SGS-THOMSON MICROELECTRONICS

APPLICATION NOTE

An Approach To Motor Control With fuzzy LOGIC

P. GUILLEMIN

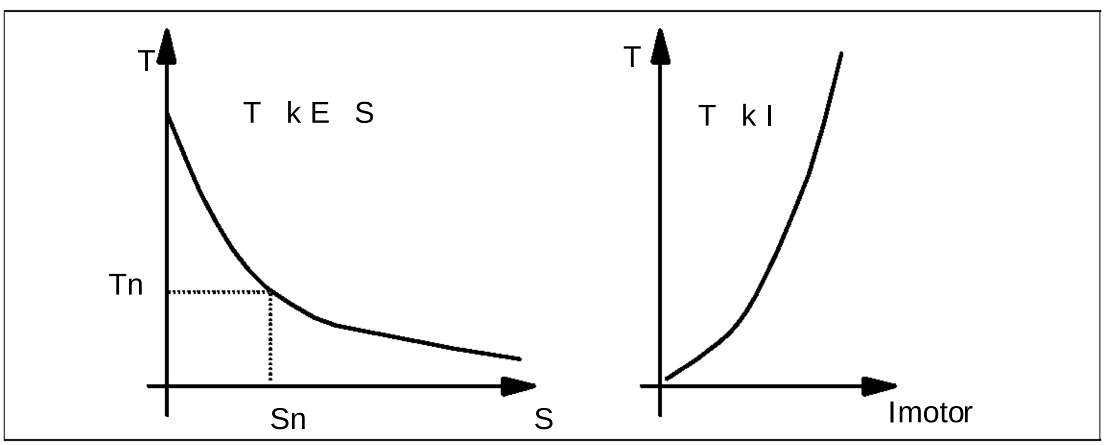
INTRODUCTION

Today home appliance applications require more and more features such as motor speed control, motor speed adaptation to accessories, an efficient and easy to use human interface and security features. These new requirements can be achieved by using electronic controls. This paper describes a universal motor speed control implemented on a standard micro-controller running software using the fuzzy LOGIC concept.
The different stages of development of this motor speed control using fuzzy LOGIC are described with an SGS-THOMSON ST6 micro-controller (MCU) and a fuzzy logic development tool: the ”fuzzyTECH ST6 Explorer Edition”, from the basic knowledge of the system to the first results.

1 WHY CONTROL MOTOR SPEED?

Most motors used in food-processors are universal motors (brush motors with serial excitation) supplied in AC or in DC mode. The stator windings of such a motor are connected in series with the rotor. The flux is proportional to the motor current and the motor torque is proportional to the square of the current. Therefore the motor speed is largely sensitive to torque variations as shown in Figure 1.

Fig 1. Universal Motor Basic Characteristics

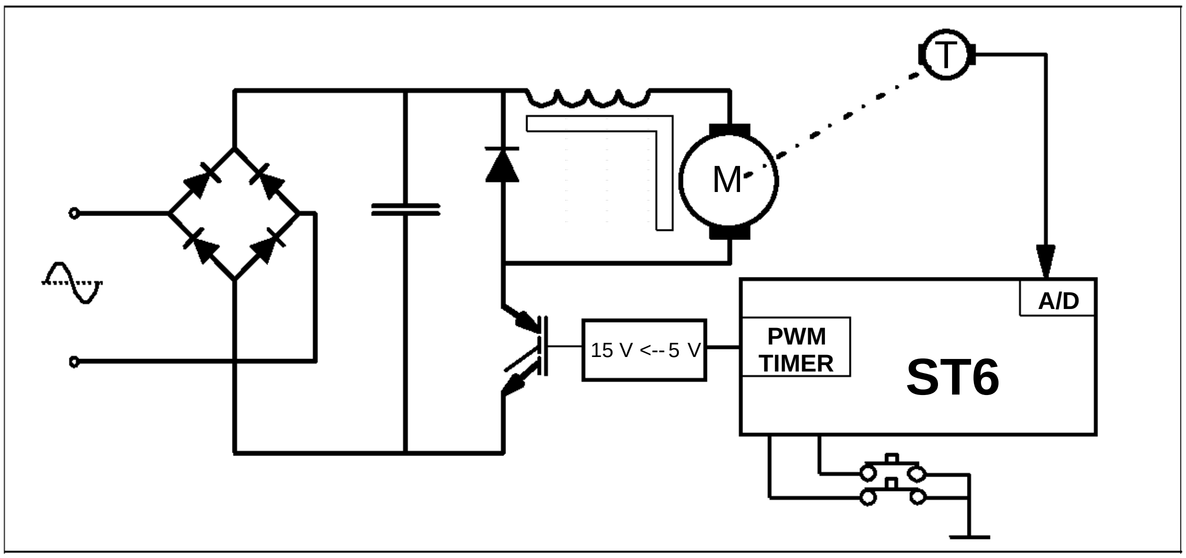


To keep the motor speed stable upon load variations, bringing a more comfortable use of the food-processor and improved cooking mixture, motor speed control is now implemented in new generations of food-processors.
This speed control can be achieved by adjustment of the motor voltage by ”phase angle” techniques or ”Pulse Width Modulation” (PWM) techniques.

