SCHOOL OF ENGINEERING AND INFORMATION TECHNOLOGY

Honeywell Experion System:

Configuration, Simulation and Process Control Software Interoperability

A thesis submitted to the school of Engineering and Information Technology, Murdoch University in partial fulfillment of the requirements for the degree of

Bachelor of Engineering Honours [BE(Hons)]
Instrumentation and Control Engineering
Industrial Computer Systems Engineering

Simon Godfrey

Project Supervisor: Associate Professor Graeme Cole

November 2016

Author's Declaration

I, Simon Godfrey, declare that this thesis is my own account of my research and contains as its main content, work which has not previously been submitted for a degree at any tertiary education institution.

Body text word count (Excluding Appendices): 15,171

Abstract

The use of automated distributed control systems has become a widely discussed topic as industry attempts to maximise efficiency. An increase in automation technology has resulted in demand for "industry ready" graduate engineers with knowledge and experience with such technologies. Murdoch University (Murdoch) provides students with exposure to industry standard automation systems such as Honeywell's Experion Process Knowledge System (PKS), a Distributed Control System (DCS) designed for the control and optimisation of industrial plants.

Presently, Murdoch uses Experion to monitor and control its pilot plant, exposing instrumentation and control students to the configuration and control of an industrial plant. Additionally, Murdoch holds a second Experion simulation license providing the ability to program and simulate process and control strategies. This project scope was to explore several key areas of the Experion simulation system followed by the development of a series of learning materials to facilitate the teaching of the Experion system to students for a new unit to be introduced at Murdoch University in 2017.

The methodology adopted to achieve the project outcome involved developing a comprehensive understanding of Experion and its associated applications, before building a series of example simulation programs for the purpose of implementing and testing a variety of Experion's control strategies. Additionally, real-time control of the Experion simulation programs, using third party process control software, was achieved. The successful interoperability of MATLAB, Simulink, and LabVIEW with Experion provides an avenue for implementing advanced control strategies both in simulation and on Murdoch's pilot plant.

At the conclusion of the project, an extensive list of learning materials was produced, providing comprehensive procedures to enact tasks within the Experion system. This includes initial setup and configuration, development of simulated programs and associated Human Machine

Interface (HMI) displays, the implementation of control strategies, and third party process control software interoperability. These learning materials provide students with an enhanced learning experience, giving them the skills and exposure required to thrive in the automation engineering industry as a Murdoch University graduate.

Acknowledgements

Associate Professor Graeme Cole

Lecturer, Murdoch University

Mr Will Stirling

Technical Officer, Murdoch University

Mr David Pol

Engineering Student, Murdoch University

Instrumentation and Control Engineering

Industrial Computer System Engineering

Mr Michael Crooke

Engineering Student, Murdoch University

Electrical Power Engineering

Industrial Computer System Engineering

Table of Contents

List of Figures

List of Tables

List of Equations

Table of Appendices

List of Abbreviations

Abbreviation	Explanation
AUTO	Automatic
CAS	Cascade
CEE	Control Execution Environment
CM	Control Module
DCS	Distributed Control System
EMB	Enterprise Model Builder
HMI	Human Machine Interface
I/O	Input / Output
ICSE	Industrial Computer Systems Engineering
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
MATLAB	Matrix Laboratory
MEDE	Microsoft Excel Data Exchange
Murdoch	Murdoch University
MV	Manipulated Variable
OLE	Object Link Embedded
OP	Operating Point
OPC	OLE for Process Control
PID	Proportional Integral Derivative
PKS	Process Knowledge Suite
PV	Process Variable
SCM	Sequential Control Module
SP	Setpoint
SVE	Shared Variable Engine

1 Introduction

1.1 Overview of the Honeywell Experion Process Knowledge System

Honeywell's Experion Process Knowledge System (PKS) is a Distributed Control System (DCS) integrating an advanced process automation platform, with a number of software applications used for the configuring, monitoring and control of an industrial process (Honeywell 2016c). A DCS characterises a system controlled by one or more controllers distributed throughout a process, connected in a network for the purpose of acquiring, communicating and monitoring data (Schaschke 2014). See Figure 1 for a simplified DCS representation.

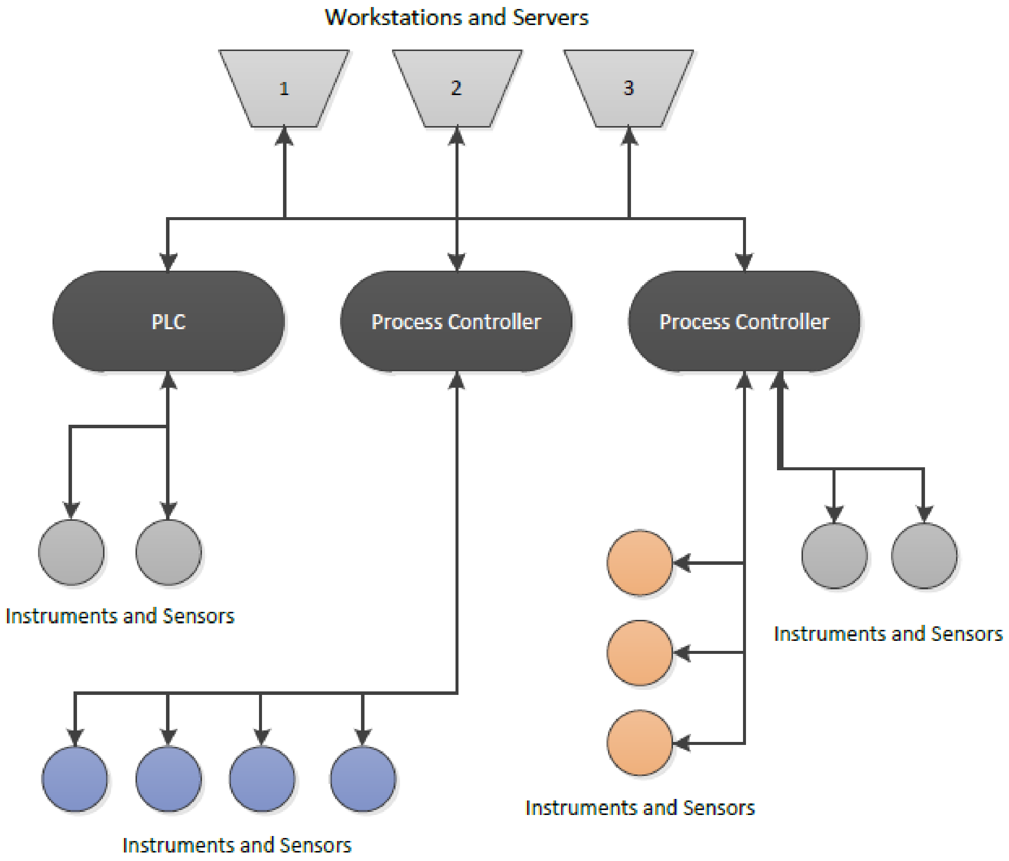


Figure 1: Basic DCS Architecture

Murdoch acquired the Experion platform and its associated C300 controller with the upgrade of Murdoch's Bayer process pilot plant in 2011 (Meiri 2015). Exposure to Experion's system architecture and programming methods provides students with an insight into standard engineering practise for process automation and control in an industrial setting.

1.2 Project Scope

The primary aim of this project was the production of a series of specific deliverables to facilitate the teaching of Honeywell Experion PKS for a new unit to be introduced in semester 1 2017 at Murdoch University.

A previous thesis indicated that the Experion software was updated in 2014 to Experion release RS-430.1 (Meiri 2015). This release made such significant changes to the operation of the Experion system that learning materials, previously developed by Gumireddy (Gumireddy 2013) and Nain (Nain 2013) would require revision. In addition, the scope of this thesis was to update and expand the learning materials to ensure the documentation became comprehensive in detail. Utilisation of the resultant learning materials will assist students develop their technical knowledge of the system, whilst guiding them through the configuration and programming of control strategies. This will improve their understanding of the configuration, programming and data exchange capabilities of Honeywell's Experion environment.

The scope of the project was broken into three distinct phases:

1. Phase 1 – Experion configuration, simulation and control;
2. Phase 2 – Object Link Embedded (OLE) for Process Control (OPC) data exchange;
3. Phase 3 – Documentation of learning materials.

1.2.1 Phase 1 – Experion Configuration, Simulation and Control

Phase 1 of the project was to understand the Experion system in its entirety. This included three key areas; system architecture and navigation of the environment (and its associated applications), the configuration and programming of these applications, and the simulation and testing of developed programs.

1.2.1.1 System Architecture and Navigation

A comprehensive understanding of the Experion system architecture was critical in ensuring the production of clear and consistent learning materials. Early stages of the project were to concentrate on gaining knowledge of the Honeywell Experion PKS system architecture, whilst navigating the Experion environment and its associated applications (Station, Control Builder, HMIWeb Display Builder and Enterprise Model Builder).

1.2.1.2 Configuring and Programming within Experion

With an understanding of Experion's architecture, focus was then to be shifted to the configuration and programming of the system. This included the initial setup of servers and asset hierarchy, programming of controllers and development of control modules.

1.2.1.3 Simulation and Testing

Emphasis was placed on the development of complete simulation examples. A number of examples were to be created to document the development and configuration of control strategies within the Experion environment. The examples to be developed were:

- Basic mathematical and logic equations;
- First order system;
- Conventional Feedback Proportional Integral Derivative (PID) control;
- Cascade PID control; and
- Model-based predictive control.

To validate the developed programs and control strategies, a number of tests and simulations were to be conducted for analysis. As the programs developed were primarily process simulations and control strategies, the operation of the controllers were to be verified by replicating the strategy with a third party control simulation application (Simulink) for comparison.

1.2.2 Phase 2 – OPC Connectivity and Data Exchange

The Industrial Computer Systems Engineering (ICSE) major of Murdoch's Bachelor of Engineering provides a focus on the exchange of process data over a variety of software platforms and applications. Included in these different communication methods is OPC. Honeywell integrates OPC capability within the Experion environment to provide users with the advantages of OPC communication (Honeywell 2016b). With correct implementation and configuration of Experion OPC, other software applications with OPC communication ability (such as OPC Quick Client, MATLAB, Simulink and LabVIEW) are able to exchange process data freely.

Experion OPC communication with third party applications, to this point, had not been achieved and validated at Murdoch. The desired outcome of Phase 2 was to establish effective communication between Experion and various OPC compliant applications. It was intended to achieve this goal by applying a systematic approach:

1. Research and implement the configuration requirements for OPC within Experion;
2. Achieve OPC communication using OPC Quick Client;
3. Achieve dynamic communication with programs such as LabVIEW and MATLAB;
4. Test and validate communication via simulations;

1.2.3 Phase 3 – Documentation of Learning Materials

Upon the successful validation of each of the objectives outlined in Phase 1 and 2, the revision of existing and production of new learning materials was to be performed. Documentation was then to be tested by a student external to the project, to provide feedback and identify any areas of improvement.

2 Technical Reviews

2.1 Honeywell Experion Process Knowledge System

Honeywell's Experion PKS is an open control and safety system designed to expand the role of a DCS by combining the functionality of a DCS with complete plant-wide infrastructure, to facilitate data acquisition and control, to optimise processes (Honeywell 2014a).

Presently, Murdoch uses the Experion system for monitoring and control of the simulated Bayer process pilot plant, located in Energy and Engineering Building. The Experion platform was commissioned in the pilot plant after it was transferred from the Rockingham campus to the South Street campus in 2009 (Meiri 2015).

With the purchase and implementation of the Honeywell Experion system on the pilot plant, a second simulation Experion license was purchased. This project was to expand the knowledge and documentation of the simulation Experion platform.

2.1.1 Experion System Topology

Using Murdoch University's pilot plant facility as an example, the topology of the Experion system is indicated in Figure 2. The operation of the system involves the user sending commands through the client workstation applications to the Experion server and subsequently to the process controller, resulting in appropriate changes to the instruments in the pilot plant. As this project is concerned with the simulation Experion license, the physical controller and plant's processes were simulated within the Experion environment.

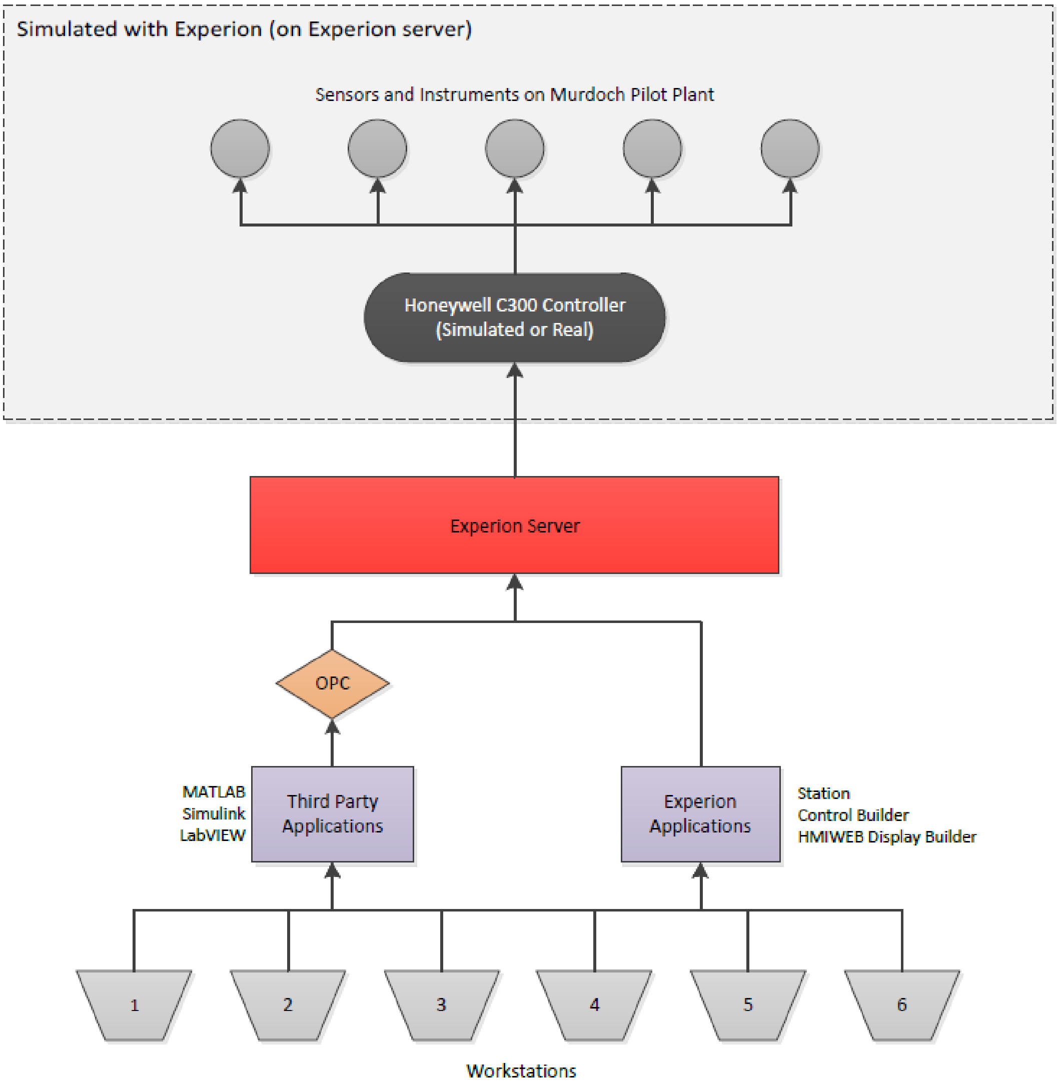


Figure 2: Experion System Topology

2.1.2 Honeywell C300 Simulated Controller and Control Execution Environment

As identified in the overview of Experion PKS, the Experion system is a DCS, integrating one or more controllers for the acquisition of data for process control. Used on Murdoch's pilot plant, Honeywell's C300 is one such controller, communicating with I/O modules to perform programmed control algorithms on the system (Honeywell 2016a). The second control license acquired by Murdoch (Experion2) provides the ability to simulate the C300 controller on the Experion server.

2.2 Experion Tools and Applications

Honeywell Experion PKS integrates an advanced process automation platform with a number of software applications used for configuring, monitoring and control of an industrial process (Honeywell 2016c). A review of each of these applications will be presented in the following sections.

2.2.1 Configuration Studio

Configuration Studio is the central location for the configuration of the Experion system. Each of the individual tools and applications required to configure, monitor and control the Experion system are launched from Configuration Studio (Honeywell 2014e). Within Configuration Studio is a list of tasks for the configuration of the system (Figure 3), and upon selection of a task, the appropriate application will be launched. These applications include:

- Enterprise Model Builder (EMB);
- Control Builder;
- Quick Builder;
- System Displays; and
- HMIWeb Display Builder.

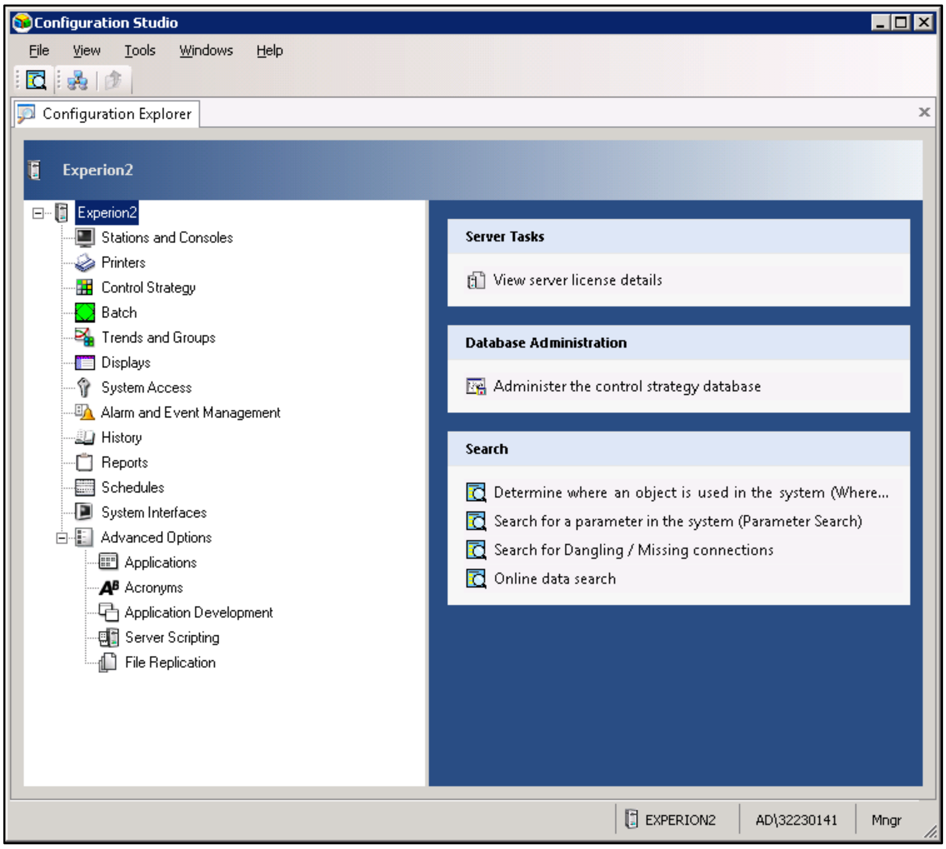


Figure 3: Configuration Studio Home

2.2.1.1 Enterprise Model Builder (EMB)

The Enterprise Model Builder is the graphical application used for configuring the Enterprise Model. The Enterprise Model reflects the configuration of the physical system of a user's enterprise including assets, materials and people (Honeywell 2014c).

Important to mention is Honeywell's use of item names. All items within the Enterprise Model are created with a unique "tag name", used by Experion as a means of structuring and identifying items (Honeywell 2014c) for use with Experion's applications.

2.2.1.2 Control Builder

Control Builder is Experion's graphical tool for the building of control strategies (Honeywell 2014b). This includes the creation of simulated C300 controllers, as well as the programming of sequential and continuous control modules.

Continuous Control Module (CM)

Continuous Control Modules (CM) within the Experion system are programs designed to continuously execute programmed control operations (Honeywell 2014a). The CM is a container for the insertion and programming of various function blocks. Experion hosts an expansive range of function blocks ranging from Boolean logic and numeric blocks through to advanced model-based controller blocks. Figure 4 displays a discretised first order system, programmed within the control module located on the Experion2 server. This example is discussed in further detail later in this report.

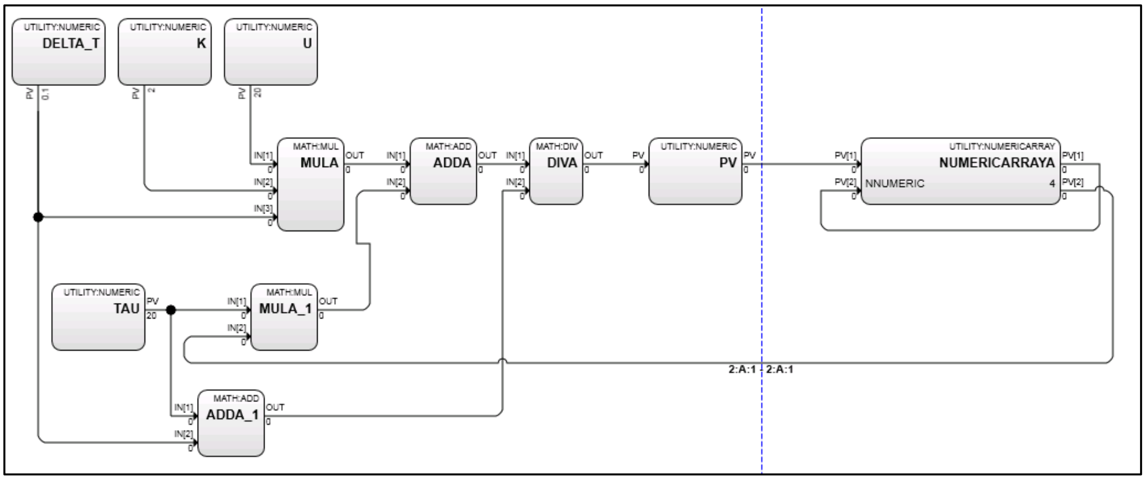
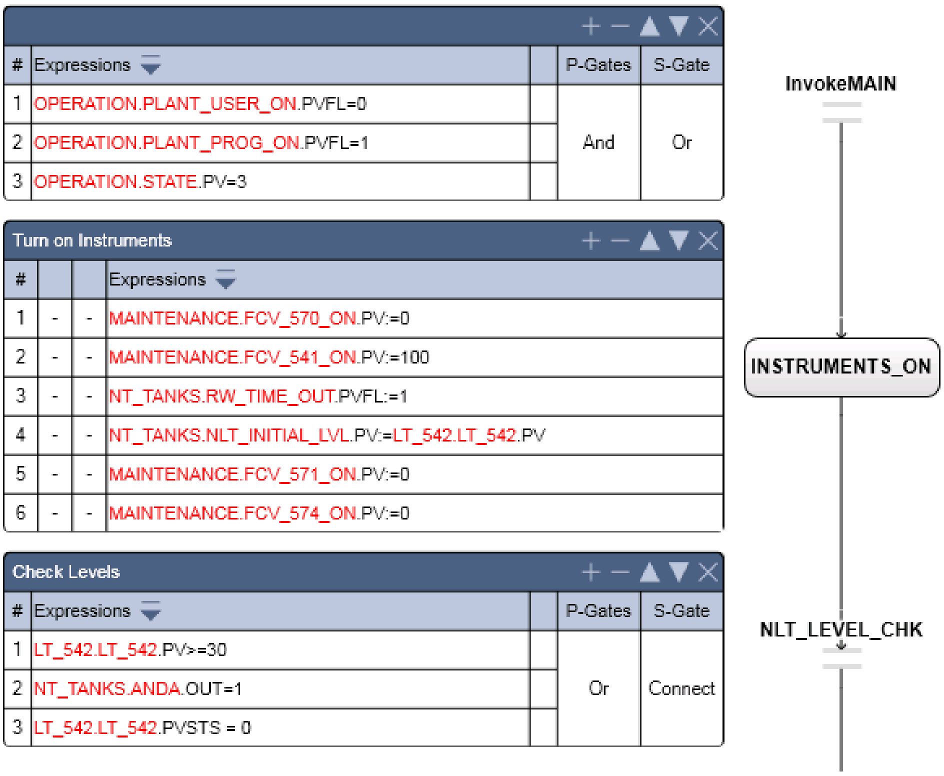


Figure 4: Control Module Example – Discretised First Order System

Sequential Control Module (SCM)

Operations such as start-up, shutdown and maintenance activities are often required to be executed sequentially. Sequential Control Modules (SCM) interact with one or more continuous control modules to achieve a sequential mode of operation. Predefined conditions invoke the SCM to activate its sequence of operations, transitioning between each step only once all conditions have been met. Figure 5 displays a small section of an example maintenance program developed for the Murdoch pilot plant.



#	Expressions		P-Gates	S-Gate
1	OPERATION.PLANT_USER_ON.PVFL=0		And	Or
2	OPERATION.PLANT_PROG_ON.PVFL=1			
3	OPERATION.STATE.PV=3			

Turn on Instruments

#			Expressions
1	-	-	MAINTENANCE.FCV_570_ON.PV:=0
2	-	-	MAINTENANCE.FCV_541_ON.PV:=100
3	-	-	NT_TANKS.RW_TIME_OUT.PVFL:=1
4	-	-	NT_TANKS.NLT_INITIAL_LVL.PV:=LT_542.LT_542.PV
5	-	-	MAINTENANCE.FCV_571_ON.PV:=0
6	-	-	MAINTENANCE.FCV_574_ON.PV:=0

Check Levels

#	Expressions		P-Gates	S-Gate
1	LT_542.LT_542.PV>=30		Or	Connect
2	NT_TANKS.ANDA.OUT=1			
3	LT_542.LT_542.PVSTS = 0			

Figure 5: Sequential Control Module Example - Maintenance Program

2.2.1.3 *Quick Builder*

Quick builder is Experion's graphical tool for the creation and configuration of equipment, such as stations, controllers and printers (Honeywell 2014e).

2.2.1.4 *System Displays*

The Experion environment includes a comprehensive set of system displays that can be called from Configuration Studio for the configuration of the system, the management of alarms and events and the monitoring of process data (Honeywell 2014e).

System Configuration

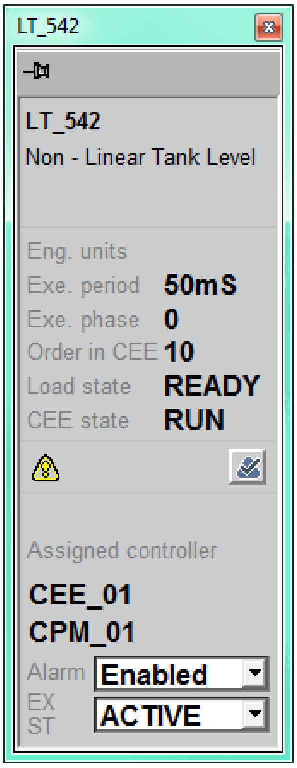
Displays such as popup dialog boxes (Figure 6) indicate the current configuration of a user's system. These displays are designed to assist engineers and developers for the configuration of the system parameters.

Alarm Management and Events

Experion's alarm management and event displays allow users to view, respond to, and acknowledge alarms quickly and efficiently. The alarm displays have added inbuilt features for the filtering and sorting of alarms and events according to priority, location and asset hierarchy (Honeywell 2014e).

Process Monitoring

Experion also provides an extensive range of system displays for the displaying and monitoring of process data including trend displays as shown in Figure 7.



LT_542

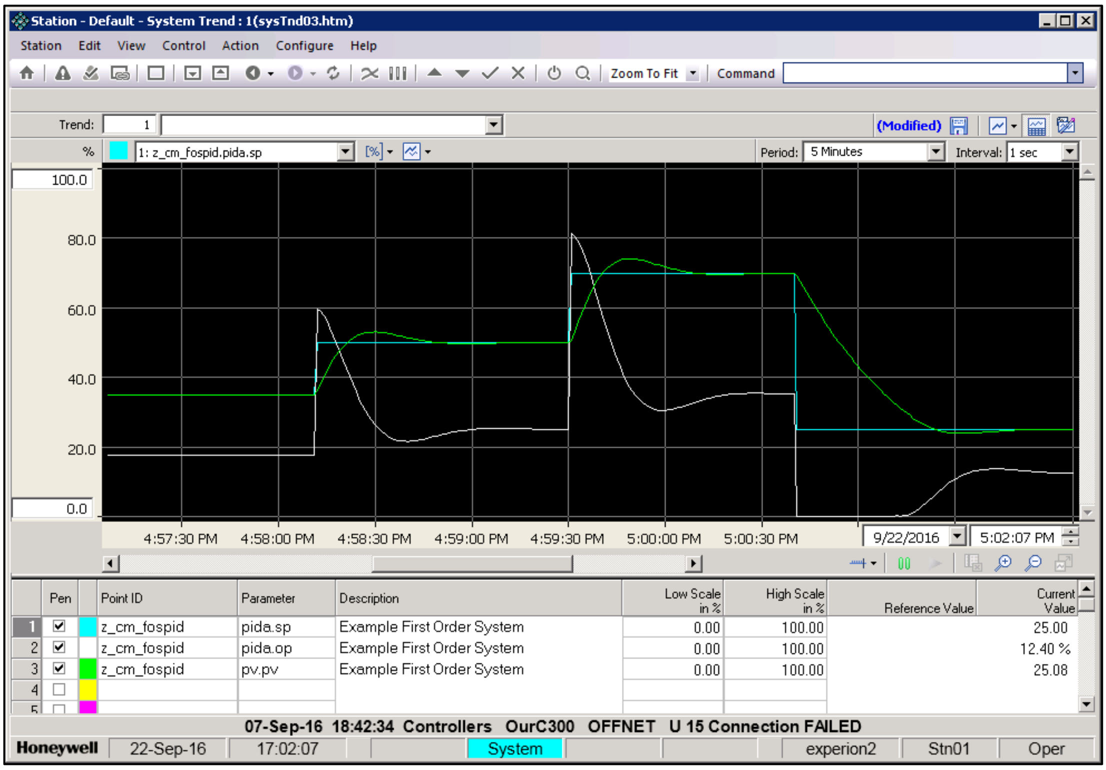
LT_542
Non - Linear Tank Level

Eng. units
Exe. period 50mS
Exe. phase 0
Order in CEE 10
Load state READY
CEE state RUN

Assigned controller
CEE_01
CPM_01

Alarm Enabled
EX ST ACTIVE

Figure 6: Faceplate Popup



Trend: 1
% 1: z_cm_fospid.pida.sp
Period: 5 Minutes
Interval: 1 sec
9/22/2016 5:02:07 PM

	Pen		Point ID	Parameter	Description	Low Scale in %	High Scale in %	Reference Value	Current Value
1	☑	🟦	z_cm_fospid	pida.sp	Example First Order System	0.00	100.00		25.00
2	☑	⬜	z_cm_fospid	pida.op	Example First Order System	0.00	100.00		12.40 %
3	☑	🟩	z_cm_fospid	pv.pv	Example First Order System	0.00	100.00		25.08
4	□	🟨							
5	□	🟪							

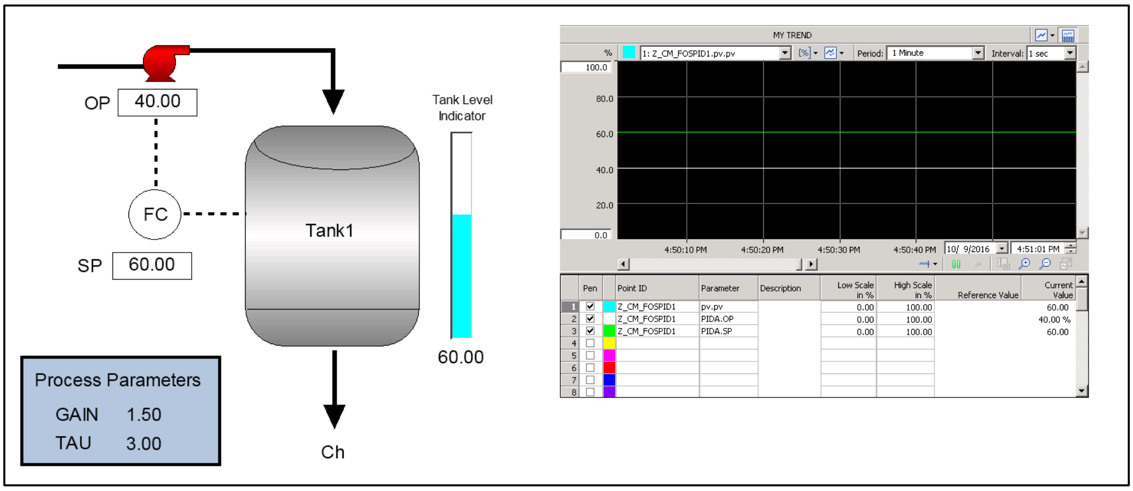
07-Sep-16 18:42:34 Controllers OurC300 OFFNET U 15 Connection FAILED

Honeywell | 22-Sep-16 | 17:02:07 | System | experion2 | Stn01 | Oper

Figure 7: Process Monitoring Trend

2.2.1.5 HMIWeb Display Builder

HMIWeb Display Builder is Honeywell's graphical drawing application used for the creation of customised Human Machine Interface (HMI) displays, to be viewed with Station (Honeywell 2014d). Figure 8 displays the HMI developed to assist in the visualisation, monitoring and control of a first order system with feedback control (tank with level controller).



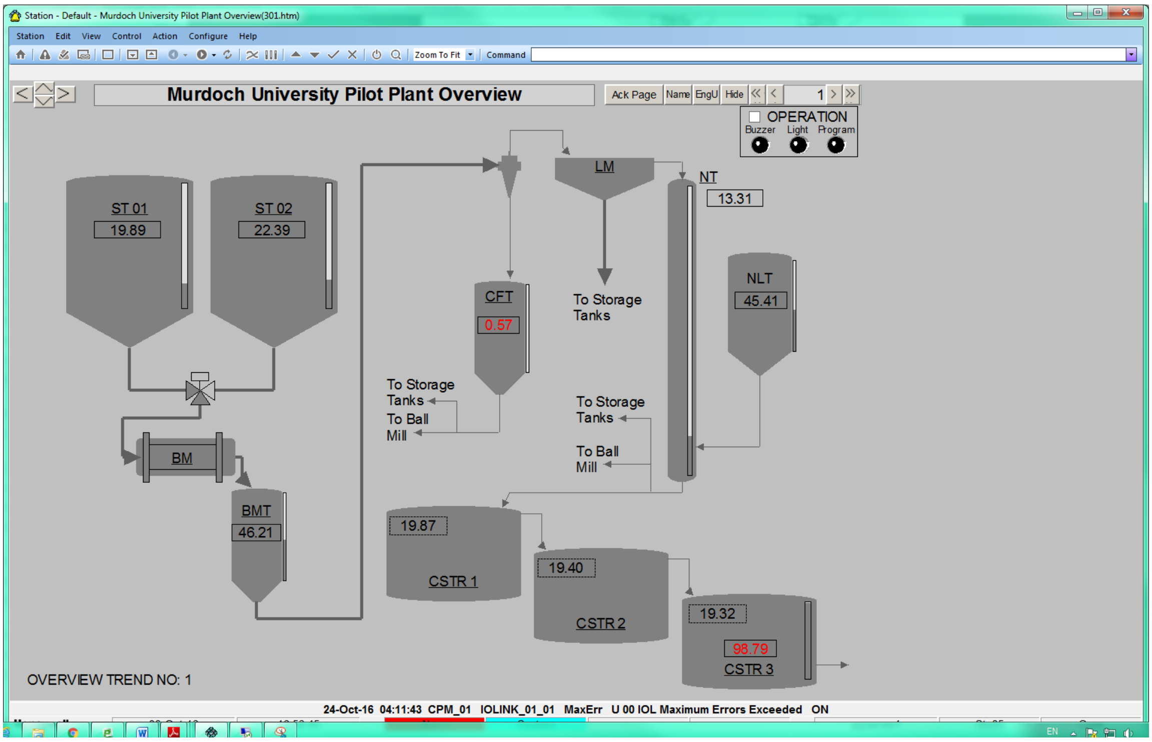
1: Z_CM_FOSPID1.pv.pv
Period: 1 Minute
Interval: 1 sec
10/ 9/2016
4:51:01 PM

	Pen		Point ID	Parameter	Description	Low Scale in %	High Scale in %	Reference Value	Current Value
1	☑	🟦	Z_CM_FOSPID1	pv.pv		0.00	100.00		60.00
2	☑	⬜	Z_CM_FOSPID1	PIDA.OP		0.00	100.00		40.00 %
3	☑	🟩	Z_CM_FOSPID1	PIDA.SP		0.00	100.00		60.00
4	□	🟨							
5	□	🟪							
6	□	🟥							
7	□	🟦							
8	□	🟪							

Figure 8: Example HMI Display - First Order System with Feedback Control

2.2.1.6 Station

Station is Experion's human interface for the viewing of developed displays (Honeywell 2014a). Using Station, it is possible to monitor and control the linked system. As an example, Figure 9 presents the HMI display for monitoring and controlling Murdoch University's pilot plant.



24-Oct-16 04:11:43 CPM_01 IOLINK_01_01 MaxErr U 00 IOL Maximum Errors Exceeded ON

Figure 9: Station Human Machine Interface – Murdoch University Pilot Plant

2.3 OPC Data Communication

OPC is a standard interface for data communication that exchanges information from a data source, such as a server, to any client application without the need for specific communication protocols (Honeywell 2016b).

As discussed in the project scope, OPC communication had not been operational since the update in software. The desired outcome of project was successful communication between Experion and OPC compliant software applications, including OPC Quick Client, MATLAB, Simulink and LabVIEW.

2.3.1 Kepware OPC Quick Client

Kepware OPC Quick Client is an application designed to confirm the configuration of both local and remote OPC servers (Kepware, Inc. 2015). Using OPC Quick Client, users can read from, and write to, an OPC server to ensure communication has been established correctly.

2.3.2 MATLAB

Matrix Laboratory (MATLAB) is an interactive, high-level text based program for technical computing developed by MathWorks Inc. (Dukkipati 2008). MATLAB integrates complex numerical and mathematical computation with data visualisation to provide a powerful programming platform.

MATLAB offers an open architecture, allowing installation of a variety of toolboxes to assist in areas such as control system design, signal processing and data communication. MATLAB's OPC Toolbox assists in providing access to live and historical OPC data (MathWorks 2016a). This toolbox was used to develop successful real-time OPC communication between Experion and MATLAB.

2.3.3 Simulink

Contained within the MATLAB umbrella is MathWorks' block diagram environment for simulation and control, Simulink. Simulink offers a range of function block libraries and solvers for the effective modelling, simulation and control of dynamic systems (MathWorks 2016b). Simulink was used extensively for testing processes and control strategies developed within the Experion environment.

2.3.4 LabVIEW

Laboratory Virtual Instrument Engineering Workbench (LabVIEW) is an integrated development environment, utilising a graphical programming language for the creation of measurement and control systems developed by National Instruments (National Instruments 2016c). Similar to MATLAB, LabVIEW offers an open architecture, enabling interoperability with other software platforms and hardware devices. This interoperability of software platforms includes data exchange via OPC communication.

3 Project Constraints

Two main project constraints encountered during the completion of the project.

3.1 Technical Issue with Experion Server

During Phase 1, difficulties with configuring the Experion simulation server (Experion2) were encountered such that Experion simulations were not operational. To ensure project progress could continue, early testing, configuration and programming of the Experion system was conducted on the Pilot Plant server known as “Ppserver1”.

Technical staff member Mr Will Stirling attempted various configuration changes, and after several weeks the Experion simulation server (Experion2) was made operational. Control modules and control strategies developed on Ppserver1 were then able to be imported to Experion2.

3.2 OPC Data Access Restrictions

At the commencement of Phase 2, difficulties were encountered with the exchange of data between Experion and third party software applications, whereby data access was restricted. Project progress was delayed while the data access restriction was resolved by Mr Sterling, by the re-configuration of security access.

Additional details on the resolution of each of these issues can be found in Appendix A and Appendix B.

4 Delivery against Project Scope – Phase 1

Phase 1 involved three key areas of focus:

1. System architecture and navigation;
2. Configuration and programming; and
3. Simulation and testing.

By systematically progressing through each area, it was possible to successfully configure the Experion system, develop simulated programs and perform validation tests on control strategies implemented.

4.1 System Architecture and Navigation

With no prior experience with Honeywell's Experion, it was necessary to develop a comprehensive understanding of the Experion system architecture to produce clear and consistent learning materials. This involved gaining a solid understanding of the role each application played within the overall system, which was achieved during initial stages of the project.

4.2 Configuration and Programming

The configuration and programming of the Experion system was divided into several sub sections, addressing each of the tasks involved to successfully configure the Experion system for control strategy implementation. These involve:

1. Enterprise Model configuration;
2. Controller and control module configuration; and
3. Process monitoring techniques.

4.2.1 Enterprise Model Configuration

Prior to the programming and testing of simulated control modules, a thorough understanding of the Experion Enterprise Model was required. Honeywell defines the Enterprise Model as a means of organising the system around the crucial items in the user's enterprise, such as assets, material, activities and people (Honeywell 2014c). As an example, considering Murdoch as the enterprise, the University's pilot plant is the system to be managed with Experion, with the assets being the physical hardware and control network.

Configuration of the Enterprise Model was achieved by use of Experion's Enterprise Model Builder. This application is launched from Configuration Studio by selecting the task to be performed, as displayed in Figure 10.

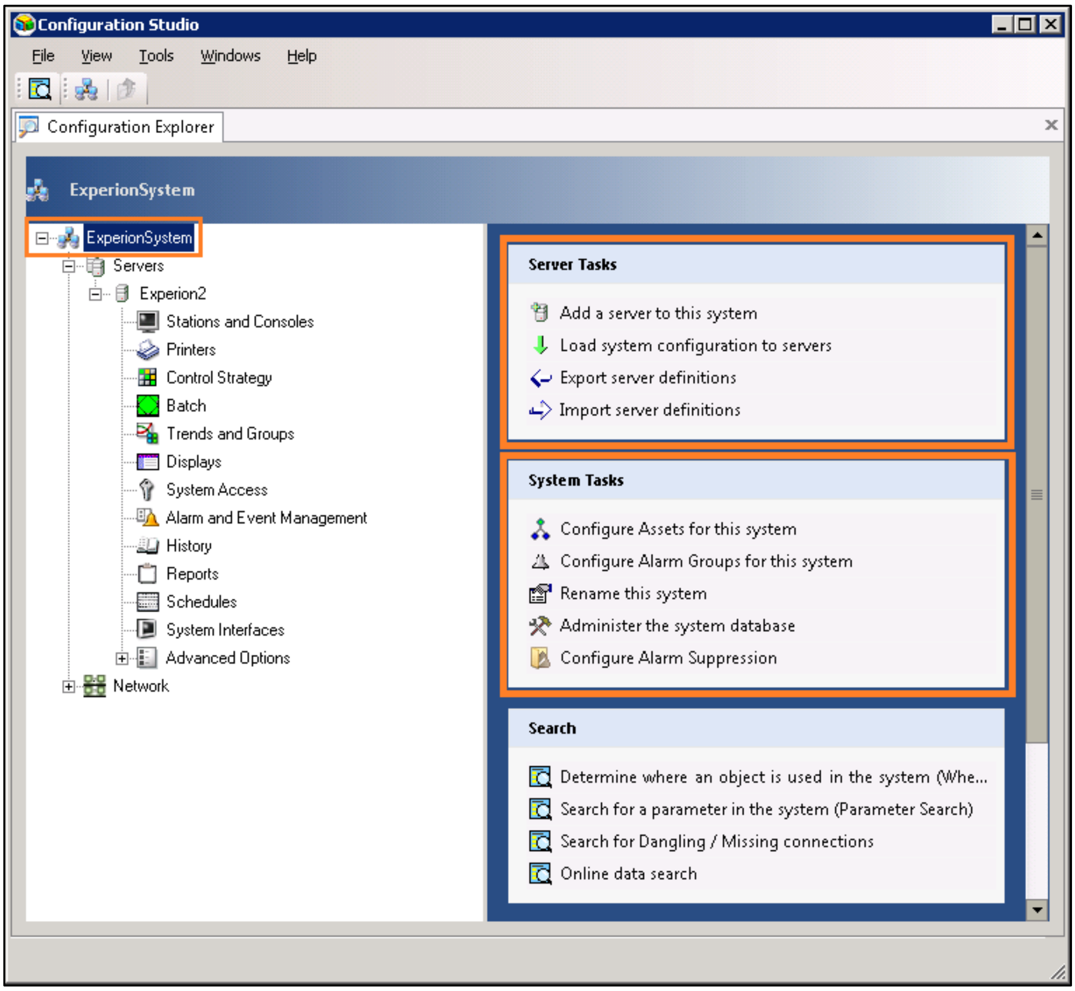

Configuration Studio
File
View
Tools
Windows
Help
Configuration Explorer
ExperionSystem
ExperionSystem
Servers
Experion2
Stations and Consoles
Printers
Control Strategy
Batch
Trends and Groups
Displays
System Access
Alarm and Event Management
History
Reports
Schedules
System Interfaces
Advanced Options
Network
Server Tasks
Add a server to this system
Load system configuration to servers
Export server definitions
Import server definitions
System Tasks
Configure Assets for this system
Configure Alarm Groups for this system
Rename this system
Administer the system database
Configure Alarm Suppression
Search
Determine where an object is used in the system (Whe...
Search for a parameter in the system (Parameter Search)
Search for Dangling / Missing connections
Online data search

Figure 10: Configuration Studio – Task list

With the knowledge of the Enterprise Model in hand, it was possible to confidently configure the Experion system for the future programming of control modules. This involved the configuration of the system model (Experion server) and asset hierarchy.

4.2.1.1 System Model (Server Configuration)

The system model signifies the boundaries of the system, defined by the servers created within the system (Honeywell 2014e). Servers are considered as assets within the Enterprise Model and are created within Configuration Studio, forming the storage location for system data. Both the Experion server and the simulation server were configured correctly after the technical issue described in Section 3.1 was resolved. Therefore, no configuration was required.

4.2.1.2 Asset Model and Configuration

Crucial to the operation of control modules and associated control strategies, the development of the asset model forms the heart of the Enterprise Model. The asset model is used to define the scope of responsibility, organise data points and references, navigate the Experion system, as well as manage alarms (Honeywell 2014a).

Honeywell defines an asset as a database entity that represents a particular physical item within that enterprise, such as plant equipment, facilities or buildings (Honeywell 2014e). This means various areas of the plant would be linked to their respective asset. For example, a control module created for monitoring the speed of a pump in a given plant may have a different linked asset than that of a control module designed to control the input of steam to a heated tank.

Additionally, Experion implements a hierarchical (child/parent) asset-based structure, allowing for effective implementation of scope of responsibility (i.e. the restriction of access to parts of the system). Rather than assigning access to individual client workstations or operators, it is possible to assign assets to a particular access group e.g. engineers, managers and operators.

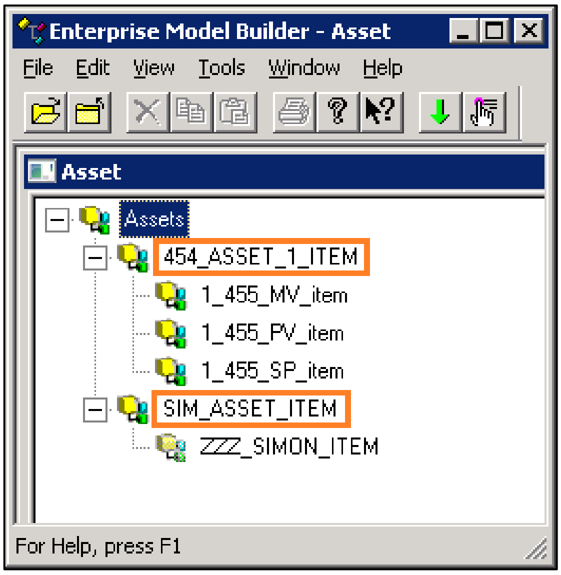
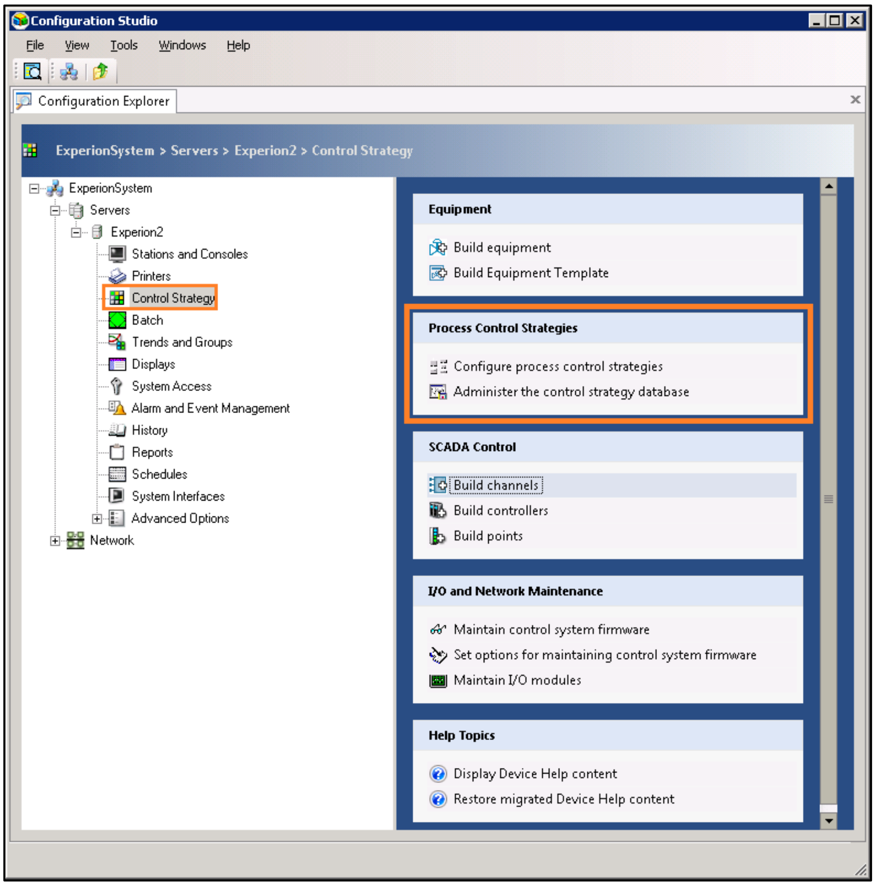


Figure 11: Asset Model

Figure 11 illustrates Experion's hierarchical asset model implemented on the Experion2 server, with parent assets indicated by the orange markings. For the configuration and testing throughout this project the asset SIM_ASSET_ITEM was used.

4.2.2 Controller and Control Module Configuration

With knowledge of the Enterprise Model, it was possible to research the requirements for control strategy building. A control strategy can be defined as a structured approach to defining a specific process using detailed information (Honeywell 2014b). The creation of these control strategies is achieved within Experion's Control Builder application, launched from the Configuration Studio by selecting the appropriate task (Figure 12).



Configuration Studio

File View Tools Windows Help

Configuration Explorer

ExperionSystem > Servers > Experion2 > Control Strategy

- ExperionSystem
  - Servers
    - Experion2
      - Stations and Consoles
      - Printers
      - Control Strategy
      - Batch
      - Trends and Groups
      - Displays
      - System Access
      - Alarm and Event Management
      - History
      - Reports
      - Schedules
      - System Interfaces
      - Advanced Options
  - Network

Equipment
- Build equipment
- Build Equipment Template

Process Control Strategies
- Configure process control strategies
- Administer the control strategy database

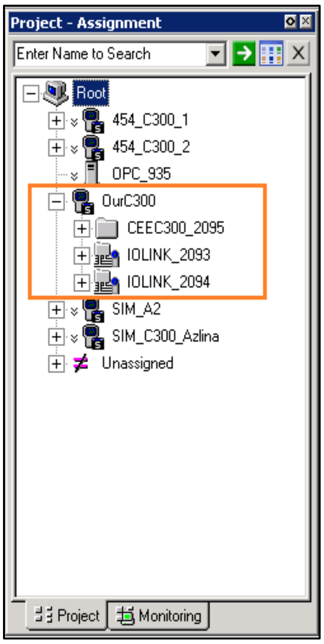
SCADA Control
- Build channels
- Build controllers
- Build points

I/O and Network Maintenance
- Maintain control system firmware
- Set options for maintaining control system firmware
- Maintain I/O modules

Help Topics
- Display Device Help content
- Restore migrated Device Help content

Figure 12: Configuration Studio - Control Builder Tasks

4.2.2.1 Simulated C300 Controller



Project - Assignment

Enter Name to Search

- Root
  - 454_C300_1
  - 454_C300_2
  - OPC_935
  - OurC300
    - CEEC300_2095
    - IOLINK_2093
    - IOLINK_2094
  - SIM_A2
  - SIM_C300_Azlina
  - Unassigned

Project Monitoring

Figure 13: Controller Composition

For the execution of Experion control strategies, Honeywell offers the use of a variety of controllers and process modules. Murdoch's pilot plant uses the Honeywell C300 Process Controller (C300) to communicate with Input/Output (I/O) modules, via Ethernet, for the execution of pre-programmed control strategies.

As identified in the technical review, Murdoch also acquired a secondary simulation license to facilitate the teaching of the Experion environment. For the purpose of this project a simulation SIMC300 controller 'OurC300' was created (as indicated in Figure 13).

The simulated C300 controller is comprised of four function blocks including:

- One platform block (the controller);
- One execution block (CEE300_2095), where all the control modules for execution are stored; and
- Two I/O link blocks used for the identification of the physical I/O of the system.

Configuration of the controller involved providing a unique tag name (OurC300) and defining a parent asset (SIM_ASSET_ITEM), before loading to the simulation environment. This controller was used for the duration of the project for the implementation of control strategies contained within the control modules.

4.2.2.2 Continuous and Sequential Control Module Development

With the simulated controller created and configured, it was possible to begin programming control modules.

Configuration was achieved by the creation of a blank control module and assigning the module to the created controller (Figure 14). Once assigned, a parent asset must be linked to the control module for the identification of function blocks and items to be programmed within the control module.

Upon completion of configuring the settings, the control module was then able to be programmed using the various function blocks contained within Experion's function block library.

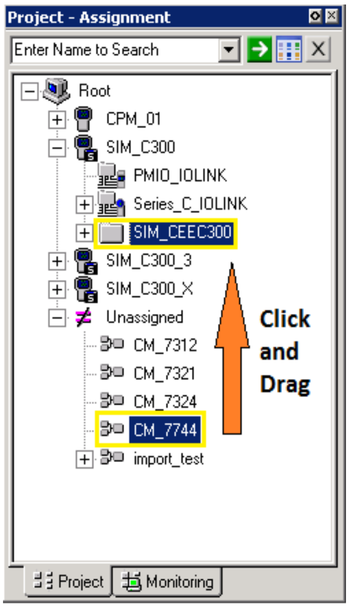


Figure 14: Control Module Assignment

Figure 15 displays the function block configuration for the Z_CM_MULT control module. In this example, three numerical function blocks and one multiplication block are used to program a mathematical expression into the control module. Whilst the control module is executing, values are able to be written to INPUT_1 and INPUT_2 tags, with the result of the multiplication written to the RESULT_MULT tag.

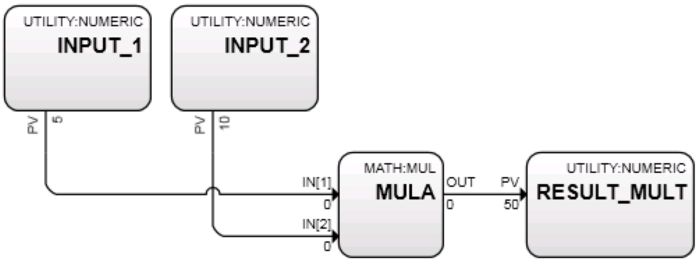


Figure 15: Mathematical Expression Control Module

4.2.3 Process Monitoring Techniques

With the configuration and implementation of control modules on the controller, steps were performed to actively monitor the developed programs. Using Experion's tag names, it was possible to dynamically monitor the values of the function blocks configured within the control modules. For example, the numerical inputs and outputs of the mathematical control module Z_CM_MULT, displayed in Figure 15, could be monitored and controlled either directly through control builder or through an external application such as:

- Microsoft Excel Data Exchange (MEDE); or
- A customised HMI displayed using Station.

4.2.3.1 Microsoft Excel Data Exchange

Identified as one of Honeywell's preferred forms of data exchange, MEDE allows for the capture of real-time data point values and historical information from the Experion server, for display within an Excel spreadsheet (Honeywell 2014e). Additionally, MEDE provides the ability to send values to data points on the Experion server. Figure 16 displays the resulting process variable response (PV), from a step in input, or manipulated variable (MV), to a first order system control module.

