ADC Resolution vs Sensor Less Motor Control Performance

Cristian Ornelas
Tyler Hawkins
Tamara Hussam A Basfar
John Adam King

Concept of Operations

REVISION – 2
26 September 2024

Concept of Operations
for
ADC Resolution vs Sensor Less Motor Control Performance

Team <12>

Approved by:

Project Leader Date

Prof. Kalafatis Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	[9/15/2024]	[ADC Resolution vs Sensor Less Motor Control Performance]		Draft Release
2	[9/26/2024]	[ADC Resolution vs Sensor Less Motor Control Performance]		Draft Release

Table of Contents

List of Figures

1.0 Executive Summary

The focus of our project is to develop software for the c2000 TI chip to implement sensorless field-oriented motor control for motor systems. A microcontroller will be used to estimate the position of the rotor instead of traditional sensors. To improve and optimize motor position estimation the ADC resolution will be increased. Testing will be conducted under various conditions such as different motor speeds to evaluate how improved ADC resolutions impact the accuracy of position estimation. To test accuracy results from the sensorless motor controller will be compared to an incremental encoder.

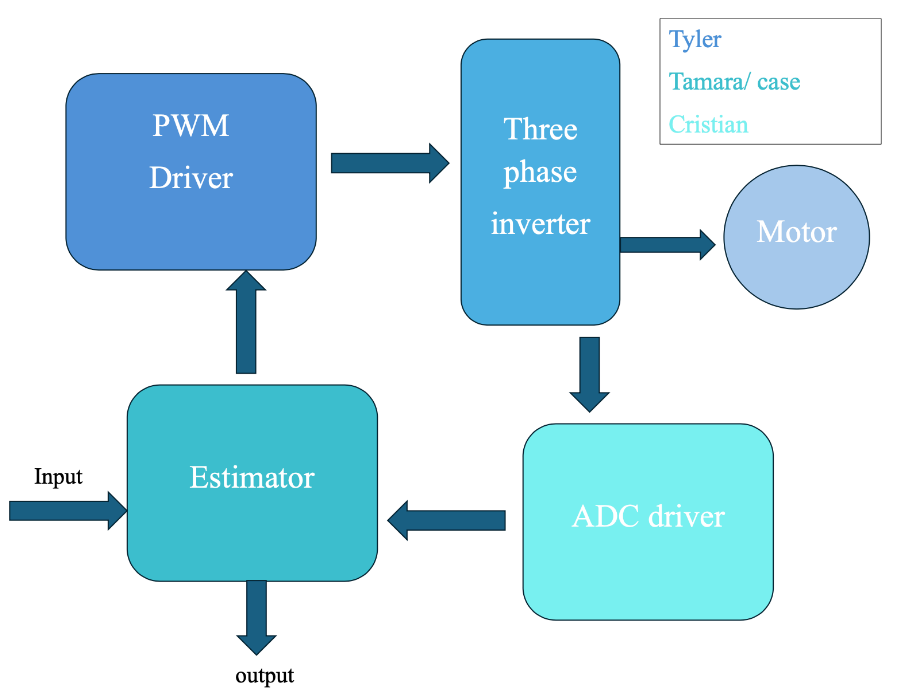

Figure 2. Simplified subsystem flow diagram

2.0 Introduction

2.1 Background

The most common motor design today is the Brushed DC Motor and the Brushless(BL) DC Motor. The brushed DC motor works by using mechanical logic and brushes to deliver electricity to the stator so that the motor spins, however, these brushes bend over time and reduce contact with the rotor. The BLDC motors often use Hall Effect sensors to detect where the rotor is, and this information is fed either into a microcontroller or analog circuit that will then control which magnets receive electricity in the stator. These motors are widely employed in all areas, but as more fields require precision motor controls, these designs have become insufficient.

The sensorless motor design aims to remove the need for sensors by using the feedback emf, electromotive force, of the motor and using this information to estimate the position of the rotor. This will allow the motor to operate with enhanced accuracy as an angle can be given for the orientation of the rotor as opposed to a region, and reduce the overall number of components needed for the motor to function.

2.2 Overview

The full system code will implement sensorless field-oriented control (FOC) of a motor. Our team will be provided hardware for the project, a motor/ motor controller, and a software development kit provided by Texas Instruments.

