

AN EDUCATION TOOL STUDY ON MECHATRONICS: EMULATION OF STEPPER MOTOR DRIVING SYSTEMS BY USING A MICROCONTROLLER BASED SYSTEM INTERFACE

Akdoğan, Erhan
eakdogan@marmara.edu.tr

Topuz, Vedat
vtopuz@marmara.edu.tr,

Akbaş, Ahmet
ahmetakbas@marmara.edu.tr

*Marmara University, Vocational School of Technical Sciences
Istanbul, Turkey*

Abstract

Some type of electromechanical actuator related driving systems, such as stepper motor driving systems, have various type of implementations in the mechatronic systems. Besides the implementation variety, rapid changes in the actuator and driver technologies make it a costly and difficult task training about the such type of mechatronic systems. It is most acute for undergraduate programs, specially. However, with the introduction of computers and flexible programming techniques, it is now possible to simulate these systems easily. Systems can also be emulated economically, in this way. Therefore, electromechanical actuator related mechatronic systems can be reached and/or trained, by computer simulations and/or emulations. This paper describes such a teaching and training tool study on stepper motor driving systems, including full-step, half-step and micro-step driving implementations. System emulations are fulfilled by using a microcontroller based system interface, in the study.

1. Introduction

The term 'mechatronics' was first introduced by a Japanese engineer in 1969. It was defined as 'the synergistic combination of precision mechanical engineering, electronic control, and systems thinking in design of products and manufacturing processes' by The International Federation for the Theory of Machines and Mechanisms (IFTMM). [1]

The term soon became popular. So that, it has been developing as one of the most common interdisciplinary technological approach in recent years. Now mechatronic is increasingly forming a crucial part in design, manufacture and maintenance of a wide range of technological products and processes.

IFTMM definition helps to steer the philosophies of mechatronics technology, including development, design, and manufacturing. Some critical characteristics of IFTMM definition, such as the 'synergistic combination of mechanical and electronic engineering' seem lacking in numerous mechatronics product that range from digital watches to electric robots. [2]

The products referred to as 'mechatronics' gained significantly enhanced function and performance, depending on the rapid development in microprocessor-related electronics, and in power electronics [3]. In this context, high-speed switching products such as various type of servo motors and their drivers have led to widespread adoption. This study is related such a subject application in connection with the education aim.

As a consequence of rapid developments in mechatronics related technologies, technologists who are related with mechatronics approach need skills and knowledge that are not confined to a single subject area. They need skills and knowledge that are confined to all the mechatronics related areas that are electricity, electronics, control, computer and mechanics. [4]

So, it is a difficult task teaching about all the subjects of a mechatronic systems. It is also costly for laboratory practices. In addition to these difficulties, it has additional costs to adopt to rapid technological developments in all the related areas of subjects.

This situation is most acute for undergraduate programs in underdevelopment countries, specially. Because of these difficulties, simulation programs have reached wide using capacity in mechatronic applications for both education and industry; so that, simulations have proven very useful for saving both time and money. [5]

However, because of that, simulators usually involve entering a data file from the keyboard into the computer

and having it calculate results on predefined parameters, simulations have limitations in analyzing data in real time and not being able to operate indefinitely. In this point, system emulation is a simple variation on the theme of simulation where the computer system operates in real time receiving data continually, processing it as if the computer is a controller and outputting results immediately. Such a system can be evaluated as a virtual system. [6]

This paper presents some examples of mechatronic system emulations about the stepper motor driving systems including full-step, half-step and micro-step drivings. In the following paragraphs, first stepper motor driving modes used in the simulation study have been presented briefly. Then, developed simulation and emulation system have been introduced with its microcontroller based system interface.

2. Stepper Motor Driving Modes

One electrical period of stepper motor corresponds to 4 full-step of mechanical movement. That is, 360 degrees of electrical angle doesn't same the stepper motor's rotation angle. One full electrical period finishes after four full mechanical step, which corresponds to 90 degrees of electrical angle.

Some applications are required smaller angle than full step angle. Half-step and micro-step driving modes were developed with this aim. Thus, the stepper-motor runing modes include full-step, half-step and micro-stepping modes. Half-step modes has 1/2 rate of full-step. Micro-stepping modes have 1/4, 1/8, 1/16, 1/32 rates of full-step.

In the steady state, the variable current applied to the stator windings can be explained by function of $\sin \theta$ and $\cos \theta$ as depicted in Figure 1.

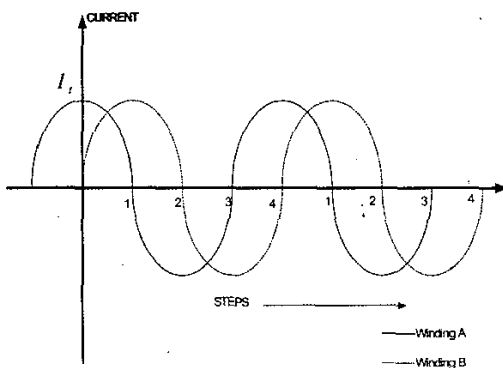


Fig 1. Stator current forms in steady state.