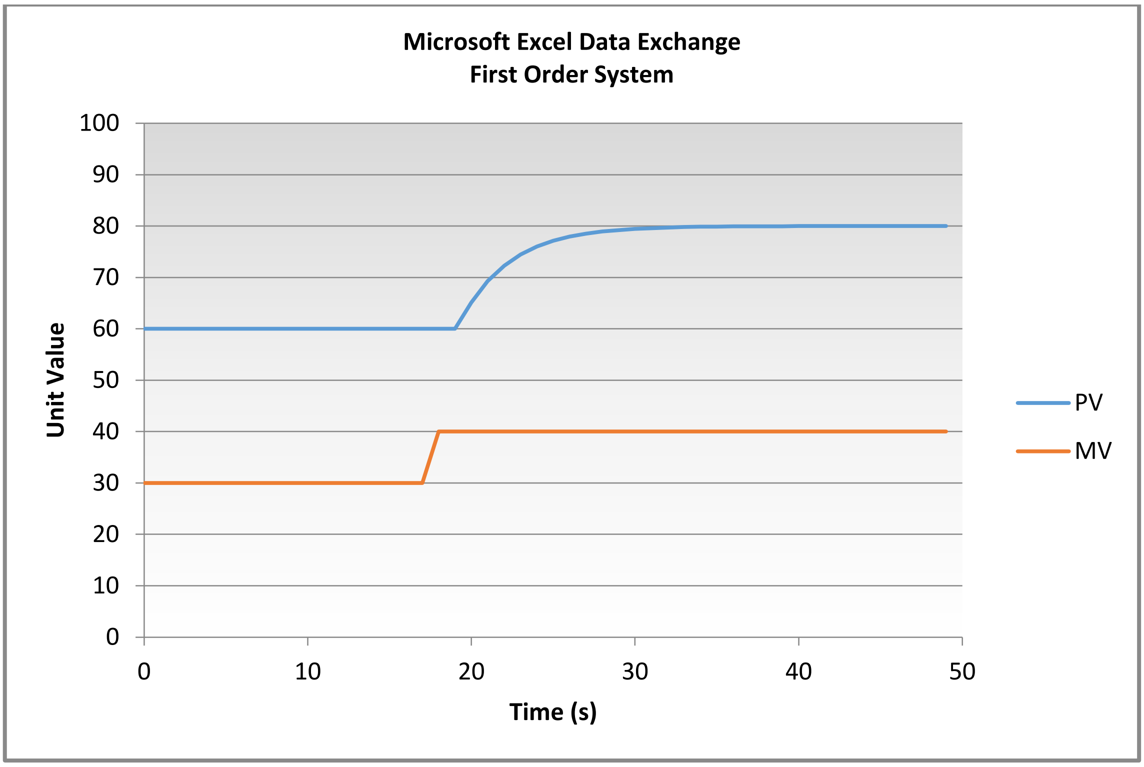


Figure 16: Microsoft Excel Data Exchange - First Order System Example

4.2.3.2 Human Machine Interface

The Experion package also contains the HMIWeb Display Builder, Honeywell's graphical drawing application for the development of HMI displays (Honeywell 2014d). For the effective monitoring and control of the control module programs developed in Experion, a range of HMI displays were created for viewing in Station.

1. Z_MULT – Mathematical Expression;
2. Z_LOGIC – Logic Expression;
3. Z_FOS – First Order System;
4. Z_FOSPID – First Order System with PID Control;
5. Z_CAS – Cascade Control;
6. Z_FOSPID-PL – Model-Based Predictive Control.

4.2.3.3 Station

In order to view any of the developed HMI displays, Station's connection properties were required to be configured. Settings were configured to recognise the location where the created HMI's were stored. Once configured, it was possible to open the displays developed for dynamically monitoring the appropriate process data.

4.3 Simulation and Testing

The production of learning materials for the setup and configuration of the Experion system was based upon a series of detailed example programs successfully created and tested during this project. These examples provide a technical review of relevant components of the program as well as three comprehensive procedures in each:

1. Development of the control module (program);
2. Monitoring and control of the control module program using MEDE;
3. Development of HMI for monitoring and control of control module using Station.

Six complete example programs have been developed with supporting documentation as specified above. These examples are:

1. First Order System;
2. Mathematical Multiplication Operation;
3. Boolean Logic Expression;
4. First Order System with Conventional Feedback PID Control;
5. Cascade Control;
6. Model-Based Predictive Control.

As the control strategies are examined at a later stage of this report, and due to the extensive nature of these tutorial examples, only the First Order System program and supporting documentation will be examined. This examination will include the development of the control module, data exchange Microsoft Excel spreadsheet and HMI display.

4.3.1 First Order System

Control modules are executed discretely at a user specified rate. This means that for a first order system to be implemented within the Experion environment, it must be discrete. For the purpose of this example program, a standard form continuous first order system (Equation 1) was discretised using the Euler method to develop an Euler approximation of a first order system (Equation 2). Equation 2 belowillustrates the equation entered into the control module to simulate the first order process. The breakdown of the Euler Approximation is included in Appendix C.

Equation 1: Continuous First Order System

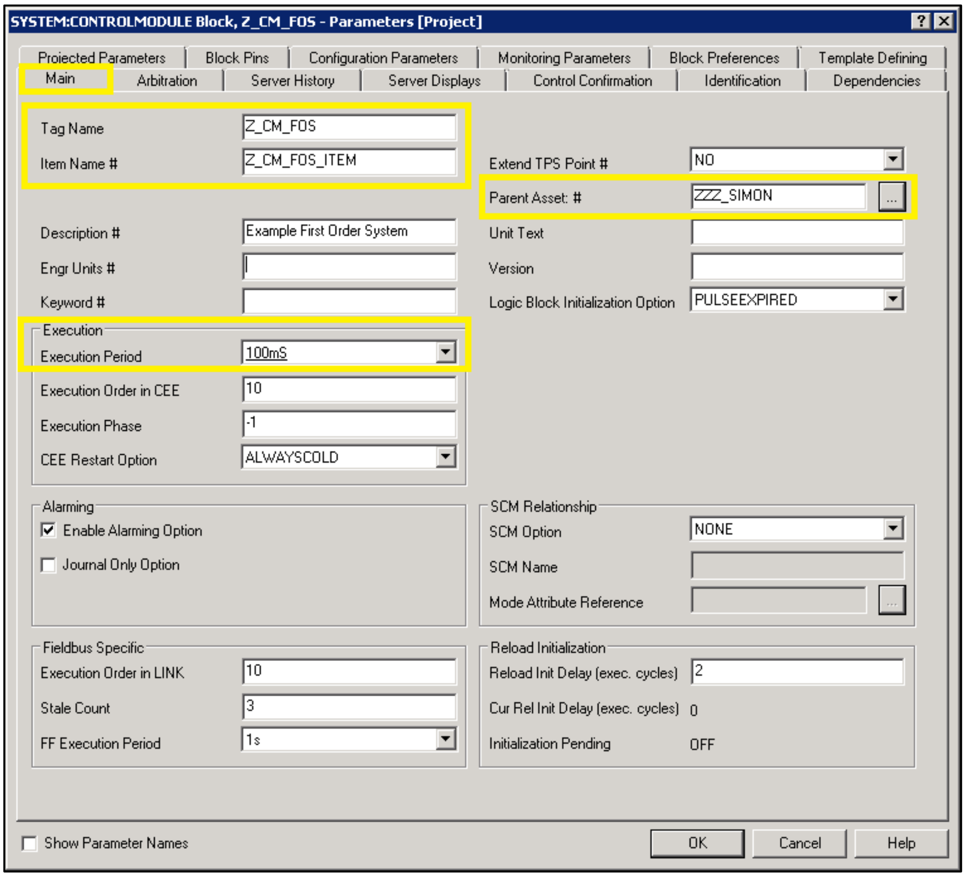
$$Transfer\ Function\colon \frac{K}{\tau s + 1}$$

Equation 2: Euler Approximation of First Order System

$$Euler\ Approximation\colon y(k) = \frac{K\ \Delta t\ u(k) + \tau\ y(k-1)}{\tau +\ \Delta t}$$

4.3.1.1 First Order System Example – Control Module Development

A control module (Z_CM_FOS) was created and configured, defining a parent asset and execution time, as displayed in Figure 17. It is important to note the execution period of the control module matches the sample time of the numeric value of ‘delta t’ (Δt) used in the Equation 2. For this example, an execution time of 100ms was selected, with Δt equal to 0.1 for the equation.



Tag Name Z_CM_FOS
Item Name # Z_CM_FOS_ITEM
Description # Example First Order System
Engr Units #
Keyword #

Execution
Execution Period 100mS
Execution Order in CEE 10
Execution Phase -1
CEE Restart Option ALWAYSCOLD

Extend TPS Point # NO
Parent Asset: # ZZZ_SIMON
Unit Text
Version
Logic Block Initialization Option PULSEEXPIRED

Alarming
☑ Enable Alarming Option
□ Journal Only Option

SCM Relationship
SCM Option NONE
SCM Name
Mode Attribute Reference

Fieldbus Specific
Execution Order in LINK 10
Stale Count 3
FF Execution Period 1s

Reload Initialization
Reload Init Delay (exec. cycles) 2
Cur Rel Init Delay (exec. cycles) 0
Initialization Pending OFF

□ Show Parameter Names

Figure 17: Control Module Configuration

Five numeric function blocks were selected from the function block library and inserted into the control module with their respective equation parameter identified in Table 1.

Table 1: Numeric Function Block Listing (First Order System Example)

Numeric Block	Equation Parameter	Description
1	K	Process gain
2	Δt	Sample time
3	u(k)	Input (manipulated variable)
4	τ	Time constant
5	y(k)	Output (process variable)

Following the numeric blocks, each of the mathematic operators were inserted into the control module to form the basis of the discrete equation (Equation 2), displayed in Figure 18.

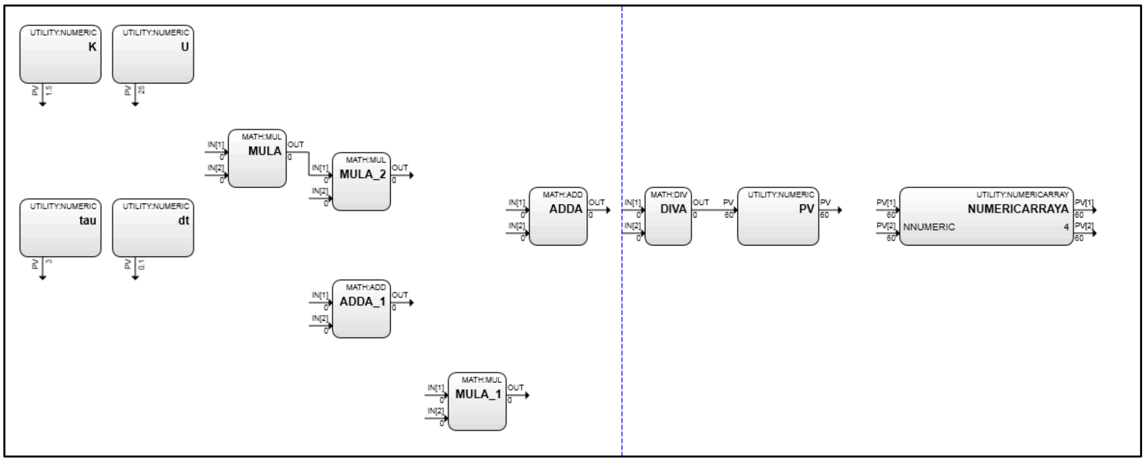


Figure 18: First Order System Function Block Arrangement

Once all blocks were entered and correctly configured, they were wired together to form the complete Euler approximation (Equation 2) as displayed in Figure 19.

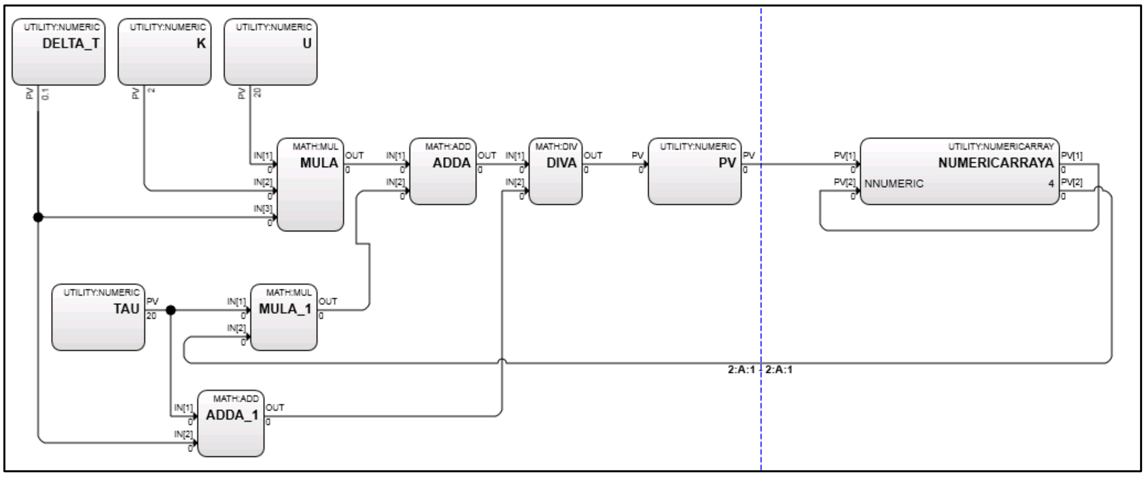


Figure 19: First Order System Control Module Complete Configuration

As Experion's programming nature has no defined flow of operation, the execution order of the function blocks contained within the control module must be defined. The correct definition of the execution order is critical to the correct operation of the module. Therefore, extensive examination and testing of the module was performed to ensure correct operation prior to final testing and validation. Execution order is specified by multiples of 10 as indicated in Table 2, the order for the first order system.

Table 2: First Order System Control Module Execution Order

Execution Order	Function Block	Function Block Type
10	DELTA_T	Numeric
20	K	Numeric
30	U	Numeric
40	TAU	Numeric
50	ADDA_1	Mathematical Operator
60	MULA	Mathematical Operator
70	MULA_1	Mathematical Operator
80	ADDA	Mathematical Operator
90	DIVA	Mathematical Operator
100	PV	Numeric Indicator
110	NUMERICARRAY	Numeric Array Indicator

The numeric function block values were then populated with the values specified in Table 3 before loading the control module to the controller for testing.

Table 3: First Order System Control Module Function Block Testing Parameters

Function Block	Numeric Value
K	2
TAU	20(s)
U	Step from 0 to 20
DELTA_T	0.1 (100ms)

Upon activating the control module, it was possible to dynamically view the numeric output of the PV block based on a change in the input (U) to the system. To illustrate dynamic monitoring of the system within Experion’s Control Builder, two separate snapshots of the system were taken at different time intervals, one at 35 seconds (Figure 20) and one at 134 seconds (Figure 21) after stepping the input.

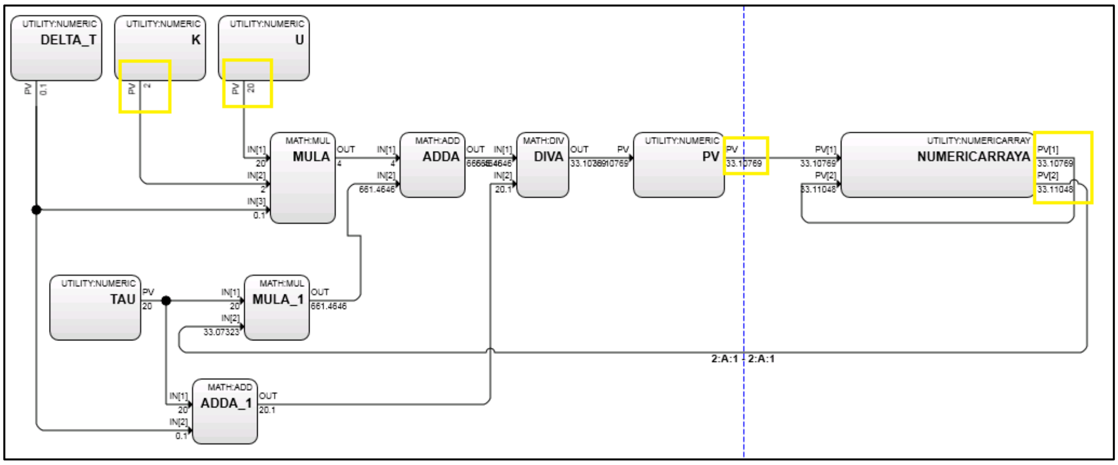


Figure 20: First Order System Control Module Dynamic Monitoring (snapshot approx. 35 after step)

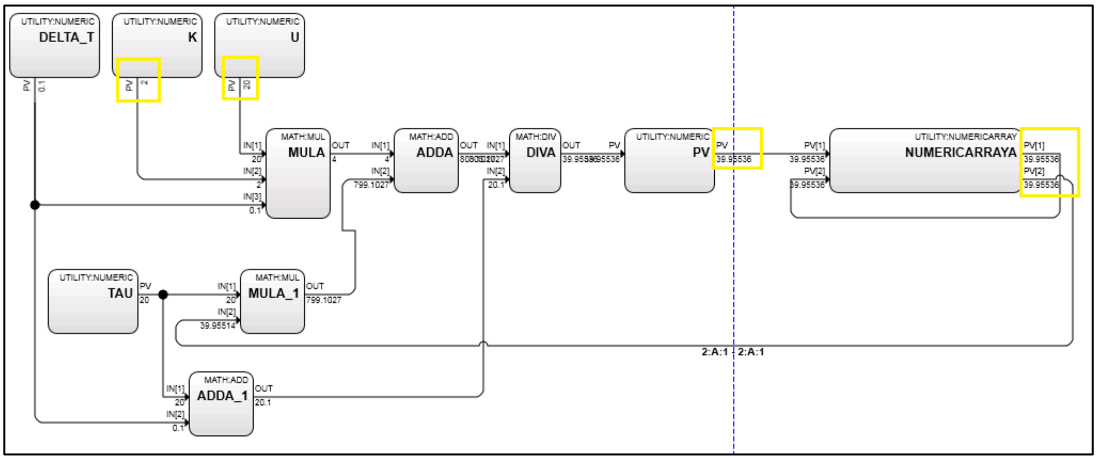


Figure 21: First Order System Control Module Dynamic Monitoring (snapshot approx. 134 after step)

Referring to Figure 20 and Figure 21, the numeric value of the output PV block has changed from value 33.10 (at 35 seconds) to 39.95 (at 134 seconds) indicating dynamic monitoring abilities.

Although Control Builder's block diagram displays system response dynamically, it was determined that for an enhanced learning experience, a graphical representation of the data would be required.

4.3.1.2 First Order System Example – Microsoft Excel Data Exchange Development

As indicated earlier, Honeywell specifies MEDE as a possible means for data exchange in the Experion documentation. Additionally, students undertaking a major in Instrumentation and

Control Engineering at Murdoch implement control strategies on the University's pilot plant using MEDE and would therefore be familiar with the application.

MEDE allows for both the sending and capturing of real-time point values and historical data from the Experion server, in an Excel spreadsheet, through the use of the following formulas:

1. GetHistValArray_offset;
2. GetPointValArray;
3. PutPointVal_Number.
4. GetHistVal_Date;
5. GetHistVal_Offset;
6. GetPointVal;

For the purpose of this example program, only three of the formulas (1, 2, 3) were used.

With each iteration of the MEDE add-in within the spreadsheet, a table is progressively populated by retrieving server values from the function blocks contained within the control module. This is achieved by the use of the function 'GetHistValArray_offset' and storing the data within appropriate columns as illustrated by Table 4.

Table 4: First Order System MEDE Historical Data

HISTORICAL VALUES FROM SERVER		
TIME	PV.PV (PV)	U.PV (MV)
1		
2		
3		
4		
…		
…		
50		

This data was displayed on a rolling chart to provide a graphical represenation of the process data (Figure 22). Altering the MEDE add-in's recalculation interval allows the chart to update dynamically displaying the process input (MV) and output (PV) at varying rates.

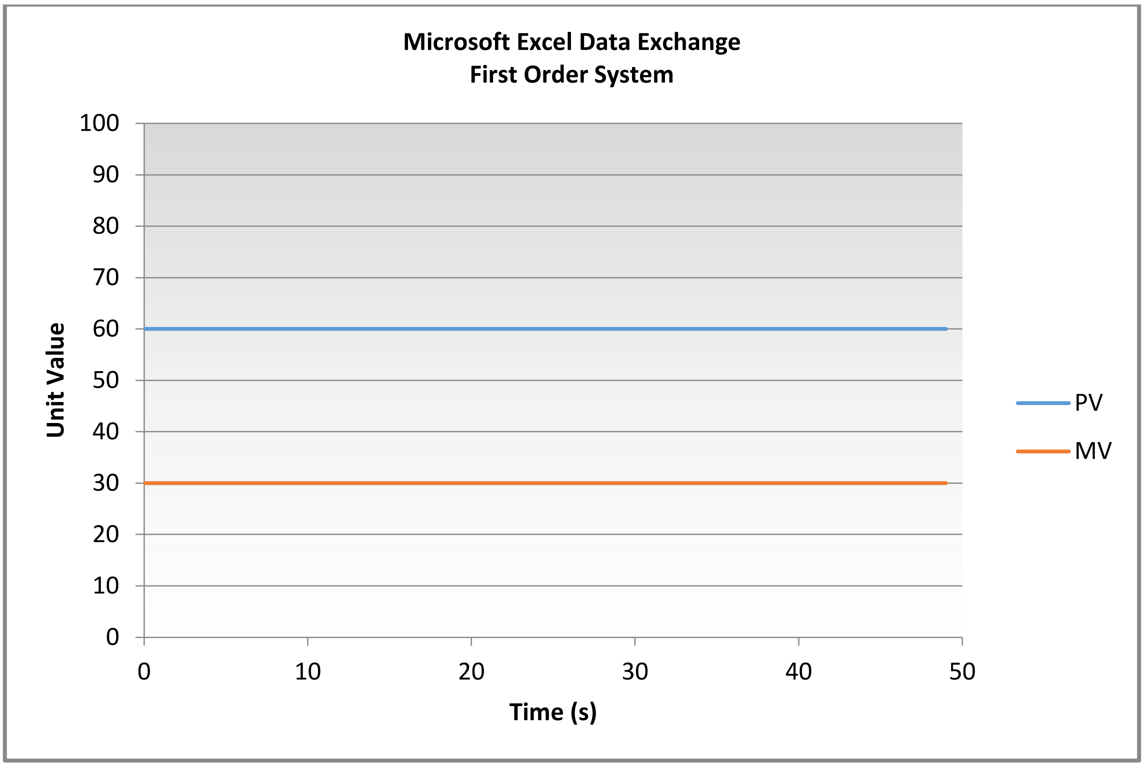


Figure 22: MEDE Chart of the First Order System Example

Once historical values were successfully being retrieved from the server, a table was created for values to be sent to the server, including the process parameters (gain and time constant) and the input to the system, as displayed in Table 5. Each of these parameters were sent to the server by making use of the formula 'PutPointVal_Number'.

Table 5: MEDE Process Parameters for the First Order System Example

PROCESS PARAMETERS

PROCESS GAIN	2
TAU	3
MV	40

With the implementation of each of the previously mentioned formulas it was possible to dynamically change the process parameters and input (MV) to the system, as well as monitor the corresponding output (PV) as displayed in Figure 23.

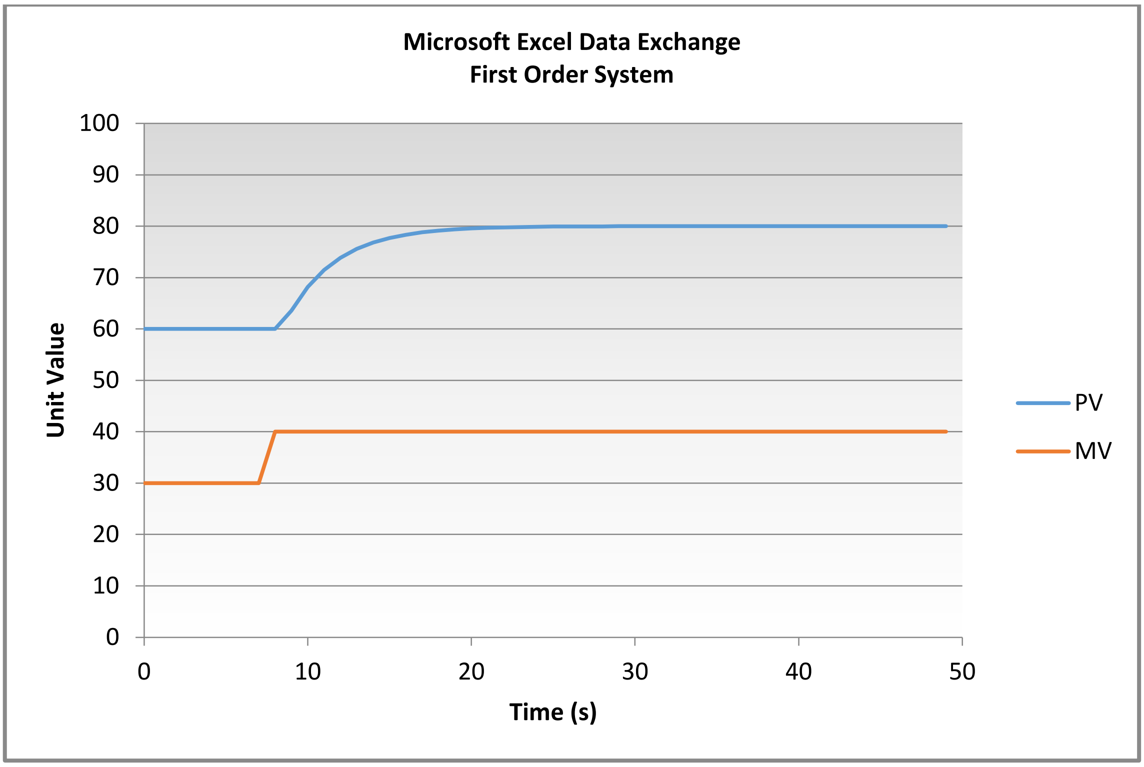


Figure 23: MEDE Chart of the First Order System Example (Dynamic Response)

A screenshot of the complete spreadsheet is available in the tutorial document located in Appendix D.

Although the Excel chart displays dynamic data on a rolling chart, it does not allow for displaying data over a period greater than 50 seconds. This is a potential problem for simulated systems with a time constant greater than 10 seconds, as the process would take over 50 seconds to reach a steady state after the introduction of a step to the input. In addition, the spreadsheet offers no graphical representation of the physical system being simulated as would typically be required by operators in an industrial setting.

To combat this, HMI display for the process was developed using Honeywell’s HMIWeb Display Builder.

4.3.1.3 First Order System Example – Human Machine Interface

As identified in the technical review, the Experion Platform includes the HMIWeb Display Builder for the development of customised HMI displays. To complete the example program, a HMI was developed to assist in the monitoring and control of the first order system. After the basic graphic had been developed (tank shape), a trend chart and appropriate numeric indicators and controls (alphanumeric objects) were added to enable monitoring and control of each of the process parameters, as displayed in Figure 24.

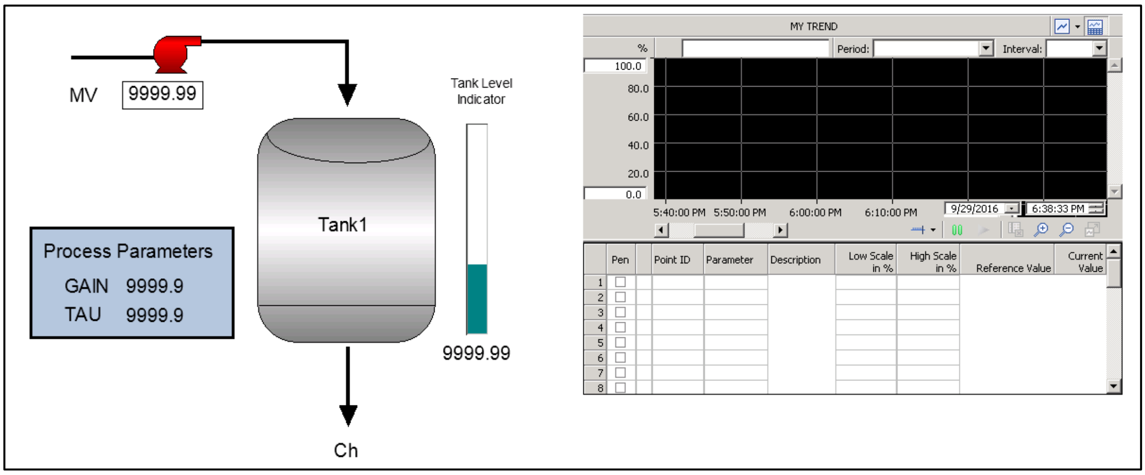


Figure 24: Complete First Order System Example HMI

By linking the alphanumeric objects (indicated by "9999.9" in Figure 24) to the tag names, these objects allow for both the reading and writing of data to and from the HMI display. With these included in the display, it was possible to dynamically change the input to the system and also make changes to the process parameters, such as the process gain and time constant.

4.3.1.4 First Order System Example – Simulation, Testing and Validation

Upon completing the first order system example, it was recognised that the control module program should be verified to ensure that the discretised Euler approximation model (Equation 2) was producing a first order system response. It was recognised that the response of the system would never be identical to that of continuous first order system, as the algorithm entered into the Experion control module is only an Euler approximation of an actual continuous transfer function system.

Two tests were conducted to validate the Experion system's control module configuration including:

- Least squares regression analysis;
- Process response comparison to a known first order system developed in Simulink.

Least Squares Regression Analysis

A linear least squares regression analysis provides a means of determining a best fit line to a set of data values (Miller n.p.). This means it was possible to determine the gain and time constant process parameters of the first order system based on the step response data values acquired from Experion.

The least squares regression was applied in Microsoft Excel and operates by making use of Microsoft Excel's Solver add-in, a tool which searches and provides for the best solution to a given optimisation problem (Microsoft 2016). To determine the process model using this technique, response data is collected from the introduction of a step in input to the system. The ideal first order response algorithm (Equation 3) is then entered for all values of time (t) to acquire an ideal process model, where A is the step input, K is process gain and τ is the process time constant.

Equation 3: First Order System Step Response Model

$$y(t) = AK(1 - e^{\frac{-t}{\tau}})$$

The error between the Experion response data and the ideal process model was then calculated for all iterations of time. Following this, the sum of all squared error was calculated within a single cell. By using Excel's Solver, it was possible to minimise the value of the squared error by making alterations to the values of K and τ to match the model to the actual data provided by Experion. The final process parameters selected by the Solver indicate the system generating the response produced by Experion. A screenshot of the linear least squares regression analysis performed is presented in Appendix E.

The least squares regression analysis was conducted by first entering the process parameters for the transfer function presented in Equation 4, into the model (Equation 3) for each iteration of time from 0 to 35 seconds. A step input of numeric value from 0 to 20 at time 2 was then introduced, before applying the Solver, to reduce the error between the model and Experion data. Upon applying the Solver, the process parameters indicated within Equation 5 were acquired. The respective responses of each of these equations are presented in Figure 25.

Equation 4: First Order System for Least Squares Regression

$$Experion\ Transfer\ Function: \frac{1.5}{3s+1}$$

Equation 5: First Order System Least Square Regression Model

$$Model\ Transfer\ Function: \frac{1.501066}{3.354871s+1}$$

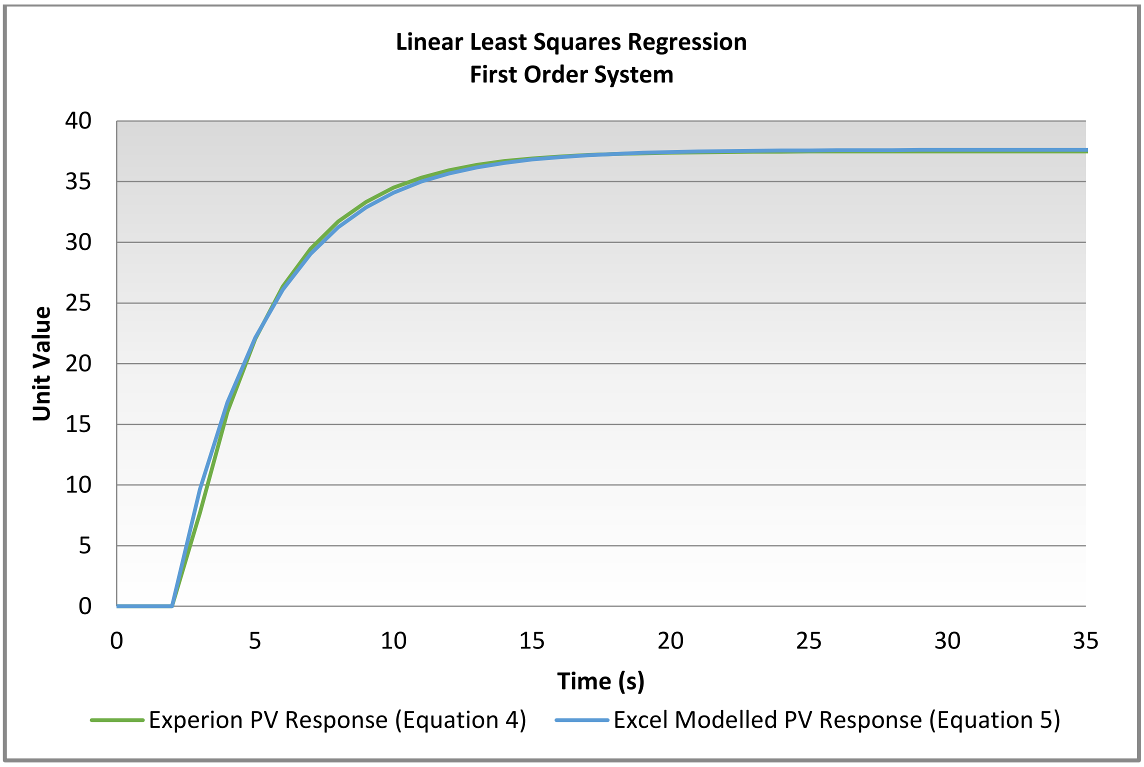


Figure 25: Least Squares Regression First Order Response

It was identified that the discrepancies in parameters and process response could be a result of the Euler approximation used. As the process is not strictly a continuous process, but rather an approximation based on a given sample time, a completely identical process response could not be expected. Prior to examining the Euler approximation effects, a secondary comparison test was conducted using Simulink to ensure the discrepancies were not a result of the regression test conducted.

Simulink Data Comparison

Following the least squares regression test, it was decided to compare the acquired Experion response data to a known first order system response generated with the Simulink application. Figure 26 illustrates the Simulink model developed for testing.

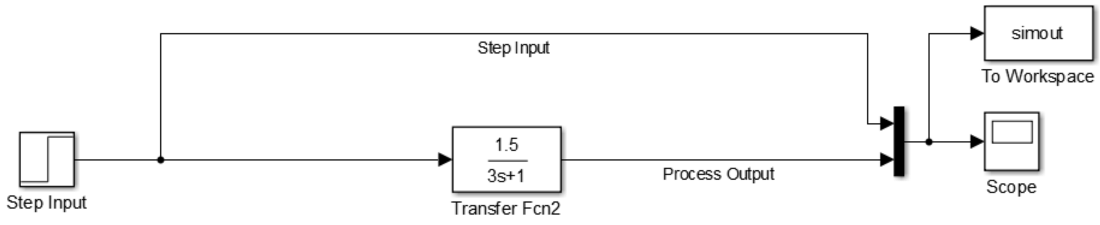


Figure 26: Simulink Model - First Order System Testing

By running a simulation with identical process parameters, and introducing a step input identical to that of the Experion test, it was possible to capture the numerical response values for comparison to the response acquired from Experion. The results of this test can be seen graphically in Figure 27.

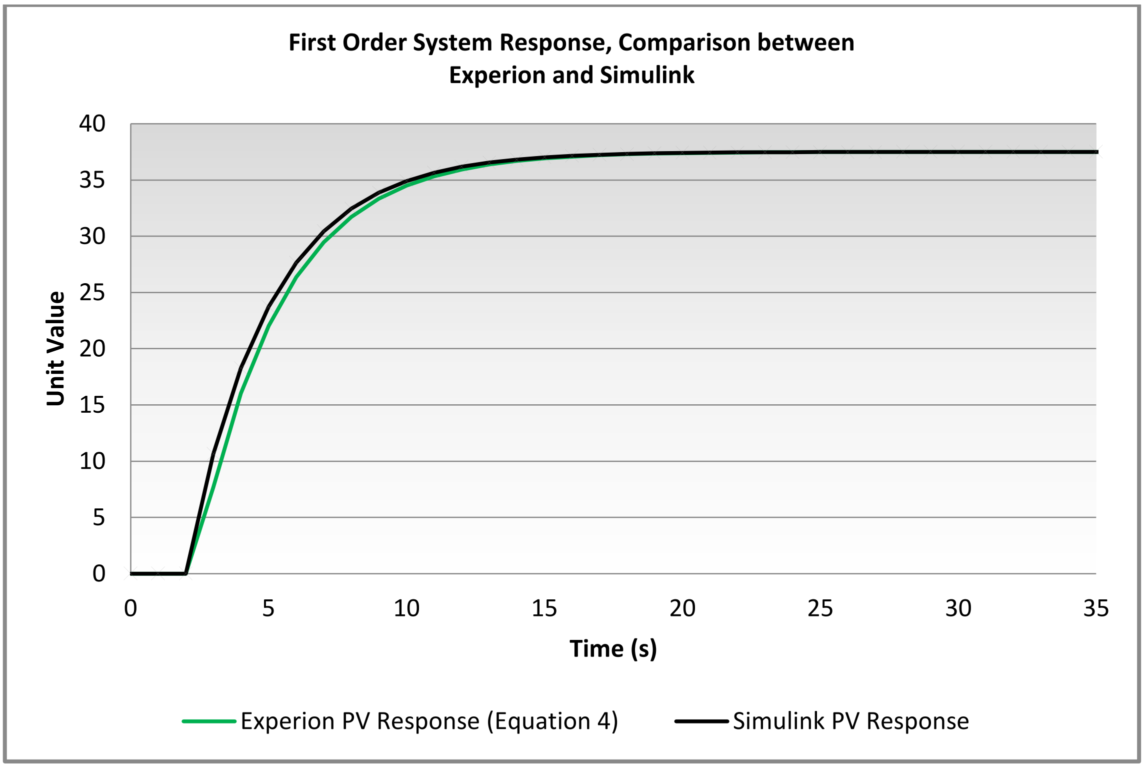


Figure 27: First Order System Response Comparison

As shown in Figure 27, trajectories are near identical with minor numerical difference between Experion and Simulink process response. As indicated earlier, these differences were thought to be a result of the Euler approximation of the first order system. To test this theory, the sample time of the Euler approximation entered into Experion was reduced from 100ms to

50ms (the fastest control module execution time available within Experion). Identical step tests were conducted with the results presented in Figure 28 below.

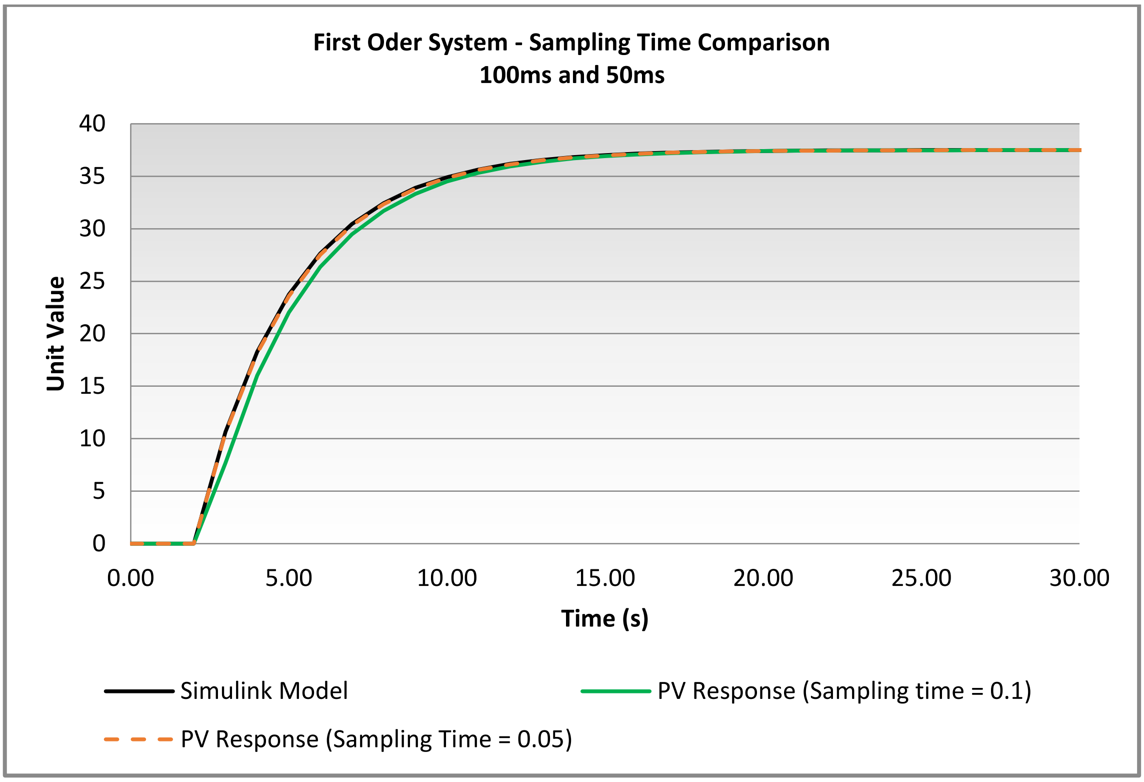


Figure 28: Sample Time Comparison Test (100ms and 50ms)

From the data presented in Figure 28, the conclusion was drawn that the discrepancies were most likely a result of the Euler approximation. It can be seen that by reducing the sample time, the response of the Experion system is almost identical to the ideal continuous model tested with Simulink.

Although reducing the sample time to 50ms produced improved results, it was decided that reducing the control module execution time, and subsequent sample time to 50ms, resulted in unnecessary additional controller computations (by as factor of 2) on the Experion server. As the responses of the system were near identical with the sample time set to 100ms it was decided to continue using the sample time value of 100ms for all future tests and simulations performed.

With the successful testing of the first order system, several control strategies available within Experion were then implemented and tested on the developed system.

4.4 Experion Control Strategy Implementation

Following the successful implementation and testing of a first order system, the implementation and testing of various control strategies within the Experion environment was performed. Experion offers a range of controllers contained within the function block library to facilitate standard feedback control, advanced control configurations model-based predictive control arrangements.

Prior to commencement of this project, it was recognised that previous students (refer to Appendix P) had configured control strategies; however, no tests had been conducted to validate the control strategies and the respective results. Furthermore, the examples developed no longer operated correctly since the Experion software upgrade. The decision was made to revisit the implementation of a range of control strategies to test and validate the operation of each, before producing appropriate learning material.

When executing the newly developed control strategies, several tests were performed to ensure the control strategy was operating as expected, validating results against known process control simulations performed in Simulink. As this project is primarily concerned with the correct implementation of the control strategy within the Experion environment, rather than achieving a desired controller outcome, once the tests validated the configuration of the control strategy, no further tests were conducted.

Three controller strategies were implemented during this phase of the project:

- Conventional feedback Proportional Integral Derivative (PID) control;
- Cascade control;
- Honeywell’s patented model-based predictive ‘Profit Loop’ Control (Honeywell 2016d).

4.4.1 Conventional Feedback Proportional Integral Derivative Control

The implementation of a feedback control strategy within the Experion environment was achieved through the use of Experion's PID block (Figure 29). Configuration was achieved by wiring the current process variable (PV) and the desired numeric setpoint (SP) into the two PID input terminals. The output of the controller, referred to as the operating point (OP), is then wired to the input to the process.

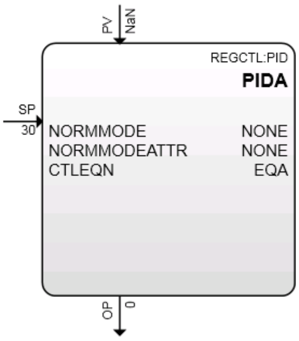


Figure 29: Experion PID Block

For the purpose of testing the configuration, an additional first order system control module was created (Z_CM_FOSPID). The PID block was then inserted into this control module with appropriate wiring connections specified above and displayed in Figure 30.

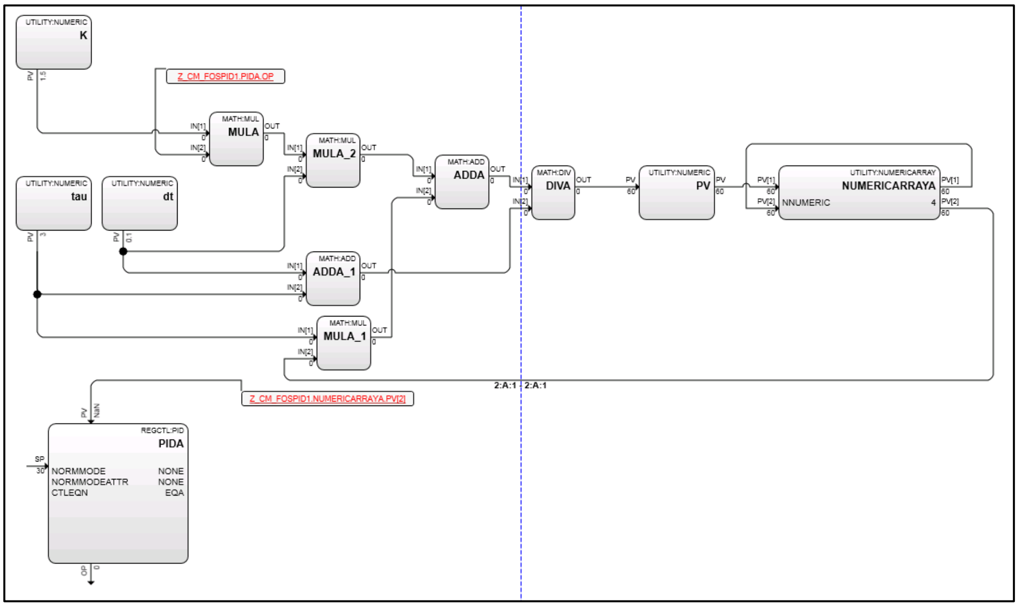


Figure 30: First Order System with PID Control Module

The PID block is configured by entering the module properties and setting the controller mode to AUTO, selecting the PID equation to be used by the controller, and entering the controller parameters. The PID block incorporates a range of different PID algorithms; however, as it is similar to a PID algorithm previously studied within the instrumentation and control major at

Murdoch (Equation 6), EQA (Equation 7) was selected for use for the duration of the project. In Equation 7, L^{-1} is inverse Laplace, $(PVP_s - SPP_s)$ is the calculation of error (ε), K is the controller gain and T1 (τ_i) and T2 (τ_D) are the controller integral and derivative terms respectively.

Equation 6: Conventional PID Algorithm

$$c(s) = K_c\left(1 + \frac{1}{\tau_i s} + \tau_D s\right)\varepsilon(s)$$

Equation 7: Experion PID Algorithm (EQA)

$$CV = K * L^{-1}[\left(1 + \frac{1}{T1_s} + \frac{T2_s}{1 + aT2_s}\right) * (PVP_s - SPP_s)\,]$$

For testing the configuration and operation of the controller within the control module, arbitrary controller values were entered into the PID block, forming a standard PI controller with the controller parameter values indicated in Table 6.

Table 6: Arbitrary PID Controller Parameters

Controller Parameter	Numeric Value
Controller Gain (K)	1.6
Integral Time (T1)	0.0333min
Derivative Time (T2)	0

Similar to the first order system example, a Microsoft Excel spreadsheet and HMI display were developed to provide the ability to monitor and control the process, as displayed in Figure 31 and Figure 32.

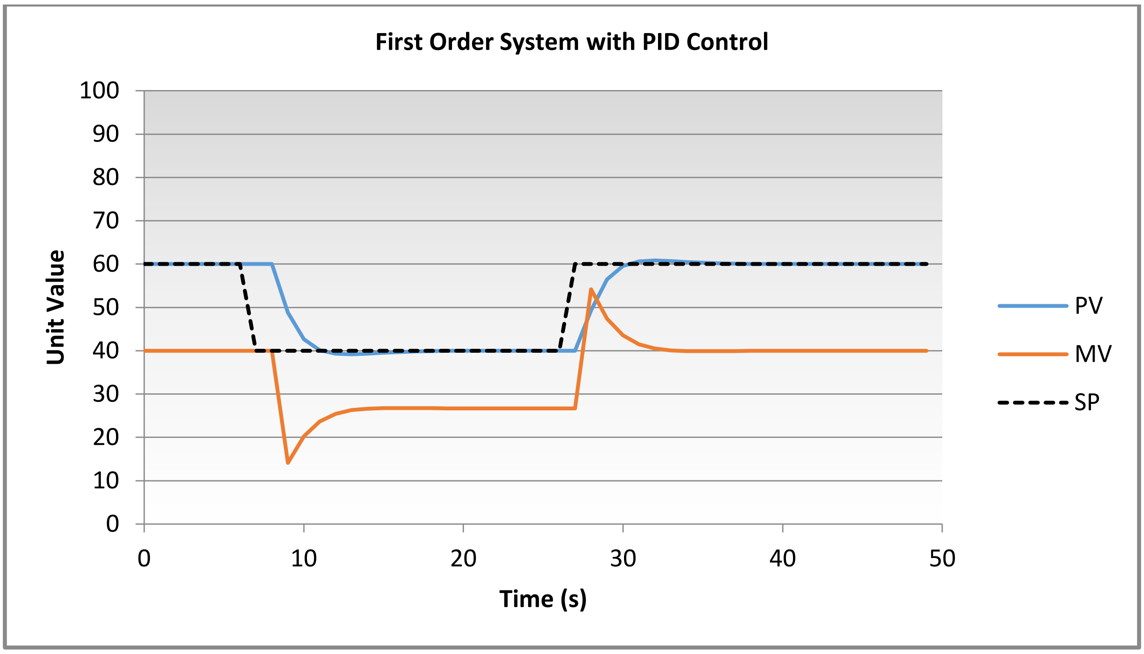
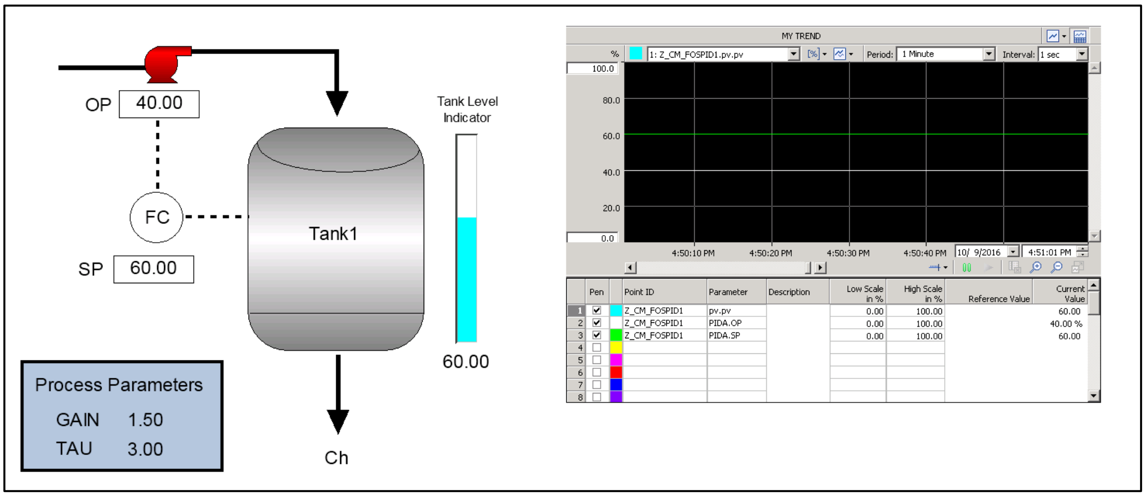


Figure 31: First Order System with PID Example MEDE Chart (Dynamic Response)



1: Z_CM_FOSPID1.pv.pv

Period: 1 Minute

Interval: 1 sec

10/ 9/2016

4:51:01 PM

	Pen		Point ID	Parameter	Description	Low Scale in %	High Scale in %	Reference Value	Current Value
1	☑	🟦	Z_CM_FOSPID1	pv.pv		0.00	100.00		60.00
2	☑	⬜	Z_CM_FOSPID1	PIDA.OP		0.00	100.00		40.00 %
3	☑	🟩	Z_CM_FOSPID1	PIDA.SP		0.00	100.00		60.00
4	□	🟨							
5	□	🟪							
6	□	🟥							
7	□	🟦							
8	□	🟪							

Figure 32: First Order System with PID Example HMI

4.4.1.1 Conventional PID Control Testing and Validation

To ensure the configuration of the feedback controller was correct, Experion response data was compared with the data attained from a Simulink model simulation with identical process and controller parameters (Figure 33 and Equation 8). It is important to recall that this project was not concerned with the tuning of controllers to achieve a desired outcome, but focussed on the testing and implementation of control strategies to confirm correct configuration and operation.

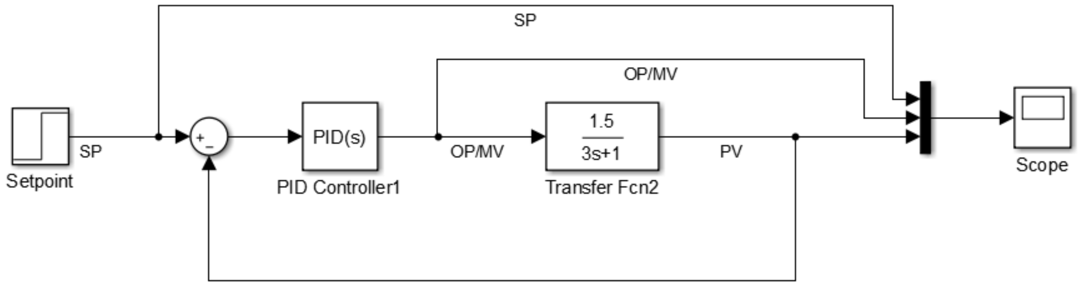


Figure 33: Simulink Model - First Order System with PID Control

Equation 8: Experion System 1 - PID Controller Testing

$$G_1(s) = \frac{1.5}{3s+1}$$

When configuring the PID controller within Simulink, alterations to the integral time were required to ensure consistency between controller outputs. Integral time for the Experion PID block is configured in minutes, while the integral time in Simulink is configured in seconds. Additionally, as displayed in Simulink's controller algorithm (Equation 9), the integral term within the algorithm (τ_i) must be reciprocated to reflect that of the Experion PID algorithm specified earlier (Equation 7).

Equation 9: Simulink PI Equation

$$C(s) = Kc\left(1 + \frac{\tau_i}{s}\right)$$

The final controller parameter values entered for comparison and testing are presented in Table 7.

Table 7: First Order System PID Testing – Controller Tuning Parameters

	Experion PID	Simulink PID
Controller Gain	1.6	1.6
Integral Time	0.0333333	0.5
Derivative Time	0	0

Tests similar to that of the first order system example were conducted, with the acquisition of Experion and Simulink process response data. Testing was conducted by introducing identical setpoint changes of value 40 to 60 at time 5 on both the Experion simulation and the Simulink model, with the comparison data presented in Figure 34.

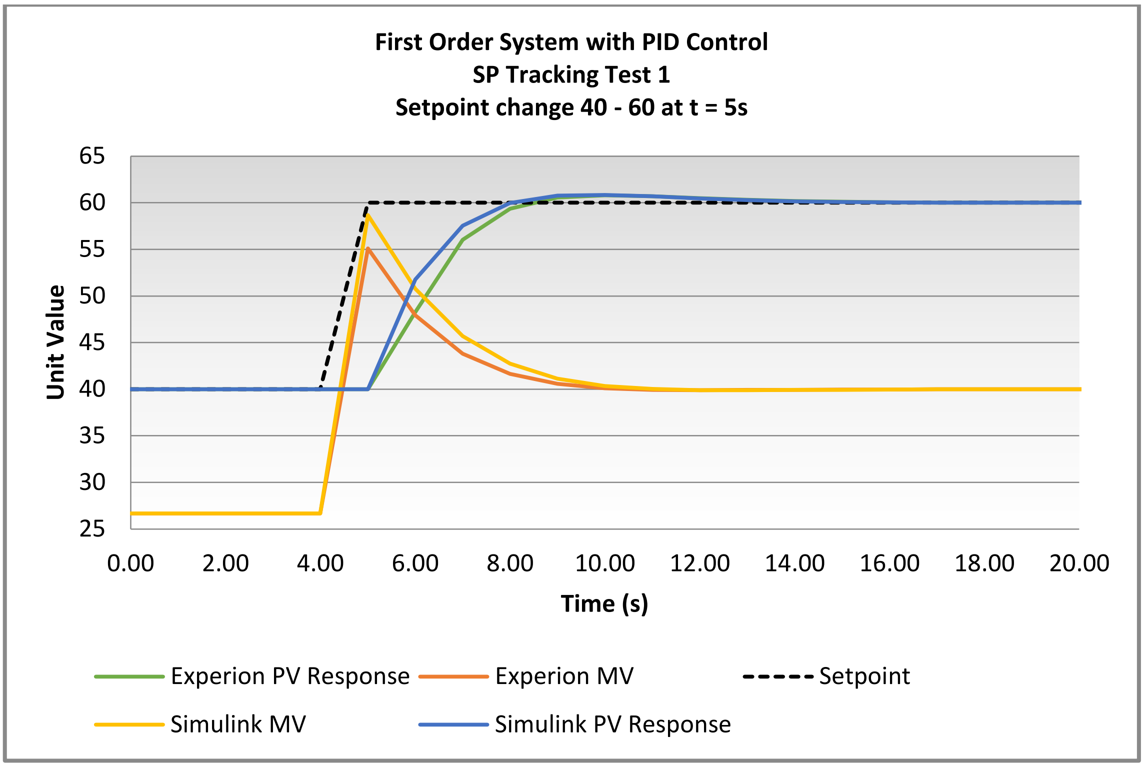


Figure 34: First Order System with PID Response Comparison

It can be seen in Figure 34 that both the Experion process response (PV) and controller action (MV) follow a near identical trajectory to that of the Simulink simulation; however, differences in both controller action and process response are present. As with the first order model example, these discrepancies can be explained by the Euler approximation system not being a true representation of a continuous first order system

As this phase was concerned with implementing correctly operating controller strategies, the test was successful with the tracking of setpoint, following a near identical trajectory to an ideal controller implemented on a simulated continuous system.

