Application Report
SLAA235A–February 2005–Revised August 2018

Fuzzy Logic Motor Control With MSP430™ MCUs

MSP430 Applications

ABSTRACT

This application report describes the implementation of a fuzzy logic control algorithm using an MSP430™ microcontroller (MCU), the MSP430F149. Fuzzy logic control algorithms can be used to solve problems that are difficult to address with traditional control techniques. As an example, a speed control system for universal motors is demonstrated.

Related source files are available from www.ti.com/lit/zip/slaa235.

Contents

1 Background Information 1
1.1 Fuzzification 2
1.2 Fuzzy Inference 3
1.3 Centroid Calculation Defuzzification Method 3
2 Hardware Description 4
3 Software Description 5
4 References 7

List of Figures

1 Membership Function 2
2 Membership Function For Error (X1[ ]) 2
3 Membership Function For dError (X2[ ]) 3
4 Output Membership Function 4
5 Output Membership Function Multiplied with Vector Y[ ] 4
6 Hardware Block Schematic 5
7 Motor Startup (No Load) 6
8 Motor Startup (Loaded) 7

Trademarks

MSP430 is a trademark of Texas Instruments.
All other trademarks are the property of their respective owners.

1 Background Information

The control of the rotation speed of serial universal motors is very complicated when done using traditional control techniques, as it requires a very complex mathematical model. Fuzzy logic eliminates the need for mathematical modeling and allows relatively easy design of a solution.

Fuzzy logic defines rules that determine the behavior of the system using word descriptions instead of mathematical equations. The algorithm consists of three steps:

1. Fuzzification
2. Fuzzy Inference
3. Defuzzification

The following sections briefly discuss these three processes. For more detailed information, refer to Fuzzy Logic: An Overview of the Latest Control Methodology.

1.1 Fuzzification

Fuzzification is the process that determines the degree of membership of the input values to defined fuzzy sets (linguistic variables). In the case of the rotation speed control of serial universal motors, the input values are:

1. Absolute error in the rotation speed

   Error = SetSpeed – CurrentSpeed

2. Differential rotation speed error

   This value is obtained by subtracting the previous error value from the current error value:

   dError = Error – LastError

In the example for this application report, five fuzzy sets are defined for the input values Error and dError:

1. NM: negative medium
2. NS: negative small
3. ZE: zero equal
4. PS: positive small
5. PM: positive medium

The membership functions (see Figure 1) are triangular-shaped and the maximum value is scaled to 400h instead of 1, which is found in other documents describing fuzzy theory. Using 400h greatly reduces the calculation complexity, because the multiplying operation becomes only one addition or subtraction.

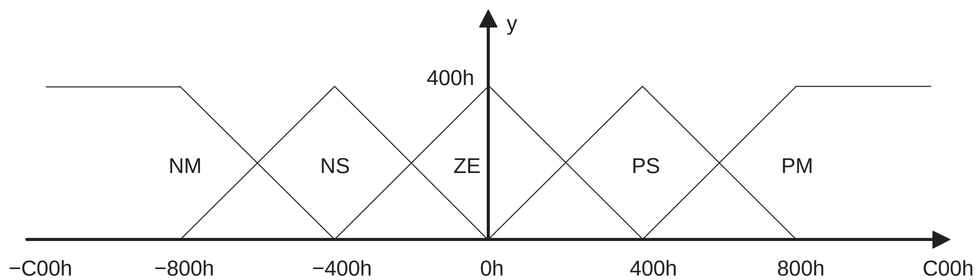

Figure 1. Membership Function

The result of the fuzzification of an input value is a vector with five elements as there are five fuzzy sets, and the value of each member defines the degree of membership of the input value to a particular fuzzy set (y-value). The vectors for the absolute and differential errors which are the results of the fuzzification are denoted as X1[ ] and X2[ ].

