CARS & FOF 2006
VIT, Vellore, INDIA

A SERIAL PORT INTERFACE CIRCUIT FOR ROBOTIC CONTROL APPLICATIONS INVOLVING MULTIPLE STEPPER MOTORS

Arun Dayal Udai*
Dr. A. N. Sinha**

** Junior Research Fellow, e-mail: arun_udai@yahoo.com*
***Professor, e-mail: an_sinha57@yahoo.com*
Department of Mechanical Engineering,
Birla Institute of Technology, Mesra, Ranchi – 835215.

ABSTRACT

A serial port interface circuit has been designed and tested for programmable control of multiple permanent magnet unipolar stepper motors to be used in a biped robot. In this work, a stepper motor control system of a biped robot having 10 degree of freedom, 5 on each leg is proposed, that will be built in-house in a economical way. The circuit consists of a microcontroller, 32K serial EEPROM, serial-in parallel-out shift register, Darlington array stepper motor drivers and power supply. A MATLAB® program is developed for calculating the motors driving sequence generation and a PBASIC microcontroller program is developed for downloading the driving sequence from the computer and driving the motors.

1. INTRODUCTION

A number of robotic control applications require precise position control, where inherent precision of stepper motors have proved itself to be very beneficial. A large number of robots used in manufacturing and automation processes use stepper motors. Various robots like robotic arm, automated material handling robot, welding robots, camera positioning systems in robots, various parts of humanoid robot etc. are examples involving some of the typical application of precise positioning and control. Power consumption, cost, reliability, high performance and various technical requirements like precision are major areas which are still to be explored in such robotic control systems involving microcontroller. A costly custom built DC Servo motors for specialized purpose is desirable and being used by the affluent country like Japan for its humanoids (ASIMO etc.). However, it is not that much attractive in the developing country where economy is the prime concern, that too where custom built motors are required. Thus, cost cutting becomes the one of the main factor in robot experimentation, design and application. In such a scenario a stepper motor with flexible circuit with discrete motion planning may prove to be very helpful.

Stepper motors are preferred here because of its high torque per step movement compared to the similar sized and servo motors if moved to per step size of stepper motor. Although, stepper motor has limitation of step resolution, it can be overcome by adding mechanical gearbox or by microstepping [11]. The stepper motor is easy to drive in open-loop, mechanically simple, and responds to a digital input which makes it possible to control directly by computer without any encoders. One of the unique characteristic of stepper motor is its holding torque [10], without any additional circuitry. Holding Torque is the minimum torque required to turn the motor from stand still position when energized. A DC servo motors are expensive as compared to stepper motors and there is additional cost for the complicated closed loop control system, which makes stepper motors very friendly for small sized projects with limited cost. Methods like microstepping [11], high speed driving [10] etc. outreach the limitations of stepper motor. A better comparison has been detailed in reference [7].

Several attempts have been made which involves stepper motors control using computer. Mike J. Johnson [6] uses parallel port for stepper motor control; the proposed scheme controls only 5 motors. Parallel port control application has limitations on number of motors that can be controlled along with the communication speed at which it should be controlled. In the proposed scheme the complete motion sequence can be downloaded or embedded to the EEPROM of the circuit or can be run directly through computer for any random motion. Once the sequence is downloaded the circuit may be detached from the computer and the sequence can be regenerated in cycle when stand alone since here in this work it is used for walking sequence of the biped robot.

A segmental motion control proposed by Mittal D. P. [4] with open loop control is used. A recently identified problem of speed reversals of stepper motors in open loop systems due to the growth of undershoot in the response to repeated steps [2]. As the torque seldom increases the pre-calculated torque, the chances of stepper motor loosing its position are very less.

In this paper, a serial port interface circuit for controlling multiple stepper motors has been presented which can be used in conjunction with Segmental Motion Control [4]. The circuit is very flexible with digital logic that can be easily modified for any special needs with similar layouts and logic.