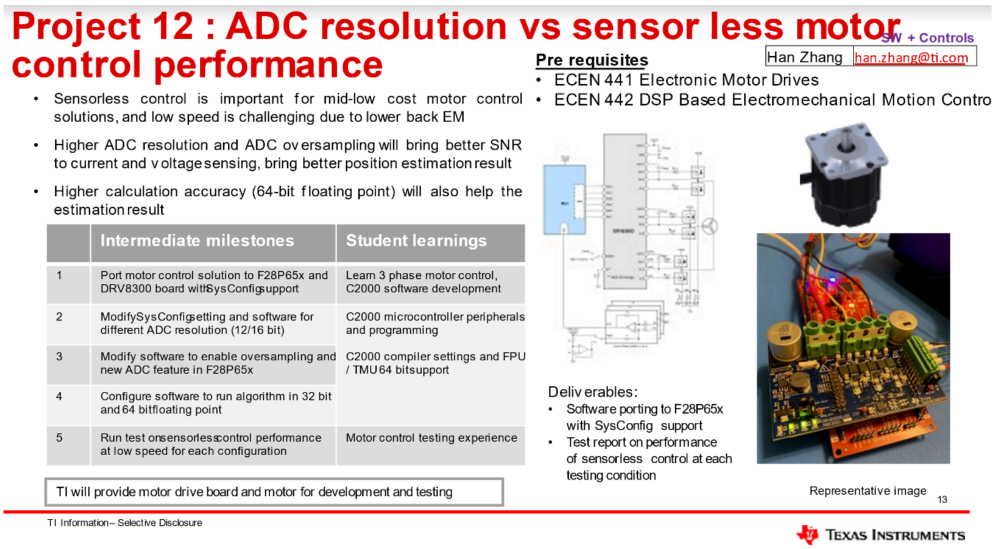

Figure 2. Project synopsis

To begin, we will port software to the F28P65x board with SysConfig support. Next, we will be implementing code to estimate the angle of the motor and load it onto the motor controller. To output results, our group will perform the first round of tests on the software that measures the motor operating with different ADC resolutions at low motor speeds. To be more precise, we will be performing additional tests on the software by enabling oversampling and 32/64-bit floating-point operators. The results will also be recorded at low motor speeds. Once this is completed, we will repeat our process at various motor speeds to capture a wide range of operating conditions that our sensorless motor could be in in the real world. Progress on developing the code and test results will be documented; this information is presented on the project as our final results.

The findings of this project should be the comparison between ADC resolutions and their effect on estimating the position accuracy. This will provide valuable insight into optimizing motor control for lower costs and how we can use this innovation to positively impact our environment and the intended users. Below is a breakdown of our team's subsystems:

Subsystem 1: Inverter - Tyler
Subsystem 2: Estimator - Tamara/ Case
Subsystem 3: ADC Driver - Cristian
System compiling will be a joint effort

2.3 Referenced Documents and Standards

1. *Motor Control Compendium*, https://www.ti.com/download/trng/docs/c2000/TI_MotorControlCompendium_2010.pdf Accessed 13 Sept. 2024.
2. “Analog-to-Digital Converter (ADC)#.”*Analog-to-Digital Converter (ADC) - C28x Academy*, dev.ti.com/tirex/explore/content/c28x_academy_1_00_00_00/_build_c28x_academy_1_00_00_00/source/c2000_analog_subsystem/c2000_analog_to_digital_converter.html. Accessed 13 Sept. 2024.
3. “Sensor vs Sensorless Motor Controllers: A Head-to-Head Comparison.” *Solo Motor Controllers*, 27 Jan. 2024, www.solomotorcontrollers.com/blog/sensor-vs-sensorless/?srsltid=AfmBOooTJsLnUA1zI53QF1GIPMYkBLSi7XJWOyLefSThTds0uwlZ2OQw.

3.0 Operating Concept

3.1 Scope

The ADC sensorless motor control performance will be designed to fit the proper parameters for professionals to carry out the finished product. The milestones for the scope are listed below:

- Port motor control solution to F28p65x and DRV8300 board with SysConfig support
- Modify SysConfig setting and software for different ADC resolutions (12/16 bit)
- Modify the software to enable oversampling and new ADC feature in F28p65x
- Configure software to run an algorithm, in 32-bit and 64-bit floating point
- Run tests on sensorless control performance at low speed for each configuration

The focus of the project will be on the development and testing of the control system under different testing conditions.

3.2 Operational Description and Constraints

Sensorless Motor control allows a user to control a motor's operations without the use of any onboard sensors. The software created will be able to determine the motor's position, speed, and torque using in/out current and voltage. The following diagram shows the flow of software operations used to control the motor.

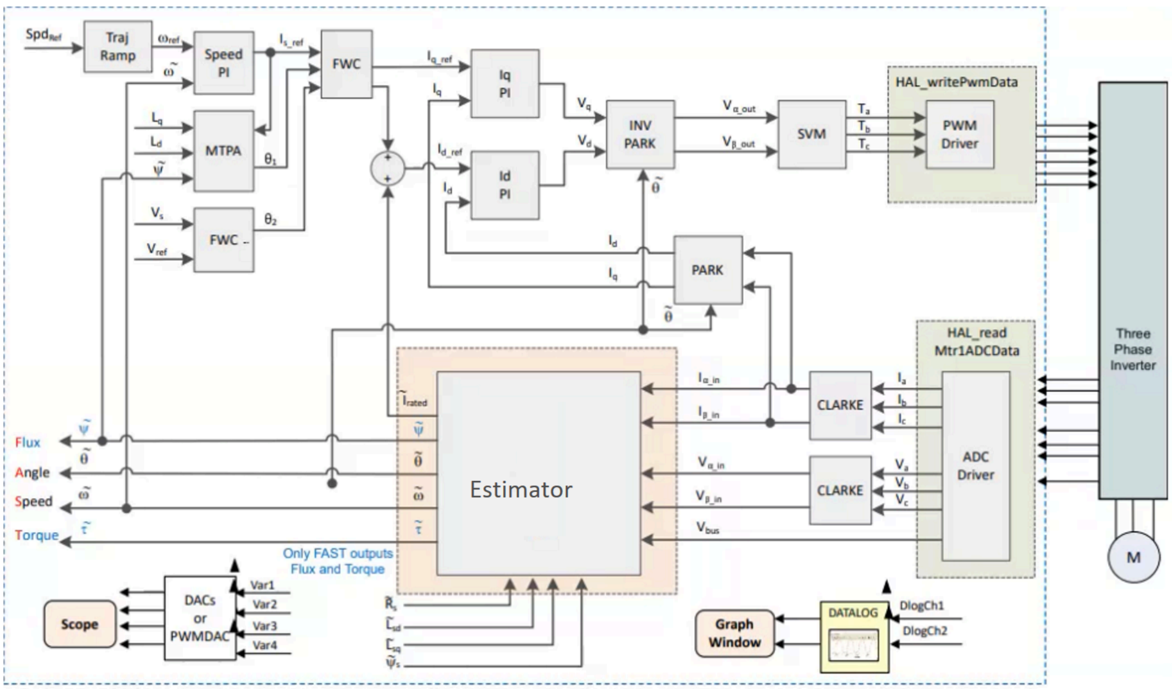

Figure 3. Block Diagram of System

The resulting constraints from this operational description are as follows:

- The Motor should be able to be controlled without any sensors
- The ability to control the motor using both 12 & 16-bit resolution
- The ability to control the motor using both 32 & 64-bit floating point operations
- All code used to control the motor should run on TI’s F28P65x Launchpad board and the DRV8300 motor driver board.
- All software should be able to be configured using sysconfig

3.3 System Description

- Microcontroller and Estimater Code: The microcontroller contains the code that will take the input voltage and current to estimate the position of the rotor. The code itself will operate by taking these values, calculating using the back EMF feedback, then force on the motor, and then using this data in conjunction with its internal clock to calculate the angular orientation of the rotor. This data will then be fed back to the connected computer, as well as the motor controller. The resulting rotor angle and speed are used for real time control purposes.

- Motor Controller and control code: The motor controller will take the position data of the rotor and determine which poles should then be powered to provide a continuous force on the rotor. This subsystem includes a module that manages the motor operation at high speeds, a controller regulating phase currents to achieve desired torque and speed, and a module that converts the reference voltage vectors into PWM signals.

- Motor: The motor is being driven by the motor controller, but is also the driving force behind the calculations run in the microcontroller. The motor operates through electrical to mechanical conversion, converting signals from the inverter to mechanical rotation, the torque generated is directly influenced by the supplied currents. Another key part of its operation are the resulting electrical motor characteristics that are imputed back into the estimate creating a feedback loop that is essential for real-time position and speed control without sensors.

3.4 Modes of Operations

The software is designed to cooperate utilizing an input ADC resolution to control the size of each sample taken by the microcontroller when calculating the rotor position. ADC resolution refers to how many bits are used for each data point. While any input is theoretically possible, the system will be tested on 32-bit and 64-bit resolutions. Additionally, the motor will operate in closed or open loop controls. In an open loop, the microcontroller will only use the BackEMF to determine the rotor position. In closed loop however, the previous estimation will be fed back into the estimation software and used in the next position test.

3.5 Users

The sensorless motor control system is intended for users who work with sensored motors and want to reduce the cost of production. Whether it's electric vehicle manufacturers who want to minimize complexity and increase reliability or HVAC engineers who wish to optimize fan motors, sensorless motors will revolutionize motor control. The intended users must have some intermediate to advanced knowledge of this system. They must understand basic motor control principles such as field-oriented control, programming languages like C++, error analysis, ADC resolution settings, etc.

3.6 Support

Support for the sensorless motor will be supplied in a user manual providing detailed instructions/ information on how to function the sensorless motor control system. There will also be troubleshooting guidance if the system doesn't work as intended.

4.0 Scenario(s)

4.1 Unfavorable/ Extreme Conditions

Without the need for any sensors, a sensorless controlled motor can operate much more reliably in harsher conditions. Motor sensors can often run into issues in environments whose variables frequently change. An example of this is frequent temperature fluctuations within an environment can often mess with the accuracy of a sensor. A sensor is also another point of failure for the motor within a harsh environment. Without the sensor, the motor is less likely to fail within these conditions.

4.2 Cheaper Alternative

Removing sensors from a motor within a device brings the cost down for that component. Examples commonly seen today include Electric transportation, such as E-bikes & scooters, Drones, and other robotic devices. A sensorless motor makes purchasing these devices more accessible to an everyday consumer, due to the cost reduction when compared to a sensored motor.

5.0 Analysis

5.1 Summary of Proposed Improvements

- Elimination of physical sensors for position estimation: This reduces hardware costs, and simplifies maintenance by reducing points of failure.
- Smaller motor designs: By eliminating physical sensors, the system enables more compact motor designs, saving space in applications.
- Increased flexibility and reliability: Sensorless motor control allows for a more flexible and reliable system due to fewer parts susceptible to damage; they have higher lifespans and can be used in harsh environments.
- Increasing ADC resolutions: an increased ADC resolution can potentially improve control speed precision which can improve performance at lower motor speeds.
- Optimal configuration identification: By testing the system at low and high speeds, the system will help identify optimal configurations for accurate position estimation and motor control.
- Environmental benefits: The system has the potential to contribute to energy-efficient designs and lower material costs, making it a more sustainable solution.

5.2 Disadvantages and Limitations

- Performance limitations at very low speeds due to lower back EMF
- sensorless motor controls use advanced algorithms which can complicate the control system needing a more sophisticated microprocessor.
- longer development time due to the need for calibration and testing.

5.3 Alternatives

- Hall effect motors: use hall effect sensors to detect the magnetic field from the rotor to estimate the position.
- Incremental motors: uses three wave pulses for position detection.
- Tachometer motor: uses speed sensors to estimate rotor position.
- Potentiometer motors: use a potentiometer to measure the angular displacement and estimate the position of the rotor.
- LVDTs: use linear differential transformers to sense linear displacement in some motors to estimate the rotor position.