2 PRACTICAL APPLICATION

Figure 2 describes a fuzzy logic motor speed control application. The 400W universal motor is supplied in DC mode. This voltage is adjusted in P.W.M. through a chopper stage composed of an Insulated Gate Bipolar Transistor (IBGT) STGP10N50 and a freewheeling diode STTA806DI. A standard micro-controller, an ST6265 with an on-board PWM timer, directly manages the IGBT through a 5/15V interface.

Fig. 2. St6 Measures Tacho Voltage And Updates Pwm Duty Cycle To Control



The motor speed can be measured by means of:

- the tachogenerator frequency and the MCU timer,
- the tachogenerator voltage and the MCU Analog to Digital (A/D) converter.

The simplest solution consists in using the A/D converter (the maximum speed of the motor corresponding to the full scale of the A/D converter i.e. 5V). The tachogenerator signal is amplified, filtered and then converted to a DC voltage by means of a simple frequency/voltage converter. This solution avoids timer and prescaler programming changes and motor speed calculation due to the large amplitude of the speed variations.

The speed measurement is done periodically and synchronized by an on-chip timer acting as a time base. This time base is defined taking into account the duration of the execution of the fuzzy rules (≈12, 15ms), the duration of the motor speed change routine and the duration of other functions for exemple data byte transmission (explained later). This synchronization time base is equal to 30ms.

The MCU manages tasks such as minimum mains voltage detection, tachogenerator voltage measurement, fuzzy logic speed control, PWM duty cycle generation, and motor speed selection.

2.1 Description of food-processor motor

The universal motor used in this application has a speed ranging up to 22000 rpm (i.e. 366 Hz). The gear ratio between motor speed and tool speed is equal to 7.5. The ratio between tachogenerator frequency and motor frequency is equal to 16.
One of the advantages of using fuzzy logic for such an application is to overcome the need for a precise mathematical model of the system. Nevertheless, the system behaviour has to be known and this knowledge can be acquired with some simple experimental graphs. Open loop motor speed versus PWM duty cycle and tachogenerator voltage versus motor speed are shown in following Figures 3 and 4.

Fig. 3. Motor Speed Versus Pwm Duty Cycle.

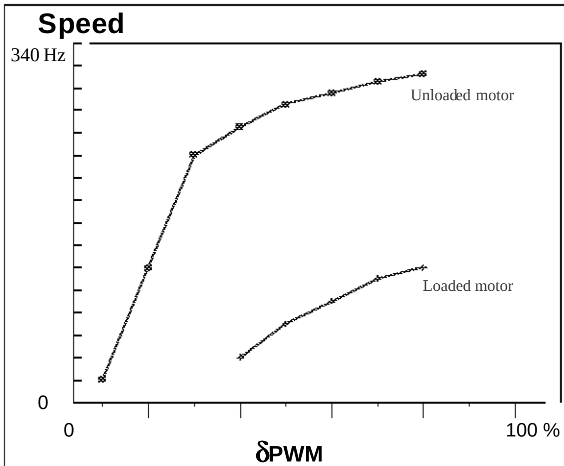
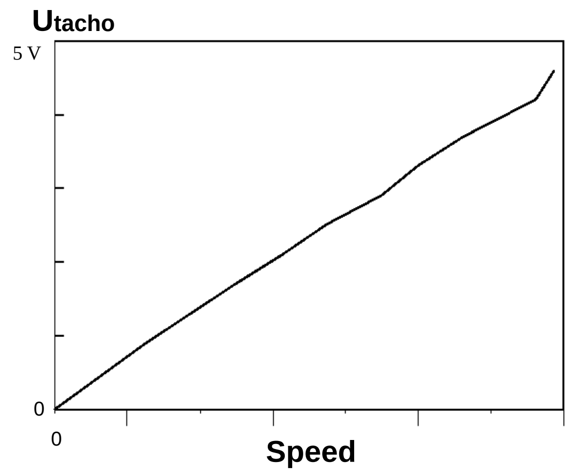


Fig. 4. Tacho Voltage Versus Motor



The slopes of these curves give the system resolution:

- 1% of PWM duty cycle variation gives a speed variation of 10 Hz for an unloaded motor at low speed, 2.5 rotations per second for a loaded motor,
- the speed measurement resolution (3 Hz) is given by the A/D converter resolution.

