DESIGN AND FUZZY CONTROL OF A FOREFINGER

Catrina Chivu
"Transilvania" University of Braşov, catrina.chivu@gmail.com

Keywords: hand prosthesis, anti-quadrilateral mechanism, fuzzy controller, simulation

Abstract: The present paper presents all the steps that have to be followed to design a high performance hand prosthesis. Thus, there is synthesized the mechanical design. There are also presented the PWM command of the DC motor that actuates the forefinger and the fuzzy controller designed for this application.

1. INTRODUCTION

Hand prostheses are very complex mechanical devices and their design has to satisfy a large number of constrains. Among these constrains stand the gripping force at the tip of the fingers, the movement of the fingers, the shape of the gripped objects and so on [2]. The fingers design begins considering the numbers of joints desired for each finger, the mechanisms chosen to model the finger and the lengths of each element. The control, in the case when there is no prototype, implies either to conceive a program similarly with an interface for electronic part of the system or all the control as an abstract thing, starting from a data base there is obtained the output and this is compared with the desired one.

Fuzzy Logic provides a completely different, unorthodox way to approach a control problem. This method focuses on what the system should do rather than trying to understand how it works. One can concentrate on solving the problem rather than trying to model the system mathematically, if that is even possible. This almost invariably leads to quicker, cheaper solutions. Once understood, this technology is not difficult to apply and the results are usually quite surprising and pleasing [3, 7].

The present paper presents the mechanical design, the command and control of forefinger of hand prosthesis.

1. MECHANICAL DESIGN

1.1. MATHEMATICAL LAWS OF MOVEMENT

The forefinger was modeled using two anti-quadrilateral mechanisms. The figure 1 presents the two mechanisms, being set off (by heavy strike) the second one [2].

To mechanically design a hand prosthesis there are some constrains to be satisfied, starting from the real, desired dimensions of the prosthesis and movement of the fingertip. To be able to determine the movement of the tip of the finger there were adopted some input values for the geometrical part of the finger. Thus [1],

$$\begin{array}{lll} OA=\ell_1=55{,}2\ [mm]; & AB=\ell_2=10{,}8\ [mm]; & BC=\ell_3=50{,}9\ [mm]; \\ CO=\ell_4=9{,}9\ [mm]; & AE=\ell_5=31{,}3\ [mm]; & EF=\ell_6=10\ [mm]; \\ FD=\ell_7=34{,}8\ [mm]; & DA=\ell_8=9{,}8\ [mm]; & EG=\ell_9=22{,}5\ [mm]; \\ \angle IOA=\alpha=125^\circ; & \angle COx=\beta=15^\circ; & \angle OAD=\gamma=50^\circ; \\ \angle BAE=\delta=120^\circ; & \angle FEG=\theta=130^\circ & \end{array} \tag{1}$$

To determine the mathematical equations for the movement of the forefinger it has to be determined the outputs of each anti-quadrilateral mechanisms. These outputs are obtained by writing the contour equations for each mechanism. The equations (2) ÷ (5) describe the output of the first and, respectively, the second mechanism.

$$\overline{OA}+\overline{AB}+\overline{BC}+\overline{CO}=0 \tag{2}$$

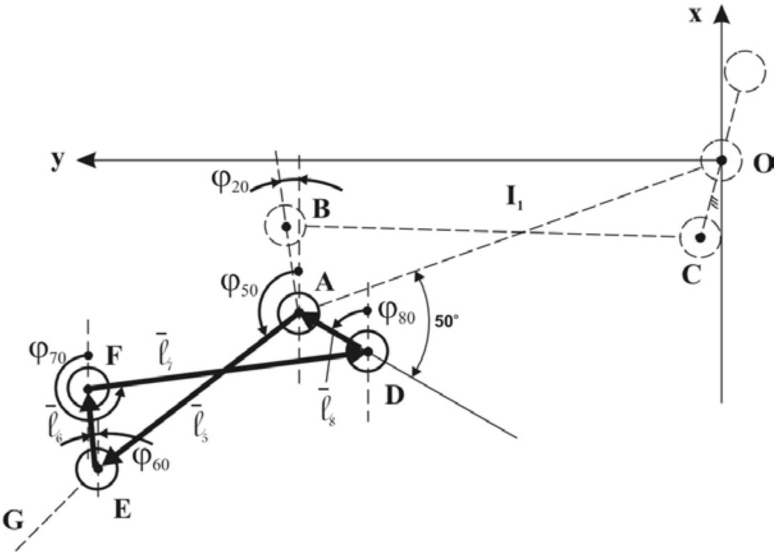


Figure 1. The diagram of the forefinger

$$(2)\xrightarrow{\text{proiectie pe axe}}\begin{cases} l_1\cos\varphi_{10}+l_2\cos\varphi_{20}+l_3\cos\varphi_{30}+l_4\cos\varphi_{40}=0\\ l_1\sin\varphi_{10}+l_2\sin\varphi_{20}+l_3\sin\varphi_{30}+l_4\sin\varphi_{40}=0\end{cases}\quad(3)$$

