Model-Predictive Control of Polyolefin Processes

Y.A. Liu and Niket Sharma
Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061, U.S.A.

Abstract

This chapter presents an in-depth exploration of model-predictive control (MPC) or advanced process control (APC) techniques in the optimization of polyolefin manufacturing processes. Drawing on the foundational motivations outlined in previous discussions, it highlights the pivotal role of APC in enhancing industrial efficiency and innovation. Through a comprehensive introduction to the basic concepts and tools of APC, including definitions of manipulated variables (MV), feedforward/disturbance variables (FF/DV), controlled variables (CV), and the intricacies of multivariable dynamic models, this work delineates the advantages of APC over traditional proportional-integral-derivative (PID) control systems. It further elucidates the mechanisms through which APC achieves its benefits, such as model CV prediction, economic optimization, and dynamic control execution, leveraging Aspen DMCplus and DMC3 control structures for illustration.

The chapter provides a detailed walkthrough of developing a dynamic matrix controller model for a copolymerization process utilizing Aspen DMC3 Builder, transitioning to the formulation and control of nonlinear processes. It addresses the challenges inherent in constructing nonlinear dynamic models for polymerization process control, introduces the Wiener model for nonlinear processes, and discusses the state-space, bounded derivative network (SS-BDN) for nonlinear controller modeling. A hands-on workshop for the development of a nonlinear model-predictive control (NMPC) of a polypropylene process is presented, culminating in an overview of recent advancements in MPC with embedded AI technologies.

Serving both as a primer for newcomers and a sophisticated reference for experienced practitioners and scholars, this work underscores the transformative potential of APC integrated with AI in the polyolefin production sphere. It advocates for the systematic adoption of these advanced control strategies to realize significant improvements in process efficiency, optimization, and innovation within the chemical processing industries.

This is a preprint version of a chapter from our book - *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing. Please cite the original work if referenced [26,35]*

8.1 Introduction to Advanced Process Control (APC)

This chapter covers the fundamentals and practice of model-predictive control (MPC), or advanced process control (APC), of polyolefin processes. The motivation for this chapter appeared previously in Chapter 1, Section 1.4.2, discussing the industrial and potential applications of advanced process control to optimizing polyolefin manufacturing.

We begin by introducing the basic concepts and tools of APC in Section 8.1. Specifically, Section 8.1.1 presents some basic definitions, including manipulated variable (MV), feedforward/disturbance (FF/DV) variables, controlled variable (CV), unit-step response curve, and integrating (ramp) variable. Section 8.1.2 explains the multivariable dynamic model, and the key differences between conventional proportional-integral-derivative (PID) control and APC. This subsection describes where the benefits of APC come from, including model CV prediction and reconciliation to online measurements, steady-state economic optimization to identify MV and CV targets, and dynamic control execution to reach MV and CV targets. This subsection presents the Aspen DMCplus control structure illustrating the three sources of benefits of APC, and the Aspen DMC3 (third-generation dynamic matric control) control structure. Section 8.1.3 introduces linear modeling of dynamic matrix control (DMC), step-response model, and finite-impulse response (FIR) model. Section 8.1.4 covers model evaluation and useful tools, including the concepts of relative gain array, ill-conditioned model matrices and collinear systems, open-loop predictions, prediction error filtering, and prediction update. Section 8.1.6 presents the important concepts and parameters in steady-state economic optimization and dynamic controller simulation. This is a key subsection that a beginner in APC should fully understand to develop and fine-tune advanced process controllers.

Section 8.2 presents a hands-on workshop for the step-by-step development of a dynamic matrix controller model for a copolymerization process using Aspen DMC3 Builder.

Section 8.3 introduces the MPC of nonlinear processes. Specifically, Section 8.3.1 discusses the challenges of developing nonlinear dynamic models for polymerization process control. Section 8.3.2 explains the concept of the Wiener model for nonlinear processes. Section 8.3.3 introduces the state-space, bounded derivative network (SS-BDN) for developing a nonlinear controller model of polyolefin processes.

Section 8.4 presents a hands-on workshop for the step-by-step development of a nonlinear model-predictive control (NMPC) of a polypropylene process. Section 8.5 discusses the new development of model-predictive control with embedded AI, and Section 8.6 gives the bibliography.

8.1.1 Some Basic Definitions

8.1.1a Independent and Dependent Variables

We begin by introducing the basic concepts of advanced process control [1,2,3,4]. Process control and monitoring are pivotal elements in ensuring the efficiency and safety of both chemical [42] and bioprocesses [25,43]. This study specifically zeroes in on the intricacies and challenges associated with polymer processes, underscoring the critical importance of advanced process management strategies in this domain. Figure 8.1 illustrates a simplified flowsheet of a solution copolymerization process. There are two monomers, methyl methacrylate (MMA) and vinyl acetate (VA), an initiator (INITIATO), and a chain-transfer agent (TRANSFER). The reactor has a cooling jacket with a cooling water (CW) stream as the cooling medium. Process details for the polymer processes are described in [31,33,39-41].

A similar practical application of model predictive control in polymerization has been showcased by [44] to control the reaction process to control the temperature and concentration for the polymer precipitation process.

We define *independent variables* as those causal variables whose values are not affected by or are independent of any other variables in the process. We classify the independent variables into two categories:

(1) *Manipulated variables (MVs)* - variables that the operator can change, particularly:

- the *setpoint* to regulate a controller, labelled by **.SP*, such as FMMA.SP, for the setpoint for the mass flow rate of monomer MMA, FMMA; and
- the *valve position* (% open) to regulate a control valve, labelled by **.VP*, such as FVA.VP for the valve position for the control valve for monomer mass flow rate, FVA.

(2) *Feedforward/Disturbance variables (FFs/DVs)* – variables that impact the process, but cannot be adjusted directly, such as:

- the temperature of cooling water through the cooling jacket of the reactor, which depends on an upstream cooling tower system, and varies with seasonal weather.
- Unmeasured temperature of a feed stream, which acts as a disturbance variable.

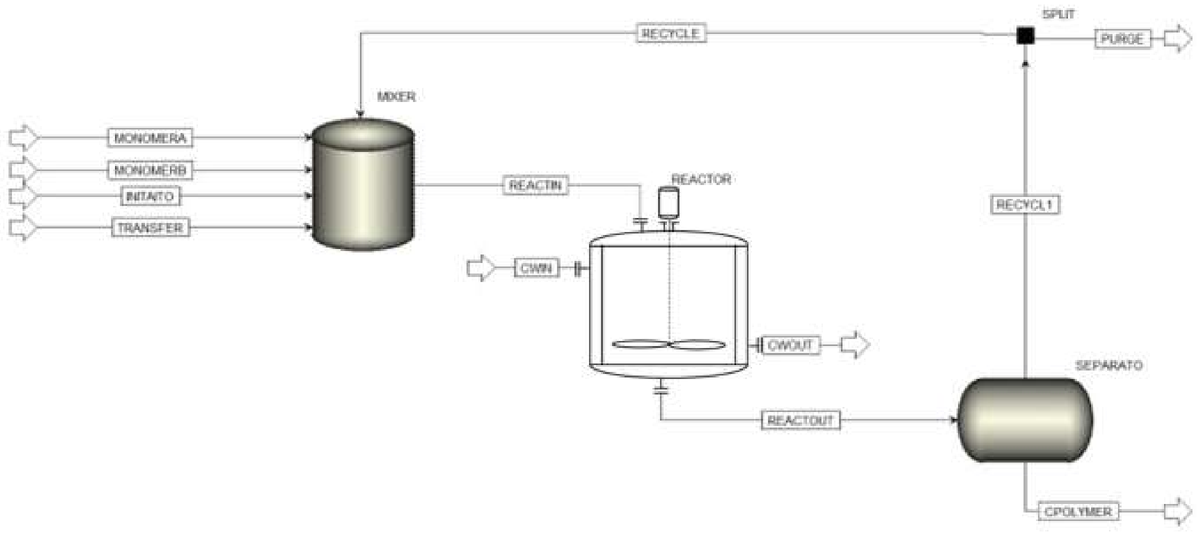


Figure 8.1 A simplified flowsheet of a solution copolymerization process.

We define *dependent variables* as those variables whose dynamic behavior could be fully described by changes in independent variables over time, particularly *control variables (CVs)*, labelled by **.PV*, such as polymer production rate, POLYMER.PV, that are typically maintained at a constant value, or between high and low limits. We note that in a process, there are many dependent variables, but we only choose those important ones as CVs.

For the copolymerization example of Figure 8.1, we consider the following variables:

- MVs: mass flow rates (kg/hr) of monomer MMA and VA, initiator and chain-transfer agent (represented by Flow_MMA.SP, Flow_VA.SP, Init.SP and Transf.SP), and temperature of the cooling jacket, T_Jkt.SP.

- CVs: polymer production rate (kg/hr), polymer molecular weight, reactor exit temperature (°C) and mole fraction of monomer MMA in the polymer product (represented by Polymer.PV, Mol_Wt.PV, T_Rx.PV, and Conc_MMA.PV).
- There is no FF/DV in this example.

8.1.1b Unit-Step Response Curve: Time to Steady State and Steady-State Gain

Figure 8.2 illustrates the step-response curve for a 2MV-1CV process, in which CV_1 varies as a response to a step change of one unit of MV_1. At time t = 12 hr, CV_1 no longer changes and reaches its steady-state value of 1.25 unit. We call the time of 12 hr as *the time to steady state (T_{ss})*, and the ratio of the changes in values of CV1 to MV1 at steady state, that is, $\Delta CV_{1,\,ss}/\Delta MV_{1,\,ss}$ of 1.25/1.0, as *the steady-state gain (SS gain)*.

