Application Report
SLAA235−February 2005

Fuzzy Logic Motor Control with MSP430x14x

Andreas Dannenberg, Courtesy of: Dr. Odry Peter, Diveki Szabolcs, Csasznyi Andor, Burany Nandor   *MSP430*

ABSTRACT

This application report discusses the implementation of a fuzzy logic control algorithm using an MSP430F14x device. Fuzzy logic control algorithms can be used to solve problems that are difficult to address with traditional control techniques. As an example, a universal serial motor speed control system is demonstrated.

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. Using Fuzzy logic eliminates the need for mathematical modeling and allows easy realization 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 in this chapter briefly discuss these three processes. For more detailed information, please refer to [1].

1.1 Fuzzification is the process which 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 this application demo, 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 (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. This way the calculation complexity is greatly reduced because the multiplying operation becomes only one addition or subtraction.

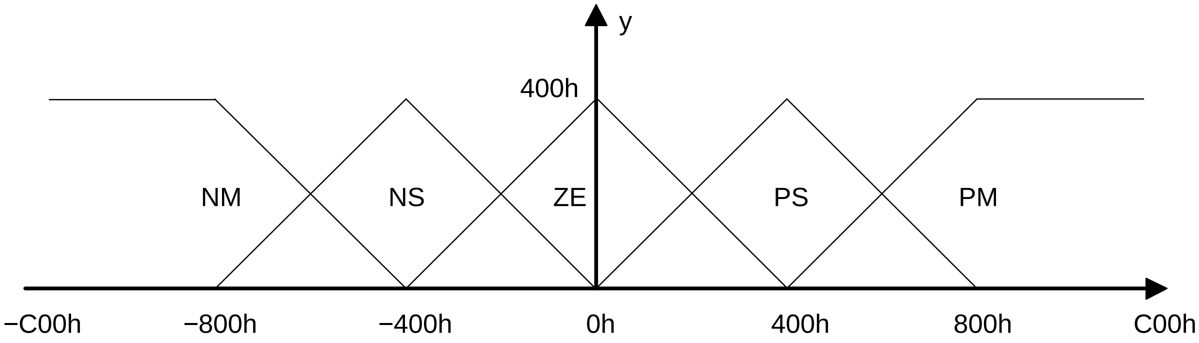

Figure 1. Membership Function

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[ ]. Example: Let Error be 30h and the dError be 10h. In this case according to Figure 2 and Figure 3, the results after fuzzification are:

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

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

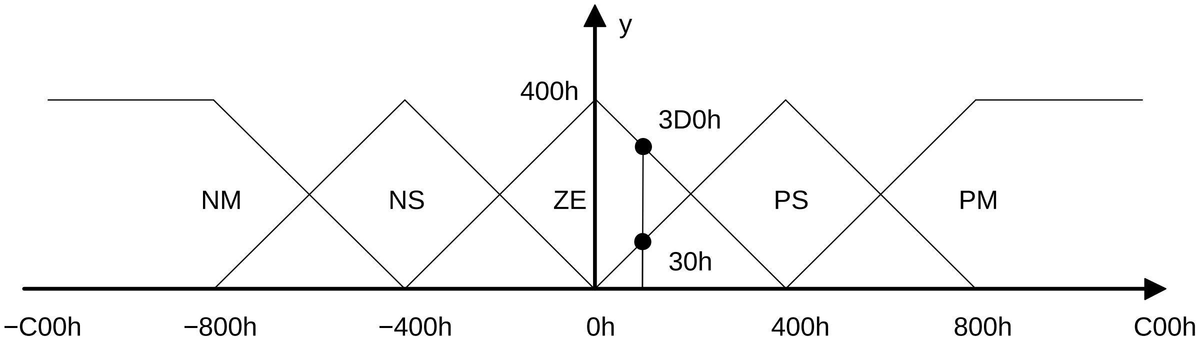

Figure 2. Membership Function For Error (X1[ ])

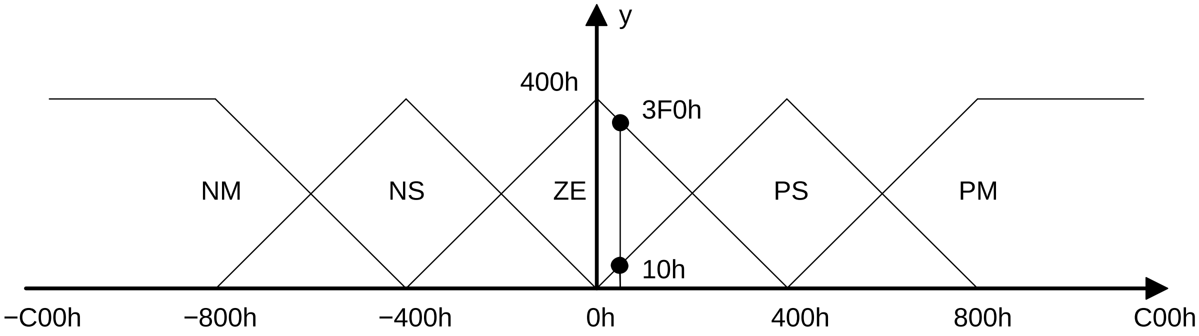

Figure 3. Membership Function For dError (X2[ ])

1.2 Fuzzy Inference

Zadeh max-min relations are used because the experience proved 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.