$$\overline{AE}+\overline{EF}+\overline{FD}+\overline{DA}=0\quad(4)$$

$$(4)\xrightarrow{\text{proiectie pe axe}}\begin{cases} l_5\cos\varphi_{50}+l_6\cos\varphi_{60}+l_7\cos\varphi_{70}+l_8\cos\varphi_{80}=0\\ l_5\sin\varphi_{50}+l_6\sin\varphi_{60}+l_7\sin\varphi_{70}+l_8\sin\varphi_{80}=0\end{cases}\quad(5)$$

The movement of each joint of the forefinger mechanisms is described by the equations (6) and (7):

$$\begin{cases} x_A=l_1\cos(180-\varphi_{10})\\ y_A=l_1\sin(180-\varphi_{10})\end{cases}\quad(6)$$

$$\begin{cases} x_E=l_5\cos(180-(\varphi_{20}+\delta))\\ y_E=l_5\sin(180-(\varphi_{20}+\delta))\end{cases}\quad(7)$$

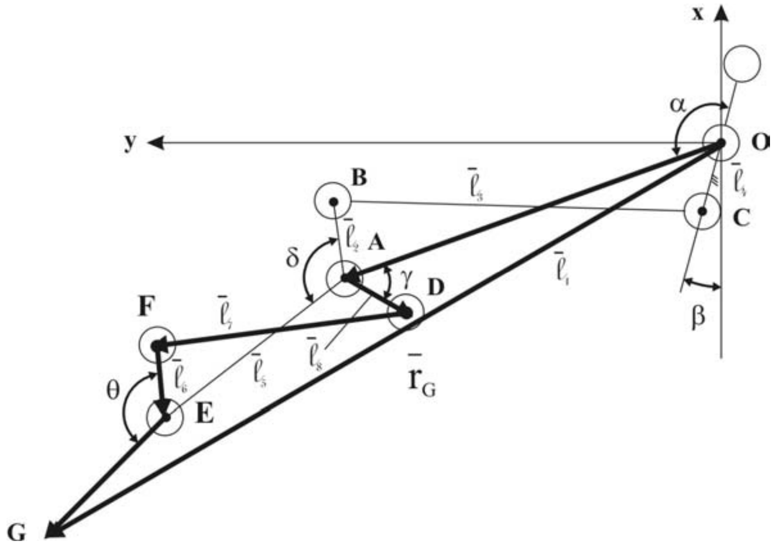


Figure 2. The diagram of $\bar{r}_G$ vector

The coordinates of the forefinger tip are given by the vector $\bar{r}_G$ (figure 2). The equations are:

$$\bar{r}_G=\overline{OA}+\overline{AD}+\overline{DF}+\overline{FE}+\overline{EG}\quad(8)$$

$$\begin{cases} \overline{OA}\left[l_1\cos\varphi_{10},\ l_1\sin\varphi_{10}\right] \\ \overline{AD}\left[l_8\cos(\varphi_{10}-\gamma+180),\ l_8\sin(\varphi_{10}-\gamma+180)\right] \\ \overline{DF}\left[l_7\cos(\varphi_{70}-180),\ l_7\sin(\varphi_{70}-180)\right] \\ \overline{FE}\left[l_6\cos(\varphi_{60}+180),\ l_6\sin(\varphi_{60}+180)\right] \\ \overline{EG}\left[l_9\cos\varphi_{90},\ l_9\sin\varphi_{90}\right] \end{cases} \tag{9}$$

In figure 3 is presented the forefinger as it resulted after the mechanical design. There is also shown the actuation (gears and dc motor).

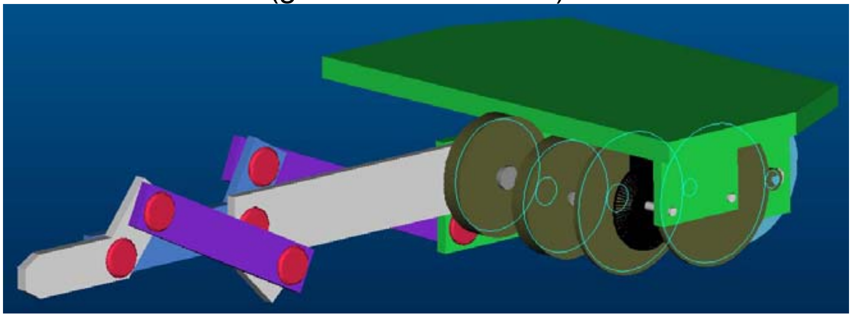


Figure 3. The mechanical and actuating structure of the forefinger [1]

1.2. SIMULATION OF THE MOVEMENT OF EACH JOINT OF FOREFINGER

Simulation was done by considering as input the geometrical values, given at the beginning of the previous subchapter, and the mathematical relations given by equations (6), (7) and (8). The figure 4 presents the three movement of the forefinger: the two joints and the tip of the finger.

