Fuzzy control of brush motor - problems of computing

Odry Péter, Divéki Szabolcs, Burány Nándor, Gyantár László

Műszaki Főiskola, Szabadka, Marka Oreskovica 16, Serbia & Montenegro

odry@vts.su.ac.yu

Abstract: This paperdeals with the problem of fuzzy control of universalmotors and with the application of microcontrollers and minimum electronic (buck converter). The article discusses the issues of realizing the program neighborhood.

Keywords: Fuzzy logic, brush motor, speed control, computations problems, microcontroler realization

1 Why fuzzy logic control?

The rotation speed control of serial universal motors is very complicated with traditional control technique, because it requires the mathematical model of control object, which is very complex. Therefore, it is simplier to solve this problem with fuzzy logic which doesn't require it [1]. We have used fuzzy for cheap brush motors and got excepcionally good performances, similar to servo motor applications [2] and we have built in even industrial enviroments. In concrete (specific) industrial applications we had some delicate problems we could not solve with the application of classic PID control (for example, when we were making a mechanic ballance equipment and when we had to control the rpm of brush motors in the neighborhood of mechanical resonance with changing rpm elements which had unknown torque values, resonanrt frequenciees debalance values - in order to avoid the breakdown of the equipment), so we used a modified version of fuzzy (neuro-, genetic-,…) control [3], [4]. We used DSP technology for this case.

2 Short description of hardware

Signal which controls the rotation speed of the motor is a PWM signal. It is led to the input of th DC/DC driver which moves the brush motor (figure 1).

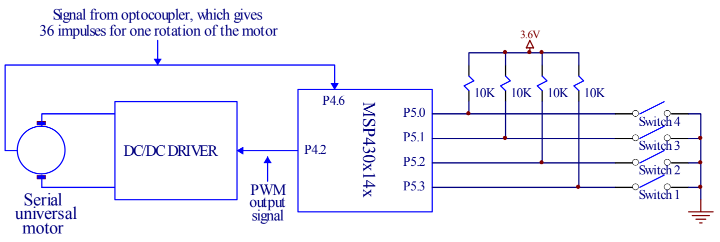


Figure 1. Blok diagram of relized fuzzy control system

Our aim was to make a system with minimum driver with relation to the full bridge. In case of a full bridge solution we have aan electronically-controlled braking, however, the buck cconverter soluttion has an uncontrolled braking with is obtained mechanically. The effect of uncontrolled braking on the ccontrol of rpm can be easily seen by ccomparing Figures 2 and 3.

The feedback signal for the fuzzy control system provides an opto-coupler. It has 24 equally positioned gaps. So it generates 24 impules for a whole rotation of the motor. We get information about the speed rotation from measuring the time passed between two succesive impulses.

3 Short description program

The parameters which define the behaviour of the system are:

```
#define PWM_Period  3999      ;The frequency of PWM is: 1/(8E+6*(3999+1))=2KHz
#define PWM_Max     2800      ;Determins the minimum duty factor of  PWM signal
#define PWM_Min     3850      ;Determins the maximum duty factor of  PWM signal
```

PWM_Period: The value of the constant PWM_Period defines the frequency of the PWM signal based on the formula:

$$f_{PWM} = \frac{8000000}{PWM\_Period + 1} \tag{1}$$

To be more accurate in control of the rotation speed we need higher resolution which means lower frequency of the PWM signal. On the other hand, for regular work of the DC-DC driver, the frequency of the PWM signal mustn't be too low. Experience proved that the value 3999 for PWM_Period satisfies both conditions.

Based on the results of defuzzification the new value for duty factor of PWM signal is obtained by:

$$\text{New\_PWM\_Set} = \text{Old\_PWM\_Set} - \text{Defuz} \qquad (2)$$

PWM_Set belongs to range [0, 1, ... ,PWM_Period]. Its value defines the moment of setting the PWM signal. If the value of PWM_Set is lower, the duty factor will be higher and vice versa.

