



A GRAPHICAL USER INTERFACE FOR PROGRAMMING STEPPER MOTORS USED AT DIFFERENT KINDS OF APPLICATIONS

Ahmet Altintas

Department of Electrical Education

Dumlupinar University, Faculty of Technical Education, 43500

Simav, Kütahya, Turkey

a_altintas@dumlupinar.edu.tr

Abstract- Stepper motors are a digital actuator whose input is in the form of programmed energization of the stator windings and whose output is in the form of discrete angular rotation. It is, therefore, ideally suited for use as an actuator in computer control systems, digital control systems and etc. Control systems employing stepper motors as actuators are known as incremental motion control systems. In this study both a GUI and a programming page driven from Matlab program are developed for programming and controlling different types of stepper motors at different drive modes. The parallel port of a PC is used to drive stepper motors. By means of this software, stepper motors having up to 8-separate-excitating signal can be programmable and controllable easily. In the software many different rotational motions are defined as built-in functions. In addition, it provides that the user can define new built-in functions, too. It can be, also, considered as a learning aid system for students to get a comprehensive understanding of stepper motors.

Key Words- Stepper motor, GUI, Programming

1. INTRODUCTION

A stepper motor is an electromechanical device which converts electrical pulses into discrete mechanical movements. The shaft or spindle of a stepper motor rotates in discrete step increments when electrical command pulses are applied to it in the proper sequence. The sequence of the applied pulses is directly related to the direction of motor rotation. The speed of the motor's shaft is directly related to the frequency of the input pulses; and the length of rotation is directly related to the number of input pulses applied. One of the most significant advantages of a stepper motor is its ability to be accurately controlled in an open loop system. This type of control eliminates the need for expensive sensing and feedback devices such as optical encoders. Its position is known simply by keeping track of the input step pulses. A stepper motor can be a good choice whenever controlled movement is required. They can be used to advantage in applications where rotation angle, speed, position and synchronism need to be controlled. Because of the inherent advantages, stepper motors have found their place in many different applications. Some of these include printers, plotters, high end office equipment, hard disk drives, medical equipment, fax machines, machine tools, automotive devices and many more [1,2,3].

There are three basic types of stepping motors: Variable-reluctance (VR), Permanent magnet (PM) and Hybrid (HB). PM motors have a magnetized rotor, while

VR motors have toothed soft-iron rotors. HB motors combine aspects of both PM and VR technology. The two most commonly used types of stepper motors are the PM and the HB types. The HB stepper motor is more expensive than the PM stepper motor but provides better performance with respect to step resolution, torque and speed. Typical step angles for the HB stepper motor range from 3.6° to 0.9° (100 – 400 steps per revolution). The HB stepper motor combines the best features of both the PM and VR type stepper motors.

There are three common types of drive modes of stepper motors: Wave Drive, Full Step Drive and Half Step Drive. In wave drive only one winding is energized at any given time. In full step drive two phases should be energized at any given time. Full step mode results in the same angular movement as one-phase-on drive; but, the mechanical position is offset by one half of a full step. Half step drive combines both wave and full step drive modes. Every second step only one phase is energized, and during the other steps two phases are energized. This process results in angular movements that are half of those in one- or two-phase-on drive modes.

Stepper motors have a wide range of applications. Some applications require that a stepper motor should rotate continuously or periodically with a constant speed or a variable speed; some applications also require that it should position a device at the right time to a certain position according to a program [4,5]; some of them require accelerating or decelerating motions up to a certain speed; some require mixed motions of them and etc. [6,7]. When these different motion types are taken into account, developing a computer program for programming and controlling the stepper motors is a practical and wise way. For this purpose, some commercial companies have produced programmable stepper motors controlled with micro-controllers (KT-5197 Serial Bipolar Stepper Motor Driver, STÖGRA Stepping Motor Controls, PIC-STEP Stepper Motor Controller Board, STPM DRV807 Optimal Engineering Systems and etc.). Micro-controllers have limited motion types, relatively. And, it is impossible to add a new motion type to micro-controllers by the user.