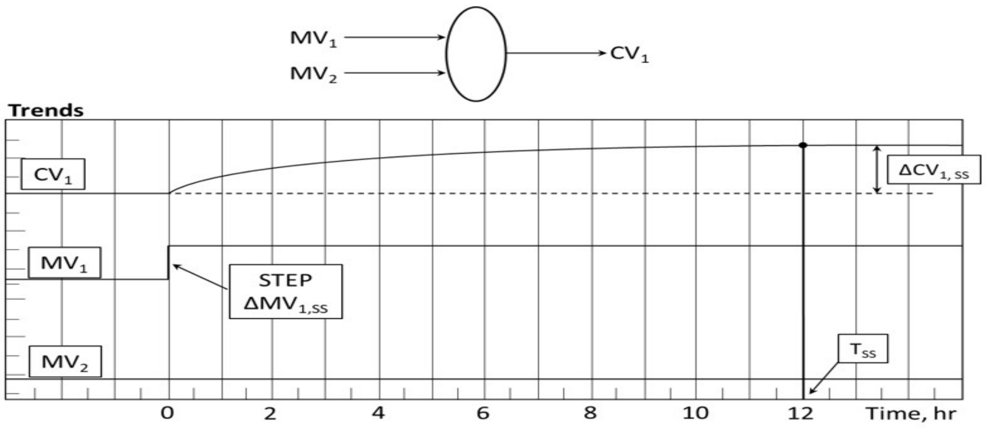


Figure 8.2 A step response curve for CV_1 in a 2MV-1CV process with a step change in MV_1. Note the steady-state gain ($\Delta CV_{1,\,ss}/\Delta MV_{1,\,ss}$) and the time to steady state (T_{ss})

8.1.1c Integrating Variable (Ramp Variable)

Liquid level in a storage vessel with both steady inlet and exit flows is a typical ramp variable or integrating variable. Let us consider a cylindrical storage vessel with inlet and exit liquid volumetric flow rates Fi and Fo m^3/hr, respectively, a cross-sectional area A m^2, a liquid height of h m, and a liquid volume V m^3. See Figure 8.3.

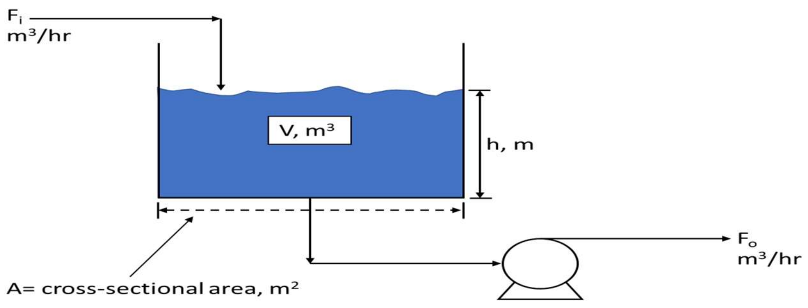


Figure 8.3 Liquid level in a storage tank

A simple volume balance gives:

$$dV/dt = A\,dh/dt = Fi - Fo \qquad (8.1)$$

$$h = \left(\frac{1}{A}\right) \int_0^t [Fi - Fo]dt \qquad (8.2)$$

Based on Eq. (8.2), we call the liquid level h *an integrating variable or a ramp variable*.

If the flow rate entering the vessel Fi is increased and the exit flow rate Fo is held fixed, the liquid level in the vessel increases. The flow exiting the vessel must be increased by the same amount to "balance the level". Therefore, the level exhibits an integrating or a ramp response to changes to the inlet flow rate.

Figure 8.4 illustrates that for an integrating or ramp variable, the step response curve has *a constant steady-state rate of change* or slope of $\Delta(CV_1)/\Delta(MV_1)$, instead of *a constant steady-state value* as in Figure 8.2, and the "traditional" time to steady state Tss does not exist.

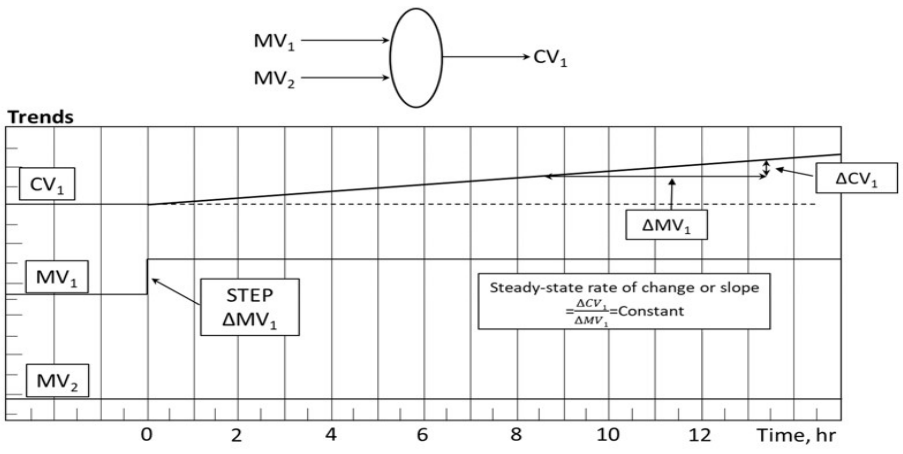


Figure 8.4 A step-response curve for CV1 in a 1MV-1CV integrating process with a step change in MV1.

In addition to liquid level, we can cite examples for other pressure and temperature integrating variables. An example is the material imbalance ramp representing the pressure in a hydroprocessing reactor, for which the hydrogen pressure is a measure of the hydrogen consumption [5]. If the make-up hydrogen flow does not equal to the amount of hydrogen consumed in the reactor, then the pressure either rise or fall. In this case, the pressure is a measure of the hydrogen material balance. Additionally, an example of an energy imbalance ramp is the dense-bed temperature in a fluid catalytic cracking (FCC) regenerator when the unit operates in a partial combustion mode [6]. This happens when the reactor temperature controller is operating in an automatic mode and continually changing the carbon balance on the catalyst. Breaking the reactor temperature controller will eliminate the ramp behavior in this case.

8.1.2 Where do the benefits of APC come from?

We describe three sources of the benefits of APC in this section.

8.1.2a Online Reconciliation of Model-Based Predictions to the Process Measurements to Provide Robustness to the Multivariable Dynamic Step-Response Model

(1) Multivariable Dynamic Model

Extending the step-response model of Figure 8.2 to a system of multiple independent and dependent variables, we can develop a multivariable step-response model to represent the time-dependent changes of control variables (CVs) to changes in manipulated variables (MVs) and feedforward/disturbance variables (FF/DVs). Workshop 8.1 in Section 8.2 gives the details of the development a multivariable predictive controller model for our copolymerization process of Figure 8.1.

Figure 8.5 shows the resulting multivariable step-response model. In the plot, each column represents a dependent variable or a CV, and each row represents an independent variable, a MV or a FF/DV. We typically arrange a FF/DV, if available, as a bottom row in the plot. For the copolymerization example, all 4 columns are CVs, and all 5 rows are MVs, and there is no FF/DV. Note that in the plot, MV Flow_MMA has a negligible impact on CV T_RX, and the model does not show any step-response curve for the MV-CV pair, as the corresponding steady-state gains become negligible. The same is true for three other MV-CV pairs with no step-response curve.

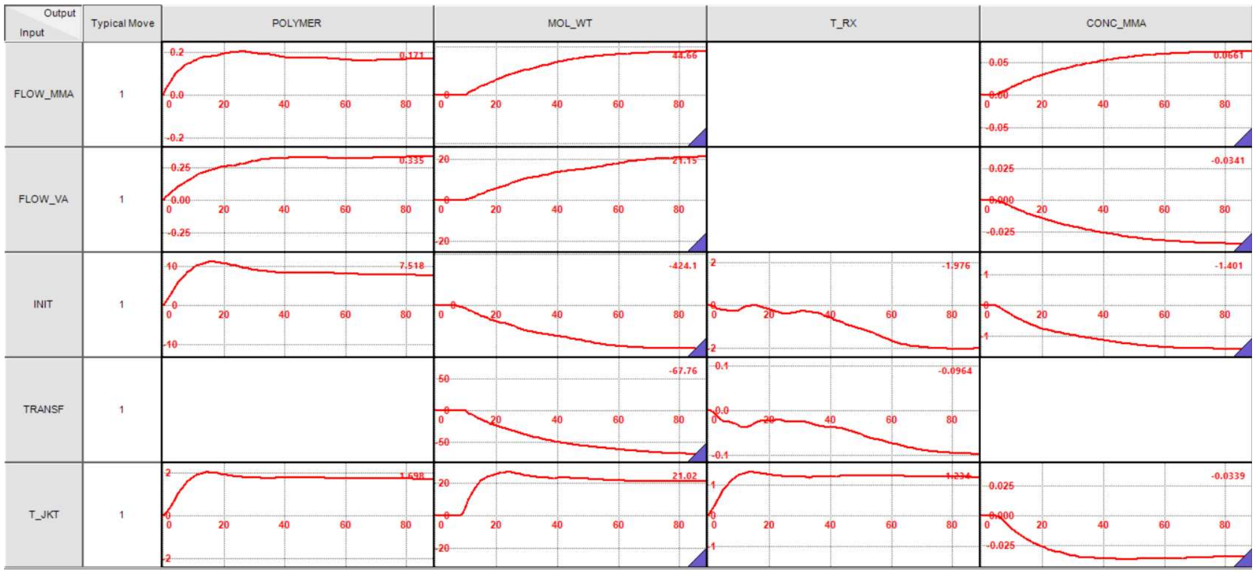
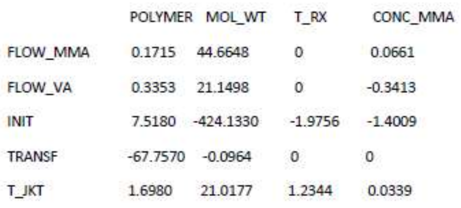


Figure 8.5 A multivariable step-response model for the copolymerization process.