The inference rules are shown in Table 1.

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[ ] is: Y[ ] = [0h, 30h, 3D0h, 0h, 0h]

1.3 Centroid Calculation Defuzzification Method

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

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

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].

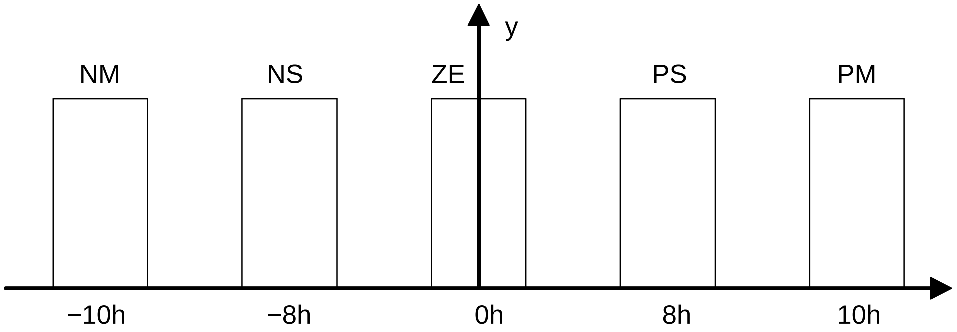

Figure 4. Output Membership Function

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

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

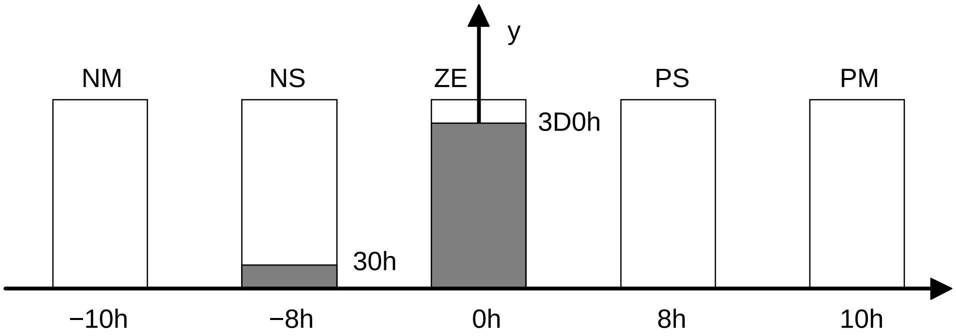

Figure 5. Output Membership Function Multiplied with Vector Y[ ]

2 Hardware Description

An MSP430F149 device is used in this motor control demo application. It can easily be substituted by any other MSP430 device with both Timer_A and Timer_B available. Figure 6 shows the application block schematic. The MSP430 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 MSP430 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 both light transmitter and receiver. Using a slotted disc which 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.

