# TMS320F28069M, TMS320F28068M InstaSPIN™-MOTION Software

# Technical Reference Manual

Literature Number: SPRUHJ0B
April 2013–Revised March 2014

# Contents

# List of Figures

# List of Tables

# TMS320F28069M, TMS320F28068M InstaSPIN™-MOTION Software

## 1 F2806xM InstaSPIN™-MOTION Enabled MCUs

InstaSPIN-MOTION [TMS320F2806xM (69M and 68M - 80 or 100 pin packages)] is the first offering from Texas Instruments to combine TI 32-bit C2000™ Piccolo™ microcontrollers with comprehensive motor-, motion-, speed-, and position-control software. InstaSPIN-MOTION delivers robust velocity and position control at the highest efficiency for motor applications that operate in various motion state transitions. InstaSPIN-MOTION is your own motion control expert, on a single chip.

InstaSPIN-MOTION is a sensorless or sensored field-oriented motor control (FOC) solution that can identify, tune, and control your motor in minutes. InstaSPIN-MOTION features the FAST™ premium software sensor and the SpinTAC™ Motion Control Suite (Figure 1). The core algorithms are embedded in the read-only-memory (ROM) on TI 32-bit C2000 Piccolo microcontrollers (MCUs).

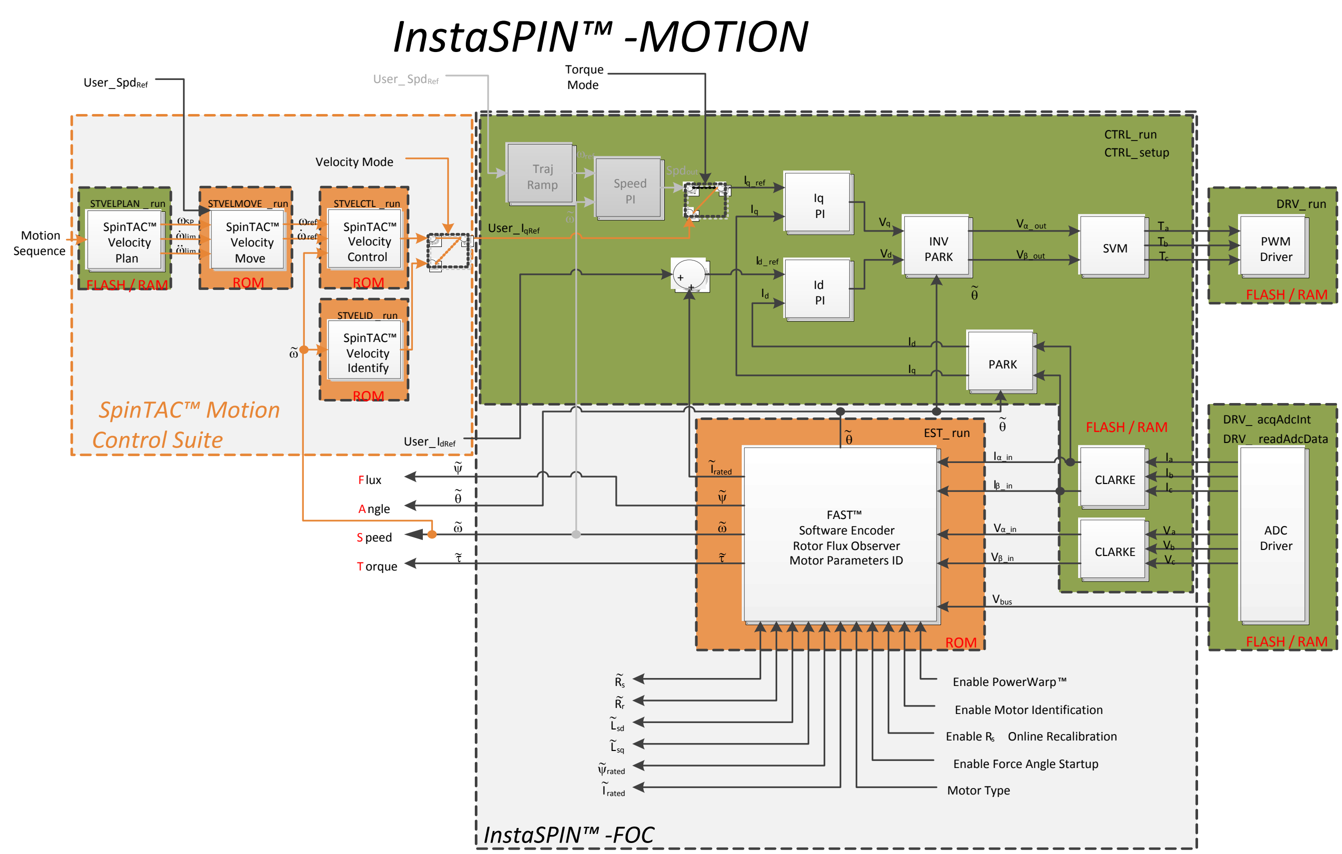


**Figure 1. InstaSPIN-MOTION = C2000 F2806xM MCU + FAST Software Sensor (optional) + Auto-Tuned Inner-Torque Controller + SpinTAC Motion Control Suite**

InstaSPIN, C2000, Piccolo, FAST, PowerWarp are trademarks of Texas Instruments.
SpinTAC is a trademark of LineStream Technologies.
All other trademarks are the property of their respective owners.

InstaSPIN-MOTION is ideal for applications that require accurate speed and position control, minimal disturbance, or undergo multiple state transitions or experience dynamic speed or load changes.

Table 1 provides examples of applications that will most benefit from InstaSPIN-MOTION.

**Table 1. InstaSPIN-MOTION Application Examples**

| Application Characteristics | Examples |
|---|---|
| Accurate speed control | Industrial fans<br>Conveyor systems<br>Elevators/escalators<br>Automotive body parts (electric windows, sunroofs, etc.)<br>Optical disc drives/hard drives<br>Medical mixing |
| Accurate position control | Surveillance systems<br>Packaging systems<br>Medical robots<br>Gimbal systems<br>Textile/Sewing machines |
| Minimal disturbance | Dental tools<br>Power tools<br>Security gates and doors |
| Undergoes multiple state transitions/dynamic changes | HVAC pumps, fans and blowers<br>Generators<br>Air conditioning compressors<br>Washing machines<br>Exercise equipment<br>Medical pumps |

This document focuses on the additive features provided in InstaSPIN-MOTION and is a supplement to all standard TMS320F2806x documentation, including:

**SPRS698** — ***TMS320F28069, TMS320F28068, TMS320F28067, TMS320F28066, TMS320F28065, TMS320F28064, TMS320F28063, TMS320F28062 Piccolo Microcontrollers Data Manual.*** Contains the pinout, signal descriptions, as well as electrical and timing specifications for the 2806x devices.

**SPRUH18** — ***TMS320x2806x Piccolo Technical Reference Manual.*** Details the integration, the environment, the functional description, and the programming models for each peripheral and subsystem in the device.

**SPRU566** — ***TMS320x28xx, 28xxx DSP Peripheral Reference Guide.*** Describes the peripheral reference guides of the 28x digital signal processors (DSPs).

Additionally, the InstaSPIN-MOTION documentation package includes:

**SPRUHJ1** — ***TMS320F2802xF InstaSPIN-FOC, TMS320F2806xF InstaSPIN-FOC, TMS320F2806xM InstaSPIN-MOTION User's Guide.*** Covers the scope and functionality of:

- F2806xM devices
- F2806xM ROM contents
- InstaSPIN-MOTION system solutions.

## 2 InstaSPIN-MOTION Key Capabilities and Benefits

InstaSPIN-MOTION replaces inefficient, older design techniques with a solution that maximizes system performance and minimizes design effort. By embedding the motor expertise on the chip, InstaSPIN-MOTION enables users to focus on optimizing their application rather than struggling with motion control.

InstaSPIN-MOTION provides the following core capabilities:

- The FAST unified 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 observer and initialize the innermost current (torque) PI controllers for Iq and Id control of the FOC system.
- SpinTAC, a comprehensive motion control suite (see Figure 2) from LineStream Technologies, simplifies tuning and ensures optimal performance across dynamic speed and position ranges.

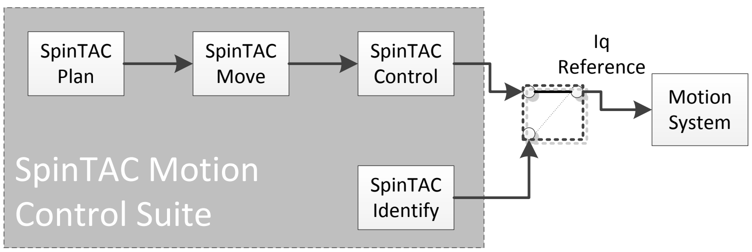


**Figure 2. SpinTAC Motion Control Suite Components**

### 2.1 FAST Unified Observer

FAST unified observer structure exploits the similarities between all motors that use magnetic flux for energy transduction:

- Supports both synchronous (BLDC, SPM, IPM) and asynchronous (ACIM) control.
- Provides salient compensation for interior permanent magnet motors. Observer tracks rotor flux and angle correctly when Ls-d and Ls-q are provided.

FAST offers unique, high-quality motor feedback signals for control systems:

- High-quality **F**lux signal for stable flux monitoring and field weakening.
- Superior rotor flux **A**ngle estimation accuracy over wider speed range compared to traditional observer techniques independent of all rotor parameters for ACIM.
- Real-time low-noise motor shaft **S**peed signal.
- Accurate high-bandwidth **T**orque signal for load monitoring and imbalance detection.

FAST replaces mechanical encoders and resolvers, and accelerates control system design:

- Angle estimator converges within first cycle of the applied waveform, regardless of speed.
- Stable operation in all power quadrants, including generator quadrants.
- Accurate angle estimation at steady state speeds below 1 Hz (typ) with full torque.
- Angle integrity maintained even during slow speed reversals through zero speed.
- Angle integrity maintained during stall conditions, enabling smooth stall recovery.
- Motor Identification measures required electrical motor parameters of unloaded motor in under 2 minutes (typ).
- "On-the-fly" stator resistance recalibration (online Rs) tracks stator resistance changes in real time, resulting in robust operation over temperature. This feature can also be used as a temperature sensor of the motor's windings (basepoint calibration required).
- Superior transient response of rotor flux angle tracking compared to traditional observers.
- PowerWarp™ adaptively reduces current consumption to minimize the combined (rotor and stator) copper losses to the lowest, without compromising ACIM output power levels.

### 2.2 SpinTAC Motion Control Suite

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:

- **Simplified Tuning** - Tune your system for the entire position and speed operating range with a single parameter.
- **Intuitive Trajectory Planning** - Easily design and execute complex motion sequences.
- **Mechanically Sound Movement -** Optimize your transitions between speeds based on your system's mechanical limitations.
- **Ideal Control -** Benefit from the most accurate speed and position control on the market, based on LineStream's patented Active Disturbance Rejection Control.

There are four components that comprise the SpinTAC Motion Control Suite: Identify, Control, Move, and Plan. Each of these components exist for both the Velocity and Position solution.

**IDENTIFY**

SpinTAC Identify estimates inertia (the resistance of an object to rotational acceleration around an axis). The greater the system inertia, the greater the torque needed to accelerate or decelerate the motor. The SpinTAC speed controller uses the system's inertia value to provide the most accurate system control. SpinTAC Identify automatically measures system inertia by spinning the motor in the application and measuring the feedback.

**CONTROL**

SpinTAC Control is an advanced speed and position controller featuring Active Disturbance Rejection Control (ADRC), which proactively estimates and compensates for system disturbance, in real-time. SpinTAC automatically compensates for undesired system behavior caused by:

- Uncertainties (for example, resonant mode)
- Nonlinear friction
- Changing loads
- Environmental changes.

SpinTAC Control presents better disturbance rejection and trajectory tracking performance than a PI controller, and can tolerate a wide range of inertia change. This means that SpinTAC improves accuracy and system performance, and minimizes mechanical system duress.

With single coefficient tuning, the SpinTAC controller allows users to quickly test and tune their velocity and position control from soft to stiff response. This single gain (bandwidth) typically works across the entire variable speed, position and load range of an application, reducing complexity and system tuning time typical in multi-variable PI-based systems. A single parameter controls both position and speed. These systems often require a dozen or more tuned coefficient sets to handle all possible dynamic conditions.

The InstaSPIN-MOTION (F2806xM) graphical user interface (GUI) (see Figure 3), in conjunction with the InstaSPIN-MOTION Quick Start Guide, allows users to quickly evaluate InstaSPIN-MOTION (speed control) using TI's evaluation kits and the TI provided motors, or their own motor. The GUI is designed to quickly guide you through the InstaSPIN-MOTION evaluation process. You can obtain the GUI, free of charge, from www.ti.com/tool/motorkitscncd69miso. Once you determine that InstaSPIN-MOTION is right for your application, use the MotorWare-based projects, in conjunction with the *TMS320F2802xF InstaSPIN-FOC, TMS320F2806xF InstaSPIN-FOC, TMS320F2806xM InstaSPIN-MOTION User's Guide* (SPRUHJ1) to design your project and conduct performance testing.

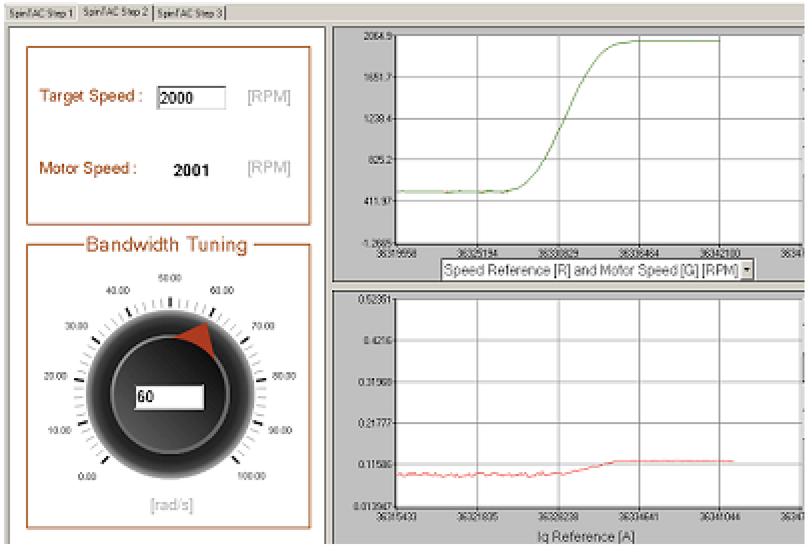


**Figure 3. Simple Tuning Interface**

### MOVE

SpinTAC Move provides an easy way to smoothly transition from one speed or position to another by computing the fastest path between Point A and Point B. SpinTAC Move generates a profile based on starting velocity or position, desired velocity or position, and configured system limitations for acceleration and jerk. Jerk represents the rate of change of acceleration. A larger jerk will increase the acceleration at a faster rate. Steps, or sharp movement between two points, can cause systems to oscillate. The bigger the step, the greater this tendency. Control over jerk can round the velocity corners, reducing oscillation. As a result, acceleration can be set higher. Controlling the jerk in your system will lead to less mechanical stress on your system components and can lead to better reliability and less failing parts.

As opposed to pre-defined lookup tables, SpinTAC Move runs on the processor, consuming less memory than traditional solutions. Besides the industry standard trapezoidal curve and s-Curve, SpinTAC also provides a proprietary st-Curve, which is even smoother than s-Curve and allows users to limit the jerk of the motion.

Figure 4 describes the curves that are available for use in SpinTAC Move. The LineStream proprietary st-Curve provides the smoothest motion by smoothing out the acceleration of the profile. For most applications the st-Curve represents the best motion profile.

| Signals | Trapezoidal | s-Curve | st-Curve |
|---|---|---|---|
| Position | Smooth | Smooth | Smooth |
| Velocity | Continuous | Smooth | Smooth |
| Acceleration | Bounded | Continuous | Smooth |
| Jerk | Infinite | Bounded | Continuous |

**Figure 4. SpinTAC™ Move Curve Descriptions**

**PLAN**

SpinTAC Plan provides easy design and execution of complex motion sequences. The trajectory planning feature allows users to quickly build various states of motion (speed or position A to speed or position B) and tie them together with state-based logic. SpinTAC Plan can be used to implement a motion sequence for nearly any application. Figure 5 displays the motion sequence for a washing machine and Figure 6 displays the motion sequence for a garage door. Both of these were easily designed using SpinTAC Plan. Once designed, the trajectories are directly embedded into the C code on the microcontroller.