For example, assume Error is 30h and dError is 10h. According to Figure 2 and Figure 3, the results after fuzzification are:

X1[ ] = [0h, 0h, 3D0h, 30h, 0h]

X2[ ] = [0h, 0h, 3F0h, 10h, 0h]

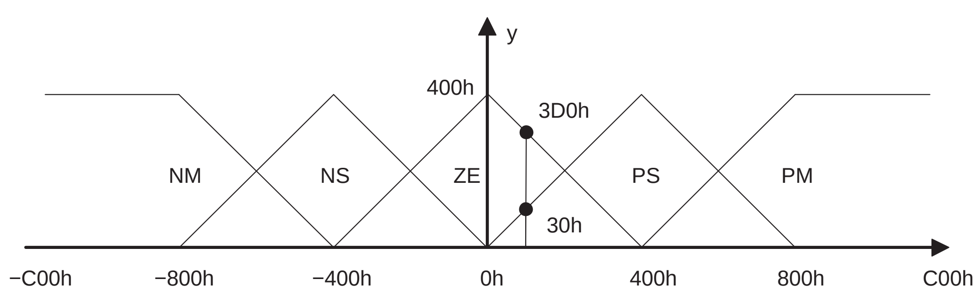

Figure 2. Membership Function For Error (X1[ ])

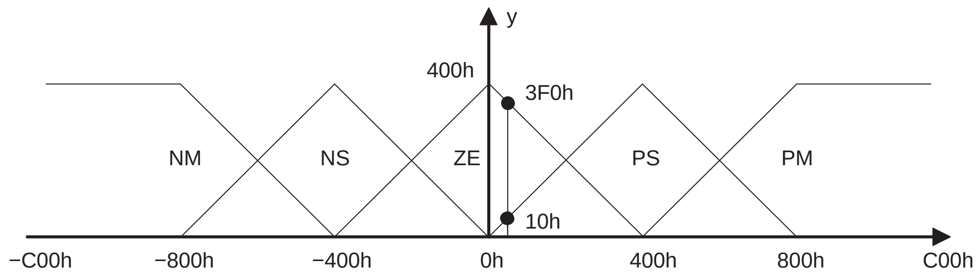

Figure 3. Membership Function For dError (X2[ ])

1.2 Fuzzy Inference

Zadeh MAX-MIN relations are used, because experience proves that they give precise results. The process consists of:

1. Clear the output vector Y[ ].
2. Take the first member of the vector X1[ ].
3. Compare it to every member of the vector X2[ ]. The smaller value is recorded on every comparison.
4. The maximum value is found among the members obtained in step 3.
5. This maximum value is then added to the output vector member which is defined by the inference rules.
6. The next member of the vector X1[ ] is taken and the process from 3 to 6 is repeated.

Table 1 lists the inference rules.

Table 1. Fuzzy Inference Rule Table

		dError, X2[ ]				
		NM	**NS**	**ZE**	**PS**	**PM**
Error, X1[ ]	NM	PM	PM	PM	PS	ZE
	NS	PM	PM	PS	ZE	NS
	ZE	PM	PS	ZE	NS	NM
	PS	PS	ZE	NS	NM	NM
	PM	ZE	NS	NM	NM	NM

In the given example, the output vector Y[ ] = [0h, 30h, 3D0h, 0h, 0h]

1.3 Centroid Calculation Defuzzification Method

The result of the defuzzification must be a numeric value that determines the duty factor of the PWM signal that drives the motor. This value is obtained by finding the centroid point of the function that is the result of the multiplication of the output membership function and the output vector Y[ ]. The general mathematical formula to obtain the centroid point is:

$$\text{Defuz} = \frac{\sum_{i=1}^{5} Y[i] \times \text{multfact}[i]}{\sum_{i=1}^{5} Y[i]} \quad (1)$$

Y[i] are the i-th members of the output vector,

multifact[i] are the multiplying coefficients of the output membership function.

Figure 4 shows a graphical representation of the output membership function as used in this application with the coefficients [-10h, -8h, 0h, 8h, 10h].

