Multivariable Model Predictive Control of a Pilot Plant Using Honeywell Profit Suite

by

Paul Wheat

Supervisor: Professor Parisa Bahri

MURDOCH UNIVERSITY
PERTH, WESTERN AUSTRALIA

Submitted to the School of Engineering and Information Technology, Murdoch University in partial fulfilment of the requirements for the ENG470 Engineering Honours Thesis

Author’s Declaration

I 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.

_____

[Author]

Abstract

This thesis documents the first implementation of Profit Suite into Murdoch University's Pilot Plant. This Pilot Plant is a small scale model of the Bayer Alumina Process. Profit Suite is a Honeywell application that uses Model Predictive Control (MPC) for Multivariable Control (MVC). The major project objective was to connect Profit Suite to the exiting Experion PKS control system then compare multivariable model predictive control to the existing PI control scheme. The project objectives were achieved. Multivariable controllers were built that controlled temperatures and levels in both halves of the plant.

The OPC connections between Profit Suite and Experion were completed and documented, as well as the procedures used to build and commission Profit Controllers in the Pilot Plant. Multivariable level controllers were designed using accurate models that performed well. These MVCs performed better than PI control in that they managed all tank levels and recycle streams throughout the plant. Linear objective functions were used to optimize flows and levels with success.

Baseline testing of the PI Controllers showed they were better than the MVCs for temperature control. The steam pressure disturbance had no effect on temperatures controlled by fast executing Experion PI controllers. Models found for steam pressure caused MVCs to overcompensate for this temperature disturbance. An MVC built that could manipulate steam valve positions to control temperature performed poorly compared to PI control. Multivariable temperature control was significantly improved when all pumps and steam valves were used as Manipulated Variables by the MVC. Models between water flow rates and temperatures enabled the MVC to use additional pump MVs to counteract the steam pressure disturbance.

There was no existing instrumentation to measure steam flowrates from each valve. This required Profit Suite to connect to the OP point of the PI Controllers to directly manipulate valve position for temperature control. Temperature control by cascaded PI steam flow control is recommended to improve the performance of multivariable temperature control. The installation of steam flow transmitters will enable the set point of a PI flow controller to be used as an MV by Profit Control. Fundamental models between steam flowrates and tank temperatures could then be acquired for multivariable control.

Acknowledgements

I wish to thank my supervisor Professor Parisa Bahri for granting me the opportunity to participate in this first implementation of Profit Suite at Murdoch University and providing guidance throughout the project. I would also like to thank technical staff Mr Mark Burt, Mr Will Stirling and Mr Graham Malzer for their invaluable help and willingness to share their knowledge.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

APC – Advanced Process Control

BMT – Ball Mill Tank

CEE – Control Execution Environment

CLid – Closed Loop Identification

CM – Control Module

CSTR – Continuously Stirred Tank Reactors

CUFT – Cyclone Underflow Tank

CV – Controlled Variable

DI – Digital Input

DMC – Dynamic Matrix Control

FB – Function Blocks

FCV – Flow Control Valve

FIR – Finite Impulse Response

GMC – Generic Model Control

HMI – Human Machine Interface

I/O – Inputs/Outputs

MIMO – Multiple Input Multiple Output

MMPC - Multivariable Model Predictive Controller

MPC – Model Predictive Control

MV - Manipulated Variable

MVC – Multi-Variable Controller

PI – Proportional Integral

PID – Proportional Integral Derivative

PKS - Process Knowledge System

PV – Process Variable

RMPC – Robust Model Predictive Control

RGA – Relative Gain Analysis

SAG - Semi-Autogenous Grinding

SISO – Single Input Single Output

SNR – Signal to Noise Ratio

SP – Set Point

SVT – Singular Value Thresholding

VSD – Variable Speed Drive

Chapter 1 Introduction

The Murdoch University Pilot Plant is a small scale model representative of the Bayer process for refining bauxite into alumina. The Pilot Plant can be operated in its entirety or separated via the overflow from the Lamella Tank into two halves which can be run independently. The first half contains the Grinding, Digestion and Clarification stages, and the second half of the plant contains three steam heated tanks simulating the Precipitation stage (Meiri 2015). This plant is currently used to develop and implement Single Input Single Output (SISO) and multiloop control strategies for tank levels and temperatures. The performance of the controllers can be evaluated by their effectiveness at maintaining their set points despite the process interactions and disturbances inherent in the plant itself. The primary objective of this thesis is to implement multivariable Model Predictive Control (MPC) and plant optimization into the Pilot Plant and then compare its performance against SISO multiloop control.

1.1 Background

Honeywell's Experion Process Knowledge System (PKS) software is used to control the Pilot Plant. Proportional Integral (PI) controllers are coded as Function Blocks inside Control Modules (CM) which are executed by Honeywell C300 controllers (Hopkinson 2010). To date, the Process Variables (PV) have been controlled as SISO processes using PI regulatory control, Generic Model Control (GMC) and Dynamic Matrix Control (DMC). While students have modelled each process and tuned their own spreadsheet based controllers using Microsoft Excel Data Exchange (MEDE), the parameters of the PI regulatory controllers inside the Experion CM have been chosen arbitrarily then adjusted through trial and error (Mackay 2012).

The Pilot Plant is an example of a Multiple Input Multiple Output (MIMO) control problem whereby numerous Controlled Variables (CV) must be controlled and numerous Manipulated Variables (MV) are present to control them (Seborg 2011). Process interactions occur where moving one MV affects two or more CVs. The current approach for mitigating the multivariable loop interactions of the Pilot Plant has been the use of Relative Gain Analysis (RGA) to pair MVs with CVs in conjunction with Decoupling Control. Decoupler transfer functions must be both realisable and derived from accurate process models to be effective. This is not always achievable in practice (Seborg 2011). To maintain the stability of the plant the individual controllers are often detuned to keep the process within safe limits which results in sub-optimal control of the plant (Ogunnaike and Ray 1994). The complexity of Decoupler Control also

increases geometrically as the number of variables in the process matrix increases (Honeywell Process Solutions 2015a).

1.2 Objectives

Murdoch University has purchased and installed Honeywell Profit Suite for Multivariable Model Predictive Control (MPC) and plant optimization (Honeywell Process Solutions 2015a). Unlike the conservative tunings of the aforementioned control strategies, multivariable MPC allows the plant set points to be moved closer to their optimum limits for plant performance. Profit Suite is composed of a set of applications running on a dedicated server for the design and implementation of multivariable MPC control.

MPC uses a model of the process to predict the future value of each CV over a prediction horizon due to current and past changes to input variables. Profit Suite will be used to model the relationships between all MVs, PVs and Disturbance Variables (DV) to be incorporated into the design of a multivariable Profit Controller. This controller makes coordinated moves to all MVs based on their relationships with each CV to drive the process towards optimal performance (Seborg 2011). Feedforward action is taken to deal with modelled disturbances such as the steam pressure from the boiler.

Profit Suite has the capability for plant optimization in which an objective function is selected and minimized. Costs (values) are assigned to CVs and MVs based on economic importance and these can be minimized (or maximised) to obtain optimum performance and efficiency from the plant (Honeywell Process Solutions 2015b). Proposed optimization strategies for the Pilot Plant could include maximizing plant throughput with good level control, or optimizing temperature set point tracking in the Heated Tanks despite loop interactions, disturbances from steam pressure, worn steam valves and safe physical limits of the pump MVs.

A key point from the Profit Suite literature is that good PI regulatory control must be achieved before implementing Profit Suite (Honeywell Process Solutions 2015b). Like other MPC applications, Profit Controller makes moves on the MVs by adjusting the set points of the regulatory (PI loop) controllers (Seborg 2011). This means that the MPC models and the performance of the multivariable control depends on the dynamics of the underlying PI controllers. The tuning parameters of the Experion PI controllers will become incorporated into the dynamic models used by the Profit Suite Multivariable controller. These Experion PI controllers must be tuned effectively before the Profit Suite implementation begins.

The main objectives of this thesis are thus:

- Literature review
- Complete and configure the installation of Profit Suite Applications and OPC communications
- Establish good regulatory PI control
- Implement Profit Suite Multivariable MPC and compare to PI control
- Implement plant optimization and compare to multivariable MPC and PI control

Chapter 2 Literature Review

2.1 Advanced Process Control

Advanced Process Control (APC) refers to any control strategies that are more sophisticated than classic PID feedback control (Fayruzov et al. 2017). Their purpose is to provide improved process performance over classical PID which must be realised to justify their expense (Smith 2010). Advanced Control is a label for Model Predictive Control, which is an example of Advanced Process Control (Smith 2010).

All processes have hard limits or constraints for each Controlled Variable that cannot be violated without causing plant shutdowns, damage, production losses or poor quality product. A plant may have many thousands of PID loops which must all be tuned optimally. Processes change due to mechanical wear, process conditions (summer/winter), economics or deviations in raw materials quality. This degrades the performance of PID loops to over time resulting in CV oscillations, and prolonged poor SP tracking (Howes et al. 2014). Without APC, operators choose safe set points so that these oscillations lie within the bounds of the high and low alarms limits so no constraints are exceeded (Howes et al. 2014). The plant is operated sub-optimally in a Comfort Zone as in Figure 2.1.

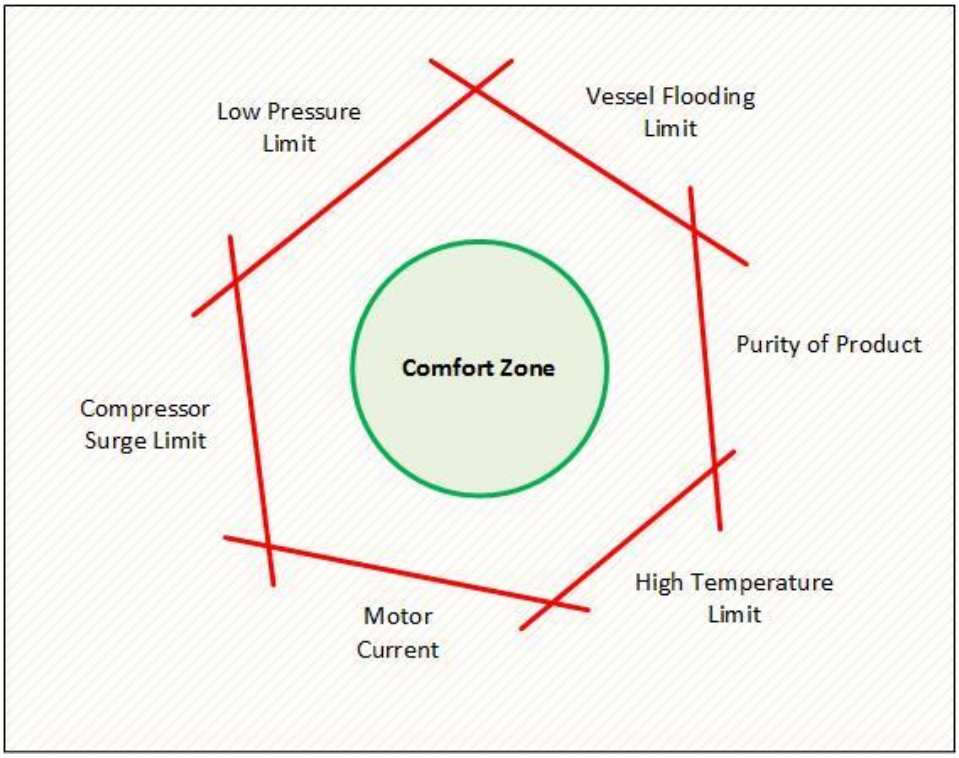


Figure 2.1 Comfort Zone Set Points avoid violating constraints (Howes et al. 2014)

A significant reduction in the amplitude of process oscillations can be achieved through the implementation of an APC software application (Howes et al. 2014). APC applications use MPC to predict the future behaviour of the plant based on current values of process variables and accurate process models obtained from automated step testing (Fayruzov et al. 2017). The APC acts as a master to the slave PID loops. At each execution interval the APC forecasts the future state of the plant to determine if any constraints will be violated (Fayruzov et al. 2017). If so, the APC moves the MVs by changing the SP of the PID regulatory controllers to avoid exceeding the constraint (Seborg 2011). This model based control is more effective at maintaining process stability and consistent product quality than operators are with PID feedback control. CV oscillations are reduced such that the process can be driven closer to the constraints which maximise production and profit as shown in Figure 2.2.

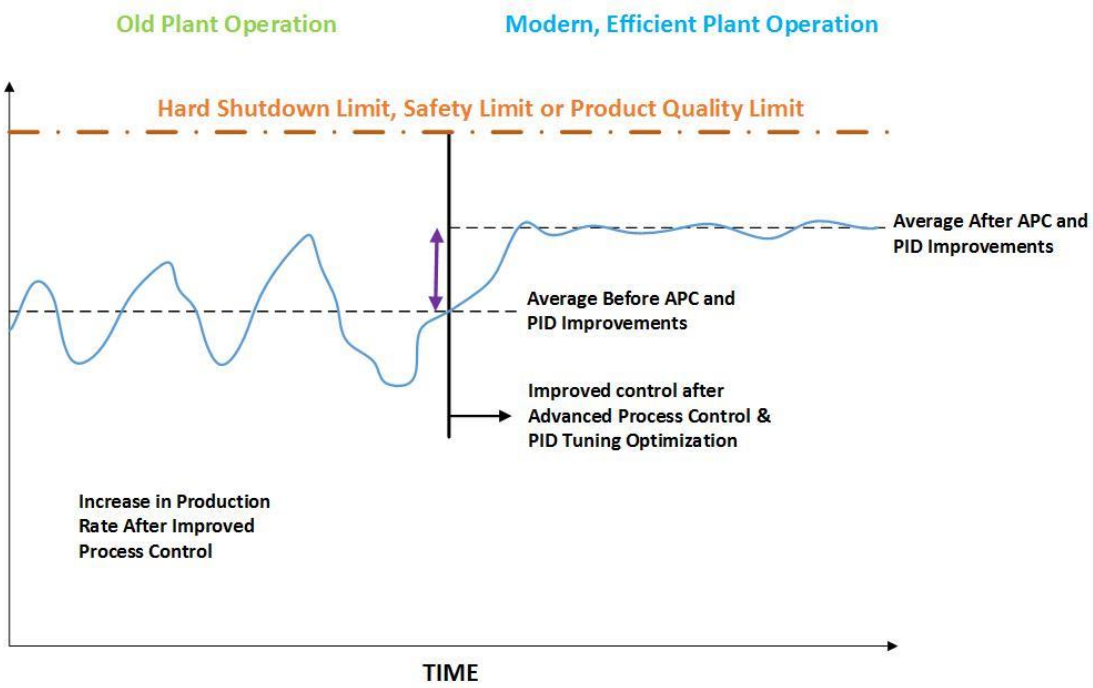


Figure 2.2 APC reduces process oscillations. The process is moved closer to constraints to maximise profit (Howes et al. 2014).

Significant profit gains of 10-15% have been achieved through APC implementation, though gains are usually around 2-10% (Howes et al. 2014). An installation of Honeywell's Profit Suite APC for a Crude Distillation unit resulted in reduction in standard deviation of the three key product qualities by 10.5 to 53%. Profit Suite APC was consistently capable of maximizing key product flows while minimizing energy inputs. The conservation of energy combined with product stabilized product quality resulted in a payback time of six months on the cost of software, engineering trials and training personnel (Fayruzov et al. 2017).

A study by Baker (2008) raised an issue with assessing the performance of APC against PID feedback control with Alcoa's Profit Suite APC implementation at their alumina refineries. Statistical methods are ineffective when comparing range control with set point control (Baker 2008). Profit Suite Range Control will be used for tank levels in this project but not temperature. Statistical methods will be used to assess performance changes by focusing on Heated Tank Set Point errors (Baker 2008).

2.1.1 Multivariable Model Predictive Control

The Pilot Plant is a MIMO control problem because it has many CVs and many MVs to choose from to control them (Seborg 2011). A hallmark of MIMO are Process Interactions, whereby a change to one MV changes two or more CVs (Seborg 2011). To date, Relative Gain Analysis (RGA) has been used to pair MVs to CVs, with the redundant MVs fixed. Decoupling Control has been used to limit process interactions (Wheat and Poonlua 2017b). Decoupler Control is not always a solution: it requires accurate process models with realisable transfer functions (Seborg 2011). Moreover, the complexity increases in geometric proportion to the size of the matrix of process variables (Honeywell Process Solutions 2015a).

MPC such as DMC uses a model of the process to form a matrix of step response coefficients. This is used to calculate the future value of the CVs over a number of time steps into the future called the prediction horizon. The goal of DMC is to minimize CV error and control the process over a target trajectory by calculating a sequence of future control moves. The control horizon is defined as the number of future control moves. Only the first control move is ever implemented as at every time step, the predictions and MV moves are recalculated using current process measurements (Seborg 2011). This is called the receding horizon approach (Seborg 2011).

With accurate process models, MPC provides significant advantages over less sophisticated SISO controllers. The MPC controller can determine from the predictions at each time step if process constraints will be violated over the prediction horizon and make SP changes to prevent it. Multivariable MPC is even more sophisticated in that no MVs are paired with CVs. The controller acquires valid models between every MV and CV and every measured DV and CV. Then using superposition, the MMPC controller predicts the future state of the process from moves made to any or all MVs based on the current value of DVs and CVs (Seborg 2011).

2.2 Honeywell Profit Suite

Honeywell Profit Suite is a group of software applications for the design and operation of Profit Controllers for Multivariable Model Predictive Control (Honeywell Process Solutions 2015c). The major programs are:

- PSES – Profit Suite Engineering Studio for the process modelling required for building both simulated and real Profit Controllers. Includes Profit Stepper for live step testing.
- PSOS – Profit Suite Operator Station which serves as the HMI for active Profit Controllers
- PSRS – Profit Suite Runtime Studio for the creation of the OPC connections between Profit Controllers and points within Experion PKS Control Modules. Also builds the Watchdog Control Module to be installed in Experion for each instance of Profit Controller.
- URT Explorer – Unified Real Time Explorer which schedules interactions between Honeywell platforms and manages OPC communications between servers.

2.2.1 Honeywell Robust Model Predictive Control

Profit Controllers use Robust Model Predictive Control, which is Honeywell jargon for the MPC controller's ability to effectively handle model mismatch. Such mismatch occurs from plant wear, non-linear processes, or errors when first acquiring the model from step testing (Honeywell Process Solutions 2015a). Profit Suite employs two methods to maintain process stability:

- Singular Value Thresholding
- Range Control Algorithm

2.2.1.1 Singular Value Thresholding

A matrix is ill-conditioned if small errors in input result in large errors in output. Singular values can be calculated from the elements of the dynamic matrix of step response coefficients (the process model matrix). An ill-conditioned matrix can be identified when the smallest singular values are much smaller than the largest singular values. If an ill-conditioned matrix is used to calculate MV moves on a mismatched process, large errors will occur with the CVs and the process can become unstable. The greater the model mismatch with the actual process, the worse the outcome will be (Honeywell International 2016c).

Profit Control removes singular values from the matrix below a minimum threshold it determines will cause large prediction errors. The control response is slower for the CVs that are contributing to the ill-conditioning, but process stability remains intact (Honeywell International 2016c). SVT has no effect on CVs not contributing to the ill-conditioning, and no effect at all on well-conditioned matrices (Honeywell International 2016c).

2.2.1.2 Range Control Algorithm

To improve control quality for strong CV interactions or significant model mismatch, Profit Suite controls to ranges rather than to set points. The ranges are used within the calculations for the dynamic control solution at each step time. The Profit Controller sees no error for CVs that lie within their ranges, which allows more degrees of freedom for economic optimisation and stability (Honeywell International 2016c).

Set Point Control Problem

If two CVs (CV1 and CV2) move together in the same direction when one MV1 is moved, it is difficult to control them independently to set point. If one CV1 deviates from it set point requiring large moves in MV1, a MIMO controller would have to make many moves to all the other MVs in the process to cancel out the disruptive effect on CV2 (Honeywell International 2016c). It may not be possible to move CV1 without also moving CV2 or disrupting the whole process.

Range Control Solution

If CV1 exceeds it high range limit, CV2 is free to move around within its range limits while CV1 is moved back below its high limit. An interacting control problem is reduced to the control of just one CV. Much smaller MV moves are required for CV1 and no other MVs need be moved to cancel disruptive effects on CV2 so long as it remains within its range.

A matrix containing constraints from interacting CVs is ill-conditioned and can generate large errors. With range control, control moves to correct CVs that have exceeded their range limits often does not move other CVs outside their ranges. Only one constraint is present in the matrix so it is not ill-conditioned.

Figure 2.3 shows Profit Controller opening a Funnel when a range is violated or the range limits are changed. This funnel length is based on the Open Loop settling time of the process and is the time the controller will take to return the CV to within its range limits. The funnel's upper and lower limits are constraints considered by the profit controller when calculating MV moves to return the CV to within the range limits. The funnel does not determine the trajectory of the CV within it. The CV can lie anywhere: as long as it remains within the funnel no MV moves will be made (Honeywell Process Solutions 2015c).

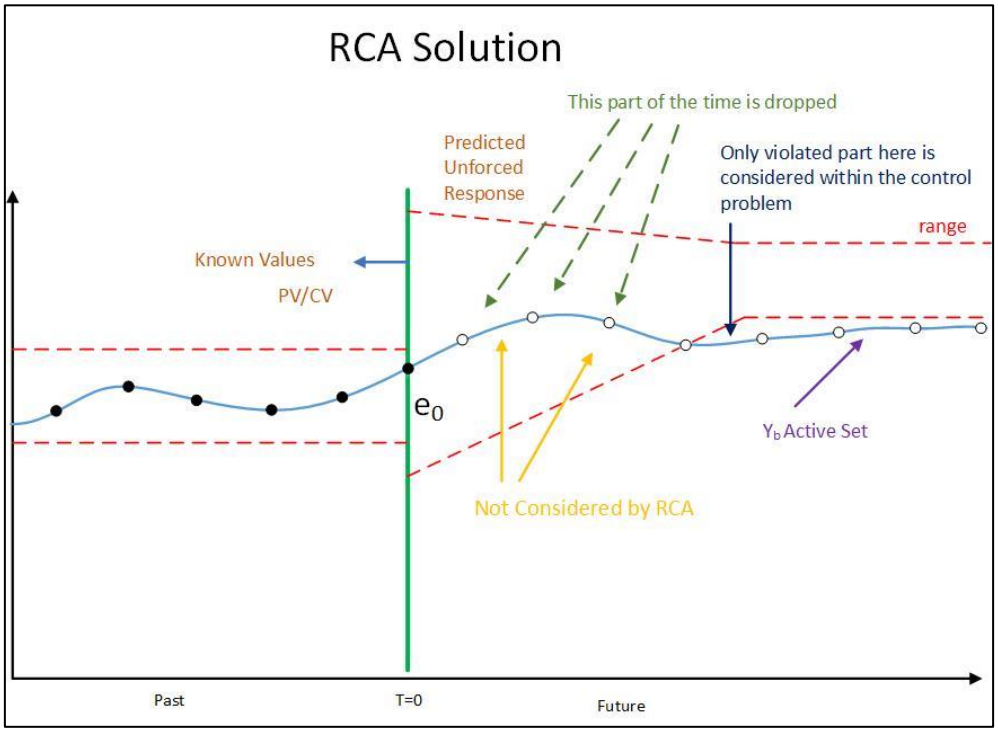


Figure 2.3 Range Control Algorithm. Only CV values outside the funnel and range limits are considered to be errors (Honeywell Process Solutions 2015c).

Shortening the length of the funnel increases the aggressiveness of the controller. This is Profit Controller's major tuning mechanism that is adjusted while online by the operator using a Performance Ratio based on settling time. Each CV can be assigned a unique Performance Ratio to tune its individual response (Honeywell Process Solutions 2015c). The other tuning parameters are called Give-ups. These are essentially rankings on the order in which Profit Controller will stop trying to correct errors for each CV when too many constraints are present.

2.2.1.3 Minimize Control Moves

Profit Suite is designed as a steady state controller that minimises input energy to meet economic objective functions. The Profit Controller minimizes MV movement while maintaining process stability within operating constraints. MVs are assigned weightings according to optimising objectives and their movement is measured as the sum of the squares of each MV change. Whenever there are more MVs available than is necessary to meet the control objective, the Profit Controller moves all MVs a small amount instead of moving one MV a larger amount. This minimizes the value of the square of the changes and therefore total MV movement (Honeywell International 2016c).

2.3 Process Optimization

Profit Suite, like other APC applications, allows for optimisation to move the process towards the constraints that maximize profit. Objective functions can be formulated for maximum profit, such as increasing alumina yield while minimising energy inputs in the Bayer Process:

> "The recovery of alumina from green liquor is driven by a number of process parameters, including alumina super-saturation in liquor, temperature, seed surface area and holding time" (Den Hond, Hiralal, and Rijkeboer 2016).

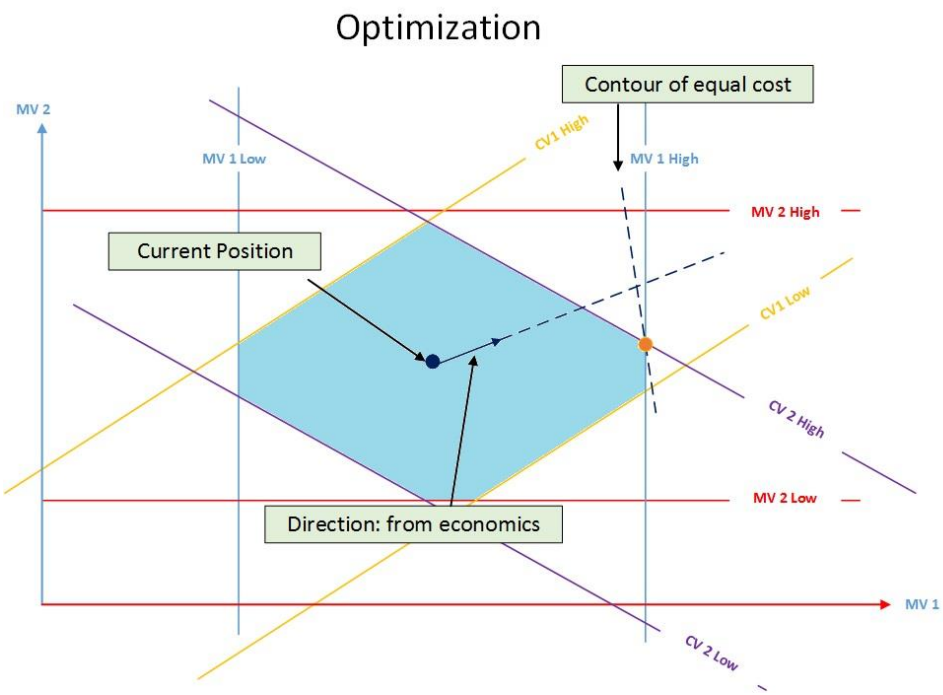


Figure 2.4 Optimisation with Profit Suite moves the process to a 'corner' (Honeywell Process Solutions 2015a).

Figure 2.4 shows the constraints from high and low limits of each CV in the process forming a diamond shape. The process traditionally lies within this comfort zone, but Profit Controller moves it closer to constraints that optimize the objective profit function, which is usually a 'corner' of the diamond in a linear process (Honeywell Process Solutions 2015a). An analogous profit function in this project will be maximising product pump flow while minimizing HT3 set point errors, with minimum possible steam energy input.

2.4 Murdoch Engineering Pilot Plant

2.4.1 The Bayer Process

The Bayer process is the dominant method of refining natural red or white bauxite clays into aluminium oxide (alumina) for the production of aluminium (Wiberg and Wiberg 2001). The most commonly used red bauxite typically consists of:

- 50-60% Aluminium hydroxide $Al(OH)_3$ (Lumley 2010)
- 20-25% Iron oxide Fe_2O_3 (Wiberg and Wiberg 2001)
- 1-5% Silicon dioxide (silica) SiO_2 (Wiberg and Wiberg 2001)
- Small quantities of titanium compounds and trace impurities (Lumley 2010)

Finely ground Bauxite is mixed with caustic (sodium hydroxide) and steam-heated in an autoclave. With constant stirring at temperatures from 140-250°C the aluminium hydroxide dissolves into the solution while the iron oxide does not (Wiberg and Wiberg 2001). The following reactions occur with the equilibrium shown in bold:

- $Al(OH)_3 + NaOH \leftarrow \rightarrow \mathbf{Na[AL(OH)_4]}$
- $\mathbf{Fe(OH)_3 + NaOH} \leftarrow \rightarrow Na[Fe(OH)_4]$
- $SiO_2 + 2NaOH + AL_2O_3 \rightarrow H_2O + Na[Al_2SiO_6]$

  (Wiberg and Wiberg 2001)

After 6-8 hours the pressure is released, the mixture is cooled to 95°C and the iron oxide is removed as red sludge from the aluminate solution through settling and filtration. The silica precipitates from the solution as aluminium silicate and is removed with the red sludge (Wiberg and Wiberg 2001).

Progressively cooling the aluminate solution to around 60°C and adding seed crystals of $Al(OH_3)$ promotes rapid crystallization. The aluminium hydroxide precipitates and the larger crystals are removed through filtration. The smaller crystals become seed for the next cycle and the sodium hydroxide is recycled (Wiberg and Wiberg 2001) (Totten and MacKenzie 2003). The aluminium hydroxide is heated above 1200°C in a furnace to form the final product of the Bayer process, Aluminium oxide (Wiberg and Wiberg 2001). This >99% pure aluminium oxide can be smelted to produce aluminium through electrolysis (Totten and MacKenzie 2003) (Schmitz 2006).

2.4.2 The Pilot Plant

The Bayer process is implemented industrially at Alcoa of Australia's alumina refineries in Western Australia (Ramboll Environ Inc 2005). Alcoa's process is divided into the following steps:

- **Bauxite Grinding and Slurry Storage**

  Ball mills or SAG mills grind bauxite into particles of size less than 1.5mm. A hot slurry is formed by the addition of recycled sodium hydroxide then pumped to holding tanks where silica removal begins (Ramboll Environ Inc 2005).

- **Digestion**

  The slurry is pumped into the digesters or autoclave units where it reacts with additional hot sodium hydroxide to dissolve the aluminium hydroxide. The aluminate solution formed is called green liquor (Ramboll Environ Inc 2005).

- **Clarification**

  The undissolved iron oxide (red sludge) and silica is removed from the green liquor (Wiberg and Wiberg 2001). Sodium hydroxide is washed from the red sludge with water for reuse. Causticisation with heated lime slurry converts any sodium carbonate that has formed in the caustic liquor back to sodium hydroxide (Ramboll Environ Inc 2005).

- **Precipitation**

  The aluminate is progressively cooled with heat exchangers in a series of tanks where alumina hydrate crystals form. The crystals from the final precipitator tank are classified according to size. Fine crystals become recycled seed for the precipitation process and large crystals are sent to calcination. Spent sodium hydroxide is recycled to digestion (Ramboll Environ Inc 2005).

- **Calcination**

  The alumina hydrate is washed, dried, then heated above 1,000 °C which removes water to produce alumina Al_2O_3 (Ramboll Environ Inc 2005). The final product is a fine white powder of similar appearance to table salt (Lumley 2010).

The Murdoch University Pilot Plant was built with input from Alcoa to represent the Bauxite grinding, Digestion, Clarification and Precipitation stages (Vu, Bahri, and Cole 2010). The flow sheet re-drawn in Figure 2.5 shows the pilot plant is greatly simplified model of the real Bayer process.

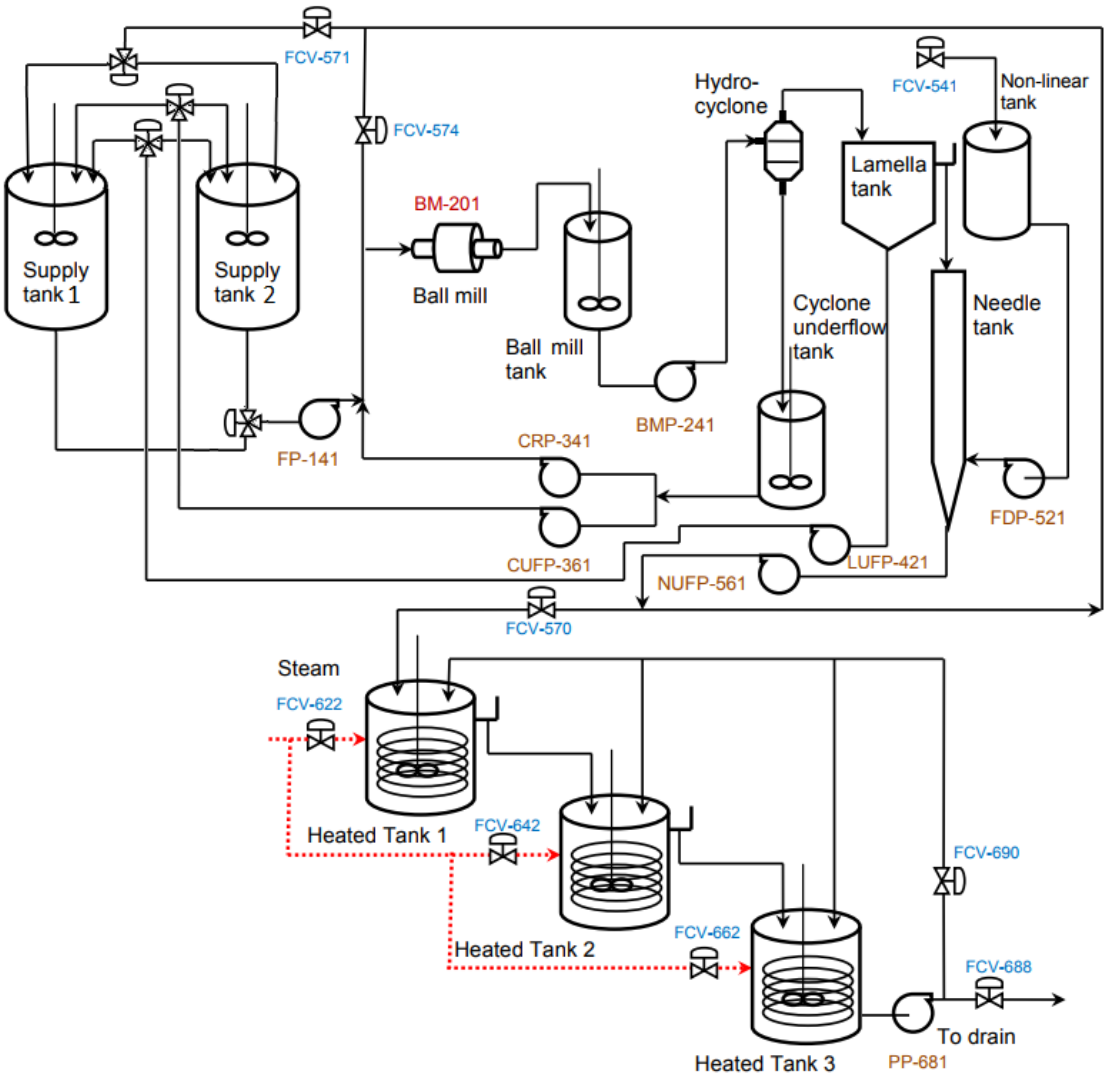


Figure 2.5 Pilot Plant Flow Sheet re-drawn from Vu, Bahri, and Cole (2010).

The Pilot Plant can be operated as one single plant, or separated at the Lamella overflow and run as two independent halves. This is done by diverting the overflow from the Lamella Tank back to the supply tanks, then introducing raw water into the Non-linear Tank with FCV-541 to supply the second half of the plant. The first half contains the Ball Mill, Ball Mill Tank, Hydro-cyclone and Cyclone Underflow Tank. The second half of the plant contains the Non-linear Tank, Needle Tank and three steam-heated tanks Heated Tank 1, 2 and 3 (Meiri 2015). The Pilot plant is mapped to the Bayer process in Table 2.1.

Table 2.1 Pilot Plant mapping to Bayer process

Bayer Process	Pilot Plant
Grinding	Supply Tanks 1 and 2 Ball Mill
Digestion	Ball Mill Tank, Cyclone Underflow Tank
Clarification	Hydro-cyclone, Lamella Tank Needle Tank
Precipitation	Heated Tanks 1, 2 and 3 (CSTR1, 2 and 3)

(Meiri 2015) (Vu, Bahri, and Cole 2010) (Hopkinson 2010) (Ramboll Environ Inc 2005)

2.4.3 Discussion

This comparison highlights the major point of difference that exists between the Digestion and Precipitation stages of the real Bayer process versus the Pilot Plant. The first half of the Pilot Plant contains only level control, whereas the Bayer Digestion stage it represents is steam-heated and temperature controlled. The Bayer Precipitation stage is progressively cooled with heat exchangers which optimises the super-saturation of aluminate solution at each tank to promote crystal growth yield (Ter Weer 2014). In contrast, the Precipitation stage in the Pilot Plant is steam-heated to progressively increase the temperature in each tank. As a result, increasing consecutive temperature SP of the Pilot Plant's Heated Tanks will be used as an analogue for the increasing rate of alumina hydrate crystal growth in the Bayer Precipitation process in this project.

2.5 Existing Status of Pilot Plant Control

The software used to control the Pilot Plant is Honeywell Experion PKS (Hopkinson 2010). The main software environment for Experion is called Configuration Studio, which launches all the programming, network configuration and HMI building software used to control the Pilot Plant (Hopkinson 2010). Configuration Studio is used to explore and configure the live Pilot Plant on server PPServer1 and also the simulation server Experion2 containing the Experion Teaching System used by students learning to program with Experion (Godfrey 2016) (Meiri 2015).

The programs required by this project inside Configuration Studio are:

- Control Builder – used to create and configure control logic; the actual Pilot Plant control code (Hopkinson 2010).
- HMIWeb Display Builder – used to create HMI pages to be viewed in the operator plant interaction and display program called Station (Hopkinson 2010).

2.5.1 Pilot Plant Code Structure

Control Builder contains Control Modules in which the code that controls the Pilot Plant is written as a structure of graphical, object-oriented Function Blocks (Hopkinson 2010). This code is executed by the Control Execution Environment on Honeywell C300 controllers (Hopkinson 2010).

2.5.1.1 Redundant and Non-commissioned Code

In 2009 the Pilot Plant code was migrated from PLC5 to the Existing Honeywell Experion PKS by Honeywell India who did not have access to the physical plant for commissioning or debugging the code (Punch 2009). Thesis students commissioned the Experion code and have been incrementally debugging it over time (Punch 2009) (Hopkinson 2010) (Meiri 2015). The Control Modules still contain erroneous and redundant code. The CM called WARN_LGHT contains code to operate the plant warning light if the plant is put into a maintenance or demonstration mode. This code cannot execute because there is nothing physically wired to the field I/O to initiate it (Meiri 2015).

Code from WARN_LGHT was removed from only the tank level Control Modules during the build of the Pilot Plant Maintenance Program. The Pilot Plant warning light and siren still function correctly when the Operation button in Station is selected (Meiri 2015). To date, redundant code from WARN_LGHT sends incorrect configuration flags to the inputs of the temperature control valve PID Function Blocks. These Experion temperature PI controllers for the Heated Tanks are to be commissioned during the course of this project.

2.5.2 PID Control Parameters

MPC applications move MVs by adjusting the set points of the existing PID regulatory controllers (Seborg 2011). Good PI control must be attained before building Profit Controllers due to the dynamics of the underlying regulatory controllers becoming a part of the MPC models used by the Multivariable Controller (Honeywell Process Solutions 2015b). The Experion PID regulatory controllers for tank level control were commissioned as part of the Pilot Plant Maintenance and Demonstration Program and PI tuning parameters were entered via the Control Loop Tuning Parameters page in Station (Mackay 2012). These PI parameters were arbitrarily selected then adjusted by trial and error to provide adequate level control (Mackay 2012). Therefore, these figures must be replaced with the best possible tuning parameters before building the Profit Controllers.

Undergraduate students control the Pilot Plant using Microsoft Excel spreadsheets. These controllers use Microsoft Excel Data Exchange (MEDE) to read and write directly to Points within Experion via OPC servers. They do not use the PID Function Blocks in the Control Modules (Godfrey 2016). The tuning parameters in Table 2.2 were obtained using relay tuning to model each tank process, then Zeigler Nichols tuning rules to calculate the gains and integral times (Ogunnaike and Ray 1994). These parameters provide good PI control for tank levels in the second half of the Pilot Plant as tested by spreadsheet controllers (Wheat and Poonlua 2017a).

Table 2.2 Tank Level PI Parameters for Second Half of Pilot Plant

Tank Level CV	MV	PI Controller Gain Kc	Integral Time (minutes)
Non-Linear Tank	FCV_541 Raw water valve	4.42	2.43
Needle Tank	NTP_561 Needle Tank Pump	2.43	1.78
Heated Tank 3 (CSTR3)	PP_681 Product Pump	2.35	2.30

2.5.3 Experion PKS PI Control Algorithm

There are five PID algorithms (Equations A to E) to choose from in the configuration of Experion PID Function Blocks (Honeywell Process Solutions 2013). Equation A in Figure 2.6 is used as it most closely resembles the classical PID algorithm introduced to Murdoch University Control Engineering Students, with the addition of a first-order filter on the derivative term (Hopkinson 2010) (Ogunnaike and Ray 1994).

$$CV = K * L^{-1}\left[\left(1 + \frac{1}{T1_s} + \frac{T2_S}{1 + \alpha * T2_s}\right) * (PVP_S - SPP_S)\right]$$

Figure 2.6 The PID Algorithm "Equation A" used in Pilot Plant Experion PID Function Blocks

(Hopkinson 2010) (Honeywell Process Solutions 2013)

The PID blocks in Experion PKS do not accept negative controller gains as is the case when controlling a tank level with the outflow as MV. Instead, PID FB Control Action is configured as either Direct Control Action or Reverse Control Action and all gains must be entered as positive values (Honeywell Process Solutions 2013). The sign of the gain is changed by changing the Control Action. Direct Control Action will increase the CV if the error increases. Reverse Control Action will decrease the CV if the error increases (Honeywell Process Solutions 2013). This is initially confusing because the Honeywell error is calculated as “PV – SP” which is the reverse of the classical PID algorithm error (Ogunnaike and Ray 1994). A tank level controlled with the outflow as MV is configured as Direct Control Action. A tank level controlled with the inflow as MV is configured as Reverse Control Action.

2.5.4 Product Pump and Temperature Interlocks

Meiri (2015) describes the procedure used to install interlocks on the new Product Pump. The Product pump had been locking up when the water temperature in the Heated Tanks was greater than 70°C which tripped out the VSD. The cause was attributed to the hot water expanding the pump internal gears and seizing them. Interlocks were installed on the steam valves such that if the temperature in Heated Tank 3 exceed 70°C all steam valves would close (Meiri 2015). This temperature limit in HT3 made temperature control a challenge for ENG445 students. The set points for all three tanks must fit inside an interval of 40°C to 65°C because the leaking steam valves heat the Heated Tanks to approximately 40°C even with the valves closed (Wheat and Poonlua 2017b).

The root cause of the Product Pump tripping out has since been identified as poor alignment. The new Product Pump was not aligned correctly with the electric motor which placed strain on the shaft making it difficult to rotate. It has been aligned using spacers underneath the feet of the motor such that the motor shaft now spins freely when turned by hand.

2.5.5 Risk Management

Risk Management is important to maintain the functionality of the Pilot Plant for teaching purposes, whilst making significant changes to its control system for this project.

2.5.5.1 Experion PKS Check-Pointing

Dring (2012) discussed backup and restore methods available to mitigate the inherent risks in altering the Pilot Plant code. Check-Pointing can create restore points to roll back to in the event coding errors are uploaded to the C300 controller. A lot of progress may be lost if Check Points are not created frequently enough. This method may be unsuitable when creating a lot of new Experion code, but will be sufficient for the purposes of this project (Dring 2012).

2.5.5.2 Import/Export

An alternative method for saving code is Import/Export. This can be used to save whole Control Modules outside the Experion environment. Code can be exported and saved elsewhere as a backup (Dring 2012). While this project will not create large quantities of Experion code, this method will be employed to import the Profit Controller Watchdog Control Modules.

Detailed instructions for both approaches are outlined in the Control Building User's Guide stored on the Murdoch 'EngShared' drive (Honeywell Process Solutions 2014).

2.5.5.3 OPC Connectivity

Profit Suite communicates to Experion PKS via an OPC server. Equipment such as the Pilot Plant is called an Asset in the Experion environment. Godfrey (2016) detailed the steps required to add assets to the OPC servers for PPServer1 so they can be accessed by third party OPC applications. This information can be used to add Assets in the Experion2 OPC server so changes required for Profit Controller OPC communication can be trialled without disrupting the Pilot Plant teaching activities (Meiri 2015) (Godfrey 2016).

Chapter 3 Profit Control Implementation

3.1 Establishing Good Regulatory PI Control

Commissioning of the Experion PI control for three CSTR temperature controllers and FDP_521 level controller for the Needle Tank was completed as part of this project. These PID controllers could not control their respective CVs in the state they were found. This section describes the code changes made with Experion Control Builder, and tuning parameters used to complete plant- wide PI Control for level and temperature.

3.1.1 Experion Code Changes

3.1.1.1 Control Action

The Experion PI Control Modules contain obsolete code from WarnLight CM. The PID blocks for the Flow Disturbance Pump 521 and all three steam valves had an input from a TYPECONVERTA block which forced their control action to be Direct Acting. This code is both incorrect and will not execute because WARNLIGHT has no field inputs as explained in 2.5.1.1.

Figure 3.1 shows how the PID algorithms were altered from Direct Acting to Reverse Acting by deleting the wire from TYPECONVERTA, updating the PID control action parameters then reloading the four updated CMs. Reverse control action makes the steam valves close further (and FDP_521 slow down) to correct an increasing positive error in the CV.

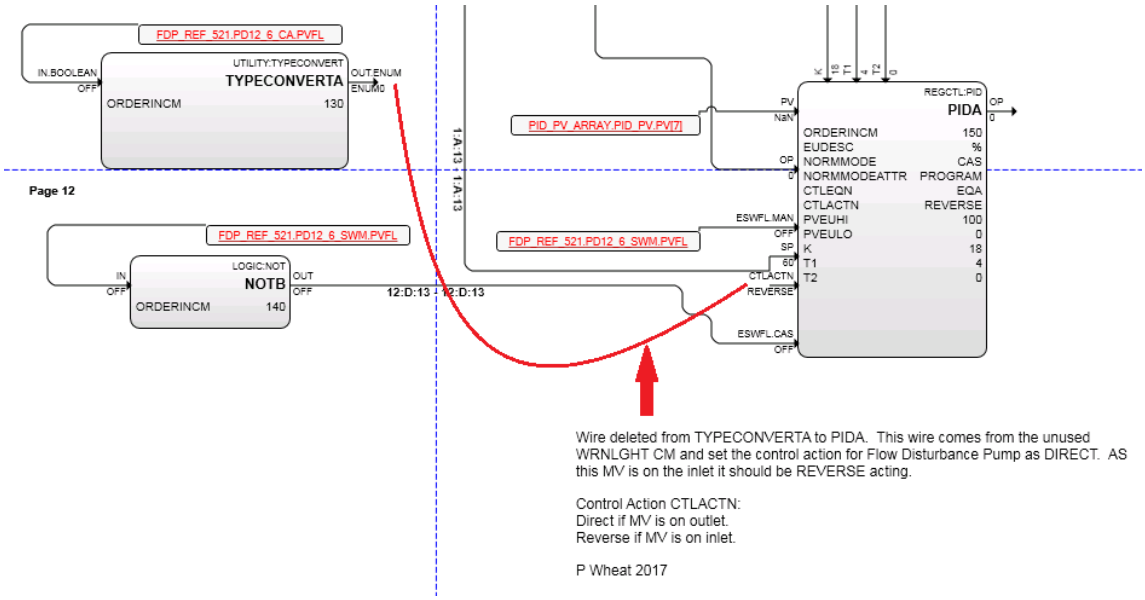


Figure 3.1 Wires deleted for Reverse Control Action in FCV_622, FCV_642, FCV_662 and FDP_521

3.1.1.2 Swap Steam Valve Set Point Function Blocks

In all three steam valve CMs, a FB called SWITCHA is wired to push a set point either from WARNLIGHT or Station into an input pin of the PIDA block via a hyperlink. SWICTHA is supposed to activate if WARNLIGHT executes to push the set point from WARNLIGHT into the PIDA block. The inputs to SWICTHA had been wired backwards such that the set point was always taken from WARNLIGHT (which cannot actually execute) and never from Station page input. The example in Figure 3.2 shows which blocks were swapped so set points can now be entered via Station. In this example of FCVB_622, the SP on the Station page updates the value of N15_13. The block N11_412 stores the SP from WARNLIGHT.

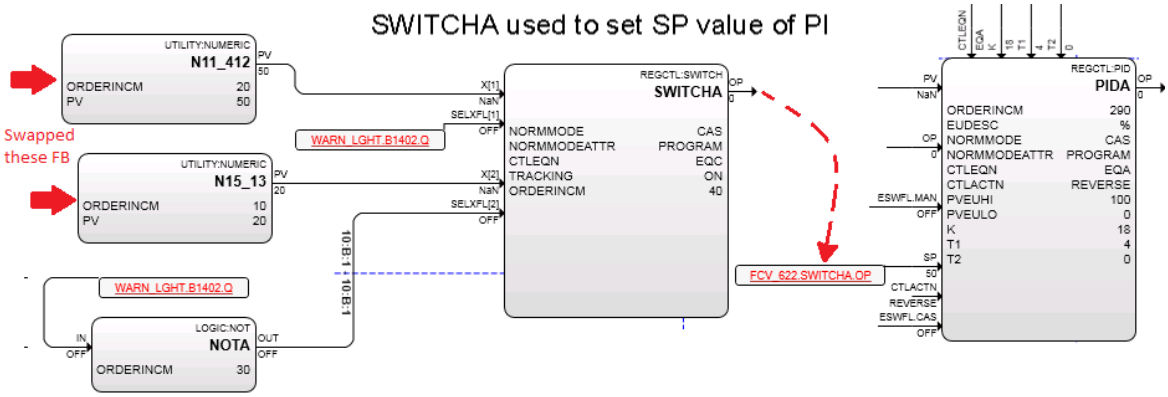


Figure 3.2 Simplified Code shows how Set Point inputs to SWICTHA swapped so CSTR temperature SP are pulled from Station page instead of obsolete WarnLight CM

This was completed for all steam valves as shown in Table 3.1 below:

Table 3.1 Set point SWITCHA input wiring changes for steam valves

Experion Control Module	Function Blocks Wiring into SWITCHA Swapped	
FCV_622	N11_412	N15_13
FCV_642	N15_413	N15_14
FCV_662	N11_414	N15_15

A similar problem was found in the CM for the Flow Disturbance Pump. FDP_521 controls the level in the Needle Tank with the inflow as MV. The block N11_406 in Figure 3.3 stores the Set point value from Station. A hyperlink from block PUSHA pushed the SP from WARNLIGHT into the PIDA block, even though N11_406 had been directly wired to the PIDA block SP input by others. This hyperlink was deleted, code comments were added, and the updated CM was loaded to the C300. Tests proved the FDP_521 can now control NT level with set points updated via Station.

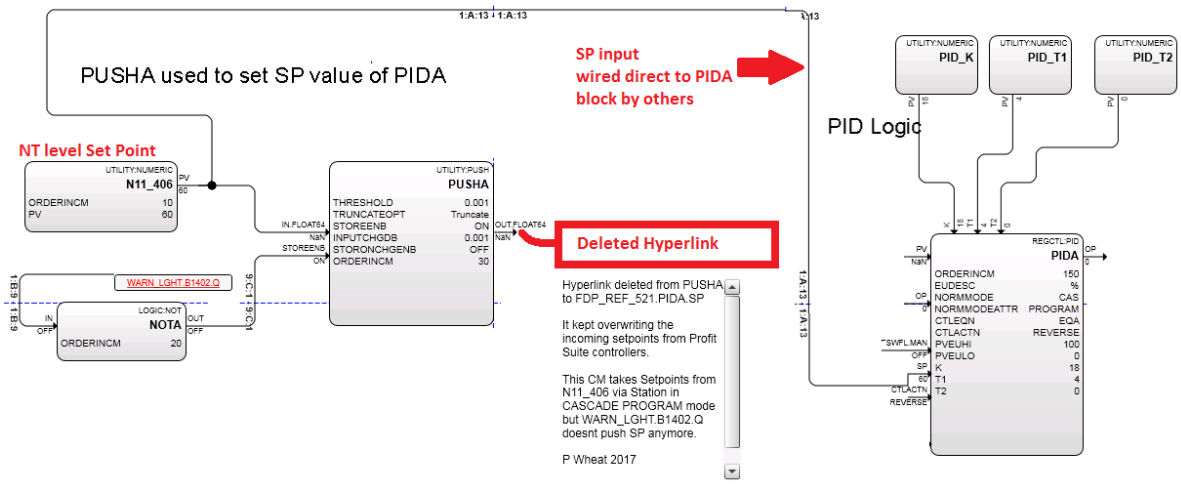


Figure 3.3 Simplified view of Set point input code changes for FDP_521

3.1.1.3 Steam Valve Interlocks

The steam valve temperature interlocks were raised from 70°C to 80°C for reasons explained in section 2.5.4 to allow a greater spread of temperature set points for the three CSTRs. The interlock values were changed to 80 in Function B locks called NUMERICA_1_1 in Control Modules FCV_622, FCV_642 and FCV_662. The DACA block alarms were also changed in Control Modules TT_623, TT_643 and TT_663 as shown in Figure 3.4. These are seen as the flashing red high temperature alarms seen on the Station page:

- PV High High = 80
- PV High = 75

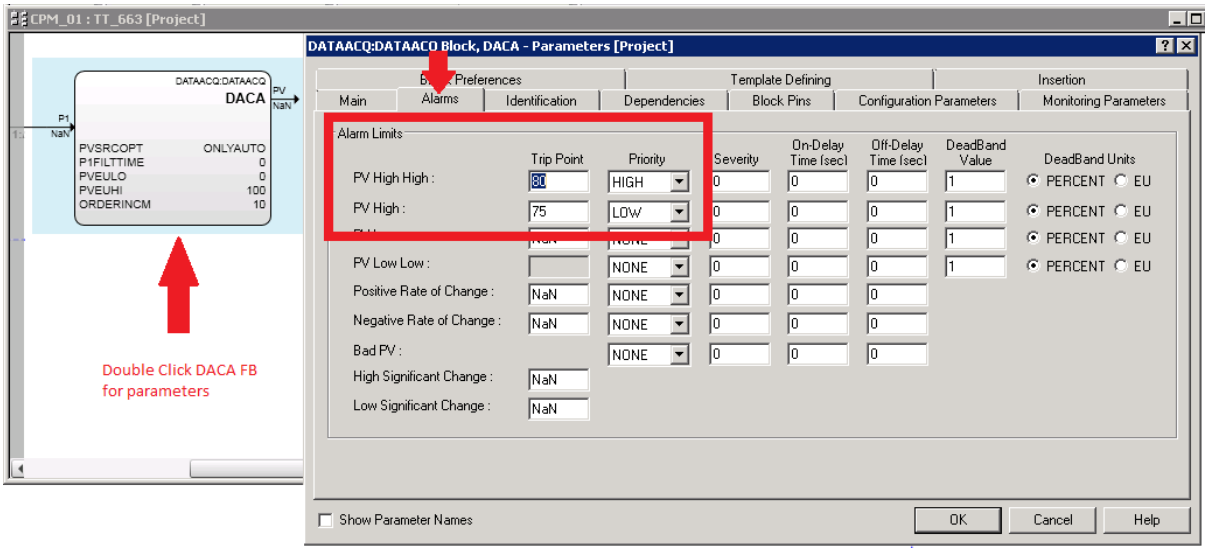


Figure 3.4 Update DACA block alarms after raising temperature interlocks

These changes were saved in all six CMs in the project tree, then these updated CMs were loaded to the C300.

3.1.2 Update Station Page

HMI Web Builder was used to create a replacement Station Page 306 which is the HMI for the CSTR tanks. Station Page 306 has been updated to correct the Set point field of FCV_662 being erroneously linked to the Product Pump Set-point in the Experion CM. Figure 3.5 shows both Set-Point fields which were previously linked to PP_REF_681.N11_415.PV point. It was impossible to change the Set-point for steam valve FCV_622 from the Station page independently of the Product Pump.

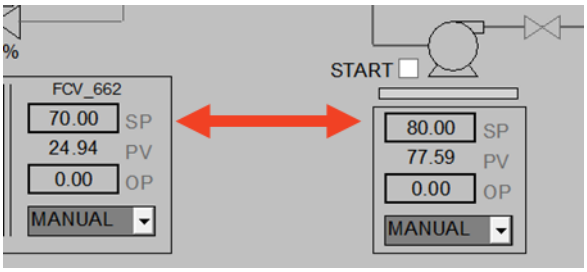


Figure 3.5 FCV_622 SP was incorrectly linked to Product Pump SP

The Set-Point field of FCV_662 was updated in the new Station page 306 files to link to FCV_662.N15_15 point. The new files were saved into the following location on local computers EE2009-01 and EE2009-02 and replaced the incorrect versions as shown in Figure 3.6:

C:\ProgramData\Honeywell\Experion PKS\Client\Abstract

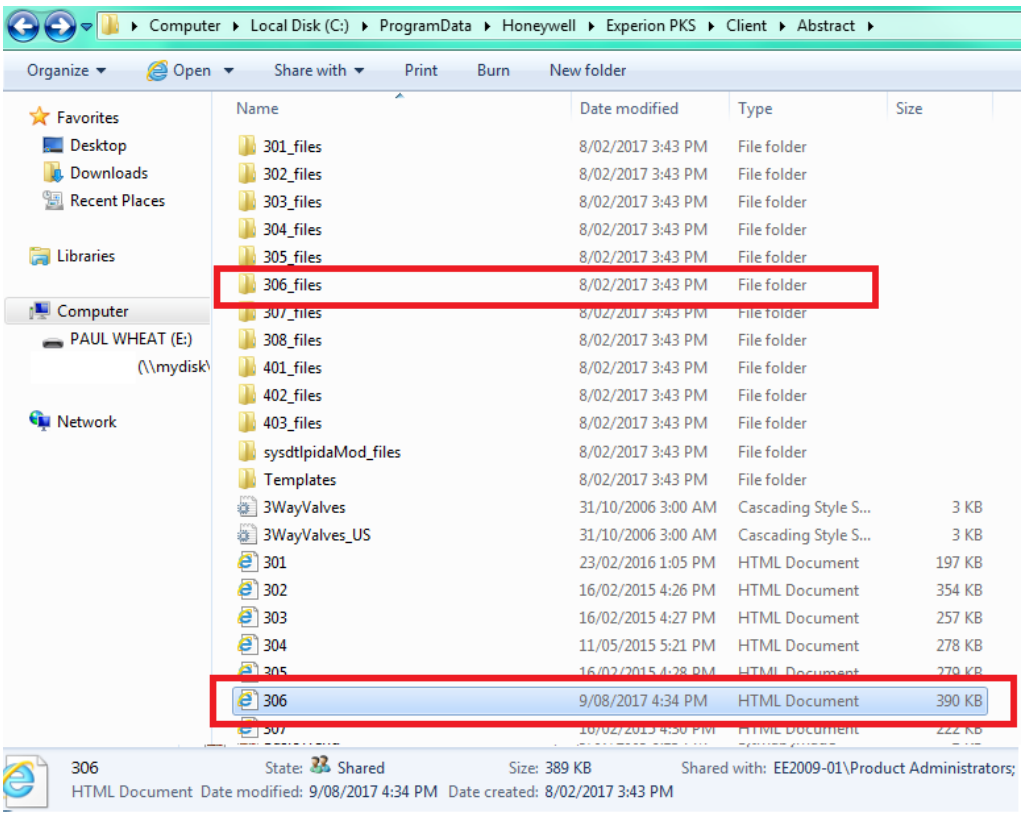


Figure 3.6 Updated Station Page 306 with corrected Set Point location on EE2009-01

The updated page 306 functions correctly. The pages on the remaining Control Room computers were not altered by request so as not to disrupt third year classes. The updated files have therefore been saved into the EngShared folder along with a Read Me text file of explanation and simple installation instructions:

\\mylab\files\engshared\Pilot_Plant_Facility\PP_Project\Updated Station Page_306 2017

3.1.3 Tuning PID Loops

All PI controllers were tuned using Zeigler-Nichols derived tuning parameters (Ogunnaike and Ray 1994). Relay tuning was used when modelling for level controllers. Step testing then Sum of Least Squares modelling was used for temperature controllers. All PI controllers used the PID Algorithm "Equation A" in Figure 2.6.

3.1.3.1 PI Level Control Parameters

The PI parameters and explanation for level control in the second half of the plant are provided in Chapter 2, and Table 2.2. The PI parameters in the first half of the plant were used as found.

3.1.3.2 PI Temperature Control Parameters

The cold water inflow rate into the CSTRs is a DV for the three CSTR temperatures, so tank level controllers were run in PI control with steady state settings listed in Table 3.2 for repeatable step testing of the temperature controllers.

Table 3.2 Steady State settings

Variable	Manual/Auto PI	Set Point %	OP Value %
FCV_541	Auto PI for NLT level Control	60	Approx. 46
FDP_521	Fixed in Manual	-	50
NTP_561	Auto PI for NT level Control	50	Approx. 72
PP_681	Auto PI for CSTR3 level Control	80	Approx. 16

The tank temperatures were step tested then modelled as First Order Systems with Time Delay. Figures 3.7 to 3.9 show the Sum of Least Squares modelling plots found for the three CSTR tanks.

