

Design and fuzzy control of hand prosthesis or anthropomorphic robotic hand

CATRINA CHIVU¹, CATALIN CHIVU², CATALIN GEORGESCU³

^{1,2}Department of Economic Engineering and Manufacturing Systems
“Transilvania” University of Brasov
ROMANIA

³Department of Mathematical Sciences
The University of South Dakota
UNITED STATES

¹catrina.chivu@gmail.com, ²catalin.chivu@gmail.com, ³Catalin.Georgescu@usd.edu

Abstract: - Hand prosthesis and anthropomorphic robotic hand can be designed starting from the same mechanical structure, actuating system and control system. Thus, it is useful to have software that helps the engineers to design the optimum system. The present paper presents such software that includes one of the possible mechanical structures used both in hand prosthesis and anthropomorphic robotic hands, the actuating system and the control system. For control system was chosen the fuzzy control.

Key-Words: - hand prosthesis, anthropomorphic robotic hand, fuzzy control, design and simulation.

1 Introduction

Hand prosthesis and anthropomorphic robotic hands are designed using the same principles and input parameters. The differences between two systems are the values of the input data (dimensions, gripping force, precision, etc.).

There are a lot of mechanical structures used to design both systems: linkages (as in Southampton hand – figure 1a), wires (prosthesis designed by Hokkaido University – figure 1b) or artificial muscles (Karlsruhe prosthesis – figure 1c).



Fig. 1a. Southampton hand [9]



Fig. 1b. Hokkaido University prosthesis [4]



Fig. 1c. Karlsruhe prosthesis [10]

All these prosthesis were designed individually. The authors of the present papers try to create general software that can be used to design the optimum hand prosthesis or anthropomorphic robotic hand. Optimum from both mechanical and control point of view.

The present paper is focused on the design of the hand using anti-quadrilateral mechanism and fuzzy control.

2 Mechanical system

As it was said there are many mechanical structures that can be used in both devices' design. The main purpose of the software is to include, in the future, as many mechanisms as possible.

The hand was designed as having 4 identical fingers and the thumb. Thus, the studies were done

by analyzing the forefinger and the thumb. All the fingers are modelled using anti-quadrilateral mechanisms: two for the thumb and three for the other fingers all made from TiAl6V4 [5].

2.1. Forefinger mechanical structure

The input data for the command and control system were the output data, or the needed data, from the mechanical structure. The whole mechanical structure was designed as it will be as close as possible of human hand. Thus, the angle in the active joint, φ_{10} (figure 2) varies between 0° and 90° relative to Oy axis, and between 90° and 180° relative to Ox axis. The dimensions for the four fingers were closed to human hand (1):

$$\begin{aligned} OA = \ell_1 = 55 \text{ mm}; AB = \ell_2 = 11 \text{ mm}; \\ BC = \ell_3 = 51 \text{ mm}; CO = \ell_4 = 10 \text{ mm}; \\ AE = \ell_5 = 31 \text{ mm}; EF = \ell_6 = 10 \text{ mm}; \\ FD = \ell_7 = 35 \text{ mm}; DA = \ell_8 = 10 \text{ mm}; \\ EG = \ell_9 = 23 \text{ mm}; \angle IOA = 125^\circ; \\ \angle COx = \beta = 15^\circ; \angle OAD = \gamma = 50^\circ; \\ \angle BAE = \delta = 120^\circ; \angle FEG = \theta = 130^\circ. \end{aligned} \quad (1)$$

Writing the movement equations (2) and considering the dimensions assessed, results a movement diagram of the forefinger tip as shown in figure 3. This diagram is a theoretical one and it was one of the desired outputs of the controlled system.

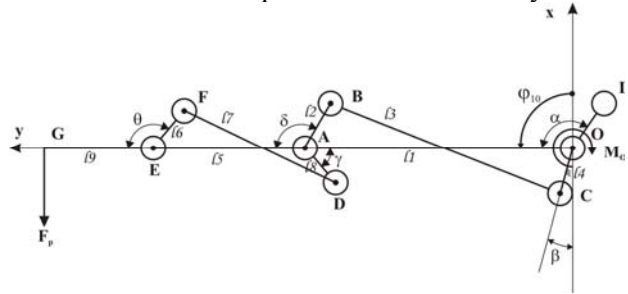


Fig. 2. Mechanism of the forefinger [2]

$$\begin{cases} \overline{OA} + \overline{AB} + \overline{BC} + \overline{CO} = 0 \\ \overline{AE} + \overline{EF} + \overline{FD} + \overline{DA} = 0 \\ \overline{r}_G = \overline{OA} + \overline{AD} + \overline{DF} + \overline{FE} + \overline{EG} \end{cases} \quad (2)$$

where

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

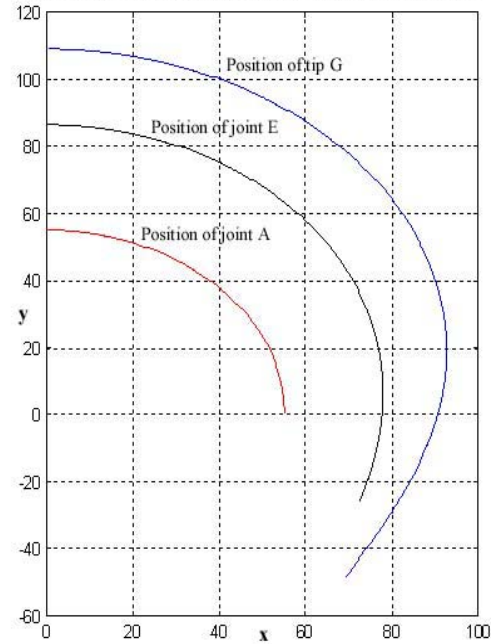


Fig.3. Trajectories of the A and E joints and of the tip of the forefinger [2]