These motor controllers have their advantages but are more costly and space-consuming.

5.4 Impact

The impact of this technology would be a reduction in the cost and material needed in precision motor applications, thus allowing more to be done for less. Overall, the impact on the environment and society would be quite small, though it would act as another point towards the idea of streamlining current technologies to use only the materials that they require.

ADC Resolution vs Sensor Less Motor Control Performance

Cristian Ornelas
Tyler Hawkins
Tamara Hussam A Basfar
John Adam King

FUNCTIONAL SYSTEM REQUIREMENTS

REVISION – 1
26 September 2024

Functional System Requirements

for

ADC Resolution vs Sensor Less Motor Control Performance

Prepared by:

Author Date

Approved by:

Project Leader Date

John Lusher, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	[9/26/24]	[ADC Resolution vs Sensor Less Motor Control Performance]		Draft Release

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1 Purpose and Scope

The project focuses on developing software for the TI c2000 chip to implement sensorless field-oriented control (FOC) for motor systems. Instead of traditional sensors, a microcontroller will estimate the rotor's position. By implementing the sensorless motor control, there will be cost reductions, improved efficiency and increased reliability. To enhance and optimize this estimation, the ADC resolution will be increased. Our team will test our sensorless motor control at different speeds to evaluate the ADC resolution and its accuracy. The accuracy of the motor controller will be validated by comparing its performance to those from an incremental encoder.

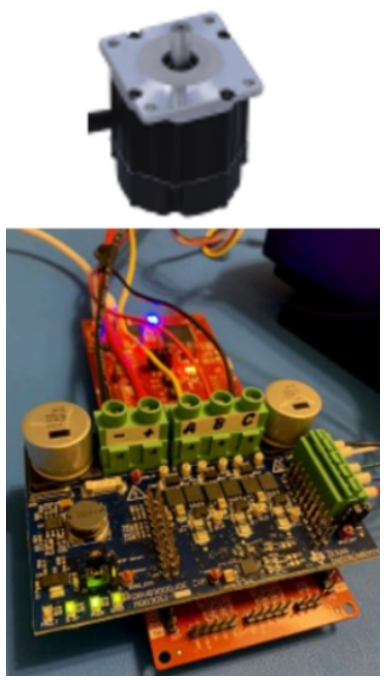

Figure 1. Conceptual Sensorless Motor Controller

The ADC sensorless motor control performance will be designed to fit the proper parameters for professionals to carry out the finished product. The milestones for the scope are listed below:

- Port motor control solution to F28p65x and DRV8300 board with SysConfig support
- Modify SysConfig setting and software for different ADC resolutions (12/16 bit)
- Modify the software to enable oversampling and new ADC feature in F28p65x
- Configure software to run an algorithm, in 32-bit and 64-bit floating point
- Run tests on sensorless control performance at low speed for each configuration

1.2 Responsibility and Change Authority

The team leader, Tyler Hawkins, is responsible for ensuring that all system specifications are met. Any changes to the deliverables or project specifications must be approved by both Tyler Hawkins and the sponsor's official representatives Han Zhang and Kevin Allen.

2. Applicable and Reference Documents

2.1 Applicable Documents

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

Document Number	Revision/Release Date	Document Title
SBVS144C	2012	Integrated MCU Power Solution for C2000 Microcontrollers
IEEE-STD-1241	2001	IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters
MCU117	2023	LAUNCHXL-F28P65X layout
v.0.3.0	2023	Servo Drive with CAN - Lab Projects User's guide

Table 1: Applicable documents

2.2 Reference Documents

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

Document Number	Revision/Release Date	Document Title
v1.3	2021	C2000 software guide
SLVUBV6	2020	DRV8300Dxxx-EVM User's guide

Table 2: Reference documents

2.3 Order of Precedence

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings or other documents that are invoked as "applicable" in this specification are incorporated as cited. All documents that are referred to within an applicable report are considered to be for guidance and information only, except ICDs that have their relevant documents considered to be incorporated as cited.

3. Requirements

This section defines the minimum requirements that the development item(s) must meet. The requirements and constraints that apply to performance, design, interoperability, reliability, etc., of the system, are covered.

3.1 System Definition

The sensorless motor controller is split up in three main subsystems: The PWM driver/three-phase inverter, ADC driver, and estimator. The PWM driver/three-phase inverter is responsible for converting a DC signal into 3 phases of AC waveforms The ADC driver focuses on ensuring that the analog signal is properly conditioned to meet the input requirements of the ADC for accurate conversion. This includes signal conditioning, impedance matching, noise filtering, and signal buffering. Finally, the estimator is responsible for taking the back EMF of the motor and using it to determine the rotor orientation. The rest of the blocks in the diagram below are responsible for general control requirements and operation dependencies.

This system/ project will include testing for ADC oversampling and different ADC resolutions (12/16 bit). This will bring better SNR to currency and voltage sensing, ultimately bringing more accurate estimation results.

