

LINEAR AXIS POSITIONING WITH INDUSTRIAL SERVO MOTOR

Xénia Erzsébet TÓTH¹, János TÓTH PhD²

email:¹tothxeniaerzs@gmail.com; ²tothjanos@eng.unideb.hu
 University of Debrecen Faculty of Engineering

Abstract— Processions controller devices and positioning tasks with different conditions are quite important in industrial automated systems. The aim of this paper was to materialize a system which is an industrial application, with using industrial devices. The main part of his teachable system is a PLC controlled servo motor. With it the user can set different positions on the linear axis, with different velocities; moreover it gives the possibility to define new places, and to make a program to control a whole sequential process. The user can control this system with an industrial display (HMI). Its interface is user-friendly with easy manageability; therefore it can be used without programming knowledge.

Keywords— servo motor, positioning, PLC, HMI, programming

I. INTRODUCTION

THE practical applications of the servo motors is very important in the industry. In this paper we use it with a tooth belt linear axis for positioning. It can be made in different ways, such as

- controlling with a PLC,
- own table which is good for repeatedly running,
- using a software on the computer.

On the other hand these options can be connected with an industrial display too.

II. GENERAL INFORMATION ABOUT THE SERVO MOTOR

Positioning means moving a mass from a point to another. For this we use a FESTO EMMS-AS-55-S-TM servo motor. The brushless servo motor contains a three phase stator and a permanent-magnet rotor. The rotational speed of the rotor depends on the frequency of the rotational magnetic field through the stator windings. The rotor moves synchronic with the magnetic field of the stator, and follows it with the same revolutions. (Fig. 1.)

$$n = \frac{2f}{p}$$

With this formula, we can see that the speed of the motor (n) is proportional with the stator frequency (f) and in inverse ration to the number of the poles (p).

The name of the used motor means that this three phased brushless servo motor with the flange size 55 mm is equipped with an inductive multi-turn encoder mounted on the motor shaft. This feeds information

goes back from the motor to the controller. It means that we use a closed-loop system (Fig. 2.).

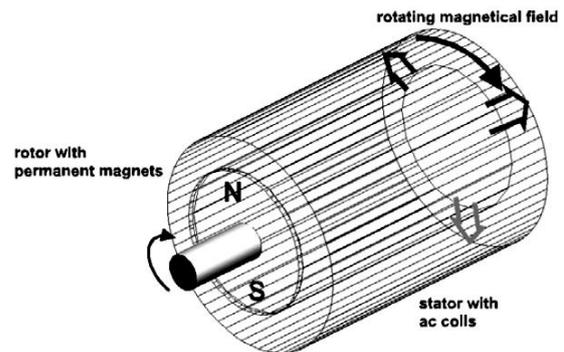


Fig. 1. Structure of a servo motor [3]

The user gives the instructions with the HMI, and it sends them to the PLC. After processing this information the PLC sends them to the motor controller which consists of a programmable controller and an amplifier. The commands about the direction, the distance, the speed and the acceleration forward to the actuator (motor). The sensor measures the real values and sends back the actual information to the controller. Based on this feedback the controller after the comparing decides how much correction is necessary. It continues repeatedly until the instruction is completed, to maintain the motor at the specified position and state. The data of the completed motion goes back to the PLC, and base on this information it gives the next instruction. The more important information for the user can be seen on the display.

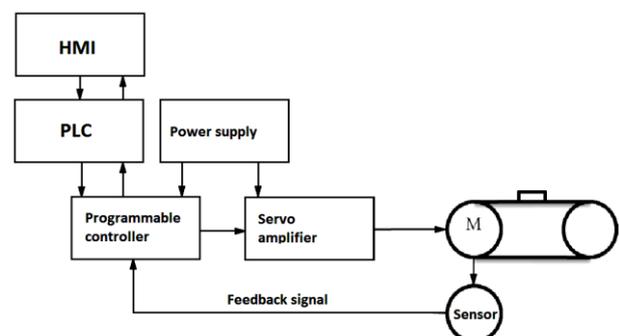


Fig. 2. The used closed-loop system

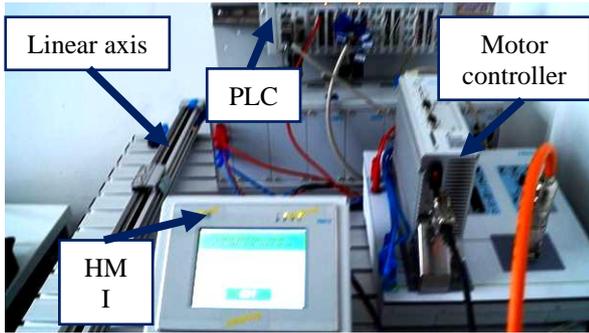


Fig. 3. The used system

III. USED MOTOR AND ITS CONTROLLER

We use a FESTO EGC-50-600-TB-KF-GK tooth belt linear axis, with 50 mm axis size, and 600 mm working stroke. Without it the data of the motor are the following:

TABLE I
 GENERAL TECHNICAL DATA OF THE SERVO MOTOR

Nominal torque	0,68 Nm
Nominal speed	6600 rpm
Nominal voltage	360 V
Nominal current	1,2 A
Torque at standstill	0,98 Nm
Maximum speed	7330 rpm
Peak torque	2,7 Nm
Maximum current	5 A

With using the linear tooth belt axis, it limits the range. In this case the maximum speed of the motor is 3000 mm/s, and the maximum acceleration can be 50 mm/s².

The used motor controller is a FESTO CMMS-AS-C4-3A-G2 which means the motor technology is AC synchronous, the nominal current of the motor is 4 A, and the nominal voltage is 230V AC/50...60 Hz. The maximal current is 10 A, and the signal speed through the RS232 port is 9600-115000 bits/s. In Table II the data of the motor controller can be seen.

TABLE II
 GENERAL TECHNICAL DATA OF THE MOTOR CONTROLLER

Nominal output current	4 A
Peak current	10 A
Max. intermediate circuit voltage	380 V DC
Mains frequency	50 ... 60 Hz
Max. nominal input current	5 A

In Fig. 4 the driver configuration can be seen in the FCT program. Every part of a system should be set with all parameters by this program, and after it can calculate the details, for example the maximum speed which is stored in its memory and it can be found in the user manual of the devices.



Fig. 4. Driver configuration

For the accuracy of the positioning we use the encoder which is mounted on the motor shaft, and the limit switch of the linear axis. We set the home position with using this Reed sensor. The slider goes to the limit switch (the side depends on the settings) and after reaching it, it goes immediately into the opposite direction. The slider stops when the signal is changing, this position is taken as the homing point.

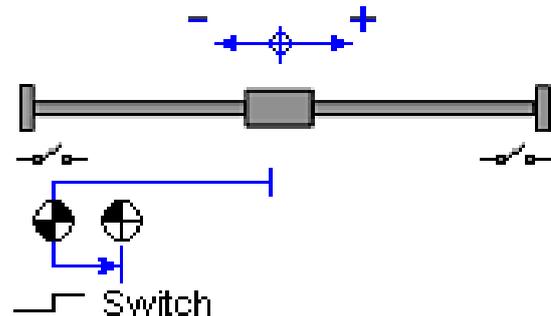


Fig. 5. Homing of the linear axis [4]

IV. INTRODUCTION OF THE SYSTEM

The configuration of the motor controller can be done with the FESTO Configuration Tool program. In one motor controller with using digital I/O mode we can store 63 different positions and 8 profiles (velocity, acceleration and deceleration). To control the system with the PLC we have to know the pin assignment of the motor controller, in order to be able to give the right commands to the enable inputs. During the process we define the number of the needed positions with the binary code of the six Record select bits (Fig.6.). The first one "000000" means the home position, and after we can choose from 63 positions. We can define also 8 profiles in the motor controller and with using these combinations we can make the PLC program which chooses between them depending on the user needs.

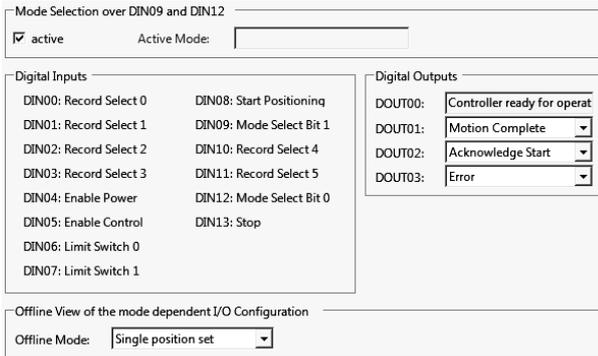


Fig. 6. Digital I/O of the motor controller

The controller also has digital outputs, with them we can make sure for example about the errors, the end of the positioning or the controller is ready for the operation. In this case we can send the next instruction to it.

