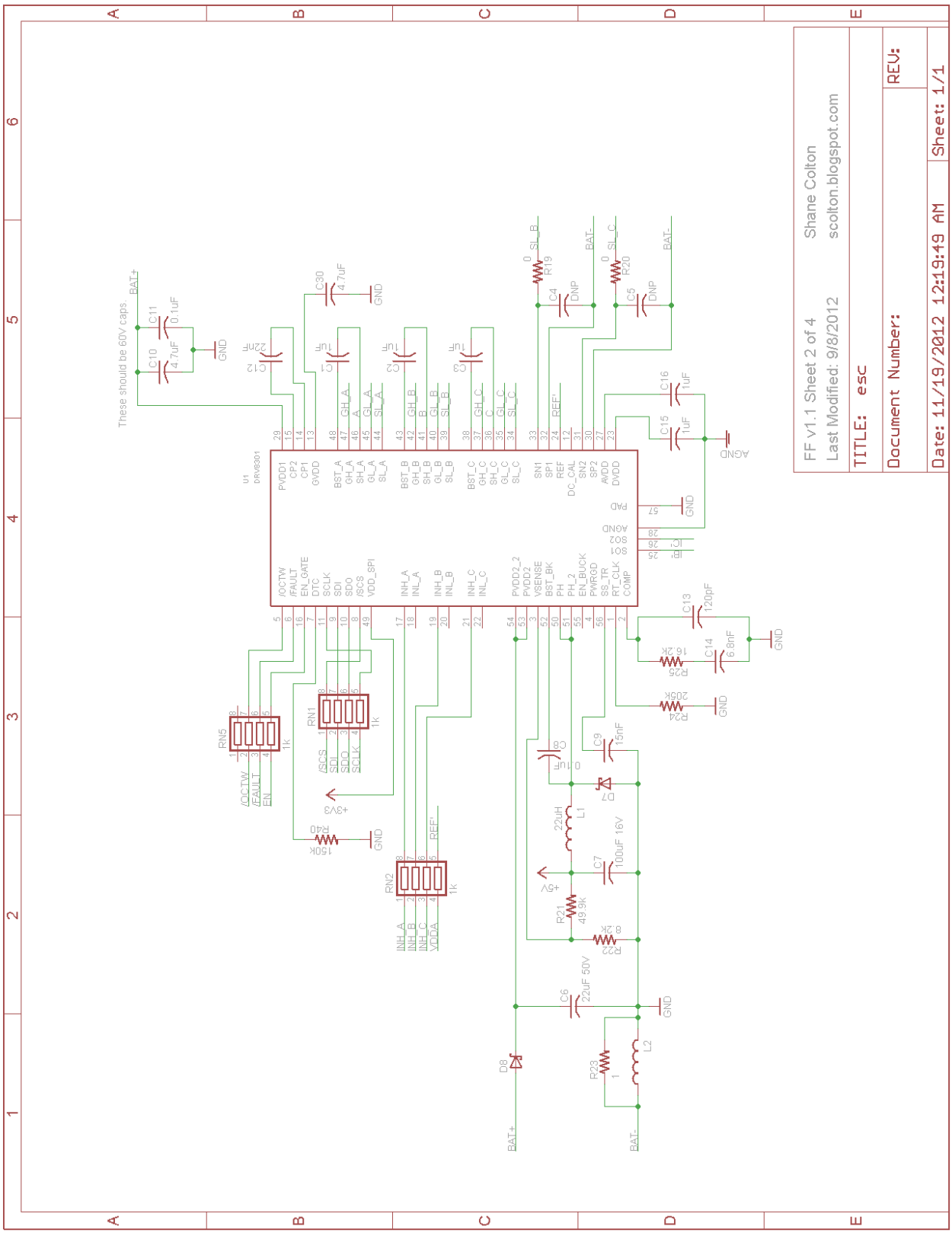


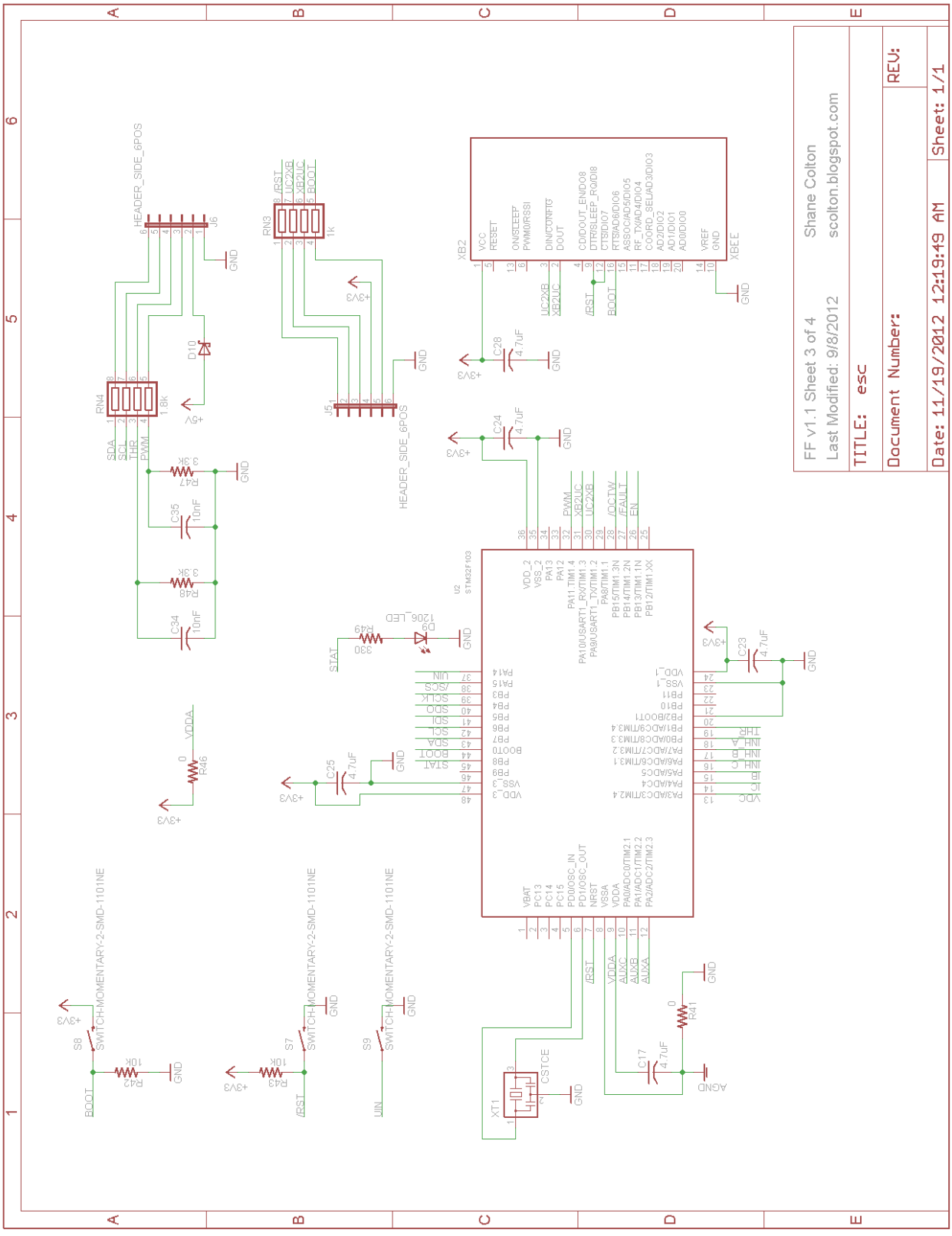


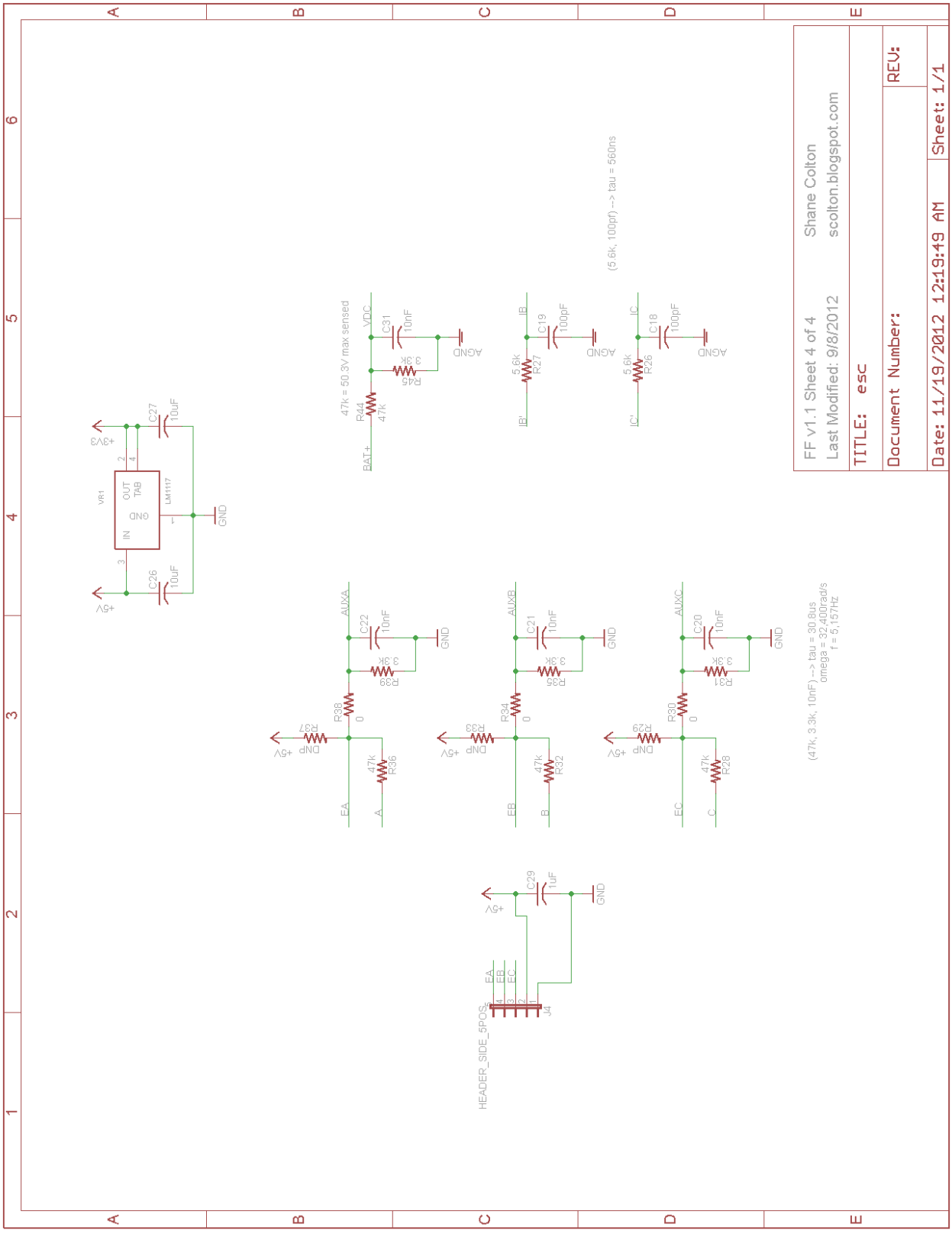


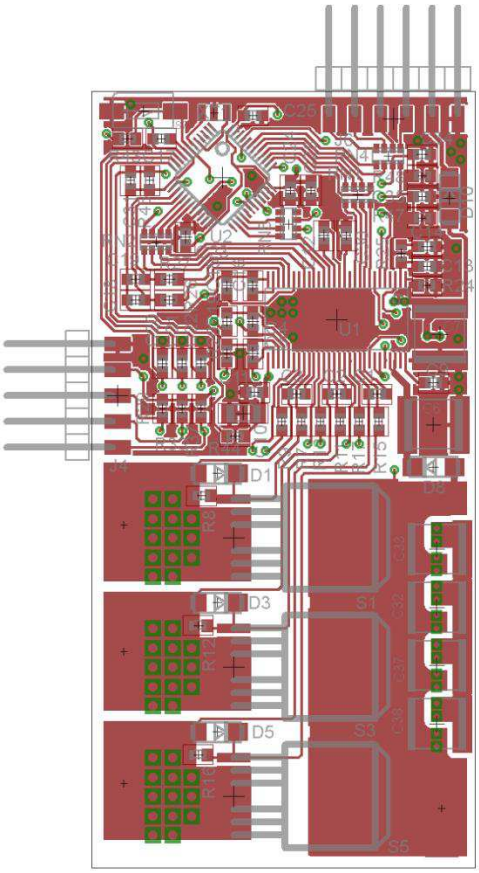
FF v1.1

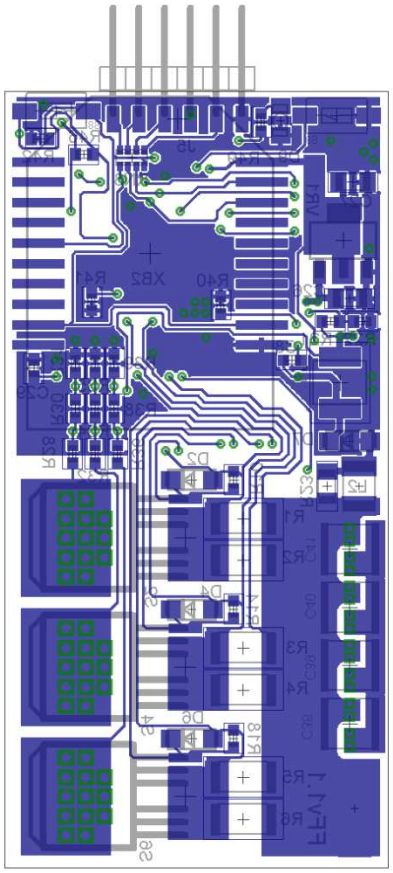

FF v1.1 Sheet 1 of 4
Last Modified: 9/8/2012
Shane Colton
scolton.blogspot.com
TITLE: esc
Document Number:
REV:
Date: 11/19/2012 12:19:49 AM
Sheet: 1/1
BAT+
BAT-
35V Capacitors


FF v1.1 Sheet 2 of 4
Last Modified: 9/8/2012
Shane Colton
scolton.blogspot.com
TITLE: esc
Document Number:
REV:
Date: 11/19/2012 12:19:49 AM
Sheet: 1/1
These should be 60V caps.
BAT+
BAT-
GND
AGND
+3V3
+5V
U1
DRV8301
RN1
RN2
RN5
L1
L2
D7
D8
C6
C7
C9
C13
C14
C15
C16
C30
R19
R20
R21
R22
R23
R24
R25
R40