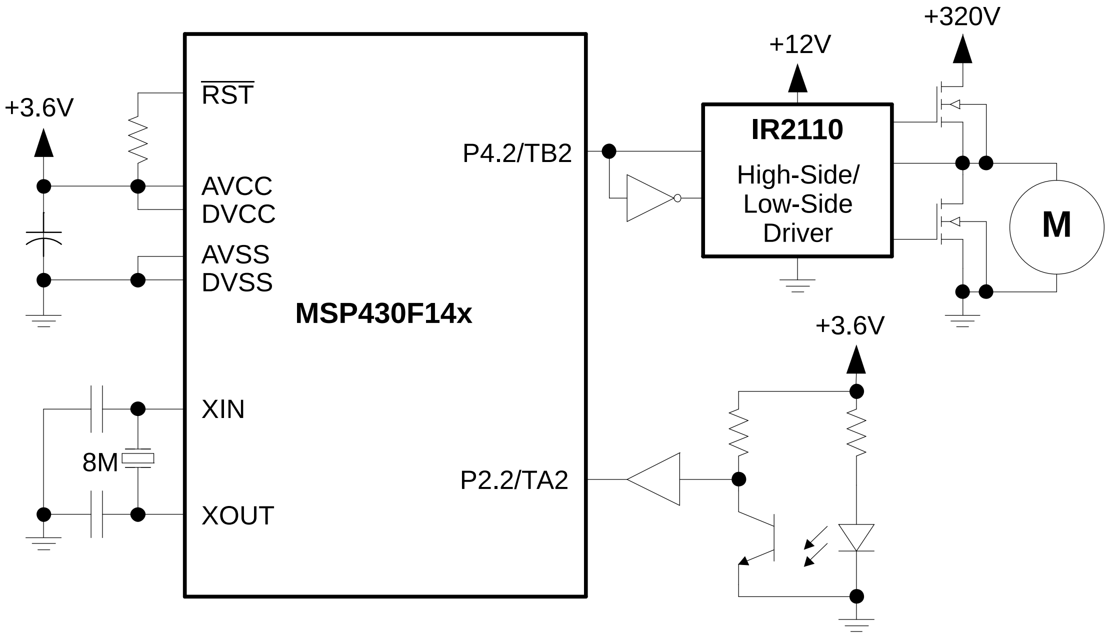

Figure 6. Hardware Block Schematic

3 Software Description

After setting up the MSP430 clock system, the peripherals are configured. Timer_A is used to determine the actual motor speed. The rising edge of the CCI0B input (port pin P2.2) is used to trigger capture events with Timer_A capture/compare block 0. In the associated ISR function, the precise, hardware-generated timestamp is read out and processed. After passing this value through an 8-tap moving average filter, the newly calculated speed value is then stored in the global variable CurrentSpeed. Capture/compare block 1 is configured to compare mode and used to provide a time-out mechanism for the speed measurement. This is needed because in the case the motor is standing still, no capture events would be generated, and the software would wrongly assume that the motor is still running. However, with implementing this timeout, the global variable CurrentSpeed still gets updated and the control algorithm will continue to work. Note that Timer_A is operated in continuous mode to avoid that any software latency affects the measurements. The 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 8MHz clock source. With the period register set to 3999, the effective PWM output frequency is 8MHz / (3999 + 1) = 2kHz. This setup gives a fine granularity to adjust the motor power, resulting in a smoother control. Note that for this application, the capture/compare block selection of Timer_A and Timer_B is arbitrary. Also, for detailed information on timer configuration and operation, please refer to [3].

Next, the MSP430 watchdog timer (WDT) module is configured as interval timer. Its associated ISR function is called 244 times a second and used to wake up the CPU from low-power mode LPM0. This mechanism provides the time interval between each control cycle.

After peripheral configuration, the actual control loop is entered. Immediately upon entering, low-power mode 0 is activated, thus switching off the CPU and stopping program execution while waiting for the wakeup caused by the WDT ISR.

On wakeup, the absolute and differential control loop errors ‘Error’ and ‘dError’ are calculated based on target speed (variable SetSpeed), current speed (variable CurrentSpeed), and the previous error value (variable LastError). These error values are then transformed into Fuzzy vectors X1[ ] and X2[ ] using the function Fuzzification(). After fuzzification, the fuzzy inference rules are applied and the Fuzzy output vector Y[ ] is generated through calling the FuzzyInference() function. This output vector is then transformed back into a single control loop output value by calling Defuzzification() and added to the current PWM duty cycle. This way the control loop is closed. Note that 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 used Fuzzy algorithms, please refer to section 1.

The graphs Figure 7 and Figure 8 show an example motor startup of the test setup that was used to develop and verify this application report. The motor target speed is set to 50 rpsec, and the graphs show the startup without and with a mechanical load attached.