2.2. Thumb mechanical structure

The thumb uses also anti-quadrilateral mechanisms but only two of them. The dimensions of the thumb were chosen also as close as possible of human hand, or, to be more accurate of human hand proportions. Thus, in figure 4 is presented the structure of the thumb and the dimensions of the elements are given in (3).

$$\begin{aligned} O_1A_1 = \ell_{11} = 48 \text{ mm}; A_1B_1 = \ell_{21} = 12 \text{ mm}; \\ B_1C_1 = \ell_{31} = 38 \text{ mm}; C_1O_1 = \ell_{41} = 11 \text{ mm}; \\ A_1D_1 = \ell_{51} = 41 \text{ mm}; \angle I_1O_1A_1 = \alpha_1 = 170^\circ; \\ \angle B_1A_1D_1 = \delta_1 = 105^\circ; \angle (C_1O_1, Ox) = \beta_1 = 15^\circ \end{aligned} \quad (3)$$

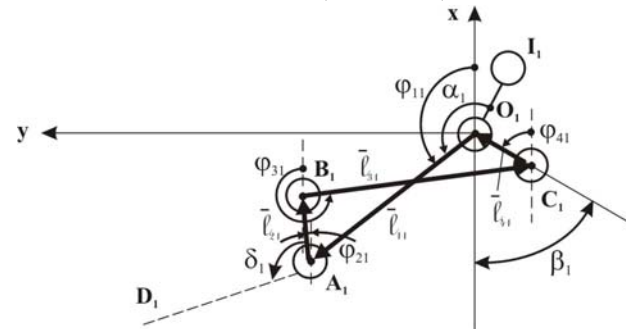


Fig. 4. Mechanisms of the thumb

The equations that characterize the movement are given in (4.1 and 4.2).

$$\begin{cases} \overline{O_1A_1} + \overline{A_1B_1} + \overline{B_1C_1} + \overline{C_1O_1} = 0 \\ \overline{r}_D = \overline{O_1C_1} + \overline{C_1B_1} + \overline{B_1A_1} + \overline{A_1D_1} \end{cases} \quad (4.1)$$

where

$$\begin{cases} \overline{O_1C_1} [l_{41} \cos(\beta_1 + 180), l_{41} \sin(\beta_1 + 180)] \\ \overline{C_1B_1} [l_{31} \cos(\varphi_{31} - 180), l_{31} \sin(\varphi_{31} - 180)] \\ \overline{B_1A_1} [l_{21} \cos(\varphi_{21} - 180), l_{21} \sin(\varphi_{21} - 180)] \\ \overline{A_1D_1} [l_{51} \cos(\gamma_1 + \varphi_{21}), l_{51} \sin(\gamma_1 + \varphi_{21})] \end{cases} \quad (4.2)$$

One of the most important requirements of an anthropomorphic gripping system is the contact between forefinger and thumb. Thus, this condition was verified by drawing the theoretically trajectories of the two fingers (figure 5).

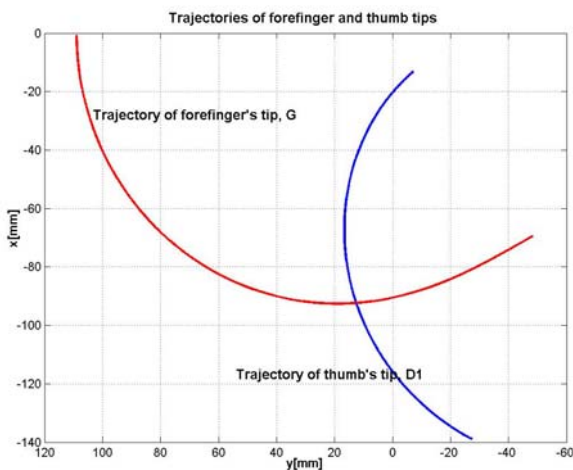


Fig. 5. Trajectories of the forefinger and thumb tips

2.3. Modelled forefinger

The mechanical structure of the forefinger was modelled using ProEngineer software. The modelled structure includes the anti-quadrilateral mechanisms the gear drive, the incremental transducer and the motor. In figures 6, 7 and 8 are shown these elements.

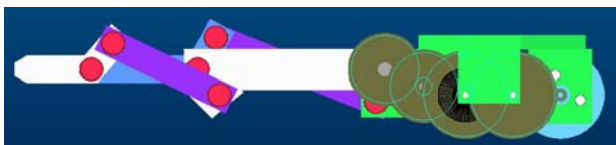


Fig. 6. Forefinger structure – front view [2]