In Figure 8.5, the number at the upper right corner of each step-response curve block represents *the steady-state gain* discussed in Figure 8.2. We can organize the displayed steady-state gains for all step-response curve blocks in *a steady-state gain matrix*, Eqs. (8.3):



	POLYMER	MOL_WT	T_RX	CONC_MMA
FLOW_MMA	0.1715	44.6648	0	0.0661
FLOW_VA	0.3353	21.1498	0	-0.3413
INIT	7.5180	-424.1330	-1.9756	-1.4009
TRANSF	-67.7570	-0.0964	0	0
T_JKT	1.6980	21.0177	1.2344	0.0339

(8.3)

This matrix represents the relationship in Eq. (8.4), which we will use below in introducing the steady-state optimization to obtain MV and CV targets to minimize the operating cost and maximize product profit:

$$\begin{matrix} \frac{\Delta(\text{Polymer})}{\Delta(\text{Flow\_MMA})} & \frac{\Delta(\text{Mol\_Wt})}{\Delta(\text{Flow\_MMA})} & \frac{\Delta(\text{T\_Rx})}{\Delta(\text{Flow\_MMA})} & \frac{\Delta(\text{Conc\_MMA})}{\Delta(\text{Flow\_MMA})} \\ \frac{\Delta(\text{Polymer})}{\Delta(\text{Flow\_VA})} & \frac{\Delta(\text{Mol\_Wt})}{\Delta(\text{Flow\_VA})} & \frac{\Delta(\text{T\_Rx})}{\Delta(\text{Flow\_VA})} & \frac{\Delta(\text{Conc\_MMA})}{\Delta(\text{Flow\_VA})} \\ \frac{\Delta(\text{Polymer})}{\Delta(\text{Init})} & \frac{\Delta(\text{Mol\_Wt})}{\Delta(\text{Init})} & \frac{\Delta(\text{T\_Rx})}{\Delta(\text{Init})} & \frac{\Delta(\text{Conc\_MMA})}{\Delta(\text{Init})} \\ \frac{\Delta(\text{Polymer})}{\Delta(\text{Transf})} & \frac{\Delta(\text{Mol\_Wt})}{\Delta(\text{Transf})} & \frac{\Delta(\text{T\_Rx})}{\Delta(\text{Transf})} & \frac{\Delta(\text{Conc\_MMA})}{\Delta(\text{Transf})} \\ \frac{\Delta(\text{Polymer})}{\Delta(\text{T\_Jkt})} & \frac{\Delta(\text{Mol\_Wt})}{\Delta(\text{T\_Jkt})} & \frac{\Delta(\text{T\_Rx})}{\Delta(\text{T\_Jkt})} & \frac{\Delta(\text{Conc\_MMA})}{\Delta(\text{T\_kJt})} \end{matrix} \qquad (8.4)$$

(2) Key Differences between Traditional PID Control and Advanced Process Control

Figure 8.6 compares the traditional PID control and advanced process control (APC). A key difference between the two is that the traditional PID control aims at keeping a CV at its setpoint, while the APC maintains a CV between its specified lower and upper limits. Thus, an operator of an APC system is to specify the lower and upper limits of a CV, but not its setpoint.

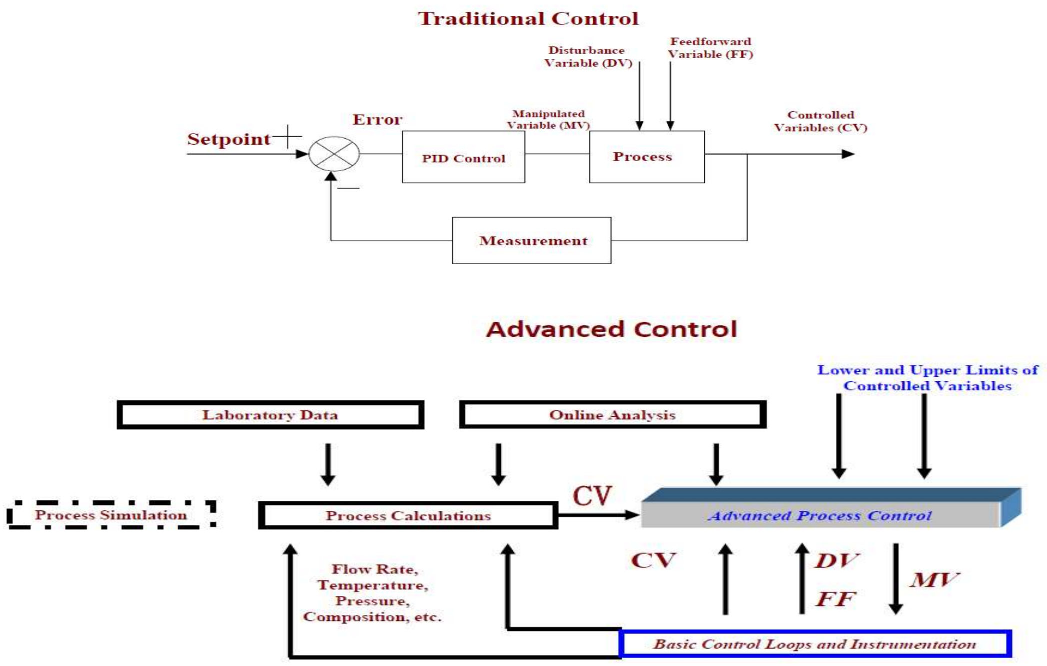

Traditional Control
Disturbance Variable (DV)
Feedforward Variable (FF)
Setpoint
+
Error
Manipulated Variable (MV)
Controlled Variables (CV)
PID Control
Process
−
Measurement
Advanced Control
Lower and Upper Limits of Controlled Variables
Laboratory Data
Online Analysis
CV
Process Simulation
Process Calculations
Advanced Process Control
CV
DV
FF
MV
Flow Rate, Temperature, Pressure, Composition, etc.
Basic Control Loops and Instrumentation

Figure 8.6 A comparison of traditional PID control (top) and advanced process control (bottom).

From a control point of view, as long as MVs are within their lower and upper limits, and the predicted value of the CV from the dynamic process model is also within its lower and upper limits, then there is no need to vary the CV value and the corresponding values of MVs that affect this CV. This minimizes the frequency of adjusting the MVs that impact a chosen CV, thus greatly minimizing the fluctuations of CVs and enhancing the operational stability of the control system. Figure 8.7 illustrates two facts: (1) APC could typically reduce the fluctuations of CVs by 30% or more; and (2) through a steady-state optimization step that we will discuss further below, APC typically operates at or near the lower or upper limits of CVs that minimize the steady-state operating cost and maximize the product profit, called *the economic optimum variable target.*

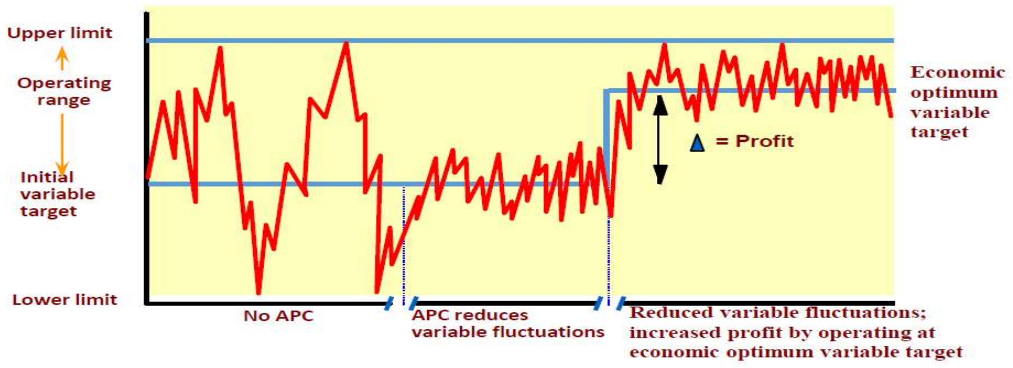


Figure 8.7 Reduced variable fluctuations and increased profit by operating at economic optimum variable target

(3) Continuing Reconciliation of Model-Based Predictions to the Process Measurements and Feedback Correction to Update the Model Predictions to the Future [2]

We illustrate a key aspect of predictive modeling of APC that makes it less sensitive to modeling errors and more accurate in predicting future CV response. Specifically, we consider a simple fire heater example of Figure 8.8, modified from [7].

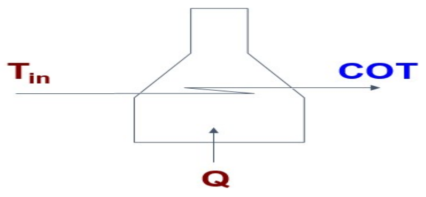


Figure 8.8 A simplified fire heater with two MVs (stream inlet temperature T_{in} and input heat duty Q) and a CV (heating coil output temperature, COT).

Figure 8.9 shows the step response curve for the fired heater.

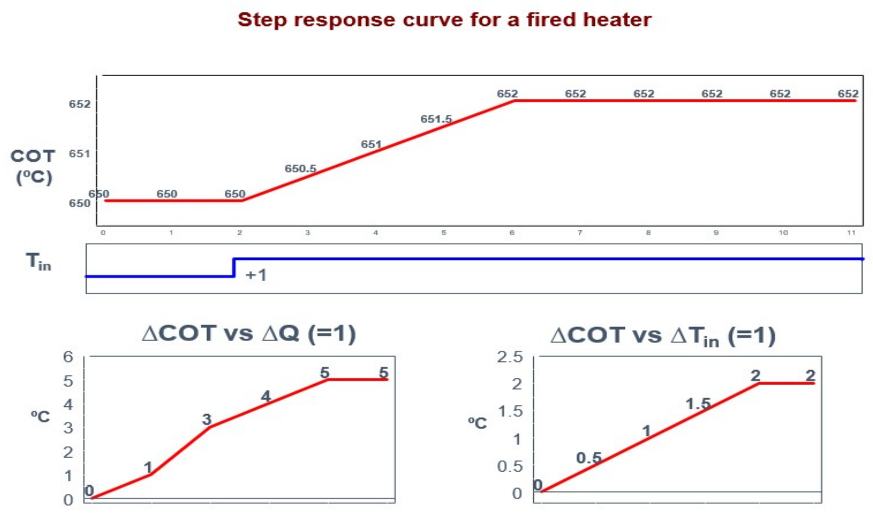


Figure 8.9 Step-response curve for a fired heater

Figures 8.10 and 8.11 demonstrate the continuing feedback corrections of the predicted CV responses based on measured CV values to minimize the prediction errors of CVs at the end of each sampling period of 1 minute.

