LAPPEENRANTA UNIVERSITY OF TECHNOLOGY
LUT School of Energy Systems
LUT Mechanical Engineering

Mikhail Proskuriakov

CONTROL OF A HYDRAULIC SERVO SYSTEM USING BECKHOFF REAL-TIME DATA ACQUISITION COMPUTER BASED ON MATLAB/SIMULINK PLATFORM

Examiners: Professor Heikki Handroos
Dr. Hamid Roozbahani

ABSTRACT

Lappeenranta University of Technology
LUT School of Energy Systems
LUT Mechanical Engineering

Mikhail Proskuriakov

Control of a Hydraulic Servo System Using Beckhoff Real-time Data Acquisition Computer Based on MATLAB/SIMULINK Platform

Master Thesis

2018

89 pages, 81 figures and 25 tables

Examiners: Professor Heikki Handroos
Dr. Hamid Roozbahani

Keywords: Control Theory; Real-time Simulation; MATLAB/SIMULINK; Beckhoff; TwinCAT; Hydraulic System; Hydraulic Manipulation; Mathematical Modeling; Optimization.

This thesis presents development, implementation and testing the control algorithm of hydraulic servo system that is utilized as a test rig with 1-D hydraulic slider in Laboratory of Intelligent Machines which is a part of Mechanical Department in Lappeenranta University of Technology. The control scheme is based on using Beckhoff Embedded PC as a Real-Time Data Acquisition Computer and industrial PLC, MATLAB/SIMULINK as a programming, data capturing and development environment and TwinCAT 3 as real-time control environment. The hydraulic system is described from both physical and mathematical points of view. Mathematical modeling consists of system's components, phenomena's and processes expressions which were modeled using MATLAB/SIMULINK blocks. The system simulation model and laboratory rig both adjusted and tested with same types of typical disturbances. As a control algorithm test results system response as load position and pressure curves were obtained. The thesis includes two case studies which serve as preliminary research steps prior to the main research work. Those case studies provide hardware module and software descriptions. The first case study is dedicated to testing software with a simple digital actuator performing on and off tasks; the second is dedicated to Beckhoff DC-servomotor rotational velocity control method. Despite the exploratory character, these case studies can be utilized as instances and detailed instructions of MATLAB/SIMULINK environment and Beckhoff TwinCAT software systems integration process.

ACKNOWLEDGMENT

First of all I would like to thank my supervisor Dr. Hamid Roozbahani for giving me an opportunity to be where I am and to do what I have done. His attitude, help and constant support during this work made it all possible.

I express my gratitude to Prof. Heikki Handroos, who gave me the research scope at the first place and introduced the Laboratory of Intelligent Machines with providing access to it.

And last but not least, this thesis would not be accomplished without help and support of Juha Koivisto, who put a lot of his effort and time trying to solve every technical issue related to this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

A_1	Piston area (m^2)
A_2	Piston rod-side area (m^2)
a_1	Effective Bulk Modulus Coefficient
a_2	Effective Bulk Modulus Coefficient
a_3	Effective Bulk Modulus Coefficient
c_s	Flow Constant ($m^3s^{-1}v^{-1}Pa^{-1/2}$)
E_{max}	Effective Bulk Modulus Coefficient (Pa)
F_c	Coulomb Friction (N)
F_f	Friction force (N)
Fs	Static Motion Friction (N)
K	Gain
K_I	Integral Gain
K_P	Proportional Gain
k_v	Viscous Friction (Ns/m)
L	Piston Maximum Stroke (m)
L_i	Laminar Leakage Flow Coefficient ($m^3s^{-1}Pa^{-1}$)
l_1	Laminar Leakage Flow Coefficient №1 (m^3/s)
l_2	Laminar Leakage Flow Coefficient №2 (m^3/s)
m	Mass (kg)
p_1	Pressure in the First Cylinder Chamber (Pa)
p_2	Pressure in the Second Cylinder Chamber (Pa)
p_{max}	Maximum System Pressure (Pa)
p_s	Supply Pressure (Pa)
p_t	Tank Pressure (Pa)
Q_1	Valve Flow №1 (m^3/s)
Q_2	Valve Flow №2 (m^3/s)
Q_{L1}	External Leakage Flow №1 (m^3/s)
Q_{L2}	External Leakage Flow №2 (m^3/s)
Q_{Li}	Internal Leakage Flow (m^3/s)
T	Time Constant (s)
t_1	Time Constant (s^{-1})
t_2	Time Constant (s^{-1})
u	Input Voltage (V)
u_s	Valve Input Voltage (V)
V_1	First Chamber Volume (m^3)
V_2	Second Chamber Volume (m^3)
v_{01}	First Pipeline Volume (m^3)
v_{02}	Second Pipeline Volume (m^3)
v_s	Stribeck Velocity (m/s)

x_p	Piston Displacement (m)
z	Internal State
β_{e1}	Effective Bulk Modulus of the First Chamber (Pa)
β_{e2}	Effective Bulk Modulus of the Second Chamber (Pa)
ζ	Damping Ratio
σ_0	Stiffness Coefficient (N/m)
σ_1	Damping Coefficient (N×s/m)
τ_1	No Unit
τ_2	Unit (1/s)
ω_n	Natural Angular Velocity (rad/s)
ADS	Automation Device Specification
CFM	Cubic Feet per Minute
CNC	Computer Numerical Control
DC	Direct Current
DCV	Directional Control Valve
DOF	Degree Of Freedom
I/O	Input/Output
IEC	International Electrotechnical Commission
LED	Light-Emitting Diode
LVDT	Linear Variable Differential Transducer
NC	Numerical Control
OS	Operating System
PC	Personal Computer
PID	Proportional Integral Derivative
PLC	Programmable Logic Controller
POU	Program Organization Unit
RPM	Revolutions Per Minute
RTDAQ	Real-Time Data Acquisition
STO	Safe Torque Off
XAE	eXtended Automation Engineering
XAR	eXtended Automation Runtime
XAT	eXtended Automation Technology

1. INTRODUCTION

1.1 Basic Information

Project Leader: Professor Heikki Handroos, Lappeenranta University of Technology

Project Supervisor: Dr. Hamid Roozbahani, Lappeenranta University of Technology

This project work was carried out in the Laboratory of Intelligent Machines which is a part of Mechanical Department at Lappeenranta University of Technology (Finland). The laboratory test rig, presented by hydraulically driven 1-D manipulator, is used as a general actuator in this work. The used hardware is Beckhoff Industrial Embedded PC. MATLAB/SIMULINK is used as control algorithm development software, and TwinCAT 3 is used as real-time software.

1.2 Background and Relevance of Work

Powerful, multifunctional compact and not expensive industrial PCs (Personal Computers) have always been in demand in automatic control sphere. Beckhoff Automation Company provides a wide range of such PCs and a special software system, called TwinCAT (The Windows Control and Automation Technology), to interact with it. The problem which can be faced when dealing with TwinCAT is a lack of knowledge and understanding of this software environment, because it is not commonly used anywhere but Beckhoff hardware. To develop sophisticated control schemes and algorithms programming languages knowledge is needed; it also requires some time to get acquainted with this new for side-user software. Meanwhile, MATLAB/SIMULINK software package is a world known and very commonly used environment for automation control schemes development. It is studied by control engineers in the very beginning of its education and has a significant meaning in industry. MATLAB/SIMULINK became a universal tool for solving many automation and control tasks nowadays, and presence of a graphical block diagram programming principle in it allows building complicated systems in a convenient and intuitive way. The ability to convert block diagram representation to C++ code is a powerful feature which allows implementing user's code to any target system and make solution of many control tasks simpler. Due to the lack of existing clear and step-by-step instructions of how to use MATLAB/SIMULINK model to control Beckhoff hardware the research work reflected is this thesis is done. The thesis presents the steps, tips and instances of how to use Beckhoff Embedded PC in actual hydraulic system as an industrial PLC and a RTDAQ computer which is programmed by the means of MATLAB/SIMULINK avoiding coding in TwinCAT.

Witherspone in his work (2014) describes the hydraulic rig control algorithm development and implementation using LabVIEW software as programming environment, dSPACE as industrial controller hardware and ControlDesk as real-time control software. In his thesis he provides the laboratory hydraulic rig description and mathematical model. Current master thesis work is using common principles of system's mathematical modeling and control but in different vision with different hard- and software.

The control algorithm implementation and test results allows to conclude that programming such a powerful industrial PC as CX-Series Beckhoff Embedded PC might be effortless, does not require special training and has an ideal transparent mapping between human, machine and task environment interaction.

1.3 Purposes of Work

The purpose of this thesis is twofold. First, it is to make a clear representation of how the Embedded PC made by Beckhoff Automation Company can operate using the program code, built as a function block diagram in MATLAB/SIMULINK software, and provide a several program implementation instances with detailed explanations which can be used as instructions of MATLAB/SIMULINK and TwinCAT 3 software integration. Testing a simple digital actuator and Beckhoff DC-servomotor served as these instances. Second, it is to develop the control algorithm for hydraulic rig which performs as a 1-D manipulator, and to implement this algorithm using Beckhoff Embedded PC which operation is based on MATLAB/SIMULINK block diagram code. After hardware setup and control scheme implementation several experiments and tests are conducted in order to study the performance of the system.

1.4 Research Scope

This thesis covers control and automation engineering, programming environments, hardware description and installation, real-time simulation, hydraulics, hydraulic system manipulation. Explanation of these topics is based on using Beckhoff hard-and software, MATLAB/SIMULINK software and hydraulic test rig.

Typically, Beckhoff hardware uses TwinCAT software system as programming and operating station, but as it was developed exclusively for Beckhoff hardware, it has several drawbacks and does not typically used as a powerful and convenient control algorithm development environment for different hardware systems. Meanwhile, MATLAB/SIMULINK is more widely used in automatic control and digital signal processing and has a graphical block diagramming tool, which is intelligible for all kind of users and does not require deep knowledge of any of programming languages or specific PLC operation principles. Combining MATLAB/SIMULINK with the Beckhoff hardware is the key point of this work.

The research work was split into three general parts. First, Beckhoff Automation Company hardware, such as CX-Series Embedded PC and DC-servomotor, and TwinCAT 3 software system were studied. Then TwinCAT 3 and MATLAB/SIMULINK interaction was studied and tested. Finally, hydraulic servo-system which is presented by a hydraulic slider with 1 DOF was investigated and its control scheme was developed and implemented.

1.5 Thesis Outline

The thesis consists of eight main chapters. The first chapter introduces the topic of the work and provides the general idea of the research.

The second chapter introduces Beckhoff as a company, describes its hardware which is used in the frames of this work and TwinCAT software system, presents the principle of TwinCAT 3 and MATLAB/SIMULINK interaction.

The third and the forth chapters are Case Study 1 and Case Study 2 respectively, where some of the Beckhoff hardware, MATLAB/SIMULINK block diagram development and importing it into TwinCAT 3, Real-Time hardware control and operation principles are studied. These chapters can also be used as instructions of how MATLAB/SIMULINK code is integrated into TwinCAT 3 and used for Beckhoff hardware.

The fifth and the sixth chapters are dedicated to hydraulic test rig and its control algorithm development and implementation. The fifth chapter is a presentation of the rig, its main parts description and mathematical model. The sixth chapter is the practical work description, which includes control algorithm development, hardware and software connection.

Chapter number seven includes hydraulic rig control scheme test results. Discussion and conclusions of the work are given in chapter eight.

2. BECHOFF EMBEDDED PC

2.1 Beckhoff Automation

Beckhoff Automation GmbH is a German manufacturer of automation technologies equipment of various power classes both in the form of system solutions and individual components.

The company focuses on PC-compatible control equipment, industrial PCs, embedded PCs, I/O modules (bus terminals), servomotors technology and automation software. In 2003 Beckhoff introduced EtherCAT technology to manage real-time systems by the means of Ethernet network.

The headquarters of Beckhoff is located in Verl, Germany and integrates the central departments of development, production, management, sales, marketing, support and service.

The graph of Beckhoff sales for the past 35 years (Figure 2.1) shows that the company is experiencing a rapid grow nowadays and can compete with the world leaders of automation technologies manufacturers.

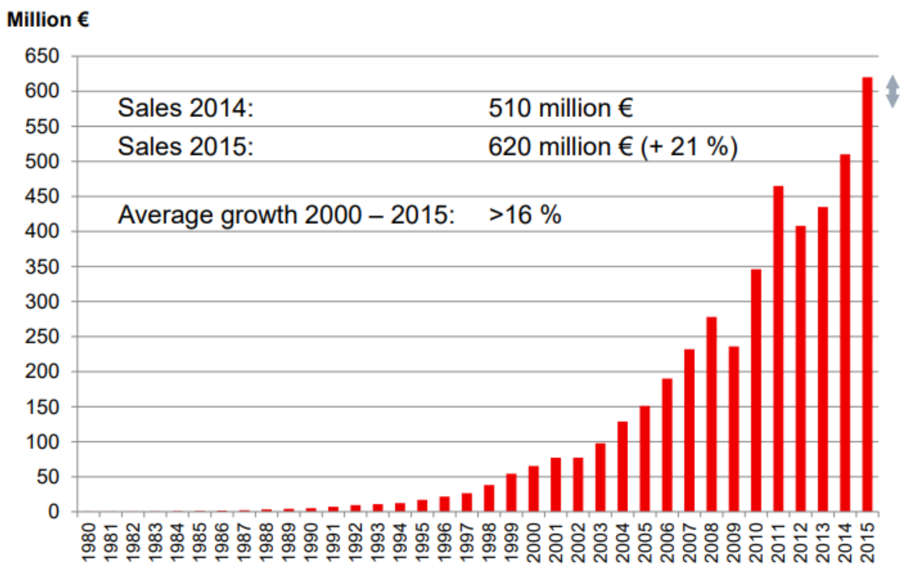

Figure 2.1. Beckhoff sales 1980 – 2015 (Beckhoff Automation GmbH).

Beckhoff is a pioneer in the development of many fundamental areas of modern automation. The main areas of application of Beckhoff products are power engineering, process control systems in the oil, gas and chemical industries, transport systems, boiler rooms, wind power, automation of factories, plants, buildings and utilities. Since 2000, the automation segment of buildings has grown at a rapid pace in both the private and corporate sectors.

Beckhoff produces four main types of products: industrial computers, I/O terminal systems, servomotors and TwinCAT software system.

Industrial PCs

The company is the second largest in Germany in terms of industrial computers production volumes. The increased volumes of production allowed organizing its own production of motherboards in Germany.

The range of industrial computers consists of:

- embedded industrial PCs and servers;
- panel PCs and control panels;
- compact modular computers with direct I/O connection;
- motherboards for industrial computers.

A series of PC-compatible controllers of the Embedded PC series differ from each other in processor performance, from simple processors such as ARM to powerful Intel Core processors. They are designed to solve automation tasks of varying complexity from managing the building's engineering systems and home automation to managing large technological lines with the number of input-output signals measured in thousands. Industrial PCs of different design with a wide variety of additional functions are equipped with high-performance components based on open standards.

I/O systems

One of the main fields of Beckhoff is the development and manufacturing of I/O systems. Modular I/O systems include more than 400 interfaces and provide a high density of installation.

Beckhoff produces four types of I/O systems:

- Bus Terminals (classical I/O system with the large number of signal types);
- EtherCAT I/O (a new generation Ethernet-based fieldbus system);

- Fieldbus Box (system for severe operating conditions - corresponds to protection class IP67);
- Lightbus (an optical system that is resistant to strong electromagnetic interference and interference in industrial environments).

Servomotor Technology

The Beckhoff servomotor technology combines the Motion Control solutions implemented in the TwinCAT automation.

Beckhoff provides a complete set of components for positioning systems creating:

- Servo amplifiers;
- Synchronous servomotors;
- Linear servomotors;
- Stepper motors;
- Planetary gearboxes.

2.2 CX-Series Embedded PCs

Beckhoff Embedded PCs are fully-functional PCs with the functionality of industrial PLCs. In this thesis, Beckhoff Embedded PC CX-series is considered.

The Embedded PC CX-series combines PC technology and a modular I/O system. These PCs are usually installed on a DIN-rail in the control cabinets. Combining the properties of industrial PCs and hardware PLCs makes a series of CX devices suitable for managing tasks in a wide range of performance. A modular system of I/O interfaces allows to create the appropriate control task for the PLC configuration, changing the number of I/O blocks and installing only the necessary components for the control system.

From Beckhoff Automation Embedded PC catalogue: “The CX family covers the whole range of Beckhoff control technology in terms of both price and performance. This product range is designed for tasks requiring the characteristics and computing capacity of Industrial PCs, but whose budget does not stretch to full-blown Industrial PCs”.

The CX-series PCs performance classes include several basic modules with different processors to adapt to the corresponding control task. In the framework of this thesis, Embedded PC CX2030 with a dual-core Intel Core i7 processor with a frequency of 1.5 GHz is considered.

In addition to the type of processors, the CX-series PC family also differs in system interfaces and power supply units. With the help of various interfaces, embedded PCs support communication via different I/O systems, such as bus terminals and EtherCAT. The selection of the CX controller configuration is usually based on the automation algorithm complexity. The transition from one CX family PC to another, for example, with higher performance, is possible even on a later stage of the control system implementation and does not require any modification of the control algorithm.

Embedded PC CX-series configuration and management are carried out using the TwinCAT software that uses IEC 61131 international standard, which describes the programming languages for the industrial PLCs. CX-series controllers are simultaneously PLCs and PCs based on Windows operating. Due to the presence of the preinstalled OS Windows, user tasks written in high level programming languages are processed in the TwinCAT environment in real time.

2.2.1 CX2030 Embedded PC

In this work Beckhoff CX2030 Embedded PC is used as a RTDAQ Computer and control algorithm implementation hardware.

The CX2030 (Figure 2.2) has a 1.5 GHz Intel Core i7 dual-core CPU.

From Beckhoff Automation Embedded PC catalogue: “The basic configuration of the CX2030 includes a CFast memory card, two independent Gbit Ethernet interfaces, four USB 2.0 interfaces and a DVI-I interface. The CPU has a 128 kB NOVRAM persistent data memory for situations where no UPS is used. The operating system is Microsoft Windows Embedded 7. TwinCAT automation software transforms a CX2030 system into a PLC and Motion Control system that can be operated with or without visualization.

The power supply for the CPU module comes from a CX2100 power supply module, which is connected on the right-hand side of the CPU".

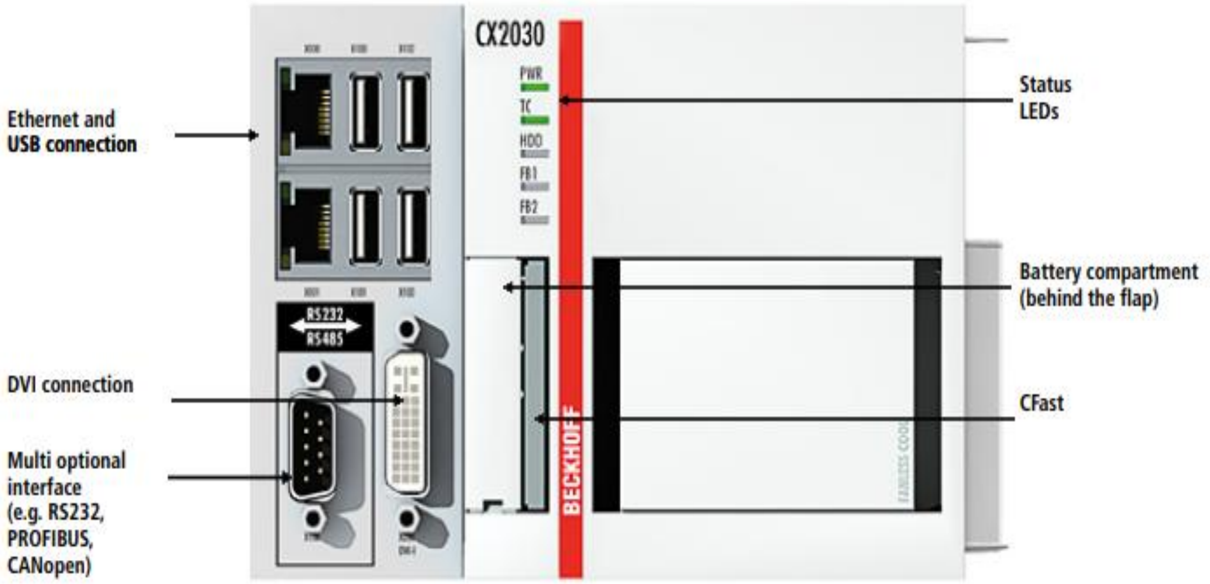

Figure 2.2. Beckhoff CX2030 Embedded PC (Beckhoff Automation).

Technical data of the CX2030 is shown in Table 2.1.

Table 2.1. Beckhoff CX2030 Embedded PC technical data. (Beckhoff Automation)

Parameter	**Characteristics**
Processor	Intel Core i7 2610UE 1.5 GHz, 2 cores
Flash memory Internal main memory	4 or 8 GB CFast card (optionally expandable) 2 GB DDR3 RAM (optionally expandable)
Persistent memory	128 KB NOVRAM integrated
Interfaces	2 x RJ45, 10/100/1000 Mbit/s, DVI-I, 4 x USB 2.0, 1 x optional interface
Diagnostics LED	1 x power, 1 x TC status, 1 x flash access, 2 x bus status
Clock	internal battery-backed clock for time and date (battery exchangeable)
Operating system	Microsoft Windows Embedded Compact 7
Control software	TwinCAT 2 runtime; TwinCAT 3 runtime (XAR)
I/O connection	via power supply module (E-bus or K-bus, automatic recognition)
Power supply	24 V DC (-15 %/+20 %)
Max. power loss	20 W (including the system interfaces)
Dimensions (W x H x D)	144 mm x 100 mm x 91 mm

Table 2.1 continues. Beckhoff CX2030 Embedded PC technical data. (Beckhoff Automation)

Weight	approx. 1165 g
Operating/storage temperature	-25…+60 °C/-40…+85 °C
Relative humidity	95 %, no condensation
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4
Protection class	IP 20
TC3 performance class	mid performance (60)

2.3 TwinCAT 3

For programming logic controllers, controlling positioning of various complexity, configuring I/O devices and for creating visualization interfaces, Beckhoff developed TwinCAT software.

According to Beckhoff Automation TwinCAT manuals, TwinCAT 3 highlights are:

- one software for programming and configuration;
- Microsoft Visual Studio integration;
- freedom in selecting programming languages;
- compatibility with the widespread standard CODESYS;
- support for the object-oriented extension of IEC 61131-3;
- use of C/C++ as the programming language for real-time applications;
- link to MATLAB/SIMULINK;
- open interfaces for expandability and adaptation to the tools landscape;
- flexible runtime environment;
- active support of multi-core and 64-bit systems;

Beckhoff Automation: “TwinCAT software system turns almost any PC-based system into a real-time control with multiple PLC, NC, CNC and/or robotics runtime systems. TwinCAT 3 is the systematic further development of TwinCAT 2.”