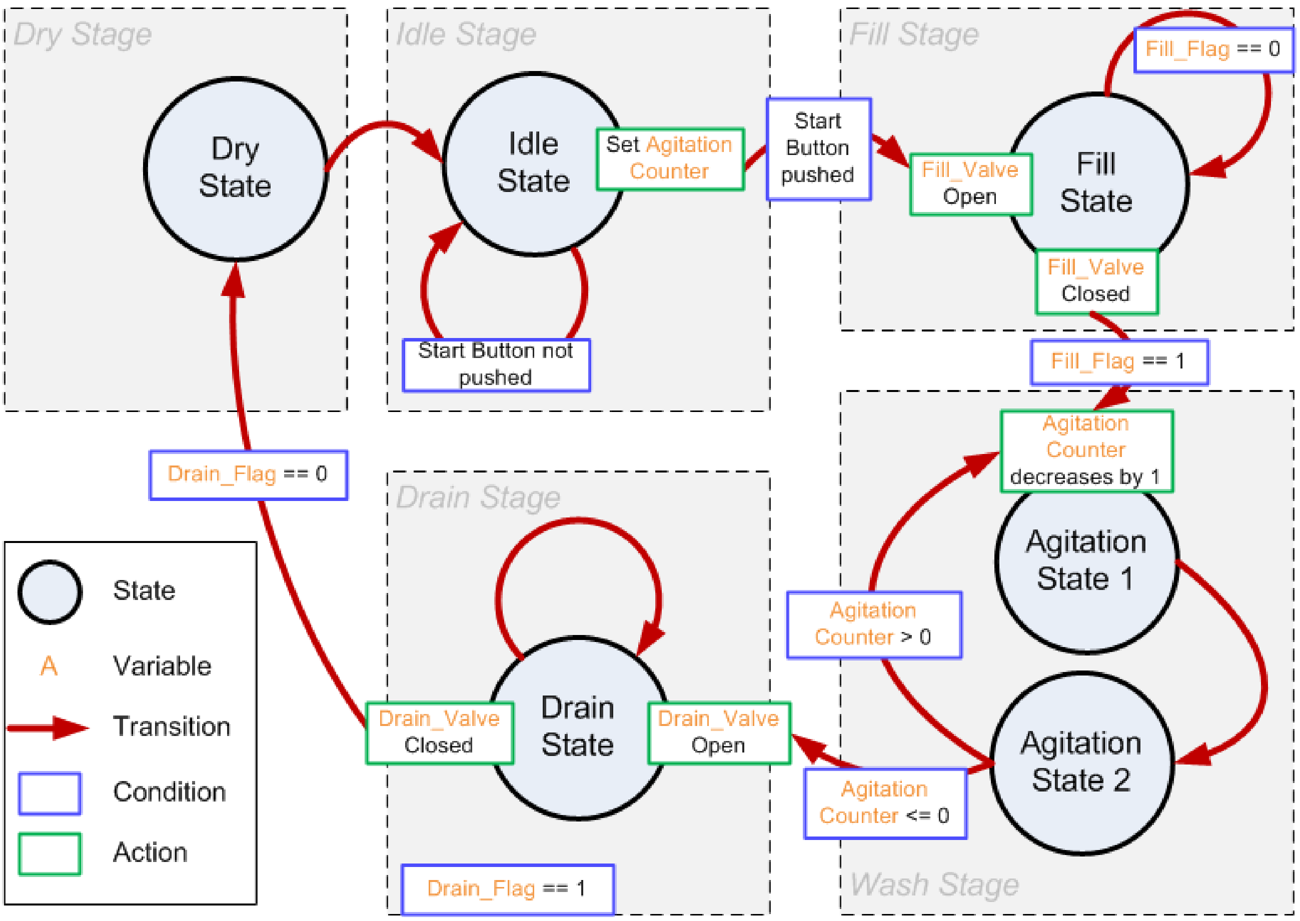


**Figure 5. State Transition Map for a Washing Machine**

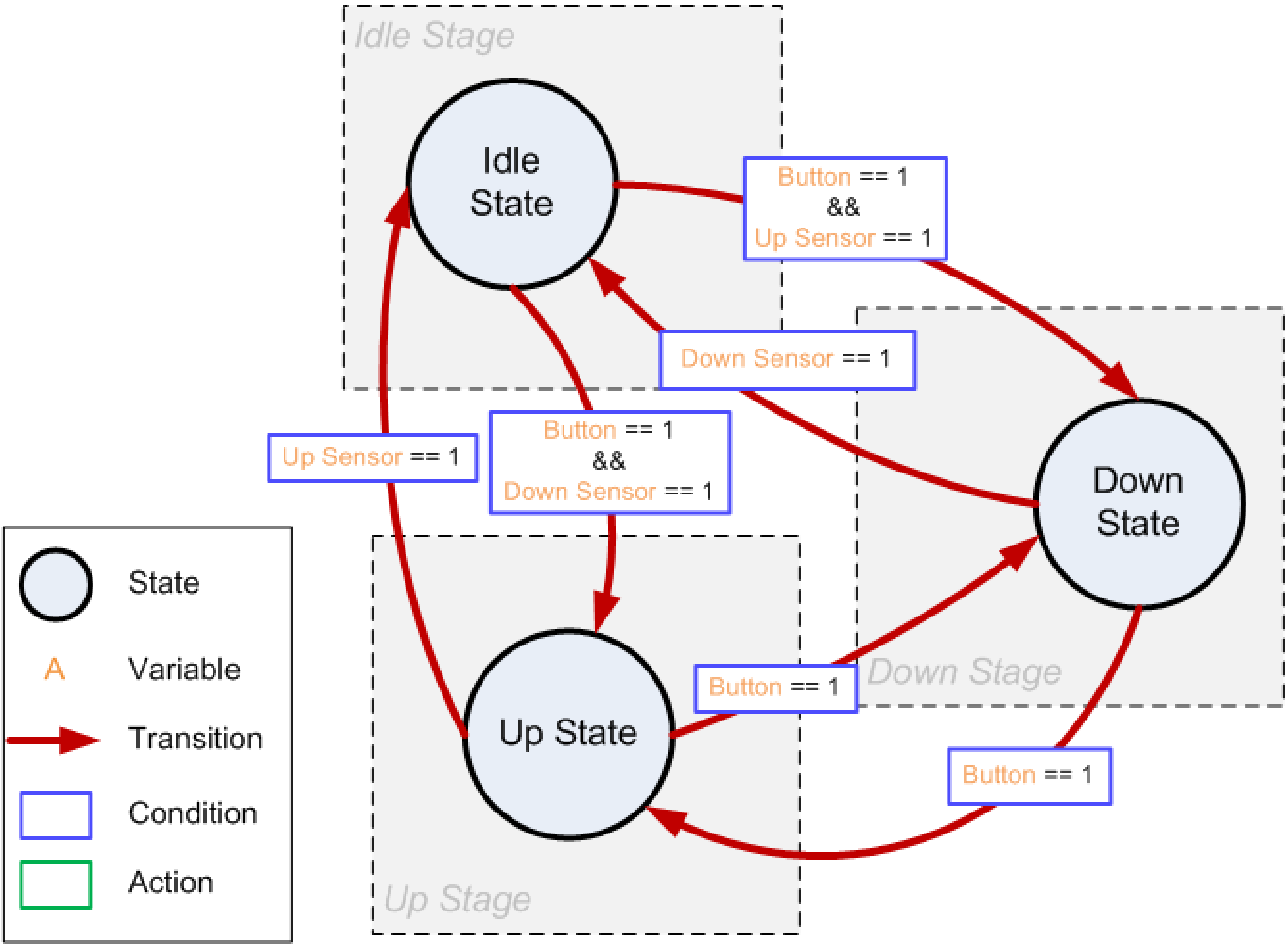


**Figure 6. State Transition Map for a Garage Door System**

## 2.3 Additional InstaSPIN-MOTION Features

- Automatic torque (current) loop tuning with option for user adjustments.
- Automatic or manual field weakening and field boosting.
- Bus voltage compensation.
- Automatic offset calibration ensures quality samples of feedback signals.

## 3 InstaSPIN-MOTION Block Diagrams

InstaSPIN-MOTION is designed in a modular structure. Customers can determine which functions will be included in their system. The FAST Observer resides in ROM. The core control algorithms of the SpinTAC library reside in ROM, and these functions are accessed by Application Program Interface (API) from the user code.

InstaSPIN-MOTION supports a wide array of system designs. InstaSPIN-MOTION uses the FAST software encoder for sensorless FOC systems [for additional information, see the *TMS320F2806xF InstaSPIN™-FOC Software Technical Reference Manual* (SPRUHI9)]. InstaSPIN-MOTION also supports solutions that leverage mechanical sensors (for example, encoders and resolvers). These scenarios are described below.

The variables in Figure 7, Figure 8, Figure 9, and Figure 10 are defined as follows:

- $\theta_{Qep}$: position angle signal from encoder
- $\theta_{SP}$: sawtooth position reference signal generated by SpinTAC Position Move
- $\omega_{lim}$: speed limit (used in position profile generation)
- $\dot{\omega}_{lim}$: acceleration limit
- $\ddot{\omega}_{lim}$: jerk limit
- $\omega_{Ref}$: speed reference
- $\dot{\omega}_{Ref}$: acceleration reference
- $\tilde{\tau}_r$: motor time constant

**Scenario 1: InstaSPIN-MOTION Speed Control with FAST Software Encoder**

In this scenario (see Figure 7 and Figure 8), SpinTAC Velocity Control receives the speed estimate from the FAST estimator and generates the torque reference signal. This works with InstaSPIN-FOC in user memory (see Figure 7) or in ROM (see Figure 8). The SpinTAC Motion Control Suite provides the motion sequence state machine, generates the reference trajectory and controls the system speed.

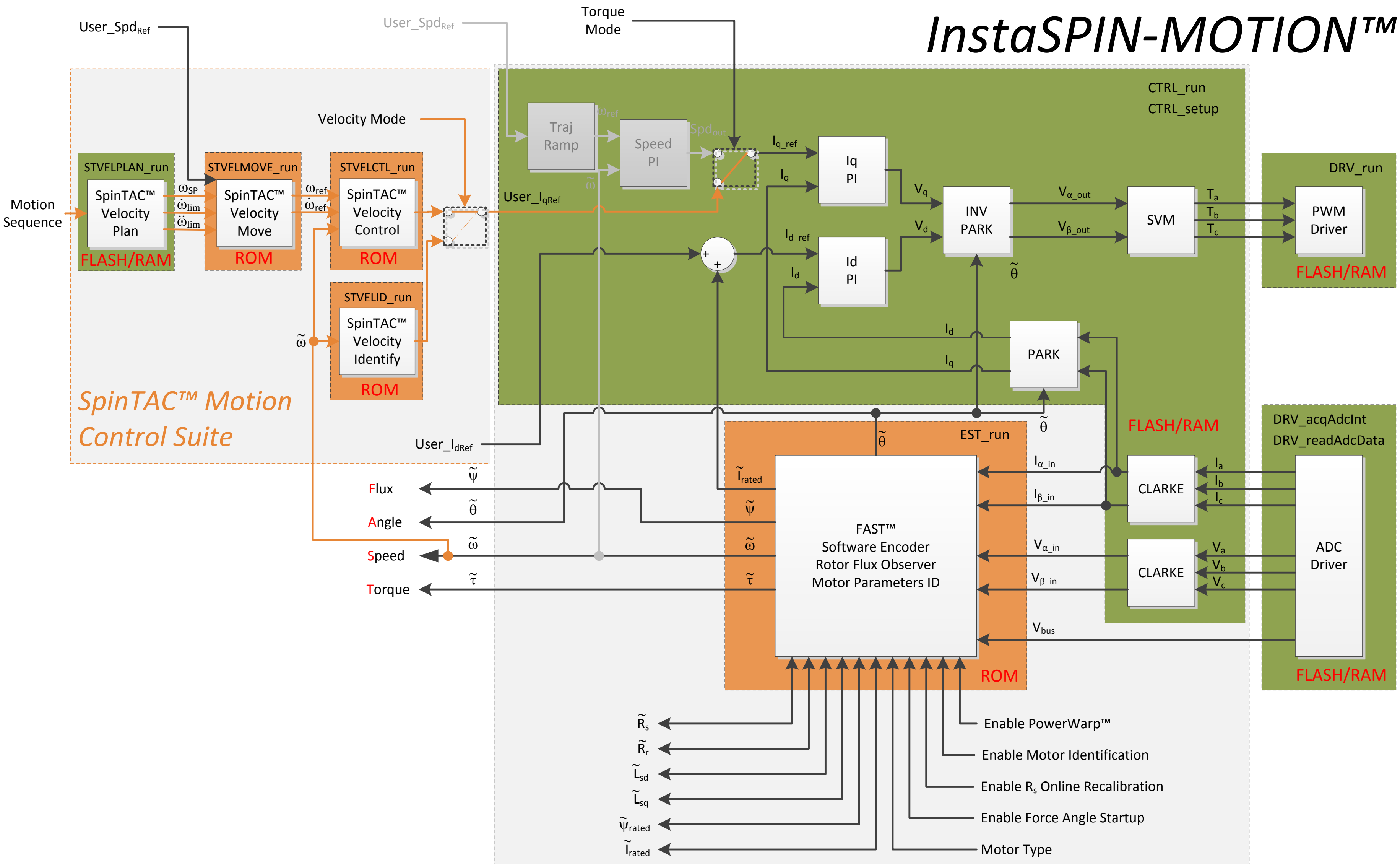


**Figure 7. InstaSPIN-MOTION in User Memory, with Exception of FAST and SpinTAC in ROM**

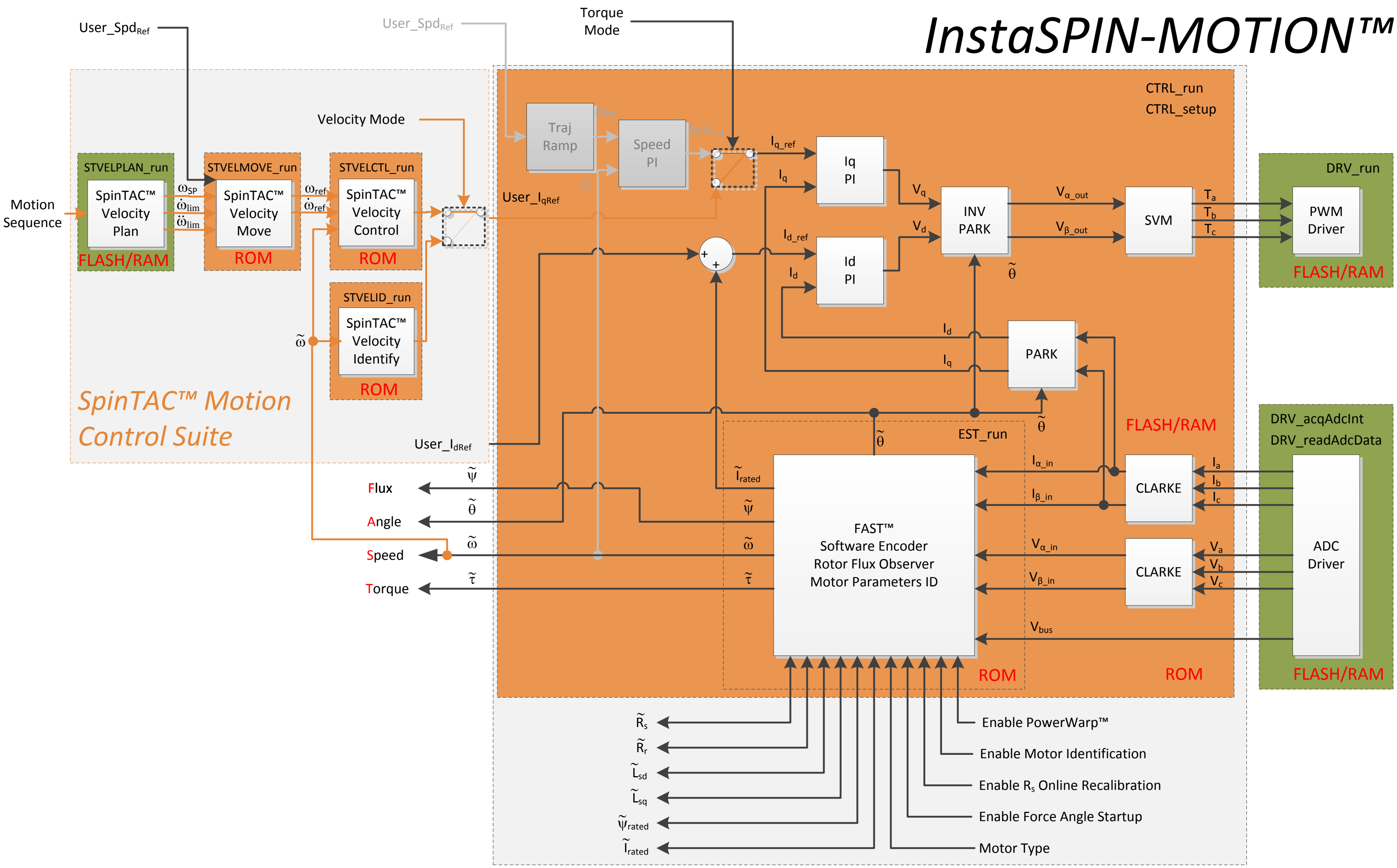


**Figure 8. InstaSPIN-MOTION in ROM**

### Scenario 2: InstaSPIN-MOTION Speed Control with a Mechanical Sensor

While sensorless solutions are appealing and cost effective for many applications, there are some applications that require the rigor and accuracy of a mechanical sensor. For these applications (see Figure 9), the quadrature encoder provides position information, which is then converted to speed feedback via the SpinTAC Position Convert. SpinTAC Velocity Control receives the speed feedback and generates the torque reference signal via IqRef. The SpinTAC Motion Control Suite provides the motion sequence state machine, generates the reference trajectory, and controls the system speed.

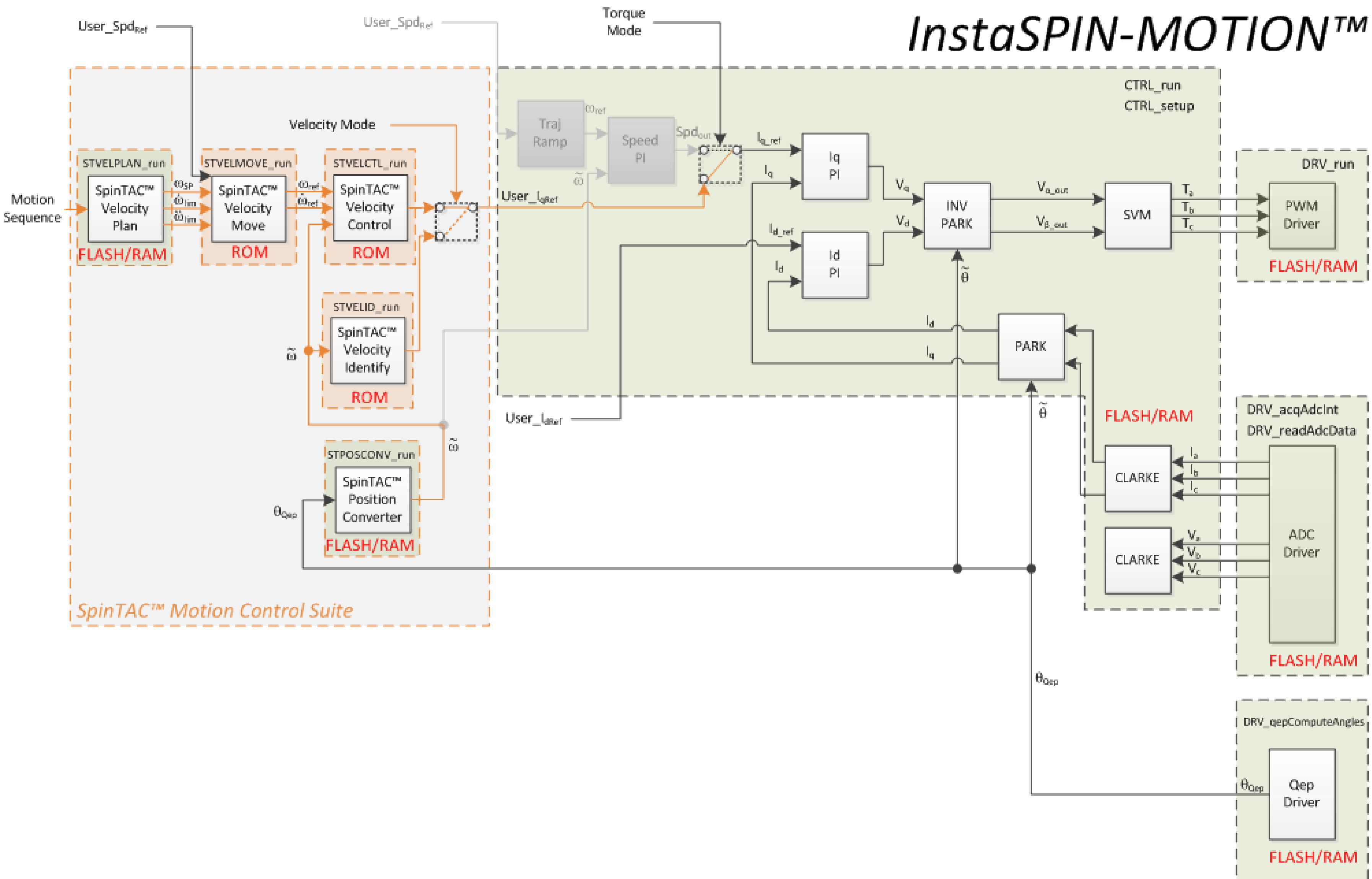


**Figure 9. InstaSPIN-MOTION Speed Control with a Mechanical Sensor**

### Scenario 3: InstaSPIN-MOTION Position Control with Mechanical Sensor and redundant FAST Software Sensor

There are many applications where precise position control is required. For these applications, it is difficult to balance the many tuning parameters that are required. InstaSPIN-MOTION features accurate position, speed, and torque control with combined position and speed single-variable tuning (see Figure 10). This simplifies the tuning challenge and allows you to focus on your application and not on tuning your motor. Position applications require a mechanical sensor in order to precisely identify the motor angle at zero and very low speeds. The FAST Software Encoder may provide redundancy in position control applications; this can be used as a safety feature in case the mechanical encoder fails.