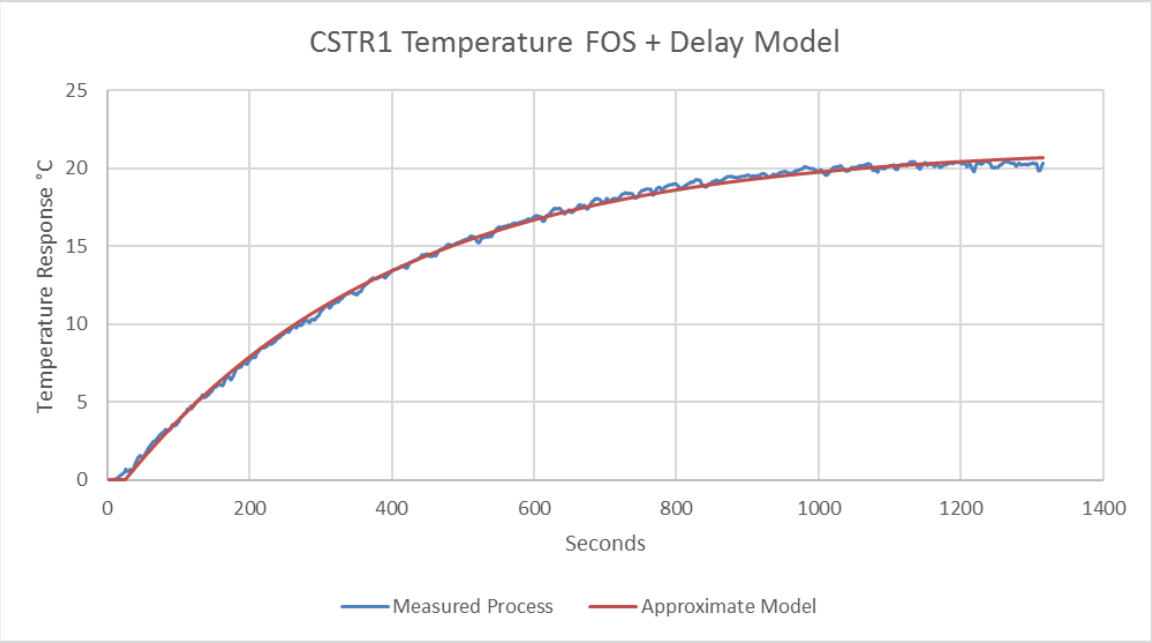


Figure 3.7 CSTR1 Temperature FOS + Delay Model

Table 3.3 CSTR1 Temperature FOS + Delay Model Parameters

Gain K	0.5360
Time Constant τ	380.2424
Delay α	21.3186
Step Size A	40

Using Zeigler Nichols calculations, the PI parameters for CSTR1 temperature control were:

$$K_c = \frac{0.9 * \boldsymbol{\tau}}{\mathbf{K} * \boldsymbol{\alpha}} = 29.95$$

$$\boldsymbol{\tau}_i = 3.33 * \boldsymbol{\alpha} = 71.0619s * \frac{1min}{60s} = 1.18\, minutes$$

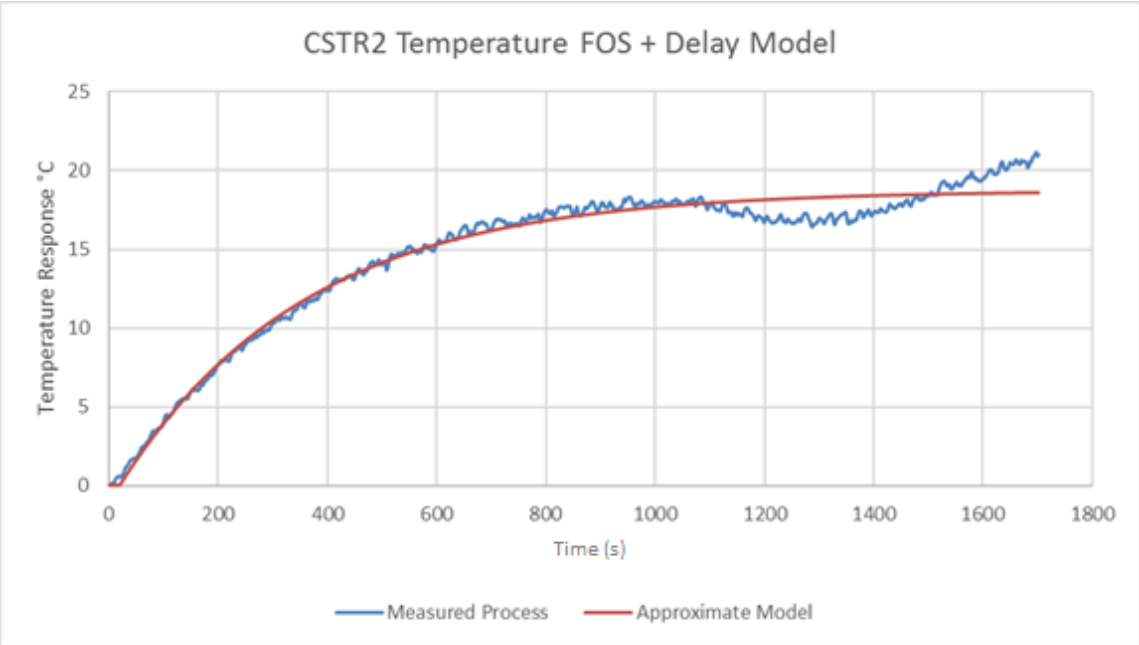


Figure 3.8 CSTR2 Temperature FOS + Delay Model

Table 3.4 CSTR2 Temperature FOS + Delay Model Parameters

Gain K	0.623451
Time Constant τ	341.2862
Delay α	17.49663
Step Size A	30

Using Zeigler Nichols calculations, the PI parameters for CSTR2 temperature control were:

$$K_c = \frac{0.9 * \boldsymbol{\tau}}{\mathbf{K} * \boldsymbol{\alpha}} = 28.16$$

$$\boldsymbol{\tau}_i = 3.33 * \boldsymbol{\alpha} = 58.3221s * \frac{1min}{60s} = 0.97\ minutes$$

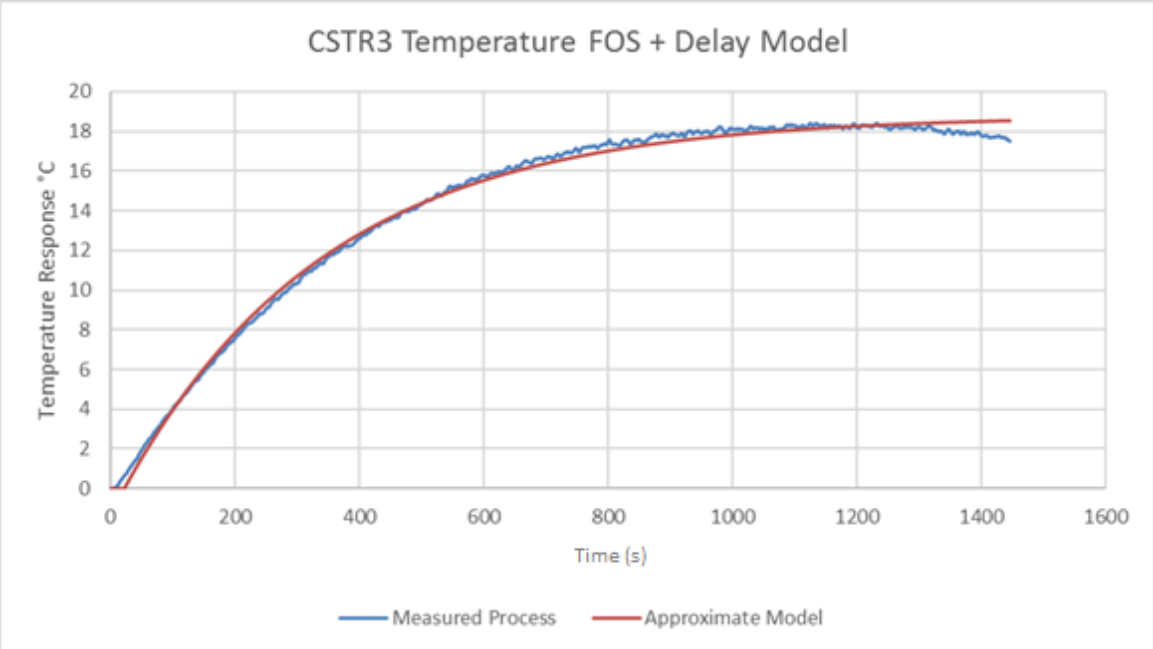


Figure 3.9 CSTR3 Temperature FOS + Delay Model

Table 3.5 CSTR3 Temperature FOS + Delay Model Parameters

Gain K	0.626789
Time Constant τ	329.4863
Delay α	19.93959
Step Size A	30

Using Zeigler Nichols calculations, the PI parameters for CSTR3 temperature control were:

$$K_c = \frac{0.9 * \boldsymbol{\tau}}{\mathbf{K} * \boldsymbol{\alpha}} = 23.73$$

$$\boldsymbol{\tau}_i = 3.33 * \boldsymbol{\alpha} = 66.4653s * \frac{1min}{60s} = 1.11\ minutes$$

Table 3.6 has the complete list of PI parameters used for regulatory control. All other variables were fixed in manual mode, or used as a disturbance variable in the case of FDP_521.

Table 3.6 Complete list of PI parameters used for level and temperature control

MV	PI Controller Gain Kc	PI Integral Time Ti in minutes
FCV_541	4.42	2.43
FDP_521	1.87	1.03
NTP_561	2.43	1.78
PP_681	2.35	2.30
FCV_622	29.95	1.18
FCV_642	28.16	1.18
FCV_662	23.73	1.11
BMP_241	20.00	4.00
CUFP_361	20.00	3.00

The PI parameters were entered via the Station page in Figure 3.10 and will revert to default values if the C300 code is restarted or reloaded:

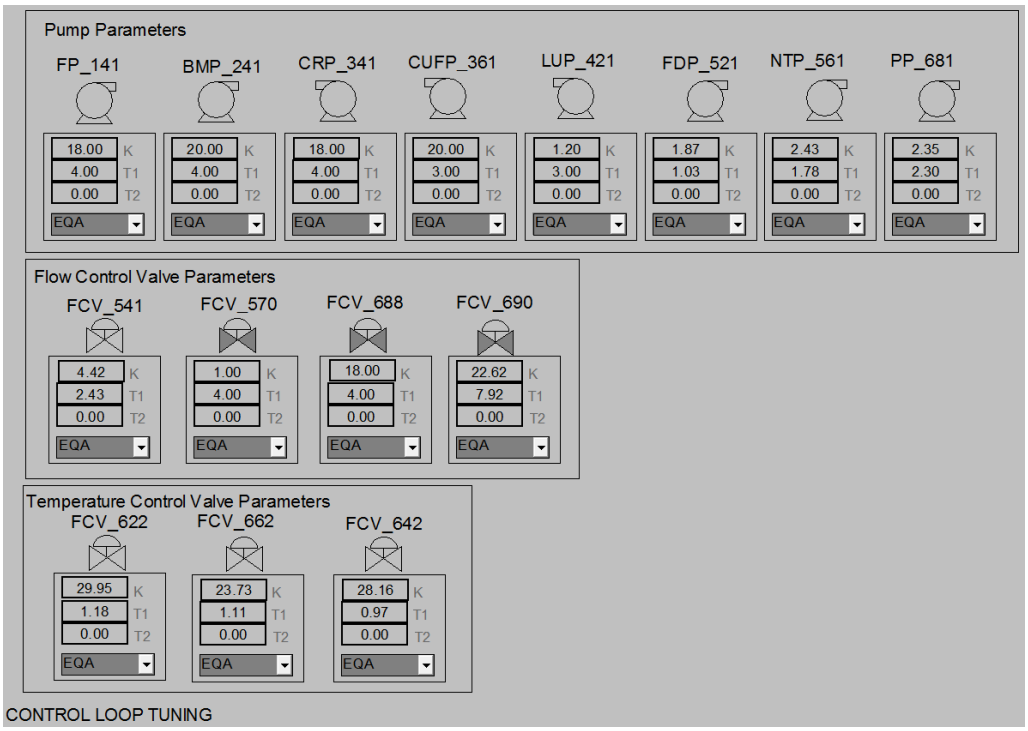


Figure 3.10 PI parameters entered via Station

3.2 Manual Step Testing

Manual step testing is required to build Profit Controllers. The data generated through manual stepping was used to define the variables in the Profit Controllers and their relationships between each other.

3.2.1 Temperature

The step testing data collected when finding PI control parameters for CSTR temperature was combined into one Excel file then imported into PSES Data Warehouse using Custom Excel Converter. All steam valves were stepped up at least three times and stepped down at least twice. The amplitude of the steps was 30 – 40 % and the step duration was approximately 25-30 minutes each. The OP point was stepped and the temperature CV data collected. The intention was to compare models of the same data from PSDS to those calculated using FOS + Delay and Sum of Least Squares. Figure 3.12 shows the manual temperature step testing data as viewed in the PSES Data Warehouse.

3.2.2 Level

The manual step testing plan for tank level controls in Figure 3.11 was constructed according to guidelines provided by Honeywell, which recommends a 12 – 16 steps based on the time constant of the process (Honeywell International 2015). The sequence recommended was 16 steps of τ, 2τ, 3τ, 4τ. The magnitude of the first and last steps was half the planned step size so the process was modelled around the set point.

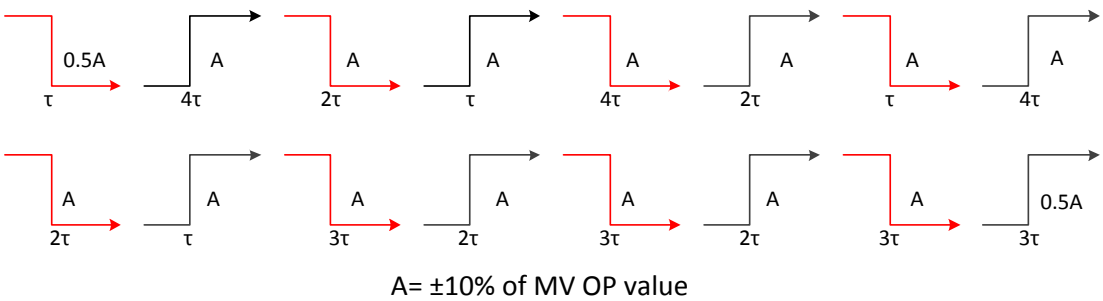


Figure 3.11 Manual step testing plan for tank levels in second half of Pilot Plant

The tank levels were deemed integrating processes because the inflows and outflows were fixed by pump speeds. They have no time constant because steady state occurs only when these flow rates are equal. The values of τ were selected empirically by comparing the size of the pump to the volume of each tank volume to avoid overflows. These are given in Table 3.7. Pump FDP_521 can rapidly overflow the relatively small Needle Tank so the time constant was halved compared to other tanks. In practice, the step testing plan was modified to avoid CV constraints in the live Pilot Plant. The raw data from the actual step tests shown in Figure 3.12 was imported from an Excel file into the Data Warehouse shown in Figure 3.16.

Table 3.7 Step testing parameters used for levels

MV	Step Amplitude A	Time Constant τ (minutes)
FCV_541	10	1
FDP_521	10	0.5
NTP_561	10	1
PP_681	10	1

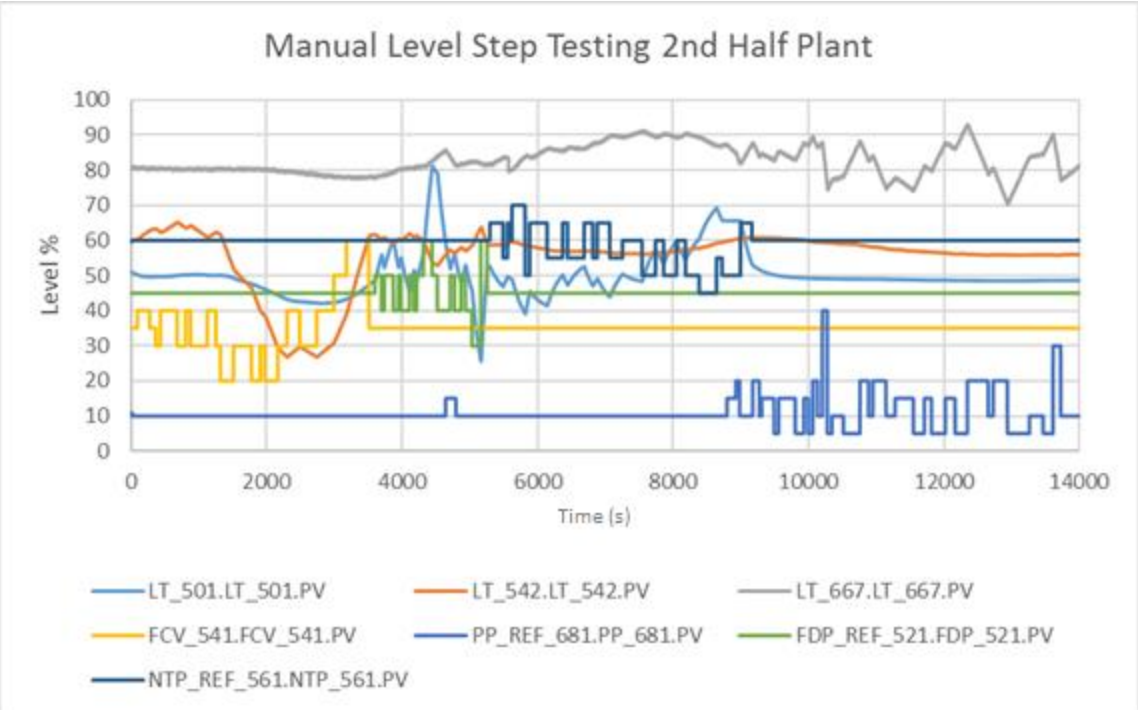


Figure 3.12 Manual Step Testing sequence for levels in the 2nd Half of the Pilot Plant

Figure 3.13 shows an Excel file imported into PSES with Custom Excel Converter. For large Excel files there was a wait time of approximately 30 minutes for the conversion to complete. Data was copied directly from Station into an Excel file then imported with this software. Imports were guided by step by step directions as shown at far right of Figure 3.13.

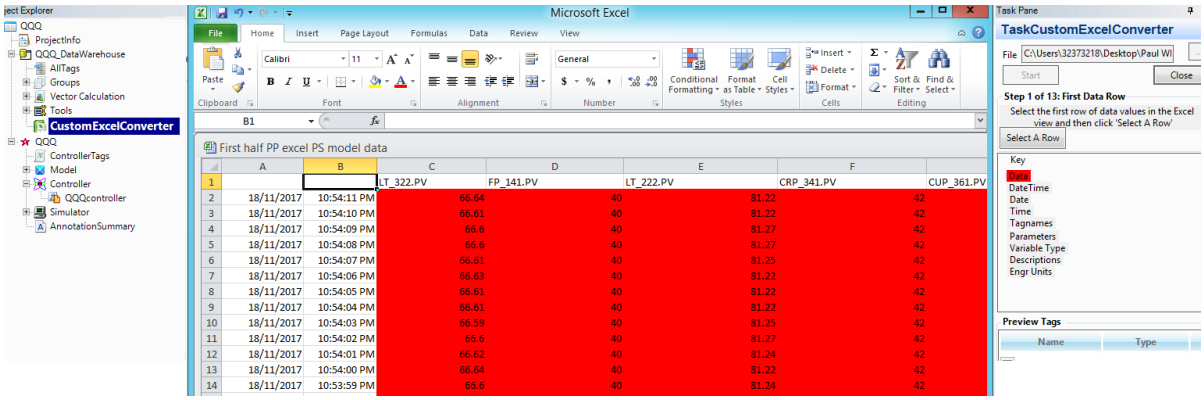


Figure 3.13 Importing step testing data with CustomExcelConverter

The converted Excel files were colour coded by the software to confirm each field selected as shown in Figure 3.14:

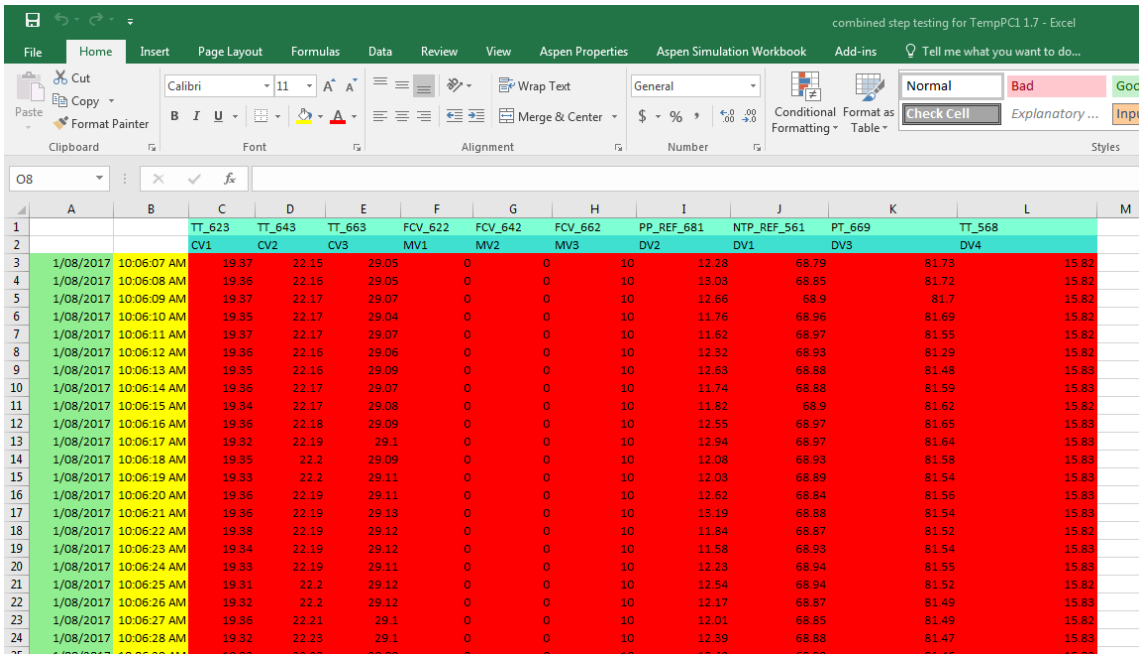


Figure 3.14 Manual Step Test data Excel files have colour coded fields after being imported with CustomExcelConverter

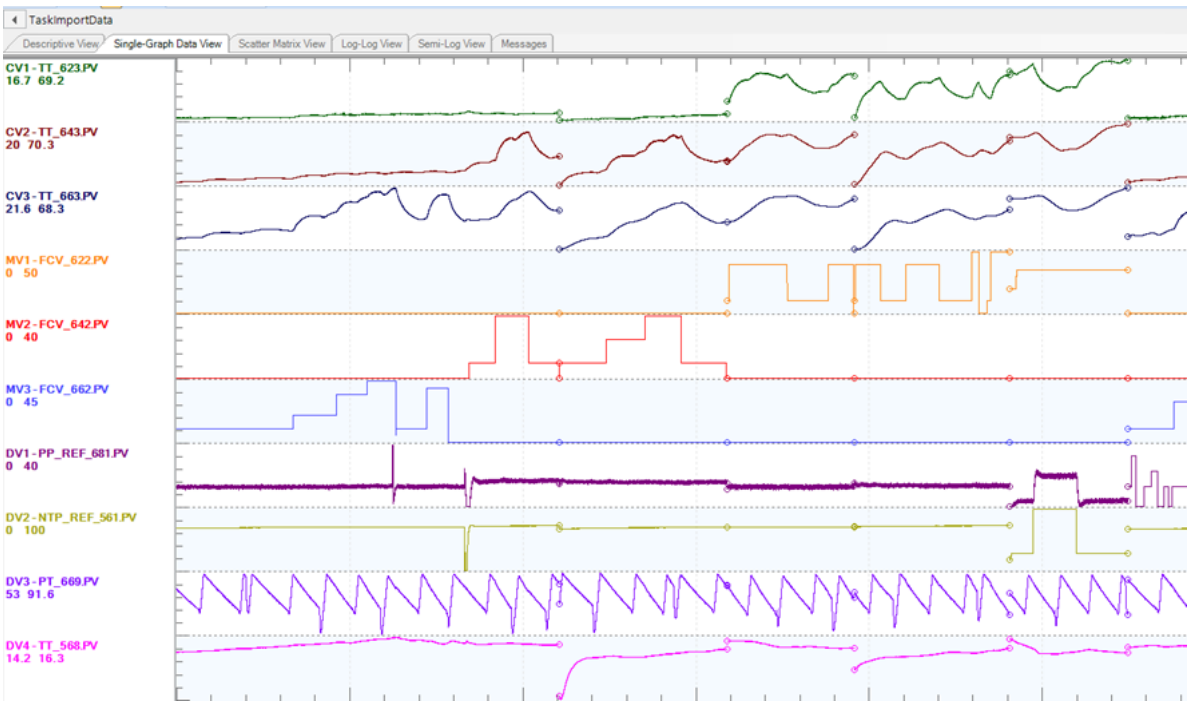


Figure 3.15 Temperature steps imported into Data Warehouse of PSES project

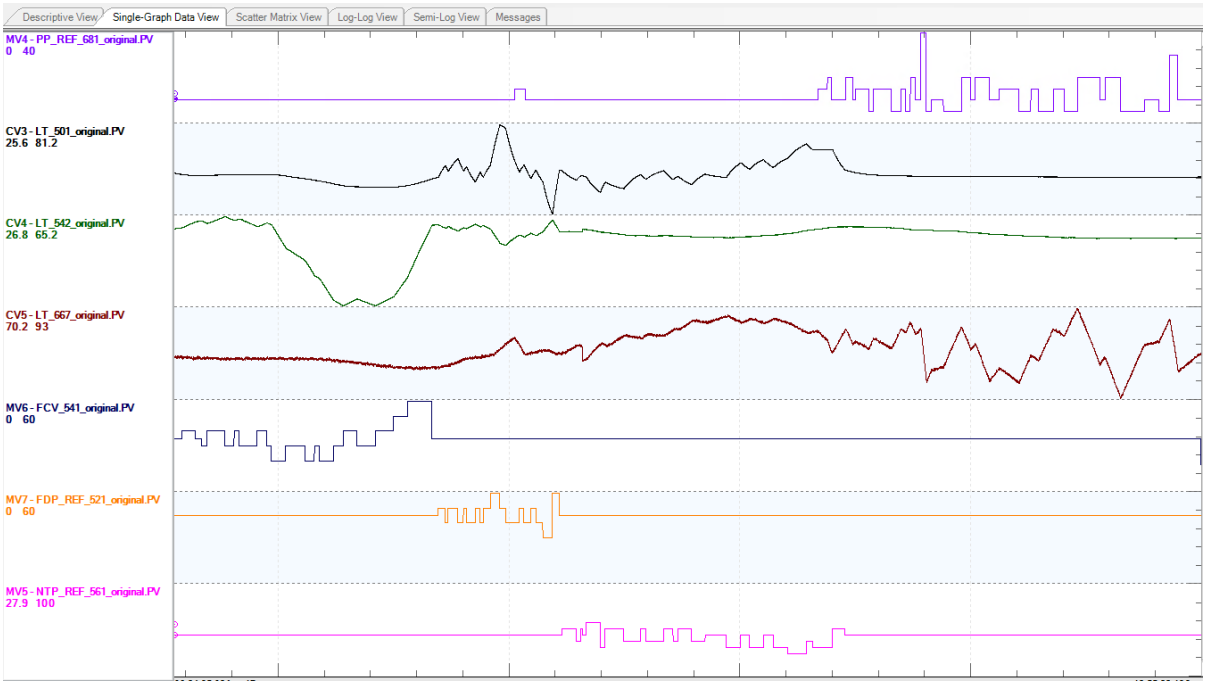


Figure 3.16 Level steps in Data Warehouse of PSES project

3.3 Model Identification using PSDS

Model Identification was completed inside PSES with the built-in PDS modelling tool. This section will describe how the initial models for temperature and level in the second half of the Pilot Plant were obtained. It will provide the steps used to obtain models using PDS then present a table of the models found for the second half of the plant. These models were used offline to build the controllers, then the models were improved online using the Profit Stepper.

3.3.1 Steps Used for Model Identification

Inside PSES, right click **G(s) Model** then select **Create ModelID** as shown in Figure 3.17:

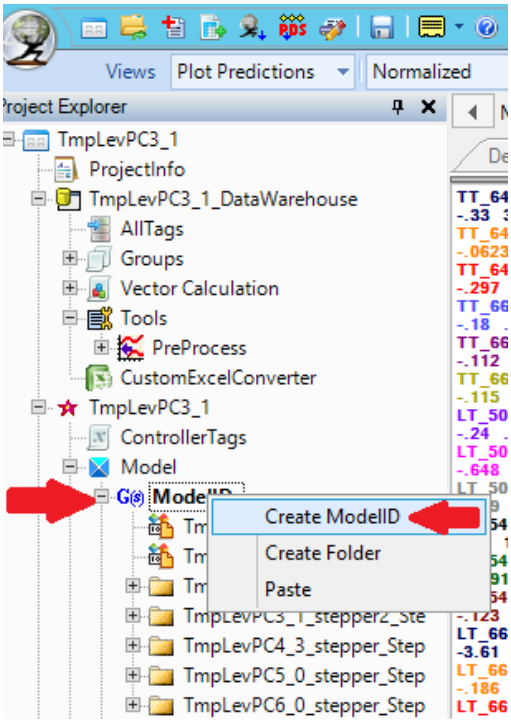


Figure 3.17 Create ModelID files in PSES

All variables imported into the Data Warehouse were available for selection. MV, CV and DV variables that were required for a Profit Controller were dragged into the SelectVariables Descriptive Info tab shown in Figure 3.18.

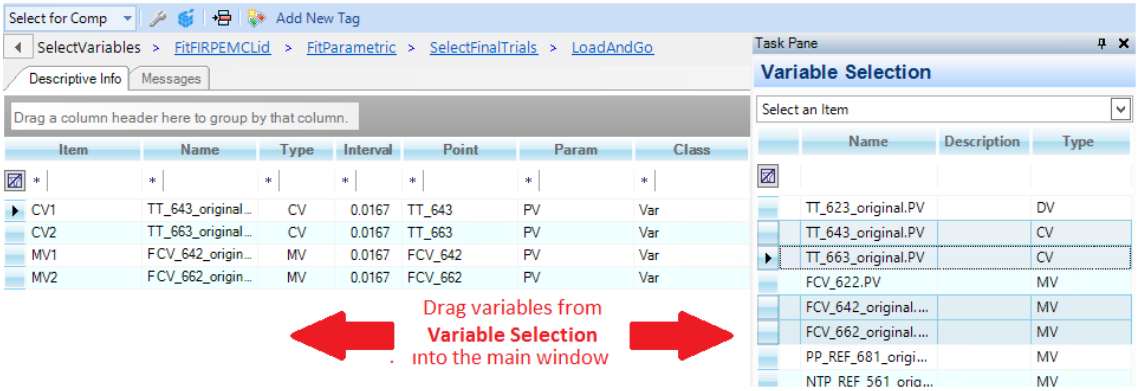


Figure 3.18 Variable Selection for Model Identification

Three trials of FIR algorithms were used for model identification. Figure 3.19 shows how the trials were set up for temperature modelling. The average setting time of the Temperature CVs for the initial step testing was 25 minutes as viewed in the Data Warehouse in Figure 3.15. Three Velocity trials of 20, 25 and 40 minutes were selected using the Characteristic Table shown in Figure 3.19. The Check and Correct button automatically reduced the number of coefficients of each FIR model to the recommended maximum of 30 without having to adjust the settling time of each trial. The trials were initiated with the Fit FIR button:

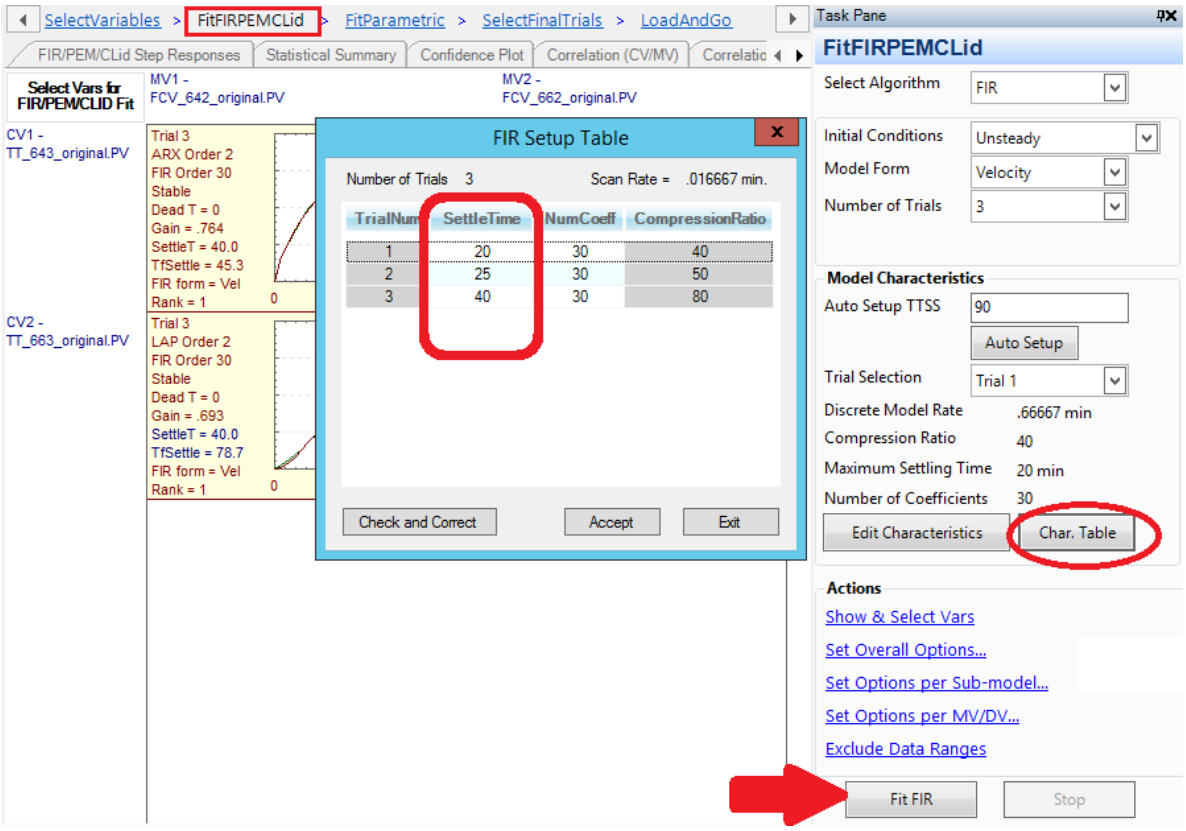


Figure 3.19 Select settling times for FIR trials based on CV settling times

For tank levels, the option to select integrator models was used. Figure 3.20 shows where **Integrating Sub-Process** was selected under **Set Options per Sub-model** to find integrator models for tank levels.

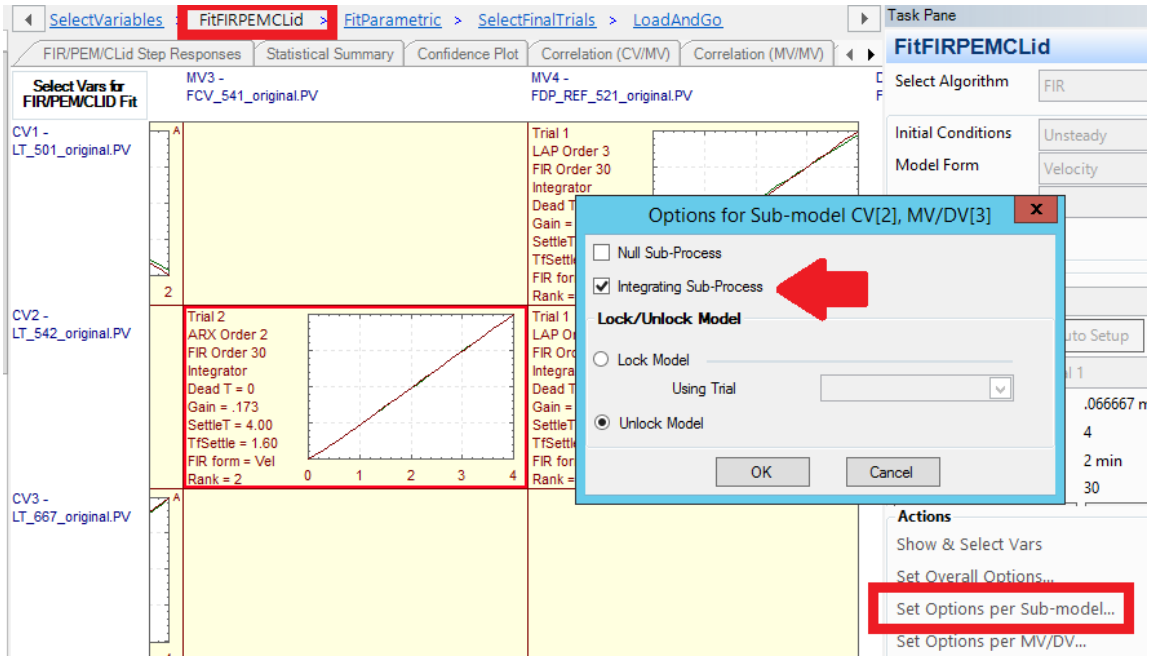


Figure 3.20 Select Integrating Sub-Process for tank levels

Parametric Models were fitted to the FIR models using the Best of Both Laplace and Discrete methods as recommended by Honeywell. Wherever the transfer function settling time was more than 100% larger than the actual CV settling time, the Profit Control build would not accept the model. New models had to be obtained by changing the settling time of the FIR trials, or by double clicking on the transfer function, editing the Individual Parametric Options then executing another parametric fit for that transfer function. Settling times were highlighted blue when this occurred as shown in Figure 3.20 for the model found between MV1 and CV1.

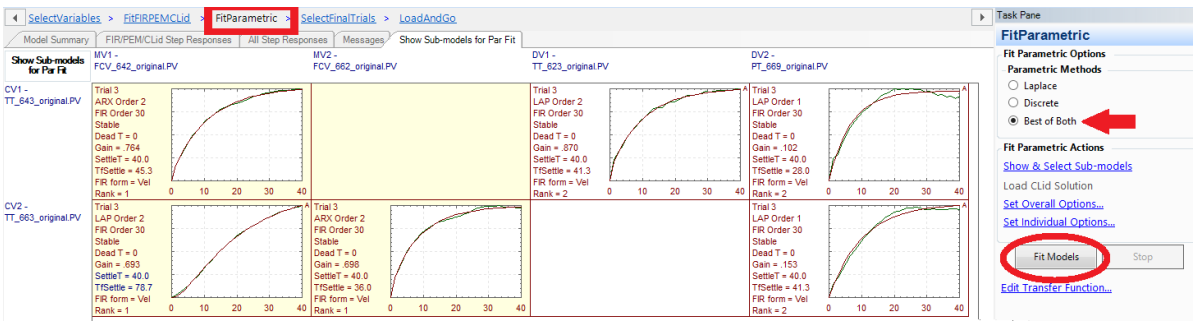


Figure 3.21 Fit Parametric models using Best of Both Laplace and Discrete Methods

The final model transfer functions were produced under the Select Final Trials tab shown in Figure 3.21. The Select Trials Source radio buttons determined which trials would be used – either User Selected or Auto Best Mixed. After pressing **Update Trials**, Velocity form predictions were available for inspection which showed the predicted behaviour of CVs using these parametric models and step testing data from the Data Warehouse. Pressing **Load Source to Final** produced the final model matrix of transfer functions ready for building Profit Controllers or Profit Simulators as shown in Figure 3.22.

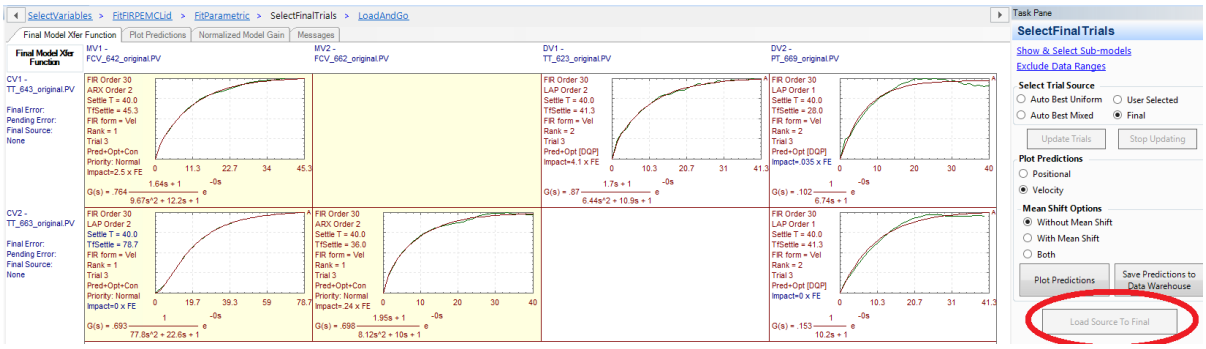


Figure 3.22 Load Source To Final produces the matrix of transfer functions required to build a Profit Controller

3.3.2 Transfer Functions Found by Offline PDS Model Identification

The complete matrix of transfer functions found during offline Model identification for the second half of the Pilot Plant are shown in Table 3.8 below. PDS assigns each model a confidence Rank from 1 to 5 based on how statistically significant the models are, and nulls models that are indistinguishable from noise. Rank 1 and 2 models can be used for control, rank 3 models may be used with caution, but lower ranks should be nulled. In this project, models of Rank 1 and 2 were used to build Profit Controllers, models of Rank 3, 4 and 5 were nulled.

Table 3.8 Transfer function matrix from first attempt at offline PDS modelling

	MV1 CSTR2 Steam FCV_642	MV2 CSTR3 Steam FCV_662	MV3 Raw Water FCV_541	MV4 Flow Dist. Pump FDP_521	MV5 Needle Tank Pump NTP_561	MV6 Product Pump PP_681	DV1 Steam Pressure PT_669
CV1 CSTR2 Temp TT_643	$\frac{(0.764) * (1.64s + 1)}{9.67s^2 + 12.2s + 1}$ $Rank = 1$						$\frac{0.1017}{6.7404547s + 1}$ $Rank = 2$
CV2 CSTR3 Temp TT_663	$\frac{0.579}{73.8s^2 + 17.2s + 1}$ $Rank = 2$	$\frac{(0.698) * (1.95s + 1)}{8.12s^2 + 10s + 1}$ $Rank = 1$					$\frac{0.1530}{10.168327s + 1}$ $Rank = 2$
CV3 Needle Tank level LT_501				$\frac{0.85380363}{4.4444 * 10^{-9}s^3 + 0.000133s^2}$ $Rank = 2$	$\frac{-0.3711408}{4.4444 * 10^{-9}s^3 + 0.000133s^2}$ $Rank = 2$		
CV4 Non Linear Tank Level LT_542			$\frac{0.17296705 * (0.11745818s + 1)}{0.19303426s^2 + s}$ Rank = 2	$\frac{-0.185968 * (0.00144993s + 1)}{0.059615407s^2 + s}$ $Rank = 2$			
CV5 CSTR# Level LT_667						$-0.40465 \frac{1.420424s + 1}{1.4717845s^2 + s}$ $Rank = 2$	

PDS does not produce the same transfer function repeatedly from the same data, and the models are not always reliable. Depending on the trial settling times used models of Rank 1 – 5, or no model at all, could result from the same data. It was also possible to obtain different Rank 1 models from the same data. An example is the Rank 1 model for CV1 versus MV1 in Table 3.8 which is:

$$Transfer\ Fcn = \frac{(0.764) * (1.64s + 1)}{9.67s^2 + 12.2s + 1}$$

Rank = 1

By altering the Trial settling times, the following Rank 1 models were produced:

$$Transfer\ Fcn1 = \frac{(0.64285326) * (1.0392262s + 1)}{5.1170897s^2 + 9.3978481s + 1}$$

$$Transfer\ Fcn2 = \frac{(0.62934059) * (1.64s + 1)}{9.67s^2 + 12.2s + 1}$$

The Simulink plot in Figure 3.23 shows all three models have similar settling times and the gain ranges from 0.63 to 0.76, yet all three are deemed suitable for control from Rankings. Similar results were obtained for the other sub-models.

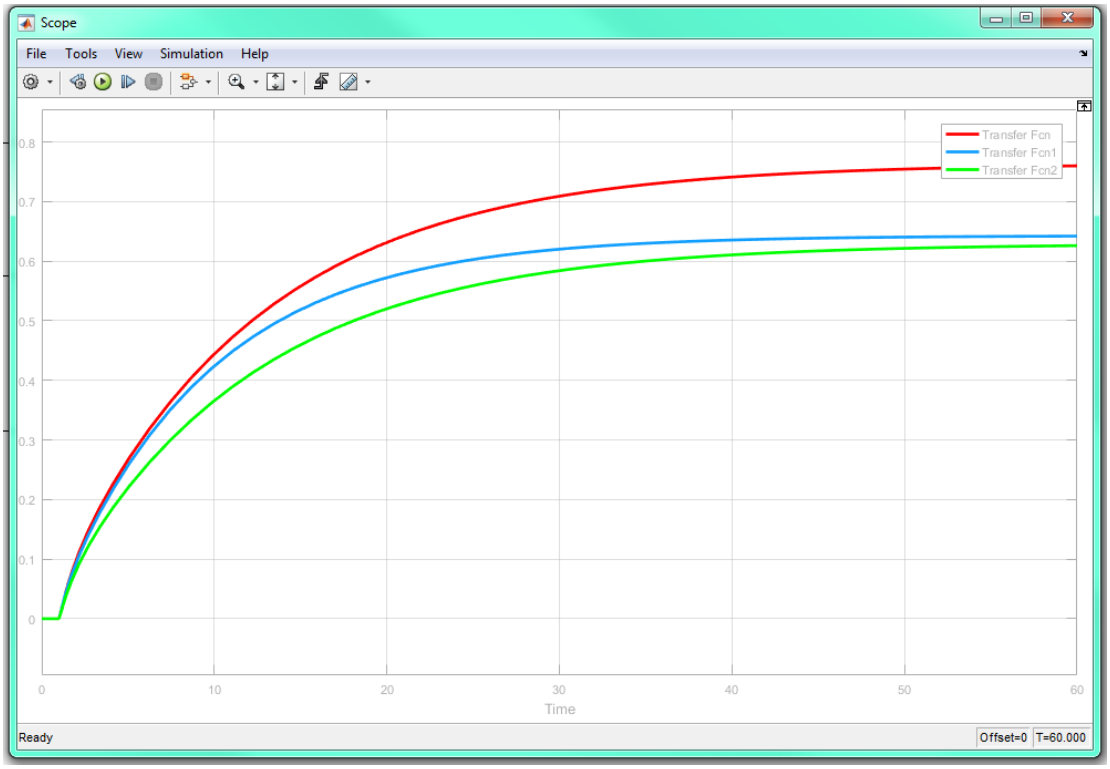


Figure 3.23 Simulink plot shows a range of Rank 1 models obtained from the same data for steam valve FCV_642 and CSTR2 temperature TT_643

Using PDS it was possible to get Rank 1 and 2 models when PDS found relationships between variables that were just noise. The models for temperature could be compared to those derived from least squares, but it was challenging to ascertain high ranked level models were valid without prior knowledge of the process, even when the polarity of the gain was correct. Some coefficients of the integrator transfer functions in Table 3.8 were so small as to effectively be zero. Based on these findings, transfer functions were later obtained online with the Profit Stepper by modifying both the step amplitude to increase the signal to noise ratio SNR and selectively targeting data to model from the data collector.

Figure 3.24 is a Simulink plot comparing least squares derived models (in minutes) to PDS models for temperature in CSTR2 and 3. The PDS models have longer time constants and higher gains.

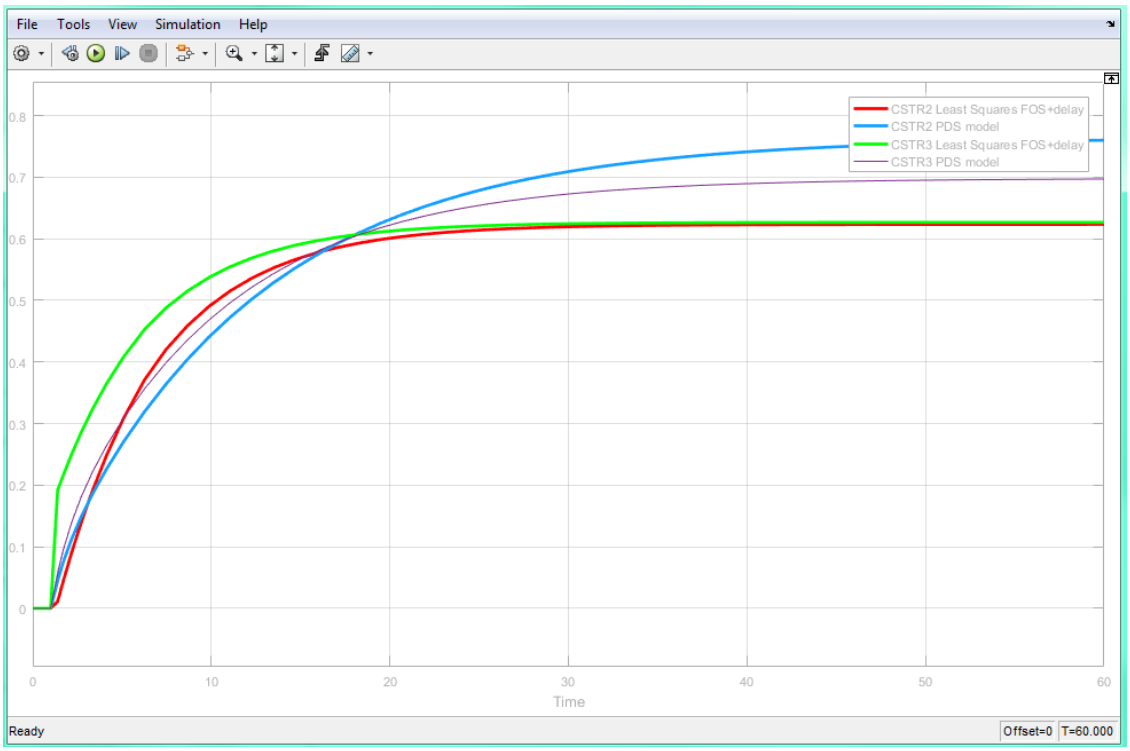


Figure 3.24 Comparison of least squares derived models to PDS models for temperature in CSTR2 and CSTR3

3.4 Building Profit Controllers

Profit Controllers were built after model identification was completed using PSDS. The form of each Profit Controller depended on the matrix of MVs, CVs and DVs used for the initial modelling in PSDS. These relationships were inherent in the model files used to build each controller and determined the type of connections required between Profit Suite and Experion. Models found from initial step testing between an OP point from a valve PID and level CV required different Point connections and Base Level Controls from models found using the SP and level CV of the same PID loop. After Point connections were decided, the main design considerations were Profit Controller Execution Rate and Base Level Controls. This section will describe all the steps as used in this project to build new profit controllers and connect them to the Pilot Plant. It includes OPC configuration identified during this project which allows the Profit Suite applications to communicate with each other and Experion PKS.

3.4.1 Adding Experion PKS Assets to OPC Server

Simulated PID controllers on the Experion2 simulation server were used to test OPC connections and practice building Profit Controllers. Initially, Profit Suite had no communication with Experion2 and could not validate any connections to points within Control Modules. To fix this issue, assets in Experion2 were added to the OPC server using Configuration Studio as shown in Figure 3.25:

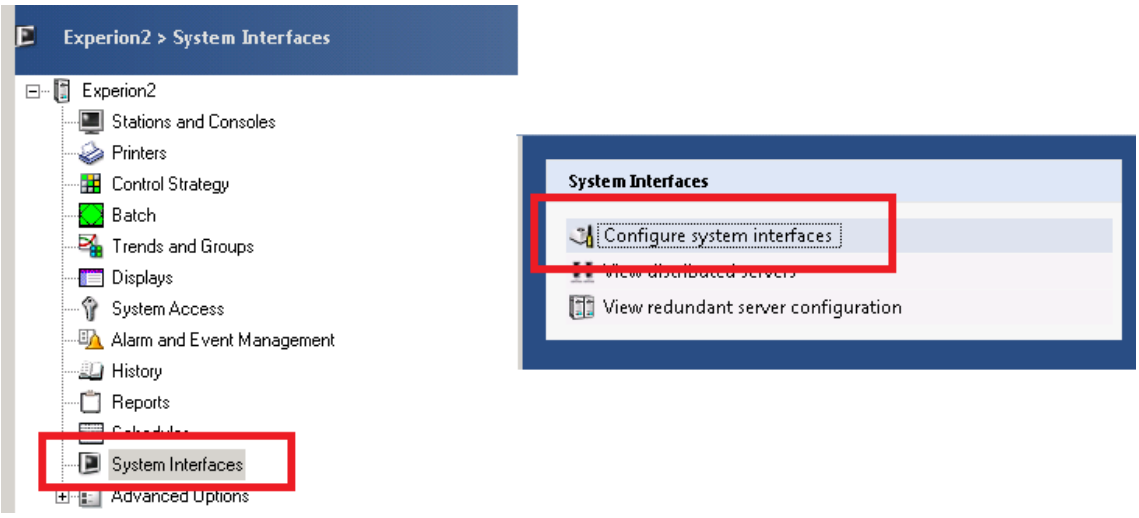


Figure 3.25 Find OPC server settings in Configuration Studio

Experion Assets were added to the OPC server by opening the Experion2_OPC link in the location shown in Figure 3.26:

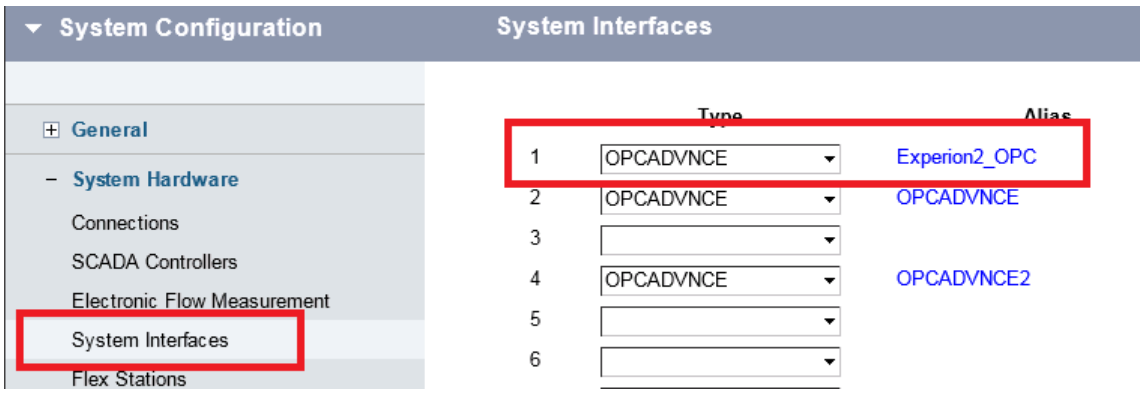


Figure 3.26 Opening Experion2_OPC server settings

Under Scope of Responsibility, the Associated Asset was opened as shown below in Figure 3.27. Then the top Parent 'Assets' was selected to add all simulated assets under it to the OPC server. This made all simulated CMs visible when browsing from Profit Suite. It was then possible to validate point connections and build working Profit Controllers that controlled PID blocks in Experion2. This is how Profit Control point connections were tested without risk to the live Pilot Plant server.

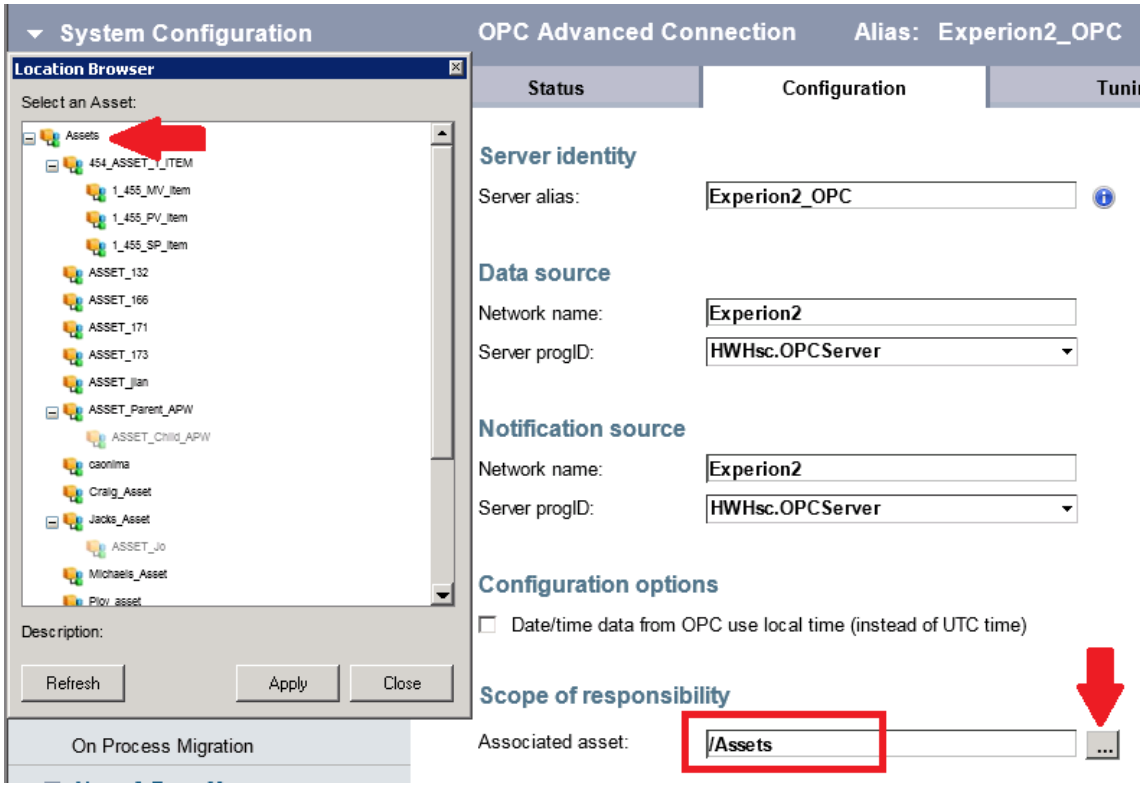


Figure 3.27 Add Parent Assets to OPC server to make them visible to Profit Suite

3.4.2 Execution Rate and Interval Offset

The Execution Rate is the interval between each time the Profit Controller reads process values, calculates predictions, then writes new MV moves to Experion to control the plant. The Execution Rate, also called Control Interval, is measured in minutes. The Execution Rate is fixed during the build and cannot be changed without rebuilding and reinstalling the Profit Controller. Honeywell's guidelines for calculating the Execution Rate of a profit controller based on CV settling times are:

- 200-300 control intervals for the CV with the longest settling time
- The shortest CV settling time should contain more than 10 control intervals (Honeywell International 2016a)

These recommendations provide a balance between controller performance and computer processing power. Increasing the Execution Rate reduces the amount of memory and processing power required to recalculate the models for very large Profit Controllers with many CVs and MVs. This is balanced against the risk of the Profit Controller not reacting fast enough to correct disturbances in CVs with short settling times (Honeywell International 2016a).

Honeywell also provides the following rule of thumb:

- $Profit\ Controller\ Settling\ Time = 4 \times Major\ Time\ Constant$ of the process CVs

From step tests, temperatures in the CSTRs had the longest time constants of approximately 400 seconds each. Therefore, Settling Time for the Profit Controllers was calculated as:

$$PC\ Settling\ Time = 4 \times 400s = 1600s$$

$$PC\ Settling\ Time\ \div 300\ Control\ Intervals = \ 5.33s$$

The Needle Tank level CV had the shortest settling time at approximately 5 minutes, which would contain 56 Control Intervals with an Execution Rate of 5.33s:

$$5min \times \frac{60s}{1min} = 300s$$

$$300s \div 5.33s\ = 56\ Control\ Intervals$$

From these calculations, 5 second Execution Rates were selected which were rounded to 0.084 minutes by PSES.

An Interval Offset is used to prevent all Profit Controllers from executing simultaneously and creating a spike in network demand. Staggering the execution intervals spreads the load on the CPUs and communications network over time. The offset is measured in minutes. Controller X with an Execution Rate of 1 will execute each minute. Controller Y with Execution Rate of 1 and Offset of 0.1 also executes every minute, but six seconds later than X (Honeywell International 2016a). At most, two Profit Controllers were run simultaneously during the project when both halves of the pilot plant were being controlled by Profit Controllers for level and temperature. Therefore, the Interval Offset was left at zero for all controllers.

3.4.3 Base Level Controls and Shed Modes

Base Level Controls are templates which serve two functions:

- Configure specific Experion Points to connect each Profit Controller CV, MV and DV to.
- Determine Shedding Modes – predetermine what will happen to the plant control system when the Profit Controller switches off.