TwinCAT 3 software interface example is presented in Figure 2.3

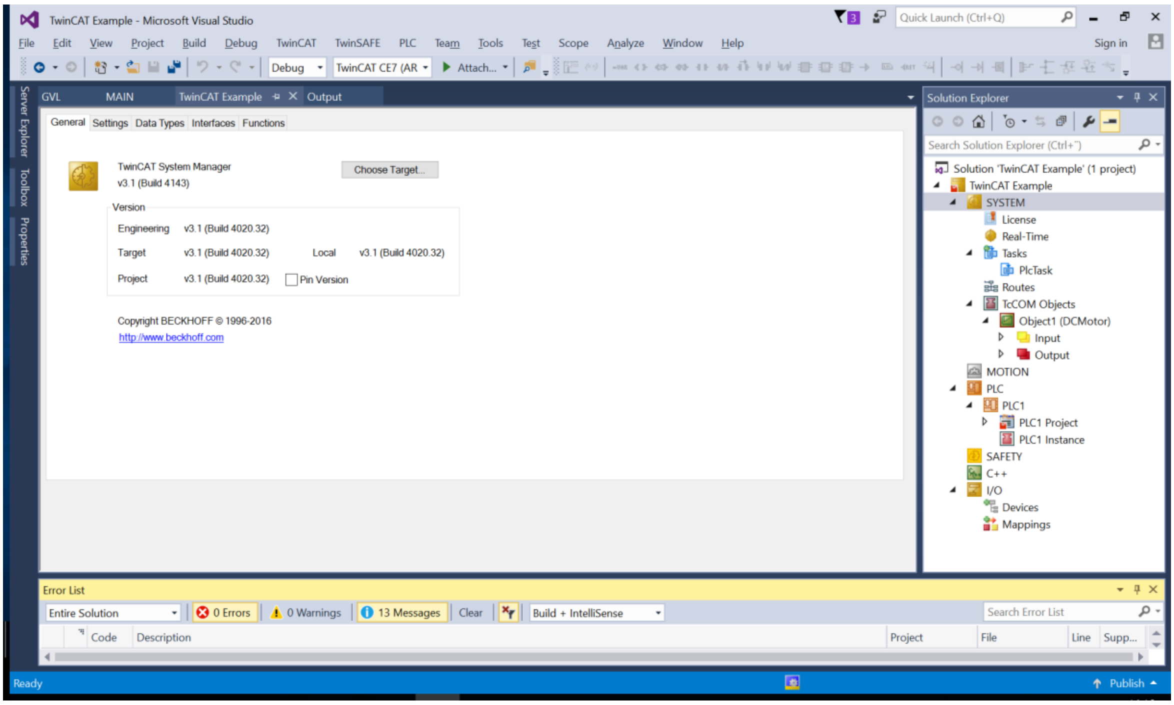

Figure 2.3. TwinCAT 3 interface instance.

TwinCAT 3 offers the possibility to program TwinCAT runtime modules in C/C++ languages. TwinCAT 3 integrates itself into the existing Microsoft Visual Studio Shell in order to use the programming language C or C++ in addition to I/O configuration and the IEC 61131-3 for real-time tasks. Integration in Microsoft Visual Studio makes it possible to program automation objects in parallel with the aid of the 3rd edition of IEC 61131-3 and the C or C++ languages. The objects (modules) generated can exchange data with each other and call each other independently of the language they were written in. The TwinCAT System Manager has been integrated into the development environment. This way just one software is required to configure, parameterize, program and to diagnose automation devices. By integrating TwinCAT 3 as an extension into the Visual Studio, it is possible to provide expandable programming platform. (Beckhoff Automation 2017).

2.4 TwinCAT 3 Interface and Target for MATLAB/SIMULINK

In addition to the classic PLC programming languages of the IEC 61131-3 it is possible also to program with the high-level languages C and C++, as well as MATLAB/SIMULINK.

According to Beckhoff Automation (2014): “The integration of MATLAB/SIMULINK enables execution of TwinCAT modules that were generated as models in the Simulink simulation environment. The chosen interfacing type displays the parameters and variables in the graphic interface of TwinCAT 3 and enables viewing and modification in the real-time environment at runtime. «TE1400 Target for MATLAB/SIMULINK» is used to generate real-time capable modules from MATLAB/SIMULINK, which can be executed in the TwinCAT 3 runtime. These modules can be instantiated, parameterized and debugged multiple times in the TwinCAT 3 environment. The Simulink Coder (previously called Real-Time Workshop) for the MATLAB/SIMULINK environment contains a code generator, which can generate corresponding C/C++ code from the model in Simulink. The TwinCAT «Target for MATLAB/SIMULINK» works on the basis of this code generator. If the code generator is configured appropriately with TwinCAT Target, a TwinCAT Object Model (TcCOM) is created with the input and output behavior of the Simulink model.”

In order to use TwinCAT Target for MATLAB/SIMULINK, «TE1400 Target for MATLAB/Simulink» module has to be installed on development platform.

TwinCAT 3 and MATLAB/SIMULINK integration principle scheme is presented in Figure 2.4.

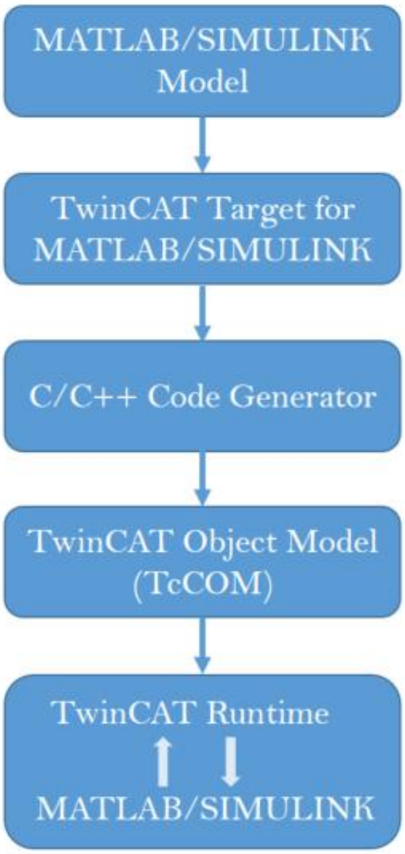

Figure 2.4. TwinCAT and MATLAB/SIMULINK integration principle.

3. CASE STUDY 1

3.1 Module Description

The first case study includes TwinCAT 3 and MATLAB/SIMULINK integration testing by executing of ON/OFF tasks using Beckhoff EL3104 Input Terminal with a cable extension sensor connected to it, and EL2024 Output Terminal with a cooling fan as an output device (actuator). The task was to build a block diagram in MATLAB/SIMULINK which presents the following algorithm: when the position sensor cable is half extended, the fan turns on and rotates until the cable is extended on less than the half of its length (Figure 3.1).

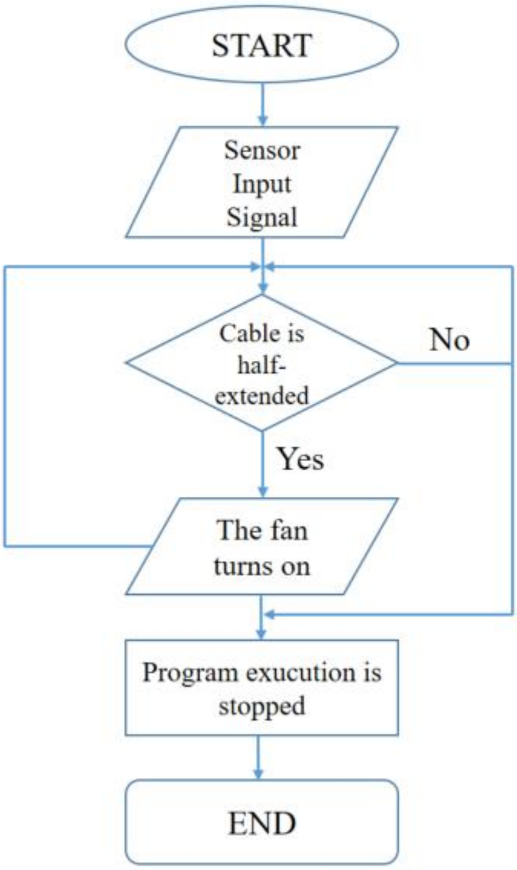

Figure 3.1. Case study 1 algorithm diagram.

The hardware which was used for this test is described below.

3.1.1 Power Supply Unit

To supply power to the components of a CX2030 system, CX2100-0004 Power Supply unit is used (Figure 3.2). The power supply units are controlled via TwinCAT.

Figure 3.2. CX2100-0004 Power Supply module (Beckhoff Automation).

Technical data of CX2100-0004 is presented in Table 3.1.

Table 3.1. CX2100-0004 Technical Data. (Beckhoff Automation)

Parameter	**Characteristics**
Power supply	24 V DC (-15 %/+20 %)
Dielectric strength	Dielectric strength
Max. output	45 W
I/O connection	E-bus (EtherCAT Terminals) or K-bus (Bus Terminals), automatic detection
Power supply I/O terminals	2 A
Connection type	Spring force technology (adapter terminal)
Display	FSTN display 2 lines x 16 characters of text, illuminated
Diagnostics LED	1 x PWR, 1 x I/O Run, 1 x I/O Err
Max. power consumption	3.5 W
Dimensions (W x H x D)	40 mm x 100 mm x 91 mm

Table 3.1 continues. CX2100-0004 Technical Data. (Beckhoff Automation.

Operating/storage temperature	-25...+60 °C/-40...+85 °C
Relative humidity	95 % no condensation
Vibration/shock resistance	Conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity/emission	Conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP 20

3.1.2 Supply Voltage Source

CX2100-0004 power supply unit is equipped with an I/O interface, which allows to establish the connection with I/O terminals. The supply voltage supplies the PC with a voltage of 24 V DC.

For this purpose Balluff BAE0002 24VDC/10A supply voltage block is used (Figure 3.3).

Figure 3.3. 24VDC/10A Supply voltage block.

3.1.3 EL3104 Input Terminal and Cable Extension Sensor

Cable Extension Sensor

In Case Study 1 ASM POSIWIRE WS10-1000-10V Cable Extension Position Sensor was used as an input device (Figure 3.4).

Technical data of the sensor is presented in Table 3.2.

Figure 3.4. Cable Extension Position Sensor ASM POSIWIRE WS10-1000.

Table 3.2. WS10-1000-10V Sensor Technical Data.

Parameter	**Characteristics**
Measurement range	0-1000 mm
Output	Voltage 0 … 10 V
Sensing device	Precision potentiometer
Cable forces: Maximum pull-out force Minimum pull-in force	 5.3 N 2.9 N
Resolution	Analog: quasi infinite
Linearity	±0.10% f.s.
Connection	Connector M12, 8 pin
Housing material	Aluminum stainless steel and plastic measuring cable: stainless steel
Protection class	IP65
Temperature range	-20 ... +85 °C
Weight	approx. 550 g
EMC	DIN EN 61326-1:2013

EL3104 Input Terminal

To read the data from Cable Extension Sensor Beckhoff EL3104 analog input terminal is used (Figure 3.5).

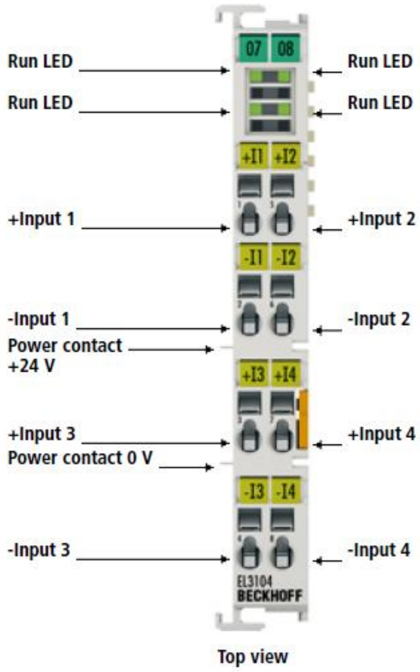

Figure 3.5. EL3104 Input terminal (Beckhoff Automation).

Beckhoff Automation: "The EL3104 analog input terminal processes signals in the range between -10 and +10 V. The voltage is digitized to a resolution of 16 bits, and is transmitted to the higher-level automation device. The input channels of the EL3104 EtherCAT Terminal are differential inputs. The signal state of the EtherCAT Terminal is indicated by light emitting diodes". Technical data of the terminal is presented in Table 3.3.

Table 3.3. EL3104 Terminal Technical Data. (Beckhoff Automation)

Parameter	**Characteristics**
Number of inputs	4
Power supply	via the E-bus
Technology	differential input
Signal voltage	-10…+10 V
Distributed clock precision	<< 1 µs
Internal resistance	> 200 kΩ
Conversion time	~ 100 µs
Resolution	16 bit (incl. sign)
Measuring error	< ±0.3 % (relative to full scale value)
Configuration	no address or configuration setting
Current consumption E-bus	typ. 130 mA
Weight	approx. 65 g
Protect. class/installation pos.	IP 20/variable

3.1.4 EL2024 Output Terminal and Simple Actuator

Actuator

A simple axial cooling fan was used as an output device in the preliminary testing of Case Study 1 (Figure 3.6).

Figure 3.6. Simple Actuator – Cooling Fan.

The fan technical data is presented in Table 3.4.

Table 3.4. Cooling fan Technical Data.

Parameter	**Characteristics**
Rating voltage	24 V DC
Power current	760 mA
Power consumption	18.2 W
Speed	4200 RPM
Air flow	190.0 CFM
Noise	54.0 dB
Weight	330 g
Size	Square – 120 mm L x 120 mm H

EL2024 Output Terminal

The fan is connected to the digital output terminal EL2024 (Figure 3.7). The EL2024 digital output terminal connects the binary 24 V control signals with the fan. Technical data of the terminal is presented in Table 3.5

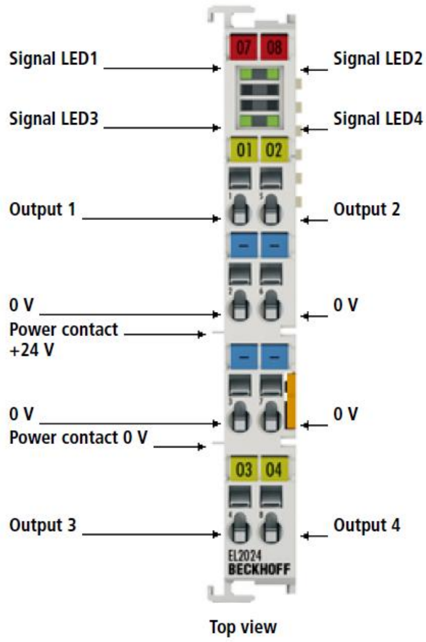

Figure 3.7. EL2024 Output terminal (Beckhoff Automation).

Table 3.5. EL2024 Terminal Technical Data. (Beckhoff Automation)

Parameter	**Characteristics**
Connection technology	2-wire
Number of outputs	4
Rated load voltage	24 V DC (-15 %/+20 %)
Load type	ohmic, inductive, lamp load
Max. output current	2.0 A (short-circuit-proof) per channel 2.0 A
Short circuit current	typ. < 70 A
Breaking energy	< 1.7 J/channel
Switching times	typ. TON: 40 µs, typ. TOFF: 200 µs
Current consumption E-bus	typ. 120 mA
Configuration	no address or configuration setting

Weight	approx. 55 g
Protect. class/installation pos.	IP 20/variable

3.2 Methods

3.2.1 Hardware Connection

EL3104 analog input terminal and EL2024 digital output terminal are connected to CX2100 power supply unit, which is in turn connected to CX2030. All devices, including 24 V DC source, are installed on a DIN rail. The view of the module is presented in Figure 3.8.

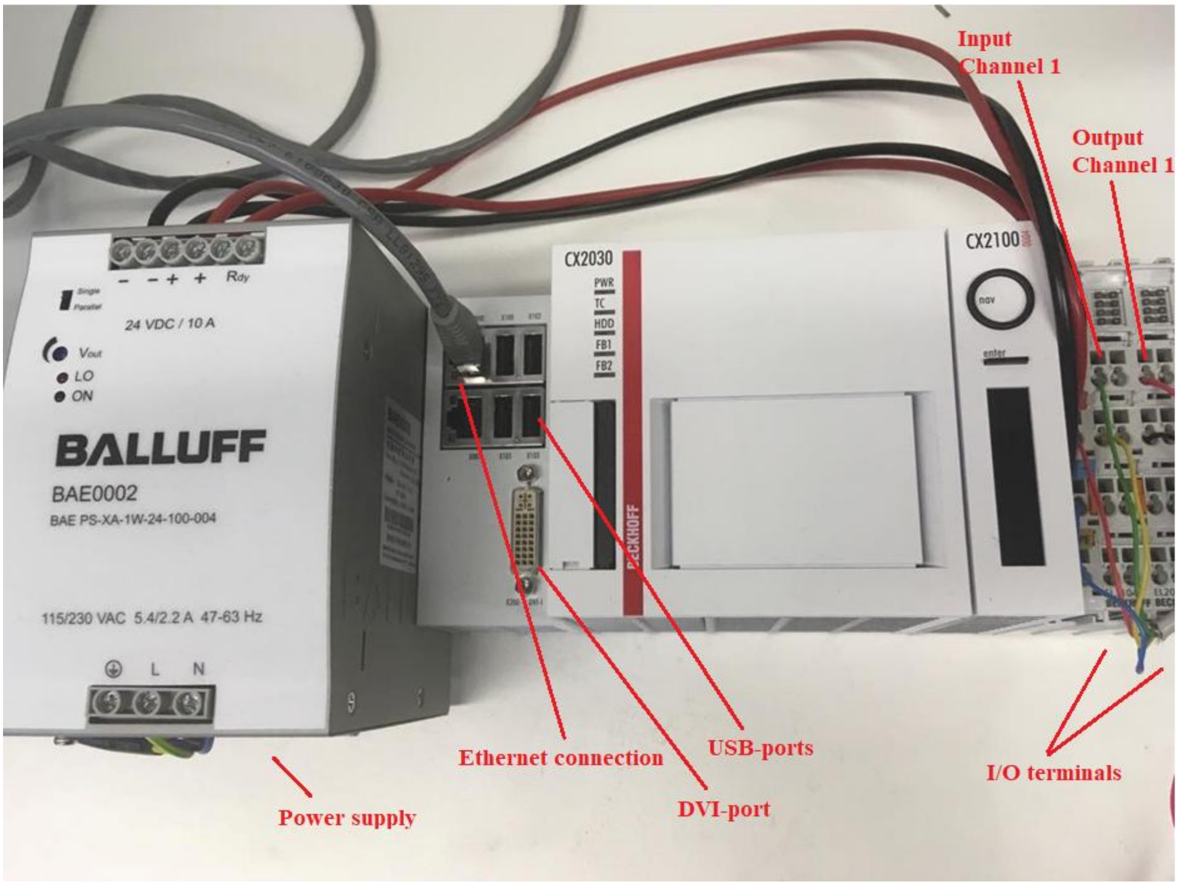

Figure 3.8. Beckhoff Embedded PC Module.

CX2030 Embedded PC is connected to the remote computer via Ethernet cable (RJ45). After IP-address configuration (both platforms must be in the same subnet), the route between CX2030 and computer is configured by right-clicking in Windows Toolbar on *TwinCAT 3 icon -> Router -> Edit Routes -> Add* (Figure 3.9). The device route is then added and the connection is established.

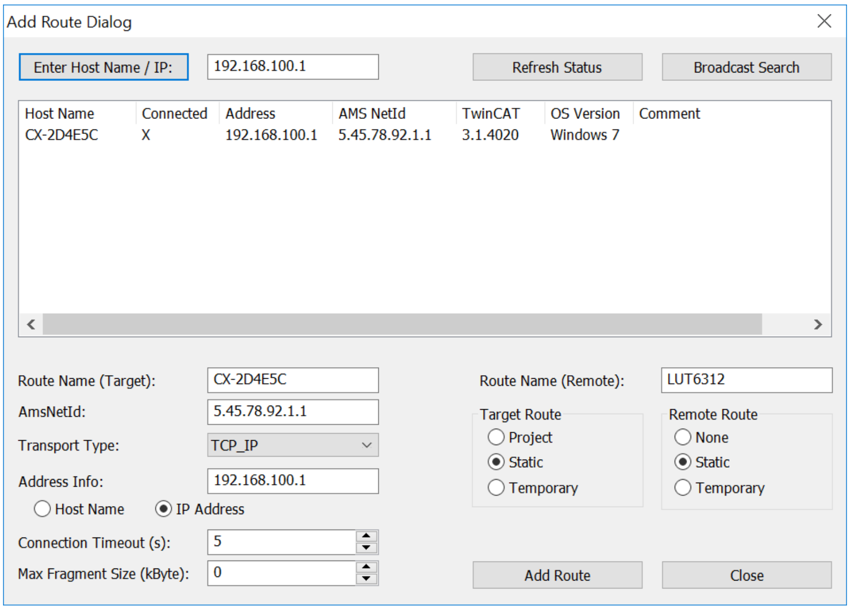

Figure 3.9. Add Route Dialog Window.

3.2.2 MATLAB/SIMULINK Model

Block Diagram

In this work MATLAB R2016b software version is used.

In order to execute the program, presented in «3.1 Module Description», the following Block Diagram was built in MATLAB/SIMULINK environment:

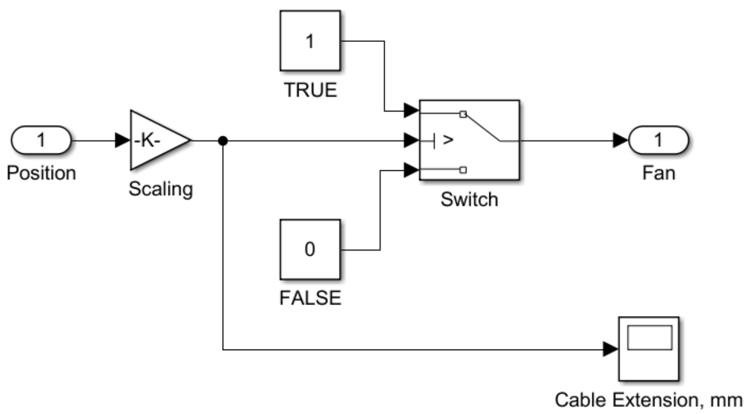

Figure 3.10. Preliminary Test MATLAB/SIMULINK Block Diagram.

In order to be linked to hardware I/O, the input and output block must have proper data types. It can be configure in *Block Parameters -> Signal Attributes* tab. The data type of “Position” Input block is int16 (INT in IEC61131-3); the data type of “Fan” Output block is Boolean.

If the input signal from the position sensor is more than 0, the Boolean 1 (TRUE) is sent the output, which activates the fan. Otherwise, the Boolean 0 (FALSE) is sent to the output. The Gain block converts position sensor input signal to the cable extension of it in millimeters.

MATLAB/SIMULINK Model Configuration

MATLAB/SIMULINK model has to be configured before integration with TwinCAT 3 environment. This configuration takes place in *Simulation -> Model Configuration Parameters* tab (pressing Ctrl+E).

The coder settings can be accessed via Code Generation menu. In Code Generation window *TwinCAT.tlc* as «System target file» is selected (Figure 3.11).

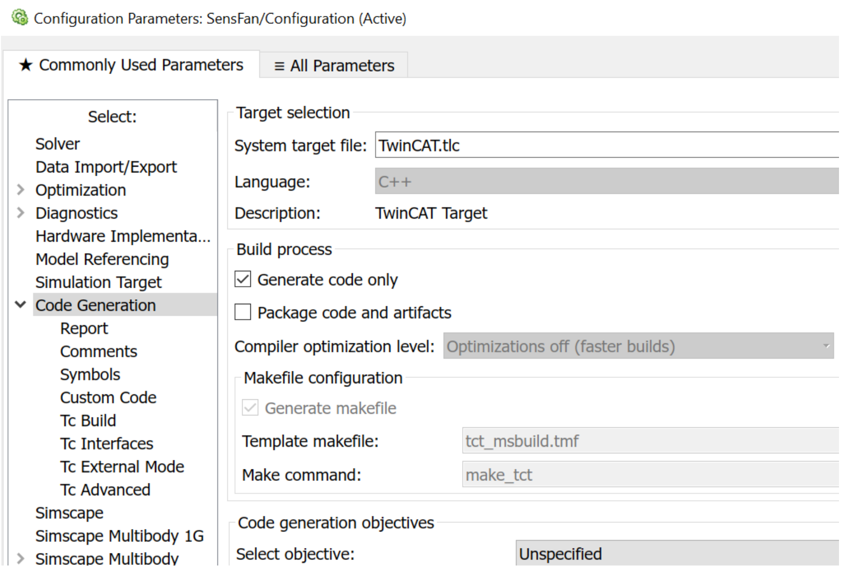

Figure 3.11. MATLAB/SIMULINK target system selection window.