In case the rotation speed of the motor is higher than the desired, the defuzzification process results in negative value (Defuz < 0), which according to eq. (2) gives a lower duty factor, so the rotation speed of the motor will be decreased.

PWM_Max, PWM_Min: The program is written the way that at the moment of start-up PWM signal has the duty factor PWM_Max, to reach the desired speed faster. The duty factor remains PWM_Max while Error < -C00h. On the other hand, if the rotation speed of the motor is too high (Error > C00h) the duty factor becomes PWM_Min. It means that the fuzzy logic regulation is active only when the error in rotation speed belongs to boundaries [-C00h, C00h]. Figures 2 and 3 showes the reactions of the motor as a function of time to Hevisaid excitation when the desired speed is 80 rpsec.

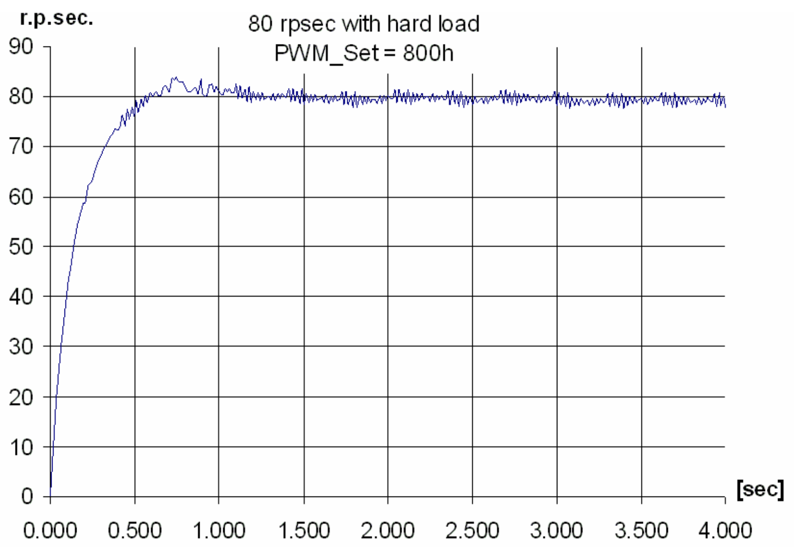


Figure 2. RPM values when the motor is heavily loaded and PWM_Max = 800h.

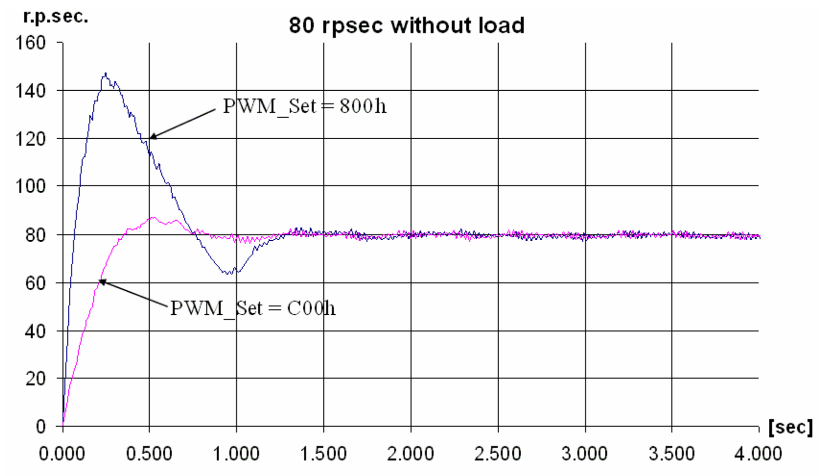


Figure 3. RPM values when the motor is not loaded and PWM_Max = 800h and PWM_Max = C00h

Figure 3 shows the case when the motor is not loaded and PWM_Max = 800h and PWM_Max = C00h. Figure 7 for the case PWM_Max = 800h shows significant excess in the moment of start-up while for PWM_Max = C00h it doesn't appear. It happens because the constant value PWM_Max determines the power of the motor. It means that when PWM_Max is small (the duty factor is high) and the load small, the motor gets too much acceleration in the start-up moment which leads to a significant excess in speed value. This problem is solved by increasing the value PWM_Max, as it is shown on Figure 3, when PWM_Max = C00h. In the other case when the duty factor is small (PWM_max high) and the load is heavy, the speed cannot reach the desired value because the motor has no enough power.