Additional tests were conducted on the same system (System 1 - Equation 8) with varying setpoint changes to ensure consistent results. Additionally, a second system (System 2 - Equation 10) with different process parameters was tested to ensure consistent results among processes. All tests performed produced results consistent to those presented in Figure 34 and are available in Appendix F.

Equation 10: System 2 - PID Controller Testing

$$G_2(s) = \frac{2}{10s + 1}$$

4.4.2 Cascaded Feedback Proportional Integral Derivative Control

Often implemented with the intent of rejecting significant disturbances on the input of a system, cascade control is configured within the Experion environment by replicating the following block diagram (Figure 35).

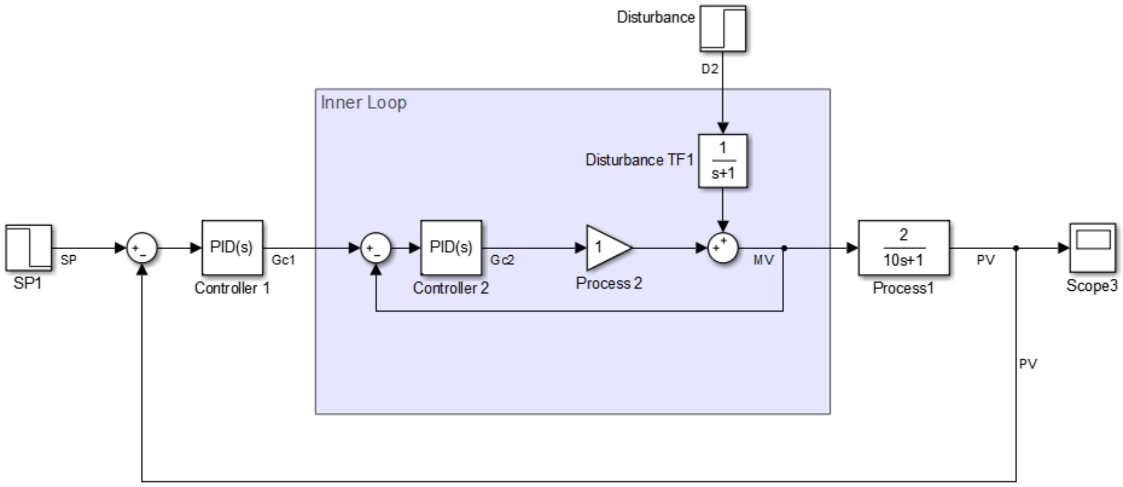


Figure 35: Simulink Model - Cascade Control Strategy

For the implementation of the cascade control strategy, the output of controller 1 becomes the setpoint of the inner loop controller 2. The role of controller 2 is to ensure that the main process (process 1) receives the input as originally intended by controller 1, should there be a disturbance on the process input (Ogunnaike and Harmon Ray 1994).

To replicate the above block diagram (Figure 35) two new control modules were created, one control module (Z_CM_CAS1) containing the necessary function blocks to make up the primary first order process (Equation 11) and controller strategy (Figure 36), and the other containing the blocks for the disturbance system (Equation 12) as displayed in Figure 37.

Equation 11: Cascade Primary Process (Process1)

$$G_1(s) = \frac{2}{10s + 1}$$

Equation 12: Cascade Disturbance System (Disturbance TF1)

$$D_1(s) = \frac{1}{s + 1}$$

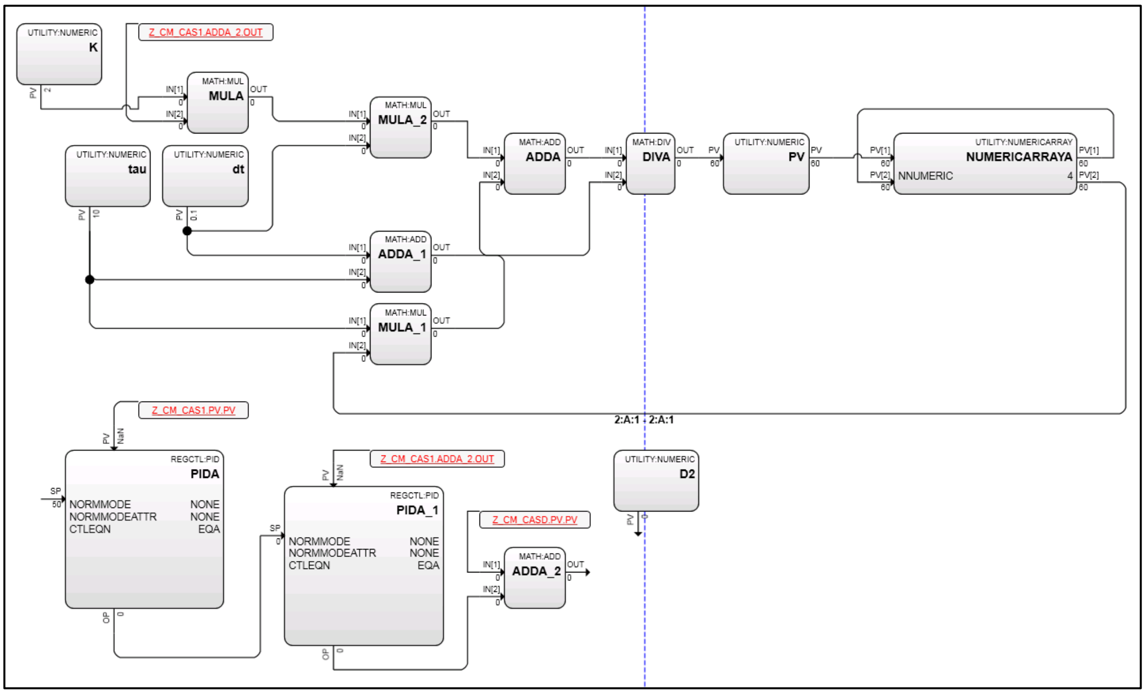


Figure 36: Experion Control Module - Cascade Control Strategy

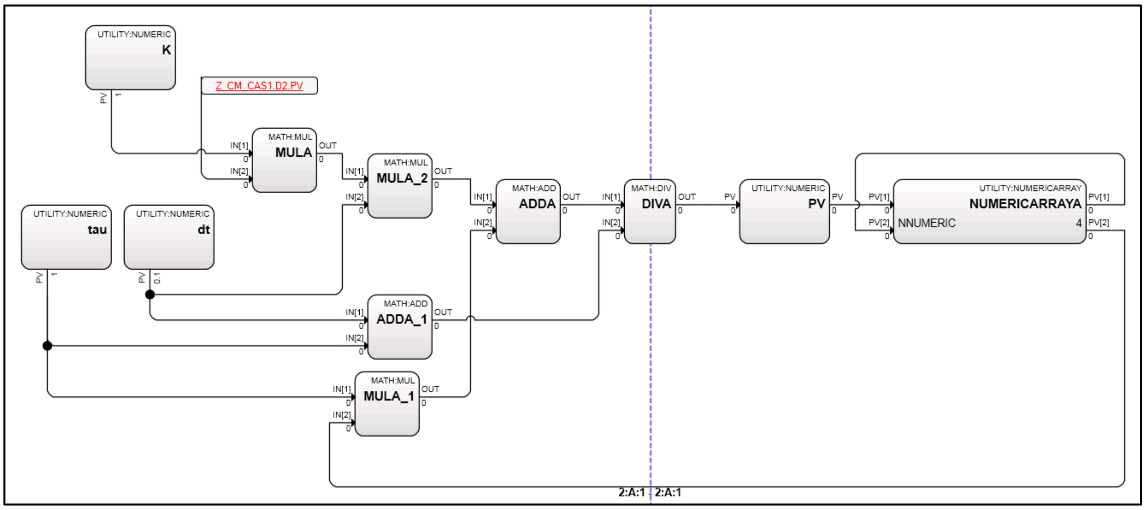
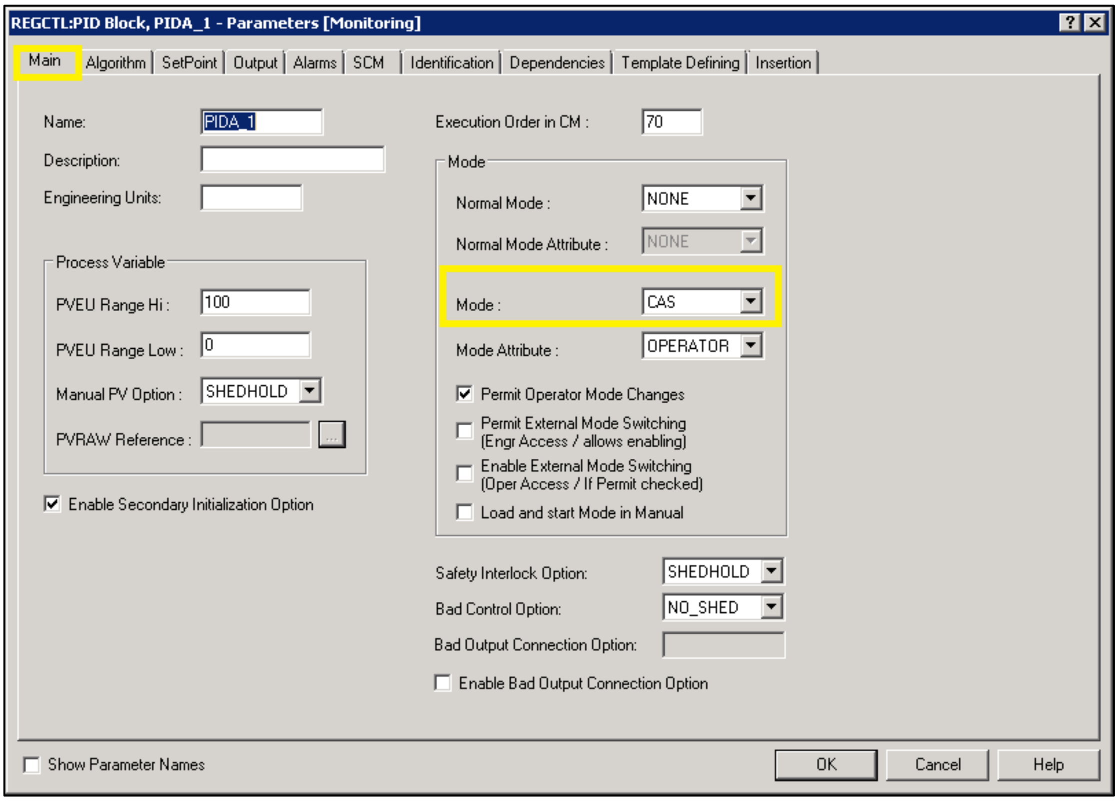


Figure 37: Experion Control Module - Cascade Disturbance

To implement the secondary controller (PIDA_1) for the inner control loop of the cascade configuration (Figure 36), the properties of the block must be adjusted to accept a setpoint fed from the output of the primary controller (PIDA). This property is easily changed within the main tab of the block parameters as indicated in Figure 38.



REGCTL:PID Block, PIDA_1 - Parameters [Monitoring]

Name: PIDA_1

Description:

Engineering Units:

Process Variable

PVEU Range Hi : 100

PVEU Range Low : 0

Manual PV Option : SHEDHOLD

PVRAW Reference :

☑ Enable Secondary Initialization Option

Execution Order in CM : 70

Mode

Normal Mode : NONE

Normal Mode Attribute : NONE

Mode : CAS

Mode Attribute : OPERATOR

☑ Permit Operator Mode Changes

□ Permit External Mode Switching (Engr Access / allows enabling)

□ Enable External Mode Switching (Oper Access / If Permit checked)

□ Load and start Mode in Manual

Safety Interlock Option: SHEDHOLD

Bad Control Option: NO_SHED

Bad Output Connection Option:

□ Enable Bad Output Connection Option

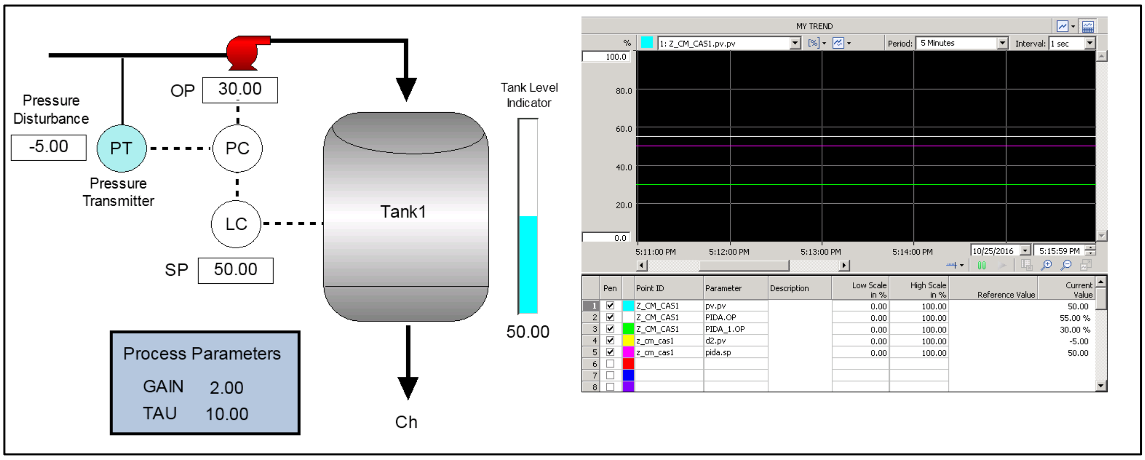
□ Show Parameter Names

Figure 38: Cascade Controller Properties

Upon the correct configuration of the controllers, the disturbance system control module was configured. As indicated by the Simulink block diagram (Figure 35), the disturbance value is added to the input of the primary process. For the simulation of a realistic process, the disturbance value (D2) from the primary control module (Z_CM_CAS1) was passed through the first order system (Z_CM_CASD) to provide a dynamic nature to the disturbance.

With this configuration, numerical changes to the function block D2 affect the input to the disturbance system (Z_CM_CASD), whose effects are reduced by the inner loop controller PIDA_1 before causing significant disturbances to the input of the primary process.

For effective monitoring and control of the system, and to assist future students to understand the control strategy configuration, a secondary HMI was developed to provide a graphical display of a system under cascade control. This HMI provides the ability to make alterations to process parameters, change the desired setpoint as well as introduce a disturbance to the system. See Figure 39.



	Pen		Point ID	Parameter	Description	Low Scale in %	High Scale in %	Reference Value	Current Value
1	☑	🟦	Z_CM_CAS1	pv.pv		0.00	100.00		50.00
2	☑	⬜	Z_CM_CAS1	PIDA.OP		0.00	100.00		55.00 %
3	☑	🟩	Z_CM_CAS1	PIDA_1.OP		0.00	100.00		30.00 %
4	☑	🟨	z_cm_cas1	d2.pv		0.00	100.00		-5.00
5	☑	🟪	z_cm_cas1	pida.sp		0.00	100.00		50.00
6	□	🟥							
7	□	🟦							
8	□	🟪							

Figure 39: Cascade Control Strategy HMI

4.4.2.1 Cascaded PID Control Testing and Validation

As with the implementation of the conventional feedback PID control strategy, it was necessary to test and validate the configuration of the cascade control strategy. Testing involved introducing a disturbance to the Simulink model identified in Figure 35 earlier, and comparing the system response to identical tests performed on the Experion system, to ensure the controllers were acting as intended.

For testing, tuning parameters were arbitrarily selected as displayed in Table 8. For simplicity in design, the primary controller was designed as a Proportional Integral (PI) controller only, while the inner loop controller was tuned as a gain only controller (P).

Table 8: Cascade Controller Testing - Controller Tuning Parameters

	Controller Gain	**Controller Integral Time**	**Controller Derivative Time**
Experion Primary Controller	2.743	0.066667	0
Simulink Primary Controller	2.743	0.25	0
Experion Inner Loop Controller	1	0	0
Simulink Inner Loop Controller	1	0	0

Similar to the conventional feedback PID control strategy, it was expected that the output of both controllers, and the process variables of the Experion test, would follow a similar trajectory to those in the Simulink model, thus confirming correct configuration.

The initial test involved introducing a step up in disturbance value (+8) at time 5 to analyse the trajectory of each controller's output (MV) and process variable (PV), the results of which are displayed in Figure 40.

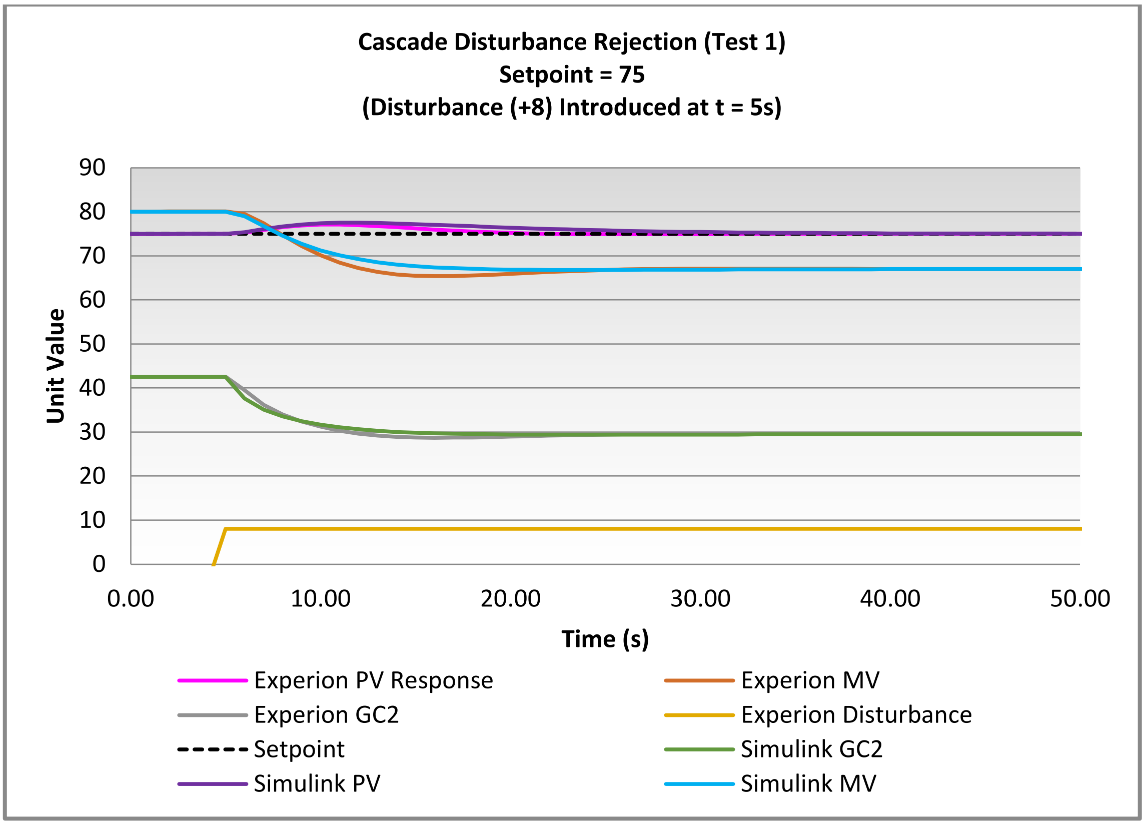


Figure 40: Cascade Control Response Comparison

By focussing solely on the process variable plots displayed in Figure 41, the error margin between Experion and Simulink is approximately 1%. Similarly, when focussing on the actions of controllers 1 and 2 (Figure 42), the differences were also minor.

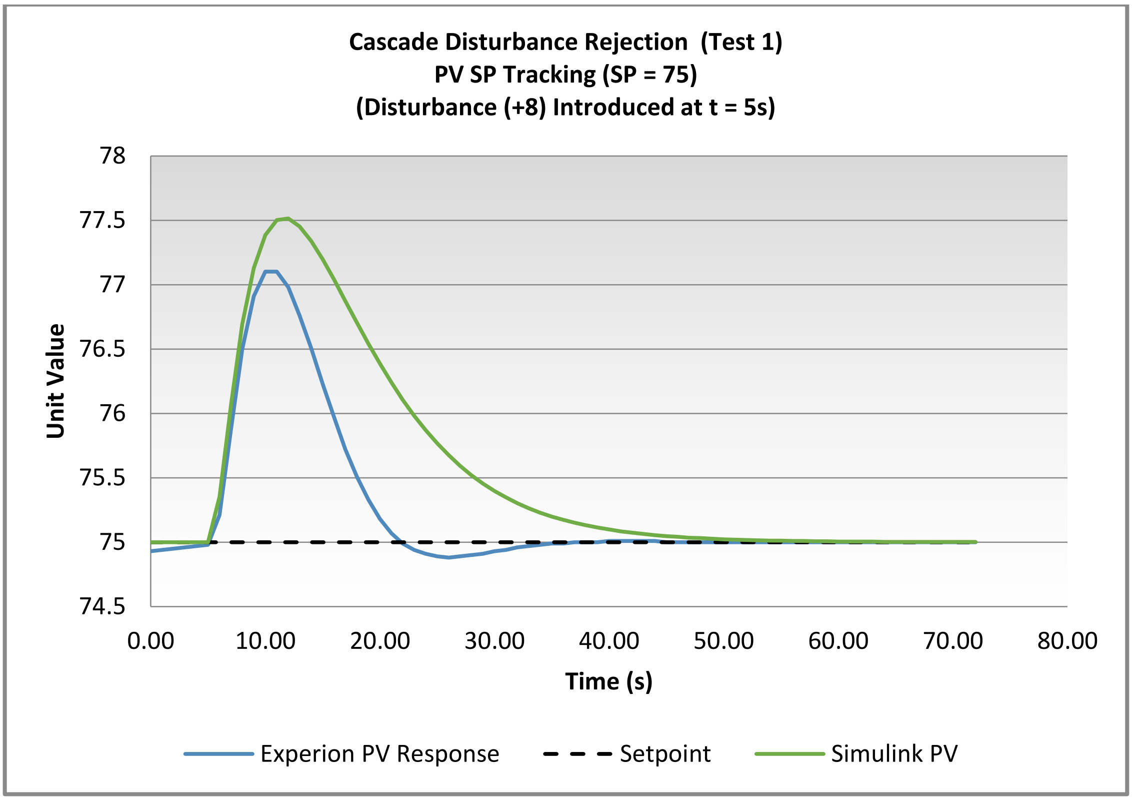


Figure 41: Cascade Control Process Variable Response Comparison

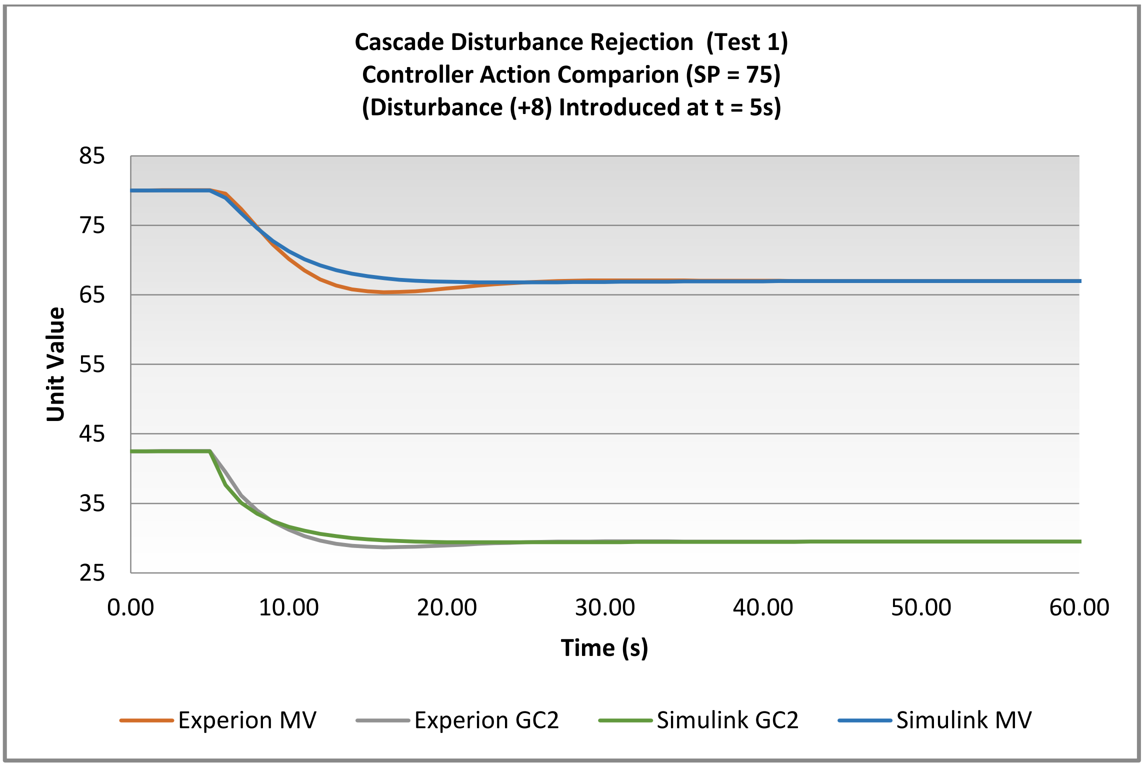


Figure 42: Cascade Control Controller Action Comparison

Based on the data presented in Figure 40, Figure 41 and Figure 42, it can be seen that each of the trajectories of the Experion system follow a near identical path to that of the Simulink model, indicating correct configuration and operation.

A total of four disturbance rejection tests were conducted on the cascade system to ensure that the results attained were consistent with those presented above. Each of these tests produced similar response and can be viewed in Appendix G.

In the early stages of the project, upon the successful implementation of a cascade control arrangement, it was intended to implement and validate the operation of both feedforward and augmented feedforward feedback control arrangements. However, upon executing the feedback PID and cascade control arrangements, it was identified that the implementation of a feedforward controller could be achieved by simply replacing the conventional PID function block. This block would be replaced with the feedforward PID block (PID-FF), with disturbance on the system wired directly into a second input to the PIDFF block. It was therefore decided to forego the implementation of these control strategies and spend additional time on the research and implementation of Experion's model-based predictive controller.

4.4.3 Model-Based Predictive Control

Model-based predictive control describes a class of control schemes that make use of a known process model to explicitly predict future system response, and make appropriate control actions to drive the predicted output to a desired target value (Seborg, Edgar and Mellichamp 2004).

Experion's model-based predictive controller block PID-PL, combines the functionality of a conventional PID block with a model-based predictive controller and optimiser algorithm, known as the Profit Loop PKS (Honeywell 2013). The predictive functionality of the controller provides an ideal means for the control of complex systems, including those with significant delays, inverse response, and processes with reduced sampling intervals (Honeywell 2013).

The PID-PL is implemented in the same manner as the PID block requiring two inputs; setpoint and current process variable (PV) value as displayed in Figure 43.

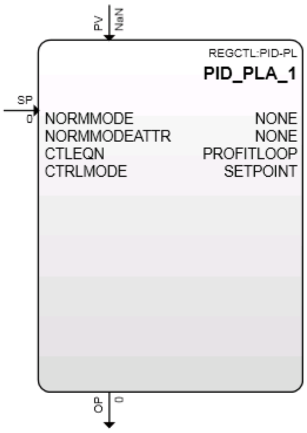
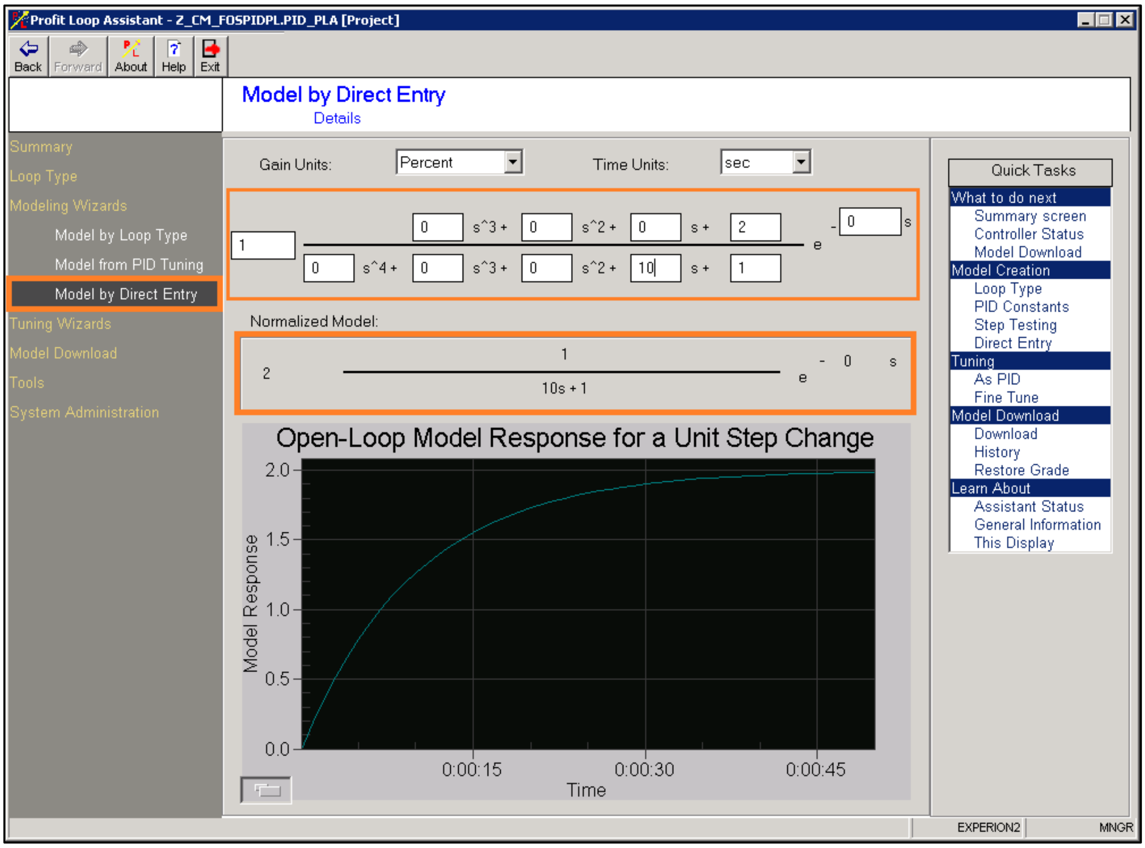


Figure 43: PID-PL Function Block

The process model can be defined within the function block in several ways, modelled by step testing, by PID tuning parameters, by loop type or, if known, entered directly (Honeywell 2013). As the process model used for testing was known (Equation 13), direct entry of the process parameters was performed using Honeywell's Profit Loop Assistant (accessible via the advanced tab of the PID_PLA_1 function block properties). Figure 44 shows the steps performed to enter the model into the Profit Loop Assistant, to be loaded to the controller.

Equation 13: Model- Based Controller (PID-PL) Process Model

$$G(s) = \frac{2}{10s + 1}$$



Model by Direct Entry

Details

Gain Units: Percent

Time Units: sec

$$1 \quad \frac{0\ s^3 + 0\ s^2 + 0\ s + 2}{0\ s^4 + 0\ s^3 + 0\ s^2 + 10\ s + 1} \quad e^{-0\ s}$$

Normalized Model:

$$2 \quad \frac{1}{10s + 1} \quad e^{-0\ s}$$

Figure 44: Profit Loop Assistant Model by Direct Entry

Once the model was loaded to the controller, it was possible to tune the controller to perform tests. Tuning the controller was achieved by defining a numerical performance ratio value between 0 and 10. Honeywell indicates that a performance ratio of 1 corresponds to the controller achieving the desired setpoint in a time approximately equal to that of the open loop response time (Honeywell 2013). Increasing the performance ratio to a value greater than 1 provides slower control with increased robustness to modelling errors. Conversely, reducing the performance ratio to a value less than 1 results in more aggressive control, driving the process variable to setpoint faster than that of the open loop response.

4.4.3.1 Model-Based Predictive Control Testing and Validation

As Honeywell's profit loop algorithm is patented (Honeywell 2016d), the algorithm is unknown, meaning the replication of previous tests conducted using Simulink was not possible. To confirm correct configuration and implementation of the model-based predictive controller, a comparison of the settling time of the open loop and controlled loop was conducted. As indicated earlier, selecting a performance ratio equal to 1, the time taken to achieve setpoint should be approximately equal to that of open loop response time for the same system. Two tests were performed to confirm this characteristic; an open loop step input change followed by a setpoint change in the controlled system, both at time 10. The results of both tests are presented in Figure 45. Original test plots can be viewed in Appendix H.

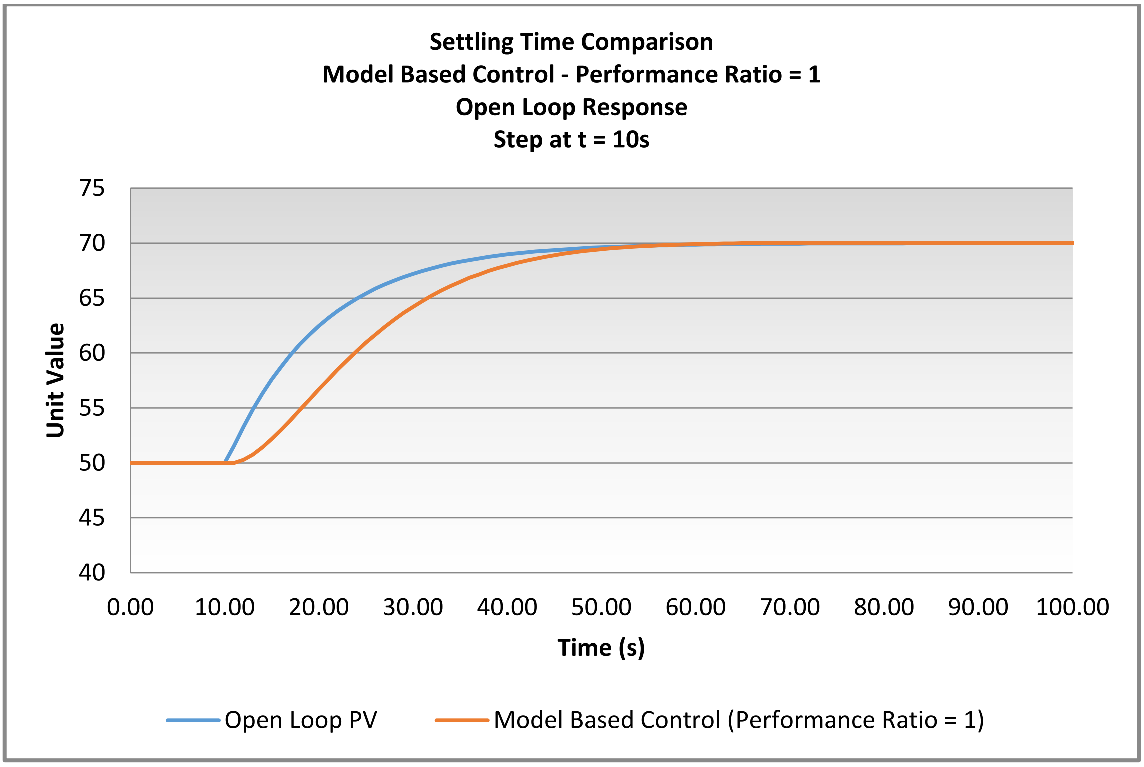


Figure 45: Model-Based Control Testing - Settling Time Comparison

Referring to Figure 45, although each of the responses follows a different trajectory, they both appear to settle to their respective final values at approximately the same time. By refining the plot to highlight the time each of the responses take to settle to their final values (Figure 46), it is evident the model-based controller settles at 95 seconds while the open loop settles at 94 seconds.

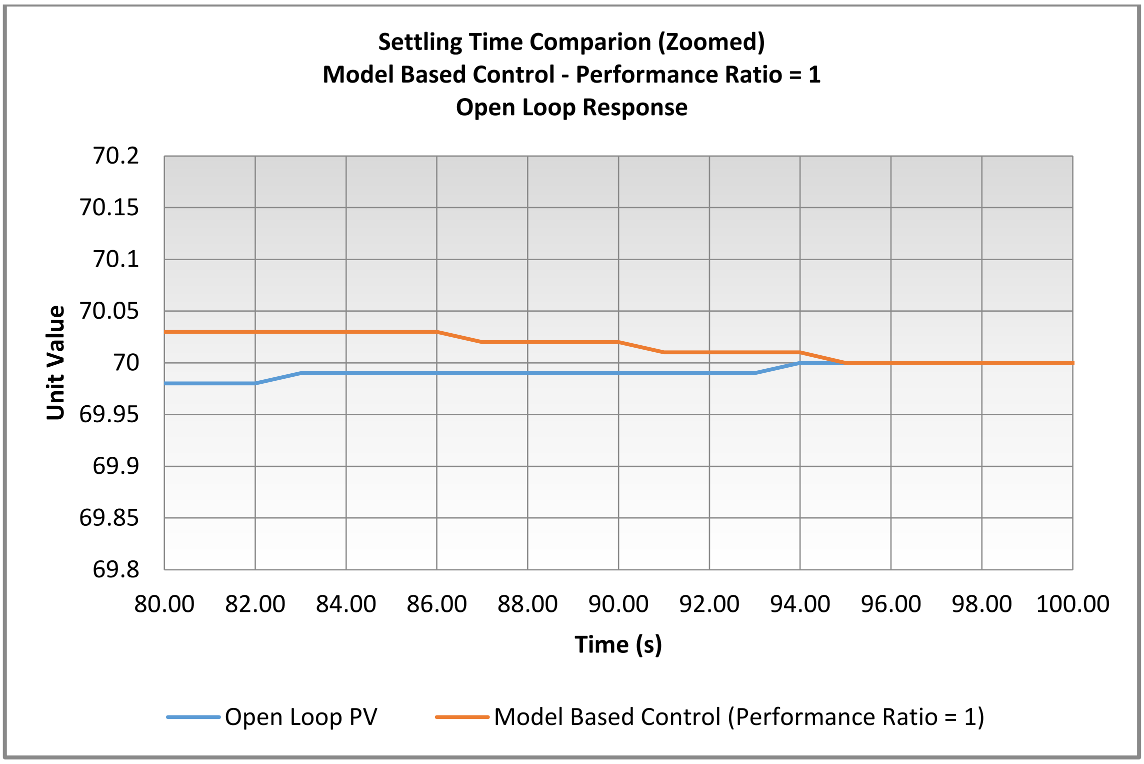


Figure 46: Model-Based Control Testing - Settling Time Comparison Refined

Based on the near identical settling times of the controlled and open loop process simulations, the test was deemed successful, confirming the implementation of the PID-PL function block within Experion was correct.

As an additional exercise, to confirm the operation of the performance ratio, five identical setpoint change tests were conducted with varying performance ratios. The tests performed confirmed that changes in the performance ratio had the correct effect on process response as identified earlier, providing further justification that the controller was implemented correctly. The comparison plot can also be viewed in Appendix H.

4.5 Phase 1 Concluding Remarks

Throughout the duration of the first phase of the project, several key achievements were made. Modifications were made to the configuration settings of the Experion system, allowing for the successful implementation of programs and control strategies. Several programs were developed to assist in understanding the programming nature of Experion and its complexities, before successful implementation and testing of three control strategies. The intensive nature of the research performed during this phase allowed for the creation of an extensive list of learning materials including comprehensive details on the configuration, simulation and control of processes contained within the Experion environment. These learning materials are discussed in greater detail in Phase 3 of this report.

5 Delivery against Project Scope – Phase 2

Following successful completion of the objectives for Phase 1, research and testing for the second phase of the project, OPC data exchange, was commenced.

Prior to attempting to successfully communicate with third party software applications, research into past findings was conducted. In previous years, students had experienced difficulty gaining reliable communication between the Experion server and third party client applications. By applying a systematic approach to the research, testing and validation, it was possible to successfully communicate in real time with four different client software applications:

- Kepware OPC Quick Client (OPC Quick Client) ;
- MathWorks MATLAB (MATLAB);
- MathWorks Simulink (Simulink);
- National Instruments LabVIEW (LabVIEW).

The approach adopted was to initially research the requirements for configuring the Experion server as an OPC server visible to client applications. Once the Experion server had been set up, the OPC Quick Client program was used to confirm correct configuration and data access. MATLAB's OPC Toolbox was then used for creating a client server for reading and writing data to the Experion server. Tests were conducted on the data acquired from MATLAB via OPC, to confirm correct operation before attempting real-time control algorithms using Simulink and LabVIEW. The methods and results of these tests are presented in the following sections.

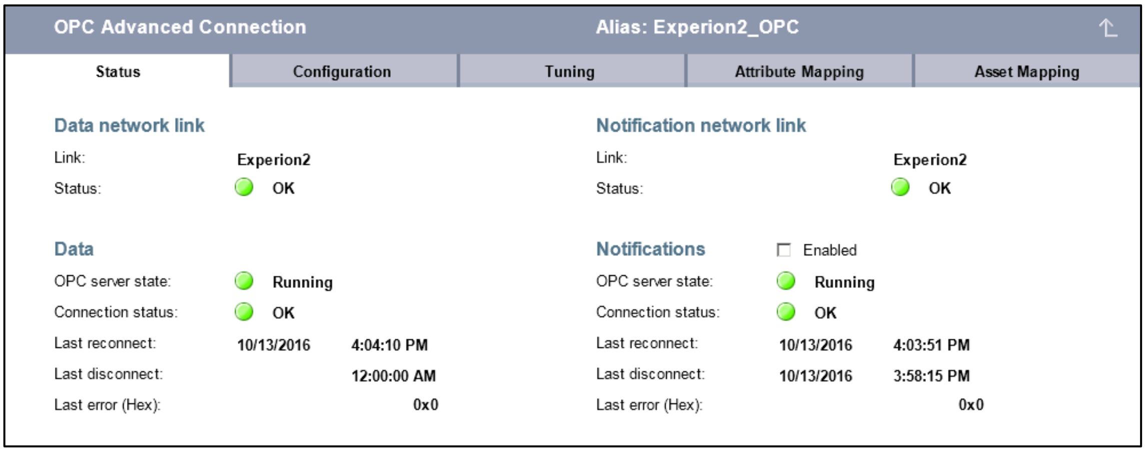
All tests presented in the following sections made use of the control module developed containing the first order system (Z_CM_FOS), with the process parameters set to create the system identified by Equation 14.

Equation 14: OPC Test System

$$G_2(s) = \frac{2}{10s+1}$$

5.1.1 Experion OPC Configuration

Before communication with client OPC programs was attempted, the Experion server was configured as the OPC server. This was achieved by opening Experion's Configuration Studio and entering the System Interface configuration settings. Within the configuration settings, it was possible to define the properties of the OPC Experion server and perform a status check on the server as indicated by Figure 47.



OPC Advanced Connection   Alias: Experion2_OPC

Data network link

Link:	Experion2	
Status:	● OK	

Data

OPC server state:	● Running	
Connection status:	● OK	
Last reconnect:	10/13/2016	4:04:10 PM
Last disconnect:		12:00:00 AM
Last error (Hex):		0x0

Notification network link

Link:	Experion2	
Status:	● OK	

Notifications □ Enabled

OPC server state:	● Running	
Connection status:	● OK	
Last reconnect:	10/13/2016	4:03:51 PM
Last disconnect:	10/13/2016	3:58:15 PM
Last error (Hex):		0x0

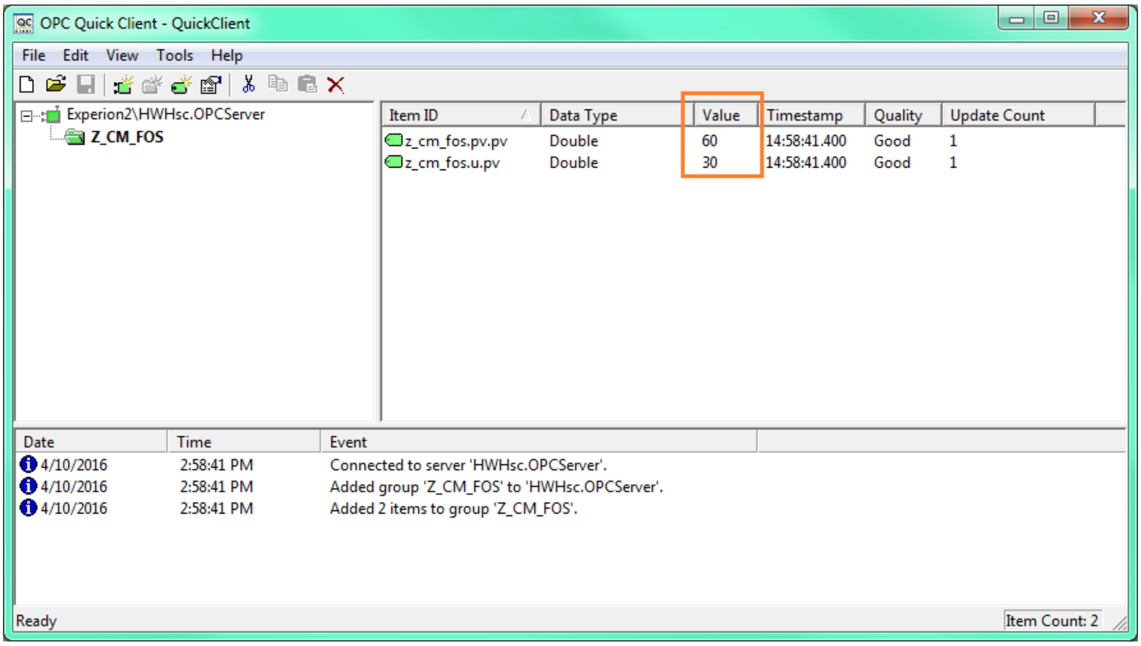
Figure 47: Experion OPC Server Status

To confirm the correct configuration of the Experion OPC server, OPC Quick Client was then opened to create a successful connection to the server.

5.1.2 Data Exchange with OPC Quick Client

To test the operation of the Experion OPC server, a data access server connection was created within the OPC Quick Client software. This connection allowed for a subgroup to be created, where items (Experion data values) could be monitored and controlled.

For testing purposes, a group (Z_CM_FOS) was created for the input and output variables of the first order system control module within Experion to be monitored and controlled. As indicated in Figure 48, the numeric values of the items PV and U of the control module were read from the Experion Server via OPC communication. These are the same values able to be monitored in Experion's Control Builder Environment as presented in Figure 49 (numerical values 60 and 30).



Experion2\HWHsc.OPCServer
Z_CM_FOS

Item ID	Data Type	Value	Timestamp	Quality	Update Count
z_cm_fos.pv.pv	Double	60	14:58:41.400	Good	1
z_cm_fos.u.pv	Double	30	14:58:41.400	Good	1

Date	Time	Event
4/10/2016	2:58:41 PM	Connected to server 'HWHsc.OPCServer'.
4/10/2016	2:58:41 PM	Added group 'Z_CM_FOS' to 'HWHsc.OPCServer'.
4/10/2016	2:58:41 PM	Added 2 items to group 'Z_CM_FOS'.

Ready
Item Count: 2

Figure 48: OPC Quick Client FOS Test Values

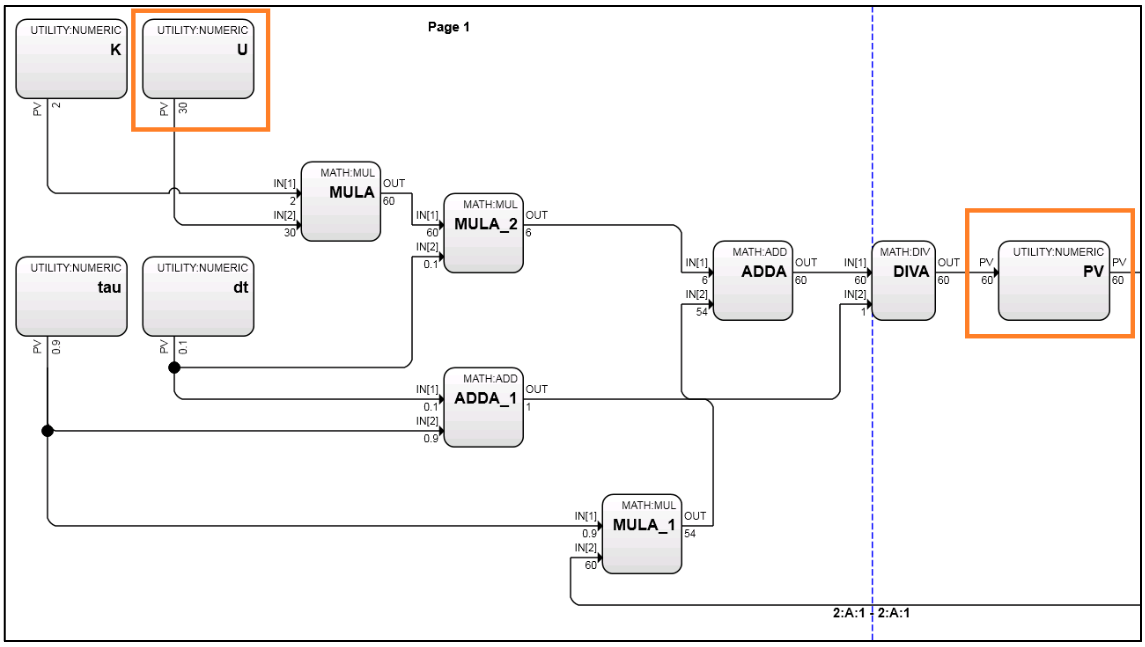
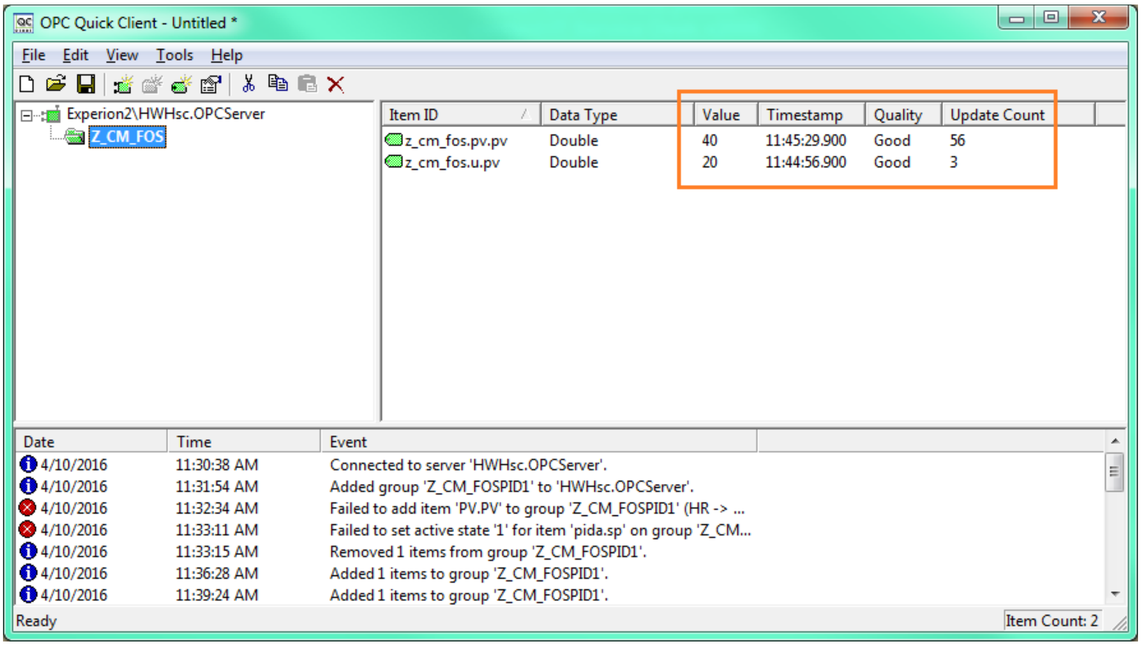


Figure 49: OPC Experion FOS Test Values

Using OPC Quick Client, it was also possible to write a new input value (U) to the control module function block and view the output of the system dynamically as the software updated. As the value PV updates with the change in input, the update count and time stamp refresh periodically, displaying the new value acquired from the Experion server. As an example, the process input (U) was changed from value 30 to 20, with the new final output (PV) reading value 40, with an updated timestamp and update count, confirming correct operation (see Figure 50).


OPC Quick Client - Untitled *
File Edit View Tools Help
Experion2\HWHsc.OPCServer
Z_CM_FOS
Ready
Item Count: 2

Item ID	Data Type	Value	Timestamp	Quality	Update Count
z_cm_fos.pv.pv	Double	40	11:45:29.900	Good	56
z_cm_fos.u.pv	Double	20	11:44:56.900	Good	3

```
Date        Time         Event
4/10/2016   11:30:38 AM  Connected to server 'HWHsc.OPCServer'.
4/10/2016   11:31:54 AM  Added group 'Z_CM_FOSPID1' to 'HWHsc.OPCServer'.
4/10/2016   11:32:34 AM  Failed to add item 'PV.PV' to group 'Z_CM_FOSPID1' (HR -> ...
4/10/2016   11:33:11 AM  Failed to set active state '1' for item 'pida.sp' on group 'Z_CM...
4/10/2016   11:33:15 AM  Removed 1 items from group 'Z_CM_FOSPID1'.
4/10/2016   11:36:28 AM  Added 1 items to group 'Z_CM_FOSPID1'.
4/10/2016   11:39:24 AM  Added 1 items to group 'Z_CM_FOSPID1'.
```

Figure 50: OPC Quick Client FOS Step Input Dynamic Monitoring

As OPC Quick Client was only intended to be used to ensure correct configuration of the OPC server, no additional tests were conducted with this application and focus was shifted to establish communication with MATLAB and LabVIEW.

5.1.3 Data Exchange with MATLAB

Following successful implementation of the Experion OPC, progression to higher level programs was then pursued. Initially, connection between MATLAB and Experion to replicate results attained using OPC Quick Client was achieved. Following the successful configuration of the MATLAB client OPC server, MATLAB's block diagram for simulation and control, Simulink, was then configured for real-time data access and process control.

5.1.3.1 MATLAB

MATLAB's OPC Toolbox provides the tools and functions required for live access to OPC server data (MathWorks 2016a). The use of this toolbox allowed for the configuration of the OPC data access client, enabling the reading and writing of data to and from the Host OPC Experion server.

MATLAB's OPC Toolbox includes a range of in-built functions for the monitoring and manipulation of data through text based commands entered in MATLAB's command window, or developed as part of a MATLAB program script.

To replicate the achievements made using OPC Quick Client, it was required to initially create an OPC data access client and establish connection to the Experion server. The creation and connection to the OPC data access server was achieved by making use of MATLAB's inbuilt functions 'opcda' and 'connect' respectively. Once the connection had been made, a group was created to contain the items which were to be read from or written to the Experion server. The group was named 'Z_CM_FOS' and created using MATLAB's 'addgroup' function.

Individual items could now be entered into the group, identified by their tag names using MATLAB's 'additem' function. With the correct connection and creation of groups and items, it was possible to read and write values using MATLABS 'read' and 'write' functions, as indicated by the block diagram in Figure 51. Further explanation and code syntax of each of the previously mentioned MATLAB functions can be found in Appendix I.

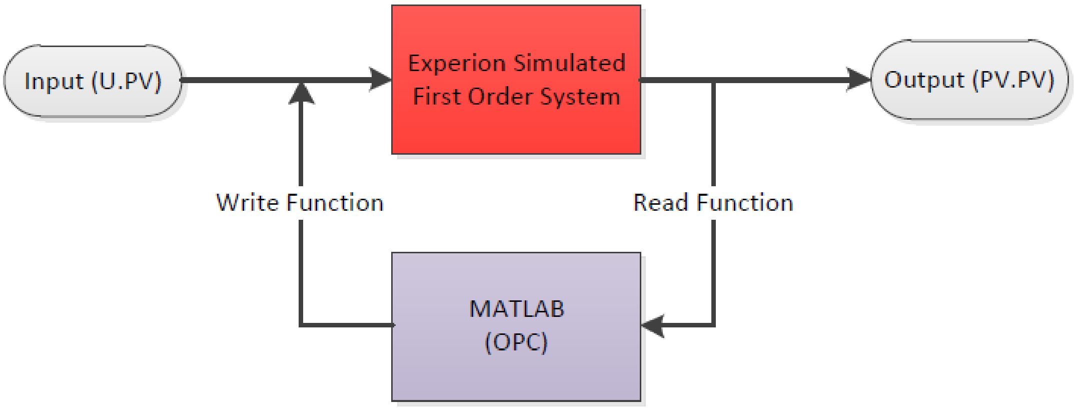


Figure 51: MATLAB / Experion OPC First Order System - Data Exchange Block Diagram

MATLAB Testing

By entering the previously discussed functions into MATLAB's command window as illustrated below, it was possible to read the process variable PV contained within the control module Z_CM_FOS on the Experion server. The item U was also added to the group for future tests to be conducted, including writing values to the Experion server.

The following MATLAB code was entered within the command window to achieve communication.