In addition, a fixed-step solver must be configured in the solver settings, to ensure real-time capability of the Simulink model (Figure 3.12).

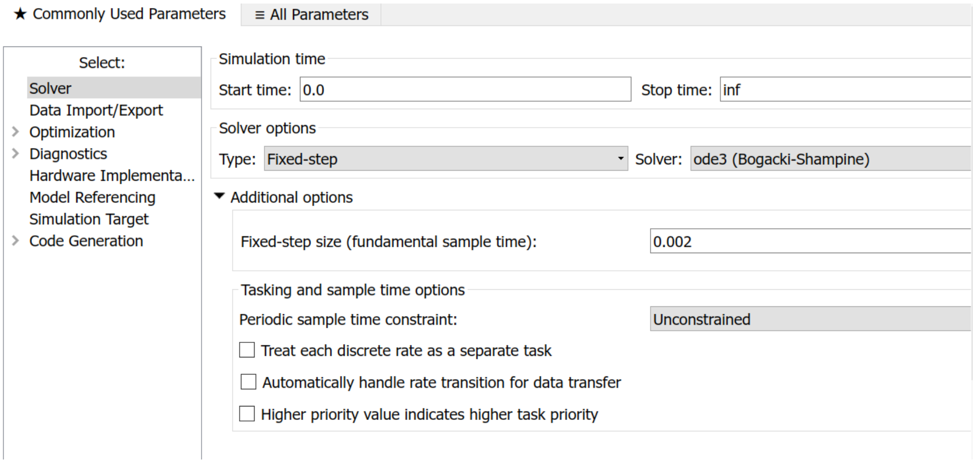

Figure 3.12. Fixed-step Solver Configuration.

Generating a TcCOM module from Simulink

Generation of the C++ code or the TcCOM module can be started with the *Build* button (or *Generate code*) in the lower section of the window for the code generator options.

The module generator stops after the C++ code and the project file for Visual Studio has been generated.

3.2.3 TwinCAT 3 Configuration

Target System

In order to address the TwinCAT runtime environment as development environment remotely, the target system must be made known first.

After creating of New XAE Project, project folders are available in Solution Explorer.

Clicking on *SYSTEM -> Choose Target…* tab opens the window:

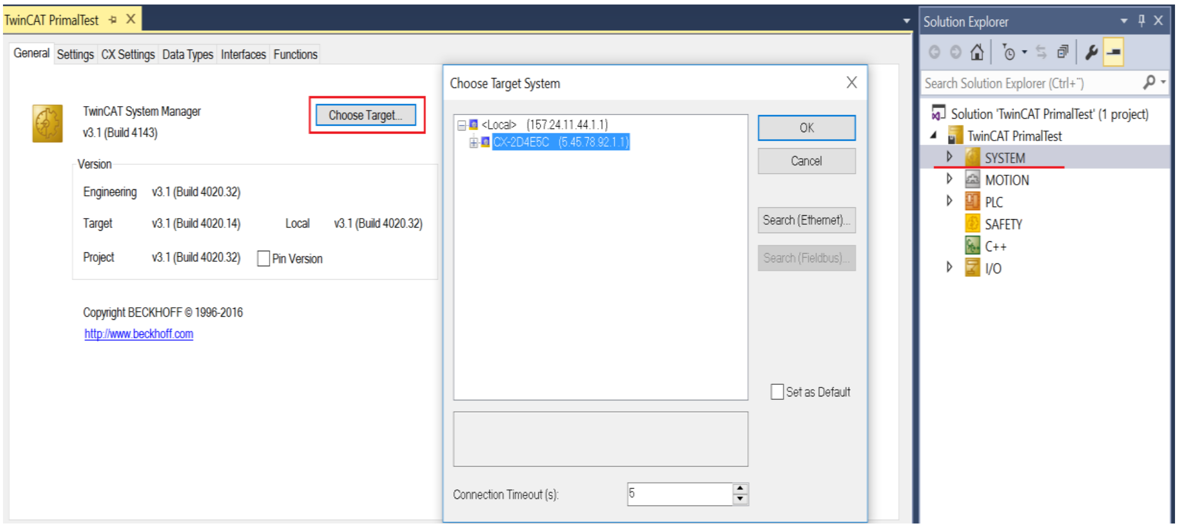

Figure 3.13. Selection dialog: Choose Target System.

After pushing the “OK” button the target system is accessed via the Visual Studio shell.

Adding Devices

To add connected to the CX2030 I/O terminals next steps are followed:

In the project folder explorer of the user interface on the left, «*Devices*» within element “I/O” are selected; then a context menu of it is opened and «*Scan*» is clicked. The TwinCAT System Manager is first set to «Config mode» via or via the menu «*TwinCAT*» → «*Restart TwinCAT (Config mode)*».

After clicking on “Scan” and confirmation the warning message, new devices are found, and “Device 1 (EtherCAT)” is selected. The message «Find new boxes» is confirmed in order to determine connected terminals. «Free Run» mode enables manipulation of I/O values in «Config mode».

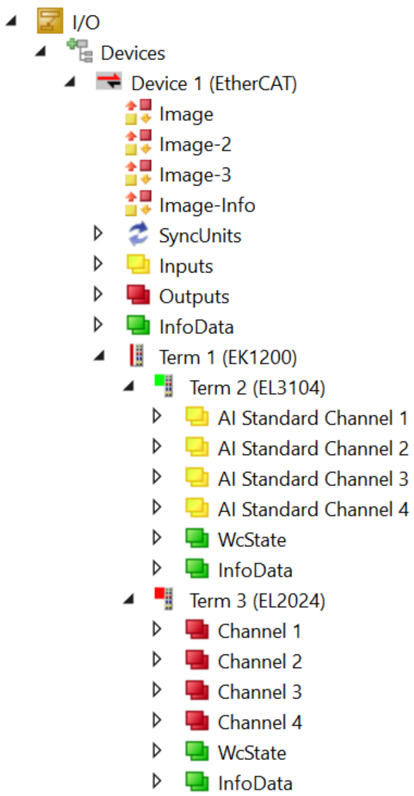

Figure 3.14. Added Devices and Terminals

As can be seen in the Figure 3.14, TwinCAT 3 added EK1200 EtherCAT Power supply as a device an then EL3104 analog Input and EL2024 digital output terminals as boxes.

The C++ project is inserted into the TwinCAT project via *Add Existing Item* in the context menu of the C++ node. The C++ project file is located in the Build directory «<MODELNAME>_tct» and has the name of the module with the file extension *.vcxproj*. The module can then be created in the TwinCAT development environment (XAE).

Adding generated MATLAB/SIMULINK Model

Generated block diagram of MATLAB/SIMULINK model is inserted into the TwinCAT project following *SYSTEM ->TcCOM Objects context menu -> Add New Item***:**

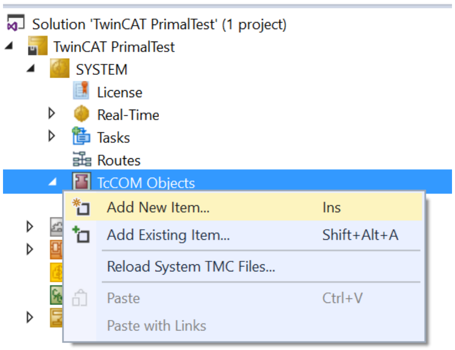

Figure 3.15. Adding New TcCOM Object.

Clicking on "*Add New Item*" opens "Insert TcCOM Object" window. "TE1400 Module Vendor" consists of previously generated in MATLAB/SIMULINK modules. The proper module is then selected (SensFan).

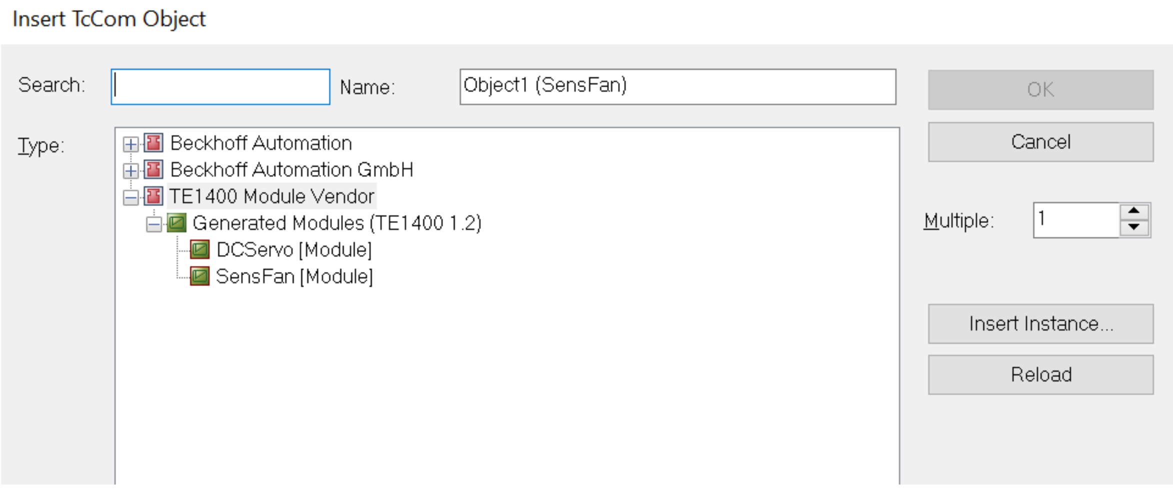

Figure 3.16. Insert TcCOM Object window.

The block diagram, previously built in Simulink, is then able as a TcCOM object in TwinCAT 3 (Figure 3.17).

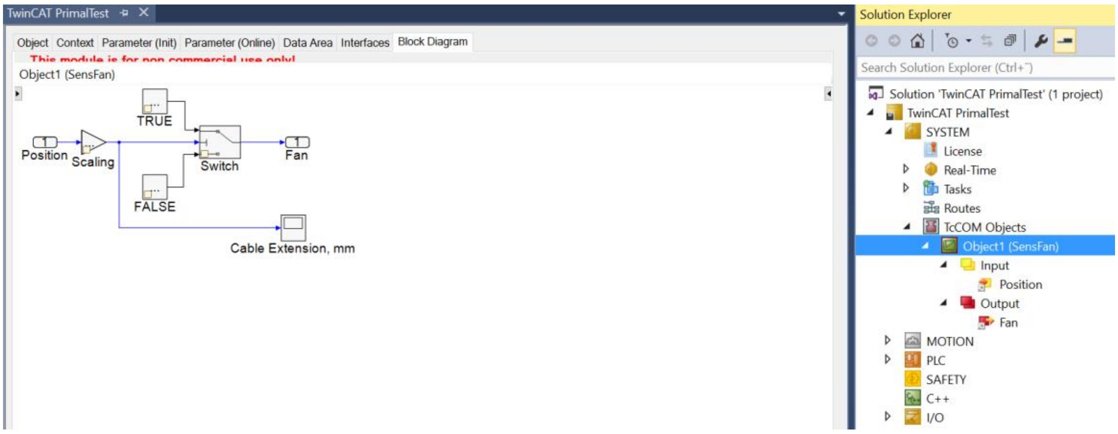

Figure 3.17. MATLAB/SIMULINK block diagram as a TcCOM Object.

Adding a Task

Added TcCOM object requires a proper Task (Figure 3.18) linked to it. To create a Task, next steps are followed:

In Solution Explorer: *SYSTEM ->Tasks context menu ->Add New Item*. The Task is named as Task 1. As the MATLAB/SIMULINK model has a fixed step with a sample time 0.2 s, the amount of cyclic ticks in Task 1 is 200.

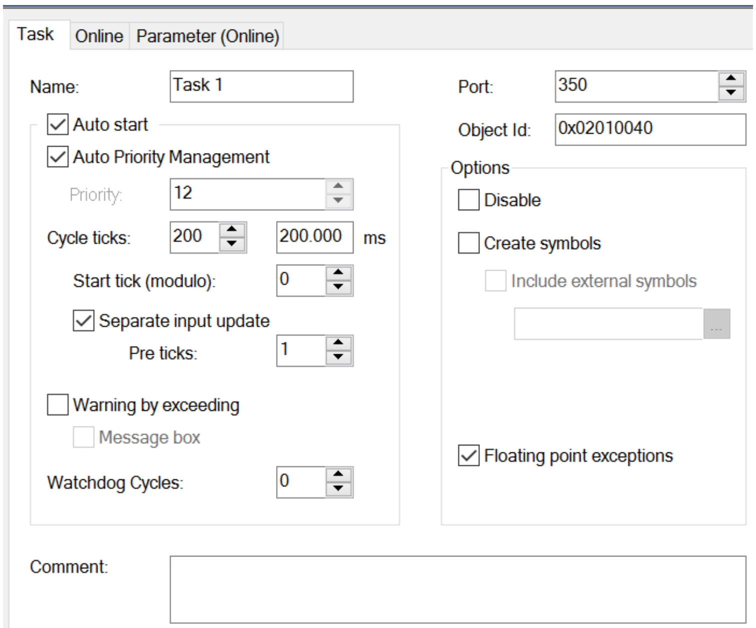

Figure 3.18. Task for TcCOM Object.

Task 1 is then linked to the added TcCOM object in the Context tab:

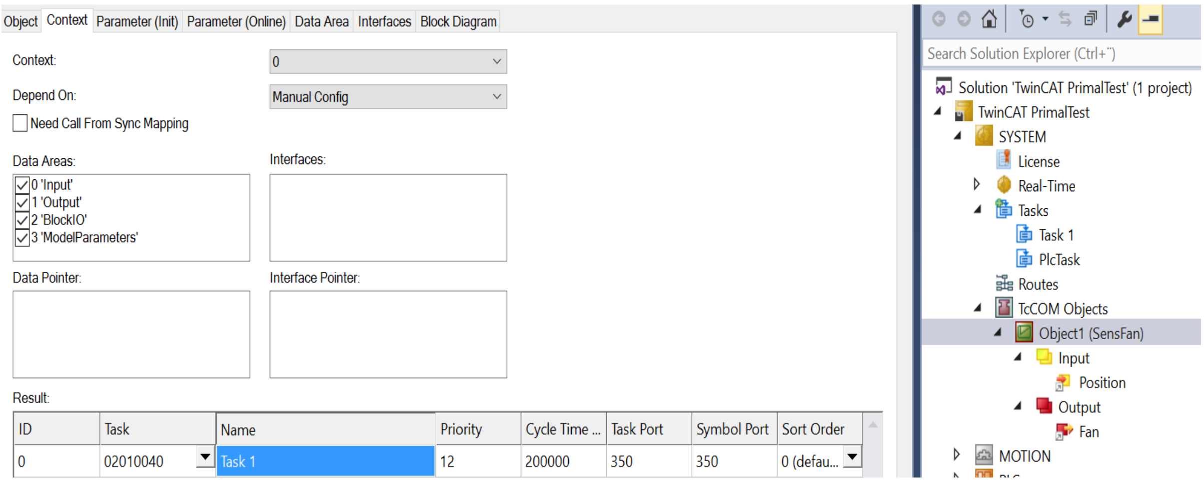

Figure 3.19. Linking the Task to TcCOM Object.

PLC and I/O links

In order to execute the program, a PLC instance is added (*System Manager -> PLC Context Menu -> Add New Item*).

The purpose of PLC in this case is to link I/O data of TcCOM Object (MATLAB/SIMULINK block diagram) and I/O of terminals. Concerning this, two variables are defined in Global Variable List as Input and Output Variables (Figure 3.20).

```
VAR_GLOBAL
    inputgvl AT %I* : BOOL;

    outputgvl AT %Q* : BOOL;
END_VAR
```

Figure 3.20. PLC Global Variables List.

Variable "inputgvl" is linked to the "Fan" output of TcCOM object. Variable "outputgvl" is linked to Channel 1 of EL2024 output terminal.

The MAIN program of PLC´s POU defines the output of EL2024 as an output "Fan"of block diagram:

```
PROGRAM MAIN

GVL.outputgvl:=GVL.inputgvl;
```

Figure 3.21. TwinCAT 3 MAIN program for Case Study 1.

PLC project tree can be seen in Figure 3.22.

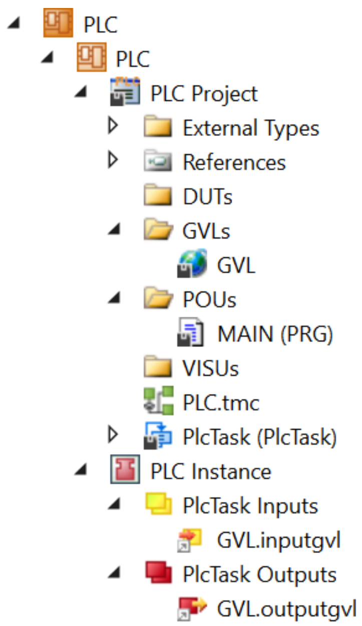

Figure 3.22. PLC tree in TwinCAT 3 System Manager.

The block diagram “Position” input is linked to the Channel 1 of EL3104 Terminal.

Once the PLC Project is built and I/O variables are linked to the I/O of terminals, the program is ready to execution.

3.2.4 External mode and ADS Connection

In addition to “Normal mode”, in which the SIMULINK model is calculated directly in the SIMULINK environment, an “External mode” is available. In this mode SIMULINK acts as a graphical interface without performing any calculations. Once the model with the corresponding settings has been converted into a TcCOM module, SIMULINK is linked to the instantiated TcCOM object that is currently running in the TwinCAT real-time environment

External Mode is activated in the settings for the Simulink Coder under TC External Mode before the module is generated:

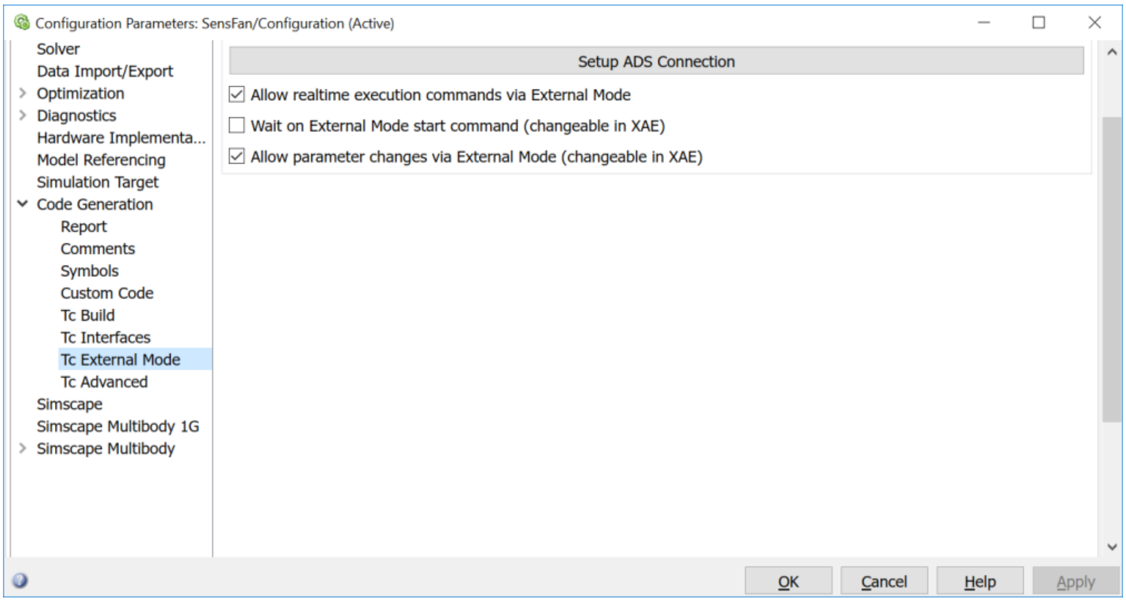

Figure 3.23. MATLAB/SIMULINK External Mode Configuration Window.

The following dialogs are displayed to configure the connection:

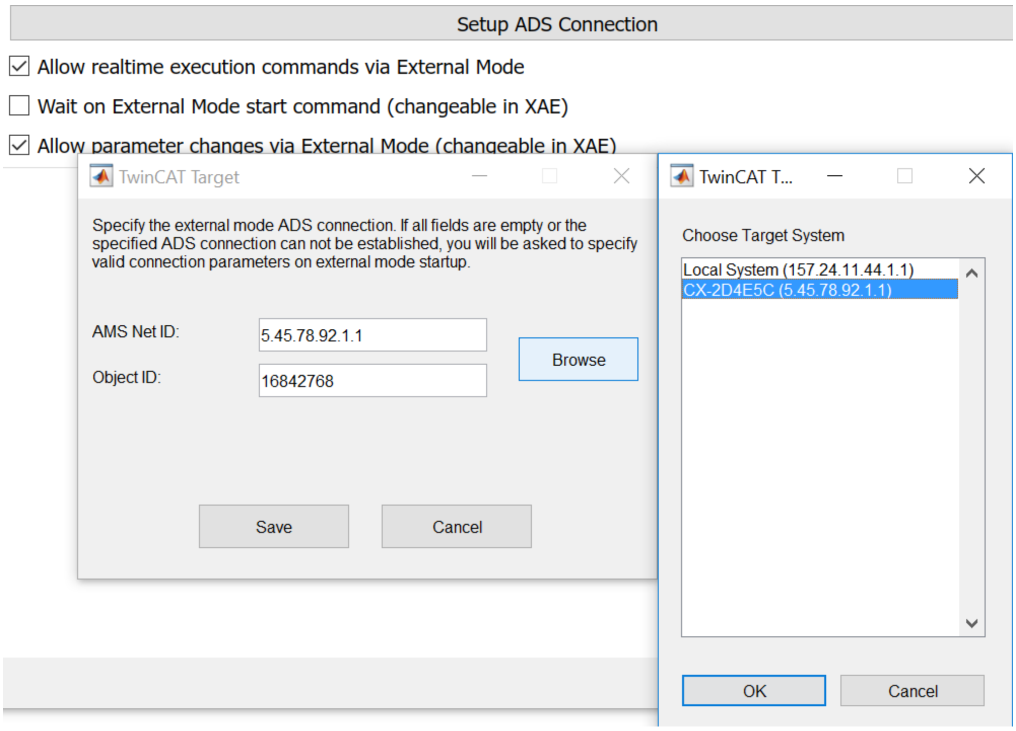

Figure 3.24. ADS Connection Configuration windows.

The «External Mode» connection can be started from Simulink via the *Connect to Target* icon, which appears in the Simulink toolbar when *External* mode is selected:

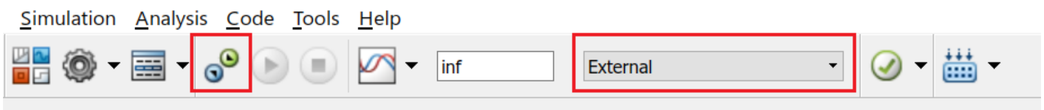

Figure 3.25. Simulink External Mode Selection.

3.3 Results

Once all the configurations are done, TwinCAT 3 Configuration is activated and PLC Project is logged in when the system is in Run mode.

Sensor cable extension (Figure 3.26) changes can be seen both in MATLAB/SIMULINK and TwinCAT 3 block diagram Scopes (Figures 3.27, 3.29).