2.2 Input/output variables definition

The input variables of the fuzzy logic controller, estimating the motor speed variations, are the speed error ε and the speed error variations $\Delta\varepsilon$:

- The speed error ε equals the measured speed minus the targeted speed $\varepsilon = V_{tacho} - V_{tgt}$. Its range of variation is defined by the maximum speed of motor, i.e. [-300Hz, +300Hz].
- The speed error variation $\Delta\varepsilon$, estimating the motor speed evolution, equals, at sampling time (30ms), measured speed minus previously measured speed $\Delta\varepsilon = V_{tacho}(n) - V_{tgt}(n-1)$.
  The output variable is the PWM duty cycle variation, $\Delta\varepsilon$, calculated by the fuzzy inference kernel. The PWM duty cycle is then calculated by the MCU: $\delta\%(n) = \delta\%(n-1) \pm \Delta\delta$ and applied to the IGBT gate each 30ms. This PWM output signal, automatically generated by the micro-controller PWM timer, ranges from 0% to 100% with a resolution of 0.4% (full range of the PWM duty cycle is coded from 0 to 255).

3 fuzzy LOGIC APPROACH WITH DEVELOPMENT TOOL

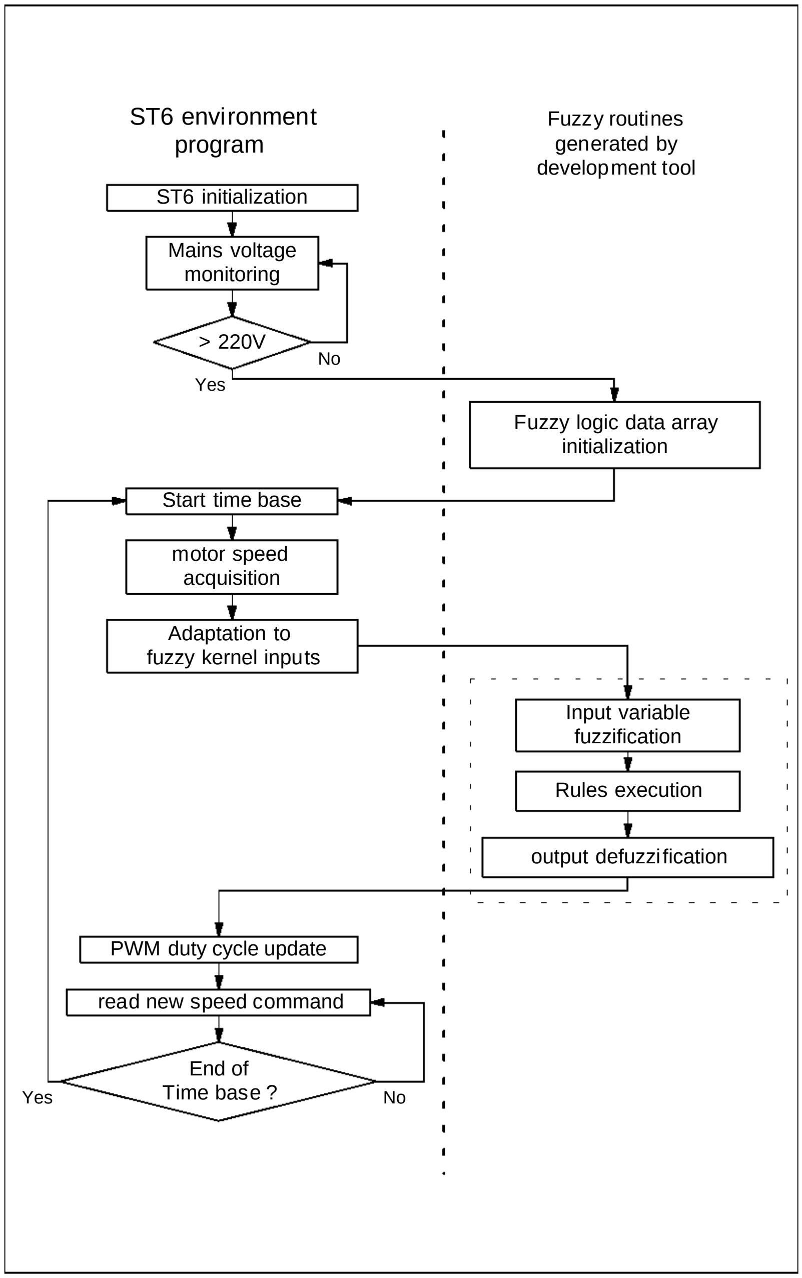
The program flow chart given in Figure 5 shows that a fuzzy logic development can be divided into two main parts: the fuzzy logic application itself and the micro-controller environment program.