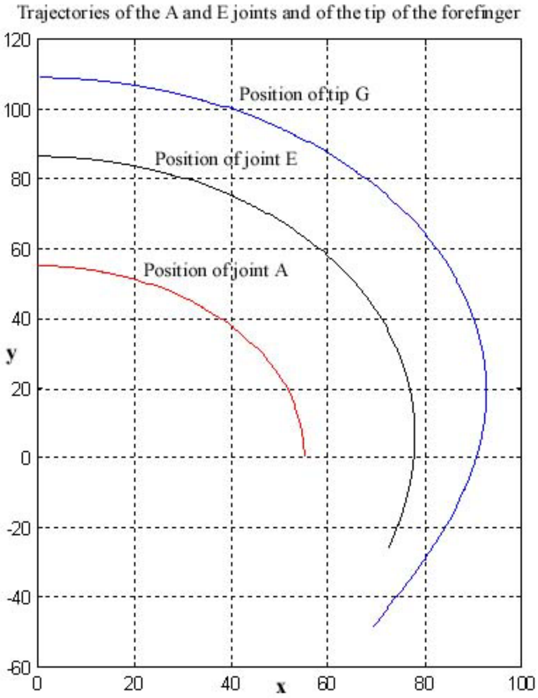


Figure 4. Diagram of movement of: tip of the finger and the joints A and E [1]

2 CONTROL MODULE

2.1. CONTROL CIRCUIT

Block diagram of control circuit is presented in figure 5.

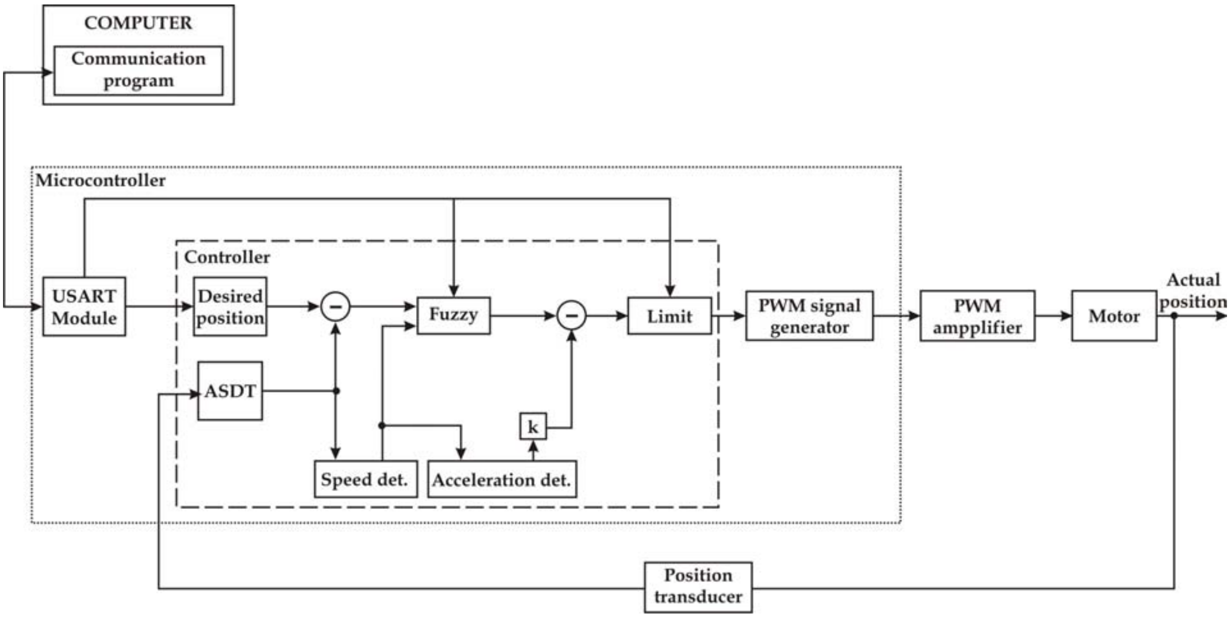


Figure 5. Bloc diagram of the control circuit [1]