In this paper both a GUI (Graphical User Interface) and a programming page driven from Matlab program are developed for programming and controlling different types of stepper motors at different drive modes. The GUI is built with benefiting from the facilities offered by the Matlab programming language. Matlab implements GUIs through GUIDE (Graphical User Interface Development Environment). All kinds of stepper motors having up to 8-separate-excitating signal can be programmable and controllable real-timely by using this software via parallel port of a PC. In the developed GUI, many different rotational motions are both assigned to push buttons and defined as built-in functions. Thus, the user can control the stepper motor by the aid of either push buttons or a computer program. The push buttons can make only their own assigned motions. On the contrary, the computer program using built-in functions can makes a lot of complex motions successively. The GUI also provides defining user-defined built-in functions which perform new motion types.

2. MATERIAL AND METHOD

Matlab is matrix-based software for scientific and engineering numeric computation and visualization. Matlab is chosen as the programming tool primarily

because of simple GUIs, immediate graphics facilities, built-in functions, the possibility of adding user-written functions, interactive mode of work, simple programming and its wide availability on computing platforms. These factors make Matlab a powerful tool for research and practical problem solving [8]. Matlab implements GUI through GUIDE. A GUI is a user interface program built with graphical objects such as buttons, text fields, sliders and menus. In fact, these definitions are well known for almost all computer users. Applications that provide GUIs are generally easier to learn and use since the person using the application does not need to know what commands are available or how to use them. GUIs assure the communication process between the user and the system's inference engine [9].

The parallel port of a PC is used to drive stepper motors by using the DB-25 parallel port connector. The PC's parallel port is a simple and inexpensive tool for building computer controlled devices and projects. It is an 8-bit bidirectional interface designed to transmit data synchronously using TTL signal level. TTL outputs can source 2.6 mA and sink 24 mA. Thus, small-size stepper motors can be driven from a PC's parallel port with a simple interface circuit containing a transistor. Almost all programming languages allow programmers to access parallel port using some library functions. Matlab program also has some useful library functions such as `digitalio`, `addline`, `putvalue`.

Animation of the variable-reluctance-type stepper motor in the main GUI window is carried out with computer graphics. Computer graphics are based on the fundamental linear transformation matrixes. Computer graphics is usually formed by rotating, translating, scaling and performing various projections on the data [10]. Like other motors, the stepper motors have stators and rotors. Because of being rotary part of the motor, only the rotor should be rotated in order to make an animation. Profiles of the motor teeth are formed with the position vectors. Actually, if the teeth numbers are increased appropriately from the GUI window, other stepper motor types will be animated basically.

3. AN OVERVIEW OF THE GUI

Basically, the developed GUI consists of two sets of programs. One set is responsible for the main GUI window; and the other is responsible for the programming page. The main GUI window offers to enter parameters of the stepper motor which will be employed, sets up the computer connection in order to run the stepper motor by using a PC, gives a list of different operating modes and runs the stepper motor according to the running mode; besides, it allows the user to animate variable-reluctance-type stepper motor. The main GUI window, displaying the default values, is given in Fig.1. All of the operating modes in question can be activated with push-buttons labeled with their own names. As seen from Fig.1, a total of twenty-seven operating modes are defined.