MATLAB code for configuring OPC data access

```
da = opcda('Experion2','HWHsc.OPCServer')
connect(da
grp = addgroup(da,'Z_CM_FOS')
PV = additem(grp, 'Z_CM_FOS.PV.PV', 'double')
U = additem(grp, 'Z_CM_FOS.U.PV', 'double');
PVread = read(PV)
```

By then entering the 'PVread' variable in the command window, the following output was presented, providing the numeric output value of the first order system PV.

Read current value PV

```
PVread =
       ItemID: 'Z_CM_FOS.PV.PV'
        Value: 60.0000
      Quality: 'Good: Non-specific'
    TimeStamp: [2016 10 23 15 23 41.6000]
        Error: ''
```

Comparing the value of ‘PVread’ via OPC to the value read directly from the Experion server (Figure 52), it can be seen that the numbers are identical, indicating MATLAB is operating correctly.

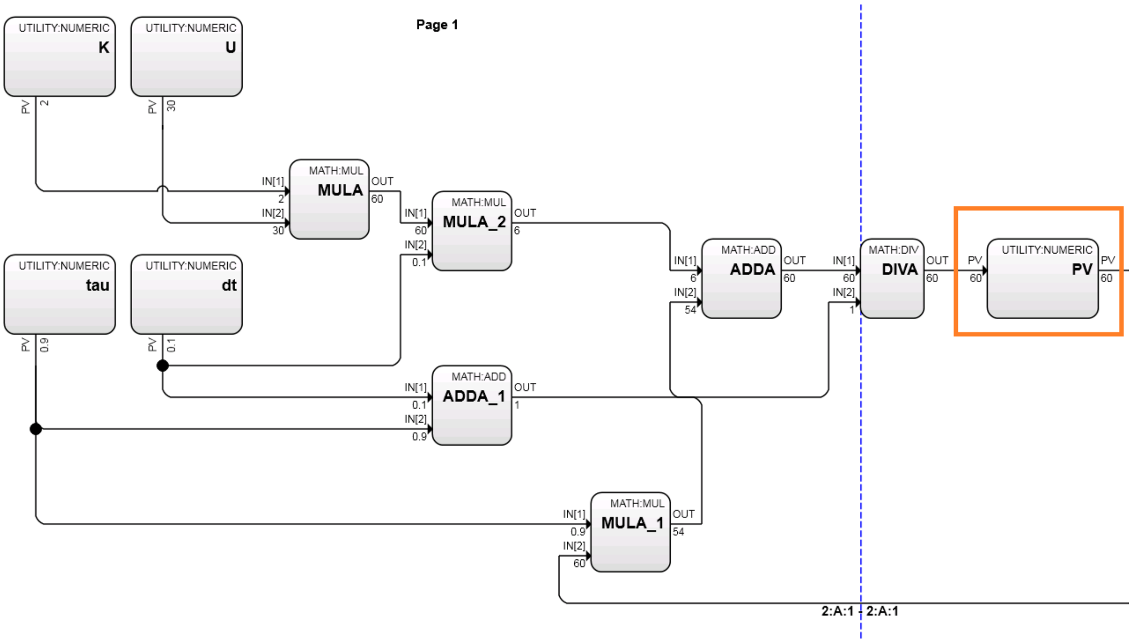


Figure 52: MATLAB Testing – PV Value

After successfully reading values from the server, a second test was performed to ensure writing values to Experion was possible. A numerical value of 20 was written to the variable U on the Experion server using MATLAB’s ‘write’ function as illustrated below.

Write value 20 to item U

```
write(U,20);
```

By entering the ‘PVread’ variable into the command window on two separate occasions (approximately 5 seconds and 80 seconds after changing the input), this step down in process input had the effect of dynamically altering the process output on Experion, yielding the following results, noting the time stamps and associated numerical values.

Read value U at t ~ 5s after step change

```
Uread = read(U);
Uread =
       ItemID: 'Z_CM_FOS.PV.PV'
        Value: 52.9860
      Quality: 'Good: Non-specific'
    TimeStamp: [2016 10 23 15 45 10.2000]
        Error: ''
```

Read Value U at t ~ 80s after step change

```
Uread = read(U);
Uread =
       ItemID: 'Z_CM_FOS.PV.PV'
        Value: 40.0000
      Quality: 'Good: Non-specific'
    TimeStamp: [2016 10 23 15 46 31.8000]
        Error: ''
```

As an additional test, to further ensure MATLAB was acquiring the correct data from the Experion server, a test was conducted whereby a step in the input U to the simulated Experion system was introduced, with the resulting output PV being acquired and logged periodically using MATLAB. The values acquired from MATLAB would then be compared with values acquired directly from Experion, with the expectation that both plots would be identical.

The data for the test was acquired by making use of the previously mentioned functions within a MATLAB script, to write a new input to the Experion server's first order system control module Z_CM_FOS, while periodically sampling and storing the output value for plotting. The final MATLAB script can be found in Appendix J.

Upon running the simulation and stepping the input from value 20 to 30, the resultant output from MATLAB was identical to the Experion data, as presented in Figure 53. The raw data and associated plot acquired from MATLAB can be found in Appendix K.

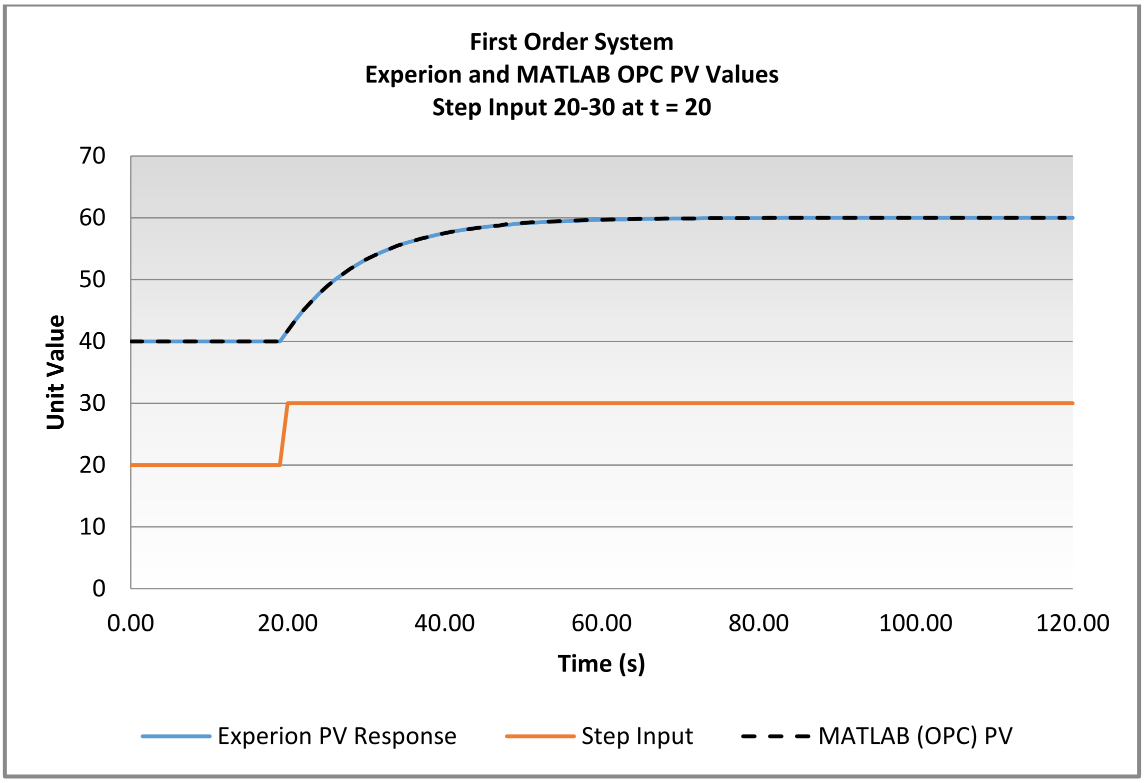


Figure 53: MATLAB Testing First Order System (Step input 20-30)

Two additional tests were conducted with varying step input changes, each producing the same result. These comparison plots can also be found in Appendix K. As the data acquired with MATLAB was identical to data acquired from Experion for all tests, no further tests were conducted using MATLAB and focus was shifted to gaining real-time communication using Simulink.

5.1.3.2 Simulink

As part of this project, it was intended to make use of Simulink's OPC data communication abilities for the testing and simulation of real-time process control strategies. Initially, communication for reading and writing server values, as previously achieved using MATLAB and OPC Quick Client, was established. Upon successful communication and data exchange, Simulink's in-built controller blocks were used to control the simulated Experion first order process (Z_CM_FOS) in real time.

Data exchange with the Experion server was achieved in a similar manner to the previous software applications; with the creation and connection to the server prior to reading and writing server values. Initial connection was achieved by inserting an OPC configuration block (as displayed in Figure 54) into a blank Simulink model and configuring settings for connection to the OPC server created within Experion. Within the same Simulink model, it was then possible to insert Simulink's 'OPC Read' and 'OPC Write' blocks for real-time data exchange.

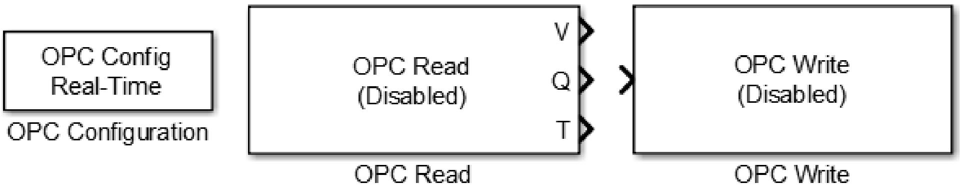


Figure 54: Simulink OPC Communication Blocks

Within both the 'OPC Read' and 'OPC Write' block configuration settings, OPC server items were configured by navigating to the appropriate item (e.g. PV.PV) on the OPC server.

Simulink Testing

Similar to MATLAB and OPC Quick Client, tests were conducted by writing to the input (U) and reading from the output (PV) of the Experion simulated first order system control module (Z_CM_FOS), as indicated by the block diagram displayed in Figure 55

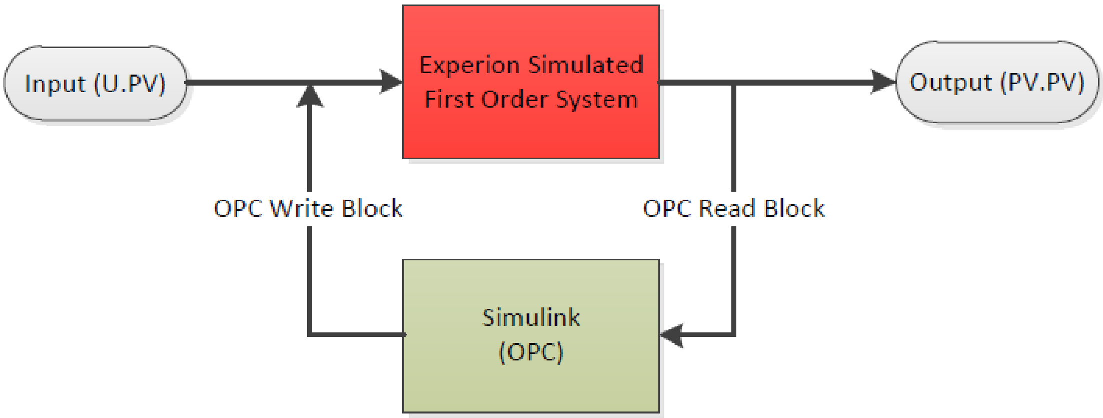


Figure 55: Simulink / Experion OPC First Order System - Data Exchange Block Diagram

As the simulation runs in real-time, a numerical display and scope were included in the model, allowing for the dynamic monitoring of the process response (see Figure 56).

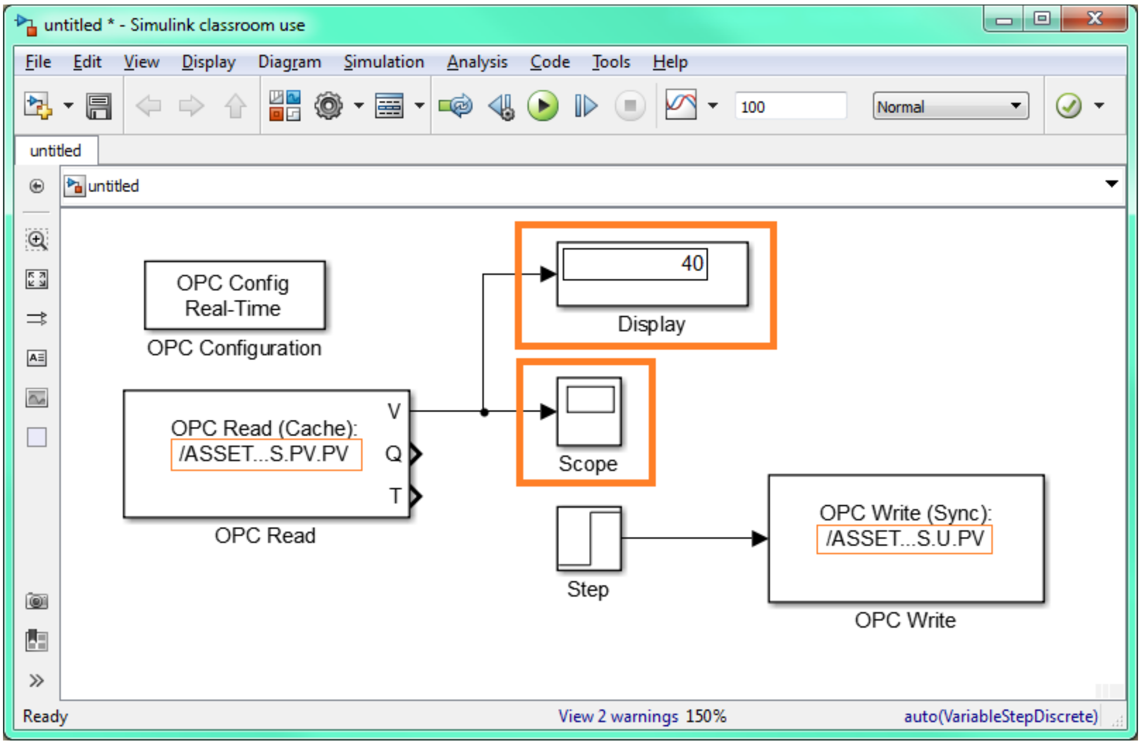


Figure 56: Simulink OPC Model - First Order System Monitoring and Control

Setting the step input to step from value 20 to 30 at time 10 produced the following output on the scope at the time intervals 50s and 100s, identifying the dynamic monitoring abilities. Refer to Figure 57 and Figure 58.

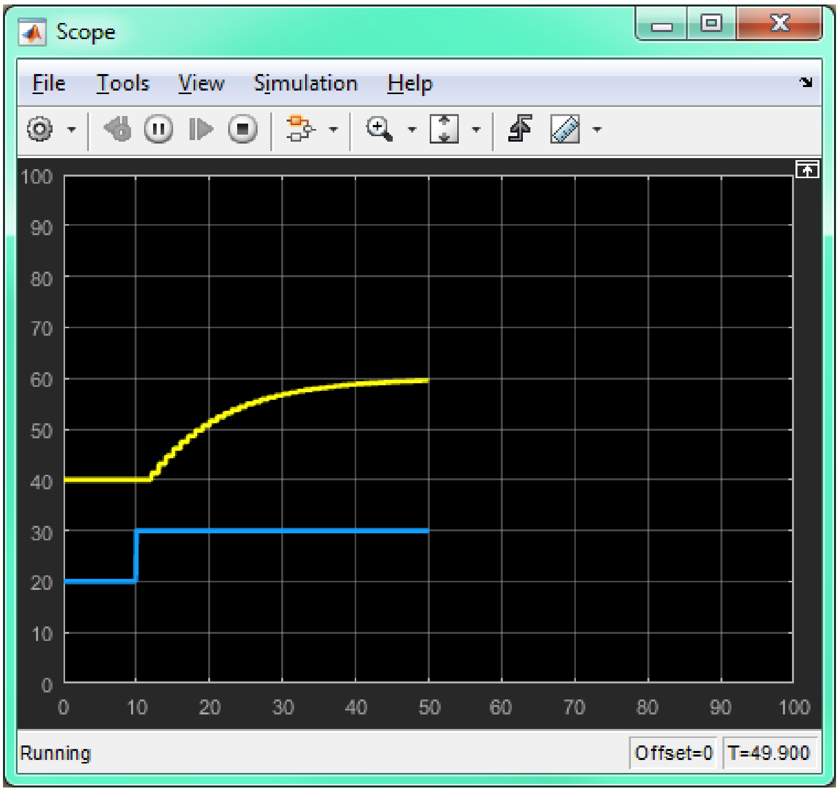


Figure 57: Simulink OPC Testing - First Order System Response at t = 50s

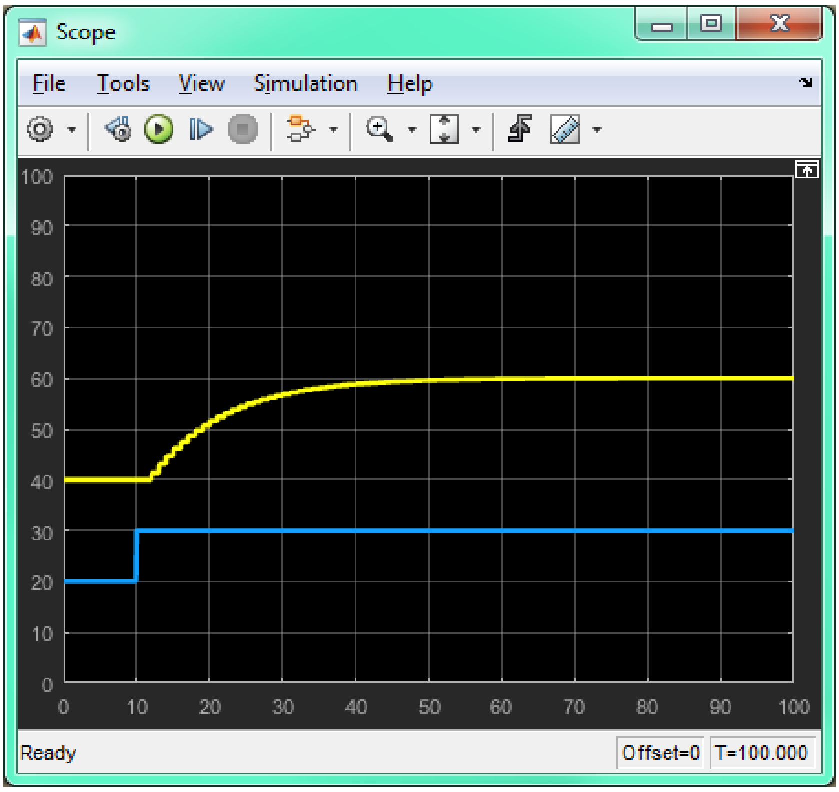


Figure 58: Simulink OPC Testing - First Order System Response at t = 100s

Based on the data presented in Figure 57 and Figure 58, it can be seen that the reading and writing of data to and from the Experion server via OPC operated correctly. However, as with the MATLAB OPC data acquired, it was desired to validate the Simulink data values against the data extracted directly from the Experion server. This was achieved by plotting the two sets of data on the same axis, with the expectation of identical results.

Both systems produced the same response with the results presented in Figure 59. As the data point values recorded were identical, no additional step tests were performed. The raw tabulated data values can be viewed in Appendix L

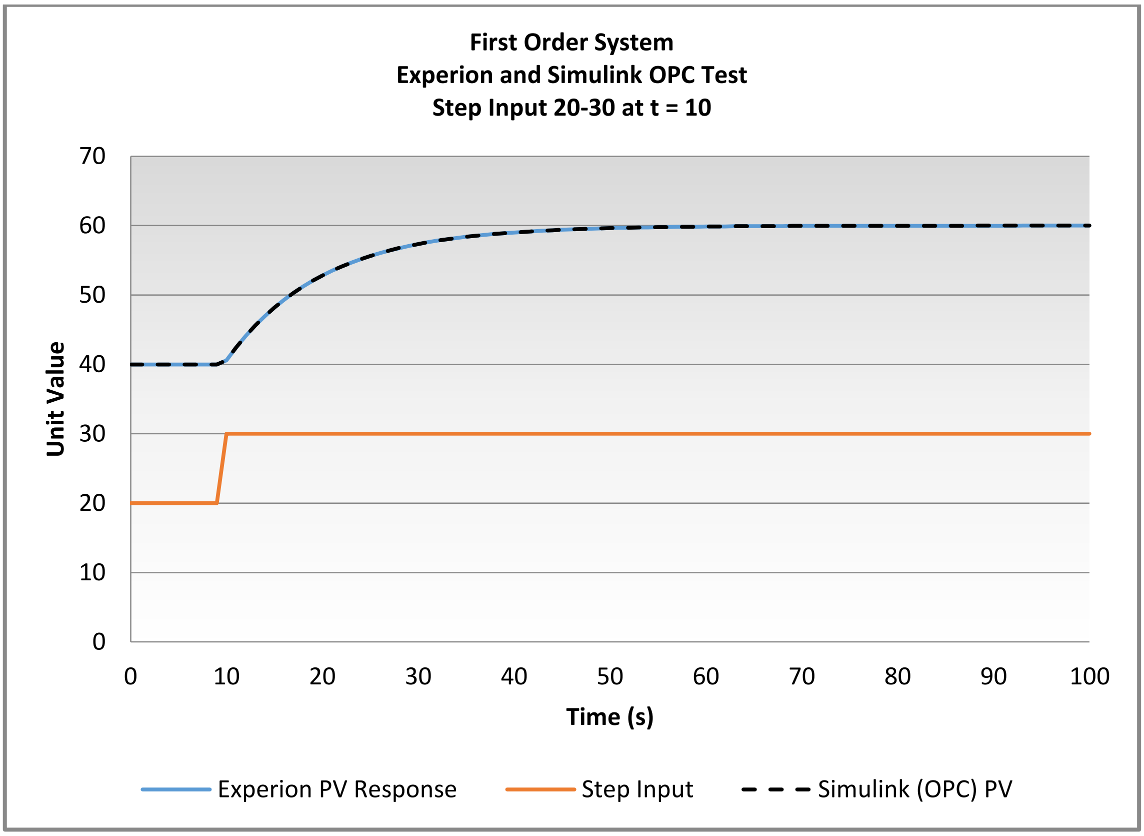


Figure 59: Simulink OPC Testing First Order System (Step input 20-30)

With the successful testing of basic read and write functions using Simulink, it was decided to control the Experion simulated first order system using Simulink's controller function blocks, as indicated by the block diagram in Figure 60.

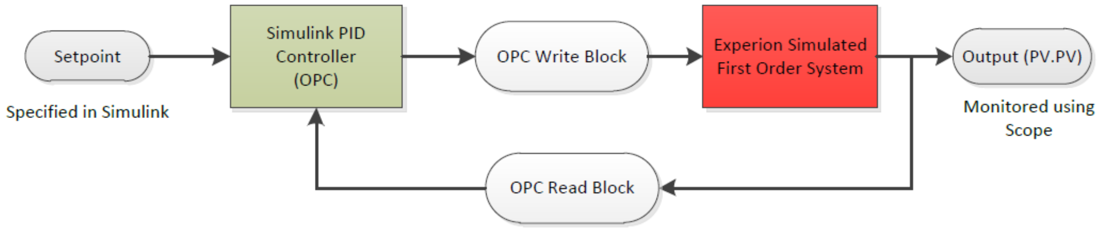


Figure 60: Simulink / Experion OPC First Order System with PID Data Exchange Block Diagram

The implementation of this control strategy configuration within the Simulink model is presented in Figure 61.

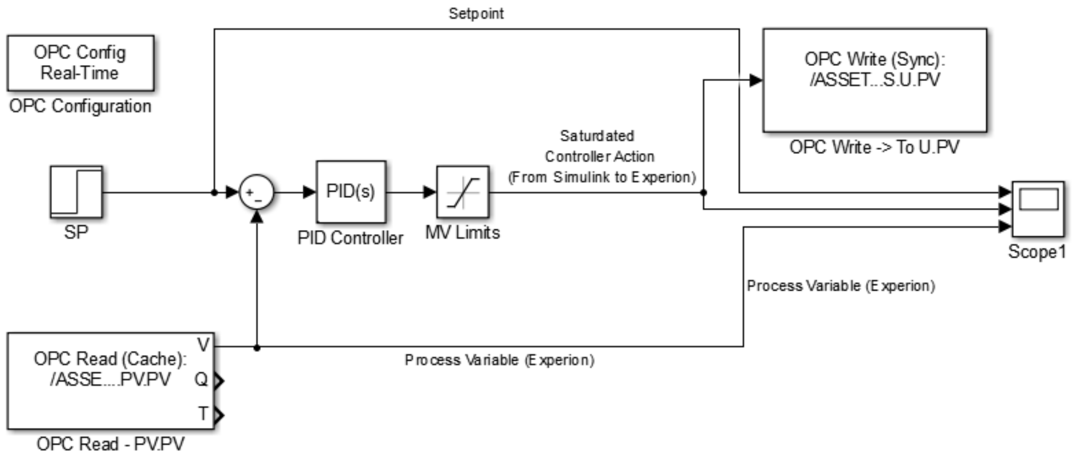

Setpoint
OPC Config Real-Time
OPC Configuration
SP
+
-
PID(s)
PID Controller
MV Limits
Saturdated Controller Action (From Simulink to Experion)
OPC Write (Sync): /ASSET...S.U.PV
OPC Write -> To U.PV
Scope1
Process Variable (Experion)
OPC Read (Cache): /ASSE....PV.PV
V
Q
T
OPC Read - PV.PV
Process Variable (Experion)

Figure 61: Simulink OPC Model – First Order System with PID Control

For each of the real-time control simulation tests conducted, the controller parameters identified in Table 9 were configured in the Simulink PID block. As with previous tests, controller tuning parameters were selected arbitrarily.

Table 9: Simulink OPC PID Testing – Controller Tuning Parameters

	Simulink PID
Controller Gain	1
Integral Time	0.1
Derivative Time	0

Testing the controller was conducted by introducing a series of setpoint changes and logging the response of the system. To verify that the controller was operating as intended, identical setpoint changes were made in a separate standalone Simulink model developed, containing the same process and controller parameters as the OPC Simulink model, as illustrated in Figure 62.

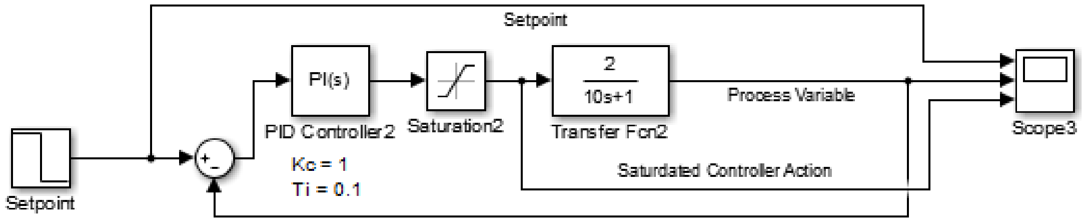


Figure 62: Simulink OPC Testing - Standalone Simulink Model

The first test conducted involved stepping the setpoint value from 40 to 60 at time 20. The response of the controller and respective system output for both simulations are displayed in Figure 63.

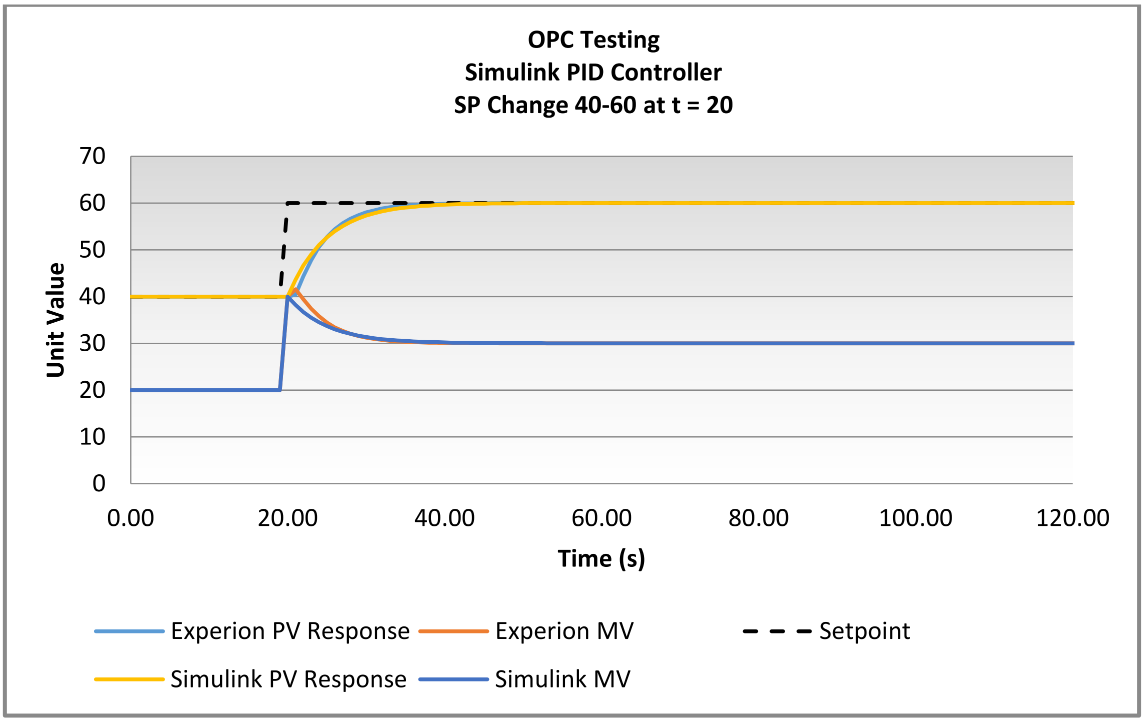


Figure 63: Simulink OPC Testing - First Order System with PID

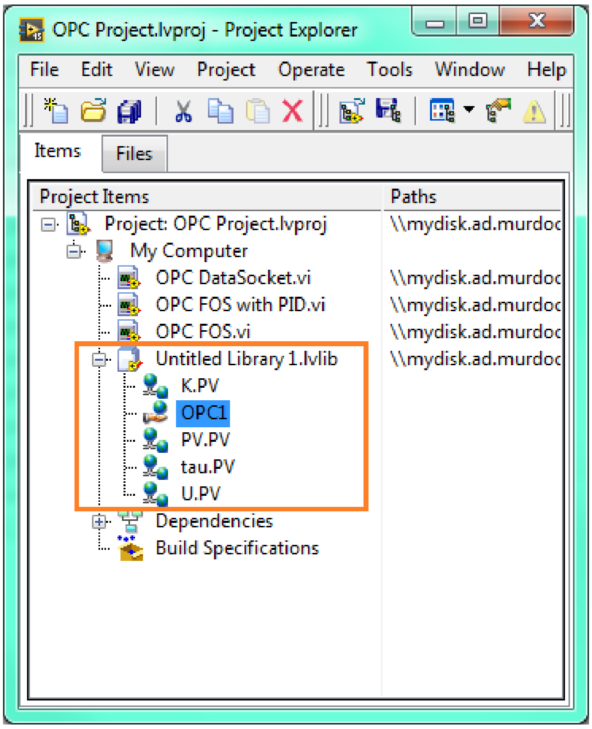
Based on the data presented, it is clear that the controller is operating correctly, producing near identical responses to the ideal simulation performed in Simulink. As with previous tests, slight discrepancies in response were recorded, however as previously mentioned, these are believed to be a result of the Euler approximation used for the Experion first order simulation system.

Three additional tests were conducted and confirmed consistent results, with these comparison plots presented in Appendix M. With the successful testing of the implementation of a Simulink controlled Experion system, no further tests were conducted. Focus was then shifted to achieving the same results using LabVIEW.

5.1.4 Data Exchange with LabVIEW

For the purpose of this project, LabVIEW's OPC connectivity property was utilised for the testing and simulation of real-time control of an Experion simulated process. To achieve a successful outcome, a systematic approach was adopted. This approach involved initially reading and displaying values from the Experion server, followed by successfully writing numerical values to the server using LabVIEW. Upon validation of the data, LabVIEW's control toolbox functions were utilised to control Experion's simulated first order process (Z_CM_FOS).

OPC communication with Experion was achieved by use of LabVIEW's Datalogging and Supervisory Control (DSC) Module. LabVIEW's DSC Module allows for the development of monitoring and control systems, with the tools necessary for historical archive creation, trending, management of alarms and events as well as networking capabilities via OPC (National Instruments 2016). Communication with an OPC server is achieved by the creation of a linked I/O server via OPC. This server connection allows for the creation of a Shared Variable Engine (SVE), permitting the continuous exchange of Experion data items (known as bound variables in LabVIEW) via OPC. Figure 64 displays the created SVE and OPC server connection, with the Experion first order system items listed for use within the LabVIEW program (K.PV, PV.PV, tau.PV and U.PV).


OPC Project.lvproj - Project Explorer

File Edit View Project Operate Tools Window Help

Items Files

Project Items	Paths
Project: OPC Project.lvproj	\\mydisk.ad.murdoc
My Computer	
OPC DataSocket.vi	\\mydisk.ad.murdoc
OPC FOS with PID.vi	\\mydisk.ad.murdoc
OPC FOS.vi	\\mydisk.ad.murdoc
Untitled Library 1.lvlib	\\mydisk.ad.murdoc
K.PV	
OPC1	
PV.PV	
tau.PV	
U.PV	
Dependencies	
Build Specifications	

Figure 64: LabVIEW OPC Shared Variable Engine

Once the SVE was created with the appropriate bound variables, these variables could be monitored and controlled using shared variable nodes within a LabVIEW simulation model, known as a Virtual Instrument or 'VI' (National Instruments 2016b) (see Figure 65).

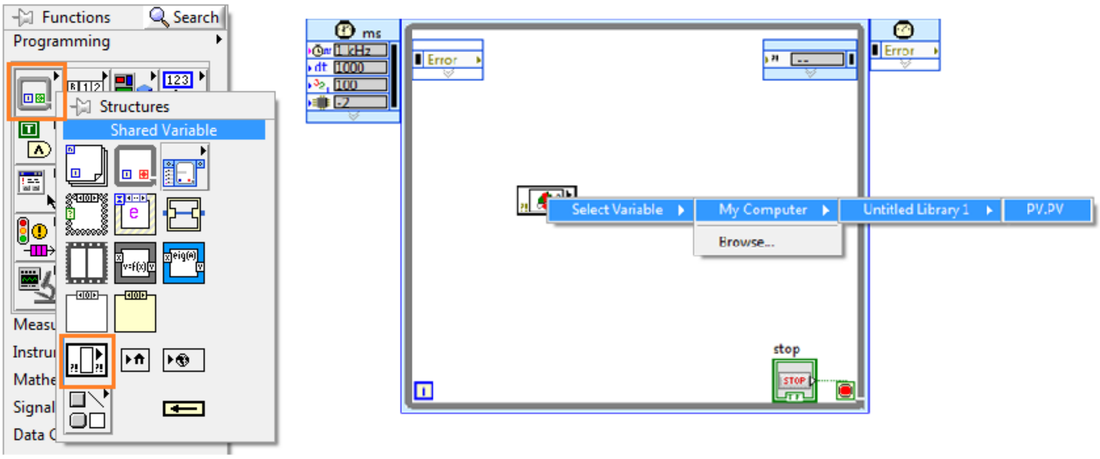


Figure 65: Shared Variable Node Configuration

Once configured, the shared variable node is then able to be read from, or written to, as illustrated by the screenshot taken from a developed LabVIEW program (Figure 66).

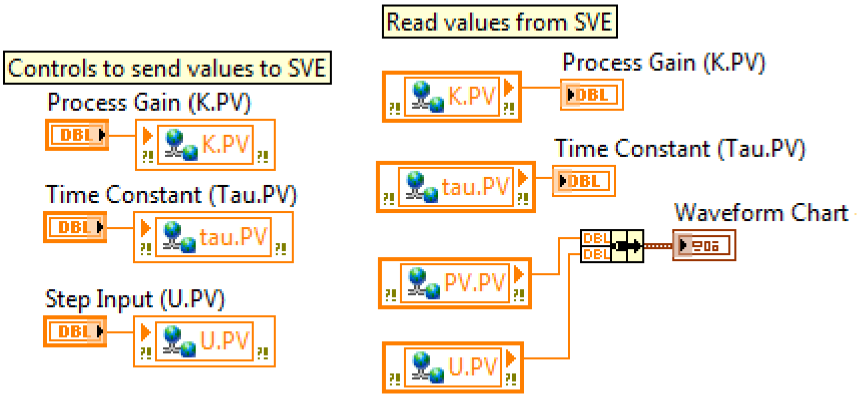


Figure 66: Shared Variable Node - Read and Write Values

LabVIEW Testing

As with the testing of the OPC communication using Simulink, tests were initially conducted by reading from, and writing to, the previously identified shared variables (Figure 64 and Figure 66), and logging the response. Again, these tests were conducted by making use of the first order system Experion control module Z_CM_FOS for consistency. See Figure 67.

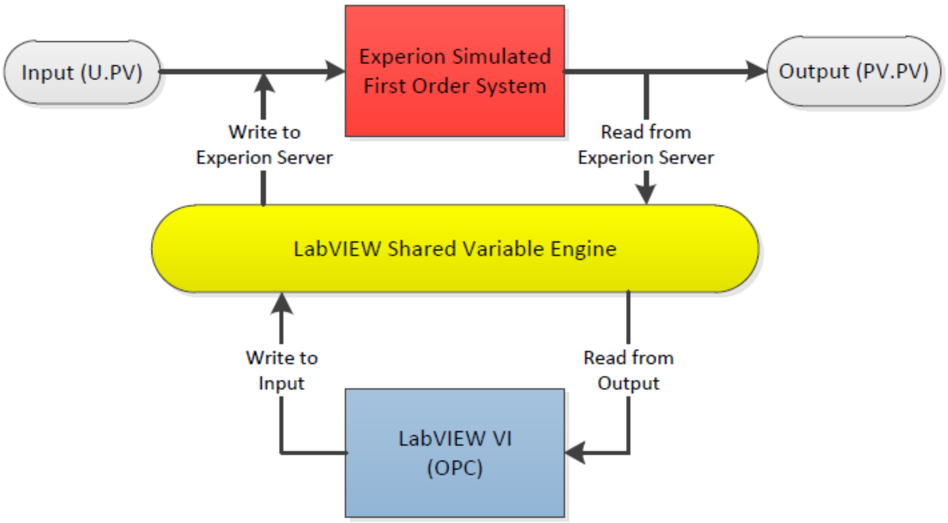
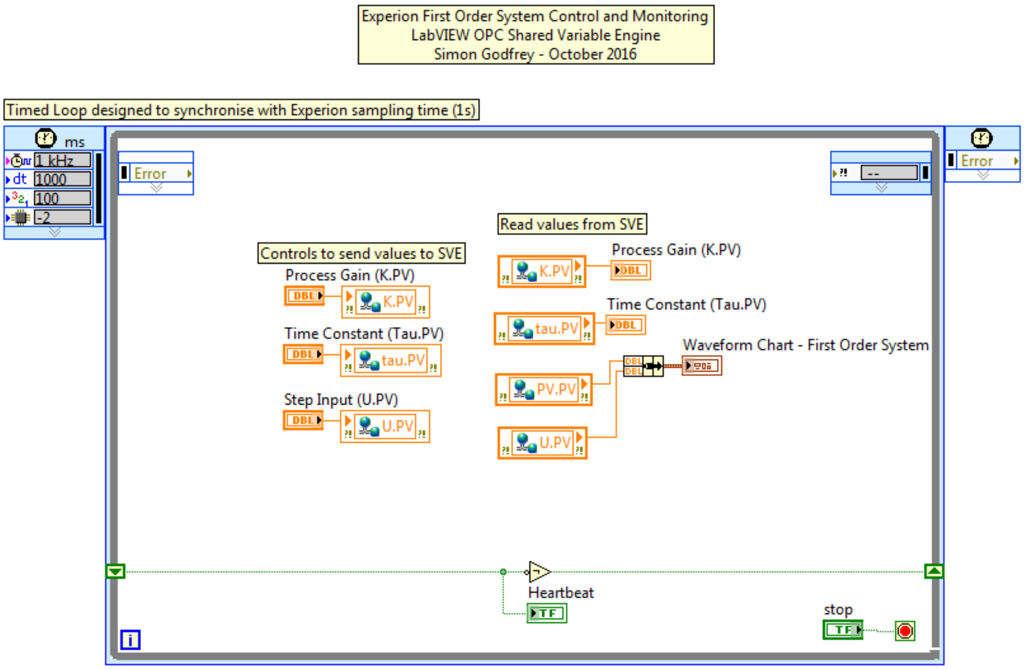


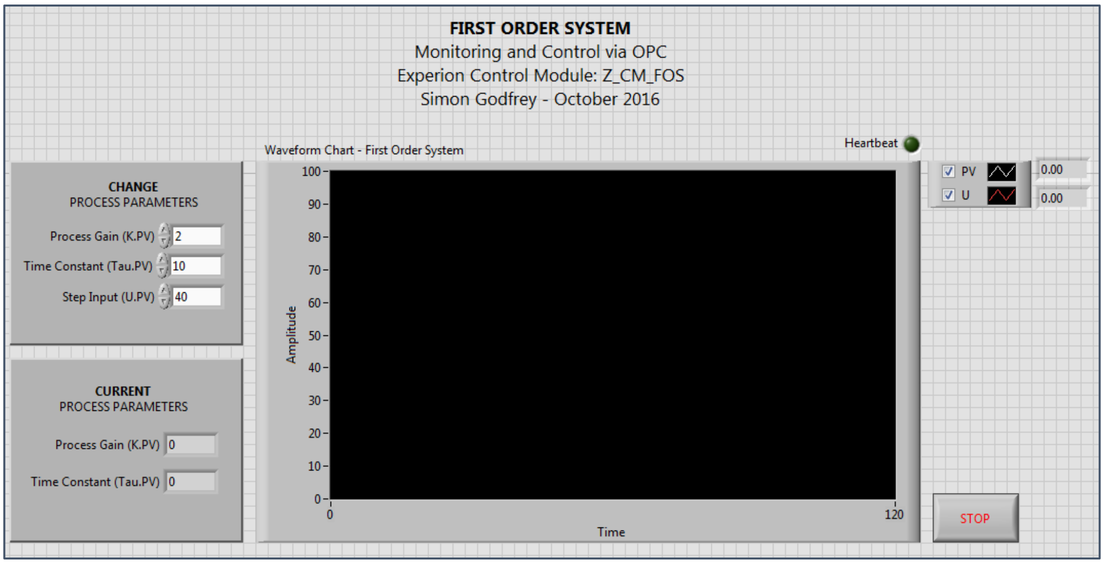
Figure 67: LabVIEW / Experion OPC First Order System - Data Exchange Block Diagram

A new LabVIEW VI was created with the appropriate controls and indicators programmed for the monitoring and control of the first order system in real-time, replicating the block diagram configuration in Figure 67. The LabVIEW block diagram and front panel of the VI are presented in Figure 68 and Figure 69.



Experion First Order System Control and Monitoring
LabVIEW OPC Shared Variable Engine
Simon Godfrey - October 2016
Timed Loop designed to synchronise with Experion sampling time (1s)
ms
1 kHz
dt 1000
100
-2
Error
Controls to send values to SVE
Process Gain (K.PV)
DBL
K.PV
Time Constant (Tau.PV)
DBL
tau.PV
Step Input (U.PV)
DBL
U.PV
Read values from SVE
Process Gain (K.PV)
K.PV
DBL
Time Constant (Tau.PV)
tau.PV
DBL
Waveform Chart - First Order System
PV.PV
U.PV
Heartbeat
TF
stop
TF
i
Error

Figure 68: LabVIEW OPC First Order System Block Diagram



FIRST ORDER SYSTEM
Monitoring and Control via OPC
Experion Control Module: Z_CM_FOS
Simon Godfrey - October 2016

CHANGE
PROCESS PARAMETERS

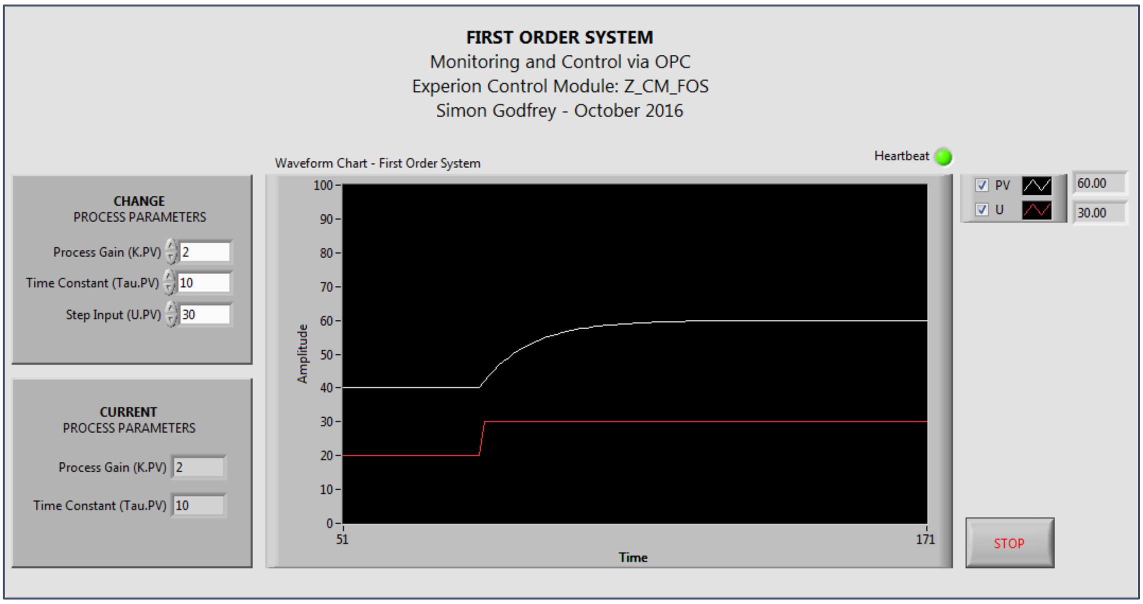
Process Gain (K.PV) 2
Time Constant (Tau.PV) 10
Step Input (U.PV) 40

CURRENT
PROCESS PARAMETERS

Process Gain (K.PV) 0
Time Constant (Tau.PV) 0

Figure 69: LabVIEW OPC First Order System Front Panel

By running the LabVIEW VI simulation, it was possible to alter the input to the system (U), and dynamically monitor the output (PV) via the waveform chart. Figure 70 displays the response of the first order system, with the introduction of step in input from value 20 to 30.



FIRST ORDER SYSTEM
Monitoring and Control via OPC
Experion Control Module: Z_CM_FOS
Simon Godfrey - October 2016

CHANGE
PROCESS PARAMETERS

Process Gain (K.PV) 2
Time Constant (Tau.PV) 10
Step Input (U.PV) 30

CURRENT
PROCESS PARAMETERS

Process Gain (K.PV) 2
Time Constant (Tau.PV) 10

☑ PV 60.00
☑ U 30.00

Figure 70: LabVIEW OPC First Order System Response

By logging the response of the LabVIEW system, it was possible to verify the data integrity, by comparing the LabVIEW acquired data to the data acquired directly from the Experion server. See Figure 71.

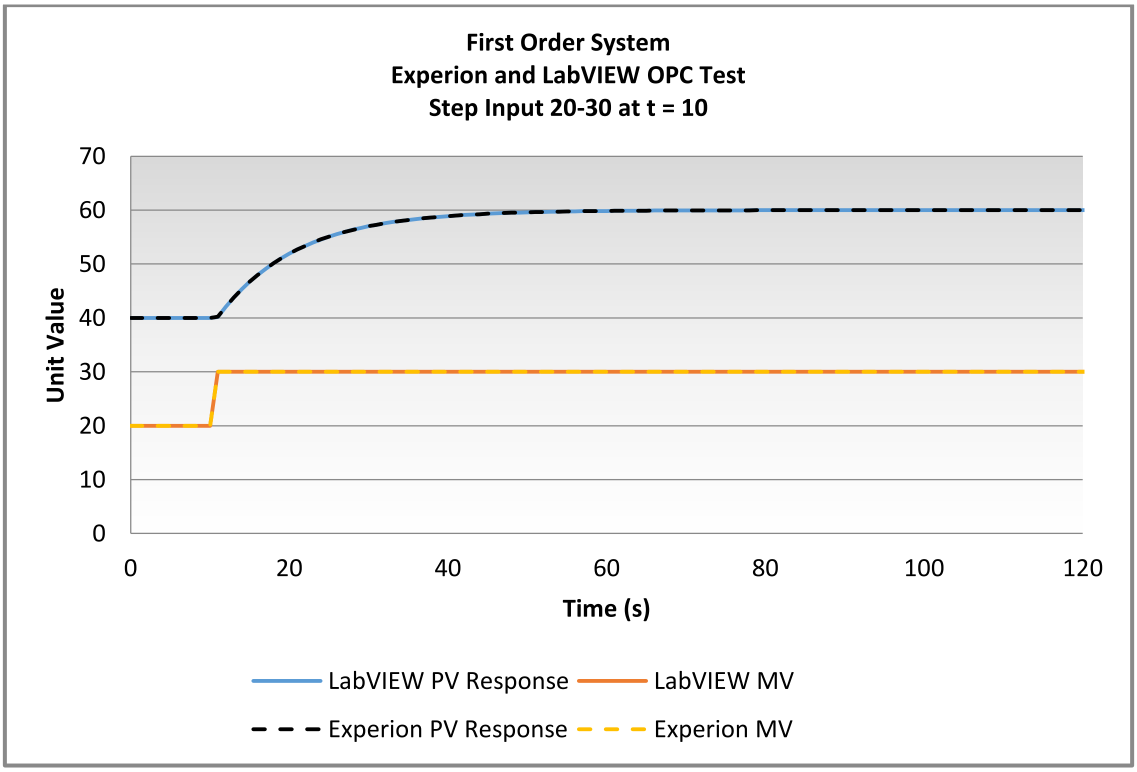


Figure 71: LabVIEW OPC Testing First Order System (Step input 20-30)

As anticipated, both responses recorded identical data values. Based on these findings, no additional step tests were recorded. The raw tabulated data values can be viewed in Appendix N.

With the successful testing of basic read and write functions using LabVIEW, implementation of a controller on the Experion simulated first order system, using LabVIEW, commenced. See Figure 72.

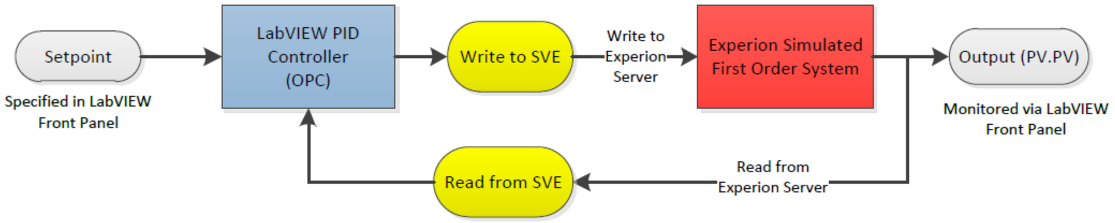

Setpoint
Specified in LabVIEW Front Panel
LabVIEW PID Controller (OPC)
Write to SVE
Write to Experion Server
Experion Simulated First Order System
Output (PV.PV)
Monitored via LabVIEW Front Panel
Read from SVE
Read from Experion Server

Figure 72: LabVIEW / Experion OPC First Order System with PID - Data Exchange Block Diagram

For the purpose of testing, LabVIEW's feedback control PID block was configured in the VI block diagram as indicated in Figure 73.

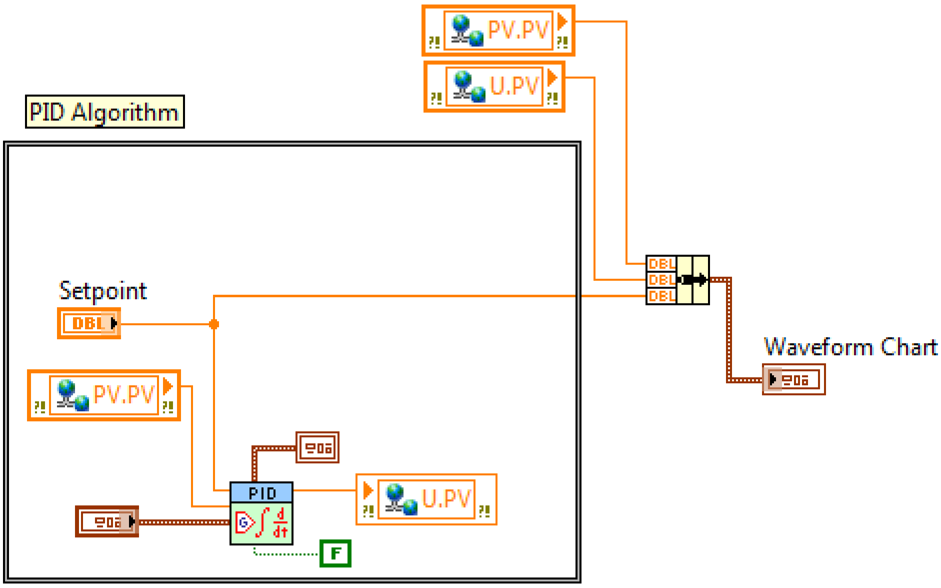


Figure 73: LabVIEW PID Block Implementation

Entering the configuration specified in Figure 73 into a time loop, allowed for the real-time simulation of the LabVIEW controlled Experion first order process. A screenshot of the complete block diagram is presented in Figure 74.

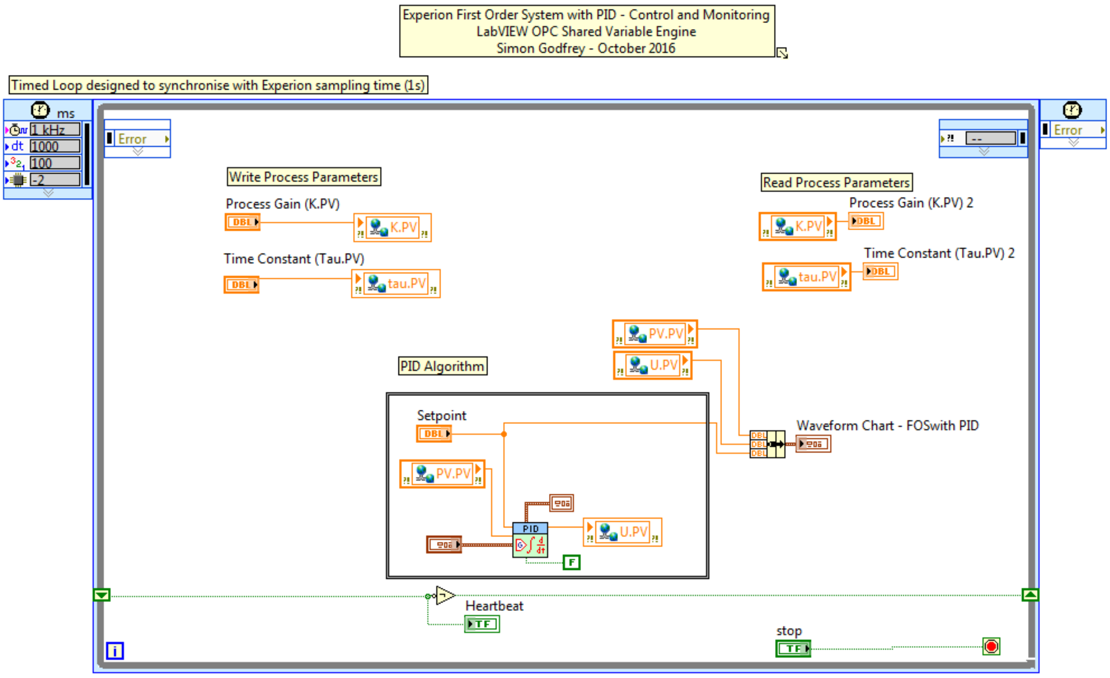

Experion First Order System with PID - Control and Monitoring
LabVIEW OPC Shared Variable Engine
Simon Godfrey - October 2016
Timed Loop designed to synchronise with Experion sampling time (1s)
ms
1 kHz
dt
1000
100
-2
Error
Write Process Parameters
Process Gain (K.PV)
K.PV
Time Constant (Tau.PV)
tau.PV
Read Process Parameters
Process Gain (K.PV) 2
Time Constant (Tau.PV) 2
PV.PV
U.PV
PID Algorithm
Setpoint
PID
Waveform Chart - FOSwith PID
Heartbeat
stop
DBL
TF
F

Figure 74: LabVIEW OPC First Order System with PID Block Diagram

For consistency, PID tuning parameters selected were the same as those used with the Simulink OPC controller. The algorithm for the calculation of the integral action differs slightly from Simulink, requiring the reciprocal of the integral term (τ_i) for implementation in LabVIEW (Equation 15 and Equation 16). A simple calculation was needed to ensure the same integral value was entered in LabVIEW's PID block. The controller tuning parameters are presented in Table 10.

Equation 15: LabVIEW Integral Action Algorithm (Discrete)

$$U_I(k) = U_I(k-1) + \frac{K_c}{\tau_i}\left(\frac{\varepsilon(k) + \varepsilon(k-1)}{2}\right)\Delta T$$

Equation 16: Simulink Integral Action Algorithm (Continuous)

$$C(s) = \frac{K_c \tau_i}{s}$$

Table 10: LabVIEW OPC PID Testing - Controller Tuning Parameters

	LabVIEW PID
Controller Gain	1
Integral Time	0.1666666¯
Derivative Time	0

With the newly configured controller parameters, it was possible to run the programmed VI to perform real-time simulations to validate OPC controller operation against data acquired from Simulink. Comparable with previous Simulink OPC tests, several setpoint change tests were conducted to ensure consistent results. The first test involved stepping the process setpoint from value 40 to 60, at time 20. As shown in Figure 75, the response of the OPC LabVIEW controller again closely resembles that of the ideal Simulink controller, indicating correct implementation in LabVIEW.

