Chapter 1
Overview

Introduction

This document describes the Hunter Electronic Motor Controller designed for speed, torque and position control of permanent magnet synchronous motors. It is designed as a sensorless controller, meaning that no position or speed sensors are needed for the controller to operate. Only the motor winding leads need to connect to the controller.

This chapter is an overview of the controller. It provides a basic understanding of the controller when operating in the speed control mode and in the torque limiting mode. Extra features such as over speed limiting, over voltage regeneration limiting, flux weakening and position control are not considered here but are described in detail in later chapters. Also, elements of the basic speed control and torque control structures outlined here are described in detail in later chapters.

A complete block diagram of the controller as actually used in the software can be found in the Design directory of the controller source code. It is updated with the source code.

Versions of the controller can operate most types of permanent magnet synchronous motors. This includes brushless DC motors and stepper motors. It includes motor sizes from the largest industrial motors down to small quadcopter motors. The possible winding geometries include 3-wire 3-phase windings and 4-wire 2-phase windings as commonly used in stepper motors.

The controller is designed for maximum performance. Even though it is a sensorless design, very few compromises have been made. In most cases, the motor performance is limited only by the motor itself, not the controller. Features of the controller include:

1. Maximum motor acceleration and deceleration is available right down to zero speed. This allows fast motor take off from standstill and fast motor reversals.

2. The motor goes into torque limit on overload, except at and near zero speed where pole slipping will occur on overload.

3. Motor speed is controlled to the accuracy of the controller's timing reference, usually a crystal controlled oscillator. The controller offers 30,000 speed steps from zero to the maximum speed.

4. The response to a step change in speed is very fast, with no overshoot, even from zero speed.

5. Automatic and fast flux weakening is provided to allow the motor speed to extend beyond the base speed.

6. Position control with trapezoidal speed trajectory is incorporated.

To achieve this performance, the control methods used are completely different to to those used in conventional motor speed controllers. This must be kept in mind when trying to understand its operation.

Handling Different Motor Types

The controller software is designed to handle different types of permanent magnet synchronous motors with either two or three phase winding configurations and any number of pole pairs.

Rather than reconfiguring the controller software for different motor types, the basic controller is designed for a 2-pole, 2-phase motor with additional software to convert the actual motor voltages, currents, torque and speed to an equivalent 2-pole, 2-phase motor.

Unless otherwise indicated, the controller design as described here always assumes a 2-phase, 2-pole motor.

Measurement Units

To simplify calculations, speed is expresses as electrical radians/second, rotor phase angle as electrical radians and torque as electrical Newtons.metre, which is motor torque in Newtons.metre divided by the number of pole pairs. Motor flux, or more correctly flux linkage, is expressed as two phase equivalent peak rotor induced voltage (back emf)/(speed in radians/sec).

Rotor Position Tracking

Like most controllers for permanent magnet synchronous motors, this controller needs to know the rotor position to operate.

The major barrier to the operation down to zero speed for conventional sensorless controllers is the sensing of the rotor position at zero speed. At high speeds, rotor position can be found from the measured back emf (the voltage induced in the motor windings by the rotating flux from the rotor magnets when the rotor is spinning). At low and zero speed, the back emf is too small to measure so it cannot be used. To overcome this, the motor must be sped up from zero speed by injecting a rotating current into the windings before switching to back emf measurement.

This controller overcomes this problem by not using position sensing at all. Instead, the controller calculates the rotor position versus time trajectory and adjusts the motor winding voltages to keep the rotor at this position using feed forward control methods. The controller also calculates the expected winding currents and compares these to the measured currents to determine the rotor position error. This is then used to adjust the winding voltages to bring the actual rotor position trajectory into alignment with the calculated trajectory. At low and zero speeds, a rotor position error does not produce a change in measured currents so instead the winding voltages are adjusted to produce an alignment current in the windings, sometimes called a holding current, to force the rotor into the calculated rotor position trajectory.

To ensure the rotor stays locked to the calculated rotor position at low and zero speed and not jump out of sync from oscillations induced by load changes or speed command changes, the controller's output impedance as seen by the motor is adjusted to maximise electrical damping of the rotor. Undamped oscillations are the major cause of low speed problems in current electronic speed controllers. For further details of damping in the controller, see Chapter 8: Stability Analysis.