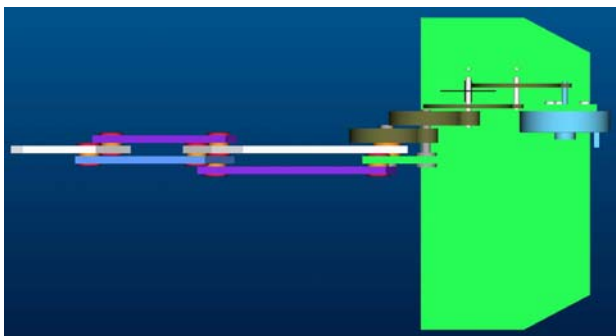


Fig. 7. Forefinger structure – top view

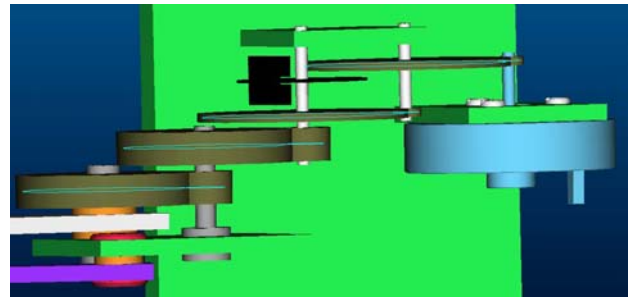


Fig. 8. Forefinger gear drive and actuator

3. Actuating system

The designed hand was considered actuated by five DC motors, one for each finger, positioned in the palm. To be able to choose the correct motor it was established the power needed to each motor. Thus, after it was considered the mechanical structure of one finger, the transmission is designed using eight cylindrical gears. The calculus started from the two possible forces needed for grasping 100N (for power grasping) and 20N (for precision grasping). Introducing the reduction ratios of the gears it resulted the two variations of the power needed to each actuator (figure 9). The ω (omega) parameter is the angular speed at the beginning of the forefinger (in joint O – figure 2).

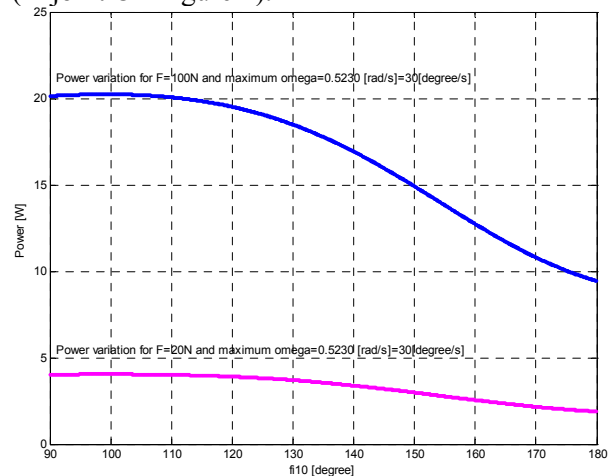


Fig.9. Variation of needed actuator power

Considering the above results for the simulations presented in this paper it was chosen a DC motor from *Faulhaber*, with torque of 3 [mNm] [6].

The most used method of control of DC motors is voltage control. By varying the supplying voltage it can be controlled the motor speed for a specific load. For the concrete case of prosthesis, the motor speed control, by varying the supplying voltage, is disadvantageous from energy consumption point of view. A decrease of energy to motor terminals determines a voltage increase on a resistance and so a wasted energy.

A more economical method, from energetic point of view, is the PWM control of a DC motor: the supplying voltage of the motor has a rectangular shape of constant frequency and a variable duty factor (the duty factor is the ratio between the one time period and signal period).

The control PWM signal is given by a microcontroller that implements a fuzzy controller [1]. The fuzzy controller is a Takagi-Sugeno one, with two inputs (position error and speed error) and one output (the PWM duty factor).

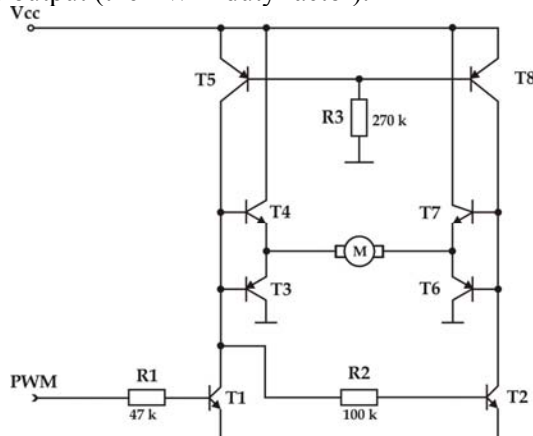


Fig.10. Simple PWM control of DC motor [2]

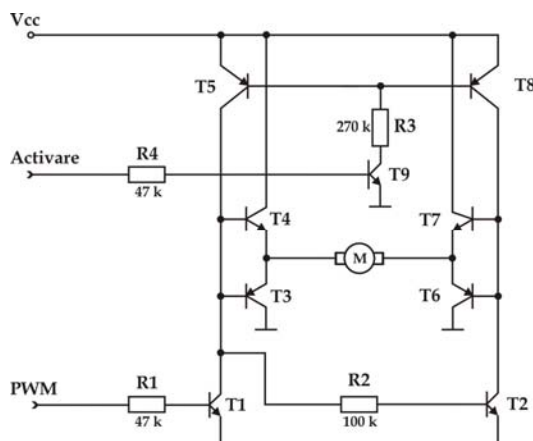


Fig. 11. Activation and PWM control of DC motor

A very simple PWM control diagram is presented in figure 10. The supplying voltage V_{cc} is continuous, stabilized (constant). The supplying circuit of the motor is a bridge circuit that has, as component elements, $T3$, $T4$ and $T6$, $T7$ transistors. The $T5$ and $T8$ transistors determine a “current-mirror” circuit that supplies the base current to $T4$ and $T7$ transistor. The $T1$ and $T2$ transistors are connected in inverter connection and so, they function in antiphase, commanding the antiphase opening of the $T4$, $T6$ transistors and, respectively, $T3$, $T7$ transistors, by absorbing the whole current from one branch of “current mirror” circuit, while on the other branch the current is not all absorbed.