4 Problems occurred during programming

4.1 Division of speed control range

During the realization of the fuzzy logic regulation it turned up that the division of speed control range is necessary. The ranges we got are:

1.15–40rpsec.
2. 40 - 120 rpsec.

The same parameters which give excellent results in speed control less than 40 rpsec(rotation per second), for higher rpsec the oscillations in speed value rises as the desired speed value grows. This kind of behaviour is a logical consequence of rotation speed measuring method. Number of impulses generated by TimerB are counted between two gaps on the encoder as the rotor revolvs. If the motor rotates slowler, TimerB can counts up to some higher value. If the rotation speed is 20rpsec, TimerB counts up to 823h, for 40 rpsec to 411h and for 80rpsec to 208h. The resolution is decreased in higher speed control range.(the relative error is four times higher for one impulse at 80 rpsec than at 20 rpsec). To avoid this problem at higher speeds impulses are counted between 8 gaps. Then the number of impulses for 80rpsec is 208h*8=1040h. The speed measuring accuracy is increased which allows smoother control at higher speed. This change allows speed control up to 120 rpsec.

The problem which occures with speed measuring slower than 40 rpsec counting impulses between 8 gaps to simplify the program is the increase of the measuring time which causes significant oscillations around the desired value.

4.2 Problems with implementing fuzzyfication algorithm

There are two input values for fuzzy control. Error in current speed value and the differential of error in current speed value. The two input values need two different membership function. Practical analysis showed that the membership functions presented on figures 8 and 9 give the best results in speed control.

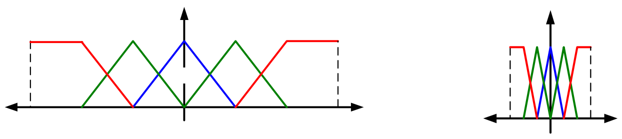


Figure 4. Membership functions of Error and ErrorDiff

The algortihms for fuzzyfication Error and ErrorDiff input values are identical but their parameters are different. A simple solution would be to write the same algorithm twice with different parameters, but there is a question: how to save in using flash memory or in other words: is it possible to use one algorithm with same parameters for both input values. If we notice that multiplying the input value ErrorDiff with 16 (or shifting right for four binary places) and using the membership function of the input value Error gives the same result as ErrorDiff with its own membership function, then the problem of fuzzyfication can be solved using half of the memory resources.

4.3 Amount of neccesary flash memory for the operation of dividing in assembler

In the process of defuzzification the centroid calculation method is applied using eq. (3):

$$\text{Defuz} = \frac{\sum_{i=1}^{5} Y[i] \cdot \text{multfact}[i]}{\sum_{i=1}^{5} Y[i]} \tag{3}$$

Microcontroller MSP430x14x owns a 16-bit hardwer multiplyer which can be used for dividing appling Q15 format. The needed amount of flash memory for storaging numbers in Q15 format depends on the range of the sum $\sum_{i=1}^{5} Y[i]$. If the membership function is given on figure 8 and if Zadeh max-min relations are used for fuzzy inference, which range of numbers presented in Q15 format has to be saved in the memory?

In the first approximation, range of numbers which would certainly contain all the possible result of $\sum_{i=1}^{5} Y[i]$ is from 0 to 800h. The lower limit is certainly greater or equal to zero because the sum of five nonnegative number is always greater or equal to zero. The upper limit is certainly less or equal to 800h because the output vectors of fuzzyfication contain up to two nonnegative numbers with maximum value of 400h.

Is it possible to reduse this range to save memory? Experimental results indicate that it can be redused. Figure 5 shows 500 successive recorded values of the sum to the flash memory of the microcontroller.