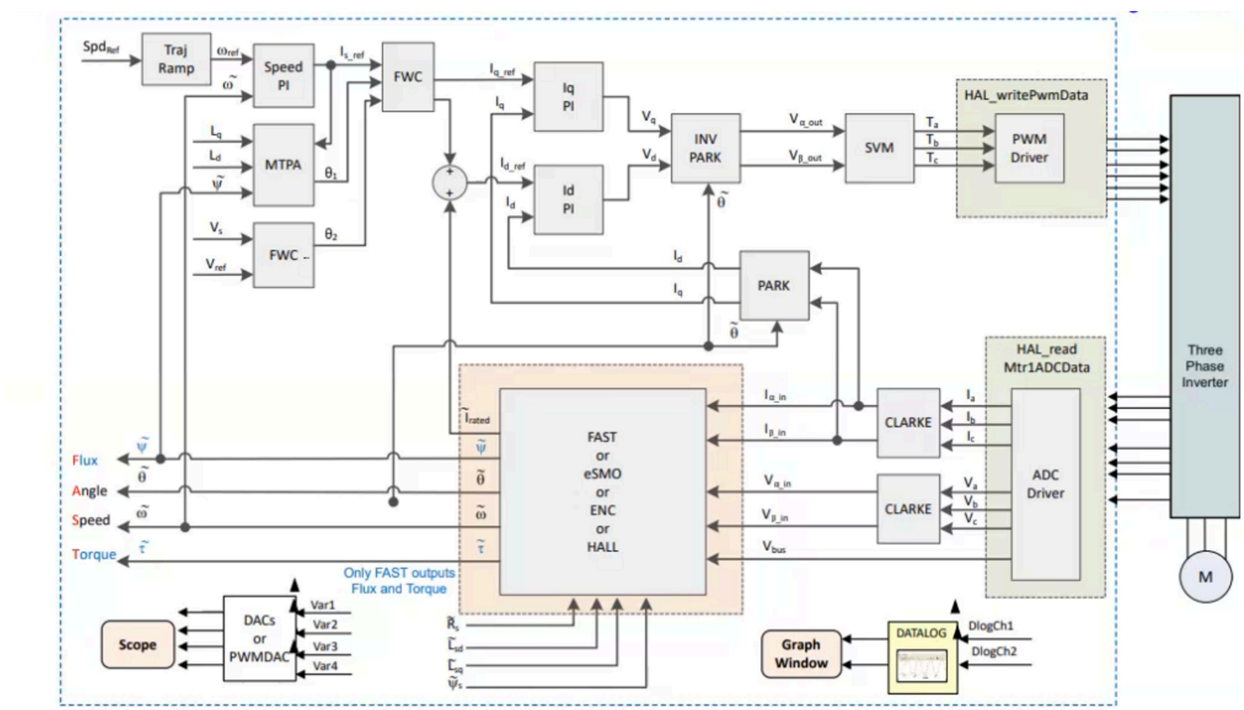

Figure 2. Block Diagram of System

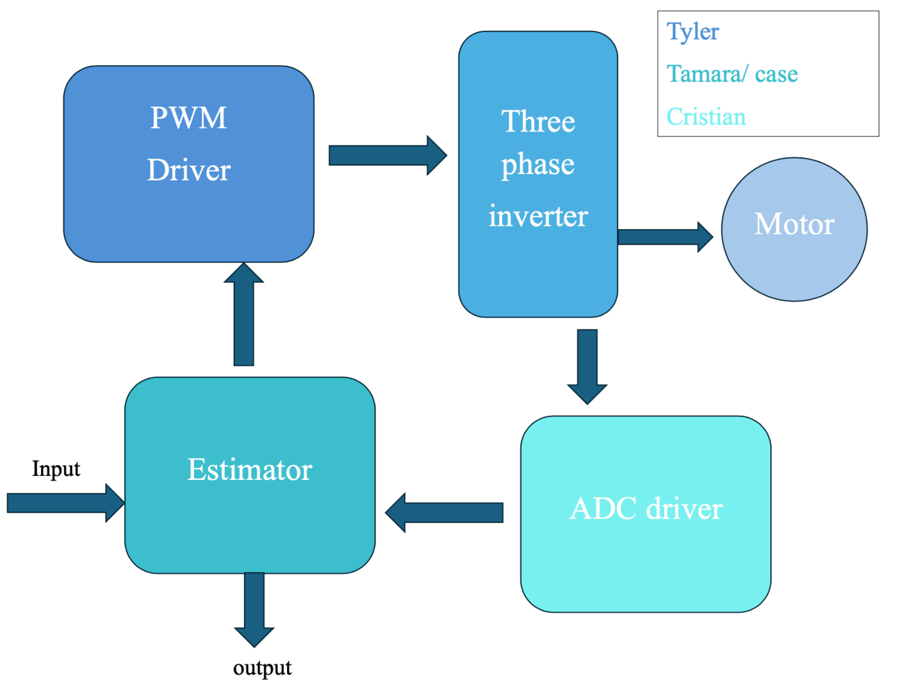

Figure 3. Simplified Block Diagram of System

3.2 Characteristics

3.2.1 Functional / Performance Requirements

3.2.1.1 Basic Motor Control

The sensorless control solution developed should effectively control the Teknic M-2310P-LN-04K Motor. The control solution should effectively determine the rotor's position, speed, torque, and rotation direction.

Rationale: This is the core system performance requirement. The sensorless control solution's performance is intended to be comparable to a motor sensor control solution.

3.2.1.2 SysConfig Support

Sensorless control of the motor should be configurable using the SysConfig IDE.

Rationale: This is the core system performance requirement. Using SysConfig allows for easy change of system options such as ADC Resolutions and 32/64 floating point.

3.2.1.3 12-Bit & 16-Bit ADC Resolutions

The sensorless control solution should support ADC Resolutions of 12 bit & 16 bit

Rationale: This is the core system performance requirement.