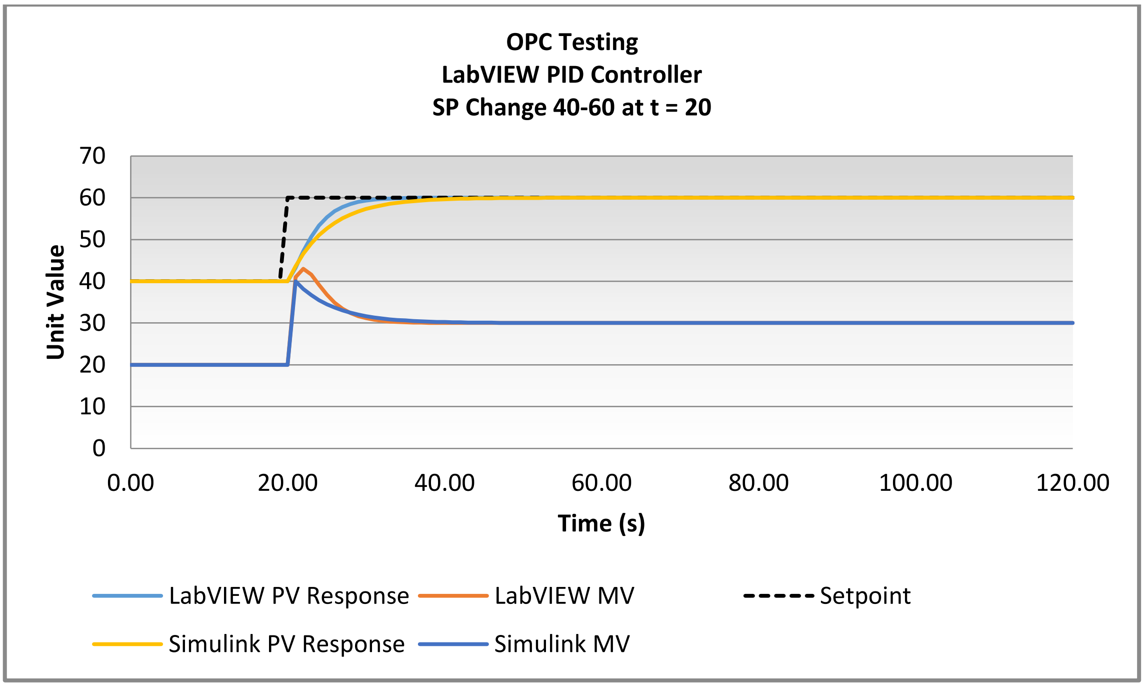


Figure 75: LabVIEW OPC Testing First Order System with PID

Several other tests were conducted which also yielded comparable results (refer to Appendix O). With successful testing of the implementation of a LabVIEW controlled Experion first order system, no further tests were conducted.

5.2 Phase 2 Concluding Remarks

The successful real-time control of the simulated system in Experion using MATLAB, Simulink and LabVIEW via OPC, provides an avenue for future research into the monitoring and control of Experion simulations via OPC compliant software. Based on the results presented, there is now a foundation upon which future students may extend the use of LabVIEW and Simulink, in addition to Microsoft Excel, for the implementation of control schemes on Murdoch University's pilot plant within the Instrumentation and Control major.

6 Delivery against Project Scope – Phase 3

As identified in the project scope, the primary objective of the project was the development of comprehensive learning materials for the configuration and operation of the Experion System.

6.1 Experion PKS Learning Material

After completion of Phase 1 and 2 of this project, previously developed learning materials were reviewed (refer to Appendix P for a list of prior authors). The existing tutorial materials lacked sufficient explanation, detail and were out-dated as a consequence of the Experion version upgrade. All existing materials were updated and re-written, with the addition of extensive detailed content.

New tutorials were also created to extend the learning opportunities as a consequence of the findings of Phases 1 and 2 of this project. Specifically new learning materials developed include:

- Mathematical Expression;
- Logic Expression;
- First Order System;
- First Order System with PID Control;
- Cascade Control Strategy;
- Model-Based Predictive Control Strategy;
- OPC - OPC Quick Client;
- OPC – MATLAB;
- OPC – Simulink; and
- OPC – LabVIEW.

As an example of the structure, content and format of the new learning materials, a small sample of 2 tutorials (HES-1006 – First Order System with PID and HES-5007 – OPC LabVIEW) is included in Appendix Q and Appendix R, respectively. The full set of learning materials is available on Murdoch's School of Engineering online directory (EngShared) at the following location: **\\mylab\files\engshared\Experion_Dev\Experion_Project**

Table 11 provides a summary of all learning materials produced.

Table 11: Learning Material Development Areas

Document Number	Title	Brief Summary
Initial Set up and Configuration		
HES-0001	System Login	- Provides procedure for the successful log in to the Experion Server
HES-0002	System Model (Server) Configuration	- Creation and configuration of the system model (servers)
HES-0003	Asset Model Configuration	- Creation and configuration of the asset model - Describes the use of assets and tag names
HES-0004	Historical Archive Configuration	- Configuration settings of historical archive
HES-0005	Sampling and Trend Configuration	- Configuration setting for setting sampling rates and trend settings
Controllers and Control Modules		
HES-1001	Simulated C300 Controller Creation	- Creation and configuration of a simulated C300 process controller - Assigning of parent asset to controller - How to load the controller to the server for simulation - How to toggle controller status.
HES-1002	Control Module Creation	- Creation and configuration of continuous control modules - Assigning of parent asset to CM - Loading CM to controller - How to perform status check on module.
Sequential Control Modules		
HES-3001	Sequential Control Module Creation	- Overview of blocks used within SCM - Creation and configuration of SCM - Assigning of parent asset to SCM

Human Machine Interface		
HES-2001	Basic Shapes	- Development of display for basic shapes - Creation of tank shape - Creation of an indicator - Creation of push button - Linking data points to shapes
HES-2002	Shape Sequences	- Creation of shape sequences
HES-2003	Dynamic Shapes	- Creation and configuration of dynamic shapes
HES-2004	Trends	- Configuration of trends - Linking data points to trends
HES-2005	Faceplates and Popups	- Configuration of popups and faceplates - Linking data points to faceplates and popups
Station		
HES-4001	Overview	- Configuration settings for Station - How to access developed displays
Microsoft Excel Data Exchange (MEDE)		
HES-6001	Configuration	- Configuring MEDE add-in to Microsoft Excel
HES-6002	Read and Write With MEDE	- Procedure for reading values from Experion server - Procedure for writing values to Experion server
Complete Simulated Examples		
HES-1003	Mathematical Expression	- Development of control module - Reading/writing values with MEDE - Development of HMI display - Testing and simulation tips
HES-1004	Logic Expression	- Development of control module - Reading/writing values with MEDE - Development of HMI display - Testing and simulation tips
HES-1005	First Order System	- Development of control module - Reading/writing values with MEDE - Development of HMI display - Testing and simulation tips
HES-1006	First Order System with PID Control	- Development of control module - Reading/writing values with MEDE - Development of HMI display - Testing and simulation tips
HES-1007	Cascade Control	- Simplified development of control module - Displayed HMI - Testing and simulation tips

HES-1008	Model-Based Control	- Simplified development of control module - Displayed HMI - Testing and simulation tips
OPC Communication		
HES-5001	OPC Introduction	- Introduction to OPC communication
HES-5002	OPC Experion Server Configuration	- Configuration of Experion server for OPC communication
HES-5003	OPC Quick Client	- Use of OPC Quick Client for testing Experion OPC server configuration - Read/write values using OPC Quick Client
HES-5005	OPC MATLAB Client	- OPC Configuration functions identified - Read/write values dynamically using MATLAB - MATLAB example script included
HES-5006	OPC Simulink Client	- OPC configuration settings identified - Read/write values dynamically using Simulink - Tips for implementing control strategies (including example)
HES-5007	OPC LabVIEW Client	- OPC configuration settings identified - Read / write values dynamically using LabVIEW - Tips for implementing control strategies (including example)

6.2 Experion Learning Material Testing

After documentation of learning materials was completed, it was considered essential to test the clarity and ease of understanding of the documents produced. As the project involved extensive time researching and working with Experion, it was not believed to be valid for the author to evaluate the quality and useability of the learning materials.

As the documentation is intended for use by students with no prior experience with Honeywell Experion PKS, the learning materials were tested by a fellow student who had no prior Experion experience (Mr Michael Crooke). He was requested to read the project learning materials and attempt to recreate two complete example programs (Boolean Logic and Mathematical Function).

Mr Crooke successfully created his own simulated C300 controller and associated asset, followed by the successful re-creation of each of the identified examples. Screenshots of the control modules and HMI's, produced by Mr Crooke, can be found in Appendix S.

Additionally, Mr Crooke provided a reflection on the learning experience as well as the learning materials produced, and indicated the following:

"Firstly, I would like to state I had no prior knowledge or experience with the Experion software before attempting these tutorials. While progressing my way through the beginner tutorials a vast range of information was presented. This information not only presented detailed steps on how to do all that was required, but also what this meant on a wider scale. Resulting in the fact not only was I making myself control modules but also learning about how they tied in within the software and how they worked.

As for the procedural steps; they were extremely detailed and concise. I would like to make a special mention about all images within the documents help dramatically, where no time was wasted searching for a particular icon/setting as it was clearly stated. Additionally, the organisation of all tutorial documents helped greatly with transitioning from one to another once completed, and lastly, when performing the later/more involved tutorials references to the previously completed documents were extremely useful. This helped tremendously as no wasted time was required searching around to find the forgotten or unknown details. Overall I must say these tutorials are presented and written in an extremely helpful and practical manner. "

6.3 Phase 3 Concluding Remarks

The learning materials produced at the completion of phase 1 and 2 of the project provide a tool for the development of technical knowledge of distributed control systems. Based on the successful testing of these materials, students will be able to configure the Experion system quickly and efficiently, before developing their own control simulations for testing. At the conclusion of learning material development, an intuitive folder structure was developed to assist students in progressing through tutorials in an appropriate manner. Additionally, each of the tutorials identifies hyperlinked additional resources for further information, as well as indicating a set of suggested pre-requisite documents for completion prior to the completion of the present document.

7 Proposed Future Work

Throughout this project, several areas requiring additional research were recognised. These areas of proposed future research are discussed below.

7.1 Sequential Control Modules

It was originally intended to produce several complete example programs and associated learning materials for the configuration of SCMs. However, due to the problems experienced with the Experion server in early weeks, the time allocated in the project schedule to SCM research had to be removed from the project scope. For this reason, only the basic configuration of SCMs and their respective function blocks were researched and documented, leaving room for further SCM program development.

It is suggested, for the next stage of research into SCMs, to apply a similar methodology to that adopted in this report (Research, Develop, Simulate, Test and Validate).

7.2 OPC Data Point Discrepancies

Additional testing of OPC controlled systems is recommended. As indicated in the preceding sections of this report, data discrepancies between ideal controllers implemented in Simulink, and controllers implemented using third party control software was present. This was assumed to be a result of the Euler approximation of the first order system tested, however further research and testing is required to confirm the cause of the discrepancies.

Another potential cause of these discrepancies that should be investigated is the possibility of inconsistent synchronisation of the third party applications and the Experion server. As an example, the execution of the LabVIEW or Simulink real-time simulations may not be precisely every one second. Further testing is required to ensure sampling synchronisation is not the cause of the data irregularities.

7.3 Advanced Control Scheme Continuation

Within this report, only a sample of the possible control strategies were implemented and validated. It is recommended to continue exploring advanced controller strategies such as feedforward control and augmented feedforward feedback control. It is suggested to adopt the same approach as presented in this report, including extensive testing and validation prior to documentation of the strategy.

Furthermore, it is recommended to expand upon the research and testing performed on Honeywell's model-based predictive controller (Profit Loop). One potential area of research involves testing the performance of the controller for a variety of complex systems, such as systems with inverse response and those with significant time delays.

Additionally, further research into the identification of the process model using the Profit Loop Assistant is required. For the purpose of this project, only direct entry was examined, it is suggested to research and test identifying the process model by way of PID tuning parameters and step testing.

7.4 Pilot Plant Modelling

The successful implementation of control strategies on the simulated Experion system provides a path for future students to simulate newly developed control algorithms. For the purpose of this report, only simple linear first order systems were developed for simulation purposes. If the University pilot plant were to be fully modelled into Experion, students could perform simulations of their control algorithms prior to implementation on the physical plant. The successful implementation of a complete simulated pilot plant would provide a number of benefits to the University, some of which include:

- Providing students with an additional resource for testing and implementation of control strategies;

- Further experience using an industry standard control system application;
- A potential reduction in the replacement/refurbishment of pilot plant instruments and hardware.

It is recognised that the modelling of the complete pilot plant is an extensive exercise, and as Experion is intended for use as a DCS, rather than a complex process simulator, Experion may not be the optimal program for use. However, Murdoch University has access to two process simulation software programs which have the potential to be configured to the Experion system for control applications; Aspen HYSYS and Honeywell's UniSim.

Aspen HYSYS

Aspen HYSYS is a simulation application used for the design, optimisation and simulation of industrial processes (Aspen Technology, Inc. 2016). Aspen Operator Training Simulator (OTS) Framework offers a single integrated platform for the simulation of developed processes. A key feature of the OTS Framework is the identification of OPC communication as a primary means for data transfer by OTS applications (Aspen Technology, Inc. 2008). Further research is required to confirm the possibility of OPC communication between Honeywell Experion and Aspen HYSYS.

Honeywell UniSim

An alternative option for pilot plant modelling is the use of Honeywell's UniSim software. UniSim is Honeywell's modelling software for the creation and simulation of steady-state and dynamic process models (Honeywell 2016e). As UniSim is part of Honeywell's process control umbrella, it is expected that the application interoperability may be achieved more efficiently than with the use of a third party application (i.e. Aspen HYSYS).

By creating the system model in either of these programs, Experion can be used as the controller for the simulated process, or as the OPC server for the implementation of control strategies using third party software applications, such as Simulink and LabVIEW.

7.5 Process Control Software for Physical Pilot Plant Control

The successful implementation of basic control strategies via OPC presented in this report provides a possible avenue for further research. With additional research and testing, it is believed the approach adopted for controlling the simulated processes via OPC could be implemented for controlling Murdoch physical pilot plant. Both Simulink and LabVIEW are believed to be worthwhile alternatives to Microsoft Excel for the implementation of real-time control strategies in the pilot plant.

To achieve control of the pilot plant with these applications, first an OPC server must be configured on the pilot plant server, allowing for the exchange of data points required for the effective monitoring and control of the plant. Then, depending on the application selected for control, the configuration steps identified in this report, or appropriate learning materials, can be followed to achieve control.

8 Summary and Concluding Remarks

At the conclusion of this project, three key achievements were identified including the successful implementation, simulation and validation of several programs and control strategies, real-time communication with third party process control applications and the creation of an extensive set of learning materials.

A total of six simulation programs were developed and tested, including three control strategy implementations; feedback PID, cascade and model-based predictive control. The validation of these control arrangements provides an avenue for future students to re-create these strategies in an attempt to gain technical knowledge and experience with the implementation of simulated control schemes in a real world industrial plant.

The successful configuration and testing of real-time data exchange with a variety of different third party applications, provides the foundation for future research into the capabilities of the interoperability between Honeywell Experion and process control software applications to be performed.

Finally, an extensive set of learning materials to facilitate the teaching of the new unit to be introduced at Murdoch University in 2017 has been produced. These materials will assist staff and students in gaining a comprehensive understanding of the role a distributed control system in an industrial setting, providing the information necessary to configure, simulate and validate control strategies developed within the Honeywell Experion package.

9 References

Aspen Technology, Inc. 2016. *Aspen HYSYS for Hydrocarbons.* http://www.aspentech.com/products/aspen-hysys/ (accessed October 30, 2016).

Aspen Technology, Inc. 2008. *Aspen OTS Framework.* Burlington.

Dukkipati, Rao V. 2008. *MATLAB Basics: MATLAB: An Introduction with Applications*, by Rao V. Dukkipati, 1-2. New Delhi: New Age International.

Gumireddy, Kiranmai. 2013. *Honeywell Experion System For Teaching Purposes.* Thesis, Perth: Murdoch University.

Honeywell. 2013. *Experion Control Builder Components Theory.* Release 430. Honeywell International. www.honeywell.com

Honeywell. 2014a. *Experion PKS Configuration Studio Overview Guide.* Release 430. Honeywell International. www.honeywell.com

Honeywell. 2014b. *Experion PKS Control Building User's Guide.* Release 430. Honeywell International. www.honeywell.com

Honeywell. 2014c. *Experion PKS Enterprise Model Builder User's Guide.* Release 430. Honeywell International. www.honeywell.com

Honeywell. 2014d. *Experion PKS HMIWeb Display Building Guide.* Release 430. Honeywell International. www.honeywell.com

Honeywell. 2014e. *Experion PKS Overview.* Release 430. Honeywell International. www.honeywell.com

Honeywell. 2016a. *C300 Controller.* https://www.honeywellprocess.com/en-US/explore/products/control-monitoring-and-safety-systems/integrated-control-and-safety-systems/experion-pks/controllers/Pages/c300.aspx (accessed November 13, 2016).

Honeywell. 2016b. *Experion PKS - OPC Technology.* https://www.honeywellprocess.com/en-US/explore/products/control-monitoring-and-safety-systems/integrated-control-and-safety-systems/experion-pks/Pages/opc-technology.aspx (accessed August 2, 2016).

Honeywell. 2016c. *Experion PKS.* https://www.honeywellprocess.com/en-US/explore/products/control-monitoring-and-safety-systems/integrated-control-and-safety-systems/experion-pks/Pages/default.aspx (accessed August 2, 2016).

Honeywell. 2016d. *Profit Loop from Honeywell - Effective Loop Tuning and Control.* https://www.honeywellprocess.com/en-US/explore/products/control-monitoring-and-safety-systems/integrated-control-and-safety-systems/experion-pks/controllers/Pages/profit-loop.aspx (accessed October 26, 2016).

Honeywell. 2016e. *UniSim Competency Suite.* https://www.honeywellprocess.com/en-US/explore/products/advanced-applications/unisim/unisim-competency-suite/Pages/default.aspx (accessed October 30, 2016).

Kepware, Inc. 2015. *OPC Quick Client.*

MathWorks. 2016a. *OPC Toolbox.* https://au.mathworks.com/products/opc/ (accessed October 27, 2016).

MathWorks. 2016b. *Simulink - Simulation and Model-Based Design.* https://au.mathworks.com/products/simulink/ (accessed October 27, 2016).

MathWorks. 2016c. *addgroup.* https://au.mathworks.com/help/opc/ug/addgroup.html (accessed November 10, 2016).

MathWorks. 2016d. *additem.* https://au.mathworks.com/help/opc/ug/additem.html (accessed November 10, 2016).

MathWorks. 2016e. *connect.* https://au.mathworks.com/help/opc/ug/connect.html (accessed November 10, 2016).

MathWorks. 2016f. *opcda.* https://au.mathworks.com/help/opc/ug/opcda.html?s_tid=srchtitle (accessed November 10, 2016).

MathWorks. 2016g. *read.* https://au.mathworks.com/help/opc/ug/read.html (accessed November 10, 2016).

MathWorks. 2016h. *write.* https://au.mathworks.com/help/opc/ug/write.html (accessed November 10, 2016).

Meiri, Timothy. *Experion Simulation and Pilot Plant Maintenance.* Thesis, Perth: Murdoch University, 2015.

Microsoft. 2016. *An introduction to optimization with the Excel Solver tool.* https://support.office.com/en-US/article/An-introduction-to-optimization-with-the-Excel-Solver-tool-1F178A70-8E8D-41C8-8A16-44A97CE99F60 (accessed November 3, 2016).

Miller, Steven J. n.p. *The Method of Least Squares.* Providence, Rhode Island: Brown University,

Nain, Nurazlina Binte Mohamad. 2013. *Honeywell Experion Process Knowledge System as a teaching guide.* Thesis, Perth: Murdoch University

National Instruments. 2016a. *LabVIEW Datalogging and Supervisory Control Module.* http://sine.ni.com/nips/cds/view/p/lang/en/nid/209851 (accessed November 12, 2016).

National Instruments. 2016b. *LabVIEW Environment Basics.* http://www.ni.com/getting-started/labview-basics/environment#frontPanel (accessed November 12, 2016).

National Instruments. 2016c. *LabVIEW System Design Software.* http://www.ni.com/labview/ (accessed October 4, 2016).

Ogunnaike, Babatunde A, and Harmon Ray W. 1994. *Process Dynamics, Modeling, and Control*, 143. New York: Oxford University Press,

Schaschke, Carl. 2014. *A Dictionary of Chemical Engineering.* Oxford: Oxford University Press,

Seborg, Dale E, Thomas F Edgar, and Duncan A Mellichamp. 2004. *Process Dynamics and Control*, 534-537. Hoboken, New Jersey: John Wiley & Sones

10 Appendices

10.1 Appendix A: Phase 1 Constraint – Solution to Technical Issues with Experion Server

Identified Cause:

Bad password for the manager account (MNGR) embedded within the Experion CDA-SP Service. This meant the service was not running on the simulation server (PROFIT-SVR). This was a result of a password misconfiguration during the installation of the simulation server.

The problem that proceeded was that it was not known that the controller simulations on the Experion server required a separate simulation server (a change from the configuration in previous release RS311). Without a separate accessible simulation server, simulations are not able to be performed.

Solution Implemented:

1. Created and configured a separate simulation server (PROFIT-SVR)

2. Re-configured Experion with correct password.

Contact:

Mr Will Stirling

Technical Officer, Murdoch University

10.2 Appendix B: Phase 2 Constraint – Solution to OPC Data Access Restrictions

Identified Cause:

The Experion OPC service uses Distributed Component Object Model (DCOM) to control access to the service. Therefore, the accessed needed to be open to all users (using the **Everyone** security group). This was not the default setting as installed during the Experion installation process as it can pose a greater security risk. The greater security risk is not a significant issue at Murdoch University as there would be minimal impact. However, there is a different evaluation for a real world plant system.

Solution Implemented:

Allow access to all users (**Everyone**).

Actual implementation on the Experion PKS OPC Server DCOM Properties is:

Everyone is granted:

- Allow launch and activation permission both local and remote.
- Allow access permission both local and remote.

Local Engineers group on Experion2 is granted:

- Configuration permission as Full Control and Read (left as set by the Honeywell Experion Installer

Contact:

Mr Will Stirling

Technical Officer, Murdoch University

10.3 Appendix C: Euler Approximation for First Order System

Starting with the equation:

$$Equation\ 1: \quad Transfer\ Function: \frac{K}{\tau s + 1}$$

Which in differential equation form and applying an input ‘U’ yields:

$$Equation\ 2: \quad \tau \frac{dy}{dt} + y = Ku(t)$$

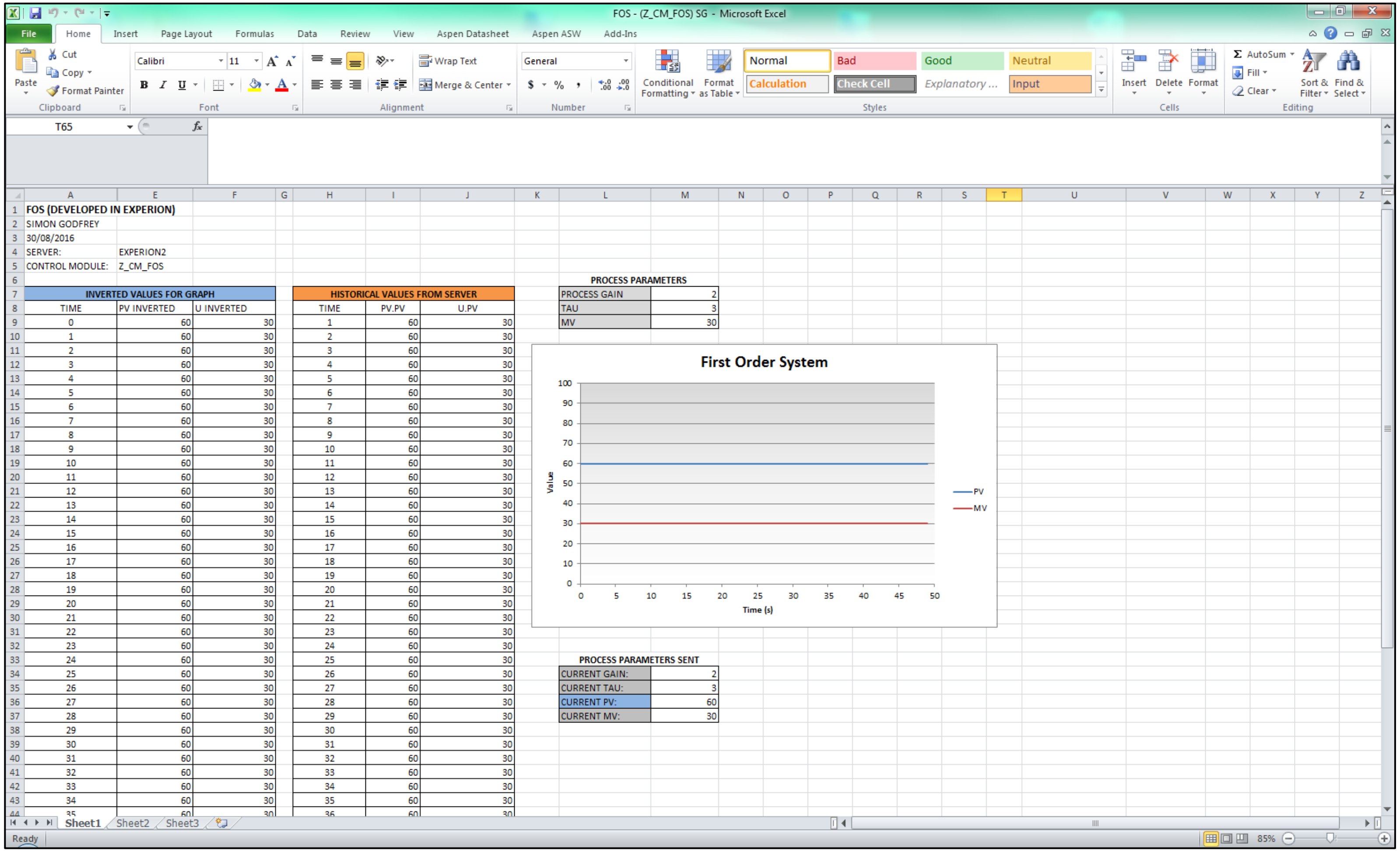
Applying Euler Approximations:

$$Equation\ 3: \quad \tau \left(\frac{y(k) - y(k-1)}{\Delta t} \right) + y(k) = Ku(k)$$

Rearranging for y(k):

$$Equation\ 4: \quad y(k) = \frac{K\, \Delta t\, u(k) + \tau\, y(k-1)}{\tau + \ \Delta t}$$

10.4 Appendix D: First Order System Monitoring and Control via Microsoft Excel Data Exchange (MEDE)



FOS (DEVELOPED IN EXPERION)
SIMON GODFREY
30/08/2016
SERVER: EXPERION2
CONTROL MODULE: Z_CM_FOS

INVERTED VALUES FOR GRAPH		
TIME	PV INVERTED	U INVERTED
0	60	30
1	60	30
2	60	30
3	60	30
4	60	30
5	60	30
6	60	30
7	60	30
8	60	30
9	60	30
10	60	30
11	60	30
12	60	30
13	60	30
14	60	30
15	60	30
16	60	30
17	60	30
18	60	30
19	60	30
20	60	30
21	60	30
22	60	30
23	60	30
24	60	30
25	60	30
26	60	30
27	60	30
28	60	30
29	60	30
30	60	30
31	60	30
32	60	30
33	60	30
34	60	30
35	60	30

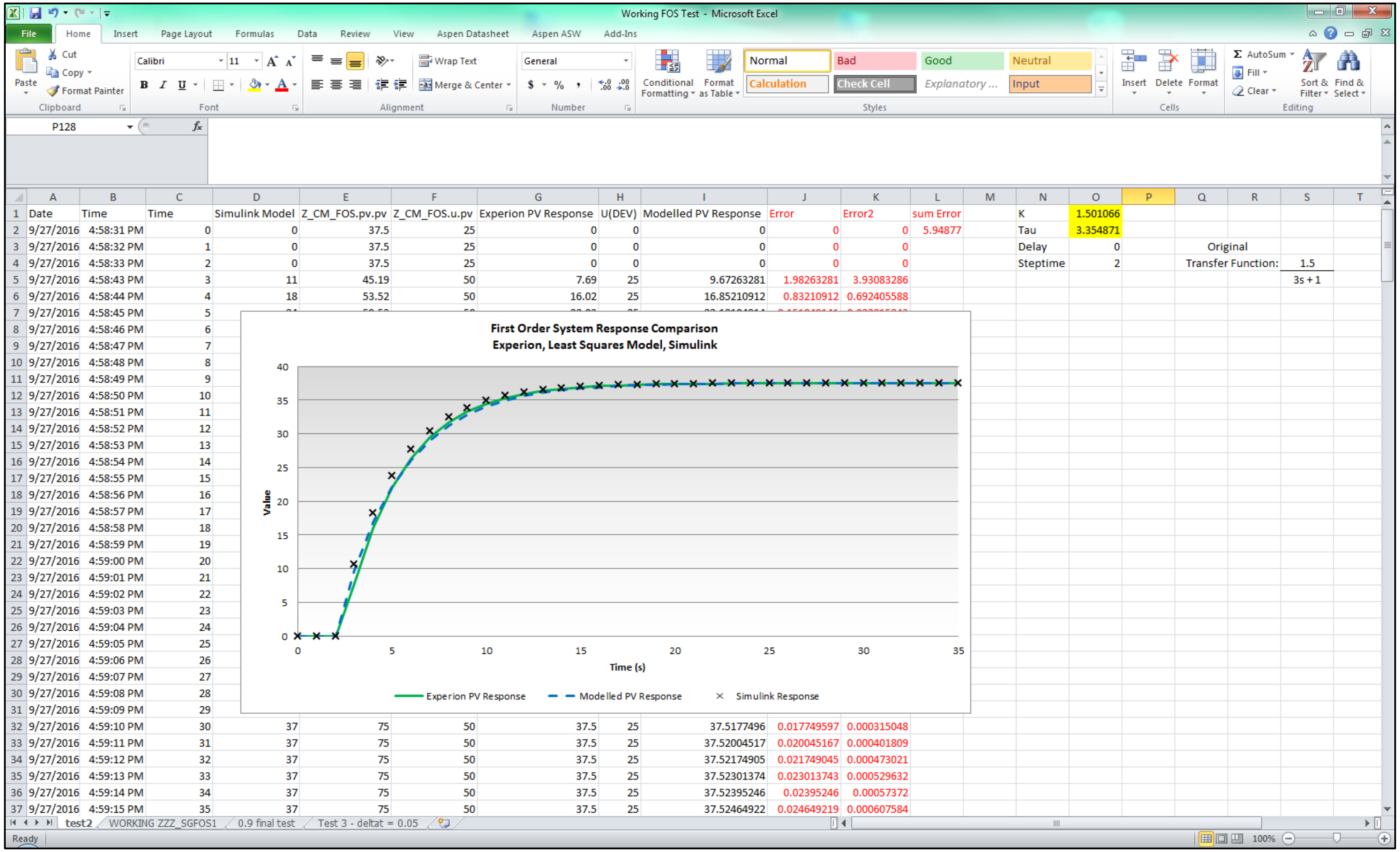
HISTORICAL VALUES FROM SERVER		
TIME	PV.PV	U.PV
1	60	30
2	60	30
3	60	30
4	60	30
5	60	30
6	60	30
7	60	30
8	60	30
9	60	30
10	60	30
11	60	30
12	60	30
13	60	30
14	60	30
15	60	30
16	60	30
17	60	30
18	60	30
19	60	30
20	60	30
21	60	30
22	60	30
23	60	30
24	60	30
25	60	30
26	60	30
27	60	30
28	60	30
29	60	30
30	60	30
31	60	30
32	60	30
33	60	30
34	60	30
35	60	30
36	60	30

PROCESS PARAMETERS	
PROCESS GAIN	2
TAU	3
MV	30

PROCESS PARAMETERS SENT	
CURRENT GAIN:	2
CURRENT TAU:	3
CURRENT PV:	60
CURRENT MV:	30

Figure 76: First Order System Monitoring and Control with MEDE

10.5 Appendix E: First Order System Least Squares Regression Analysis Screenshot



	A	B	C	D	E	F	G	H	I	J	K	L
1	Date	Time	Time	Simulink Model	Z_CM_FOS.pv.pv	Z_CM_FOS.u.pv	Experion PV Response	U(DEV)	Modelled PV Response	Error	Error2	sum Error
2	9/27/2016	4:58:31 PM	0	0	37.5	25	0	0	0	0	0	5.94877
3	9/27/2016	4:58:32 PM	1	0	37.5	25	0	0	0	0	0	
4	9/27/2016	4:58:33 PM	2	0	37.5	25	0	0	0	0	0	
5	9/27/2016	4:58:43 PM	3	11	45.19	50	7.69	25	9.67263281	1.98263281	3.93083286	
6	9/27/2016	4:58:44 PM	4	18	53.52	50	16.02	25	16.85210912	0.83210912	0.692405588	
7	9/27/2016	4:58:45 PM	5									
8	9/27/2016	4:58:46 PM	6									
9	9/27/2016	4:58:47 PM	7									
10	9/27/2016	4:58:48 PM	8									
11	9/27/2016	4:58:49 PM	9									
12	9/27/2016	4:58:50 PM	10									
13	9/27/2016	4:58:51 PM	11									
14	9/27/2016	4:58:52 PM	12									
15	9/27/2016	4:58:53 PM	13									
16	9/27/2016	4:58:54 PM	14									
17	9/27/2016	4:58:55 PM	15									
18	9/27/2016	4:58:56 PM	16									
19	9/27/2016	4:58:57 PM	17									
20	9/27/2016	4:58:58 PM	18									
21	9/27/2016	4:58:59 PM	19									
22	9/27/2016	4:59:00 PM	20									
23	9/27/2016	4:59:01 PM	21									
24	9/27/2016	4:59:02 PM	22									
25	9/27/2016	4:59:03 PM	23									
26	9/27/2016	4:59:04 PM	24									
27	9/27/2016	4:59:05 PM	25									
28	9/27/2016	4:59:06 PM	26									
29	9/27/2016	4:59:07 PM	27									
30	9/27/2016	4:59:08 PM	28									
31	9/27/2016	4:59:09 PM	29									
32	9/27/2016	4:59:10 PM	30	37	75	50	37.5	25	37.5177496	0.017749597	0.000315048	
33	9/27/2016	4:59:11 PM	31	37	75	50	37.5	25	37.52004517	0.020045167	0.000401809	
34	9/27/2016	4:59:12 PM	32	37	75	50	37.5	25	37.52174905	0.021749045	0.000473021	
35	9/27/2016	4:59:13 PM	33	37	75	50	37.5	25	37.52301374	0.023013743	0.000529632	
36	9/27/2016	4:59:14 PM	34	37	75	50	37.5	25	37.52395246	0.02395246	0.00057372	
37	9/27/2016	4:59:15 PM	35	37	75	50	37.5	25	37.52464922	0.024649219	0.000607584	

	N	O	P	Q	R	S
1	K	1.501066				
2	Tau	3.354871				
3	Delay	0		Original		
4	Steptime	2		Transfer Function:		1.5
5						3s + 1

Figure 77: First Order System Least Squares Regression Analysis Screenshot

10.6 Appendix F: First Order System with PID Control Testing

System 1

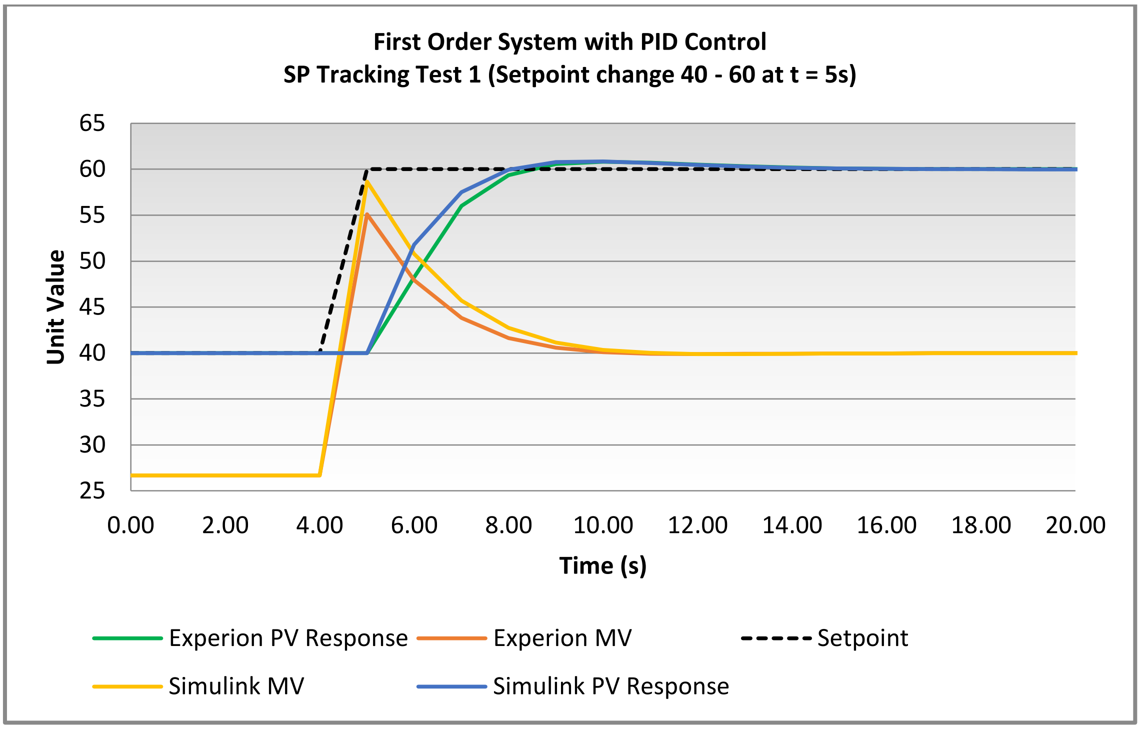


Figure 78: System 1 - First Order System with PID - Test 1

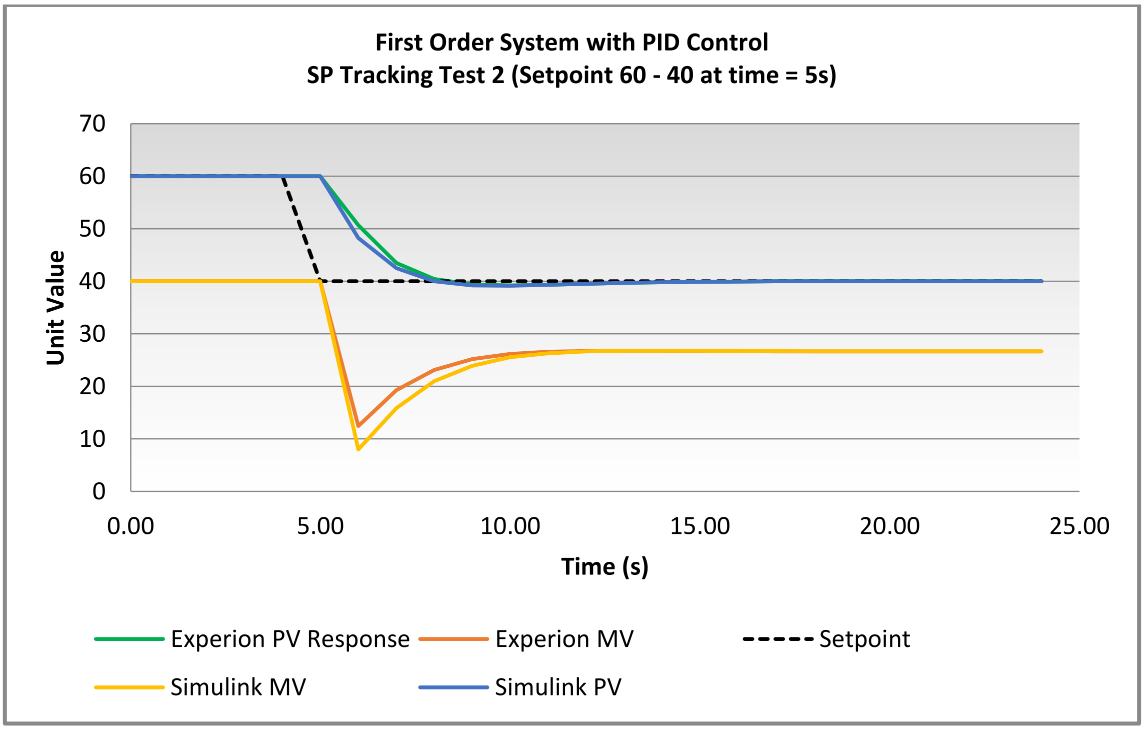


Figure 79: System 1 - First Order System with PID - Test 2

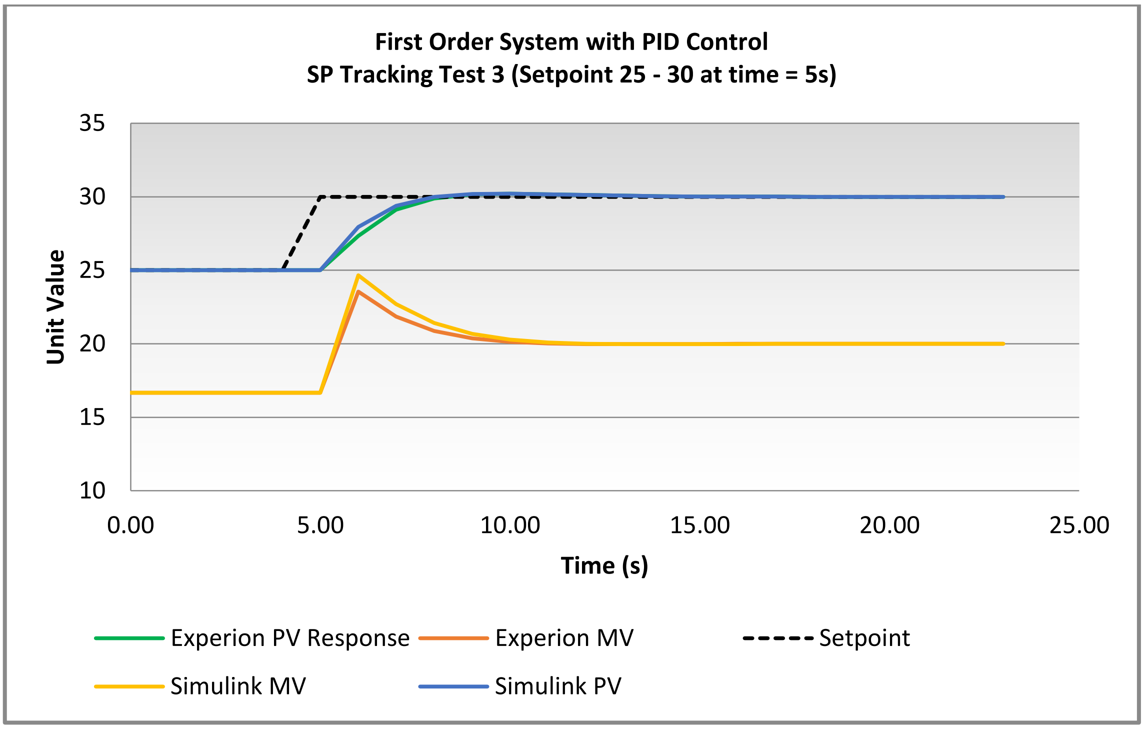


Figure 80: System 1 - First Order System with PID - Test 3

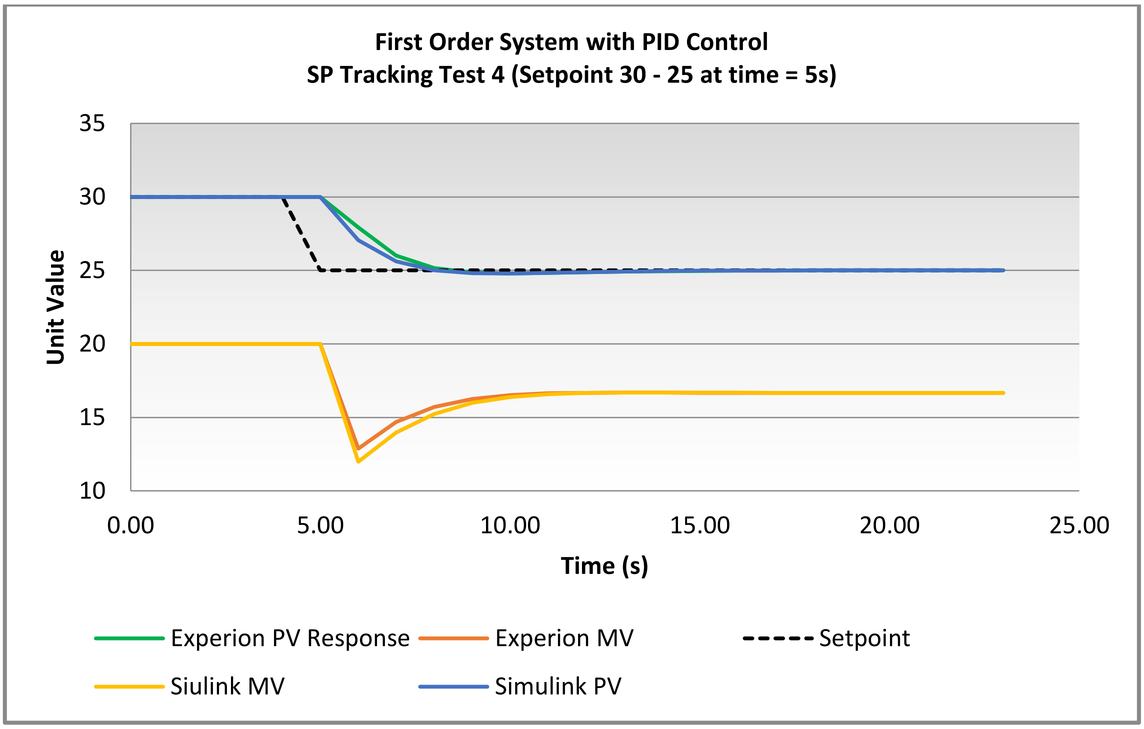


Figure 81: System 1 - First Order System with PID - Test 4

System 2

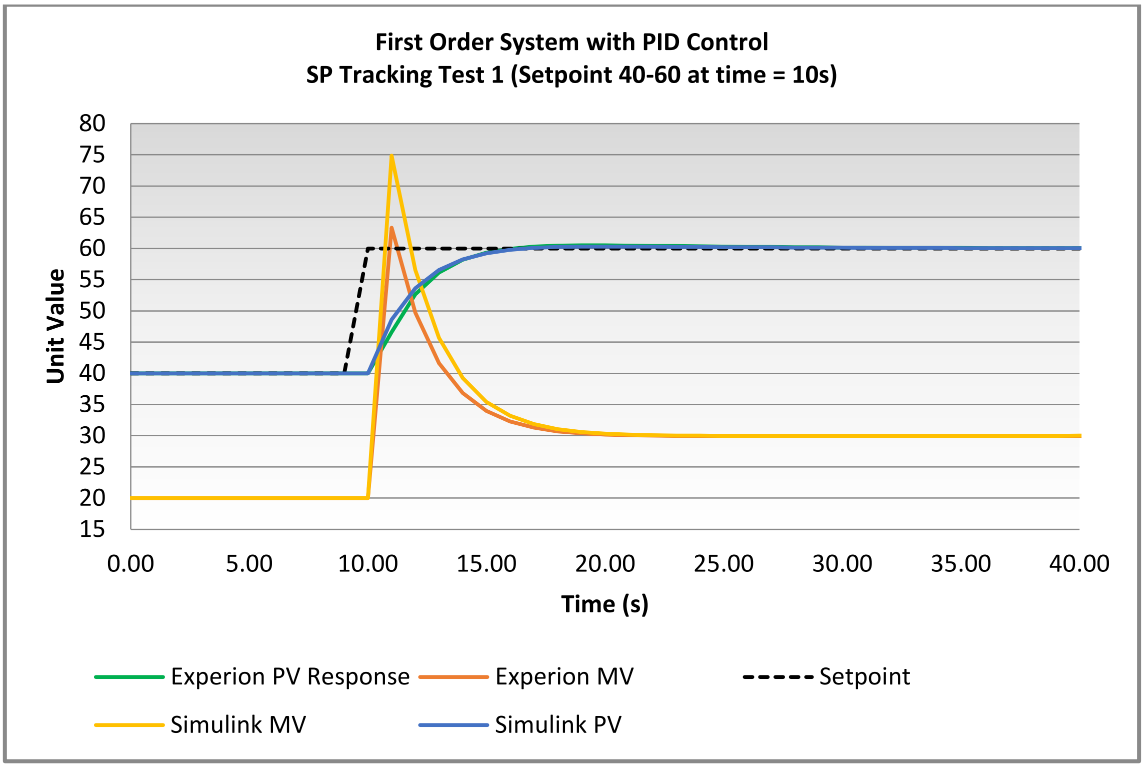


Figure 82: System 2 - First Order System with PID - Test 1

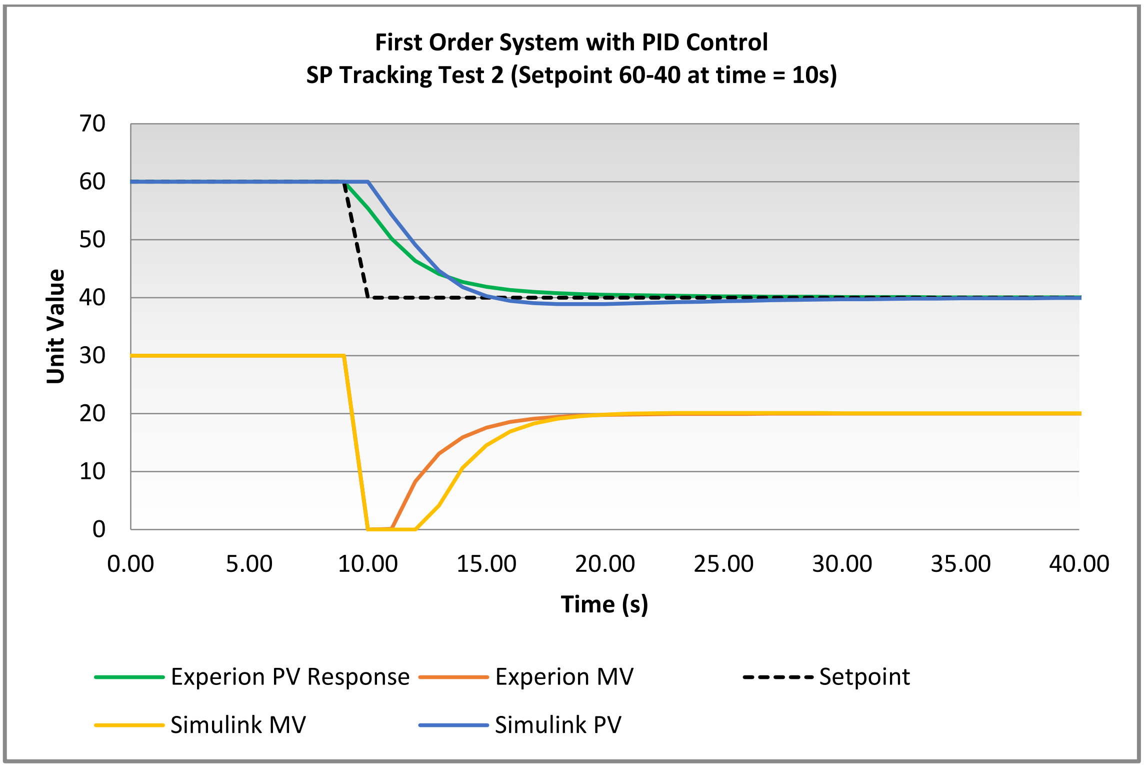


Figure 83: System 2 - First Order System with PID - Test 2

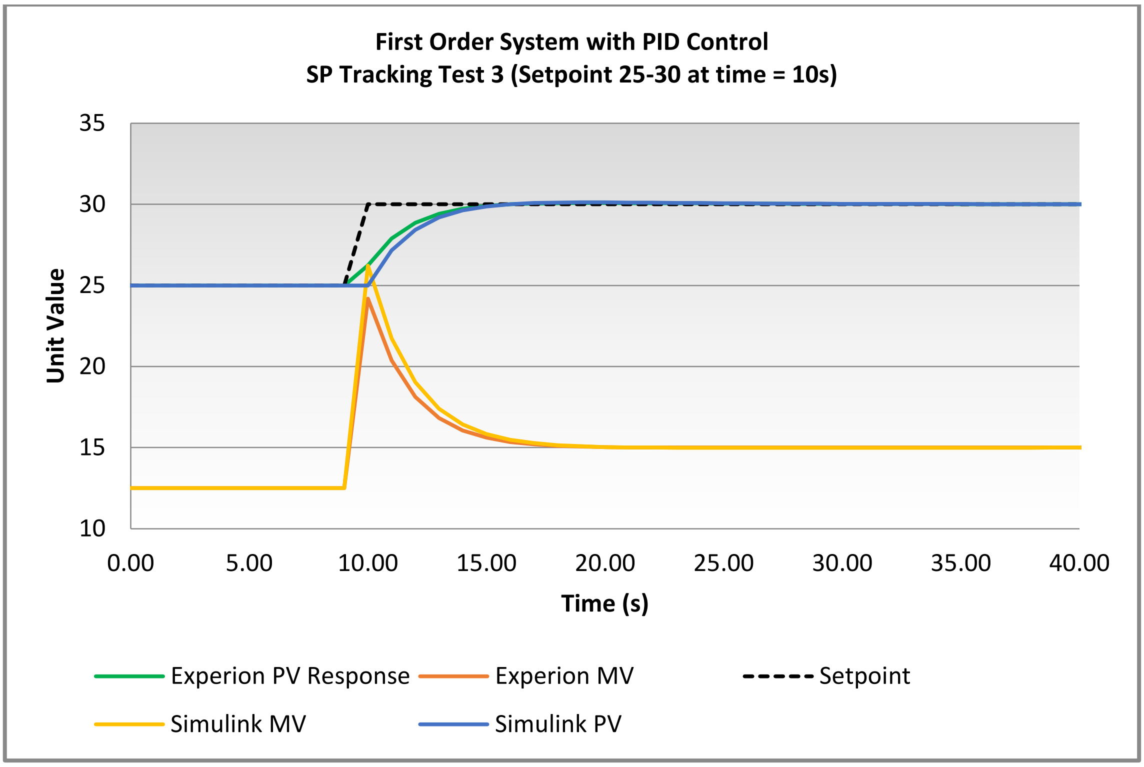


Figure 84: System 2 - First Order System with PID - Test 3

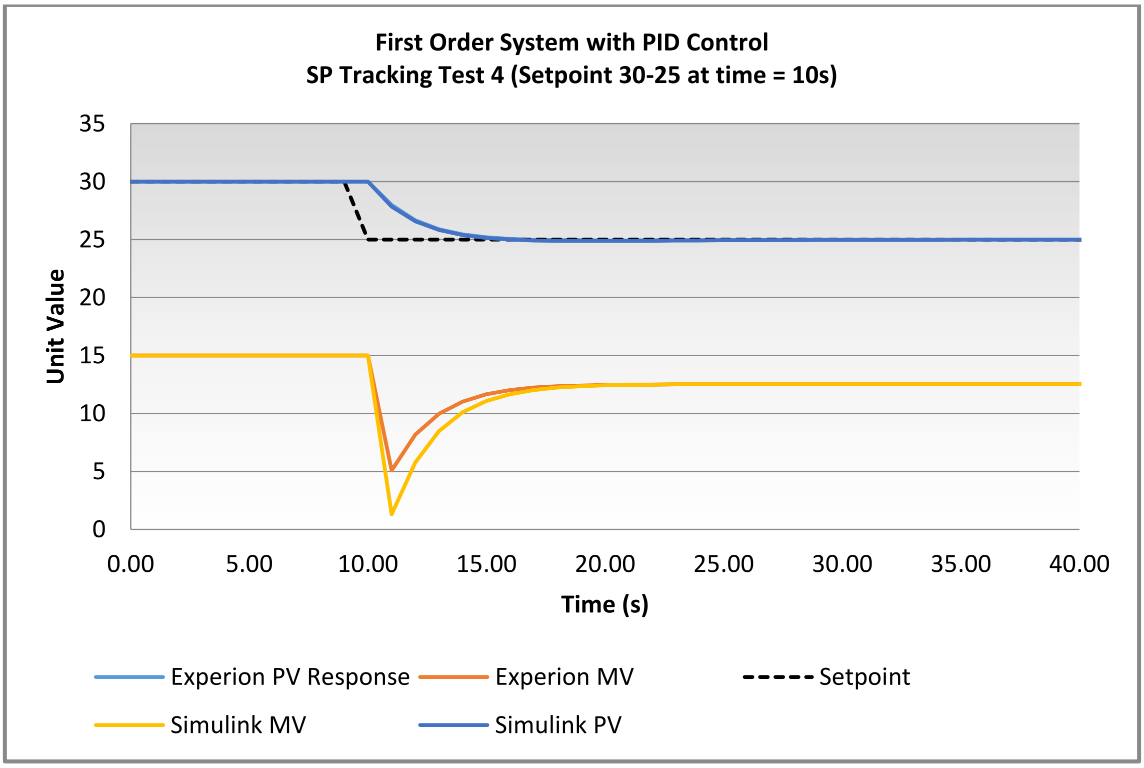


Figure 85: System 2 - First Order System with PID - Test 4

10.7 Appendix G: Cascade Control Strategy Disturbance Introduction Tests

Test 1: Setpoint = 75, Disturbance -8 at time = 5

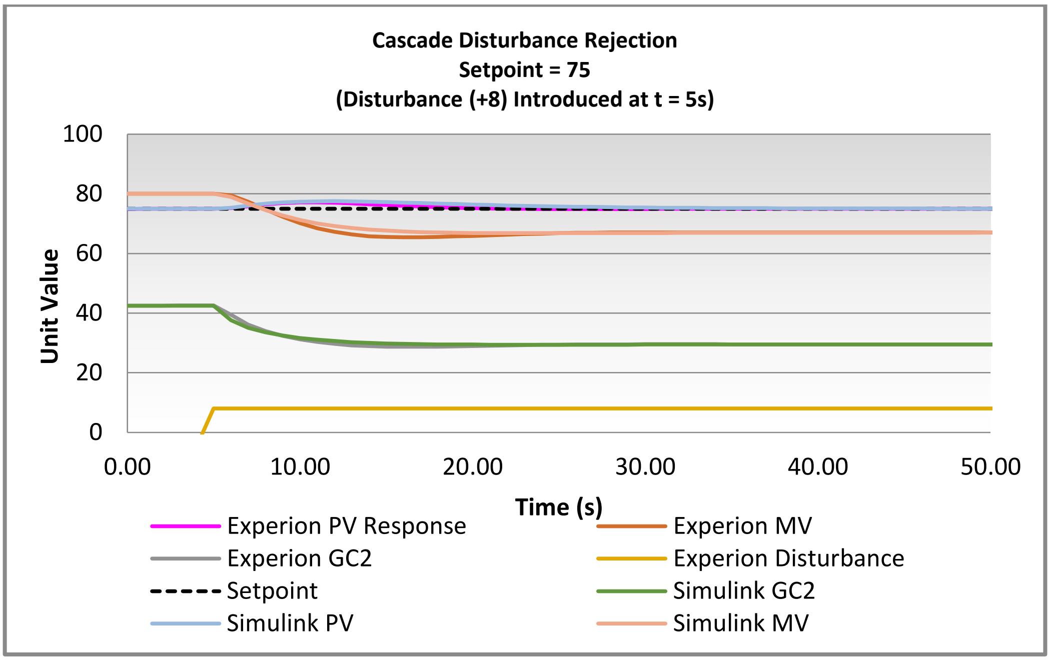


Figure 86: Cascade Control Response Comparison Test 1 (Disturbance +8 at Time =5s)

