

Force control on direct driven servo hydraulic actuator

ALEKS PETROVIČ, MIHAEL JANEŽIČ & VITO TIČ

Abstract Direct Driven Servo Hydraulic Actuator also known as Pump Direct Driven Cylinder (PDDC) represents a decentralized modern concept of energy efficient cylinder control without damping losses of direction valves. Such systems have many advantages over conventional hydraulic systems and combine benefits of hydraulic and electric drives. PDDC system developed in Laboratory for Oil Hydraulics at University of Maribor consists of hydro motor, which is used as a reversible pump that is directly driven by servomotor and is designed for experimental testing with differential hydraulic cylinder. In this paper, the aforementioned system runs experimental setup for force control of hydraulic cylinder, with load produced by pneumatic bellow.

Keywords: • PDDC • PID • trajectory • step • HMI •

CORRESPONDENCE ADDRESS: Aleks Petrovič, University of Maribor, Faculty of Mechanical Engineering, Smetanova ulica 17, 2000 Maribor, Slovenia, e-mail: aleks.petrovic@student.um.si. Mihael Janežič, University of Maribor, Faculty of Mechanical Engineering, Smetanova ulica 17, 2000 Maribor, Slovenia, e-mail: mihael.janezic@student.um.si. Vito Tič, University of Maribor, Faculty of Mechanical Engineering, Smetanova ulica 17, 2000 Maribor, Slovenia, e-mail: vito.tic@um.si.

1 Introduction

In comparison to conventional systems, PDDC systems don't use direction valves. Direction is controlled by rotation of servomotor connected to reversible pump. PDDC system are very compact, mobile, universal and powerful which bring many advantages such as:

- productivity,
- overload safety,
- robustness (longer maintenance intervals),
- easier maintenance,
- accuracy and dynamic of servomotor,
- small use of hydraulic oil (because of system compactness),
- ability to connect to other systems (industry 4.0).

Such systems are usually used to bend, press, move, lift and position objects [11].

For this paper, an experiment using different ways of force trajectory type was designed. PDDC system is force controlled using step, three-phase and sinus trajectory. For maintaining force value, trajectories are used as an input value of PID regulator. Parameters of various trajectories and PID regulator parameters are initialized using Windows forms C# application developed in Visual Studio. User can easily change all of parameters of specific control for better control of the desired trajectory.

2 Experimental setup

Experimental system consist of Rexroth Indramat permanent magnet servomotor MKD071B that is speed and torque controlled by Indramat DKC03 servo motor driver. For this particular application, motor torque is set to 24 Nm with speed limited to $\pm 3000 \text{ min}^{-1}$. It's connected to directly driven hydraulic pump using bell housing with coupler, which is connected to custom made hydraulic block with screw in check valves, filters, hydraulic accumulator and adjustTable pressure relief valves set to 210 bar for both sides of differential cylinders chambers, shown in Figure 1. Each cylinder chamber pressure is measured with Hawe DT2-4 pressure sensor. Additional pressure sensor is also mounted on hydraulic block to monitor pressure in tank/accumulator line. Complete hydraulic scheme is shown in Figure 3.

Differential hydraulic cylinder ($d=22$ mm, $D=32$ mm, cylinder stroke = 200 mm) made by Bosch Rexroth is used as hydraulic actuator. Given above max. pressure and size of cylinder, it can push up to 16 kN of force and pull up to 8,5 kN. Closed loop system control is based on Beckhoff industrial soft PLC mounted in electrical cabinet shown in Figure 2.