3.2.1.4 32 Bit & 64 Bit Floating Point

The sensorless control solution should support 32 Bit & 64 Bit Floating Point

Rationale: This is the core system performance requirement.

3.2.2 Physical Characteristics

The project is centered around a software application, however, this software is being designed to run with specific components; a microcontroller, motor controller, and motor. For development, the microcontroller is an F28p65x launchpad from Texas Instruments. While this board has dual processing cores, only Core-1 is being used. The motor controller is a DRV8300DIPW-EVM evaluation module from Texas Instruments. Lastly, the motor itself is a Low Voltage Servo Motor from Texas Instruments.

3.2.2.1 Board Connections

Our F28p65x Launchpad Board should be connected to the DRV8300 motor driver board and the Teknic M-2310P-LN-04K Motor.

Rationale: In order to control the motor listed above, the boards should be connected using the board's contact pins, in addition to jumper wires.

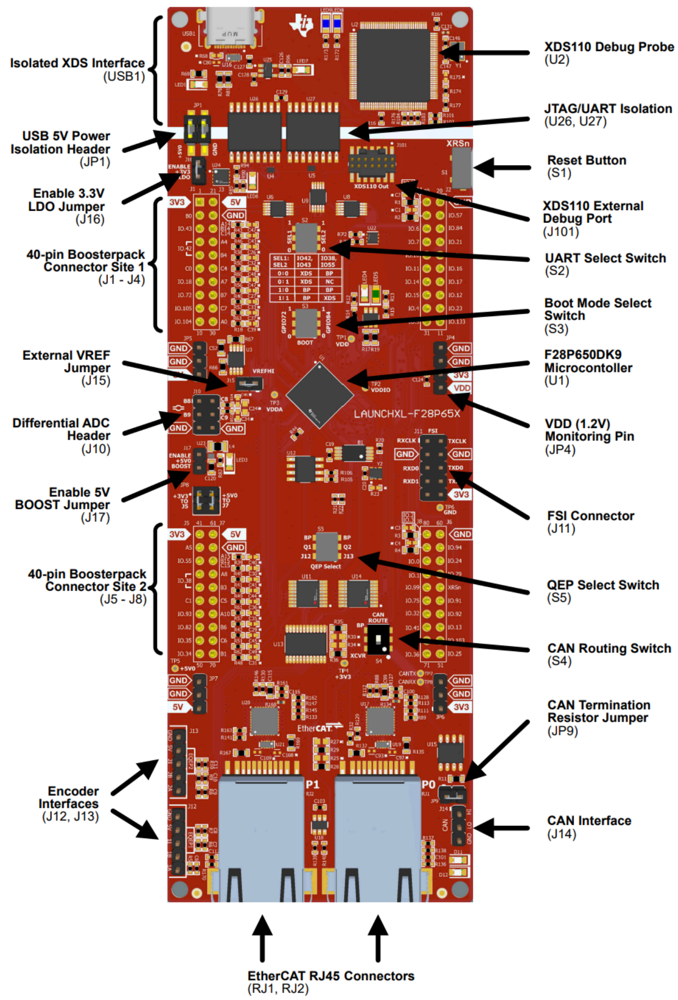

Figure 4. F28P65x Board Connections

3.2.3 Electrical Characteristics

3.2.3.1 Inputs

a. USB-C connection to the host device for the purpose of receiving control commands.

b. The Motor Driver Booster Pack is connected using the 40-pin connection at site 2 shown in the image above.

c. The motor is connected to the driver board’s corresponding phase a,b & c headers, as well as positive power & ground.

d. When applicable for testing connections to the motor’s sensor will be connected to the respective phase connectors on the driver board

Rationale: By design, should limit the chance of damage or malfunction by user/technician error.

3.2.3.2 Input Voltage Level

The input voltage of the F28P65x Board shall be 5 VDC through the USB connection
The DRV8300Dxxx-EVM is designed for an input external supply from 6 VDC to 100 VDC and up to 25-A continuous drive current (software limited).

Rationale: TI F28P65x & DRV83000 Board specifications

3.2.3.3 External Commands

The motor control solution shall document all external commands in the appropriate ICD.

Rationale: The ICD will capture all interface details from the low-level electrical to the high-level packet format.

3.2.3.6 Outputs

3.2.3.7 Data Output

The motor control solution shall include the Sysconfig data interface that is compatible with the system.

Rationale: Using Sysconfig, data output motoring can be viewed through the host device.

3.2.3.8 Diagnostic Output

The motor control solution shall include the Sysconfig control and data logging interface.

Rationale: Provides the ability to control things for debugging.

3.2.3.9 Wiring

The motor control solution shall follow the guidelines outlined in DRV8300xxx-EVM User's Guide paragraph 2.1 Hardware Connections Overview.

Rationale: Conform to Driver Board standard.

3.2.4 Failure Propagation

3.2.4.1 Diagnostic Errors

The motor control solution will monitor the voltage & current levels allowing visibility of any errors of the hardware and its operation.

Rationale: This will help preserve the integrity of the data being collected

4. Support Requirements

4.1.1. Computer USB Port

A computer is required to compile & run motor control solutions using the CCS environment. This requires a minimum of 4GBs of RAM, 2.5GBs disk space, & 2.0GHz single-core processor.