Table 3.9 contains the BLC Templates used in this project and their meaning. The BLC for DVs and CVs was selected automatically and connected to the DACA block in the CM. The BLC for MVs was user defined and depended on whether a Profit Controller used the SP or the OP of a PID block as an MV.

Table 3.9 BLC Templates used to connect Profit Controllers to Pilot Plant

BLC Template	Control System	PID Block Mode for Profit Control	PID Block Mode after Shedding	MV used for Profit Control	Variable Type
HW_EPKS_Ctrl-P Auto_Shed-O Auto	Honeywell Experion PKS	Program Auto	Operator Auto	SP	MV
HW_EPKS_Ctrl-P Man_Shed-O Auto	Honeywell Experion PKS	Program Manual	Operator Auto	OP	MV
HW_EPKS_Ctrl-P Man_Shed-O Man	Honeywell Experion PKS	Program Manual	Operator Manual	OP	MV
HW_EPKS_CV_PV	Honeywell Experion PKS	-	-	-	CV
HW_EPKS_DV_PV	Honeywell Experion PKS	-	-	-	DV

3.4.3.1 PID Block Control Modes for Profit Control

Using PID block SP parameters as MVs required the PID blocks to be in Program Auto for Profit Control and a **Ctrl-P Auto** BLC Template. This control mode cannot be selected directly in Station. The PID modes available to students from Station were Program Manual and Cascade Program. This constraint was designed to allow students to change PID modes and read/write OPs and SPs values using links in Station, deliberately limiting their access to the Experion code to reduce the risk of code corruption. When a user selects Auto for a CV from a drop down box in Station, the PID block is put into Cascade Program mode. In this context Cascade simply means the PID block draws its SP from any upstream Function Block, not necessarily another PID block.

The parameters of the PID block in every CM used as an MV had to be changed in Experion to enable the Profit Controllers to write to its SP Point. These parameters were changed as follows:

1. Open the monitoring tree in Configuration Studio showing green CM icons

2. Open CM containing an MV – e.g. FCV_622 for CSTR1 steam valve

3. Scroll to locate the PID Block and double click on a blank area (not a parameter) of the FB to bring up the Parameters window shown in Figure 3.28

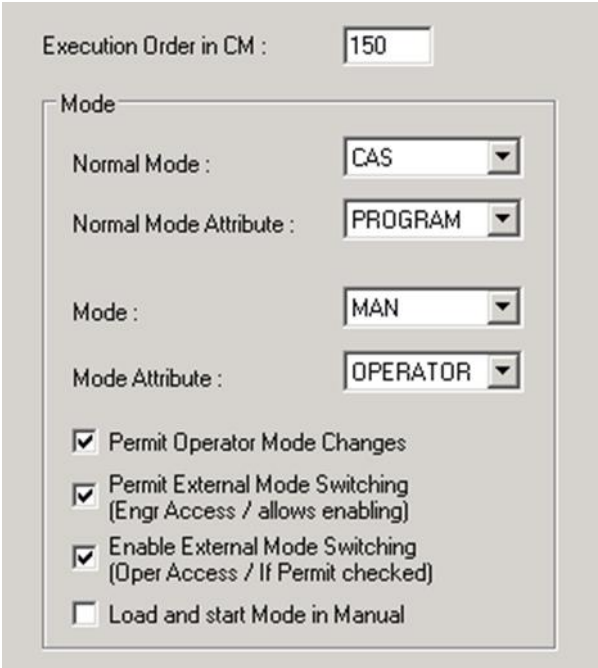


Figure 3.28 Enable PID Block External Mode Switching

4. Ensuring the block mode is first in **MAN** using Station, uncheck both **Enable External Mode Switching** and **Permit External Mode Switching**

5. Repeat for all CMs containing MVs

Any control mode and attribute combination could then be selected using the PID block Faceplates in Station shown in Figure 3.29. These Faceplates were accessed by either double clicking on a Valve icon or the Set-point window of a pump then selecting:

- MD = Program
- MD attr. = Auto

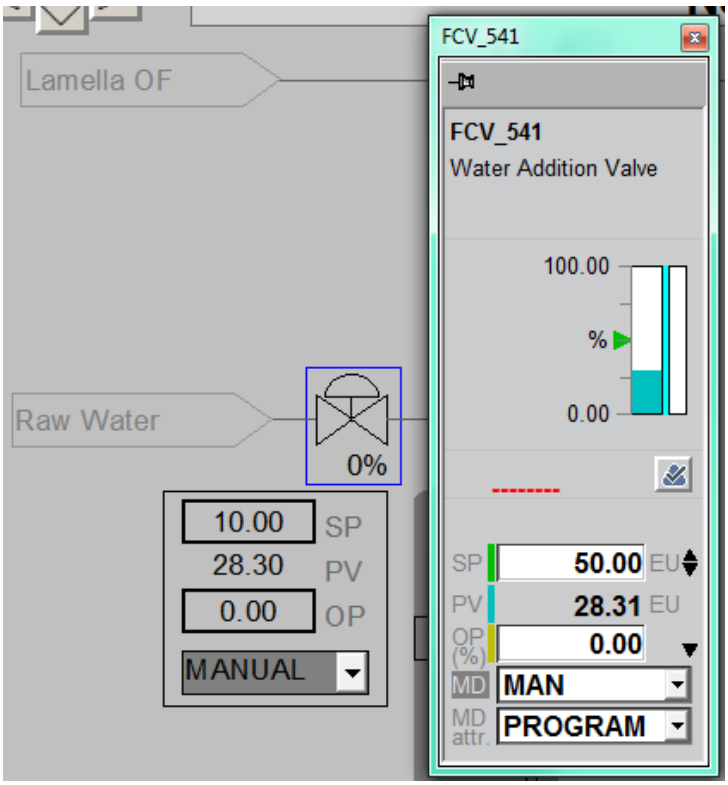


Figure 3.29 Double click Raw Water Valve icon to access Faceplate

The modes shown in the drop-down boxes in Station were invalid until the checkboxes in the PID block Parameters window were restored each time Profit Control testing ended.

The steps required when SP was used as an MV were:

1. Select a BLC containing CTRL-P Auto and connected to SP point during the Profit Controller build

2. Disable external mode switching with PIDA Block parameters as described above

3. Select Program Auto mode and attribute using PID Faceplate via Station

4. Reverse this procedure upon completion of Profit Control testing

3.4.3.2 Experion Code Changes to allow Profit Control with OP as MV

To use the OP point as MV, the BLCs require Manual Program mode to determine that the PID Block is controllable. Unfortunately, the PIDA blocks in Experion have inputs for manually setting the OP via Station when a PIDA is in Manual Program mode. When first trying the **Ctrl - P Man** BLC, the PID blocks were taking the OP value from the upstream OP input block pin, and the OP value from Profit Suite was ignored. This prevented Profit Control from directly controlling a valve position Output. A full explanation of a previous CM code example with a diagram is provided in Appendix A.

The simplified example shown in Figure 3.30 below shows why Profit Controllers could not write to the OP of the PIDA blocks. The value of FDP_521 is controlled by a Station user or an Excel Spreadsheet controller. The value held in FDP_521 is fed back via N11_306 then SWITCHB to the OP input pin of PIDA preventing Profit Control from changing the PIDA OP value.

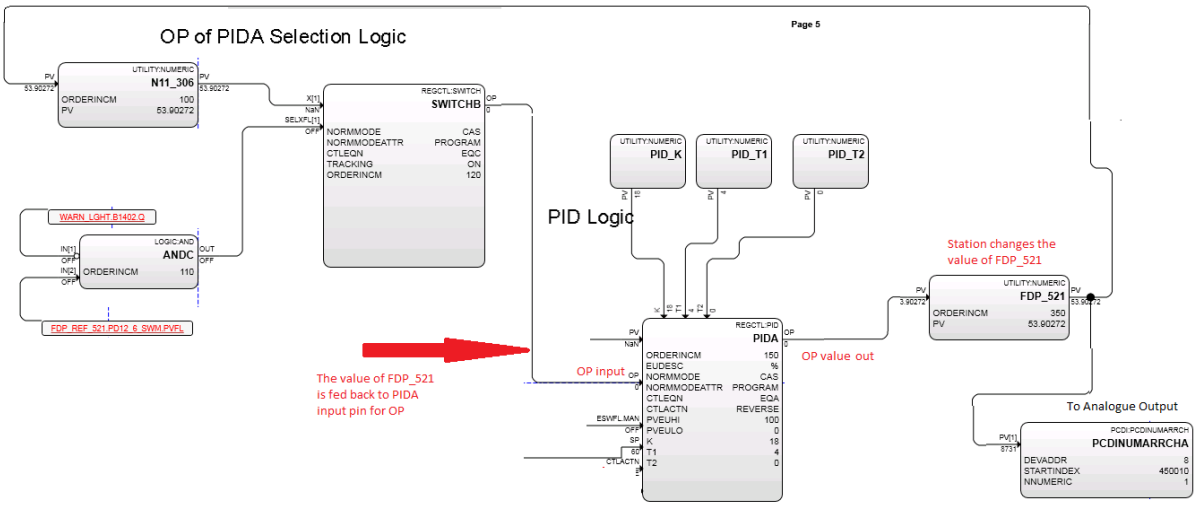


Figure 3.30 Simplified code example shows PIDA block OP is fed back from FDP_521 block

Experiments on the Experion2 simulation server confirmed that Profit Controllers could write to an OP point when the wire to the OP input was deleted. The code shown in Figure 3.31 was then developed to selectively remove the input to the PIDA OP point when a utility flag is set. When the Initial Flag Value checkbox is set in the Profit_Control Utility Flag block the OP input pin value on PIDA is set with a SELREAL logic block to NAN. This allows Profit Controllers to write to the OP point. When the flag is unchecked the OP value is pulled from the Station page input as usual. The order in which these FB execute inside each CM was set in the range between SWITCHB and PIDA to ensure the code worked properly. This code was successfully implemented on all temperature and level PIDA CMs in the Pilot Plant. Comments describing how to use it were saved inside each CM as shown in Figure 3.31.

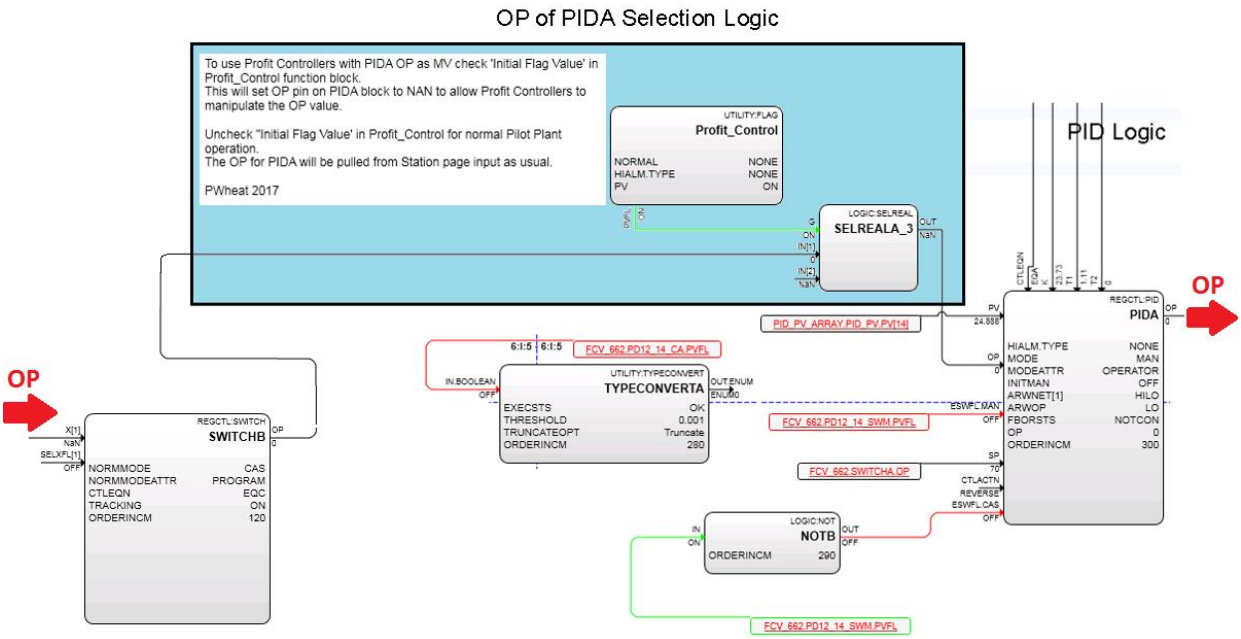


Figure 3.31 Utility Flag block sets the OP pin of PIDA block to NAN for Profit Control using OP

Following these code changes, the steps required when OP was used as an MV were:

5. Select a BLC containing CTRL-P Man and connected to OP point during the Profit Controller build

6. Disable external mode switching with PIDA Block parameters as described in 3.3.2.1

7. Select Program Manual mode and attribute using PID Faceplate via Station

8. Check the "Initial Flag Value" box in the Profit_Control Utility FB inside Experion CM

9. Reverse this procedure upon completion of Profit Control testing

3.4.3.3 Shed Modes

A CV or MV is user defined as critical whenever it is preferable for a Profit Controller to switch itself off rather than continue to operate without this CV/MV should it become unavailable. Variables can be set as critical or not by Managers inside PSOS while the controller is running. When the Profit Controller loses the ability to control or communicate with a critical variable it will shed control of the plant back to the operator.

When the controller sheds, the Experion PID block modes were switched into either Manual or Auto modes, both with Operator attributes by the Watchdog CM. Essentially, Watchdogs contain timers which reset each time a Profit Controller executes. When the Watchdog does not receive this signal from the Profit Controller, it changed the modes of the PID Blocks to return the plant to Operator control. This also happened when the Profit Controller was simply switched off.

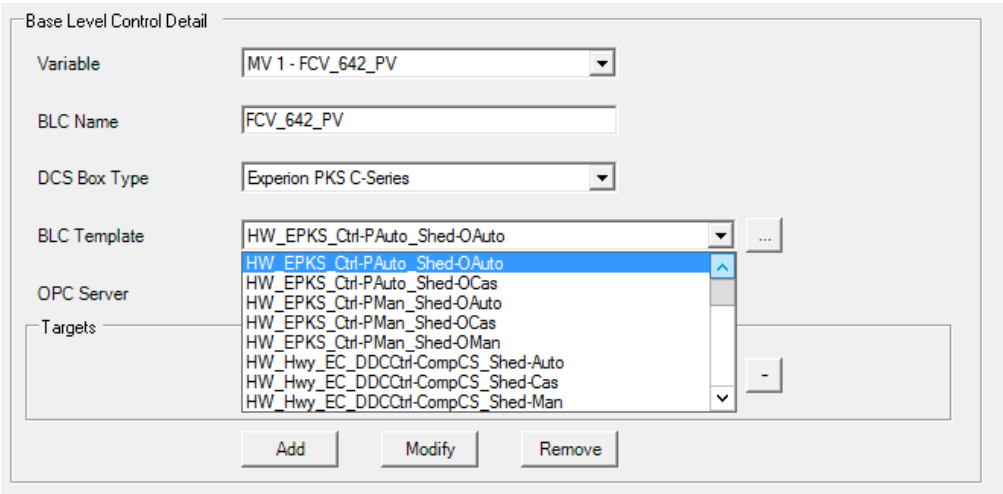


Figure 3.32 BLC Templates for Honeywell Experion PKS with Shed Modes

Generally, PID blocks were shed to Operator Auto as shown in Figure 3.32. This meant the associated CV (temperature or level) was returned to automatic PID regulatory control whenever the Profit Controller switched off. The CVs for PID regulatory control are predefined by code inside the Experion CM. Therefore, three exceptions which were shed to Operator Manual mode were:

- Feed Pump FP_141: If shed to auto the CV is Supply Tank 02 Level which was out of service. Operator Manual mode was used instead to fix the inflow into BMT
- Cyclone Recycle Pump CRP_341: If shed to auto the CV is the recycle flow rate. Operator Manual was selected to keep level in CUFT and BMT stable
- Flow Disturbance Pump FDP_521

Both FDP_521 and NTP_561 have the level in the Needle Tank as the CV when in PID control as illustrated by Figure 3.33.

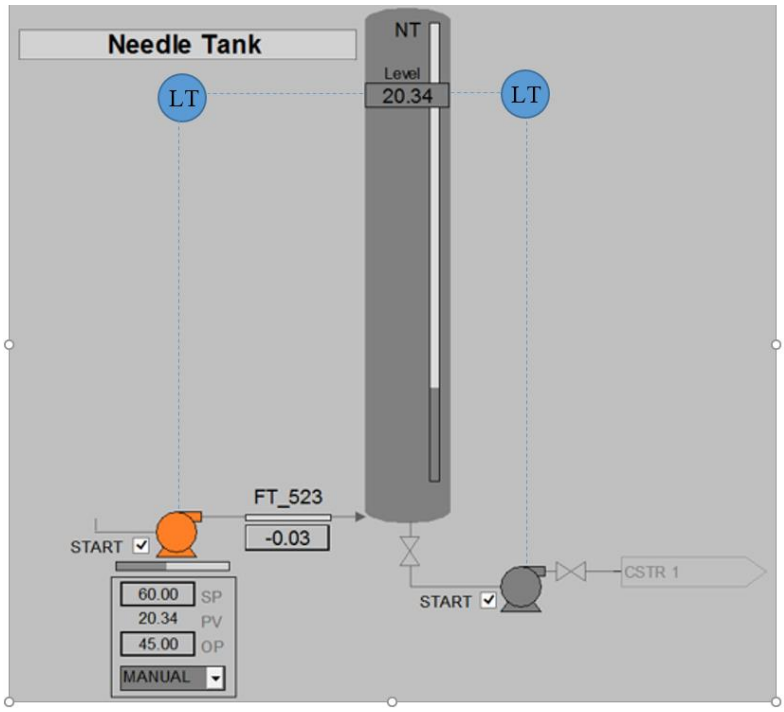


Figure 3.33 FDP sheds to Operator Manual and NTP sheds to Operator Auto

Where both MVs were shed to Operator Auto, the PID controllers would interact adversely as disturbances and the process became unstable. For this reason, FDP_141 was shed to Operator Manual and NTP was shed to Operator Auto. With FDP in Manual mode, the outflow from the NLT and inflow to NT were both fixed. The Raw Water Valve and the NTP pump were shed into Auto mode to control the CVs. Table 3.10 lists the Shed Modes used for MVs in this project which left the plant in a stable condition.

Table 3.10 Shed modes used for stable PID Operator control

MV	FCV_622 FCV_642 FCV_662	FCV_541 NTP_561 PP_681	FDP_521	**MV**	FP_141 CRP_341	BMP_241 CUP_361
Shed Mode	O Auto	O Auto	O Man	**Shed Mode**	O Man	O Auto

It was important to learn that the SPs and OPs for each CV are handed back to the operator with whatever values they last held from the Profit Controller. The operator was still required to intervene to select sensible SP and manual OP values via Station Faceplates even though careful BLC selection meant the plant was shed to stable PID control.

3.4.4 Steps to create a new Profit Controller

The major steps used to create then connect an operational Profit Controller to the Pilot Plant were:

- Complete Model ID in PSDS prior to building controller or use pre-existing model files copied from other platforms (Profit Steppers/merged models from other controllers)
- Create and configure a new Profit Controller in PSES
- Import controller files into PSRS and create a new Profit Controller application in PSRS
- Connect Profit Controller points to Experion points, validate the OPC connections and build Watchdog
- Edit OPC connection in URT Explorer
- Import then load the Watchdog into Experion
- View the newly created and running Profit Controller application using PSOS

3.4.4.1 Create and configure a new Profit Controller in PSES

Multiple Profit Controllers of different configurations can be created inside a single PSES Project. A PSES project can contain model files from initial model identification using PDS and models created from any Profit Steppers in the PSES Project. These can also be merged with each other or with new models imported into the Project through the Data Warehouse.

In PSES, click on **Open Project** (if not open already from PDS model ID) as shown in Figure 3.34:

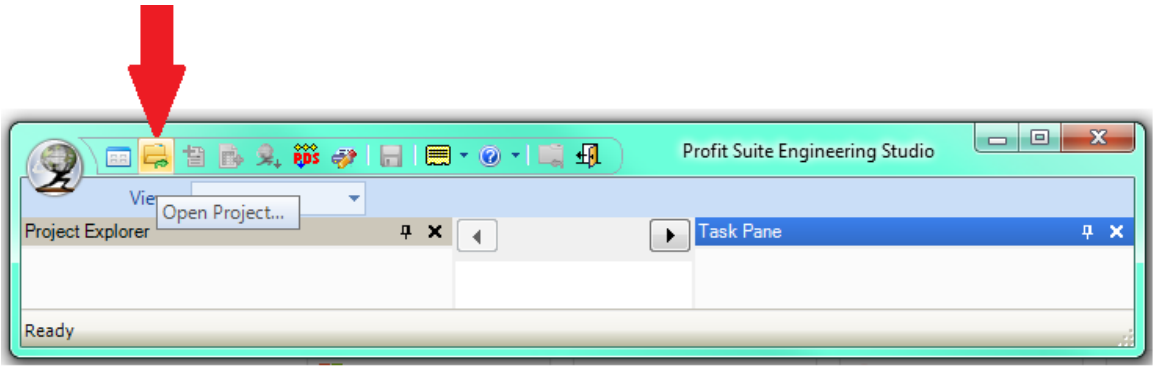


Figure 3.34 Open existing Project in PSES

Browse to select a project **SLN file type** then click **Open** as shown in Figure 3.35:

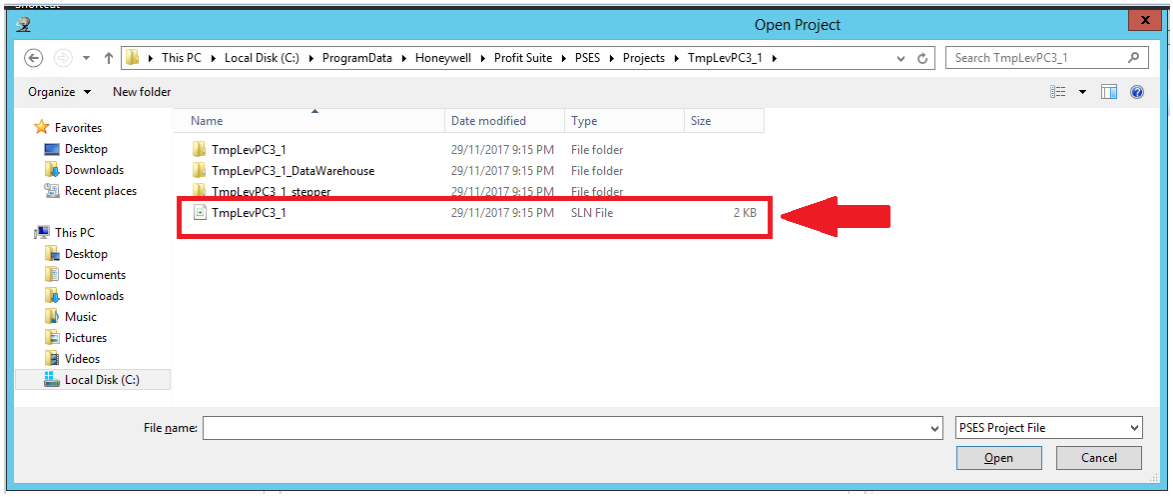


Figure 3.35 Select the Project file with SLN extension then click Open

In the Project Explorer tree, right click **Controller** > **Create Controller** then name it something meaningful as shown in Figure 3.36:

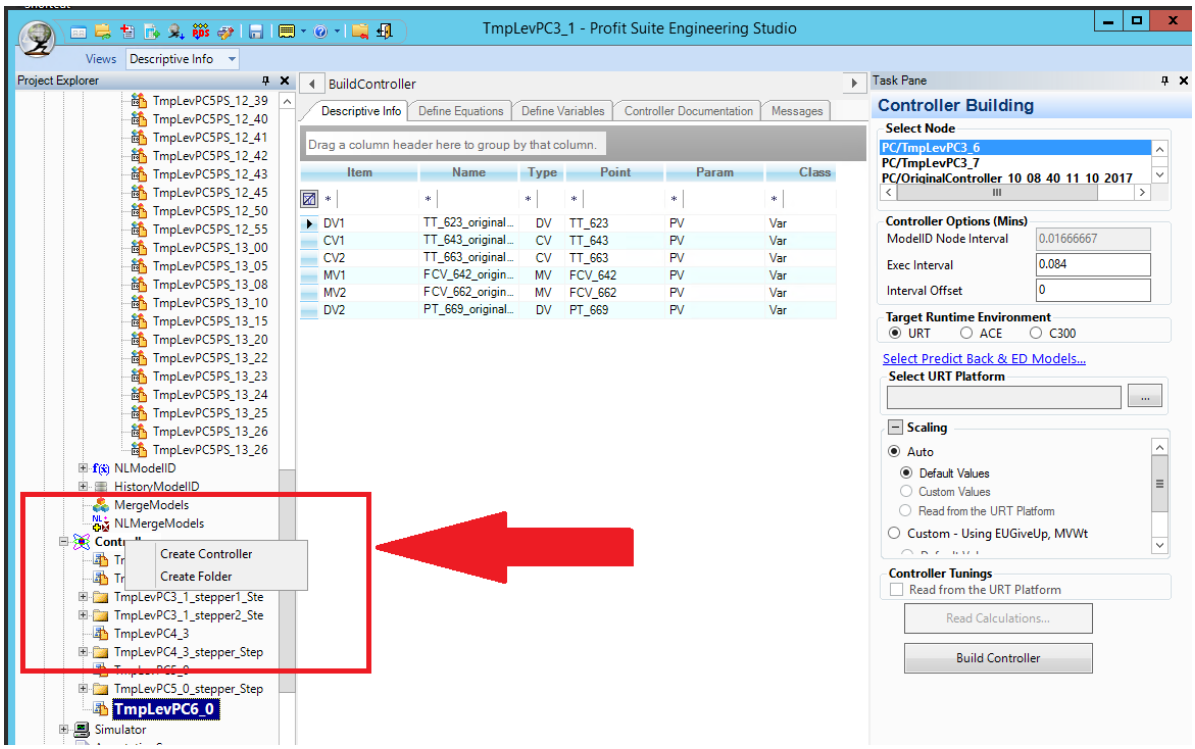


Figure 3.36 Create a new Profit Controller in Project Explorer tree

Model files must be added before building the controller. These files can be examined in detail in the **G(s) ModelID** nodes in the Project Explorer tree and were built from PSDS modelling or by Profit Steppers. Under **Controller Building** use **select node** to choose the desired model file, then set the **Execution interval** and **Interval offset.** Click on **Controller Documentation** to enter the name and write a description for the Profit Controller's objectives as shown in Figure 3.37:

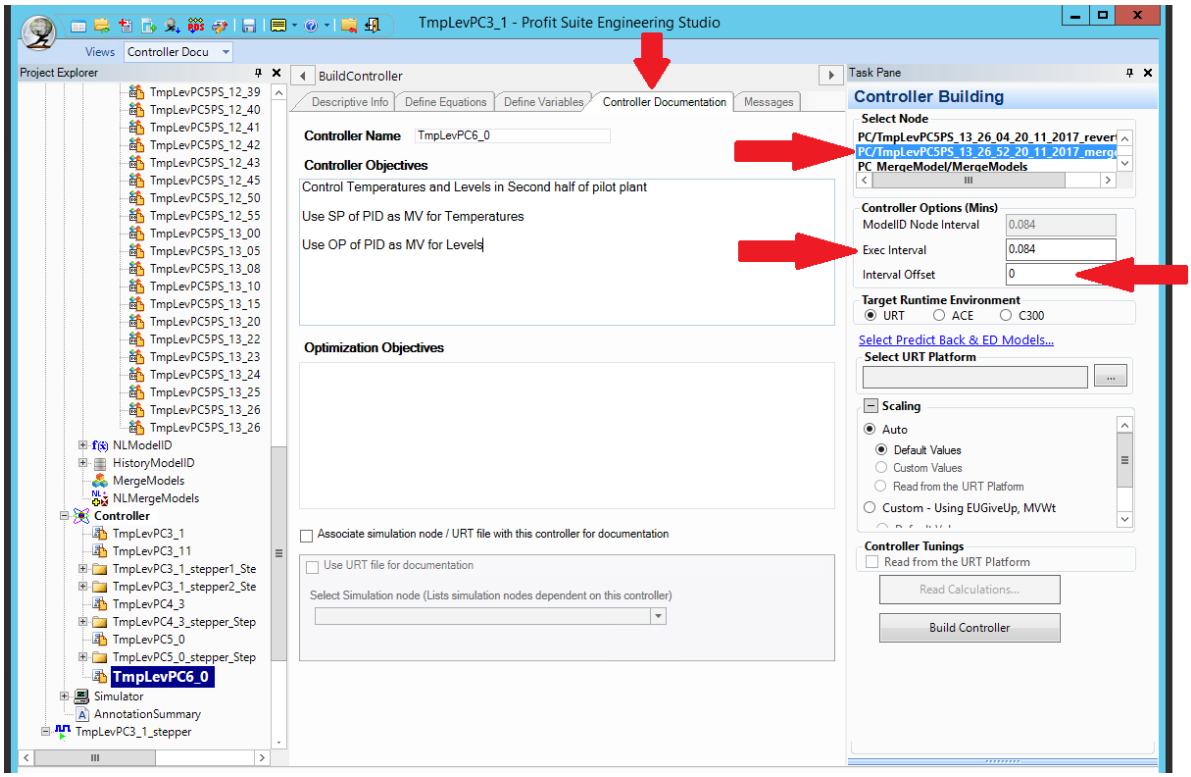


Figure 3.37 Select Model File Node, Execution Interval, Interval offset and document Profit Controller objectives

Click **Build Controller** then click **OK** to the resulting **User Changeable Scaling** popup as shown in Figure 3.38:

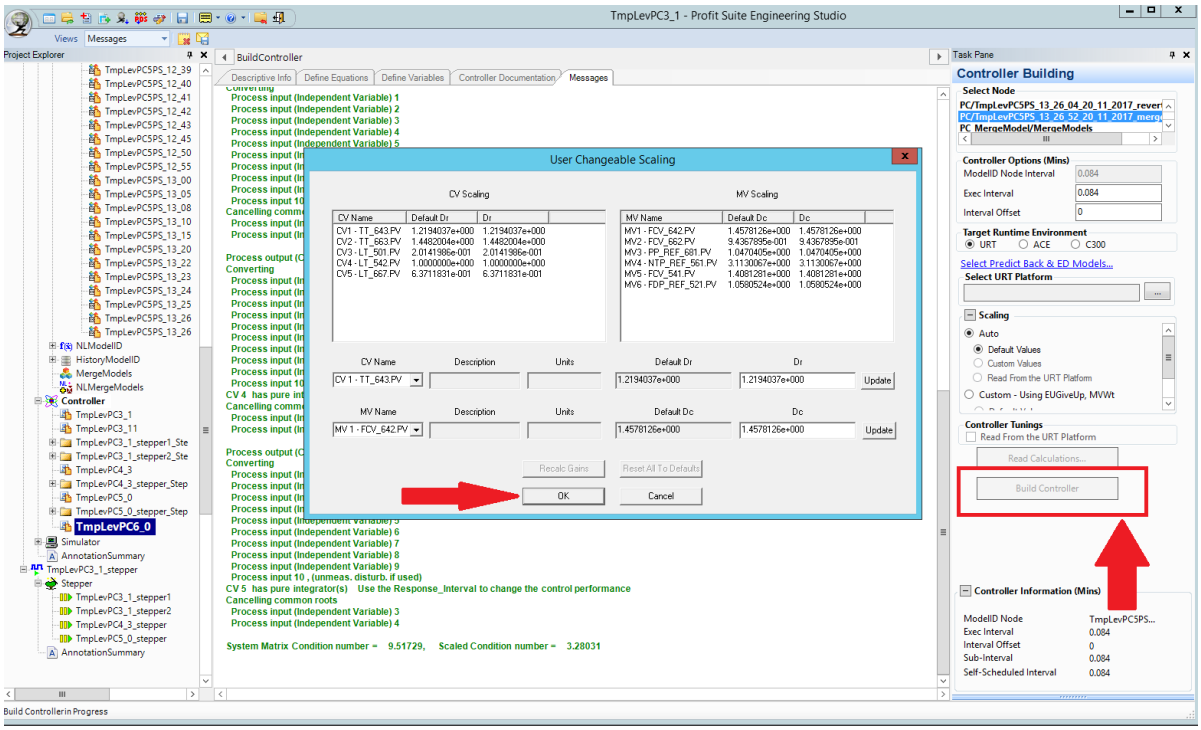


Figure 3.38 Build the newly configured Profit Controller

If the Profit Controller build has been successful "Build Controller Completed" will appear in the bottom left and Controller files will have been generated. The names and pathname for the location of the controller files is listed in the **Messages tab** and must be noted for PSRS. This is shown in Figure 3.39.

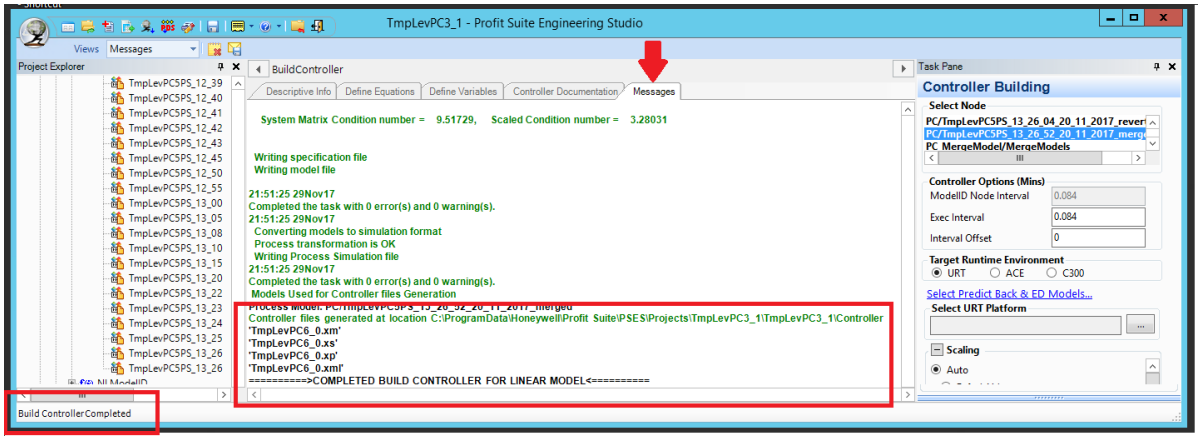


Figure 3.39 Profit Controller files and location shown in Messages tab

The Controller files take the following form:

NewPC.**xm**: the modelling data NewPC.**xp**: for simulating the process

NewPC.**xs** : contains controller settings

NewPC.**xml**: process and model data for 3rd party applications (Honeywell International 2016b)

C:\ProgramData\Honeywell\Profit Suite\PSES\Projects\NewPC_Project\NewPC\Controller

The newly created profit controller must next be connected to points in Experion using PSRS and URT Explorer.

3.4.4.2 Create and configure a new Application in PSRS

Open PSRS and click the **Create a new profit suite application** icon. Select **Profit Controller** in the popup window then click **OK** as shown in Figure 3.40:

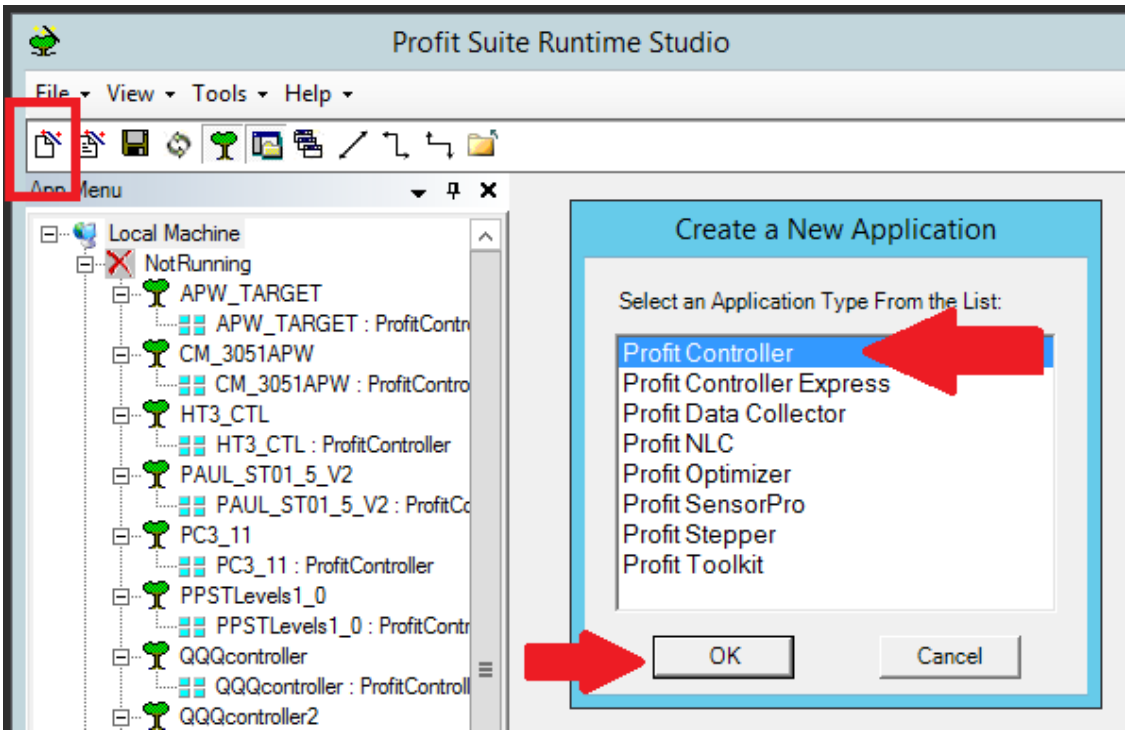


Figure 3.40 Create a new Profit Suite application in PSRS

Click **xm file** and type in the location where PSES created the controller files. Only the **xm file** need be selected as all others are automatically filled in, as shown in Figure 3.41. The Server field must be edited to contain the OPC server pathname that allow the Profit Controller to communicate with the correct Experion server:

Pilot Plant	**\\ppserver1\HWHsc.OPCServer**
Experion2	**\\Experion2\HWHsc.OPCServer**

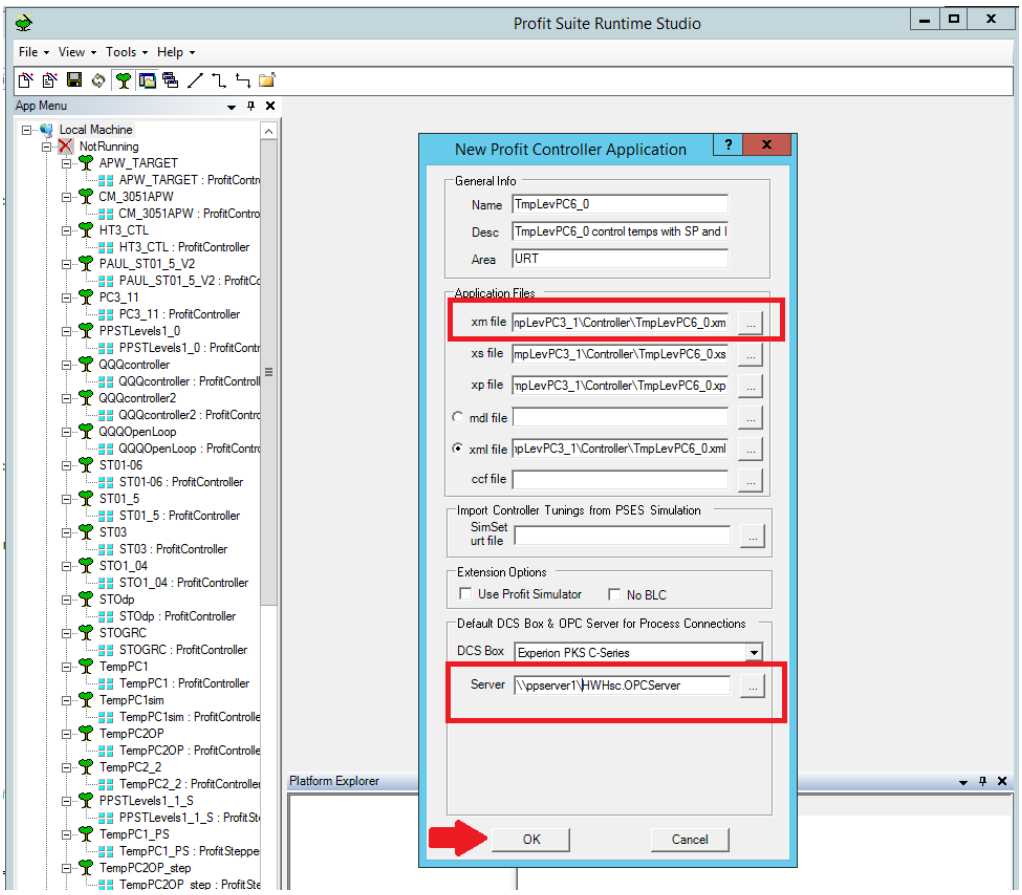


Figure 3.41 Select the XM controller files then edit OPC server path for either ppserver1 or Experion2

Click **OK** and the new Profit Controller Platform is created as shown in Figure 3.42:

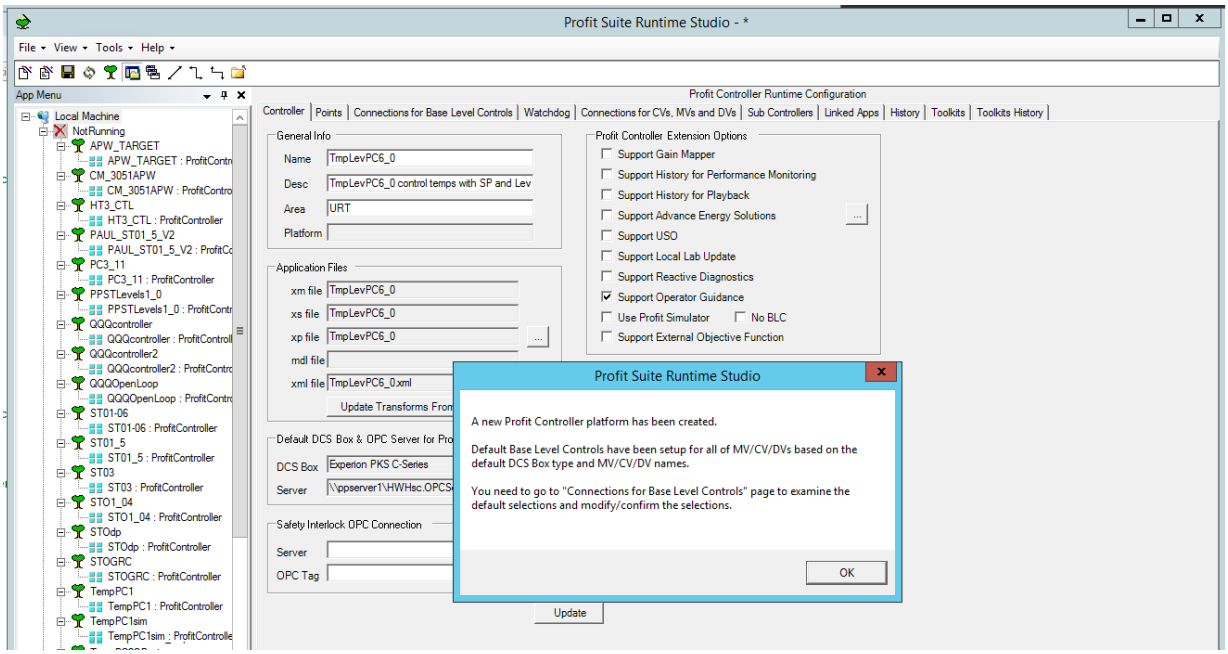


Figure 3.42 New Profit Controller is created

3.4.4.3 Connect Profit Controller points to Experion points, validate the OPC connections and build Watchdog

The tabs for **Profit Controller Runtime Configuration** must be worked through in order from **Controller** through to **Connections for CVs,MVs and DVs**, and then sub controllers if required. After filling out each tab, click **Update** before proceeding to next tab. If a mistake is made, return to **Controller** and start again, clicking update on each tab again before continuing to the next tab.

On the **Controller tab, c**heck that **Server** contains the correct OPC server address then click **Update** as shown in Figure 3.43:

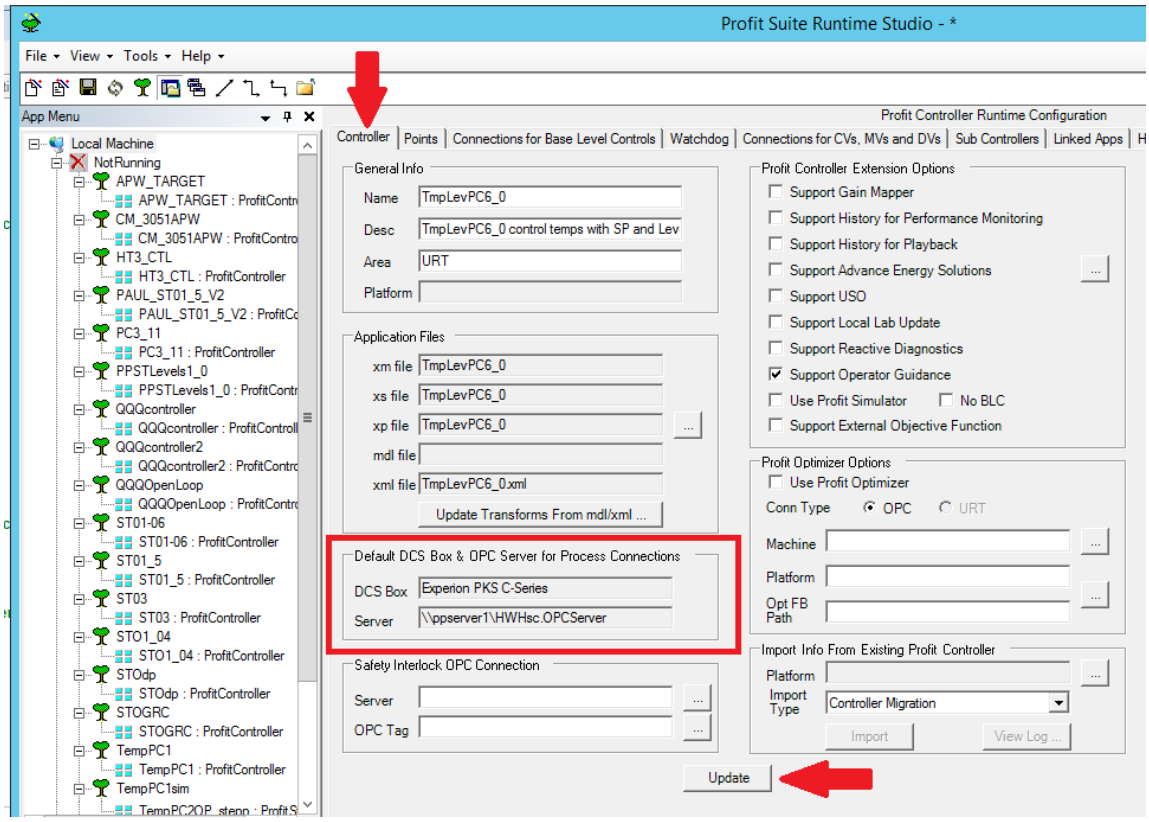


Figure 3.43 Check OPC server address on Controller tab and click Update

On **Points** tab, check that all CVs, DVs and MVs are in the correct categories. It is possible but not a requirement to enter descriptions and engineering units. Click **Update** as shown in Figure 3.44:

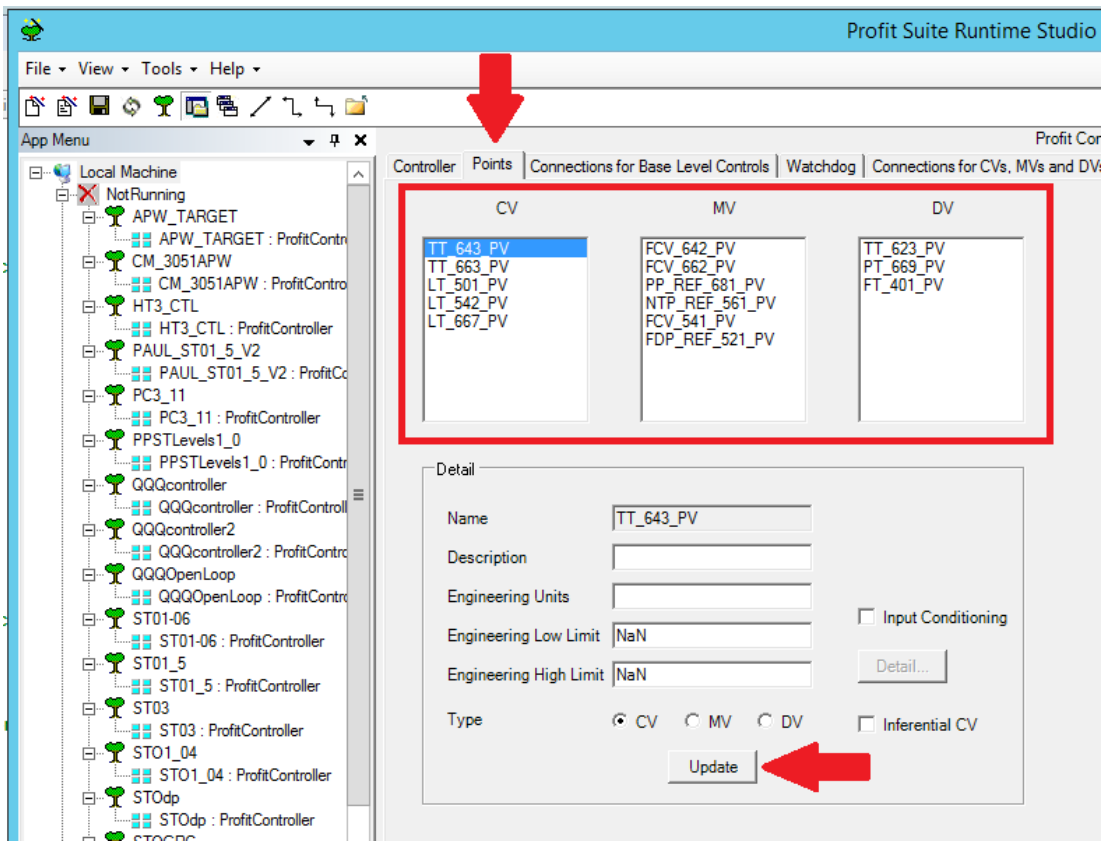


Figure 3.44 Review CVs DVs and MVs are in correct category then click Update

Connections for Base Level Controls Tab

Work through each MV, DV and PV and connect it to the correct point inside the correct Control Module on the Experion server. Points in Experion are called Targets. For each variable, browse to the correct Target in Experion server; FCV_642.PIDA.OP or FCV_642.PIDA.SP for example. If the Targets are not visible, use Configuration Studio on the Experion server to check the assets have been added to the OPC server. This has been done for Experion2 and ppserver1.

1. Select a **Variable**

2. Choose a **BLC template** for the type of **Target** that the **Variable** will connect to. It must have the desired Shed Mode for the Experion PID controller when the Profit Controller is switched off

3. Browse through Experion Assets to connect each **Variable** to each **Target**

Select a **Variable** and appropriate **BLC Template** from the dropdown list as in Figure 3.45:

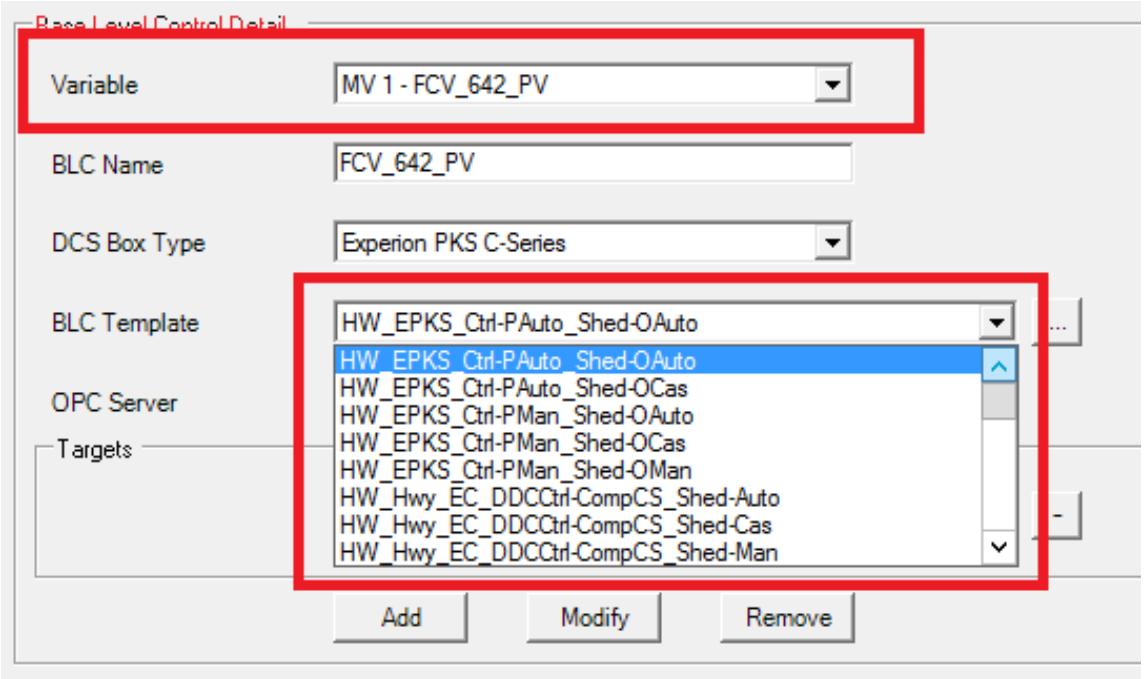


Figure 3.45 Choose Base Level Controls

To edit the Experion point the Variable will connect to, click the browse icon in **Targets (…)**. A window popup shows all the Assets visible on the OPC server. For Pilot Plant assets browse to **Assets** folder then **Pilot** folder. Figure 3.46 shows a list of folders which are the Control Modules in Experion on ppserver1:

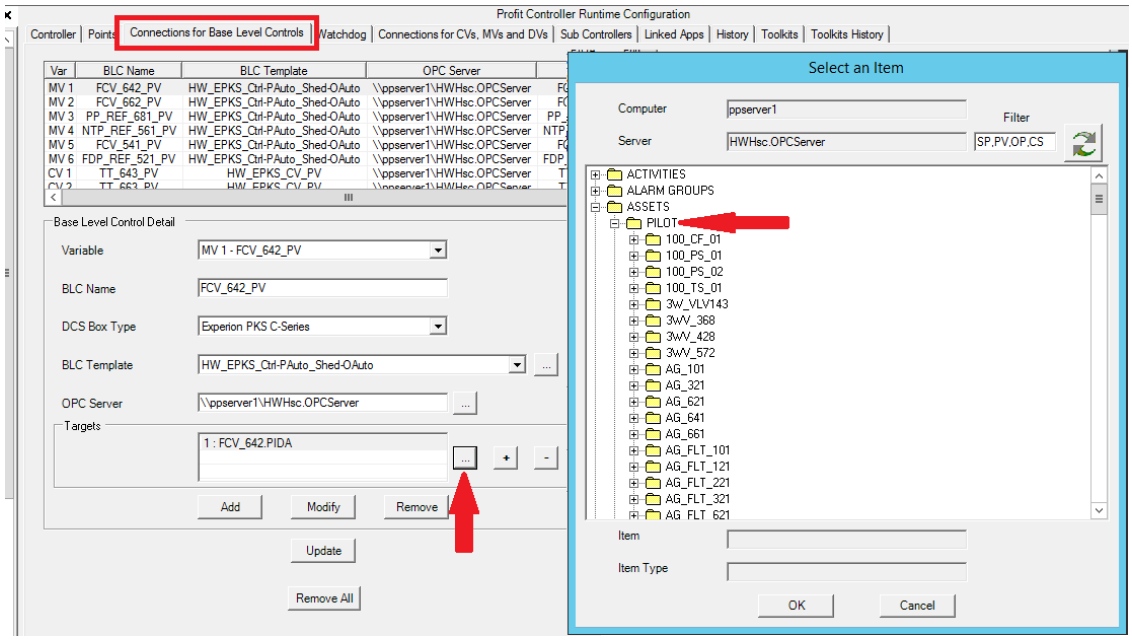


Figure 3.46 Browse Experion Control Modules to select Targets

Expand the correct Control Module, then select point to connect it. The example in Figure 3.47 shows a connection to the Set Point for the PID block of steam valve FCV_642 for use as an MV:

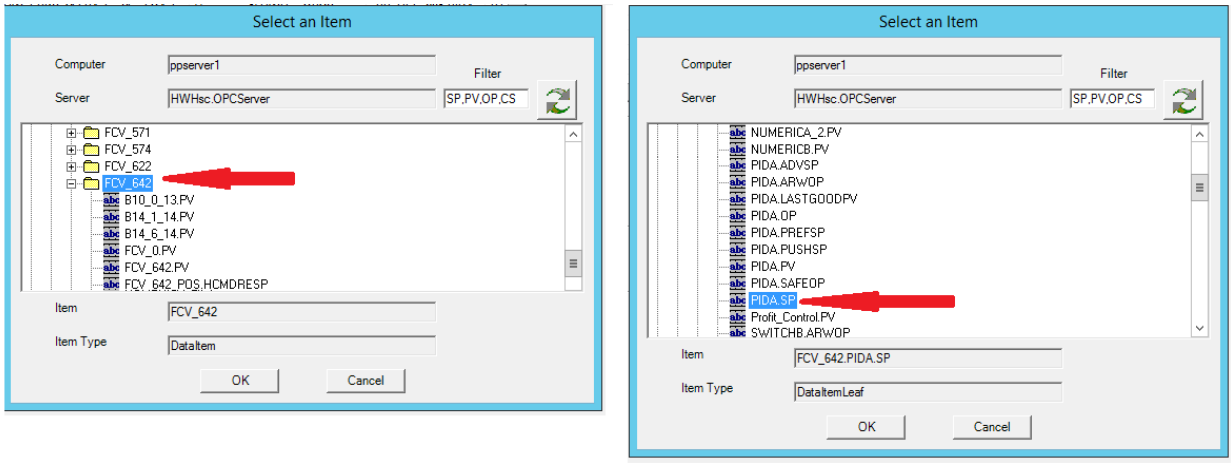


Figure 3.47 Selecting PID set point of steam valve FCV_642 as an MV

Use the + and – icons to ensure only one Target is selected per Variable. Click **Modify** after selecting the **Target** for each **Variable** as shown in Figure 3.48:

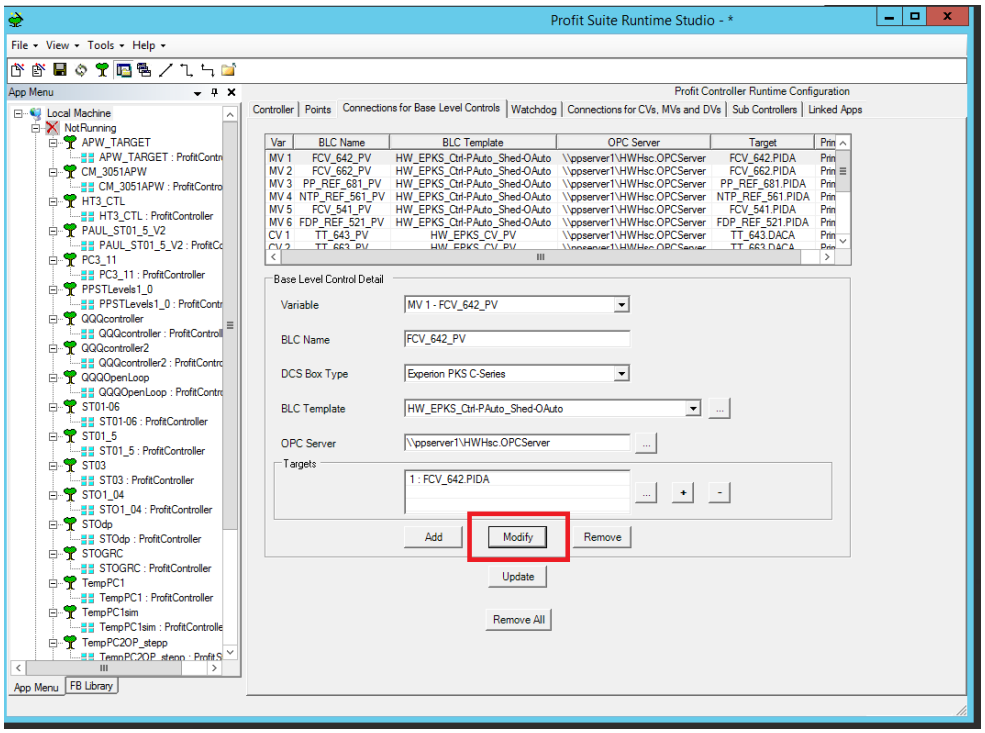


Figure 3.48 Modify after editing Target without creating multiple Targets per Variable

The Targets for CVs and DVs are DACA points in Experion CMs, shown in Figure 3.49.