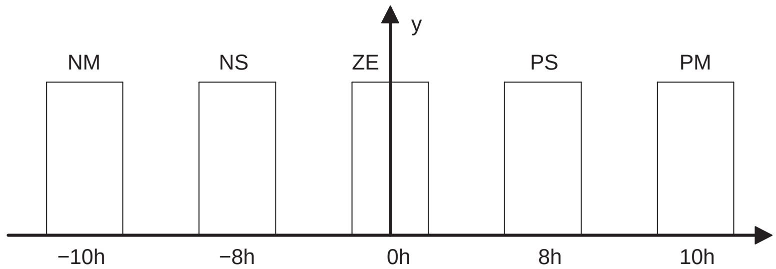

Figure 4. Output Membership Function

Using the example output vector Y[ ] = [0h, 30h, 3D0h, 0h, 0h], the following defuzzification output value can be calculated:

$$\text{Defuz} = \frac{0 \times (-16) + 48 \times (-8) + 976 \times (0) + 0 \times (8) + 0 \times 16}{0 + 48 + 976 + 0 + 0} = \frac{-384}{1024} = -0.375 \quad (2)$$

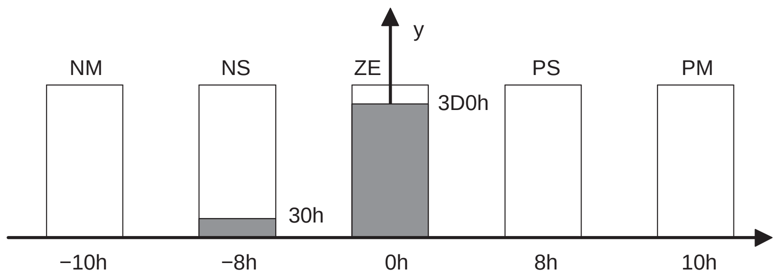

Figure 5. Output Membership Function Multiplied with Vector Y[ ]

2 Hardware Description

An MSP430F149 device is used in this motor control application. You can substitute by any other MSP430 MCU with both Timer_A and Timer_B. Figure 6 shows the application block schematic. The MSP430 MCU is sourced by an 8-MHz crystal to provide a high-resolution clock source that is used for both PWM generation and speed measurement.

The universal serial motor is driven with a PWM signal that is generated using a Timer_B capture/compare block operated in compare mode. The MCU output signal is then delivered to the actual motor driver output stage. In this application demo, this driver stage is supplied by the rectified 230-V mains voltage.

To provide feedback for the fuzzy logic control loop, an optical sensor is used. This sensor consists of a light transmitter and a receiver. Using a slotted disc that is attached to the motor shaft, 24 impulses for each motor revolution are generated. The actual motor speed is obtained by measuring the time between two successive impulses with a Timer_A capture/compare block operated in capture mode. An 8-tap moving average filter is used to minimize the measurement error that occurs with higher motor speeds.

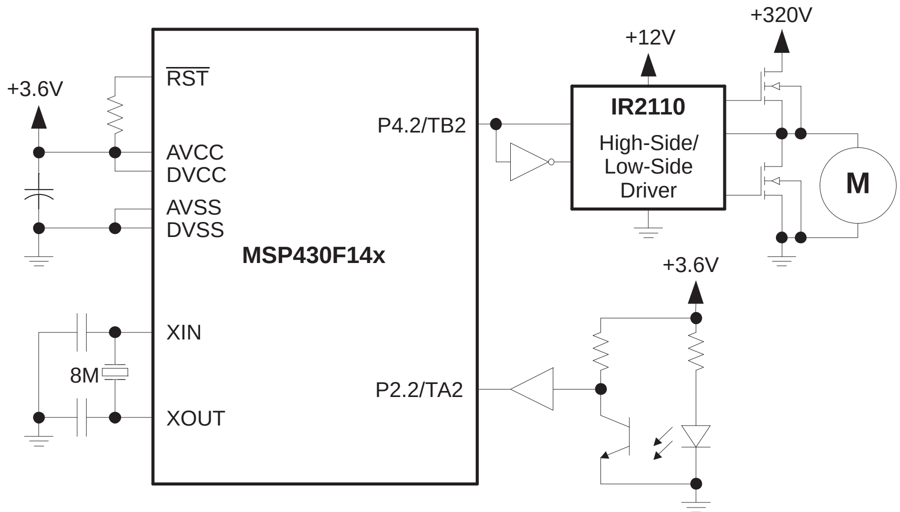

Figure 6. Hardware Block Schematic

3 Software Description

The software performs these operations:

1. Configures the MCU clock system
2. Configures the peripherals
3. Uses Timer_A to determine the actual motor speed.
   a. The rising edge of the CCI0B input (port pin P2.2) triggers capture events with Timer_A capture/compare block 0.
   b. In the associated ISR function, the precise, hardware-generated timestamp is read and processed.
   c. After passing this value through an 8-tap moving average filter, the newly calculated speed value is then stored in the global variable CurrentSpeed.
   d. Capture/compare block 1 is configured to compare mode and used to provide a time-out mechanism for the speed measurement.