Appendix A: Acronyms and Abbreviations

EMF	Electro-motive Force
BIT	Built-In Test
CCA	Circuit Card Assembly
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EO/IR	Electro-optical Infrared
FOR	Field of Regard
FOV	Field of View
GPS	Global Positioning System
GUI	Graphical User Interface
Hz	Hertz
ICD	Interface Control Document
kHz	Kilohertz (1,000 Hz)
LCD	Liquid Crystal Display
LED	Light-emitting Diode
mA	Milliamp
MHz	Megahertz (1,000,000 Hz)
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair
mW	Milliwatt
PCB	Printed Circuit Board
RMS	Root Mean Square
TBD	To Be Determined
TTL	Transistor-Transistor Logic
USB	Universal Serial Bus
VME	VERSA-Module Europe
PWM	pulse width modulation
ADC	Analog to Digital converter
INV	inverter
c2000	real-time microcontroller
sysConfig	development tool for c2000
CCS	Code Compiler Studio

Appendix B: Definition of Terms

Sysconfig: a configuration tool designed to simplify hardware and software configuration challenges to accelerate software development. SysConfig provides an intuitive graphical user interface for configuring pins, peripherals, radios, software stacks, RTOS, clock tree and other components. SysConfig will automatically detect, expose and resolve conflicts to speed software development.

ADC Resolution vs Sensor Less Motor Control Performance

Cristian Ornelas
Tyler Hawkins
Tamara Basfar
John Adam King

Interface Control Document

REVISION – 1
26 September 2024

INTERFACE CONTROL DOCUMENT

FOR

ADC Resolution vs Sensor Less Motor Control Performance

PREPARED BY:

____

Author Date

APPROVED BY:

____

Project Leader Date

____

John Lusher II, P.E. Date

____

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
1	[9/26/24]	[ADC Resolution vs Sensor Less Motor Control Performance]		Draft Release

Table of Contents

List of Tables

List of Figures

1. Overview

This Interface Control Document (ICD) defines the technical specifications and interface requirements for the ADC Resolution vs Sensor Less Motor Control Performance project. It outlines how the subsystems defined in the Concept of Operations report and the Functional System Requirements report will be implemented and integrated. The ICD includes detailed descriptions of reference documents, key definitions, and physical and electrical interfaces. The report outlines the physical interface specifications, such as weight, dimensions, and mounting locations, as well as the electrical interface, covering primary input power, polarity reversal protection, and signal interfaces. Additionally, it details the user control interface and communication protocols necessary for seamless integration between the host device and peripheral components. This ICD shows how all subsystems are going to be implemented and produced in accordance with the defined requirements, facilitating compatibility and seamless operation across the entire system.

2. References and Definitions

2.1 References

SBVS144C
Integrated MCU Power Solution for C2000 Microcontrollers
2012

IEEE-STD-1241
IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters
2001

v1.3
C2000 software guide
2021

SLVUBV6
DRV8300Dxxx-EVM User’s Guide
2020

Motor Control Compendium

2011

Sensor vs Sensorless Motor Controllers: A Head-to-Head Comparison

2024

2.2 Definitions

CCS	Code Compiler Studio
mA	Milliamp
mW	Milliwatt
MHz	Megahertz (1,000,000 Hz)
TBD	To Be Determined
TTL	Transistor-Transistor Logic
VME	VERSA-Module Europe
PWM	pulse width modulation
ADC	Analog to Digital converter
INV	inverter
c2000	real-time microcontroller
sysConfig	development tool for c2000

3. Physical Interface

3.1 Weight

Component	Weight
C2000™ real-time MCU F28P65x LaunchPad™ development kit	1 oz
DRV8300DIPW evaluation module for three-phase BLDC	1 oz
Low Voltage Servo Motor - Low voltage servo (encoder) motor and wiring harness	23.1 oz
Total	25.1 oz

Table 1. Component's Weight

3.2 Dimensions

Component	Length	Width	Height
C2000™ real-time MCU F28P65x LaunchPad™ development kit	6.75 in.	2.3 in.	0.925 in.
DRV8300DIPW evaluation module for three-phase BLDC	3.5 in.	3.125 in.	0.75 in.
Low Voltage Servo Motor - Low voltage servo (encoder) motor and wiring harness	2.73 in.	2.73 in.	3.98 in.

Table 2. Component Dimensions

3.3 Mounting Locations

3.1.1 Mounting of the DRV8300Dxxx-EVM

The DRV8300Dxxx-EVM must plug into the lower LAUNCHXL-F280049C Launchpad headers as shown below.

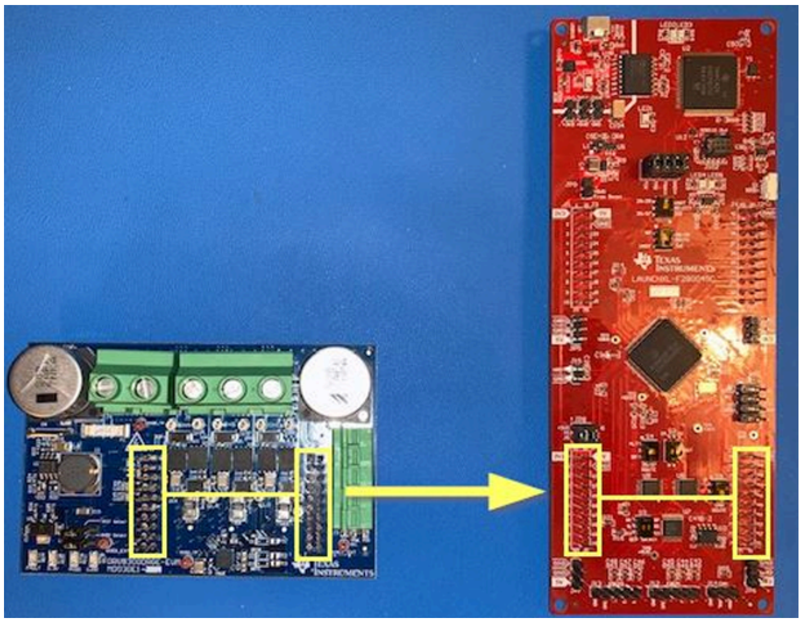

Figure 1. Mounting Location of the DRV8300Cxxx-EVM

4. Electrical Interface

4.1 Primary Input Power

Primary power through the DRV8300 board shall be connected to an external power supply. Power to the F28P65x board shall be provided through the USB connection.