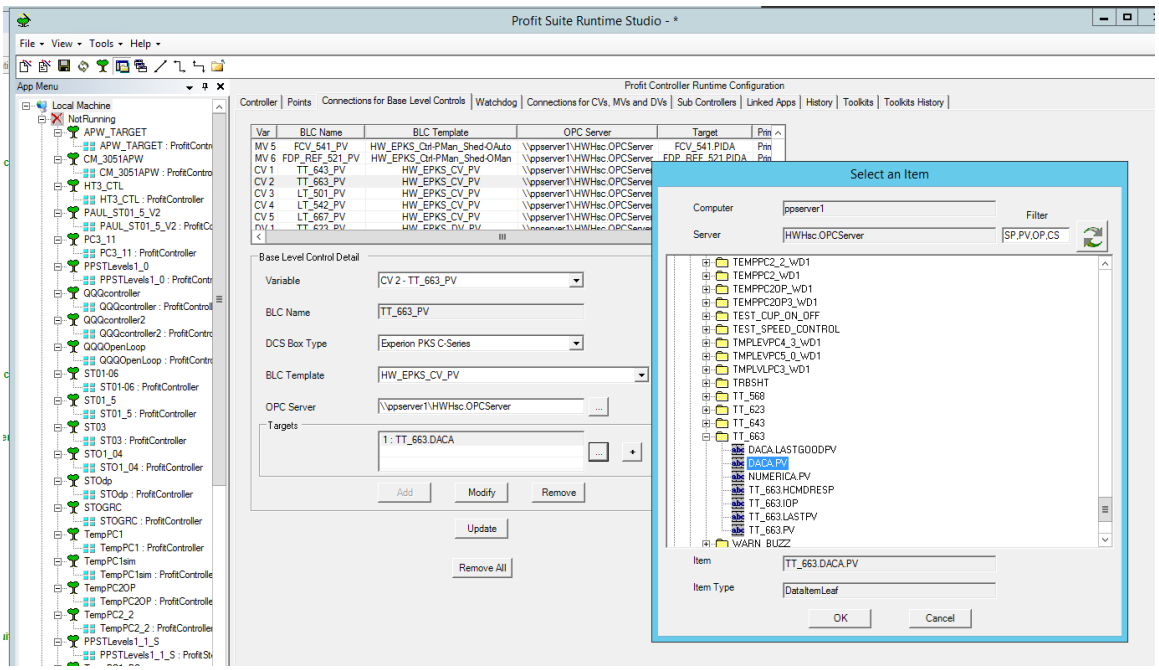


Figure 3.49 CVs and DVs are connected to DACA.PV points in Experion CMs

Repeat this process for all **Variables** clicking Modify after editing each one. Click Update only after all **Variables** have been configured. Figure 3.50 shows that the **Watchdog XML** file is automatically created - remember the name and location of this file:

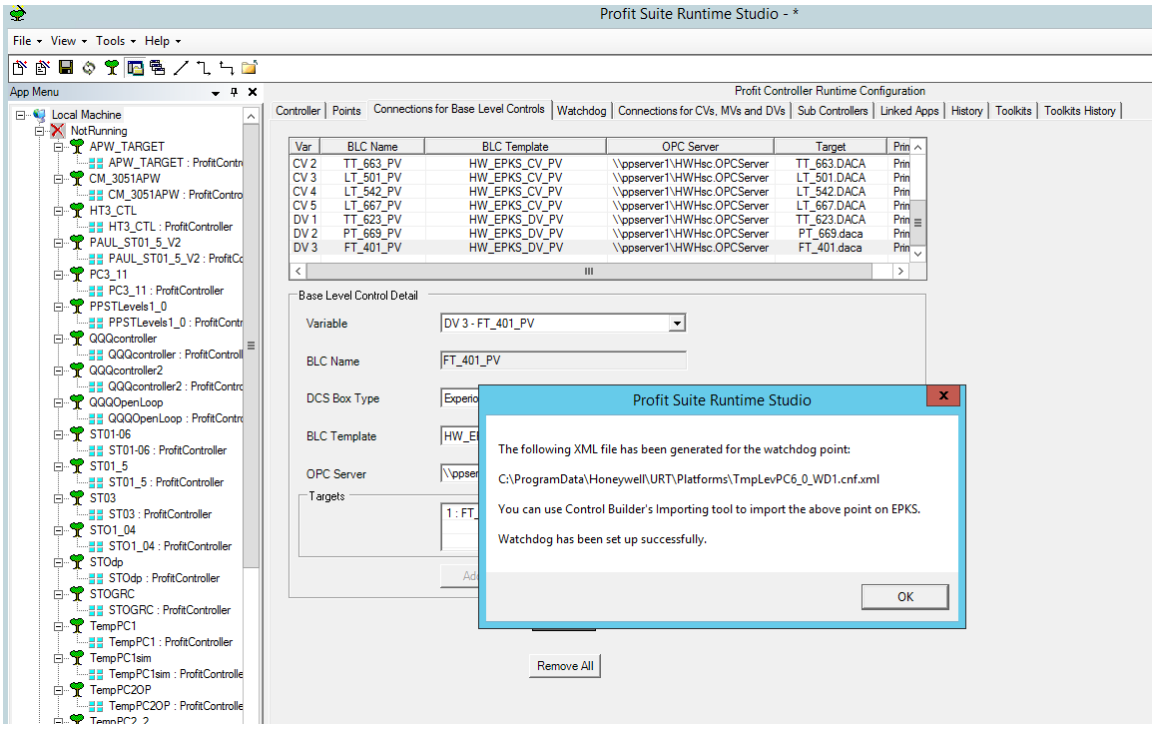


Figure 3.50 Remember the name and location of the Watchdog XML file

Click **OK** to proceed to the **Watchdog** tab. Click **OK** to ignore the popup window concerning primary and secondary swaps. On the **Watchdog** tab review the configuration data then click **Update** if everything is correct. Disregard the primary secondary swap warning. A message appears to confirm all Base Level Controls have been set up correctly as shown in Figure 3.51:

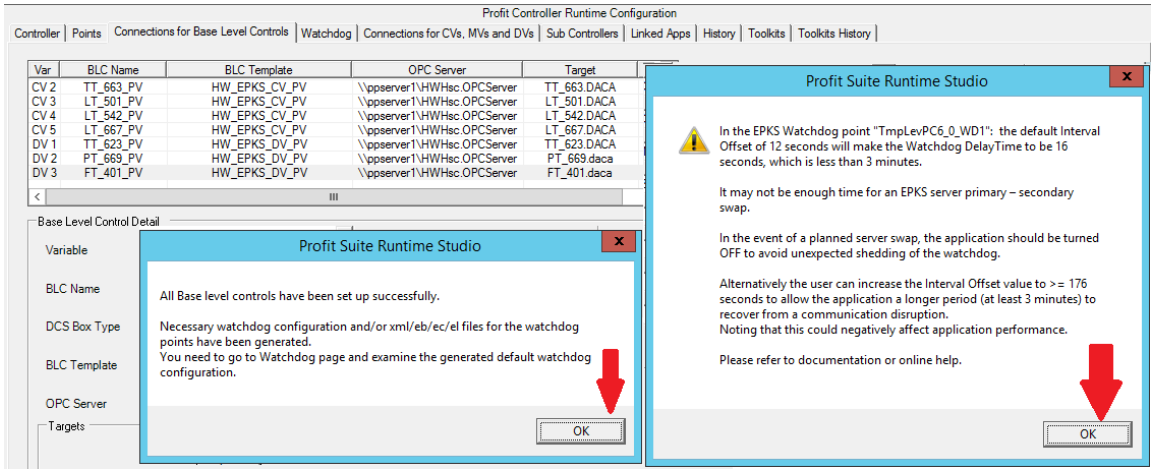


Figure 3.51 Base Level Controls set up successfully

Click **Validate All** on the **Connections for CVs, MVs and DVs** tab as shown in Figure 3.52. This will verify that the Profit Controller can communicate with all configured Targets inside the Experion control modules via the Experion OPC server (ppserver1 or Experion2).

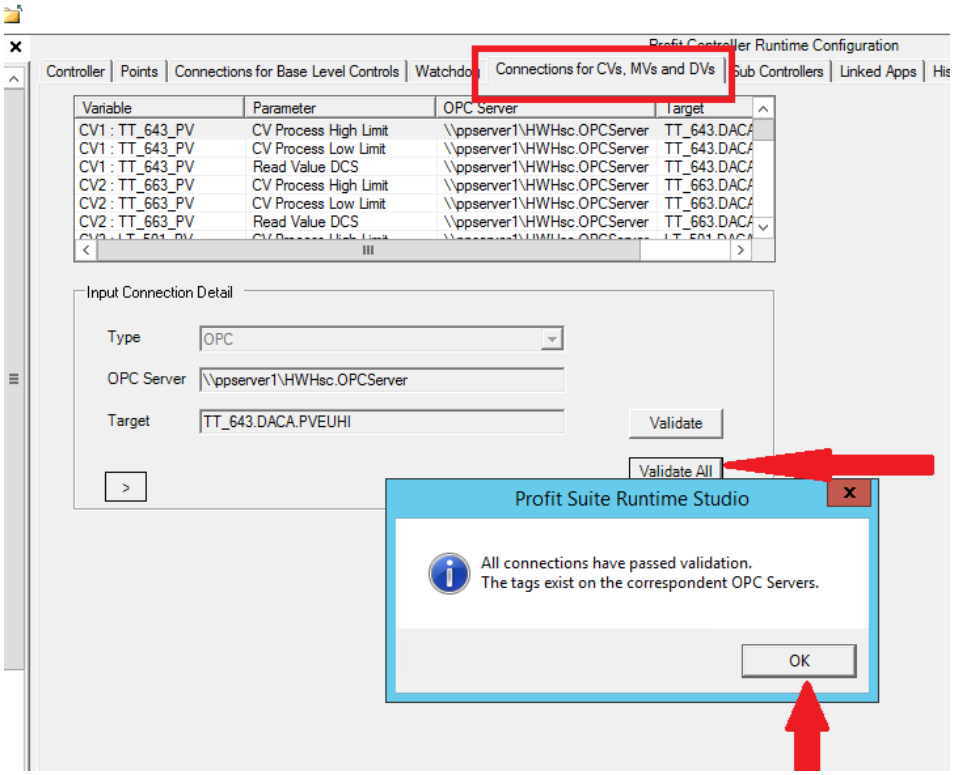


Figure 3.52 Validate all Points to test the OPC communication between Profit Controller and Experion Points is successful

Sub Controllers Tab

Profit Controller could be broken into sub-controllers, one for levels and another for temperatures for example. The advantage is a temperature sub-controller could shed to operator control if a critical temperature CV became unavailable, without shedding the level sub-controller, or vice versa. Each Profit Sub-Controller will still have the same **Execution Rate** and model files etc. A view in PSOS of a Profit Controller created with sub controllers in PSOS is shown in Figure 3.53.

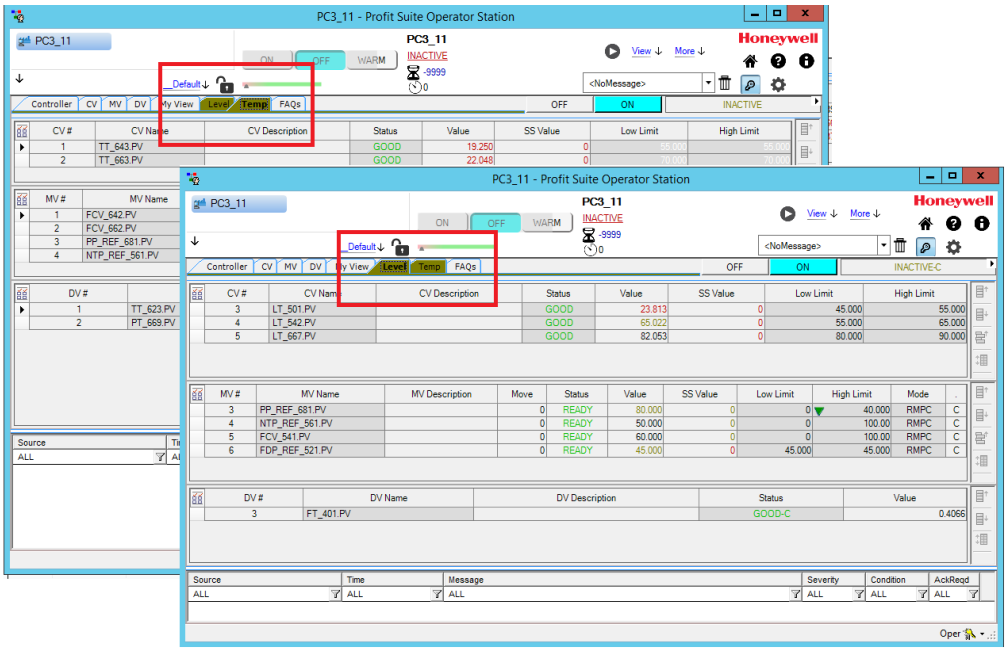


Figure 3.53 A view in PSOS showing sub-controllers for level and temperature

Each sub controller was given a name and assigned Variables on the **Sub Controller Tab** as shown in Figure 3.54:

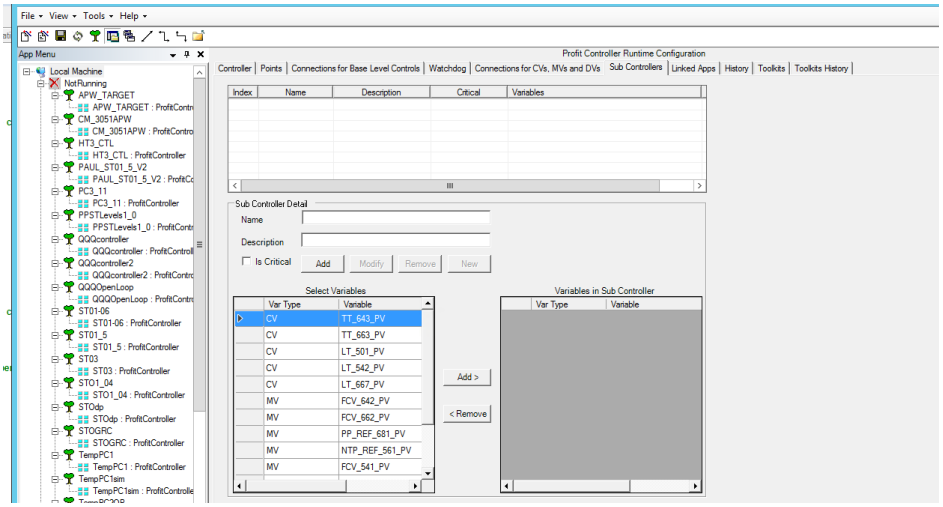


Figure 3.54 Assign variables to Sub Controllers then save the whole PSRS configuration

Configuration of the PSRS application is complete. Click the Save icon to save the configuration and record the name and file location for future reference as shown in Figure 3.55:

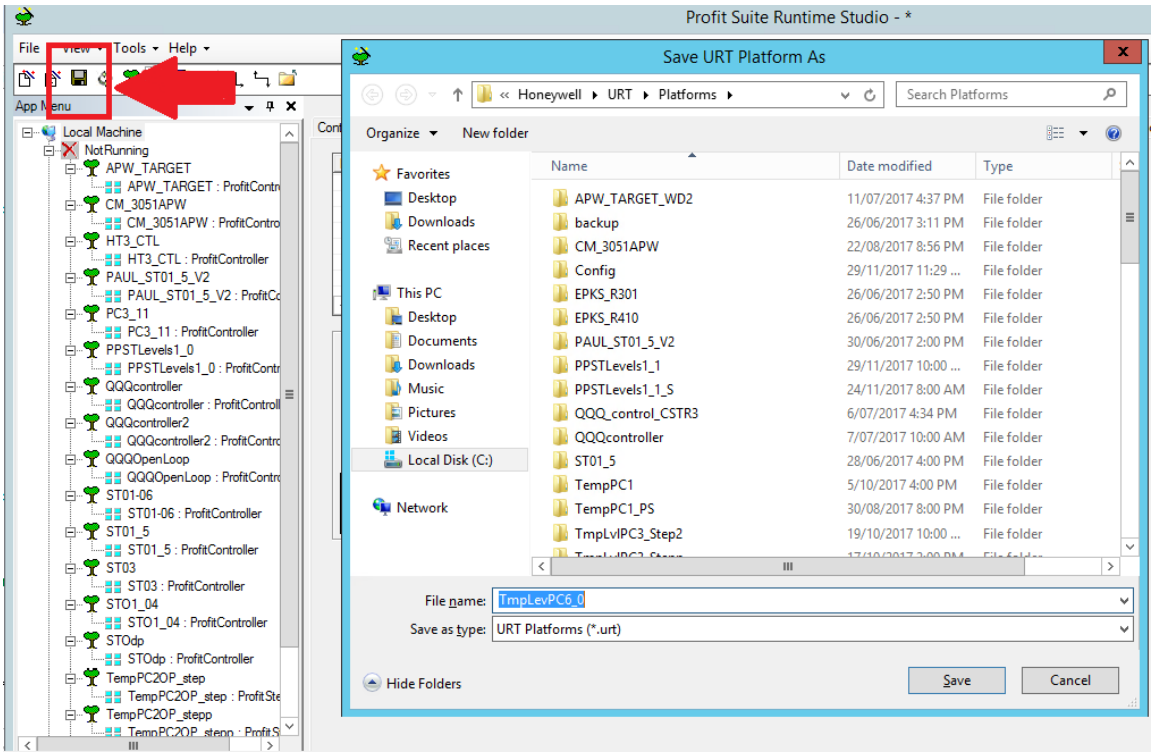


Figure 3.55 Save the PSRS application to complete the configuration and record the name and file location

3.4.4.4 Edit OPC connection to Experion PKS in URT Explorer

Profit Controllers would not communicate with Experion PKS until the OPC server was configured using URT Explorer. To enable this users should open URT Explorer, right click on the newly created platform and select **View** to explore the parameters of the Platform. This is shown in Figure 3.56:

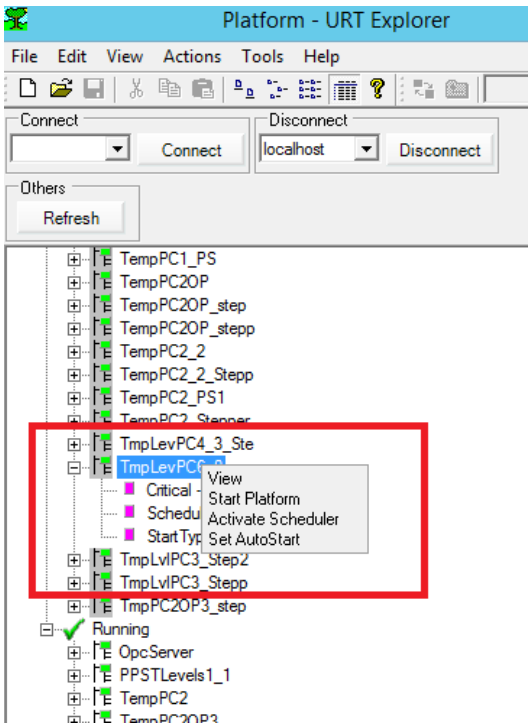


Figure 3.56 Right click on Platform and select View to explore Platform configuration

Browse to **OpcServerInfor-EpksOpServer** then click **ocServerID** to find the OPC server settings, as shown in Figure 3.57:

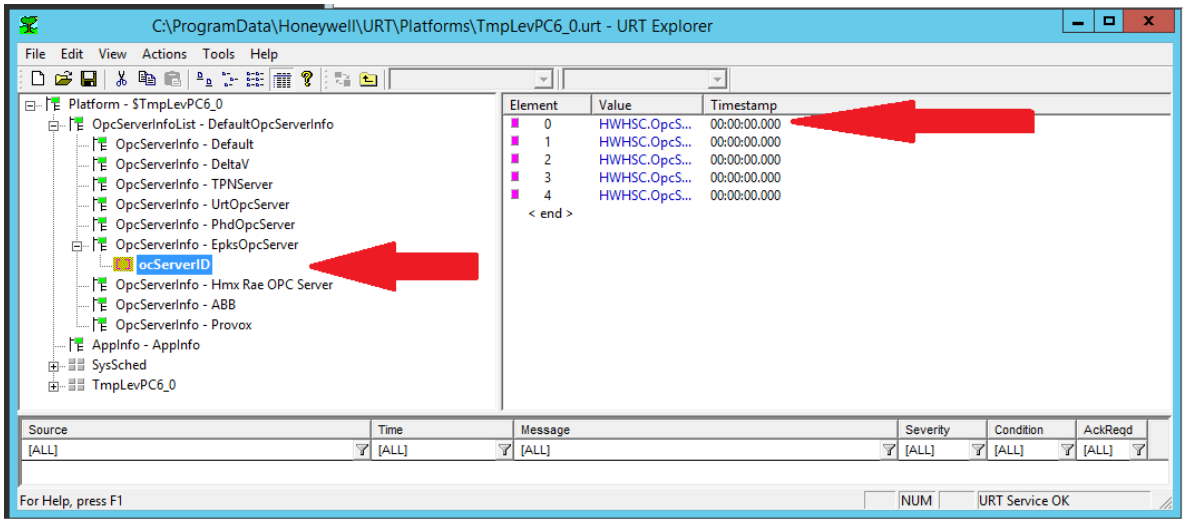


Figure 3.57 Location of OPC server settings inside URT Explorer

Double click on element 0 the top row, as shown in Figure 3.58:

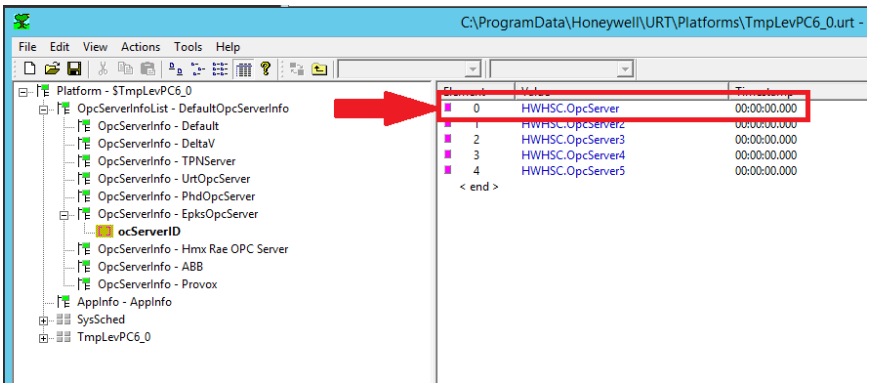


Figure 3.58 Edit the OPC server address in element 0 in URT Explorer

Change the Working Value to either **\\ppserver1\HWHSC.OpcServer** for the Pilot Plant or **\\Experion2\HWHSC.OpcServer** for the simulation Experion2 server, as shown in Figure 3.59:

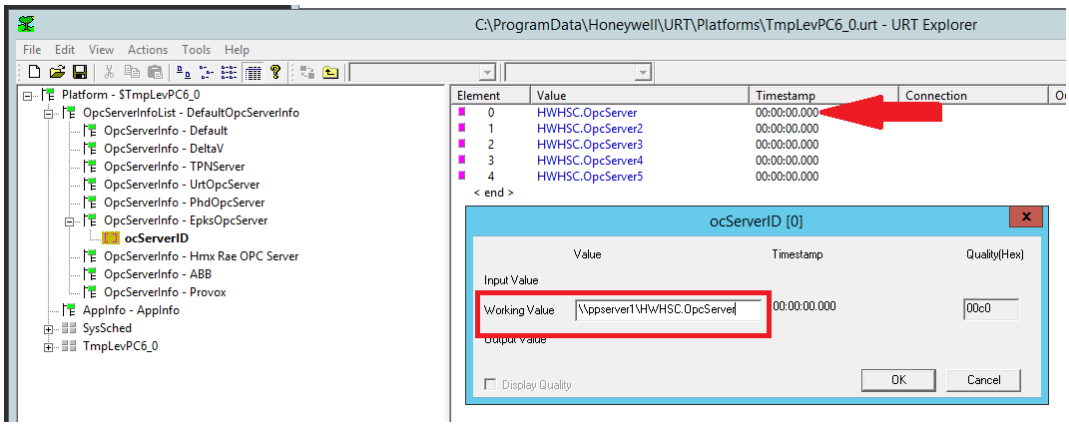


Figure 3.59 OPC Working Value changed to \\ppserver1\HWHSC.OpcServer

Check the address in element 0 is correct then save the platform, as shown in Figure 3.60:

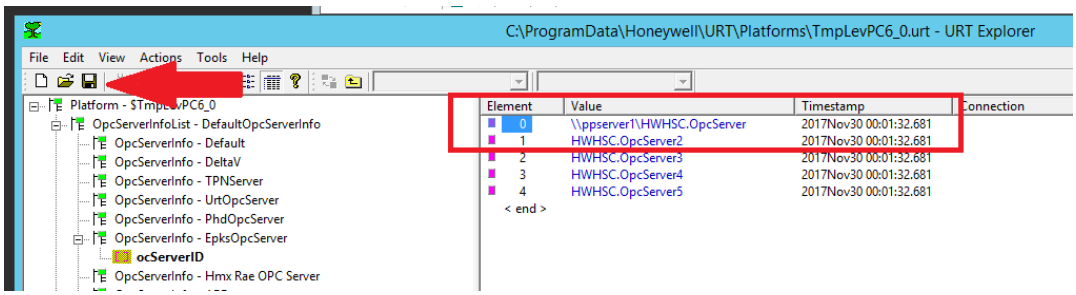


Figure 3.60 Save the OPC server settings in URT Explorer

Close (but not Terminate) the platform to exit the explorer view, as shown in Figure 3.61:

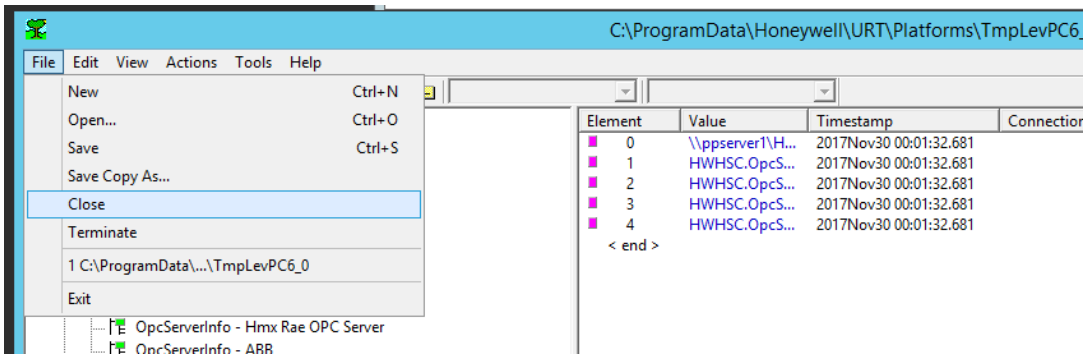


Figure 3.61 Close the Platform to exit the explorer view

The Platform will still be running as seen in the list of Nodes, as shown in Figure 3.62:

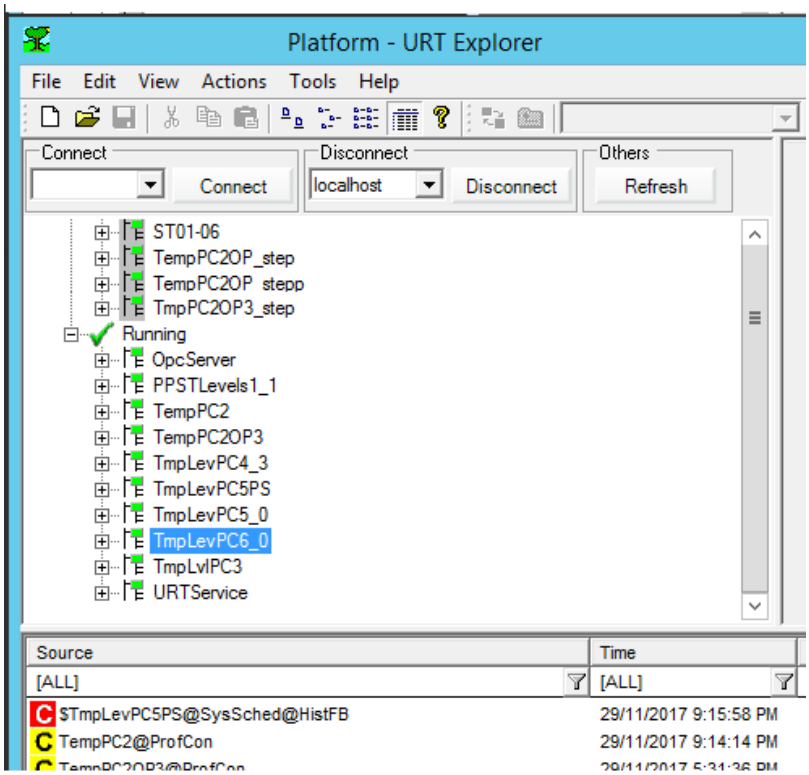


Figure 3.62 Newly created Profit Controller Platform listed as running in URT Explorer

Leave the Platform running to use the Profit Controller, or right click and select **Terminate Platform** to totally shut it down, as shown in Figure 3.63:

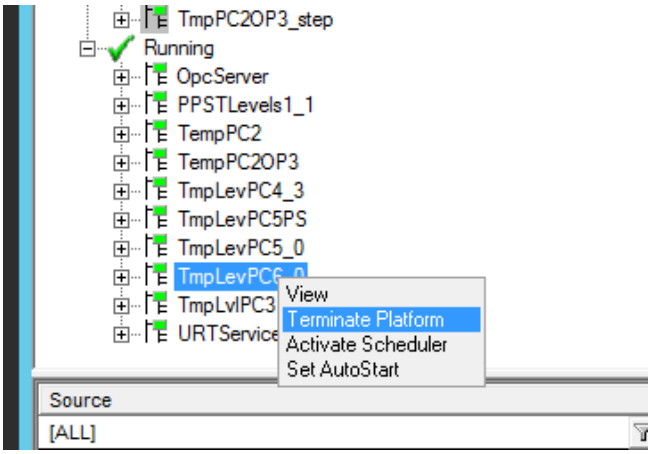


Figure 3.63 Terminate Platform to shut a Profit Controller down completely

3.4.4.5 Load the Profit Controller Watchdog CM into Experion

The Profit Controller Watchdog CM must be copied from the Profit Suite server to the Experion server then loaded into the Experion CEE.

In **profit-svr2**, locate the **Watchdog** files created in the previous section by PSRS, as shown in Figure 3.64:

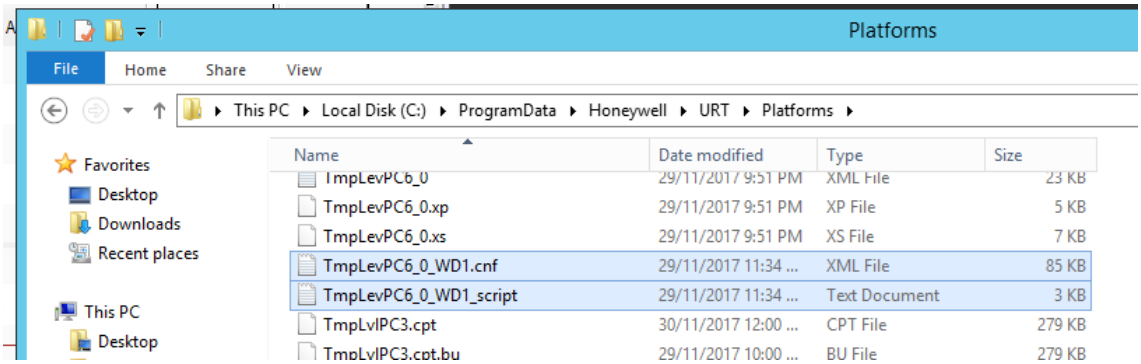


Figure 3.64 Locate the Watchdog XML and Text Document files created by PSRS

Log into the Experion sever (i.e. ppserver1) and create a new folder to copy the watchdog files to. Copy them into a folder on the desktop of ppserver1 for example, as shown in Figure 3.65:

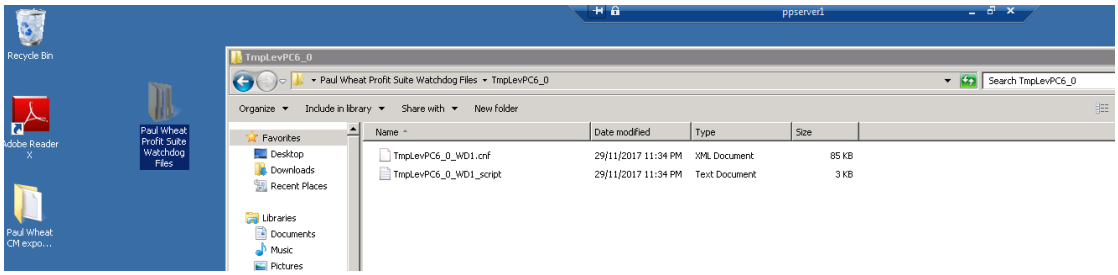


Figure 3.65 Watchdog XML files copied into a folder on the desktop of ppserver1 ready to be imported with Configuration Studio

Open Configuration Studio on the Experion server (i.e. ppserver1) and connect to the server Target, as shown in Figure 3.66:

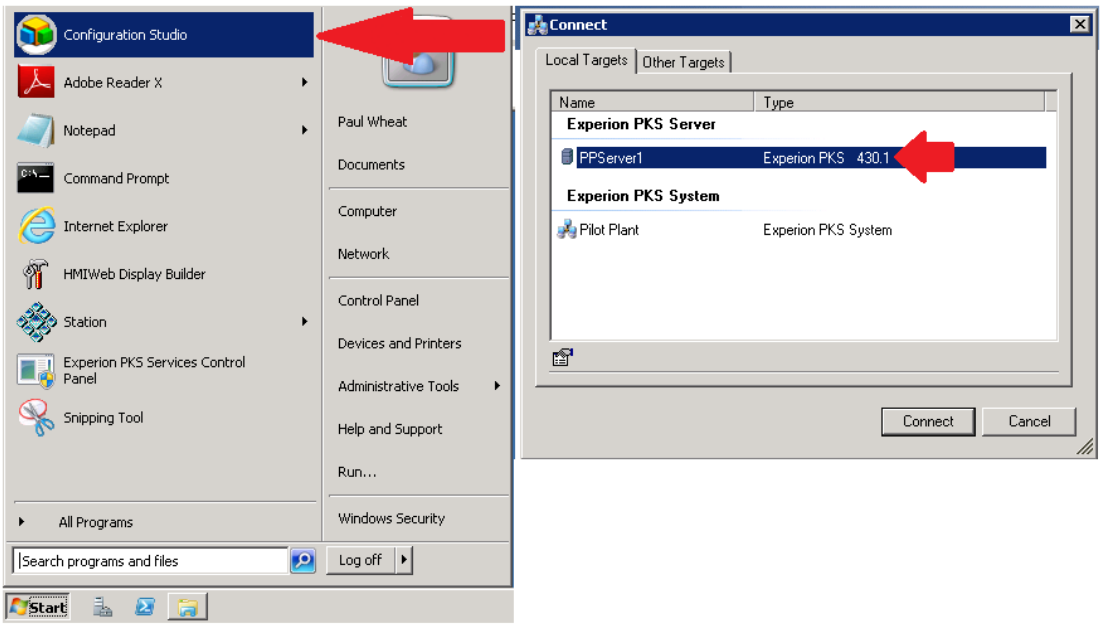


Figure 3.66 Configuration Studio in Experion server and connect to PPserver1

Click on **Control Strategy** then click c**onfigure process control strategies**, this opens **Control Builder** to view the Pilot Plant code, as shown in Figure 3.67:

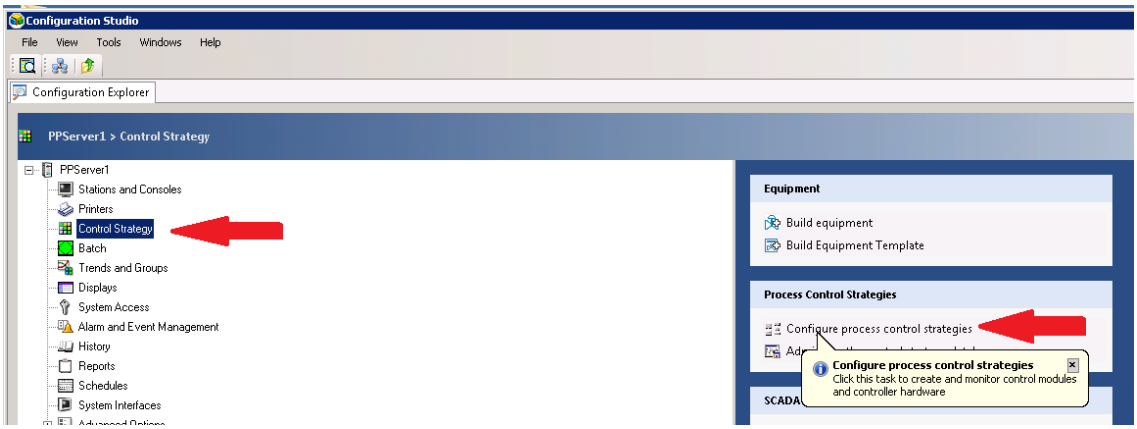


Figure 3.67 Click Configure process control strategies to open Control Builder

In **Control Builder** select **Project Tree** then expand **Unassigned**, as shown in Figure 3.68 part (a). Click **File** and then **Import** as shown in Figure 3.68 part (b):

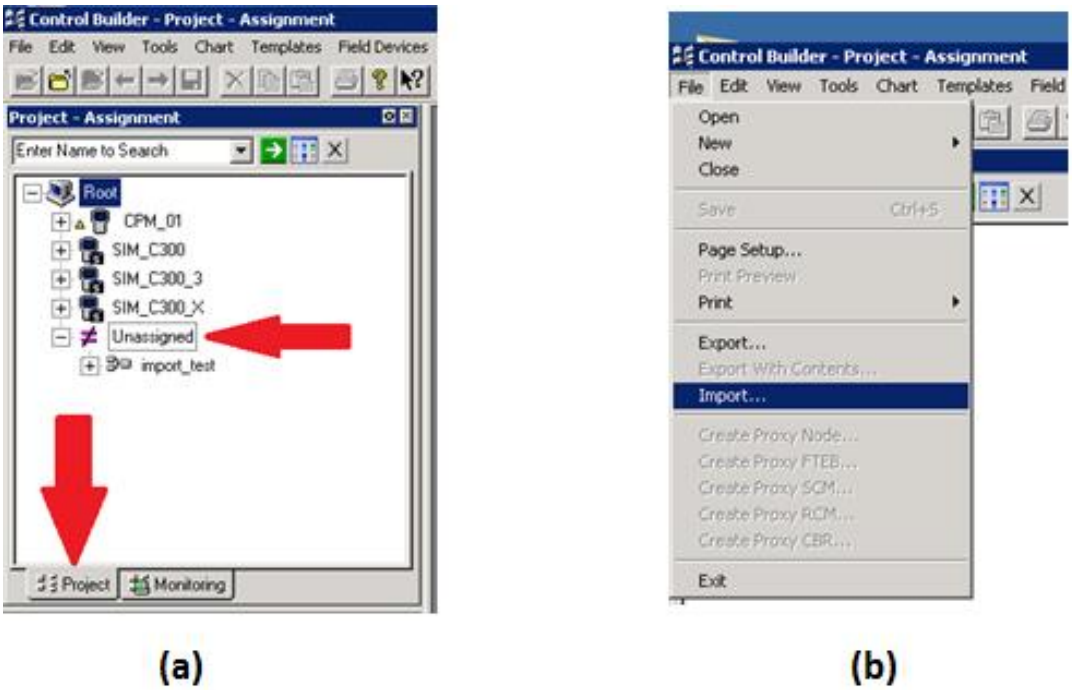


Figure 3.68 Use Import to add the Watchdog CM to the project

Browse to the folder that contains the Watchdog files copied from PSRS. The window message says 'no items match your search' even though it is the correct folder. Click OK to ignore this, as shown in Figure 3.69:

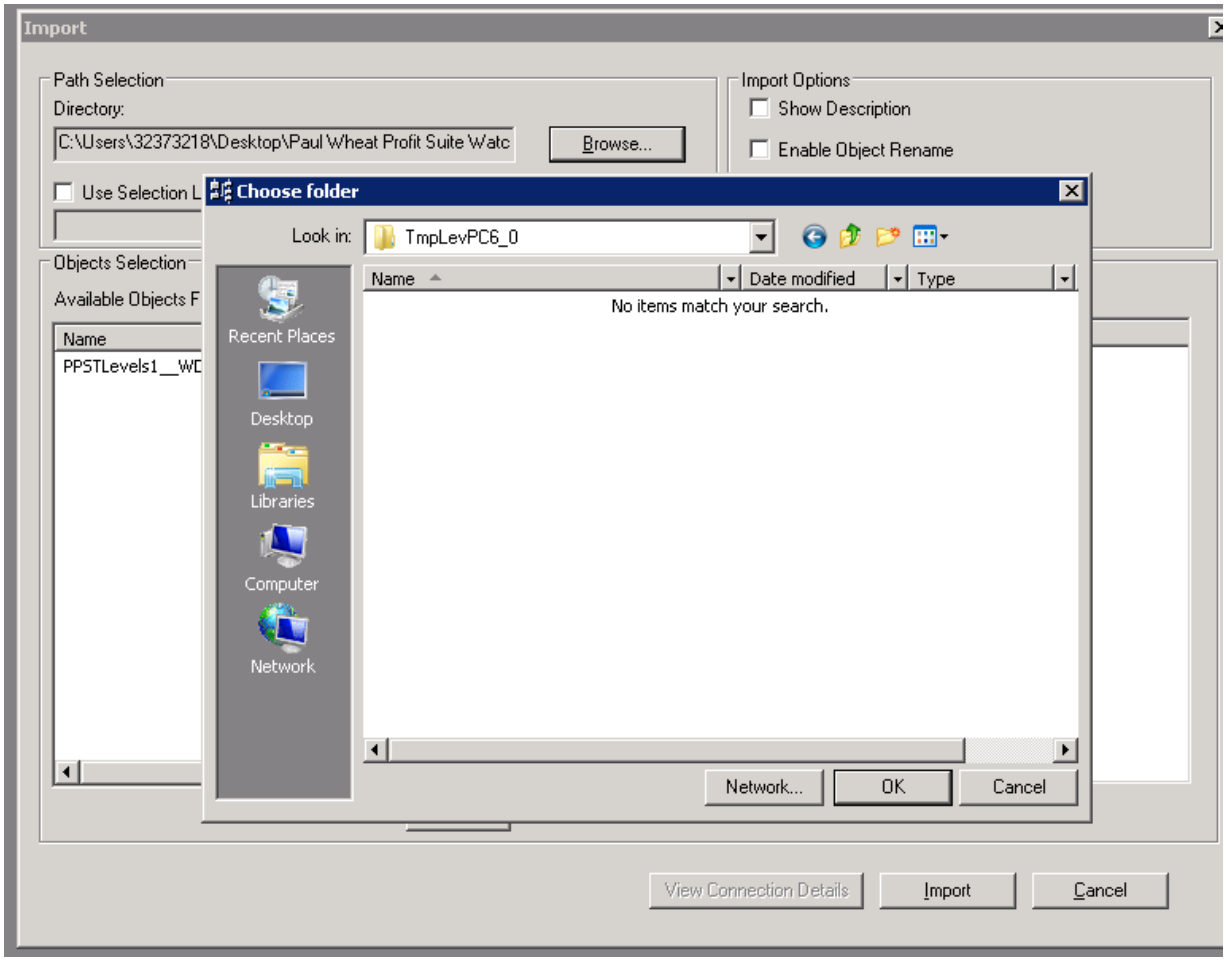


Figure 3.69 Choose the folder containing the Watchdog files for Import. Ignore 'No items match your search' message.

Select the Watchdog object to be imported then click **Import**, as shown in Figure 3.70:

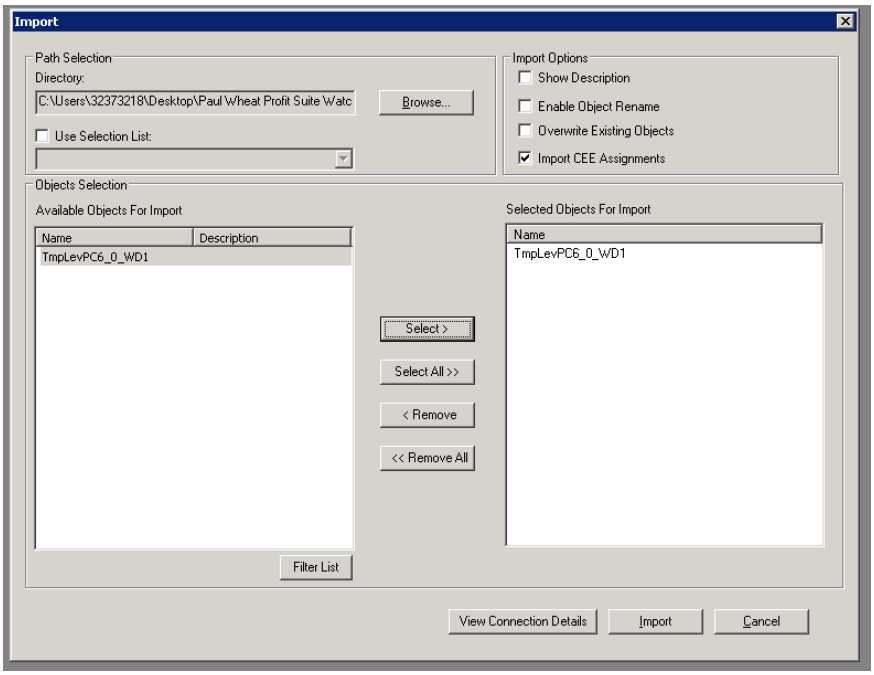


Figure 3.70 Import the Watchdog CM

Find the watchdog in **Unassigned** in the project tree, as shown in Figure 3.71:

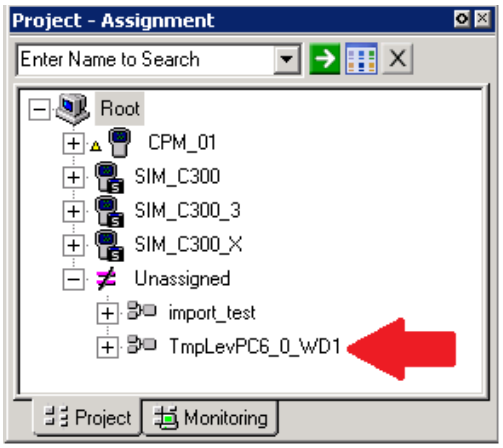


Figure 3.71 Newly imported Watchdog CM is not yet assigned to a CEE

Watchdogs are created with the default Parent Asset A1. Watchdogs cannot be loaded into the CEE without first being assigned to a valid Parent Asset. Right click on the **Watchdog CM** and select **Module Properties,** as shown in Figure 3.72:

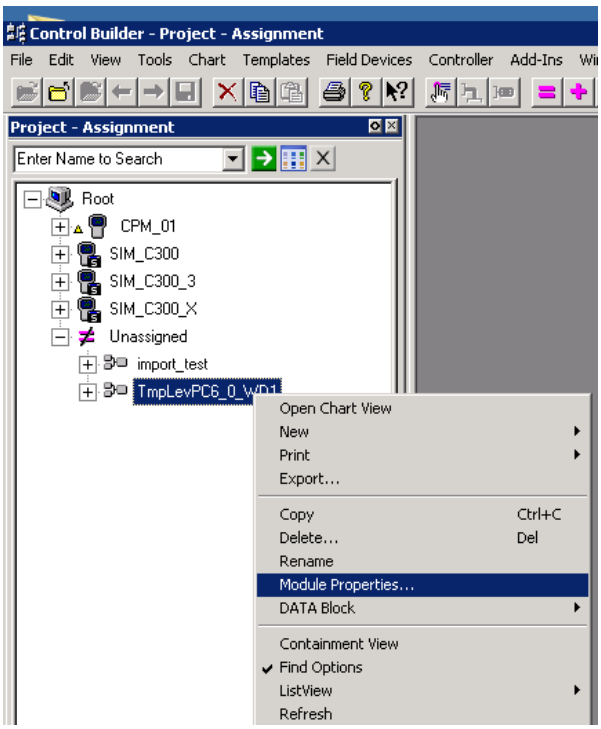


Figure 3.72 The Watchdog CM will not load until assigned to a parent asset

Browse from **Parent Asset (…)** icon and assign the Watchdog to the PILOT asset then click **OK** as shown in Figure 3.73:

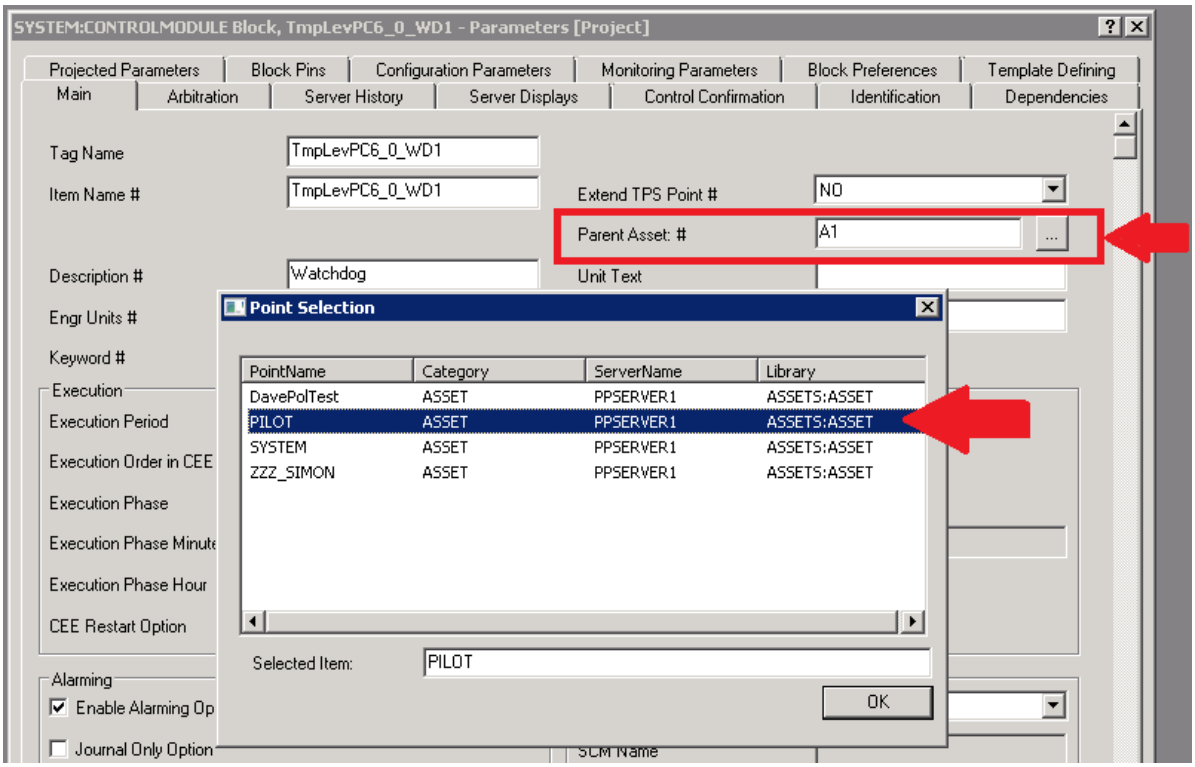


Figure 3.73 Assign Watchdog to PILOT asset to enable loading to CEE

Drag the **Watchdog CM** from **Unassigned** and add it to **CEE_01.** This controller contains the Control Modules that the Watchdog will be monitoring for the Profit Controller. Locate the **Watchdog CM** in the Project tree assigned to the correct CEE. Two adjacent chevrons indicate changes not yet loaded to the CEE, as shown in Figure 3.74:

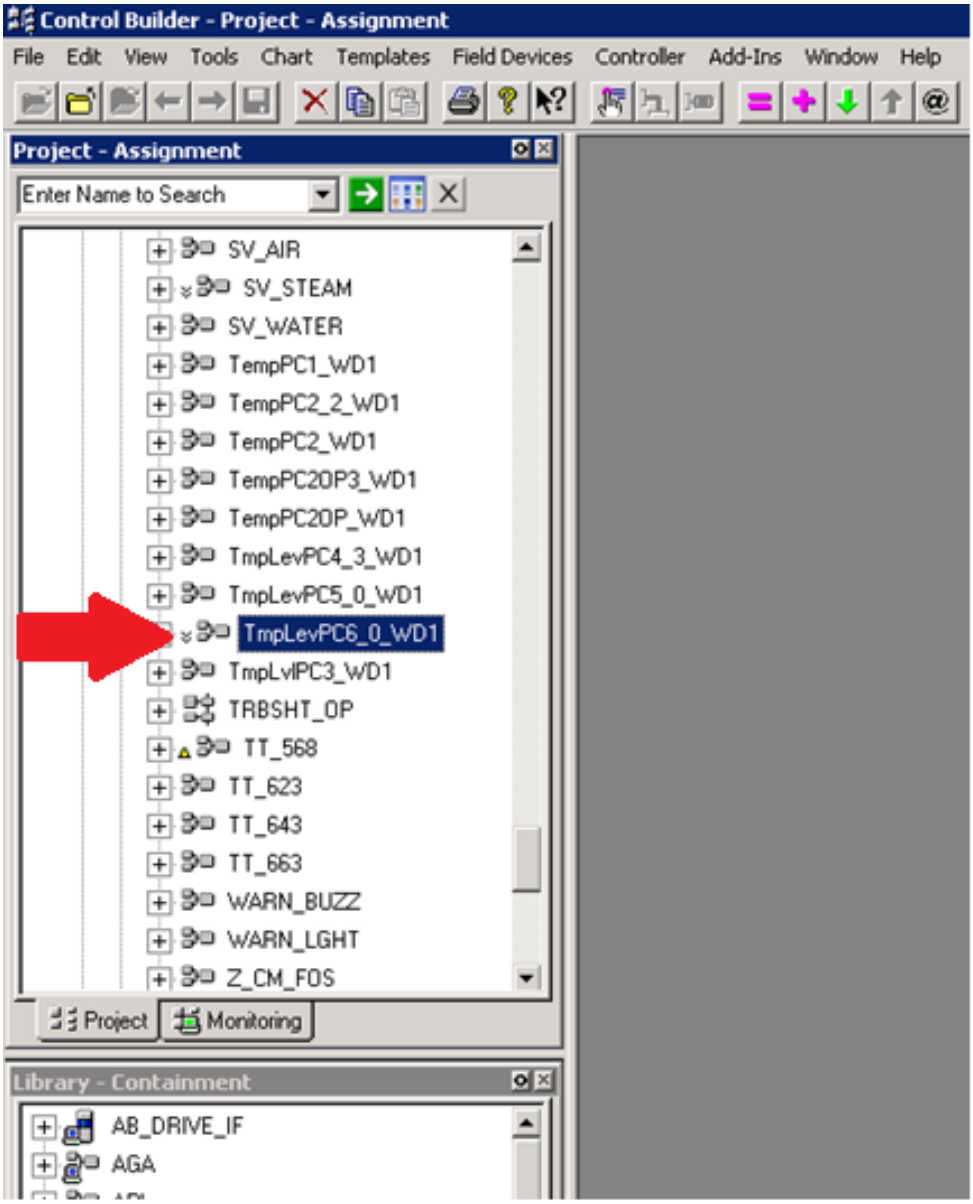


Figure 3.74 Watchdog assigned to CEE with correct Parent Asset ready to load

Select the **Watchdog CM** and **Load** it to the live controller, as shown in Figure 3.75:

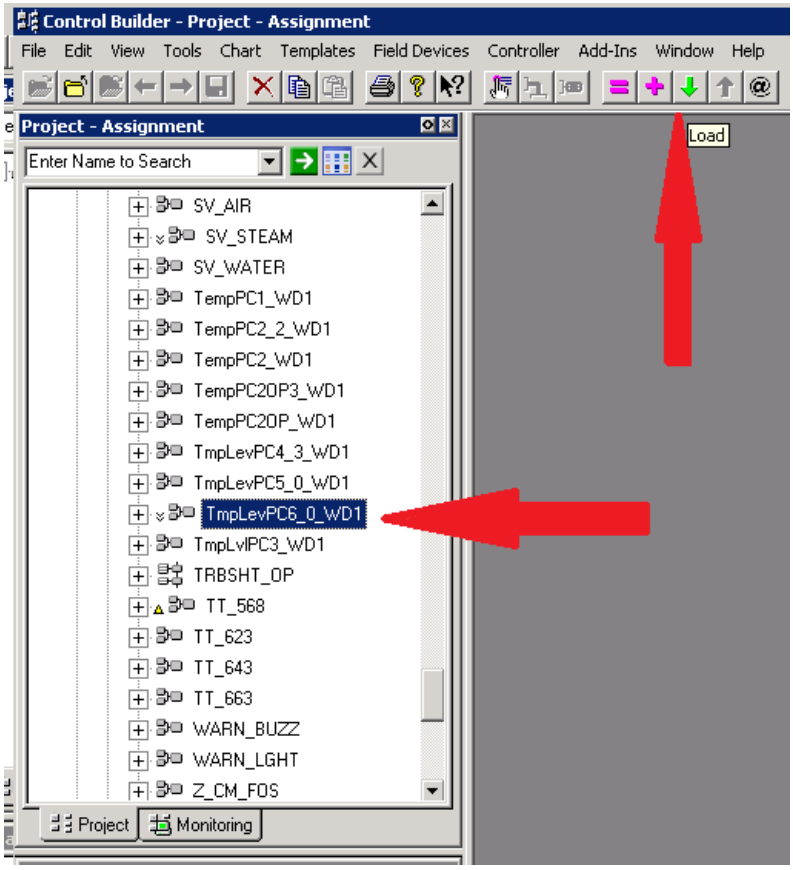


Figure 3.75 Select then Load the Watchdog

Click Continue to the following warning if there are no other users, as shown in Figure 3.76:

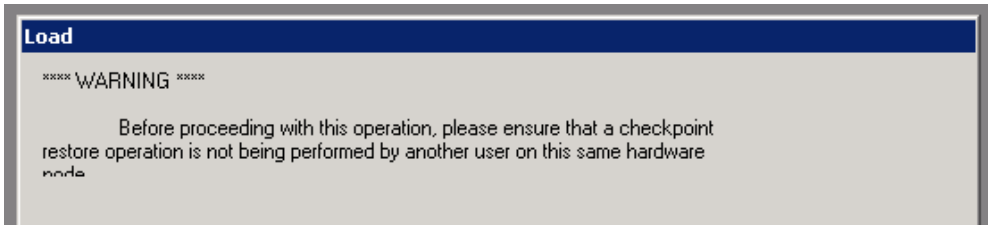


Figure 3.76 Proceed with the Watchdog Load if no other users are using checkpoint restore

Tick the checkbox in bottom left corner to activate the Watchdog after loading, as shown in Figure 3.77:

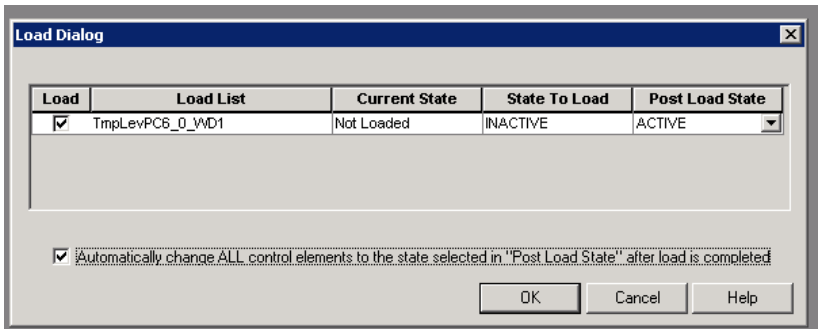


Figure 3.77 Automatically change Watchdog state to Active after Load

The Watchdog icon in the Monitoring Tree turns green to indicate the CM is running, as shown in Figure 3.78:

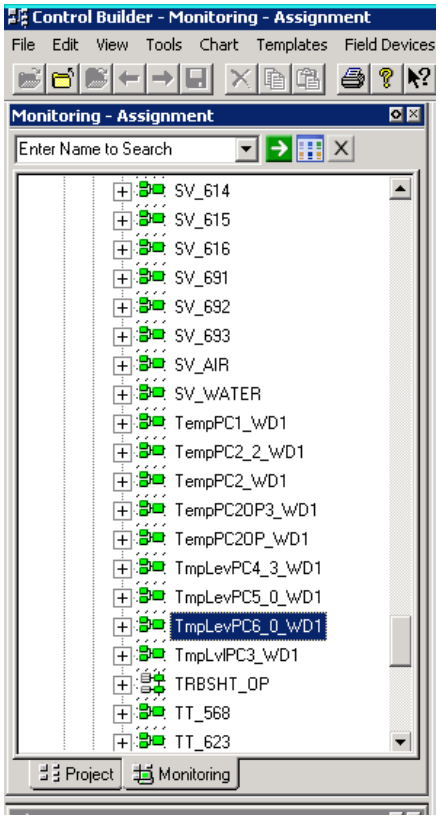


Figure 3.78 Watchdog icon is green in Monitoring Tree tab indicating it is running

3.4.4.6 View and configure Profit Controller Themes in PSOS

The PSOS enables operator interaction with a Profit Controller platform. PSOS is much more than a simple view of the platform, it contains many parameters which can dramatically alter the behaviour of a Profit Controller. This section details the steps used to configure themes in PSOS for running Profit Controller platforms.