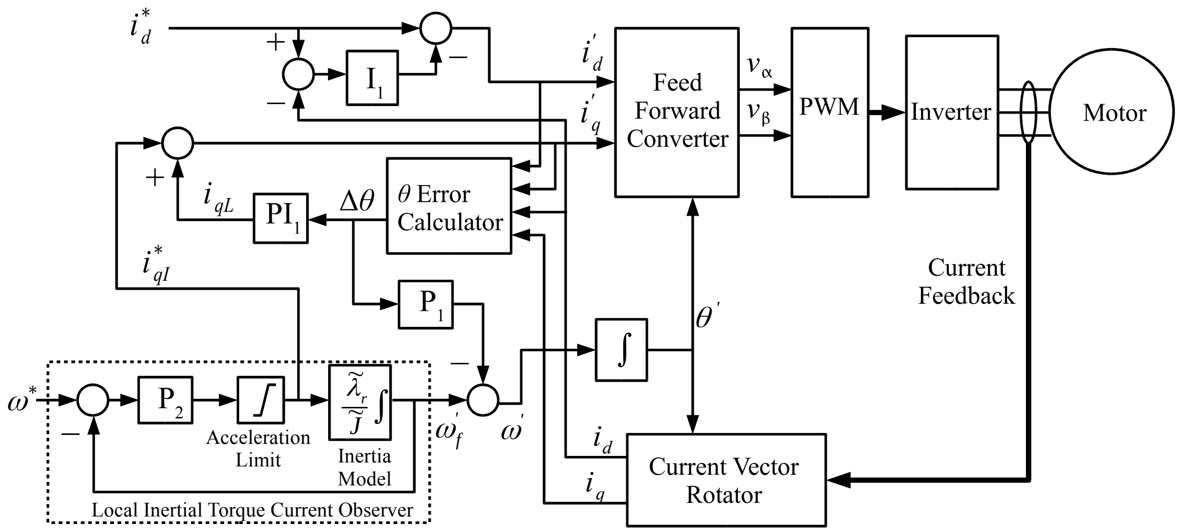

Figure 1.1. Control structure of the motor controller for speed control.

Speed Control using Feed Forward Control

In a conventional controller, feedback is used extensively to control motor currents and speed. This leads to many problems. The feedback current controllers must operate with a fast response time for sufficiently fast control of the output voltages, but this response time is limited by sampling delays and pulse width modulation delays. The delays severely limit the maximum frequency at which the controller can operate. Also, only limited filtering can be applied to the current measurement signals to avoid filter delays which would otherwise further limit the maximum frequency, often resulting in noisy waveforms. Another problem with the feedback current controllers is that an output voltage overhead must be maintained for the control loops to remain stable. Typically the output voltage must be limited to about 85-90% of the maximum available voltage for a given DC link voltage, reducing the maximum motor speed that can be obtained.

The outer feedback speed controller also has limitations, especially when operating in sensorless mode. With sensorless control, a speed feedback signal must be derived from the position signal, which in turn is derived from back emf measurements. Because the back emf is not measurable below a certain speed, speed control is not possible below this speed. Also, because speed must be derived by differentiating the position signal, usually with the help of a phase locked loop structure to reduce noise, speed response is slower then it could be.

In the Hunter Motor Controller, extensive use of feed forward control techniques are used to avoid these problems. The structure of the motor controller for speed control, showing how feed forward control is implemented, is shown in Figure 1.1.

Control is implemented in d,q rotating co-ordinates aligned with the rotor flux direction. The controller can thus be classified as a vector controller. In this arrangement, the d axis current affects the rotor flux and the q axis current affects the rotor torque.

The Feed Forward Converter uses a model of the motor to convert the applied d and q axes currents i'_d and i'_q and the applied rotor electrical position angle θ' to the 2-phase equivalent of the actual motor voltages, v_α and v_β. The tick superscript (') is used here to indicate applied values generated by the controller. These may differ to the actual values in the motor, depending on controller errors although in reality, the controller will keep the applied values very close to the actual motor values. The 2-phase equivalent motor voltage reference signals v_α and v_β generated by the Feed Forward Converter are converted to pulse widths by the pulse width modulator (PWM) after conversion to 3-phase voltage signals, if required, then fed as gate drive signals to the output inverter.

The speed reference input ω^* is filtered by the Local Inertial Torque Current Observer, a simple observer structure to limit acceleration, extract the estimated inertial torque current i^*_{qI} and filter out high frequencies. The output of this block is the applied filtered speed ω'_f. The observer uses a simple inertial model of the motor load with an estimated inertia $\tilde{J}$ and an estimated rotor flux linkage $\tilde{\lambda}_r$. Note that the tilde accent (~) is used to indicate estimated quantities. The filtered speed is not used directly to generate the applied rotor angle but is modulated by the rotor angle error $\Delta\theta$ via the proportional gain P_1 to provide damping to the motor and load. The resulting modulated speed ω' is then integrated to generate the applied rotor angle θ'. There is normally no difference between ω' and ω'_f since the rotor angle error is kept at zero and the feed back damping proportional gain is very low.

The Current Vector Rotator converts the measured motor currents into the rotating vector components id and iq using the applied flux angle θ'.

The rotor angle error $\Delta\theta$ is calculated from the applied and actual motor currents and estimates of the motor parameters. In reality, a rotor angle error cannot be detected at or near zero speed when there is no back emf present, so the calculated rotor angle error $\Delta\theta$ just stays at zero for these speeds. To ensure the actual rotor angle tracks the controller's applied value at these speeds, a positive *d* axis current is applied to the motor by lifting the *d* axis reference current above zero. This naturally locks the rotor into position, provided the *d* axis current is sufficient to overcome any load torque on the motor.