- The fuzzy logic part consists of ST6 executable code generated by the development tool. This part is made of the fuzzification of the input variables, execution of the activated rules and defuzzification producing the output variable.

- The environment program consists of micro-controller initialization, motor speed acquisition, speed error and speed error variation calculation, input variable adaptation to fuzzy logic kernel code values, the calling fuzzy logic kernel, PWM duty cycle update, new motor speed command acquisition and waiting for the end of synchronization time base .

The "fuzzyTECH ST6 Explorer Edition" used to develop this application covers all the steps of a fuzzy logic design from the project, linguistic variables and rules definitions up to the ST6 executable code generation.

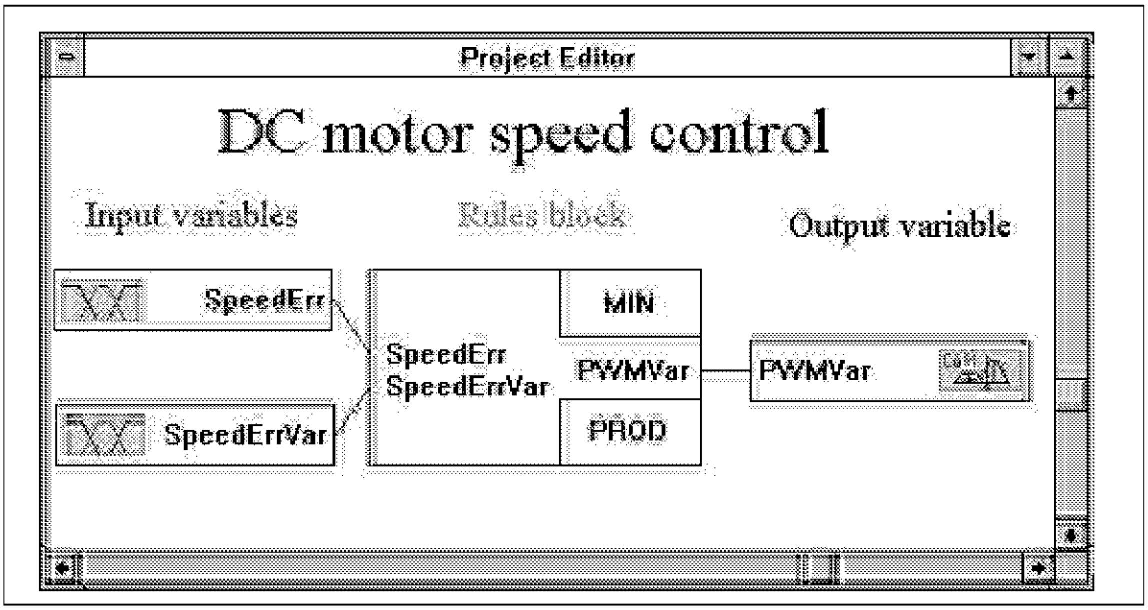
Fig. 5. fuzzy Logic Application Flow Chart.



3.1 Project definition in development tool

The first step when using the fuzzyTECH ST62 Explorer Edition is to define the project by means of the project editor window. The project editor displays the controller structure and allows the designer to directly access linguistic variables and rule definitions. Figure 6 shows the project window of the motor speed control.

Fig. 6. Motor Speed Control Project.



3.2 Linguistic variables definition

The next step of the controller design is the definition of linguistic variables. The graphic interface of the development tool allows the designer to easily create the most suitable linguistic variables and the membership functions for the application.
During the definition of linguistic variables, the development tool allows the user to define two representations for the variables: the ”shell value” and the ”code value”. The shell value is a representation of variables only used to display actual data with the tool. The code values are used for the micro-controller executable code generation and range from 0 to 255 (8-bit representation). Having the same scale for the code value and the shell value gives a better understanding of the control behaviour by directly comparing the real fuzzy inputs/output to the linguistic variables.

The input membership functions shown in Figure 7 are defined taking into account the speed and the acceleration of the motor and the system resolution. The motor speed range is well covered with five membership functions. According to the sensibility of the speed measurement line and to the speed sampling rate (3Hz each 30ms corresponds to an acceleration of 100Hz/s), the motor speed variation range is described with three membership functions. The triangular shaped (”Pi-Type”) membership functions ”Zero” and ”Nul” make the PWM duty cycle variations less sensitive to the A/D converter resolution.