Open PSOS from the desktop and select the Profit Controller. The Status column shows the current state of all platforms as explained in Table 3.11.

Table 3.11 Profit Controller status meaning in PSOS

Status	Platform State	To Open in PSOS
NotRunning	The Profit Controller application has been Terminated	Start Application
INACTIVE	The platform is running but the Profit Controller is dormant.	View Application
CONTROLLER_OFF	The platform is ACTIVE and running. The Profit Controller is waiting to be put into WARM or ON modes.	View Application
CONTROL_OK	Platform is running and Profit Control is actively controlling the plant	View Application
OPTIMIZING	Platform is running and Profit Control is actively optimizing the plant	View Application

Running Platforms can be terminated in URT Explorer. Start a controller with **NotRunning** status by left clicking on its name and following the prompts. To view a controller, click on desired Profit Controller then select **View Application**, as shown in Figure 3.79:

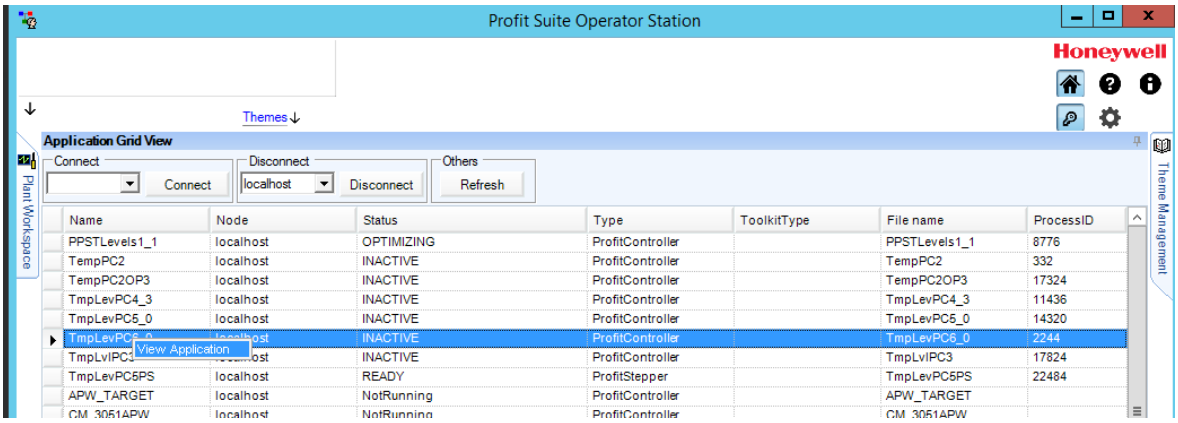


Figure 3.79 Opening a Profit Controller with PSOS

Click on **Oper** in bottom right corner and enter the password "**mngr**" to select manager user role, as shown in Figure 3.80. Manager privileges permit opening trends, saving new Themes and configuring PSOS.

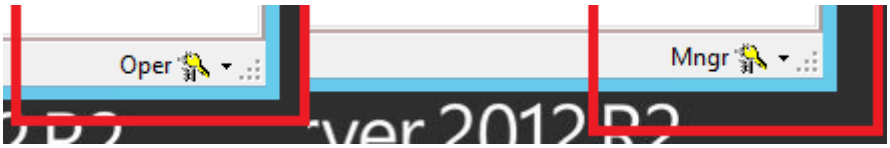


Figure 3.80 Change user roles in PSOS for Manager privileges

Open the CVs tab, click and hold a CV then drag it onto the My View tab, shown in Figure 3.81. This CV can then be viewed in the MY View tab. Repeat this for all CVs, DVs and MVs. The MY View tab becomes the customised window for the operator of this Profit Controller. It can contain as many/few variables as desired.

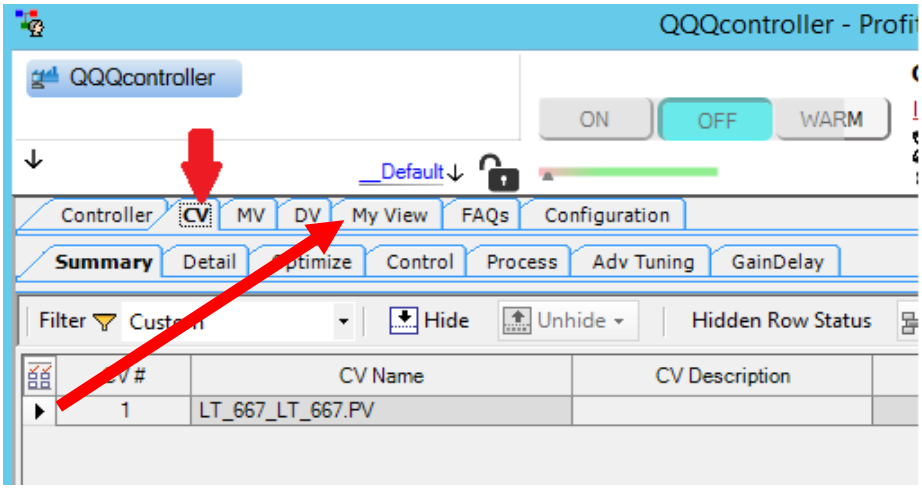


Figure 3.81 Drag CVs DVs and MVs into My View tab

My View can be customised to show Set Points, Profit Stepper moves, Soft Limits, Performance Ratios and so forth by clicking the icon in the top left of each section and selecting from the list as in Figure 3.82:

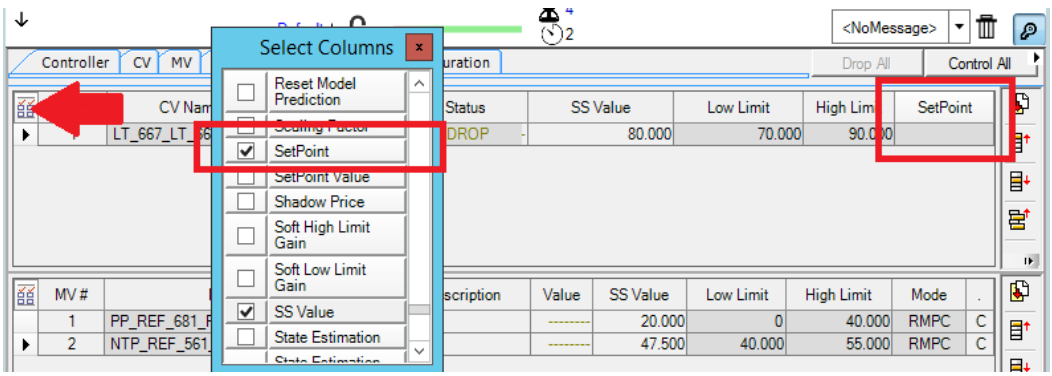


Figure 3.82 Add parameters such as SetPoint or Soft Limits to customise My View

Add a trend display by clicking **View** then **Trends,** as shown in Figure 3.83:

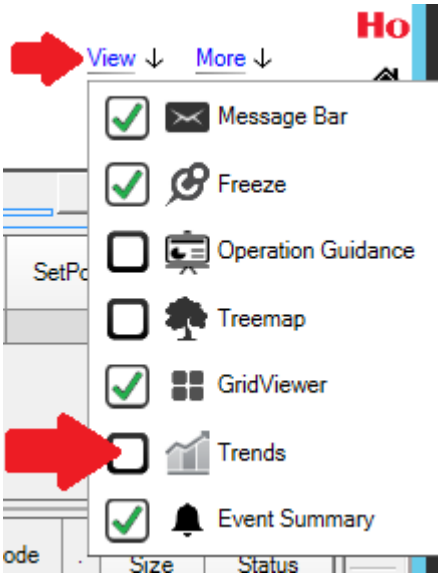


Figure 3.83 Add Trend displays to PSOS

Figure 3.84 shows the Trend window's display options. Future predictions of process values can be shown on the trend with the **Show Future Values** checkbox. Add variables by clicking on the downwards pointing arrow icon in the top right of the window then selecting **Options**.

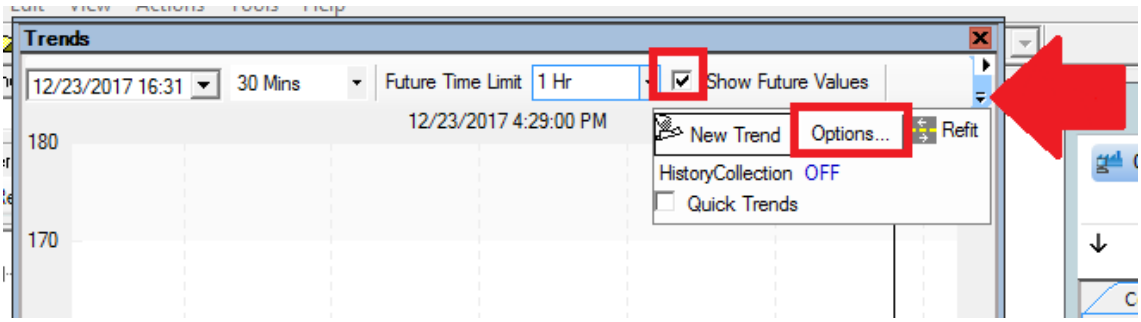


Figure 3.84 Show future model predictions and add variables to a Trend in PSOS

There are five trend Stacks in which to display variables. To place a variable on the graph, first expand the variable in the Trends tree, then right click on **Read Value** to add it to a Stack. Change the display High and Low limits for each variable. Select Show Future Trend for CVs and MVs. Figure 3.85 shows this sequence.

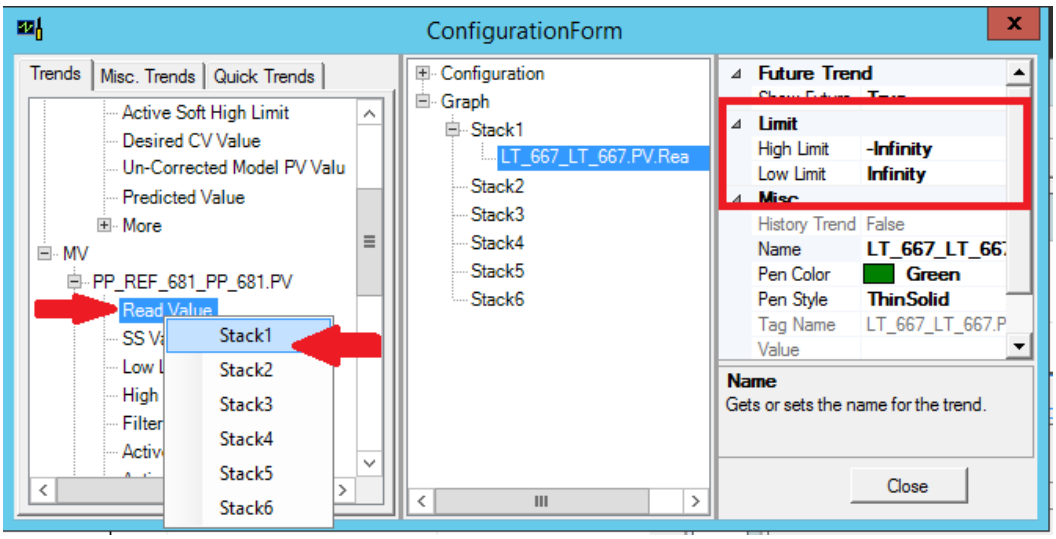


Figure 3.85 Add each variable Read Value to a trend Stack and edit High and Low limits

A trend display with past and future values can be configured to look similar to Figure 3.86 which shows three stacks in the Trend display of TmpLevPC5_0 used in this project.

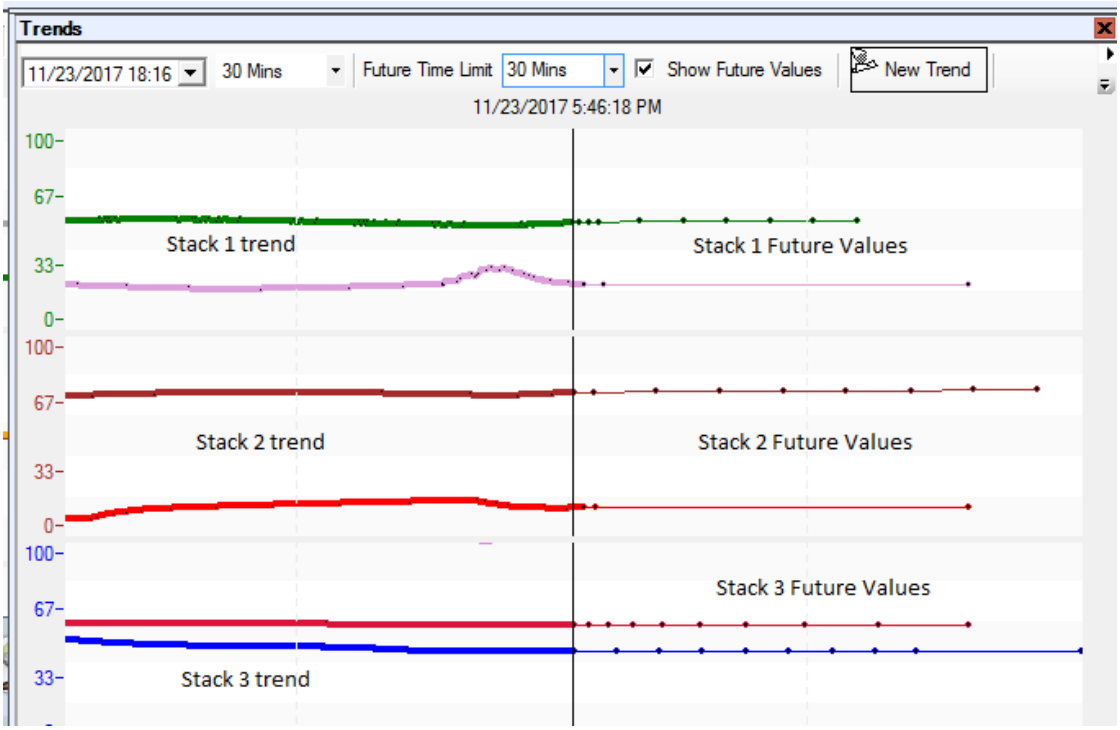


Figure 3.86 Configured trend Stacks showing past and future predicted process values

When closing the PSOS window opt to save the current configuration as a Theme with a unique name. This will store all the customization. Next time the Profit Controller is viewed, load this Theme to view the Profit Controller display. Significant changes to the theme can be saved each time the PSOS window is closed. Themes have user privilege constraints. If it is saved for Managers, the drop down menu will not show it to an Operator. Figure 3.87 shows where themes are accessed.

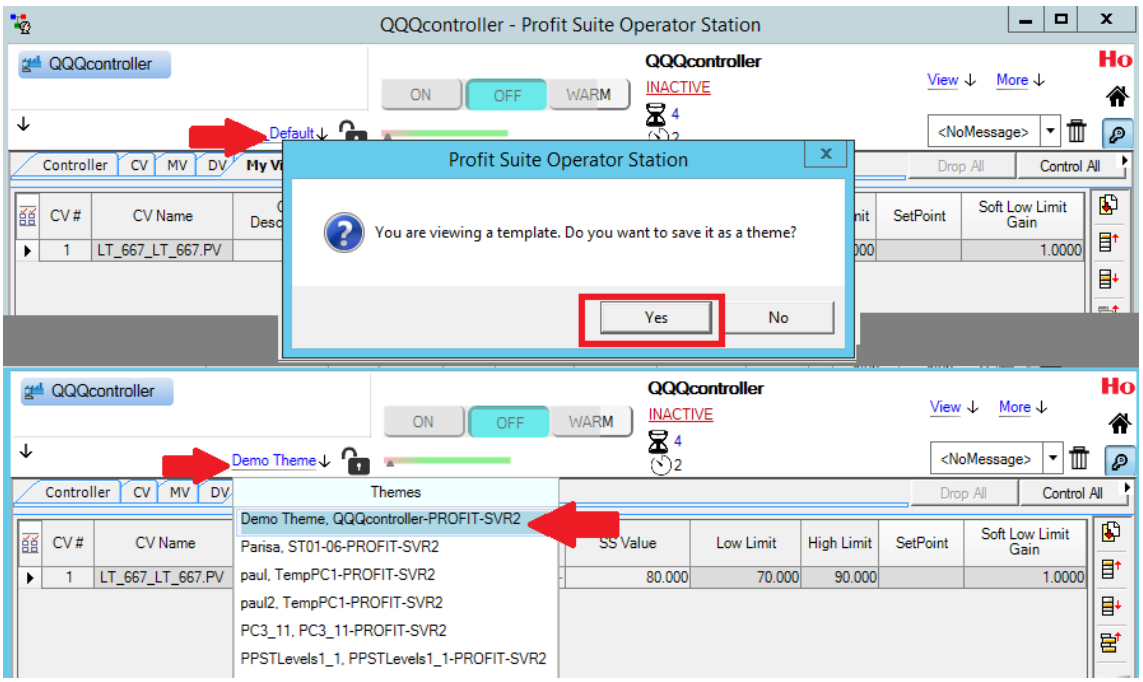


Figure 3.87 Load and save custom Themes in PSOS using Manager privileges

To start the Profit Controller, click **INACTIVE** then select **ACTIVE**. The controller will switch to **Controller_Off** mode. If **WARM** is selected, the controller will read process values from the plant to make predictions and calculate control moves, but will not execute them. The Profit Controller is turned on once the operator is confident the moves the Profit Controller is making in **WARM** mode are satisfactory. To do this, first use Experion to change the modes of all MV PID Blocks to those required by each BLC for Profit Control (e.g. Program Auto), then use PSOS to select **ON** to change Profit Control mode to **Control_ok**. The Profit Controller will control the live plant unless a critical CV or MV becomes unavailable, in which case it sheds control back to the Experion PID Operator modes configured with BLC. Closing PSOS will not stop Profit Control or terminate the platform. To stop the Profit Controller completely, first switch it to **OFF** mode, then terminate the platform in URT Explorer.

3.5 Profit Stepper

The Profit Stepper was used to model the process while the Profit Controllers were actively controlling the plant. The Profit Stepper made moves to MVs then collected the CVs' response data. This data was periodically modelled using the Load and Go feature in PDS, which automatically executes FIR and parametric fitting to produce the final models. The user can choose whether to load the models into the Profit Controller or not. The setup of Profit Steppers was critical to achieving any results at all. If the SNR was too small and response data was not selectively edited for modelling, the Profit Stepper could run all day without acquiring any useful models. This section outlines the steps used to create Profit Steppers and the settings required to obtain useful models.

New Profit Steppers were created in the PSES project by clicking **Add Application**, as shown in Figure 3.88:

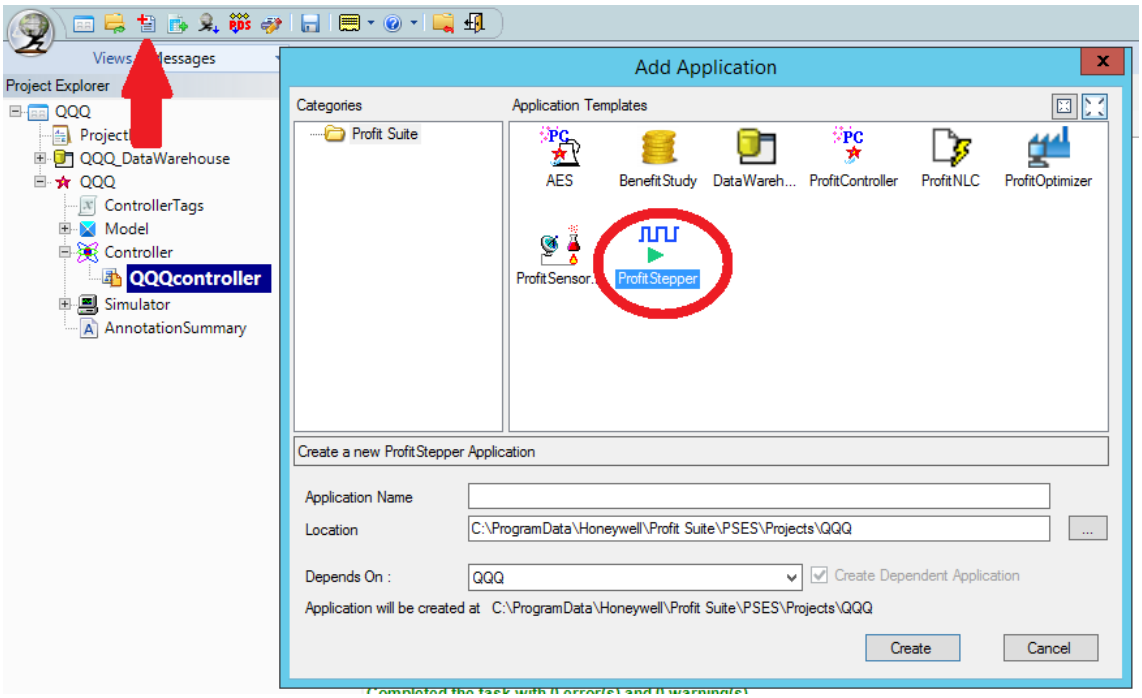


Figure 3.88 Add Profit Steppers to existing PSES Projects

A Profit Stepper can model on behalf of a Profit Controller. Therefore, the Profit Stepper could use the same OPC and Experion Point connections as the Profit Controller for which it was modelling. This was selected by choosing Controller Platform then browsing through available controllers in Selecting the Profit Controller Application, as shown in Figure 3.89:

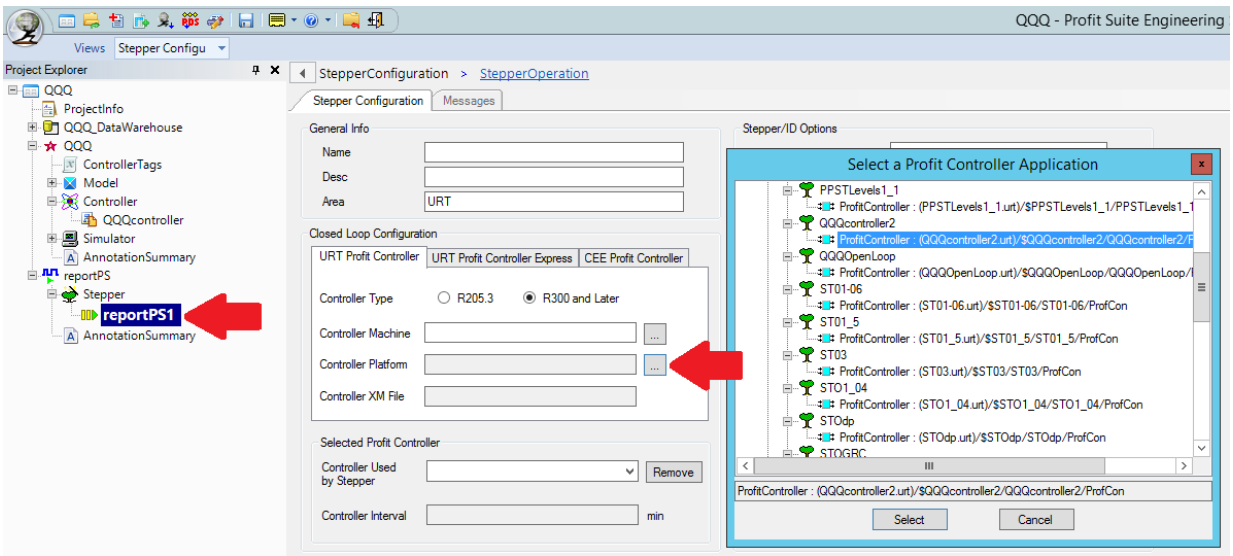


Figure 3.89 Connect Profit Stepper to an existing Profit Controller for step testing

All Profit Controller parameters were automatically imported, such as Controller Interval and Execution Interval. The MVs, CVs and DVs of the Profit Controller were imported into the Profit Stepper also. The ID Update Frequency is the interval between model identification attempts. A default value of 10 minutes was entered during the build; though it was possible to alter the interval after the Profit Stepper was created. The profit Stepper was automatically given a unique name (not exceeding 15 characters) and description then built by clicking Build Platform as shown in Figure 3.90:

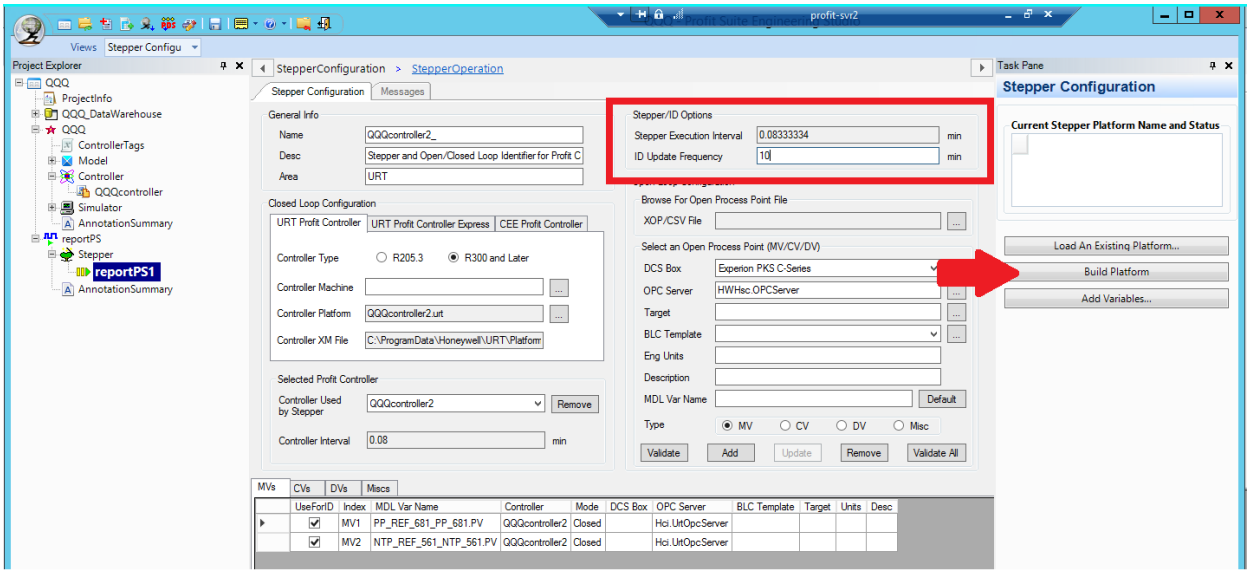


Figure 3.90 Parameters automatically imported into Profit Stepper from Profit Controller

When the Profit Stepper Execution Interval was different to the sample interval of the manual step testing data in the Data Warehouse, the Profit Stepper build was interrupted by an error message about conflicting intervals. This was caused by collecting data from Experion at 1 second intervals for initial modelling, then later choosing 5 second Execution Intervals for Profit Controller/Steppers. The conflict was resolved by returning to the Data Warehouse and renaming the original variables i.e.:

FCV_541_original.PV

Profit Steppers build was then initiated successfully at any desired Execution Interval, as shown in Figure 3.91:

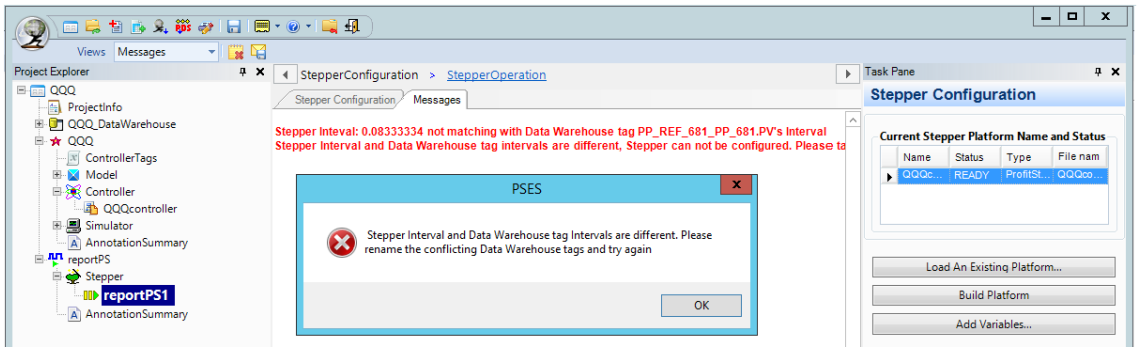


Figure 3.91 Rename original variables when Data Warehouse sample interval and Profit Stepper Execution interval are different

Correct set up of the parameters in the Profit Stepper was the key to getting any models from it. The amplitude of the Steps had to be as large as possible for the Model Identification to be able to distinguish the CVs' response from noise. Though the response could be clearly seen by the operator, this did not mean that the Profit Stepper could recognize the correct response. The Step Magnitude had to be set high enough for the Signal to Noise Ratio (SNR) indicator to turn green which indicated that the Model Identification software could differentiate between noise and the step response.

- SNR >= 3 are good and indicated in green
- SNR >= 2 are marginal and indicated in yellow
- SNR < 2 are bad and indicated in red

Models that were obtained with low SNRs were not reliable. Rank 1 models were even produced with their gain polarity reversed from Profit Stepper finding relationships in random noise.

The Step Magnitude and settling time of each CV was entered. These parameters were influenced by the MV being an OP or a SP of a closed loop PI controller via BLC. The Profit Stepper could be stepping, then finding models between either a PID SP and CV or an OP and CV. The Step magnitude for an MV could not be larger than the limits set for that variable in PSOS. These limits range had to be widened in PSOS to allow steps large enough to enable high SNRs in Profit Stepper. Profit Stepping was initiated by starting the data **Collector** and then **Stepping**, as shown in Figure 3.92**:**

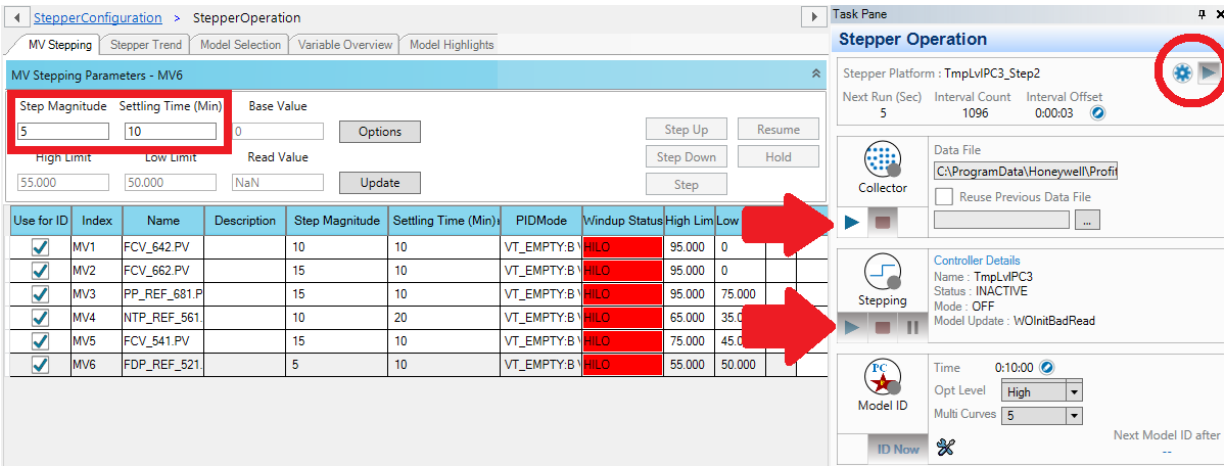


Figure 3.92 Enter Step Magnitude, the direction of the first step then press Update. Start the Collector then begin Profit Stepping.

The Model Selection page was used to set Integrator options, choose which sub-models were to be modelled and which were to be nulled. Models could also be locked only once reliable Rank 1 models were found for a particular sub-model. This is shown in Figure 3.93.

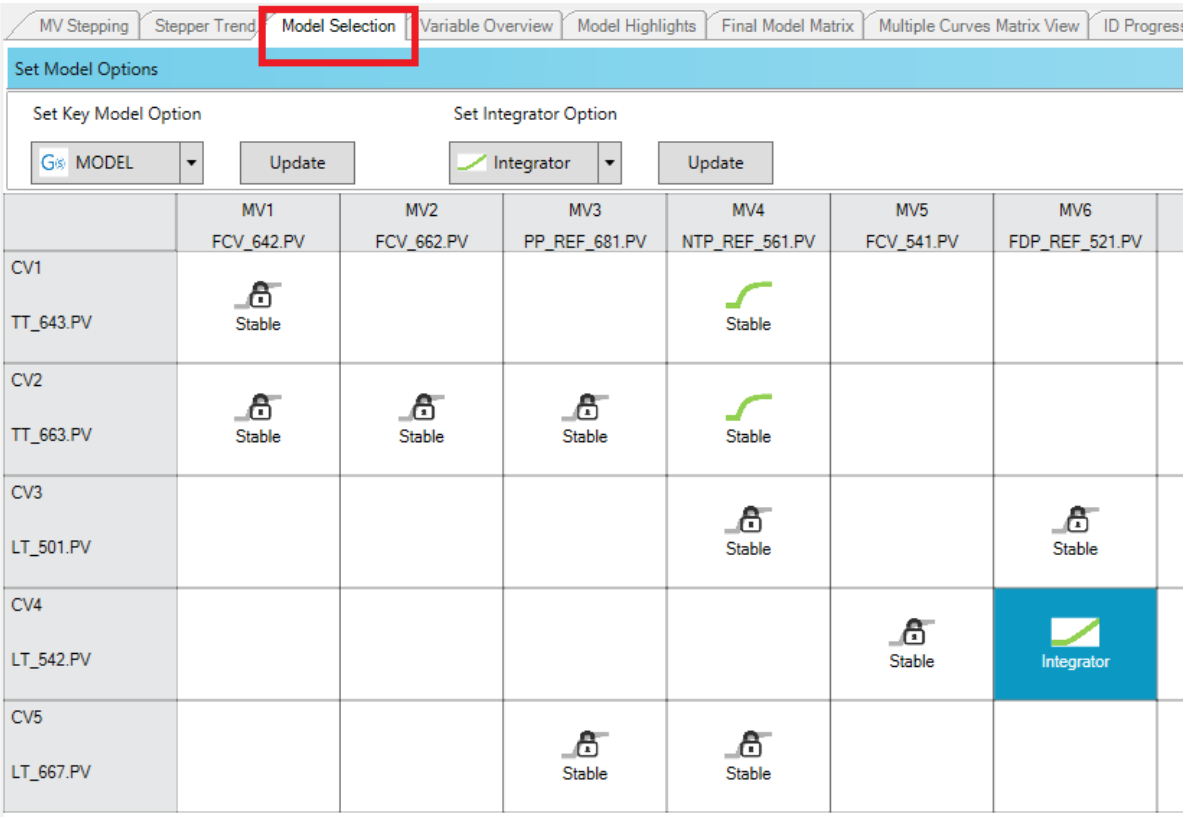


Figure 3.93 Set integrator options and null sub-processes or lock in Rank 1 models

Each time the Model ID is executed, the Model Highlights tab shows the quality of each model found:

- Rank 1 and 2 models are indicated in green;
- Rank 3 models are indicated in yellow;
- Rank 4 and 5 models are shown in red;

The key to obtaining models quickly was to adjust the MV step and settling time settings in Profit Stepper and PSOS until both the SNR and Sub-Model Rank indicators both repeatedly turned green after each Model ID run. Steps had to be as large as possible without exceeding CV constraints to find models quickly. If not, the Profit Stepper could run unsuccessfully for hours. With success, Rank 1 models were locked in as acquired and profit stepping ceased for these sub models to enable quicker model acquisition for the remaining sub-models as shown in Figure 3.94.

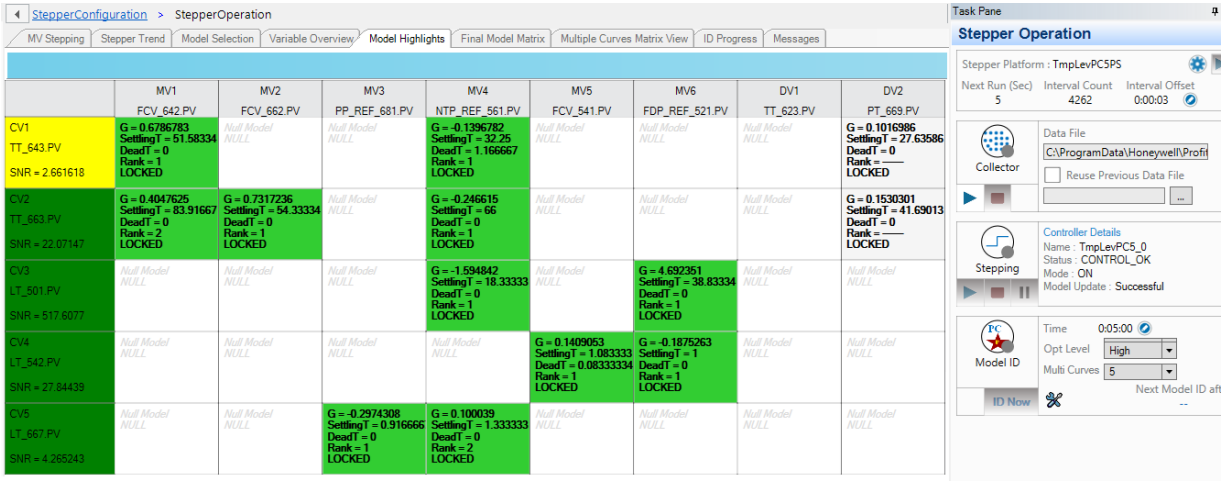


Figure 3.94 High SNRs from large steps produced faster more accurate results from the Profit Stepper. SNR indicators shown green and yellow in far left column.

When first stepping OP points, the Profit Controller would not execute the steps. This was caused by an MV parameter in PSOS in the Detail tab preventing large moves. The default MV Maximum Move was increased to allow the Step Magnitude move selected in Profit Stepper as shown in Figure 3.95:

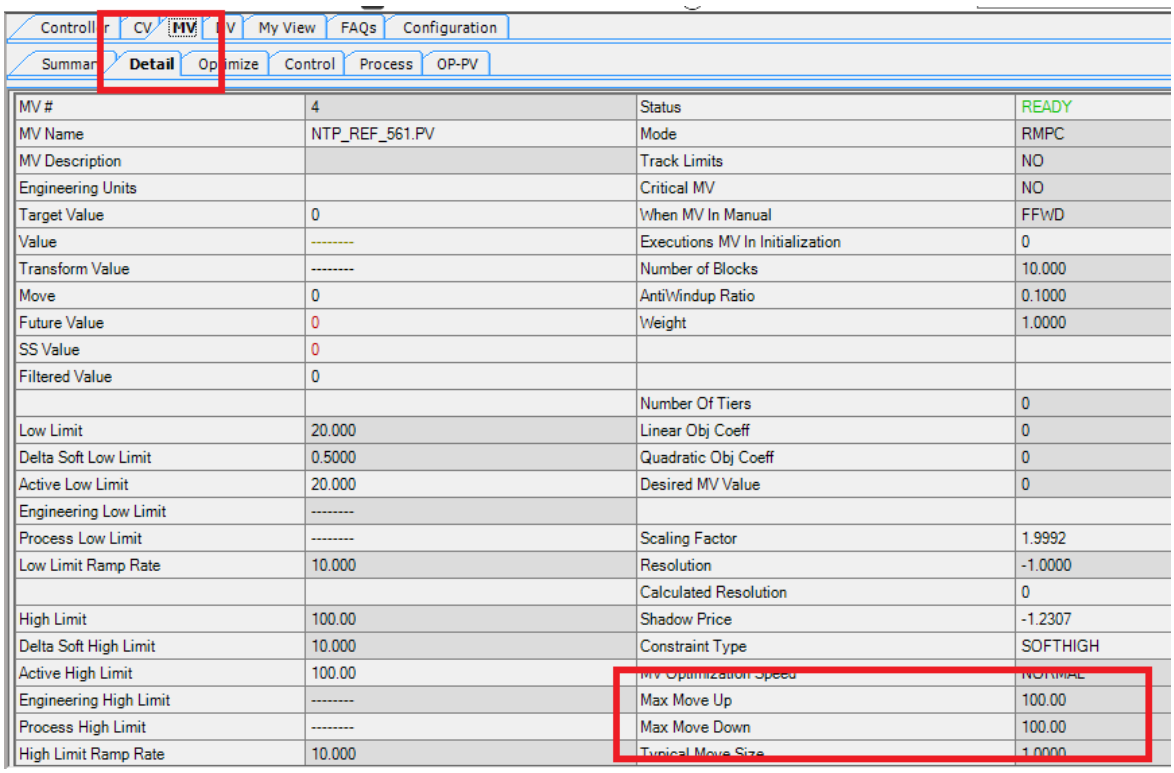


Figure 3.95 MV maximum move parameters must be large enough to permit Profit Stepper moves

Once Step Size and Step Status parameters were added to MyView inside PSOS, the moves made by Profit Stepper could be confirmed in PSOS each time they executed as shown in Figure 3.96.

MV #	MV Name	Move	Status	Value	SS Value	Low Limit	High Limit	Mode	Step Size	Step Status
1	FCV_642.PV	0	READY	--------	54.593	0	100.00	RMPC	0	NO
2	FCV_662.PV	0	READY	--------	69.454	0	100.00	RMPC	0	NO
3	PP_REF_681.PV	0	READY	--------	0	0	40.000	RMPC	0	NO
4	NTP_REF_561.PV	0	READY	--------	0	20.000	100.00	RMPC	0	NO
5	FCV_541.PV	0	READY	--------	0	0	100.00	RMPC	0	NO
6	FDP_REF_521.PV	0	READY	--------	41.422	20.000	70.000	RMPC	0	NO

Figure 3.96 View Profit Step Size and Status in PSOS

The moves MV moves made by Profit Stepper and the CV responses were observed as shown in Figure 3.97. The Profit Controller unwound these moves to keep the process stable:

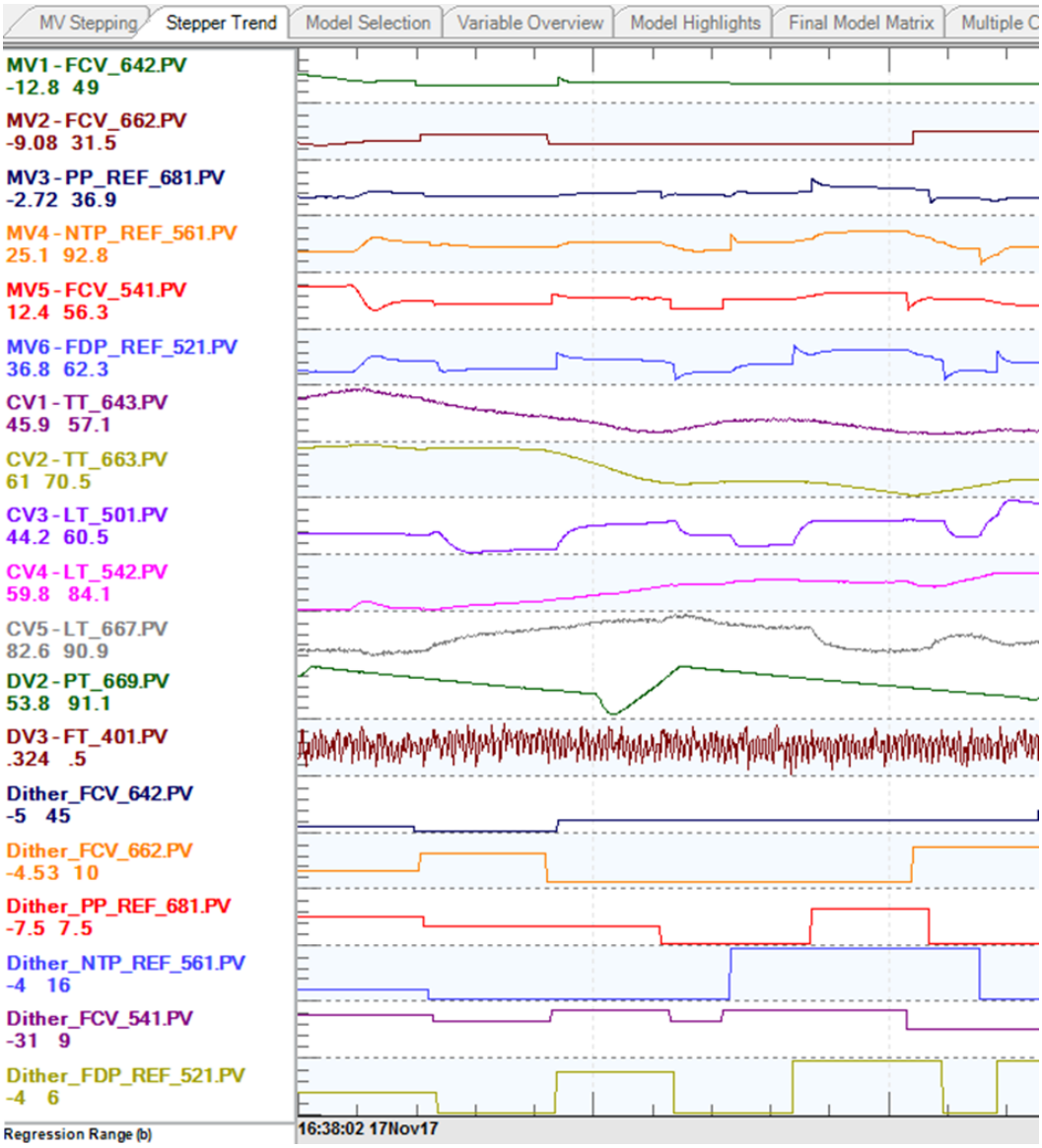


Figure 3.97 Profit Stepper MV moves executed then unwound by Profit Controller

It was possible to combine Rank 1 models from successive Model ID runs into a single model matrix using the Use Prior Models function shown in Figure 3.98. Locked Rank1 and 2 sub-models were loaded into the Profit Controller with the Select Sub-models and Update functions:

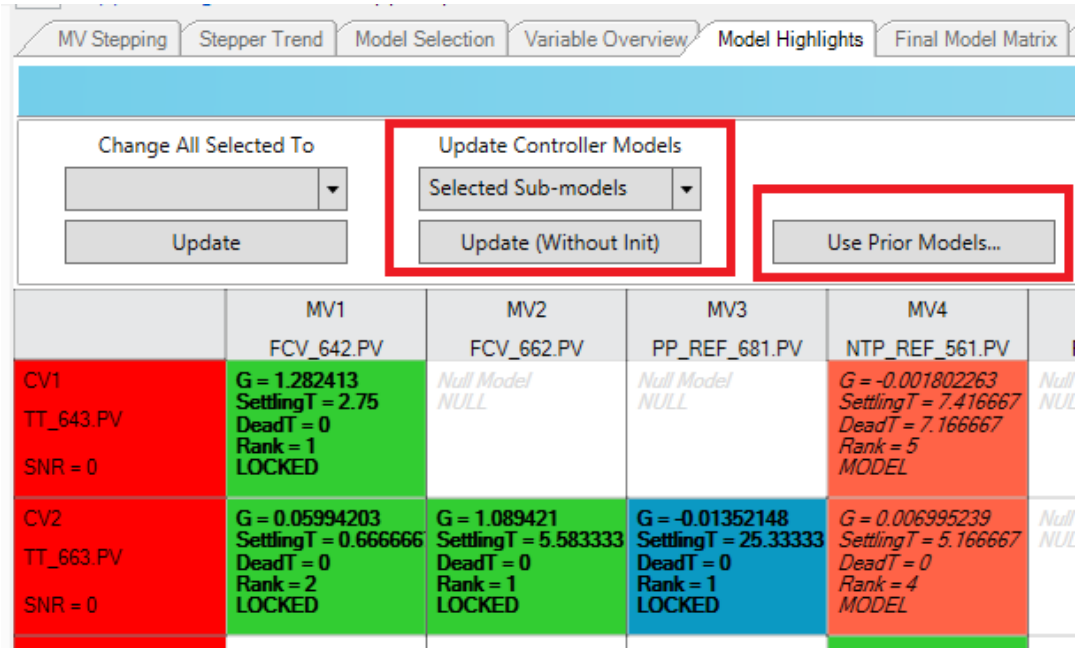
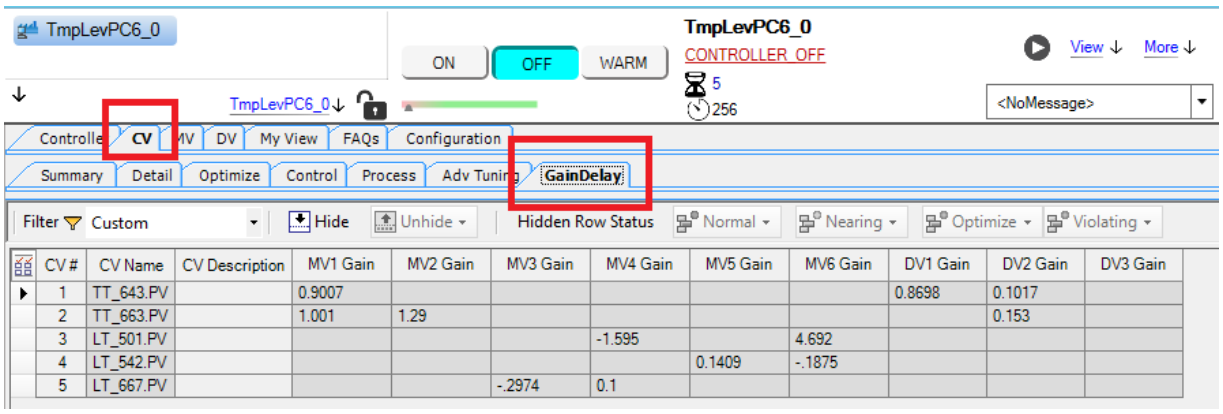


Figure 3.98 Upload Rank 1 or 2 models to Profit Controller

The models of the active Profit Controller did not dynamically change following an update from Profit Stepper. This was discovered in the **GainDelay** tab shown in Figure 3.99. The model gains updated successfully only after the Profit Controller was made Inactive and then Active. The gains in this matrix in PSOS then matched the models found using Profit Stepper. New model performance was tested after confirming they had been loaded into Profit Controller.



CV #	CV Name	CV Description	MV1 Gain	MV2 Gain	MV3 Gain	MV4 Gain	MV5 Gain	MV6 Gain	DV1 Gain	DV2 Gain	DV3 Gain
1	TT_643.PV		0.9007						0.8698	0.1017	
2	TT_663.PV		1.001	1.29						0.153	
3	LT_501.PV					-1.595		4.692			
4	LT_542.PV						0.1409	-.1875			
5	LT_667.PV				-.2974	0.1					

Figure 3.99 Gains Matrix matched models from Profit Stepper once the Profit Controller is re-activated following a model update

3.5.1 Modelling Non-linear Sub-systems

The CSTR temperatures are non-linear systems because it takes far longer to cool the water down than it does to raise the temperature. Equal magnitude positive and negative steps of the steam valves MV do not produce an equal and opposite temperature CV response. However, the Profit Stepper and PDS modelling software aim to fit a linear transfer function to model these systems, which resulted in the Profit Stepper running all day without finding quality models, even with good SNRs. This also caused difficulty in completing the initial offline modelling in Section 3.3.2.

Performing additional step tests in an effort to improve model acquisition was both ineffective and time consuming because of the large time constant of the temperature steps. The solution was found in editing the data in the collector warehouse such that Model ID runs focused only on positive temperature steps to present linear data for modelling. A tool in Profit Stepper called Exclude Data for Regression Calculation was used to select which data was used to fit transfer functions to. Negative steps and irregular plant data (plant start/stop and valve faults) were edited out of the Model ID runs. Figure 3.100 shows how excluded data was shaded brown. The result was the Model ID runs targeted only quality step test data for calculations so the transfer functions did not have to fit large amounts of historical data. The outcome was that models were obtained faster with fewer step tests.

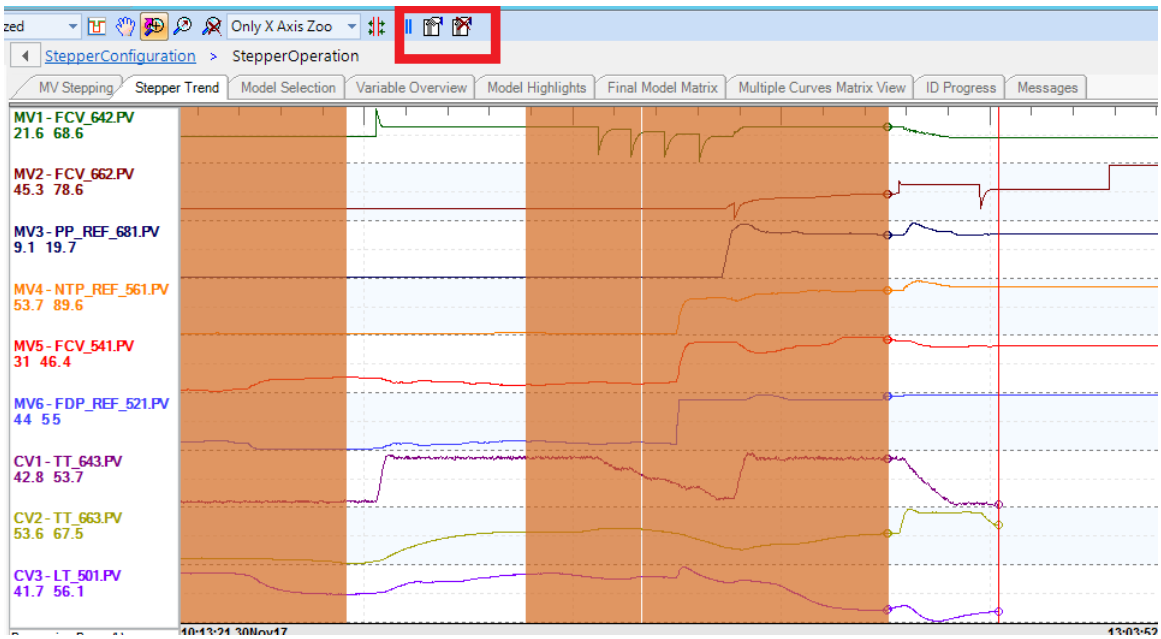


Figure 3.100 Exclude Data for Regression Calculation used to target data for linear Model ID runs

TempPC2 was a Profit Controller used to control Temperatures in CSTR2 and 3, with Needle Tank Pump, Product Pump and steam pressure modelled as disturbances. The models in Figure 3.101 used for the build are between the OP point and the CV, as was normally used in step testing in the Pilot Plant.

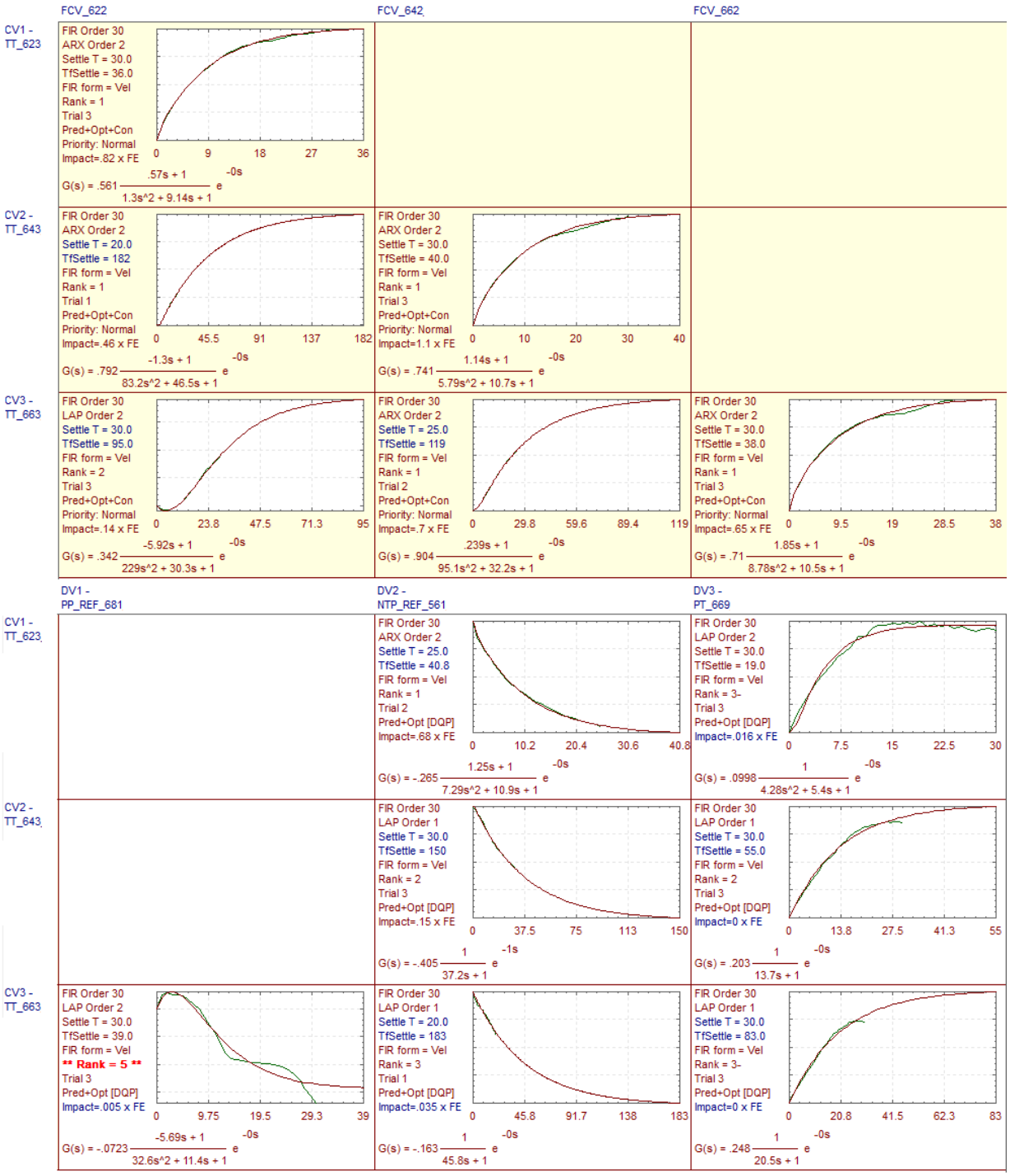


Figure 3.101 Offline PDS models used to build TempPC2. This controlled CSTR2 and 3 temperatures with steam pressure, Needle Tank Pump and Product Pump modelled as DVs

The Profit Stepper models in Figure 3.102 show that when the BLC connections are used to connect to SP instead of OP, the models are of the effect of the PI Controller on the CV. This means the gains should ideally be equal to one, as raising the set-point by 10 degrees should raise/lower the temperature by 10 degrees. The gain of the transfer function between FCV_642 and TT_663 in Figure 3.102 is 0.993 which is to be expected of the PI controller. When the same issue was identified with outlet pumps used as MVs on level controllers, the PDS transfer functions had negative gains when they should have had positive gains. The Profit Stepper was used to acquire correct models, but in future it is recommended that the SP should be manipulated in manual step testing rather than the OP if the intention is for the Profit controller to use the SP as an MV for control.

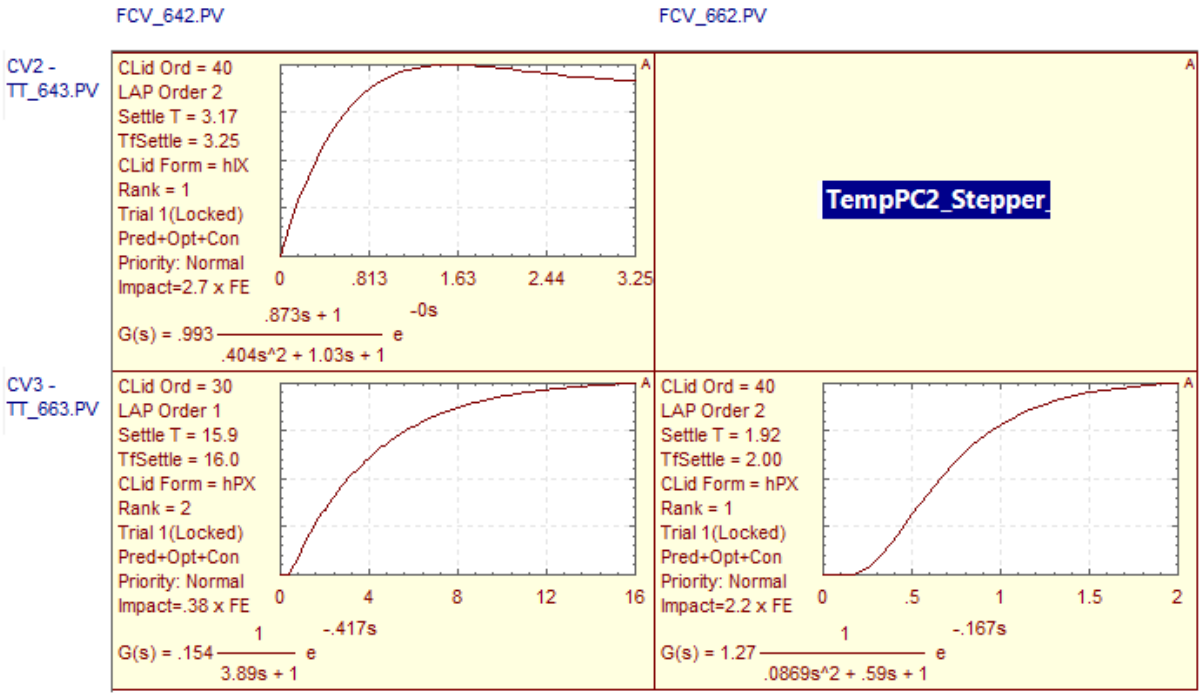


Figure 3.102 Profit Stepper found models between SP of PI controller and CSTR2 and 3 Temperature CVs

The open loop PDS models for CSTR2 and CSTR3 are plotted with these closed loop PI models found using Profit Stepper in Figure 3.104. It shows clearly the temperature PI controller's rapid response in raising the CV when its SP is used as an MV by the Profit Controller as compared to the larger time constant of the open loop model. This information should influence future controller designs and step testing plans.