If the PWM input signal is $0V$ than the $T1$ transistor is blocked, allowing the collector potential to tend to the supplying voltage (through $T5$ transistor of the “current mirror” circuit). This determines a current through $R2$ resistance to the $T2$ transistor base, which will conduct. As effect, the potential of the $T3$ and $T4$ transistors bases tends to the supplying voltage ($T4$ is strongly conducting and $T3$ is blocked) and the potential of the $T6$ and $T7$ transistors bases tends to zero ($T6$ is strongly conducting and $T7$ is blocked), determining a positive maximum voltage on motor terminals.

If the PWM input signal is $5V$ (supplied by the controller) then $T1$ is conducting, its potential tends to $0V$, the current through $R2$ resistance tends to zero and $T2$ is blocked. As effect, the $T3$ and $T7$ transistors are strongly conducting and $T4$ and $T6$ transistors are blocked. This fact determines, on motor terminals, a negative maximum voltage. If the PWM signal is constant for a period of time long enough then the motor would have maximum speed in one sense and, on changing the signal status would have the maximum speed on reverse sense.

If the PWM signal status will change with a frequency high enough and with a $\frac{1}{2}$ duty factor the motor will be kept on a passive state (the speed being zero). The motor current has small values because of the high commutation frequency relative to the motor equivalent inductance.

By changing the duty factor of the control PWM signal it is controlled the current through motor and so, the speed of the motor. In steady-state (duty factor is constant) the speed of the motor is constant when it works with no-load. A small variation of the duty factor determines a small variation of the speed of the motor. The covariation characteristic of the two parameters is, approximately, a linear one as it is the inherit mechanical characteristic of the DC motor. A small variation of the PWM duty factor, inside the interval $[0, 1]$, determines a variation of the speed between the negative maximum value and the positive maximum one.

The $R3$ resistance gives the base current for the two mirror transistors. If this resistance will not be used, the $T5$ and $T8$ transistors will block and so, the $T4$ and $T7$ transistors will also block and $T3$ and $T6$ will conduct, determining a zero potential on the motor terminals.

In figure 11 [2] the $T9$ transistor could cancel the effect of the $R3$ resistance if it is blocked (the “activation” input signal is $5V$). The activation signal could be used, in the same time, also to control a brake of the motor axle. When the “activation” signal activates the PWM device, it deactivates the motor brake and vice versa. The

brake is useful in the case of manipulation of objects, when it is desired to maintain the object gripped (this doesn't necessary mean that there is a constant force). An advantage of using such a brake is the increase of the energy utilization, which, in the prosthesis case, is a critical resource.

4 Control System

In the above paragraphs were presented the statements of the problem. Thus, for hand prosthesis or anthropomorphic robot hand there are important to be controlled the position of the fingers, because of grasping condition, and the force. This implies a control of the motor-transmission-fingers system. In the following are presented the fuzzy controller with his rules, the electronic circuit and the program that includes the motor-transmission-forefinger system.

4.1. Fuzzy controller

The fuzzy controller applied for hand is a Takagi-Sugeno controller. The designed fuzzy controller 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 "ė" are determined using the equations [8]:

$$\begin{aligned} e(k) &= n^*(k) - n(k) \\ \dot{e}(k) &= e(k) - e(k-1) \end{aligned} \tag{5}$$

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 fig. 12. 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 were attributed an estimative set of values and, after controller implementation those values were modified in the simulation process.

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 were defined the rules *R1*, ..., *R25* that were used to determine the numeric values of the fuzzy controller outputs (Table 1).

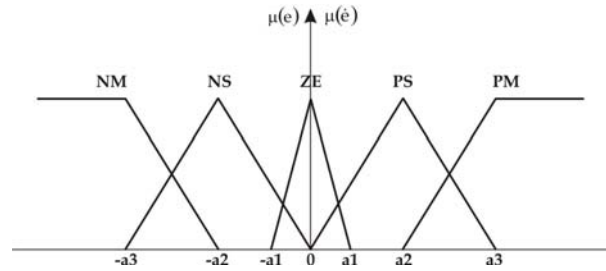


Fig.12. Membership functions of the fuzzy controller

Table 1.

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

From Table 1 it can be concluded that the fuzzy set of rules can be decomposed, based on the two input variables (angular position error and angular speed error). Thus, for angular position error, x , the rules are presented in Table 2 and the rules for angular speed error, v , in Table 3.

Table 2

NM	$S_{x1}x$
NS	$S_{x2}x$
ZE	$S_{x3}x$
PS	$S_{x2}x$
PM	$S_{x1}x$

Table 3

NM	$S_{v1}v$
NS	$S_{v2}v$
ZE	$S_{v3}v$
PS	$S_{v2}v$
PM	$S_{v1}v$

Thus, the numeric value of the output that should be computed by the controller is:

$$\text{“output”} = \text{“angular speed correction”} - \text{“angular position correction”}.$$

Starting from figure 12, there can be written the computing mode and the value of the angular position error (Table 4).

