InstaSPIN-MOTION

Position Control Motor Commissioning

Adam Reynolds

12/4/13

1 Introduction

This document will walk you through the process of taking an unknown motor and have it running under position control. This is based on InstaSPIN-MOTION and the MotorWare software suite.

InstaSPIN-MOTION is available on TMS320F2806xM and TMS320F2805xM devices, and provides maximum control with minimum effort. InstaSPIN-MOTION provides the following core capabilities:

- The FAST unified software observer, which exploits the similarities between all motors that use magnetic flux for energy transduction. The FAST estimator measures rotor flux (magnitude, angle, and speed) as well as shaft torque in a sensorless FOC system.
- Motor parameter identification, used to tune the FAST software observer and initialize the innermost current (torque) PI controllers for Iq and Id control of the FOC system.
- SpinTAC™, a comprehensive motion control suite from LineStream Technologies, SpinTAC minimizes the time you spend defining how you want your motor to spin and ensures that your motor runs at its optimal level for ideal performance. Key benefits include:
- **Control** - Based on LineStream's patented Active Disturbance Rejection Control, it features a combined position and velocity control loop that is tuned with a single parameter. A parameter works across a wide range of speeds and loads, reducing or eliminating the need for gain scheduling.
- **Identify –** Identifies the system inertia and friction, which are needed for optimal control.
- **Move** - Provides the ability to control the velocity, acceleration/deceleration, and jerk of position transitions, which results in extremely smooth position movements.
- **Plan** - Easily design and execute complex motion sequences.

.

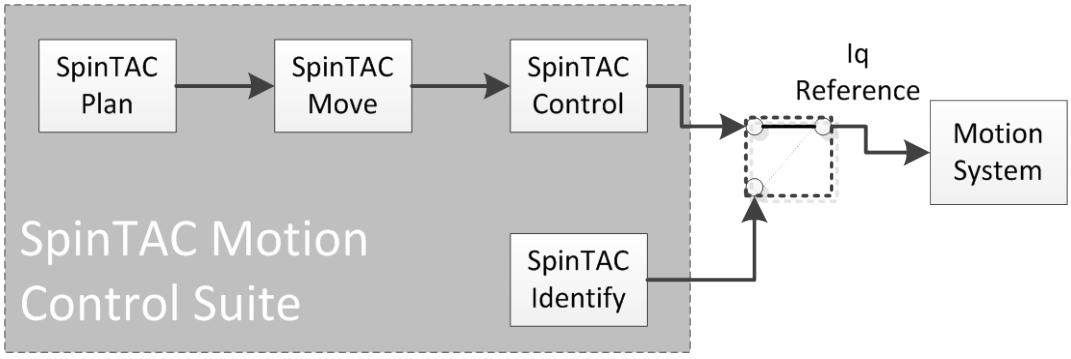

Figure 1 - SpinTAC Motion Control Suite

2 Prerequisites

2.1 Motor Prerequisites

InstaSPIN-MOTION position control is available for permanent magnet motors, with plans to extend the capability to AC induction motors. The example software assumes that you are using a quadrature encoder to provide the electrical angle feedback. InstaSPIN-MOTION requires the electrical angle of the motor. If you are using a different sensor, you will need to modify the code to provide the electrical angle information from that sensor.

2.2 Hardware Prerequisites

This document assumes that you are using a TI motor control reference kit. This document has been written to cover the following TI motor control evaluation kits:

- DRV8301-69M-KIT
- DRV8312-69M-KIT (NOTE: This kit might require an additional 5V power supply to power the encoder)
- TMDSHVMTRINSPIN

The steps will be the same for custom motor control hardware.

2.3 Software Prerequisites

InstaSPIN-MOTION position solution examples are provided as part of MotorWare. We will be referring to the MotorWare lab software (available here: http://www.ti.com/tool/motorware) and guide (located here: C:\ti\motorware\motorware_1_01_00_11\docs\labs) as well as the InstaSPIN-MOTION User's Guide (available here: http://www.ti.com/lit/pdf/spruhj1) throughout this document.

3 Setup

Running in position control requires that the motor and encoder are setup to work together. In this section we describe the steps required to let these components work together.

3.1 Motor

3.1.1 Physical Connection

It is important that the motor phases are connected in the correct order. If they are not connected in the right order, the motor will not be able to run using the encoder for electrical angle feedback. The motor should be connected with the phases in sequential order. This information is typically available on the motor or via the motor datasheet. If you cannot find the motor phase order, we will determine it when we identify the motor parameters.

3.1.2 Software

We need to create an entry for the target motor in the user.h file. This file is located in a directory based on the hardware kit that you are using.