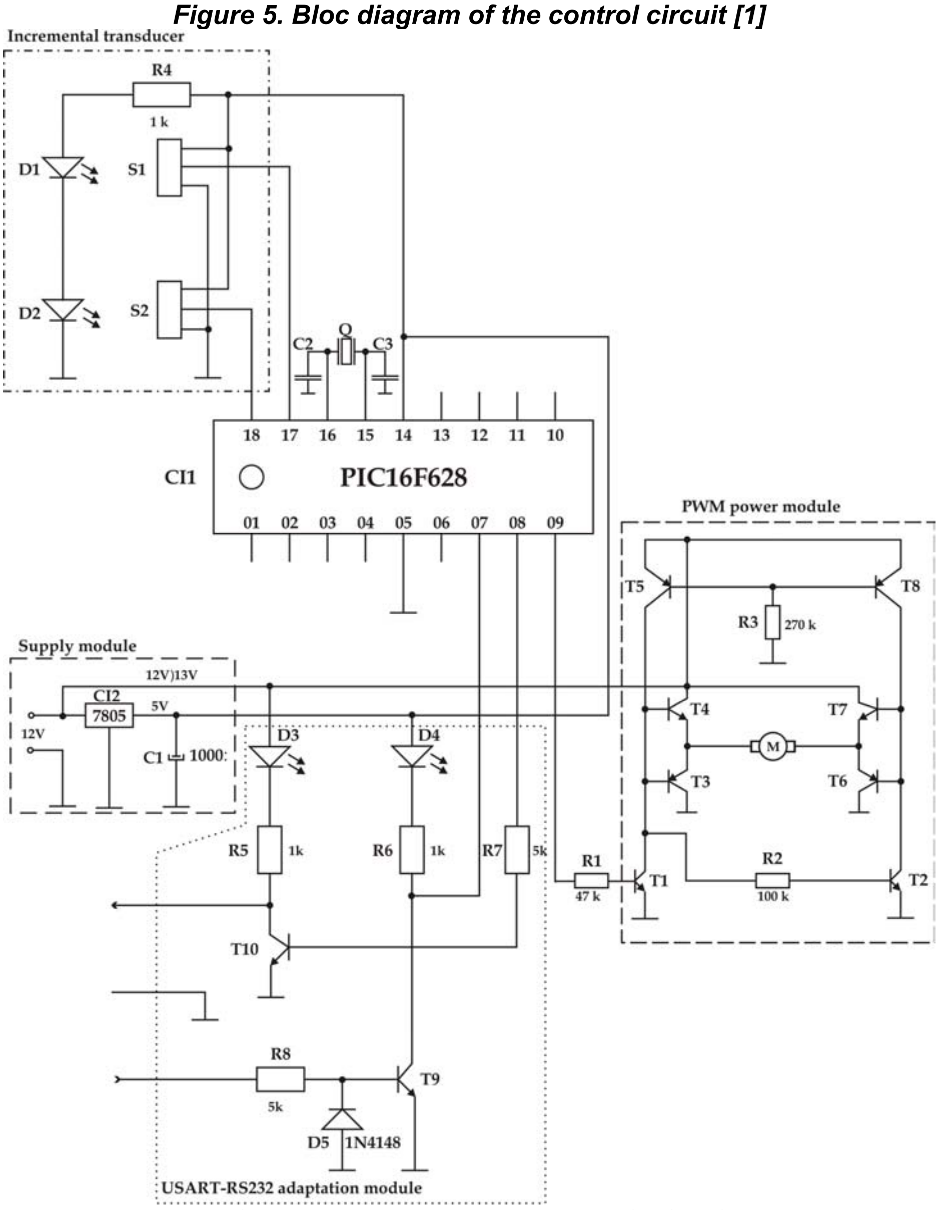


Figure 6. Command and control circuit of a forefinger [1]

The component blocks of the diagram presented in figure 5 are:
Motor – DC motor with permanent magnet excitation.
Power circuit – power circuit presented in figure 6.
Position transducer – is an incremental transducer with two channels directly mounted to the motor axe.
Microcontroller – PIC16F628, implements, by its program, many of the functional blocks of the diagram.
SAISD – Sequential Automat for Incremental Sensor Decoding.
Speed det. – computes the speed, by numerical derivation of the position.
Acceleration det. – computes the acceleration, by numerical derivation of the speed.
Fuzzy – the fuzzy controller that has as inputs: position error and motor speed.
k – constant amplification.
Limit – block that limits the PWM output value on a specific interval.
PWM –hardware module integrated on the microcontroller circuit.

The electronic circuit of the controller is shown in figure 6. The diagram contains the PWM power module of the motor, PIC16F628 microcontroller and the incremental transducer that has two IR LEDs and two phototransistors.

