

HOW TO CONTROL A PNEUMATICAL AND AN ELECTRICAL DRIVEN LINEAR AXIS

János TÓTH¹, Xénia Erzsébet TÓTH²

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

Abstract— In the industrial processes the positioning with high accuracy is really important. In this paper we controlled an electropneumatic and an electrical powered linear axes. The pneumatical part of this model (a pneumatical transporter with magnetic clutch) is moving to the x direction, and the electrical part (servo engine with its linear axis) is positioning to the y direction. The communication network is created with the PLC (Programmable Logic Controller) via CANopen protocol (based on CAN –Controlled Area Network- industrial standard bus) and FHPP (Festo Handling and Positioning Profile) data profile. We used a PID controller (proportional-integral-derivative controller) for the pneumatic positioning, and made it in the PLC program.

Keywords— closed-loop control, PID controller; pneumatic axis, servo motor, PLC, distance sensor

I. INTRODUCTION

The base of this paper was to create a positioning execution with two different driven linear axis. In our earlier works [1] we dealt with controlling of stepper and servo motors, and we used our experiences to create an innovative, combined system, which means that we connected a pneumatical and an electrical driven axis. These axes realized a fixed working area in which the positioning of an optional chosen tool –for example a pen or a gripper – takes place.

II. GENERAL INFORMATION ABOUT THE PNEUMATIC PART

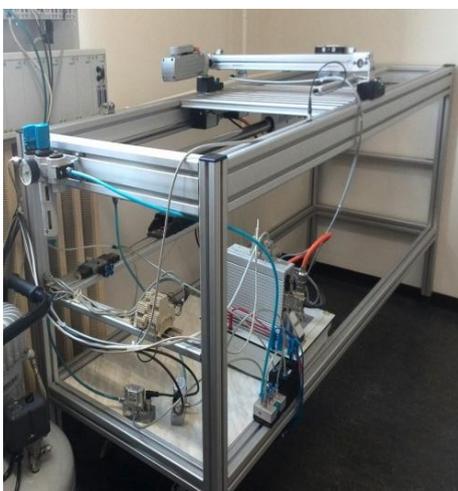


Fig. 1. Pneumatical transporter with magnetic clutch

In the case of the pneumatic drives the compressibility of the gas should be taken notice during the positioning method.

To move the pneumatic transporter (Fig. 1.) two electropneumatic pressure regulators were used.

These valves are controlled with 4-20 mA and proportional to this current they set the filling pressure between 0 and 10 bar of the chambers of the transporter. To reach the accurate position of the workbench of the transporter, a distance measuring sensor - operating on the Hall effect principle - with 0-503 mm measurement range was used, its output voltage - the feedback signal - according to this range is 0-10 V.

The pneumatic circuit was created with the Rexroth D&C Scheme Editor software (Fig. 2.).

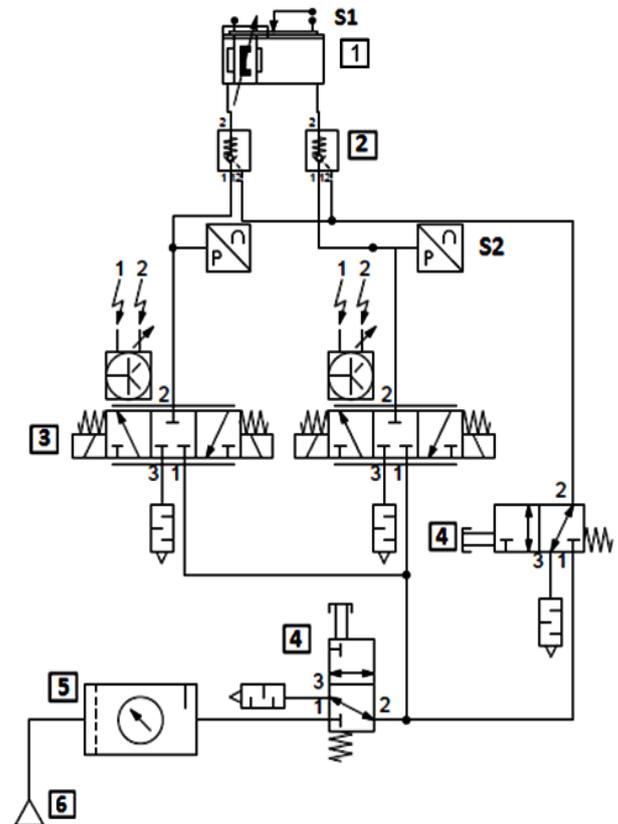


Fig. 2. Pneumatic circuit

It contains the following elements: double acting pneumatic transporter with magnetic clutch (1) with a distance measuring sensor (S1), controlled non-return valves (2), integrated pressure sensor (S2) in a 3/3

monostable electropneumatic pressure regulator valve (3), monostable manually operated 3/2 pneumatic valve (4), air service unit (5), compressor (6).