Example:
C:\ti\motorware\motorware_1_01_00_11\sw\solutions\instaspin_motion\boards\<kit>\f28x\f2806xM\src

An example of what this entry should look like is included in Figure 1.

```
#elif (USER_MOTOR == YOUR_MOTOR)
#define USER_MOTOR_TYPE                 MOTOR_Type_Pm
#define USER_MOTOR_NUM_POLE_PAIRS       (4)
#define USER_MOTOR_Rr                   (NULL)
#define USER_MOTOR_Rs                   (0.4110007)
#define USER_MOTOR_Ls_d                 (0.0007092811)
#define USER_MOTOR_Ls_q                 (0.0007092811)
#define USER_MOTOR_RATED_FLUX           (0.03279636)
#define USER_MOTOR_MAGNETIZING_CURRENT  (NULL)
#define USER_MOTOR_RES_EST_CURRENT      (1.0)
#define USER_MOTOR_IND_EST_CURRENT      (-1.0)
#define USER_MOTOR_MAX_CURRENT          (5.0)
#define USER_MOTOR_FLUX_EST_FREQ_Hz     (20.0)
#define USER_MOTOR_ENCODER_LINES        (2000.0)
#define USER_MOTOR_MAX_SPEED_KRPM       (4.0)
#define USER_SYSTEM_INERTIA             (0.02)
#define USER_SYSTEM_FRICTION            (0.01)
```

Figure 2: Example of a Motor Entry in user.h

If you don't know the values for all of these parameters we will be identifying the motor parameters in the first step.

The other consideration in the user.h is the base frequency. This is set by USER_IQ_FULL_SCALE_FREQ_Hz. This setting is used to scale the position and velocity signals from real units to scaled units. It is important that this value is not too large for position control. This value should be set according to the following inequality.

$$\text{USER\_IQ\_FULL\_SCALE\_FREQ\_Hz} \leq 125 * \text{USER\_MOTOR\_NUM\_POLE\_PAIRS}$$

If the value in USER_IQ_FULL_SCALE_FREQ_Hz is not set according to the inequality, it will result in configuration errors in the SpinTAC components.

3.2 Encoder

3.2.1 Physical Connection

For the encoder it is important that it is connected A to A, B to B, and I to I. The pinout for the TI motor control evaluation kits is provided in Table 1.

Table 1: Pinout for Encoder Connection

Pin	Signal
1	A
2	B
3	I
4	+5 Volt
5	Ground

For the DRV8312 and DRV8301 kits the header is J4. For the High Voltage Kit the connection is the top row of H1. If these connections are not in the proper sequence, the motor will not spin.

3.2.2 Software

The number of encoder lines is configured by setting the appropriate value in USER_MOTOR_ENCODER_LINES. An example is provided in Figure 1. Note that this value should be set to the number of lines on the encoder also referred to as pulses per revolution, not the resultant number of counts after figuring the quadrature accuracy.

It is important that the angle of the quadrature encoder is aligned with the electrical angle of the motor. This is accomplished in the example MotorWare projects by forcing the rotor to zero degree electrical angle during the stator resistance measurement. This calibrates the encoder counts to the electrical angle of the motor. This procedure does require that the motor will move in order to align along a zero degree electrical

angle. If the motor cannot move to perform this alignment, then other arrangements need to be provided in order to align the quadrature encoder angle with the motor electrical angle.

4 Motor Parameter Identification

The first step when working with an unknown motor is to identify the motor parameters. This will ensure that the motor and hardware are functioning properly. This step will also automatically tune the current regulators. You can bypass this step if you want to manually tune the current regulators.

There are three options for identifying the motor parameters:

- InstaSPIN-MOTION GUI (available here: http://www.ti.com/tool/motorkitscncd69miso)
  - Follow instructions in the InstaSPIN-MOTION Quick Start Guide
- Lab 02a
  - If your motor parameter identification fails, attempt to use Lab 02c which is designed for low inductance motors.
  - Follow instructions in the MotorWare lab guide
- Motor Datasheet
  - Follow instructions in the InstaSPIN-MOTION User's Guide Section 6.8.1

During the motor parameter identification the motor must spin in the anti-clockwise direction in order to align the motor and encoder. If the motor spins clockwise during the motor parameter identification, swap any two motor phase lines and repeat the process.

Once the motor parameters have been identified they need to be stored in the user.h file. This file is located in a directory based on that hardware kit that you are using.

Example:
C:\ti\motorware\motorware_1_01_00_11\sw\solutions\instaspin_motion\boards\<kit>\f28x\f2806xM\src

Table 1 summarizes the motor parameters that are required in the user.h header file for PMSM motors.