In the middle plot of Figure 8.10, we see the initial CV prediction (dark black curve) deviates from its measured value (dark square point) at 12:01. The online feedback correction shifts the CV prediction curve downward to match the measured CV value at 12:01 in the bottom plot. The black “real process” in the bottom plot also shows that the deviation of the initial CV prediction only exists within a sampling period of 1 minute. At the end of a sampling period of 1 minute, that is, at 12:01, the online feedback correction based on the CV measured value completely eliminates the deviation.

Figure 8.11 repeats the same online feedback correction process, making the corrected CV prediction at 12:02 equal to the CV measured value.

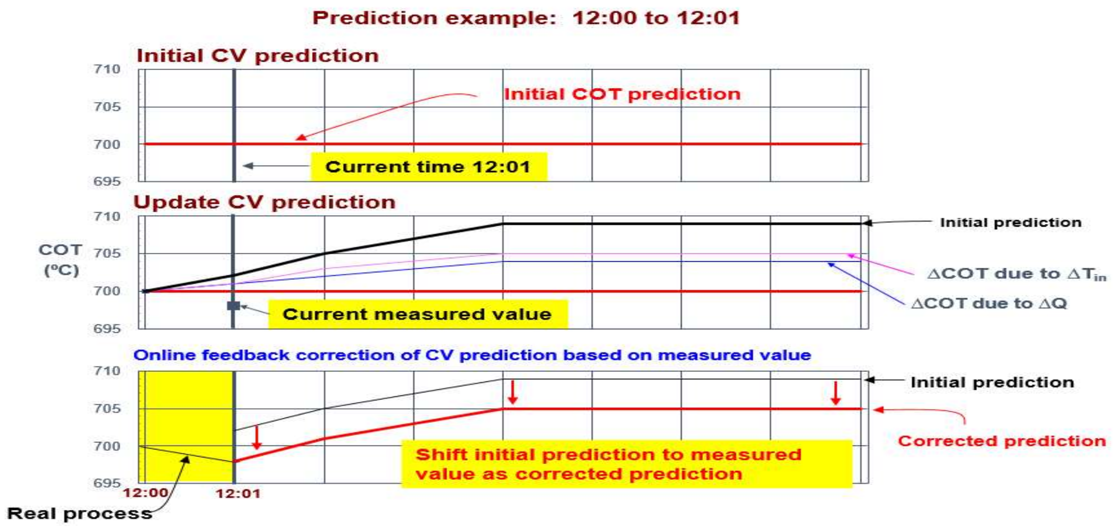


Figure 8.10 Online feedback correction of CV prediction based on measured value from 12:00 to 12:01.

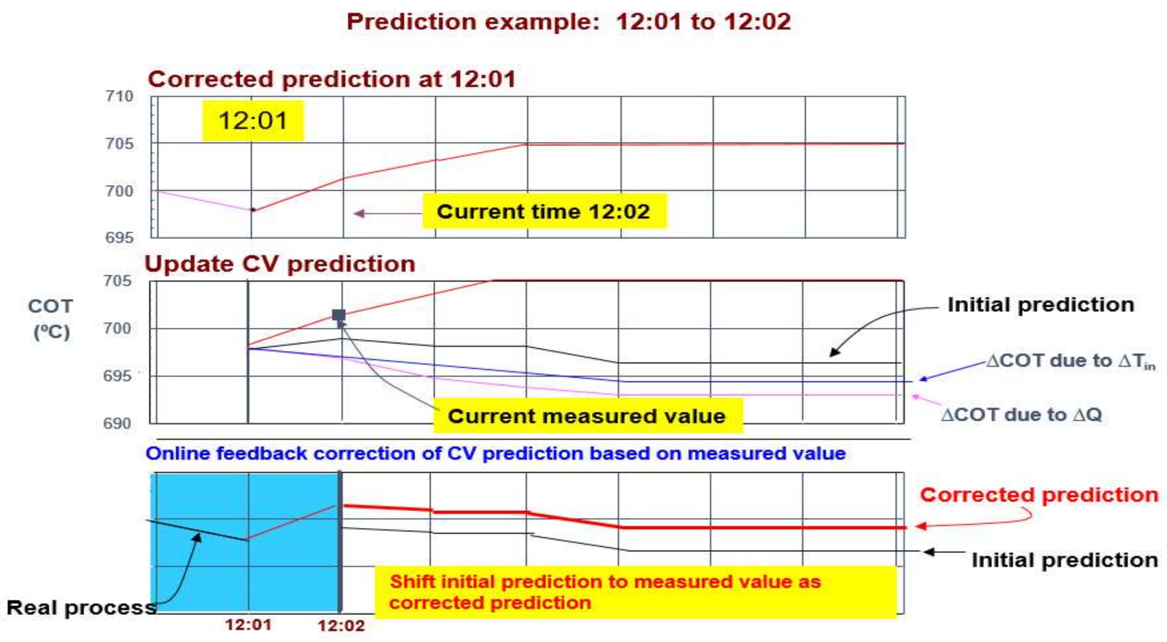


Figure 8.11 Online feedback correction of CV prediction based on measured value from 12:01 to 12:02.

Because the multivariable dynamic model based on step-response tests are data-driven and is not 100 percent accurate, APC strategy includes the online feedback corrections of initial CV predictions based on CV measured values to eliminate the CV prediction errors at the end of each sampling period. This approach reconciles the model-based predictions to the process measurements, and then feeds the information back to update the model predictions into the future [2]. This results in the robustness to the multivariable dynamic model in predicting accurately CV responses to changes in MVs, and *this accurate model prediction capability represents the first source of benefits of APC.*

8.1.2b Steady-State Economic Optimization to Determine MV and CV Targets to Minimize Cost and Maximize Profit

The second source of benefits of APC results from the steady-state optimization to determine the MV and CV targets to minimize the cost and maximize the profit. Under the constraints of the lower and upper limits of all MVs and CVs, the DMC strategy typically minimizes a linear objective function of the form [7]:

$$\text{Min } \phi = \text{Cost}_1 \times \Delta MV_1 + \text{Cost}_2 \times \Delta MV_2 + \ldots\ldots + \text{Cost}_i \times \Delta MV_i \qquad (8.5)$$

where Cost_i is essentially the steady-state gain:

$$\text{Cost}_i = \left(\frac{\Delta\phi}{\Delta MV_i}\right)_{\Delta MV_j} \; (j \neq i) \qquad (8.6)$$

To minimize cost and maximize profit, we may write the objective function as

$$\begin{aligned}\phi &= \text{Cost} - \text{Profit}\\ &= + (\text{steady-state change in feed/utilities}) * (\$ \text{ cost of feed/utilities})\\ &\quad - (\text{steady-state change in production}) * (\$ \text{ value of products}) \qquad (8.7)\end{aligned}$$

For the copolymerization example, we write:

ϕ= Cost – Profit

$$= \{\Delta\text{Flow\_MMA x (cost of Flow\_MMA)} + \Delta\text{Flow\_VA x (cost of Flow\_VA)} + \Delta\text{Flow\_INIT x (cost of INIT)} + \Delta\text{Flow\_Transf x (cost of Transf)} + \Delta T_{jkt} \text{ x (cost of } T_{jkt})\}$$

$$- \{(\frac{\Delta(\text{Polymer})}{\Delta(\text{Flow\_MMA})})(\Delta\text{Flow\_MMA}) + (\frac{\Delta(\text{Polymer})}{\Delta(\text{Flow\_VA})})(\Delta\text{Flow\_VA}) + (\frac{\Delta(\text{Polymer})}{\Delta(\text{Init})})(\Delta\text{Flow\_INIT}) + (\frac{\Delta(\text{Polymer})}{\Delta(\text{Transf})})(\Delta\text{Flow}_{\text{Transf}}) + (\frac{\Delta(\text{Polymer})}{\Delta(T_{\text{Jkt}})})(\Delta T_{jkt})\}*(\$ \text{ value of polymer})$$

$$=\{(\text{cost of Flow\_MMA}) - (\frac{\Delta(\text{Polymer})}{\Delta(\text{Flow\_MMA})}) * (\$\text{ value of polymer})\} * \Delta\text{Flow\_MMA} +$$

$$\{(\text{cost of Flow\_VA}) - (\frac{\Delta(\text{Polymer})}{\Delta(\text{Flow\_VA})}) * (\$\text{ value of polymer})\} * \Delta\text{Flow\_VA} +$$

$$\{(\text{cost of Flow\_INIT}) - (\frac{\Delta(\text{Polymer})}{\Delta(\text{Flow\_INIT})}) * (\$\text{ value of polymer})\} * \Delta\text{Flow\_INIT} +$$

$$\{(\text{cost of Flow\_Transf}) - (\frac{\Delta(\text{Polymer})}{\Delta(\text{Flow\_Transf})}) * (\$\text{ value of polymer})\} * \Delta\text{Flow\_Transf} +$$

$$\{(\text{cost of } T_{jkt}) - ((\Delta(\text{Polymer}))/(\Delta(\text{Flow\_VA})))*(\$\text{ value of polymer})\}* \Delta\text{Flow\_VA}$$

$$= \sum_{i=1}^{i=5} \left[(\text{cost of } MV_{i,SS}) - \frac{\Delta(\text{Polymer})}{\Delta(MV_i)} * (\$value\ of\ polymer)\right] * (\Delta MV_{i,SS})$$