Fig. 7A. Membership Functions For Speederr

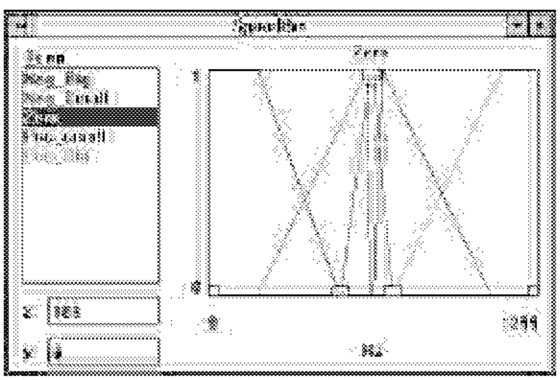
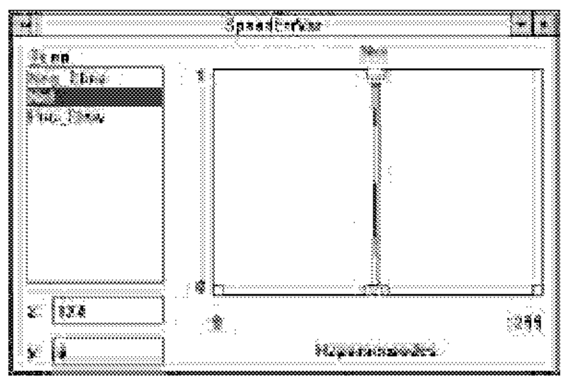


Fig. 7B. Membership Functions For Speederrvar

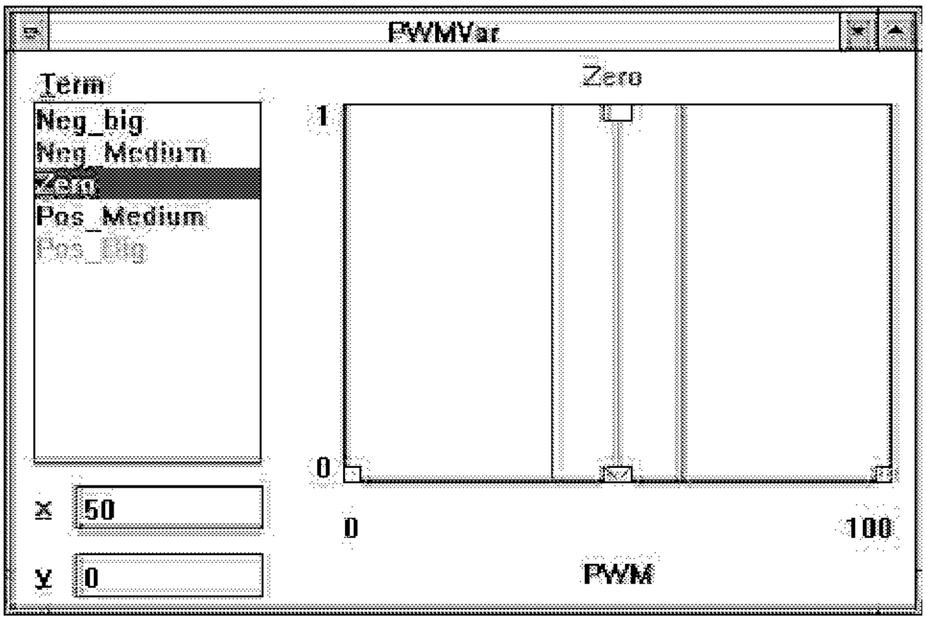


.

As the PWM duty cycle variation is computed using the Center of Maximum method, the output linguistic variable can be represented with lines instead of the triangular shape as shown in Figure 8.

The PWMVar code value has been reduced to the range [0, 24] producing each 30ms a maximum duty cycle variation of ± 5% with a resolution of 0.4% directly compatible with the PWM resolution generated by the ST6. This avoids additional calculation when adding the PWM duty cycle variation $\pm \Delta\varepsilon(\%)$ to the current PWM duty cycle $\delta(\%)$.

Fig. 8. Membership Functions For Pwm Duty Cycle



3.3 Rule definition

Fig. 9. Spreadsheet Rule Editor.