Table 2: PMSM Motor Parameters in user.h

PMSM Motor Parameter in user.h	PMSM Motor Parameter and Units
USER_MOTOR_Rs	Stator Resistance (Ω)
USER_MOTOR_Ls_d	Stator Direct Inductance (H)
USER_MOTOR_Ls_q	Stator Quadrature Inductance (H)
USER_MOTOR_RATED_FLUX	Rated Flux (V/Hz)

There are some additional motor & system parameters that need to be provided.

- USER_MOTOR_ENCODER_LINES
  - Set this value equal to the number of lines on the encoder wheel also called pulses per revolution. This will be used to setup the encoder.
- USER_MOTOR_MAX_SPEED_KRPM
  - Set this to the maximum rated speed of the motor. It will be used to establish maximum limits for the profile generator.
- USER_SYSTEM_INERTIA
  - Set this to the default value of 0.02 (we will update this in the next step)
- USER_SYSTEM_FRICTION
  - Set this to the default value of 0.01 (we will update this in the next step)

Once these parameters have been placed in the user.h you are ready to identify the system inertia.

5 Inertia Identification

Once the motor parameters have been identified and the current controllers are tuned you need to identify the system inertia and friction. Inertia is the resistance to rotational acceleration. In order to identify the system inertia you need to couple the motor to the application, specifically anything that is rigidly coupled with a motor shaft. The system inertia is a critical piece of information to achieve the great level of control provided by InstaSPIN-MOTION.

InstaSPIN-MOTION uses the same inertia value for velocity and position control.

Run Lab 05c to identify the system inertia. Follow the instructions in the MotorWare lab guide.

Once you have completed Lab 05c, place the identified inertia and friction values into the user.h.

6 Encoder Setup

We need to confirm that the encoder is functioning properly, and that it can generate the electrical angle needed to run the FOC. Since the encoder will be generating the electrical angle for the FOC it is important that the pole pairs for the motor are correct.

We will do this step with Lab 12. Follow the instructions in the MotorWare lab guide to run this lab.

If the motor does not spin with this lab, double check the motor phase order and the encoder wiring.

If the motor phase order and the encoder wiring look correct, we can manually confirm that the encoder is working.

- Add “st_obj.vel.conv.Pos_erev” & “st_obj.vel.conv.Pos_mrev” into the watch window
- Set “gMotorVars.Flag_enableSys” to 1.
- Manually rotate the motor in the anti-clockwise direction for one mechanical revolution and watch the value in “st_obj.vel.conv.Pos_erev”.
    - If the value decreases than the encoder wiring is incorrect, A and B should be switched
- The value in “st_obj.vel.conv.Pos_mrev” should be approximately equal to 1
    - If the value is not 1 then the number of pole pairs in the motor is incorrect or the number of encoder lines is incorrect.

7 Position Control

We have identified the motor parameters, the system inertia, and confirmed that the encoder is operational. At this point we can run under position control.

Instructions for tuning the InstaSPIN-MOTION Position Controller are available in Lab 13a. Once you have found a good tuned value for your system you can place the tuning value in the user.h.

7.1 Lab Synopsis

- Lab 13a –Tune the InstaSPIN-MOTION Position Controller

Tuning position control applications can be very difficult and time consuming. InstaSPIN-MOTION provides a position-velocity controller that can be tuned using a single coefficient. This single gain (bandwidth) typically works across the entire range of loads and transitions in applications, reducing their complexity. This lab demonstrates how to connect the InstaSPIN-MOTION position controller and tune it for your application.

- Lab 13b – Smooth Position Transitions with SpinTAC™ Move

InstaSPIN-MOTION includes SpinTAC Move, a motion profile generator that generates constraint-based, time-optimal position trajectory curves. It removes the need for lookup tables, and runs in real time to generate the desired motion profile. This lab will demonstrate the different configurations and their impact on the final position transition of the motor.

- Lab 13c – Motion Sequence Position Example

InstaSPIN-MOTION includes SpinTAC Plan, a motion sequence planner that allows you to easily build complex motion sequences. You can use this functionality to quickly build your application's motion sequence and speed up development time. This lab provides a very simple example of a motion sequence.

- Lab 13d – Motion Sequence Real World Example: Vending Machine

This lab builds off Lab 13c and provides a very complex example of a motion sequence.

- Lab 13e - Smooth Velocity Transitions in Position Control

In addition to providing smooth position transitions, SpinTAC Move can also provide smooth speed transitions while still operating in a position control system. This lab demonstrates how to configure SpinTAC Move to generate speed transitions in position mode.