When the input value is less than 500 (mm), the fan is in inactive state. Once the extension exceeds the value of 500 (mm), the fan turns on.

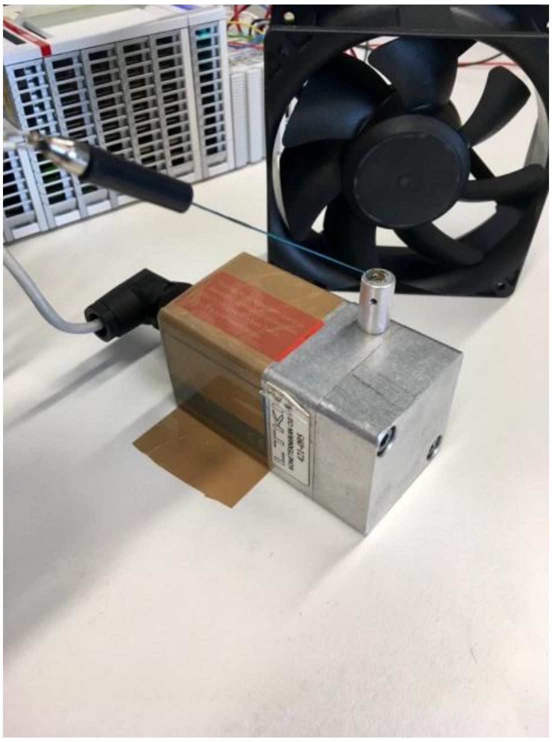

Figure 3.26. Position sensor cable extension.

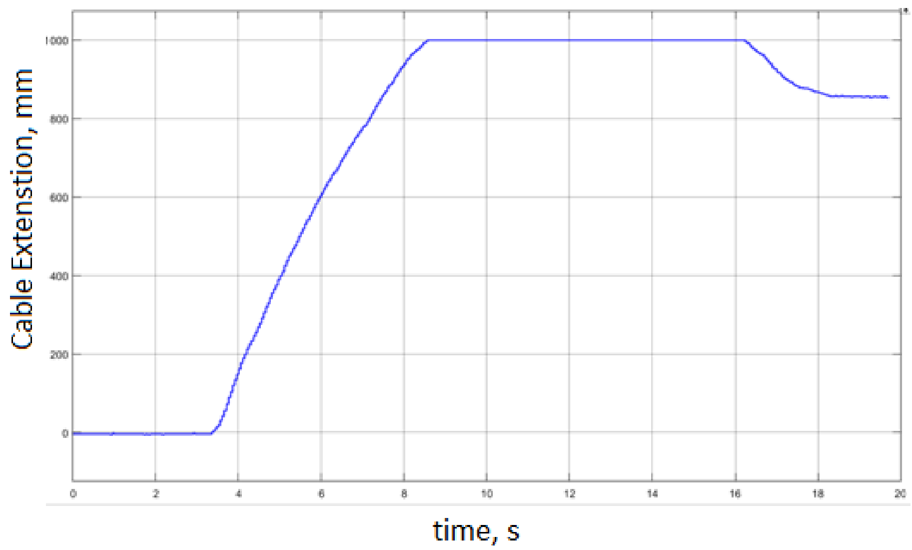

Figure 3.27. Sensor cable extension in MATALB/SIMULINK Scope.

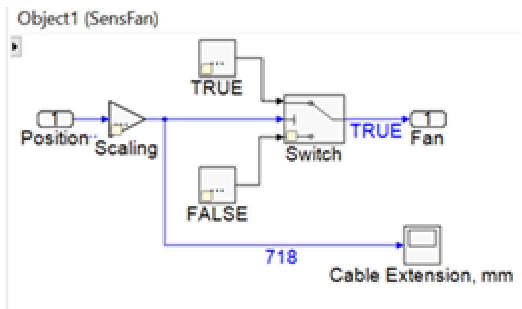

Figure 3.28. TwinCAT 3 block diagram in system run mode.

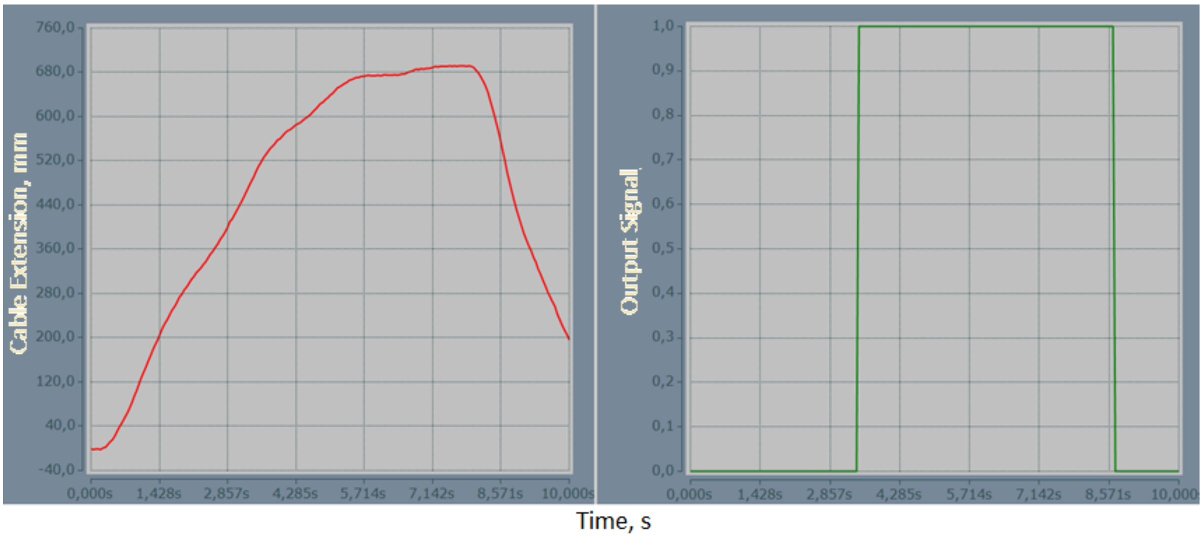

Figure 3.29. TwinCAT 3 input (red) and output (green) charts.

Simulation results show that the program works correctly and the output actuator follows the desired behavior.

4. CASE STUDY 2

4.1 Module Description

In the second Case Study, where TwinCAT 3 and MATLAB/SIMULINK integration is tested additionally, the fan is replaced by Beckhoff DC-servomotor, and servomotor terminal EL7211-9014 is used instead of EL2024 digital terminal. The main task of this test is to develop an algorithm which allows to control the rotational speed of motor's shaft by extension of position sensor cable.

As previously, ASM POSIWIRE WS10-1000-10V cable extension position sensor is connected to EL3104 input terminal. Beckhoff AM8121 DC-servomotor is connected to EL7211-9014 terminal. The aim of the test is to make the velocity of the motor dependent on the input signal of EL3104, or in other words, to create the constant link between cable extension and motor velocity and make the velocity change proportionally to cable extension. When the cable is in steady state, motor velocity is constant, and when the cable is not extended, motor's shaft does not rotate.

4.1.1 Beckhoff AM8121 DC-servomotor

Beckhoff AM8121 servomotor (Figure 4.1) is brushless DC-motor. As most of the servomotors, this motor demonstrates its advantages in highly dynamic and precise positioning applications with demanding requirements in terms of dynamics and stability. The servo motor is equipped with permanent magnets in the rotor. As the motor does not have brushes, the commutation being implemented electronically in the servo terminal. AM8121 motor is intended for speed- and/or torque-controlled operation via servo terminal EtherCAT EL7211.

Technical data of AM8121 servomotor is presented in Table 4.1.

Figure 4.1 Beckhoff AM8121-1F20 DC-servomotor.

Table 4.1. AM821 DC-servomotor Technical Data. (Beckhoff Automation)

Parameter	**Characteristics**
Mains voltage, UN	48 [V]
Max. mains voltage	50 [V]
Standstill current	4,0 [A]
Standstill torque	0,5 [Nm]
Max. mech. speed	12000 [min-1]
Rated speed	3000 [min-1]
Rated torque	0,5 [Nm]
Rated power	157 [W]
Peak current	17 [A]
Torque constant	0,125 [Nm/A]
Voltage constant	8 [mVmin]
Winding resistance	1.6 [Ohm]
Winding inductance	3 [mH]
Rotor moment of inertia	0,134 [kgcm2]
Number of contacts	8
Static friction torque	0,002 [Nm]
Thermal time constant	10 [min]
Weight	1.10 [kg]

4.1.2 EL7211-9014 Terminal

The servo-motor EtherCAT terminal EL7211-9014 (50 VDC, 2.8 Arms) is designed for the AM81xx series motor types (Figure 4.2).

The control technology is based on field-orientated current and PI speed control and supports dynamic positioning tasks. Operational reliability is supported by monitoring of numerous parameters, such as overvoltage and under voltage, overcurrent, terminal temperature or motor load. With the One Cable Technology (OCT) the encoder cable is omitted by transmitting the signals of the encoder digitally via the existing motor cable (Beckhoff Automation).

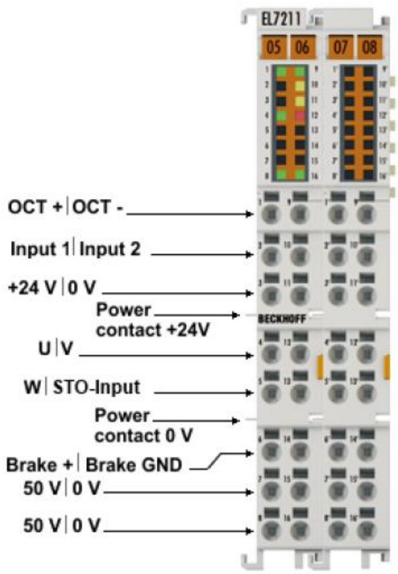

Figure 4.2. Beckhoff EL7211-9014 Servomotor Terminal (Beckhoff Automation).

As with the others Beckhoff I/O terminals, EL7211-9014 terminal is inserted into the terminal network. The terminal network is scanned via the TwinCAT System Manager.

Technical data of EL7201-9014 is showed in Table 4.2

Table 4.2. EL7211-9014 Terminal Technical Data. (Beckhoff Automation)

Parameter	**Characteristics**
Number of outputs	3 motor phases, 2 motor holding brake
Number of inputs	2 (4) DC link voltage, 2 absolute feedback, 2 digital inputs. 1 STO input
DC link supply voltage	8...50 VDC
Supply voltage	24 VDC via the power contacts / via the E-bus
Output current	4.5 Arms
Peak current	9 Arms for 1 second
Rated power	276 W
Motor holding brake output voltage	24 V (+ 6 %, - 10 %)
Load type	permanently excited synchronous motors, inductive (series AM81xx)
PWM switching frequency	16kHz
Current controller frequency	double PWM switching frequency
Velocity controller frequency	16 kHz
Diagnostic LED	Status, warning, errors and limits
Power loss	typ. 1.6 W
Current consumption via E-bus	typ. 120 mA
Current consumption from the 24 V	typ. 55 mA + holding brake
Reverse voltage protection	through the body diode of the overvoltage protection device
Possible EtherCAT cycle times	Multiple of 125 µs
Configuration	no address setting required; configuration via TwinCAT System Manager
Weight	approx. 95 g
Dimensions (W x H x D)	approx. 27 mm x 100 mm x 70 mm (width aligned: 24 mm)

4.2 Methods and Results

4.2.1 Configuration

MATLAB/SIMULINK Block Diagram

In order to make an algorithm, described in “4.1 Module Description”, the following block diagram is built in Simulink:

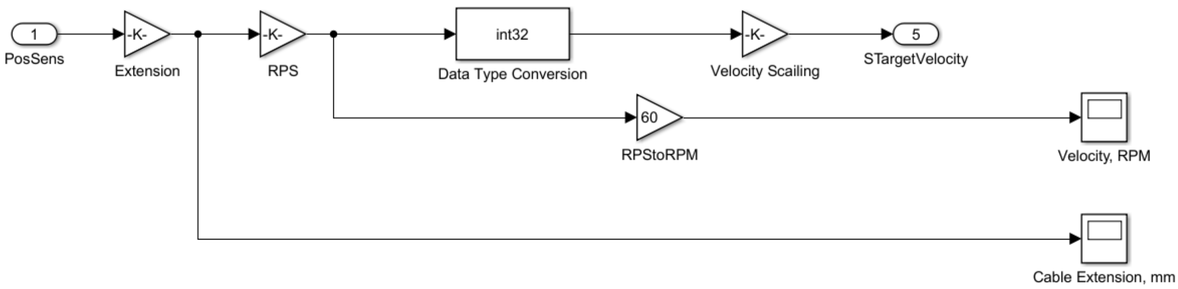

Figure 4.3. Case Study 2 MATLAB/SIMULINK Block Diagram.

The input signal from the cable extension position sensor comes to “PosSens” input block. In order to convert input signal value into cable extension in millimeters, “Extension” gain block is used. “RPS” converts extension into velocity value in revolutions per second, and then it is converted to revolutions per minute. Data Type Conversion block is needed to match the position sensor signal with the target velocity of motor. “Velocity Scaling” block converts incoming velocity value into appropriate for the AM8121 type.

TwinCAT 3 Configuration

As previously, the MATLAB/SIMULINK model is configured and generated for TwinCAT 3 target system. In TwinCAT 3 this model is added as a TcCOM Object (Figure 4.4).

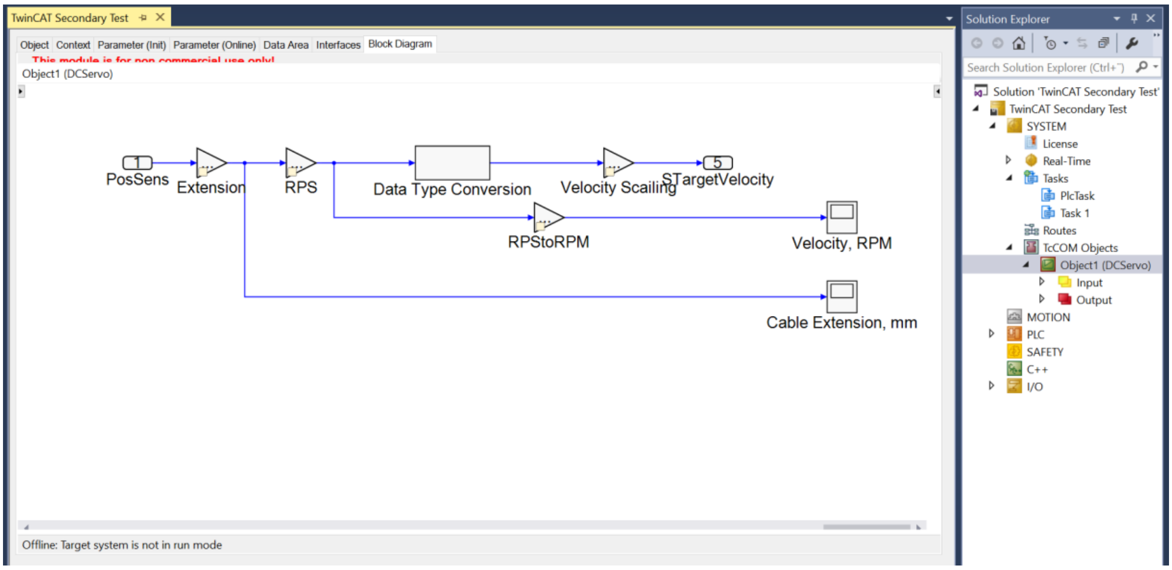

Figure 4.4. MATLAB/SIMULINK Block Diagram in TwinCAT 3.

As the system needs a fast response, the Task for TcCOM object was configured with the lowest possible in this case amount of cycle ticks:

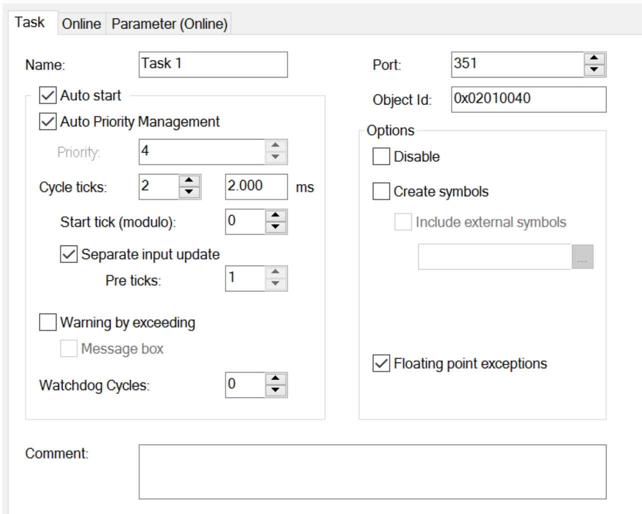

Figure 4.5. Task for TcCOM Object of Case Study 2.

The next step is device scan. When the EL7211-9014 Terminal is found, system automatically recognizes connected to the terminal drives and configures it with default settings. In the case of this secondary test there is no need in commissioning with NC or CNC, so automatically added Motion elements are deleted.

As Outputs EL7211-9014 terminal has DRV Controlword and DRV Target velocity as the motor's operation mode is CSV (Cyclic Synchronous Velocity). DRV Target velocity defines the speed of motor's shaft.

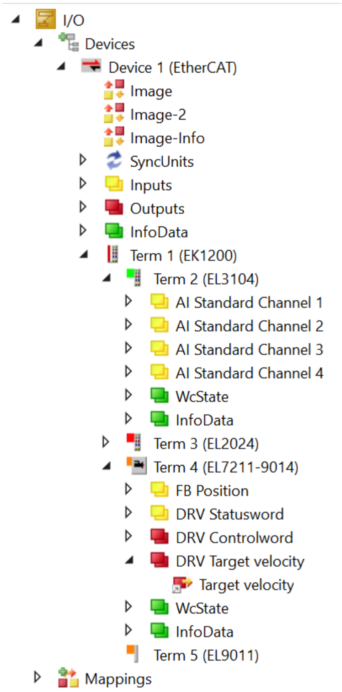

Figure 4.6. Scanned Devices and EL7211-9014 Outputs.

PLC and I/O links

The task of the PLC Project in the secondary test is to link target velocity variable, defined by extension of sensor cable, from the block diagram with the EL7211-9014 terminal "DRV Target velocity", which defines AM8121 servomotor rotational speed. Concerning this the following GVL was created:

```
VAR_GLOBAL

    SensPosition AT %I* : DINT;
    TargetVelocity AT %Q* : DINT;

END_VAR
```

Figure 4.7. GVL for Case Study 2.

Variable "SensPosition" is linked to the output block diagram block "STargetVelocity". Variable "TargetVelocity" is linked to DRV Target velocity terminal output.

The MAIN program of PLC has the following code:

```
IF GVL.SensPosition > 0 THEN
GVL.TargetVelocity := GVL.SensPosition;
ELSE
    GVL.TargetVelocity :=0;
END_IF
```

Figure 4.8. MAIN program for Case Study 2.

The IF condition assigns the motor's velocity to the block diagram output and prevents it of being negative, which can cause terminal error.

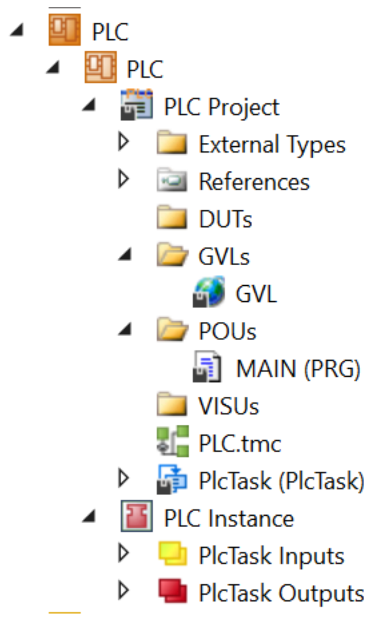

Figure 4.9. Case Study 2 PLC Project Tree.

Once the PLC Project is built and I/O variables are linked to the I/O of terminals, the program is ready to execution. After TwinCAT 3 configuration is activated, PLC Project is logged in, and MATLAB/SIMULINK platform is switched to External mode.

4.2.2 Results

As can be seen from TwinCAT 3 and MATLAB/SIMULINK Scopes, the motor's velocity follows the extension of position sensor cable.

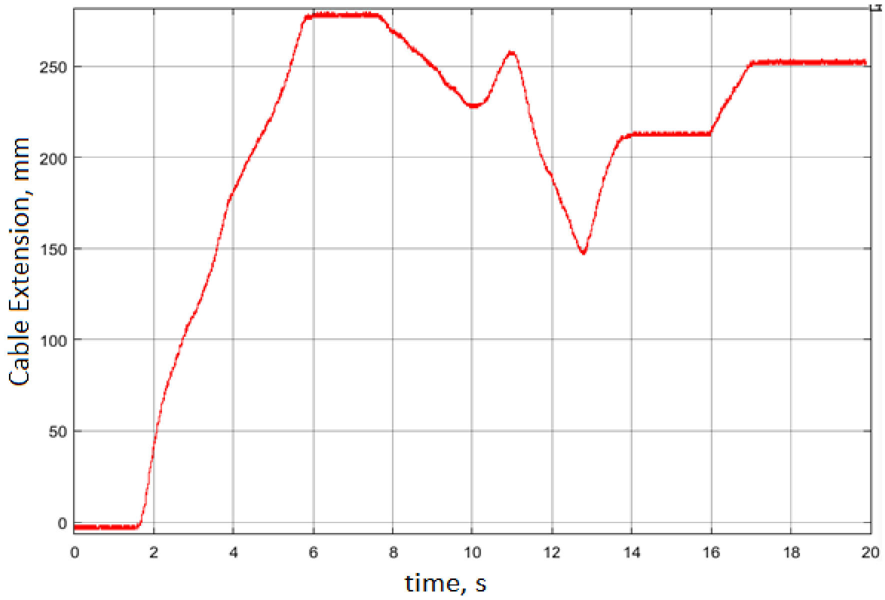

Figure 4.10. Cable Extension scope in MATLAB/SIMULINK.

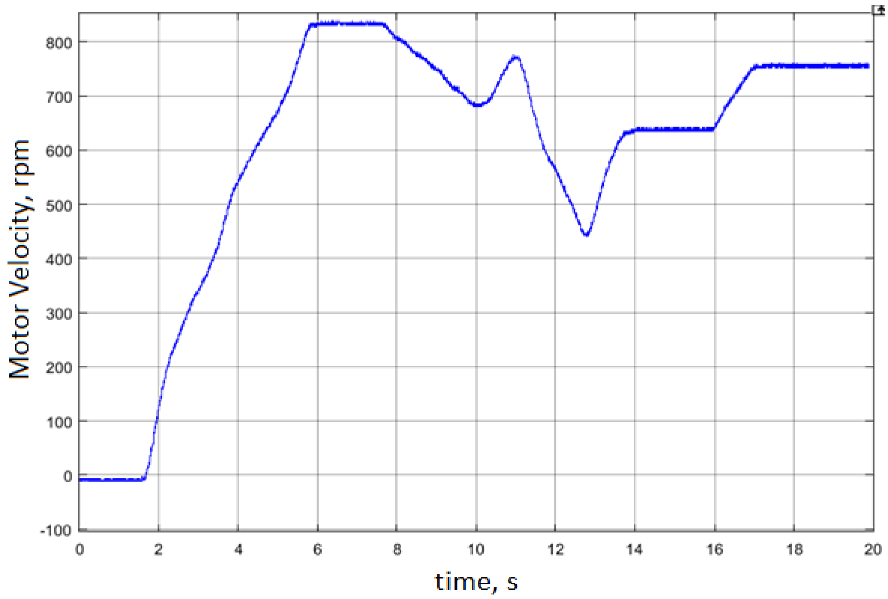

Figure 4.11. Servomotor rotational velocity scope in MATLAB/SIMULINK.