4.2 Voltage Levels

The F28P65x Board shall use 5 VDC
The DRV8300 Board shall use 6 VDC to 100 VDC

4.3 User Control Interface

The user control interface will be the Sysconfig IDE on the host device.

5. Communications / Device Interface Protocols

5.1 Host Device

The host device will be the computer system connected to the microcontroller. Here, there commands will be given to the motor. As the host device requires Texas Instruments Code Composer Studio, it is recommended that the host device meet the same minimum and recommended requirements. The minimum hardware requirements are a 4GB memory, 2.5 GB of disk space, and a 2.0 GHz single core processor. The recommended hardware requirements are 8GB of memory, 5.0 GB of disk space, and a multicore processor.

5.2 Device Peripheral Interface

For connecting the computer system to the microcontroller, the project will be utilizing a wire protocol. The wire required is a USB to UART connector. The launchpad used in this project features a Fast Serial Communications Peripheral, enabling robust high-speed communications. See document Users Guide C2000™ F28P65x Series LaunchPad™ Development Kit, paragraph 3.1.5 FSI for specific details, and 3.1.6 and 3.1.7 for additional attributes.

Status Indicators	
On schedule	
In progress	
Completed	
Behind schedule	
Important Deadlines	

Task	Owner	Deadline	Status
Conops Report	All	9/15/24	
FSR Report	All	9/26/24	
ICD Report	All	9/26/24	
Order required hardware	All	9/15/24	
Become familiar with the CCS & sysconfig environments	All	9/21/24	
Become familiar with the motor control SDK	All	9/28/24	
Midterm Presentation	All	9/30/24	
Subsystem Research	Tyler	10/1/24	
Subsystem Research	Tamara	10/1/24	
Subsystem Research	Case	10/1/24	
Subsystem Research	Cristian	10/1/24	
Subsystem Introduction Project	Tyler	10/8/24	
Subsystem Introduction Project	Tamara	10/8/24	
Subsystem Introduction Project	Case	10/8/24	
Subsystem Introduction Project	Cristian	10/8/24	

Understand needed Functions/Data for the Estimator	Case & Tamara	10/10/24	
Have Code Outline Complete	Case & Tamara	10/12/24	
Begin porting process	All	10/12/24	
Port over Inverter & PWM dependencies to F28P65x device	Tyler	10/18/24	
Port over ADC driver dependencies to F28P65x device	Cristian	10/18/24	
Port over Estimator dependencies to F28P65x device	Case & Tamara	10/18/24	
Project Update Presentation	All	10/21/24	
Port over Inverter & PWM to F28P65x device	Tyler	10/28/24	
Port over ADC driver to F28P65x device	Cristian	10/28/24	
Port over Estimator to F28P65x device	Case & Tamara	10/28/24	
Port existing solution to F28P65x	All	10/31/24	
Begin working on the Sensorless Estimator	Case & Tamara	10/31/24	
Begin working on 16 ADC resolution	Cristian	10/31/24	
Begin Working on 64 Floating point	Tyler	10/31/24	
Finish Sensorless Estimator	Case & Tamara(All if needed)	11/21/24	
Final Presentation	All	11/18/24	
Subsystem Demo	All	11/26/24	

Final Report	All	12/5/24	

Validation Plan:

Task	Specification	Result	Owner
Test Inverter Operation	Be able to spin the motor in both directions at a high and low speed		Tyler
Test Estimator Operation	The estimator correctly collects motor information This is done by comparing with sensored motor outputs		Tamara & Case
ADC Driver Operation	ADC Correctly converts Analog signal to digital signal Through test cases and plotting		Cristian
System Compiling	Full system compile		All
Motor control With sensor Operation	The full system of a motor using a sensor operates correctly. Porting existing sensored motor control software onto 02x launch pad successfully.		All
SysConfig support	The system correctly operates using the SysConfig IDE		All
16 Bit ADC resolution	The system correctly operates using 16-bit ADC resolution		Cristian
12 Bit & 16 Bit ADC resolution Support	The system correctly operates using either 12-bit or 16-bit ADC resolutions		Cristian
64-bit Floating point operation	The system Correctly Operates using 64-bit Floating Point		Tyler
32-bit & 64-bit Floating point operation	The system Correctly Operates using either 32-bit or 64-bit Floating Point		Tyler
Test 12 vs 16 Bit performance	Test the accuracy of both 12-bit and 16-bit ADC resolutions when		All

	compared to a motor with a sensor		
Test 32-bit & 64-bit Floating point operation	Test the accuracy of both 32-bit and 64-bit ADC Floating point operations when compared to a motor with a sensor		All
Test the effectiveness of oversampling	Test the effectiveness of oversampling by comparison to the old system using a sensor(older board did not support oversampling)		All