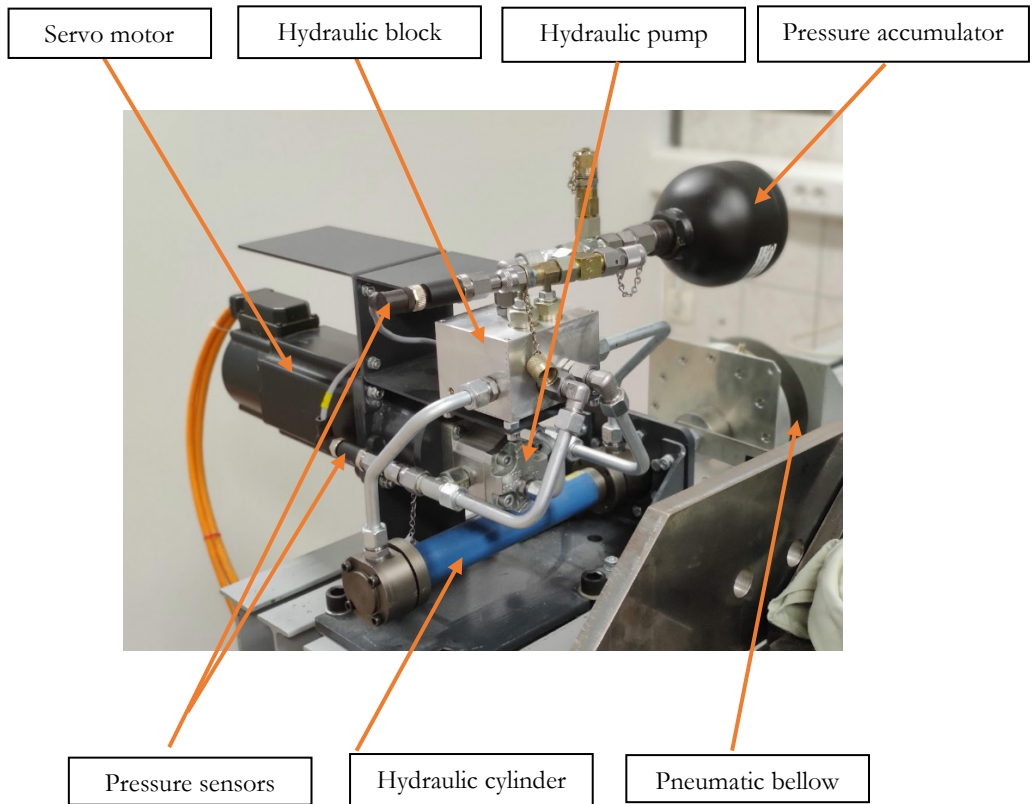


Figure 1: PDDC.

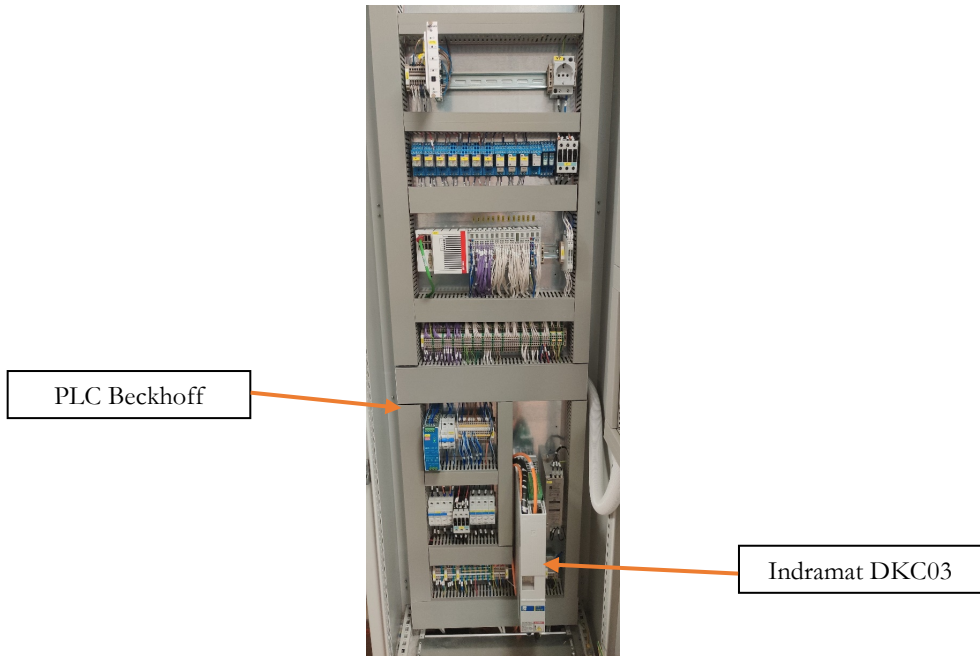


Figure 2: Electrical cabinet.