2. CIRCUIT DESIGN AND IMPLEMENTATION

A Serial port interface circuit is designed with serial connector shown in Figure 1. Once the sequence of motor steps is calculated, it is sent via serial communication from any of the PC's COM# Port (RS232) through MATLAB Program. This data is decoded by the microcontroller and it sends the pulses to the corresponding motor required for its forward or reverse driving. The first digit of the serial data contains the motor direction and the remaining digits form the motor number to be driven. E.g. a serial data 91 will be decoded as 9^{th} motor in forward direction and similarly 90 would mean the 9^{th} motor in reverse direction. As long as the motor is not moved by the microcontroller program remains powered with the last phase status of the motor coils. This holds the motor in its current position. With the current layout eight motors can be controlled which can be very easily extended for sixteen motors by cascading of shift registers explained later.

2.1 Microcontroller

The microcontroller used is BASIC Stamp 2px24 which is used because of its simplicity in its programming language i.e. PBASIC. It serves the purpose of synchronous serial communication between microcontroller and the shift register 74HC595. It also has a dedicated serial communication port for interfacing with computer which can take in asynchronous serial data from computer by a single SERIN command at 19200 bauds. Only three pins of the microcontroller are engaged in asynchronous serial communication. Two of the I/O pins is engaged for serial communication with the external EEPROM. Thus, 16 I/O pins of the microcontroller can serve for 4 non cascaded shift registers, which in turn can drive 8 stepper motors. A multiple number of shift registers can be cascaded together if more number of motors is required to be driven. Remaining two pins can be used for feedback purposes if required. The microcontroller chip is smaller in size which is best suited for compact applications like robot.

2.2 4K x 8 (32K bit) Serial Electrically Erasable PROM

This component is an addition to the circuit discussed by Ranjan R., et. al. [8]. It makes the circuit capable of working offline without the computer attached to the working robot. The 24LC32A is a 4K x 8 (32K bit) Serial Electrically Erasable PROM capable of operation across a broad voltage range (2.5V to 6.0V). It has been developed for advanced, low power applications such as personal communications or data acquisition. The 24LC32A also has a page-write capability of up to 32 bytes of data. The 24LC32A is capable of both random and sequential reads up to the 32K boundary. Functional address lines allow up to eight 24LC32A devices on the same bus, for up to 256K bits address space. It is used here for storing walking sequence data table which is generated through MATLAB Program. The program reads the data and then sends to the steppers during walking.