III. THE PID CONTROLLER FOR THE PNEUMATIC TRANSPORTER

A PID controller (proportional-integral-derivative controller) (Fig. 3.) was used for the accurate positioning of the workbench of the pneumatic transporter. It is a predefined function block in the CODESYS programming interface of the PLC (Programmable Logic Controller), which provides an easily configurable, parameterized, and simple to set adjustments.

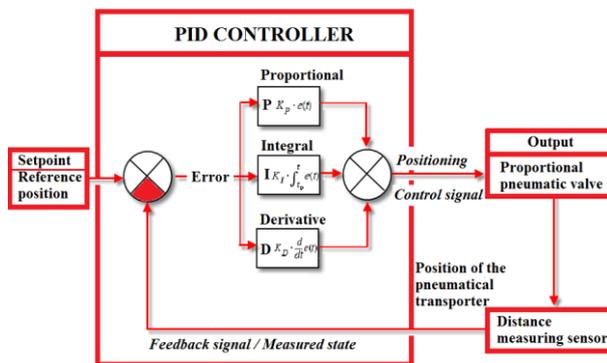


Fig. 3. The used closed-loop system

Based on the output of the PID the electropneumatic pressure regulator valves are controlled in an inverse way to each other (PneuValve1 and PneuValve2 in the PLC program). The feedback from the actual position of the workbench is the signal of the distance sensor (PneuActualPosition). With the xPIDMan variable a manual set point can be chosen (a slider in a control interface for testing), in the other case this set point depends on the target coordinates and the position of the servo driven linear axis (the PLC program makes the connection between them) (Fig. 4.).

```

0083 PneuValve1 := REAL_TO_WORD(PneuPIDOut);
0084 PneuValve2 := REAL_TO_WORD(30000-PneuPIDOut);
0085
0086 PIDpneu(
0087   ACTUAL:= PneuActualPosition ,
0088   SET_POINT:= PneuSetpointPosition,
0089   KP:= rKP,
0090   TN:= rTN,
0091   TV:= rTV,
0092   Y_MANUAL:= PneuSetpointPosition ,
0093   Y_OFFSET:= 15000 ,
0094   Y_MIN:= 7000 ,
0095   Y_MAX:= 23000 ,
0096   MANUAL:= xPIDMan,
0097   RESET:= xPIDReset,
0098   Y=>PneuPIDOut,
0099   LIMITS_ACTIVE=> ,
0100   OVERFLOW=> );
    
```

Fig. 4. The PLC program for the PID control

Defining $u(t)$ as the PID controller output, the transfer function is:

$$u(t) = K_P e(t) + K_I \int_0^t e(t) dt + K_D \frac{de}{dt} \quad (1)$$

where: K_P : proportional gain, a tuning parameter, K_I : integral gain, a tuning parameter [1/s], K_D : derivative gain, a tuning parameter [s], $e(t)$: the actual error.

Another form, containing the time terms:

$$K_{PID}(s) = K_P \left(1 + \frac{1}{T_N s} + T_V s \right) \quad (2)$$

$$K_I = \frac{K_P}{T_N} \quad \text{and} \quad K_D = K_P \cdot T_V \quad (3)$$

The substance of the control is that it compares the measured process variable (y) with the desired, set point value (y_0). The controller calculates the difference – it is called error - and it decides how to reduce it, with the new control variable (u), which brings closer y to y_0 [2].

The PID controller is the sum – parallel connection - of the proportional (P), integral (I), and derivative (D) terms. If its parameters are chosen incorrectly the output can oscillate, and the control can be instable, thus the tuning of the PID loop is necessary. The chosen method was the Ziegler-Nichols, according to this method the effect of the integral and the derivative terms should be set to zero. By increasing the proportional term to the output starts to oscillate, the K_u ultimate gain and its ultimate period P_u can be defined (Fig. 5.). We could monitor the tuning method and the transfer functions with the graphs on the CODESYS programming interface.