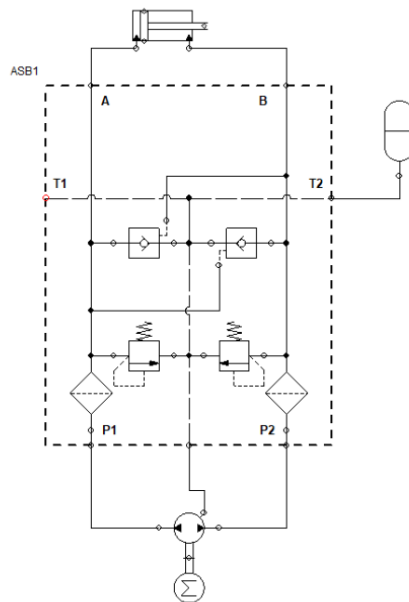


Figure 3: Hydraulic scheme [1].

2.1 Beckhoff soft PLC

PLC used in system is CX5140 that is made by company Beckhoff. This PLC was chosen because of its modularity whit I/O cards (current configuration is shown in Figure 4), that are DIN rail compatible so they can be easily changed. Its technical data can be found in Table 1 [2].

PLC – PC connection is done with Ethernet connection via RJ45 port. PLC – PC connection is used for programming, troubleshooting and real time monitoring. Final program and HMI interface are uploaded to PLC and run locally on soft PLC that simultaneously run real-time PLC program and developed Windows Forms application on Windows 7 Embedded.

Table 1: PLC technical data

| Tehcnical data | | CX5140 |
|---|---|--------|
| Procesor | Intel Atom® E3845, 1.91 GHz | |
| Number of cores | 4 | |
| Flash memory | Slot for Cfast card and microSD cards | |
| Main memory | 4 GB DDR3 RAM | |
| Interfaces | 2x RJ45 10-1000Mbit/s 1x DVI-I, 4 x USB 2.0 | |
| Diagnostic LED | 1 x power, 1 x TC status, 1 x flash access, 2 x bus status | |
| Operating system | Windows Embedded Compact 7 (supports only one CPU core), | |
| | Windows Embedded Standard 7 P, Windows 10 IoT Enterprise 2016 LTSC, | |
| | Windows 10 IoT Enterprise 2019 LTSC, TwinCAT/BSO | |
| Control software | TwinCAT 2 runtime, TwinCAT 3 runtime (XAR) | |
| I/O connection | E-bus or K-bus, automatic recognition | |
| Power supply | 24 V DC | |
| Current supply E-bus/K-bus | 2A | |
| Max. power consumption | 16 W | |
| Max. power consumption (with loading UPS) | 23 W | |
| Dimensions (W x H x D) | 142 mm x 100 mm x 92 mm | |

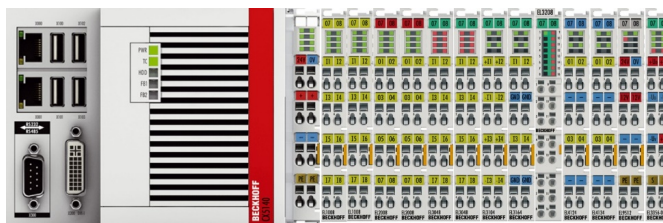


Figure 4: PLC with I/O cards [4].

2.1.1 I/O Devices

There are multiple cards in system that are not used for current experiment. They will be used in future experiments. List of all cards and their current configuration are listed in Table 2.

Beckhoff EL1008 [5] card is used as an electrical cabinet switch state input and Indramat digital output state check. For relay activating and controlling of digital inputs on Indramat servo-motor controller, EL2008 [6] was used. Analog values received from HAWE DT pressure sensors and Omron ZX-1 CMOS sensor are read using EL3048 [7] card. EL3014 [8] card is used to read analog outputs from Indramat servo-motor controller. Analog inputs of Rexroth Indramat servo-motor controller are controlled using EL4134 [9] card.