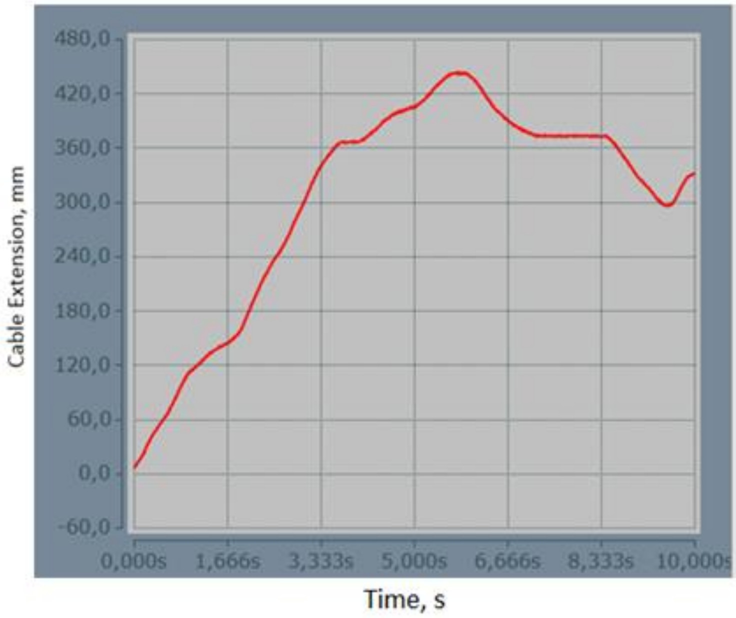

Figure 4.12. Cable Extension scope in TwinCAT 3.

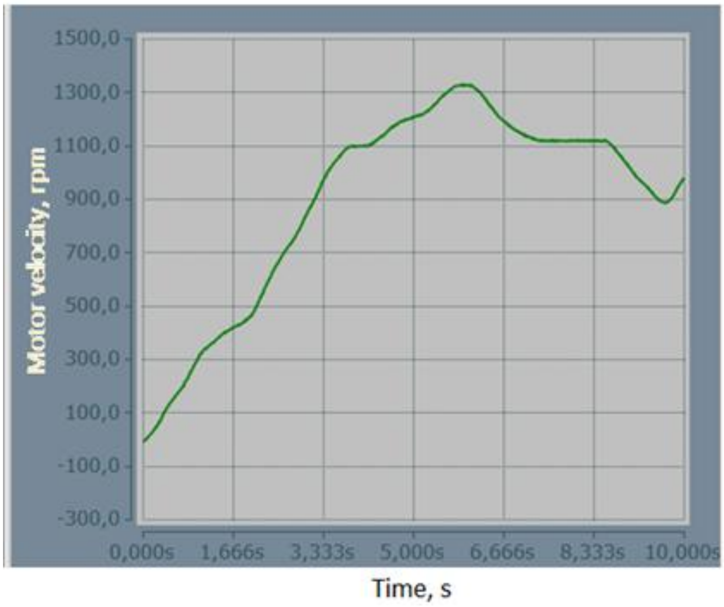

Figure 4.13. Servomotor rotational velocity scope in TwinCAT 3.

According to figures 4.10 and 4.11 it can be concluded that servomotor reacts to position sensor input signal changes immediately, and response is fast.

Both Case Study 1 and Case study 2 provide explanations and examples of how the TwinCAT 3 and MATLAB/SIMULINK integration is carried out, and a control algorithm, built in MATLAB/SIMULINK, is implemented using Beckhoff hard- and software. In the frames of research work these Case Studies allowed to move to the next stage which is hydraulic system control algorithm development and implementation.

5. HYDRAULIC RIG

5.1 System Description

The hydraulic system under investigation is represented by rail slider with 1 DOF. The system consists of following main components:

- Hydraulic cylinder;
- Direct-operated proportional servo valve;
- Pump and tank;
- Pressure sensors, position sensor, electrical convertors and amplifiers;
- Load (mass).

The actuator of the system is a hydraulic cylinder where a pressurized mineral oil is utilized to move the stroke and thereby the mass along the rails in horizontal direction. The cylinder flows are controlled by the servo valve. The system diagram is presented in Figure 5.1.

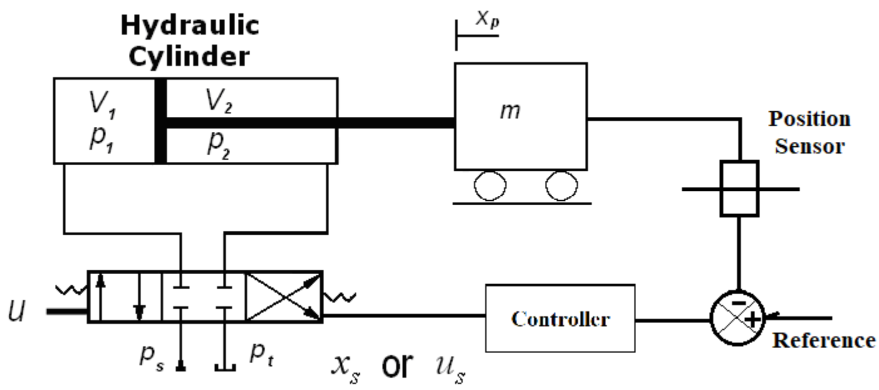

Figure 5.1. Experimental servo hydraulic system diagram (Roozbahani 2011).

The system is taken into closed loop control with the mass position as an output and feedback signal. The input signal for valve is voltage. Applying the input to the valve makes the valve's spool shift and regulates the liquid flows. The valve's main spool is a mass-held in position by a spring system. The close-loop block diagram of the system is presented in Figure 5.2.

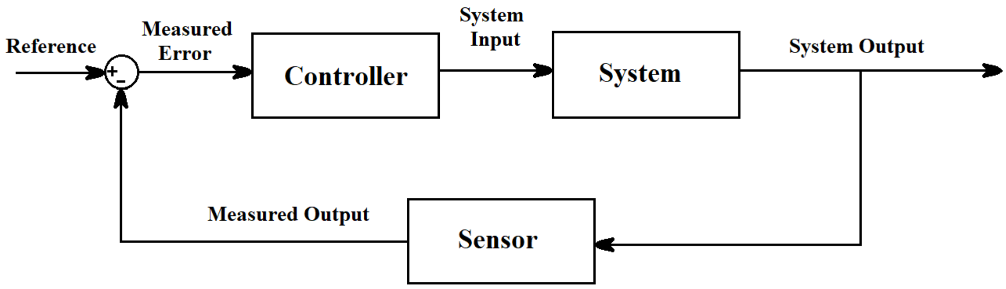

Figure 5.2. Experimental hydraulic system close loop diagram.

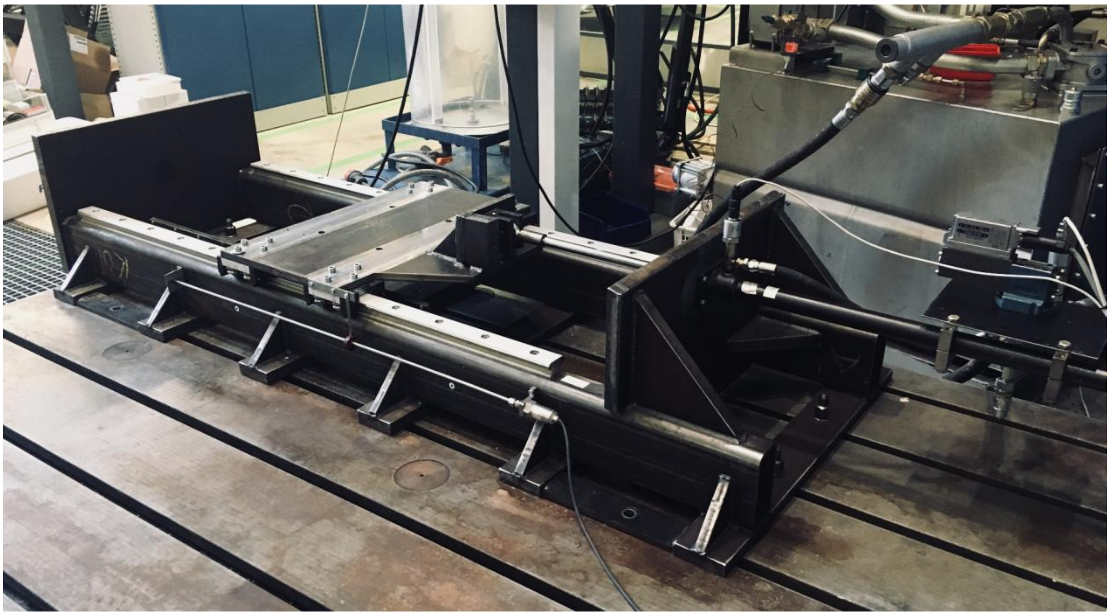

Figure 5.3. The system under investigation - hydraulic slider.

5.1.1 Hydraulic Cylinder

According to Roozbahani (2011, p. 119): “A hydraulic cylinder (also called a linear hydraulic motor) is a mechanical actuator that is used to give a unidirectional force through a unidirectional stroke. Hydraulic cylinders get their power from pressurized hydraulic fluid, which is typically oil. The hydraulic cylinder consists of a cylinder barrel, in which a piston connected to a piston rod moves back and forth. The barrel is closed on each end by the cylinder bottom (also called the cap end) and by the cylinder head where the piston rod comes out of the cylinder. The piston has sliding rings and seals. The piston divides the inside of the cylinder in two chambers, the bottom chamber (cap end) and the piston rod side chamber (rod end). The hydraulic pressure acts on the piston to do linear work and motion. A hydraulic cylinder is the actuator or «motor» side of this system. The «generator» side of the hydraulic system is the hydraulic pump which brings in a fixed or regulated flow of oil to the bottom side of the hydraulic cylinder, to move the piston rod upwards. The piston pushes the oil in the other chamber back to the reservoir. The piston moves instead downwards if oil is pumped into the piston rod side chamber and the oil from the piston area flows back to the reservoir without pressure."

The main components of hydraulic cylinder are: cylinder barrel, bottom, head, piston, piston rod, rod gland.

The cylinder in studied hydraulic system is a non-symmetrical position sensing cylinder, which provides instant electronic position feedback information from the cylinder indicating of how much the rod is extended throughout the range of stroke. The maximum stroke length of rod is 1 meter.

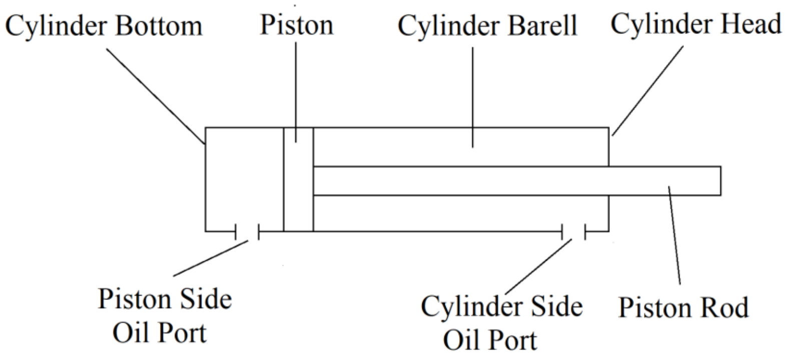

Figure 5.4. System's hydraulic cylinder scheme.

5.1.2 Hydraulic Servo Valve

To control the hydraulic cylinder piston motion servo solenoid directional control valve is utilized.

"Directional control valves (DCVs) are used to start, stop, or change the direction of fluid flow. These valves are specified by the number of connected lines (ways) and the number of control positions. The control positions determine the way in which the lines are interconnected, and consequently the directions of fluid flow. The application of a DCV in controlling the direction of motion of hydraulic cylinders is illustrated in Figure 5.5. A 4/3 directional control valve is connected to the pressure line (P), return line (T), and cylinder lines (A and B). In its neutral position, the valve closes all of the four lines and the cylinder is stopped. By switching the valve to any of the other positions, the cylinder moves in the corresponding direction." (M. Galal Rabie 2009, p. 157.)

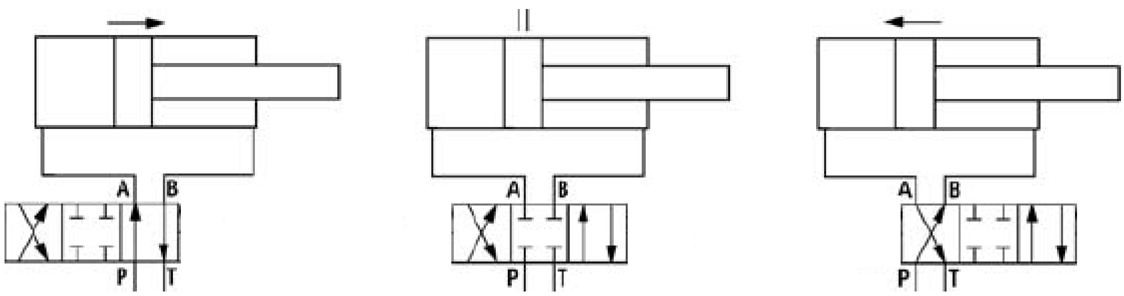

Figure 5.5. A hydraulic cylinder controlled by a 4/3 directional control valve (M. Galal Rabie 2009).

Considered hydraulic system has a valve of Bosch Rexroth 4WRPEH 6 model. It is a 4/4-way servo solenoid directional control valve, directly operated, with electrical position feedback and on-board electronics. Its symbol is presented in Figure 5.6.

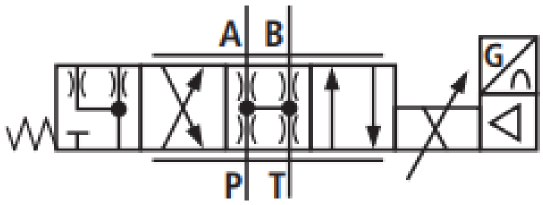

Figure 5.6. Bosch Rexroth 4/4-way servo solenoid directional control valve symbol (Bosch Rexroth AG Hydraulics 2010).

The valve's specified command value is compared with the actual position value. In case of deviations from the standard, the lifting solenoid is activated. Due to the changed magnetic force, the lifting solenoid adjusts the control valve against the spring. Lifting/control cross-section are adjusted proportionally to the command value. In case of a command value provision of 0 V, the electronics adjusts the control valve against the spring to center position. In deactivated condition, the spring is unloaded to a maximum and the valve is in fail-safe position. (Bosch Rexroth AG Hydraulics 2010.)

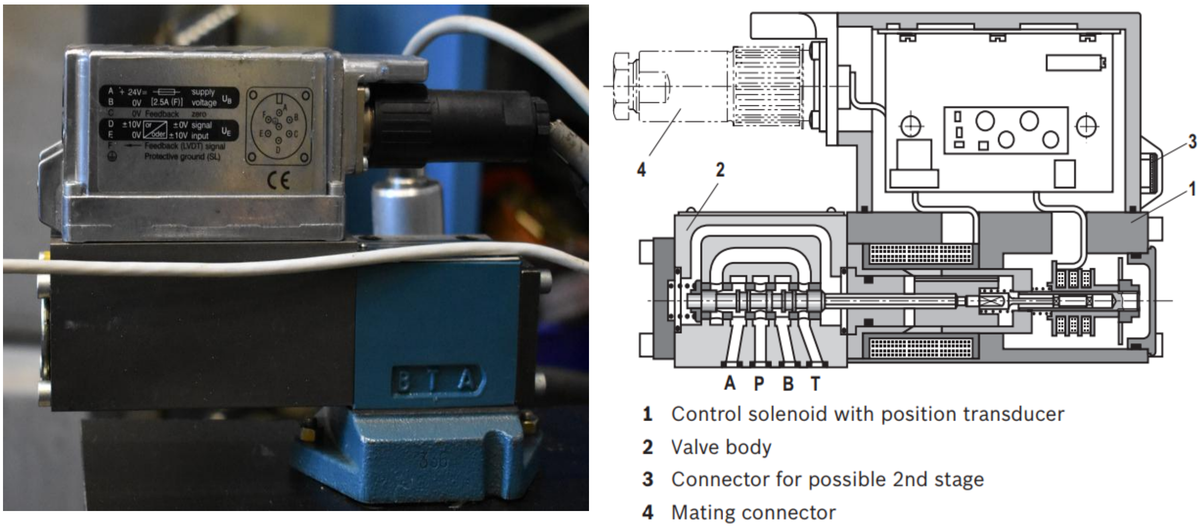

Figure 5.7. Bosch Rexroth 4WRPEH 6 servo solenoid directional control valve.

The valve’s spool motion is provided by applying the input voltage of the range of $\pm$10 V DC. To provide the information of the valve’s spool displacement the voltage signals are measured. For this purpose the spool is connected to linear variable differential transducer (LVDT) with the signals range $\pm$10 V.

5.1.3 Load

The system’s load (Figure 5.8) is a 62.15 kg mass on the hydraulic slider attached to a cylinder piston.

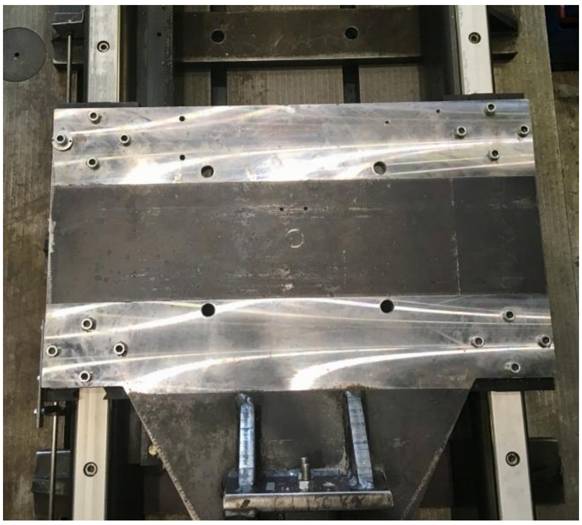

Figure 5.8. Hydraulic slider load.

5.1.4 Pump and Tank

To increase the energy of the liquid flow in hydraulic circuit an axial piston pump is used (Figure 5.9).

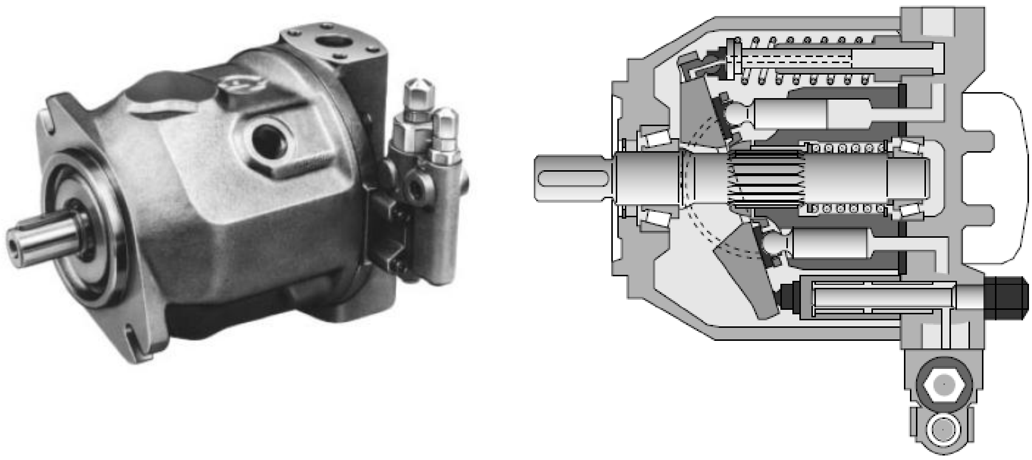

Figure 5.9. Axial piston pump (Bosch Rexroth AG Hydraulics 2004).

Axial piston pump converts input shaft rotary motion to an axial reciprocating pistons motion. This pump uses a swash plate to translate the rotary motion of a shaft into the reciprocating motion. The linear motion increases with increase in the angle between disk and the shaft and the disk attached to a shaft. The reciprocating pistons motion causes the drawing in and pumping out of the oil.

The variable displacement axial piston pump A10VSO is utilized in hydraulic circuit. Its plate is designed for hydrostatic motors used in open hydraulic circuits. The pump's volumetric flow is proportional to the motor rotational speed. By adjusting the position of the plate the flow is varied steplessly. The pump is able to control pressure and liquid flow rates.

Pump's technical data is presented in Table 5.1

Table 5.1. Axial piston pump A10VSO Technical Data. (Bosch Rexroth AG Hydraulics 2004)

Parameter	**Characteristics**
Absolute pressure at inlet port, min.	0.8 bar
Absolute pressure at inlet port, max.	30 bar
Nominal pressure	280 bar

Table 5.1 continues. Axial piston pump A10VSO Technical Data.(Bosch Rexroth AG Hydraulics 2004)

Peak pressure	350 bar
Displacement	18 cm^3
Max. speed	3300 rpm
Max. permissible speed (speed limit)	3900
Max. flow	59.4 L/min
Max. power	27.7 kW
Max. torque	80.1 Nm
Moment of inertia about drive axis	0.00093 kgm^2
Fill capacity	0.4 L

The pump is driven by induction motor with power 55 kW and nominal rotational speed 1450 rpm.

The pump supply pressure range is 0-100 bar and the voltage range is 0-100 V. Based on empirical knowledge, the operational pressure and flow rate are set at 40 bar and 40 m^3/s respectively.

To create circuit liquid flows the hydraulic system is equipped with a closed tank with the atmospheric pressure inside it. The hydraulic oil is drawn from the tank and pressurized in the pump to the set pressure.

5.1.5 Sensors, converters and amplifiers

Position sensor

The linear position of the hydraulic slider load is measured using magnetostrictive transducer based on Widermann effect (Figure 5.10). The transducer is supplied with 24 V DC. The magnetostrictive sensor consists of position magnet, strain pulse detection system, waveguide and damping module. The load position is sensed by determination the distance between the permanent magnet and the sensor head.

The waveguide is made of ferromagnetic material. When the magnetic field of the load magnet interacts with the magnetic field around the waveguide, a strain pulse is generated. This pulse is detected by the strain pulse detection system and then converted into electrical pulse. The load position is determined using the time the strain pulse takes to reach the detection system.

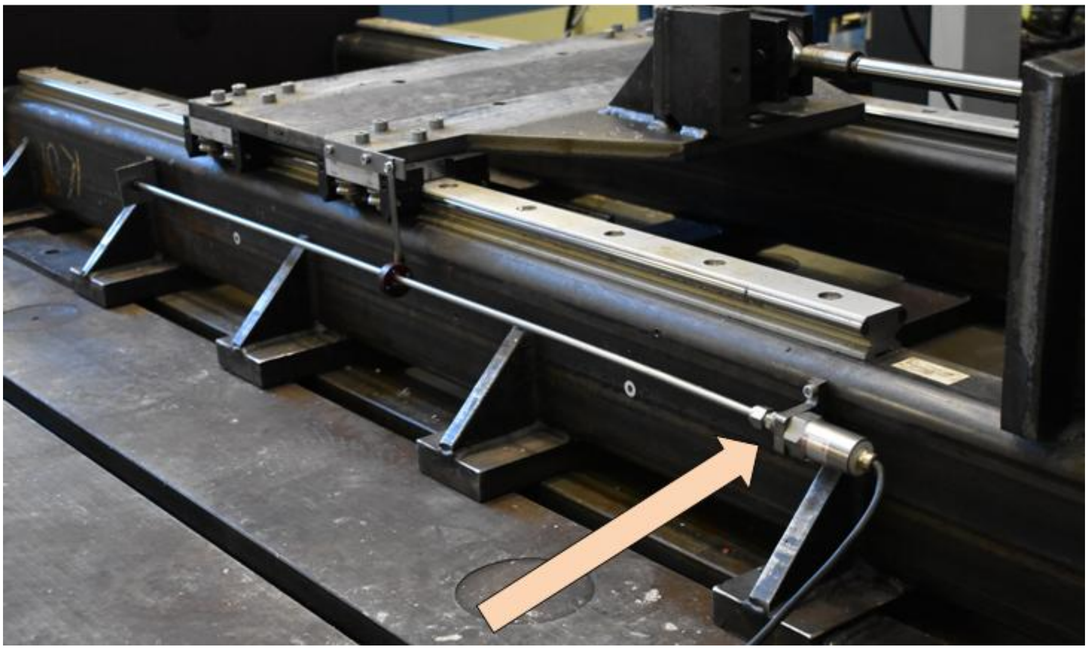

Figure 5.10. Position sensor - magnetostrictive displacement transducer.