2.2. FUZZY CONTROLLER RULES

The designed fuzzy controller is a Takagi-Sugeno one and has as inputs the *angular position error* and *differential of angular position error* and as output the *PWM duty factor*. The two inputs “e” and “$\dot{e}$” are determined, at any moment, using the equations:

$$\begin{aligned} e(k) &= n^*(k) - n(k) \\ \dot{e}(k) &= e(k) - e(k-1) \end{aligned} \qquad (10)$$

where $n^*(k)$ is the reference angular position at that moment and $n(k)$ is the determined actual position.

The two linguistic variables, corresponding to the inputs, have five linguistic terms: NM – negative medium, NS – negative small, ZE – zero, PS – positive small, PM – positive medium.

The membership functions are triangular one and their definition is shown in figure 7. The *PR1X, PR2X, PR3X, CX* and *PR1V, PR2V, PR3V, CV* parameters define the definition interval of the two membership functions (*PR1X* – a1*, PR2X* – a2*, PR3X* – a3*, PR1V* – a1*, PR2V* – a2*, PR3V* – a3) and, respectively, the maximums of *PM* and *NM* intervals (*CX, CV*). To these parameters where attributed an estimative set of values and, after controller implementation those values where modified in the simulation process.

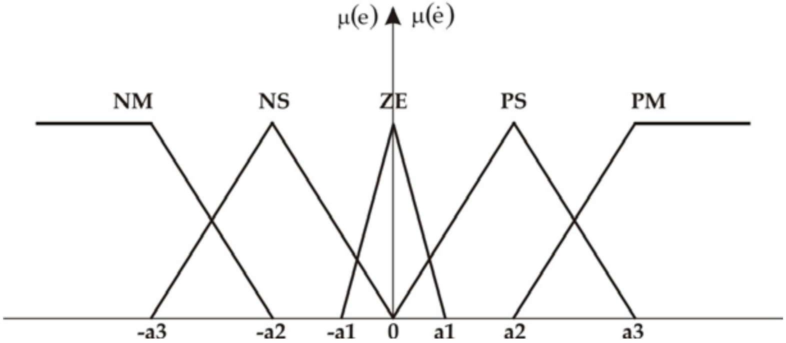


Figure 7. Membership function of the inputs

The constants of the fuzzy controller are symbolized by S_{x1}, S_{x2}, S_{x3} and S_{v1}, S_{v2}, S_{v3}.

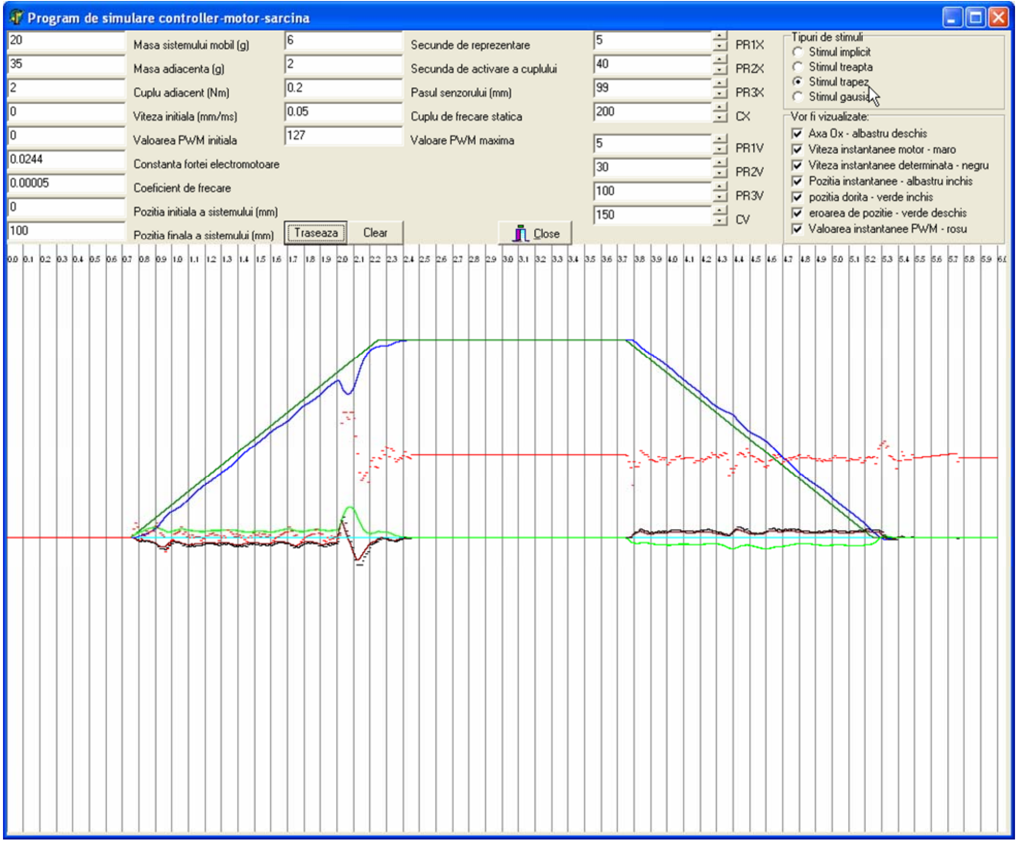
Using these notations where defined the rules *R1,…, R25* that where used to determine the numeric values of the fuzzy controller outputs (Table 1).