      This timeout is needed if the motor is not running, because no capture events would be generated, and the software would wrongly assume that the motor is still running. However, this timeout updates the global variable CurrentSpeed, and the control algorithm continues to work. Timer_A is operated in continuous mode to avoid that any software latency affects the measurements.
   e. Capture/compare block 2 of Timer_B is configured to compare mode and used for PWM signal generation.

      Timer_B is operated in up mode and driven by the 8-MHz clock source. With the period register set to 3999, the effective PWM output frequency is 8 MHz / (3999 + 1) = 2 kHz. This frequency gives a fine granularity to adjust the motor power, resulting in a smoother control. For this application, the capture/compare block selection of Timer_A and Timer_B is arbitrary.

      For detailed information on timer configuration and operation, see the MSP430x1xx Family User's Guide.
4. Configures the MSP430 watchdog timer (WDT) module as interval timer. Its associated ISR function is called 244 times per second and wakes the CPU from low-power mode LPM0. This mechanism provides the time interval between each control cycle.
5. Enters the control loop. Immediately upon entering, the MCU enters LPM0 to turn off the CPU and stop program execution while waiting for the wakeup from the WDT ISR.
6. On wakeup, calculates the absolute and differential control loop errors Error and dError based on target speed (variable SetSpeed), current speed (variable CurrentSpeed), and the previous error value (variable LastError).
7. The function Fuzzification() transforms these error values into fuzzy vectors X1[ ] and X2[ ].

8. The fuzzy inference rules are applied, and the function FuzzyInference() generates the fuzzy output vector Y[ ].
9. The function Defuzzification() transforms this output vector back into a single control loop output value that is added to the current PWM duty cycle. This process closes the control loop.

   The two definitions PWM_Min and PWM_Max are used to limit the motor duty cycle and may need to be adjusted depending on the application and load conditions.

For a discussion of the fuzzy algorithms, see Section 1.

Figure 7 and Figure 8 show example motor startup operations using the test setup developed to verify this application report. The motor target speed is set to 50 rotations per second. The graphs show the startup without a mechanical load and with a mechanical load.