With the Mode Select Bits we can choose between two modes: “00” is simple positioning and “10” means teaching new positions or crawling on the axis. In Single Position Set Mode we use the selection bits to define positions. We can see the used positions and profiles in the Fig. 7.

Position List		Position Profiles						
FCT	No.	Mode	Position [mm]	Profile	Command	Dest.	Input	
	1	A	20,00	0	END			
	2	A	100,00	1	END			
	3	A	180,00	2	END			
	4	A	260,00	3	END			
	5	A	340,00	4	END			
	6	A	420,00	5	END			
	7	A	500,00	6	END			
	8	A	580,00	7	END			

No.	Vel. [mm/s]	Accel. [m/s ²]	Decel. [m/s ²]	Smooth [%]	Time [ms]	Start D. [ms]	Fin.Vel. [mm/s]	Startcond.
0	100,00	1,000	1,000	0	0	0	0,00	Ignore
1	500,00	1,000	1,000	0	0	0	0,00	Ignore
2	900,00	1,000	1,000	0	0	0	0,00	Ignore
3	1300,00	1,000	1,000	0	0	0	0,00	Ignore
4	1600,00	1,000	1,000	0	0	0	0,00	Ignore
5	2000,00	1,000	1,000	0	0	0	0,00	Ignore
6	2400,00	1,000	1,000	0	0	0	0,00	Ignore
7	200,00	1,000	1,000	0	0	0	0,00	Ignore

Fig. 7. Used positions, velocities in the FCT

In Jogging/Teaching Mode with DIN10 and DIN11 inputs we can change the direction and rotate the motor slowly with 50 mm/s and after 1s the speed changes to 200 mm/s (Fig.8.). With this function we can find new positions, and with sending an impulse to DIN8 we can store them to the DIN1-DIN4 defined place.

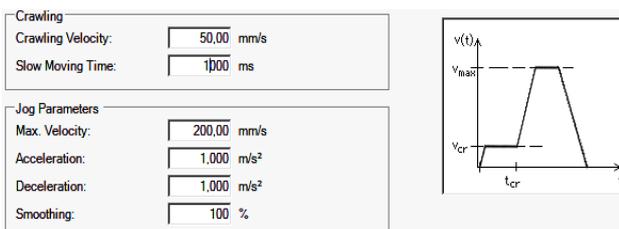


Fig. 8. Jog parameters

V. THE CONNECTION BETWEEN PLC AND HMI

The center part of the controlling is a FESTO CPX Programmable Logic Controller which has a modular structure and its programming interface is the Festo Software Tools software. We used an STL programming language, which describes sequentially processes.

A Festo FED-500 type HMI (Human-Machine Interface) provides the necessary information about the system’s functioning for the user, and makes the connection between the system and the user. It has a pressure sensitive (resistive) touch screen and it communicates with the PC and the PLC via serial port.

The signals of the limit switch and the digital outputs of the controller go into the two digital inputs module of the PLC. From the two digital outputs module of the PLC goes the necessary enable, and input information towards the motor controller through a parallel cable which has 25 pins.

The variables that were used during the HMI programming are defined as flags inside the PLC program. The structure of the program consists of two parts: the main program that operates the other low-class programs, it makes them active or inactive depends on the criterions.

```

STEP MODE_SEL
IF
THEN JMP TO HOME          F 0.06      'HOME

IF
THEN RESET                F 0.07      'RESTART
JMP TO INIT2              F 0.07      'RESTART

IF
THEN RESET                N          F 0.04      'MODE_SELECT
RESET                    02.6      'MODE_DIN12
SET                       02.7      'MODE_DIN9
P2                        P2        'POSITION
RESET                    P3        'JOG
JMP TO MODE_SEL

IF
THEN RESET                F 0.04      'MODE_SELECT
RESET                    02.2      'START/ TEACH
RESET                    02.7      'MODE_DIN9
SET                       02.6      'MODE_DIN12
SET                       P3        'JOG
RESET                    P2        'POSITION
JMP TO MODE_SEL
    
```

Fig. 9. Part of the main program of the PLC

In Fig. 9. can be seen a part of the main program, when it is deciding about the mode, in step Mode_sel the states is checking all the time. With the first “if” it is looking for whether the Home button has been pushed or not. With the second “if” it does the same with the reset button. If these variables are true the program jumps into another steps. We are also pay attention of the Mode select bits statements, and base on them it activates or deactivates programs. If there is not homing the main program is continually be in this step.