Table 1. The outputs of the fuzzy controller

		$\dot{e}$				
		NM	*NS*	*ZE*	*PS*	*PM*
e	*NM*	$-S_{x1}x+S_{v1}v$	$-S_{x1}x+S_{v2}v$	$-S_{x1}x+S_{v3}v$	$-S_{x1}x+S_{v2}v$	$-S_{x1}x+S_{v1}v$
	NS	$-S_{x2}x+S_{v1}v$	$-S_{x2}x+S_{v2}v$	$-S_{x2}x+S_{v3}v$	$-S_{x2}x+S_{v2}v$	$-S_{x2}x+S_{v1}v$
	ZE	$-S_{x3}x+S_{v1}v$	$-S_{x3}x+S_{v2}v$	$-S_{x3}x+S_{v3}v$	$-S_{x3}x+S_{v2}v$	$-S_{x3}x+S_{v1}v$
	PS	$-S_{x2}x+S_{v1}v$	$-S_{x2}x+S_{v2}v$	$-S_{x2}x+S_{v3}v$	$-S_{x2}x+S_{v2}v$	$-S_{x2}x+S_{v1}v$
	PM	$-S_{x1}x+S_{v1}v$	$-S_{x1}x+S_{v2}v$	$-S_{x1}x+S_{v3}v$	$-S_{x1}x+S_{v2}v$	$-S_{x1}x+S_{v1}v$

2.3. SIMULATION PROGRAM FOR CONTROLLER-MOTOR-LOAD SYSTEM

The simulation program is created in Delphi [5]. The interface of the program (figure 8) allows the user to visualize the simulation characteristics of the evolution of controller-motor-load system and also of some important internal parameters of the controller. It was created a physical electronic circuit that uses a microcontroller (PIC16F628) and the fuzzy controller implemented on that circuit is an identical one with the controller designed in the simulation program presented in this paper. This is the reason why it can be said that the simulated behavior of the system and the real one are identically. The results of the simulations are presented in [1].



Program de simulare controller-motor-sarcina

20 Masa sistemului mobil (g)
35 Masa adiacenta (g)
2 Cuplu adiacent (Nm)
0 Viteza initiala (mm/ms)
0 Valoarea PWM initiala
0.0244 Constanta fortei electromotoare
0.00005 Coeficient de frecare
0 Pozitia initiala a sistemului (mm)
100 Pozitia finala a sistemului (mm)

6 Secunde de reprezentare
2 Secunda de activare a cuplului
0.2 Pasul senzorului (mm)
0.05 Cuplu de frecare statica
127 Valoare PWM maxima

5 PR1X
40 PR2X
99 PR3X
200 CX
5 PR1V
30 PR2V
100 PR3V
150 CV

Tipuri de stimuli
○ Stimul implicit
○ Stimul treapta
● Stimul trapez
○ Stimul gausian

Vor fi vizualizate:
☑ Axa Ox - albastru deschis
☑ Viteza instantanee motor - maro
☑ Viteza instantanee determinata - negru
☑ Pozitia instantanee - albastru inchis
☑ pozitia dorita - verde inchis
☑ eroarea de pozitie - verde deschis
☑ Valoarea instantanee PWM - rosu

Figure 8. Interface of the simulation program [1]

The functional parameters of the program (that can be seen also on the interface) can be classified depending on the physical parameters that they characterized. Thus:
- mechanical parameters: *mobile system mass (masa sistemului mobil)* – represents the equivalent mass of the motor rotor and of all mechanical elements connected with it (including the actuated finger), excepting the manipulated object; *adjacent mass (masa adiacenta)* – represents the equivalent mass of the manipulated object, reduced to the motor axle; *adjacent torque (cuplul adiacent)* – equivalent torque of the force that actuates on the finger, at one moment, reduced to motor axle; *torque activation second (secunda de activare a cuplului)* – the moment when appears the force that actuates on the finger, relative with the simulation start moment; *static friction torque (cuplul de frecare statica)* – the static friction torque at zero speed of the system; *friction coefficient (coefficient de frecare); initial speed (viteza initiala); system initial position (pozitia initiala a sistemului); system final position (pozitia finala a sistemului); step of the sensor (pasul senzorului); simulation time (secunde de reprezentare)*.
- electrical parameters: *constant of electromotive force; initial PWM value; maximum PWM value*.
- **fuzzy parameters**: *PR1X, PR2X, PR3X, CX, PR1V, PR2V, PR3V, CV* – the domain of the membership functions that define the two input variables: position error and speed error [1].