Table 2: List of I/O cards

| Card (current order) ▼ | characteristics ▼ |
|------------------------|---|
| EL1008 | 8-channel digital input, 24 V DC |
| EL1008 | 8-channel digital input, 24 V DC |
| EL2008 | 8-channel digital output, 24 V DC, 0.5 A |
| EL2008 | 8-channel digital output, 24 V DC, 0.5 A |
| EL3048 | 8-channel analog input, current, 0...20 mA, 12 bit, single-ended |
| EL3048 | 8-channel analog input, current, 0...20 mA, 12 bit, single-ended |
| EL3104 | 4-channel analog input, voltage, ± 10 V, 16 bit, differential |
| EL3164 | 4-channel analog input, voltage, 0...10 V, 16 bit, single-ended |
| EL3208 | 8-channel analog input, temperature, RTD (Pt100), 16 bit |
| EL4124 | 4-channel analog output, current, 4...20 mA, 16 bit |
| EL4134 | 4-channel analog output, voltage, ± 10 V, 16 bit |
| EL9512 | Power supply terminal 12 V DC |
| EL3356 | 1-channel analog input, measuring bridge, full bridge, 16 bit |

2.2 OMRON ZX1-LD600A81

For measuring piston rod movement, OMRON laser displacement sensor with CMOS sensor is used. It allows us to measure up to 0.08 mm of movement in standard mode with 50 ms response time. With some accuracy loss, sensor is also capable of high speed capturing with 1.5 ms response time. In our application sensor does not play any role in regulation of the system. Sensor is mounted in custom-made 3D printed enclosure visible in Figure 5.

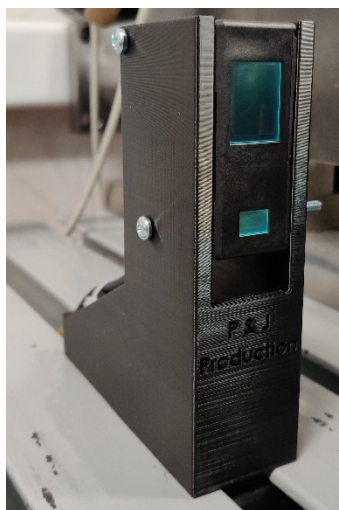


Figure 5: Omron laser.

3 Software solutions

PLC program was designed in TwinCAT 3 development environment. Different program languages such as FBD, SFC and ST were used. For human machine interface (HMI), we have developed user interface using Windows Forms C# for controlling and monitoring different parameters of PLC program.

As mentioned in introduction, PID controller is used to control trajectory of the system by following three different types of generated trajectory (step, three-phase trajectory and sinus).

3.1 PLC Program

PLC program developed in Twincat3 is running locally on Beckhoff soft PLC controller. Main program is written in Structured Text language (ST), and uses Case statements for selecting subprograms (Figure 6), which determine working mode of PDDC system.

```
1 Read ();
2
3 CASE gvl.stanje OF
4 0:
5   Count ();
6 1:
7   Manual ();
8 2:
9   EHAClosedLoopPosition ();
10 3:
11  SERVO_CL ();
12 4:
13  HAWK ();
14 5:
15   EHA_Force_PID ();
16
17 END_CASE
18
19 CycleTest_calc ();
20 Write ();
```

Figure 6: Main program.

Mode subprograms are also written in Structured Text language with Case statements that determine the behaviour of the system using subprogram. For example in Figure 7, program is in Force control mode and in system start up procedure.

```
1 CASE gvl.EHA_MODE OF
2 1:
3   EHAstart ();
4 2:
5   EHAstop ();
6 3:
7   EHA_FORCE ();
8 END_CASE
```

Figure 7: Mode subprograms.

For start-up and stopping procedure of Rexroth Indramat servo driver, subprograms EHA_Start and EHA_Stop are used. Both subprograms are used multiple times in PLC program. Subprogram for stopping procedure is visible in Figure 8. Program logic is shown in Figure 9.


```

start1:= gvl.timerEHAStop;

gvl.DriveEnable :=FALSE;
gvl.INDR_PS_BR:= FALSE;
gvl.E_stop:= FALSE;
gvl.DriveHalt:= FALSE;

Timer1(IN:= start1, PT:= T#1000MS);
IF Timer1.Q THEN
  gvl.INDRAMAT_POWER_ON := FALSE;
  gvl.timerEHAStop :=FALSE;
  gvl.EHA:= 0;
  gvl.HAWE_SET_MODE:= 5;
  gvl.EHA_MODE := 0;

END_IF
    
```

Figure 8: Stopping procedure subprogram.