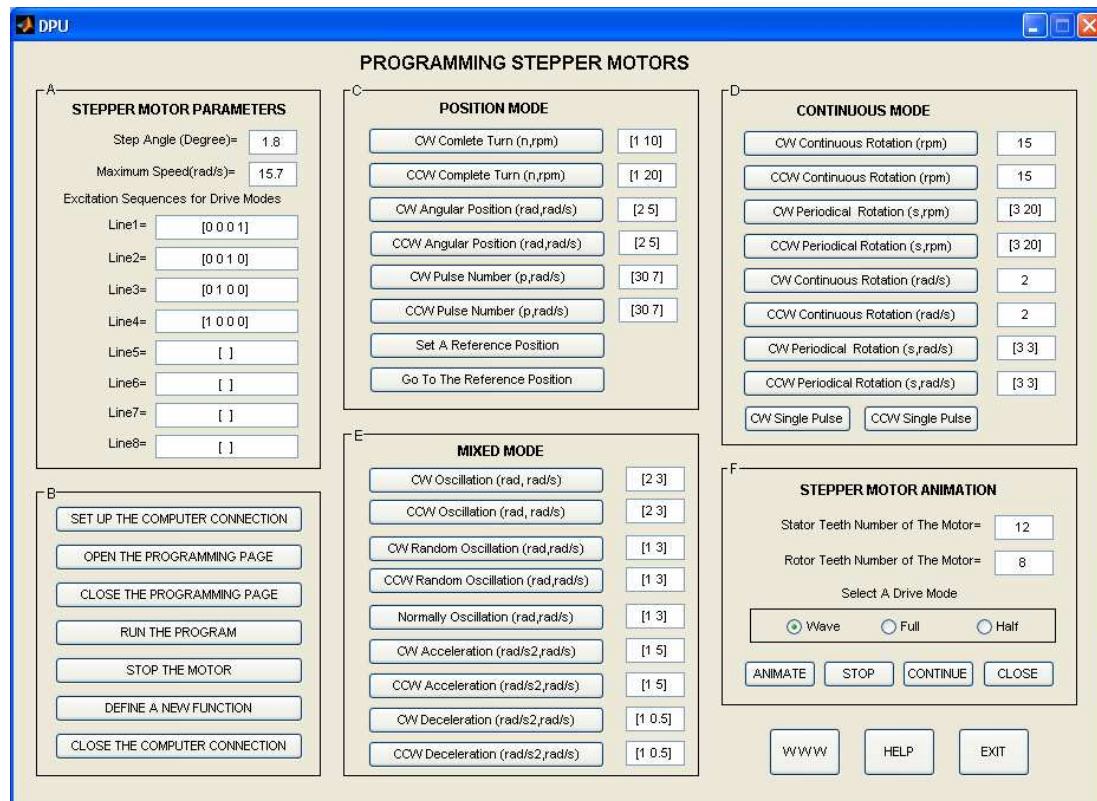


Figure 1. Main GUI window

The main GUI window mainly consists of six-frame (A, B, C, D, E, F). Frame-A is the part for inputting the stepper motor's parameters such as the step angle, maximum speed and excitation sequences. Usually stepper motors have two phases, but three-and five-phase motors also exist. A bipolar motor with two phases has one winding/phase and a unipolar motor has one winding, with a center tap per phase. Taking into account this case, a computer connection with 8-line via parallel port is set up. So, 8-different excitation signals can be taken from the PC. According to the selected stepper motor type and drive modes (wave, full, half), the user must enter the excitation sequences. Frame-B is the part that the user carries out various tasks, such as setting up or closing the computer connection, opening or closing the programming page, running and stopping the stepper motor and etc. Frame-C, D and E are the parts for position, continuous and mixed operating modes, sequentially. The stepper motor can be controlled in position mode, continuous (velocity) mode or mixed mode. Briefly, position mode moves a device to a position; continuous mode causes the motor to revolve continuously; mixed mode makes different physical movements. Frame-F is the part for displaying a visual animation about VR type stepper motor according to the stator & rotor teeth numbers and drive mode. Exemplary views of animation window are given in Fig.2. The arrow sketch positioned to the rotor axis represents rotor's movement in respect of step angle and drive mode.