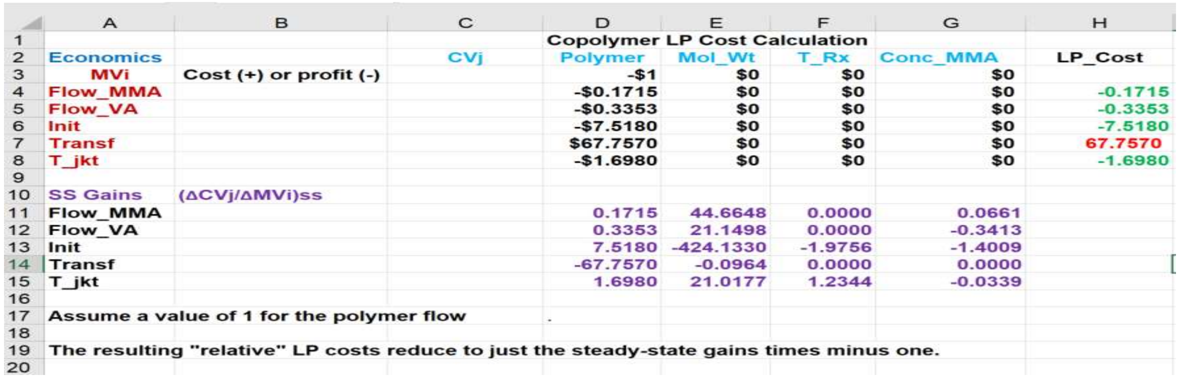
$$=\sum_{i=1}^{i=5} Cost_i * \Delta MV_{i,SS} \qquad (8.8)$$

where the subscript *SS* represents steady state, and

$$Cost_i = = (\frac{\Delta\phi}{\Delta MV_i})_{\Delta MV_j}\ (j \neq i) = (\text{cost of } MV_{i,SS}) - \frac{\Delta(\text{Polymer})}{\Delta(MV_i)} * (\$value\ of\ polymer) \qquad (8.9)$$

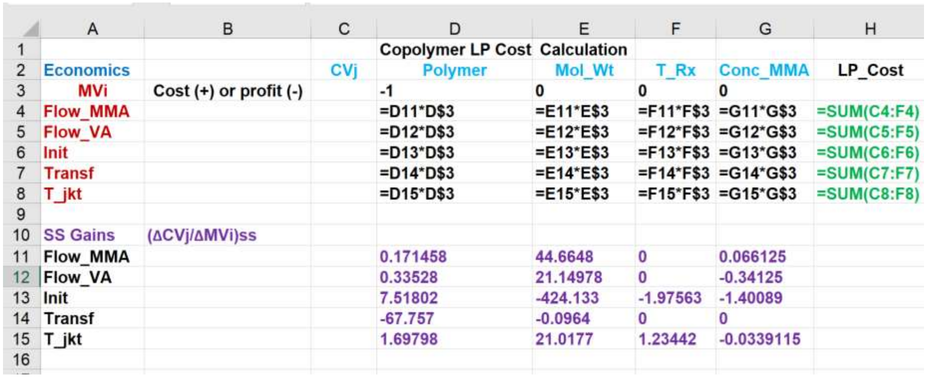
We call $Cost_i$ *the steady-state LP cost* that minimizes the objective function ϕ (= cost – profit) by linear programming (LP). This minimization is constrained by the lower and upper limits of all MVs and CVs.

Figure 8.12 illustrates an Excel calculation of the steady-state LP costs (LP_Cost) based on the steady-state gains in Eq. (8.3), assuming the cost of $MV_{i,SS}$ ($i = 1$ to 5) to be insignificant when compared to the profit of polymer product (if this is not true, we can enter the cost of of $MV_{i,SS}$, $i = 1$ to 5, into the spreadsheet). We also assume the profit of 1 kg/hr of polymer product to be \$1. Figure 8.13 shows the calculation formulas for the LP_Cost.



	A	B	C	D	E	F	G	H
1				Copolymer LP Cost Calculation				
2	Economics		CVj	Polymer	Mol_Wt	T_Rx	Conc_MMA	LP_Cost
3	MVi	Cost (+) or profit (-)		-\$1	\$0	\$0	\$0	
4	Flow_MMA			-\$0.1715	\$0	\$0	\$0	-0.1715
5	Flow_VA			-\$0.3353	\$0	\$0	\$0	-0.3353
6	Init			-\$7.5180	\$0	\$0	\$0	-7.5180
7	Transf			\$67.7570	\$0	\$0	\$0	67.7570
8	T_jkt			-\$1.6980	\$0	\$0	\$0	-1.6980
9								
10	SS Gains	(ΔCVj/ΔMVi)ss						
11	Flow_MMA			0.1715	44.6648	0.0000	0.0661	
12	Flow_VA			0.3353	21.1498	0.0000	-0.3413	
13	Init			7.5180	-424.1330	-1.9756	-1.4009	
14	Transf			-67.7570	-0.0964	0.0000	0.0000	
15	T_jkt			1.6980	21.0177	1.2344	-0.0339	
16								
17	Assume a value of 1 for the polymer flow							
18								
19	The resulting "relative" LP costs reduce to just the steady-state gains times minus one.							
20								

Figure 8.12 Excel calculation of steady-state LP costs.



	A	B	C	D	E	F	G	H
1				Copolymer LP Cost	Calculation			
2	Economics		CVj	Polymer	Mol_Wt	T_Rx	Conc_MMA	LP_Cost
3	MVi	Cost (+) or profit (-)		-1	0	0	0	
4	Flow_MMA			=D11*D$3	=E11*E$3	=F11*F$3	=G11*G$3	=SUM(C4:F4)
5	Flow_VA			=D12*D$3	=E12*E$3	=F12*F$3	=G12*G$3	=SUM(C5:F5)
6	Init			=D13*D$3	=E13*E$3	=F13*F$3	=G13*G$3	=SUM(C6:F6)
7	Transf			=D14*D$3	=E14*E$3	=F14*F$3	=G14*G$3	=SUM(C7:F7)
8	T_jkt			=D15*D$3	=E15*E$3	=F15*F$3	=G15*G$3	=SUM(C8:F8)
9								
10	SS Gains	(ΔCVj/ΔMVi)ss						
11	Flow_MMA			0.171458	44.6648	0	0.066125	
12	Flow_VA			0.33528	21.14978	0	-0.34125	
13	Init			7.51802	-424.133	-1.97563	-1.40089	
14	Transf			-67.757	-0.0964	0	0	
15	T_jkt			1.69798	21.0177	1.23442	-0.0339115	
16								

Figure 8.13 Formulas for Excel calculation of steady-state LP costs.

We wish to minimize the objective function ϕ (=cost –profit) following Eq. (8.8). Based on the calculated steady-state LP costs, *Cost$_i$,* in Figure 8.12, should we move a specific MV$_i$ toward its lower limit or upper limit? In other words, what is the MV$_i$ target value based on steady-state optimization to minimize cost and maximize profit? The resulting MV$_i$ target value appears in Table 8.1. We note that the CV$_j$ target value is to maximize the polymer production within the given lower and upper limits.

Table 8.1 Recommended MV$_i$ target value based on steady-state optimization to minimize cost and maximize profit

MV$_i$	Cost$_i$	$\Delta\mathbf{MV}_{i,ss}$
MV$_1$: Flow_MMA	-0.1716	Small positive increase toward upper limit
MV$_2$: Flow_VA	-0.3353	Small positive increase toward upper limit
MV$_3$:Flow_INIT	-7.7518	Large positive increase toward upper limit
MV$_4$:Flow_Transf	+67.760	Large negative decrease toward lower limit
MV$_5$:T$_{jkt}$	-1.1980	Medium positive increase toward upper limit

This example illustrates how DMC uses the steady-state optimization to identify *the economic optimum steady-state MV and CV target values* to minimize cost and maximize profit, which represents *the second source of benefits of APC*.

8.1.2c Determination of Future MV Moves to Minimize the Least-Squares Errors between Predicted and Desired Economic Optimum Target Values of CVs

Having identified the economic optimum steady-state MV and CV targets, the DMC strategy determines a set of future MV adjustments that will drive the process to the CV to its desired economic optimum operating point without violating the lower and upper limits of MVs and CVs. Figure 8.14 shows the open-loop CV prediction reflecting the effects of past MV changes, and the error between the predicted CV value and its setpoint or the economic optimum target value [7]. The desired future response of each CV is to have it reach its setpoint or economic optimum, steady-state target value. Figure 8.15 illustrates the desired future CV response that is defined by the mirror image of the CV prediction about the setpoint or economic optimum, steady-state target value [2]. Figure 8.16 displays the development of MV moves to minimize the least-squares errors between the predicted and desired values of CVs. We will demonstrate this step quantitatively in Section 8.2.12 below. Depending on the time to steady state, sampling period, and controller execution interval, the DMC strategy calculates 8 to 14 future moves of each MV extending approximately one-half of the time to steady state into the future (see Section 8.2.7). This step of dynamic control execution to reach economic optimum, steady-state MV and CV targets represents *the third source of benefits of APC.*

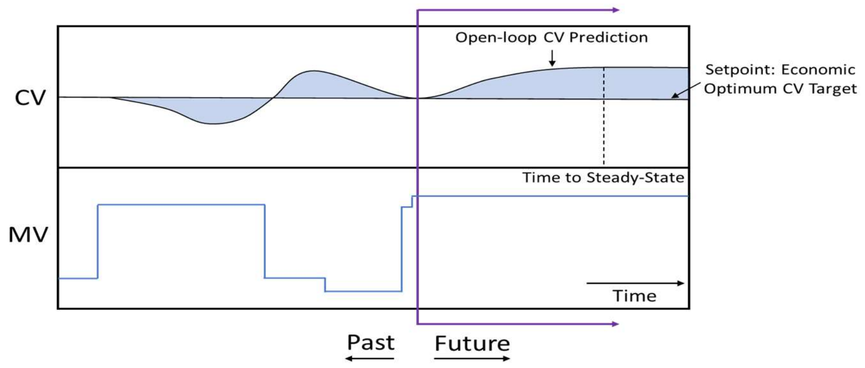


Figure 8.14 The open-loop CV prediction reflecting the effects of past MV changes and the shaded area of errors between the CV prediction and its setpoint or economic optimum target value

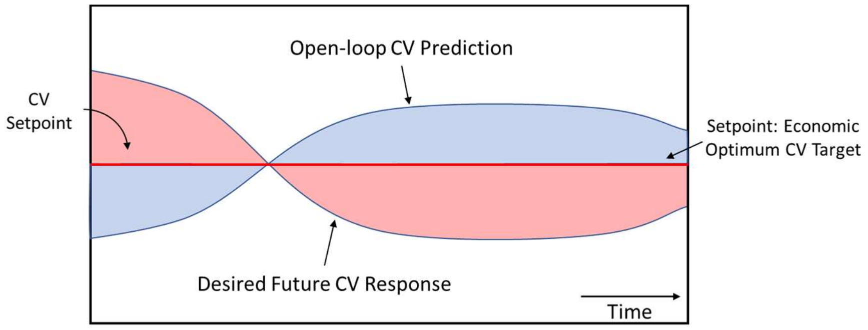

Open-loop CV Prediction
CV Setpoint
Setpoint: Economic Optimum CV Target
Desired Future CV Response
Time

Figure 8.15 An illustration of the desired future CV response that best fits the mirror image of the CV prediction about the setpoint. The shaded area represents the CV errors.

