Positioning system with stepper motor

Crisan Vasile-Rares[*], Todor Daniel

Technical University of Cluj-Napoca, Romania

This paper will present the types of stepper motors, principles of choice, calculus for verifications in positioning systems, constructive solutions of positioning systems and the realizations of authors concerning positioning systems controlled from a personal computer and command from distance using the micro controller and infra-red remote control.

NOTATION

Mpp	stepper motor
M_r^r	reduced resistant momentum
ω	angular velocity
F_j	force applied of the 'j' element
V_j	velocity of the 'j' element
a_j	angle between force and direction
M_j	friction momentum
w_j	angular velocity of the j element
I^r	reduced momentum of inertia
m_j	the mass of the j element
V_{Gj}	velocity of the center mass of 'j' element
I_{Gj}	momentum of inertia for the 'j' element
f	frequency
f'	frecquency after corrections
I_m	inertia momentum of the motor

GENERAL ASPECTS

Positioning systems actuated with stepper motors cover a large area of applications as the primary energy used is non polluting electrical energy which also assures high precision in the positioning process even in an open loop command. Other advantages can be given by the fact that the motor torque is large versus geometrical dimensions.

The stepper motor is an electromagnetic converter of electrical energy materialized in a train of pulses and applied on phase coils as mechanical energy manifesting discrete movements of the motor axis. For correct functioning, every pulse corresponds with a discrete movement of motor shaft - the step. The frequency of pulses gives the velocity of revolute movement of the motor axis.

In the present, sorting, dosing and positioning mechanisms actuated with stepper motors present a large area of implementation in industry, because better control of position can be obtained than in pneumatic systems through conditions of superior ergonomics. In later years, due to these conditions, manufacturers have developed special controllers for stepper motors.

In direct continuos motors the general purpose is to maintain constant revolute movement or change in relative wide limits, whilst in systems with stepper motors the main problem is to control the position process.

An essential problem in projecting positioning systems with stepper motors is the choice and verification of the motor for associated mechanical structure and deriving utile tasks from the process.

The basics principle for choosing stepper motors

In choosing stepper motors we have taken into consideration the following aspects:

Type of motor, from the point of view of constructive principles there exist 6 type of stepper motors [4]: solenoid's, electromechanical, with permanent magnet in stator, with permanent magnet in rotor, with variable reluctance and hybrid types. The most used types are with permanent magnet, variable reluctance and hybrid, because of robust construction, durability and precision of stepping.

Functioning characteristic, is the characteristic is traced experimental by the manufacturer at no mechanical loads. This diagram expresses the dependence of the frequency at applied pulses and the torque.

Maximum command frequency, is the parameter which practically influences the maximum speed of positioning.

Type of command, it can be possible to make a classification of the possibilities of command of stepper motors [1]: after the timing of alimentation with electrical energy; the command can be potential or by pulses, after the polarity of phases; it can be unipolar or bipolar and after the number of phases fed simultaneously; symmetric or asymmetric.

The amount of step, usually the values are: 0.72, 1.8, 3.6, 7.2, 45, and 90. For positioning systems we recommend motors with small values for the angle step in order to obtain small increments.

Geometrical characteristics, these parameters are restrictive in some applications, for instance in mini systems.

Verification calculus

For the motor chosen, if the mechanical system is known, it is necessary to see if the motor will function correctly and normally. Verification consists of the calculus of effective frequency and reduced resistant torque at the motor axis. So, usually the method is as follows:

1. Calculus of utile forces and torque's
2. Calculus of reduced resistant torque

$$M_r^{\,r} \cdot \omega = \sum_{j=1}^{n} \left(F_j \cdot V_j \cos \alpha_j + M_j \cdot \omega_j\right) \qquad [1]$$

   It took into consideration all resistances such as: in this formulae the value related to forces is entered which appears in linear guided movement and linear velocity; this formulae took into consideration the friction forces, the velocity, the friction momentum and the angular velocity of the kinematical elements.
3. For the velocity known/given, we can estimate the revolution velocity needed at the motor axis
4. Calculus of frequency of pulses
5. Inertia reduced momentum

$$\frac{1}{2} I_r \cdot \omega^2 = \sum_{j=1}^{n} \left(m_j \frac{V_{Gj}^{\,2}}{2} + I_{Gj} \frac{\omega_j^{\,2}}{2} \right) \qquad [2]$$

   It took into consideration the masses in movement and the velocity of the kinematical elements
6. The frequency must be a reestimate in order to obtain accurate results

$$f^{'} = f \cdot \sqrt{\frac{I_m}{I^r}} \qquad [3]$$

7. We check if the point of coordinates A ($M_r^{\,r}$, f') is below the functioning characteristics drawn by the producer. In the affirmative case we consider that motor choice was good.