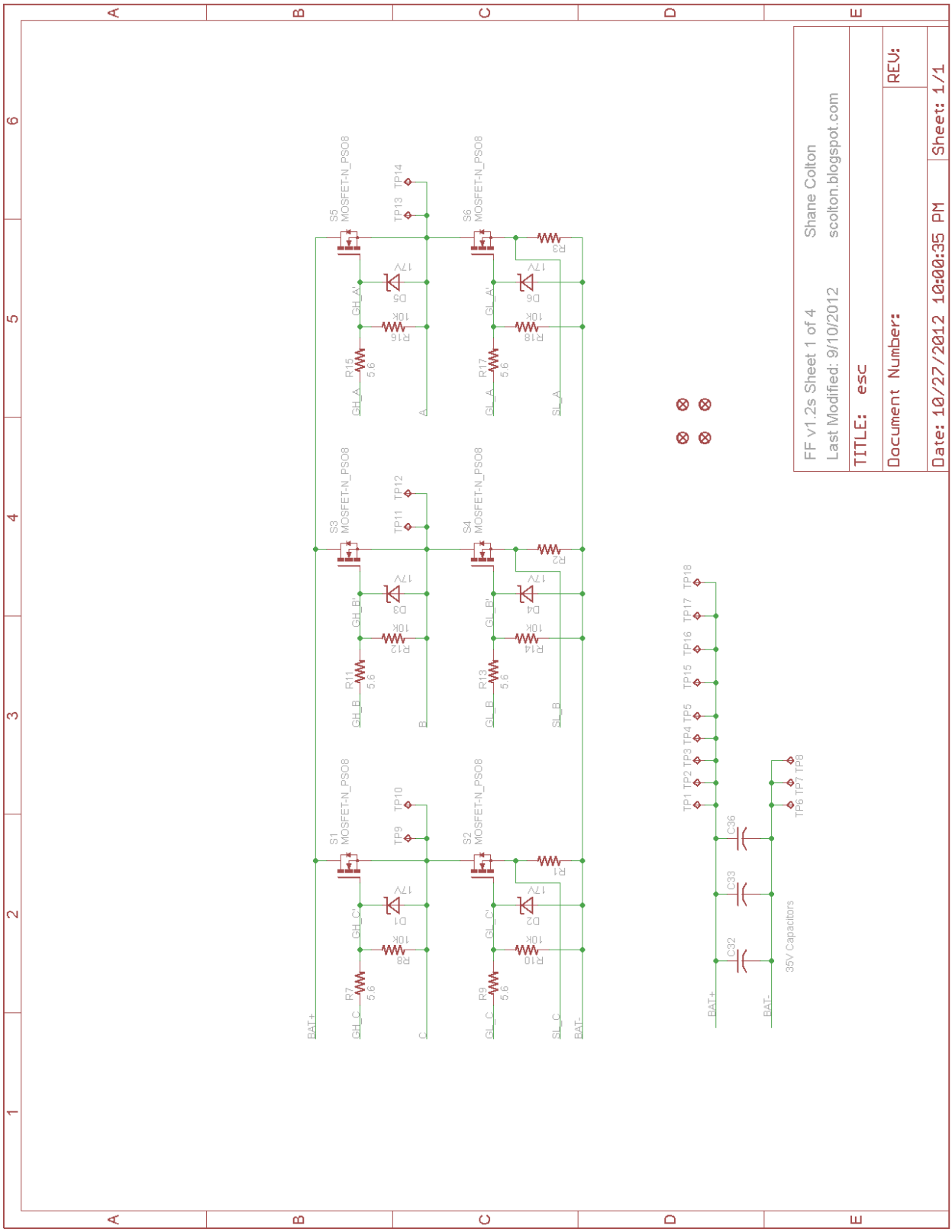

FF v1.1 Sheet 3 of 4
Last Modified: 9/8/2012
Shane Colton
scolton.blogspot.com
TITLE: esc
Document Number:
REV:
Date: 11/19/2012 12:19:49 AM
Sheet: 1/1
HEADER_SIDE_6POS
RN4
1.8K
SDA
SCL
THR
PWM
D10
RN3
1K
/RST
UC2XB
XB2UC
BOOT
J5
J6
XB2
XBEE
VCC
RESET
ON/SLEEP
PWM0/RSSI
DIN/CONFIG
DOUT
CD/DOUT_EN/DIO8
DTR/SLEEP_RQ/DI8
CTS/DIO7
RTS/ADB/DIO6
ASSOC/AD5/DIO5
RF_TX/AD4/DIO4
COORD_SEL/AD3/DIO3
AD2/DIO2
AD1/DIO1
AD0/DIO0
VREF
GND
C28
4.7uF
C24
4.7uF
U2
STM32F103
VDD_2
VSS_2
PA13
PA12
PA11_TIM1.4
PA10/USART1_RX/TIM1.3