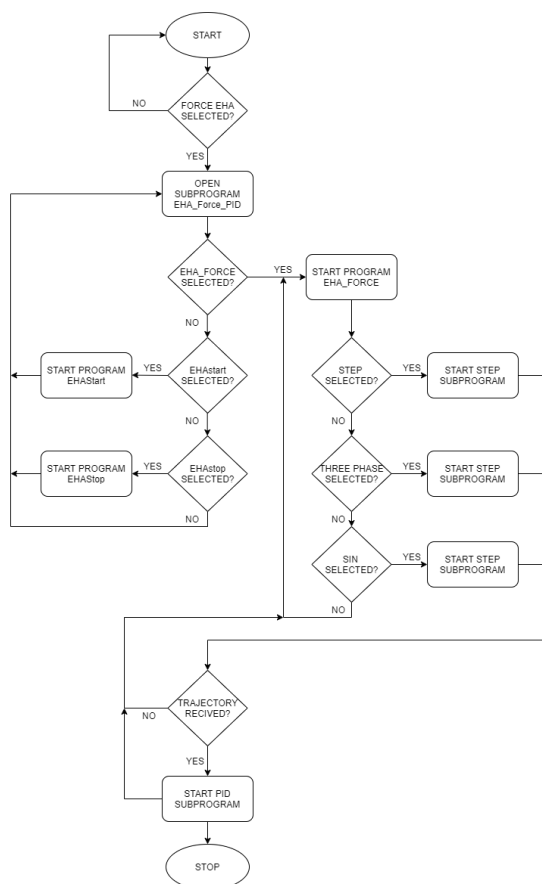


Figure 9: Program logic.

3.2 Step response

Subprogram *forceSTEP_EHA*, written in ST, is used to generate trajectory for step response.

It reads user input value and directly applies it to the input of PID controller as desired value (Figure 10).

```
gvl.trajectory_OUT_eha := gvl.STargetPosForceEHA;
```

Figure 10: Step trajectory subprogram

3.3 Trajectory

Trajectory control is one of the most widely used control of movement in industry. To generate trajectory, we have used three phase trajectory generator that can be found in Twincat 3 libraries. This subprogram was written in FBD language (Figure 11). Generator is controlled by binary values that determine its state. Generator parameters are defined in initialization subprogram and are combined in array structure called *stParams* that consist of start position, target position, start velocity, desired velocity, start acceleration/deceleration, desired acceleration/deceleration, control cycle time and task cycle time (Figure 12). Generated structure is inserted in one of the inputs of the three-phase generator.

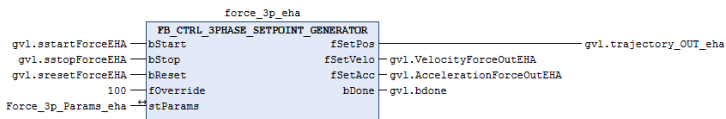


Figure 11: Three phase set point generator.

```
// 3p

Force_3p_Params_eha.fStartPos := gvl.EHAForceValue;
Force_3p_Params_eha.fTargetPos := gvl.STargetPosForceEHA;
Force_3p_Params_eha.fStartVelo := 0;
Force_3p_Params_eha.fVelocity := gvl.SVelocityForceEHA;
Force_3p_Params_eha.fTargetVelo := 0;
Force_3p_Params_eha.fAcceleration := gvl.SAccelerationForceEHA;
Force_3p_Params_eha.fDeceleration := gvl.SDecelerationForceEHA;
Force_3p_Params_eha.tCtrlCycleTime := T#1MS;
Force_3p_Params_eha.tTaskCycleTime := T#1MS;

eModeFORCEgen := gvl.EHA_SIN_Emode;
eModeFORCE := gvl.EHA_PID_Emode;
```

Figure 12: StParams structure

3.4 SIN

For quick constant changes of trajectory, sinus generator from Twincat 3 library was used (Figure 13). It generates sin wave using amplitude and offset parameters that are defined in array structure called *stParams* shown in Figure 14. Working mode of generator is defined in *eMode* (passive, active). *E state* parameters show current state of the generator.

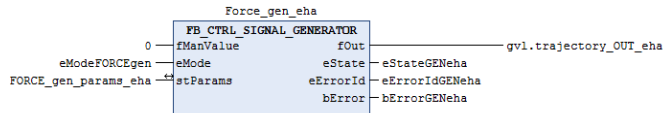


Figure 13: Sinus signal generator.

```
//sin
FORCE_gen_params_eha.fAmplitude := gvl.FORCEAmplitudeEHA;
FORCE_gen_params_eha.fOffset := gvl.FORCEOffsetEHA;
FORCE_gen_params_eha.tCtrlCycleTime := T#1MS;
FORCE_gen_params_eha.tTaskCycleTime := T#1MS;
FORCE_gen_params_eha.eSignalType := 1;
// E_CTRL_SIGNAL_TYPE;
FORCE_gen_params_eha.tCycleDuration := gvl.FORCEFreqEHA;
FORCE_gen_params_eha.tStart := T#00MS;
```

Figure 14: Sinus StParams structure.