So that, when energy is applied to stator windings, the rotor is balanced due to final current vector. So that,

$$I_{stat} = \sqrt{(I_t \sin \theta)^2 + (I_t \cos \theta)^2} \quad (1)$$

$$I_{stat} = I_t * \sqrt{\sin^2 \theta + \cos^2 \theta}$$

Here, I_{stat} represent the final stator current. θ is electrical phase angle. I_t is rated maximum winding current, I_a and I_b are the current of stator winding A and B, respectively, so that,

$$I_a = I_t * \sin \theta \quad (2)$$

$$I_b = I_t * \cos \theta$$

In practice, current that in one winding is constant in half of full step and it changes as a function of $\sin \theta$, in other winding. Then, the final stator current is,

$$I_{stat} = \sqrt{I_t^2 + (I_t * \sin \theta)^2} = I_t \sqrt{1 + \sin^2 \theta} \quad (3)$$

This relation is depicted in Fig 2.

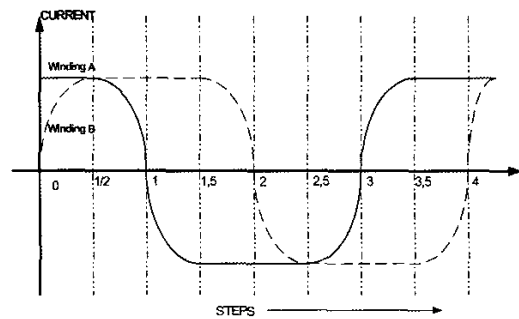


Figure 2. Stator current forms in practice.

Therefore, by dividing the 90 degrees of electrical phase angle to its smaller parts, in the currents applied to stator windings, step angle of rotation can be increased. Stepper motor driving modes are developed by this way. This situation is depicted in Figure 3.

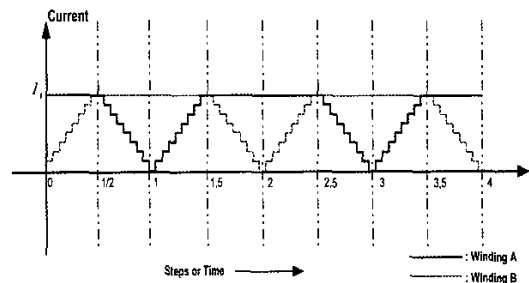


Fig 3. Waveforms of stator winding currents in micro-stepping mode [7]

Each time, stepper motors are driven by applied digital pulses to its stator windings. The step angle can be changed up to some small portion of full step, depending on the applied pulse modes. The rotor rotates and stops with one pulse up to generated step angle.

In the full-step and half-step driving modes, currents in stator windings are either exist or not. In the micro-step driving mode, stator winding currents are determined depending on some current amplitude functions as depicted in Figure 3.

Some advantages of microstepping driving are [8],

- Smooth movement at low speeds
- Increased step positioning resolution, as a result of a smaller step angle,
- Maximum torque at both low and high step-rates.

3. Emulations of Stepper Motor Drivers

The block diagram of the developed emulation system hardware is depicted in Figure 4. Communications between the simulator (PC side) and the stepper motor driver interface unit is realized by using RS232 serial interface port.

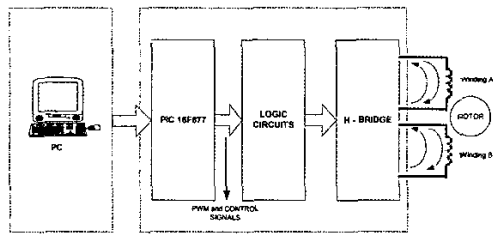


Figure 4. Block Diagram of Emulation System

Stepper motor interface unit receives the control parameters related to the selected stepping mode and create the convenient PWM outputs for logic circuits.

Because of the two PWM output capability and the low cost, the microcontroller based interface unit has been created around the PIC 16F877 microcontroller chip. [9]

The main specifications of PIC 16F877 microcontroller unit is as following:

- 20 Mhz clock input
- 8K Flash Memory
- 35 instruction
- RISC architecture
- Direct, indirect and relative addressing
- Power on Reset (POR)
- Power up Timer (PWT)
- Programable code protect
- 3 timer
- 10 Bit A/D convertor

Developed software consists of two main parts. The block diagram of software is shown in Figure 5.

First part of the software is implemented in the PIC 16F877 microcontroller program memory. This software has been created by using PIC assembly codes. It manages the operating requirements of stepper motor in different modes. It supports various rotation sequences such as, one full-step forward or backward, continuous full-step forward or backward, one half-step forward or backward, continuous half-step forward or backward, one micro-step forward or backward, continuous micro-step forward or backward .etc.

Second part of the software is developed by Delphi 7.0 programming language. It runs on the PC and has two main goal. One of the goals is connecting the PC and PIC 16F877 stepper motor interface unit using RS 232 port. By this way, the mode control information of the stepper motor is send to the motor interface unit. Assembly programming inside the delphi has been used for this purpose.