Table 4

Interval	Computing mode	Value
NM	NM	$S_{x1}x = C_x x$
$NM \cap NS$	NM+NS	$S_{x1}x + S_{x2}x = C_{1x}x$
NS	NS	$S_{x2}x = C_{2x}x$
$NS \cap ZE$	NS+ZE	$S_{x2}x + S_{x3}x = C_{3x}x$
ZE	ZE	$S_{x3}x = C_{4x}x$
$ZE \cap PS$	ZE+PS	$S_{x2}x + S_{x3}x = C_{3x}x$
PS	PS	$S_{x2}x = C_{2x}x$
$PS \cap PM$	PS+PM	$S_{x1}x + S_{x2}x = C_{1x}x$
PM	PM	$S_{x1}x = C_x x$

From figure 12 results the angular correction (*AngCorr*):

- If $x < -PR_{3x}$ then *AngCorr* is $C_x x$.
- If $-PR_{3x} \leq x < -PR_{2x}$ then *AngCorr* is $C_{1x}x$.
- If $-PR_{2x} \leq x < -PR_{1x}$ then *AngCorr* is $C_{2x}x$.
- If $-PR_{1x} \leq x < 0$ then *AngCorr* is $C_{3x}x$.
- If $x = 0$ then *AngCorr* is $C_{4x}x$.
- If $0 < x \leq PR_{1x}$ then *AngCorr* is $C_{3x}x$.
- If $PR_{1x} < x \leq PR_{2x}$ then *AngCorr* is $C_{2x}x$.
- If $PR_{2x} < x \leq PR_{3x}$ then *AngCorr* is $C_{1x}x$.
- If $PR_{3x} < x$ then *AngCorr* is $C_x x$

The above rules can be simplified by observing that:

- when angular error is $x = 0$, the constant $C_{4x}x$ may take any finite value. Thus, this rule can be included in one or both adjacent rules;
- for the extreme rules ($x < -PR_{3x}$ or $PR_{3x} < x$) the resulted values for correction are very big in modulus, relatively to numeric limits for the variation interval of PWM duty factor (the output of fuzzy controller). These values will be truncated, thus the angular correction will be considered $C_x x$;
- the rules are symmetrical relative to $x = 0$.

Considering these observations the rules become:

- If $0 \leq |x| \leq PR_{1x}$ then *AngCorr* is $C_{3x}x$.
- If $PR_{1x} < |x| \leq PR_{2x}$ then *AngCorr* is $C_{2x}x$.
- If $PR_{2x} < |x| \leq PR_{3x}$ then *AngCorr* is $C_{1x}x$.
- If $PR_{3x} < |x|$ then *AngCorr* is $C_x \text{sgn}(x)$.

The same analysis and simplification can be done for the angular speed error. Thus, the final rules are:

- If $0 \leq |v| \leq PR_{1v}$ then the *AngCorr* is $C_{3v}v$.
- If $PR_{1v} < |v| \leq PR_{2v}$ then *AngCorr* is $C_{2v}v$.

If $PR_{2v} < |v| \leq PR_{3v}$ then *AngCorr* is $C_{1v}v$.

If $PR_{3v} < |v|$ then *AngCorr* is $C_v \text{sgn}(v)$.

Where $PR_{1x}, PR_{2x}, PR_{3x}$ are the domains of the membership functions of the angular position error and $PR_{1v}, PR_{2v}, PR_{3v}$ represents the domains of the membership functions of the angular speed error (see a_1, a_2 and a_3 in figure 12). These values can be modified after the experiments or simulations.

The values for C_{1x}, C_{2x}, C_{3x} and C_{1v}, C_{2v}, C_{3v} are constants and were chosen as powers of 2 (to minimize the execution time of the microcontroller program). C_x and C_v are the outputs of extreme rules ($PR_{3x} < x$ and $PR_{3v} < v$) and can be modified based on experiments and simulations.

The output of the fuzzy controller is the difference between the angular speed correction and angular position correction. This output is then linearly corrected relative to instantaneous angular acceleration. The result is truncated to the limit of PWM duty factor domain (the output of the controller). Thus, the maximum motor torque is controlled limited in both rotation senses.

These two sets of rules will be easier to be implemented in the microcontroller's program, which has as critical resource the execution time of a minimal module (module that can not have interrupt) between two successive samplings of the signals given by the incremental transducer.

4.2. Control circuit

The rules of the fuzzy controller were introduced in a program that allowed us to simulate the hand prosthesis and to verify the mechanical design of it. Thus, was conceived a program that includes all the subsystems of the prosthesis, but also an electronic circuit that contains the motor and a microcontroller, in which has been implemented the fuzzy controller and the PWM command.

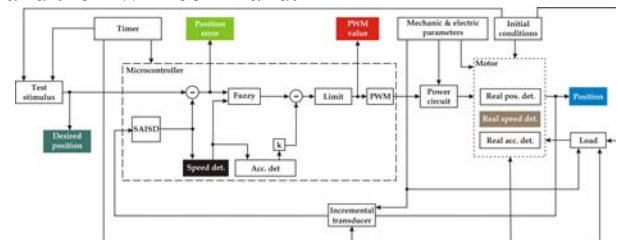


Fig. 13 Block diagram of fuzzy controller

The component blocks presented in figure 13 are: *Motor*–DC motor with permanent magnet excitation.

Power circuit – power circuit presented in figure 14.

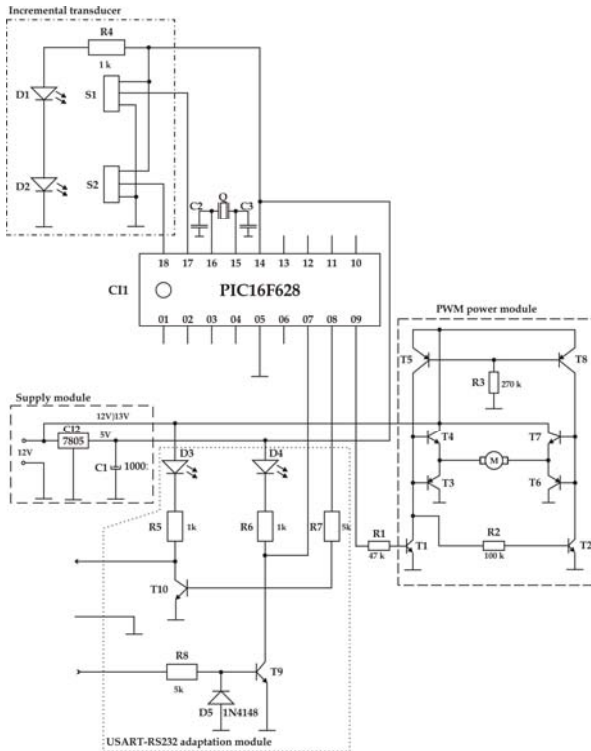


Fig. 14 Electronic circuit for forefinger control

Position transducer –incremental transducer with two channels directly mounted to the motor axle.
Microcontroller – PIC16F628, implements 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 14. The diagram contains the PWM power module of the motor, PIC16F628 microcontroller and the incremental transducer that has two IR LEDs and two phototransistors.