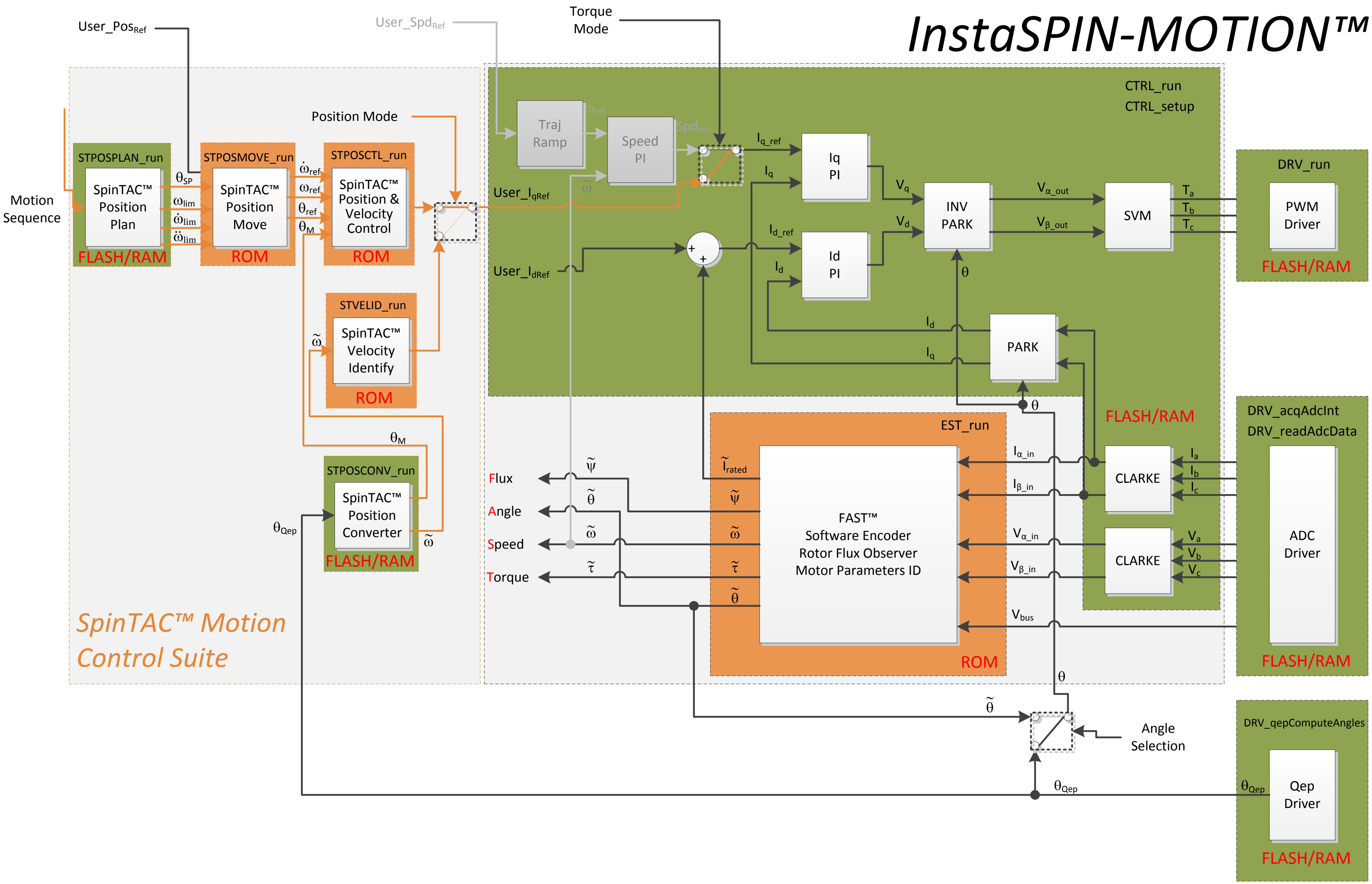


Figure 10. InstaSPIN-MOTION Position Control with Mechanical Sensor and Redundant FAST Software Sensor

# 4 Application Examples

InstaSPIN-MOTION is ideal for applications that require accurate speed and position control, minimal disturbance, and for applications that undergo multiple state transitions or experience dynamic changes. A few examples are provided in the following sections.

## *4.1 Treadmill Conveyor: Smooth Motion Across Varying Speeds and Loads*

Consistent speed control is critical for treadmill conveyor belts. When a person runs on a treadmill, their stride disturbs the motion of the belt. The runner's stride will be choppy if the motor driving the belt cannot quickly provide enough torque to overcome the disturbance. This problem is exacerbated when the user changes speeds as part of their exercise regime. If the belt does not smoothly accelerate or decelerate it seems like the treadmill is not operating correctly. In addition, at low speeds when a user steps on the belt, their weight can cause the belt to stop.

InstaSPIN-MOTION was applied to a commercial treadmill using a 4-HP, 220-V AC induction motor to drive the conveyor belt. The treadmill was tested across a variable speed range: 42 rpm at the low end to 3300 rpm at top speed.

The customer found that InstaSPIN-MOTION's advanced controller automatically compensated for disturbances, keeping the speed consistent while running and across changing speeds. The controller prevented the belt from stopping at low speeds when a load was applied. In addition, a single gain was used to control the entire operating range.

## *4.2 Video Camera: Smooth Motion and Position Accuracy at Low Speeds*

High-end security and conference room cameras operate at very low speeds (for example, 0.1 rpm) and require accurate and smooth position control to pan, tilt, and zoom. The motors that drive these cameras are difficult to tune for low speed, and they usually require a minimum of four tuning sets. In addition, there can be choppy movement at startup, which results in a shaky or unfocused picture.

InstaSPIN-MOTION was applied to a high-precision security camera driven by a 2-pole BLDC motor with a magnetic encoder. InstaSPIN-MOTION was able to control both velocity and position using a single tuning parameter that was effective across the entire operating range. SpinTAC Move was used to control the motor jerk, resulting in smooth startup.

## *4.3 Washing Machine: Smooth Motion and Position Accuracy at Low Speeds*

Cycle transitions, changing loads, and environmental disturbances cause significant wear and tear on motors. Automatic, real-time reduction of disturbances can extend the life and performance of motors.

Consider washing machines, for example. Figure 11 displays the motion profile for three stages of a standard washing machine. The first stage represents the agitation cycle, rotating between 250 rpm and -250 rpm, repeatedly. The second and third stages represent two different spin cycles. The second stage spins at 500 rpm and the third stage spins at 2000 rpm. This profile was easily created using SpinTAC Plan [as documented in the *TMS320F2802xF InstaSPIN-FOC, TMS320F2806xF InstaSPIN-FOC, TMS320F2806xM InstaSPIN-MOTION User's Guide* (SPRUHJ1)].

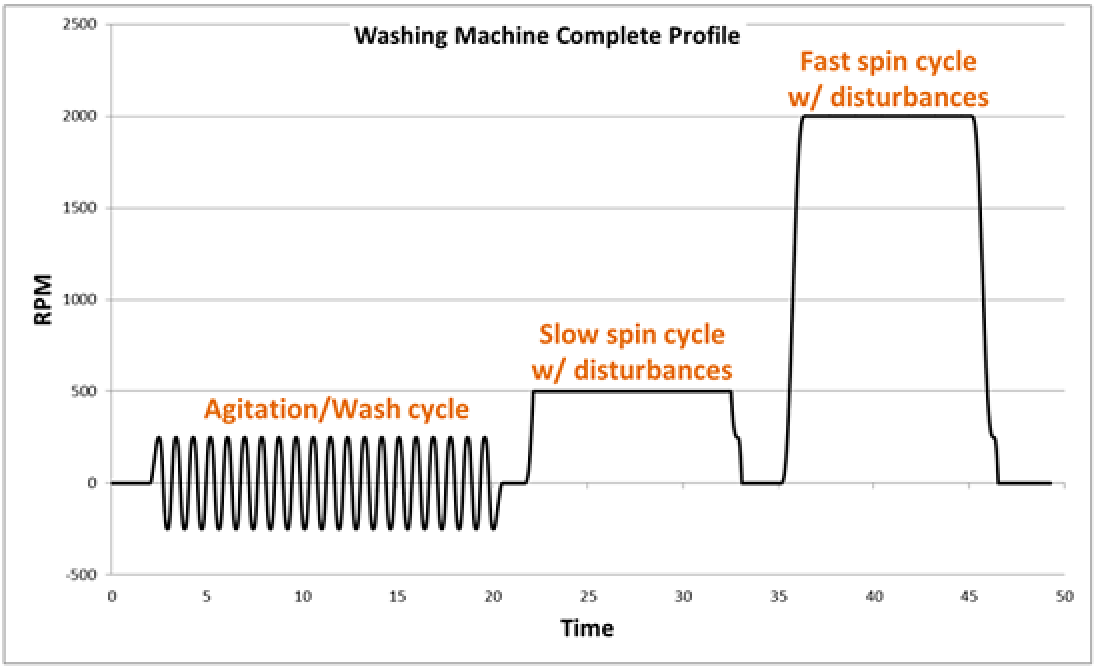


**Figure 11. Washing Machine Profile**

InstaSPIN-MOTION was applied to a washing machine application. The SpinTAC Plan trajectory planning feature was used to quickly build various states of motion (speed A to speed B) and tie them together with state based logic (see Figure 5).

The washing machine application was run twice — once using a standard PI controller and once using LineStream's SpinTAC controller. The data was then plotted against the reference curve for comparison.

### Agitation Cycle

During agitation, the motor switched between the 250 rpm and -250 rpm setpoints 20 times. The results, shown in Figure 12, demonstrate that InstaSPIN-MOTION more closely matched the reference profile. Additionally, the maximum error for PI was 91 rpm (341 - 250 = 91 rpm); whereas, the maximum error for SpinTAC was 30 rpm (280 - 250 = 30 rpm).

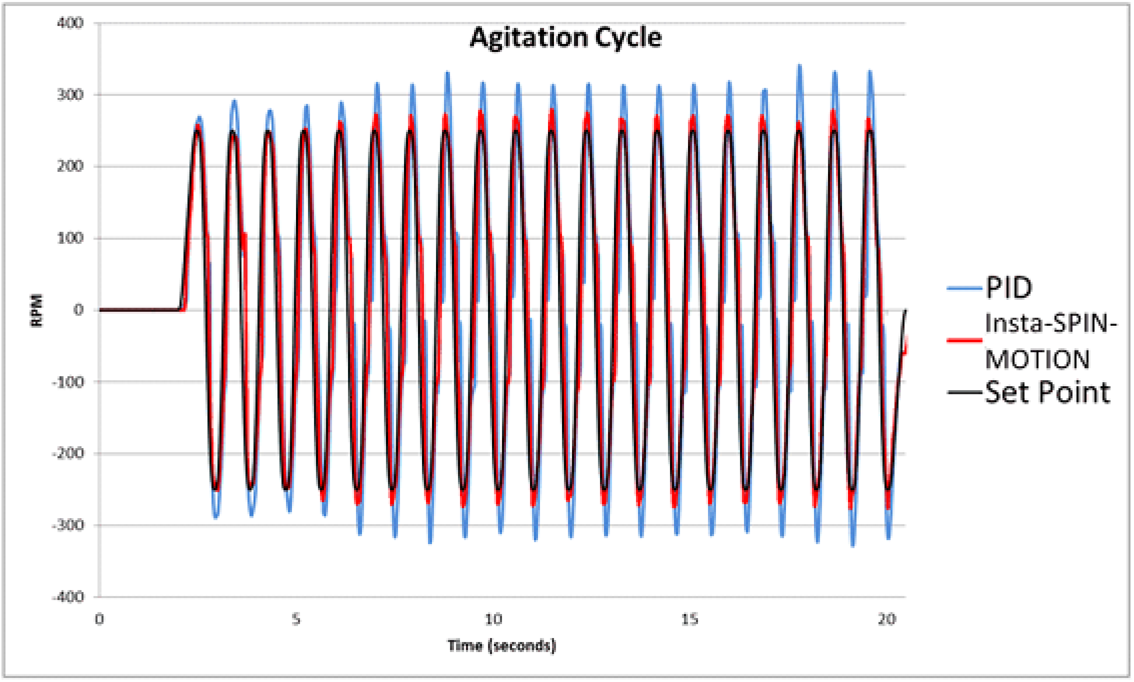


**Figure 12. InstaSPIN-MOTION Minimizes Error**

### Spin Cycles

For the first spin cycle, the objective is to maintain 500 rpm, even when disturbances are introduced. Figure 13shows that InstaSPIN-MOTION recovered from disturbances more quickly, and with less oscillation, than the PI controller. Additionally, InstaSPIN-MOTION does not suffer from the overshoot and undershoot shown by the PI controller when it tries to reach the initial 500 rpm setpoint.

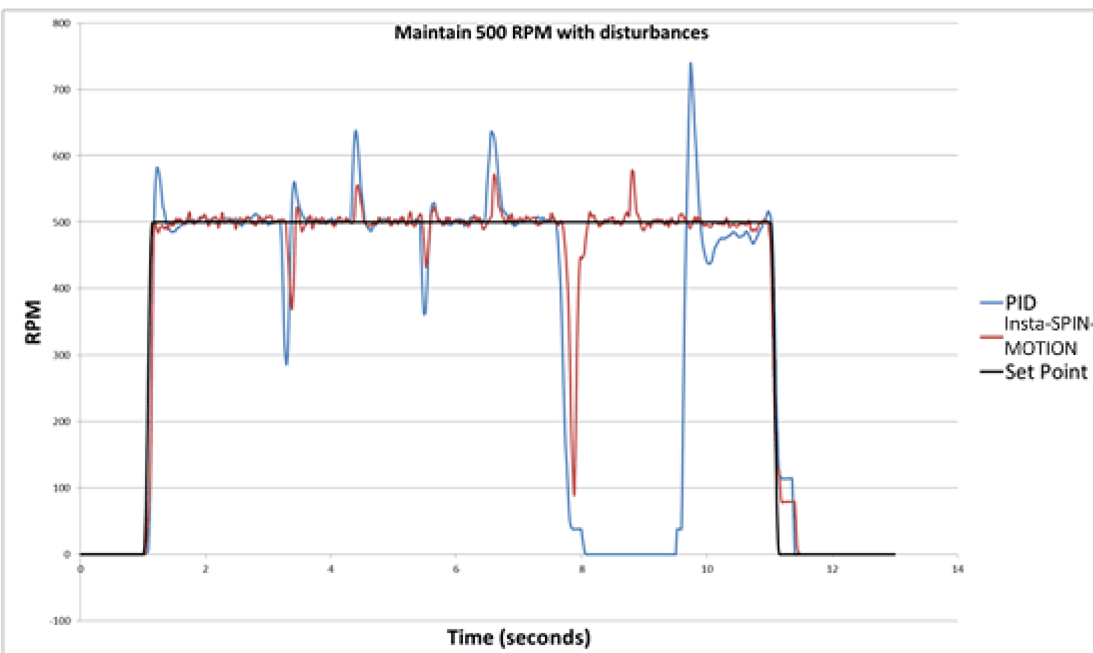


**Figure 13. First Spin Cycle - 500 rpm**

During the second spin cycle, shown in Figure 14, the SpinTAC controller consistently recovered from disturbances at 2000 rpm more quickly, and with less oscillation, than the PI controller. Note that SpinTAC does not suffer from the overshoot and undershoot shown by the PI controller when it tries to reach the initial 2000 rpm setpoint.

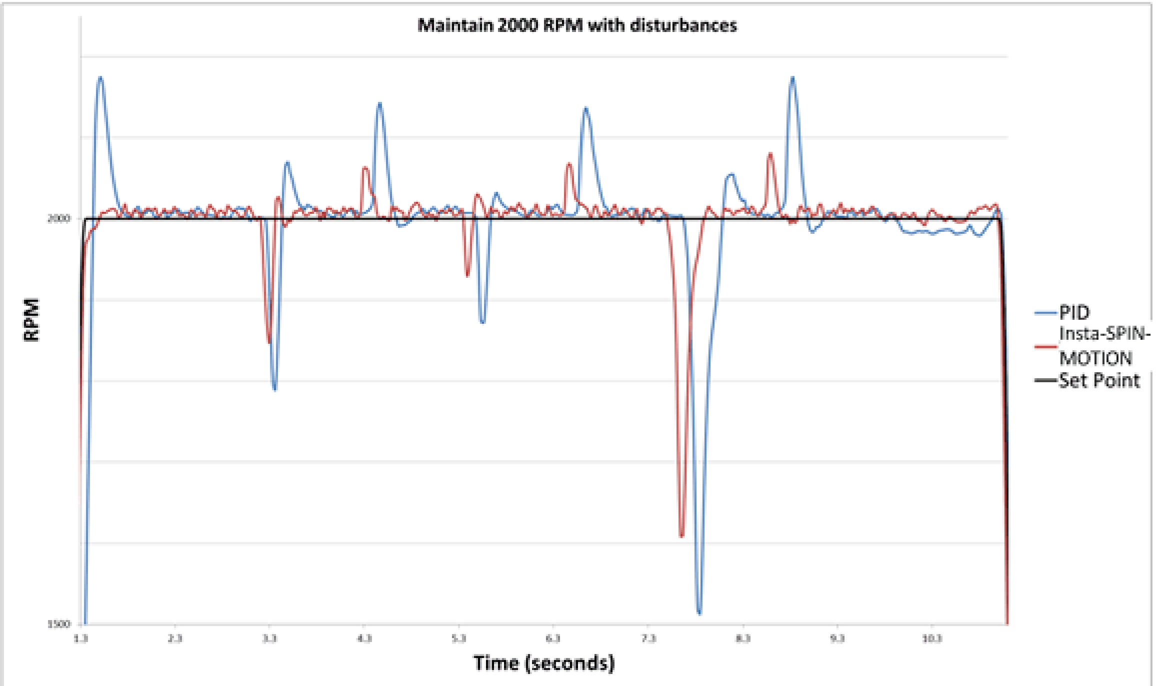


**Figure 14. Second Spin Cycle - 2000 rpm**

Additionally, the PI controller could not recover from the ramp disturbance at the 9.75 second mark. Instead, it shows a steady-state error of roughly 20 rpm.

## *4.4 InstaSPIN-MOTION Works Over the Entire Operating Range*

Both the InstaSPIN-MOTION controller and the PI controller were tuned once, before executing the washing machine application. From the examples shown above, it is evident that InstaSPIN-MOTION's tuning works over the entire operating range. Whether the motor switches between the 250 rpm and -250 rpm, or maintains 500 rpm or 2000 rpm spin cycles, there is no need for new tuning sets.

# 5 InstaSPIN-MOTION Replaces Hard-to-Tune PID Controllers

Large original equipment manufacturers (OEMs) may work with external design houses to optimize their tuning parameters. However, many small and medium size OEMs are challenged by tuning their own PI controls. With InstaSPIN-MOTION, OEMs can save weeks of tuning time.

While PI controls are versatile and applicable to numerous control systems, the tuning process can be complex and time-intensive. Similarly, performance of PI loops can be unpredictable over a wide operating range and as system dynamics change over time. While the use of advanced control techniques, such as model-based compensation, can address these challenges, they require significant time for implementation and testing. They also add complexity to the system, and can result in performance degradation when the system dynamics vary.

InstaSPIN-MOTION addresses these challenges by replacing traditional PI controls. The advantages are highlighted in Table 2. InstaSPIN-MOTION incorporates advanced control features, such as feed-forward, an observer, and can be tuned in a fraction of the time required to tuned PI control. Instead of having to tune multiple control parameters, the SpinTAC controller can be tuned by adjusting a single parameter for both position and speed. Once tuned, it performs across a wide-operating range.

**Table 2. PI vs InstaSPIN-MOTION**

| Topic | PI Controllers | InstaSPIN-MOTION |
|---|---|---|
| ***Performance*** | Unpredictable | Compensating |
| ***Tuning Parameters*** | Multiple | Single |
| ***Tuning Process*** | Complex | Simple - the identification process takes a few minutes |
| ***Startup*** | Difficult - requires control expertise | Simple - accomplished in three steps |
| ***Disturbance Recovery*** | Overshoot and undershoot when disturbances are introduced and during transitions | Advanced disturbance rejection holds setpoints more closely |
| ***Ongoing Maintenance*** | Requires retuning | None |

InstaSPIN-MOTION is designed to perform in all kinds of systems. In systems that remain unaffected by outside influences, the InstaSPIN-MOTION controller demonstrates its performance benefits by eliminating overshoot and undershoot during transitions.

In systems that are affected by outside disturbances, the InstaSPIN-MOTION controller provides advanced disturbance rejection, holding setpoints more closely than is achievable with standard PI control, as shown in Figure 15 and Figure 16.