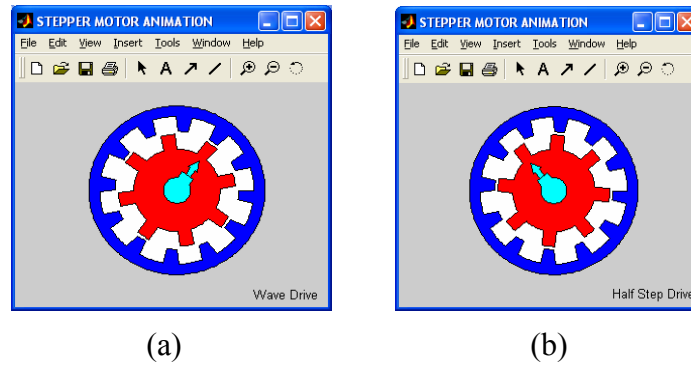


Figure 2. Exemplary views of animation window, a) Wave drive, b) Half-step drive

The other main window is the programming page. When the user pushes the push button labeled with ‘OPEN THE PROGRAMMING PAGE’ in Frame-B, the programming page will pop-up and offer the user writing the program code for controlling the stepper motor independent from the main GUI window. An exemplary view of the programming page is given in Fig.3. The menus and toolbar located to top of the window can be used for classical window operations such as saving, copying, printing and etc. First thirteen-line in Fig.3 is the part for defining global parameters and setting up the computer connection.

```

1 % A SAMPLE PROGRAM FOR STEPPER MOTOR CONTROL
2 global dio line0 test driving_pulses sa maxh a aa, a=1; aa=0;
3
4 sa=1.8; %Step angle
5 maxh=15.7; %Maximum Speed
6 driving_pulses=[0 0 1;
7               0 0 1 0;
8               0 1 0 0;
9               1 0 0 0]; %Excitation Signals
10
11 dio=digitalio('parallel','LPT1');
12 line0=addline(dio,0:3,0,'out','line0');
13 putvalue(dio.line0,not([0 0 0 0]));
14
15 for k=1:10
16     SetRefPos % Set A Reference Position
17     CWrw(3.1,5) % rotate 3.1 radyan with 5 radyan/s.
18     pause(2) % wait 2 seconds.
19     CCWrw(1.2,3.5) % rotate 1.2 radyan with 3.5 radyan/s.
20     pause(1) % wait 1 second.
21     CWpw(33,7) % apply 33 pulses with 7 radyan/s.
22     CWsn(2.2,30) % rotate 2.2 seconds with 30 rpm.
23     GoRefPos % Go to the Reference Position
24 end

```

Figure 3. A sample program

```

1 %FUNCTION FOR CW PERIODICAL ROTATION
2 function CWsn(s,n)
3 global dio line0 test driving_pulses sa maxh a aa
4 test=0; % used for stopping the motor
5 tic
6 for rotate=1:inf
7     putvalue(dio.line0,driving_pulses(a,:));
8     toc
9     if toc<s
10         pause(sa/(n*6))
11     else
12         test=1;
13     end
14     a=a+1; aa=aa+1;
15     if a>4,a=1;end
16     if test==1, break,end
17 end
18 putvalue(dio.line0,not([0 0 0 0]));
19

```

Figure 4. A sample built-in function

In order to use operating modes as shortened codes in the programming page, all of the operating modes are defined as built-in functions separately. Naturally, input parameters of the built-in functions will be different from each other. A sample built-in function performing periodical rotation in CW direction (CWsn) is given in Fig.4. Names of the built-in functions and detailed descriptions of those are presented in Table 1. As seen from Table 1, almost all operating modes of the stepper motor fall into the

two categories: the clockwise direction (CW) and the counterclockwise direction (CCW). The functions given in Table 1 can be used together in the programming page, and complex motions can be obtainable. After writing & saving a program and closing the program page by push button labeled with 'CLOSE THE PROGRAMMING PAGE', the stepper motor can be run by push button labeled with 'RUN THE PROGRAM'. At any time the stepper motor can be stopped by push button labeled with 'STOP THE STEPPER MOTOR'. If the built-in functions given in Table 1 are not sufficient for a specific motion, the programmer can define a new function himself by push button labeled with 'DEFINE A NEW FUNCTION'. In this case, the programmer must add the global parameters to his own function file. Thus, the newly defined function can access the global parameters defined in the main program.