Constructive solutions

There exists a large diversity of constructive solutions for positioning systems actuated with steppers motors. A formal classification of positioning systems can be made after the coordinates of positioning. That is angular systems and linear systems.

Usually, in order to obtain a greater torque between the motor and the transmission a gear reducer constructed usually with tooth wheels is used. For transforming revolution movements into linear movements a transmission with flexible elements, screw movements, and pinion - crew is utilized [3].

REALIZATONS OF THE AUTHORS

One axis positioning system with command from the computer

We designed the experimental stand, figure 1, to study the positioning process in an open loop command. The mechanical structure is very simple, the revolute movement is transformed into linear movement by the transmission with a flexible element.

The motor tooth wheel is fixed rigid on the motor axis. There is a rigidified element on the belt which makes linear movement guided by the stalk. Because in many cases the pretension of belt is needed in order to prevent sliding on the wheel we used a excentric mechanism.

Figure 1. Positioning system SIP1x

The power-electronics interface is presented in figure 2

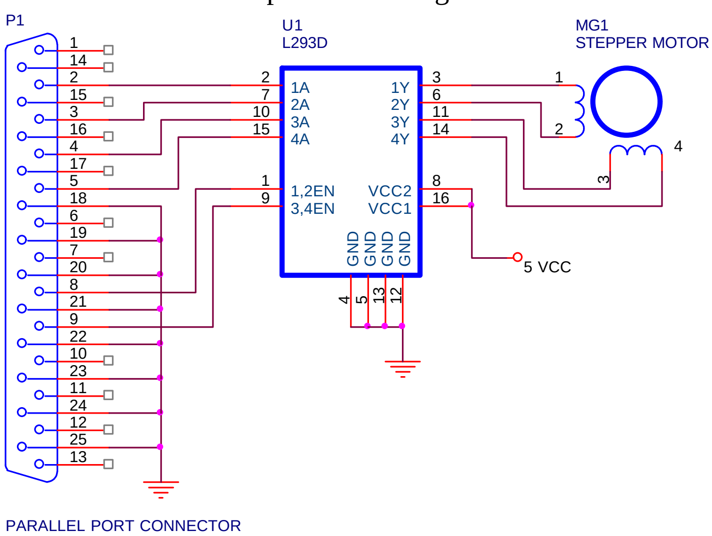

Figure 2. Electronic circuit for stepper motor command

We used a bipolar stepper motor from Shinano Kenshi Co., therefore we needed two H bridges in order to be able to reverse the polarity on the motor coils. The L293D circuit provided by ST contained two protected H bridge cells with separate commands, internally protected by fly-back voltages.

The program was written in C standard language. We used the parallel port of a personal computer to generate the command signals.

```
//-------------------------------------------
// Sip 1X
// Made by:Crisan Vasile-Rares 2001
// cvr@email.ro
// -------------------------------------------

        #include<stdio.h>
        #include<conio.h>
        #include<dos.h>
        #include<math.h>

int port=0x378;
unsigned char bit[4] =
        {0x41,0x84,0x42,0x88}

  /* ----------------------
        | D7 | 0 | 1 | 0 | 1 |
        | D6 | 1 | 0 | 1 | 0 |
        | D5 | 0 | 0 | 0 | 0 |
        | D4 | 0 | 0 | 0 | 0 |
        | D3 | 0 | 0 | 0 | 1 |
        | D2 | 0 | 1 | 0 | 0 |
        | D1 | 0 | 0 | 1 | 0 |
        | D0 | 1 | 0 | 0 | 0 |
        ----------------------
        |data|41h|84h|42h|88h|
         ---------------> +
        <---------------  - */

void step_fwd(void);
void step_bwd(void);
void clear_port(void);

 int rst;
 int index=0;
 int f;  // frequency
 int T;  // period
 float coord;
 float no_steps;
 float increment=0.397;
 float er_poz;

 main()
{     clear_port();
  jmp1:
        clrscr();
        #include "c:\sip1x\antet.h"
        printf("\n\n   \xAF Give coordinate between
(0,220mm] ");
        scanf("%f",&coord);
        if(coord>220||coord<=0)
          {
           printf("\n\t\t ERROR!!!");
           delay(1000);
           goto jmp1;
          }
  jmp2:
        printf("   \xAF Give frecquency of pulses between
[10Hz;250Hz] ");
        scanf("%d",&f);
        if(f<10||f>250)
          {
           printf("\n\t\t ERROR!!!\n");
           delay(1000);
clrscr();
#include"c:\borlandc\stepper\antet.h"printf("\n\n");
   goto jmp2;
          }
        clrscr();
#include"c:\borlandc\stepper\antet.h"
T=1000/f;  // in ms
printf("\n\n   \xAF Coord. is %g mm ",coord);
printf("\n   \xAF Frequency is %d Hz",f);
printf("\n   \xAF Period is %d ms",T);
no_steps=coord/increment;
float j,k;
k=no_steps;
j=abs(no_steps);
printf("\n\n\t\t\t\t\t  \xAE Number of steps:  %g",j);
k=k-j;
er_poz=k*increment;
printf("\n\t\t\t\t\t  \xAE Positioning error is: %g mm
",er_poz);
        int i;
        for(i=1;i<=j;i++)
           {
           step_fwd();
           }
        printf("\n\n\n\t\t........ For moving back press any
key");
        getch();
        for(i=1;i<j;i++)
           {
           step_bwd();
           }
        outp(0x378,0x0);
        printf("\n\n\n\n\n\n\t   Another   positioning   ?
<Yes=1/No=2> ");
        scanf("%d",&rst);
        if(rst==1)
           {
           goto jmp1;
           }
        else {
                printf("\n\t  Press  any  key  for  end  of
program!");
                getch();
             }
   return(0);
 }
   void step_fwd()
     {
        if(++index==4)
        index=0;
```

```
        outp(port,bit[index]);
        delay(T);
  }

void step_bwd()
  {
        if(--index<0)
        index=3;
        outp(port,bit[index]);
        delay(T);
  }

void clear_port()
 {
        outp(port,0x0);
        delay(T);
 }
```

```
// antet.h
{
int w;
printf("\xC9");
for(w=1;w<79;w++)
        {
                printf("\xCD");
        }
printf("\xBB");
printf("\xBA            SISTEM   INCREMENTAL   DE
POZITIONARE   SIP1x   ver1.0          \xBA");
printf("\xC8");
for(w=1;w<79;w++)
        {
                printf("\xCD");
        }
printf("\xBC");
 }
```