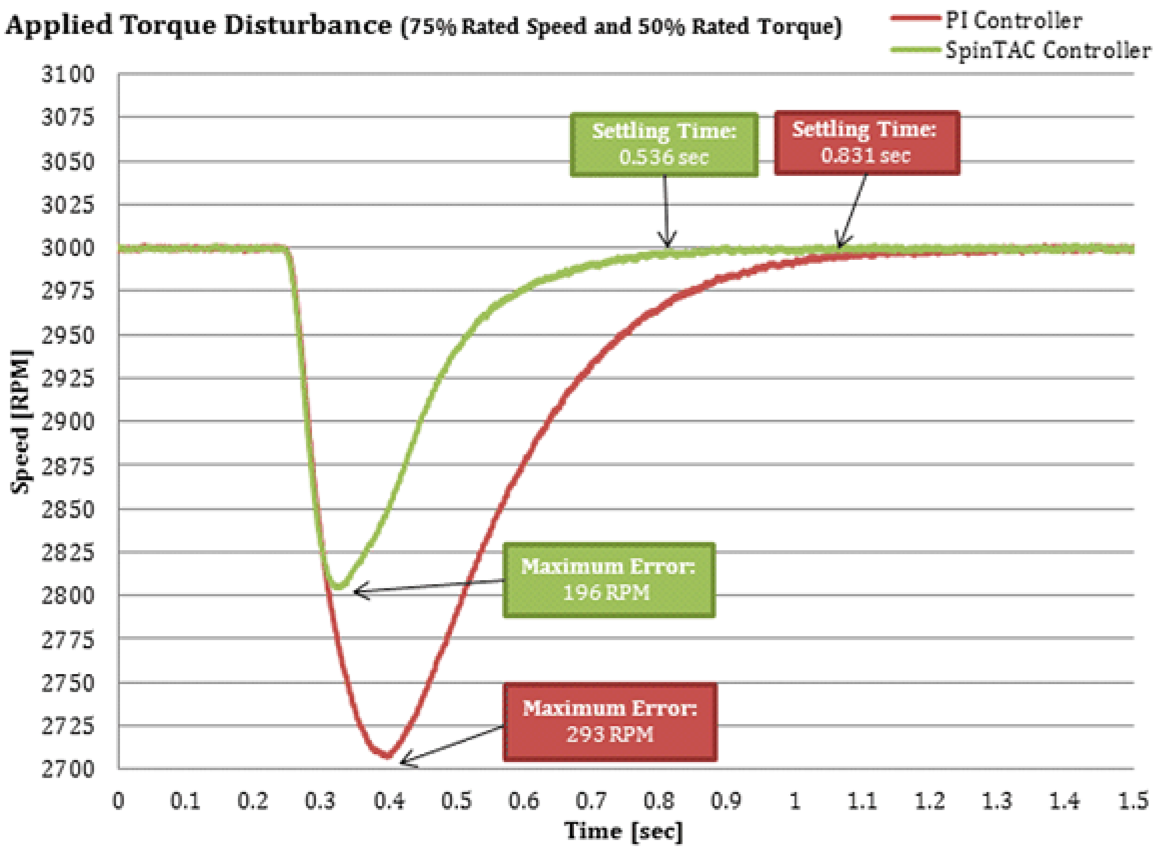


**Figure 15. Applied Torque Disturbance Comparison**

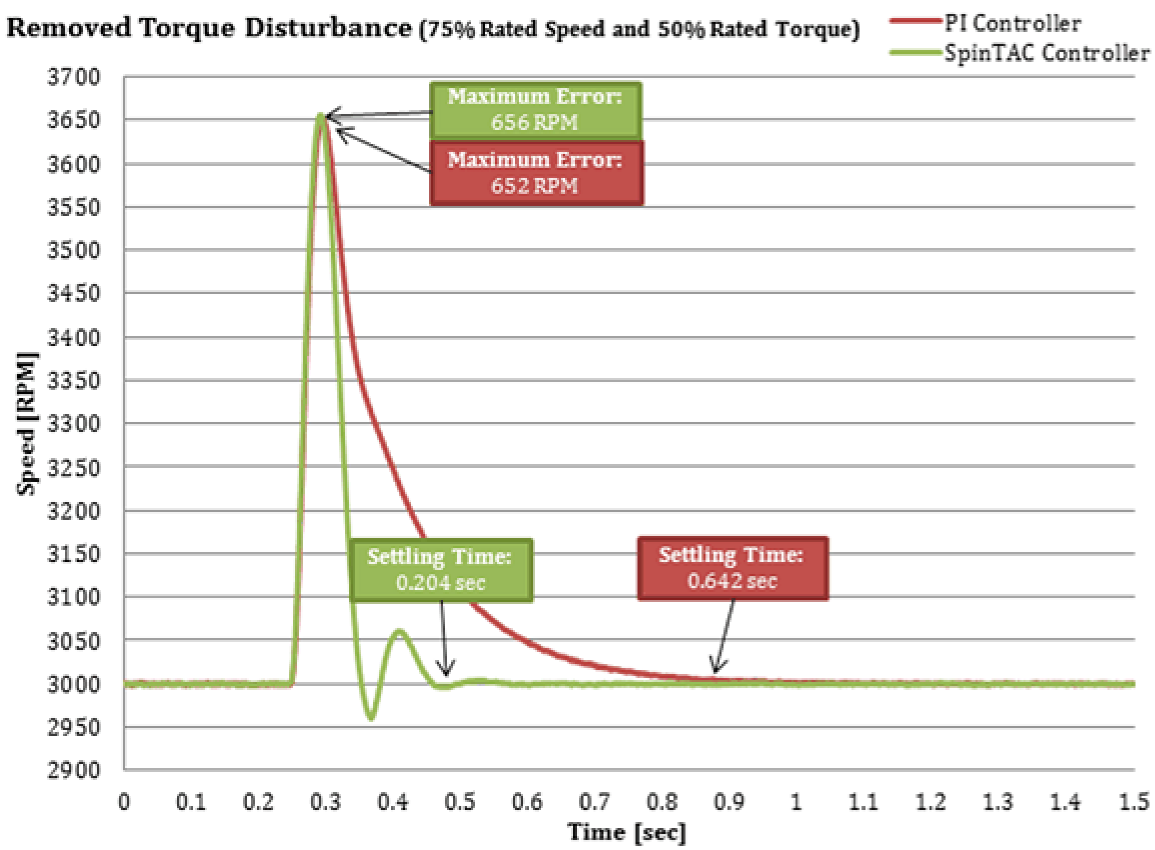


**Figure 16. Removed Torque Disturbance Comparison**

# 6 Evaluating InstaSPIN-MOTION Performance

Getting the best possible performance out of your motion system is important. A poorly tuned regulator can result in wasted energy, wasted material, or an unstable system. It is important that speed and position controller performance be evaluated at many different speed and load operating points in order to determine how well it works in your application.

Speed and position controllers can be compared on a number of different factors. However, two metrics — disturbance rejection and profile tracking — can be used to test performance and determine how well your controller is tuned for your application.

### Test-Bed Description

Tests were conducted using the following equipment:

- TMS320F28069M Control Card with InstaSPIN-MOTION Version 1.6
- Texas Instruments Code Composer Studio Version 5.3
- Texas Instruments Inverters:
  - DRV8301-69M-KIT
  - TMDSHVMTRPFCKit Version 1.1
- Motors:
  - Teknic M2310P
  - Estun EMJ-04APB22
- Dynamometer:
  - Magtrol HD-715 Dynamometer
  - Magtrol DSP7001 Controller
  - Magtrol 6510e Power Analyser
  - DC Power Supply

## *6.1 Velocity Control Performance: SpinTAC vs PI*

For Velocity control, we compared the performance of the SpinTAC controller to a PI controller. Each controller was tuned using the same method.

For the Estun motor these controllers were tuned experimentally by injecting 25% rated torque (45 oz-in) disturbances while running the motor at 100% rated speed (3000 rpm). This resulted in the following gains:

- PI Speed Controller
  - Kp = 20
  - Ki = 0.098
- SpinTAC Speed Controller
  - Bandwidth = 35 radians/s

The inertia used by the SpinTAC speed controller was estimated with the dyne coupled with the motor. The value was found to be 0.483 A / (krpm/s).

For the Teknic motor these controllers were tuned experimentally by injecting 50% rated torque (19.4 oz-in) disturbances while running the motor at 50% rated speed (2000 rpm). This resulted in the following gains:

- PI Speed Controller
  - Kp = 9
  - Ki = 0.03
- SpinTAC Speed Controller
  - Bandwidth = 16 radians/s

The inertia used by the SpinTAC speed controller was estimted with the dyne coupled with the motor. The value was found to be 4.23 A / (krpm/s).

These determined gains were held constant throughout all of the tests. This was done purposefully in order to highlight the wide operating range of the SpinTAC speed controller.

### 6.1.1 Disturbance Rejection

Disturbance rejection tests the controller's ability to compensate for external disturbances, which impact the motor speed. In the disturbance rejection test, a load torque is applied to the system, held on for a short period of time, and then removed from the system. Figure 17 is an example of a disturbance rejection test. The response of the controller is measured using the maximum speed error and settling time. The maximum speed error shows the deviation from the goal speed, and is an indication of how aggressively your controller is tuned. Aggressive tuning produces a low maximum error. In Figure 17 the PI controller presents a greater maximum speed error than the SpinTAC controller, indicating that the SpinTAC controller is more responsive in compensating for system error.

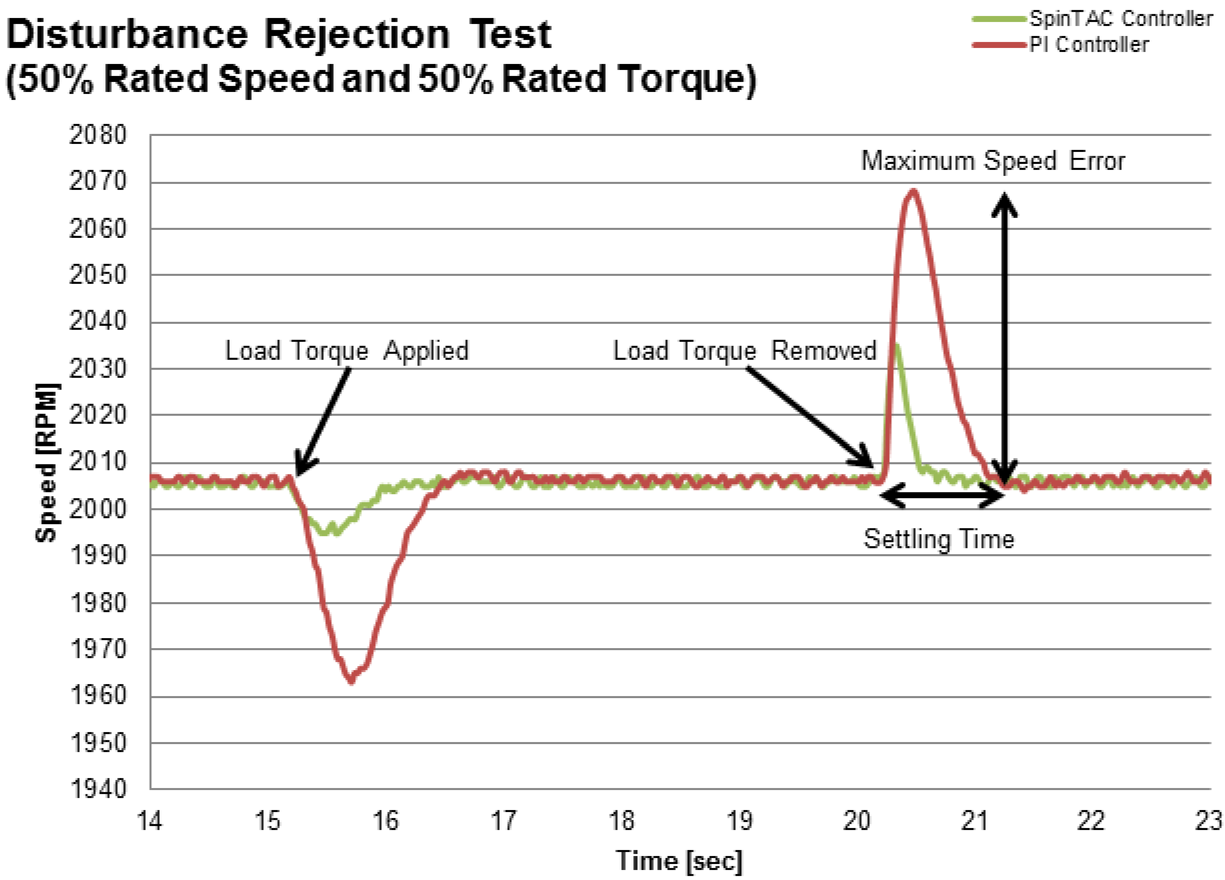


**Figure 17. Disturbance Rejection Test of Maximum Speed Error and Settling Time**

Settling time refers to the amount of time from the point when the disturbance happens until the speed returns to a fixed band around the goal speed. This is also an indication of how aggressively your control loop is tuned. If the controller is tuned too aggressively it will have a long settling time because it will oscillate around the goal speed before settling. In Figure 17, the PI controller has a longer setting time that the SpinTAC controller. Note that there is very little oscillation in either controller as they settle back to the goal speed.

There may be a difference in settling time when loads are applied to the system, and when loads are removed from the system. When a load is applied to a motor, the controller may reach saturation, at which point the controller's output is limited. However, when the load is removed, the motor transitions from a loaded state to zero load. The settling time and overshoot is entirely dependent upon the controller. Figure 18 shows an example of this case where the controller was placed into saturation.

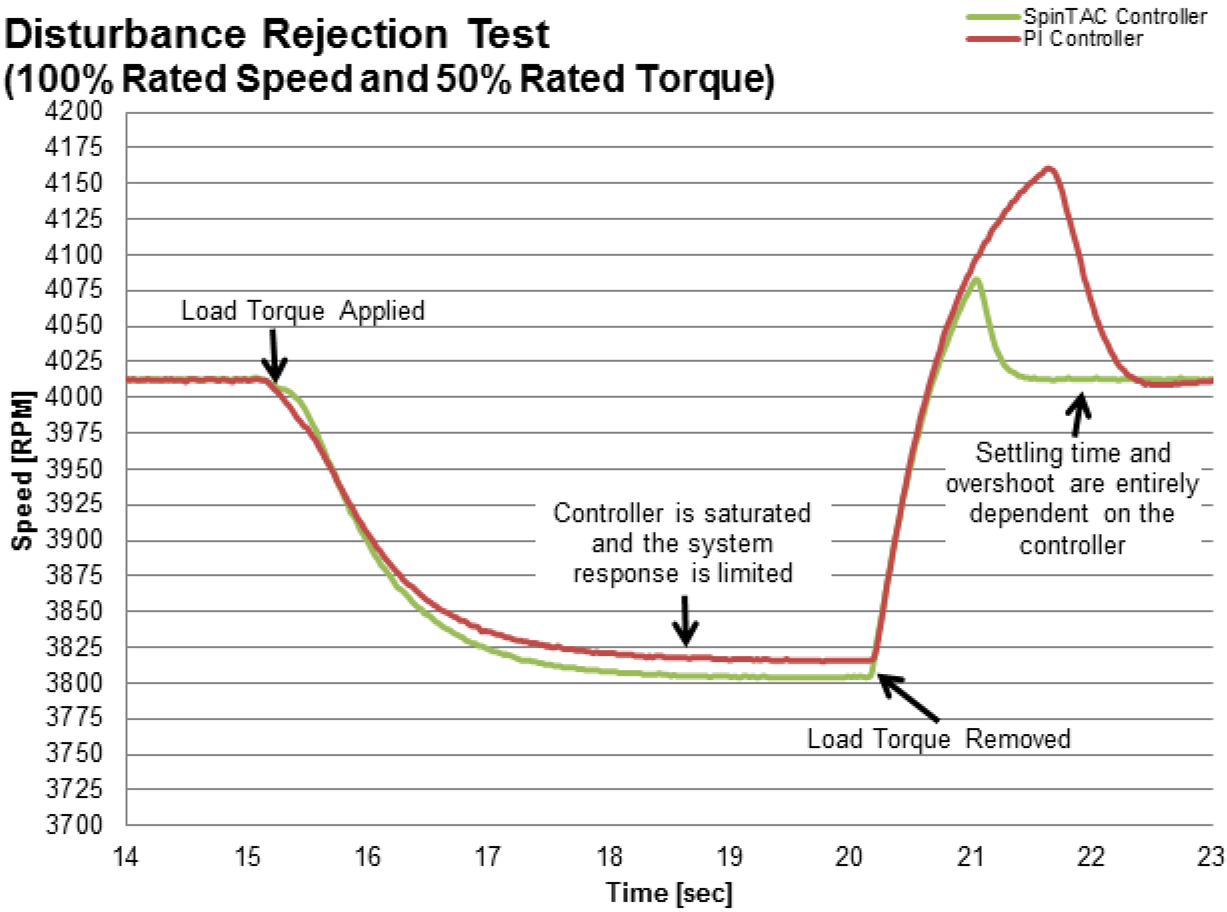


**Figure 18. Disturbance Rejection Test with Controller Saturation**

Speed controllers have different performance characteristics when placed into different situations. In order to properly evaluate the effectiveness of your speed controller, tests should be done across the entire application range. The test results indicate whether the controller meets the application specifications, or whether the controller needs to be tuned multiple times for different operating points. These tests were conducted at nine different speed and load combinations in order to test a wide range of operation.

It is also important to be able to create repeatable disturbances. This can be accomplished using a dynamometer or a disturbance motor. Creating repeatable disturbance is an important factor when evaluating multiple controllers. If test conditions cannot be replicated, it is difficult to adequately compare the responses of two controllers.

For the test results shown in Table 4 and Table 3, a disturbance load profile was created that applied 25%, 50%, and 100% of rated torque to the motor. The test compared the performance of the SpinTAC speed controller to a standard PI controller, and the following parameters were measured for each:

- Average Recovery Time (from the point of disturbance until within 2% of the target speed): The average recovery time was measured when the load was applied and when the load was removed from the system.
- Absolute Average Speed Error: The positive or negative deviation from the goal speed when a system disturbance is introduced.
- Maximum Speed Error: The maximum deviation from goal speed when a disturbance is introduced.

**Table 3. SpinTAC vs PI Disturbance Rejection Test Results (Teknic Motor)**

| | 1000 rpm | | | 2000 rpm | | | 4000 rpm | | |
|---|---|---|---|---|---|---|---|---|---|
| | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) |
| **25% rated torque** | | | | | | | | | |
| Avg Recovery Time(s) - load applied | 1.34 | 1.84 | 27.34 | 1.20 | 1.67 | 28.27 | 0.85 | 1.79 | 52.46 |
| Avg Recovery Time(s) - load removed | 0.43 | 0.93 | 53.29 | 0.41 | 0.89 | 53.55 | 0.54 | 1.02 | 46.82 |
| Abs Avg Error (rpm) | 3.06 | 4.16 | 26.44 | 5.98 | 6.63 | 9.74 | 12.23 | 12.42 | 1.53 |
| Avg Max Error (rpm) | 17 | 29 | 41.38 | 16 | 27 | 40.74 | 18 | 26 | 30.77 |
| **50% rated torque** | | | | | | | | | |
| Avg Recovery Time(s) - load applied | 1.01 | 1.33 | 23.81 | 1.01 | 1.45 | 30.34 | 4.98 | 5.04 | 1.19 |
| Avg Recovery Time(s) - load removed | 0.51 | 1.04 | 51.30 | 0.56 | 1.06 | 46.52 | 1.33 | 2.54 | 47.20 |
| Abs Avg Error (rpm) | 3.7 | 7.9 | 53.16 | 6.21 | 10.2 | 39.12 | 81.92 | 87.66 | 6.55 |
| Avg Max Error (rpm) | 36 | 71 | 49.30 | 35 | 69 | 49.28 | 197 | 185 | -6.49 |
| **100% rated torque** | | | | | | | | | |
| Avg Recovery Time(s) - load applied | 0.76 | 1.20 | 36.67 | 0.78 | 1.16 | 32.73 | 4.98 | 5.08 | 1.95 |
| Avg Recovery Time(s) - load removed | 0.40 | 1.00 | 59.84 | 0.52 | 1.02 | 48.89 | 1.90 | 3.12 | 39.09 |
| Abs Avg Error (rpm) | 5.4 | 15.39 | 64.91 | 7.99 | 17.54 | 54.45 | 345.42 | 360.74 | 4.25 |
| Avg Max Error (rpm) | 87 | 158 | 44.94 | 80 | 151 | 47.02 | 829 | 837 | 0.96 |

**Table 4. SpinTAC vs PI Disturbance Rejection Test Results (Estun Motor)**

| | 750 rpm | | | 1500 rpm | | | 3000 rpm | | |
|---|---|---|---|---|---|---|---|---|---|
| | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) |
| **25% rated torque** | | | | | | | | | |
| Avg Recovery Time(s) - load applied | 0.69 | 1.52 | 54.34 | 0.60 | 1.36 | 57.96 | 0.60 | 1.35 | 52.87 |
| Avg Recovery Time(s) - load removed | 0.42 | 1.12 | 61.75 | 0.40 | 1.03 | 61.58 | 0.41 | 1.01 | 59.01 |
| Abs Avg Error (rpm) | 1.97 | 3.81 | 48.13 | 3.05 | 5.25 | 41.90 | 6.16 | 9.08 | 32.12 |
| Avg Max Error (rpm) | 37 | 47 | 21.28 | 36 | 47 | 23.40 | 38 | 49 | 22.45 |
| **50% rated torque** | | | | | | | | | |
| Avg Recovery Time(s) - load applied | 0.35 | 1.31 | 72.97 | 0.33 | 1.33 | 76.23 | 0.36 | 1.13 | 67.07 |
| Avg Recovery Time(s) - load removed | 0.44 | 1.36 | 67.32 | 0.40 | 1.25 | 68.02 | 0.364 | 1.14 | 67.33 |
| Abs Avg Error (rpm) | 2.67 | 5.91 | 54.87 | 3.86 | 7.13 | 45.89 | 6.89 | 11.14 | 38.19 |
| Avg Max Error (rpm) | 76 | 96 | 20.83 | 74 | 95 | 22.11 | 76 | 97 | 21.65 |
| **100% rated torque** | | | | | | | | | |
| Avg Recovery Time(s) - load applied | 0.56 | 2.26 | 75.09 | 0.5 | 2.14 | 76.68 | 4.98 | 5.06 | 1.58 |
| Avg Recovery Time(s) - load removed | 0.38 | 1.16 | 66.78 | 0.4 | 0.92 | 55.93 | 0.44 | 0.74 | 40.60 |
| Abs Avg Error (rpm) | 8.64 | 57.98 | 85.09 | 9.54 | 59.95 | 84.09 | 94.25 | 103.74 | 9.15 |
| Avg Max Error (rpm) | 440 | 697 | 36.87 | 440 | 665 | 33.83 | 585 | 646 | 9.44 |

### 6.1.2 Reference Tracking

Reference tracking tests how well the controller follows a changing target speed. The two metrics to evaluate in this testing are the maximum error and the absolute average error. The maximum speed error shows how much the controller overshoots while changing speeds. This is an indication of how aggressively your controller is tuned. If your controller is not tuned aggressively enough, the speed will overshoot the target, and take a long time to recover. If the controller is tuned too aggressively it will overshoot, and then oscillate as it settles on the goal speed. If the controller is correctly tuned, it will minimally overshoot and then smoothly return to the goal speed.