After the Experion code changes outlined in Section 3.4.3.2 enabled OP manipulation by Profit Suite, the Profit Stepper was used to acquire the open loop transfer functions shown in Figure 3.103. These Rank 1 models are shown alongside the PDS and FOS + Delay models in Figure 3.105.

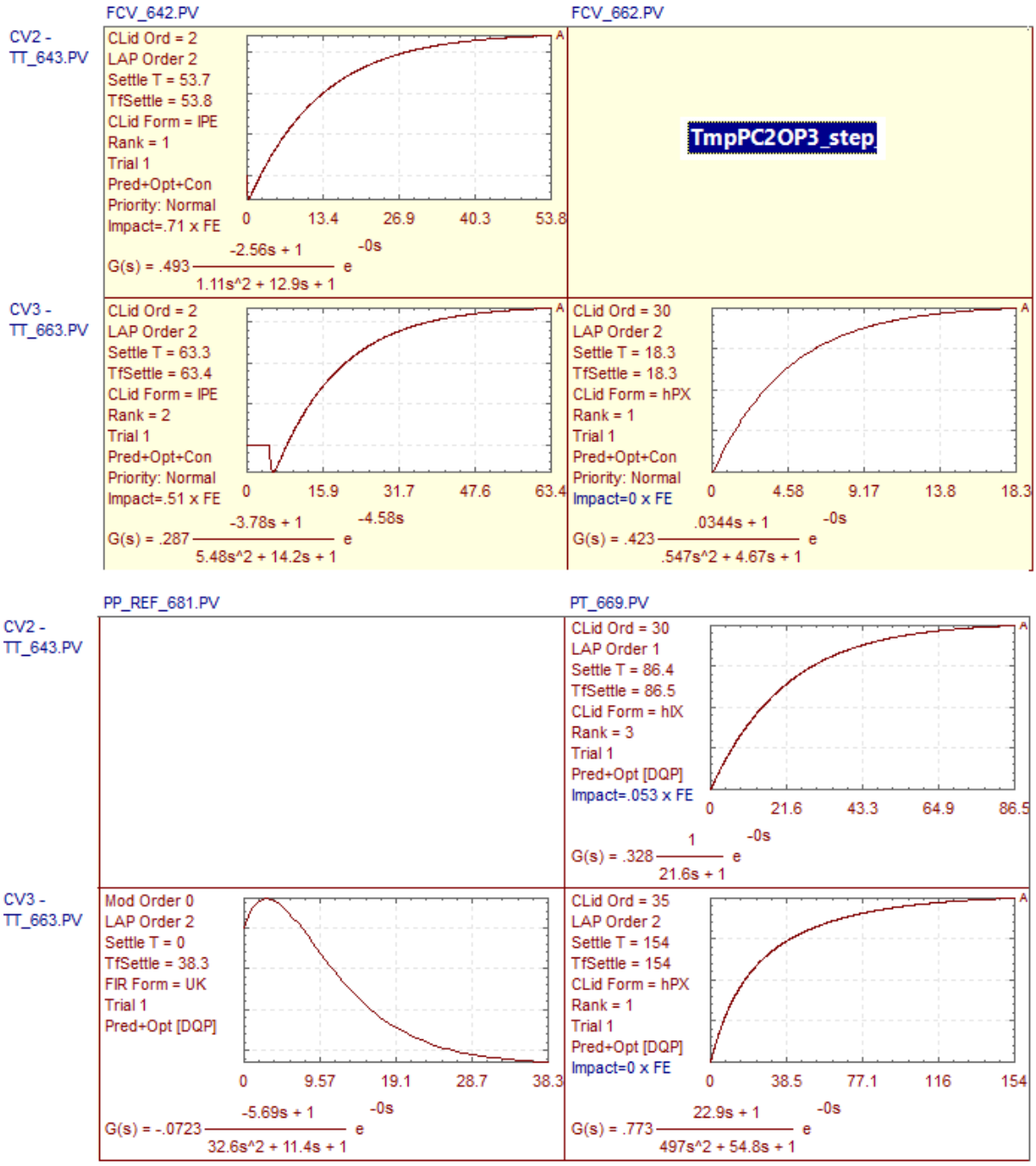


Figure 3.103 OP models found for TmpPC2OP3 Profit Controller with Profit Stepper. This controller used OP points to control temperatures only CSTR2 and 3. CV1 FCV_622 was out of service. Product pump and steam pressure were modelled as DVs for CSTR3 temperature.

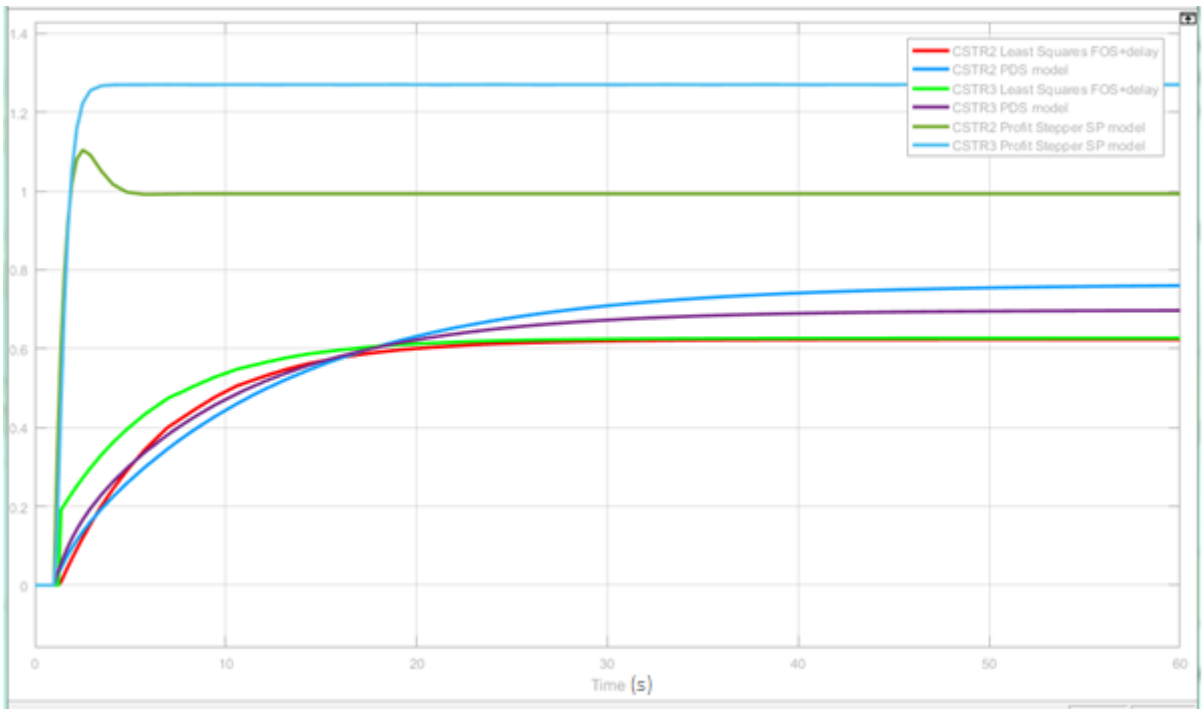


Figure 3.104 Temperature models found with Profit Stepper using SP as MV compared to PDS and FOS+Delay models for FCV_642/CSTR2 and FCV_662/CSTR3

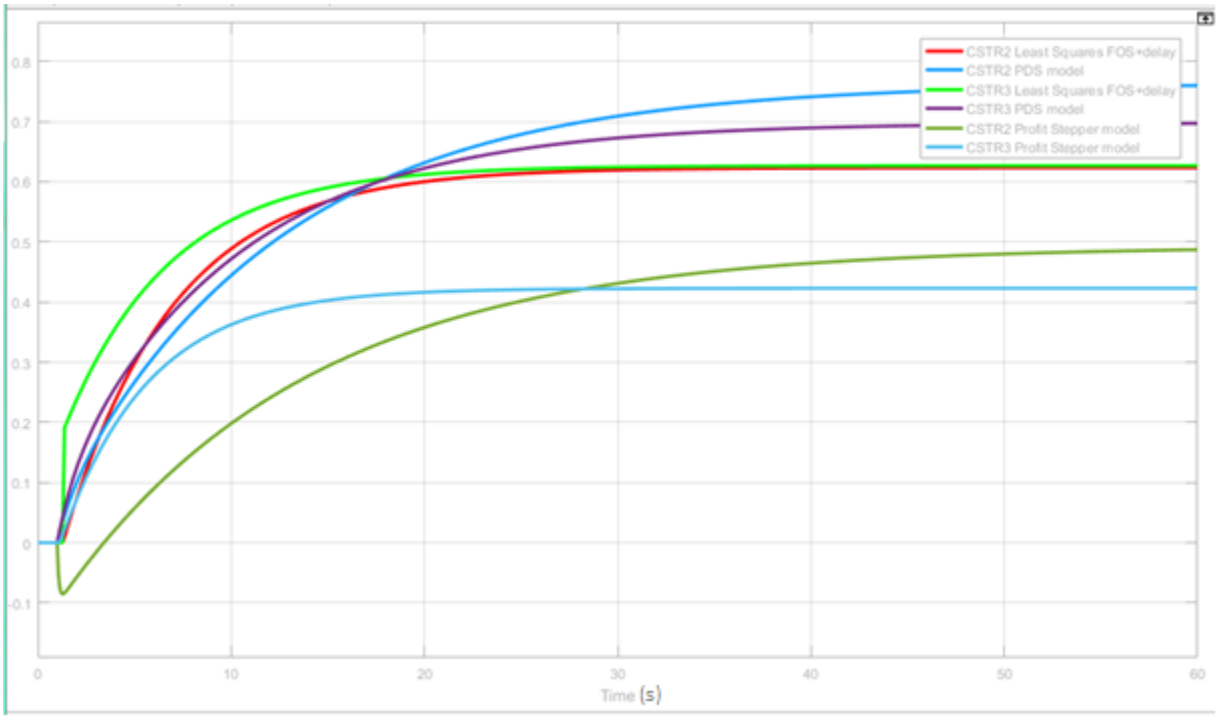


Figure 3.105 Rank 1 Temperature models found with Profit Stepper using OP as MV compared to PDS and FOS+Delay models for FCV_642/CSTR2 and FCV_662/CSTR3

Chapter 4 Profit Controller Results and Analysis

Seven different working Profit Controllers with Profit Steppers were built and tested on the Pilot Plant as part of this project. The objective was to first build MVCs for CSTR temperatures only, then MVCs for temperatures and levels in the second half of the plant. Finally, two MVCs were built to control the entire Pilot Plant. The first controller built was TempPC1 which controlled all three CSTR temperatures with a 1 second Execution rate. Unfortunately, steam valve FCV_622 failed during testing so the performance of this controller could not be used for comparison to other controllers. Baseline PI Control testing was repeated without FCV_622. The results discussed in this chapter are of six subsequent Multivariable Profit Controllers built without FCV_622. These six controllers had 5 second Execution Rates. Table 4.1 provides a summary of the Profit Controllers.

Table 4.1 Summary of Profit Controllers implemented in the Pilot Plant

Profit Controller	Control Structure	Comments
TempPC1	Temperature Control of 3 CSTRs using PID SP as MV	1 second Execution Rate Completed Step testing and Disturbance testing in Pilot Plant before Steam Valve FCV622 failed
TempPC2	Temperature Control of 2 CSTRs using PID SP as MV	Control of Temperatures in Pilot Plant same as for PI Control
TmpLevPC3	Temperature and Level control for 2nd half of plant using SP as MV	Feedback between level controllers around Needle Tank made the process unstable
TempLevPC4_3	Controlled 2 CSTR Temperatures and Level control for 2nd half of plant using SP as MV OP for MV for Flow Disturbance Pump only	Step tests on Needle Tank with Profit Controller using FDP OP were successful Needle Tank Pump PID controller still unstable under Profit Control
TempPC2OP3	Temperature Control of 2 CSTRs using OPs as MVs	Completed Step and Disturbance testing Temperature control sluggish without very low PR
TempLevPC5_0	Temperature and Level Control for 2nd half of Pilot Plant OP as MV for all levels and 2 CSTR temperatures	Code changes in all Experion CM enabled Profit Controller to manipulate OP of all PID blocks Good Level controls; Poor Steam models Completed Step and Disturbance testing **Tested optimization strategies**
PPSTLevels1_1	Tank Level Controls in 1st Half of Pilot Plant using OPs as MVs	Very good steady state control of plant **Tested optimization strategies** to maximize feed rate with limits on recycle

The key performance measure of each control strategy was temperature control of CSTR3. All controllers were tested first for set point tracking then disturbance rejection. Three sources of disturbance were available for testing controller performance; the Flow Disturbance Pump (FDP) flow rate, the Lamella Tank Overflow flow rate and the CSTR recycle stream.

The FDP was used as a manual disturbance when running only the second half of the Pilot Plant. The FDP OP was set at 45% for set-point tracking evaluation, then stepped up to 60% and down to 30% in 15% increments to disturb the temperature and level controllers. Though the FDP can pump much faster, 60% produces the maximum FDP flow rate that the downstream Needle Tank Pump can handle without overflowing the Needle Tank. Conversely, the minimum selectable Product Pump speed will still empty CSTR3 if the FDP is set less than 30% without an additional Lamella Overflow stream. These FDP settings provided the maximum disturbance to the control system while keeping within all pumps' limitations.

The Lamella Tank Overflow connects both halves of the Pilot Plant via the Needle Tank (Appendix B). This flow is a rapidly changing level disturbance to the Needle Tank level. PI controllers that use the Needle Tank Pump as MV respond with aggressive movements to the flowrate into the CSTRs, disturbing the temperature CVs. The amplitude of the lamella disturbance changes as the flow rate increases as shown in Figure 4.1.

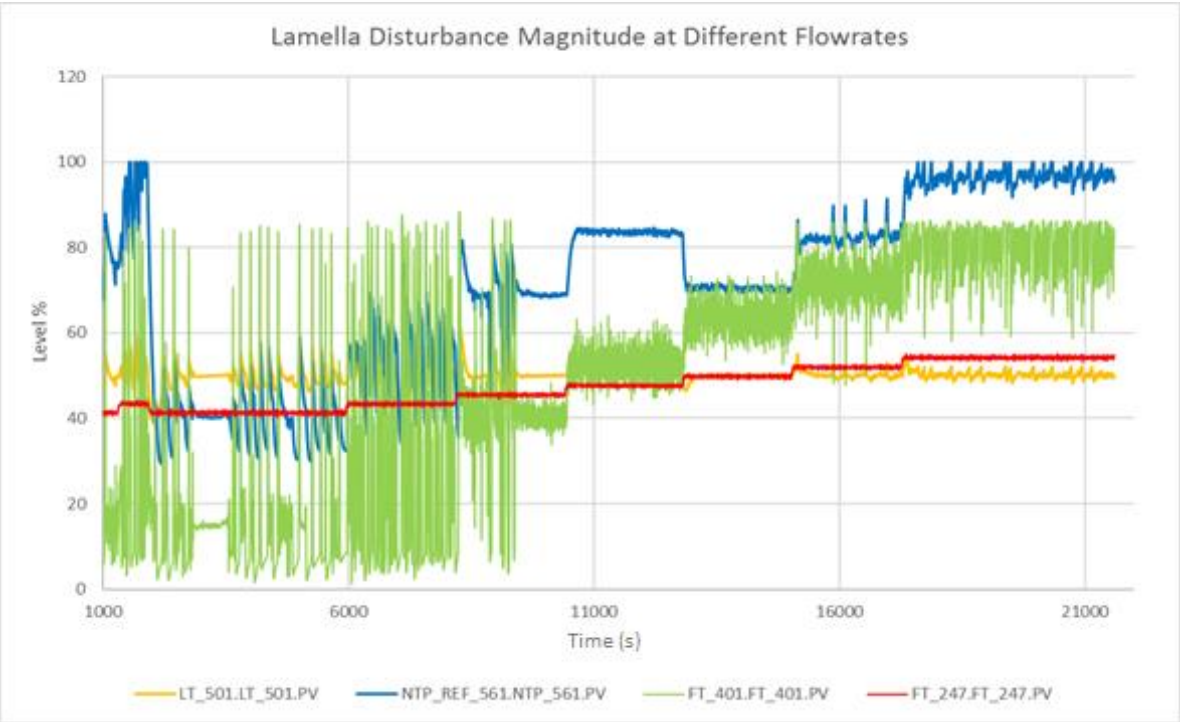


Figure 4.1 Lamella overflows (green) show largest disturbances occurred at around 40% FT_247 flowrates (red). This is the lowest flowrate at which the Lamella Tank will overflow.

The Lamella Overflow was controlled by stepping the feed flow from the supply tank with FP_141. When the flow rate into the Cyclone underflow tank FT_247 increases above 40% the Lamella Tank level will increase and overflow into the Needle Tank. The CUFT level was controlled with PI at 70%. FT_247 flowrates below 40% did not cause the Lamella Tank to overflow. Figure 4.1 shows the magnitude of the disturbance is greatest at low flows. As the flow rate increases the mean flow into the Needle Tank increases linearly, but the variance of the flow values decreases. These characteristics are reflected in the Needle Tank level LT_501 (yellow) and the NT pump speed (blue).

The recycle stream pumped hot water from CSTR3 back into CSTR2 by opening FCV_690 (Figure 2.5) and solenoid valve SV_692. The Product Pump control valve FCV_688 was closed from 100% to 70% to force water back into the recycle stream. This caused a temperature disturbance to the CSTRs and a level disturbance to CSTR3. During initial tests valve FCV_689 would intermittently fail (stuck closed). Replacing the current to pressure (I/P) converter solved the problem.

4.1 Baseline PI Testing and Results

The temperature and level PI controllers for the second half of the plant were commissioned with the tuning parameters outlined in Table 3.6. The performance of these controllers was recorded as a baseline to compare and evaluate the MVC Profit Controllers. The set points are provided in Table 4.2.

Table 4.2 PI Control Set Points

CV		MV		Process	Set Point
Non Linear Tank	LT_542	Raw Water Valve	FCV_541	Level	60%
Needle Tank	LT_501	Needle Tank Pump	NTP_REF_561	Level	50%
CSTR3 Tank	LT_667	Product Pump	PP_REF_681	Level	80%
CSTR2	TT_643	Steam Valve	FCV_642	Temperature	55°C
CSTR3	TT_663	Steam Valve	FCV_662	Temperature	70°C
Flow Disturbance Pump FPD_REF_521		**DV**		Fixed in Manual	OP 60%
Steam Pressure		**DV**		Temperature Disturbance	-

4.1.1 Set Point Tracking

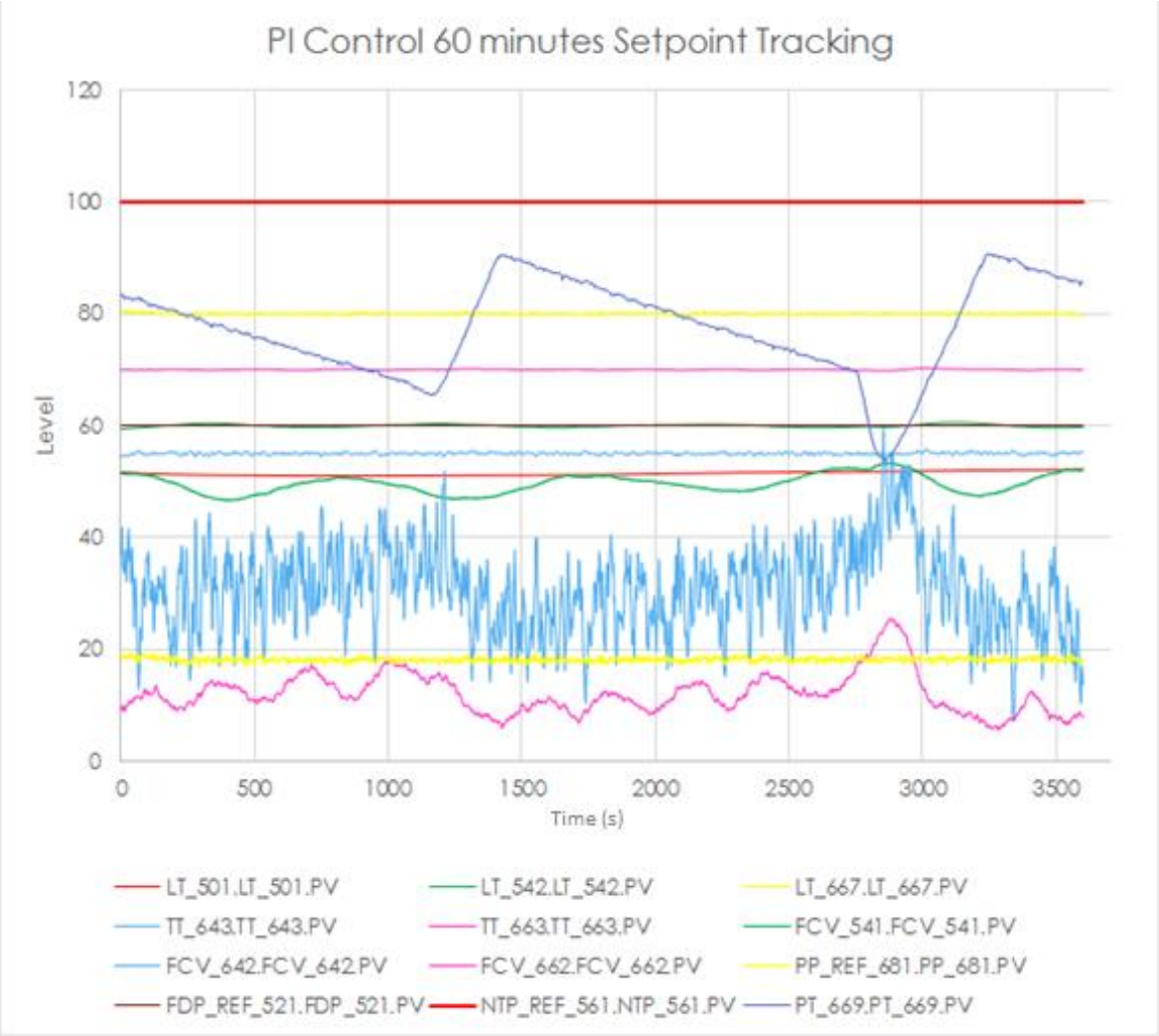


Figure 4.2 Set point tracking for PI control shows negligible disturbance from steam pressure

The set point tracking performance of the Experion PI controllers for both levels and temperatures was excellent.

Level Control

The CVs for all level controllers chart as straight lines in Figure 4.2. The pump MVs are neither aggressive nor oscillating. The MV for the Non Linear Tank is FCV_541 which is the Raw Water Valve. This is a control valve with no feedback and no flow meter on the inlet to NLT. The MV could be the valve position only. The valve oscillates in response to the mains water pressure which act as a disturbance to the level of the NLT. Adding Raw Water to the Supply Tanks also disturbs this controller by reducing mains supply water pressure.

Temperature Control

The set point tracking of the Experion PI temperature controllers was excellent. Both temperatures tracked their respective set points as straight lines with no oscillations regardless of the steam pressure drop around the 2600 second mark. The steam valve MVs responded rapidly to the fluctuations of the steam pressure from the boiler. As a result, the changing steam pressure had a negligible effect on the temperature in either tank.

The temperature in CSTR1 was not being controlled. The MV for CSTR2 temperature is quite aggressive in response to the cold water inflow from CSTR1. However, before the upstream steam valve FCV_622 failed it displayed the same aggressive behaviour in response to the cold water inflow disturbance from the Needle Tank. With the same tuning parameters FCV_642 was as subdued then as FCV_663 is shown here. The maximum errors recorded for the temperature CVs were:

- CSTR2 TT_643 (SP-0.76°C) < SP < (SP +0.7°C)
- CSTR3 TT_663 (SP-0.2°C) < SP < (SP+0.27°C)

4.1.2 Disturbance Rejection Flow Disturbance Pump

4.1.2.1 FDP Step Down

Level Control

The Flow Disturbance pump was stepped down from 45 to 30 %. Figure 4.3 shows the PI level controllers for NLT and NT had returned the processes to set point by approximately 8 minutes. The PI controller took over 3 minutes to reduce the Product Pump OP value to zero. When the product pump OP is set at zero the actual pump is not stopped. Its minimum value still pumped water out of CSTR3 faster than the Needle tank pump was filling it. The level in CSTR3 had not returned to set point after 30 minutes.

Temperature Control

The temperature in CSTR2 was barely affected by this test. Both steam valves closed in response to the reduced water inflow stream, but after a slight initial rise in temperature for the first 2 minutes, CSTR2 continued to track its set point unaffected by the steam pressure disturbance. With the reduced water flow through the tanks, the temperature in CSTR3 began to float up and down with the steam pressure. Although FCV_663 remained closed, this valve was probably passing steam into the CSTR3 heating coil to cause this effect. This tank did not cool down enough to return to tracking the 70°C set point after 30 minutes and had an error of nearly 2 degrees above set point.

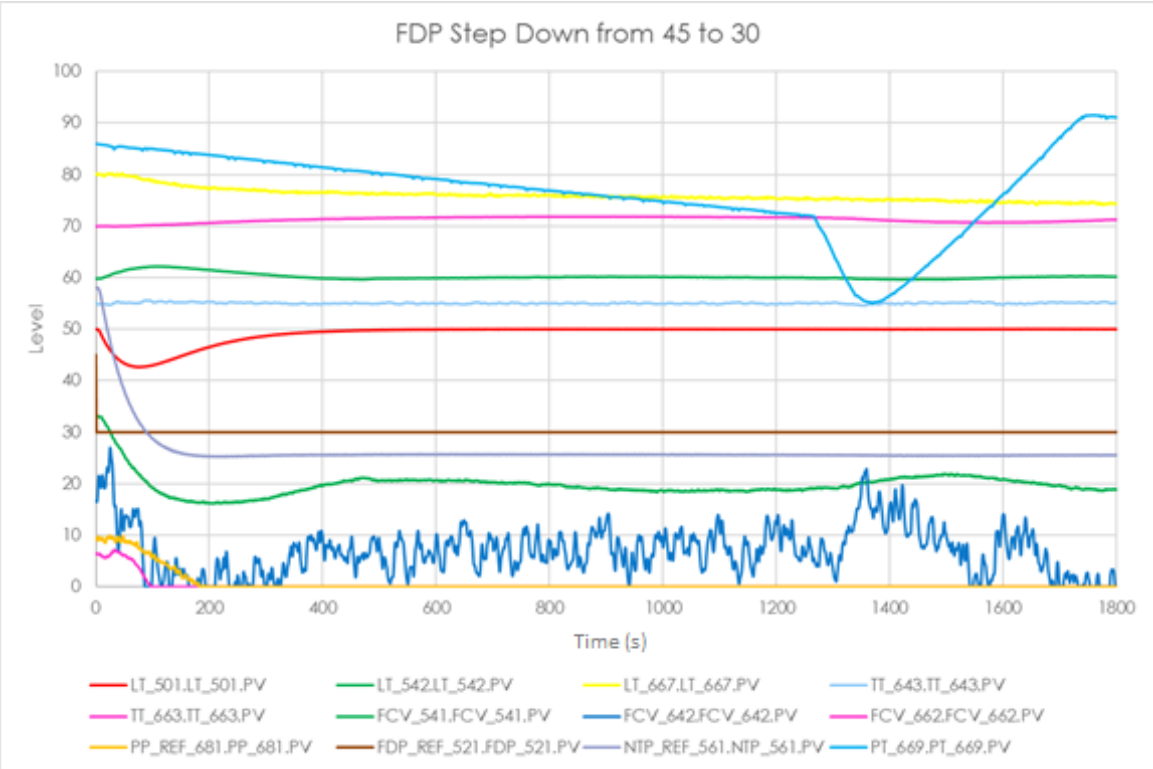


Figure 4.3 PI disturbance: FPD stepped down from 45% to 30%

The maximum errors recorded for the temperature CVs were:

- CSTR2 TT_643 (SP-0.42°C) < SP < (SP+0.58°C)
- CSTR3 TT_663 (SP-0.06°C) < SP < (SP+1.87°C)

4.1.2.2 FDP Step Up

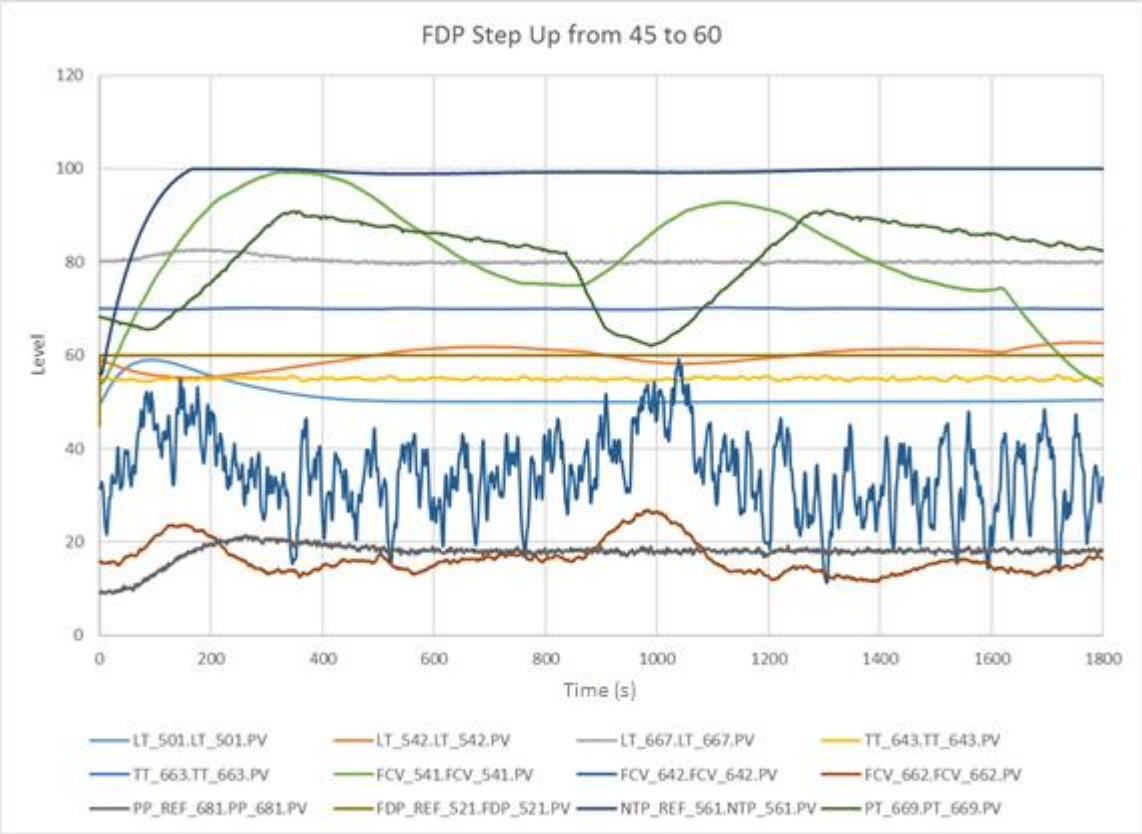


Figure 4.4 PI disturbance: FPD stepped up from 45% to 60%

Level Control

The Flow Disturbance pump was stepped up from 45 to 60 %. Figure 4.4 shows the PI level controllers for CSTR3 and NT had returned the processes to set point after approximately 8 minutes. The level in NLT and its MV FCV_541 show large oscillations for the 30-minute duration. The NLT is approaching steady state at the set point after 1200 seconds but this sub-process was disturbed by the raw water solenoid valve closing at the Supply Tanks. The increased of water pressure available at FCV_541 caused the PI controller to further close the valve to compensate for the resulting level increase in the tank. The level controllers for the Needle Tank and CSTR3 were unaffected by the mains pressure disturbance.

Temperature Control

The temperatures in both tanks were unaffected by either the water flow increase or the steam pressure fluctuations during this test. There is adequate energy capacity available with the steam valves to compensate for the increased water flow and the mean steam valve positions increased to compensate. The temperatures track as flat lines on Figure 4.5 even with the large steam pressure drop around the 1000-second mark. The temperature errors were similar to the set point tracking test.

The maximum errors recorded for the temperature CVs were:

- CSTR2 TT_643 (SP-0.59°C) < SP < (SP +0.7°C)
- CSTR3 TT_663 (SP-0.21°C) < SP < (SP+0.23°C)

4.1.3 Disturbance Rejection Lamella Overflow

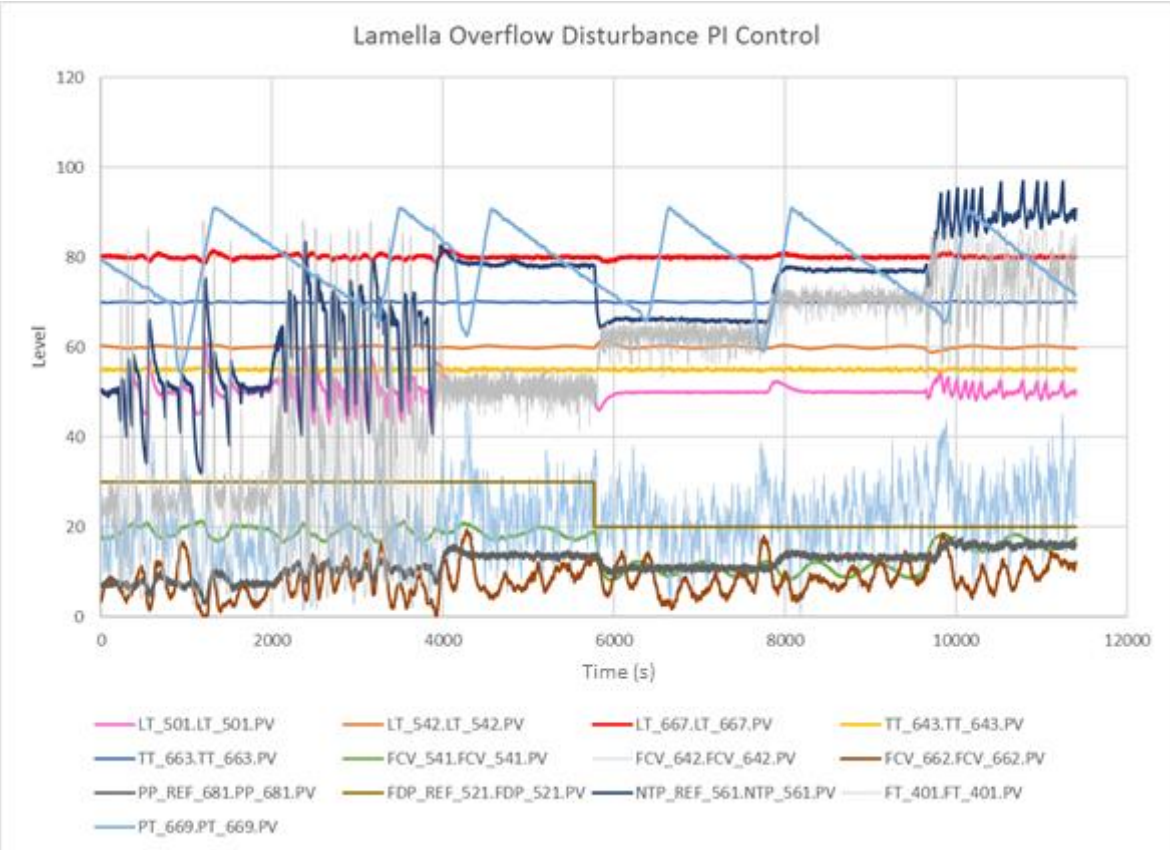


Figure 4.5 Lamella Overflow Disturbance for PI Control in Second Half of Plant

Level Control

The Lamella Overflow measured by FT_401 (Figure 4.5) caused frequent and large NT level disturbances to which the PI controller responded aggressively. The combined inflows from the FDP and the Lamella tank were too much for the NTP once FT_401 average flows were greater than 60%, which is the reason that FDP was reduced to 20% near the 6000-second mark. Before the 4000-second mark when the Lamella disturbance was most prominent, the levels in all three tanks were affected. Once the Lamella overflow became more stable at higher flow rates these tank levels were affected only at the flow step change, not the steady state flow characteristics.

Temperature Control

The irregular flow rate from the NTP caused a disturbance to the temperature in CSTR2 to which steam valve FCV_642 responds. CSTR3 was far less affected as most of the temperature disturbances were dealt with by the CSTR2 controller. The major MV trend of FCV_662 is in effective rejection of the steam pressure disturbance.

The maximum errors recorded for the temperature CVs were:

- CSTR2 TT_643 (SP-0.71°C) < SP < (SP+0.74°C)
- CSTR3 TT_663 (SP-0.27°C) < SP < (SP+0.26°C)

4.1.4 Disturbance Rejection CSTR Recycle Stream

The final 30-minute Lamella Overflow test on at the highest FT_401 flow rate on Figure 4.6 shows significant level disturbances to the Needle Tank. With the FDP reduced to 20%, this was the maximum flow rate possible within the capacity of the Needle Tank Pump. This setting was used to represent the maximum production rate of the entire Pilot Plant. The CSTR recycle stream between CSTR3 and CSTR2 was switched on in addition to this maximum Lamella flow to simulate maximum production with alumina seed/caustic recycle.

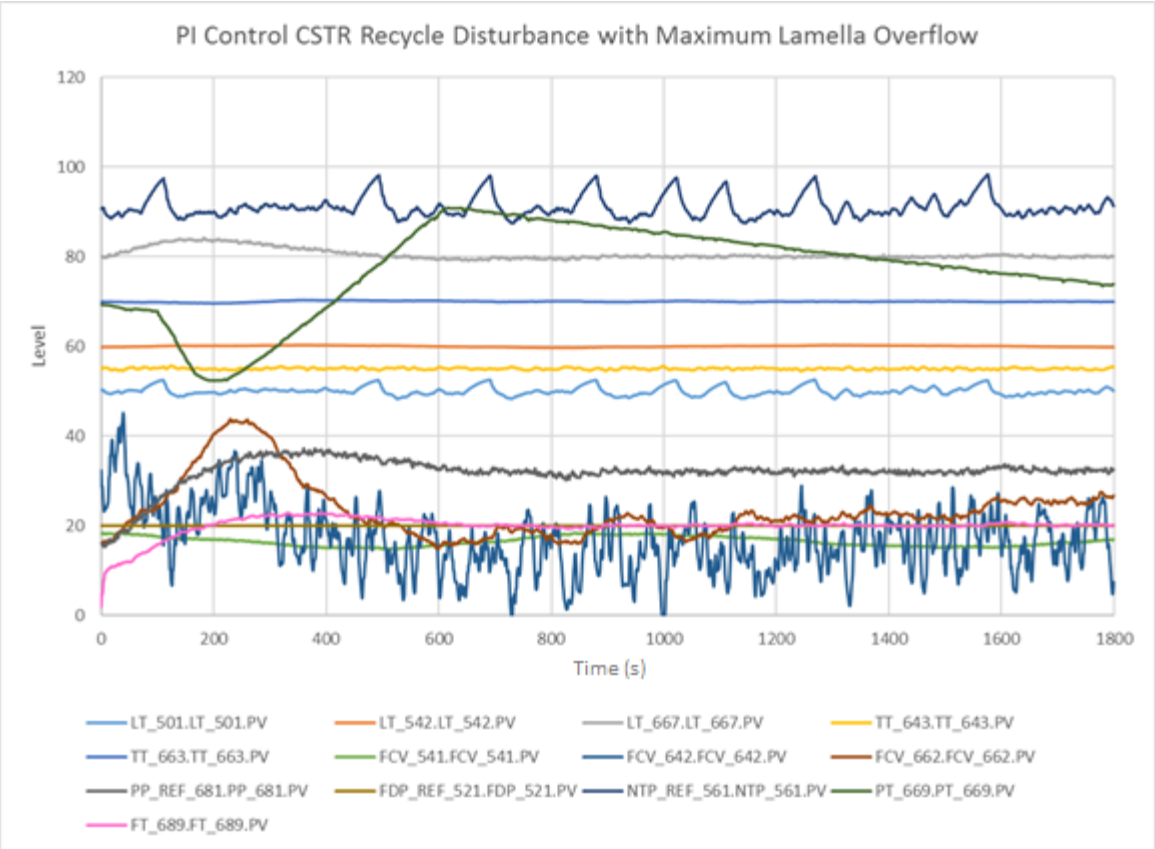


Figure 4.6 CSTR recycle stream shown in pink. The tank level spikes are due to the maximum Lamella overflow into the Needle Tank.

Level Control

The Product Pump speed increased to cope with the extra water from the recycle stream and the returned CSTR3 level to the set point in about 13 minutes. CSTR3 returned slowly to steady state over the 30-minute test. The Lamella flow disturbance to the NT level (not plotted in Figure 4.6 for clarity) caused the aggressive spikes in the NTP speed as the MV response.

Temperature Control

The major action of the temperature controllers during this test was in dealing with the steam pressure disturbance. The recycle stream has less impact. Both steam valves opened in response to the drop in steam pressure at 200s, then the average position of FCV_642 decreased due to the influx of hot water from the recycle stream. The Experion PI temperature CVs were as unaffected by this disturbance test as those previous, as the MV responses effectively dealt with the disturbances.

The maximum errors recorded for the temperature CVs were:

- CSTR2 TT_643 (SP-0.55°C) < SP < (SP+0.66°C)
- CSTR3 TT_663 (SP-0.42°C) < SP < (SP+0.3°C)

4.2 Profit Control in Second Half of Pilot Plant

MVCs were built to control temperature only, then both level and temperature in the second half of the Pilot Plant.

4.2.1 Temperature Profit Controllers

4.2.1.1 TempPC2

The first attempt at MVC of the temperatures of CSTR2 and CSTR3 used the set points of the PI Temperature Controllers as MVs for the Profit Controller. When first training, the impression was that the Profit Controller would manipulate the OP value sent to the valve by assuming command of the Experion PID controller. In practice it was found that the Profit controller simply altered the value of exactly the Point that it was connected to. When connected to the SP as MVs the Profit Controller made miniscule adjustments to SPs of the temperature PI controllers and the OP values behaved as for PI control.

The Profit Stepper was required to find new models between the SP MVs and the temperature CVs in each tank as discussed in chapter 3. The models in Figure 3.102 were used in place of the open loop models. The flow from the NTP and the steam pressure were modelled as disturbances, but these were dealt with by the underlying PI Temperature controller rather than the Profit Controller. The results in Figure 4.7 show that the controller's action with models for steam and NTP caused worse disturbances than the actual DVs themselves.

The PI controllers would react too fast to reject disturbances to enable valid models to be found between FCV_642 and CSTR3 temperature, because the SNR was too low. For example, when the upstream FCV_642 valve was stepped up to increase the temperature in downstream CSTR3, the CSTR3 PI controller reacted as rapidly to reject this disturbance as for any disturbance presented in the baseline PI control test results. CSTR3 temperature would not rise enough for the modelling software to recognize it as anything other than noise.

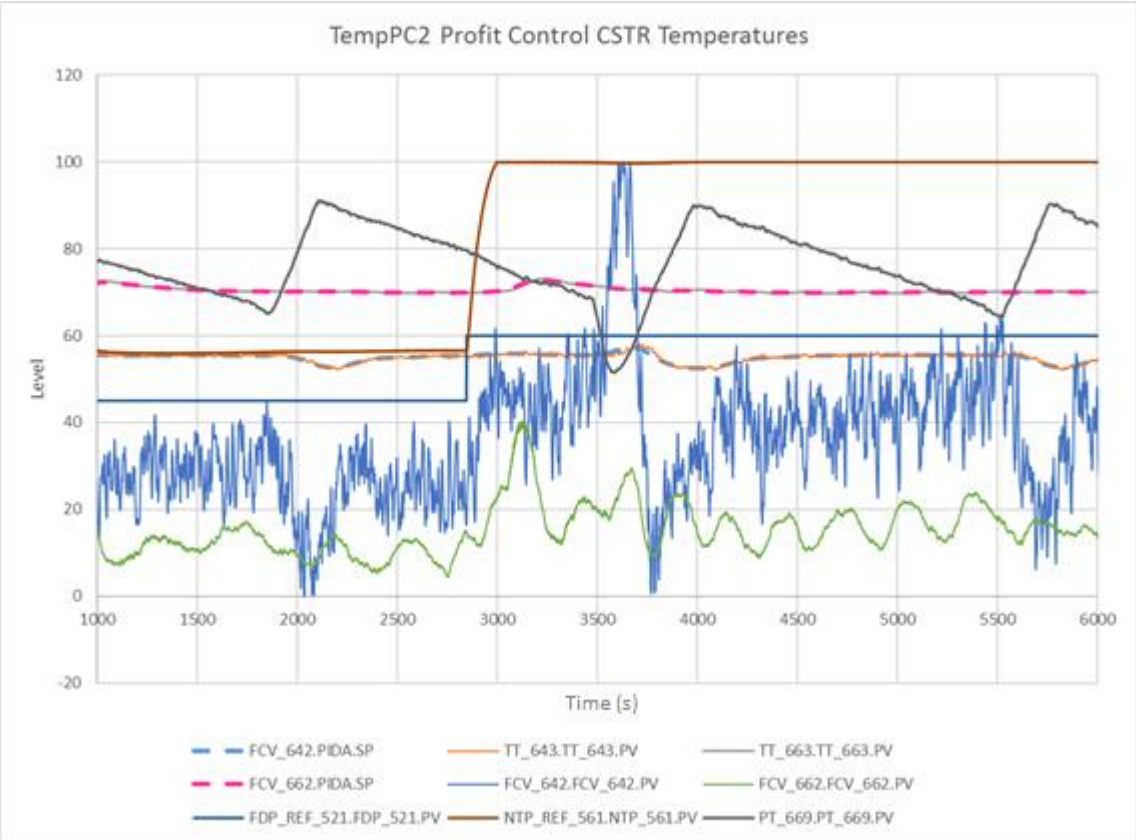


Figure 4.7 Profit Control for CSTR temperatures with TempPC2 controller

The temperature set points are plotted as dotted lines along with the CVs in Figure 4.7. This shows that the Profit Controller was moving the SP of the PI controllers to reject steam pressure and needle tank pump disturbances. The PI controllers tracked the set point exactly as instructed by the MVC. The poor DV models caused the Profit Controller to overcompensate for the disturbances which made control worse than for SISO PI Control. Stepping the FDP up from 45 to 60% at 2800-seconds on Figure 4.7 had little impact on temperatures when compared to the excessive control action due to the DV models.

4.2.1.2 TempPC2OP3

The results of TempPC2 showed that the Profit Controller should directly manipulate the steam flow into each CSTR tank, not the set points of the PI temperature controllers. This would best be achieved using the inner loop of a cascaded PI controller for steam flow. The MVC could manipulate the SP of the steam flow PI controller as an MV to control the CSTR temperature CV. This would eliminate the feedback between PI temperature controllers and enable models to be acquired between each steam valve and all downstream temperature CVs. The Profit Controller should shed the PI steam flow controller to an outer PI temperature controller as necessary. This strategy was not possible because the Pilot Plant has no instruments to measure the steam flowrates downstream of each steam valve.

An alternative strategy to directly control steam flow was trialled in TempPC2OP3. The Experion control code was changed to enable the Profit Controller to write to the OP points to manipulate steam flow instead of the temperature SP. This strategy was successful in that the Profit Stepper was able to step each steam valve with SNRs large enough to acquire Rank 1 models between both MVs and both temperature CVs as shown in Figure 3.103. Steam pressure was modelled as a DV because PI steam flow control was not possible.

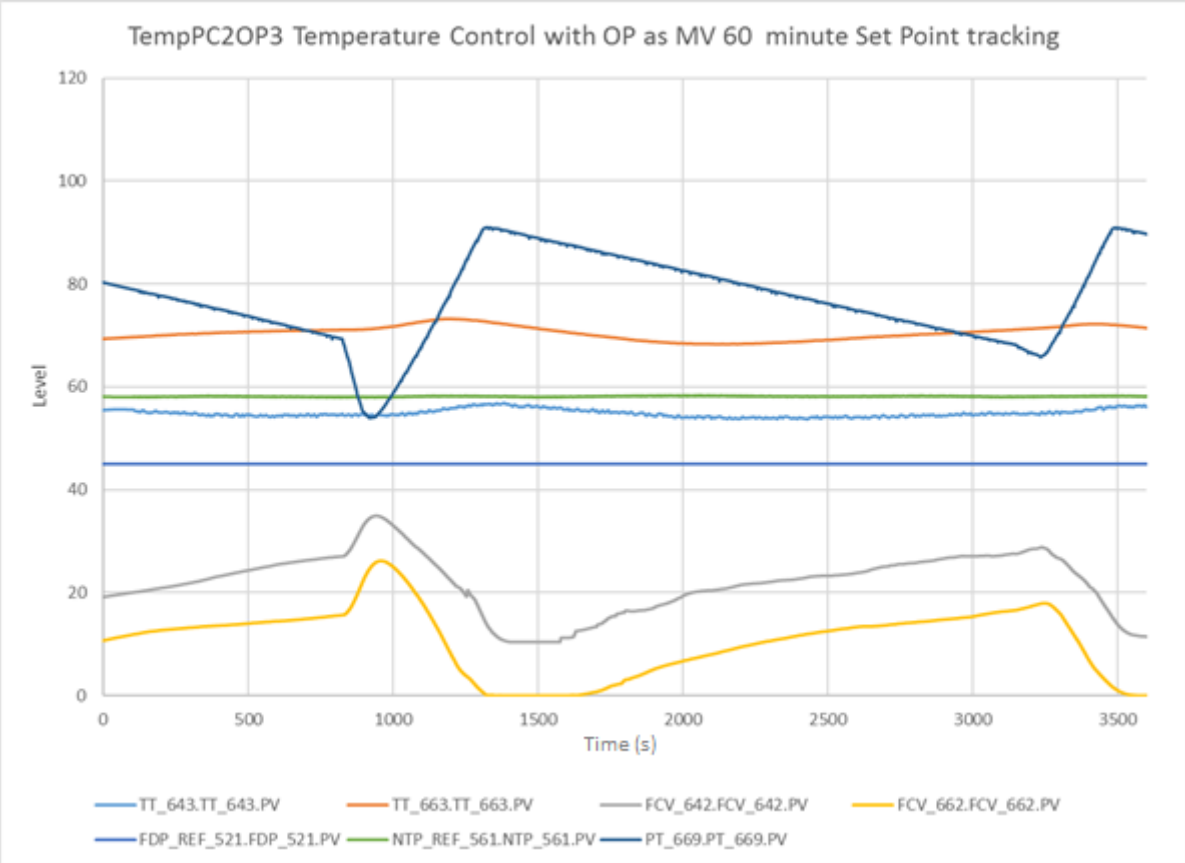


Figure 4.8 Multivariable temperature controlled using PI OP points as MVs

The steady state temperature control of the Profit Controller shown in Figure 4.8 was sluggish. The Performance Ratios were decreased to 0.8 to speed up the response but the results were still worse than for PI Temperature control. The Profit Controller was able to bring the temperatures to set point, but both CVs oscillated in response to the fluctuating steam pressure.

The steam valve MVs exhibited the correct behaviour in counteracting the steam pressure disturbance and are almost a mirror image of the steam pressure curve. However, the modelling between steam pressure and temperatures caused excessive control action. Both steam valves responded immediately to the steam pressure drop at the 1000 second mark on Figure 4.8 because of the DV modelling, but the temperature overshot the set point. The MV action from the Profit Controller was far less aggressive than for PI. Control moves were made only every 5 seconds as opposed to the much faster scan rate of the PI temperature controllers.

The maximum errors recorded at steady state for the temperature CVs were:

- CSTR2 TT_643 (SP-1.3°C) < SP < (SP+1.84°C)
- CSTR3 TT_663 (SP-1.72°C) < SP < (SP+3.23°C)

4.2.1.3 TempPC2OP3 Disturbance Rejection Flow Disturbance Pump

The FDP was stepped up from 45 to 60% and 30 minutes of the temperature response data is shown in Figure 4.9. The temperature CVs were affected more by the steam pressure disturbance and the controller's response to it at t = 1000s, than for the FDP step test at t = 0s.

The maximum errors recorded for the temperature CVs were:

- CSTR2 TT_643 (SP -5.93°C) < SP< (SP -0.23°C) *CV remained below SP*
- CSTR3 TT_663 (SP-1.8°C) < SP < (SP+0.86°C)

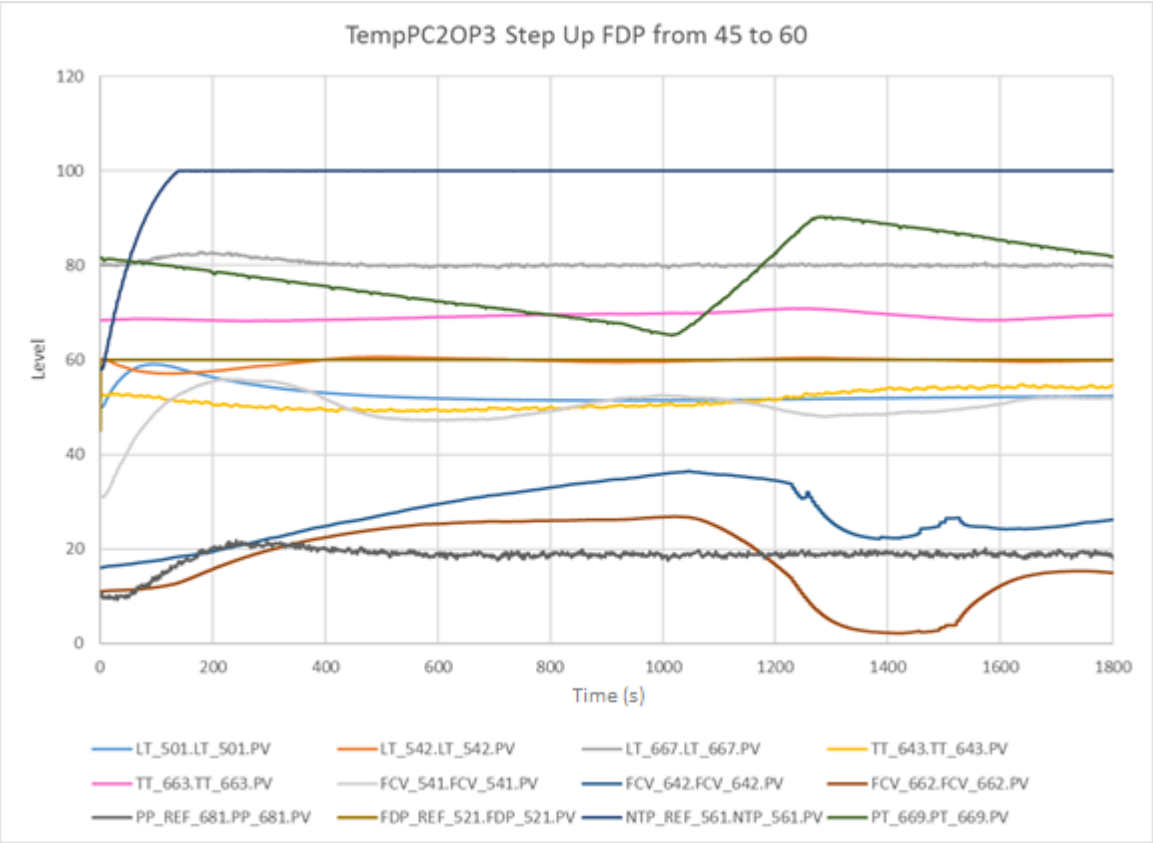


Figure 4.9 FDP stepped up from 45 to 60 at time = 0s

The FDP was stepped down from 45 to 30 and 30 minutes of the Profit Controllers response is plotted in Figure 4.10. The MV response was too sluggish for good temperature control. The steam valves began to close in response to the temperature increase, but then opened for feed forward action for the drop in steam pressure. The Performance ratios for both CVs were further decreased to speed up the response. FCV_642 became unstable at t=500s so its PR was changed back to 0.8.

The maximum errors recorded for the temperature CVs were:

- CSTR2 TT_643 (SP-2.64°C) < SP < (SP+4.41°C)
- CSTR3 TT_663 (SP +0.18°C) <SP< (SP +4.92°C) *CV remained above SP*

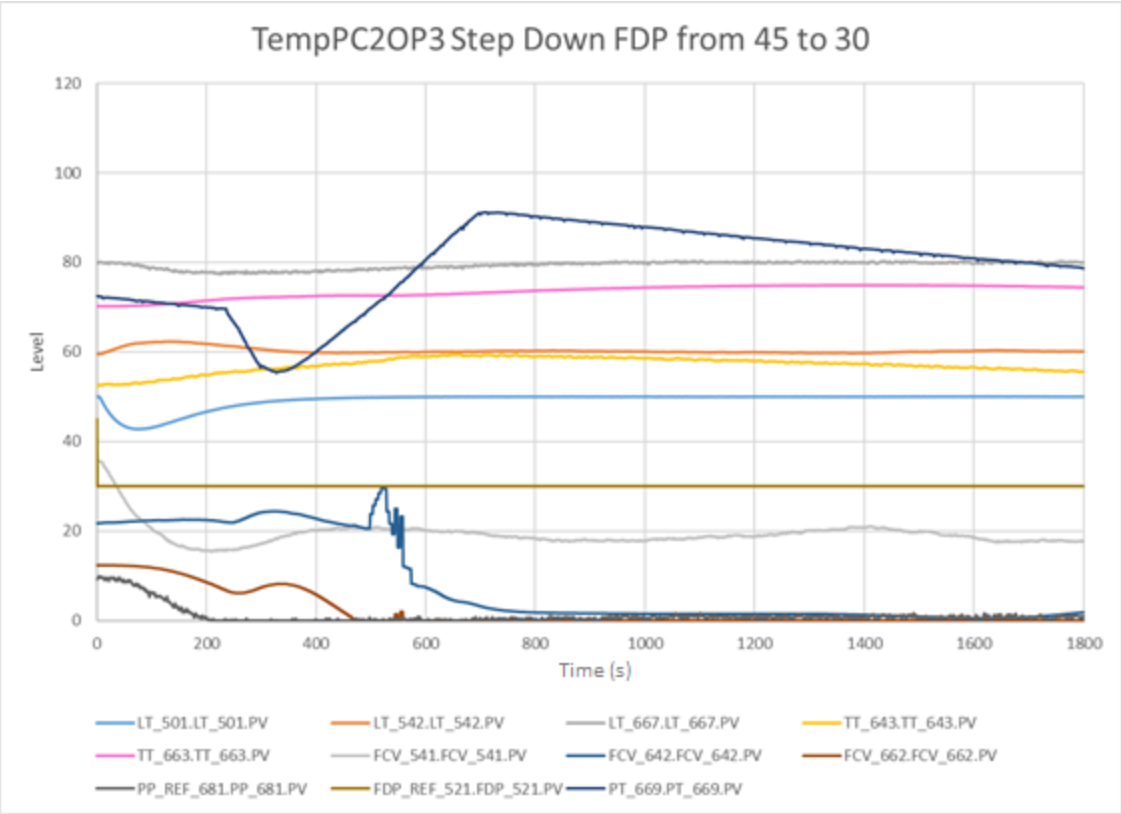


Figure 4.10 FDP stepped down from 45 to 30 at time = 0s

When tested for controller response to the Lamella Overflow Disturbance, the dominant feature was still the steam pressure disturbance. Figure 4.11 shows the Needle Tank level and NTP were greatly affected by the Lamella disturbance, but the erratic flow into the CSTR tanks had little effect in comparison to the steam disturbance. Both temperature trends float up and down with the steam pressure even though the steam valves move to counteract it. The response of the controller is too slow to deal with the disturbance, and the Lamella disturbance appears as noise in comparison.