4.3. Controller – motor – load program

The interface of the simulation program is presented in figure 15 [2] (the window for control system). The simulation program is created in Delphi [7]. The interface of the program 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 behaviour of the system and the real one are identically.

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)* –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)* – 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)* – moment of actuating force of the finger, relative with the simulation start moment; *static friction torque (cuplul de frecare statica)* – 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 senzorialului); 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 of the two input variables: position error and speed error.

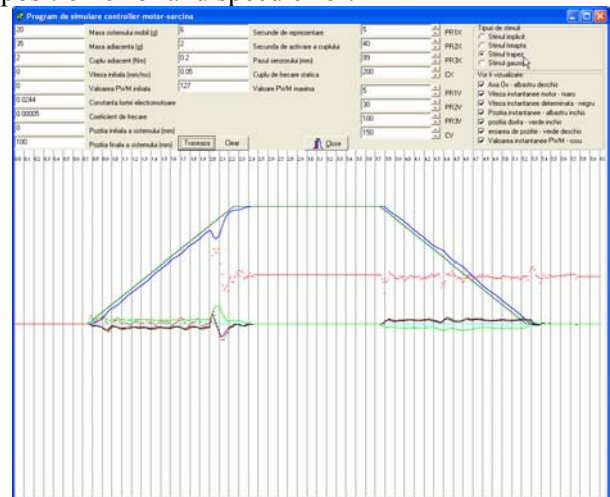


Fig. 15 Interface of simulation program

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)*; *Gaussian input*.

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 desired to be visualized. After the inputs were chosen and the type of diagrams that 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 15). 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.

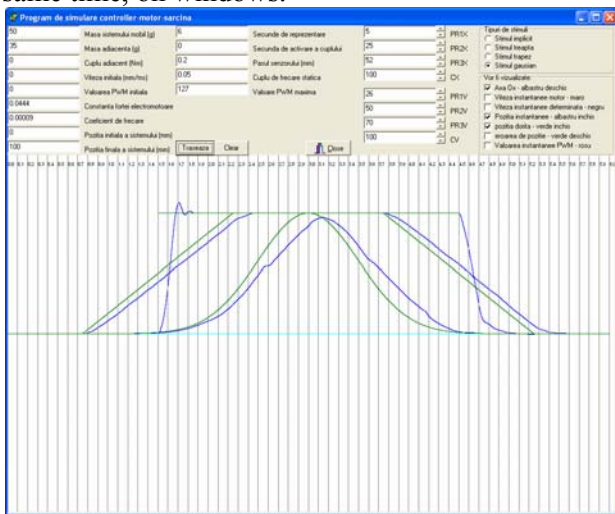


Fig. 16 Simulation for $PRIX = 5$

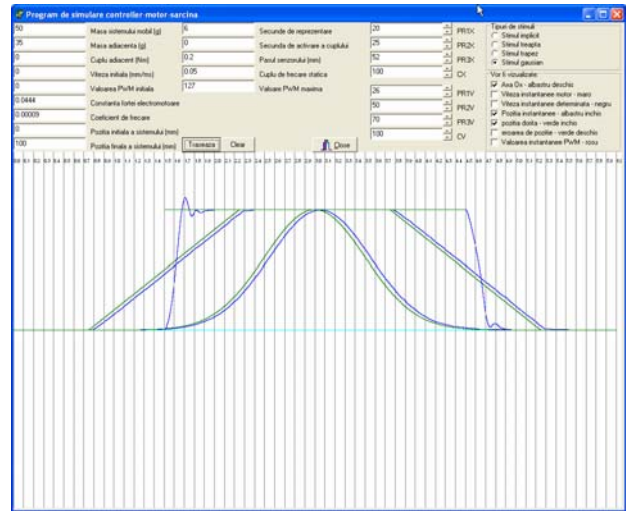


Fig. 17 Simulation for $PRIX = 20$

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. There were done many simulations to test the program and the fuzzy rules. In the above, are presented two simulation, difference between them is giving by a $PRIX$ parameter of the fuzzy controller. The simulation period of time is in both cases is 6s. If there are compared the two simulations it can be remarked that the responses to trapeze and, respectively, Gaussian input are more accurate, while the response to impulse input has a insignificant degradation (the maximum amplitude of oscillation increases, and the total damping time is also increases with approximately 0.1s). Although, in most of the applications, the response of step-input or impulse-input is more often used, and using the parameters shown in figure 17 this response is not the best one, these parameters were chosen to be the final ones (the decrease of the quality of step-input or impulse-input is insignificant).

4.4. Simulation program

There was made a program that can be used in designing both hand prosthesis and anthropomorphic robotic hand. The program was done using Delphi software.

The program includes the mathematical model of all the above subsystems that form the hand system: mechanical, electrical and control subsystems

The logic diagram of the program is given in figure 18.