Table 1. Names of the built-in functions and descriptions

GUI Name	Function Name	Description
CW Complete Turn (n,rpm)	CWtn(n,rpm)	Rotating n turns with the speed of rpm (CW)
CCW Complete Turn (n,rpm)	CCWtn(n,rpm)	Rotating n turns with the speed of rpm (CCW)
CW Angular Position (rad,rad/s)	CWrw(rad,rad/s)	Rotating rad radians wit the speed of rad/s (CW)
CCW Angular Position (rad,rad/s)	CWrw(rad,rad/s)	Rotating rad radians wit the speed of rad/s (CCW)
CW Pulse Number (p,rad/s)	CWpw(p,rad/s)	Applyig p pulses to the motor with the speed of rad/s (CW)
CCW Pulse Number (p,rad/s)	CCWpw(p,rad/s)	Applyig p pulses to the motor with the speed of rad/s (CCW)
Set A Reference Position	SetRefPos	Setting up a reference position
Go to Reference Position	GoRefPos	Going to the reference position set up before
CW Continuous Rotation (rpm)	CWn(rpm)	Rotating continuously with the speed of rpm (CW)
CCW Continuous Rotation (rpm)	CCWn(rpm)	Rotating continuously with the speed of rpm (CCW)
CW Periodical Rotation (s,rpm)	CWsn(s,rpm)	Rotating s seconds with the speed of rpm (CW)
CCW Periodical Rotation (s,rpm)	CCWsn(s,rpm)	Rotating s seconds with the speed of rpm (CCW)
CW Continuous Rotation (rad/s)	CWw(rad/s)	Rotating continuously with the speed of rad/s (CW)
CCW Continuous Rotation (rad/s)	CCWw(rad/s)	Rotating continuously with the speed of rad/s (CCW)
CW Periodical Rotation (s,rad/s)	CWsw(s,rad/s)	Rotating s seconds with the speed of rad/s (CW)
CCW Periodical Rotation (s,rad/s)	CCWsw(s,rad/s)	Rotating s seconds with the speed of rad/s (CCW)
CW Single Pulse	CWp	Applyig a single pulse in a CW direction
CCW Single Pulse	CCWp	Applyig a single pulse in a CCW direction
CW Oscillation (rad,rad/s)	CWOrw(rad,rad/s)	Oscillating from the last position to the rad radian with the speed of rad/s (CW)
CCW Oscillation (rad,rad/s)	CCWOrw(rad,rad/s)	Oscillating from the last position to the rad radian with the speed of rad/s (CCW)
CW Random Osc. (rad,rad/s)	CWRrw(rad,rad/s)	Oscillating randomly with the speed of rad/s (CW); rad is normalized to 1
CCW Random Osc. (rad,rad/s)	CCWRrw(rad,rad/s)	Oscillating randomly with the speed of rad/s (CCW); rad is normalized to 1
Normally Oscillation (rad,rad/s)	NOrw(rad,rad/s)	Oscillating normally with the speed of rad/s; rad is normalized to 1
CW Acceleration (rad/s ² ,rad/s)	CWAaw(rad/s ² ,rad/s)	Accelerating motion of rad/s ² to a certain speed of rad/s (CW)
CCW Acceleration (rad/s ² ,rad/s)	CCWAaw(rad/s ² ,rad/s)	Accelerating motion of rad/s ² to a certain speed of rad/s (CCW)
CW Deceleration (rad/s ² ,rad/s)	CWDaw(rad/s ² ,rad/s)	Decelerating motion of rad/s ² to a certain speed of rad/s (CW)
CCW Deceleration (rad/s ² ,rad/s)	CCWDaw(rad/s ² ,rad/s)	Decelerating motion of rad/s ² to a certain speed of rad/s (CCW)

Nameplate data of the hybrid stepper motor, used in the exemplary application, is given in Table 2. Measured value of the loadless maximum speed is 15.7 rad/s (~150 rpm). The motor is run with each excitation mode; and good results are obtained. The stepper motor can maintain step-angle accuracy within $\pm 0.45^\circ$ at half step drive, $\pm 0.9^\circ$ at wave and full step drive. Step losses are not occurred if maximum speed is not exceeded. When the stepper motor has a load, the maximum speed of the motor will naturally decrease. In this case, maximum speed parameter in Frame-A must be changed correctly.

Table 2. Nameplate data of the hybrid stepper motor

MARCA MINEBEA Co. Ltd – ASTROSYN, Stepper Motor, Type 23LM-C355-P6V	
Rated Current: 1.5 Amps	Detent Torque Nm (g-Cm): 0.0540 (550)
Winding Resistance: 2.2 Ohms	Rotor Inertia (Kg-Cm2): 0.110
Inductance: 5.5 mH	1.8 Deg/Step
Holding Torque Nm (Kg-Cm): 0.614 (6.2)	200 Steps/Rev.