Test 2: Setpoint = 75, Disturbance -8 at time = 5

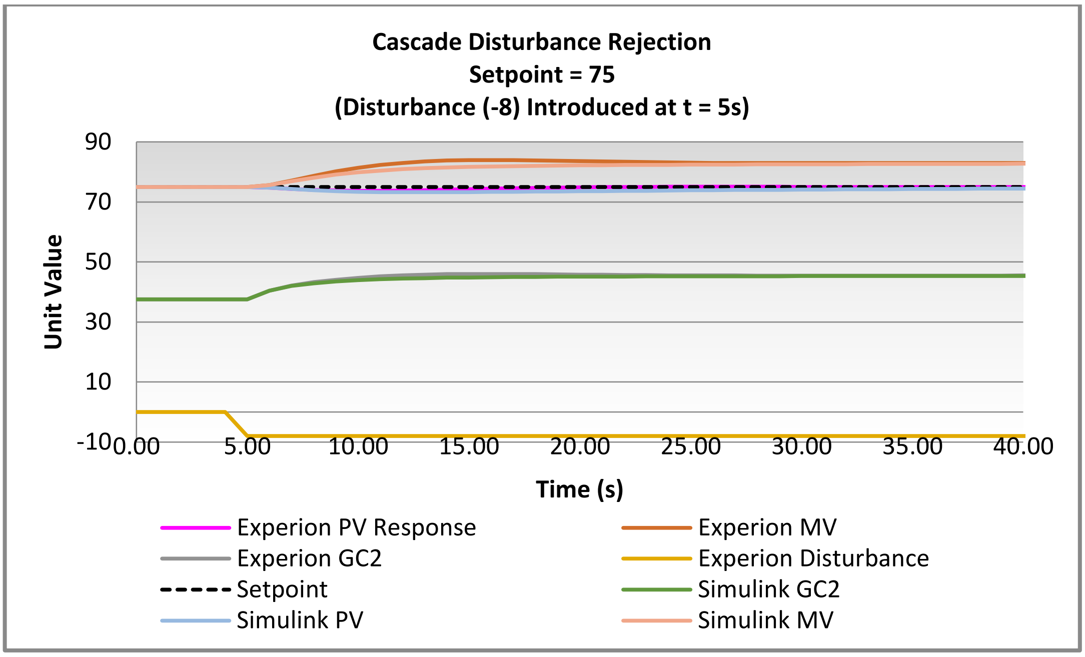


Figure 87: Cascade Control Response Comparison Test 2 (Disturbance -8 at Time =5s)

Test 3: Setpoint = 75, Disturbance +5 at time = 5

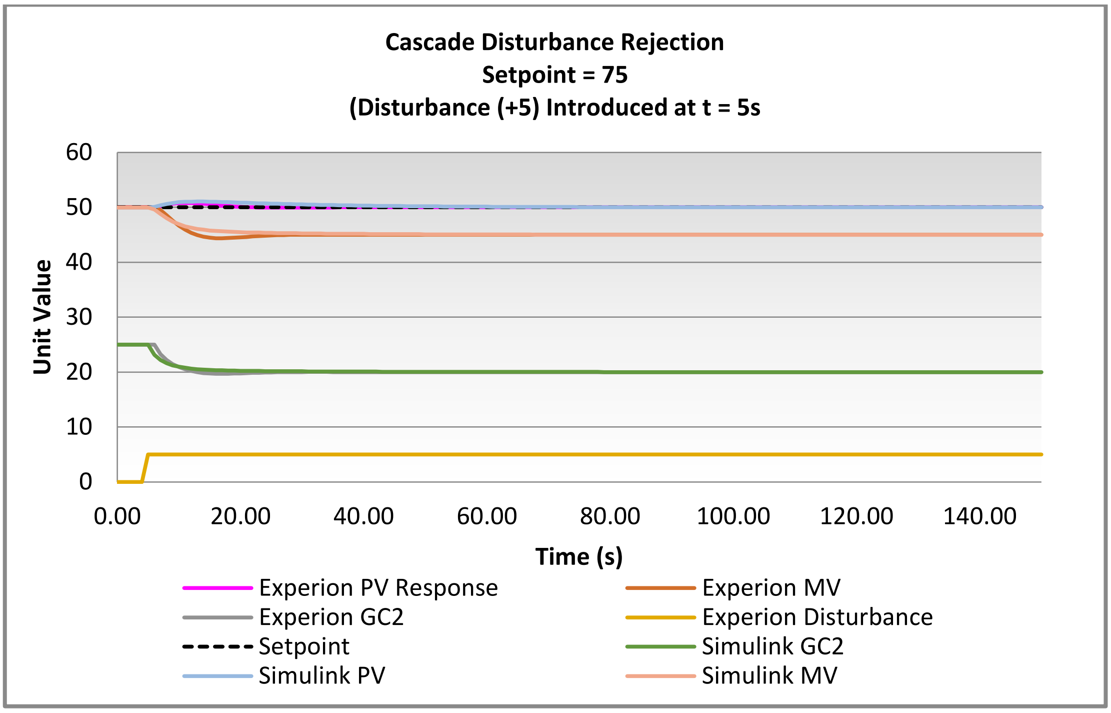


Figure 88: Cascade Control Response Comparison Test 3 (Disturbance +5 at Time =5s)

Test 4: Setpoint = 75, Disturbance -5 at time = 5

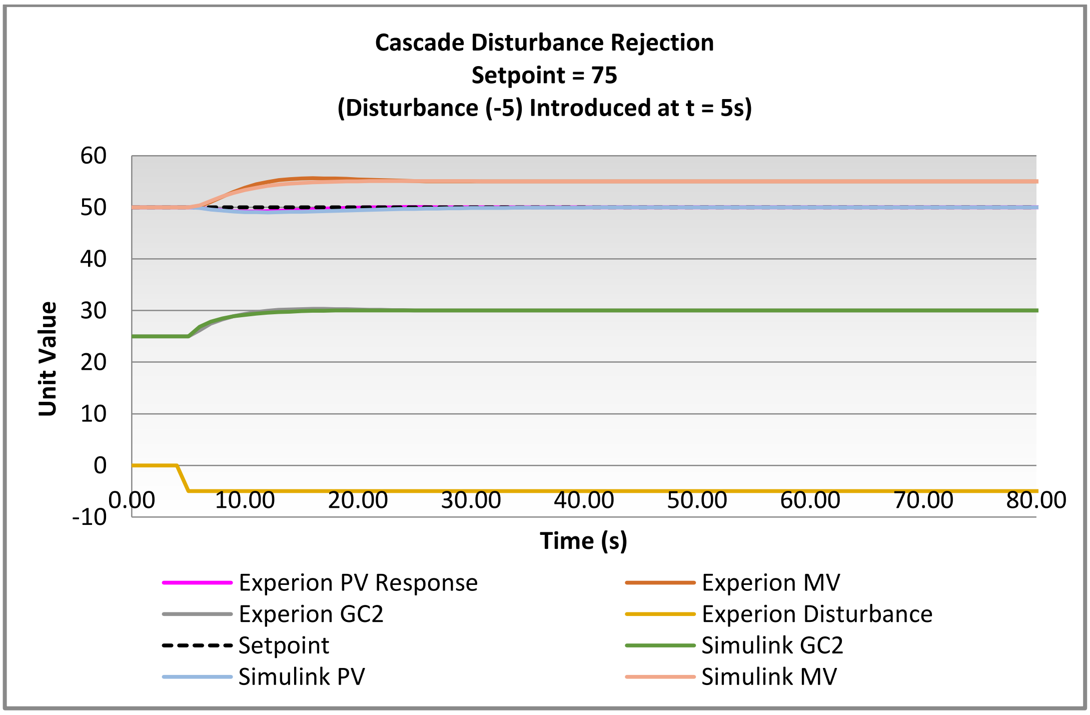


Figure 89: Cascade Control Response Comparison Test 4 (Disturbance -5 at Time =5s)

10.8 Appendix H: Model-Based Predictive Controller Testing

Open Loop Step Test: Step input 25-35 at time = 10s

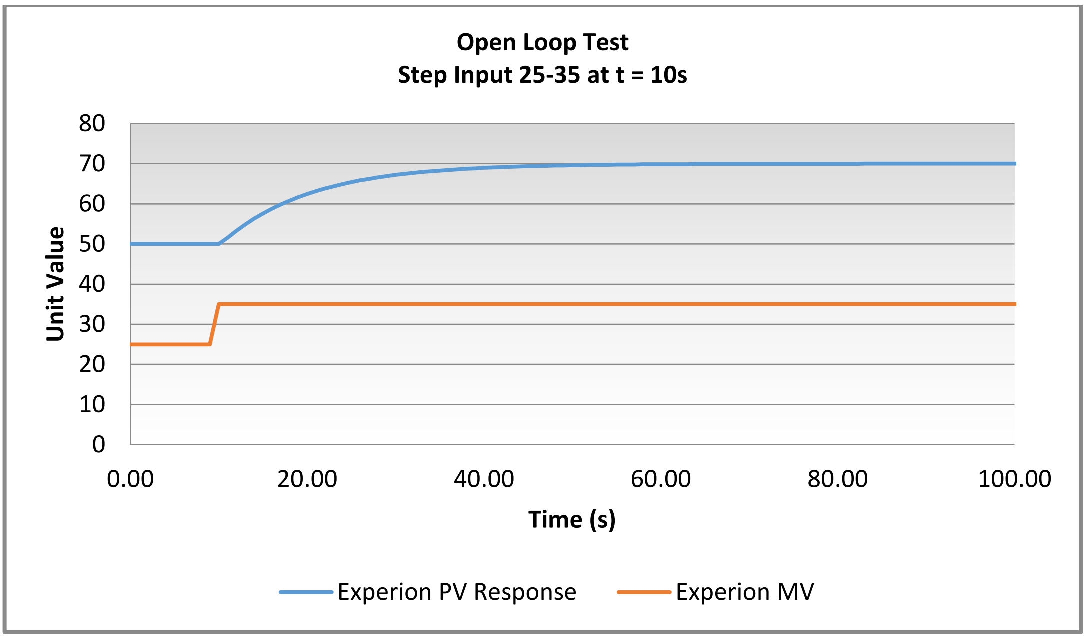


Figure 90: Open Loop Step Test (Step input 25-35 at time = 10s)

Model-Based Predictive Control (PID-PL): Setpoint Change 50-70 at time = 10s [Performance Ratio = 1]

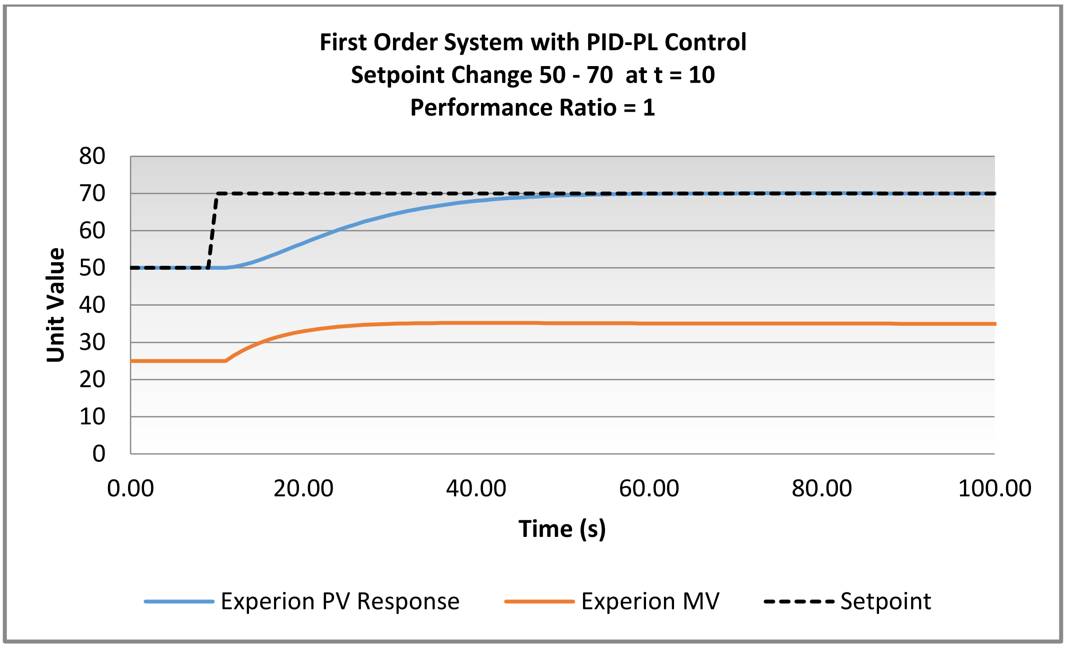


Figure 91: Model-Based Predictive Controller Test (Setpoint Change 50-70 at time = 10s)

Model-Based Predictive Control (PID-PL): Performance Ratio Comparison

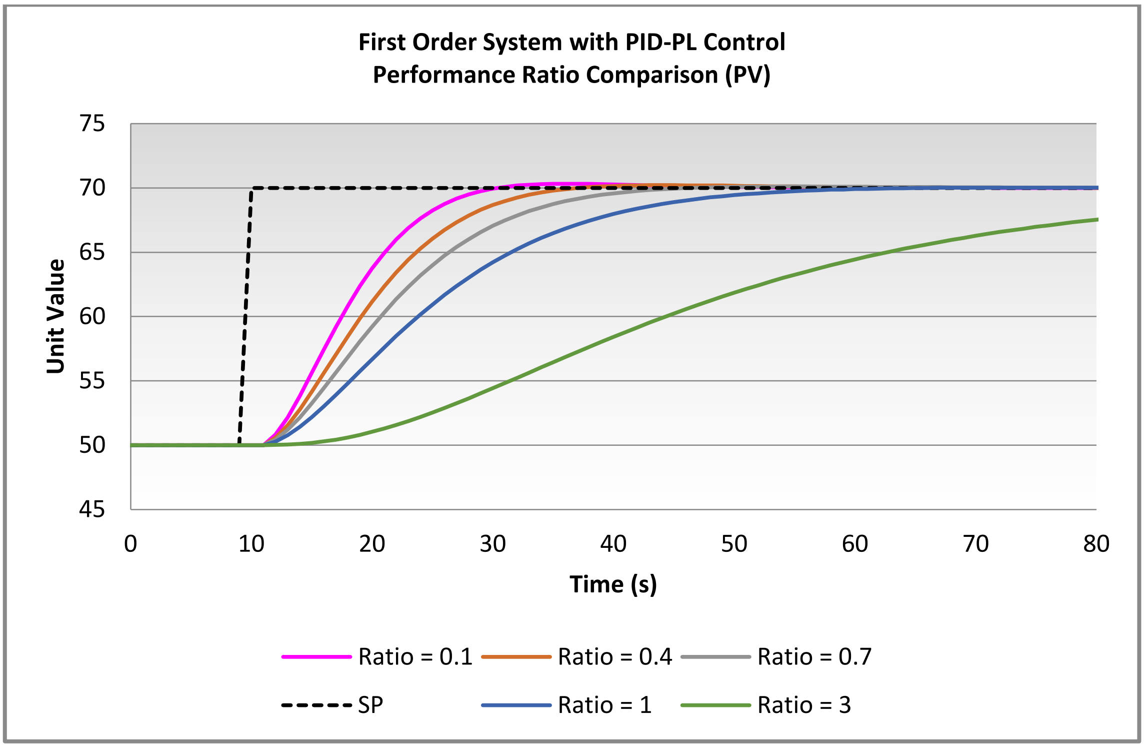


Figure 92: Model-Based Control - Performance Ratio Comparison

10.9 Appendix I: MATLAB OPC Data Exchange Functions

opcda

The **opcda** function is designed to create a data access client server (object). For creating the MATLAB object, MATLAB expects the following code syntax (MathWorks 2016f):

```
Obj = opcda('Host','ServerID')
```

For the creation of a connection to the Experion server the following syntax was used:

```
da = opcda('Experion2','HWHsc.OPCServer')
```

connect

The **connect** function connects the created data access object to the OPC server (MathWorks 2016e). For connecting to the Experion server the following syntax was used:

```
connect(da)
```

Once connected to the Experion server, it was then possible create a group for adding items (point parameters) for monitoring and control

addgroup

The **addgroup** function is designed to create a group for adding items for monitoring and control. Typically, different groups are added to support different sample rates and configuration settings (MathWorks 2016c). The following syntax was used for the creation of a group to monitor and control the point contained within the control module Z_CM_FOS:

```
grp = addgroup(da,'Z_CM_FOS')
```

additem

As the name suggests, the **additem** function add the items (point parameters) to the created group (MathWorks 2016d) . Using intuitive names, the following items from the control module Z_CM_FOS were added:

```
PV = additem(grp, 'Z_CM_FOS.PV.PV', 'double');
U = additem(grp, 'Z_CM_FOS.U.PV', 'double');
tau = additem(grp, 'Z_CM_FOS.tau.PV', 'double');
K = additem(grp, 'Z_CM_FOS.K.PV', 'double');
```

Each of these items are now able to be read and written to using the **read** and **write** functions

read

The **read** function provides the ability to read data synchronously from created groups or items (MathWorks 2016g). The example blow shows the output of the read function for the item PV:

```
PVread = read(PV);
PVread =
       ItemID: 'Z_CM_FOS.PV.PV'
        Value: 40.0000
      Quality: 'Good: Non-specific'
    TimeStamp: [2016 10 23 15 23 41.6000]
        Error: ''
```

Additionally, the **read** function can be used to read an entire group storing the values in an structured array.

write

The **write** function provides the ability to write values to groups or items (MathWorks 2016h). The example below writes the value 20 to the variable U. This variable U is for the item Z_CM_FOS.U.PV.

```
write(U,20);
```

10.10 Appendix J: MATLAB Script – First Order System Step Test

```
%% OPC MATLAB Script
% Simon Godfrey 6/10/2016
% 6/10/2016
% This script allows the user to monitor and control the variables
contained
% within the Experion2 control module Z_CM_FOS.

% Users are able to alter the Process Gain (K) and Time Constant (Tau)
as well as
% introduce a step in the input (U) to the system.
% The response of the system will be plotted on Figure 1 at the
completion
% of the algorithm.
% Note: As this program operates with real-time data acquisiton from
the
% Experion Server (@ 1 sample per second)

%% Clear Workspace
clear
clc

%% Client and Server Configuration
da = opcda('Experion2','HWHsc.OPCServer'); %Create Client Server for
connecting to Experion
connect(da); % Connect to above server (da = Data Access)

%% Create client group
grp = addgroup(da,'Z_CM_FOS');

%% Add items for read/write to group (From Z_CM_FOS)
PV = additem(grp, 'Z_CM_FOS.PV.PV', 'double'); %Add item PV.PV from
Z_CM_FOS
U = additem(grp, 'Z_CM_FOS.U.PV', 'double'); %Add item U.PV from
Z_CM_FOS
tau = additem(grp, 'Z_CM_FOS.tau.PV', 'double'); %Add item tau.PV from
Z_CM_FOS
K = additem(grp, 'Z_CM_FOS.K.PV', 'double'); %Add item K.PV from
Z_CM_FOS

%% Write Value(s) to Experion server
u_write = 20; %Value to be written to U - STEP INPUT
tau_write = 10; %Value to be written to tau - PROCESS TIME CONSTANT
K_write = 2; %Value to be written to K - PROCESS GAIN

%% Simulation time
% Note: consider process time constant when entering simulation time
sim_time = 100; %Set the time of simulation

%% Values Written to Server
write(U,u_write); %Write Value u_write to U.PV
write(tau,tau_write); %Write Value tau_write to tau.PV
write(K,K_write); %Write Value K_write to K.PV

%% Read Group
v1 = read(grp); %Reads all values contained within group created above
```

```
%% Read Value(s) from Experion server
Uread = struct2cell(read(U)); % convert structure of U to cell
dispU = Uread(2); %extract second index from Cell
dispU = dispU{1}; %extract numeric value for displaying

PVread = struct2cell(read(PV));
dispPV = PVread(2);
dispPV = dispPV{1};

tauread = struct2cell(read(tau));
disptau = tauread(2);
disptau = disptau{1};

kread = struct2cell(read(K));
dispK = kread(2);
dispK = dispK{1};

fprintf('Gain = %3.1f   Tau = %3d   Input(U) = %3d  Current PV = %3.1f
\n' ,dispK ,disptau, dispU, dispPV)

%% For loop to grab OPC server values and store in array
i=0;
for i = 1:1:(sim_time + 1);
    pause(1); %1 second pause - acquire data from OPC Experion Server
every 1 second (Fastest rate by Experion)

    PVread = read(PV);
    Uread = read(U);

    PVread = struct2cell(PVread); %convert structure of U to cell
    Uread = struct2cell(Uread);

    PVvalue = PVread(2); %extract second index from Cell
    Uvalue = Uread(2);

    PVvalue = PVvalue{1}; %Grabs Numeric value
    Uvalue = Uvalue{1};

    PVvaluearray(i) = PVvalue; %Store numeric values into array
    Uvaluearray(i) = Uvalue;
end

 %% Plotting of steps for testing
t=0:1:sim_time; %Time value for x axis plotting
figure(1)
plot(t,PVvaluearray,'-ro', t, Uvaluearray, '-bo') %Plot PV and U
arrays
title('Z CM FOS : First Order System OPC Values');
legend('Process Variable PV.PV', 'Input Variable U.PV');
ylabel('Value');
xlabel('Time(s)');
ylim([0 100]);

%% Transpose Arrays for Excel Plots (Comparison to Experion Data)
PV_transpose = transpose(PVvaluearray);
U_Transpose = transpose(Uvaluearray);
```

10.11 Appendix K: MATLAB First Order System Testing

Example 1: Input Step 20-30 – Raw MATLAB Data Acquired

```
Gain = 2.0   Tau = 10   Input(U) = 30  Current PV = 40.0
```

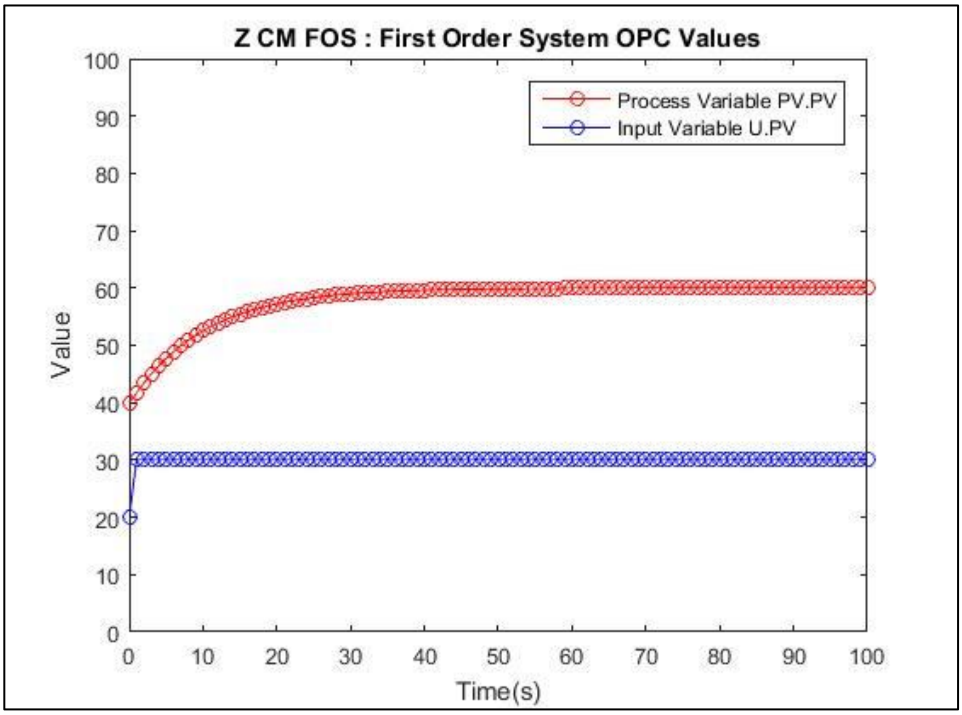


Figure 93: MATLAB OPC Step Test (Step input 20-30)

Table 12: MATLAB OPC First Order System Step Test Raw Tabulated Results (Experion vs MATLAB OPC)

Date	Time	Time	Experion PV Response	Step Input	MATLAB (OPC) PV
10/18/2016	########	0.00	40	20	40
10/18/2016	########	1.00	40	20	40
10/18/2016	########	2.00	40	20	40
10/18/2016	########	3.00	40	20	40
10/18/2016	########	4.00	40	20	40
10/18/2016	########	5.00	40	20	40
10/18/2016	########	6.00	40	20	40
10/18/2016	########	7.00	40	20	40
10/18/2016	########	8.00	40	20	40
10/18/2016	########	9.00	40	20	40
10/18/2016	########	10.00	40	20	40
10/18/2016	########	11.00	40	20	40
10/18/2016	########	12.00	40	20	40
10/18/2016	########	13.00	40	20	40
10/18/2016	########	14.00	40	20	40

10/18/2016	########	15.00	40	20	40
10/18/2016	########	16.00	40	20	40
10/18/2016	########	17.00	40	20	40
10/18/2016	########	18.00	40	20	40
10/18/2016	########	19.00	40	20	40
10/18/2016	########	20.00	41.94	40	41.94137257
10/18/2016	########	21.00	45.55	40	45.54602108
10/18/2016	########	22.00	48.81	40	48.80926234
10/18/2016	########	23.00	51.76	40	51.76343209
10/18/2016	########	24.00	54.44	40	54.43780343
10/18/2016	########	25.00	56.86	40	56.85887691
10/18/2016	########	26.00	59.05	40	59.05064315
10/18/2016	########	27.00	61.03	40	61.03482053
10/18/2016	########	28.00	62.83	40	62.83107044
10/18/2016	########	29.00	64.46	40	64.45719204
10/18/2016	########	30.00	65.93	40	65.92929871
10/18/2016	########	31.00	67.26	40	67.26197768
10/18/2016	########	32.00	68.47	40	68.46843457
10/18/2016	########	33.00	69.56	40	69.56062425
10/18/2016	########	34.00	70.55	40	70.54936932
10/18/2016	########	35.00	71.44	40	71.44446733
10/18/2016	########	36.00	72.25	40	72.25478788
10/18/2016	########	37.00	72.99	40	72.98836051
10/18/2016	########	38.00	73.65	40	73.65245424
10/18/2016	########	39.00	74.25	40	74.25364963
10/18/2016	########	40.00	74.8	40	74.79790397
10/18/2016	########	41.00	75.29	40	75.29061033
10/18/2016	########	42.00	75.74	40	75.73665096
10/18/2016	########	43.00	76.14	40	76.14044573
10/18/2016	########	44.00	76.51	40	76.50599587
10/18/2016	########	45.00	76.84	40	76.83692364
10/18/2016	########	46.00	77.14	40	77.13650824
10/18/2016	########	47.00	77.41	40	77.40771826
10/18/2016	########	48.00	77.65	40	77.65324116
10/18/2016	########	49.00	77.88	40	77.87550984
10/18/2016	########	50.00	78.08	40	78.07672677
10/18/2016	########	51.00	78.26	40	78.25888583
10/18/2016	########	52.00	78.42	40	78.42379206
10/18/2016	########	53.00	78.57	40	78.57307951
10/18/2016	########	54.00	78.71	40	78.7082275
10/18/2016	########	55.00	78.83	40	78.83057521
10/18/2016	########	56.00	78.94	40	78.94133499
10/18/2016	########	57.00	79.04	40	79.04160438
10/18/2016	########	58.00	79.13	40	79.13237694
10/18/2016	########	59.00	79.21	40	79.21455217

10/18/2016	########	60.00	79.29	40	79.28894432
10/18/2016	########	61.00	79.36	40	79.35629057
10/18/2016	########	62.00	79.42	40	79.41725825
10/18/2016	########	63.00	79.47	40	79.4724515
10/18/2016	########	64.00	79.52	40	79.52241722
10/18/2016	########	65.00	79.57	40	79.56765054
10/18/2016	########	66.00	79.61	40	79.60859968
10/18/2016	########	67.00	79.65	40	79.64567039
10/18/2016	########	68.00	79.68	40	79.67923003
10/18/2016	########	69.00	79.71	40	79.70961113
10/18/2016	########	70.00	79.74	40	79.73711474
10/18/2016	########	71.00	79.76	40	79.76201341
10/18/2016	########	72.00	79.78	40	79.78455384
10/18/2016	########	73.00	79.8	40	79.8049594
10/18/2016	########	74.00	79.82	40	79.82343229
10/18/2016	########	75.00	79.84	40	79.84015556
10/18/2016	########	76.00	79.86	40	79.85529491
10/18/2016	########	77.00	79.87	40	79.86900037
10/18/2016	########	78.00	79.88	40	79.88140774
10/18/2016	########	79.00	79.89	40	79.89263998
10/18/2016	########	80.00	79.9	40	79.90280837
10/18/2016	########	81.00	79.91	40	79.91201369
10/18/2016	########	82.00	79.92	40	79.9278913
10/18/2016	########	83.00	79.93	40	79.93472094
10/18/2016	########	84.00	79.93	40	79.94090372
10/18/2016	########	85.00	79.94	40	79.94650091
10/18/2016	########	86.00	79.95	40	79.95156797
10/18/2016	########	87.00	79.95	40	79.95615511
10/18/2016	########	88.00	79.96	40	79.9603078
10/18/2016	########	89.00	79.96	40	79.96406717
10/18/2016	########	90.00	79.96	40	79.96747047
10/18/2016	########	91.00	79.97	40	79.97055144
10/18/2016	########	92.00	79.97	40	79.97334061
10/18/2016	########	93.00	79.97	40	79.9758656
10/18/2016	########	94.00	79.98	40	79.97815144
10/18/2016	########	95.00	79.98	40	79.98022079
10/18/2016	########	96.00	79.98	40	79.98209413
10/18/2016	########	97.00	79.98	40	79.98379005
10/18/2016	########	98.00	79.98	40	79.98532535
10/18/2016	########	99.00	79.99	40	79.98671523
10/18/2016	########	100.00	79.99	40	79.98797347

Example 2: Input Step 20-40

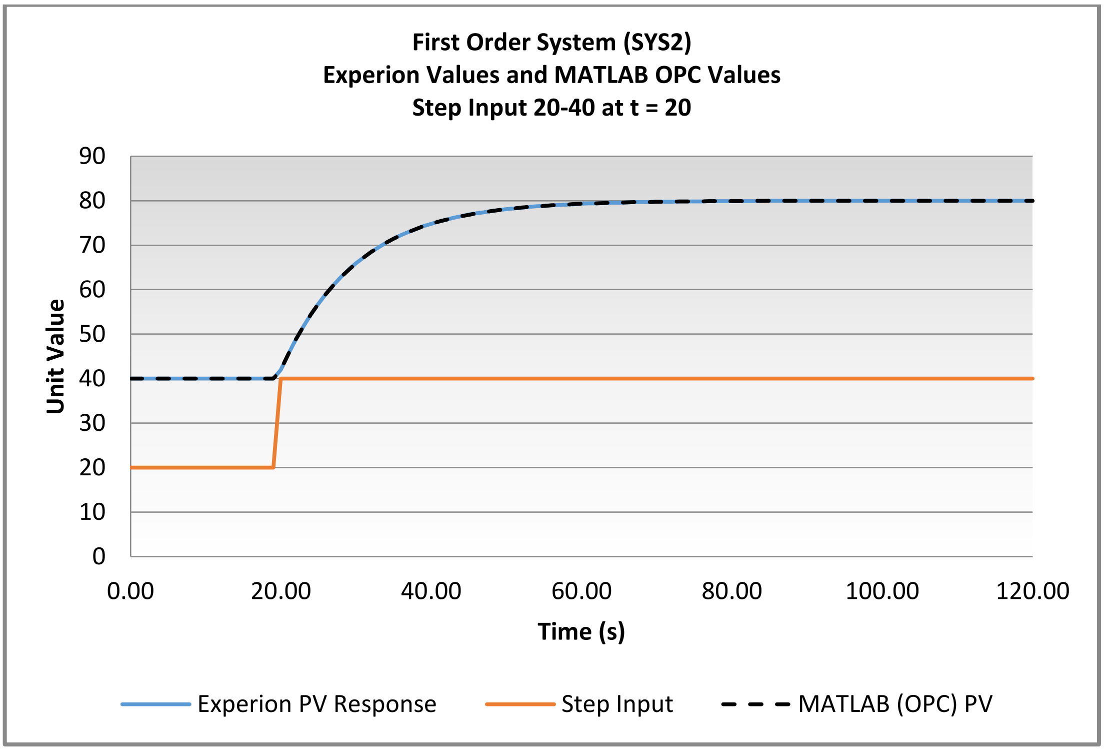


Figure 94: MATLAB OPC Testing - First Order System Test (Step input 20-40 at time = 20s)

Example 3: Input Step 30-20

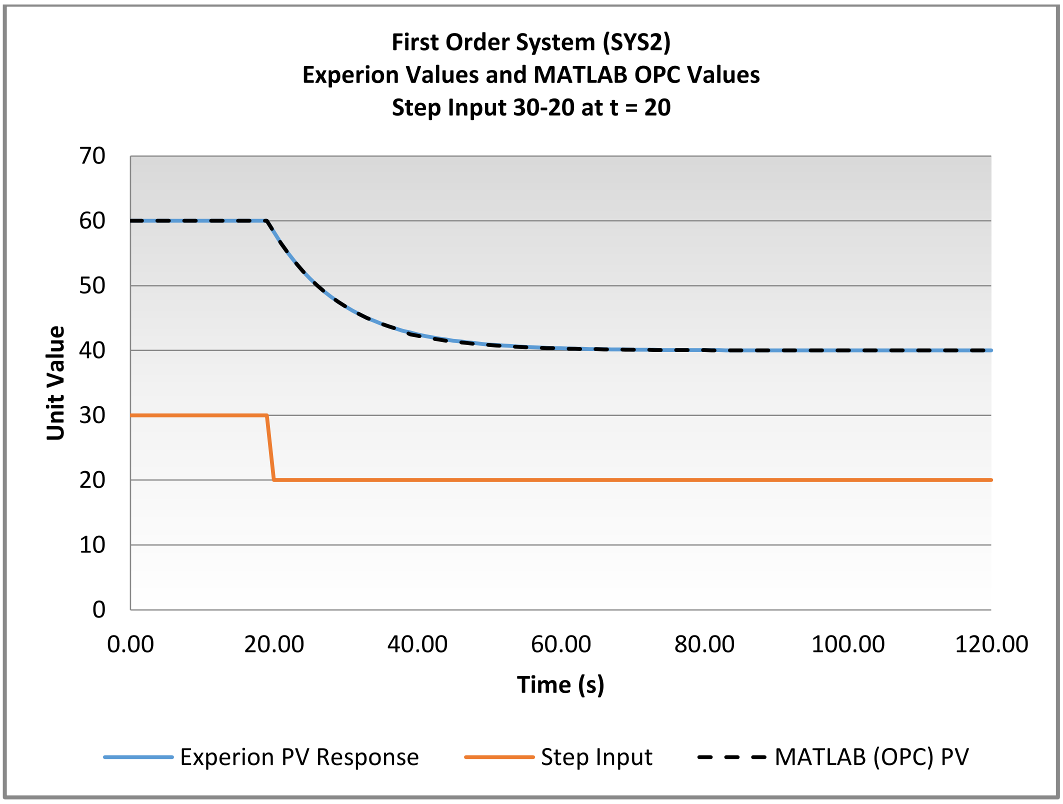


Figure 95: MATLAB OPC Testing - First Order System Test (Step input 30-20 at time = 20s)

10.12 Appendix L: Simulink First Order System OPC Testing

Example 1: Input Step 20-30 – Raw Simulink Data Acquired

Table 13: Simulink OPC First Order System Step Test Raw Tabulated Results (Experion vs Simulink OPC)

Date	Time	Time	Experion PV Response	Step Input	Simulink (OPC) PV
10/27/2016	5:13:05 PM	0	40.00	20	40.00
10/27/2016	5:13:06 PM	1	40.00	20	40.00
10/27/2016	5:13:07 PM	2	40.00	20	40.00
10/27/2016	5:13:08 PM	3	40.00	20	40.00
10/27/2016	5:13:09 PM	4	40.00	20	40.00
10/27/2016	5:13:10 PM	5	40.00	20	40.00
10/27/2016	5:13:11 PM	6	40.00	20	40.00
10/27/2016	5:13:12 PM	7	40.00	20	40.00
10/27/2016	5:13:13 PM	8	40.00	20	40.00
10/27/2016	5:13:14 PM	9	40.00	20	40.00
10/27/2016	5:13:15 PM	10	40.59	30	40.59
10/27/2016	5:13:16 PM	11	42.43	30	42.43
10/27/2016	5:13:17 PM	12	44.09	30	44.09
10/27/2016	5:13:18 PM	13	45.60	30	45.60
10/27/2016	5:13:19 PM	14	46.96	30	46.96
10/27/2016	5:13:20 PM	15	48.20	30	48.20
10/27/2016	5:13:21 PM	16	49.31	30	49.31
10/27/2016	5:13:22 PM	17	50.33	30	50.33
10/27/2016	5:13:23 PM	18	51.24	30	51.24
10/27/2016	5:13:24 PM	19	52.07	30	52.07
10/27/2016	5:13:25 PM	20	52.82	30	52.82
10/27/2016	5:13:26 PM	21	53.50	30	53.50
10/27/2016	5:13:27 PM	22	54.12	30	54.12
10/27/2016	5:13:28 PM	23	54.68	30	54.68
10/27/2016	5:13:29 PM	24	55.18	30	55.18
10/27/2016	5:13:30 PM	25	55.64	30	55.64
10/27/2016	5:13:31 PM	26	56.05	30	56.05
10/27/2016	5:13:32 PM	27	56.42	30	56.42
10/27/2016	5:13:33 PM	28	56.76	30	56.76
10/27/2016	5:13:34 PM	29	57.07	30	57.07
10/27/2016	5:13:35 PM	30	57.35	30	57.35
10/27/2016	5:13:36 PM	31	57.60	30	57.60
10/27/2016	5:13:37 PM	32	57.83	30	57.83
10/27/2016	5:13:38 PM	33	58.03	30	58.03
10/27/2016	5:13:39 PM	34	58.22	30	58.22
10/27/2016	5:13:40 PM	35	58.39	30	58.39

10/27/2016	5:13:41 PM	36	58.54	30	58.54
10/27/2016	5:13:42 PM	37	58.68	30	58.68
10/27/2016	5:13:43 PM	38	58.80	30	58.80
10/27/2016	5:13:44 PM	39	58.92	30	58.92
10/27/2016	5:13:45 PM	40	59.02	30	59.02
10/27/2016	5:13:46 PM	41	59.11	30	59.11
10/27/2016	5:13:47 PM	42	59.20	30	59.20
10/27/2016	5:13:48 PM	43	59.27	30	59.27
10/27/2016	5:13:49 PM	44	59.34	30	59.34
10/27/2016	5:13:50 PM	45	59.40	30	59.40
10/27/2016	5:13:51 PM	46	59.46	30	59.46
10/27/2016	5:13:52 PM	47	59.51	30	59.51
10/27/2016	5:13:53 PM	48	59.56	30	59.56
10/27/2016	5:13:54 PM	49	59.60	30	59.60
10/27/2016	5:13:55 PM	50	59.64	30	59.64
10/27/2016	5:13:56 PM	51	59.67	30	59.67
10/27/2016	5:13:57 PM	52	59.70	30	59.70
10/27/2016	5:13:58 PM	53	59.73	30	59.73
10/27/2016	5:13:59 PM	54	59.76	30	59.76
10/27/2016	5:14:00 PM	55	59.78	30	59.78
10/27/2016	5:14:01 PM	56	59.80	30	59.80
10/27/2016	5:14:02 PM	57	59.82	30	59.82
10/27/2016	5:14:03 PM	58	59.84	30	59.84
10/27/2016	5:14:04 PM	59	59.85	30	59.85
10/27/2016	5:14:05 PM	60	59.87	30	59.87
10/27/2016	5:14:06 PM	61	59.88	30	59.88
10/27/2016	5:14:07 PM	62	59.89	30	59.89
10/27/2016	5:14:08 PM	63	59.90	30	59.90
10/27/2016	5:14:09 PM	64	59.91	30	59.91
10/27/2016	5:14:10 PM	65	59.92	30	59.92
10/27/2016	5:14:11 PM	66	59.93	30	59.93
10/27/2016	5:14:12 PM	67	59.93	30	59.93
10/27/2016	5:14:13 PM	68	59.94	30	59.94
10/27/2016	5:14:14 PM	69	59.95	30	59.95
10/27/2016	5:14:15 PM	70	59.95	30	59.95
10/27/2016	5:14:16 PM	71	59.96	30	59.96
10/27/2016	5:14:17 PM	72	59.96	30	59.96
10/27/2016	5:14:18 PM	73	59.96	30	59.96
10/27/2016	5:14:19 PM	74	59.97	30	59.97
10/27/2016	5:14:20 PM	75	59.97	30	59.97
10/27/2016	5:14:21 PM	76	59.97	30	59.97
10/27/2016	5:14:22 PM	77	59.98	30	59.98
10/27/2016	5:14:23 PM	78	59.98	30	59.98
10/27/2016	5:14:24 PM	79	59.98	30	59.98
10/27/2016	5:14:25 PM	80	59.98	30	59.98

10/27/2016	5:14:26 PM	81	59.98	30	59.98
10/27/2016	5:14:27 PM	82	59.98	30	59.98
10/27/2016	5:14:28 PM	83	59.99	30	59.99
10/27/2016	5:14:29 PM	84	59.99	30	59.99
10/27/2016	5:14:30 PM	85	59.99	30	59.99
10/27/2016	5:14:31 PM	86	59.99	30	59.99
10/27/2016	5:14:32 PM	87	59.99	30	59.99
10/27/2016	5:14:33 PM	88	59.99	30	59.99
10/27/2016	5:14:34 PM	89	59.99	30	59.99
10/27/2016	5:14:35 PM	90	59.99	30	59.99
10/27/2016	5:14:36 PM	91	59.99	30	59.99
10/27/2016	5:14:37 PM	92	59.99	30	59.99
10/27/2016	5:14:38 PM	93	60.00	30	60.00
10/27/2016	5:14:39 PM	94	60.00	30	60.00
10/27/2016	5:14:40 PM	95	60.00	30	60.00
10/27/2016	5:14:41 PM	96	60.00	30	60.00
10/27/2016	5:14:42 PM	97	60.00	30	60.00
10/27/2016	5:14:43 PM	98	60.00	30	60.00
10/27/2016	5:14:44 PM	99	60.00	30	60.00
10/27/2016	5:14:45 PM	100	60.00	30	60.00

10.13 Appendix M: Simulink OPC Control Testing

Test 2 – Setpoint change 60-40 at time = 20s

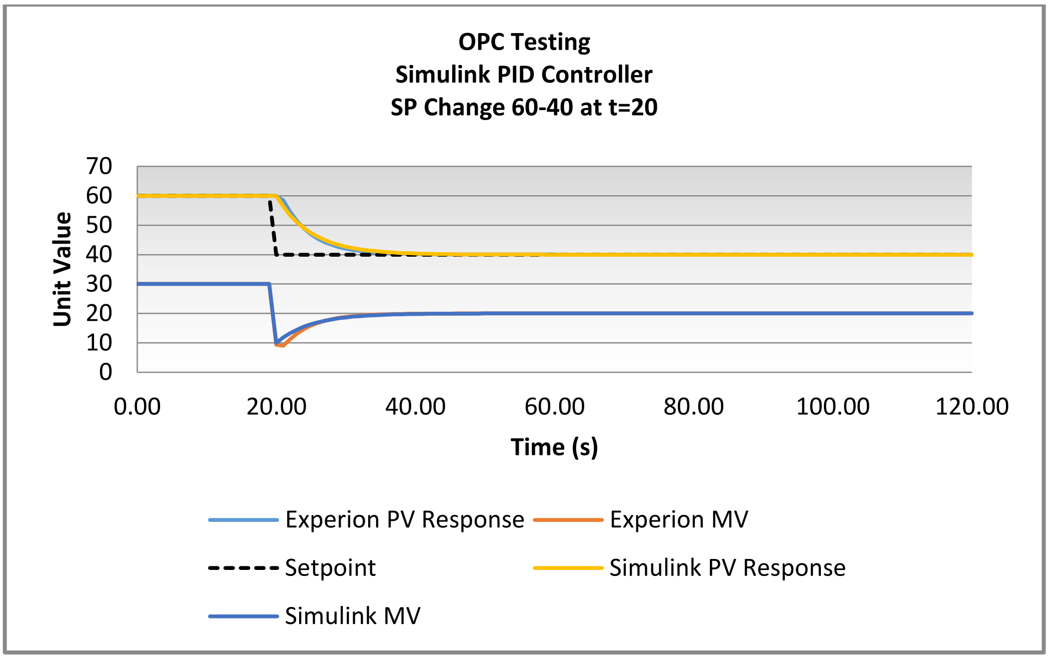


Figure 96: Simulink OPC Controller Testing (Setpoint 60-40 at time = 20s)

Test 3 – Setpoint change 40-30 at time = 20s

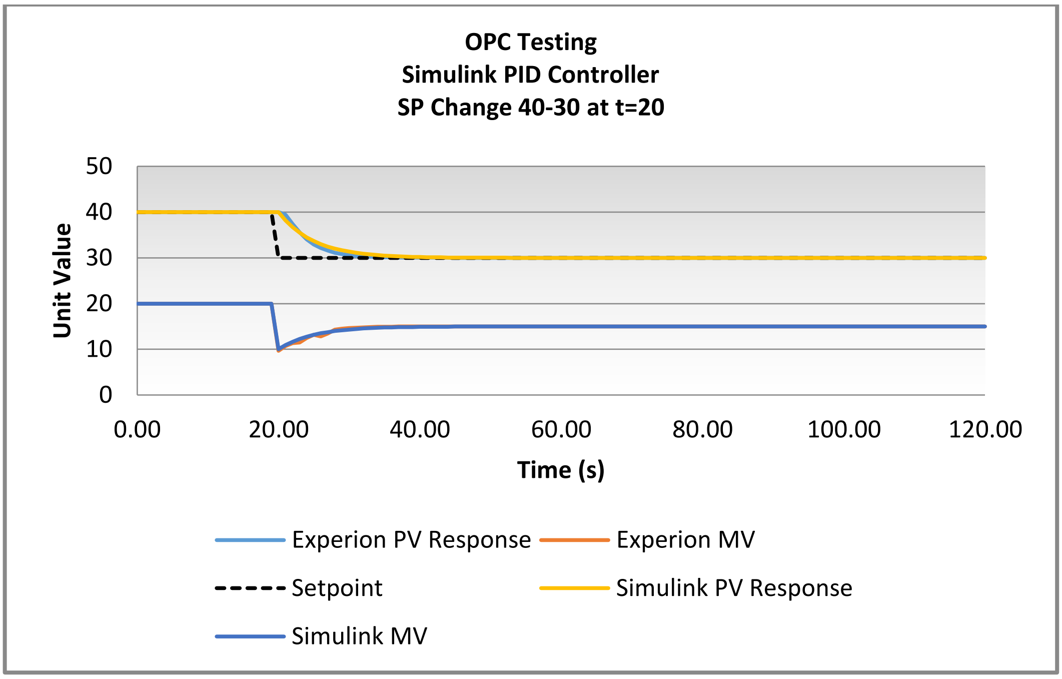


Figure 97: Simulink OPC Controller Testing (Setpoint 40-30 at time = 20s)

Test 3 – Setpoint change 30-40 at time = 20s

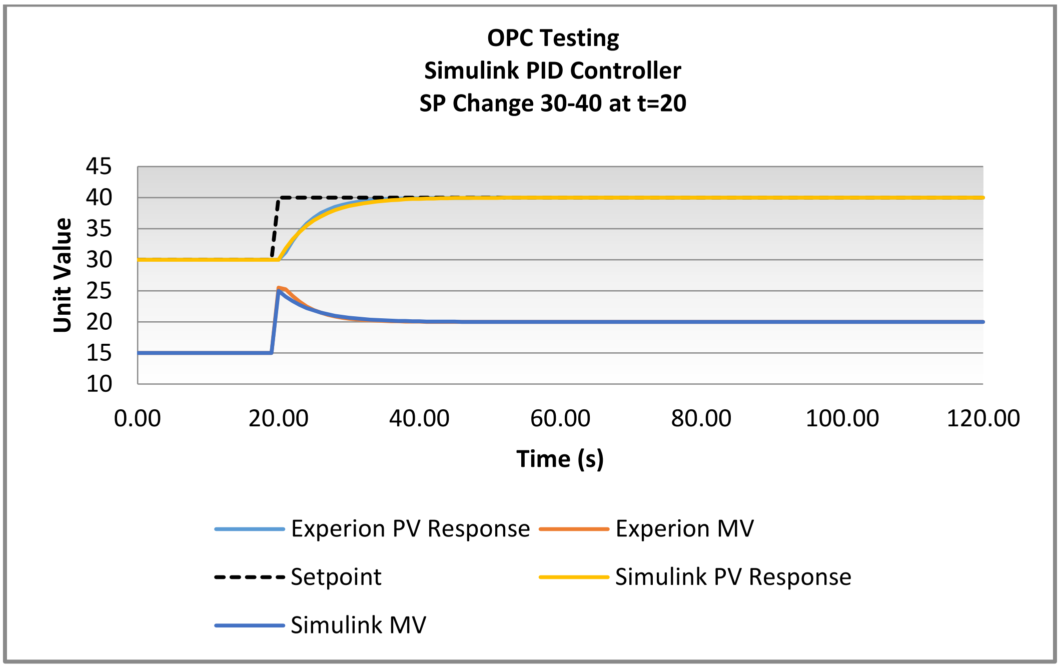


Figure 98: Simulink OPC Controller Testing (Setpoint 30-40 at time = 20s)

10.14 Appendix N: LabVIEW First Order System OPC Testing

Example 1: Input Step 20-30 – Raw LabVIEW Data Acquired

Table 14: LabVIEW OPC First Order System Step Test Raw Tabulated Results (Experion vs LabVIEW OPC)

Date	Time	Experion PV Response	Experion MV	Simulink MV	Simulink PV Response
10/18/2016	6:13:05 PM	40	20	20	40
10/18/2016	6:13:06 PM	40	20	20	40
10/18/2016	6:13:07 PM	40	20	20	40
10/18/2016	6:13:08 PM	40	20	20	40
10/18/2016	6:13:09 PM	40	20	20	40
10/18/2016	6:13:10 PM	40	20	20	40
10/18/2016	6:13:11 PM	40	20	20	40
10/18/2016	6:13:12 PM	40	20	20	40
10/18/2016	6:13:13 PM	40	20	20	40
10/18/2016	6:13:14 PM	40	20	20	40
10/18/2016	6:13:15 PM	40	20	20	40
10/18/2016	6:13:16 PM	40.2	30	30	40.2
10/18/2016	6:13:17 PM	42.07	30	30	42.07
10/18/2016	6:13:18 PM	43.77	30	30	43.77
10/18/2016	6:13:19 PM	45.31	30	30	45.31
10/18/2016	6:13:20 PM	46.7	30	30	46.7
10/18/2016	6:13:21 PM	47.96	30	30	47.96
10/18/2016	6:13:22 PM	49.1	30	30	49.1
10/18/2016	6:13:23 PM	50.13	30	30	50.13
10/18/2016	6:13:24 PM	51.07	30	30	51.07
10/18/2016	6:13:25 PM	51.91	30	30	51.91
10/18/2016	6:13:26 PM	52.68	30	30	52.68
10/18/2016	6:13:27 PM	53.37	30	30	53.37
10/18/2016	6:13:28 PM	54	30	30	54
10/18/2016	6:13:29 PM	54.57	30	30	54.57
10/18/2016	6:13:30 PM	55.08	30	30	55.08
10/18/2016	6:13:31 PM	55.55	30	30	55.55
10/18/2016	6:13:32 PM	55.97	30	30	55.97
10/18/2016	6:13:33 PM	56.35	30	30	56.35
10/18/2016	6:13:34 PM	56.7	30	30	56.7
10/18/2016	6:13:35 PM	57.01	30	30	57.01
10/18/2016	6:13:36 PM	57.29	30	30	57.29
10/18/2016	6:13:37 PM	57.55	30	30	57.55
10/18/2016	6:13:38 PM	57.78	30	30	57.78
10/18/2016	6:13:39 PM	57.99	30	30	57.99

10/18/2016	6:13:40 PM	58.18	30	30	58.18
10/18/2016	6:13:41 PM	58.35	30	30	58.35
10/18/2016	6:13:42 PM	58.51	30	30	58.51
10/18/2016	6:13:43 PM	58.65	30	30	58.65
10/18/2016	6:13:44 PM	58.78	30	30	58.78
10/18/2016	6:13:45 PM	58.89	30	30	58.89
10/18/2016	6:13:46 PM	59	30	30	59
10/18/2016	6:13:47 PM	59.09	30	30	59.09
10/18/2016	6:13:48 PM	59.18	30	30	59.18
10/18/2016	6:13:49 PM	59.26	30	30	59.26
10/18/2016	6:13:50 PM	59.33	30	30	59.33
10/18/2016	6:13:51 PM	59.39	30	30	59.39
10/18/2016	6:13:52 PM	59.45	30	30	59.45
10/18/2016	6:13:53 PM	59.5	30	30	59.5
10/18/2016	6:13:54 PM	59.55	30	30	59.55
10/18/2016	6:13:55 PM	59.59	30	30	59.59
10/18/2016	6:13:56 PM	59.63	30	30	59.63
10/18/2016	6:13:57 PM	59.67	30	30	59.67
10/18/2016	6:13:58 PM	59.7	30	30	59.7
10/18/2016	6:13:59 PM	59.73	30	30	59.73
10/18/2016	6:14:00 PM	59.75	30	30	59.75
10/18/2016	6:14:01 PM	59.78	30	30	59.78
10/18/2016	6:14:02 PM	59.8	30	30	59.8
10/18/2016	6:14:03 PM	59.82	30	30	59.82
10/18/2016	6:14:04 PM	59.83	30	30	59.83
10/18/2016	6:14:05 PM	59.85	30	30	59.85
10/18/2016	6:14:06 PM	59.86	30	30	59.86
10/18/2016	6:14:07 PM	59.88	30	30	59.88
10/18/2016	6:14:08 PM	59.89	30	30	59.89
10/18/2016	6:14:09 PM	59.9	30	30	59.9
10/18/2016	6:14:10 PM	59.91	30	30	59.91
10/18/2016	6:14:11 PM	59.92	30	30	59.92
10/18/2016	6:14:12 PM	59.92	30	30	59.92
10/18/2016	6:14:13 PM	59.93	30	30	59.93
10/18/2016	6:14:14 PM	59.94	30	30	59.94
10/18/2016	6:14:15 PM	59.94	30	30	59.94
10/18/2016	6:14:16 PM	59.95	30	30	59.95
10/18/2016	6:14:17 PM	59.95	30	30	59.95
10/18/2016	6:14:18 PM	59.96	30	30	59.96
10/18/2016	6:14:19 PM	59.96	30	30	59.96
10/18/2016	6:14:20 PM	59.97	30	30	59.97
10/18/2016	6:14:21 PM	59.97	30	30	59.97
10/18/2016	6:14:22 PM	59.97	30	30	59.97
10/18/2016	6:14:23 PM	59.97	30	30	59.97
10/18/2016	6:14:24 PM	59.98	30	30	59.98