Absolute average error is an average of the absolute value of the instantaneous speed error over the entire profile. This measure shows the amount of deviation throughout the entire profile. It takes into account all of the little errors as the motor is running. If the controller is tuned too aggressively it will result in larger absolute average error because the controller will be oscillating throughout the profile. If the controller is not tuned aggressively enough, it will result in a larger absolute average error because it is continuously falling behind what the profile is commanding the motor to do.

A tracking profile was created to exercise the motor in a repeatable a pattern. The profile was used to compare the performance of SpinTAC and PI controllers. The profile included quick transitions as well as gentle sweeping transitions. Figure 19 is a plot of the speed profile used during the reference tracking tests. The blocked off areas are areas where additional plots will be shown detailing the differences between the SpinTAC and PI controllers.

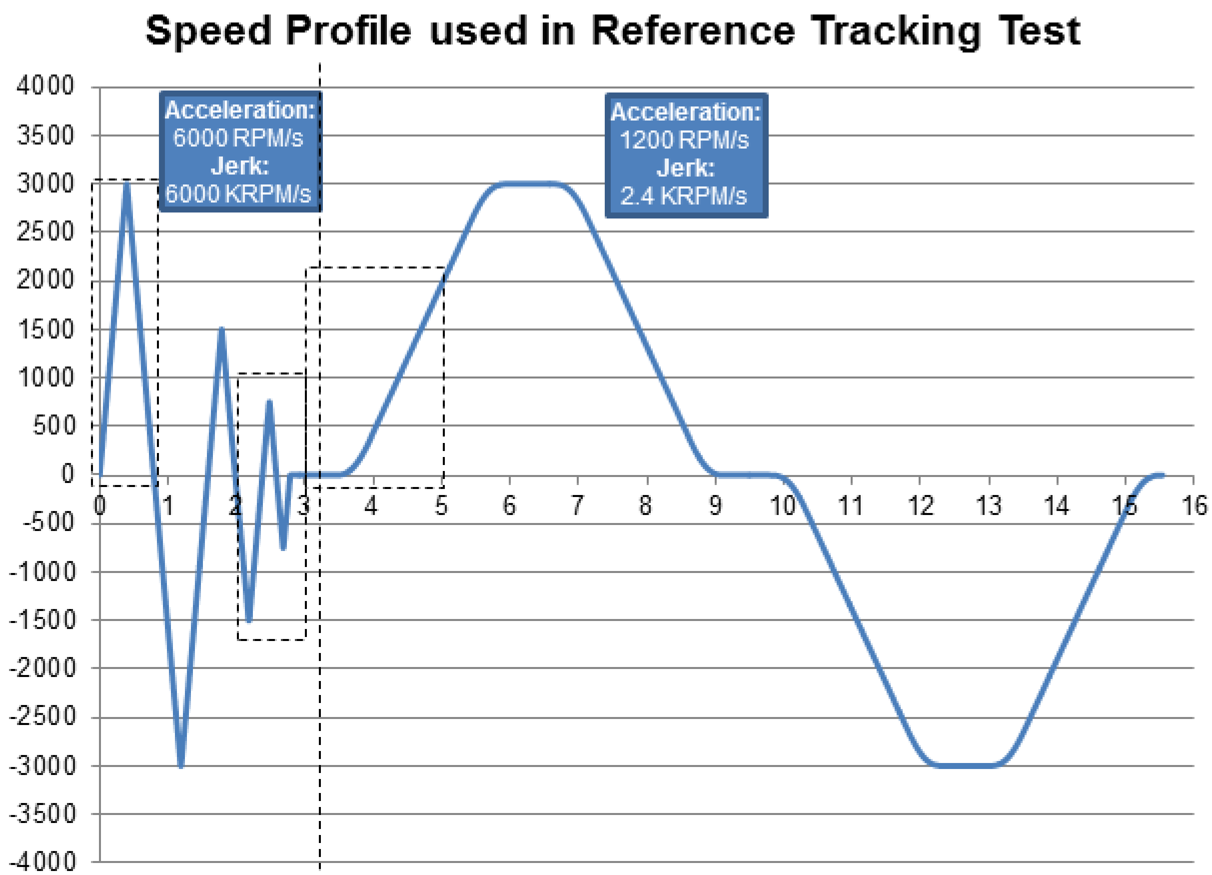


**Figure 19. Speed Profile Used During Reference Tracking Test**

There are some differences between the SpinTAC and PI controller that should be called out and evaluated in addition to the absolute average error and maximum error of the profile tracking. Figure 20 shows how the PI controller greatly overshoots the speed reference when it makes a very drastic change. It then takes quite a bit of time to recover from that initial error, while the SpinTAC controller has no problems with the drastic reference change.

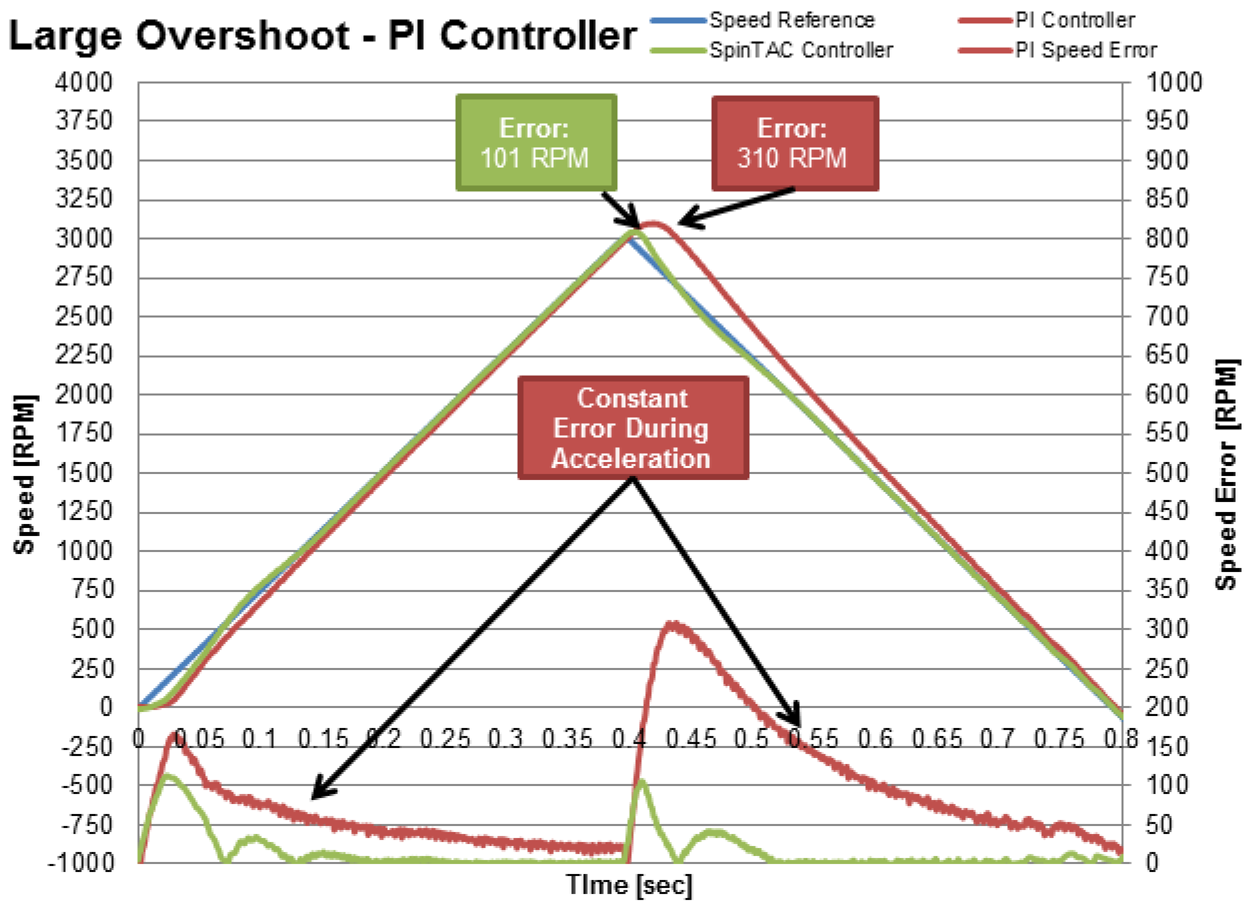


**Figure 20. Large Overshoot for PI Controller During Reference Tracking**

Figure 21 shows how the PI controller falls behind the reference as the drastic changes in the reference continue. The PI controller cannot keep up, while the SpinTAC controller has no difficultly in accurately tracking the speed reference.

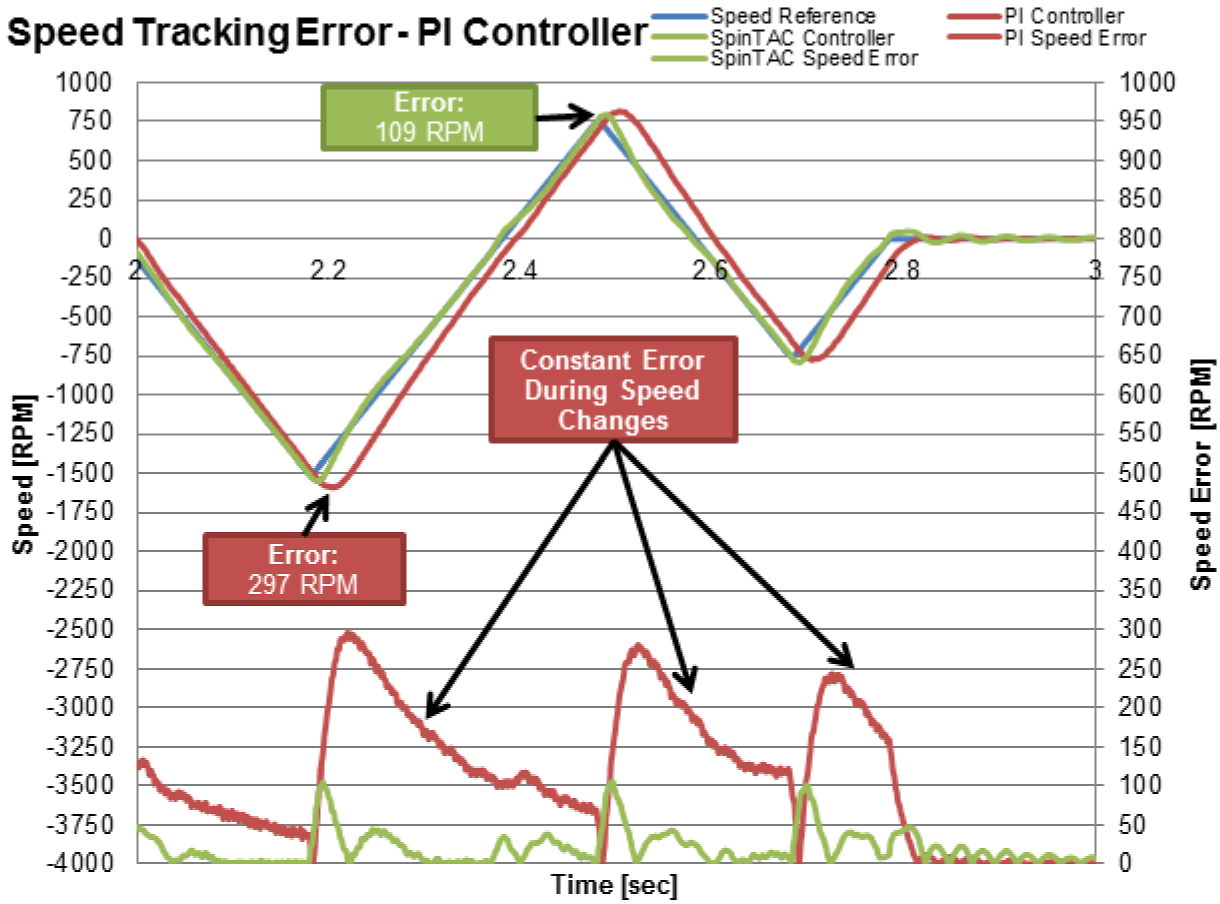


**Figure 21. Speed Tracking Error for PI Controller During Reference Tracking**

When starting from zero speed with a slow trajectory change, the PI controller has more difficulty than the SpinTAC controller. This is due to the SpinTAC controller's superior ability to track a changing reference signal. Figure 22 shows an example of this from the reference tracking test.

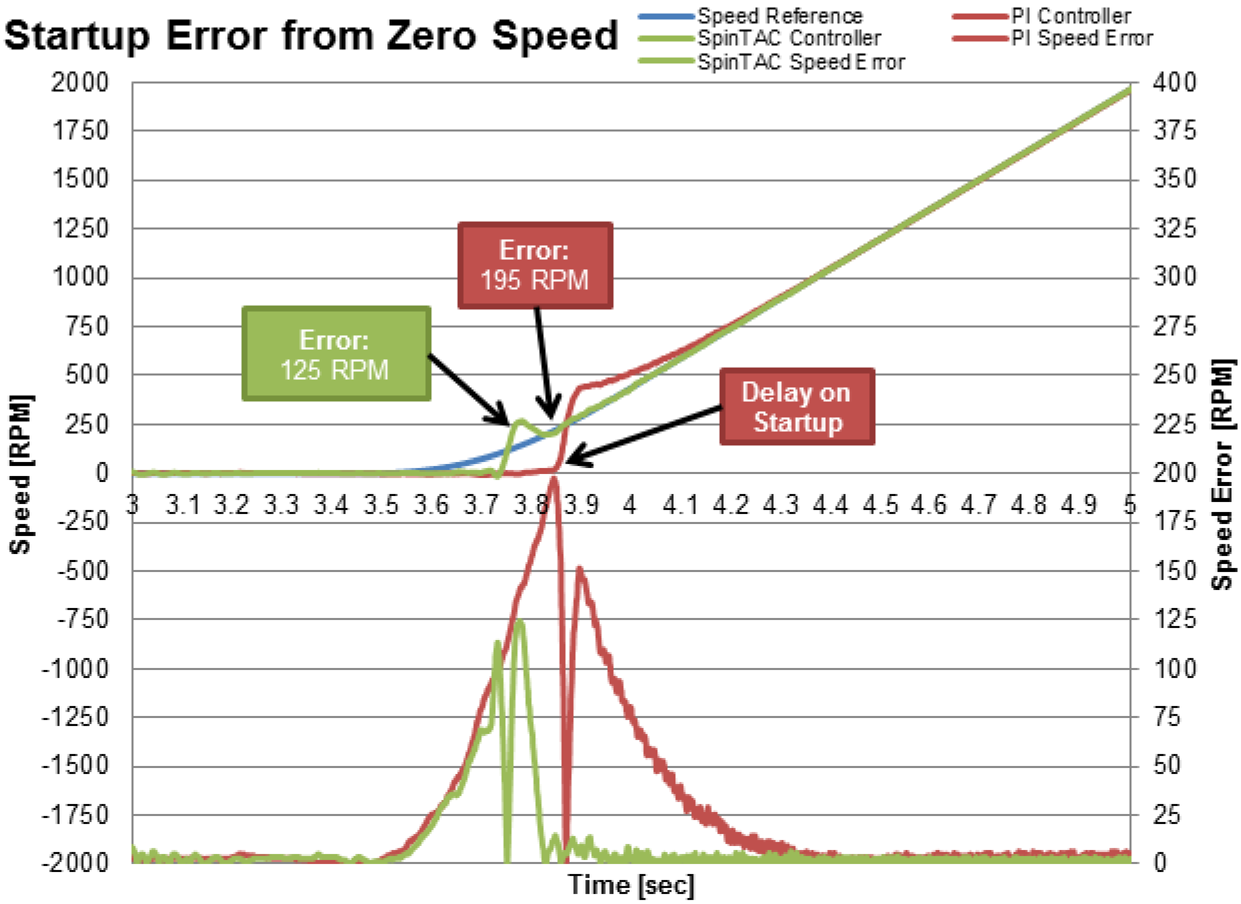


**Figure 22. Startup Error From Zero Start for PI Controller**

It is important to test multiple speeds and accelerations in your profile as well as multiple different loads. Speed controllers have different performance characteristics when placed into different situations. In order to properly evaluate the effectiveness of your speed controller, tests should be conducted across the entire application range. This includes when you design the profile for testing. Care needs to be taken to ensure that the application speeds and accelerations are built into the profile. The results of these tests will inform you if your controller will meet the application specifications or if your controller needs to be tuned multiple times for different operating points. You should notice in Figure 19 that there is a wide range of speeds and accelerations that are tested.

It is also important to be able to create repeatable profiles. Creating a repeatable profile can be done using SpinTAC Move and SpinTAC Plan [the *TMS320F2802xF InstaSPIN-FOC, TMS320F2806xF InstaSPIN-FOC, TMS320F2806xM InstaSPIN-MOTION User's Guide* (SPRUHJ1) provides detailed information on SpinTAC Move and SpinTAC Plan]. Repeatable profiles are required so that all controllers are tested using the same reference in the same order and for the same length of time. This ensures that test conditions are as identical as possible. The profile for this test was made using SpinTAC Move and SpinTAC Plan to ensure that an identical profile was presented to both the SpinTAC and PI controllers.

The following parameters were measured:

- Absolute Average Speed Error: The positive or negative deviation from the goal speed over the entire speed profile.
- Maximum Speed Error: The maximum deviation from the goal speed during the speed profile.

**Table 5. SpinTAC vs PI Profile Tracking Test Results**

| | Trapezoidal Curve | | | st-Curve | | |
|---|---|---|---|---|---|---|
| | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) |
| **Teknic Motor** | | | | | | |
| Abs Avg Error (rpm) | 6.01 | 19.94 | 69.86 | 5.98 | 19.51 | 69.35 |
| Max Error (rpm) | 266.06 | 430.00 | 38.13 | 263.00 | 334.71 | 21.42 |
| **Estun Motor** | | | | | | |
| Abs Avg Error (rpm) | 5.39 | 16.11 | 66.54 | 5.73 | 16.124 | 64.46 |
| Max Error (rpm) | 248.29 | 312.74 | 20.61 | 181.64 | 307.61 | 40.95 |

### 6.1.3 Step Response

Step response tests how quickly a controller can respond to a sudden input change. The two metrics to evaluate during this testing are settling time and maximum overshoot. This test is also a measure of stability of your controller. If the controller oscillates upon reaching the goal speed, then it is not very stable.

A step profile was applied to each controller. This step input bypassed the profile generator. The following parameters were measured:

- Settling Time (from the step input until within 2% of the target speed): The settling time reflects how long it takes the controller to reach the goal speed and bring the speed of the motor within a narrow band around the goal speed.
- Maximum Overshoot: The maximum speed the motor reaches after the step input.