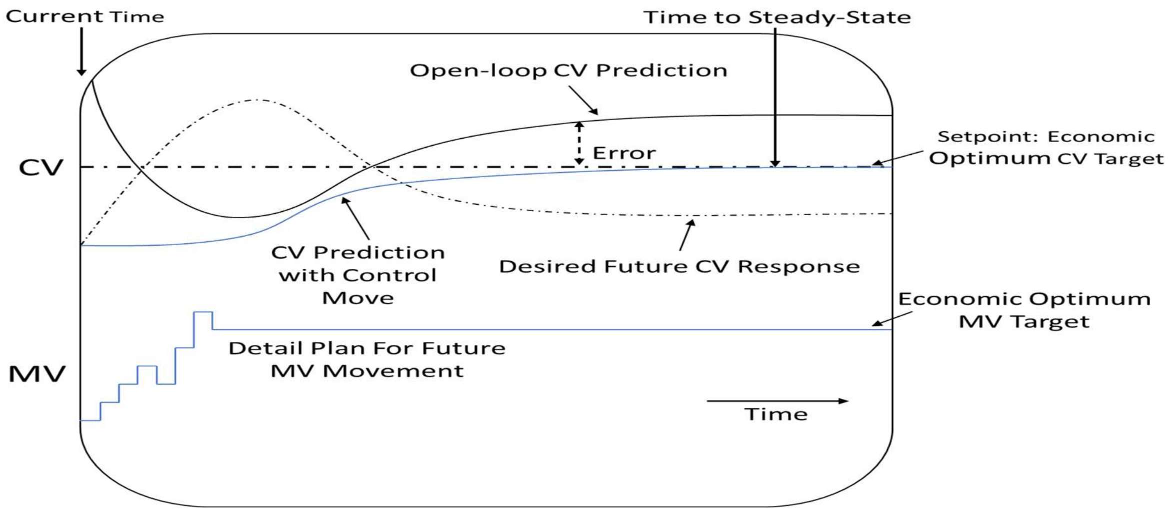

Current Time
Time to Steady-State
Open-loop CV Prediction
Error
CV
Setpoint: Economic Optimum CV Target
CV Prediction with Control Move
Desired Future CV Response
Economic Optimum MV Target
Detail Plan For Future MV Movement
MV
Time

Figure 8.16 An illustration of the development of MV moves to minimize the lease-squares errors between predicted and desired values of CVs

To summarize, in Sections 8.1.2a to 8.1.2c, we have introduced three aspects of the DMC strategy that represent the sources of benefits of APC: (1) *model CV prediction and reconciliation to online measurements*: continuing reconciliation of model-based predictions to the process measurements and feedback correction to update the model predictions to the future; (2) *steady-state economic optimization*: steady-state economic optimization to determine MV and CV targets to minimize cost and maximize profit; and (3) *dynamic control execution to reach MV and CV targets:* determination of future MV moves to minimize the least-squares errors between predicted and desired economic optimum target values of CVs. In Figure 8.17, we have modified a diagram in [7] to illustrate these three sources of benefits of APC in the context of the DMCplus control structure.

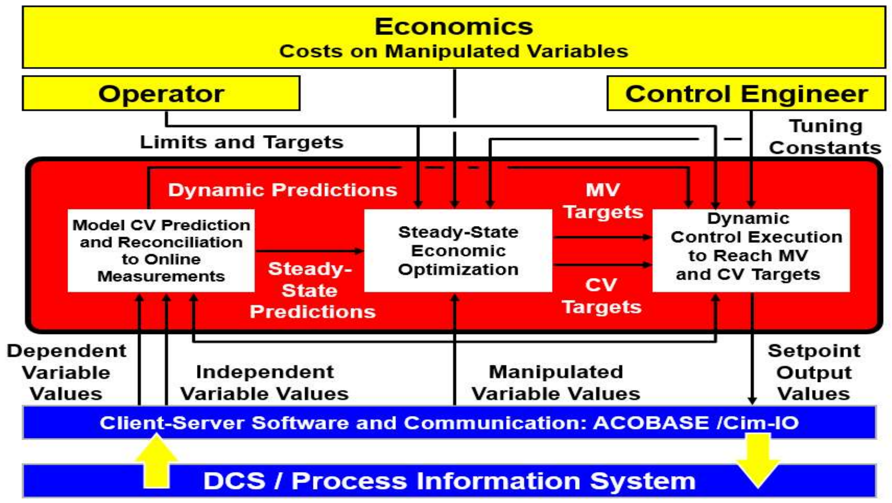

Economics
Costs on Manipulated Variables
Operator
Control Engineer
Limits and Targets
Tuning Constants
Dynamic Predictions
MV Targets
Model CV Prediction and Reconciliation to Online Measurements
Steady-State Economic Optimization
Dynamic Control Execution to Reach MV and CV Targets
Steady-State Predictions
CV Targets
Dependent Variable Values
Independent Variable Values
Manipulated Variable Values
Setpoint Output Values
Client-Server Software and Communication: ACOBASE /Cim-IO
DCS / Process Information System

Figure 8.17 The Aspen DMCplus control structure illustrating three sources of benefits of APC [7]. Used with permission from Aspen Technology, Inc.

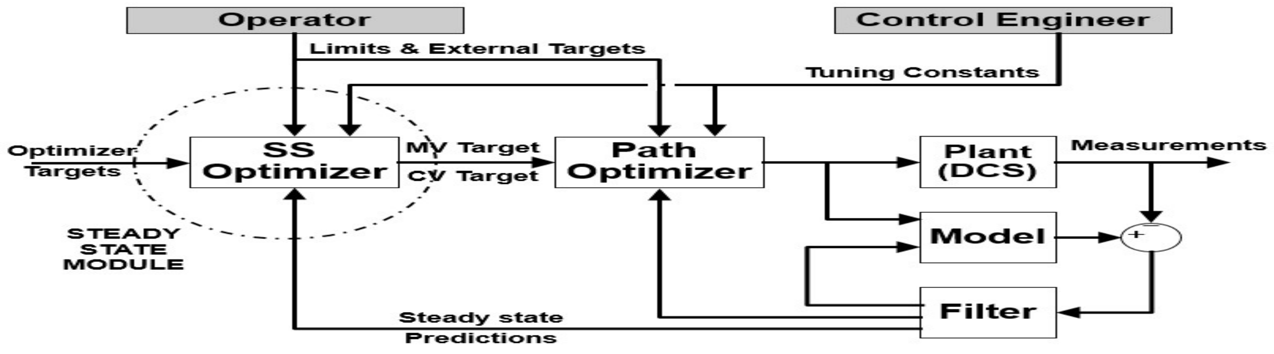


Figure 8.18 The Aspen DMC3 control structure [7]. Used with permission from Aspen Technology, Inc.

Figure 8.18 shows the Aspen DMC3 control structure taken from the DMC3 online help that extends the DMC control structure to provide more robust dynamic control. We note the five key *blocks or controller applications* in the figure. We follow the DMC3 online help to briefly describe these blocks or controller applications below.

(1) **The "plant" ("process") block or controller application**: The "plant" stage of application development occurs in the controller deployment stage, where we specify input/output (or MV/CV) connection parameters to prepare the controller for online operation.

(2) **The "model" block or controller application:** It represents the "controller model" that we will discuss in detail beginning in Section 8.1.3.

(3) **The "SS (steady-state) optimizer" block or controller application**: It performs the "steady-state economic optimization" to find the MV and CV targets, as we illustrated in Section 8.1.2b. In other words, the SS optimizer determines the best steady-state operating point for the plant, subject to the constraints for MVs and CVs.

(4) **The "controller or path optimizer" block or controller application**: It represents "the dynamic control execution" to reach MV and CV targets, or externally specified MV and CV targets that we will discuss more in Section 8.1.6d below. This application develops the move plan to take the plant from its current operating point to the economic optimum steady-state targets or externally specified targets with minimum least-squares errors, while respecting MV and CV constraints.

(5) **The "filter" block or controller application**: It compares the model predictions of CVs with the actual measured CV values at each execution. The filter application helps us understand the current prediction errors by estimating the size of unmeasured disturbances that enter the plant. This comparison tells us where the process is currently operating, and which direction the CVs will go if the MVs remain constant. Disturbance and dynamic state estimate from the filter are then passed to the optimizer.

8.1.3 Linear Modeling for Dynamic Matrix Control (DMC)

8.1.3a Step-Response Model

We use a simple step-response curve of Figure 8.19 to develop a linear matrix-based dynamic process model. In the figure, the MV has a unit step change at time zero, that is, $\Delta MV_0 = 1$.