10/18/2016	6:14:25 PM	59.98	30	30	59.98
10/18/2016	6:14:26 PM	59.98	30	30	59.98
10/18/2016	6:14:27 PM	59.98	30	30	59.98
10/18/2016	6:14:28 PM	59.98	30	30	59.98
10/18/2016	6:14:29 PM	59.99	30	30	59.99
10/18/2016	6:14:30 PM	59.99	30	30	59.99
10/18/2016	6:14:31 PM	59.99	30	30	59.99
10/18/2016	6:14:32 PM	59.99	30	30	59.99
10/18/2016	6:14:33 PM	59.99	30	30	59.99
10/18/2016	6:14:34 PM	59.99	30	30	59.99
10/18/2016	6:14:35 PM	59.99	30	30	59.99
10/18/2016	6:14:36 PM	59.99	30	30	59.99
10/18/2016	6:14:37 PM	59.99	30	30	59.99
10/18/2016	6:14:38 PM	59.99	30	30	59.99
10/18/2016	6:14:39 PM	59.99	30	30	59.99
10/18/2016	6:14:40 PM	60	30	30	60
10/18/2016	6:14:41 PM	60	30	30	60
10/18/2016	6:14:42 PM	60	30	30	60
10/18/2016	6:14:43 PM	60	30	30	60
10/18/2016	6:14:44 PM	60	30	30	60
10/18/2016	6:14:45 PM	60	30	30	60
10/18/2016	6:14:46 PM	60	30	30	60
10/18/2016	6:14:47 PM	60	30	30	60
10/18/2016	6:14:48 PM	60	30	30	60
10/18/2016	6:14:49 PM	60	30	30	60
10/18/2016	6:14:50 PM	60	30	30	60
10/18/2016	6:14:51 PM	60	30	30	60
10/18/2016	6:14:52 PM	60	30	30	60
10/18/2016	6:14:53 PM	60	30	30	60
10/18/2016	6:14:54 PM	60	30	30	60
10/18/2016	6:14:55 PM	60	30	30	60
10/18/2016	6:14:56 PM	60	30	30	60
10/18/2016	6:14:57 PM	60	30	30	60
10/18/2016	6:14:58 PM	60	30	30	60
10/18/2016	6:14:59 PM	60	30	30	60
10/18/2016	6:15:00 PM	60	30	30	60
10/18/2016	6:15:01 PM	60	30	30	60
10/18/2016	6:15:02 PM	60	30	30	60
10/18/2016	6:15:03 PM	60	30	30	60
10/18/2016	6:15:04 PM	60	30	30	60
10/18/2016	6:15:05 PM	60	30	30	60

10.15 Appendix O: LabVIEW First Order System with PID (OPC) Testing

Test 2 – Setpoint change 60-40 at time = 20s

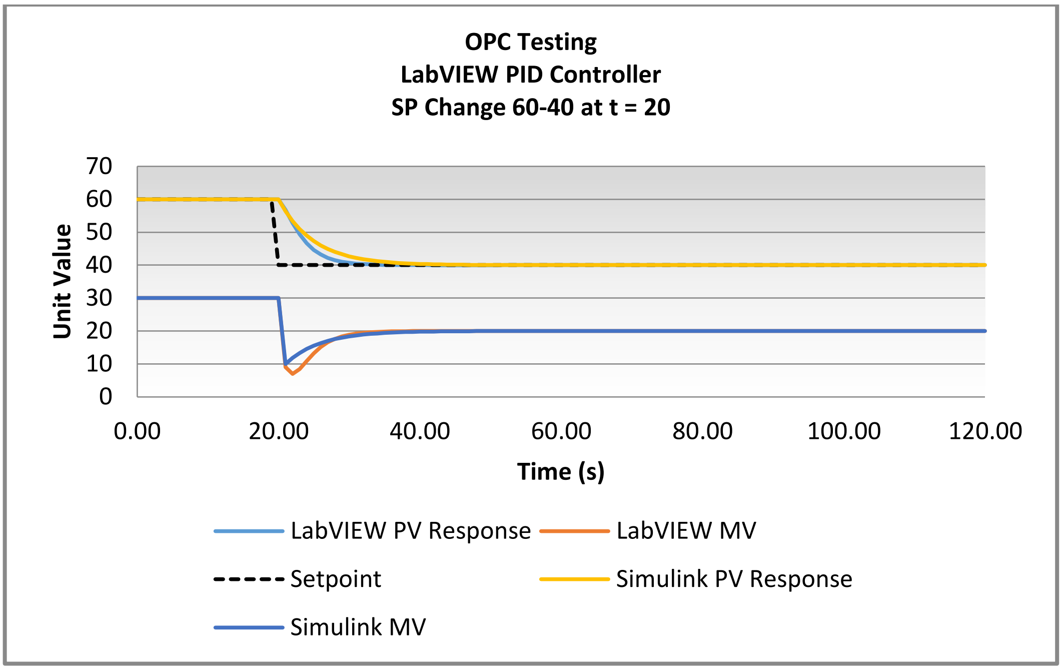


Figure 99: LabVIEW OPC Controller Testing (Setpoint 60-40 at time = 20s)

Test 3 – Setpoint change 40-30 at time = 20s

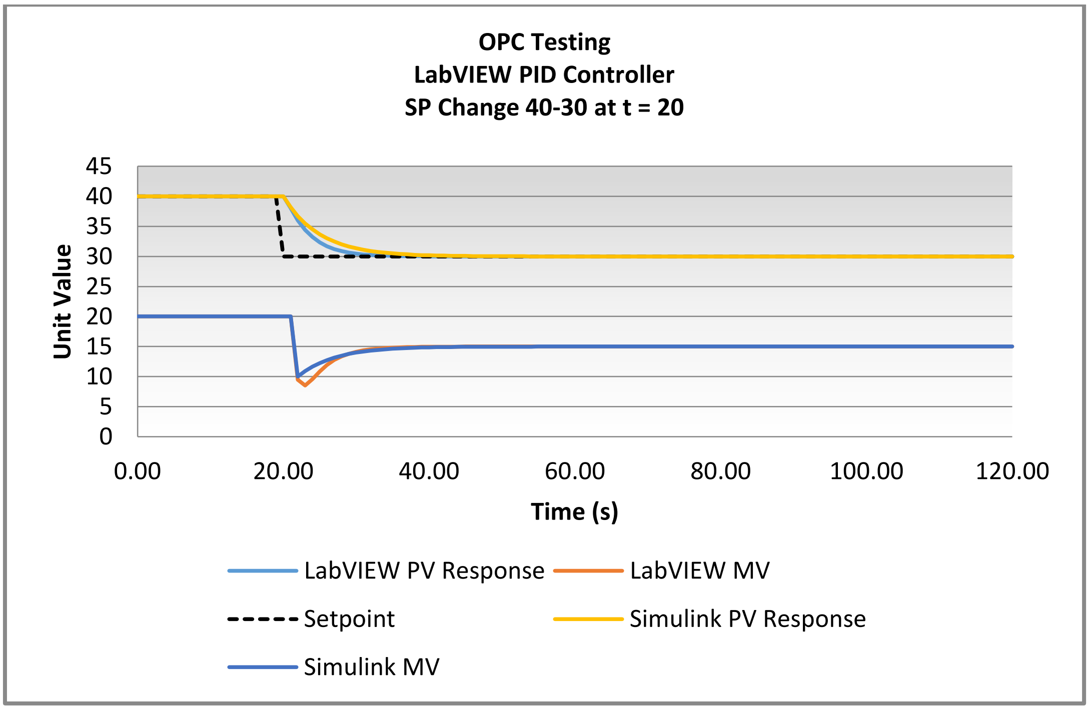


Figure 100: LabVIEW OPC Controller Testing (Setpoint 40-30 at time = 20s)

Test 4 – Setpoint change 30-40 at time = 20s

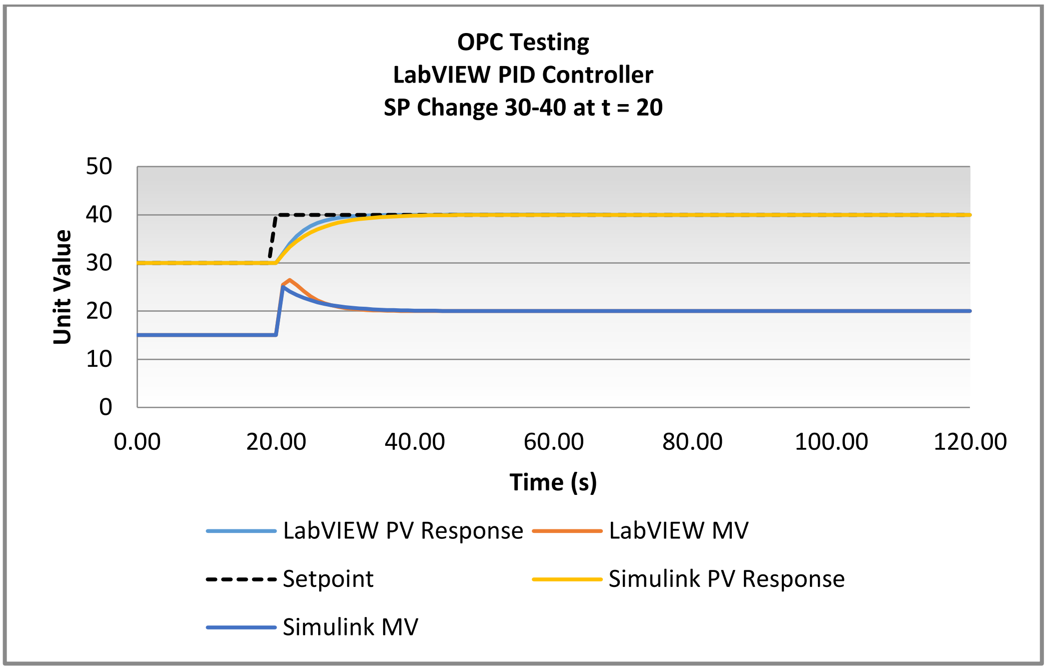


Figure 101: LabVIEW OPC Controller Testing (Setpoint 30-40 at time = 20s)

10.16 Appendix P: Previous Learning Material Authors

Credit is given to the previous authors of learning materials revised during the completion of this project. These authors are:

- Charlotte Moss (2013)
- Fasil Worku (2013)
- James Baker (2013)
- Shane Walker (2013)
- Nathan Froese (2013)
- Matthew Kenneday (2013)
- Dan Murphy (2013)
- Nurazlina Binte Mohamad Nain (2013)
- Kiranmai Gumireddy (2013)
- Michael Gant (2014)
- Michael D'Uva (2014)
- Sean Dean (2014)
- Michael Maynard (2014)
- Scott Zanerva (2014)
- Ali Kharrazi (2014)
- Timothy Meiri (2015)

10.17 Appendix Q: Tutorial HES-1006 Example: First Order System with PID Control.

Experion2 First Order System with PID

Document Number: HES-1004

Project Name: Honeywell Experion Process Knowledge System for Teaching Purposes

Department: School of Engineering & Information Technology

Focus Area: Industrial Computer Systems Engineering

Product/Process: Honeywell Experion PKS RS430.1

Prepared By:

Document Owner(s)	Project/Organization Role
Simon Godfrey	Engineering Thesis 2016

Project Status Report Version Control

Version	Date	Author	Change Description
1.0	September 2016	Simon Godfrey	Document Created

Introduction

The purpose of this tutorial is to demonstrate the procedure for implementing and configuring a Proportional Integral Derivative (PID) controller on a First Order System (FOS), within the Experion simulation environment. The tutorial will make use of the first order system and associated excel spreadsheet and HMI from the previous tutorial HES-1006.

Upon completion of this tutorial you will be able to:

- Explain the operation of a PID controller;
- Create and configure a PID block to a first order system;
- Test and monitor a PID controller via Experion Monitoring Project Tree, Microsoft Excel and Station HMI.

Proportional Integral Derivative (PID) Controller

A PID controller is an algorithm which calculates the error between a desired setpoint and the current process output. The algorithm is designed to minimise this error by way of changes to the process' manipulated variable, to achieve the desired process setpoint.

Prerequisites

This tutorial assumes basic knowledge of a first order process and conventional feedback control methods. This tutorial also assumes the completion of tutorials listed in the prerequisites. Should you be unsure of any steps, return to appropriate tutorials.

- Security level of ENGR or higher;
- Server(s) have been created and configured to the Experion2 system;
- Completion of tutorials:
  - Tutorials included in HES-0000 (Initial Setup and Configuration);
  - HES-1001 (Creating a Simulated C300 Controller);
  - HES-1003 (Example Mathematical Function);
  - HES-1004 (Example Logic Function);
  - Tutorials included in HES-2000 (Human Machine Interface);
  - HES-1005 (Example Modelling and First Order System).

Additional Resources and Troubleshooting

1. Control Building Users Guide EDPOC-XX19 en 430
2. C300 Controller Users Guide EPDOC-XX11 en 430
3. SIM C300 Users Guide EPDOC-X133 en 430
4. Control Builder Components Reference EPDOC-XX15 en 430
5. Control Builder Components Theory EPDOC-XX16 en 430

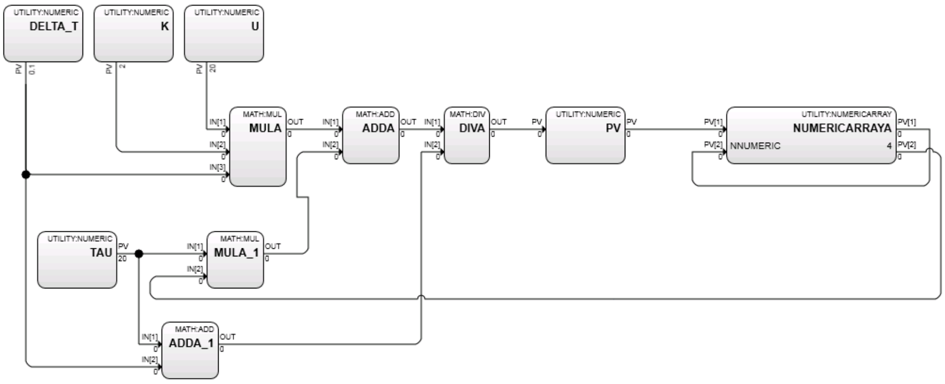
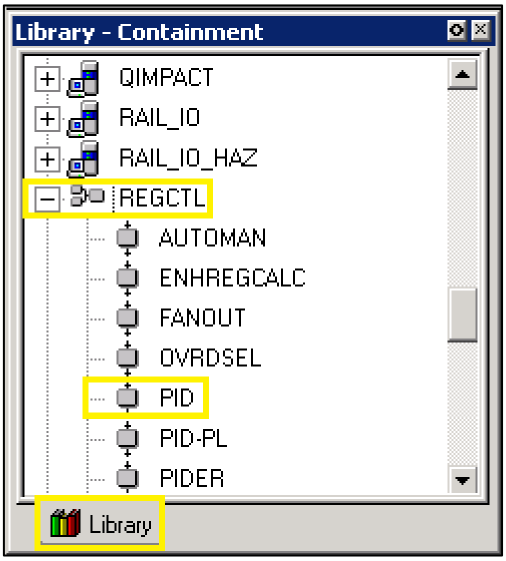
Procedures

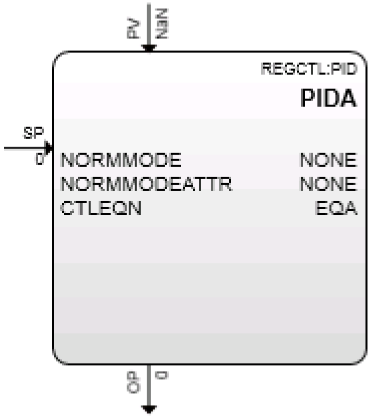
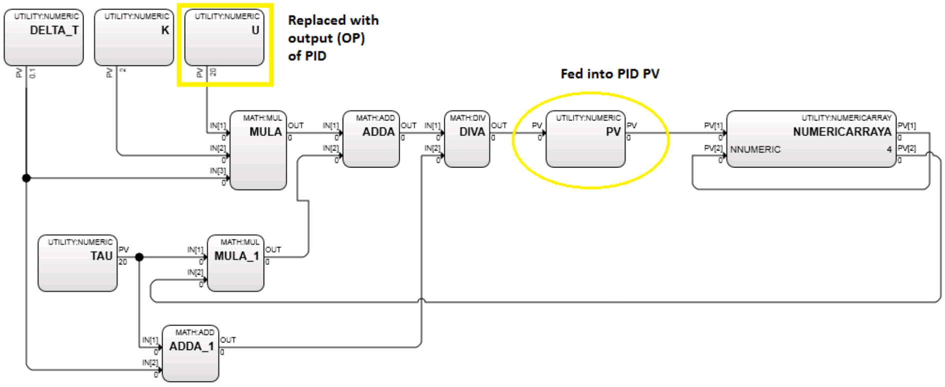
Procedure 1: Program CM and Monitor Dynamically with Experion

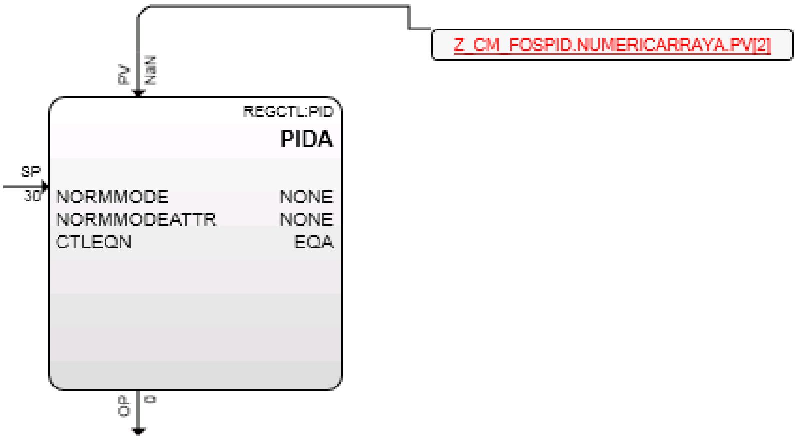
Procedure 2: Read/Write values with Microsoft Excel (MEDE)

Procedure 3: Read/Write Values with Experion HMI (Station)

Procedure 1: Program CM and Monitor Dynamically with Experion

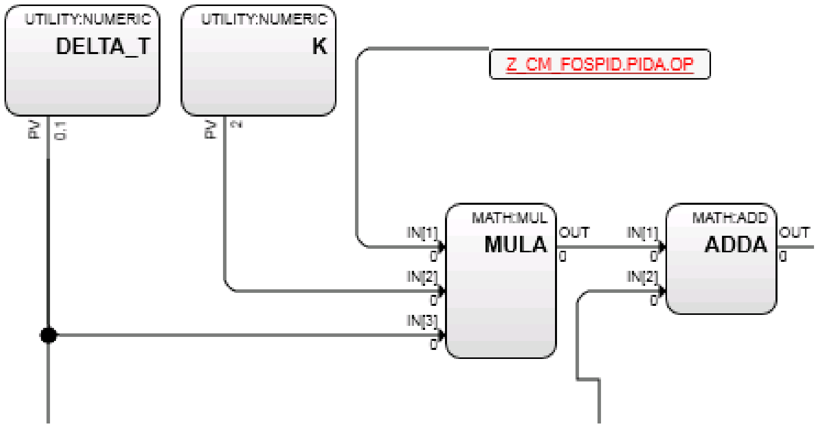
✓	Step	Action
	1	**Create** and **Assign** a **Control Module (CM)** for FOS to your Simulated Controller (See Tutorial HES-1002). For this example, a copy of the first order system CM (Z_FOS - Tutorial HES-1006) was made and renamed Z_FOSPID. It is intended for this CM to contain both the FOS coding as well as the PID controller. The PID can also be built in a separate CM with appropriate links (Param Connectors) between CM function blocks.
	2	**Open** the new CM with the first order system program.  UTILITY:NUMERIC DELTA_T UTILITY:NUMERIC K UTILITY:NUMERIC U PV 0.1 PV 2 PV 20 MATH:MUL MULA MATH:ADD ADDA MATH:DIV DIVA UTILITY:NUMERIC PV UTILITY:NUMERICARRAY NUMERICARRAYA NNUMERIC 4 PV[1] PV[2] IN[1] IN[2] IN[3] OUT UTILITY:NUMERIC TAU PV 20 MATH:MUL MULA_1 MATH:ADD ADDA_1
	3	Navigate to and insert a PID Block into your CM **Function Block Library** > **REGCTL** > **PID**  Library - Containment QIMPACT RAIL_IO RAIL_IO_HAZ REGCTL AUTOMAN ENHREGCALC FANOUT OVRDSEL PID PID-PL PIDER Library

	4	By default, the PID block contains: - 2x Inputs ○ Setpoint (SP) ○ Process Variable (PV) - 1 x Output ○ Operating Point (OP) [MV  PV NEN REGCTL:PID PIDA SP 0 NORMMODE NONE NORMMODEATTR NONE CTLEQN EQA OP 0
	5	The operation of a PID involves the calculation of error in output (PV) and makes adjustments to the process input (MV) to minimise this error. This means: - The output (PV) of the FOS will need to be fed into the input PV of the PID block - The output (OP) of the PID block will become the input to the Process (MV or U) - A setpoint input to the PID block will need to be configured  UTILITY:NUMERIC DELTA_T PV 0.1 UTILITY:NUMERIC K PV 2 UTILITY:NUMERIC U PV 20 MATH:MUL MULA IN[1] IN[2] IN[3] OUT MATH:ADD ADDA MATH:DIV DIVA UTILITY:NUMERIC PV UTILITY:NUMERICARRAY NUMERICARRAYA PV[1] PV[2] NNUMERIC 4 UTILITY:NUMERIC TAU 20 MATH:MUL MULA_1 MATH:ADD ADDA_1 0 Replaced with output (OP) of PID Fed into PID PV

	6	The output from the **Numericarray** PV block can simply be wired to the input PV of the PID block. Alternatively, making use of the **Param Connector** keeps the code tidy and must be used if containing the PID block within a different CM. Upon selecting your point parameter and clicking **OK** the **Param Connector** accesses data from the specified point parameter.  Z_CM_FOSPID.NUMERICARRAYA.PV[2] PV NaN REGCTL:PID PIDA SP 30 NORMMODE NONE NORMMODEATTR NONE CTLEQN EQA OP 0

7

Now the output of the PID (OP) needs to be wired to the input of the FOS. This can be simply wired together or connected by way of a **Param Connector** (as indicated in **Step 6**).



Note: In the example above, the output of the PID was fetched using a **Param Connector**.

8

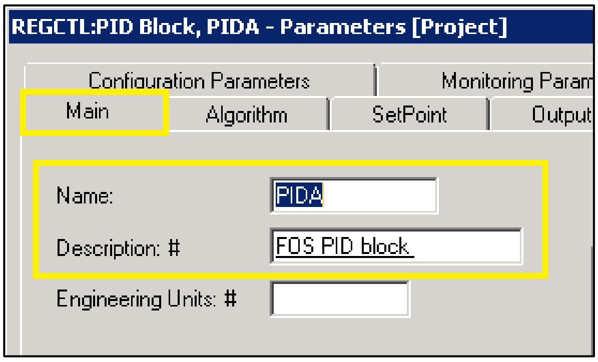
Once all wired together the **Execution Order** must be defined to ensure correct operation of the CM. The following order was selected for the example **Z_CM_FOSPID**:

10	DELTA_T	70	MULA
20	K	80	MULA_1
30	U	90	ADDA
40	TAU	100	DIVA
50	ADDA_1	110	PV
60	PIDA	120	NUMERICARRAY

Execution order must be selected to ensure all mathematical operations can be completed.
Ensure the **Numeric array block** has initialised values (Outlined in Tutorial HES-1005)

9

Enter the PID block **Properties** (**Double-click PID** block).
Under the **Main** tab, provide a unique name and description for the block.


REGCTL:PID Block, PIDA - Parameters [Project]

Name: PIDA

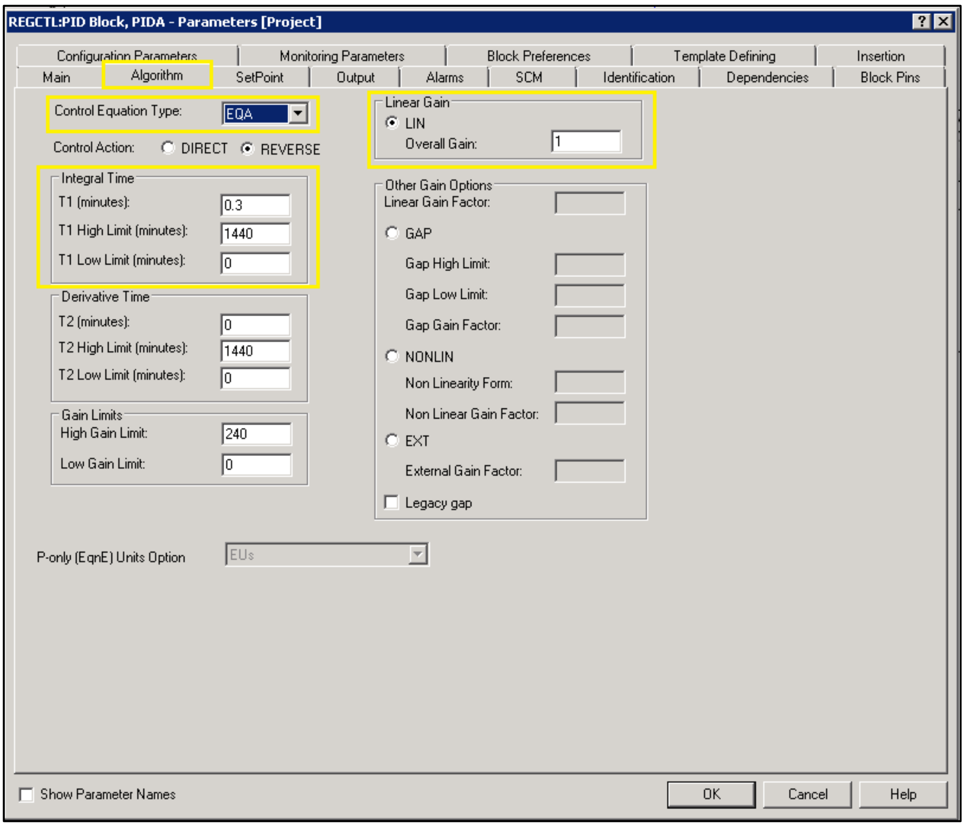
Description: # FOS PID block

Engineering Units: #

10

Within the **Algorithm** tab of the **PID** block **Properties**, set the controller to **Linear Gain** and select a value.

Set a value for the **Integral** and **Derivative** time if desired.



REGCTL:PID Block, PIDA - Parameters [Project]

Control Equation Type: EQA

Control Action: ○ DIRECT ● REVERSE

Integral Time
T1 (minutes): 0.3
T1 High Limit (minutes): 1440
T1 Low Limit (minutes): 0

Derivative Time
T2 (minutes): 0
T2 High Limit (minutes): 1440
T2 Low Limit (minutes): 0

Gain Limits
High Gain Limit: 240
Low Gain Limit: 0

Linear Gain
● LIN
Overall Gain: 1

Other Gain Options
Linear Gain Factor:
○ GAP
Gap High Limit:
Gap Low Limit:
Gap Gain Factor:
○ NONLIN
Non Linearity Form:
Non Linear Gain Factor:
○ EXT
External Gain Factor:
□ Legacy gap

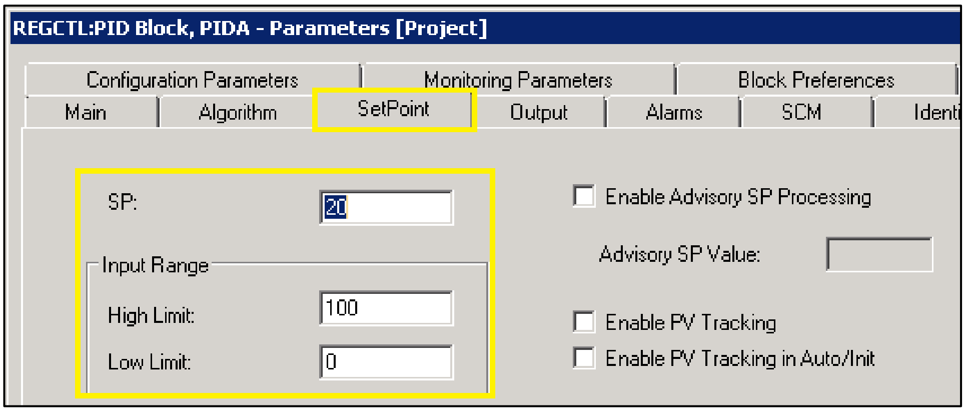
P-only (EqnE) Units Option: EUs

□ Show Parameter Names

Note: For further information on the PID block including the difference in equations refer to the Control Builder Components Reference and Theory documents. (See Additional Resources)

11

Enter the **SetPoint** tab of the **PID** block **Properties** and select a default setpoint value.



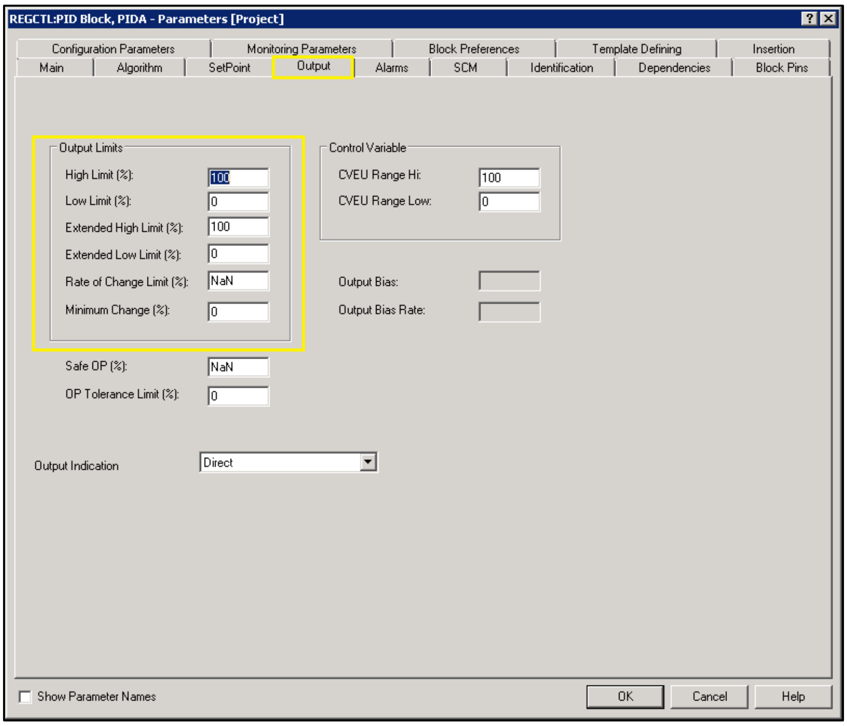
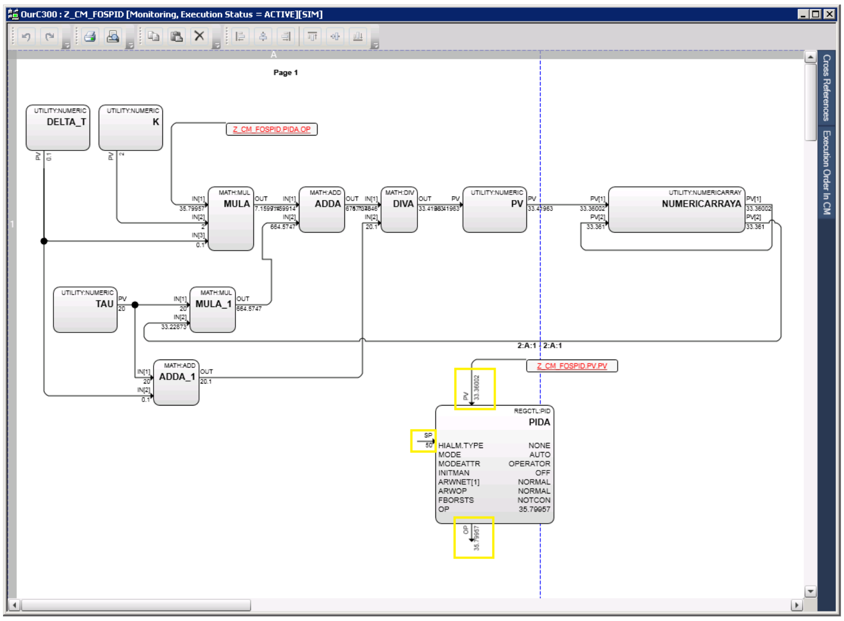
REGCTL:PID Block, PIDA - Parameters [Project]

SP: 20

Input Range
High Limit: 100
Low Limit: 0

□ Enable Advisory SP Processing
Advisory SP Value:
□ Enable PV Tracking
□ Enable PV Tracking in Auto/Init

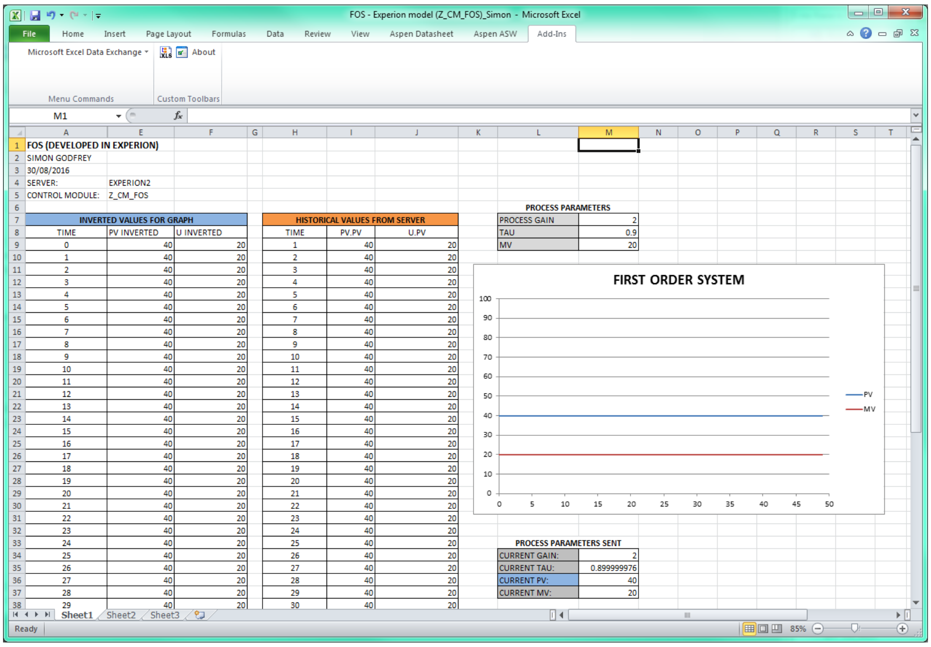
From the same tab is it also possible to set input saturation limits.

	12	Enter the **Output** tab and set output limits of the PID.  REGCTL:PID Block, PIDA - Parameters [Project] Configuration Parameters Monitoring Parameters Block Preferences Template Defining Insertion Main Algorithm SetPoint Output Alarms SCM Identification Dependencies Block Pins Output Limits High Limit (%): 100 Low Limit (%): 0 Extended High Limit (%): 100 Extended Low Limit (%): 0 Rate of Change Limit (%): NaN Minimum Change (%): 0 Control Variable CVEU Range Hi: 100 CVEU Range Low: 0 Output Bias: Output Bias Rate: Safe OP (%): NaN OP Tolerance Limit (%): 0 Output Indication: Direct ☐ Show Parameter Names
	13	**Save** and **Close** the CM
	14	**Load** the CM to the simulated Controller.
	15	From the Project **Monitoring** tab, it is now possible to monitor the output of the system based on changes to the Setpoint and process parameters.  OurC300 : Z_CM_FOSPID [Monitoring, Execution Status = ACTIVE][SIM] Page 1 UTILITY:NUMERIC DELTA_T PV 0.1 UTILITY:NUMERIC K PV 2 Z_CM_FOSPID.PIDA.OP MATH:MUL MULA IN[1] 35.79957 IN[2] 2 IN[3] 0.1 OUT 7.159914 MATH:ADD ADDA IN[1] 7.159914 IN[2] 664.5747 OUT 671.7346 MATH:DIV DIVA IN[1] 671.7346 IN[2] 20.1 OUT 33.41963 UTILITY:NUMERIC PV PV 33.41963 UTILITY:NUMERICARRAY NUMERICARRAYA PV[1] 33.36002 PV[2] 33.361 UTILITY:NUMERIC TAU PV 20 MATH:MUL MULA_1 IN[1] 20 IN[2] 33.22873 OUT 664.5747 MATH:ADD ADDA_1 IN[1] 20 IN[2] 0.1 OUT 20.1 2:A:1 2:A:1 Z_CM_FOSPID.PV.PV PV 33.36002 SP 50 REGCTL:PID PIDA HIALM.TYPE NONE MODE AUTO MODEATTR OPERATOR INITMAN OFF ARWNET[1] NORMAL ARWOP NORMAL FBORSTS NOTCON OP 35.79957 OP 35.79957 Cross References Execution Order In CM

Procedure 2: Read/Write values with Microsoft Excel (MEDE)

This tutorial assumes prior experience with Microsoft Excel Data Exchange. The program builds upon the excel program developed for the monitoring of the FOS created in Tutorial HES-1006

✓	Step	Action
	1	Open Microsoft Excel and ensure the MSEDE Add-in has been installed.
	2	Open the previously developed Excel program for the FOS (FOS with PID)



FOS (DEVELOPED IN EXPERION)
SIMON GODFREY
30/08/2016
SERVER: EXPERION2
CONTROL MODULE: Z_CM_FOS

INVERTED VALUES FOR GRAPH			HISTORICAL VALUES FROM SERVER		
TIME	PV INVERTED	U INVERTED	TIME	PV.PV	U.PV
0	40	20	1	40	20
1	40	20	2	40	20
2	40	20	3	40	20
3	40	20	4	40	20
4	40	20	5	40	20
5	40	20	6	40	20
6	40	20	7	40	20
7	40	20	8	40	20
8	40	20	9	40	20
9	40	20	10	40	20
10	40	20	11	40	20
11	40	20	12	40	20
12	40	20	13	40	20
13	40	20	14	40	20
14	40	20	15	40	20
15	40	20	16	40	20
16	40	20	17	40	20
17	40	20	18	40	20
18	40	20	19	40	20
19	40	20	20	40	20
20	40	20	21	40	20
21	40	20	22	40	20
22	40	20	23	40	20
23	40	20	24	40	20
24	40	20	25	40	20
25	40	20	26	40	20
26	40	20	27	40	20
27	40	20	28	40	20
28	40	20	29	40	20
29	40	20	30	40	20

PROCESS PARAMETERS

PROCESS GAIN	2
TAU	0.9
MV	20

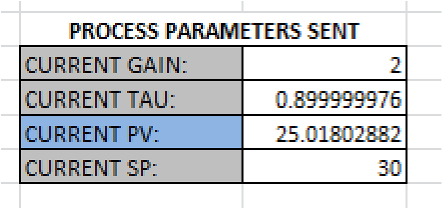
PROCESS PARAMETERS SENT

CURRENT GAIN:	2
CURRENT TAU:	0.899999976
CURRENT PV:	40
CURRENT MV:	20

✓	Step	Action
	3	Add additional columns for SP to both of the tables: - Historical Values from the server - Inverted values for graph Using the Wizard (or formula), acquire the fast history snapshots of the SP from the new CM (Z_CM_FOSPID)
	4	Alter the formulas of each of the historical columns to display: - Setpoint (PIDA.SP) - Manipulated Vairable (PIDA.OP) - Process Variable (PV.PV) **{=GetHistValArray_Offset(1,"EXPERION2","Z_CM_FOSPID","PIDA.SP","H5SF",1,50,"V","")}** **{=GetHistValArray_Offset(1,"experion2","Z_CM_FOSPID","PIDA.OP","H5SF",1,50,"V","")}** **{=GetHistValArray_Offset(1,"experion2","Z_CM_FOSPID","PV.PV","H5SF",1,50,"V","")**

5

Alter the formulas where new process parameters are being sent to the server.



PROCESS PARAMETERS SENT	
CURRENT GAIN:	2
CURRENT TAU:	0.899999976
CURRENT PV:	25.01802882
CURRENT SP:	30

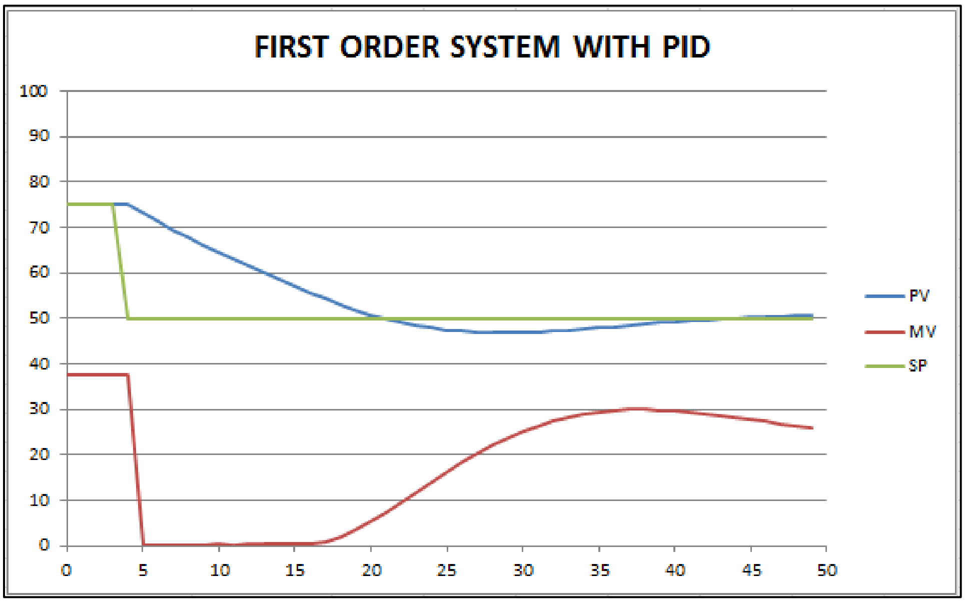
Note: Difficulties were experienced trying to directly send setpoint values to the PID block. This can be overcome by setting the PID block mode or CAS (Cascade) and feeding a numeric to the PID's SP input pin. It is then possible to send numerical values via excel to the Numeric block.

6

Add the Setpoint data to the plot.

7

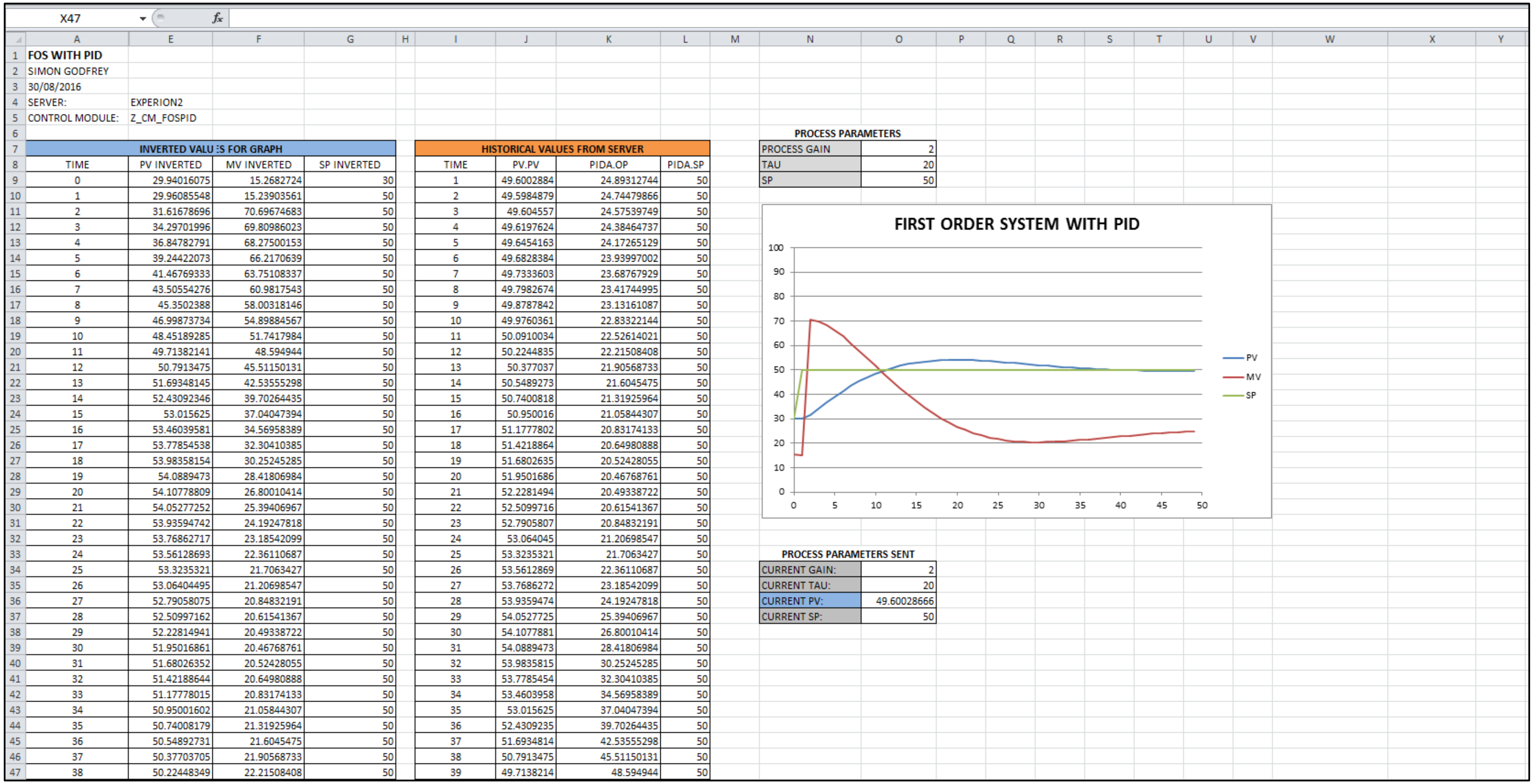
Upon setting the re-calculation interval of MSEDE to 1s, it is now possible to dynamically change the process parameters and view a rolling graph of the process response.



Note: The above controller was tuned with the following parameters:
Controller Gain: 2.4
Integral Time: 0.08

8

See the following page for a full screen view of the Excel Spreadsheet.



FOS WITH PID
SIMON GODFREY
30/08/2016
SERVER: EXPERION2
CONTROL MODULE: Z_CM_FOSPID

INVERTED VALU :S FOR GRAPH				HISTORICAL VALUES FROM SERVER			
TIME	PV INVERTED	MV INVERTED	SP INVERTED	TIME	PV.PV	PIDA.OP	PIDA.SP
0	29.94016075	15.2682724	30	1	49.6002884	24.89312744	50
1	29.96085548	15.23903561	50	2	49.5984879	24.74479866	50
2	31.61678696	70.69674683	50	3	49.604557	24.57539749	50
3	34.29701996	69.80986023	50	4	49.6197624	24.38464737	50
4	36.84782791	68.27500153	50	5	49.6454163	24.17265129	50
5	39.24422073	66.2170639	50	6	49.6828384	23.93997002	50
6	41.46769333	63.75108337	50	7	49.7333603	23.68767929	50
7	43.50554276	60.9817543	50	8	49.7982674	23.41744995	50
8	45.3502388	58.00318146	50	9	49.8787842	23.13161087	50
9	46.99873734	54.89884567	50	10	49.9760361	22.83322144	50
10	48.45189285	51.7417984	50	11	50.0910034	22.52614021	50
11	49.71382141	48.594944	50	12	50.2244835	22.21508408	50
12	50.7913475	45.51150131	50	13	50.377037	21.90568733	50
13	51.69348145	42.53555298	50	14	50.5489273	21.6045475	50
14	52.43092346	39.70264435	50	15	50.7400818	21.31925964	50
15	53.015625	37.04047394	50	16	50.950016	21.05844307	50
16	53.46039581	34.56958389	50	17	51.1777802	20.83174133	50
17	53.77854538	32.30410385	50	18	51.4218864	20.64980888	50
18	53.98358154	30.25245285	50	19	51.6802635	20.52428055	50
19	54.0889473	28.41806984	50	20	51.9501686	20.46768761	50
20	54.10778809	26.80010414	50	21	52.2281494	20.49338722	50
21	54.05277252	25.39406967	50	22	52.5099716	20.61541367	50
22	53.93594742	24.19247818	50	23	52.7905807	20.84832191	50
23	53.76862717	23.18542099	50	24	53.064045	21.20698547	50
24	53.56128693	22.36110687	50	25	53.3235321	21.7063427	50
25	53.3235321	21.7063427	50	26	53.5612869	22.36110687	50
26	53.06404495	21.20698547	50	27	53.7686272	23.18542099	50
27	52.79058075	20.84832191	50	28	53.9359474	24.19247818	50
28	52.50997162	20.61541367	50	29	54.0527725	25.39406967	50
29	52.22814941	20.49338722	50	30	54.1077881	26.80010414	50
30	51.95016861	20.46768761	50	31	54.0889473	28.41806984	50
31	51.68026352	20.52428055	50	32	53.9835815	30.25245285	50
32	51.42188644	20.64980888	50	33	53.7785454	32.30410385	50
33	51.17778015	20.83174133	50	34	53.4603958	34.56958389	50
34	50.95001602	21.05844307	50	35	53.015625	37.04047394	50
35	50.74008179	21.31925964	50	36	52.4309235	39.70264435	50
36	50.54892731	21.6045475	50	37	51.6934814	42.53555298	50
37	50.37703705	21.90568733	50	38	50.7913475	45.51150131	50
38	50.22448349	22.21508408	50	39	49.7138214	48.594944	50

PROCESS PARAMETERS

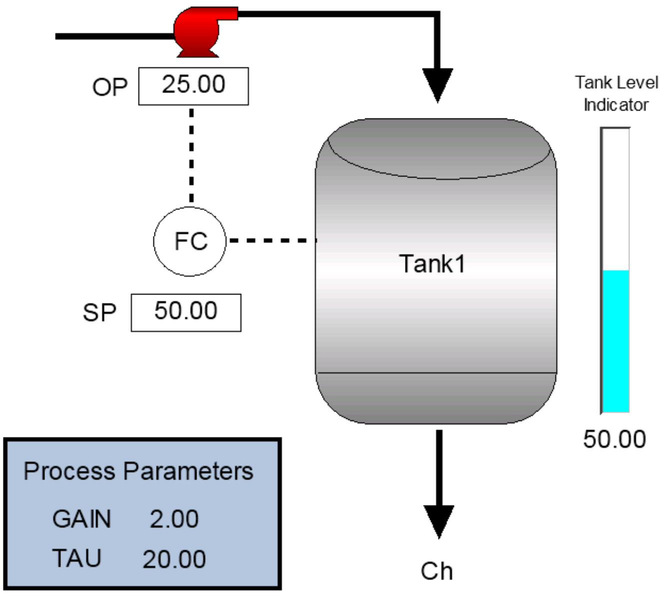
PROCESS GAIN	2
TAU	20
SP	50

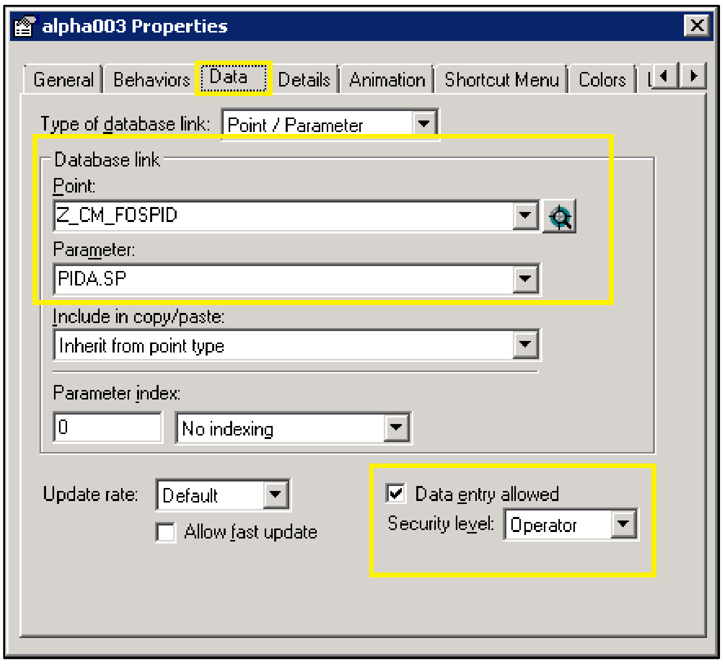
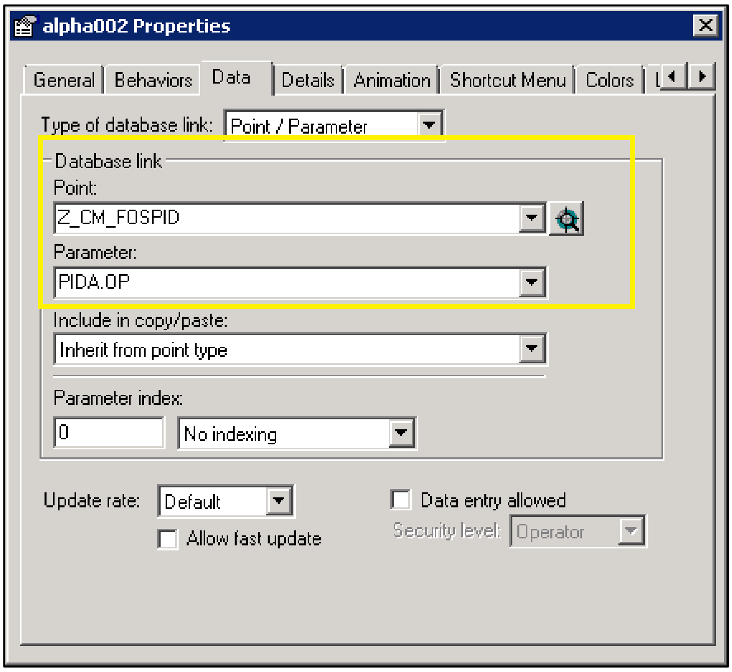
PROCESS PARAMETERS SENT

CURRENT GAIN:	2
CURRENT TAU:	20
CURRENT PV:	49.60028666
CURRENT SP:	50

Procedure 3: Read/Write Values with Experion HMI (Station)

This tutorial assumes prior experience with HMIWeb Display Builder. The program builds upon the HMI developed for the monitoring of the FOS created in Tutorial HES-1006

✓	Step	Action
	1	Open HMIWeb Display Builder and open the HMI developed from Tutorial HES-1006 (Z_FOS)
	2	Save the display with a new unique name for future changes (Z_FOSPID)
	3	Alter the **point parameters** for each of the **Process Parameters** as well as the **Tank level Indicator** ensuring the correct CM (**Z_CM_FOSPID**) is used: - Gain (K.PV) - Tau (TAU.PV) - Tank Level Indicator (PV.PV) - Tank Level Alphanumeric (PV.PV)
	4	Using various shapes and techniques outlined in previous Tutorials, develop: **Flow Controller** - SP (Setpoint) Text box - SP (Setpoint) Alphanumeric - Electrical Lines - **Pump** - Alter the text box of the pump to OP (Operating Point)  OP 25.00 FC SP 50.00 Tank1 Tank Level Indicator 50.00 Process Parameters GAIN 2.00 TAU 20.00 Ch

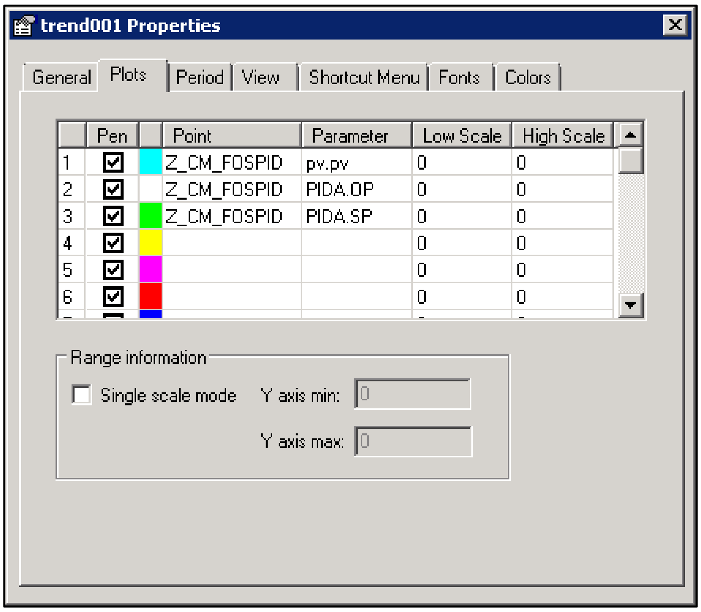
	5	**Setpoint Alphanumeric** Enter the **Properties** and set the **Point Parameters** under the **Data** tab.  alpha003 Properties General Behaviors Data Details Animation Shortcut Menu Colors Type of database link: Point / Parameter Database link Point: Z_CM_FOSPID Parameter: PIDA.SP Include in copy/paste: Inherit from point type Parameter index: 0 No indexing Update rate: Default ☐ Allow fast update ☑ Data entry allowed Security level: Operator Note: Ensure you have ticked the **Data entry allowed** checkbox to allow the setpoint to be altered from the HMI.
	6	**Pump Operating Point Alphanumeric** Enter the P**roperties** and set the **Point Parameters** under the **Data** tab.  alpha002 Properties General Behaviors Data Details Animation Shortcut Menu Colors Type of database link: Point / Parameter Database link Point: Z_CM_FOSPID Parameter: PIDA.OP Include in copy/paste: Inherit from point type Parameter index: 0 No indexing Update rate: Default ☐ Allow fast update ☐ Data entry allowed Security level: Operator

7

Trend

Alter the Plots included in the trend to include:

- Process Variable (PV.PV)
- Manipulated Variable - PID Operating Point (PIDA.OP)
- Setpoint (PIDA.SP)



	Pen		Point	Parameter	Low Scale	High Scale
1	☑	🟦	Z_CM_FOSPID	pv.pv	0	0
2	☑	⬜	Z_CM_FOSPID	PIDA.OP	0	0
3	☑	🟩	Z_CM_FOSPID	PIDA.SP	0	0
4	☑	🟨			0	0
5	☑	🟪			0	0
6	☑	🟥			0	0

Range information

☐ Single scale mode Y axis min: 0

Y axis max: 0

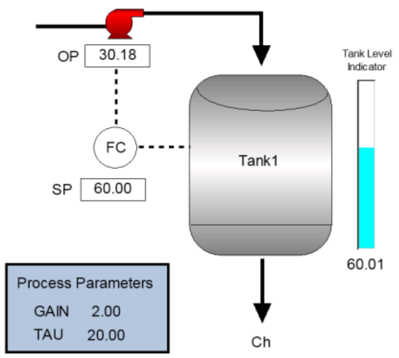
Note: Ensure that Setpoint (PIDA.SP) has been included in the **Server History** of the CM.