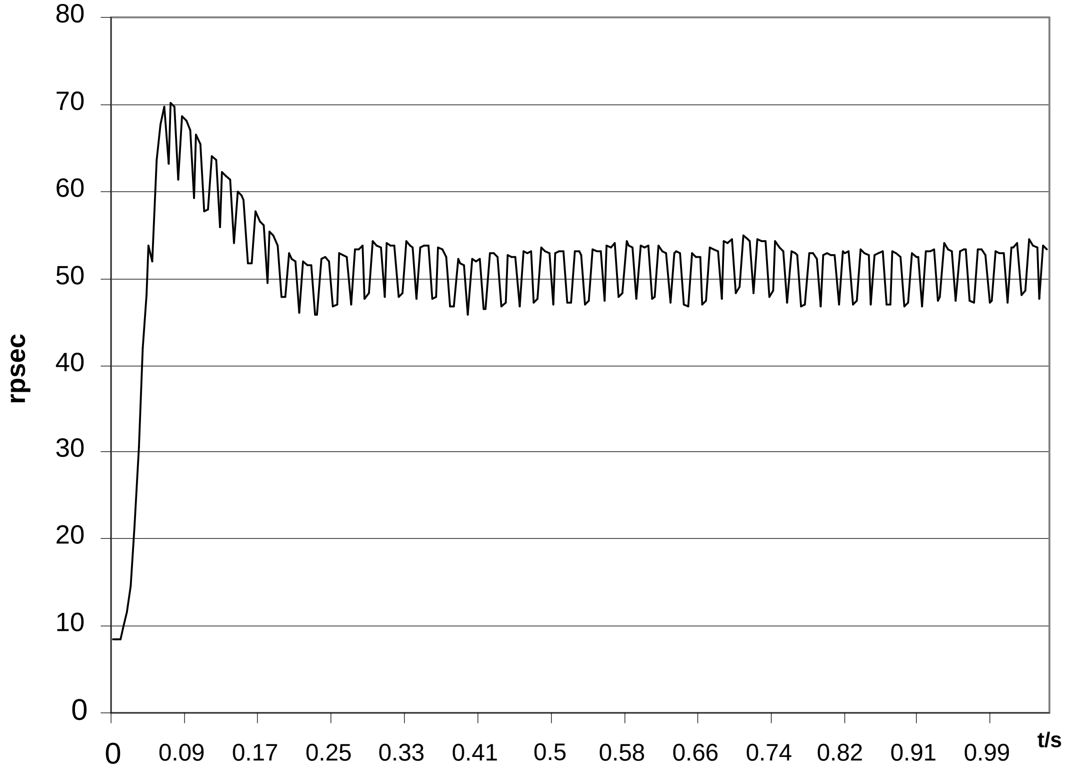

Figure 7. Motor Startup (No Load)

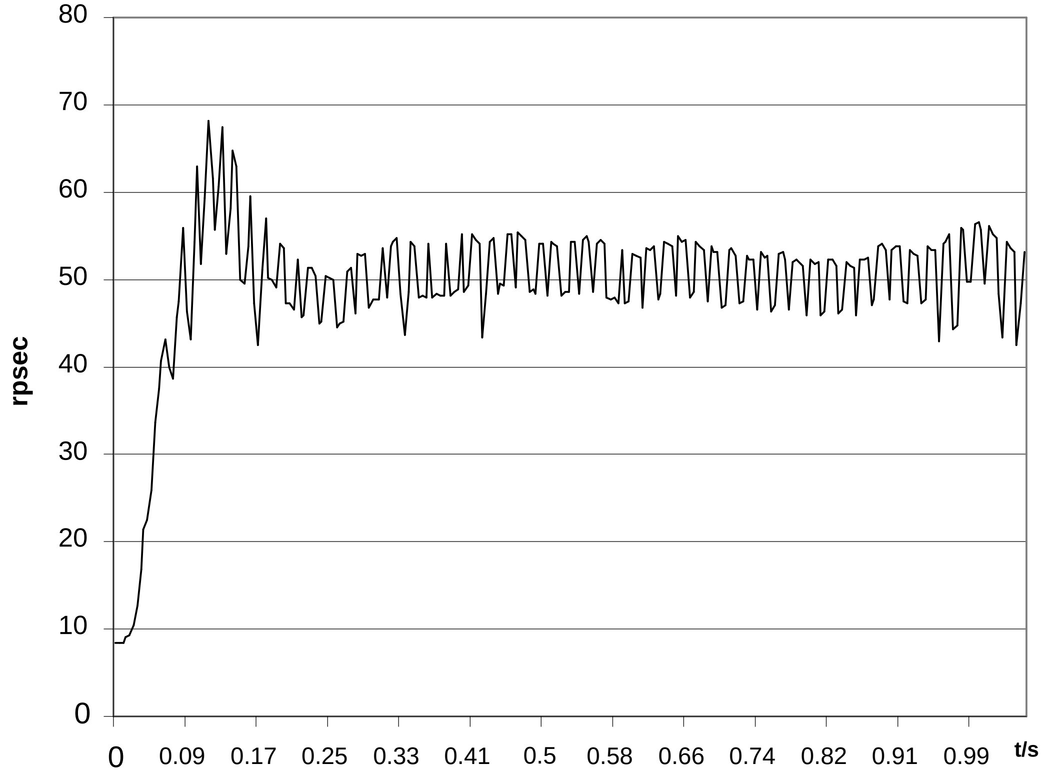

Figure 8. Motor Startup (Loaded)

4 References

1. Fuzzy Logic: An Overview of the Latest Control Methodology (SPRA028)
2. MSP430x13x, MSP430x14x Mixed Signal Microcontroller Data Sheet (SLAS272)
3. MSP430x1xx Family User’s Guide (SLAU049)

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information 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.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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