Figure 23 compares the step responses of the SpinTAC and PI controllers. It also gives a visual representation of how these metrics were calculated. The SpinTAC controller was able to reach the goal speed with zero overshoot and the minimal amount of settling time. It achieved the minimal amount of settling time because the controller did not allow the motor speed to exceed the goal speed.

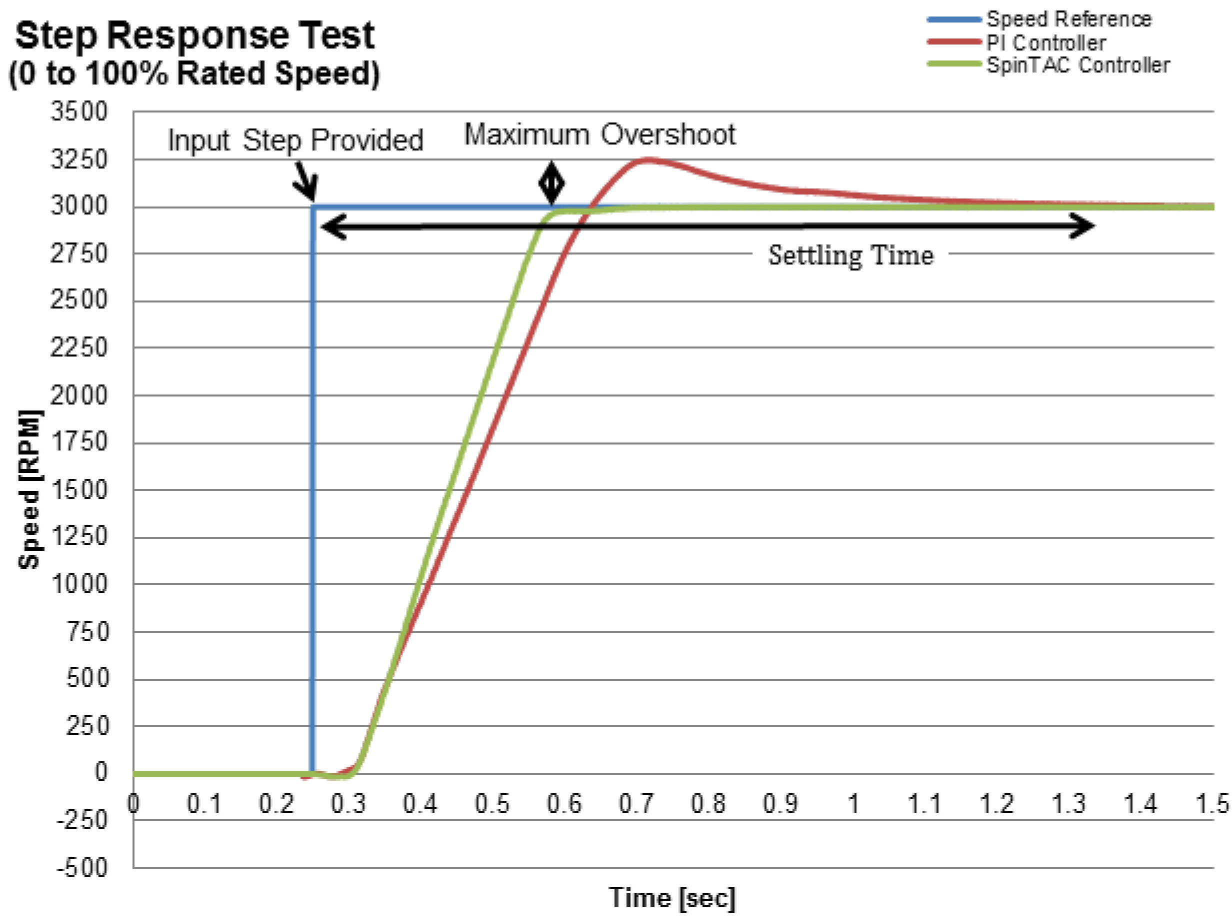


**Figure 23. Step Response Test of Maximum Overshoot and Settling Time**

**Table 6. SpinTAC vs PI Step Response Test Results (Teknic Motor)**

| Settling Time (s) | | | Overshoot (rpm) | | |
|---|---|---|---|---|---|
| SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) |
| **0-4000 rpm** | | | | | |
| 2.67 | 3.45 | 22.66 | 73 | 140.4 | 48.01 |
| 2.48 | 3.39 | 26.84 | 71.5 | 139.6 | 48.78 |
| 2.52 | 3.51 | 28.21 | 70.1 | 138.2 | 49.28 |
| 2.54 | 3.59 | 29.25 | 71.5 | 138.2 | 48.26 |
| 2.64 | 3.39 | 22.12 | 70.1 | 139.6 | 49.79 |
| **0-2000 rpm** | | | | | |
| 0.84 | 1.51 | 44.37 | 317 | 319.6 | 0.81 |
| 0.9 | 1.42 | 36.62 | 325 | 320.3 | -1.47 |
| 0.86 | 1.42 | 39.44 | 332 | 320.3 | -3.65 |
| 0.93 | 1.45 | 35.86 | 390 | 320.3 | -21.76 |
| 0.92 | 1.42 | 35.21 | 325 | 319.6 | -1.69 |
| **0-1000 rpm** | | | | | |
| 0.65 | 1.28 | 49.38 | 239 | 316.2 | 24.41 |
| 0.79 | 1.1 | 28.18 | 269 | 317.6 | 15.3 |
| 0.57 | 1.27 | 55.12 | 195 | 319.1 | 38.89 |
| 0.59 | 1.38 | 57.25 | 203 | 318.4 | 36.24 |
| 0.59 | 1.18 | 50 | 24.16 | 318.4 | 38.76 |

**Table 7. SpinTac vs PI Step Response Test Results (Estun Motor)**

| Settling Time (s) | | | Overshoot (rpm) | | |
|---|---|---|---|---|---|
| SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) |
| **0-3000 rpm** | | | | | |
| 0.44 | 1.16 | 61.81 | 2.9 | 250.5 | 98.84 |
| 0.54 | 1.1 | 50.91 | 2.9 | 251.2 | 98.85 |
| 0.38 | 1.32 | 71.21 | 2.9 | 252.7 | 98.85 |
| 0.4 | 1.21 | 66.94 | 2.9 | 252 | 98.85 |
| 0.42 | 1.41 | 70.21 | 2.9 | 252.7 | 98.85 |
| **0-1500 rpm** | | | | | |
| 0.25 | 1.12 | 78 | 3.7 | 261.5 | 98.59 |
| 0.37 | 1.3 | 71.54 | 19 | 262.2 | 92.75 |
| 0.47 | 1.12 | 58.04 | 3.7 | 263.7 | 98.6 |
| 0.42 | 1.1 | 61.82 | 3.7 | 263.7 | 98.6 |
| 0.44 | 1.13 | 61.06 | 3.7 | 262.9 | 98.59 |
| **0-750 rpm** | | | | | |
| 0.41 | 0.81 | 49.63 | 63.72 | 286.4 | 77.75 |
| 0.32 | 0.61 | 47.54 | 76.17 | 179.44 | 57.55 |
| 0.43 | 0.71 | 39.44 | 71.04 | 279.8 | 74.61 |
| 0.32 | 0.62 | 48.39 | 13.91 | 292.2 | 95.24 |
| 0.32 | 0.73 | 56.16 | 24.16 | 177.97 | 86.42 |

## 6.2 Position Control Performance: SpinTAC vs PI

The performance of SpinTAC Position Control was compared against a typical PI-PI control system. Each controller was tuned via the same method.

For the Estun motor, the controllers were tuned by injecting 25% rated torque (45 oz-in) disturbances while running the motor at 100% rated speed (3000 rpm). This resulted in the following gains:

PI Control System

- Position Loop
  - Kp = 20
  - Ki = 0.002
- Speed Loop
  - Kp = 23
  - Ki = 0.5

SpinTAC Position Control (single gain for position and speed)

- Bandwidth = 36 radians/s

The inertia used by SpinTAC Position Control was estimated with the dyne coupled with the motor. The value was found to be 0.44 A / (krpm/s).

For the Teknic motor, the controllers were tuned by injecting 50% rated torque (19.4 oz-in) disturbances while running the motor at 50% rated speed (2000 rpm). This resulted in the following gains:

PI Control System

- Position Loop
  - Kp = 17
  - Ki = 0.01
- Speed Loop
  - Kp = 1.2
  - Ki =0.1

SpinTAC Position Control (single gain for position and speed)

- Bandwidth = 50 radians/s

The inertia used by SpinTAC Position Control was estimated with the dyne coupled with the motor. The value was found to be 0.07 A / (krpm/s).

These determined gains were held constant throughout all of the tests. This was done purposefully in order to highlight the wide operating range of SpinTAC Position Control.

### 6.2.1 Disturbance Rejection

The disturbance rejection tests demonstrate the controller's ability to compensate for external disturbances, which impact the motor position. In this test a load torque is applied to the system, held for short period of time, and then removed from the system. Figure 24 is an example of a disturbance rejection test. The response of the controller is measured using the maximum position error and settling time. The maximum position error shows the deviation from the goal position, and is an indication of how aggressively the controller is tuned. Aggressive tuning produces a low maximum error. In Figure 24, the PI controller presents a greater maximum error than the SpinTAC controller, indicating that the SpinTAC controller is more responsive in compensating for system error.

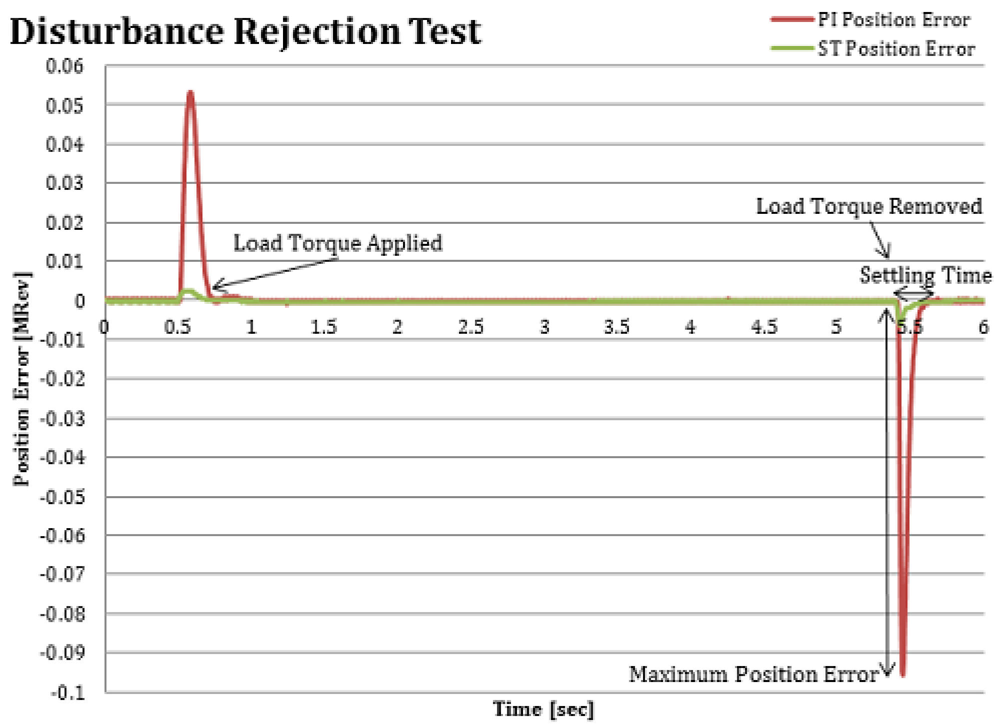


**Figure 24. Disturbance Rejection Test of Maximum Position Error and Settling Time**

Settling time refers to the amount of time from the point when the disturbance occurs until the position feedback returns to a fixed band around the goal position reference. This is also an indication of how aggressively the control loop is tuned. If the controller is tuned too aggressively it will have a long settling time, and will oscillate around the goal position before settling. Figure 24 shows that the PI controller has a longer settling time than the SpinTAC controller. Note that there is little oscillation from either controller as they return to the goal position.

When doing disturbance rejection testing it is important to test at multiple speed and load combinations. In order to properly evaluate the effectiveness of your position controller, tests should be done across the entire speed range since the speed at which the motor is running will impact the ability to control the position. The test results will indicate whether the controller will meet the application specifications or whether the controller needs to be tuned multiple times for different operating points. The disturbance rejection tests were conducted at nine different speed and load combinations (Table 8 and Table 9) in order to test a wide range of operation.

It is also important to be able to create repeatable disturbances. This can be accomplished using a dynamometer or a disturbance motor. Creating repeatable disturbance is an important factor when evaluating multiple controllers. If test conditions cannot be replicated, it is difficult to adequately compare the responses of two controllers.

For the following test results, a disturbance load profile was created that applied 25%, 50%, and 100% of rated torque to the motor. The test results compare the performance of SpinTAC Position Control to a standard PI position control system. The following parameters were measured:

- Average Recovery Time (from the point of disturbance until within 2% of the reference position) - the average recovery time was measured when the load was applied, and when the load was removed from the system.
- Absolute Average Position Error - positive or negative deviation from the reference position in mechanical revolutions (MRev) when a system disturbance is introduced.
- Maximum Position Error - maximum deviation in mechanical revolutions (MRev) from the goal position when a disturbance is introduced.

**Table 8. SpinTAC vs PI Position Control Disturbance Rejection Test Results (Teknic Motor)**

| | 1000 rpm | | | 2000 rpm | | | 4000 rpm | | |
|---|---|---|---|---|---|---|---|---|---|
| | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) |
| **25% rated torque** | | | | | | | | | |
| Avg Recovery Time (s) - load applied | 0.40 | 0.55 | 25.9 | 0.44 | 0.52 | 15.2 | 0.41 | 0.53 | 23.2 |
| Avg Recovery Time (s) - load removed | 0.2 | 0.50 | 59.6 | 0.25 | 0.38 | 33.6 | 0.20 | 0.39 | 49.5 |
| Abs Avg Error (MRev) | 8.9e-4 | 6.6e-3 | 86.4 | 7.5e-4 | 5.4e-3 | 86.1 | 7.3e-4 | 5.5e-3 | 86.7 |
| Avg Max Error (MRev) | 0.05 | 0.37 | 86.7 | 0.05 | 0.36 | 86.9 | 0.04 | 0.35 | 87.6 |
| **50% rated torque** | | | | | | | | | |
| Avg Recovery Time (s) - load applied | 0.28 | 0.39 | 27.9 | 0.44 | 0.49 | 11.0 | 0.27 | 0.56 | 51.4 |
| Avg Recovery Time (s) - load removed | 0.23 | 0.51 | 55.3 | 0.25 | 0.39 | 35.5 | 0.25 | 0.45 | 44.2 |
| Abs Avg Error (MRev) | 1.7e-3 | 1.3e-2 | 87.4 | 1.4e-3 | 1.2e-2 | 88.0 | 1.4e-3 | 1.2e-2 | 87.9 |
| Avg Max Error (MRev) | 0.10 | 0.75 | 86.8 | 0.10 | 0.74 | 87.0 | 0.09 | 0.71 | 86.8 |
| **100% rated torque** | | | | | | | | | |
| Avg Recovery Time (s) - load applied | 0.24 | 0.39 | 37.4 | 0.38 | 0.65 | 41.6 | 0.32 | 0.43 | 26.6 |
| Avg Recovery Time (s) - load removed | 0.26 | 0.64 | 59.7 | 0.28 | 0.43 | 35.3 | 0.32 | 0.42 | 23.8 |
| Abs Avg Error (MRev) | 3.2e-3 | 3.2e-2 | 90.0 | 3.0e-3 | 3.8e-2 | 92.1 | 2.4e-3 | 2.2e-3 | 89.4 |
| Avg Max Error (MRev) | 0.17 | 1.33 | 87.3 | 0.17 | 2.03 | 91.8 | 0.14 | 0.97 | 85.1 |

**Table 9. SpinTAC vs PI Position Control Disturbance Rejection Test Results (Estun Motor)**

| | 750 rpm | | | 1500 rpm | | | 3000 rpm | | |
|---|---|---|---|---|---|---|---|---|---|
| | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) |
| **25% rated torque** | | | | | | | | | |
| Avg Recovery Time (s) - load applied | 0.40 | 0.62 | 34.3 | 0.39 | 0.54 | 26.4 | 0.40 | 0.51 | 20.5 |
| Avg Recovery Time (s) - load removed | 0.22 | 0.30 | 24.8 | 0.19 | 0.29 | 34.0 | 0.20 | 0.33 | 39.5 |
| Abs Avg Error (MRev) | 2.6e-4 | 3.0e-3 | 91.5 | 2.6e-4 | 2.8e-3 | 90.6 | 2.6e-4 | 3.1e-3 | 91.6 |
| Avg Max Error (MRev) | 7.3e-3 | 0.08 | 90.8 | 6.8e-3 | 0.08 | 90.9 | 6.3e-3 | 0.11 | 94.5 |

**Table 9. SpinTAC vs PI Position Control Disturbance Rejection Test Results (Estun Motor) (continued)**

| | 750 rpm | | | 1500 rpm | | | 3000 rpm | | |
|---|---|---|---|---|---|---|---|---|---|
| | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) |
| **50% rated torque** | | | | | | | | | |
| Avg Recovery Time (s) - load applied | 0.29 | 0.46 | 36.2 | 0.24 | 0.42 | 43.0 | 0.27 | 0.35 | 21.7 |
| Avg Recovery Time (s) - load removed | 0.27 | 0.35 | 22.9 | 0.19 | 0.32 | 40.0 | 0.24 | 0.42 | 41.2 |
| Abs Avg Error (MRev) | 4.3e-4 | 4.8e-3 | 91.1 | 4.4e-4 | 4.7e-3 | 90.7 | 4.6e-4 | 0.04 | 98.9 |
| Avg Max Error (MRev) | 9.3e-3 | 0.16 | 94.1 | 9.3e-3 | 0.15 | 93.9 | 9.8e-3 | 0.15 | 93.4 |
| **100% rated torque** | | | | | | | | | |
| Avg Recovery Time (s) - load applied | 0.58 | 0.76 | 23.0 | 0.54 | 0.78 | 30.9 | 0.46 | X(1) | 100 |
| Avg Recovery Time (s) - load removed | 0.29 | 0.41 | 28.9 | 0.25 | 0.36 | 29.0 | 0.28 | X | 100 |
| Abs Avg Error (MRev) | 7.4e-3 | 0.07 | 89.9 | 5.8e-3 | 0.07 | 92.2 | 3.8e-3 | X | 100 |
| Avg Max Error (MRev) | 0.37 | 3.46 | 89.2 | 0.32 | 3.43 | 90.7 | 0.23 | X | 100 |

(1) X indicates that the controller could not complete the test because the motor began oscillating and was unable to return to stability when the load torque was removed.

### 6.2.2 Reference Tracking

Reference tracking tests how well the controller follows a changing position target. The two metrics to evaluate in these tests are the maximum error and the absolute average error. The maximum position error shows how much the controller overshoots while changing speeds. This is an indication of how aggressively your controller is tuned. If you controller is not tuned aggressively enough, the speed will overshoot the target, and will take a long time to recover. If the controller is tuned too aggressively it will overshoot, and then oscillate as it settles on the position target. If the controller is correctly tuned, it will minimally overshoot and then smoothly return to the position target.

Absolute average error is an average of the absolute value of the instantaneous position error over the entire profile. This measure shows the amount of deviation throughout the entire profile. It takes into account all of the little errors as the motor is running. If the controller is tuned too aggressively it will result in larger absolute average error because the controller will be oscillating throughout the profile. If the controller is not tuned aggressively enough, it will result in a larger absolute average error because it is continuously falling behind the commanded profile.

A position tracking profile was created to exercise the motor in a repeatable a pattern. The profile was used to compare the performance of SpinTAC Position Control and the PI control system. The profile included quick transitions as well as gentle sweeping transitions. Figure 25 is a plot of the reference profile that was used in the tests.