3.5 PID

Trajectory generated from step, 3 phase trajectory or sinus are written in global variable called *trajectory_OUT_eha* which is used as input for PID controller (Figure 15). PID controller used is taken from Twincat 3 library. It compares actual (*gvl.EHAForceValue*) and desired value (*gvl.trajectory_OUT_eha*). Parameters for PID controller are written in array structure *stParams* and set in initialization subprogram. MUL block is used to convert motor direction. Limit is set to limit motor speed. Actual force of hydraulic cylinder is calculated from acquired pressure from both sides of cylinder chambers (by subtracting forces in each cylinder chamber in *Read* subprogram) [12].

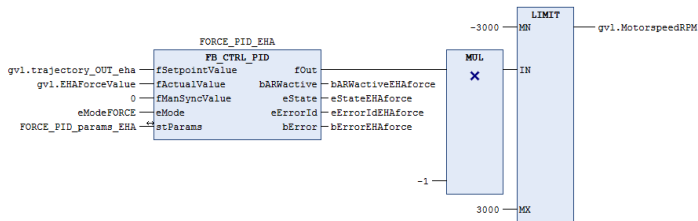


Figure 15: PID controller

4 HMI

For easier control and data monitoring HMI was created in Windows Forms C# application using Visual studio. It connects to PLC with ADS communication protocol.

Main goal when designing HMI was to make it simple and easy to use. From HMI user can initialize Start/Stop procedure, select work mode (Step, Sinus, 3-phase trajectory) and type of regulator (P, PI, PD, PID). User can also define the parameters for work mode, trajectory and closed-loop control parameters.

Scope is used to monitor and display actual and desired force value of piston. It also shows position of hydraulic cylinder from Omron distance laser.

HMI can also be used in case of emergency. Emergency stop button turns of system and stops servomotor instantly. System can be started again using Clear Error button, which clears errors on Rexroth Indramat and enables the system. System state can be monitored by true/false indicators shown in Figure 16.

TwinCAT 3 ADS notifications protocol is used for event driven reading of variables. Library waits for designated variables to change state and sends them to HMI using separate thread of UI thread. In HMI, communication thread is opened, variables are then transferred to UI thread. When transfer is completed, communication thread is closed and variables are assigned to their local variables. Those local variables can then be displayed without UI lag.

Library can work with up to 500 variables without causing delays in responsiveness of UI.

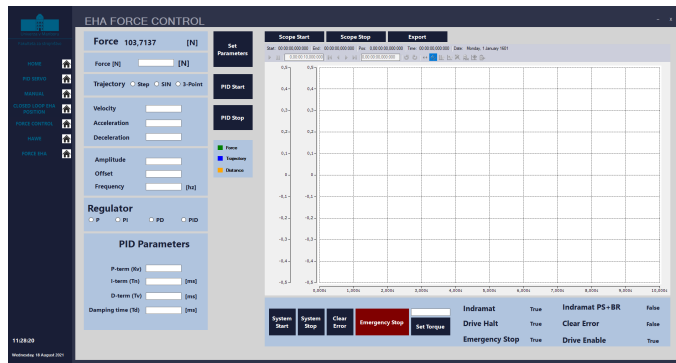


Figure 16: HMI

5 Experimental results

System responses are monitored in Visual studio using Twincat 3 Scope. Experiment was executed on three various type of trajectory (step, three-phase and sinus). The responses are commented in paragraphs below.

5.1 Step trajectory response

First experiment was done using step trajectory. As visible in Figure 17 response of the system with PID regulator follows desired trajectory with small amount of phase shift and error in amplitude. At holding constant force, we can notice jerking of force caused by jerking of PID controlled servomotor.