Pressure sensors

The valve outlets are equipped with pressure sensors - pressure transducers with on-board electronics. The measured pressures are supply pressure p_s and pressure in cylinder chambers p_1 and p_2 (Figure 5.11).

The pressure sensors convert mechanical pressure variable into voltage signal. The output signal is proportional to pressure. The sensing element is a diaphragm made of spring material with thin-layer strain gauge sensor. The presence of on-board electronics provides temperature compensation. All the necessary adjustments and calibrations are done by manufacturer.

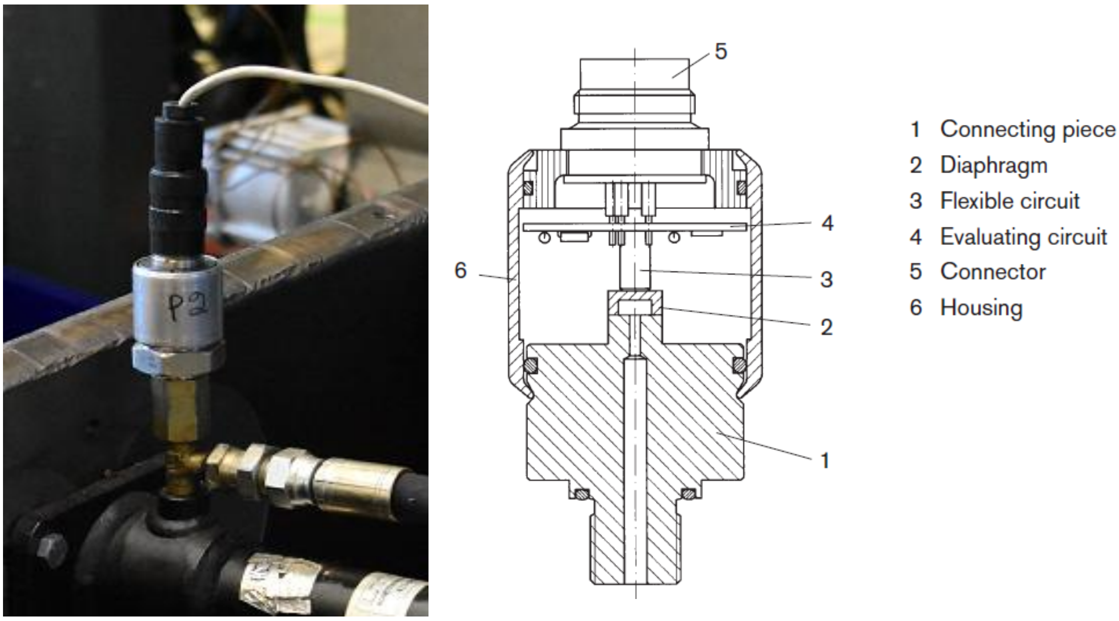

Figure 5.11. Pressure transducer utilized in hydraulic circuit (Bosch Rexroth AG Hydraulics 2005).

The technical data of pressure sensors is given in Table 5.2.

Table 5.2. Pressure transducers Technical Data. (Bosch Rexroth AG Hydraulics 2005)

Parameter	**Characteristics**
Measuring range	up to 210 bar
Output signal range	0-10 V
Sypply voltage	12-28 V
Accuracy class	0.5
Connector type	7-pin concentric
Bursting pressure	>1,500 bar
“Dynamic” overload capacity	2 x *p*nom
“Static” overload capacity	3 x *p*nom
Linearity error incl. hysteresis	<±0.5 %
Hydr. dead volume	approx. 0.5 cm^3
Measuring frequency	approx. 1 kHz

Electrical converters and amplifiers

To connect position and pressure sensors as well as actuators to a remote platform (computer) an electrical convertor and amplifiers box is utilized (Figure 5.12). This box provides necessary components to establish the connection between hardware and data center. It also includes displacement transducer base and fuses for system safe operation.

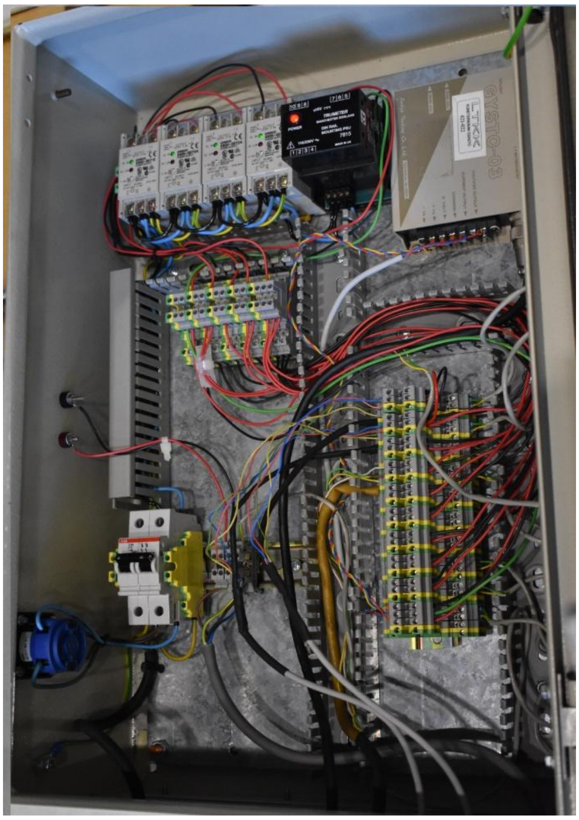

Figure 5.12. Electrical convertors and amplifiers box.

5.2 System Mathematical Model

Every component and phenomena in the considered hydraulic system is represented by mathematical model. These mathematical models combination describes the whole system mathematical model which accuracy depends on sub-models authenticity and phenomena’s neglected in the simulation model.

The system mathematical model is built based on formulations and phenomena’s, described below.

Newton's second law, the equation of motion for the hydraulic system

Handroos, Roozbahani and Wu (2011) wrote the Newton's second law for the system as:

$$m \cdot \ddot{x}_p = p_1 \cdot A_1 - p_2 \cdot A_2 - F_f \ , \qquad (1)$$

where m denotes the mass weight, x_p the displacement of piston, A_1 and A_2 the piston areas, p_1 and p_2 the pressures, and F_f the friction force. (Handroos, Roozbahani and Wu 2011)

The equation parameters have the values, presented in Table 5.1.

Table 5.3. Equation 1 parameters.

Parameter	**Value**	**Units**
A_1	8.04×10^{-4}	m^2
A_2	4.24×10^{-4}	m^2
m	62.5	kg

Friction Force

Friction force is taken into account as an external disturbance for the hydraulic cylinder. Friction is typically modeled as the velocity and friction force static mapping. There are Coulomb and Viscous friction components. But several friction properties cannot be modeled by static models. The analytic model of friction dynamics, proposed by LuGre, has high accuracy and addresses the characteristics of the friction which observed in the system but cannot be explained by static models (e.g. stick-slip motion, pre-sliding displacement and friction lag). (Roozbahani 2011)

LuGre model for friction is defined as (Liu, Handroos, Haario and Wu 2009):

$$\begin{cases} F_f = \sigma_0 \cdot z + \sigma_1 \cdot \frac{dz}{dt} + k_v \cdot \dot{x}_p \\ \frac{dz}{dt} = \dot{x}_p - \frac{|\dot{x}_p|}{g(\dot{x}_p)} z \\ g(\dot{x}_p) = \frac{1}{\sigma_0}\left[F_c + (F_s - F_c) \cdot e^{-(\frac{\dot{x}_p}{v_s})^2}\right] \end{cases} \qquad (2)$$

where z is an internal state, $g(\dot{x}_p)$ describes part of the steady-state characteristics of the model for constant velocity motion, v_s is the Stribeck velocity, Fs is the static motion friction, F_c is Coulomb friction, k_v is the viscous friction, σ_0 is the stiffness coefficient and σ_1 is the damping coefficient. (Liu, Handroos, Haario and Wu 2009)

Table 5.4. Equation 2 parameters.

Parameter	**Value**	**Units**
k_v	87.74	Ns/m
v_s	0.1624	m/s
Fs	2921	N
F_c	74.81	N
σ_0	1521	N/m
σ_1	848.3	N×s/m

Cylinder Chambers Volume

According to Handroos, Roozbahani and Wu (2011), cylinder volumes are:

$$V_1 = A_1 \cdot x_p + v_{01}$$
$$V_2 = A_2 \cdot (L - x_p) + v_{02} \qquad (3)$$

where v_{01} and v_{02} the pipeline volumes at the two ports respectively, and L is the maximum stroke of the piston. (Handroos, Roozbahani and Wu 2011)

Table 5.5. Equation 3 parameters.

Parameter	**Value**	**Units**
v_{01}	1.07×10-4	m^3
v_{02}	1.07×10-4	m^3
L	1	m

Valve flows

The following equations describe the flow in the valve (Handroos, Roozbahani and Wu 2011):

$$Q_1 = \begin{cases} c_s \cdot u_s \cdot sign(p_s - p_1) \cdot \sqrt{|p_s - p_1|}, u_s \geq 0, \\ c_s \cdot u_s \cdot sign(p_1 - p_t) \cdot \sqrt{|p_1 - p_t|}, u_s < 0, \end{cases}$$
$$Q_2 = \begin{cases} c_s \cdot u_s \cdot sign(p_2 - p_t) \cdot \sqrt{|p_2 - p_t|}, u_s \geq 0, \\ c_s \cdot u_s \cdot sign(p_s - p_2) \cdot \sqrt{|p_s - p_2|}, u_s < 0, \end{cases} \qquad (4)$$

where u_s is the valve input voltage, c_s is the flow constant, p_s is the supply pressure, and p_t is the tank pressure. (Handroos, Roozbahani and Wu 2011)

Table 5.6. Equation 4 parameters.

Parameter	**Value**	**Units**
c_s	3.021×10^{-8}	$m^3s^{-1}v^{-1}Pa^{-1/2}$
p_s	14×10^{6}	Pa
p_t	0.3×10^{6}	Pa

Valve leakage

The valve has internal and external kinds of leakage. Leakage flows are depicted in Figure 5.14.

The internal leakage flow is calculated as (Handroos, Roozbahani and Wu 2011):

$$Q_{Li} = L_i \cdot (p_2 - p_1) \quad , \qquad (5)$$

where L_i is the laminar leakage flow coefficient. (Handroos, Roozbahani and Wu 2011)

Table 5.7. Equation 5 parameters.

Parameter	**Value**	**Units**
L_i	1.19×10^{-12}	$m^3s^{-1}Pa^{-1}$

The model of the external leakage flows is built as follows (Handroos, Roozbahani and Wu 2011):

$$Q_{L1} = l_1 \cdot (p_1 - p_t)$$
$$Q_{L2} = l_2 \cdot (p_2 - p_t) \qquad (6)$$

where l_1 and l_2 are the laminar leakage flow coefficients. (Handroos, Roozbahani and Wu 2011)

Table 5.8. Equation 6 parameters.

Parameter	**Value**	**Units**
l_1	$1{,}038\times10^{-13}$	m^3s
l_2	$8{,}485\times10^{-13}$	m^3s

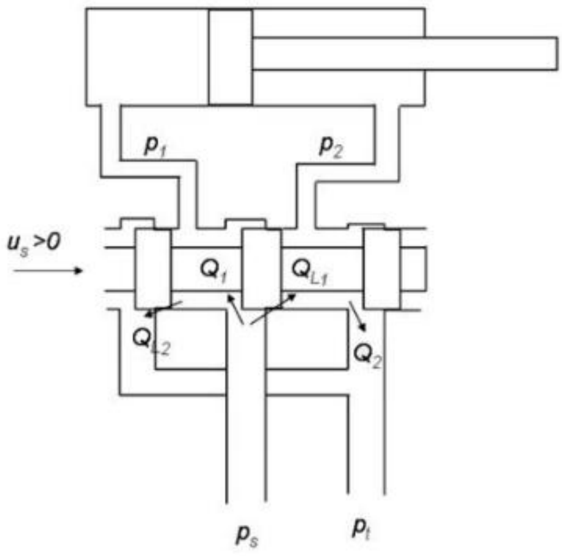

Figure 5.13. Flow diagram in the valve (Roozbahani 2011).

Pressures at the valve ports

The pressures at valve ports are described as (Handroos, Roozbahani and Wu 2011):

$$\frac{dp_1}{dt} = \frac{\beta_{e1}}{V_1}(Q_1 - A_1 \cdot \dot{x}_p + Q_{Li} - Q_{L1})$$
$$\frac{dp_2}{dt} = \frac{\beta_{e2}}{V_2}(-Q_2 + A_2 \cdot \dot{x}_p - Q_{Li} - Q_{L2}) \quad (7)$$

where, p_1 and p_2 are pressures at valve ports, Q_1 and Q_2 the valve flows, Q_{Li} the internal leakage flow, Q_{L1} and Q_{L2} the leakage flows, V_1 and V_2 the chamber volumes, $\beta_{e1}$and β_{e2} the effective bulk modules of the cylinder. (Handroos, Roozbahani and Wu 2011)

The effective bulk modules of the cylinder are characterized by (Handroos, Roozbahani and Wu 2011):

$$\beta_e = a_1 \cdot E_{max} \cdot \log\left(a_2 \cdot \frac{p_i}{p_{max}} + a_3\right) \quad (8)$$

where E_{max}, $a_1 - a_3$ represent coefficients of effective bulk modules, p_{max} is the maximum system pressure. (Handroos, Roozbahani and Wu 2011)

Table 5.9. Equation 8 parameters.

Parameter	Value	Units
E_{max}	$1{,}8\times10^{9}$	Pa
p_{max}	2.8×10^{-7}	Pa
a_1	0.3102	no unit
a_2	49.18	no unit
a_3	1.843	no unit

Valve Model

The valve's spool is connected to LDVT transducer. The LVDT signals are u_s and lay in the range of -10 to +10 V.

The relation between the simulated valve spool position x_s and the input voltage u (Handroos, Roozbahani and Wu 2011):

$$G_v(s) = \frac{x_s(s)}{u(s)} = \frac{\tau_1}{s+\tau_2}, or\ \dot{x}_s = t_1 \cdot u - t_2 \cdot x_s \qquad (9)$$

where term τ_1 has no unit, and τ_2 has unit (1/sec). The term τ_1 should have unit as (m·s^{-1}V^{-1}) since $\tau_1$multiply by u is measured in (m·s^{-1}). (Handroos, Roozbahani and Wu 2011)

The transfer function of the valve dynamics between *us* and *u* using the first order system (Handroos, Roozbahani and Wu 2011):

$$\dot{u}_s = t_1 \cdot u - t_2 \cdot u_s \qquad (10)$$

where t_1 and t_2 are the time constants (s^{-1}).

The relationship between *us* and *u* can also be expressed as (Handroos, Roozbahani and Wu 2011):

$$\dot{u}_s = (K \cdot u - u_s)/T \qquad (11)$$

where *K* is the gain and *T* the time constant (s). (Handroos, Roozbahani and Wu 2011)

Witherspone (2014): "A first order model can only be applied in case of limited frequency range, well below the natural frequency of the valve; the second order model responds the servo valve dynamics through a wider frequency range. A linearized model for an electro hydraulic servo system with a two-stage flow control servo valve and a double ended actuator has revealed that the higher order model fits closer to the experimental data because of the reduced un-modeled dynamics."

The valve's dynamics with a second order transfer function for the valve model (Handroos, Roozbahani and Wu 2011):

$$\ddot{u}_s = k \cdot \omega_n^2 \cdot u - 2 \cdot \zeta \cdot \omega_n \cdot \dot{u}_s - \omega_n^2 \cdot u_s \qquad (12)$$

where k is the gain, ζ the damping ratio and ω_n the natural angular frequency (Handroos, Roozbahani and Wu 2011).

Table 5.10. Equation 12 parameters.

Parameter	Value	Units
k	0.9907	no unit
ζ	0.5588	no unit
ω_n	481.3	rad/s

5.3 System Model in MATLAB/SIMULINK

The hydraulic system is modeled using mathematical formulation described above. Each formulation is expressed by using MATLAB/SIMULINK library blocks. The blocks are consolidated in one unite subsystem, which represents the hydraulic slider simulation model (Figure 5.14).

This simulation model is used together with the real-time control algorithm model to compare the control system implementation and test results. The real-time model is described in Chapter 6 “Implementation” of this thesis. It is to notice that the task of real-time model is to provide the proper movement for the slider load and to send the reference position to the slider and reading the slider real position at the same time, when the slider simulation model task is to provide the ideal response to output signals for theoretical and real system behavior comparison.

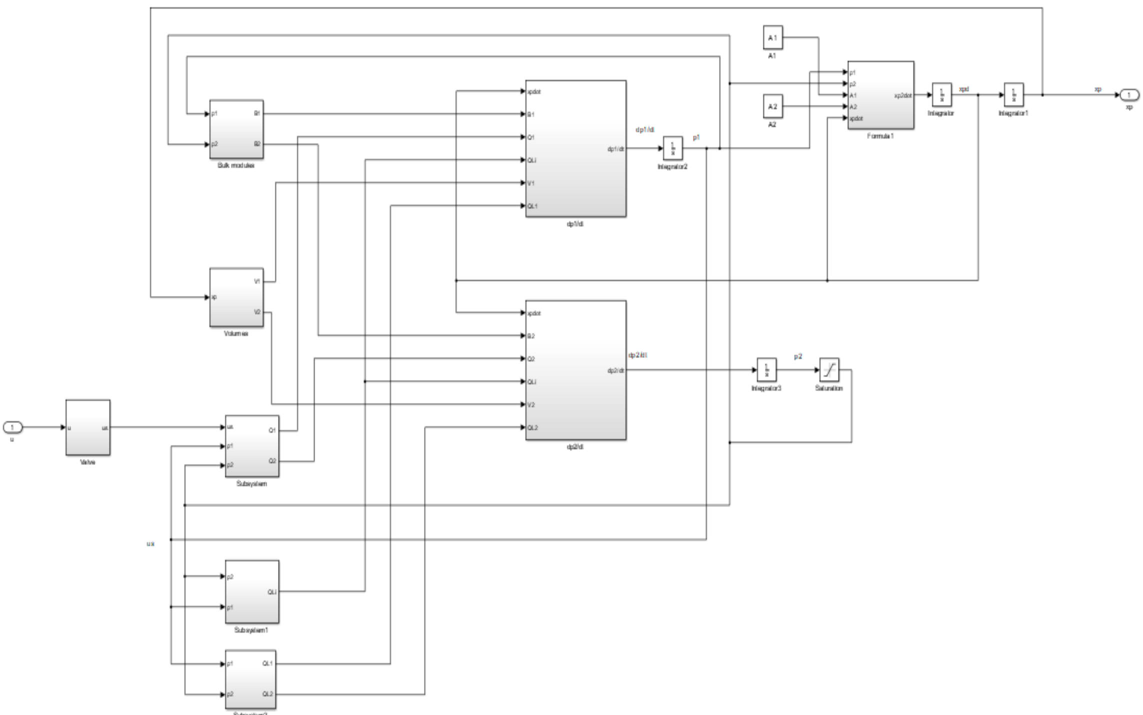

Figure 5.14. The hydraulic slider simulation model in MATLAB/SIMULINK.

6. IMPLEMENTATION

6.1 Control Method

The hydraulic system control method is based on using different types of input disturbances applied to a directional servo valve as an intelligent switching input signals. The task of control scheme is to make the slider motion follow the desired input and achieve the highest possible performance of the system. The control system has a closed loop structure with a PI controller as a correction mechanism and load position as a feedback. The disturbances are chosen as the most typical for the hydraulic systems. The laboratory system performance is compared to a simulation model behavior which configuration is based on mathematical formulations of real system. The real and simulation model disturbances are the same. Described control algorithm scheme is presented in Figure 6.1.

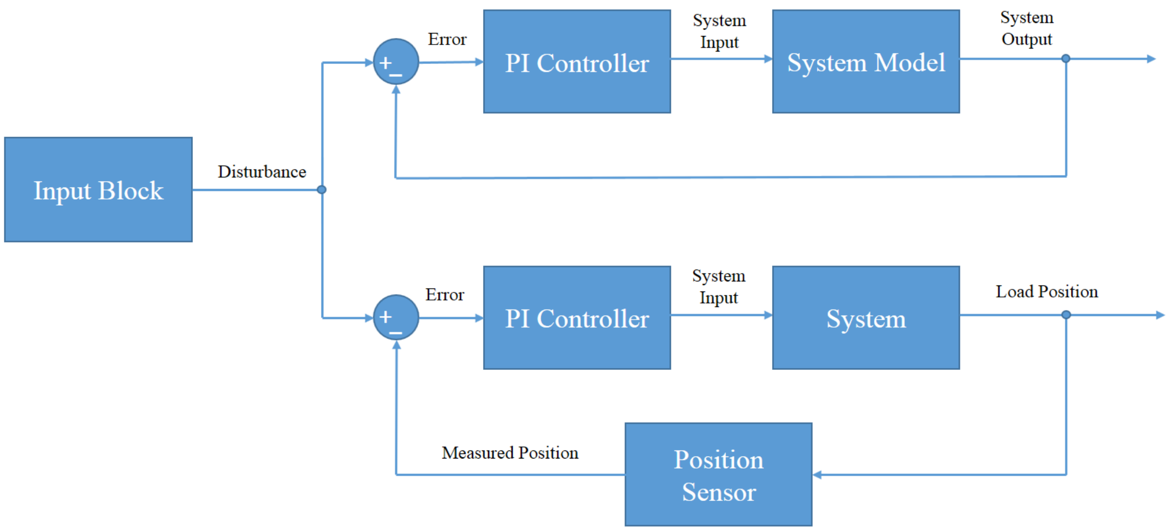

Figure 6.1. Hydraulic system control algorithm scheme.

The input block includes switching disturbance sources. For the hydraulic system test following types of typical disturbances are applied:

- Step Input;
- Pulse Input;
- Sinusoidal Input.

As a control loop feedback mechanism PI controller is utilized which is a variation of Proportional Integral Derivative (PID) control, where only the proportional and integral terms are used. The controller's output value is fed into the system as the manipulated variable. The PI controller measures the difference between a desired reference signal and actual output value and applies a correction based on proportional K_p and integral K_i terms.

The process of finding proper K_p and K_i values is called controller tuning, which aim is to obtain the system desired performance: fast response and stability. The way of controller tuning used in this work is finding K_p and K_i values by manual searching for proper values observing the system behavior and gradually applying gains changes.

6.2 MATLAB/SIMULINK Model

To implement the control algorithm for hydraulic system described above, the real-time control model was built in MATLAB/SIMULINK environment (Figure 6.2).