TABLE I
 ZIEGLER-NICHOLS METHOD – PARAMETERS [3]

P	$K_p = K_u / 2$	-	-
PI	$K_p = K_u / 2.2$	$T_I = P_u / 1.2$	-
PID	$K_p = K_u / 1.7$	$T_I = P_u / 2$	$T_D = P_u / 8$

The control loop became a stable system without oscillates.

```

0007 PneuValve1 AT %QW16 : WORD;
0008 PneuValve2 AT %QW17 : WORD;
0009
0010 PneuActualPosition AT %IW22 : WORD;
0011 PneuSetpointPosition : WORD;
0012 PneuPIDOut : REAL;
0013 PIDpneu: PID;
0014
0015 xPIDMan: BOOL;
0016 xPIDReset: BOOL;
0017 rKP: REAL :=1.5;
0018 rTN: REAL :=1;
0019 rTV: REAL :=0.004;
    
```

Fig. 5. The variables in the PLC program for the PID control

By measuring the pressure and the distance of the workbench (Fig. 6.) the power and velocity in function of the time can be defined.

$$v(t) = \frac{ds}{dt} \quad (4)$$

This $v(t)$ velocity can be calculated by dividing the displacements between the sampling time with the period time (Fig. 7.).

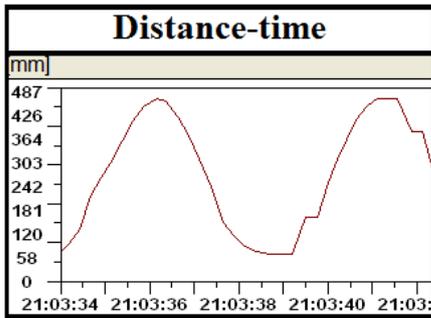


Fig. 6. The distance of the pneumatic workbench in function of the time

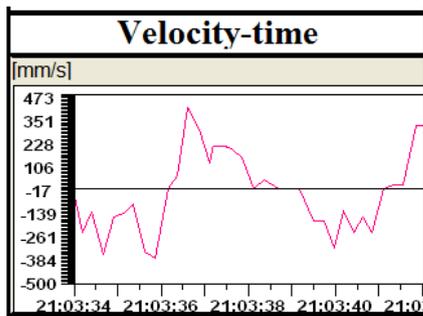


Fig. 7. The velocity of the pneumatic workbench in function of the time

IV. THE CONTROL UNIT

We use a FESTO CPX-CEC-M1 modular type Programmable Logic Controller (Fig. 8.). It controls the motor, the pneumatic valves, makes the PID controller, collects the signals of the sensors and makes the communication between the different parts.

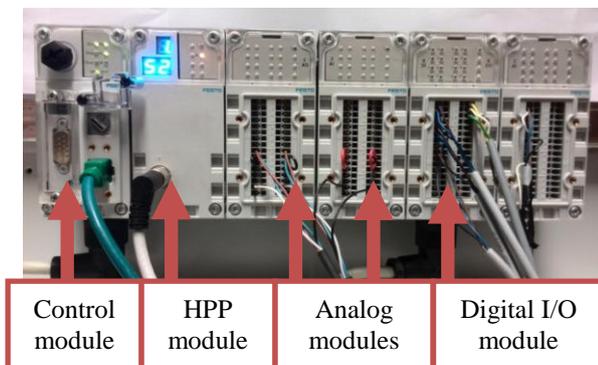


Fig. 8. The structure of the used PLC

With the CPX-CM-HPP communication module the PLC is able to control the motor via CAN

(Controlled Area Network) bus. The communication takes place on FHPP (Festo Handling and Positioning Profile) data profile.

The PLC program was written with CODESYS user interface in structured text programming language and it is based on the case function [5].

V. THE ELECTRICAL DRIVEN AXIS

A FESTO EMMS-AS-55-S-TM type servo motor with a linear tooth belt axis is moving perpendicular to the pneumatic axis. The range of the positioning is between 0 and 300 mm (Fig. 9.).



Fig. 9. The used motor and linear axis

The used motor controller is a FESTO CMMS-AS-C4-3A-G2. The configuration of the motor controller can be done with the FESTO Configuration Tool (FCT) program. Every part of a system should be set with all parameters by this program, and after it calculates the details, for example the maximum speed, which is stored in its memory. 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.

In the FCT program we can monitor the movement according to different parameters, and the results can save to the computer (Fig.10.).