Three different digital oscilloscope views taken from experimental setup are given in Fig.5. The two signals in all frames represent first two-driving signals of the motor. Fig.5a shows an application of CW periodic rotation function (CWsw(3,2)); in this figure, the motor has rotated 3 seconds with a speed of 2 rad/s. Fig.5b displays an application of CW pulse number function (CWpw(30,2)); in this frame, 30 pulses have been applied to the motor two times with a speed of 2 rad/s. Fig.5c displays an application of CW acceleration function (CWAaw(1,3)); in here, the motor has made an accelerating motion of 1 rad/s² from stationary position to a speed of 3 rad/s.

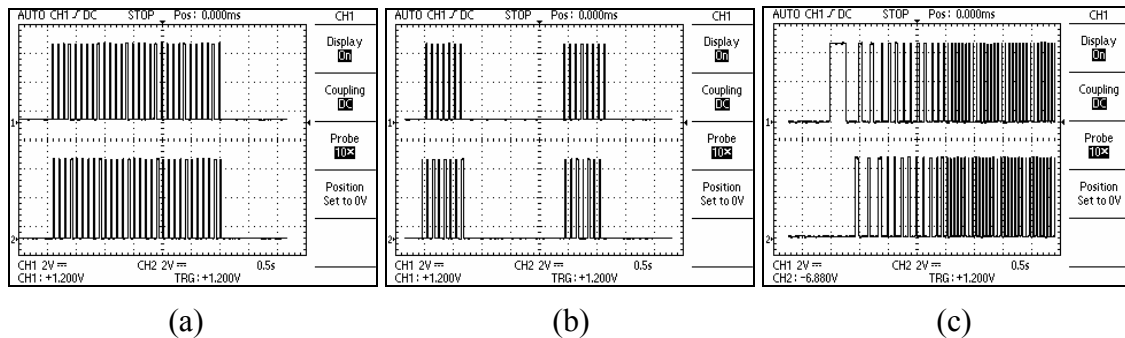


Figure 5. Measurement results, a) CWsw(3,2), b) CWpw(30,2), b) CWAaw(1,3)

4. CONCLUSIONS

Special abilities to provide accurate control over speed and position combined with their small size and relatively low cost make stepper motors a popular choice in a range of applications. Stepper motors make very different rotating motions in applications. In this paper a computer program has been developed for programming and driving all kinds of stepper motors up to 8-different excitation signals via the parallel port of a PC; a GUI performing all defined motions individually are also improved. In addition, the GUI allows programmers to define user-defined functions. So, the programmers can make stepper motors rotate in all manners. The developed program is tested with a stepper motor of 1.8° step-angle in wave, full and half drive modes. Very satisfactory results are obtained from experimental setup. The program can be used in applications containing stepper motors performing complex motions.

5. REFERENCES

1. T. Kenjo and A. Sugawara, *Stepping Motors and Their Microprocessor Controls*, 2.ed., Oxford University Press, England, 1994.
2. P. P. Acarnley, *Stepping Motors: A Guide to Modern Theory and Practice*, 4.ed, IEE Control Engineering Series 63, The Institution of Electrical Engineers, London, 1984.
3. P. Krause and O. Wasynczuk, *Electromechanical Motion Devices*, McGraw-Hill Pub., New York, 1989.
4. J. Petrovic, and S. Strmcnik, A microcomputer-based speed controller for lift drives, *IEEE Trans. Ind. Appl.* **24**(3), 487-498, 1988.
5. M. Zribi and J. Chiasson, Position control of a PM stepper motor by exact linearization, *IEEE Trans. Automatic Control* **36**, 620-625, 1991.
6. D.O. Carrica, S.A. Gonzalez, M. Benedetti, A novel algorithm for controlling stepper motors at higher speed, *Proc. IMechE Part I:J Systems and Control Engineering* **217**, 359-365, 2003.
7. D.O. Carrica, S.A. Gonzalez, M. Benedetti, A high speed velocity control algorithm of multiple stepper motors, *Elsevier- mechatronics* **14**, 675-684, 2004.
8. A. Biran and M. Breiner, *MATLAB for Engineers*, Addison-Wesley Pub., New York, 1995.
9. Matlab, *Creating Graphical User Interfaces, Version 7*, The Mathworks Inc., Natick/MA, 2006.
10. D.F. Rogers and J.A. Adams, *Mathematical Elements for Computer Graphics 2.ed*, McGraw-Hill Pub., New York, 1990.