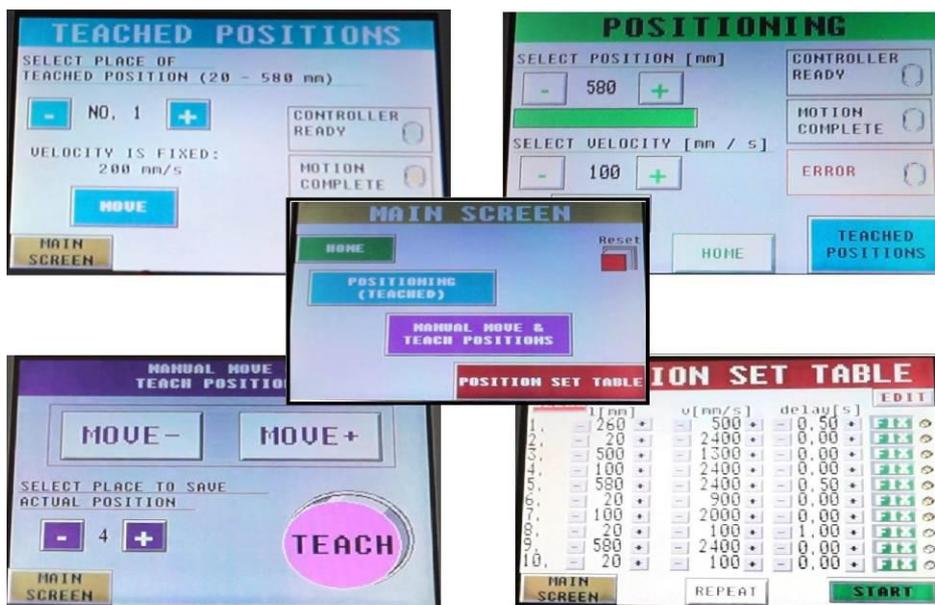


Fig. 10. The menu structure [6]

VI. INTRODUCTION OF THE USER INTERFACE

For the easier operating we created a menu structure which can be seen in Fig. 10. With pushing the buttons on the main screen (in the middle) we can select the other screens. Before starting the program it goes automatically to its home position, but with the home button the user can do it manually whenever it is necessary. The program can be restarted with the reset button.

In the Positioning screen (right on the top) we can select between 8 fixed positions, which can be seen in the Fig. 7. The long green line under this function symbolizes the linear axis, and the target position on it. We can choose fixed velocities (Fig. 7.) to every position and the indicator lights on the right side show information about the state of the controller.

In Manual move & Teach position screen (left below) we can move the dray with pushing move – or move + buttons. If it stops we can select a place (1-7) in the controller for the new position, and save it with the Teach button.

If we have manually taught positions we can load them with the screen on the left top. We can choose the number of the position and push the move button. It will go to the defined place with a fixed velocity (200 mm/s).

With the last screen on the right below, called Position set table, we can set a whole sequential program. We can describe the positions, choose a velocity and set a delay time, which means the time for waiting before starting the movement. After giving the details every used line should be fixed, otherwise the program skips them. If a position is reached the indicator light of the row shows it. The process starts repeatedly from the beginning if the repeat button is pushed.

VII. CONCLUSION AND RECOMMENDATIONS

This process and these settings can be quite difficult, and needs the good knowledge of the devices. Our aim was to create an interface to this system, with a simple structure for a common user. It means that it can be used easily without the deep knowledge of programming, or of using softwares.

VIII. REFERENCES

- [1] Mohamed A. El-Sharkawi: Fundamentals of electric drives, Cengage Learning, 2000
- [2] János Tóth: Electrical Actuators, University of Debrecen Faculty of Engineering, 2012
- [3] David Hoey, Dan Sandoval, Edward Gasper: TP 800 e-drives - Reference notes, Festo Didactic GmbH & Co., D-73770 Denkendorf, 2002
- [4] http://www.festo.com/net/SupportPortal/Files/10273/EMMS-AS_ENUS.pdf download: 20.04.2014.
- [5] http://www.festo.com/net/SupportPortal/Files/10272/CMMS-AS_ENUS.pdf download: 20.04.2014.
- [6] <http://www.youtube.com/watch?v=c2RhXUsSxtE>