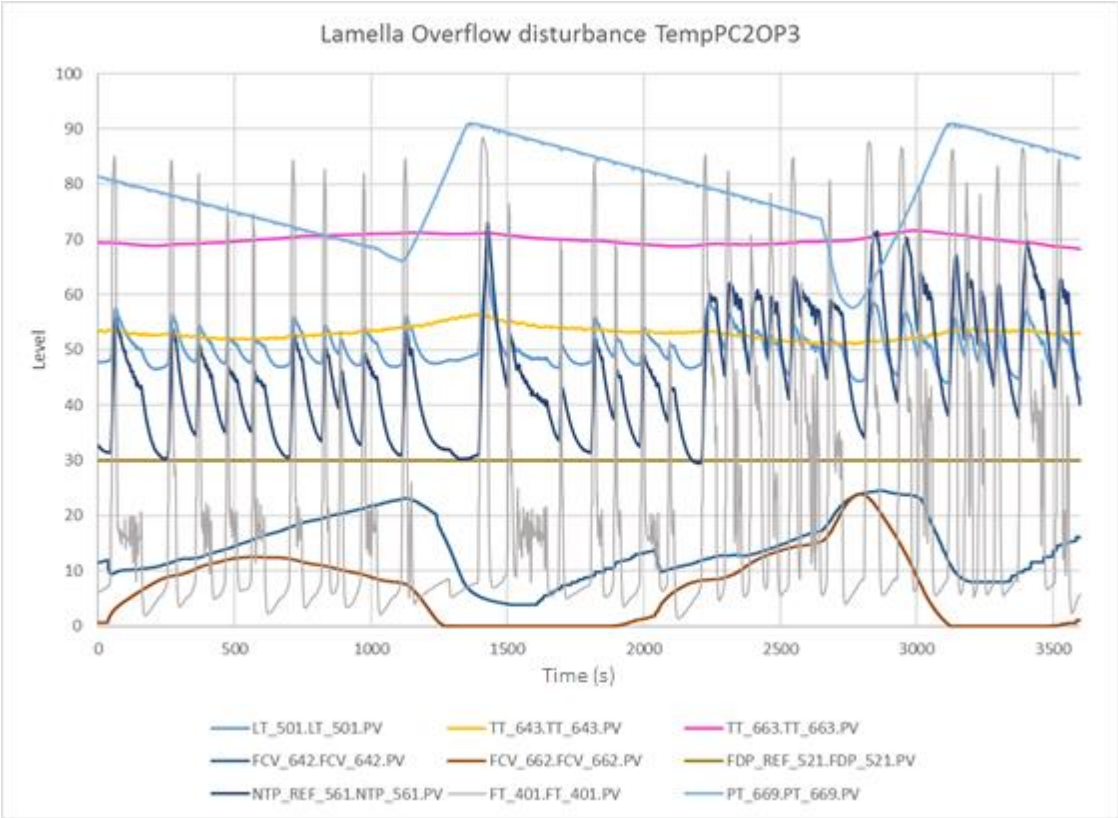


Figure 4.11 TempPC2OP3 Lamella Overflow Disturbance Rejection

The maximum errors recorded for the temperature CVs were:

- CSTR2 TT_643 (SP-4.08 °C) < SP < (SP+1.46 °C)
- CSTR3 TT_663 (SP-1.72°C) < SP < (SP +1.63°C)

The CSTR Recycle Stream was switched on for a 30-minute disturbance test as shown in red on Figure 4.12. The controller did not have precise enough control on the temperature CVs to clearly observe the effect of the recycle disturbance. The dominant characteristic was still the MVs tracking the steam pressure trend, and slowly oscillating CVs. The controller response could not be sped up with the PRs enough without becoming unstable when disturbed. Adjustments of the PR resulted in the jagged steps on the FCV_642 trend. The PR for FCV_662 could be set as low as 0.5 without becoming unstable, but the performance was still worse than for PI control.

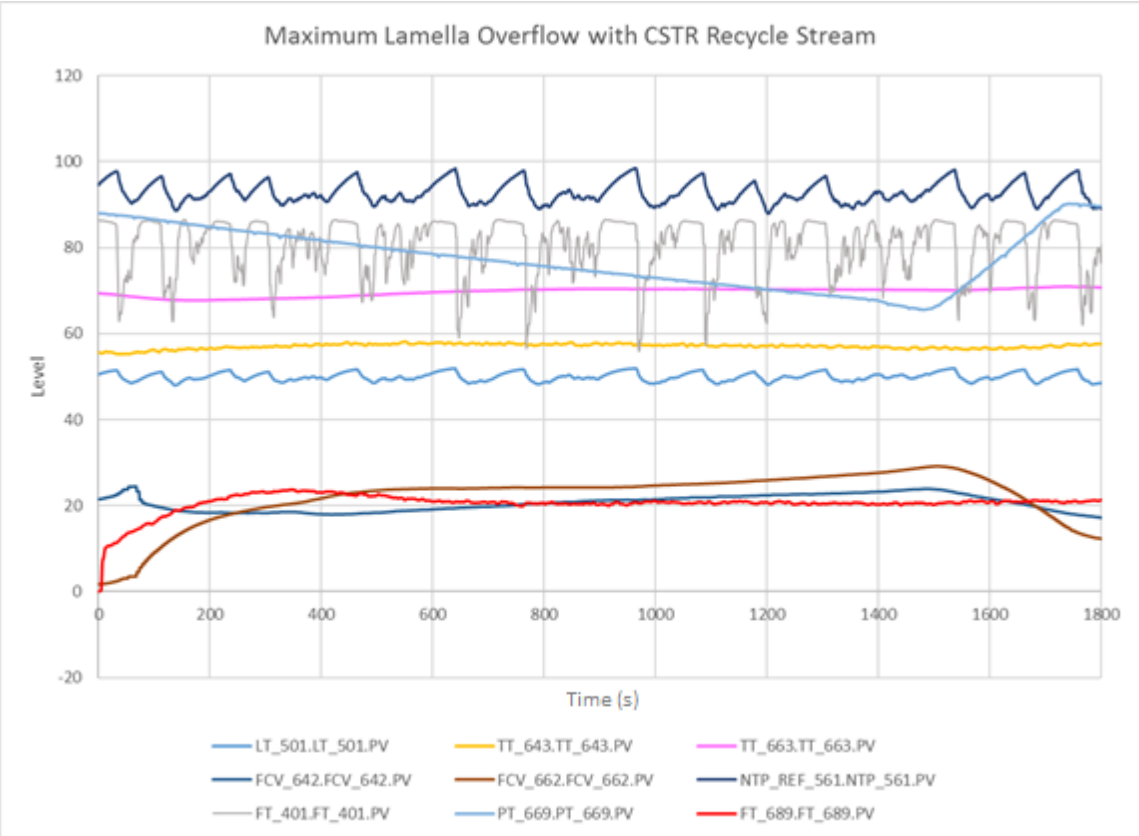


Figure 4.12 Recycle stream disturbance with lamella disturbance for TempPC2OP3

The maximum errors recorded for the temperature CVs were:

- CSTR2 TT_643 (SP +0.21°C) <SP< (SP +3.16°C)
- CSTR3 TT_663 (SP -2.25°C) <SP< (SP +0.96°C)

4.2.2 Tank Level Profit Controllers for Second Half of Plant

Three Profit Controllers which controlled both level and temperature were built with different control structures and BLC:

- TmpLevPC3
- TempLevPC4_3
- TempLevPC5_0

4.2.2.1 TmpLevPC3 Level and Temperature Profit Controller

The first Profit Controller was designed using the SPs of the PI controllers as MVs for the MVC. As discussed for Figure 3.33, feedback between the two PI level controllers for the Needle Tank caused the process to become unstable. There is no fundamental relationship between the SP of a level controller and the level in a downstream tank, so multivariable control was not possible with this approach. For example, the MVC cannot use the PID Level SP to change the level in the Needle Tank and predict the effect on the temperature in CSTR2. The fundamental relationship is between the water flow from the Needle Tank Pump and the temperature in CSTR2; not the level in the Needle Tank.

4.2.2.2 TempLevPC4_3 Level and Temperature Profit Controller

TmpLevPC3 was rebuilt as TempLevPC4_3 with new BLC to try to experiment with controlling tank levels using the OP point instead of the SP. The OP is the position or speed value written to a valve or pump output. This Profit Controller connected to FDP using the OP point as MV to directly control the speed of the pump and therefore the flow into the Needle Tank. Code changes in Experion were required to enable the Profit Controller to write to the OP point.

The Profit Stepper found a stable model for FDP OP and Needle Tank Level CV as in Figures 4.13 and 4.14, not an integrator. Step tests on the NT level using the FDP with Profit Control proved that using the OP for level control was successful (Figure 4.15). The Profit Controller manipulated the flow into the Needle Tank and controlled the level in the tank, not the SP of a PI level controller.

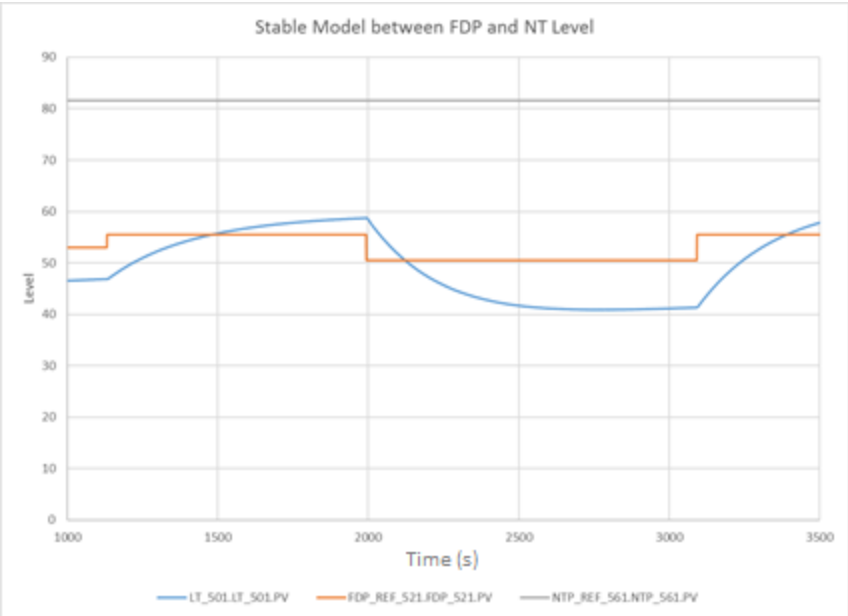


Figure 4.13 Profit stepping found a stable model for FDP and NT level not an integrator

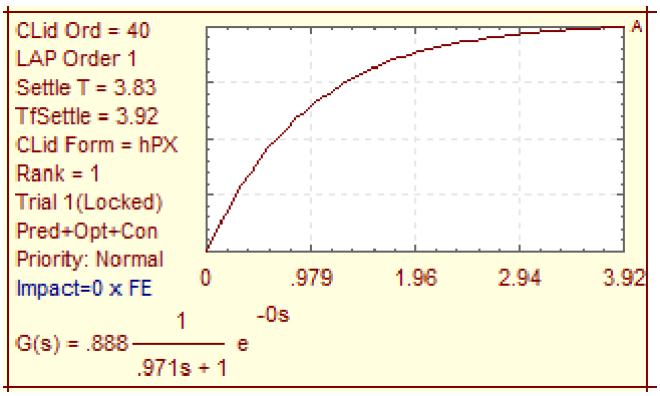


Figure 4.14 Rank 1 stable model between FDP OP and NT level CV enabled successful control of NT Level with NTP fixed in manual

The maximum MV moves in PSOS were increased for PDP to allow the controller to make large moves to the OP point for good control of the CV.

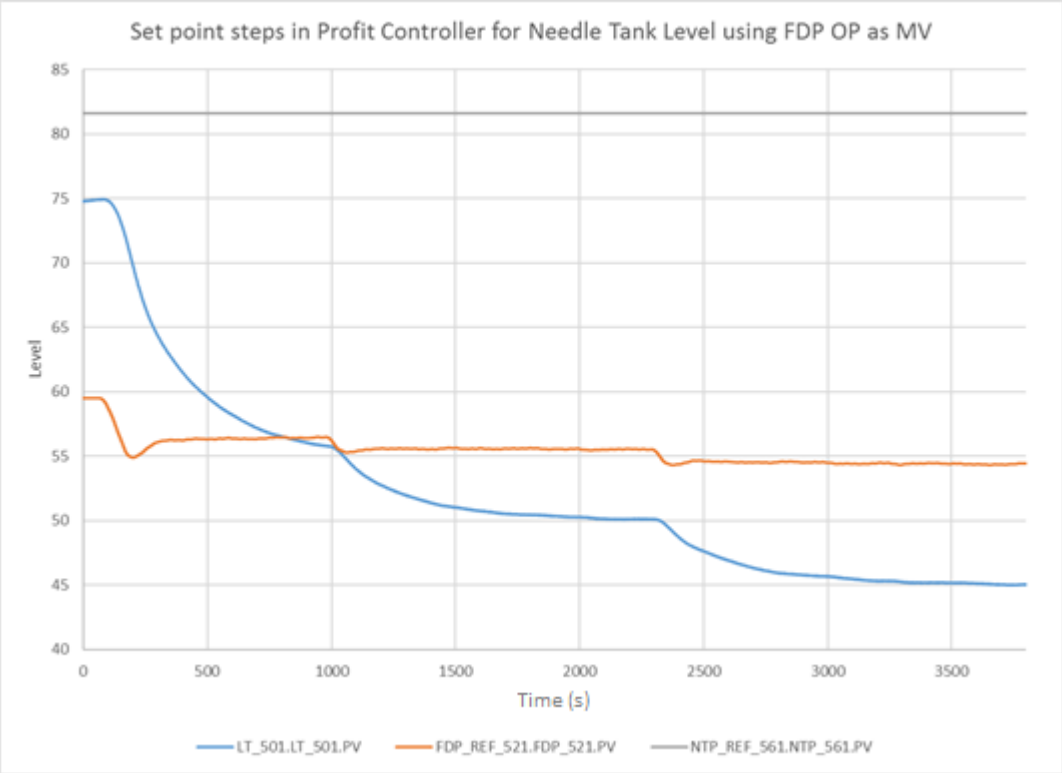


Figure 4.15 Profit Controller successfully controlled NT level with FDP using OP point. NTP fixed in manual.

The NTP MV was dropped from Profit Control when testing the performance of using FDP OP as MV. When the NTP was given back to the Profit Controller as an MV, the Needle Tank became unstable due to interactions between the PI level controller and Profit Control. The PI controller reacted to the FDP as a disturbance. This controller did confirm that levels could be controlled successfully by connecting to the OP point to directly manipulate flow, instead of the level SPs of the PI controllers.

4.2.2.3 TempLevPC5_0 Level and Temperature Profit Controller

Code developed to enable Profit Suite to write to PIDA OP points was applied to all Experion CMs for Level and Temperature. This resulted in actual multivariable control of all levels and temperatures in the second half of the plant. Tank level control performance was good; however, temperature control was still affected by the steam pressure disturbance.

The Profit Stepper was used to acquire Rank 1 models between OPs and CVs (Appendix B Figures 9.3 - 9.8). Models were found for the NTP/CSTR2 Temperature and NTP/CSTR3 Temperature, as well as FCV_642/CSTR2 Temperature and FCV_642/CSTR3 Temperature. This resulted in clear observable multivariable control action (Figure 4.16). The NTP speed tracked the steam pressure disturbance because the Profit Controller used it as an MV for temperature in addition to the two steam valves. When the steam pressure increased the energy flow into the tank, more cold water was supplied from the needle tank to counteract the disturbance and the steam valves were closed.

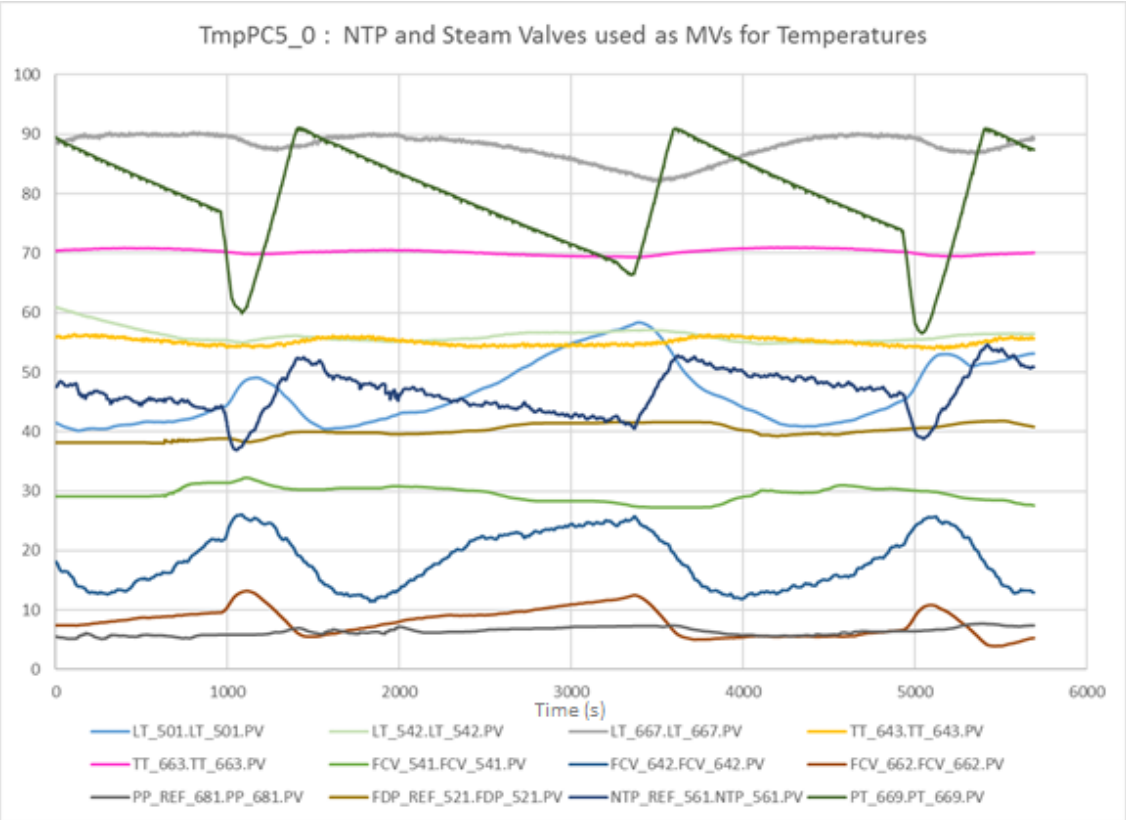


Figure 4.16 Observable multivariable control with NTP and FCV_642 and FCV_662 controlling temperatures

The CV give-ups were structured in PSOS to give the highest priority to CSTR3 temperature, then CSTR2 temperature, followed by CTR3 level then NT level and finally NLT level. The plant was run at steady state for 60 minutes and the results recorded in Figure 4.17.

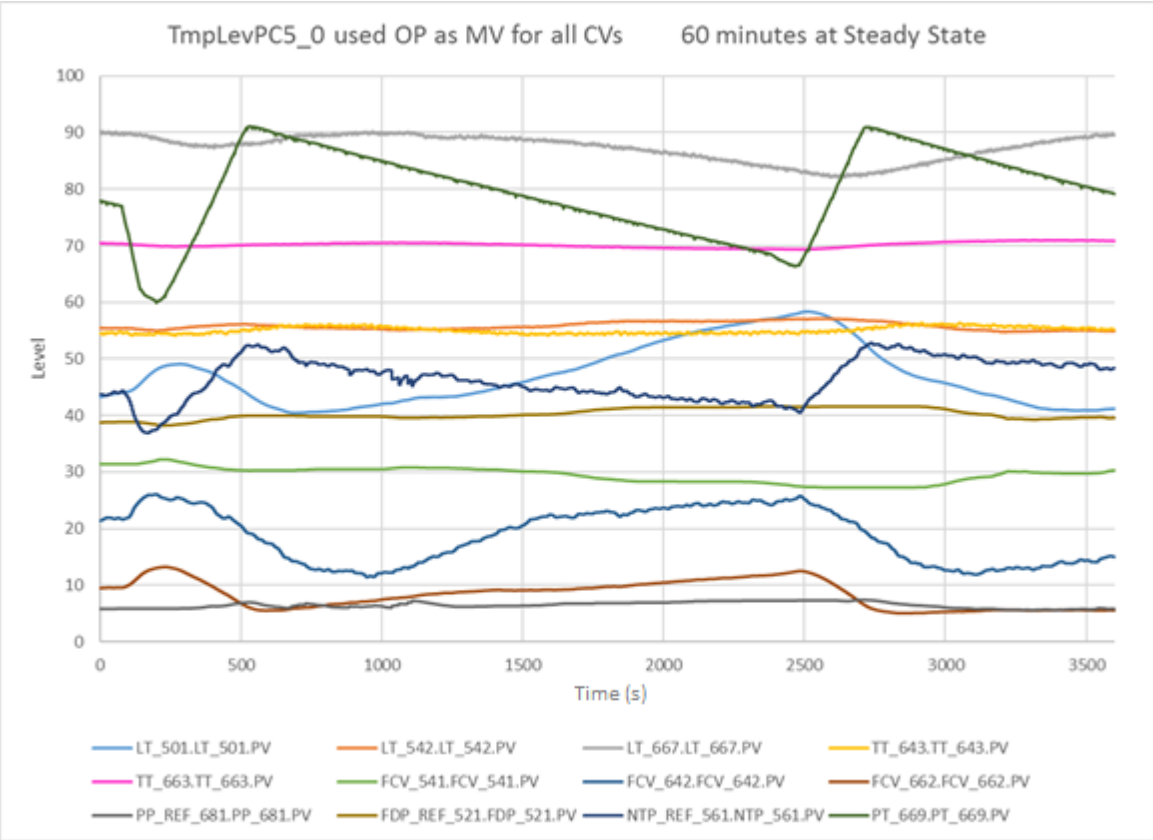


Figure 4.17 60 minutes at steady state with all levels and temperatures controlled using OP as MV by Profit Controller

The temperature control performed better with multivariable level control than for TempPC2OP3. The tank levels were controlled to ranges instead of set points. These settings gave the MVC the degrees of freedom required to use the NTP for temperature control instead of just NT level control. Table 4.3 shows the settings used. The temperatures were controlled first to set points as well as ranges with Profit Control. When temperature ranges were used, soft limits were set at the set point levels to drive the temperatures to the previous PI set points for comparison.

Table 4.3 Ranges used as hard limits for multivariable temperature and level control

CV		Set Point previously for PI control	Sub-Process	Profit Control Range of hard CV limits
Non Linear Tank	LT_542	60%	Level	55 – 65 %
Needle Tank	LT_501	50%	Level	45 -55 %
CSTR3 Tank	LT_667	80%	Level	80 - 90%
CSTR2	TT_643	55°C	Temperature	50 -60°C or SP =55°C
CSTR3	TT_663	70°C	Temperature	65 - 75°C or SP =70°C
Steam Pressure		**DV**	Temperature Disturbance	-

The results of Profit level control were quite different to the PI control because the tank level were controlled to ranges instead of set points. The give-ups used dictated that levels were not as important as the temperatures, so the tank levels were allowed to drift somewhat within their ranges. This was most evident in the level of the Needle Tank because the NTP was an MV for the higher priority temperature CVs. The FDP was instead used as MV for both the NT and NLT. However, there was not much control action by the level MVs while the level CVs stayed within hard limits.

The maximum errors recorded for the temperature CVs were:

- CSTR2 TT_643 (SP -0.89°C) <SP< (SP +1.36°C)
- CSTR3 TT_663 (SP -0.65°C) <SP< (SP +0.98°C)

4.2.2.4 TempLevPC5_0 FDP Disturbance rejection

The FDP was dropped from Profit Control so it could be used as a disturbance. The FDP was stepped up from 45 to 60 % in Figure 4.18. All other pumps are immediately sped up due to the models between pump speeds and levels. The temperature of CSTR3 is unaffected by the disturbance because the steam valves are also opened immediately. The Profit controller enacted feedforward action to the disturbance, unlike the PI control which required feedback after the levels had changed.

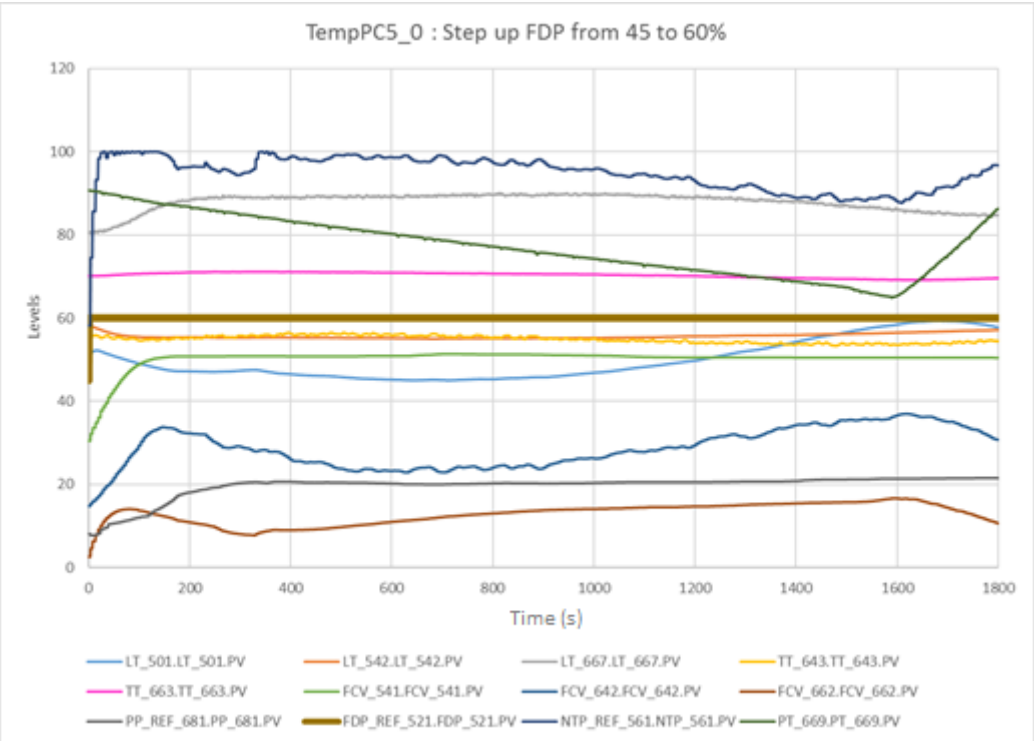


Figure 4.18 FDP stepped up by 15% saw feedforward action taken by the Profit Controller

The maximum errors recorded for the temperature CVs were:

- CSTR2 TT_643 (SP -1.71°C) <SP< (SP +1.51°C)
- CSTR3 TT_663 (SP -0.85°C) <SP< (SP +1.14°C)

The same behaviour was observed in Figure 4.19 when the FDP was stepped down from 45 – 30%. Reducing the PR for the temperature CVs below 0.8 caused erratic behaviour of the NTP in 4.19 so it was increased to 0.8 again.

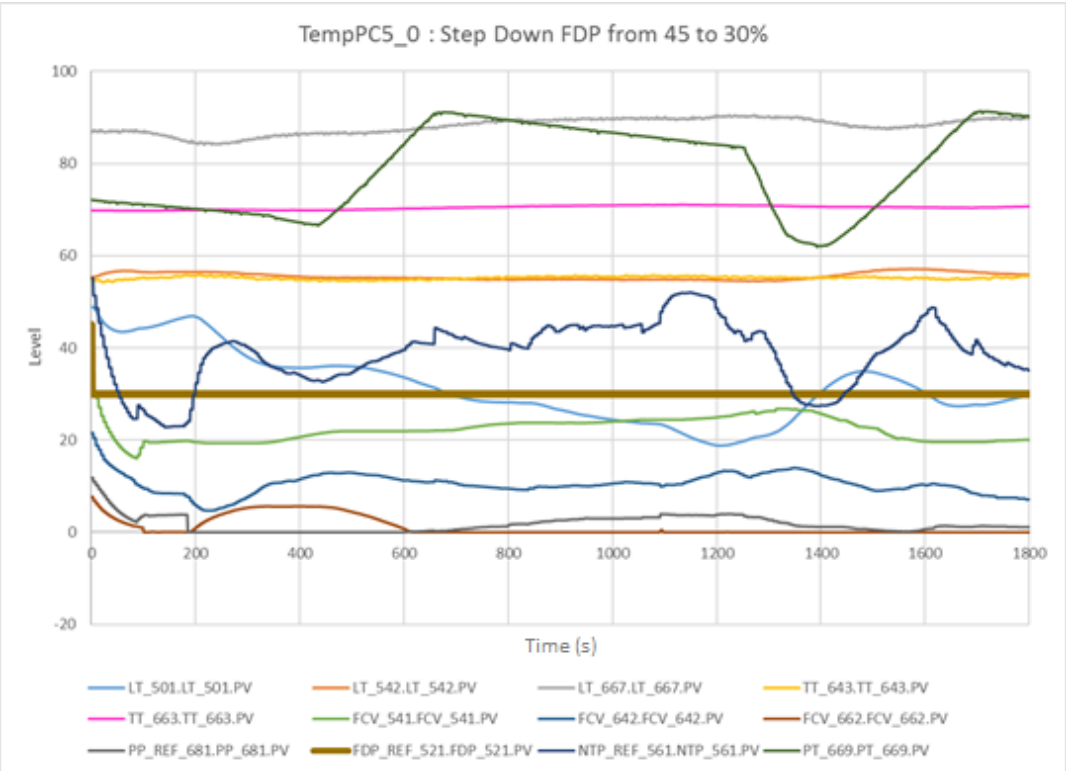


Figure 4.19 FDP stepped down by 15% saw feedforward action taken by the Profit Controller

The maximum errors recorded for the temperature CVs were:

- CSTR2 TT_643 (SP -0.75°C) <SP< (SP +0.87°C)
- CSTR3 TT_663 (SP -0.25°C) <SP< (SP +1.07°C)

The feedforward action of the multivariable controller is more clear in the combined step tests shown in Figure 4.20. At 2000s all MVs increase to account for the FDP step, and for example, CSTR3 level is kept within the 80 – 90% hard limits throughout the tests. The characteristic trend of the steam pressure is visible in the NTP and both steam valves throughout the test as these MVs controlled the temperature CVs at a new steady states regardless of the flow rate from the FDP.

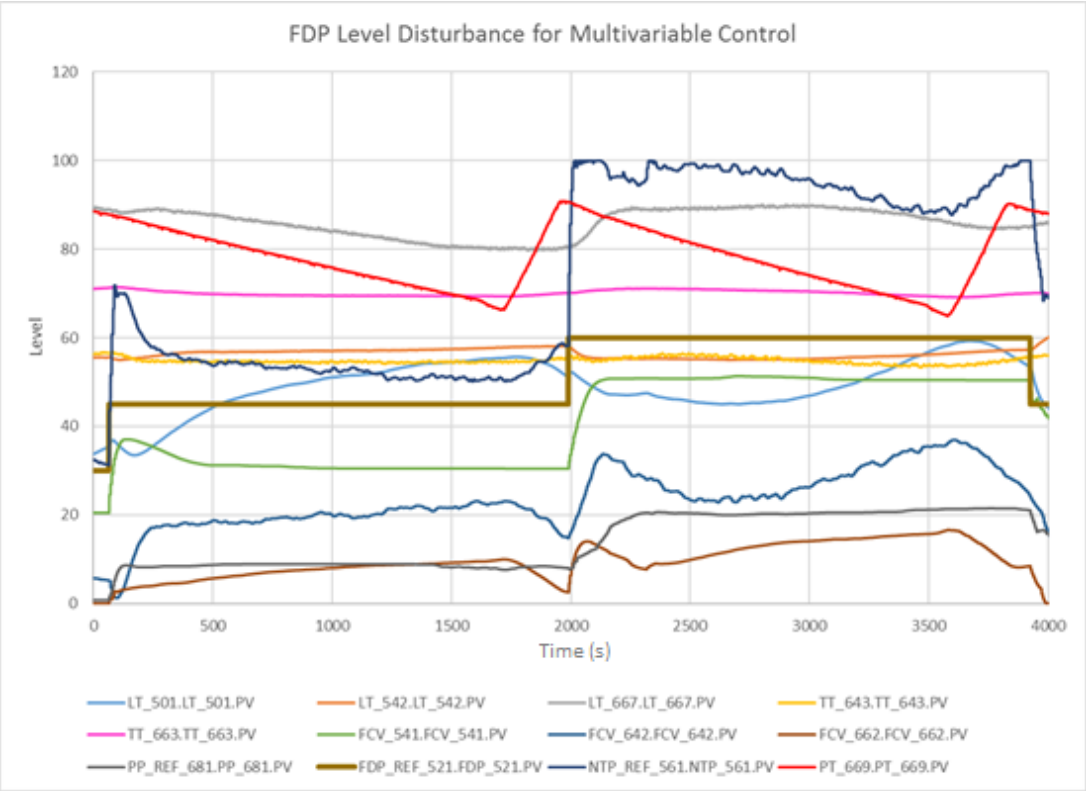


Figure 4.20 A series of FDP steps shows all valves open and pumps instantly speed up for an increase in FDP output.

The Lamella disturbance for Multivariable Control shows the controller ignoring level disturbances in the NT while the level remains within the range limits (Figure 4.21).

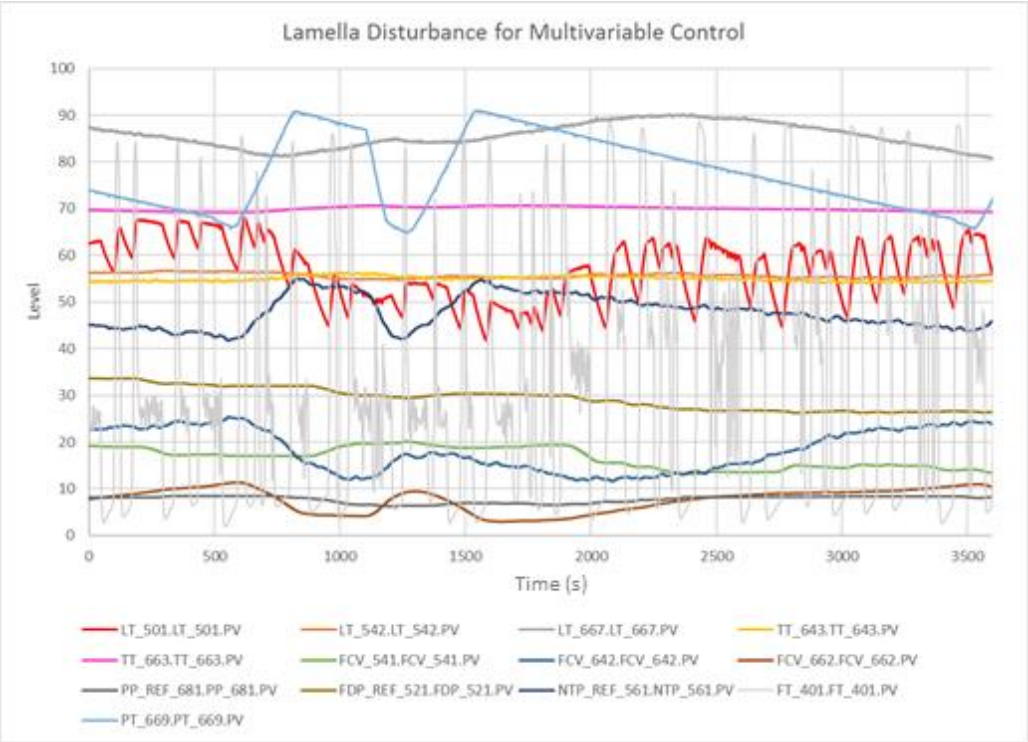


Figure 4.21 Lamella disturbance for multivariable control

The FDP was handed back to the Profit Controller so all pumps and all valves were available to it for multivariable control. The FDP flow rate was steadily reduced over the first hour to compensate for the increased flow into the Needle Tank. The Needle Tank level was allowed to fluctuate with the Lamella disturbance once the level was back within range. The NTP was still used to control temperatures in the CSTRs as they had higher priority than the tank levels.

The results of the 30-minute Lamella disturbance test are shown in more detail in Figure 4.22 below. The red line shows the average level in the Needle Tank was brought down below the upper high limit by reducing the FDP speed. The CSTRs did not receive the same erratic flow from the NTP as they did for the aggressive PI controller that tried to hold the Needle Tank level to a set point instead of a range.

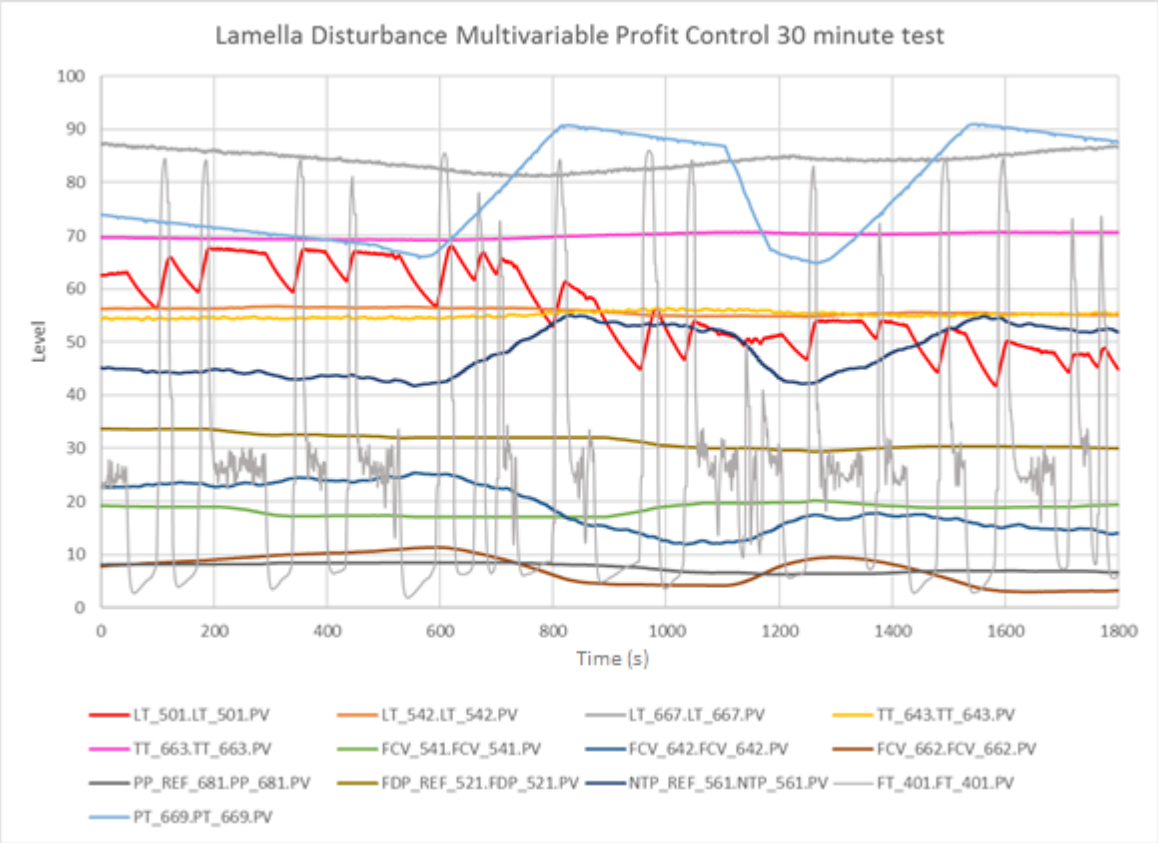


Figure 4.22 Profit Control with Lamella Tank disturbance flow

The maximum errors recorded for the temperature CVs were:

- CSTR2 TT_643 (SP -0.9°C) <SP< (SP +1.34°C)
- CSTR3 TT_663 (SP -0.81°C) <SP< (SP +0.67°C)

The results of the recycle disturbance test are provided in Figure 4.23. A hard lower limit had been set on the FDP because this MV cannot be operated as speed below 20%. A low flow interlock on downstream from FDP will switch the pump for OP settings lower than 20%. Figure 4.23 shows that the Profit Controller had reduced the FDP to 20% to accommodate the maximum flow from the Lamella Tank, just as a human operator would have done. This meant the NTP had to be used to control the NT level as well as CSTR temperatures and the trade-off is seen in the temperature CVs. The NTP speed hovered around 80% to keep the NT level below the upper hard limit because the FDP MV was wound up. The steam pressure disturbance is more pronounced because the Needle Tank Pump could no longer provide effective feed forward action while still maintaining the NT level.

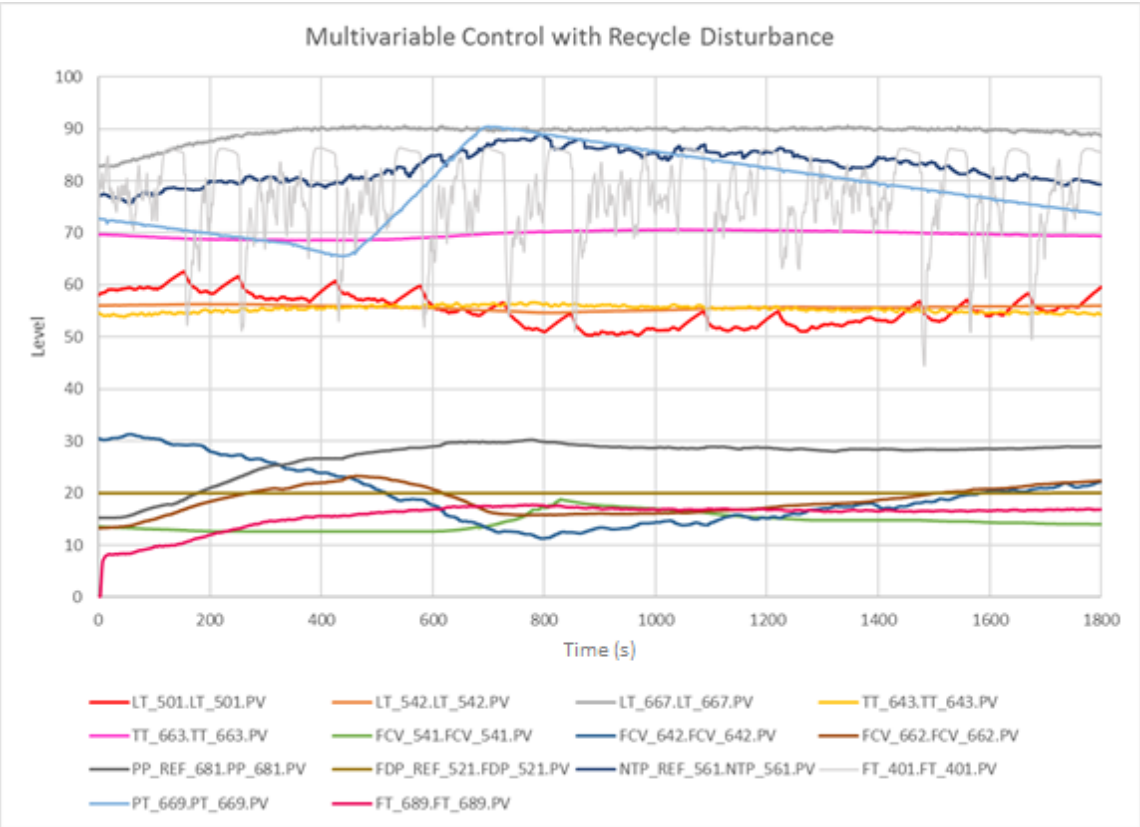


Figure 4.23 Multivariable Control with recycle and Lamella overflow disturbance

The maximum errors recorded for the temperature CVs were:

- CSTR2 TT_643 (SP -1.08°C) <SP< (SP +1.67°C)
- CSTR3 TT_663 (SP -1.4°C) <SP< (SP +0.69°C)

4.3 Profit Control in First Half of Pilot Plant

Multivariable Profit Control for tank levels in the first half of the Pilot Plant was very successful. Throughout the previous tests PI Control was used to control the first half of the plant and create the Lamella Disturbance. This required a lot of operator attention to manage the recycle stream between the BMT and the CUFT. The Cyclone Underflow pump seals were leaking and at the beginning of the project this pump was blocked completely. The pump output was unpredictable and could not always control the level in the CUFT for which it was the MV in PI control. Operator intervention was required to adjust the recycle stream feed rate FP_141 to avoid overflowing the tanks. The multivariable Profit Controller managed this half of the plant easily by manipulating all four pumps to control the two tank levels.

One Profit Controller called PPSTLevelPC1_1 was built to control levels in the first half of the plant. The Experion CMs were altered to allow Profit Suite to write to OP points. Profit Stepper was then employed to find classic integrator models between the four pump OP points and the tank level CVs as shown in Figure 4.24 and 4.25. These Rank 1 models enabled excellent level control.

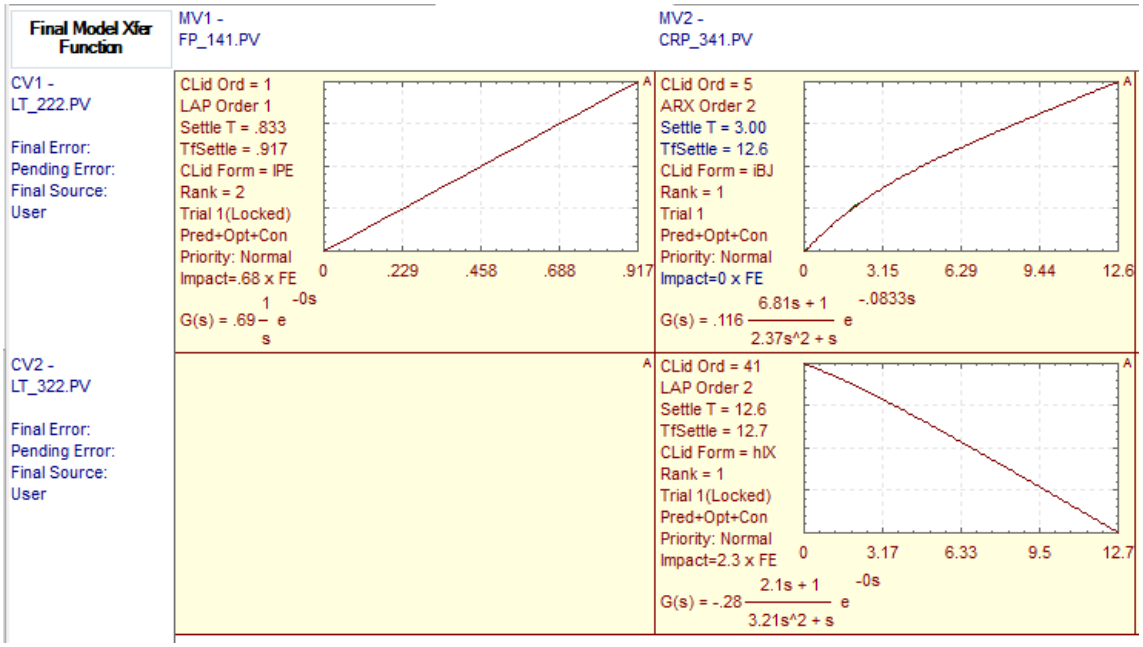


Figure 4.24 Rank 1 models for tank levels in first half of plant

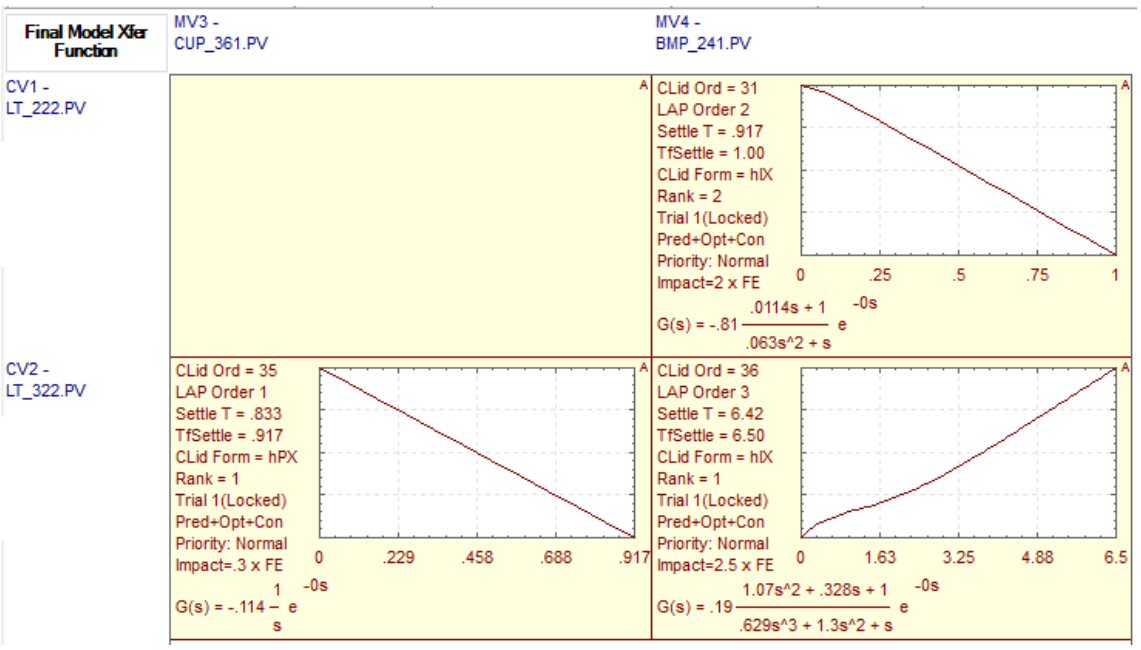


Figure 4.25 Rank 1 models for tank levels in first half of plant

The BMT was controlled within a range of hard limits of 55 to 65%. The CUFT had a CV range of hard limits of 65 to 75%. When controlled with PI controllers, the BMT set point was 60% and the CUFT set point was 70%. To mimic control to set point with Profit Control, the soft limits were brought in to 60% and 70% respectively and the optimizer was used to drive the plant towards these limits. The hard limit ranges allowed the controller the freedom to control all both CVs, and the soft limits were used in place of set points. Figure 4.26 shows the excellent performance of the MVC controlling both CVs at steady state. The tanks were held to their soft limits just as effectively as when held to SP under PI control. The added benefit was the Profit Controller would use the BMT pump and the Recycle pump in addition to the CUP to control the level in the CUFT. It would automatically reduce the feed rate from the Supply Tank to prevent overflowing the tank level CVs.

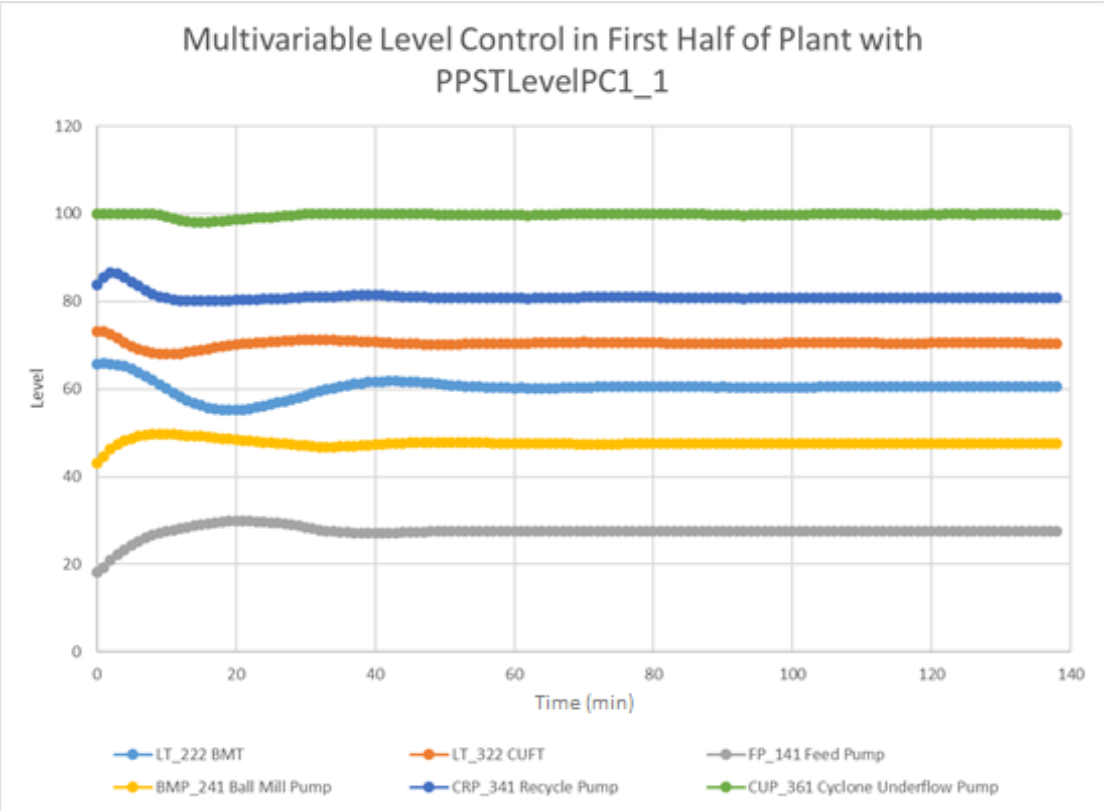


Figure 4.26 Multivariable control of BMT and CUFT levels with four pumps as MVs

The period from 0 – 100 minutes in Figure 4.27 shows the Profit Controller dealing with disturbances created by switching the tap that feeds the Lamella Tank on and off when preparing to connect both halves of the plant. The dotted lines show all MVs were used as the controller returned the CVs to their soft limits. It was much easier to bring the plant to steady state with the Profit Controller than to juggle the levels and recycle streams with PI control.

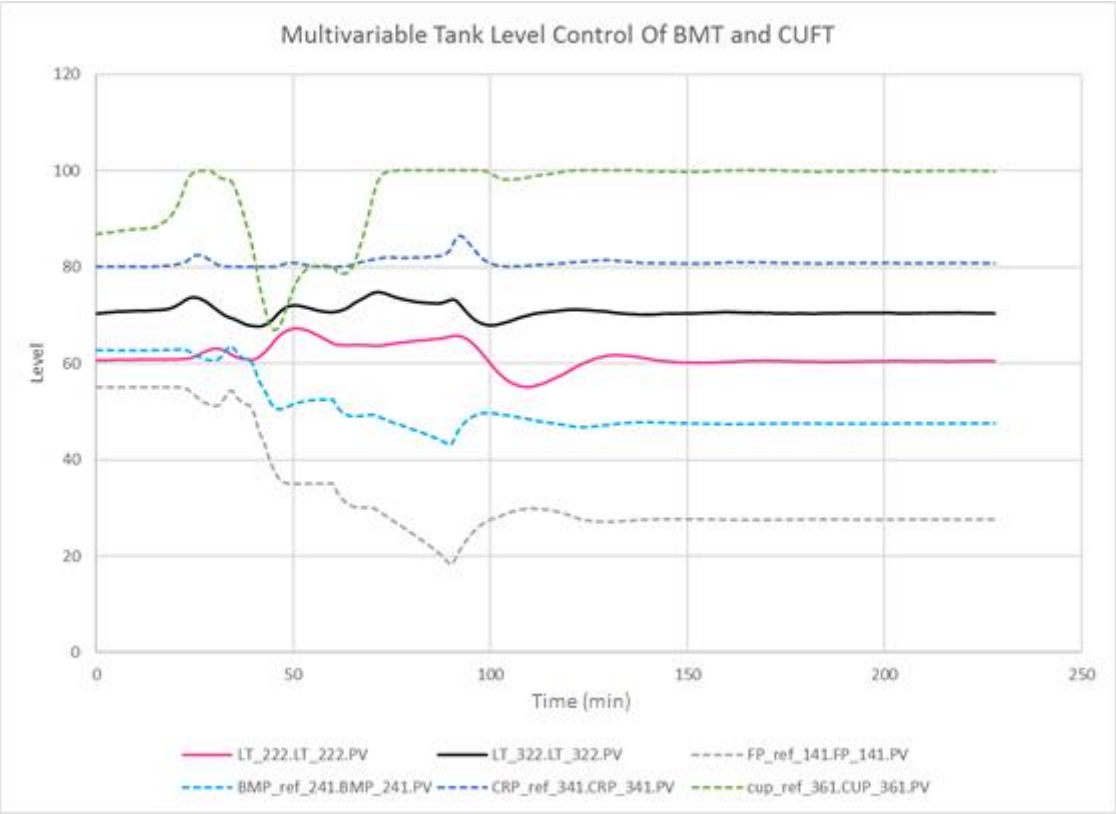


Figure 4.27 Profit Control dealt with disturbances effectively in the first half of the Pilot Plant

4.4 Analysis and Discussion

The performance of the PI Controllers exceeded that of the MVC controllers by the metrics given in Tables 4.4 to 4.7. These are measurements of the CV errors for the set point tracking and recycle disturbance tests presented in the results section. For example, the PI controller has an Integral Squared Error (ISE) value of 3.55 compared to MVC temperature-only controller TempPC2OP3 with 2006.6. However, all these measures show that when Profit Controller had the NTP available as an additional MV to combat the steam disturbance for temperature control with TempPC5_0, the MVC performance dramatically improved.

Table 4.4 CSTR3 Set point tracking performance metrics

Controller	ISE	IAE
PI Control	**3.5536**	**47.6**
TempPC2OP3	**2006.622**	**1052.145**
TempPC5_0	**82.03885**	**230.415**

Table 4.5 CSTR2 Set point tracking performance metrics

Controller	ISE	IAE
PI Control	**42.0909**	**157.87**
TempPC2OP3	**479.4855**	**534.06**
TempPC5_0	**300.1199**	**466.295**

Table 4.6 CSTR3 Lamella overflow plus recycle disturbance performance metrics

Controller	ISE	IAE
PI Control	**17.74225**	**90.055**
TempPC2OP3	**878.2088**	**660.13**
TempPC5_0	**499.1487**	**555.72**

Table 4.7 CSTR2 Lamella overflow plus recycle disturbance performance metrics

Controller	ISE	IAE
PI Control	**36.70555**	**147.205**
TempPC2OP3	**4181.342**	**1868.965**
TempPC5_0	**486.5604**	**554.3**

The statistical performance charts presented in Figure 4.28 to Figure 4.31 also confirm the PI Controller outperformed the MVC controllers. These chart the mean of groups of samples of each CV compared to the Grand Mean. The Sample Sub-groups plotted in these figures comprise ten CV samples per minute. They provide another indicator of how closely the CVs tracked their set points over time. The Experion PI controllers were barely affected by disturbances in comparison to the MVC controllers: the PI x-bar charts have very tight control limits even for the disturbance rejection tests. The MVC controllers are not under statistical control in any of the charts as their mean sample plots break all warning and control limits. The tighter control limits for the PI Controllers plot are well inside those of the MVC controllers.

The charts do confirm that multivariable control for both level and temperature was successful when using the OP as MV for all pumps and valves. This is seen in Figure 4.28 where the set point tracking of TmpPC5_0 shows significant statistical improvement over TempPC2OP3. This MVC Profit Controller's performance was much nearer to that of the PI Controller. The MVCs performed well for levels. The fluctuating steam pressure was not a problem for the Experion PI controllers. If PI control could be used for steam flow these results suggest the steam disturbance would be dealt with very effectively and enable better performance for MVC controller.