The program allows positioning in prescribed limits, after that by pressing any key, the elements move backward to the original position, generated by software.

Control by distance

We proposed to construct an autonomous minirobot with control via distance by infrared communication.

The mechanical structure consists of a mobile platform on wheels, as it can observed in figure 3..The motrics wheels, two by number, are fixed on the axes of the motors. The stepper motor is from the permanent magnet family, therefore permitting better control of position.

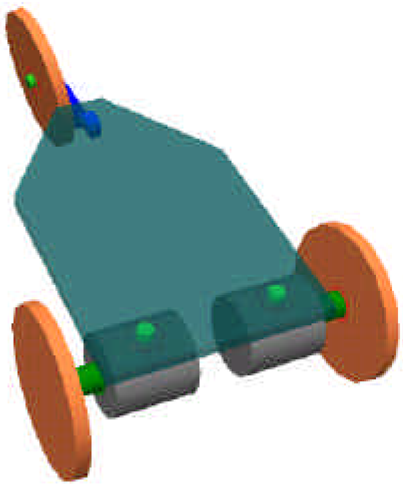

Figure 3. Mechanical structure of platform on wheels

The electronic schematics is very simple and easy to implement, presented in figure 3, implement the interface between robot and user. The command signals are generated from distance by a remote control. The coded signals are received and demodulated by special circuitry (SFH5110) provided by OSRAM, which generate digital code standard known as RC5. This signal is read over one pin of the microcontroller (AVR AT90S2313). The software was written in BASCOM AVR (an extension of basic language) and on the basis of the signal received, the program generates the logic signals for the command of steppers.

The program for the command of the motors, through the infra red remote control, simplified, is listed below:

```
$crystal = 4000000
Config Portb = Output
Config Rc5 = Pind.2
Config Pind.5 = Output
Enable Interrupts
Goto Start0
Start0:
Dim A As Byte
Dim C As Byte
Getrc5(a , C)'receiving IR-RC5
code
  If A = 0 Then
        C = C And &B10111111
        Select Case C
         Case 1 : Goto M_fwd
         Case 2 : Goto Left_curve
         Case 3 : Goto Right_curve
        End Select
  End If
M_fwd:
        Set Portb.3
        Set Portb.2
        Waitms 500
        Reset Portb.3
        Reset Portb.2
        Set Portb.5
        Set Portb.0
        Waitms 500
        Reset Portb.5
        Reset Portb.0
        Set Portb.4
        Set Portb.1
        Waitms 500
        Reset Portb.4
        Reset Portb.1
        Set Portb.6
        Set Portd.5
        Waitms 500
        Reset Portb.6
        Reset Portd.5
        Goto Start0
Left_curve:
        Set Portb.2
        Waitms 500
        Reset Portb.2
        Set Portb.0
        Waitms 500
        Reset Portb.0
        Set Portb.1
        Waitms 500
        Reset Portb.1
        Set Portd.5
        Waitms 500
        Reset Portd.5
        Goto Start0
Right_curve:
        Set Portb.3
        Waitms 500
        Reset Portb.3
        Set Portb.5
        Waitms 500
        Reset Portb.0
        Set Portb.4
        Waitms 500
        Reset Portb.4
        Set Portb.6
        Waitms 500
        Reset Portb.6
        Goto Start0
```