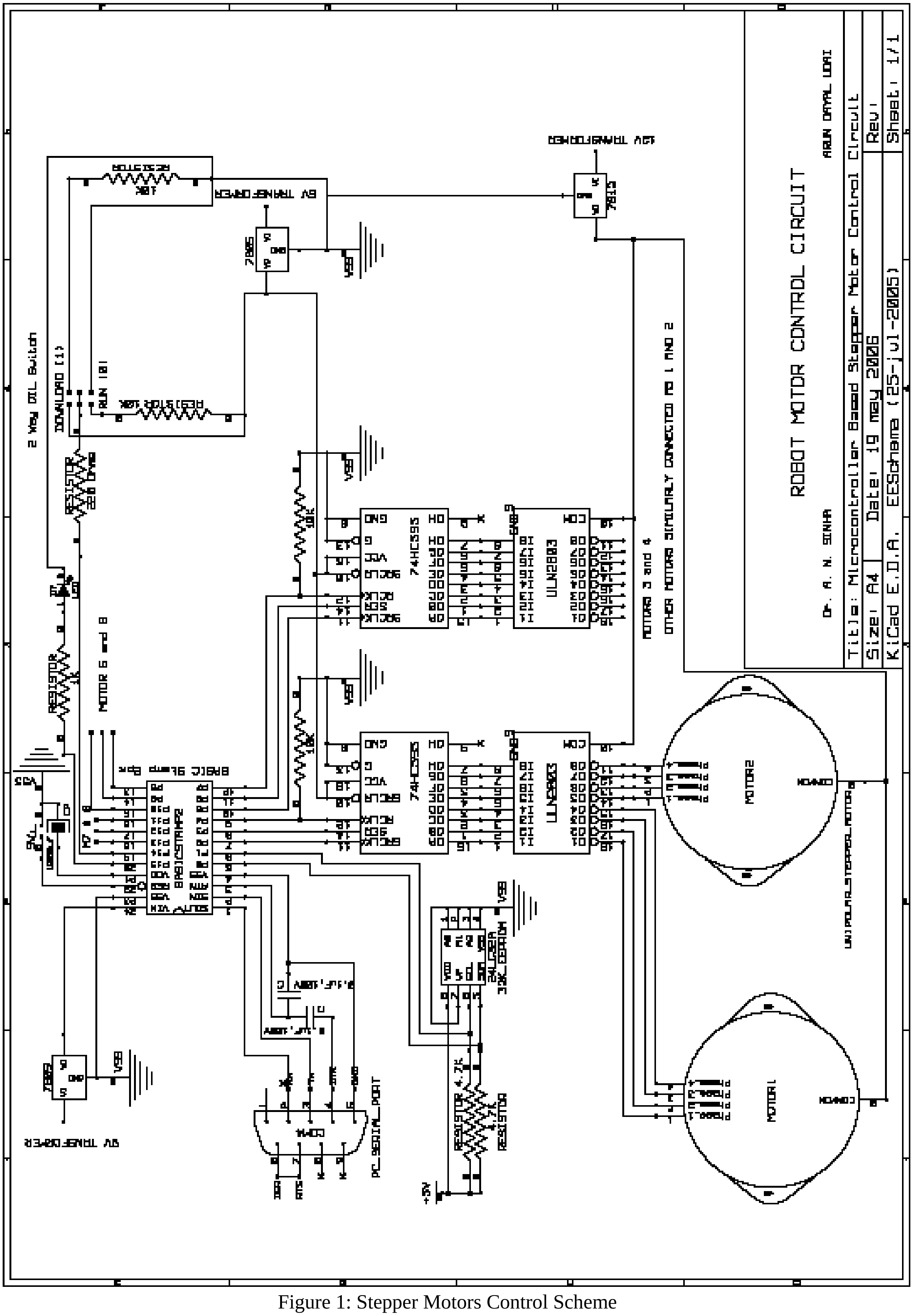

Figure 1: Stepper Motors Control Scheme

2.3 16 pin Serial-in parallel-out Shift Register with Latch

This IC provides the 4 bit sequential pulse data required for driving each stepper motor. Having an 8 – bit output pins it can drive two unipolar stepper motors at a time. Pin 11 receives the clock pulses and pin 14 receives the series data given by the microcontroller by a single SHIFTOUT command to any microcontroller pin. Pin 12 receives the Latch signal given by the single PULSEOUT command of the microcontroller, which enables output pins to collect the finally prepared signal at the registers.

The IC is implemented to expand the output pins of the microcontroller. The chip uses only 3 data pins to drive 8 output pins when used in non cascaded form. Two 74HC595 IC's can be cascaded together to control 8 ×2 = 16 output pins, with same 3 input lines. Thus cascading is used to drive 4 motor with only 3 microcontroller pins.