Spreadsheet Rule Editor

Matrix	IF		THEN	
#	SpeedErr	SpeedErrVar	DoS	PWMVar
1	Zero	Neg_Slow	1.00	Pos_Medium
2	Zero	Nul	1.00	Zero
3	Zero	Pos_Slow	1.00	Neg_Medium
4	Pos_small	Neg_Slow	1.00	Zero
5	Pos_small	Nul	1.00	Neg_Medium
6	Pos_small	Pos_Slow	1.00	Neg_Medium
7	Neg_Small	Neg_Slow	1.00	Pos_Medium
8	Neg_Small	Nul	1.00	Pos_Medium
9	Neg_Small	Pos_Slow	1.00	Zero
10	Neg_Big	Nul	1.00	Pos_Big
11	Pos_Big	Nul	1.00	Neg_big
12	Neg_Big	Pos_Slow	1.00	Pos_Medium
13	Pos_Big	Pos_Slow	1.00	Neg_big
14	Neg_Big	Neg_Slow	1.00	Pos_Big
15	Pos_Big	Neg_Slow	1.00	Neg_Medium
16				

Figure 9 shows the fuzzy controller rules. They have been defined by the understanding of the behaviour of the system and one can find rules maintaining speed error near zero (steady state rules), rules avoiding motor speed overshoot (small variation rules) and rules providing rapid response to large error due to command change (big variation rules).
The fuzzy development tool with its Spreadsheet rule editor (fig. 9) provides easy rule acquisition and also permits the definition of rule aggregation (MIN and MAX operators are available) and rule composition (PROD operator with degree of support of 0 or 1). Here, the rules aggregation is done using the MIN operator and the degree of support

3.4 Adaptation of real variables to fuzzy kernel input

Working with variations of input variables, which can have negative values, translation has to be done to code the whole range of the input variables within the range [0, 255]. The motor speed representation (given by the A/D converter) can vary from 0 to 255, so the speed error and the speed error variation range from -255 to +255. This variation is coded within the range [0, 255] by dividing the input variables by 2 and adding 127 (07fh) to normalize the value.

3.5 Open loop testing and optimization

The fuzzy development tool offers two ways to test and optimize the rules and membership function definition:

- the interactive debug mode provides the designer with a graphical verification of every design step even while the design is being performed.

- the batch mode associated with a pattern generator records the output variables versus each of the input variables.

3.6 Real time test

The real time test of the application can be done using an emulator or an EPROM version of ST6 device. In both cases, real time recording of the application variables measuring the motor behaviour (tachogenerator voltage, PWM variation, linguistic variables) can be helpful. This can be achieved by a 9600 Baud serial link between the ST6 and a computer recording the data. This data file can then be used in a spreadsheet editor capable of displaying curves. This method, replacing an oscilloscope, has been used to record the following results (e.g. Figure 10).

4 PRACTICAL RESULTS

Recordings of tachogenerator voltage (corresponding to the motor speed) and PWM duty cycle while the motor is running are given in the following figures.
The motor reaches the steady state speed in 1.3s corresponding to the execution of about 40 fuzzy logic loops.
No overshoot of the motor speed has been observed.
In steady state, we can notice motor speed fluctuations of ± 3Hz. These fast fluctuations (30ms) are due to the resolution limit of the speed measurement line (tachogenerator and A/D converter) and remain inaudible.
Figure 10 shows the speed step response when the motor is not loaded.

Fig. 10. Speed Step Response Versus Time For Unloaded Motor.