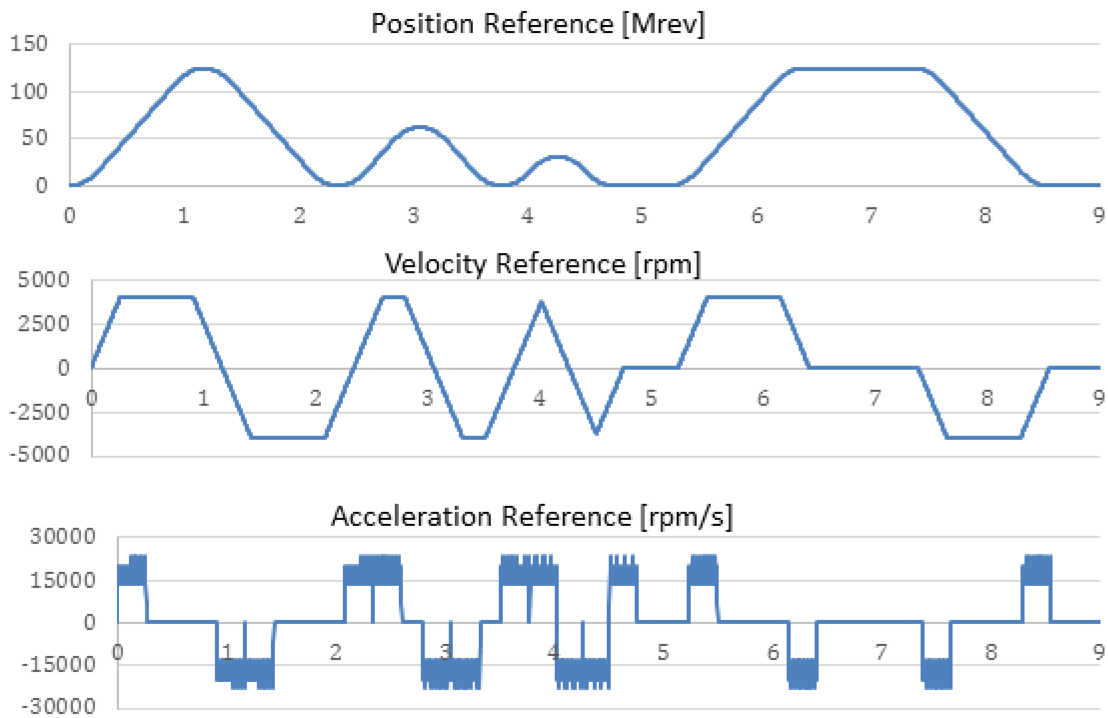


**Figure 25. Position Profile Used During Reference Tracking Test**

It is important to test multiple speeds and accelerations in your position profile as well as multiple different loads. Position controllers have different performance characteristics when placed into different situations. In order to properly evaluate the effectiveness of your controller, tests should be conducted across the entire application range. This includes when you design the profile for testing. Care needs to be taken to ensure that the application speeds and accelerations are built into the position profile. The results of these tests will indicate whether the controller will meet the application specifications, or if the controller needs to be tuned multiple times for different operating points. You should notice in Figure 25 that there is a wide range of speeds and accelerations that are tested.

It is also important to be able to create repeatable profiles. Creating a repeatable profile can be done using SpinTAC Move and SpinTAC Plan [the *TMS320F2802xF InstaSPIN-FOC, TMS320F2806xF InstaSPIN-FOC, TMS320F2806xM InstaSPIN-MOTION User's Guide* (SPRUHJ1) provides detailed information on SpinTAC Move and SpinTAC Plan]. Repeatable profiles are required if position controllers are being compared. A repeatable profile ensures that the controllers will be tested using the same reference in the same order and for the same length of time, and that test conditions are as identical as possible. The profile for this test was made using SpinTAC Move and SpinTAC Plan.

The following parameters were measured:

- Absolute Average Position Error - deviation in mechanical revolutions (MRev) from the target over the entire position profile.
- Maximum Position Error - maximum deviation in mechanical revolutions (MRev) from the goal position during the position profile.

**Table 10. SpinTAC vs PI Position Profile Tracking Test Results**

| | Trapezoidal Curve | | | st-Curve | | |
|---|---|---|---|---|---|---|
| | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) |
| **Teknic Motor** | | | | | | |
| Abs Avg Error (MRev) | 1.0e-3 | 1.7e-2 | 94.0 | 1.1e-3 | 1.3e-2 | 91.5 |
| Max Error (MRev) | 1.1e-2 | 0.27 | 96.0 | 1.1e-2 | 0.15 | 92.5 |
| **Estun Motor** | | | | | | |
| Abs Avg Error (MRev) | 9.2e-4 | 2.8e-3 | 96.7 | 9.7e-4 | 0.14 | 99.3 |
| Max Error (MRev) | 5.9e-3 | 0.32 | 98.3 | 4.9e-3 | 0.35 | 98.4 |

### 6.2.3 Step Response

The step tests for a position controller provide giving a step input to the controller to determine how quickly the controller can respond to a sudden input change. The two metrics evaluated during these tests are settling time and maximum overshoot. This test is also a measure of stability of your controller. If the controller oscillates upon reaching the goal speed then it is not very stable.

A step profile was applied to SpinTAC Position Control and the PI control system. This step input bypassed the profile generator. The following parameters were measured:

- Settling Time (from the step input until within 2% of the target position) - the settling time reflects how long it takes the controller to reach the goal position.
- Maximum Overshoot - maximum motor mechanical revolutions measured after the step input.

Figure 26 compares the step responses of SpinTAC Position Control and the PI control system. It also provides a visual representation showing how these metrics were calculated. SpinTAC Position Control was able to reach the goal position with zero overshoot and a shorter settling time than the PI control system.

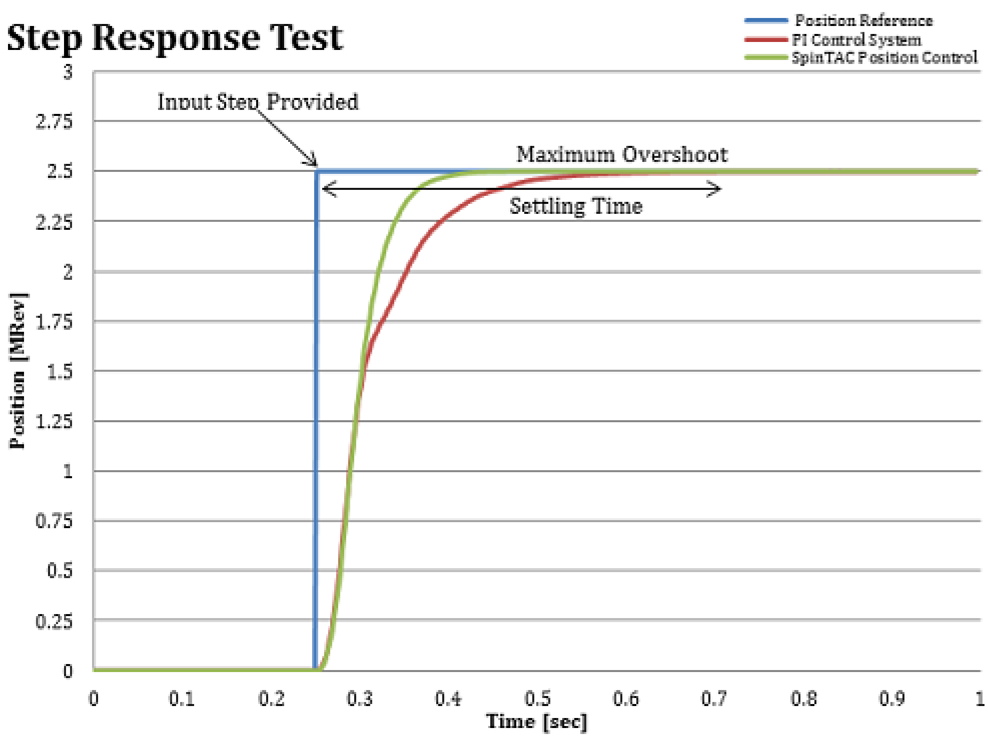


**Figure 26. Step Response Test of Maximum Overshoot and Settling Time**

**Table 11. SpinTAC vs PI Position Control Step Response Test Results (Teknic Motor)**

| | Settling Time (s) | | | Overshoot (MRev) | | |
|---|---|---|---|---|---|---|
| | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) |
| **0-4.9 MRev** | | | | | | |
| | 0.23 | 0.54 | 57.4 | 0.00 | 0.00 | 0.0 |
| | 0.22 | 0.45 | 51.1 | 0.00 | 0.00 | 0.0 |
| | 0.25 | 0.46 | 45.7 | 0.00 | 0.00 | 0.0 |
| | 0.28 | 0.45 | 37.8 | 0.00 | 0.00 | 0.0 |
| | 0.27 | 0.43 | 37.2 | 0.00 | 0.00 | 0.0 |
| **0-2.5 MRev** | | | | | | |
| | 0.24 | 0.36 | 34.4 | 0.00 | 0.00 | 0.0 |
| | 0.23 | 0.38 | 39.5 | 0.00 | 0.00 | 0.0 |
| | 0.24 | 0.41 | 41.5 | 0.00 | 0.00 | 0.0 |
| | 0.23 | 0.40 | 42.5 | 0.00 | 0.00 | 0.0 |
| | 0.21 | 0.41 | 48.8 | 0.00 | 0.00 | 0.0 |
| **0-1.25 MRev** | | | | | | |
| | 0.21 | 0.41 | 48.5 | 0.00 | 0.00 | 0.0 |
| | 0.20 | 0.41 | 51.2 | 0.00 | 0.00 | 0.0 |
| | 0.18 | 0.46 | 60.9 | 0.00 | 0.00 | 0.0 |
| | 0.18 | 0.40 | 55.0 | 0.00 | 0.00 | 0.0 |
| | 0.23 | 0.38 | 39.5 | 0.00 | 0.00 | 0.0 |

**Table 12. SpinTac vs PI Position Control Step Response Test Results (Estun Motor)**

| | Settling Time (s) | | | Overshoot (MRev) | | |
|---|---|---|---|---|---|---|
| | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) | SpinTAC | PI | SpinTAC Advantage (percentage improvement over PI) |
| **0-4.9 MRev** | | | | | | |
| | 0.67 | 0.59 | -13.6 | 1.26 | 3.49 | 63.9 |
| | 0.94 | 0.57 | -66.1 | 1.27 | 3.50 | 63.7 |
| | 0.86 | 0.57 | -49.6 | 1.27 | 3.52 | 63.8 |
| | 0.70 | 0.60 | -15.5 | 1.27 | 3.52 | 63.8 |
| | 0.81 | 0.62 | -30.1 | 1.27 | 3.56 | 64.3 |
| **0-2.5 MRev** | | | | | | |
| | 0.53 | 0.50 | -5.8 | 0.23 | 1.12 | 79.8 |
| | 0.52 | 0.48 | -8.3 | 0.23 | 1.16 | 80.5 |
| | 0.62 | 0.49 | -27.2 | 0.23 | 1.14 | 80.1 |
| | 0.56 | 0.47 | -18.2 | 0.23 | 1.16 | 80.5 |
| | 0.17 | 0.49 | 65.2 | 0.23 | 1.03 | 78.0 |
| **0-1.25 MRev** | | | | | | |
| | 0.53 | 0.42 | -26.2 | 0.00 | 0.05 | 100 |
| | 0.53 | 0.32 | -65.6 | 0.00 | 0.04 | 100 |
| | 0.53 | 0.33 | -63.1 | 0.00 | 0.05 | 100 |
| | 0.53 | 0.41 | -30.6 | 0.00 | 0.05 | 100 |
| | 0.60 | 0.36 | -68.4 | 0.00 | 0.05 | 100 |

### 6.2.4 Inertia Estimation Repeatability

The system inertia is an important input into both SpinTAC Velocity Control and SpinTAC Position Control. The inertia value is estimated using SpinTAC Velocity Identify. SpinTAC Velocity Identify is produces a very accurate inertia estimation. In order to test the repeatability of SpinTAC Velocity Identify, the inertia identification process was ran 100 times for each motor. The results are collected in Figure 27 and Figure 28.

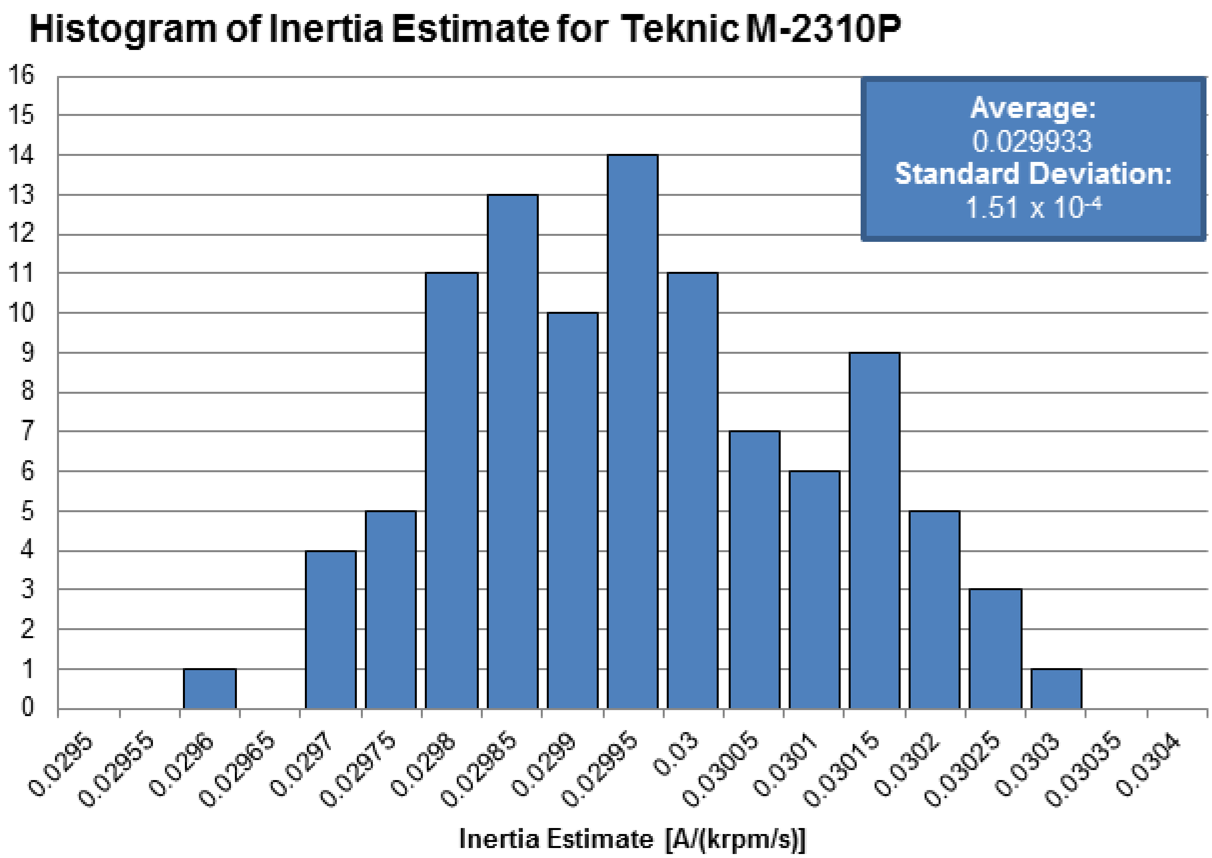


**Figure 27. Inertia Estimation Results for Teknic M-2310P**

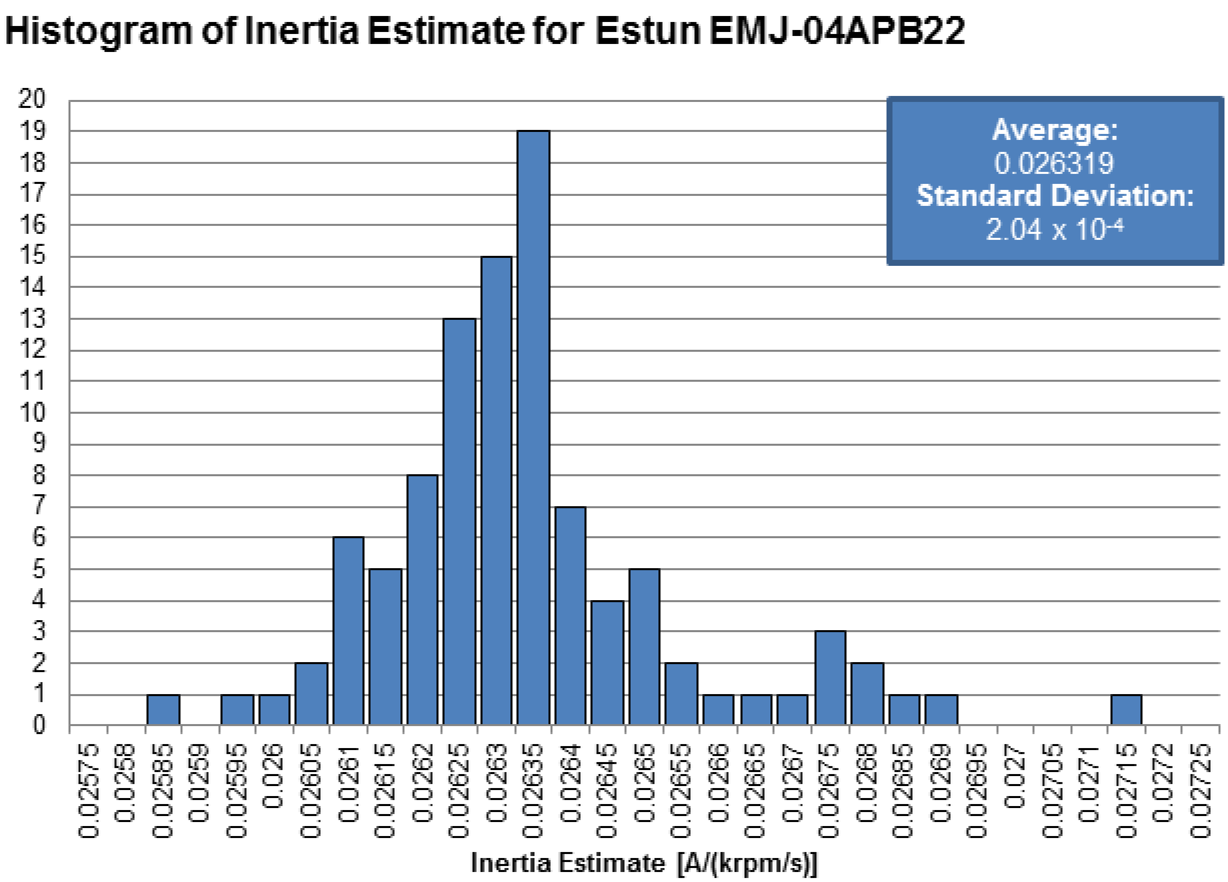


**Figure 28. Inertia Estimation Results for Estun EMJ-04APB22**

From the above results, it is clear that SpinTAC Velocity Identify has a very high degree of repeatability. It is not required that the inertia estimation that is provided to the SpinTAC controller be the perfect inertia value. As a robustness feature, a SpinTAC speed controller can accept a wide range of inertia changes. However, the SpinTAC speed controller will always produce the best performance if the inertia value provided to it is accurate.

# 7 Microcontroller Resources

The TMS320F2806xM microcontroller resources required by the InstaSPIN libraries are discussed in detail in the *TMS320F2802xF InstaSPIN-FOC, TMS320F2806xF InstaSPIN-FOC, TMS320F2806xM InstaSPIN-MOTION User's Guide* (SPRUHJ1).

The following resources were measured to determine their consumption by the InstaSPIN-MOTION library:

- CPU Utilization
- Memory Allocation
- Stack Utilization
- Digital and Analog Pins Utilization

## 7.1 CPU Utilization

Figure 29 illustrates the options available to the designer to manage the real-time scheduling of each of the major software functions.