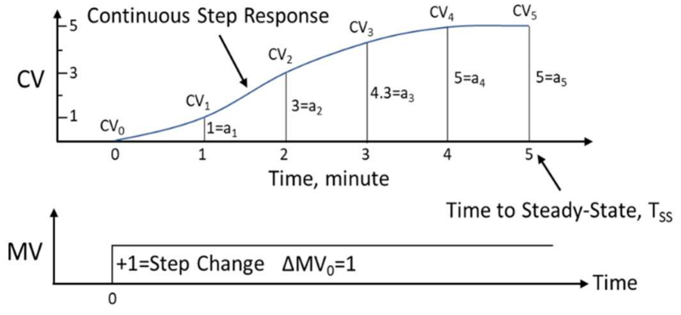


Figure 8.19 Representation of a continuous step response curve by a series of discrete values CV_0, CV_1, CV_2….at time t = 1,2,3, … minute for a unit step change of MV at time zero, $\Delta MV_0 = 1$.

Based on Figure 8.19, we write the following relationships:

$$\Delta CV_1 = CV_1 - CV_0 = 1*\Delta MV_0 = a_1*\Delta MV_0$$

$$\Delta CV_2 = CV_2 - CV_0 = 3*\Delta MV_0 = a_2*\Delta MV_0$$

$$\Delta CV_3 = CV_3 - CV_0 = 4.3*\Delta MV_0 = a_3*\Delta MV_0 \qquad (8.10)$$

$$\Delta CV_4 = CV_4 - CV_0 = 5*\Delta MV_0 = a_4*\Delta MV_0$$

$$\Delta CV_5 = CV_5 - CV_0 = 5*\Delta MV_0 = a_5*\Delta MV_0$$

We illustrate two characteristics of the linearity of the process model in Figures 8.20 and 8.21.

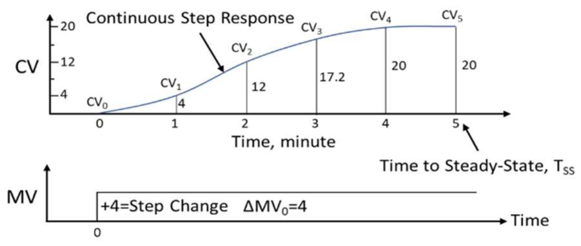


Figure 8.20 An illustration of the model linearity, i.e., preservation of scale, with reference to Figure 8.19

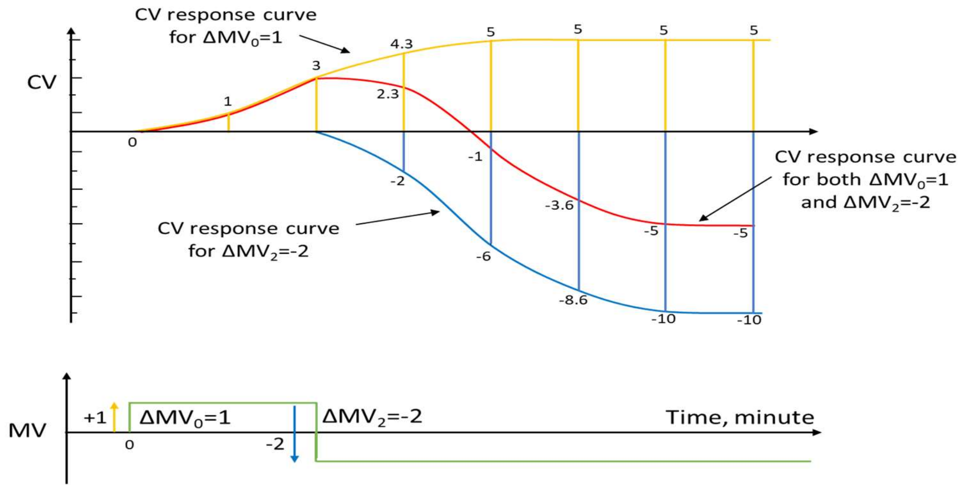


Figure 8.21 An illustration of the superposition principle of model linearity, adding the CV response curve for ΔMV_2 to the CV response curve for ΔMV_0

The first characteristic is *the preservation of scale*, which suggests that if we increase the step change of MV at time zero by four times, that is, $\Delta MV_0 = 1$ to $\Delta MV_0 = 4$, ΔCV_i (i = 1 to 5, ….) will also increase four times, as seen in Figure 8.20. The second characteristic is *the superposition principle*; in Figure 8.21, we add the CV response curve for ΔMV_2 to the CV response curve for ΔMV_0. Extending Eq. (8.11), we write the following relationship for Figure 8.21:

$$\Delta CV_1 = CV_1 - CV_0 = 1*\Delta MV_0 = a_1*\Delta MV_0 = (1)*(1) = 1 \quad (8.11a)$$

$$\Delta CV_2 = CV_2 - CV_0 = 3*\Delta MV_0 + 1*\Delta MV_1 = a_2*\Delta MV_0 + a_1*\Delta MV_1 = 3*(1) + 1*(0) = 3 \quad (8.11b)$$

$$\begin{aligned}\Delta CV_3 = CV_3 - CV_0 &= 4.3*\Delta MV_0 + 3*\Delta MV_1 + 1*\Delta MV_2 = a_3*\Delta MV_0 + a_2*\Delta MV_1 + a_1*\Delta MV_2 \\ &= 4.3*(1) + 3*(0) + 1*(-2) = 2.3 \quad (8.11c)\end{aligned}$$

$$\begin{aligned}\Delta CV_4 = CV_4 - CV_0 &= 5*\Delta MV_0 + 4.3*\Delta MV_1 + 3*\Delta MV_2 \\ &= a_4*\Delta MV_0 + a_3*\Delta MV_1 + a_2*\Delta MV_2 = 5*(1) + 4.3*(0) + 3*(-2) = -1 \quad (8.11d)\end{aligned}$$

$\Delta CV_5 = CV_5 - CV_0 = 5*\Delta MV_0 + 5*\Delta MV_1 + 4.3*\Delta MV_2 = a_5*\Delta MV_0 + a_4*\Delta MV_1 + a_3*\Delta MV_2$
$= 5*(1) + 5*(0) + 4.3*(-2) = -3.6$ (8.11e)

$\Delta CV_6 = CV_6 - CV_0 = 5*\Delta MV_0 + 5*\Delta MV_1 + 4.3*\Delta MV_2 = a_6*\Delta MV_0 + a_5*\Delta MV_1 + a_4*\Delta MV_2$
$= 5*(1) + 5*(0) + 5*(-2) = -5$ (8.11f)

$\Delta CV_7 = CV_7 - CV_0 = 5*\Delta MV_0 + 5*\Delta MV_1 + 5*\Delta MV_2 = a_7*\Delta MV_0 + a_6*\Delta MV_1 + a_5*\Delta MV_2$
$= 5*(1) + 5*(0) + 5*(-2) = -5$ (8.11g)

We may re-write Eq. (8.11a-g) in a matrix form:

$$\begin{bmatrix}\Delta CV_1\\ \Delta CV_2\\ \Delta CV_3\\ \Delta CV_4\\ \Delta CV_5\\ \Delta CV_6\\ \Delta CV_7\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 3 & 1 & 0\\ 4.3 & 3 & 1\\ 5 & 4.3 & 3\\ 5 & 5 & 4.3\\ 5 & 5 & 5\\ 5 & 5 & 5\end{bmatrix}\begin{bmatrix}\Delta MV_1\\ \Delta MV_2\\ \Delta MV_3\end{bmatrix} = \begin{bmatrix}a_1 & 0 & 0\\ a_2 & a_1 & 0\\ a_3 & a_2 & a_1\\ a_4 & a_3 & a_2\\ a_5 & a_4 & a_3\\ a_6 & a_5 & a_4\\ a_7 & a_6 & a_5\end{bmatrix}\begin{bmatrix}\Delta MV_1\\ \Delta MV_2\\ \Delta MV_3\end{bmatrix} \qquad (8.12)$$

For this example, *$a_4 = a_5 = a_6 = a_7$,* implying that the process reaches its steady state when time t = 5 min. Additionally, $\Delta MV_4 = \Delta MV_5 = \Delta MV_6 = \Delta MV_7 = 0$.

We write the time-dependent or dynamic linear matrix equation, Eq. (8.12), in a general matrix form that represents the step-response model:

$$\mathbf{\Delta CV = A * \Delta MV} \qquad (8.13)$$

Eq. (8.13) is *the foundational model for dynamic matrix control (DMC)*. It represents three classes of problems**.**

- *Prediction*: Knowing the model matrix A and MV move vector ΔMV, calculate the resulting CV change vector ΔCV.
- *Control*: Knowing the control change vector ΔCV and the model matrix A, find the required MV move vector ΔMV.
- *Identification or modeling*: Knowing the MV move vector and the resulting CV change vector, find the corresponding model matrix A.

8.1.3b Finite-Impulse Response (FIR) Model

Figure 8.22 illustrates that in actual practice, there could be missing input data for MV for a certain time duration, and discontinuous CV response curve with CV instrument failure. The figure shows only three slices of valid MV-CV response curves. We call the process of identifying slices of good continuous MV-CV response curves without any instrument failure or missing data as "*data slicing*". We show below how to modify our modeling equations, (8.11a) to (8.11g), to represent discontinuous MV-CV response curves.