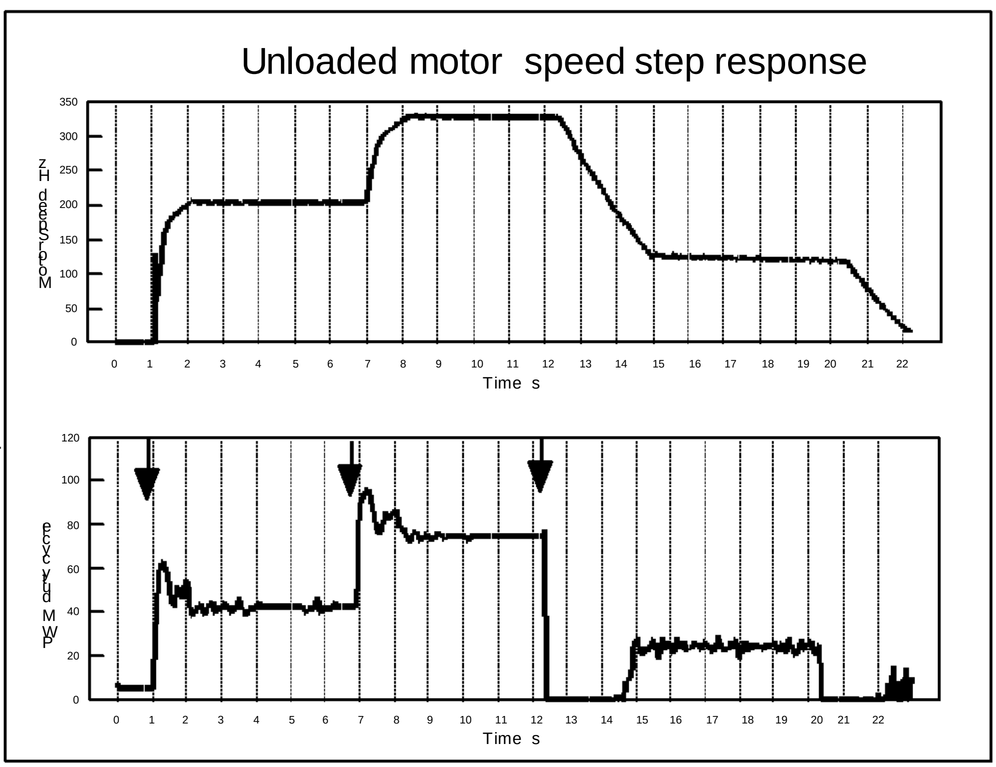
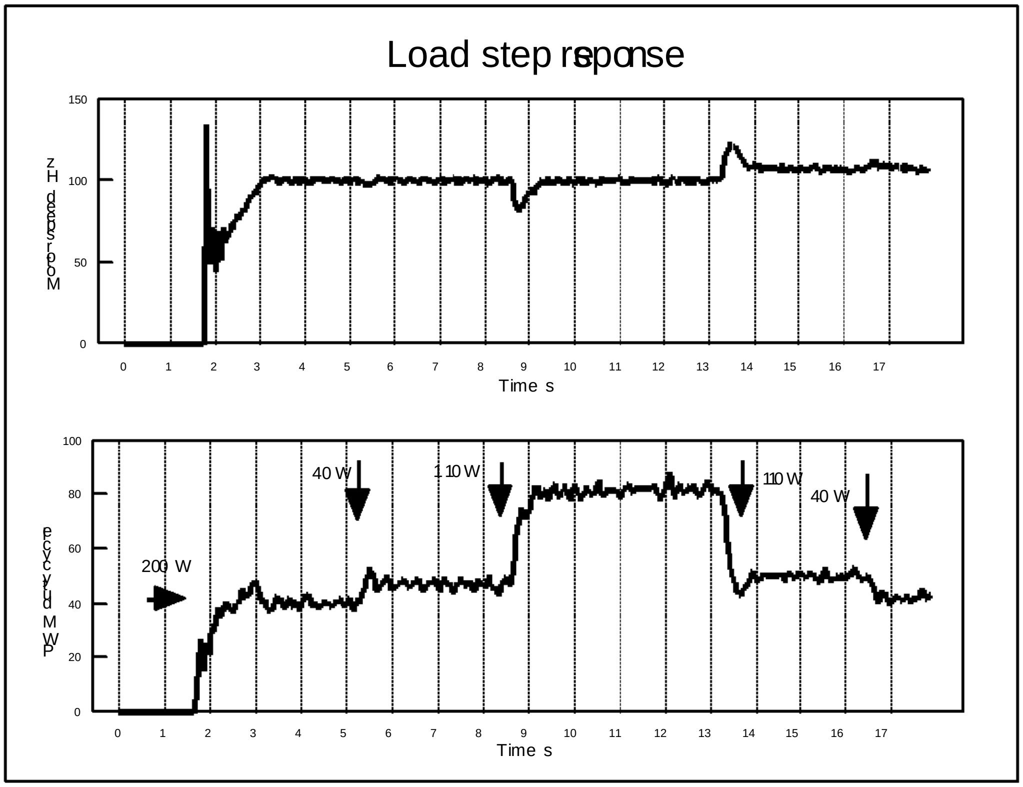


Figure 11 shows the load step response of the motor.

The motor is permanently loaded with 200W; 40W and 110W load variations are applied. Upon these load variations, the motor recovered the targeted speed within less than 1s without any speed overshoot.

Fig. 11. Load Step Response Versus Time 200W Loaded Motor.



5 CONCLUSION

This note presents a speed control with fuzzy logic for a DC universal motor. This motor control, implemented using a standard ST6265 micro-controller running a fuzzy logic kernel, gives acceptable results for home appliances, such as drills, washing machines, and food processors, which do not require high precision level control.

Improvements can be achieved by increasing the sensitivity of the speed error variation and by adding a 3rd parameter giving an image of the motor load in order to adapt the PWM variation to the motor load.

Using fuzzy logic for this application has avoided needing the knowledge of a precise mathematical model of the system regulation loop, and permits first results within some days. Furthermore, the fuzzy logic program is flexible and can be easily adapted to other motor characteristics.

Optimizing the results (response time and overoscillations) can be achieved by making several trials; no precise method for fuzzy logic variables and rule definitions is available. These optimization trials may be reduced in number if the effect of the Fuzzy logic variable modifications on the system behaviour is well understood.

6 REFERENCES

[1] - Adaptative fuzzy logic control for DC motor speed loop
P. Kosc, F. Profumo - Electrical Drives and Power Electronics - September 1992.