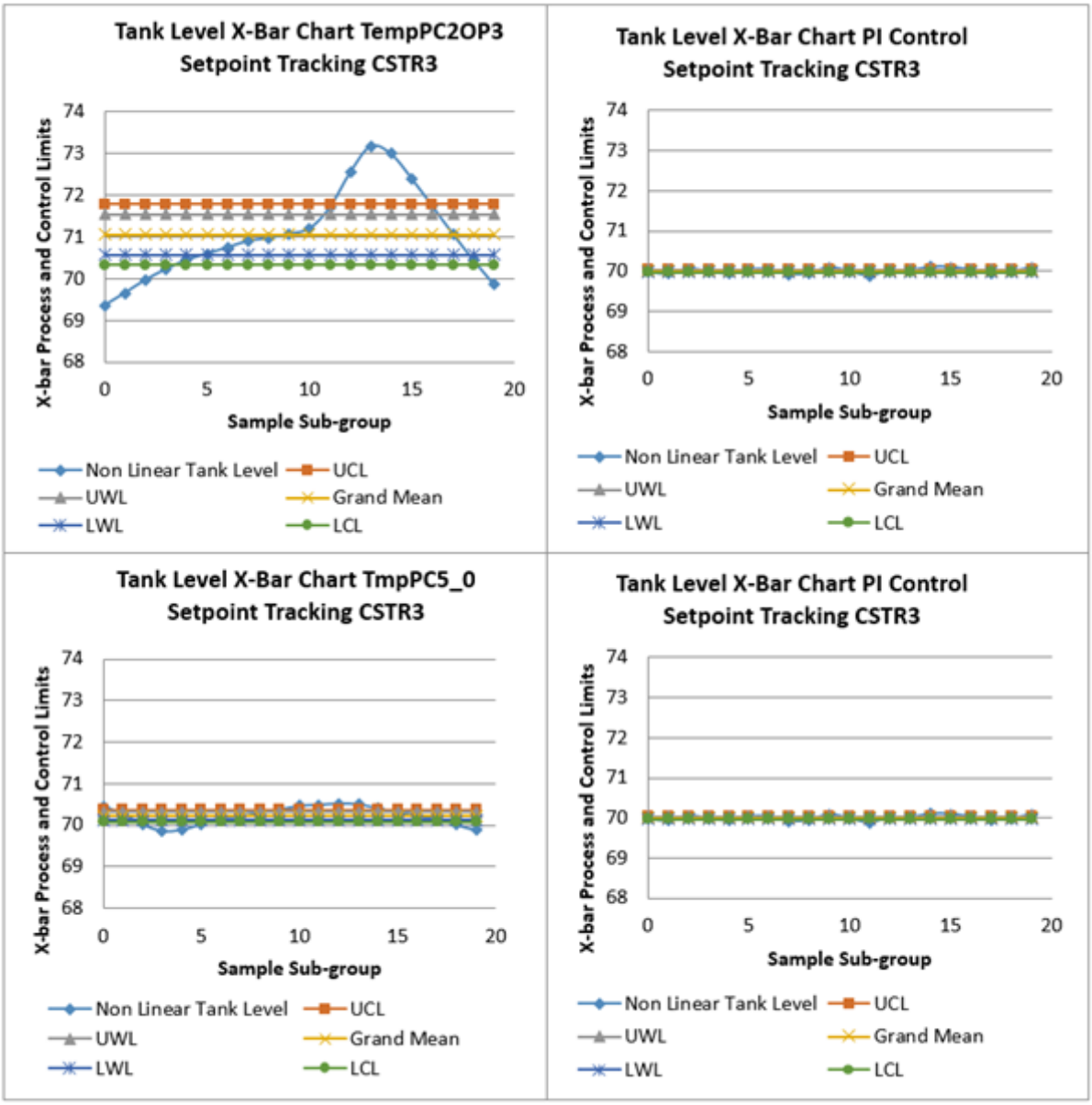


Figure 4.28 X-bar charts for CSTR3 set point tracking comparing MVC to PI Control

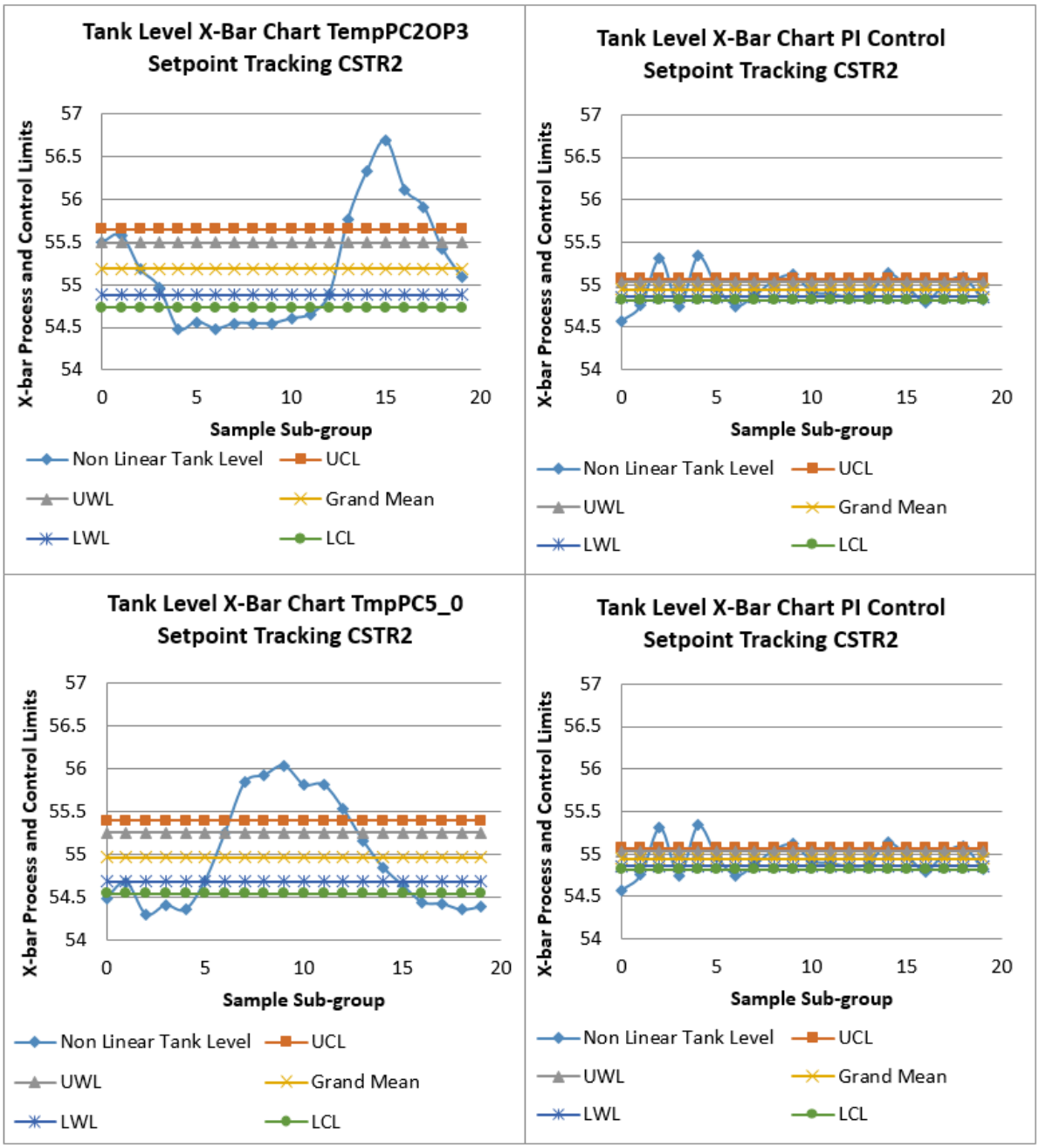


Figure 4.29 X-bar charts for CSTR2 set point tracking comparing MVC to PI Control

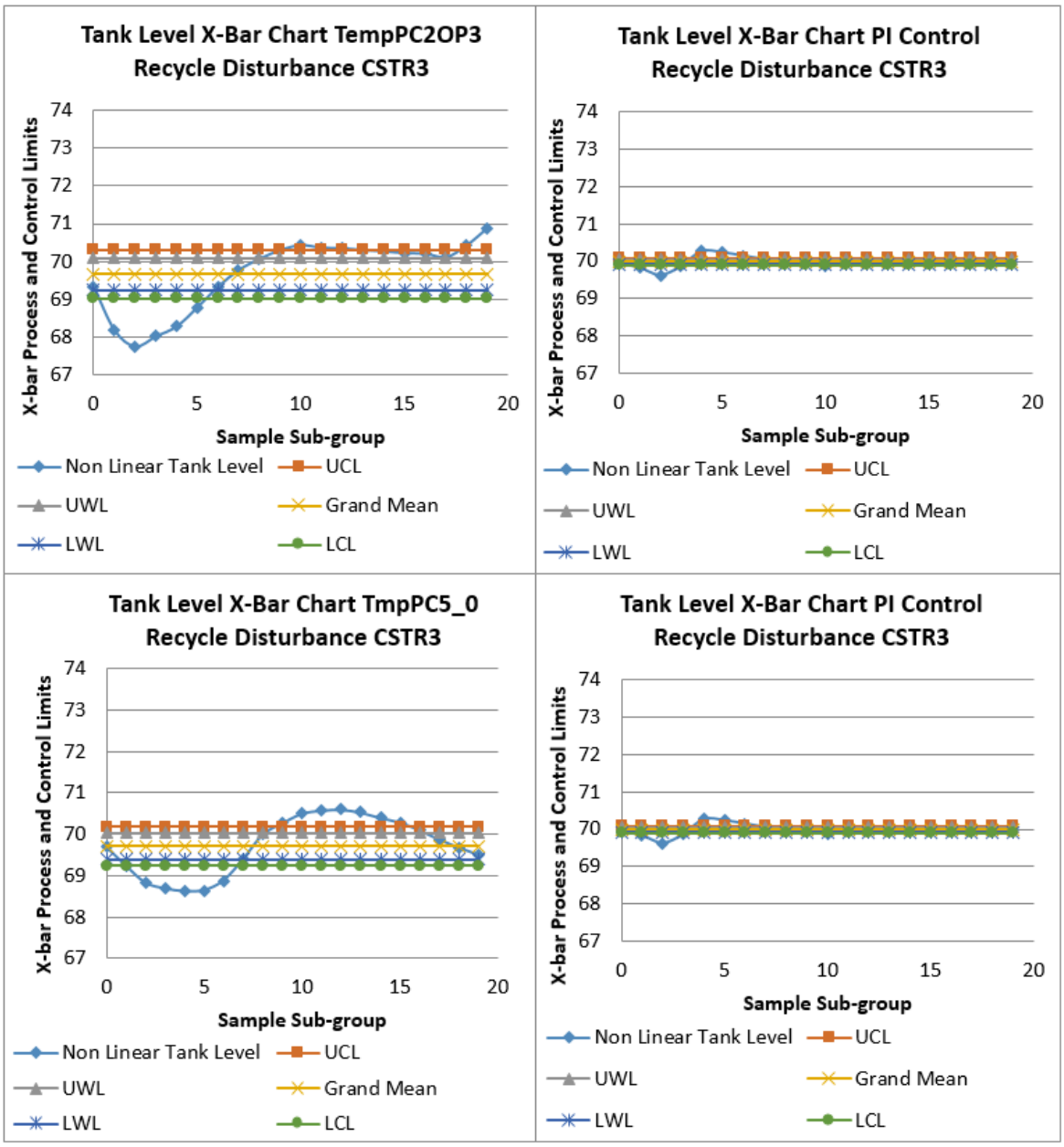


Figure 4.30 X-bar charts for CSTR3 recycle disturbance comparing MVC to PI Control

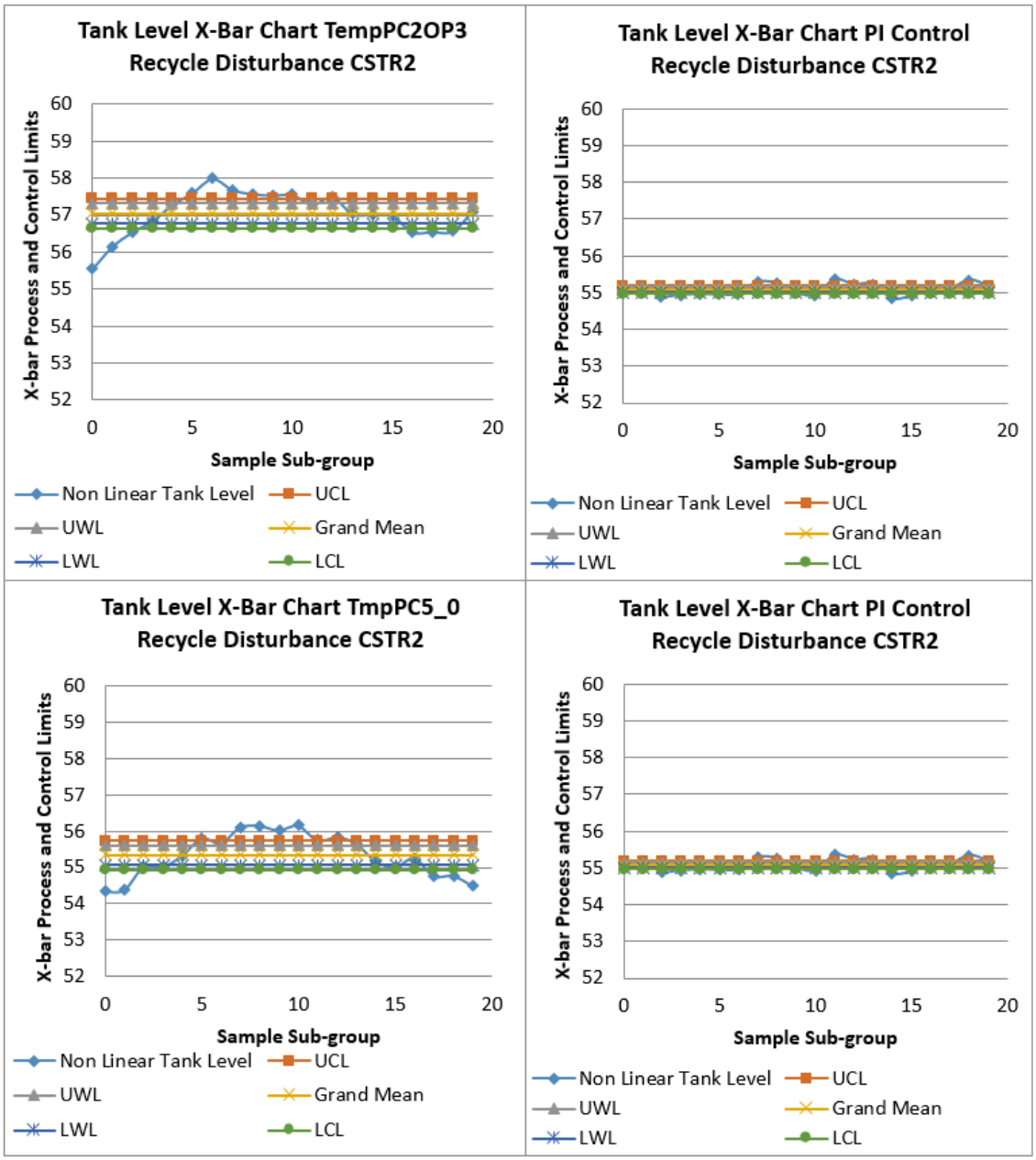


Figure 4.31 X-bar charts for CSTR2 recycle disturbance comparing MVC to PI Control

Chapter 5 Optimization

5.1 Optimizing First Half of Pilot Plant

Without optimization, the CVs of the Profit Controller were free to drift anywhere within their hard limits; the controller saw no error so made no control moves. Soft control limits for each CV were selected and the linear objective functions were used to drive the CVs to either the upper or lower of these limits. These soft limits targets were used successfully like set points in the first half of the plant. The hard limits were left at 5% either side of the desired CV targets for 10% of range control. The results in Figure 4.26 were the result of the following optimizations strategy:

- Objective function: Maximize supply from Feed Pump 141 to High Soft Limit of 30.5%
- Optimize Recycle pump to soft high limit 80.5%
- Optimize Ball Mill Tank to soft high limit 60.5%
- Optimize Cyclone Underflow Tank to soft high limit 70.5%

Table 5.1 shows the initial values for the CVs and MVs and their final values after 140 minutes of optimization. As Figures 4.26 and 4.27 show, optimization resulted in a very stable process. The feed pump speed at the Supply Tank was optimized after 63 minutes at 27.58%.

Table 5.1 Optimization results

	Hard Limits	Delta Soft Limits	Initial Value	Optimized Value at 140 minutes
Ball Mill Tank	55<CV<65	4.5%	65.69	60.57
Cyclone Underflow Tank	65<CV<75	4.5%	73.23	70.38
Feed Pump	8<MV<60	Delta High 29.5%	18.2	27.58
Ball Mill Pump	10<MV<100	2%	43.17	47.62
Recycle Pump	5<MV<100	Delta High 19.5%	83.75	80.75
Cyclone Underflow Pump	10<MV<100	2%	100	99.79

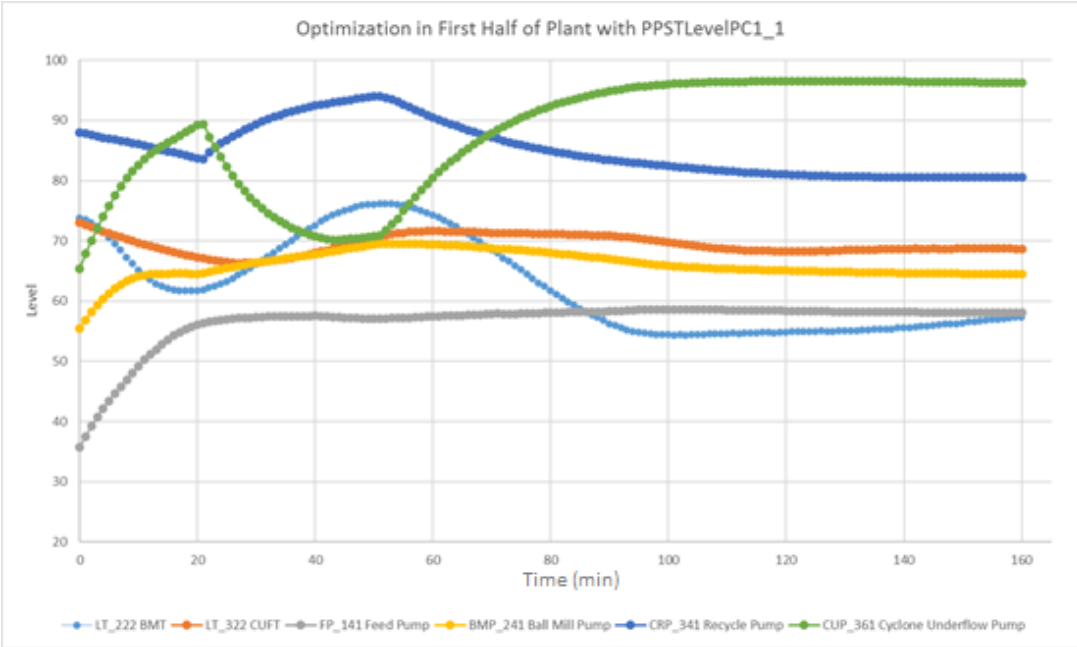


Figure 5.1 First maximizing then minimizing the recycle stream for different steady states

Figure 5.1 shows how easy it was to choose steady states using the Profit Controller Optimizer. The recycle stream was maximised from t=20min to t=50min. The recycle pump speed was then minimized to its soft limit from t=50min to t=160min at which point the tank levels CVs were at steady state.

5.2 Optimizing Second Half of Pilot Plant

The objective function for optimization of the second half of the Pilot Plant was to maximize production and minimize costs. Costs were assigned to each steam valve were based on empirically collected data of the steam flow from each valve. These cost coefficients were entered into the optimizer via PSOS and the speed of the Product Pump was optimized to the upper soft limit. Table 5.2 shows negative coefficients were used to maximize variables.

Table 5.2 Linear Optimization coefficients for the second half of Pilot Plant

Variable		Cost Coefficient
Product Pump	PP_681	-4
Needle Tank Pump	NTP_561	0
Steam Valve CSTR2	FCV_642	0.84
Steam Valve CSTR3	FCV_662	0.74
CSTR2 Temperature	LT_642	0
CSTR3 Temperature	LT_662	-1.5

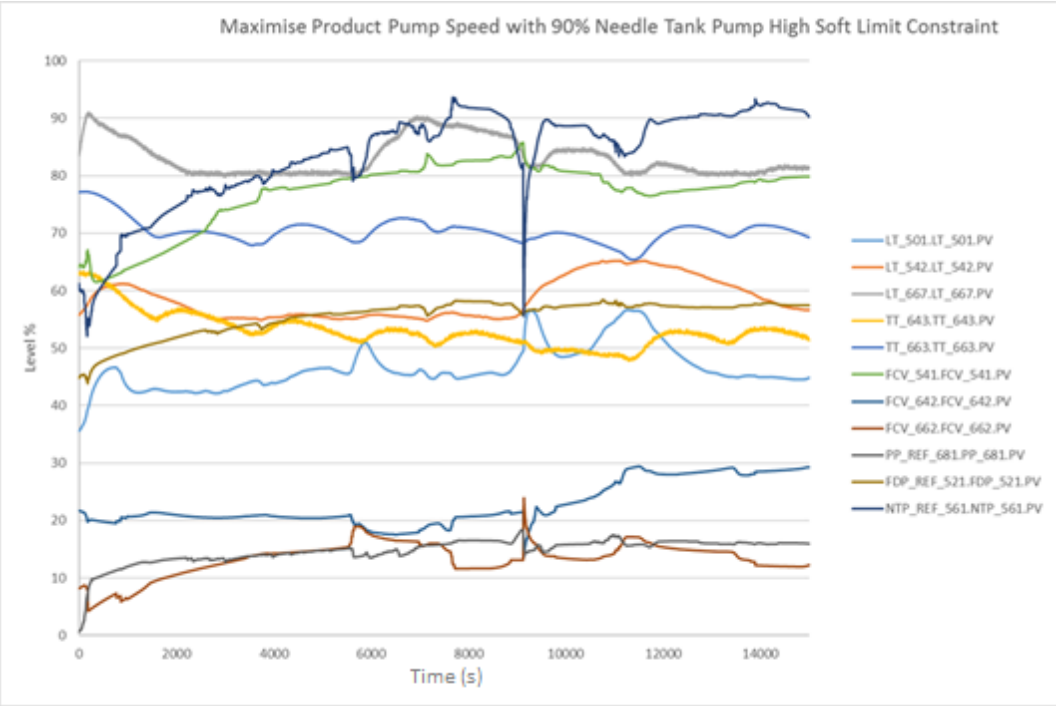


Figure 5.2 Maximizing production in the second half of Pilot Plant with the Optimizer

With an optimization speed factor of 1, the Profit Controller took three hours to maximize the production rate. The NTP is the flow constraint in the plant because of its small capacity compared to the other pumps. Figure 5.2 shows that maximum production is reached once the NTP was steadily tracking its high soft limit of 90%.

5.2.1 Optimizing as Plant Conditions Change

The lamella disturbance was introduced while the optimizer was running to observe the behaviour of the controller. Figure 5.3 shows the Lamella Overflow occurred at t=660s with a sharp rise in the NT level. The Profit Controller reduced flow from the Raw Water Valve and Flow Disturbance Pump to compensate while it continued to optimize for maximum Product Pump flow. The NTP had returned to the high soft limit of 90% and the whole plant moved to a new steady state by t=4000s.

The same disturbance conducted with PI control shows its aggressive MV behaviour for comparison in Figure 5.4. For Profit Control, the Needle Tank Pump was returned for use as a MV for temperatures to counteract steam pressure disturbance once it was tracking the soft limit. Towards the end of the optimization test the FDP had an OP of 46.74. It is unlikely that a human operator would know that the maximum production rate would be achieved with the FDP at 46.74 when the Supply Tank feed FP_141 in the first half of the plant was maximised at 27.58. The FDP OP was instead manually dropped to 42 in PI control. This is the benefit of multivariable control in that the Profit Controller continually drove the process towards maximum production as plant conditions changed. An experienced Pilot Plant operator would need to choose much lower FDP settings and constantly monitor the level of the NT to prevent it overflowing. The Profit Controller automatically adjusted the FDP at 5 second intervals.

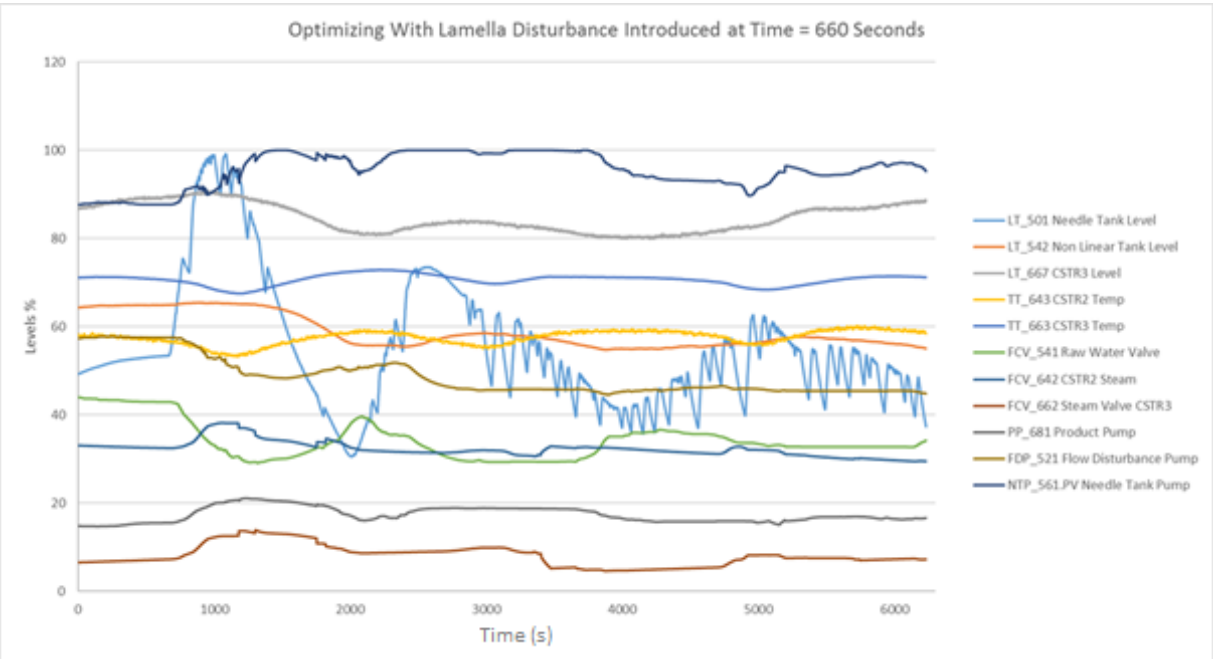


Figure 5.3 Optimization with changing plant conditions

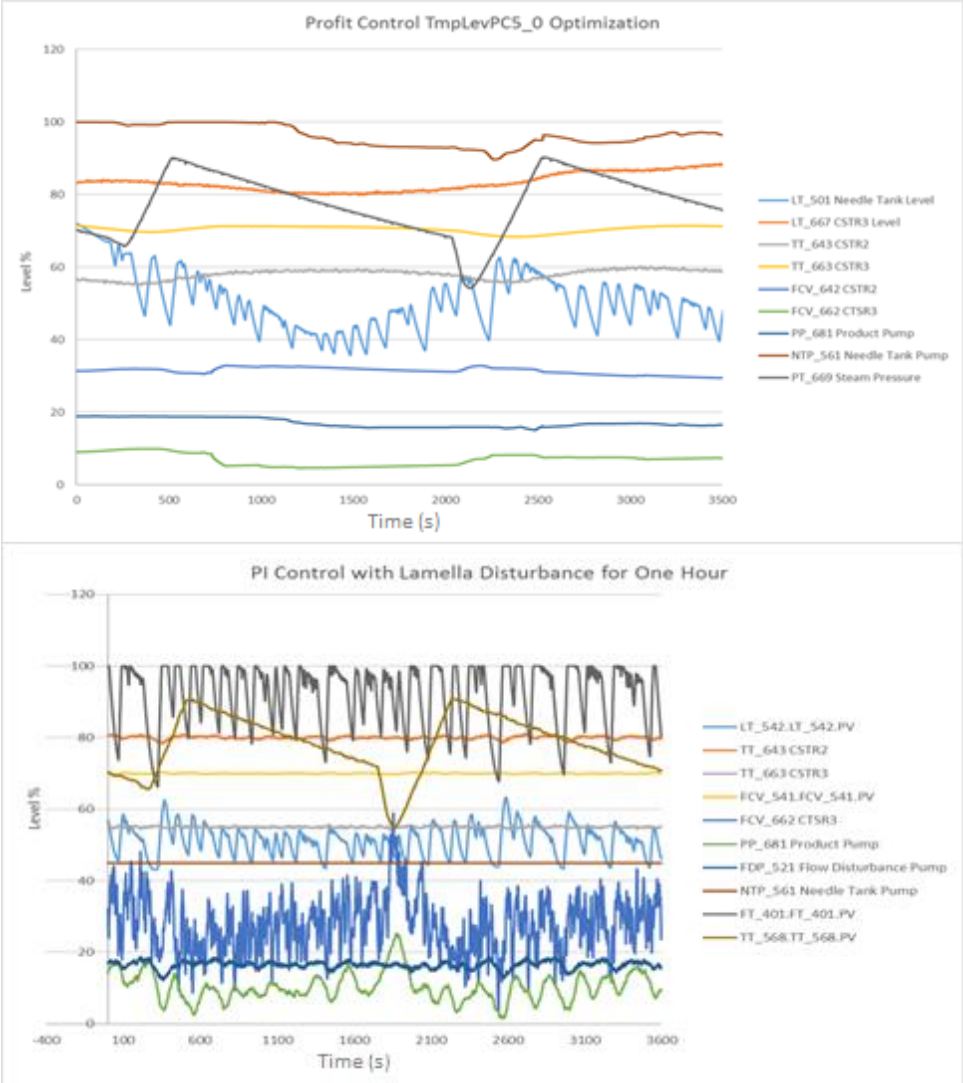


Figure 5.4 PI control compared to optimized MVC control with Lamella disturbance

Chapter 6 Proposed Future Work

The control structure arrived at in this project is depicted in Figure 6.1. The MVC used the OP point of Temperature PI controllers as the MV to control CSTR temperature CVs. The Profit Controller is a slow, steady state controller that is supposed to execute infrequently. The steam pressure DV models found were not accurate and caused the MVC to overcompensate with excessive control action. These two factors prevented good temperature control with the MVC. However, the fast executing Experion PI Regulatory controllers were excellent at dealing with the steam pressure disturbance.

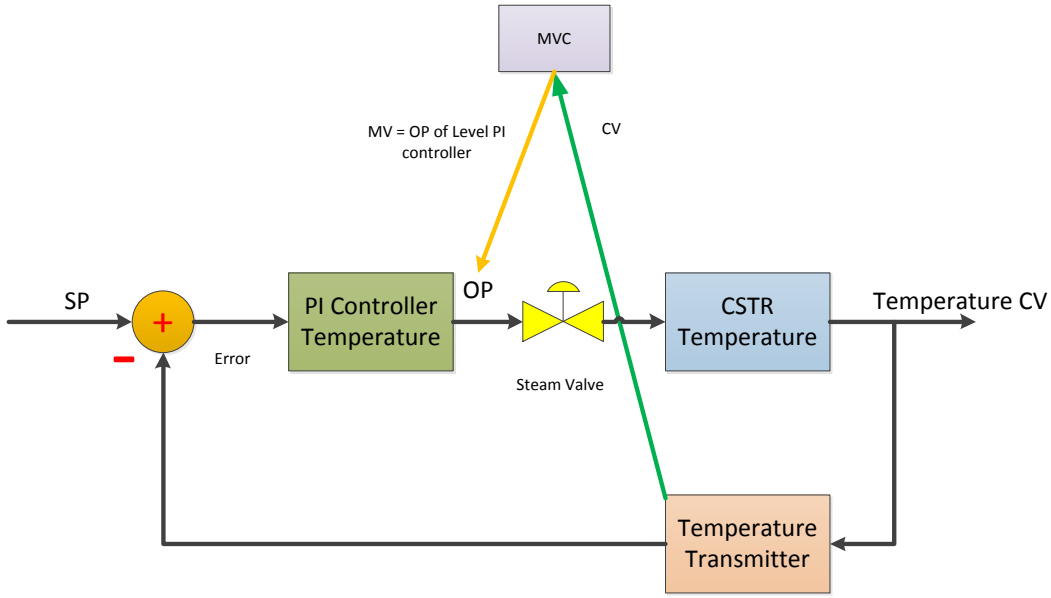


Figure 6.1 Current BLC control structure

The proposed solution is depicted in Figure 6.2. Steam flow transmitters should be installed to measure the flow of steam from each valve. (These could possibly be wireless instruments given the University's recent installation of a Honeywell OneWireless network.) This flow transmitter would permit PI steam flow control which under normal conditions would be the inner loop of a cascaded temperature PI controller. The results of the PI experiments in this project suggest that excellent PI control of steam flow would be obtained and the steam disturbance would be eliminated from the CSTR temperature sub-process. The Profit Control MVC would connect to the SP of the inner steam flow PI controller and use it to accurately manipulate the flowrate of steam into the CSTRs. When the MVC is terminated, BLC are already available in PSRS that shed the inner cascade flow loop back to the outer temperature PI controller.

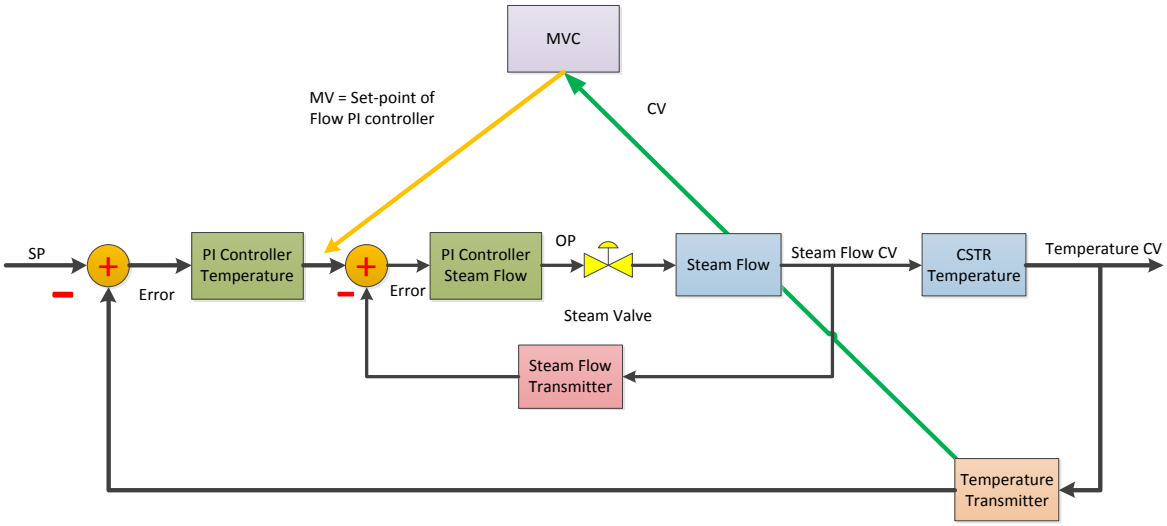


Figure 6.2 Proposed BLC control structure. Profit Control uses SP of inner flow PI Controller as MV and sheds to the outer Level PI controller

Flow transmitters are installed around some pumps which will allow cascaded PI flow controllers to be built in Experion for tank levels. MVC level controllers could then be built to trial the use of PI water flow SPs as MVs instead of pump speeds. This requires the PIDA pin connections to be rewired in the Experion CMs which will affect the automated maintenance and demonstration programs.

Until Steam PI flow control is possible, the only option is to continue using OP points for multivariable temperature control. Accurate Rank 1 models for the steam pressure disturbance will be required. Extensive modelling of the effect of the steam flow on the CSTR temperature CVs should be conducted and tested. The Lamella overflow disturbance could be modelled against the Needle Tank level with the Profit Stepper to enable one Profit Controller to optimize flow through the entire plant. This may be possible by stepping the BMT underflow pump and measuring the mean level of the Needle Tank; results showed higher Lamella flow rates produced less erratic flows.

Another proposal is to remove the need to use Experion CM to enable profit control by using the Station faceplates to change PID controller modes. A check box on Station page similar to the Operation button might be used to enable the Profit Controller flags in Experion instead of requiring future students to have access to the Pilot Plant code to use Profit Suite.

Chapter 7 Conclusion

This thesis represents the first implementation of Profit Suite into Murdoch University's Pilot Plant. The main objective of the project was to build and test multivariable controllers on the live plant and this was achieved. This required the commissioning and tuning of Experion PI controllers which was completed as part of this project. The performance of these PI controllers was excellent. Multivariable controllers were built that controlled temperatures and levels in both halves of the plant.

The results from testing Multivariable Profit Control using PI set points as MVs proved that the MVC must manipulate the fundamental inputs of a process. The models that the MVC required were between steam flow and temperature, and water flow and tank level. This was not possible with the existing instrumentation and Experion code in the Pilot Plant, so OP points were used as MVs for Profit Control. Code was developed that enabled Profit Controllers to write to the OP points of the PID level and temperature controllers in the Experion CM.

The level controllers that were built with OP points as MVs performed well. This is because it was possible to find accurate models between the pump speeds and the tank levels: the pump speed is directly related to flow. In particular, the MVC for the first half of the Pilot Plant had excellent control. The Profit Controller managed two levels and recycle streams using the four pumps as MVs. It was easy to choose soft limits targets and watch as the MVC moved the plant to new steady states with good level control. The control was better than for PI which can allow tanks to overflow if a human operator does not constantly monitor each individual CV. The first half of the plant was managed by the Profit Controller which allowed the operator to focus on controlling the second half of the plant.

The PI Controllers were better than the MVCs for temperature control. The fast executing Experion PI controllers dealt with the steam pressure disturbance such that there was no effect on the temperature CVs. This disturbance was only visible in the MV action in response to the steam trend. The models found for steam pressure were not accurate enough for the MVCs to perform as well as PI control and the CVs were affected by both the disturbance and the overcompensation of the MVC controllers. The MVC for levels and temperature had more MVs available to it and performed much better than the temperature-only MVC. Multivariable control action was clearly visible when the NTP and both steam valves were used to control two temperature CVs. All three MVs tracked the steam pressure disturbance trend and temperature control improved measurably.

Optimization strategies to maximise Feed Rate and Production rate while minimizing steam costs were successful. Because MVC level control was good, the optimizer could be used to drive both halves of the plant in any desired direction. Various linear objective functions were implemented to maximise or minimize flows and levels with success.

For the performance of the temperature Profit Controllers to exceed that of PI Control in future, changes to the control system of the Pilot Plant are required. Cascade temperature control is recommended. The slow acting MVC should manipulate the SP of an inner PI flow controller for precise steam flow. This fast executing PI controller would eliminate the steam disturbance allowing Profit Controllers to be built which do not require models for steam pressure.

References

Baker, Peter. 2008. "Development and evaluation of control performance assessment indices for Alcoa's advanced process control applications."

Den Hond, Roelof, Iwan Hiralal, and Ab Rijkeboer. 2016. "Alumina Yield in the Bayer Process Past, Present and Prospects." In *Essential Readings in Light Metals: Volume 1 Alumina and Bauxite*, edited by Don Donaldson and Benny E. Raahauge, 528-533. Cham: Springer International Publishing.

Dring, Joshua. 2012. "Pilot plant automation." Murdoch University.

Fayruzov, D. Kh, Yu N. Bel'kov, D. V. Kneller, and A. Yu Torgashov. 2017. "Advanced process control system for a crude distillation unit. A case study." *Automation and Remote Control* 78 (2):357-367. doi: 10.1134/S0005117917020138.

Godfrey, Simon. 2016. "Honeywell Experion System: Configuration, simulation and process control software interoperability." Murdoch University.

Honeywell International. 2015. "Plant Test Design." In: Honeywell International,.

Honeywell International. 2016a. Control Interval. In *Profit Suite Concepts*. USA: Honywell International.

Honeywell International. 2016b. Profit Controller Designer Help. Honeywell International.

Honeywell International. 2016c. Robust Control. In *Profit Suite Concepts*. USA: Honeywell International,.

Honeywell Process Solutions. 2013. *Experion Control Builder Components Theory Release 430*: Honeywell Process Solutions.

Honeywell Process Solutions. 2014. *Experion PKS Control Building User's Guide Release 430*: Honeywell Process Solutions,.

Honeywell Process Solutions. 2015a. Advanced Control Overview. Honeywell

Honeywell Process Solutions. 2015b. Profit Controller. Honeywell.

Honeywell Process Solutions. 2015c. *Profit Controller Introduction*: Honeywell Process Solutions,.

Hopkinson, Earl. 2010. "Murdoch University pilot plant advanced control technology upgrade." Murdoch University.

Howes, Steve, Janarde Le Pore, Ivan Mohler, and Nenad Bolf. 2014. "IMPLEMENTING ADVANCED PROCESS CONTROL FOR REFINERIES AND CHEMICAL PLANTS." *Goriva i Maziva* 53 (2):97.

Lumley, R. 2010. *Fundamentals of Aluminium Metallurgy: Production, Processing and Applications*: Elsevier Science.

Mackay, Shane. 2012. "Development of Murdoch University pilot plant maintenance & demonstration programs." Murdoch University.

Meiri, Timothy. 2015. "Experion simulation and pilot plant maintenance." Murdoch University.

Ogunnaike, Babatunde A., and W. Harmon Ray. 1994. *Process dynamics, modeling, and control*. New York: Oxford University Press.

Punch, Alan. 2009. "Implementation of advanced control technology in the Murdoch University Pilot Plant." Murdoch University.

Ramboll Environ Inc. 2005. *Wagerup Refinery Unit Three expansion : environmental review and management program / McAuliffe & Associates, Environ [for] Alcoa*. Edited by Australia Alcoa World Alumina, Authority Western Australia. Environmental Protection, Australia Environ, McAuliffe, Associates and Csiro Atmospheric Research. [Applecross, W.A: Alcoa World Alumina Australia].

Schmitz, C. 2006. *Handbook of Aluminium Recycling*: Vulkan-Verlag.

Seborg, Dale E. 2011. *Process dynamics and control*. Vol. 3rd, international student version. Hoboken, N.J: John Wiley & Sons.

Smith, Cecil L. 2010. *Advanced process control: beyond single-loop control*. Hoboken, N.J: Wiley.

Ter Weer, Peter-Hans. 2014. "Relationship Between Liquor Yield, Plant Capacity Increases, and Energy Savings in Alumina Refining." *JOM* 66 (9):1939-1943. doi: 10.1007/s11837-014-1069-x.

Totten, G.E., and D.S. MacKenzie. 2003. *Handbook of Aluminum: Volume 2: Alloy Production and Materials Manufacturing*: CRC Press.

Vu, L.T.T., P.A. Bahri, and G. Cole. 2010. "The use of the pilot plant facility in teaching process control engineering at Murdoch University." *Chemeca 2010: Engineering at the Edge*.

Wheat, Paul, and Warunthon Poonlua. 2017a. *Pilot Plant Case Study*, *Murdoch Internal Report,*: School of Engineering and Information Technology - Murdoch University,.

Wheat, Paul, and Warunthon Poonlua. 2017b. *Pilot Plant Case Study Part 2, Murdoch Internal Report,*: School of Engineering and Information Technology - Murdoch University,.

Wiberg, E., and N. Wiberg. 2001. *Inorganic Chemistry*: Academic Press.

Appendix A Example Control Module prior to OP code changes

Figure 9.1 shows the code in CM FCV_541 as found prior to this project. Troubleshooting is hampered because most FB have meaningless names and there are no code comments describing how each CM operates. This CM code was typical of the way the OP values are pulled from the Station page input and fed back into an input pin on the PID block. This prevents Profit Controller writing to the OP point and using it as an MV.

A FB named FCV_541 is used to control the valve directly from an input field on a Station page. When either the Station user or an Excel spreadsheet controller changes the value of FCV_541, this value is written directly to the Analogue Output for the valve. All the upstream safety interlocks are ignored for one scan cycle. The value of FCV_541 is then fed back around via N11_307 then SWITCHB to an input pin on the PIDA block to store the value of the OP for subsequent scan cycles.

Before code changes implemented as part of this project, Profit Controllers could not write to the OP point because the OP value was always fixed with the value stored in FCV_541 function block.

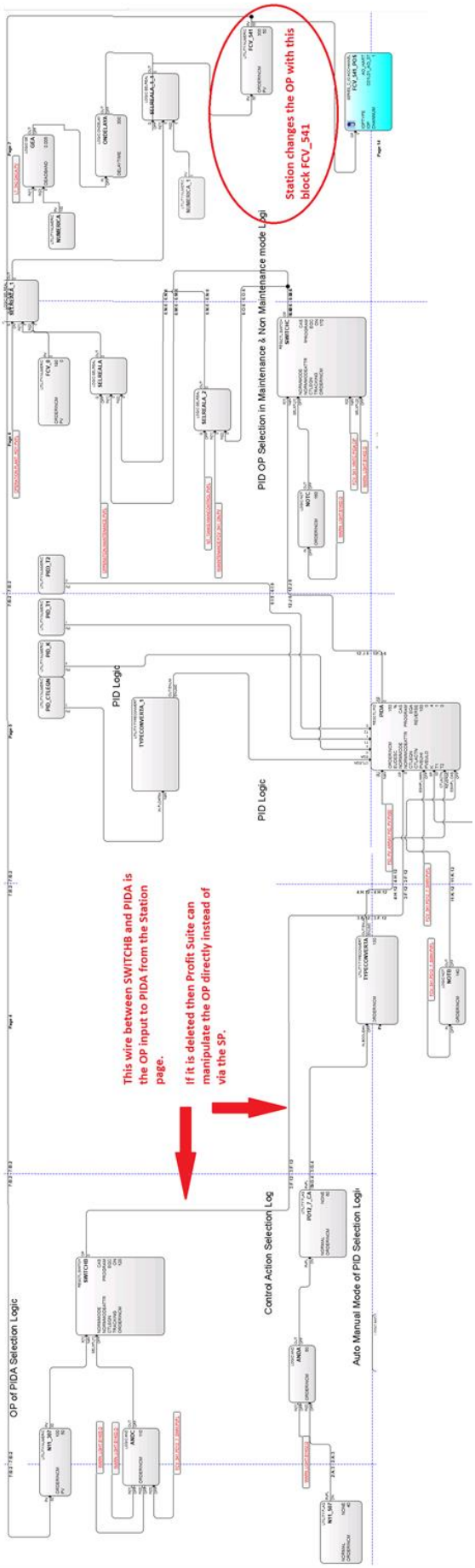


Figure 7.1 Example of PIDA block taking OP value from upstream block

Appendix B Overview diagrams

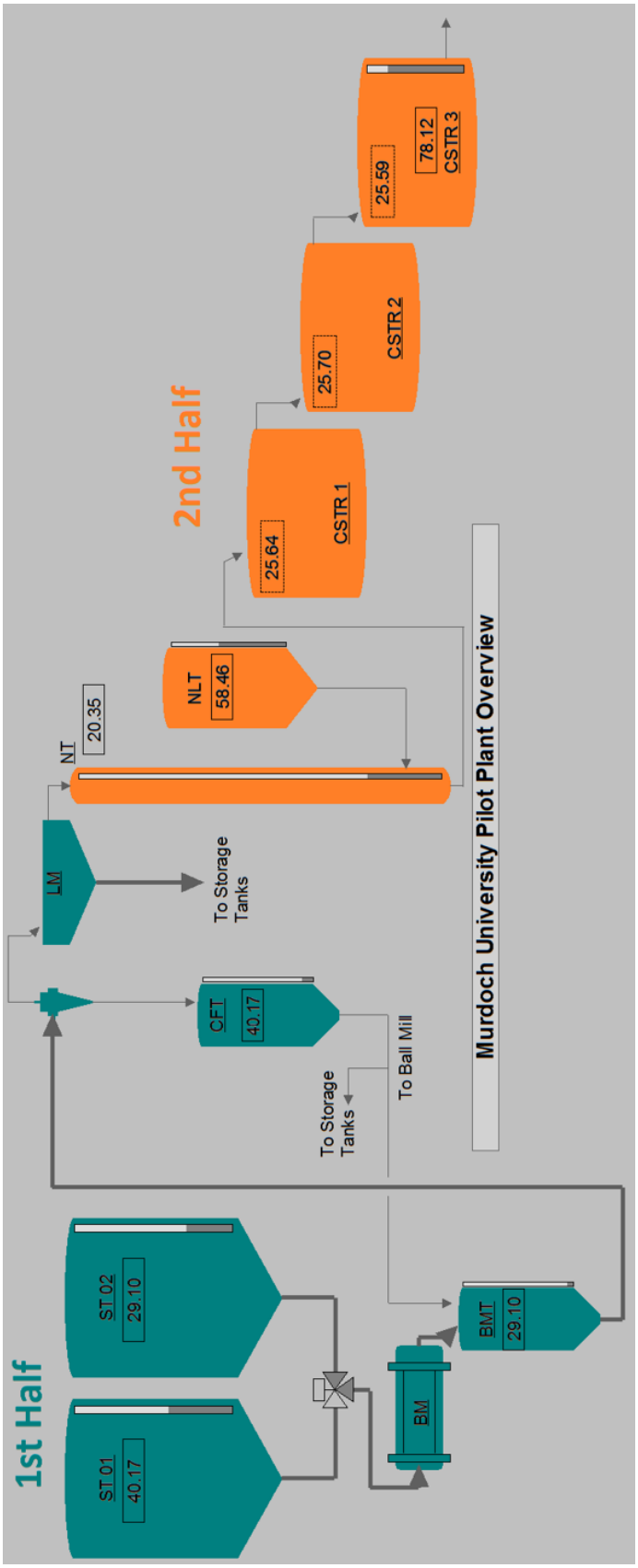


Figure 7.2 Pilot Plant overview illustrating both halves of plant

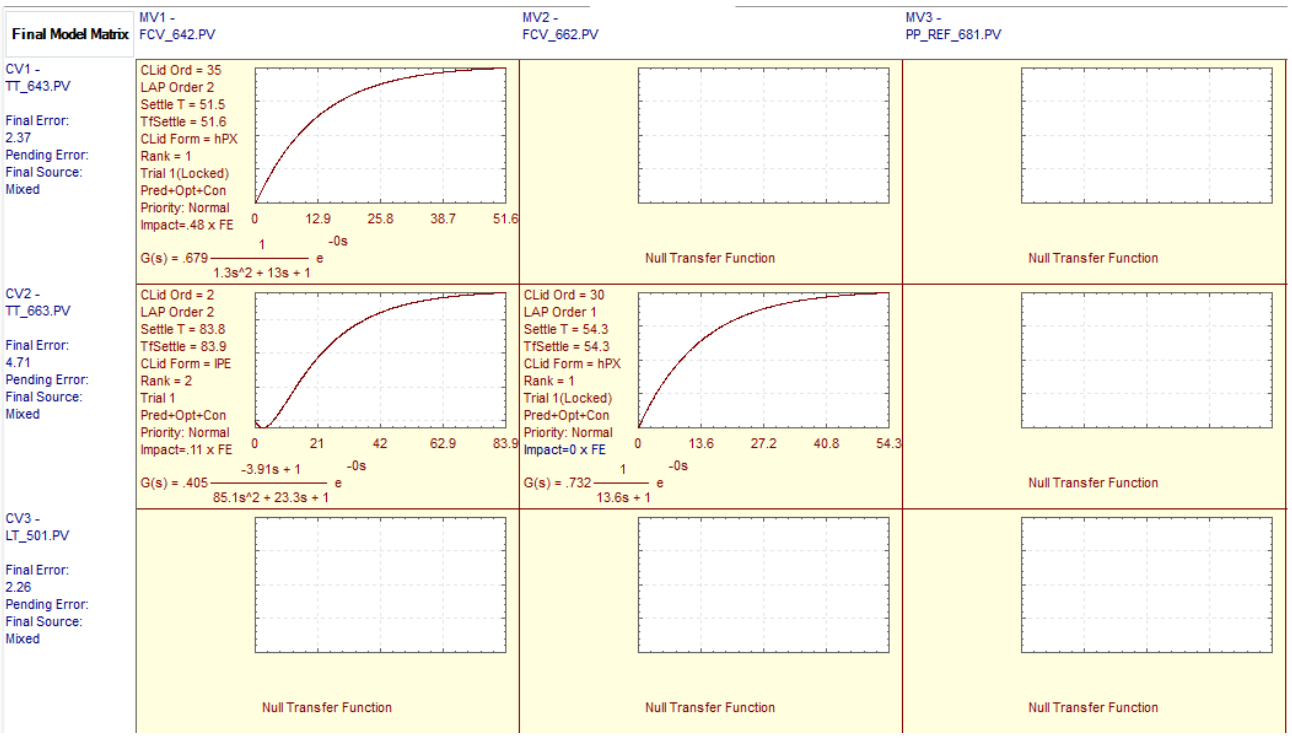


Figure 7.3 TmpPC5_0 models

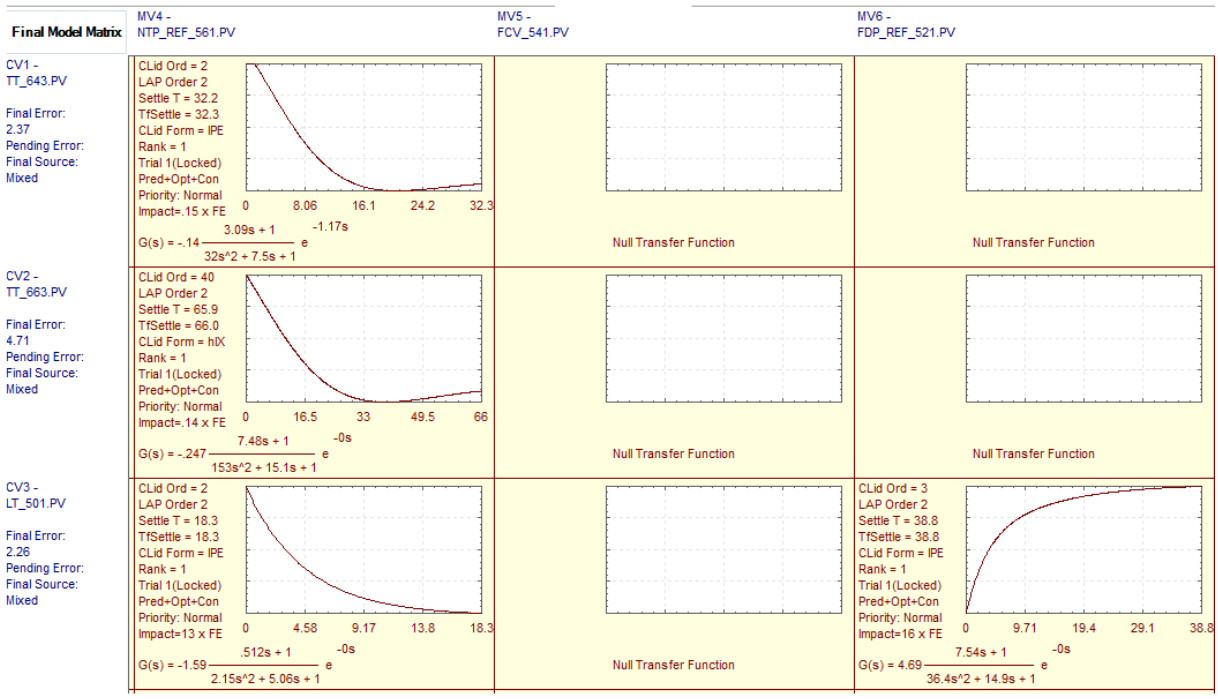


Figure 7.4 TmpPC5_0 models

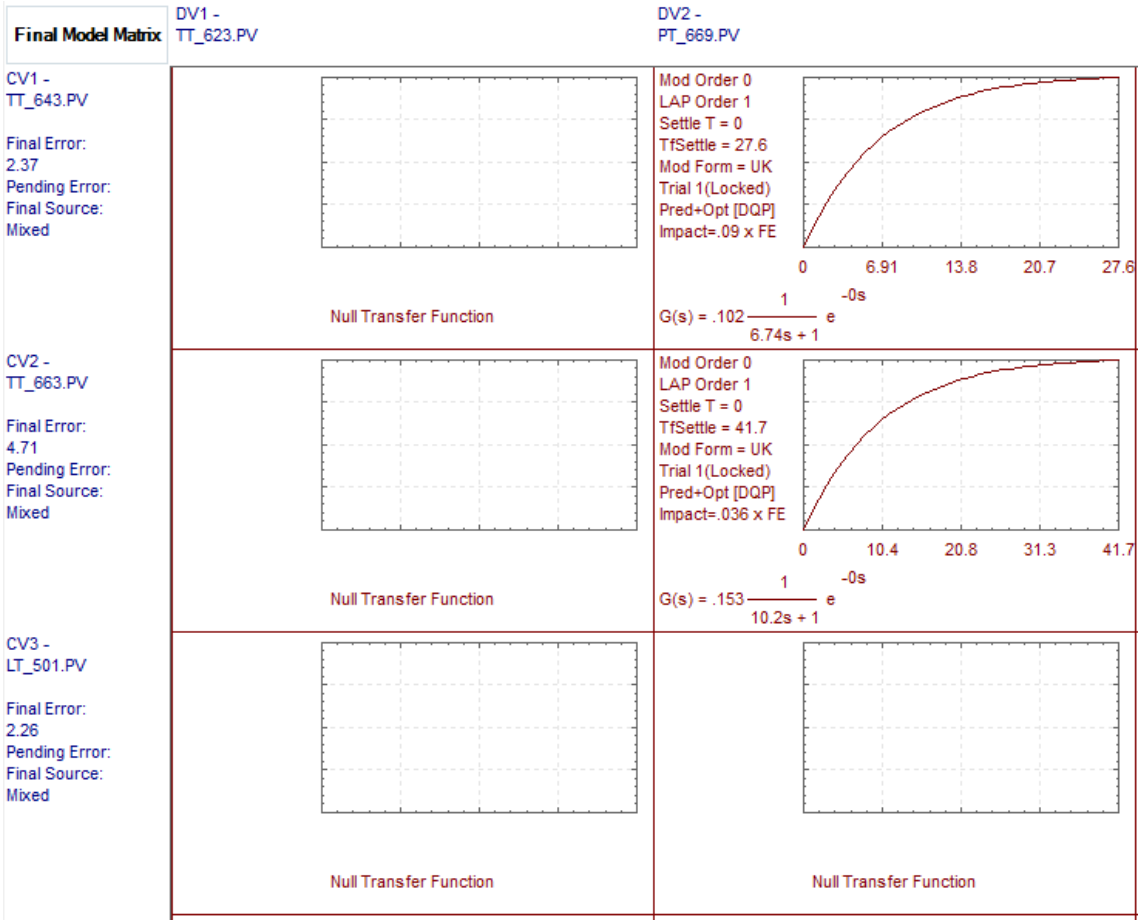


Figure 7.5 TmpPC5_0 models

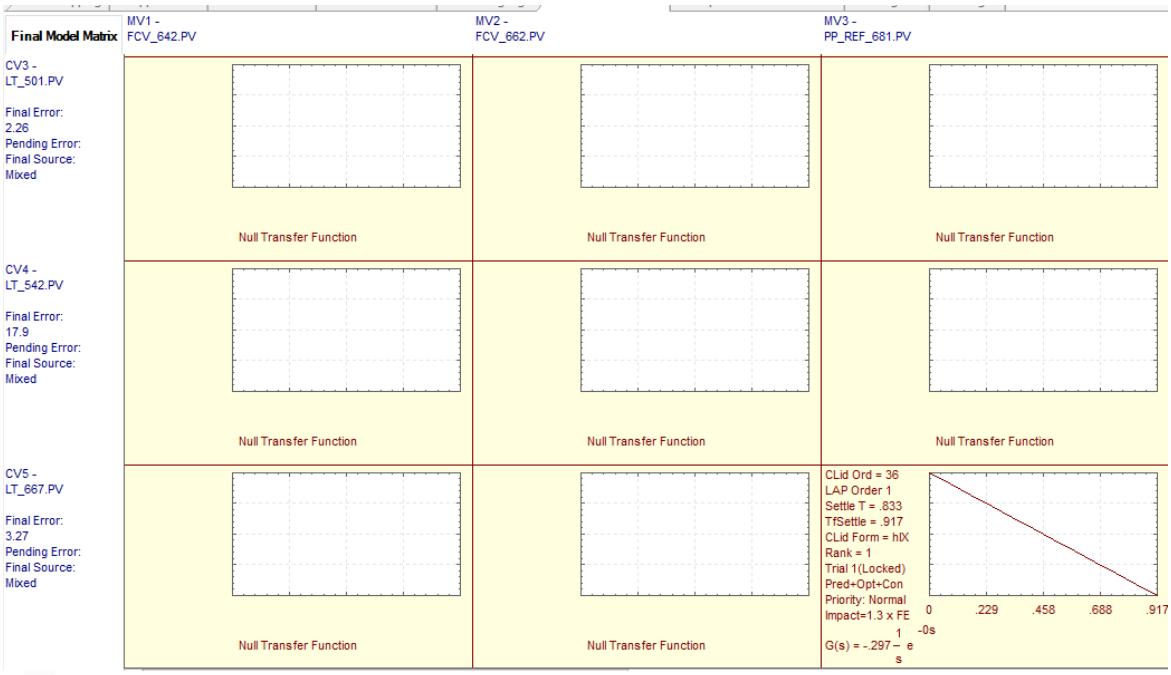


Figure 7.6 TmpPC5_0 models

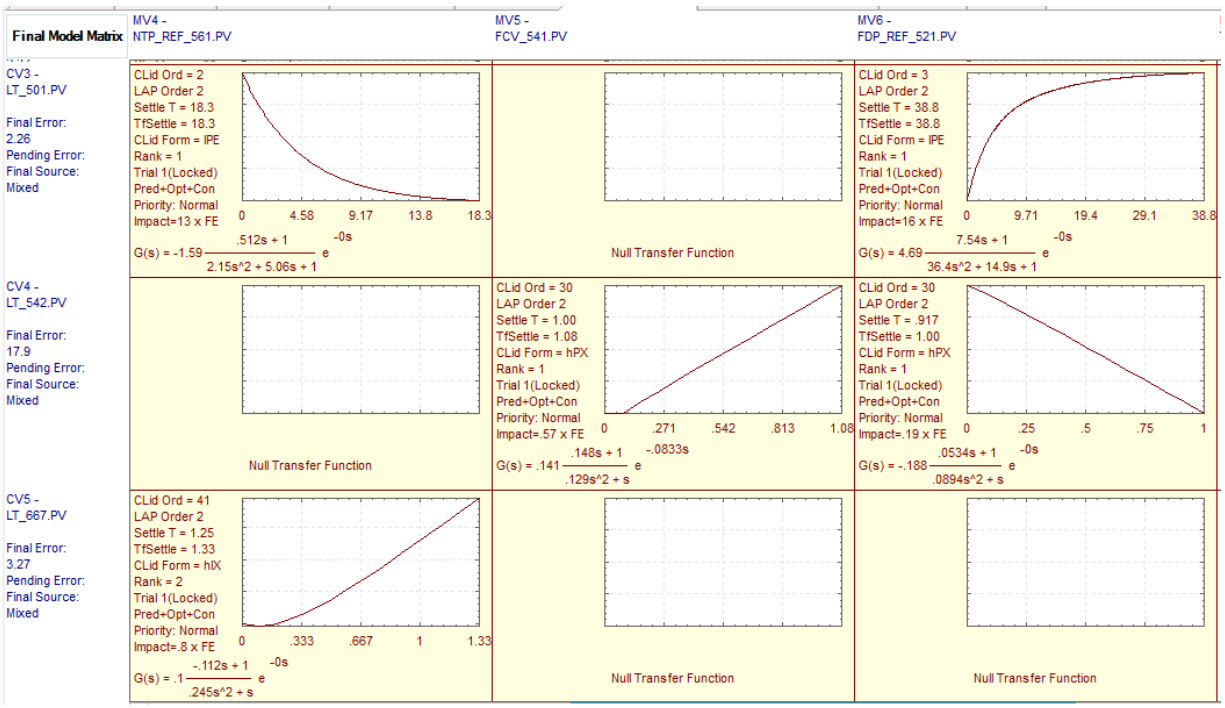


Figure 7.7 TmpPC5_0 models

Figure 7.8 TmpPC5_0 models