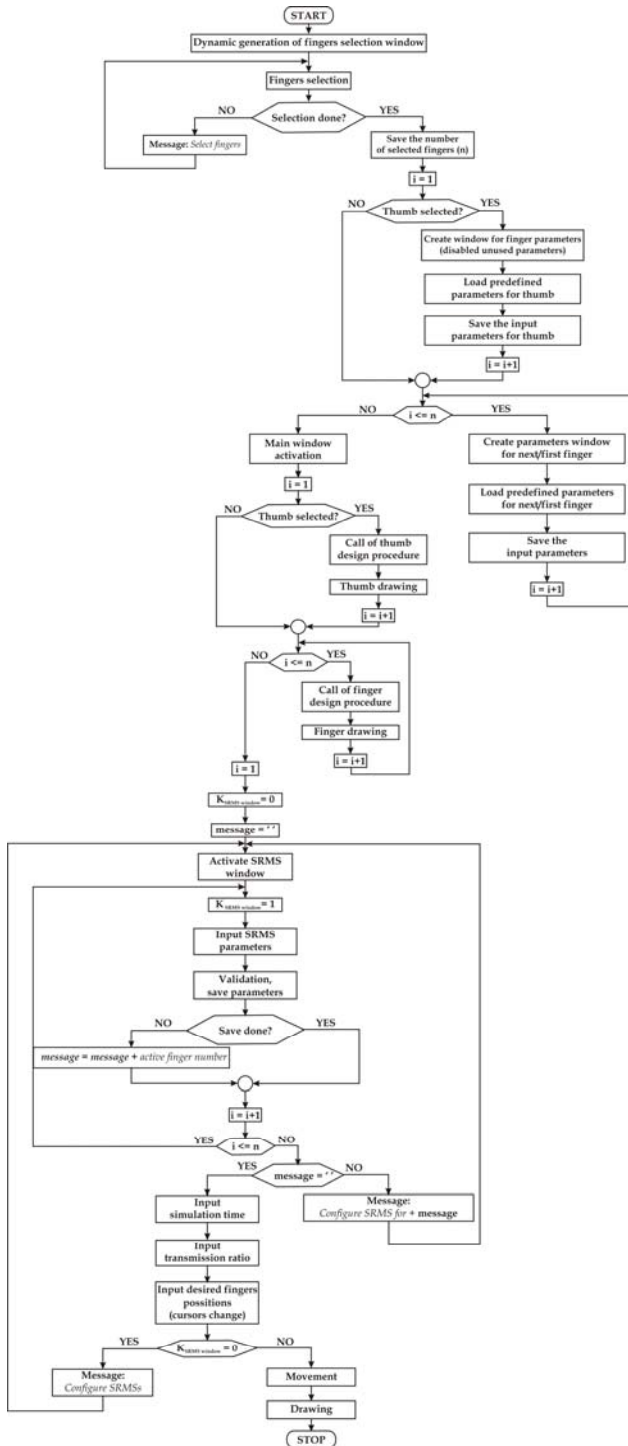


Fig. 18. Logic diagram of the program [1]

The program interface contains a main window (figure 19), which allows the visualization of the fingers, dynamic movement (animation based on mathematical equations described above) and access to controller window [1] and graphical visualization of the movement of the forefinger tip.

The first window that appears when the program is started allows the user to choose the number of the fingers that are needed and also the type of the finger (figure 20a). After that, is opened, for each

chosen finger, a window that contains all the geometrical parameters that describe the finger (figure 20b) [1]. The main window of the program will not be active until all the geometrical elements and all the desired fingers are completely defined.

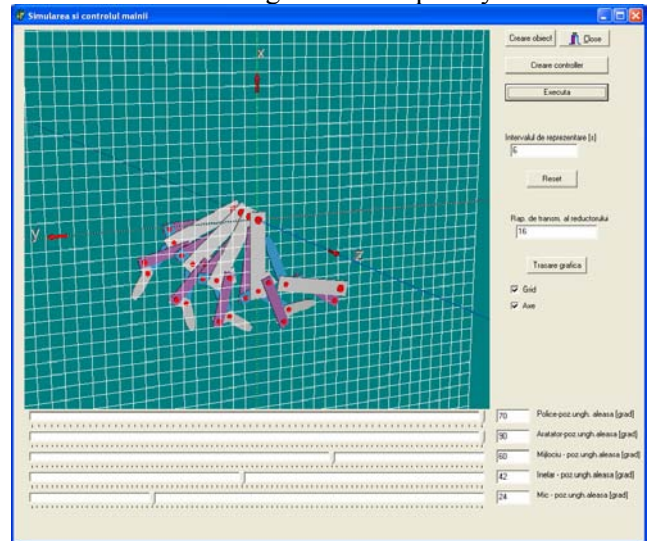


Fig. 19. The main window of the program [1]

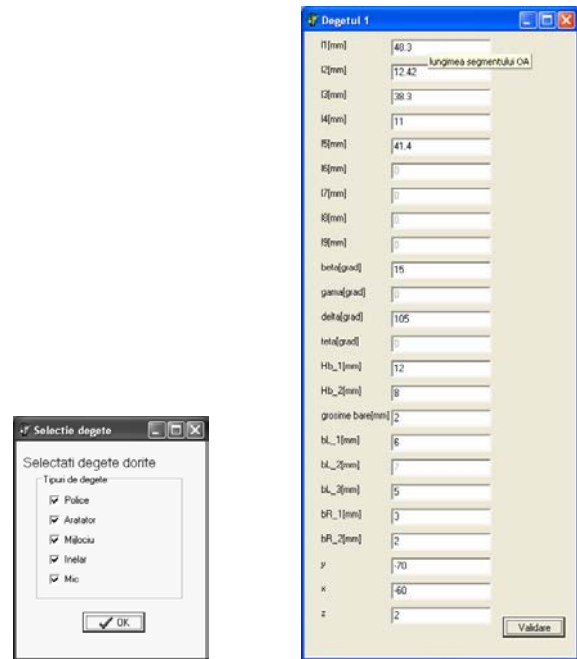


Fig. 20. The Input data windows of the simulation program

The main window allows introducing the desired positions for each chosen finger (from the scroll bars at the bottom of the window), to obtain the graphic of the tip of the forefinger (button *Trasare grafica*), to visualize or not the axis and the grid and to execute the movements.