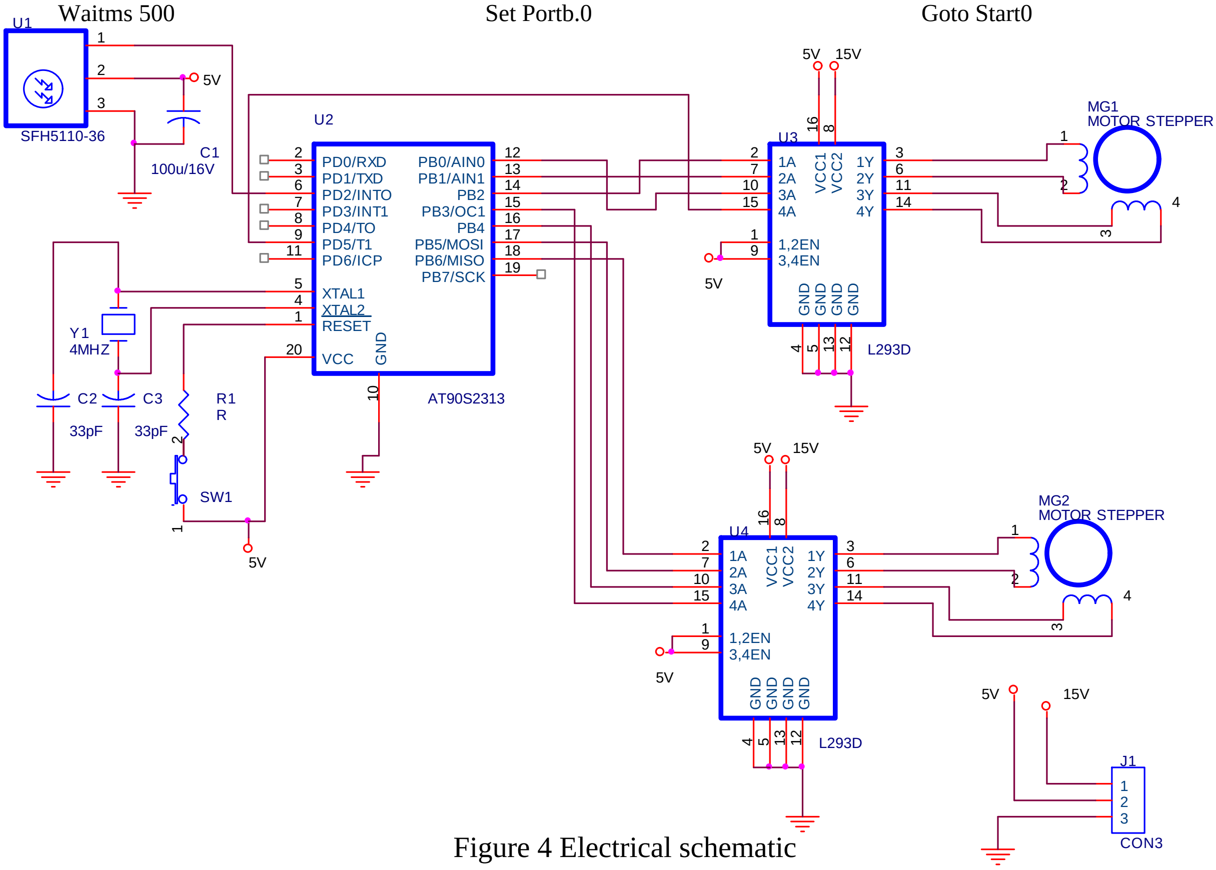

Figure 4 Electrical schematic

BIBLIOGRAPHY

1 Bogdanov I, The microprocessor in command of electrical actuations, Timisoara, Ed. Facla, 1989

2 Costea D, Tutorial for C language, Romania, Teora, 1996

3 Demian T, Constructivc elements of precision mechanics, Romania, Ed. Didactica si pedagogica, 1980

4 Kuo B. Kelemen A., Systems for command and incremental adjusting of position, Romania, Ed. Tehnica 1981

6 http:\\www.atmel.com, the official site of Atmel

7 http:\\www.shinano.com, the official site of Shinano Kenshi