PA9/USART1_TX/TIM1.2
PA8/TIM1.1
PB15/TIM1.3N
PB14/TIM1.2N
PB13/TIM1.1N
PB12/TIM1.XX
PWM
XB2UC
UC2XB
/DOUTEN
/FAULT
EN
VDD_1
VSS_1
PB11
PB10
PB2/BOOT1
PB1/ADC9/TIM3.4
PB0/ADC8/TIM3.3
PA7/ADC7/TIM3.2
PA6/ADC6/TIM3.1
PA5/ADC5
PA4/ADC4
PA3/ADC3/TIM2.4
VBAT
PC13
PC14
PC15
PD0/OSC_IN
PD1/OSC_OUT
NRST
VSSA
VDDA
PA0/ADC0/TIM2.1
PA1/ADC1/TIM2.2
PA2/ADC2/TIM2.3
C23
4.7uF
R49
330
D9
1206_LED
STAT
R46
0
VDDA
C25
4.7uF
R41
0
C17
4.7uF
AGND
XT1
CSTCE
S8
S7
S9
SWITCH-MOMENTARY-2-SMD-1101NE
R42
10K
R43
10K
BOOT
/RST
UIN
C35
10nF
C34
10nF
R47
3.3K
R48
3.3K
+3V3
+5V
GND