The program allows, by using the *Stimuli type (Tipuri de stimuli)* buttons group, to select the type of the input. Thus, it can be chosen among: *implicit stimulus (stimul implicit)* – that connects the initial value and final value without an enforced form of the signal; *step input (stimul treapta); trapeze input (stimul trapez)* – a trapeze input that has its definition elements given as percent of simulation time; *Gaussian input* – the input is so called "Gauss bell", given by equation $m(x) = A \cdot e^{-\frac{(x-x_0)^2}{2\cdot\sigma^2}}$).

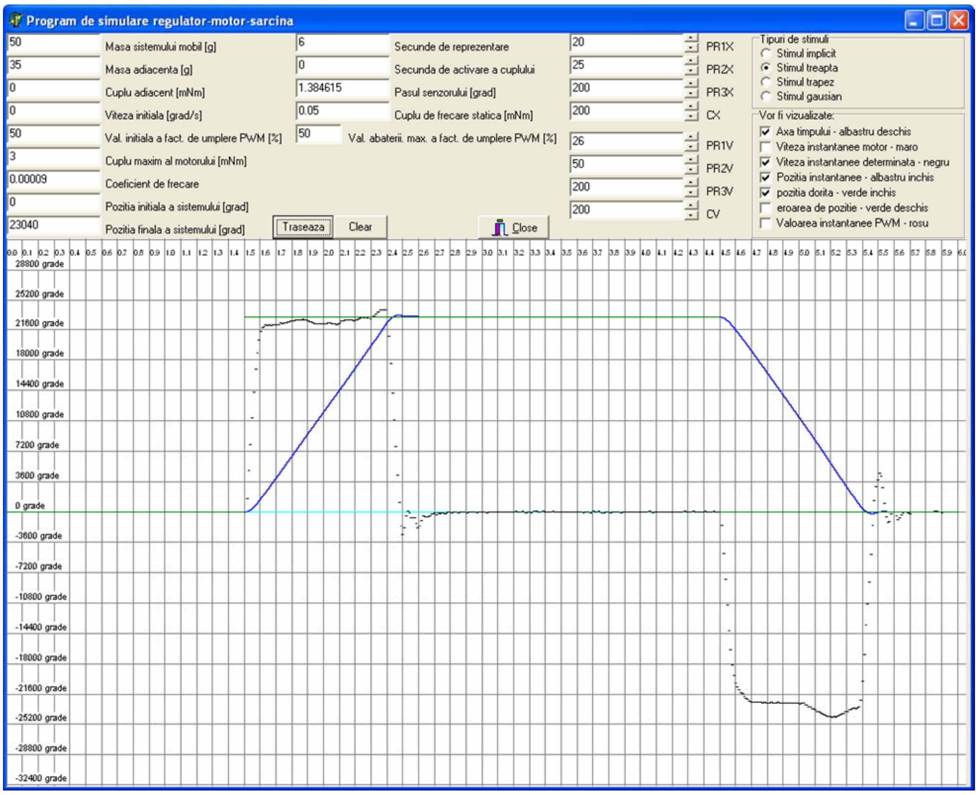
After is chosen the input form of the signal, there can be selected, using the group of button called *Will be visualized (Vor fi vizualizate)*, the diagrams that one wants to be visualized.

After the inputs were chosen, type of diagrams that it will be represented, it could be pushed the button *Draw (Traseaza)*. This button starts the simulation process. After this process ends on the interface window appears the selected diagrams (figure 8). Thus, the diagram drawn with: *light blue* represents *Ox* axis (time axis); *maroon* – motor instantaneous speed; *black* – determined instantaneous speed; *bluemarin* – instantaneous position; *dark green* – desired position; *light green* – system relative position (positioning error) and *red* – instantaneous PWM value at controller output.

If the current simulation is done and it is wanted to do another one, there should be selected the new values for the desired parameters and, after that, there are two possibilities: to keep the last diagrams on the window (in this case it just should be pushed the button *Draw* - *Traseaza*) or to clean the last diagrams by pushing the button *Clear* and after that go to another simulation by pushing the button *Draw (Traseaza)*. The clear procedure is done manually in order to give the possibility to compare two or more simulation on the same diagram, by keeping the desired simulation diagrams, in the same time, on windows.

To obtain real simulation results there should be introduced, as inputs of the controller-motor-load simulation program, real values, computed or chosen from the motor catalogue.