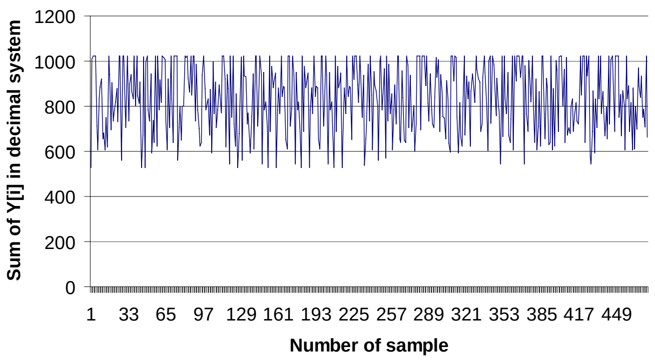


Figure 5. Sum value presentation in memory domain

According to graph on figure 5 the sum never goes out of range [512(200h), 1024(400h)]. Can it be proved mathematically?

1.Let

$$\begin{matrix} X_1 = [0,0,X_{11},X_{12},0] \\ X_2 = [0,0,X_{21},X_{22},0] \end{matrix} \wedge X_{11} + X_{12} = 400\text{h} \wedge X_{21} + X_{22} = 400\text{h}.$$

$$\text{If } X_{11}, X_{21} \in [200\text{h}, 400\text{h}] \wedge X_{11} > X_{21} \text{ then}$$

$$X_{12} < X_{22} \text{ then}$$

$$\sum_i Y[i] = A, \quad 400\text{h} \geq A \geq 200\text{h}$$

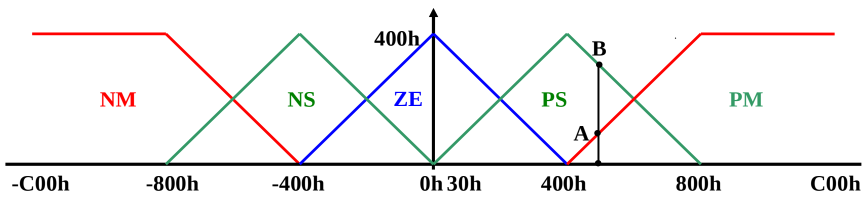


Figure 6. Representing a pair of values A and B

The equations $X_{11}+X_{12} = 400\text{h}$ and $X_{21}+X_{22} = 400\text{h}$ are always true (In our case the sum of the input membership function A+B for every input value will be 400h as it is presented on figure 6), because of the shape and position of input membership functions. Then:

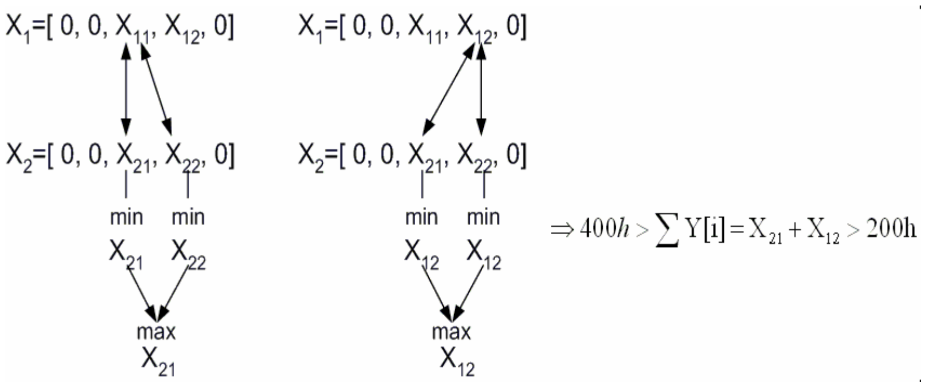


Figure 7. Represent calculation of Let 1.

The position of the elements X_{11} and X_{12} in vector X1 is unimportant for proof because the result of max-min aproximation will be the same.