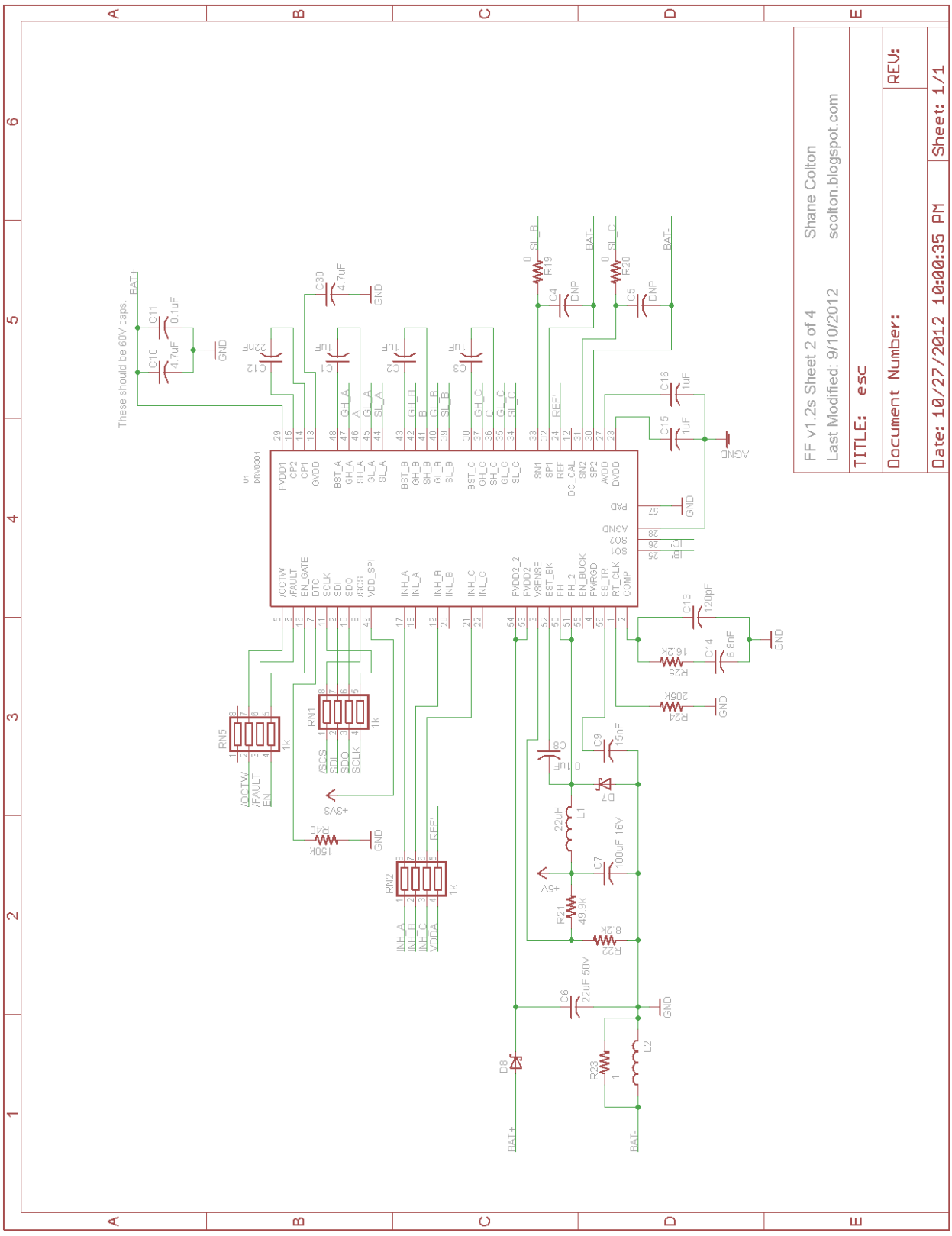

FF v1.1 Sheet 4 of 4
Last Modified: 9/8/2012
Shane Colton
scolton.blogspot.com
TITLE: esc
Document Number:
REV:
Date: 11/19/2012 12:19:49 AM
Sheet: 1/1
+3V3
+5V
GND
AGND
VR1
LM1117
IN
OUT
TAB
C27
10uF
C26
10uF
47k = 50.3V max sensed
VDC
BAT+
R44
47k
C31
10nF
R45
3.3k
IB
IB'
R27
5.6k
C19
100pF
IC
IC'
R26
5.6k
C18
100pF
(5.6k, 100pf) --> tau = 560ns
AUXA
AUXB
AUXC
C22
C21
C20
10nF
R39
R35
R31
3.3k
R38
R34
R30
0
R37
R33
R29
DNP
EA
EB
EC
A
B
C
R36
R32
R28
47k
(47k, 3.3k, 10nF) --> tau = 30.8us
omega = 32,400rad/s
f = 5,157Hz
C29
1uF
HEADER_SIDE_5POS
J4