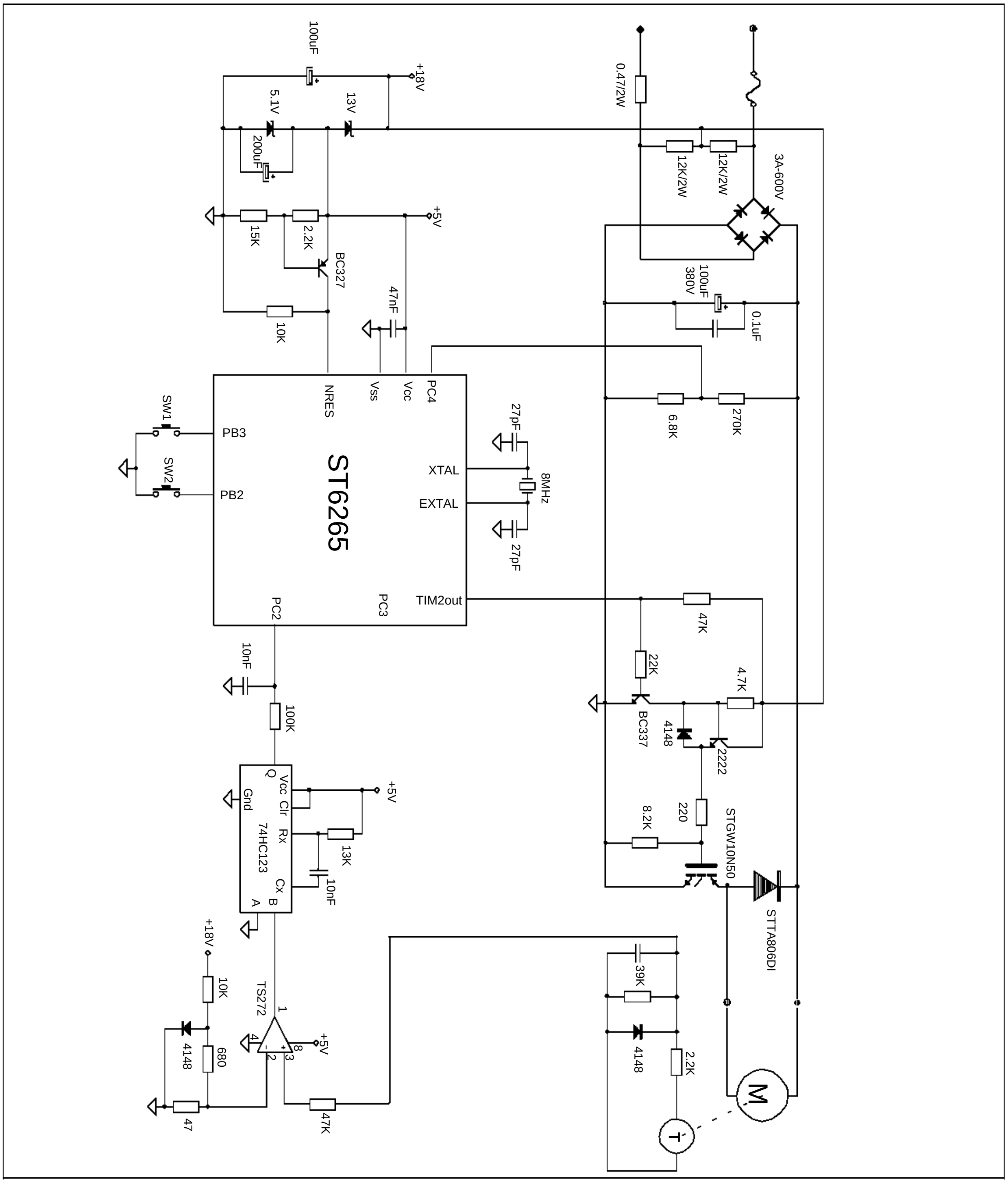
[2] - Twenty years of fuzzy control: Experiences gained and lessons learnt
E.H. Mamdani - IEEE 1993.

[3] - fuzzyTech ST6 Explorer Edition User Manual
SGS-THOMSON Microelectronics

[4] - ST6260-6265 data sheet SGS-THOMSON Microelectronics

[5] - Sensorless speed control for universal motor
Thierry Castagnet, Corporate Application Laboratory, SGS-THOMSON Microelectronics.

Appendix 1. Practical schematic of the universal motor speed control with fuzzy logic.
The fuzzy logic regulation loop is achieved by a standard ST6 MCU. The motor is supplied in DC mode through a chopper stage composed of an IGBT STGP10N50 and a freew heeling diode STTA806DI. The board is directly connected on the mains.



NOTES :