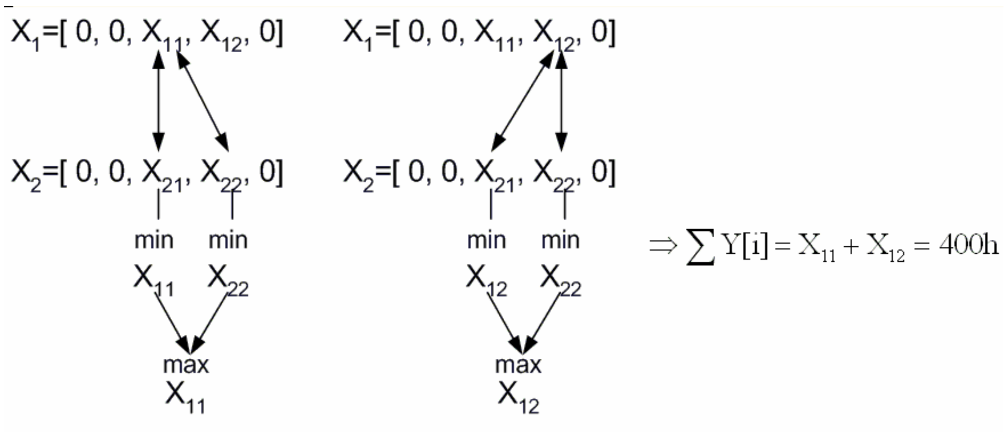
2. Let

$$\begin{matrix} X_1 = [0,0,X_{11},X_{12},0] \\ X_2 = [0,0,X_{21},X_{22},0] \end{matrix} \wedge X_{11} + X_{12} = 400h \wedge X_{21} + X_{22} = 400h.$$

If $X_{11}, X_{21} \in [200h, 400h] \wedge X_{11} < X_{21}$ then

$X_{12} > X_{22}$ then

$\sum_i Y[i] = 400h.$



X_1=[0, 0, X_{11}, X_{12}, 0]
X_2=[0, 0, X_{21}, X_{22}, 0]
min
min
X_{11}
X_{22}
max
X_{11}
X_1=[0, 0, X_{11}, X_{12}, 0]
X_2=[0, 0, X_{21}, X_{22}, 0]
min
min
X_{12}
X_{22}
max
X_{12}
$\Rightarrow \sum Y[i] = X_{11} + X_{12} = 400h$

Figure 8. Represent calculation of Let 2.

3. Let

$$\begin{matrix} X_1 = [0,0,X_{11},0] \\ X_2 = [0,0,X_{21},X_{22},0] \end{matrix} \wedge X_{11} = 400h \wedge X_{21} + X_{22} = 400h.$$

If $X_{21} \in [200h, 400h]$ then $\sum_i Y[i] = A, \quad 400h \geq A \geq 200h.$

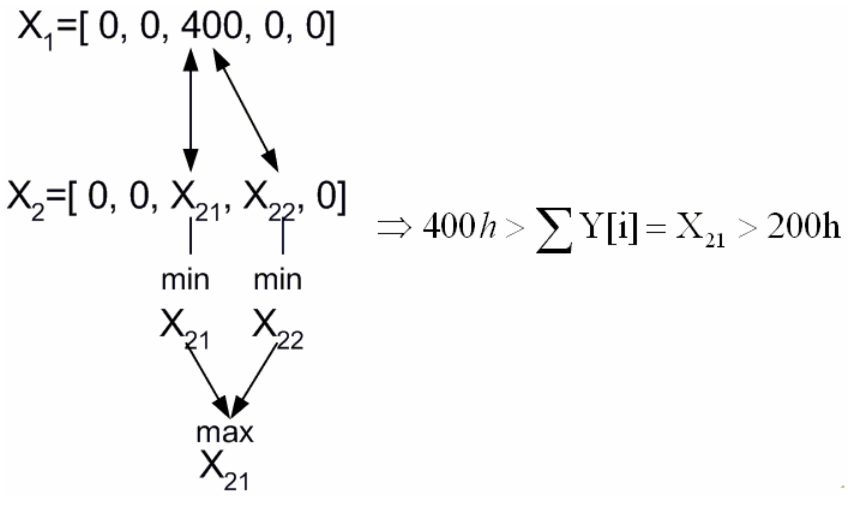


Figure 8. Represent calculation of Let 3.