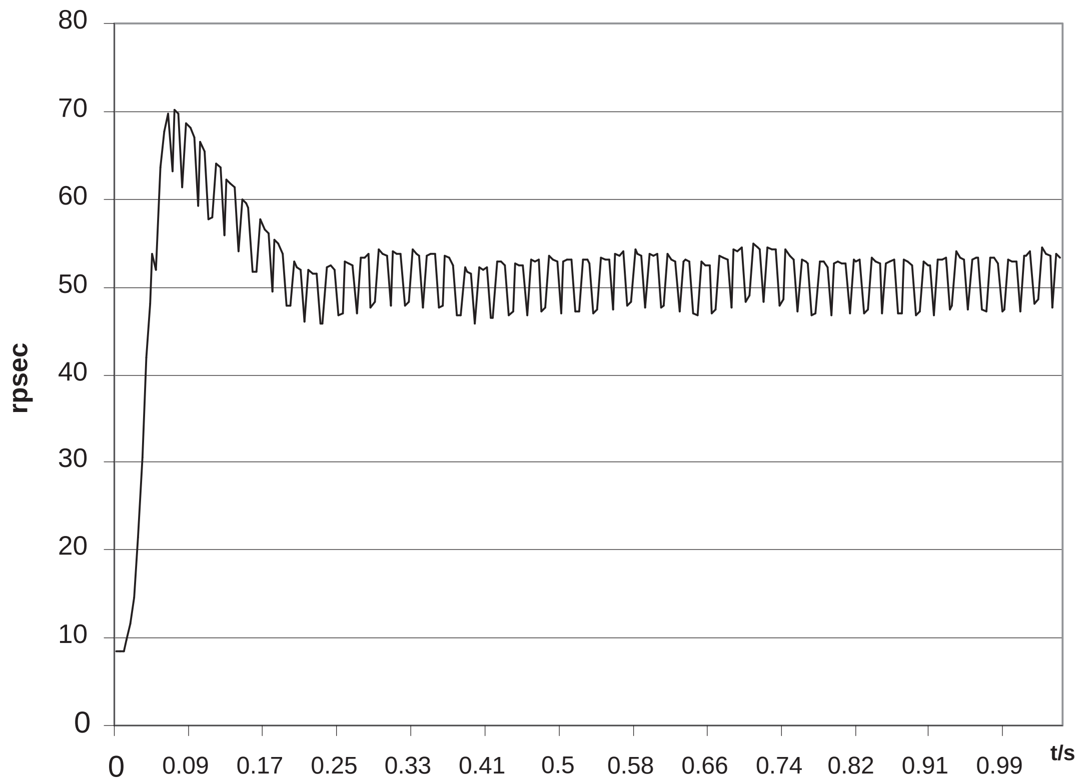

Figure 7. Motor Startup (No Load)

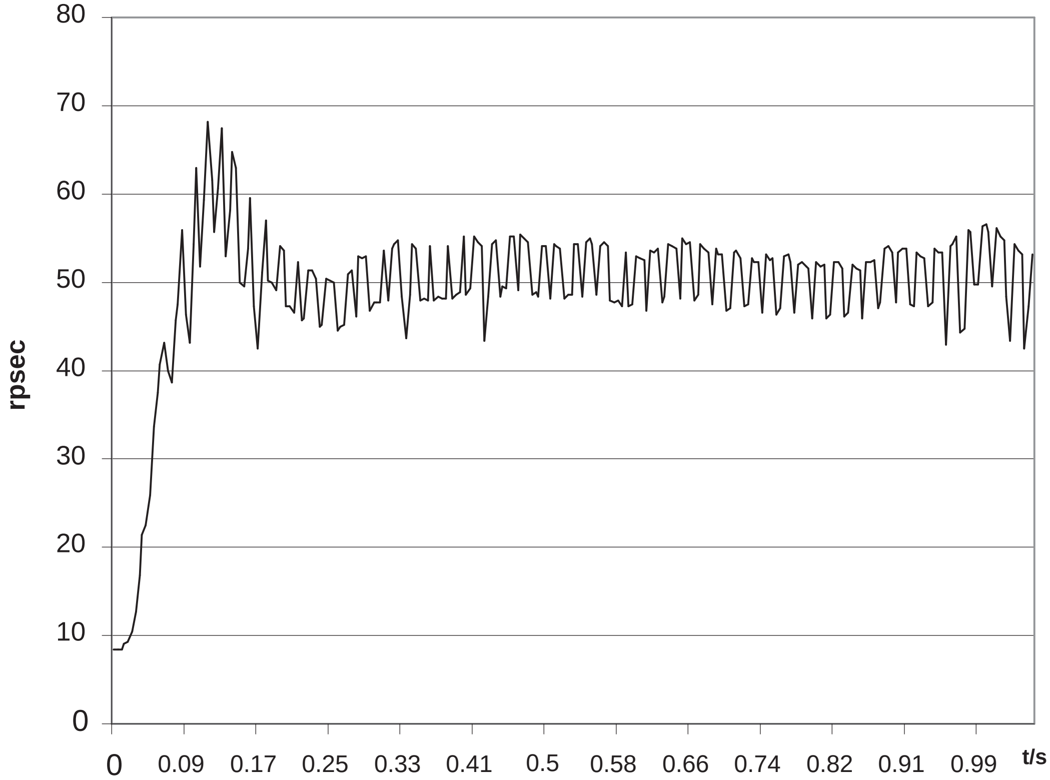

Figure 8. Motor Startup (Loaded)

4 References

1. Fuzzy Logic: An Overview of the Latest Control Methodology
2. MSP430F13x, MSP430F14x, MSP430F14x1 Mixed-Signal Microcontrollers data sheet
3. MSP430x1xx Family User's Guide

Revision History

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

Changes from February 17, 2005 to August 3, 2018	Page
• Editorial and formatting changes throughout document	1

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

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