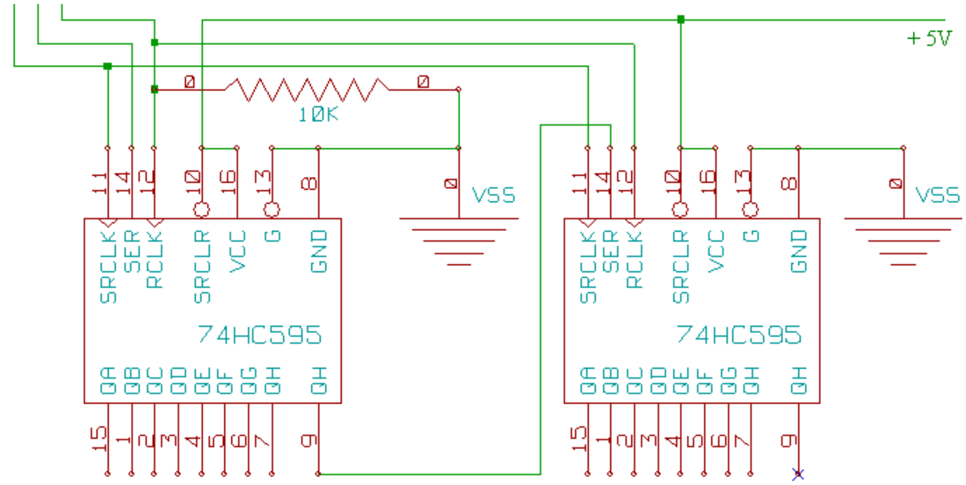

Figure 2: Cascading of two Shift Registers

2.4 Darlington Array Stepper Motor Drivers

The ULN2803A/ULN2003 contains eight/seven darlington transistors with common emitters and integral suppression diodes for inductive loads. Each darlington features a peak load current rating of 600mA (500mA continuous) and can withstand at least 50V in the off state. Outputs may be paralleled for higher current capability for driving higher torque motors.

It is used to convert the TTL logic 5V signal levels to 12V driving signals for driving motor actuators.

2.5 Unipolar Stepper Motor

Permanent Magnet Unipolar stepping motors (MITSUMI M42SP-5) is selected because of its compact size, low power consumption, high torque 26.5 mN·m/200pps and precise step angle of 7.5^{o}/step and a maximum of 365 pulses per second. The driving of unipolar motor is simpler as compared to bipolar motor which requires H-Bridge for its driving. Unipolar motor also supports precise half step or micro step driving. If a motion of steps lesser than half steps i.e. 3.75^{o} is required additional reduction gear heads can be attached which will also increase the driving torque of the shaft. Full step and Half step driving of stepper motors is detailed in reference [1, 4 and 11].

Voltage applied to the stepper motor remains constant, and the speed can be varied with the number of pulses per second applied through ULN2803. However, the torque of the motor torque varies inversely as per the speed which is shown in the graph below for MITSUMI M42SP-5 motor.

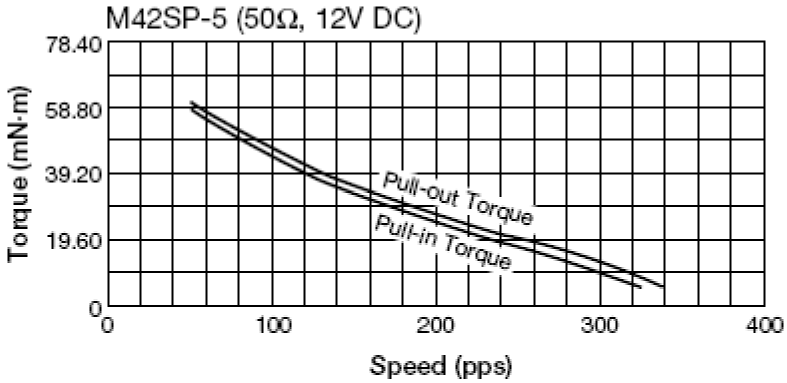

Figure 3: A Typical Speed – Torque characteristics of a stepper motor [12].

Based on the speed-torque characteristic of the stepper motor selected, maximum and minimum speed for the motion is to be decided for the working range of the trajectory. For any specific motor, this can be obtained from the datasheet. The variation of the pulse can be made very easily as this is a digital process achieved through microcontroller. This is an added advantage over simple DC motor where the torque characteristic is based on the voltage applied to its terminals that requires digital to analog converter for speed control.