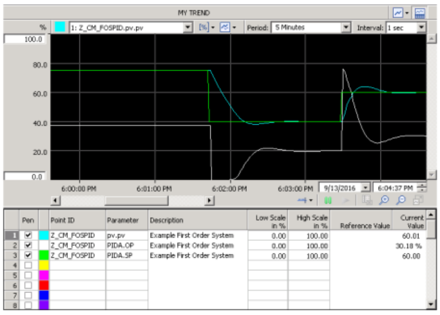
8

Group any objects together and **Save** the display (Take note of the display page name). Ensure the save location is accessible via Station. See Tutorial HES-4001 for configuring the connection properties of Station with created HMI's.

9

It is now possible to monitor and control the process via the display page in Station.





	Pen		Point ID	Parameter	Description	Low Scale in %	High Scale in %	Reference Value	Current Value
1	☑	🟦	Z_CM_FOSPID	pv.pv	Example First Order System	0.00	100.00		60.01
2	☑	⬜	Z_CM_FOSPID	PIDA.OP	Example First Order System	0.00	100.00		30.18 %
3	☑	🟩	Z_CM_FOSPID	PIDA.SP	Example First Order System	0.00	100.00		60.00
4	☐	🟨							
5	☐	🟪							
6	☐	🟥							
7	☐	🟦							
8	☐	🟪							

10.18 Appendix R: Tutorial HES-5007 OPC Data Exchange using LabVIEW

Murdoch
UNIVERSITY

Experion2 OPC LabVIEW Configuration and Testing

Document Number: HES-5006

Project Name:	Honeywell Experion Process Knowledge System for Teaching Purposes
Department:	School of Engineering & Information Technology
Focus Area:	Industrial Computer Systems Engineering
Product/Process:	Honeywell Experion PKS RS430.1

Prepared By

Document Owner(s)	Project/Organization Role
Simon Godfrey	Engineering Thesis – Semester 2 - 2016

Project Status Report Version

Version	Date	Author	Change Description
1.0	October 2016	Simon Godfrey	Document Created

Introduction

The purpose of this tutorial is to demonstrate the procedure for using National Instruments' LabVIEW software application to access Experion server data via OPC. Upon completion of this tutorial, you will be able to:

- Create a server connection to Experion2;
- Monitor and control point parameters on the Experion server;
- Implement a simple feedback PID control strategy.

LabVIEW

Laboratory Virtual Instrument Engineering Workbench (LabVIEW) is an integrated development environment utilising a graphical programming language for the creation of measurement and control systems. LabVIEW offers an open architecture enabling the interoperability with other software platforms and hardware devices. This interoperability of software platforms includes data exchange between Experion and LabVIEW via OPC communication.

For the effective monitoring and control of Experion server items, a real-time Shared Variable Engine (SVE) was created. The SVE allows provides the ability the bind shared variables to I/O servers, such as the OPC server.

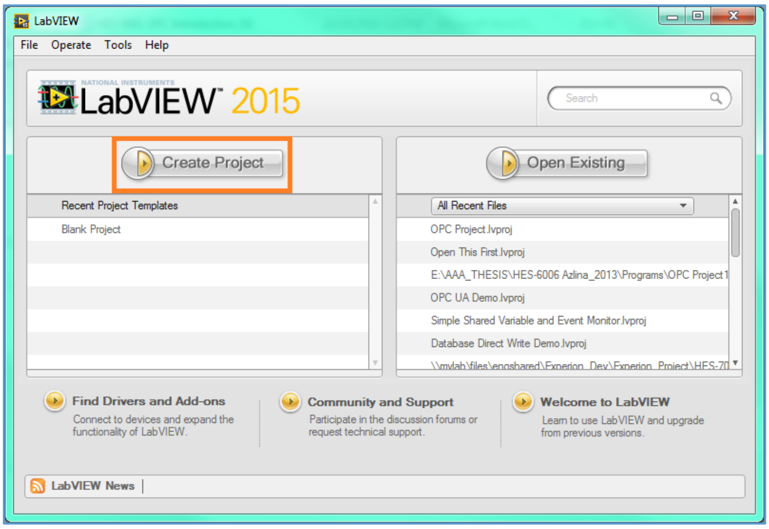
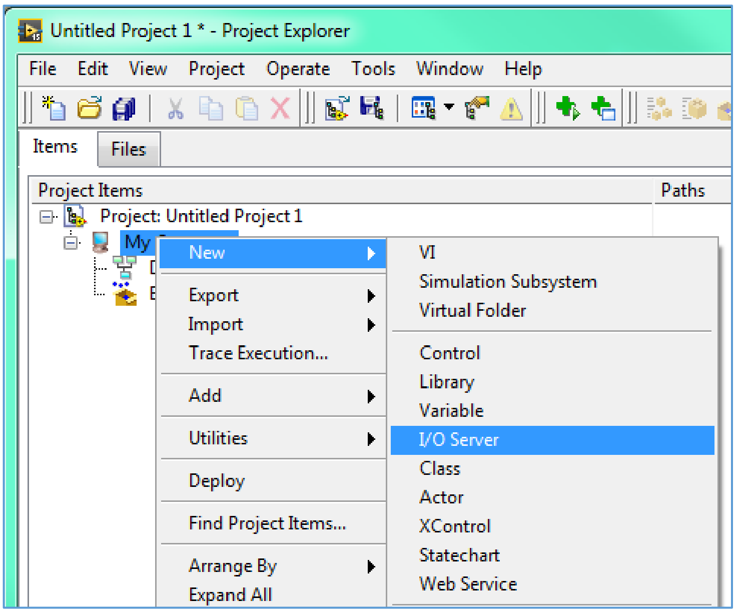
Prerequisites

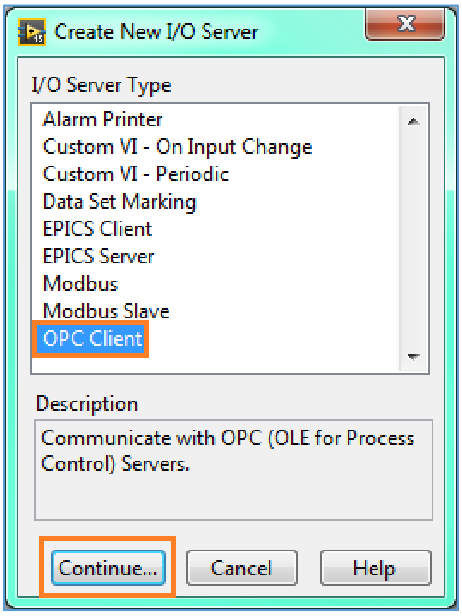
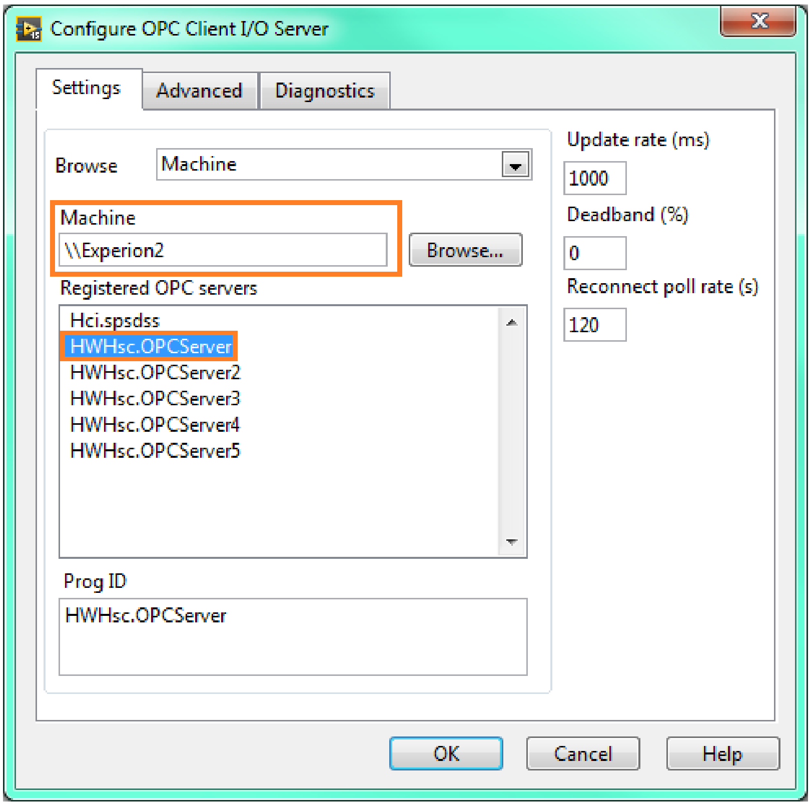
- Security level of ENGR or higher
- Server(s) have been created and configured to the Experion2 system.
- Completion of tutorials contained within:
  - HES-0000
  - HES-1000
  - HES-2000
  - HES-3000
  - HES-4000

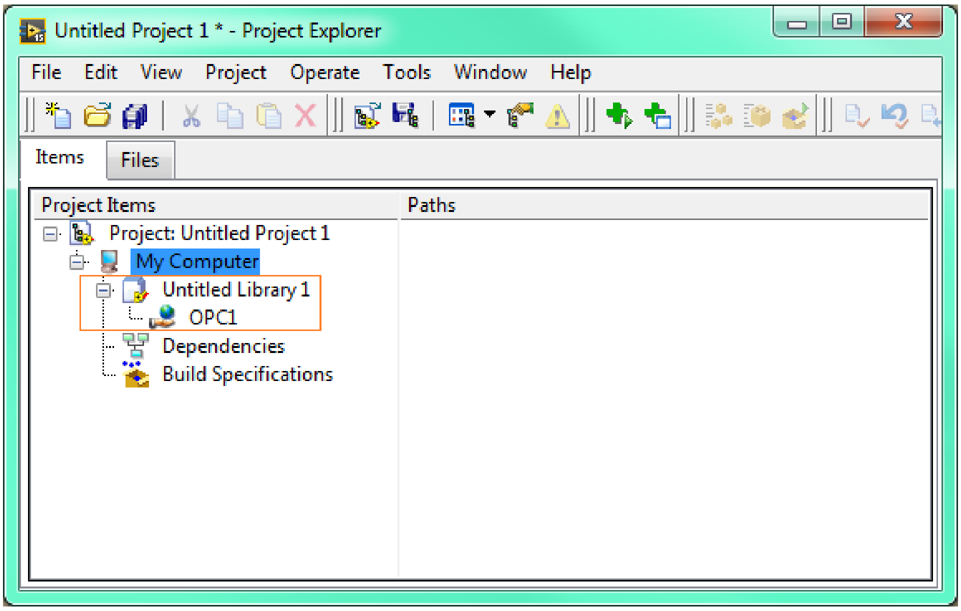
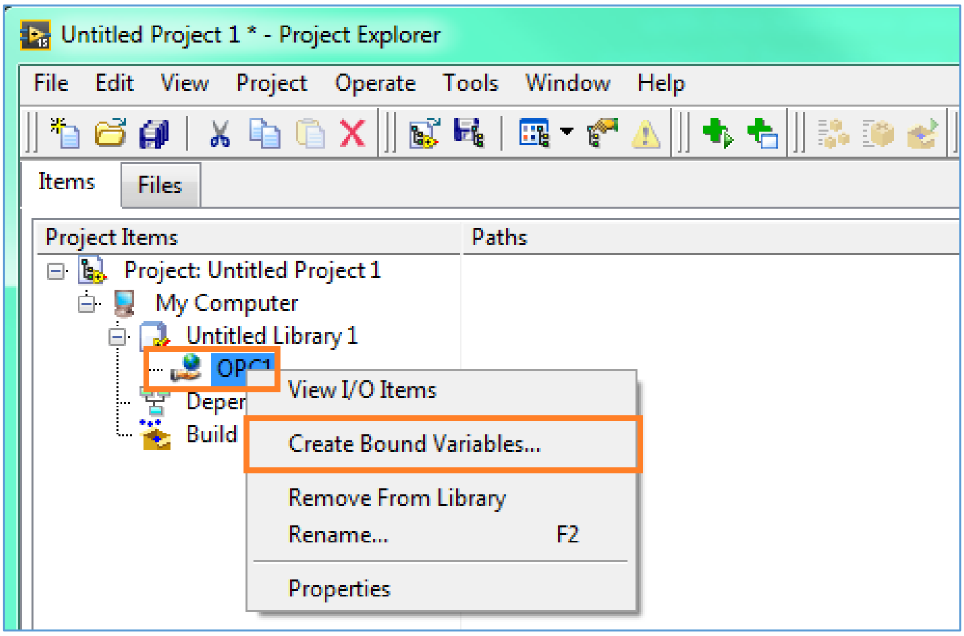
Additional Resources and Troubleshooting

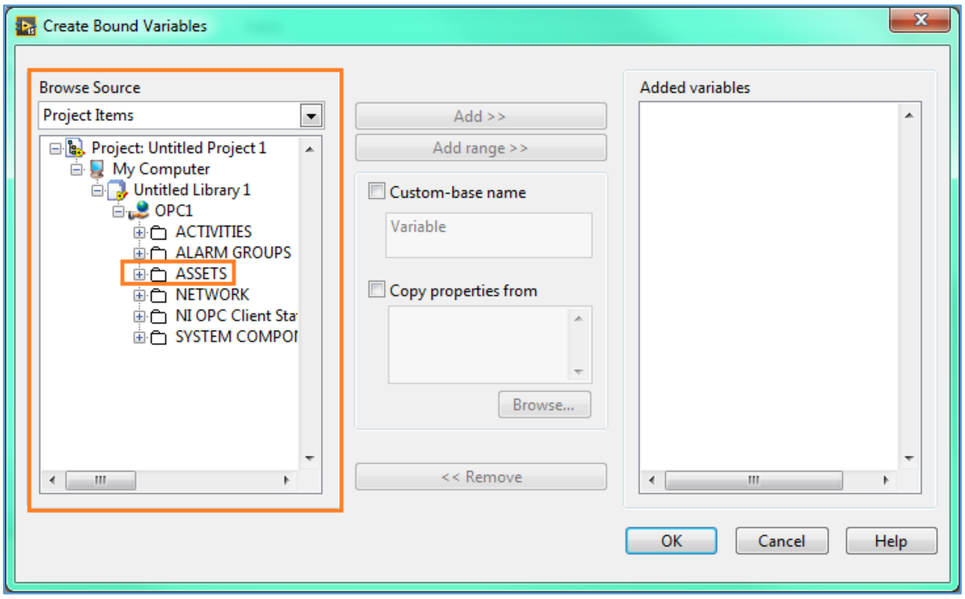
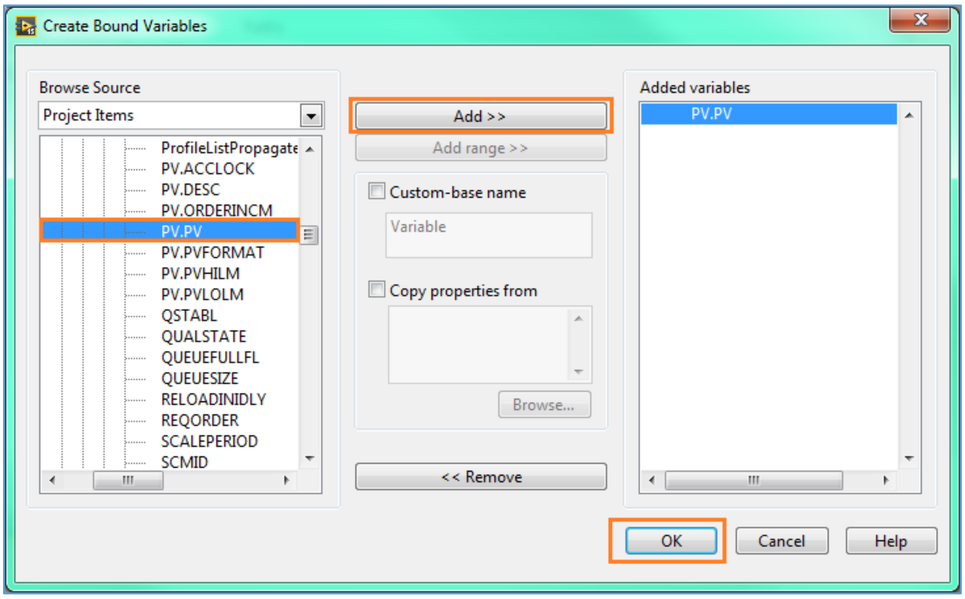
1. OPC Client Interface Reference EPDOC-XX78 en 430
2. OPC Gateway for ACE Interface Reference EPDOC-XX79 en 430
3. LabVIEW online help documentation.

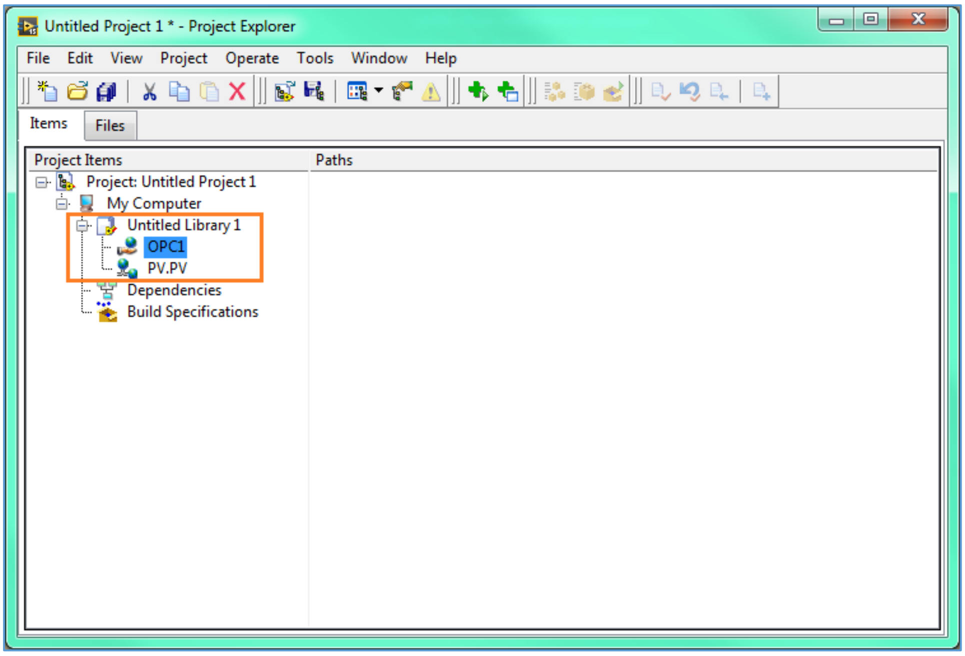
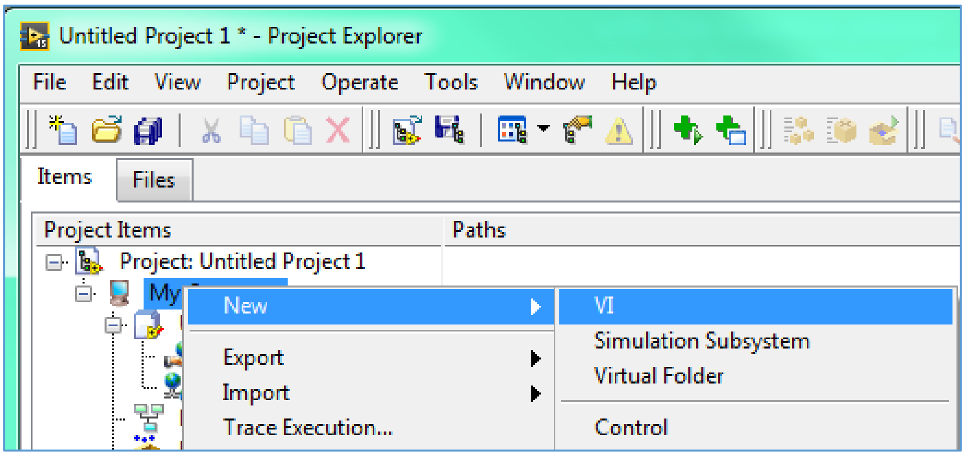
Procedure

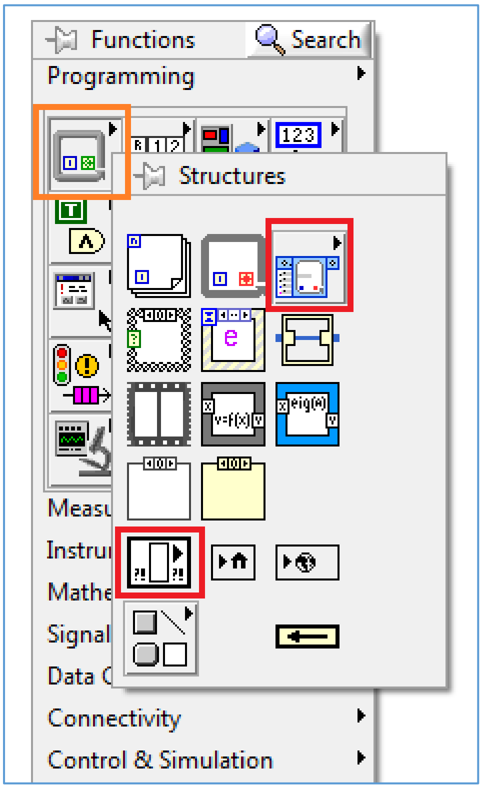
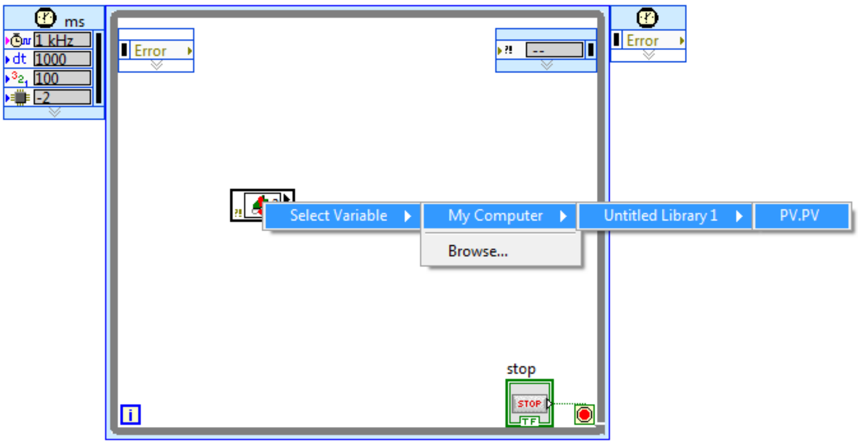
✓	Step	Action
	1	Open **LabVIEW** **Start** > **All Programs** > **National Instruments** > **LabVIEW 32-bit** Note: You must use LabVIEW 32-bit (Not 64-bit).
	2	Create a **blank Project**  LabVIEW File Operate Tools Help LabVIEW 2015 Search Create Project Open Existing Recent Project Templates Blank Project All Recent Files OPC Project.lvproj Open This First.lvproj E:\AAA_THESIS\HES-6006 Azlina_2013\Programs\OPC Project1 OPC UA Demo.lvproj Simple Shared Variable and Event Monitor.lvproj Database Direct Write Demo.lvproj Find Drivers and Add-ons Connect to devices and expand the functionality of LabVIEW. Community and Support Participate in the discussion forums or request technical support. Welcome to LabVIEW Learn to use LabVIEW and upgrade from previous versions. LabVIEW News This will open a new window.
	3	Create an OPC I/O Server by right-clicking **My Computer** and selecting **New** > **I/O Server**  Untitled Project 1 * - Project Explorer File Edit View Project Operate Tools Window Help Items Files Project Items Paths Project: Untitled Project 1 My New Export Import Trace Execution... Add Utilities Deploy Find Project Items... Arrange By Expand All VI Simulation Subsystem Virtual Folder Control Library Variable I/O Server Class Actor XControl Statechart Web Service

	4	Select **OPC Client** and click **Continue**  Create New I/O Server I/O Server Type Alarm Printer Custom VI - On Input Change Custom VI - Periodic Data Set Marking EPICS Client EPICS Server Modbus Modbus Slave OPC Client Continue... Cancel Help Description Communicate with OPC (OLE for Process Control) Servers. This will open a new window for configuring the Client I/O OPC server.
	5	Enter **Experion2** into the **Machine** and then select the appropriate OPC server.  Configure OPC Client I/O Server Settings Advanced Diagnostics Browse... Registered OPC servers Hci.spsdss HWHsc.OPCServer HWHsc.OPCServer2 HWHsc.OPCServer3 HWHsc.OPCServer4 HWHsc.OPCServer5 OK Cancel Help Browse: Machine Machine: \\Experion2 Prog ID: HWHsc.OPCServer Update rate (ms): 1000 Deadband (%): 0 Reconnect poll rate (s): 120

	6	The **I/O Server** is now present within the project tree.  Untitled Project 1 * - Project Explorer File Edit View Project Operate Tools Window Help Items Files Project Items Paths Project: Untitled Project 1 My Computer Untitled Library 1 OPC1 Dependencies Build Specifications
	7	It is now possible to create **Bound Variables** from the Experion Server for monitoring and control. **Right-click** the OPC I/O Server and select **Create Bound Variables**  Untitled Project 1 * - Project Explorer File Edit View Project Operate Tools Window Help Items Files Project Items Paths Project: Untitled Project 1 My Computer Untitled Library 1 OPC1 Deper Build View I/O Items Create Bound Variables... Remove From Library Rename... F2 Properties This will open a new window.

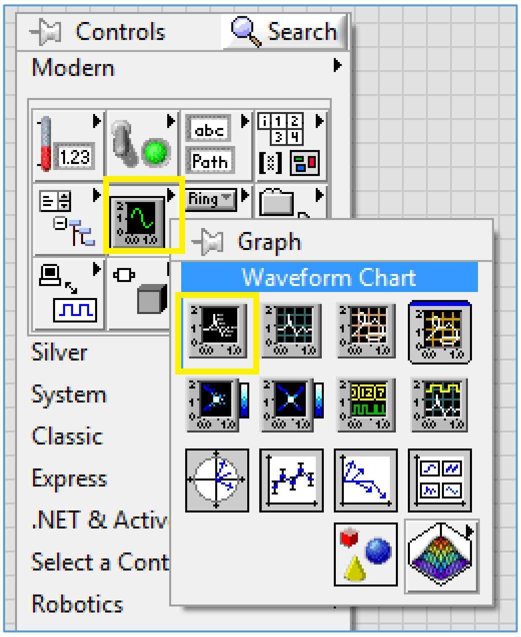
	8	It is now possible to navigate through to the appropriate item (point parameters) you wish to monitor/control. 
	9	Once you have navigated to your item, select the item and click **Add >>** followed by **OK**. For the example below, the item PV.PV from the Control Module Z_CM_FOS was monitored. **ASSETS** > **SIM_ASSET_ITEM** > **ZZZ_SIMON_ITEM** > **Z_CM_FOS** > **PV.PV** 

	10	The newly created **bound variable** will now be visible in the Project's variable library  Untitled Project 1 * - Project Explorer File Edit View Project Operate Tools Window Help Items Files Project Items Paths Project: Untitled Project 1 My Computer Untitled Library 1 OPC1 PV.PV Dependencies Build Specifications This variable can now be accessed within a **LabVIEW** VI by way of a **Shared Variable node.**
	11	Create a new VI by **right-clicking** the **My Computer** and selecting **New** > **VI**  Untitled Project 1 * - Project Explorer File Edit View Project Operate Tools Window Help Items Files Project Items Paths Project: Untitled Project 1 My New Export Import Trace Execution... VI Simulation Subsystem Virtual Folder Control This will open 2 new windows. - 1 x Block Diagram - 1 x Front Panel

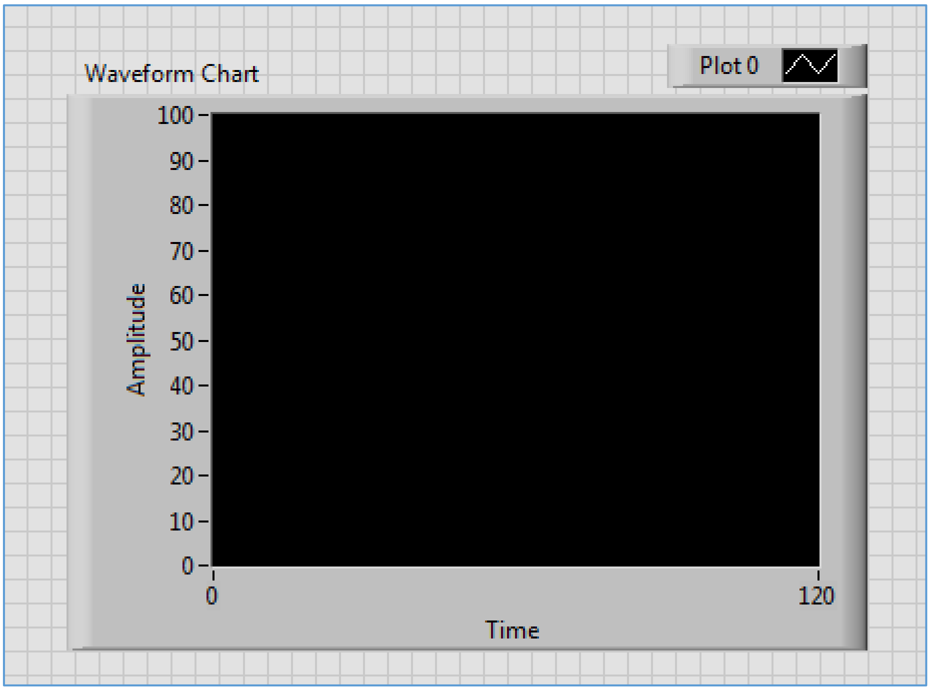
	12	Within the **Block Diagram** create a **Timed Loop** and a **Shared Variable**.  Functions Search Programming 123 Structures Measu Instru Mathe Signal Data C Connectivity Control & Simulation
	13	The Timed loop will default to executing every 1000ms (keep these settings) Select the appropriate **Shared Variable** from the icon.  ms 1 kHz dt 1000 100 -2 Error Error Select Variable My Computer Untitled Library 1 PV.PV Browse... stop STOP

14

Create a Waveform Chart in the Front Panel Window.
Navigate to the Wavechart.

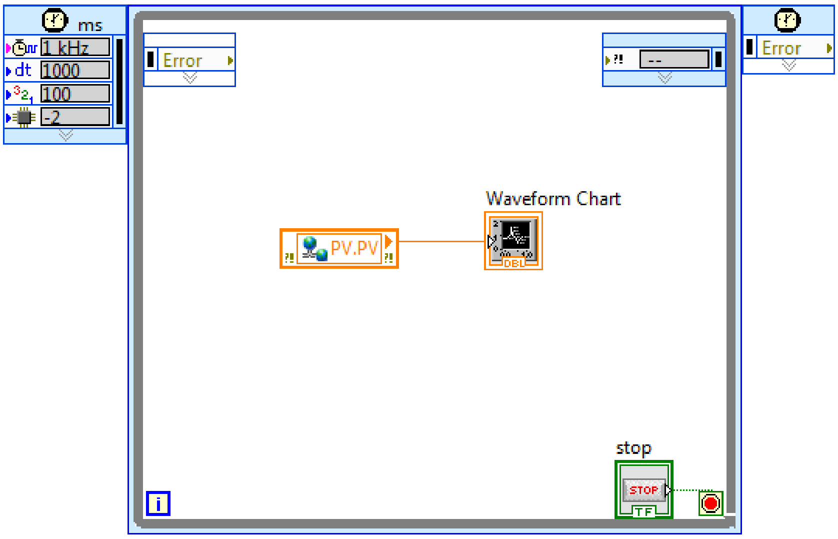


Set the Y-axis and X-axis limits appropiately.

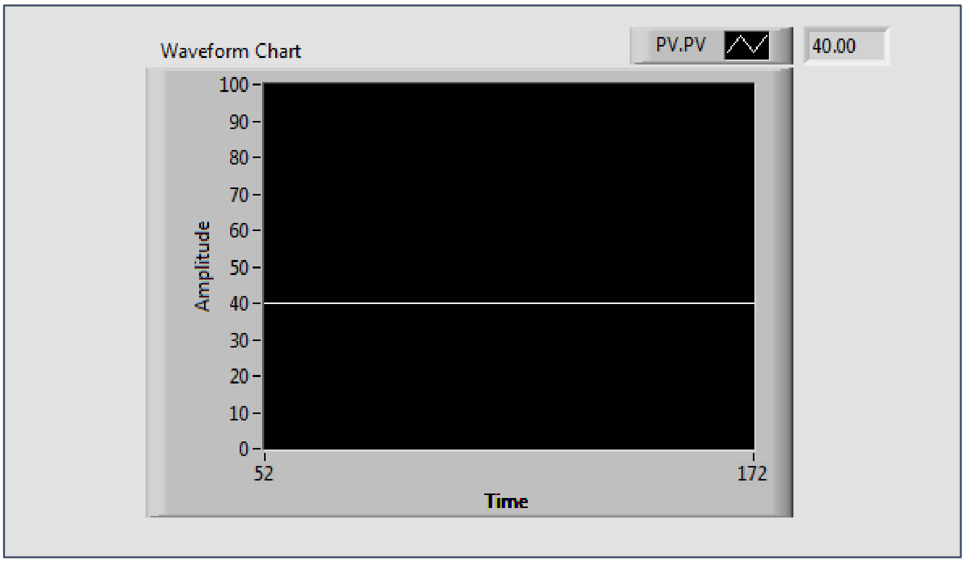


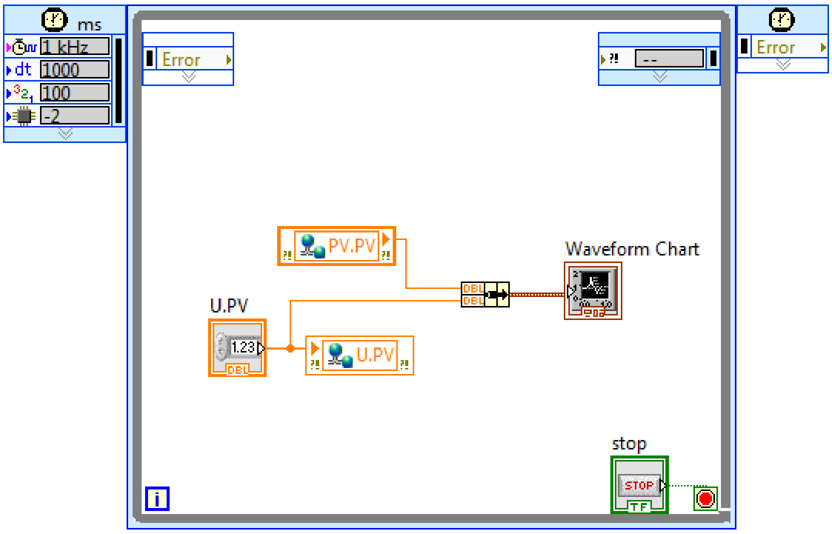
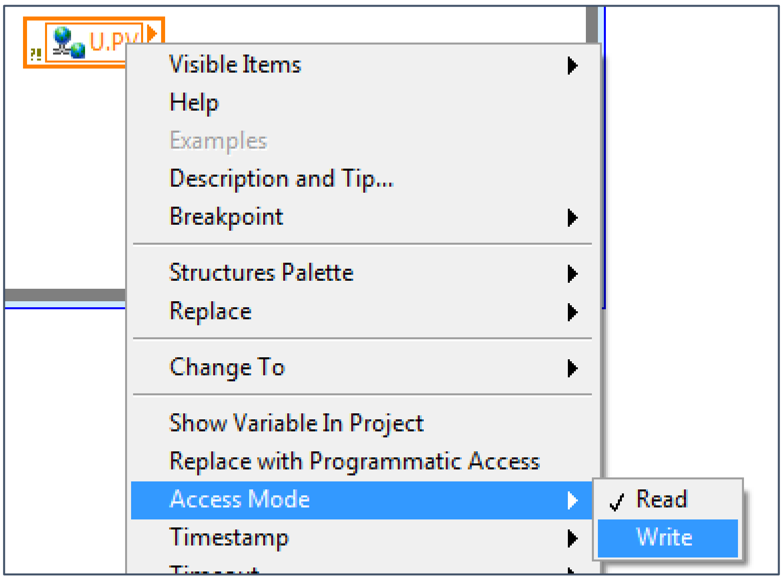
15

Wire the **Shared Variable** to the Waveform chart in the Block Diagram window



It is now possible to run the simulation and view the output of the item on the chart in real time.

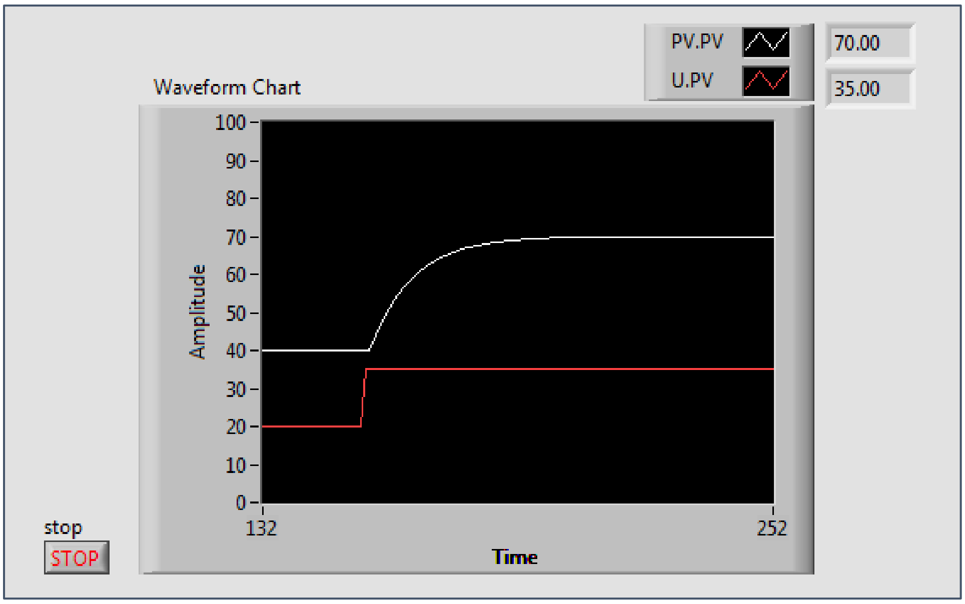


	16	Following the same procedure for adding a Shared Variable, we can add a second varibale for controlling the input (U.PV) to the First Order System contained within the Experion Control Module.  ms 1 kHz dt 1000 100 -2 Error PV.PV Waveform Chart DBL U.PV 1.23 U.PV stop STOP TF i Error Note: To change the Shared Varable to a Write Block, **Right-Click** the block and alter the access mode as shown below.  U.PV Visible Items Help Examples Description and Tip... Breakpoint Structures Palette Replace Change To Show Variable In Project Replace with Programmatic Access Access Mode Timestamp Timeout ✓ Read Write

17

The results of a change in input to the first order system are shown below.

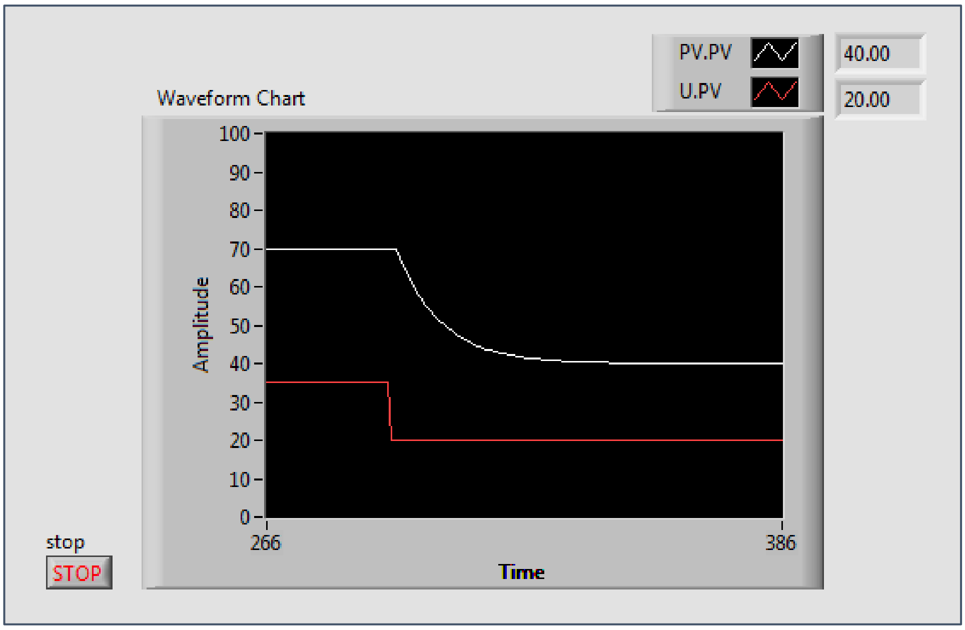
STEP INPUT 20 → 35



PV.PV 70.00

U.PV 35.00

STEP INPUT 35 → 20

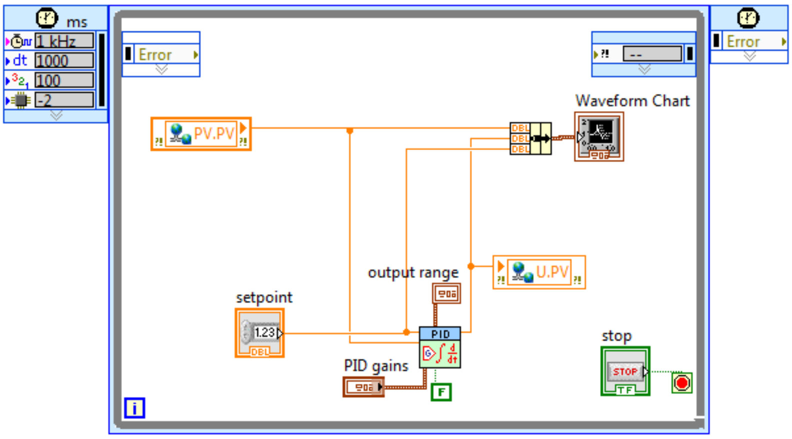


PV.PV 40.00

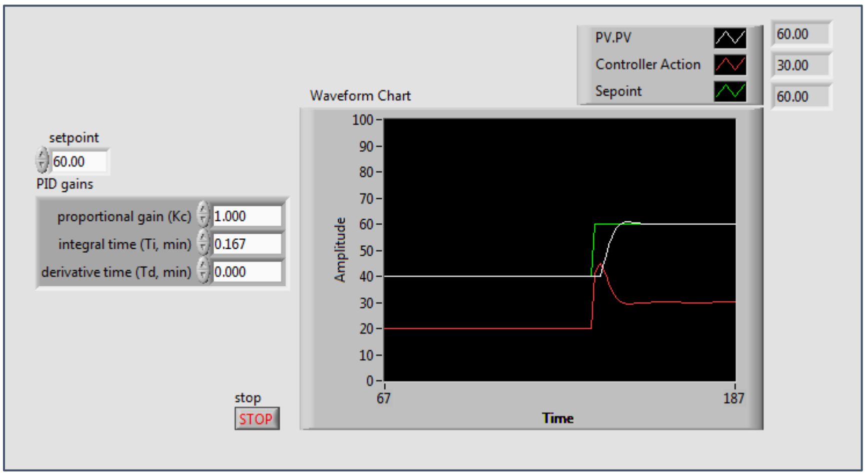
U.PV 20.00

18

It is also now possible to add a variety of control strategies to the system. As an example, a conventional feedback PID controller is displayed below.



SP Change 40 → 60



PV.PV 60.00

Controller Action 30.00

Sepoint 60.00

setpoint 60.00

PID gains

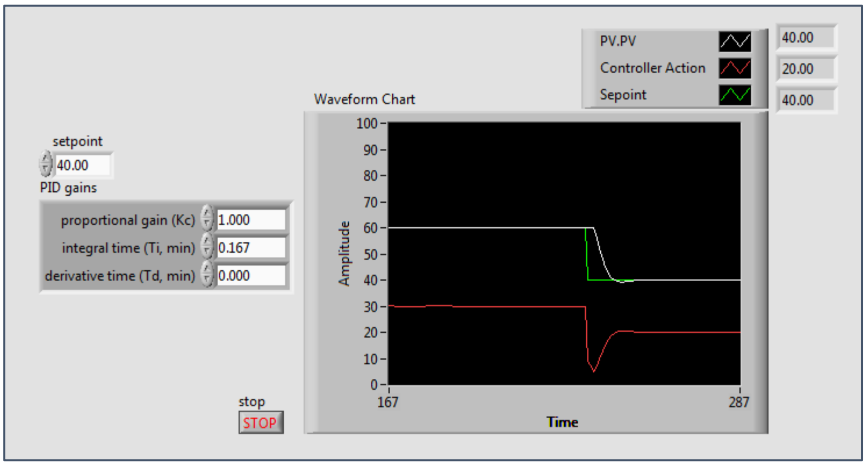
proportional gain (Kc) 1.000

integral time (Ti, min) 0.167

derivative time (Td, min) 0.000

stop

SP Change 60 → 40



PV.PV 40.00

Controller Action 20.00

Sepoint 40.00

setpoint 40.00

PID gains

proportional gain (Kc) 1.000

integral time (Ti, min) 0.167

derivative time (Td, min) 0.000

stop

10.19 Appendix S: Documentation Testing - Michael Crooke Control Modules and HMI

Mathematical Example

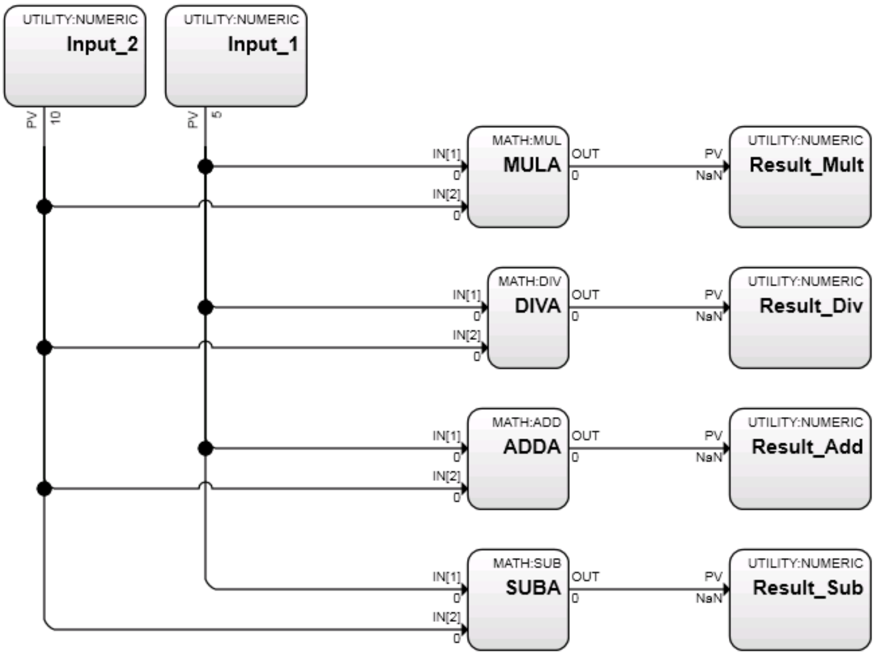
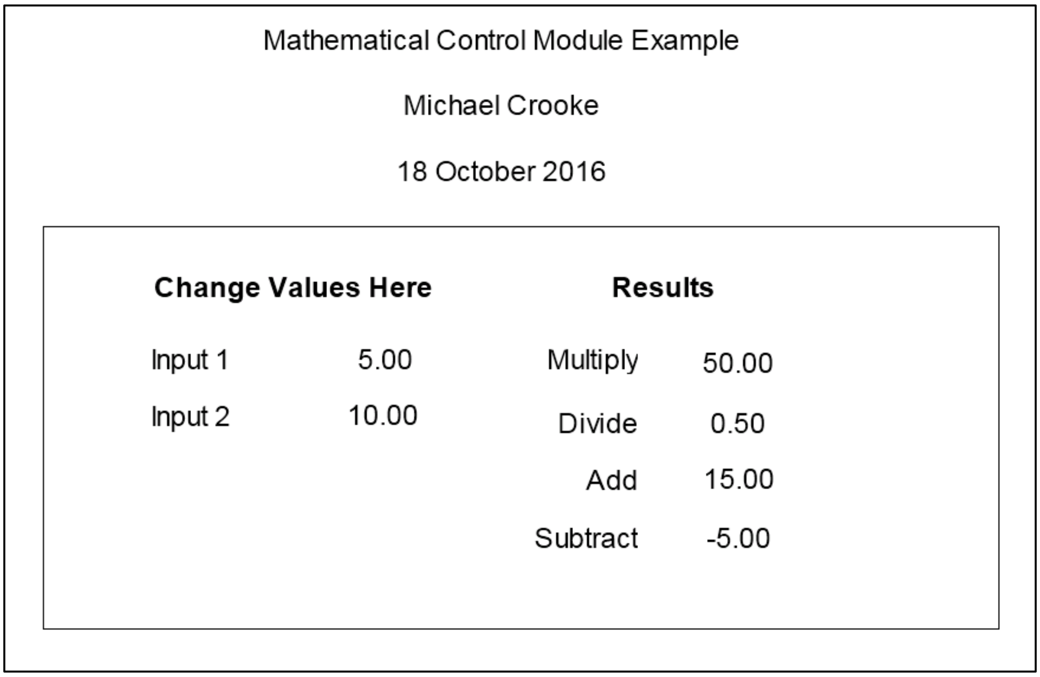


Figure 102: Michael Crooke Example Mathematical Control Module



Mathematical Control Module Example

Michael Crooke

18 October 2016

Change Values Here		Results	
Input 1	5.00	Multiply	50.00
Input 2	10.00	Divide	0.50
		Add	15.00
		Subtract	-5.00

Figure 103: Michael Crooke Example Mathematical HMI

Boolean Logic Example

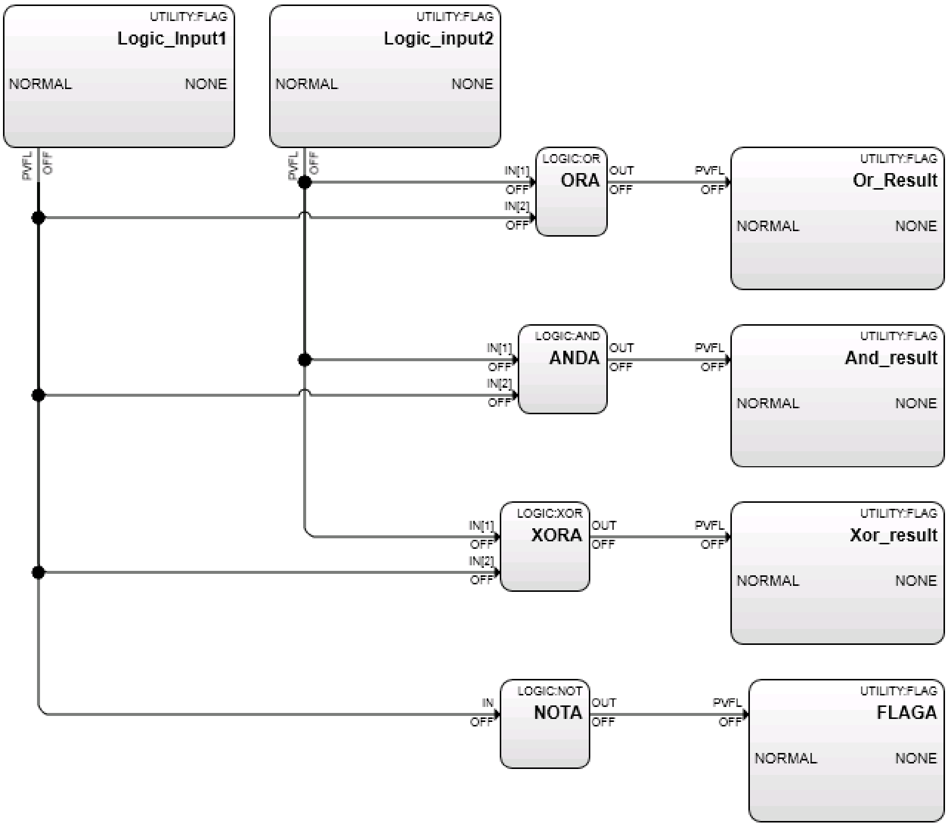
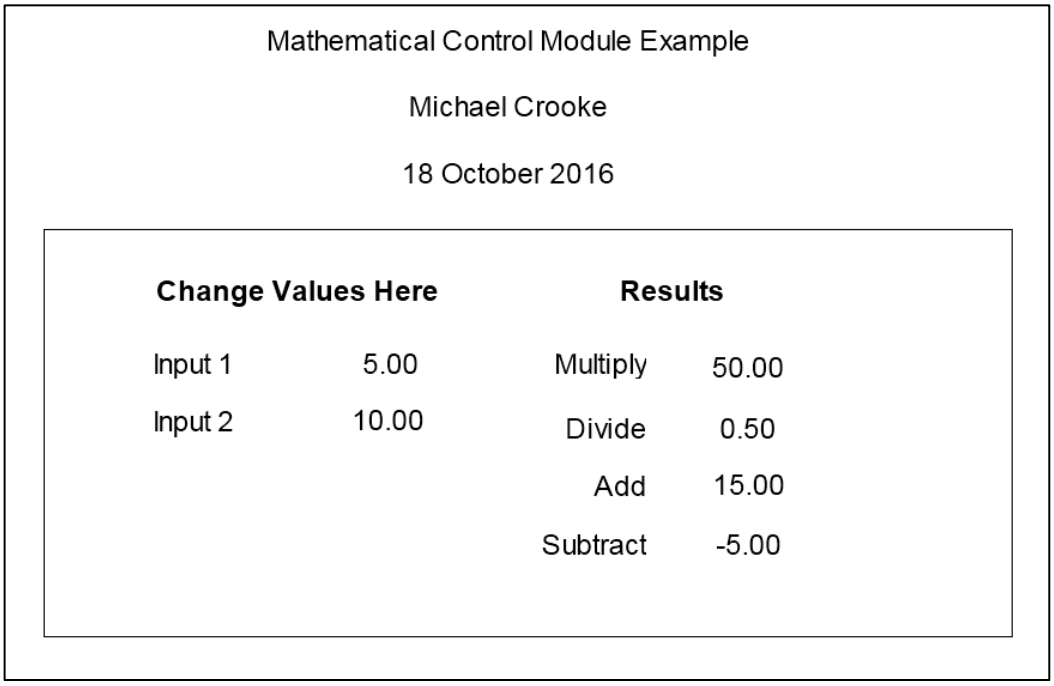


Figure 104: Michael Crooke Example Boolean Control Module



Mathematical Control Module Example

Michael Crooke

18 October 2016

Change Values Here		Results	
Input 1	5.00	Multiply	50.00
Input 2	10.00	Divide	0.50
		Add	15.00
		Subtract	-5.00

Figure 105: Michael Crooke Example Boolean HMI