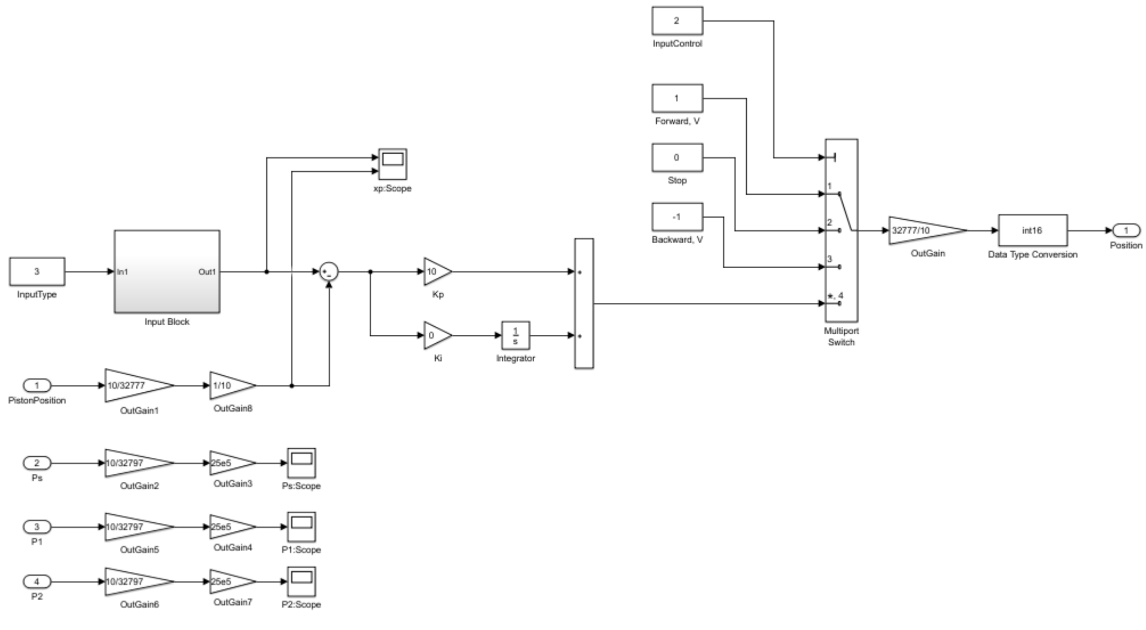

Figure 6.2. MATLAB/SIMULINK real-time control model for hydraulic slider.

The model consists of four main components: I/Os, manual control block, switching input block and PI controller.

Inputs/Outputs

The model has four input variables: slider's position *InPosition*, supply pressure *Ps* and pressures in cylinder chambers *P1* and *P2.*

As *InPosition* variable has INT16 data type, it is converted to signal in Volts using 32767/10 gain: 10 V (analog signal) = 32767 (INT16 digital data type). After conversion into voltage, the input value is then matched as load position in meters, as applying the 10 V input voltage to valve moves the slider to 1 m along the rails.

The same operation is carried out for pressures *Ps*, *P1* and *P2* input variables with the difference of Volts (V) to Pascal (Pa) conversion.

The output variable for the model is *OutPosition*. The data type of output variable is INT16 as well, so before coming directly to output block the data is converted to this type by using gain and *Data Type Conversion* blocks.

Input *InPosition* and output *OutPosition* variables represent the position sensor data. It is used as feedback for closed loop control.

Manual Control Block

Manual control component of the model is presented in Figure 6.3. Switching the Multiport Switch ports by changing the control port with constant block *InputControl* leads to applying different input commands for the system. Physically it means, that Port 1 sends 1 V signal to the valve, which makes the slider move forward. Switching to Port 2 sends no voltage at the valve input and the slider is stopped at its current position. Port 3 sends negative voltage to the valve, which make the slider move backward. These signals are coming to the output variable *OutPosition.*

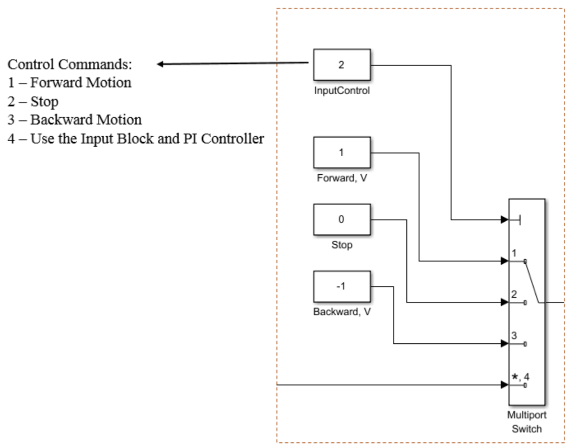

Figure 6.3. Manual control block and control commands for the system.

Input Block

The Input Block represents system disturbances and consists of three different types of input sources which are switched using Multiport Switch Block (Figure 6.4). Control port commands for the Multiport Switch are sent manually from the system model layout as constant values.

Switching to Port 1 applies Step input for the system with initial value 0 m and final value 1 m. Port 2 is a Pulse function input with slider starting position 0.4 m, amplitude 0.2 m, period 5 s and 2.5 s of pulse width. Port 3 is a Sine Wave input with slider starting position 0.4 m, amplitude 0.2 m and frequency 1 rad/s.

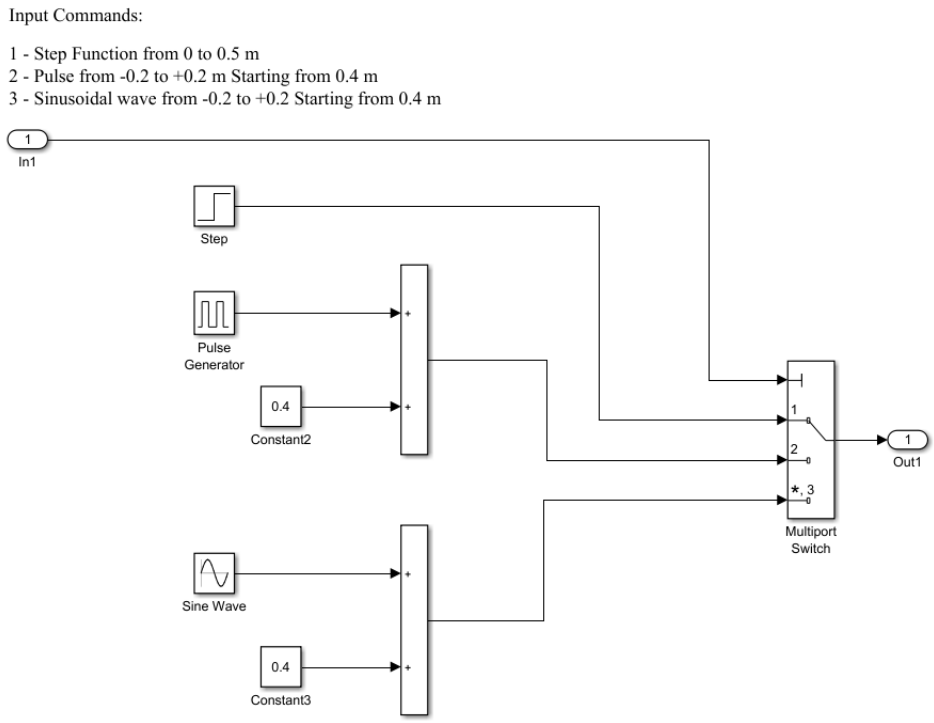

Figure 6.4. Input block with switching disturbances in MATLAB/SIMULINK.

PI controller

PI controller (Figure 6.5) was modeled using K_P and K_I gains responsible for PI controller proportional and integral terms and Integrator block. Gains values are changeable in real-time.

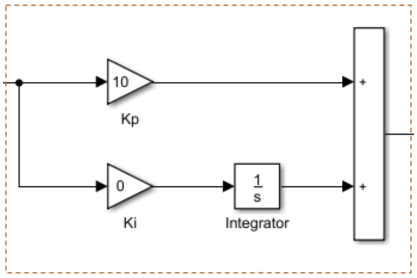

Figure 6.5. PI controller in system MATLAB/SIMULINK model.

The real-time model is combined with the simulation model as described in section 6.1. The final MATLAB/SIMULINK model both for real-time and simulation control is presented in Figure 6.6.

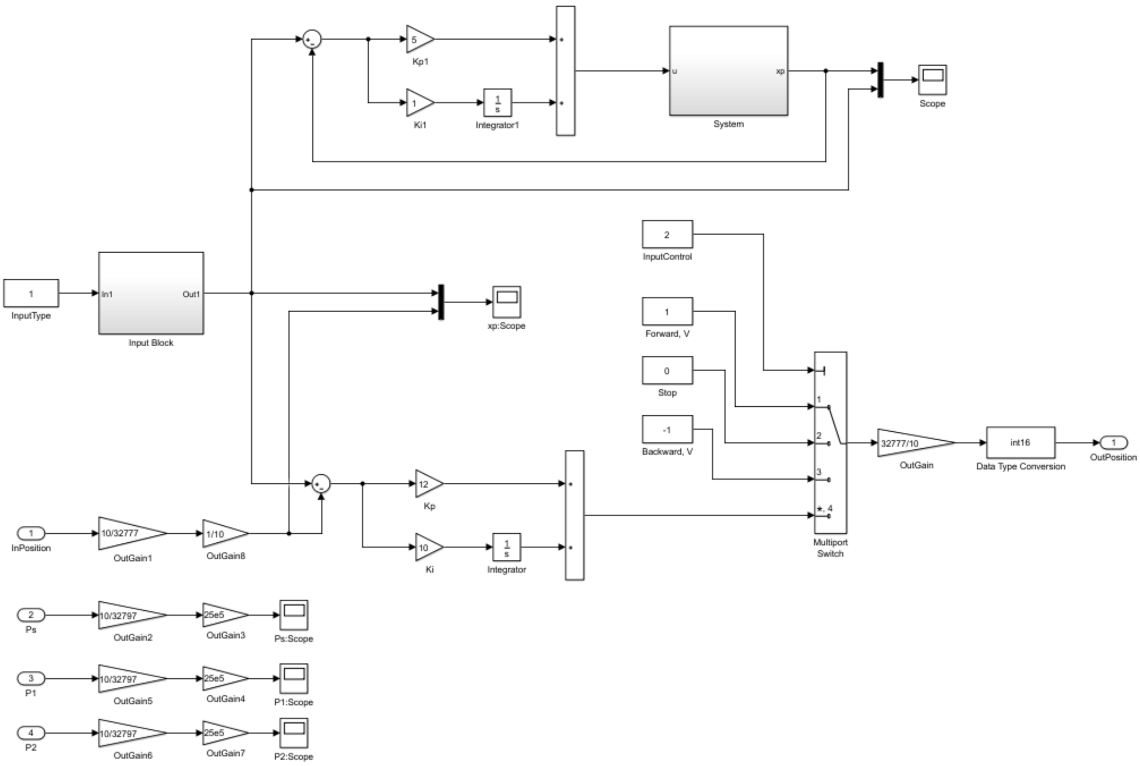

Figure 6.6 MATLAB/SIMULINK model for real-time and simulation control.

6.3 Hardware and Software Connections

Previously described Beckhoff embedded PC module is used for real-time system control.

The electrical convertors and amplifiers box has analog output pins for system position and pressures. The active connections are shown in Figure 6.7. The inputs are connected to Beckhoff EL3104 analog input terminal, and the output which is load position is connected to Beckhoff EL4134 analog output terminal (Figure 6.8 and Table 6.1).

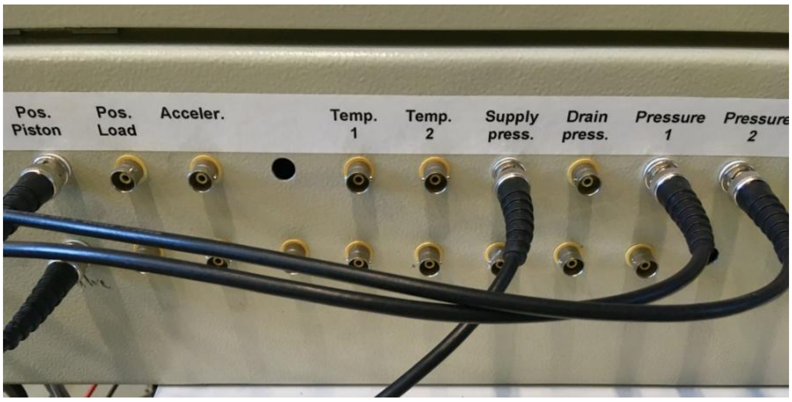

Figure 6.7 Electrical convertors and amplifiers box active inputs and outputs.

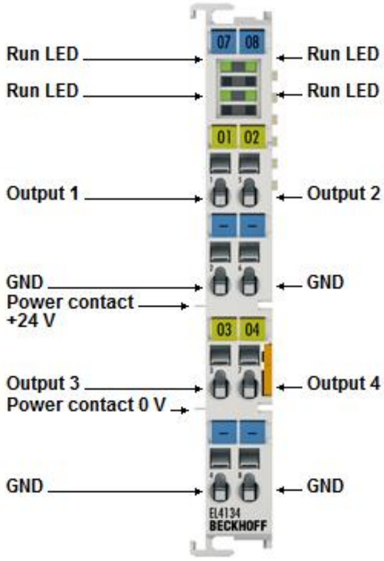

Figure 6.8. Beckhoff EL4134 analog output terminal (Beckhoff Automation).

Table 6.1. Beckhoff EL4134 terminal technical data. (Beckhoff Automation)

Parameter	**Characteristics**
Number of outputs	4
Connection technology	2-wire, single-ended
Power supply	via the E-bus
Signal voltage	-10…+10 V
Distributed clock precision	<< 1 µs
Load	> 5 kΩ (short-circuit-proof)

Table 6.1 continues. Beckhoff EL4134 terminal technical data (Beckhoff Automation).

Conversion time	~ 290 µs
Resolution	16 bit (incl. sign)
Current consumption E-bus	typ. 265 mA
Weight	approx. 65 g
Protect. class/installation pos.	IP 20/variable

The connection between embedded PC and remote platform is established and MATLAB/SIMULINK real-time control model is generated as C++ code and imported to TwinCAT 3 environment as TcCOM object (Figure 6.9) following the steps described in Case Studies section. The TwinCAT PLC project has the code, presented in Figure 6.10.

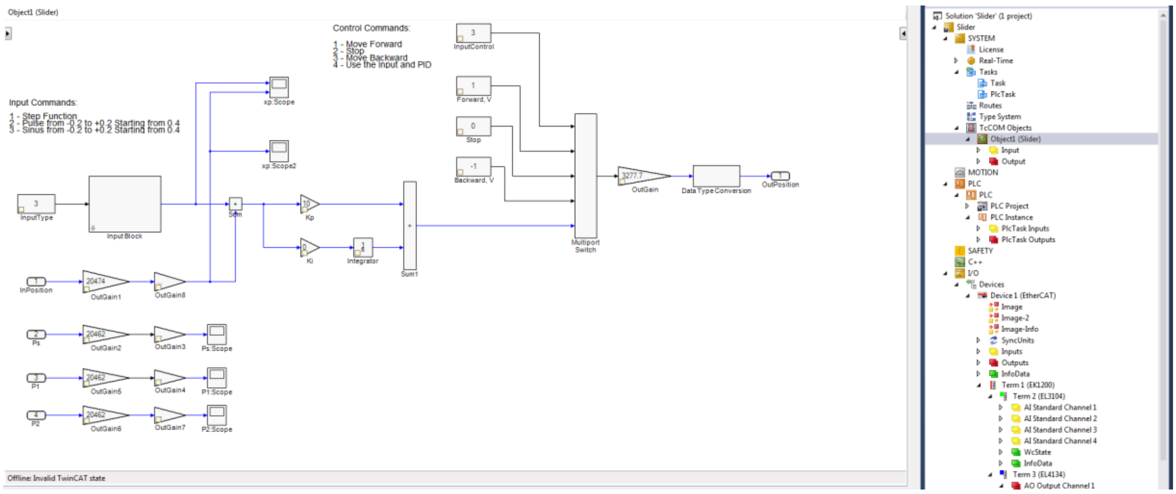

Figure 6.9. MATLAB/SIMULINK model for hydraulic system control in TwinCAT 3 environment.

```
VAR_GLOBAL
    InPos AT %I* : INT;
    OutPos AT %Q* : INT;
END_VAR

GVL.OutPos := GVL.InPos;
```

Figure 6.10. TwinCAT PLC project code for real-time system control.

The input variable *InPos* is linked to a model's output variable *OutPosition*. The output variable *OutPos* is linked to a Channel 1 of EL4134 output terminal.

After TwinCAT 3 project compilation the system is set to run mode with the data exchange with MATLAB/SIMULINK environment through the External mode. The system behavior is followed using Scope block in MATLAB/SIMULINK.

7. RESULTS

1) Manual Control

The hydraulic system is first tested with manual control block by applying forward and backward motion commands. The control port change was carried out in MATLAB/SIMULINK environment. As the model has 2 ms fixed step the system response time is fast enough to not notice any delays between command set and its execution. The data exchange between MATLAB/SIMULINK and TwinCAT 3 environment during the test was also smooth, fast and infallible. The slider's load position graph is presented in Figure 7.1. Pressure changes are depicted in Figure 7.2.

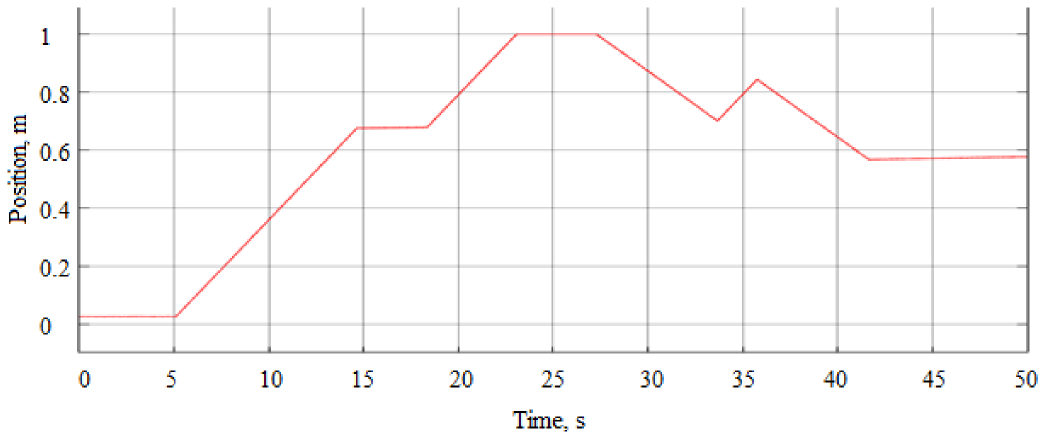

Figure 7.1. Load position graph for slider manual control.

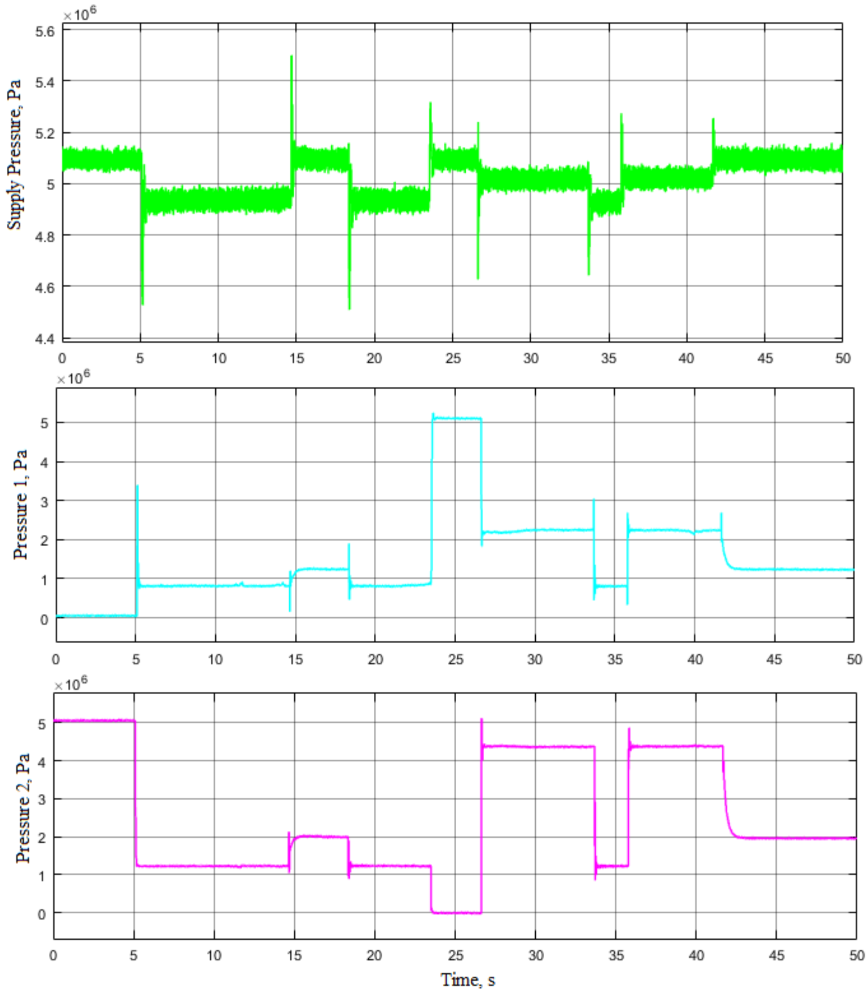

Figure 7.2. Pressures charts for slider manual control.

2) Step Input

Step input is the first type of disturbance the system was tested with. The PI controllers were tuned both for real-time control and simulation model to achieve the desired system performance. PI controller K_P and K_I terms values are presented in Table 7.1 for the simulation model and in Table 7.2 for real system control model. The output graphs are presented in Figures 7.3 and 7.4. Pressure charts are in Figure 7.5.

Table 7.1. Simulation model PI controller settings for Step input.

Parameter	Value
K_P	20
K_I	1

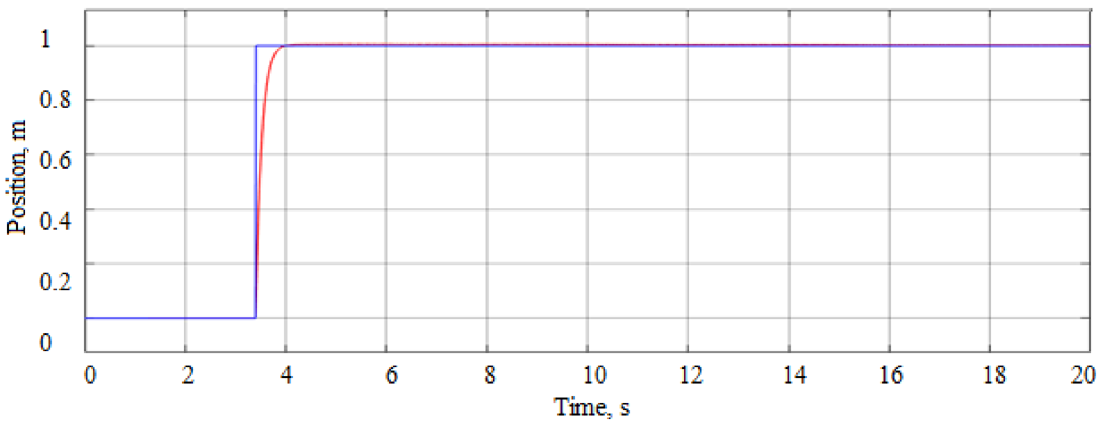

Figure 7.3. Simulation model step response.

Table 7.2. System control model PI controller settings for Step input.