3. SOFTWARE IMPLEMENTATION

The software is implemented in two parts. MATLAB Program is used to calculate the joint angles of each motor, based on the trajectory in consideration. Inverse kinematics is used to know the joint angles at each point on the micro steps of trajectory. The cumulative angles are obtained at each step for each motor to be turned successively from one state to other while moving. This will enable to find change in joint angle along the micro step. This data can be coded in any suitable format and sent to the microcontroller which receives it and stores it in the external EEPROM in download mode. As the PC COM Port can send the maximum numeric value of 255 i.e. FF in Hexadecimal or one byte, the maximum number of steps are limited to 255. A typical coding of motor data can be sent in two bytes of information. First byte consists of motor number and the direction, and second byte says the number of steps to turn in that direction. For example, 91 means motor number 9 to be rotated in clockwise direction. Similarly, the value 80 is to be decoded as motor number 8 and direction anticlockwise. The motor data is stored in EEPROM in groups of two bytes of information at two consecutive locations.

The microcontroller program consists of two parts. One is the download code which receives the motor data through PC and stores in its EEPROM. The second part of the microcontroller code reads the data from the EEPROM and sends the pulses to the required motor in decoded direction. There is a mode change switch implemented in the circuit PIN 14 of the microcontroller (refer Figure 1) which decides the download or working code to be functional based on the position of the switch. The indicator pin PIN 15 of the microcontroller fitted with a Light Emitting Diode (LED) is used to indicate the download completion while data transfer and cycle completion in running mode. In the running mode, if the switch position is changed to download mode, it acts to halt the motors.

4. RESULTS AND DISCUSSION

Figure 4 shows a spline path for the biped ankle movement with the thigh and shank length of 110 cm each. The selection of stepper motor is based on the maximum torque required to traverse the path calculated. The stepper motor torque should be sufficiently large to avoid any undershoot and thereby any position loss. Figure 5 shows the corresponding angular movement of stepper motor actuator at hip, knee and ankle.

The microcontroller has sufficient command execution speed to drive 16 motors at a time with its maximum pulse rate. Further enhancement of the circuit can be to run the multiple motors in micro steps as detailed in [11]. If more number of motors are used the speed may drop significantly in such situation a method discussed in reference [10] may be used. If feedback is required to the computer, MAX232 chip can be used for communicating with Microcontroller as used in [8], but this consumes two additional pins of the microcontroller I/O pins. It acts as a level shifter and converts the RS-232 level signals from our PC to the TTL levels required by the BASIC Stamp microcontroller. The entire circuit has been found to attain the required precise motion satisfactorily.

A sample motor actuator for joint angle variation corresponding to a segmented Spline path is shown in Figure 5. This is the stable path which is calculated using Zero Moment Point and Normal Projection of Centre of Mass [9] into consideration for biped balancing is taken. The path is ramped because of segmented motion attained by stepper motors. This path can be optimized with balancing constraints for a biped robot based on ZMP.

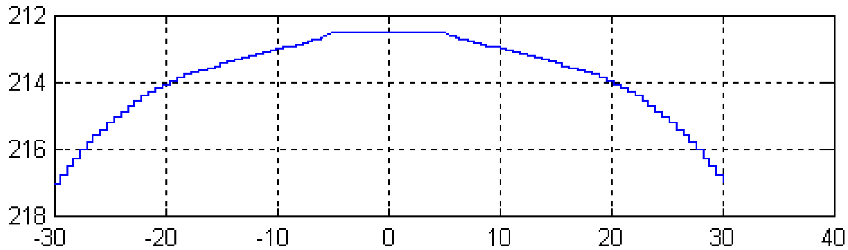

Figure 4: A Ramped Spline Path for the Biped ankle movement with thigh and shank length 110cm each.