The following simulation (figure 9) where done considering a motor produced by Faulhaber with a torque of 3 [mNm] [FAU05].



Program de simulare regulator-motor-sarcina

50	Masa sistemului mobil [g]	6	Secunde de reprezentare	20	PR1X
35	Masa adiacenta [g]	0	Secunda de activare a cuplului	25	PR2X
0	Cuplu adiacent [mNm]	1.384615	Pasul senzorului [grad]	200	PR3X
0	Viteza initiala [grad/s]	0.05	Cuplu de frecare statica [mNm]	200	CX
50	Val. initiala a fact. de umplere PWM [%]	50	Val. abaterii. max. a fact. de umplere PWM [%]	26	PR1V
3	Cuplu maxim al motorului [mNm]			50	PR2V
0.00009	Coeficient de frecare			200	PR3V
0	Pozitia initiala a sistemului [grad]			200	CV
23040	Pozitia finala a sistemului [grad]				

Tipuri de stimuli
○ Stimul implicit
● Stimul treapta
○ Stimul trapez
○ Stimul gausian

Vor fi vizualizate:
☑ Axa timpului - albastru deschis
☐ Viteza instantanee motor - maro
☑ Viteza instantanee determinata - negru
☑ Pozitia instantanee - albastru inchis
☑ pozitia dorita - verde inchis
☐ eroarea de pozitie - verde deschis
☐ Valoarea instantanee PWM - rosu

Figure 9. System output when the input was positive and negative step functions [1]

There were represented the *determined instantaneous speed* (*viteza instantanee determinata*) – black line – determined by a numeric derivation of angular position, it is an internal parameter; *instantaneous position* (*pozitia instantanee*) – blue line – and *desired position* (*pozitia dorita*) – green line.

As value for *motor maximum torque* (*Cuplu maxim al motorului*) – from interface – was introduced the value from the motor catalogue: 3 [mNm]. The mechanical subsystem of the prosthesis includes the four spur gearings that allow to reduce the motor speed at the entry of the forefinger (the first two gearings – from motor to the forefinger entry – have the reduction ration of 8 and the other two gearings have the reduction ration 4). Thus, the total reduction ration of the first two gearings is 64 and was included in the program as constant input. The reduction ratio of the last two gearings it is not included in the simulation program because it was considered that the incremental transducer is mounted after the two gearings. This position of the incremental transducer was chosen because otherwise there is a limitation of the motor speed (because of the controller) and, implicit, appears a decrease of the prosthesis speed.

The value introduced in the *System final position* (*pozitia finala a sistemului*) was obtained by multiplying 360^o (at the motor ax) with 64 and thus, obtaining a full rotation after the second gearing.

The step time of the positive step input is 1.5s and the step time of the negative step input is 4.5s. Thus, the stimulus interval is 3s, with amplitude of 23040 [degree].

If there is analyzed the response of the system it results: the dead time is very small (under 10ms); rise time (between 10% and 90% from step value) is almost 0.7s. Using a stabilization tolerance of ± 5% the system is damped after 0.7s, the complete damp is obtained at 0.9s. The overshoot is approximately 270 [degree] that is almost 1.17%.

3. CONCLUSION

Designing human hand prosthesis or robotic hand implies to do an initial simulation of both mechanical system and command and control system. Present paper presents the main steps followed: design of the mechanical part, of command system (PWM command of a DC motor) and of control system (fuzzy controller).

The simulation results (in the present paper was presented only the response to positive and negative step input, but there were done the simulation for trapeze and Gaussian inputs) help us to conclude that the designed fuzzy controller satisfies the constrains of the designer.

The simulation program allows implementing other type of controllers, other than fuzzy, thus the results can be compared to those obtained with fuzzy controller. However, because the fuzzy controller is easier to be implemented and more friendly with the user will be preferred if the simulation results will be close enough.

REFERENCES

[1] Chivu, C., Tehnici inteligente de conducere a unei proteze pentru membrul superior (Intelligent control method for control of upper limb prosthesis). PhD Thesis. Politehnica University of Bucharest.
[2] Cutkosky, M., Howe, R. - Human grasp choice and robotic grasp analysis. Dextrous Robot Hands, pp.5-31, Springer Verlag, 1990.
[3] Dario, P., Carozza, M.C., Micera, S. Design and experiments on a novel biomechatronic robotics. IROS, 2000.
[4] Design manufacturing. Linkage – MIT Course, 2002
[5] Kovacs, S. Delphi 3.0. Ghid de utilizare. Editura Albastră, Cluj-Napoca, 2001.
[6] OpenGL, GLScene for Delphi 5.0, 2000.
[7] Zecca, M., On the development of a cybernetic prosthetic hand. Scuola superiore Sant' Anna. PhD Thesis, 2003.