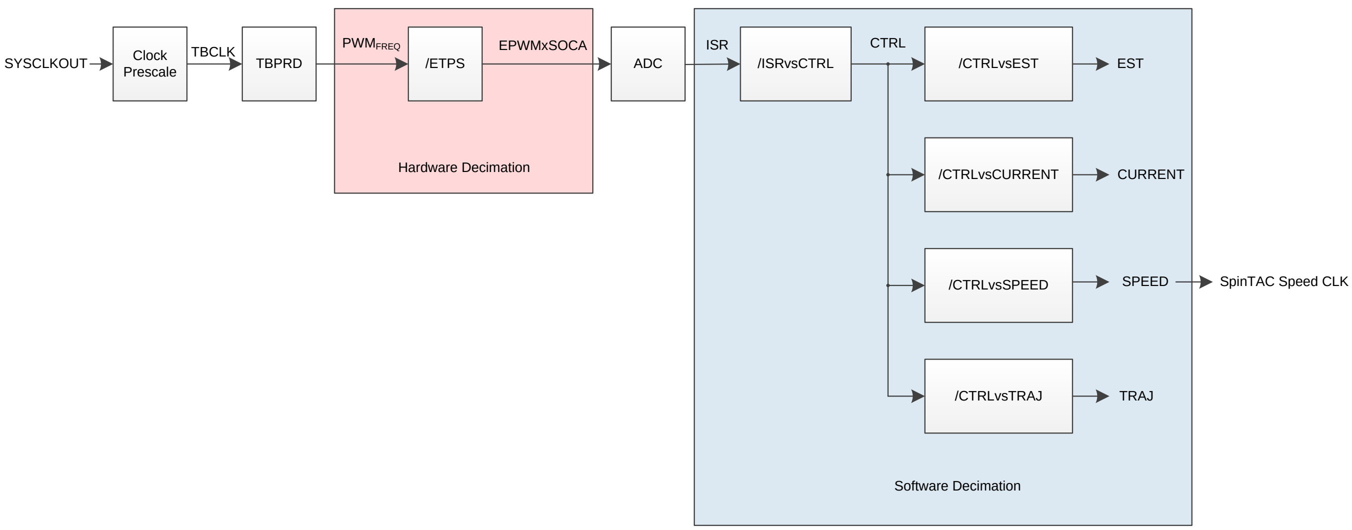


**Figure 29. Software Execution Clock Tree Provides Flexibility with Real-Time Scheduling**

Table 13 indicates CPU cycle utilization of SpinTAC components running out of ROM on the 28069M and 28068M devices. Note that this data is for the SpinTAC components only, and does include InstaSPIN-FOC. For InstaSPIN-FOC CPU utilization and memory footprints, see the *TMS320F28069F, TMS320F28068F, TMS320F28062F InstaSPIN™-FOC Software Technical Reference Manual* (SPRUHI9).

The Speed Count is used to calculate the SpinTAC sample time. SpinTAC is called from main ISR but can be decimated using the same decimation rate used to run the speed controller. The correct decimation rate can be calculated by multiplying the USER_NUM_ISR_TICKS_PER_CTRL_TICK by the USER_NUM_CTRL_TICKS_PER_SPEED_TICK.

### Table 13. SpinTAC CPU Cycle Utilization with Library Executing in RAM[(1)(2)(3)(4)]

| Function Name | | CPU Cycles Min | Avg | Max | Executed From ROM | RAM | FLASH |
|---|---|---|---|---|---|---|---|
| **STVELCTL_run (Velocity Control)** | | | | | ✓ | ✓ | × |
| | RES = 1, ENB = 0 | 158 | 158 | 158 | | | |
| | **RES = 0, ENB = 1** | **573** | **573** | **573** | | | |
| | First call after ENB = 1 | 1010 | 1010 | 1010 | | | |
| | Change Bandwidth parameter | 786 | 786 | 786 | | | |
| | Change Inertia parameter | 786 | 786 | 786 | | | |
| | RES = 1, ENB = 1 | 165 | 165 | 165 | | | |
| **STVELMOVE_run (Velocity Move)** | | | | | ✓ | ✓ | × |
| | RES = 1, ENB = 0 | 202 | 202 | 202 | | | |
| **stcurve** | **RES = 0, ENB = 1** | **713** | **743** | **1444** | | | |
| **scurve** | **RES = 0, ENB = 1** | **676** | **709** | **1409** | | | |
| **trap** | **RES = 0, ENB = 1** | **547** | **620** | **1134** | | | |
| | RES = 1, ENB = 1 | 191 | 191 | 191 | | | |
| **STVELPLAN_run (Velocity Plan)** | | | | | ✓ | ✓ | × |
| | RES = 1, ENB = 0 | 159 | 159 | 159 | | | |
| | **RES = 0, ENB = 1** | **169** | **169** | **169** | | | |
| | First call after ENB = 1 | 326 | 326 | 326 | | | |
| | STAY FSM State | 188 | 188 | 188 | | | |
| | Condition FSM State Calculation must be done for each State | 324 (fixed) + 290 * Number of Transitions + 68 * Number of EXIT Actions | | | | | |
| | Transition FSM State Calculation must be done for each State | 226 (fixed) + 63 * Number of ENTER Actions | | | | | |
| **STVELPLAN_runTick (ISR function)** | | 58 | 79.7 | 80 | | | |
| **STVELID_run (Velocity Identify)** | | | | | ✓ | ✓ | × |
| | RES = 1, ENB = 0 | 142 | 142 | 142 | | | |
| | **RES = 0, ENB = 1** | **217** | **239** | **667** | | | |
| | First call after ENB = 1 | 1083 | 1083 | 1083 | | | |
| | RES = 1, ENB = 1 | 149 | 149 | 149 | | | |
| **STPOSCOV_run (Position Converter)** | | | | | ✓ | ✓ | × |
| | RES = 1, ENB = 0 | 110 | 110 | 110 | | | |
| | **RES = 0, ENB = 1** | **322** | **331** | **333** | | | |
| | First call after ENB = 1 | 985 | 985 | 985 | | | |
| | RES = 1, ENB = 1 | 118 | 118 | 118 | | | |
| **STPOSCTL_run (Position Control)** | | | | | ✓ | ✓ | × |
| | RES = 0, ENB =0 | 176 | 176 | 176 | | | |
| | **RES = 0, ENB = 1** | **1134** | **1139** | **1154** | | | |
| | First call after ENB = 1 | 1927 | 1927 | 1927 | | | |
| | Change Bandwidth parameter | 1632 | 1632 | 1632 | | | |
| | Change Inertia parameter | 1632 | 1632 | 1632 | | | |
| | RES = 1, ENB = 1 | 185 | 185 | 185 | | | |

**Table 13. SpinTAC CPU Cycle Utilization with Library Executing in RAM[(1)(2)(3)(4)] (continued)**

| Function Name | | CPU Cycles Min | Avg | Max | Executed From ROM | RAM | FLASH |
|---|---|---|---|---|---|---|---|
| **STPOSMOVE_run (Position Move)** | | | | | ✓ | ✓ | × |
| | RES = 0, ENB = 0 | 406 | 406 | 406 | | | |
| **stcurve** | **RES = 0, ENB = 1** | **611** | **1400** | **2729** | | | |
| | Velocity Controlled Profile | 1272 | 1409 | 2475 | | | |
| **scurve** | **RES = 0, ENB = 1** | **611** | **1345** | **2560** | | | |
| | Velocity Controlled Profile | 1219 | 1367 | 2431 | | | |
| **trap** | **RES = 0, ENB = 1** | **611** | **1274** | **2498** | | | |
| | Velocity Controlled Profile | 1673 | 1850 | 2029 | | | |
| | RES = 1, ENB = 1 | 381 | 381 | 381 | | | |
| **STPOSPLAN_run (Position Plan)** | | | | | ✓ | ✓ | × |
| | RES = 1, ENB = 0 | 173 | 173 | 173 | | | |
| | RES = 0, ENB = 1 | 201 | 201 | 201 | | | |
| | First call after ENB = 1 | 363 | 363 | 363 | | | |
| | STAY FSM State | 201 | 201 | 201 | | | |
| | Condition FSM State Calculation must be done for each State | 388 (fixed) + 296 * Number of Transitions + 68 * Number of EXIT Actions | | | | | |
| | Transition FSM State Calculation must be done for each State | 240 (fixed) + 63 * Number of ENTER Actions | | | | | |
| **STPOSPLAN_runTick (ISR function)** | | 58 | 80 | 80 | | | |

(1) Microcontrollers were run at 90 MHz with the PWM at 15 kHz.
(2) RES = 1 indicates that the component is in Reset.
(3) ENB = 1 indicates that the component is enabled.
(4) The typical state appears in **bold**.

**Table 14. SpinTAC CPU Cycle Utilization with Library Executing in Flash[(1)(2)(3)(4)]**

| Function Name | | CPU Cycles | | | Executed From | | |
|---|---|---|---|---|---|---|---|
| | | Min | Avg | Max | ROM | RAM | FLASH |
| **STVELCTL_run (Velocity Control)** | | | | | ✓ | × | ✓ |
| | RES = 1, ENB = 0 | 216 | 216 | 216 | | | |
| | **RES = 0, ENB = 1** | **672** | **672** | **672** | | | |
| | First call after ENB = 1 | 1191 | 1191 | 1191 | | | |
| | Change Bandwidth | 938 | 938 | 938 | | | |
| | Change Inertia parameter | 938 | 938 | 938 | | | |
| | RES = 1, ENB = 1 | 220 | 220 | 220 | | | |
| **STVELMOVE_run (Velocity Move)** | | | | | ✓ | × | ✓ |
| | RES = 1, ENB = 0 | 275 | 275 | 275 | | | |
| **stcurve** | **RES = 0, ENB = 1** | **823** | **859** | **1725** | | | |
| **scurve** | **RES = 0, ENB = 1** | **786** | **825** | **1689** | | | |
| **trap** | **RES = 0, ENB = 1** | **654** | **743** | **1401** | | | |
| | RES = 1, ENB = 1 | 251 | 251 | 251 | | | |
| **STVELPLAN_run (Velocity Plan)** | | | | | ✓ | × | ✓ |
| | RES = 1, ENB = 0 | 219 | 219 | 219 | | | |
| | **RES = 0, ENB = 1** | **266** | **266** | **266** | | | |
| | First call after ENB = 1 | 427 | 427 | 427 | | | |
| | STAY FSM State | 266 | 266 | 266 | | | |
| | Transiation FSM State Calculation must be done for each State | 415 (fixed) + 386 * Number of Transitions + 84 * Number of EXIT Actions | | | | | |
| | Condition FSM State Calculation must be done for each State | 320 (fixed) + 83 * Number of ENTER Actions | | | | | |
| **STVELPLAN_runTick (ISR function)** | | 91 | 119 | 119 | | | |
| **STVELID_run (Velocity Identify)** | | | | | ✓ | × | ✓ |
| | RES = 1, ENB = 0 | 198 | 198 | 198 | | | |
| | **RES = 0, ENB = 1** | **305** | **321** | **813** | | | |
| | First call after ENB = 1 | 1366 | 1366 | 1366 | | | |
| | RES = 1, ENB = 1 | 206 | 206 | 206 | | | |
| **STPOSCOV_run (Position Converter)** | | | | | ✓ | × | ✓ |
| | RES = 1, ENB = 0 | 145 | 145 | 145 | | | |
| | **RES = 0, ENB = 1** | **437** | **442** | **4444** | | | |
| | First call after ENB = 1 | 1275 | 1275 | 1275 | | | |
| | RES = 1, ENB = 1 | 170 | 170 | 170 | | | |
| **STPOSCTL_run (Position Control)** | | | | | ✓ | × | ✓ |
| | RES = 0, ENB =0 | 246 | 246 | 246 | | | |
| | **RES = 0, ENB = 1** | **1315** | **1320** | **1330** | | | |
| | First call after ENB = 1 | 2262 | 2262 | 2262 | | | |
| | Change Bandwidth parameter | 1928 | 1928 | 1928 | | | |
| | Change Inertia parameter | 1928 | 1928 | 1928 | | | |
| | RES = 1, ENB = 1 | 254 | 254 | 254 | | | |

### Table 14. SpinTAC CPU Cycle Utilization with Library Executing in Flash[1][2][3][4] (continued)

| Function Name | | CPU Cycles Min | CPU Cycles Avg | CPU Cycles Max | Executed From ROM | Executed From RAM | Executed From FLASH |
|---|---|---|---|---|---|---|---|
| **STPOSMOVE_run (Position Move)** | | | | | ✓ | × | ✓ |
| | RES = 0, ENB = 0 | 520 | 520 | 520 | | | |
| **stcurve** | **RES = 0, ENB = 1** | **772** | **1635** | **3742** | | | |
| | Velocity Controlled Profile | 1425 | 1623 | 2917 | | | |
| **scurve** | **RES = 0, ENB = 1** | **772** | **1578** | **3188** | | | |
| | Velocity Controlled Profile | 1412 | 1582 | 2873 | | | |
| **trap** | **RES = 0, ENB = 1** | **772** | **1526** | **3119** | | | |
| | Velocity Controlled Profile | 1953 | 2188 | 2421 | | | |
| | RES = 1, ENB = 1 | 495 | 495 | 495 | | | |
| **STPOSPLAN_run (Position Plan)** | | | | | ✓ | × | ✓ |
| | RES = 1, ENB = 0 | 229 | 229 | 229 | | | |
| | RES = 0, ENB = 1 | 276 | 276 | 276 | | | |
| | First call after ENB = 1 | 473 | 473 | 473 | | | |
| | STAY FSM State | 276 | 276 | 276 | | | |
| | Condition FSM State Calculation must be done for each State | 473 (fixed) + 378 * Number of Transitions + 83 * Number of EXIT Actions | | | | | |
| | Transition FSM State Calculation must be done for each State | 327 (fixed) + 83 * Number of ENTER Actions | | | | | |
| **STPOSPLAN_runTick (ISR function)** | | 92 | 116 | 116 | | | |

(1) Microcontrollers were run at 90 MHz with the PWM at 15 kHz.
(2) RES = 1 indicates that the component is in Reset.
(3) ENB = 1 indicates that the component is enabled.
(4) The typical state appears in **bold**.

## 7.2 Memory Utilization

Table 15 lists the memory usage for each SpinTAC component. In SpinTAC Plan, there are a large number of configuration functions.

**Table 15. Code Size and RAM Usage for SpinTAC Components**

| Component | Code (.text) (16-bit Words) | RAM (.ebss) (16-bit Words) |
|---|---|---|
| Velocity Control | 0x2F4 | 0x4C |
| Velocity Move | 0x50D | 0x5C |
| Velocity Plan (Min) | 0x680 | 0x4E |
| Velocity Plan (Max) | 0x13BA | 0x4E |
| Velocity Identify | 0x3A5 | 0x3C |
| Position Converter | 0x20D | 0x4A |
| Position Control | 0x41E | 0x62 |
| Position Move | 0x141C | 0xCC |
| Position Plan (Min) | 0x7EE | 0x60 |
| Position Plan (Max) | 0x16AA | 0x60 |

Table 16 breaks down the maximum stack utilization of SpinTAC components when run individually. The stack consumption of InstaSPIN-FOC is included. To calculate the stack usage, the entire memory section where the stack is placed is filled with known values. The corresponding code was then run for a few minutes. The memory area where the stack was allocated was analyzed and the amount of used memory was calculated.

**Table 16. Stack Utilization of SpinTAC Components + InstaSPIN-FOC**

| Configuration | Maximum Stack Used (16-bit Words) |
|---|---|
| Velocity Control | 0x0120 |
| Velocity Move | 0x0120 |
| Velocity Plan + Move + Control | 0x0120 |
| Velocity Identify | 0x0120 |
| Position Converter | 0x0120 |
| Position Control | 0x0120 |
| Position Move | 0x0120 |
| Position Plan + Move + Control | 0x0120 |

For the F2806x and F2806xM devices, InstaSPIN-FOC v1.6 and SpinTAC v2.2.6 is stored in the address range of 0x3F8000 to 0x3FBFF and the last part of L8-RAM is reserved for InstaSPIN variables, address range 0x013800 to 0x013FFF. Figure 30 shows the memory locations that have been assigned for InstaSPIN-FOC and SpinTAC.

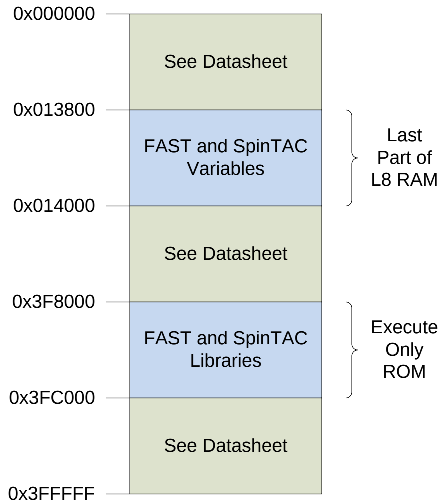


**Figure 30. F2806xF and F2806xM Allocated Memory for InstaSPIN-FOC and SpinTAC Library**

## Appendix A Definition of Terms and Acronyms

**ACIM —** Alternating current induction motor.

**ADRC —** Active Disturbance Rejection Control. Estimates and compensates for system disturbance, in real-time.

**CCStudio —** Code Composer Studio.

**FAST —** Unified observer structure which exploits the similarities between all motors that use magnetic flux for energy transduction, automatically identifying required motor parameters and providing the following motor feedback signals:

- High-quality Flux signal for stable flux monitoring and field weakening.
- Superior rotor flux Angle estimation accuracy over wider speed range compared to traditional observer techniques independent of all rotor parameters for ACIM.
- Real-time low-noise motor shaft Speed signal.
- Accurate high bandwidth Torque signal for load monitoring and imbalance detection.

**FOC —** Field-oriented control.

**Forced-Angle —** Used for 100% torque at start-up until the FAST rotor flux angle tracker converges within first electrical cycle.

**InstaSPIN-FOC —** Complete sensorless FOC solution provided by TI on-chip in ROM on select devices (FAST observer, FOC, speed and current loops), efficiently controlling your motor without the use of any mechanical rotor sensors.

**InstaSPIN-MOTION —** A comprehensive motor-, motion- and speed-control software solution that delivers robust system performance at the highest efficiency for motor applications that operate in various motion state transitions. InstaSPIN-MOTION builds on and includes InstaSPIN-FOC, combined with SpinTAC™ Motion Control Suite from LineStream Technologies.

**IPM —** Interior permanent magnet motor.

**LineStream Technologies —** Pioneers in the world of embedded controls software. Boasting a team of motor control experts from six different countries cumulatively speaking fifteen languages and possessing over eighty years of industry experience, LineStream is fast becoming the world's preeminent stronghold of embedded motor control knowledge.

**Motor Parameters ID or Motor Identification —** A feature added to InstaSPIN-FOC, providing a tool to the user so that there is no barrier between running a motor to its highest performance even though the motor parameters are unknown.

**PI —** Proportional-integral regulator.

**PMSM —** Permanent magnet synchronous motor.

**PowerWarp™ —** Mode of operation used for AC induction motors (ACIM) that allows minimum current consumption.

**Rs-Offline Recalibration —** InstaSPIN-FOC feature that is used to recalibrate the stator resistance, Rs, when the motor is not running.

**Rs-Online Recalibration —** InstaSPIN-FOC feature that is used to recalibrate the stator resistance, Rs, while the motor is running in closed loop.

**SpinTAC™ Motion Control Suite —** Includes and advanced speed controller, a motion engine, and a motion sequence planner. The SpinTAC disturbance-rejecting speed controller proactively estimates and compensates for system disturbances in real-time, improving overall product performance. The SpinTAC motion engine calculates the ideal reference signal (with feed forward) based on user-defined parameters. SpinTAC supports standard industry curves, and LineStream's proprietary "smooth trajectory" curve. The SpinTAC motion sequence planner operates user-defined state transition maps, making it easy to design complex motion sequences.

**SVM —** Space-vector modulation.

# Revision History

**Changes from A Revision (November 2013) to B Revision** **Page**



NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

## IMPORTANT NOTICE



| **Products** | | **Applications** | |
|---|---|---|---|
| Audio | www.ti.com/audio | Automotive and Transportation | www.ti.com/automotive |
| Amplifiers | amplifier.ti.com | Communications and Telecom | www.ti.com/communications |
| Data Converters | dataconverter.ti.com | Computers and Peripherals | www.ti.com/computers |
| DLP® Products | www.dlp.com | Consumer Electronics | www.ti.com/consumer-apps |
| DSP | dsp.ti.com | Energy and Lighting | www.ti.com/energy |
| Clocks and Timers | www.ti.com/clocks | Industrial | www.ti.com/industrial |
| Interface | interface.ti.com | Medical | www.ti.com/medical |
| Logic | logic.ti.com | Security | www.ti.com/security |
| Power Mgmt | power.ti.com | Space, Avionics and Defense | www.ti.com/space-avionics-defense |
| Microcontrollers | microcontroller.ti.com | Video and Imaging | www.ti.com/video |
| RFID | www.ti-rfid.com | | |
| OMAP Applications Processors | www.ti.com/omap | **TI E2E Community** | e2e.ti.com |
| Wireless Connectivity | www.ti.com/wirelessconnectivity | | |

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265