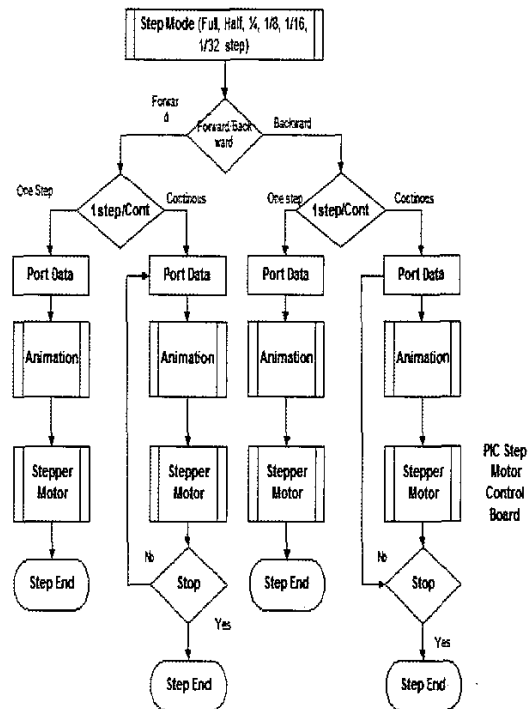


Figure 5. Software block diagram.

Second goal of the software is creating a user friendly graphic screen by which user easily gathers the information about stepper motor control process. For this purpose, some animations, graphics and help files are made in different fields of the screen. A view of graphical user interface screen is shown in Figure 6.

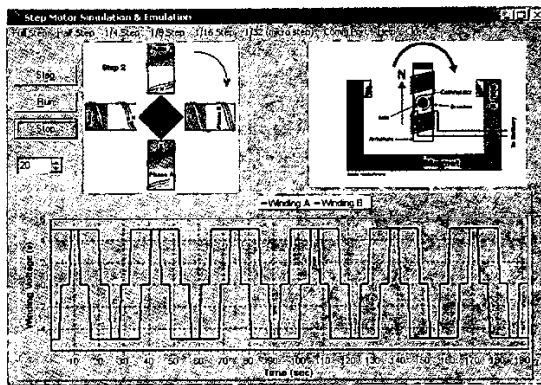


Figure 6. Graphical user interface

To start the program first, we must select the serial port. The program supports the ports Comm-1 and Comm-2. After that, we must select the stepping mode (Full step, Half step, 1/4 step, 1/8 step, 1/16 step, Micro step), stepping direction (Forward, Backward) and stepping type which are single step and continuous step (run mode). In the continuous mode, we must also select the step frequency.

In the single step mode, direction and mode information are transferred once to the microcontroller based unit whenever the step button is pressed. Then motor rotates one step. In the same time, A and B coil currents are displayed with two different animations on the graphical screen.

In the continuous mode, by pressing to run button, the direction, mode and frequency information are transferred to the microcontroller based unit. While the motor rotates in selected mode and frequency, A and B coil currents are continuously displayed on the graphical screen.

4. Conclusion:

Mechatronics is becoming a common elective in most undergraduate technological programs throughout the world. The response of the educational infrastructure to these changes has been richly diverse [4]. So it is needed hardly to develop many educational tools in future.

With this relation, an emulation system for stepper motor driving system has been developed for **mechatronic education** purposes, in this study. In our researchs we couldn't find similar study in literature. So we hope that, this is a new study related to emulation based education of stepper motor actuation systems. Successes in PC programming studies and microcontroller based system development studies has been encouraged us for planning similar studies on mechatronics education in future.

REFERENCES:

- [1]. Comerford R., (1994), 'Mech...what?' IEEE Spectrum, pp.46-49.
- [2]. Hori Y., at all, (1990), 'Theories and application of advanced motion control', Trans. IEE Jpn., vol 110-D, no. 11, pp. 1110-1118, 1990.
- [3]. Kyura N., (1996), 'The development of a controller for mechatronics equipment', IEEE Trans. Ind. Electron., vol. 43, no. 1, pp. 30-37.
- [4]. Ebert-Uphoff I., at all, (2000), 'Preparing for the Next Century: The State of Mechatronic Education', IEEE/ASME Trans. on Mechatronics, vol. 5, no. 2, pp. 226-227.
- [5]. Ibrahim E.T., at all, (1993), 'Integrating Computers and Industrial Hardware in Instrumentation and Control Education', ASEE Conference, Instrumentation Division, Urbana, Illinois, pp. 1425-1433.
- [6]. Ibrahim E.T., Smith R.F., (1993), 'New Approach to Undergraduate Instrumentation Education', Advances in Instrumentation and Control, Chicago, Illinois, pp. 437-445.
- [7]. Akdoğan E., Çelik H.H, (2003), "A Microstep Driver For Stepper Motor By Using PIC Microcontroller", Third Int. Advanced Technologies Symposium, pp 59-65.
- [8]. Microchip Technology Inc. (2002), "Stepper Motor Microstepping with 18C452", Microchip Application Notes-AN822.
- [9]. Microchip Technology Inc. (1999), "PIC 16F87X 28/40 pin 8-bit CMOS FLASH Microcontroller Data Sheet", DS30292B.