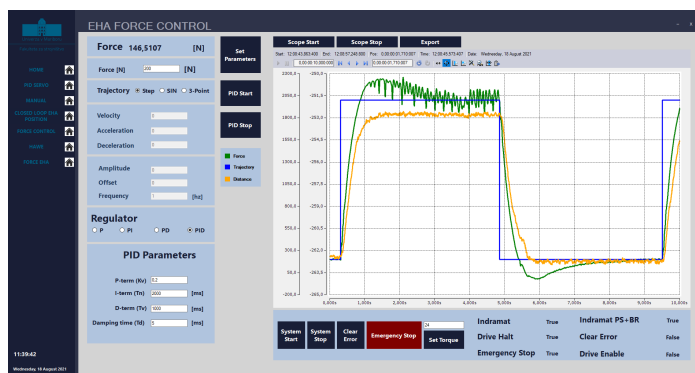


Figure 17: Step response.

5.2 Three phase trajectory response

Second type of trajectory used in experiment was three-phase trajectory. Figure 18 shows a small amount of overshoot visible in lowering of force and jerking of PID controlled servomotor when trajectory levels out and force is constant.



Figure 18: Three phase response

5.3 Sinus trajectory response

Sinus trajectory was the last trajectory tested in experiment. System follows desired force very well, shows only a small amount of overshooting, and phase delay (Figure 19). Jerking of force on peaks off sine wave is caused by changing of the servo motor rotation direction.



Figure 19: Sinus response

6 Conclusion

Experimental results have shown that PID controller has tracking error and slower dynamics. Satisfying results are achieved only in slower trajectories. As soon as trajectories are changed quickly, more significant overshoot is seen and system lags behind trajectory, which results in phase shift in square and sine waves. Parameters of PID controller were selected with trial and error method.

For better responses, non-linear control algorithms should be considered. Better accuracy of force control could also be achieved using load cell, which would eliminate reading errors of calculated forces of cylinder.

Experimental setup represents a good foundation for further experiments on different PID controllers for PDDC systems.

References

- [1] Tine, J. (2020), *Zasnova in izdelava direktno vodenega elektro-hidravličnega aktuatorja*: magistrsko delo [online]. University of Maribor, Faculty of Mechanical Engineering. (Accessed 24 August 2021)
Retrieved from: <https://dk.um.si/IzpisGradiva.php?lang=eng&id=77183>
- [2] Beckhoff Automation. [Accessed 24 August 2021]. Retrieved from: <https://www.beckhoff.com/en-en/company/>
- [3] Beckhoff Information System [Accessed 24 August 2021]. Retrieved from: https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/tcplcctrl_statement_if.htm&id=2531424188494754278
- [4] Beckhoff Automation EtherCAT terminal [Accessed 24 August 2021] Retrieved from: <https://www.beckhoff.com/sl-si/products/i-o/ethercat-terminals/>
- [5] Beckhoff Automation EL1008 EtherCAT terminal [Accessed 24 August 2021] Retrieved from: <https://www.beckhoff.com/sl-si/products/i-o/ethercat-terminals/el1xxx-digital-input/el1008.html>
- [6] Beckhoff Automation EL2008 EtherCAT terminal [Accessed 24 August 2021] Retrieved from: <https://www.beckhoff.com/sl-si/products/i-o/ethercat-terminals/el2xxx-digital-output/el2008.html>
- [7] Beckhoff Automation EL3048 EtherCAT terminal [Accessed 24 August 2021] Retrieved from: <https://www.beckhoff.com/sl-si/products/i-o/ethercat-terminals/el3xxx-analog-input/el3048.html>
- [8] Beckhoff Automation EL3104 EtherCAT terminal [Accessed 24 August 2021] Retrieved from: <https://www.beckhoff.com/sl-si/products/i-o/ethercat-terminals/el3xxx-analog-input/el3104.html>
- [9] Beckhoff Automation EL4134 EtherCAT terminal [Accessed 24 August 2021] Retrieved from: <https://www.beckhoff.com/sl-si/products/i-o/ethercat-terminals/el4xxx-analog-output/el4134.html>

- [10] Omron Industrial Automation [Accessed 24 August 2021] Retrieved from: <https://industrial.omron.eu/en/products/ZX1-LD600A81-2M>
- [11] Lovrec Darko, Uvod v hidravlično pogonsko-krmilno tehniko. Univerzitetna založba, November 2018
- [12] Kiker Edo, Regulacijska tehnika. Univerza v Mariboru, 5. 9. 2000