Also, the main window of the program connects the mechanical part to the control system, through

Creare controllere (Controllers design) button, which opens the controllers' window (figure 21).



Fig. 21. Controllers' parameters window

The controllers' parameters window allows introducing the main parameters of each controller (for each chosen finger there is a controller): membership functions intervals, weight of the system, the supplementary weight, PWM duty factor, step of the incremental transducer, etc.

4 Conclusion and future work

The program made by the authors of this paper tries to solve the problem of designing hand prosthesis or anthropomorphic robotic hand by offering software that includes all the component subsystems.

The program allows visualising the movement of the hand (figure 22) and the trajectory of the tip of forefinger (figure 23). It is a very complex program and a very general one, because, from control point of view, can be determined the optimum control for each finger by having one controller/finger.

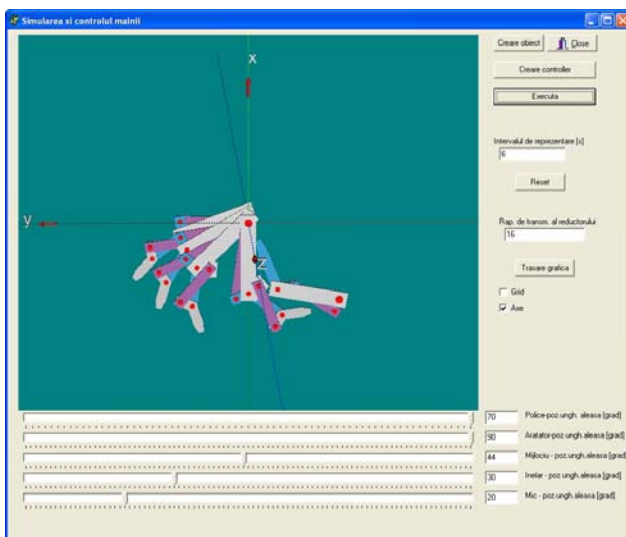


Fig. 22. Movement of the hand

In the future, the program will be developed thus it will contain different types of mechanical structures and also different types of controllers. It is also desired that each mechanical structure to have a subprogram dedicated to the stress analysis.

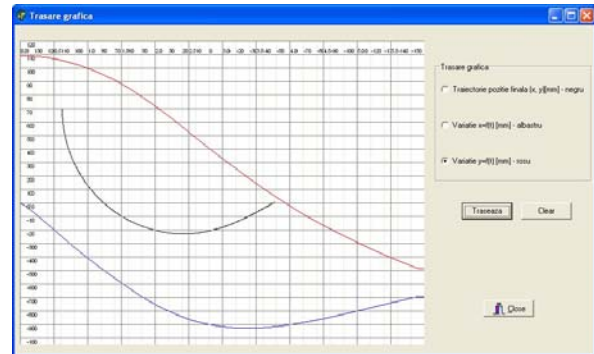


Fig. 23. Movement of the forefinger's tip

The authors of the present paper developed also software dedicated to vocal command of hand prosthesis that can be adapted to each structure [3].

References:

- [1] C. Chivu, *Intelligent control for upper limb prosthesis*, PhD Thesis, "Politehnica" University of Bucharest, 2006.
- [2] C. Chivu, C. Chivu, *Fuzzy control applied in prosthetics*, Advanced Topics on Fuzzy Systems, Published by WSEAS Press, ISBN 978-960-6766-57-2, ISSN 1790-5109, pg. 100-107, 2008.
- [3] C. Chivu, C. Chivu, *Automated dialog systems for Romanian language*, Computational methods and intelligent systems, Published by WSEAS Press, ISBN 978-960-6766-60-2, ISSN 1790-5117, pg. 83-91, 2008.
- [4] S. Fujii, D. Nishikawa, H. Yoko, *Development of a Prosthetic Hand Using Adaptable Control Method for Human Characteristics*, vol.5, IOS Press, Amsterdam, Olanda, pg. 360-376, 1998
- [5] Ho-Rim Lee, *Comparative Study of Bond Characteristics between Titanium/Titanium Alloy and Ceramic*, PhD Thesis, Tübingen University, 2004.
- [6] Faulhaber Group, *DC motors catalogue*, 2607SR
- [7] S. Kovacs, *Delphi 3.0. User guide*. (in Romanian), "Albastră" Publishing House, Cluj-Napoca, Romania, 2001.
- [8] K. Ogata, *Modern control engineering*, Prentice Hall Intern., Minnessota, 1990.
- [9] *Southampton hand*, Univ. Southampton, 2003, www.bib.ecs.soton.ac.uk/
- [10] S. Schulz, C. Pylatiuk, G. Bretthauer, *A new ultralight antropomorphic hand*. In IEEE Conf. On Robotics and Automation ICRA 2001, Seoul, Coreea de Sud, May 21-26 2001, pg. 2437-2441.