The calculated rotor angle error $\Delta\theta$ is used to adjust the *q* axis torque-producing current via the proportional-integral controller PI_1 to bring the actual rotor angle in track with the applied rotor angle θ'. At low and zero speed, $\Delta\theta$ no longer responds to a change in rotor angle and in theory drops to zero but in practice drops to an error value. To prevent this from being amplified by the integral part of the PI controller, which would cause a large offset error in the *q* axis current, the DC gain of the integrator is reduced at these speeds to a level where the $\Delta\theta$ zero error will not cause a problem.

The Feed Forward Converter relies on a motor model in order to generate the correct output voltages for the given input applied currents. For the motor model, accurate estimates of the motor parameters is not possible, especially since these will vary depending on the motor currents and motor temperature. To overcome this problem, the *d* axis current is corrected by feedback. The reference current i^*_d is compared with the measured current i_d and the difference i_{qL} is used to adjust the reference current via the integral controller I_1. No extra correction is needed for the *q* axis current since it is automatically adjusted by the PI controller. These correction mechanisms are sufficient to get the speed controller to operate successfully for even fairly large errors in motor parameters although the more accurate the estimates of the motor parameters are, the higher is the performance that can be achieved from the controller.

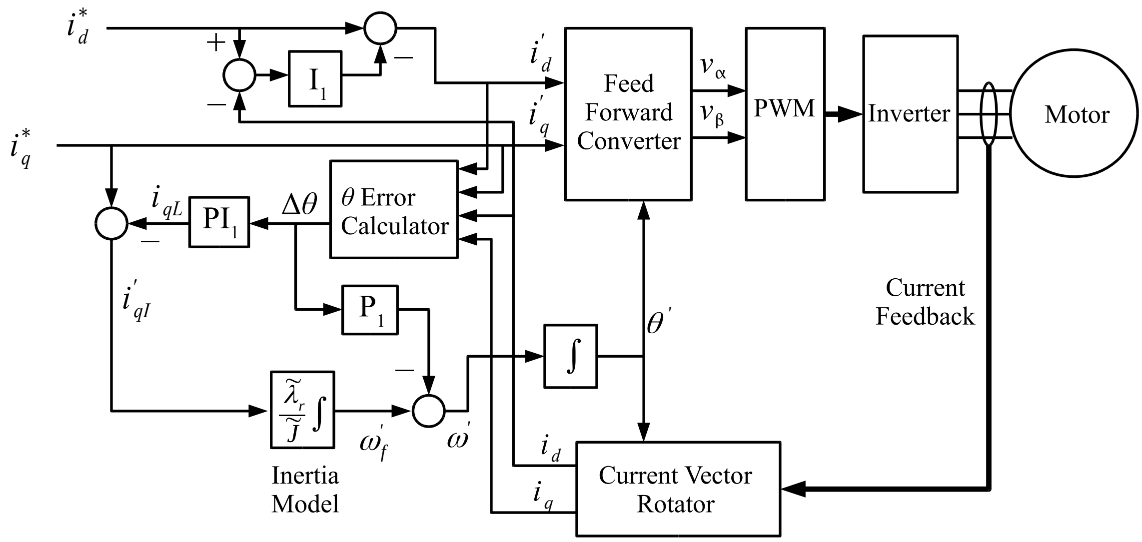

Figure 1.2. Control structure of the motor controller for torque control.

Torque Limiting

Ideally, when the motor load torque exceeds the maximum torque available from the motor, the controller should change from operating the motor at the reference speed to operating the motor at the maximum torque setting. For a traditional motor controller, this is implemented by using a cascaded control structure where an inner torque control loop is in turn controlled by an outer speed control loop, which adjusts the torque via the *q* axis current reference to the required value to keep the speed at the reference value. The *q* axis reference current is limited to the maximum torque value so that when the motor load torque is exceeded, the controller automatically swaps to constant torque operation mode.

The speed controller does not use a cascade structure. As shown in Figure 1.1, speed is instead controlled by directly controlling the rate of change of the rotor angle with the reference speed. There is no automatic method of switching to constant torque operation in the event of a motor overload. Instead, parallel speed and torque control structures are used. When motor torque is found to exceed a maximum value, detected when the applied torque current i'_q exceeds the value corresponding to the maximum torque, the controller structure is re-configured to a new structure that puts the motor in a constant torque operating mode. In this mode, with the motor overloaded, the motor speed will drop whilst keeping output torque at the maximum value. When the now torque controlled output speed rises back to the reference speed, the controller structure is swapped back to the constant speed operating mode.

The torque controller structure used when operating in constant torque mode is shown in Figure 1.2. Torque is controlled by the corresponding reference torque current i^*_q. The output of the angle error PI controller, which is actually the load component of the *q* axis current i_{qL}, is subtracted from this reference current to create an acceleration or inertial current i'_{qI} and applied to the input of the inertia model instead of the inertial current in the Local Torque Current Observer of Figure 1.1. Note that as the speed drops to near zero under torque limit, the angle error signal disappears and Patent due date 13/4/2026the motor will lose synchronism with the controller. Thus the motor cannot be controlled for motor overloads at or near zero speed. This is not a problem for most applications.