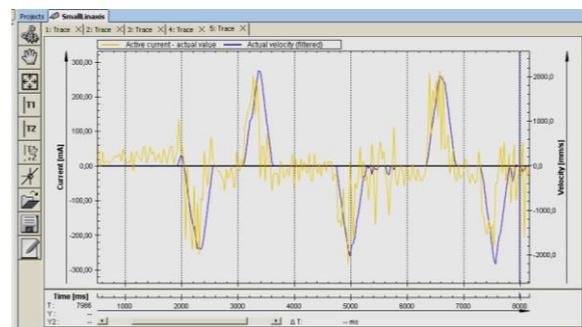


Fig.10. The velocity and the current of the motor in function of time

By using the FHPP data profile we have the possibility to control 4 different motor controllers via CAN bus. To solve this we have 32 byte data, 8 byte for each motor controller.

We can choose between two different modes the Record Select and the Direct mode. Each mode have

their own control, status and reserved bytes, they can be seen in Table 2.

TABLE II
 CONTROL AND STATUS BYTES OF THE MODES [4]

In Record Select mode we can use in the motor controller defined and configured discrete positions and dynamic parameters. In one motor controller with

Record Select mode					
	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5-8
Output data	CCON	CPOS	Record number	Reserved	Reserved
Input data	SCONS	SPOS	Record number	RSB	Actual position
Direct mode					
	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5-8
Output data	CCON	CPOS	CDIR	Set value (velocity)	Setpoint 2 (position, force, torque)
Input data	SCONS	SPOS	SDIR	Actual value (velocity)	Actual value (position, force, torque)

using digital I/O mode we can store 63 different positions and 8 profiles (velocity, acceleration and deceleration).

By choosing the Direct mode all the motor parameters can be set and dynamically change from the PLC program. In this mode with the CDIR byte we can choose between the position, velocity, force and torque control mode, and use them in the program with input and output bytes 5-8, their actual and set points.

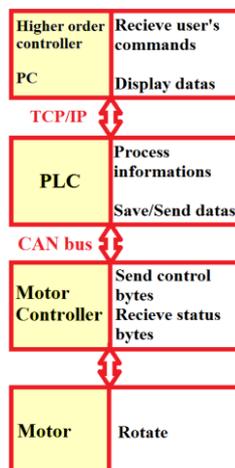


Fig. 11. The communication between the devices

In the CODESYS programming interface we created a user interface for testing our system. Our aim was to make 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. The user can send commands on it, and set velocity values of both axes, monitor the motions and change the modes.

In the Fig. 11. the structure of the communication between the devices of the electrical system can be seen.

VI. CONCLUSION AND RECOMMENDATIONS

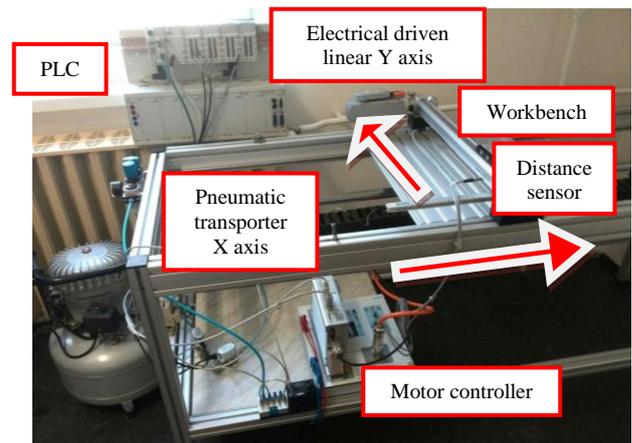


Fig. 12. The main components of the created system

The created system can be seen in Fig.12. it can be used in different industrial field, it depends on the used tool. This construction can be extended with different tools, for example with a pen (drawing method, plotter function), a blade (stamping out), a gripper (packing). By increasing the accuracy of the pneumatic part –for precision positioning - and integrating a new motor driven axis into the z direction, this system could work as 3D printer, CNC (Computer Numerical Control) machine, etc.

REFERENCES

- [1] Xénia Erzsébet Tóth, János Tóth: Linear axis positioning with industrial servo motor , IMT Oradea – 2014, ISBN:978 606 10 1276 3
- [2] U. Tietze, Ch. Schenk: *Advanced Electronic Circuits*, Springer Berlin Heidelberg, 1978, ISBN: 978-3-642-81243-9, pp: 464-490
- [3] Ziegler-Nichols Method: <http://www.chem.mtu.edu/~tbc/cm416/zn.html>, Downloaded: 2015.04.03.
- [4] Festo handling and positioning profile: <http://www.festo.com/net/SupportPortal/Files/319397/555696g1.pdf>, Downloaded: 2014. 11.12.
- [5] Karl Heinz John, Michael Tiegelkamp: *IEC 61131-3: Programming Industrial Automation Systems*, Springer, London, 2010, ISBN: 9783642120145