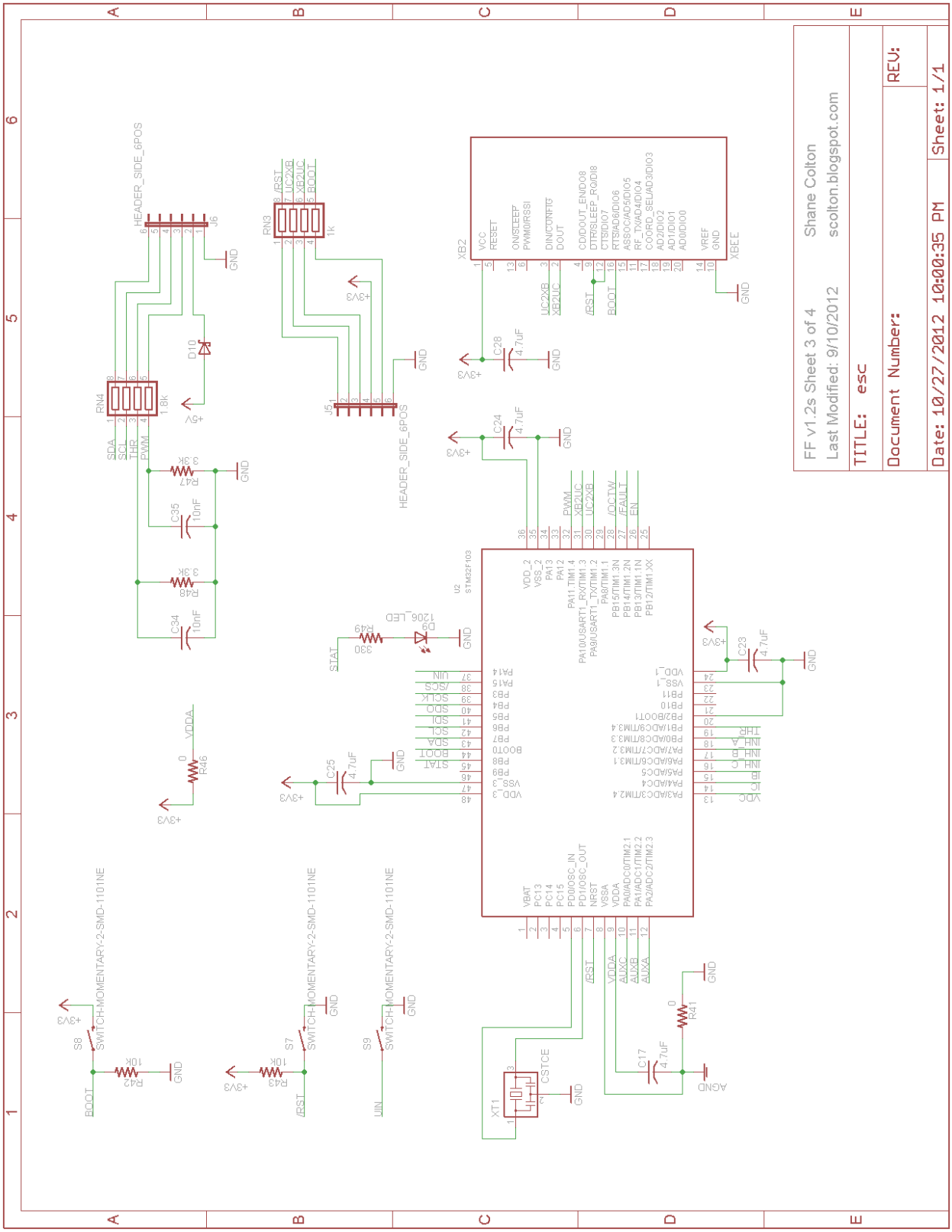


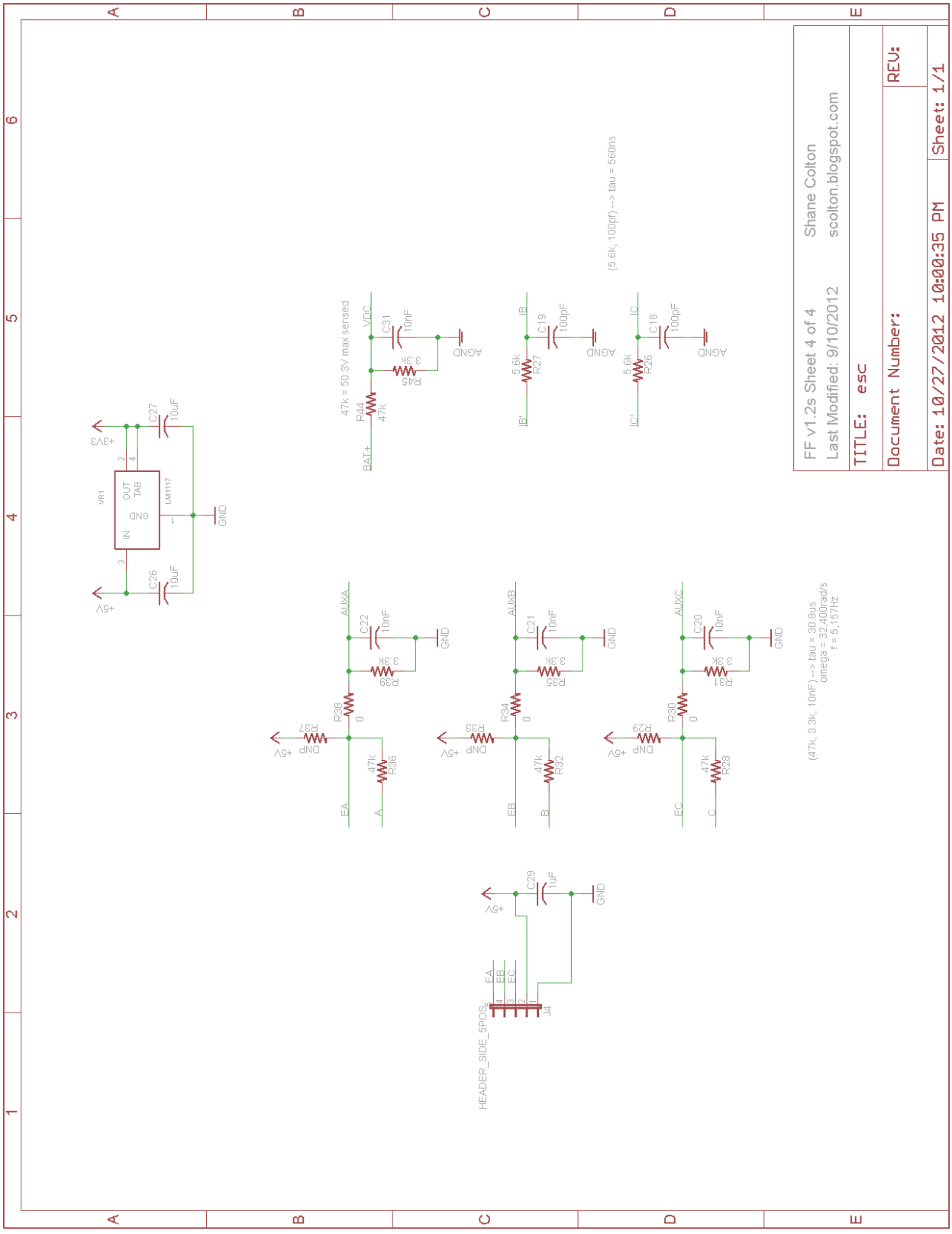


FF v1.2s


FF v1.2s Sheet 1 of 4
Last Modified: 9/10/2012
Shane Colton
scolton.blogspot.com
TITLE: esc
Document Number:
REV:
Date: 10/27/2012 10:00:35 PM
Sheet: 1/1
BAT+
BAT-
35V Capacitors


FF v1.2s Sheet 2 of 4
Last Modified: 9/10/2012
Shane Colton
scolton.blogspot.com
TITLE: esc
Document Number:
REV:
Date: 10/27/2012 10:00:35 PM
Sheet: 1/1
These should be 60V caps.
BAT+
BAT-
GND
AGND
+3V3
+5V
U1
DRV8301
C10
4.7uF
C11
0.1uF
C30
4.7uF
C4
DNP
C5
DNP
C15
1uF
C16
1uF
C13
120pF
C14
6.8nF
C9
15nF
C7
100uF 16V
C6
22uF 50V
L1
22uH
L2
R19
0
R20
0
R21
49.9k
R22
2
R23
1
R24
205k
R25
16.2k
R40
150k
RN1
1k
RN2
1k
RN5
1k
D7
D8


FF v1.2s Sheet 3 of 4
Last Modified: 9/10/2012
Shane Colton
scolton.blogspot.com
TITLE: esc
Document Number:
REV:
Date: 10/27/2012 10:00:35 PM
Sheet: 1/1


FF v1.2s Sheet 4 of 4
Last Modified: 9/10/2012
Shane Colton
scolton.blogspot.com
TITLE: esc
Document Number:
REV:
Date: 10/27/2012 10:00:35 PM
Sheet: 1/1
(47k, 3.3k, 10nF) --> tau = 30.8us
omega = 32.400rad/s
f = 5,157Hz
(5.6k, 100pf) --> tau = 560ns
47k = 50.3V max sensed
VR1
LM1117
HEADER_SIDE_5POS
J4
EA
EB
EC
AUXA
AUXB
AUXC
VDC
BAT+
IA
IB
IC
IA'
IB'
IC'
+3V3
+5V
GND
AGND
DNP