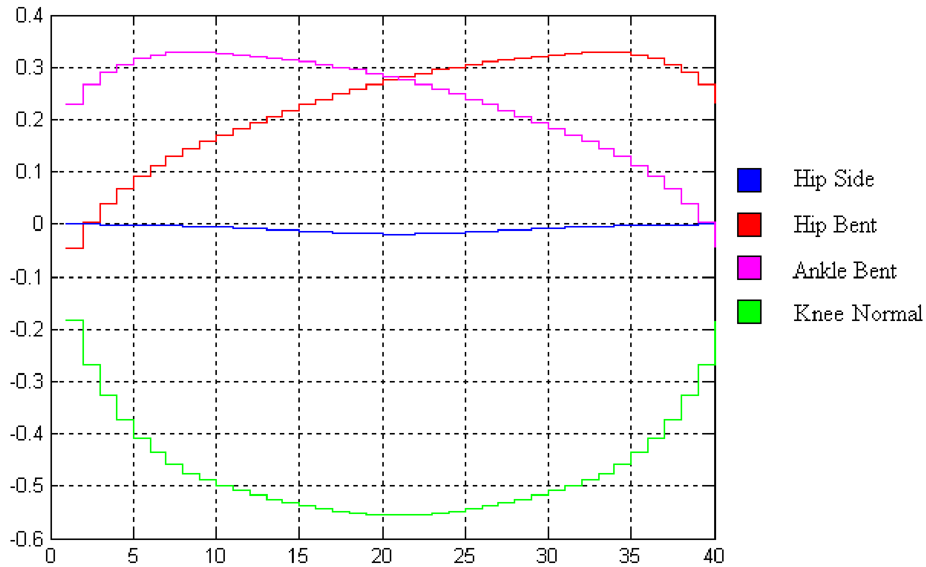

Figure 5: Stepper motor angles corresponding to spline path trajectories.

5. CONCLUSION

The circuit presented here is suitable for low cost applications and experimentation with rapid and flexible design requirements. The motion of the robotic manipulator with this arrangement is precise. The position of the manipulator or biped legs is very stable when loaded even when it is not moving. However, for higher resolution gear reduction is required to be coupled with stepper motors. As it is an open loop system the supply to the motors should not be switched off while it is operating the motors. This will result in loss of position, and the motors are required to be reset to initial position from where the motor sequence is to be started again. The gear should have minimum backlash for positional accuracy. The gear box may contain clutch arrangement which enables easy resetting to the initial position.

6. REFERENCES

1. Ackdogan E., Topuz V. and Akbas, A. “An Educational Tool on Mechatronics: Emulation of Stepper Motor Driving Systems by Using a Microcontroller Based System Interface”, IEEE 2004.
2. Bodson Marc, Jeffrey S. Sato and Stephen R. Silver, “A Spontaneous Speed Reversals in Stepper Motor” IEEE Transactions on Control Systems Technology, Vol. 14, No. 2, March 2006.
3. Douglas W. Jones, “Stepping Motors”, web book at http://www.cs.uiowa.edu/~jones/step/index.html.
4. Mittal D. P. and Chin T. C., “A 2 – Axis Stepper Motor Motion Control System”, IECON’ 91, pp: 397 – 401.
5. Malvino Albert Paul and Leach Donald, “Digital Principles and Applications”, Fourth Edition, Tata McGraw Hill, 1991.
6. Mike J. Jhonson and Guru Subramanyam, “A Parallel Port Interface Circuit for Computer Control Applications Involving Multiple Stepper Motors”, IEEE, 1997, pp: 889 – 892.
7. Mort N., Abbod, M. F. and Linkens D. A., “Comparative Study of Fuzzy DC Servo Motors and Stepper Motors For Mechatronic Systems”, IEEE, 1995.
8. Ranjan R., Sinha A. N. and Udai A D, “Kinematics and Control of Biped Robot”, proceedings of All India Seminar on Advances in Product Development”, February 2006.
9. Udai A. D, Sinha A. N., “Trajectory Planning of a Biped Robot”, proceedings of All India Seminar on Robotics and Futuristic Factories, December 2005.
10. Vadell J., Lira I. and Dixon J., “CED - A New Method for High Speed Stepping Motor Driving”, IEEE 1997, pp: 539 – 545.
11. Yedamale P, Chattopadhaya S., “Stepper Motor Microstepping with PIC 18C452”, Microchip Technology Inc. 2002.
12. http://www.mitsumi.com, Datasheet “MITSUMI” Stepper Motor M42SP-5.