All the possible values of vectors X_1 and X_2 are covered in this proof, therefore the sum of elements of Y vector is always in boundaries from 200h to 400h. It means that 512 values are enough for the operation of dividing. Every number presented in format Q15 is size of 2 byte, so the necessary amount of flash memory is only 1Kbyte.

4.4 Problems with PWM output of MSP430F148

Timer_B of the microcontroller is used for the PWM signal generating which controls the rotation speed of the motor through the DC-DC driver. The value of register TBCCR0 determins the period of the PWM signal while the value of register TBCCR2 determins it's duty factor. In C program OUTMOD_7 was firstly used for generation the PWM signal whose function is showed on figure 9. The next few lines are taken from C program code which sets up the Timer_B.

```
// Setup Timer_B for PWM generation
 TBCCR0 = PWM_Period;    // PWM Period
 TBCCR2 = PWM_Max;       // PWM Duty Cycle
 TBCCTL2 = OUTMOD_7;
```

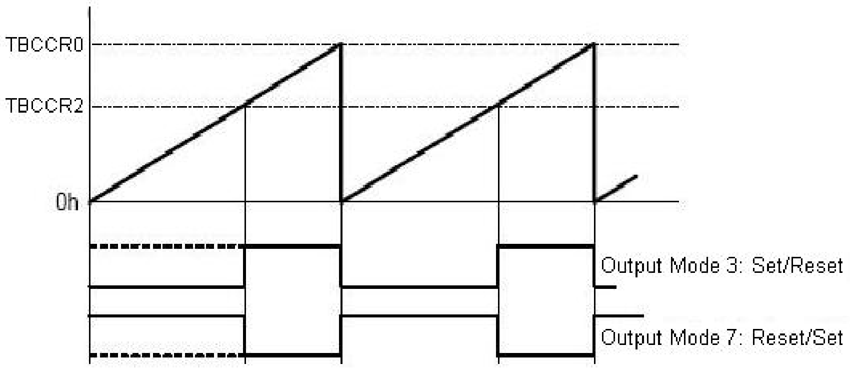


Figure 9. Selection output mode in PWM signal generations

When the program was tested an intresting problem appeared. While the motor was revolving a quit knocking sound was present which caused significant oscillations in rotation speed. The cause of this problem was that the PWM signal was setted for a whole period in random moments without any apparent reason. According to algorithm this situations couldn't happen. To solve this problem we reversed the logic of generating PWM signal from OUTMOD_7 to OUTMOD_3 and the problem vanished. The only conclusion for this behaviour remaind that an error has been made in the hardware of OUTMODE_7 of the microcontroller.

Conclusions

Based on the above, we can conclude that by using a simple construction (brush motor, buck converter, microcontroller and simple fuzzy algoritm) we obtain a ssystem wich has acceptable responses on the Hevisaid's exitation without the need of adaptation to a wider range of load and rpm.

References

[1] *Mathew George, Jr.:* **Implementation of Fuzzy Logic on Servo Motor Control,** *Selected Applications***,** SPRA028, Texas Instruments, January 1993

[2] *S. Beierke, R. Königbauer, B. Krause, C. Altrock:* **Enhanced Control of an Alternating Current Motor Using Fuzzy Logic and a TMS320 Digital Signal Processor**, SPRA057, Texas Instruments, March 1996

[3] Szilveszter Pletl : **Some Robot Control Algorithms Based on Fuzzy, Neural and Genetic Methods**, IEEE International Conference on Computational Cybernetics, ICCC 2003, pp.135-140, Siófok 2003.

[4] Márta Takács: **Approximate reasoning based on distance based operators and sismilarity measures**, in Prrinciples of Fuzzy Preference Modelling and Decision Making, edited by Bernard de Baets and Janos Fodor, Academic Press, Gent, 2003, ISBN 90-382-0567-8, pp: 83-94.