Parameter	Value
K_P	12
K_I	10

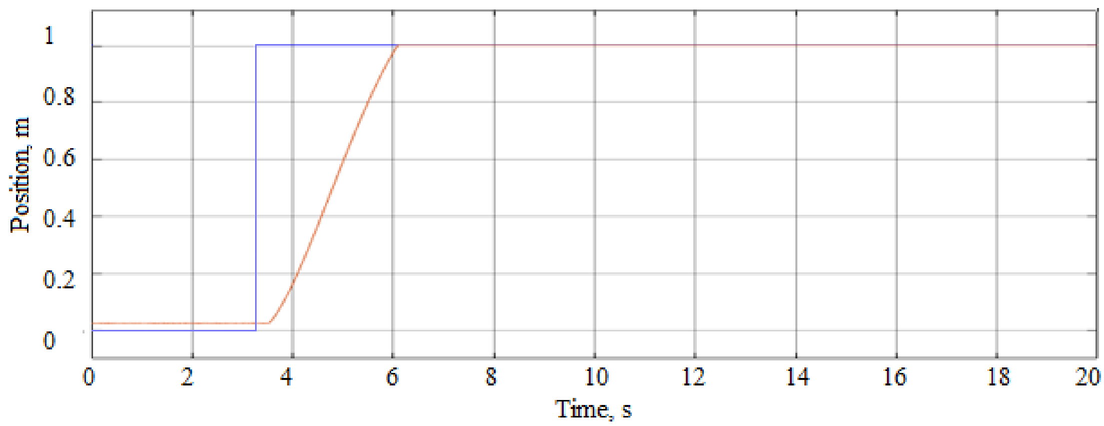

Figure 7.4. Hydraulic slider step response.

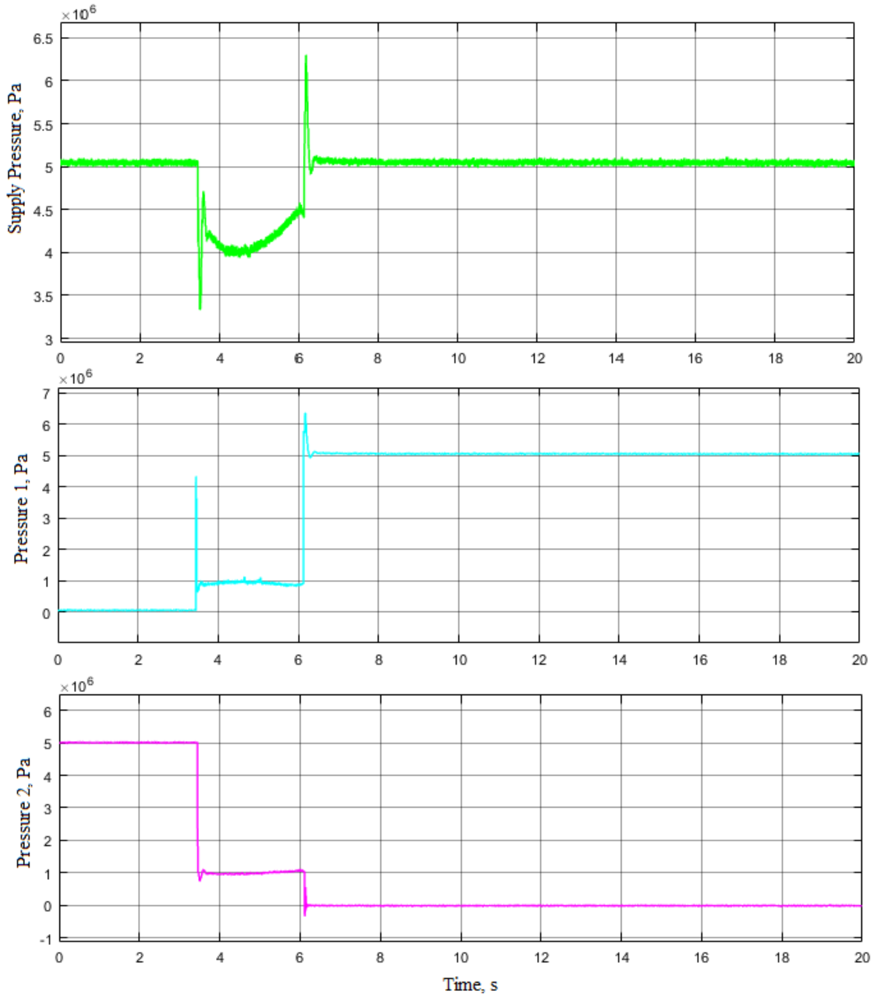

Figure 7.5. Hydraulic system pressure charts for Step input.

3) Pulse Input

The system's pulse signal reaction is affected by period, amplitude, frequency and noise. PI controller parameters were adjusted for pulse signal output (Tables 7.3 and 7.4.) again. The outputs graphs are presented in Figures below.

Table 7.3. Simulation model PI controller settings for Pulse input.

Parameter	**Value**
K_P	45
K_I	2

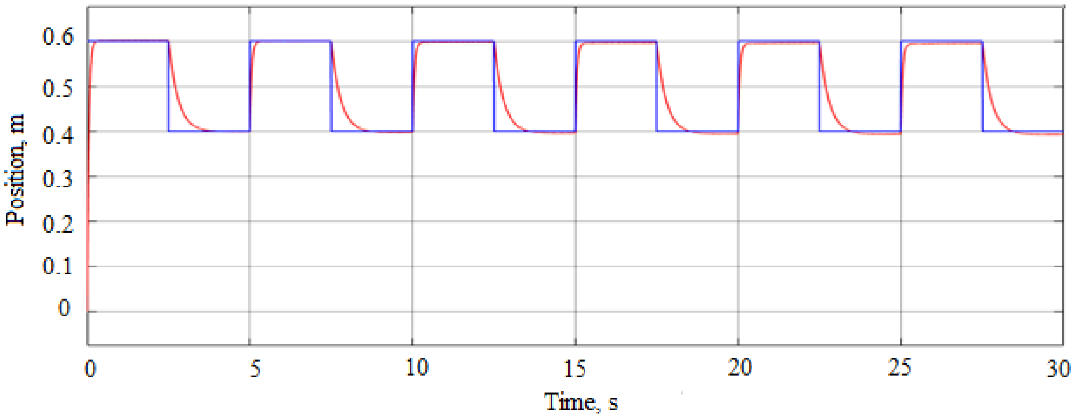

Figure 7.6. Simulation model output for Pulse input.

Table 7.4. System control model PI controller settings for Pulse input.

Parameter	**Value**
K_P	48
K_I	2

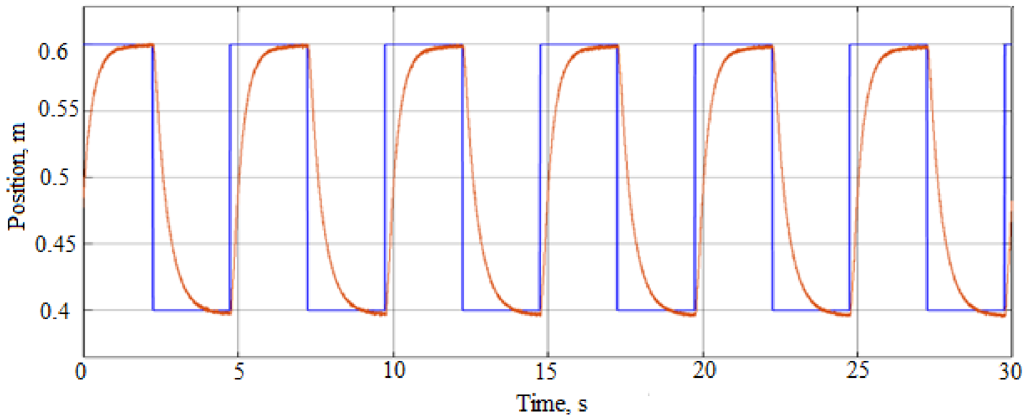

Figure 7.7. Hydraulic slider output for Pulse input.

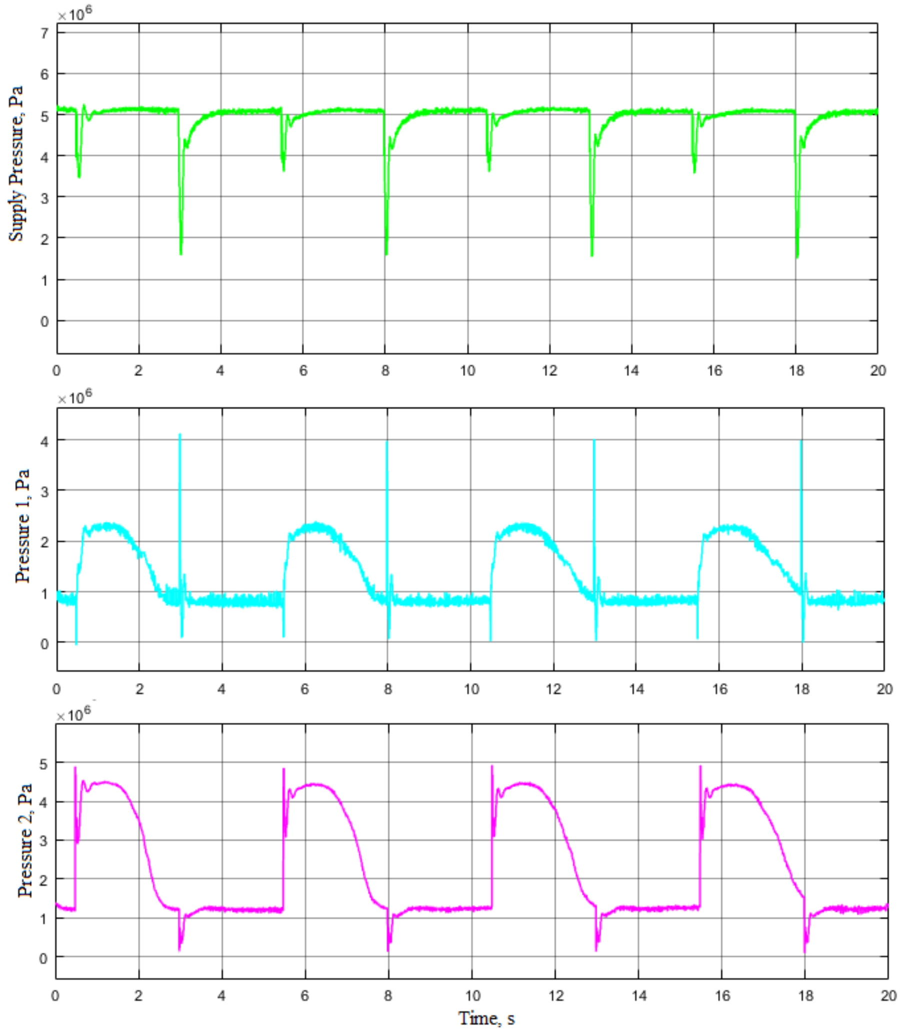

Figure 7.8. Hydraulic system pressure charts for Pulse input.

4) Sinusoidal Input

The sine wave is the last type of disturbance the system was tested with. Both real-time control and simulation model PI controllers settings (Tables 7.5. and 7.6) and responses (Figures 7.9 and 7.10) are presented below.

Table 7.5. Simulation model PI controller settings for Sinusoidal input.

Parameter	Value
K_P	58
K_I	1

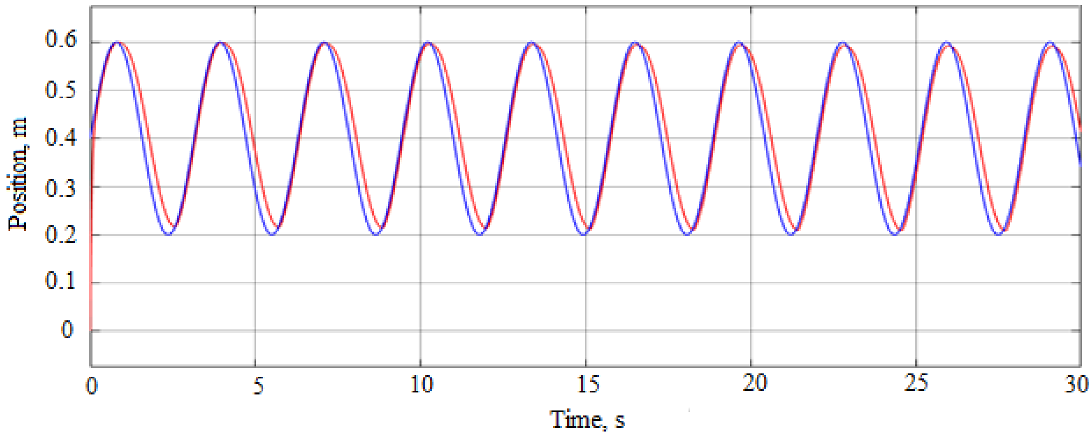

Figure 7.9. Simulation model output for Sinusoidal input.

Table 7.6. System control model PI controller settings for Sinusoidal input.

Parameter	Value
K_P	110
K_I	8

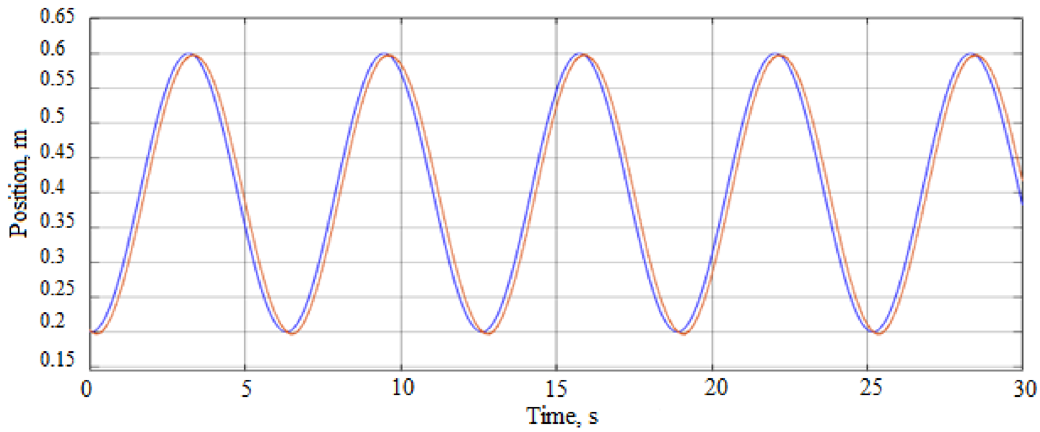

Figure 7.10. Hydraulic slider output for Sinusoidal input.

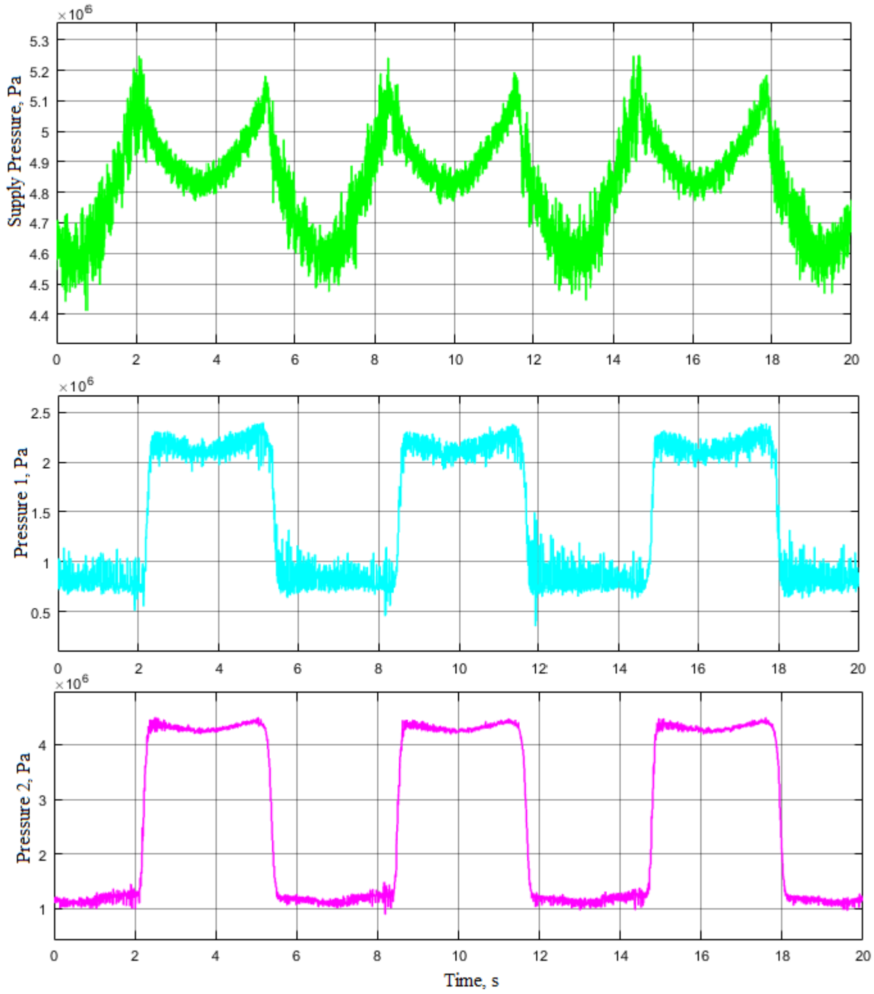

Figure 7.11. Hydraulic system pressure charts for Sinusoidal input.

8. DISCUSSION AND CONCLUSION

Using the chain of Beckhoff Embedded PC-TwinCAT 3 real-time environment-MATLAB/SIMULINK programming environment is a powerful tool for industrial systems control. This chain can be utilized for a wide variety of individual user applications due to its multipurpose principle and simplicity. Each of the control scheme components can be adjusted according to a user task in “on the fly” mode and does not require control algorithm rebuilt. Case studies, described in this work, and laboratory hydraulic system control algorithm implementation results show that the link between the chain components is robust and implementation method is straightforward. Considering the fact that most of the industrial manufacturers are prone to develop their own software environments for hardware programming and control, the presence of MATLAB/SIMULINK in this chain makes it valuable and worth to study for control engineers at the first place.

There are several ways of how the MATLAB/SIMULINK model can be used for real-time control in TwinCAT 3 environment. The main principle of integration chosen for this work is based on using C++ code generated with the help of Simulink Coder for TwinCAT 3 target system and establishing the connection between two environments in MATLAB/SIMULINK external mode. This method requires a powerful remote PC connected to Beckhoff Embedded PC in order to process all the necessary data in real-time operation mode and develop a control algorithm with required structure. It is also important to have all the necessary software and drivers installed on the remote platform. In any other aspects this method has no significant hardware and software requirements and limitations.

TwinCAT 3 integration with MATLAB/SIMULINK is described by Beckhoff Automation in their manuals, but for the laboratory of Intelligent Machines in Lappeenranta University of Technology it is the first time experience of its implementation and testing. During the case studies the integration method was studied and described, and this material can be used as step-by-step instructions of how it can be fulfilled for particular industrial applications. Preliminary studies and tests proved that these two environments are compliable with each other and have no significant data exchange delay. Control tasks, described in Case Study 1 and 2 as well as hydraulic system control scheme showed that the described control algorithm development and implementation method using the integration of two environments is suitable for variety of applications for different levels of complexity.

Hydraulic system test results show that the laboratory system follows control tasks in desired manner together with the system's simulation model. Comparison of these results makes it possible to conclude that the PI controller tuning method and chosen parameters were selected in a proper way both for simulation and real system. The difference between PI controller settings is caused by real system phenomena's neglected in simulation model (e.g. hardware influence, transients, cylinder and hose flexibility, etc.). Control tasks for the system presented as MATLAB/SIMULINK output data is not misrepresented by actuators, no errors were occurred during tests, no data losses and distortions. Result curves show that the system is stable, its response for three typical disturbances is adequate, the system's load has no oscillations or sudden/jerky motions, the system pressures vibrate in acceptable form. Overall test results allow concluding that the system desired behavior is reached, which matches the research goals. Mathematical model is reliable and corresponds the real system parameters.

During the research and studying the main aspects of this work several drawbacks of the described method were discovered. In MATLAB/SIMULINK environment the number of blocks for model C++ code generation is limited. In order to successfully build the model for export to TwinCAT 3 target system the total amount of blocks must be less than one hundred, which in some cases might be an obstacle for sophisticated control algorithm development. Also MATLAB/SIMULINK interface is not supplied with advanced real-time data acquisition tools. Gauges and Scopes from Dashboard library do not operate in external mode, and the only way to display and capture the data in real-time is to use classic Scope or Display blocks. To overcome this problem one might want to use TwinCAT 3 built-in visualizations, which provide a whole spectrum of data measurement and indication tools. One more issue that needs to be considered is data type mismatch. Every output/input block must have appropriate type of data to be utilized in TwinCAT 3 environment, where hardware I/O have predefined types which cannot be changed by user. Data type mismatching does not allow linking the MATLAB/SIMULINK model with hardware, thus the model configuration sometimes might be not compliable (e.g. when using such blocks as Mux/Demux) and lead to data flows entanglement. It is very important to make sure that every input and output port, and in some cases even operation blocks, have proper data type settings.

Many of the industrial companies can have profits from using the control method described in this thesis. Beckhoff hardware has several prospects nowadays cause of wide variety of products and price ranges, performance indexes and the company's rapid growth. TwinCAT 3 is free software which is a powerful tool in a stand-alone version, and MATLAB/SIMULINK integration makes it even more powerful. Combining the features of these two software packages allows using the hardware in more efficient way, especially in the sphere of hydraulics where due to large power ranges computing abilities of industrial controllers play significant role.

LIST OF REFERENCES

Beckhoff Automation GmbH. Beckhoff short profile. Available: https://download.beckhoff.com/download/press/Short_profile_Beckhoff.pdf

Beckhoff Automation. 2012 TwinCAT 3 | eXtended Automation (XA). Available: https://download.beckhoff.com/download/document/catalog/TwinCAT_3_Booklet.pdf

Beckhoff Automation. 2014. Product Overview. Available: https://download.beckhoff.com/download/Document/Catalog/Beckhoff_Products_2014.pdf

Beckhoff Automation. 2018. TC3 C++. Manual**.** 316 p.

Beckhoff Automation. 2018. TC3 Target for Matlab®/Simulink®. Manual. 95 p.

Beckhoff Automation. CX2030 | Basic CPU module. Available: https://download.beckhoff.com/download/Document/Catalog/Main_Catalog/english/separate-pages/Embedded_PC/CX2030.pdf

Beckhoff New Automation Technology. Embedded PC, Modular DIN rail IPCs and Industrial Motherboards catalogue. Available: www.beckhoff.com/Emdedded-PC

Bosch Rexroth AG Hydraulics. 2004. Variable Axial Piston Pump A10VSO. 36 p.

Bosch Rexroth AG Hydraulics. 2005. Pressure transducer with on-board electronics, type HM 18. 8 p.

Bosch Rexroth AG Hydraulics. 2010. Type 4WRPEH 6 Valve. 16 p.

Galal Rabie, M. 2009. Fluid power engineering. McGraw-Hill Osborne Media. 443 p.

Handroos, H., Roozbahani, H, Wu, H. 2011. Real-time simulation based robust adaptive control of hydraulic servo system. Mechatronics (ICM), IEEE International Conference. p. 6-10.

Liu, J., Handroos, H., Haario, H. and Wu, H. 2009. Reliability analysis of a model for servo hydraulic system by utilizing MCMC, 7th Int'l Conf. on Fluid Power and Control, China. p. 8-12

Roozbahani, H. 2011. Novel force control methods in a teleoperation system of a hydraulic slider. Lappeenranta University of Technology. 161 p.

Witherspone, I. 2014. Comparison between dSPACE and NI systems based on real-time intelligent control of a teleoperated hydraulic servo system. Lappeenranta University of Technology. 76 p.