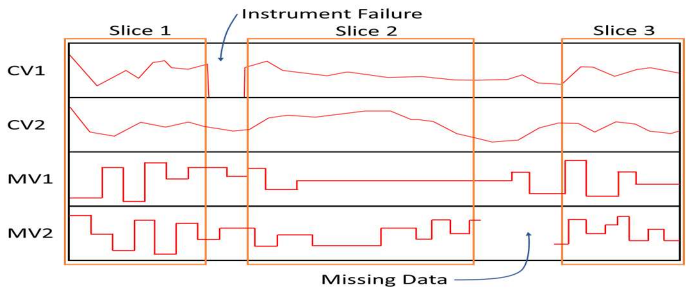


Figure 8.22 An illustration of discontinuous MV-CV response curves with missing MV input data and CV instrument failure, resulting in three slices of valid MV-CV response curves

First, we write Eqs. (8.11b) and (8.11a) as follows.

$$CV_2 – CV_0 = a_2*\Delta MV_0 + a_1*\Delta MV_1 \qquad (8.11b)$$

$$\mathbf{CV_1 – CV_0 = a_1*\Delta MV_0} \qquad (8.11a)$$

Subtracting Eq. (8.11b) by Eq. (8.11a), to remove CV_0 and we get

$$\mathbf{CV_2 – CV_1 = a_1*\Delta MV_1 + (a_2 – a_1)*\Delta MV_0} \qquad (8.14a)$$

Next, we write Eqs. (8.11c) and (8.11b) as follows:

$$CV_3 – CV_0 = a_3*\Delta MV_0 + a_2*\Delta MV_1 + a_1*\Delta MV_2 \qquad (8.11c)$$

$$CV_2 – CV_0 = a_2*\Delta MV_0 + a_1*\Delta MV_1 \qquad (8.11b)$$

Subtracting Eq. (8.11c) by Eq. (8.11b) gives

$$\mathbf{CV_3 – CV_2 = a_1*\Delta MV_2 + (a_2 – a_1)*\Delta MV_1 + (a_3 – a_2)*\Delta MV_0} \qquad (8.14b)$$

Following the same procedure, we can get:

$$\mathbf{CV_4 – CV_3 = a_1*\Delta MV_3 + (a_2 – a_1)*\Delta MV_2 + (a_3 – a_2)*\Delta MV_1 + (a_4 – a_3)*\Delta MV_0} \qquad (8.14c)$$

$$\mathbf{CV_5 – CV_4 = a_1*\Delta MV_4 + (a_2 – a_1)*\Delta MV_3 + (a_3 – a_2)*\Delta MV_2 + (a_4 – a_3)*\Delta MV_1 + (a_5 - a_4)*\Delta MV_0} \qquad (8.14d)$$

For convenience, let us define a new set of model coefficients b_i as follows:

$$b_1 = a_1 \quad b_2 = a_2 – a_1 \quad b_3 = a_3 – a_2 \quad b_4 = a_4 – a_3 \quad b_5 = a_5 – a_4 \qquad (8.15)$$

We also write

$$\partial CV_i = CV_i – CV_{i-1} \qquad (8.16)$$

which applies to any slice of continuous CV response curve with two neighboring CV values, CV_i and CV_{i-1}. Eq. (8.16) is different from Eq. (8.11),

$$\Delta CV_i = CV_i – CV_0 \qquad (8.11)$$

which assumes a continuous CV response curve from CV_0 to CV_i.

Applying Eqs. (8.15) and (8.16) to Eqs. (8.11a), (8.14a) to (8.14d) gives the following “impulse form” of our model equations:

$$\begin{aligned}
əCV_1 &= b_1*\Delta MV_0 \\
əCV_2 &= b_1*\Delta MV_1 + b_2 * \Delta MV_0 \\
əCV_3 &= b_1*\Delta MV_2 + b_2 * \Delta MV_1 + b_3 *\Delta MV_0 \\
əCV_4 &= b_1*\Delta MV_3 + b_2 * \Delta MV_2 + b_3 *\Delta MV_1 + b_4 *\Delta MV_0 \\
əCV_5 &= b_1*\Delta MV_4 + b_2 * \Delta MV_3 + b_3 *\Delta MV_2 + b_4 *\Delta MV_1 + b_5 *\Delta MV_0
\end{aligned} \tag{8.17}$$

We write the resulting “impulse form” of our dynamic matrix model equation as follows:

$$\begin{bmatrix} əCV_1 \\ əCV_2 \\ əCV_3 \\ əCV_4 \\ əCV_5 \end{bmatrix} = \begin{bmatrix} b_1 & 0 & 0 & 0 & 0 \\ b_2 & b_1 & 0 & 0 & 0 \\ b_3 & b_2 & b_1 & 0 & 0 \\ b_4 & b_3 & b_2 & b_1 & 0 \\ b_5 & b_4 & b_3 & b_2 & b_1 \end{bmatrix} \begin{bmatrix} \Delta MV_0 \\ \Delta MV_1 \\ \Delta MV_2 \\ \Delta MV_3 \\ \Delta MV_4 \end{bmatrix} \tag{8.18}$$

In a matrix form, Eq. (8.18) becomes

$$\mathbf{əCV = B * \Delta MV} \tag{8.19}$$

Eq. (8.19) represents <u>*the finite impulse response (FIR) form*</u> of the linear model equation for dynamic matrix control. We can extend Eq. (8.19) to allow more than one manipulated variable to vary at the same time to give Eq. (8.20).

$$\begin{bmatrix} əCV_1 \\ əCV_2 \\ əCV_3 \\ əCV_4 \\ əCV_5 \end{bmatrix} = \begin{bmatrix} \Delta MV_{1,1} & 0 & 0 & 0 & 0 & \Delta MV_{2,1} & 0 & 0 & 0 & 0 \\ \Delta MV_{1,2} & \Delta MV_{1,1} & 0 & 0 & 0 & \Delta MV_{2,2} & \Delta MV_{2,1} & 0 & 0 & 0 \\ \Delta MV_{1,3} & \Delta MV_{1,2} & \Delta MV_{1,1} & 0 & 0 & \Delta MV_{2,3} & \Delta MV_{2,2} & \Delta MV_{2,1} & 0 & 0 \\ \Delta MV_{1,4} & \Delta MV_{1,3} & \Delta MV_{1,2} & \Delta MV_{1,1} & 0 & \Delta MV_{2,4} & \Delta MV_{2,3} & \Delta MV_{2,2} & \Delta MV_{2,1} & 0 \\ \Delta MV_{1,5} & \Delta MV_{1,4} & \Delta MV_{1,3} & \Delta MV_{1,2} & \Delta MV_{1,1} & \Delta MV_{2,5} & \Delta MV_{2,4} & \Delta MV_{2,3} & \Delta MV_{2,2} & \Delta MV_{2,1} \end{bmatrix} \begin{bmatrix} b_{1,1} \\ b_{1,2} \\ b_{1,3} \\ b_{1,4} \\ b_{1,5} \\ b_{2,1} \\ b_{2,2} \\ b_{2,3} \\ b_{2,4} \\ b_{2,5} \end{bmatrix} \tag{8.20}$$

To identify the model coefficients *b_{i,j}* or matrix ***B***, we convert Eq.(8.20) to a residual form involving the error residual $r_{i,j}$:

$$\begin{bmatrix} əCV_1 \\ əCV_2 \\ əCV_3 \\ əCV_4 \\ əCV_5 \end{bmatrix} - \begin{bmatrix} \Delta MV_{1,1} & 0 & 0 & 0 & 0 & \Delta MV_{2,1} & 0 & 0 & 0 & 0 \\ \Delta MV_{1,2} & \Delta MV_{1,1} & 0 & 0 & 0 & \Delta MV_{2,2} & \Delta MV_{2,1} & 0 & 0 & 0 \\ \Delta MV_{1,3} & \Delta MV_{1,2} & \Delta MV_{1,1} & 0 & 0 & \Delta MV_{2,3} & \Delta MV_{2,2} & \Delta MV_{2,1} & 0 & 0 \\ \Delta MV_{1,4} & \Delta MV_{1,3} & \Delta MV_{1,2} & \Delta MV_{1,1} & 0 & \Delta MV_{2,4} & \Delta MV_{2,3} & \Delta MV_{2,2} & \Delta MV_{2,1} & 0 \\ \Delta MV_{1,5} & \Delta MV_{1,4} & \Delta MV_{1,3} & \Delta MV_{1,2} & \Delta MV_{1,1} & \Delta MV_{2,5} & \Delta MV_{2,4} & \Delta MV_{2,3} & \Delta MV_{2,2} & \Delta MV_{2,1} \end{bmatrix} \begin{bmatrix} b_{1,1} \\ b_{1,2} \\ b_{1,3} \\ b_{1,4} \\ b_{1,5} \\ b_{2,1} \\ b_{2,2} \\ b_{2,3} \\ b_{2,4} \\ b_{2,5} \end{bmatrix} = \begin{bmatrix} r_{1,1} \\ r_{1,2} \\ r_{1,3} \\ r_{1,4} \\ r_{1,5} \\ r_{2,1} \\ r_{2,2} \\ r_{2,3} \\ r_{2,4} \\ r_{2,5} \end{bmatrix} \tag{8.21}$$

We typically find the values of the model coefficients $b_{i,j}$ or matrix ***B,*** by minimizing the sum of the squared residual error terms:

$$Min \sum r_{i,j}^2 = Min\,(r_{1,1}^2 + r_{1,2}^2 + r_{1,3}^2 + r_{1,4}^2 + r_{1,5}^2 + r_{2,1}^2 + r_{2,2}^2 + r_{2,3}^2 + r_{2,4}^2 + r_{2,5}^2) \quad (8.22)$$

To summarize, the finite impulse response (FIR) model identification procedure discussed so far has incorporated effective means to handle several practical issues for model-predictive control in the real world: (1) The data slicing permits the use of discontinuous MV-CV response curves with missing MV input data and CV instrument failure; (2) The procedure works with the incremental or "delta" world (∂CV_i or ΔCV_i), thus not requiring the process to be at a steady state; and (3) The approach allows for continuous updates of model coefficients when changing multiple manipulated variables at the same time. These features represent significant advances of the finite impulse-response (FIR) model over the step-response modeling approach.

8.1.4 Model Evaluation and Useful Tools

In our hands-on workshop WS8.1, Section 8.2, we will go through the details of applying several tools to evaluate how accurate our model prediction is and how robust our model response is to disturbances Section 8.2.9 demonstrates uncertainty and correlation analysis of the model, and Section 8.2.11 illustrates a five-step procedure to collinearity analysis of the model. In this section, we introduce some basic concepts and tools relating to collinearity analysis.

8.1.4a Relative Gain Array (RGA)

We introduce the concept of relative gain array (RGA) [8,9,10] by considering the liquid level in a tank shown in Figure 8.23.

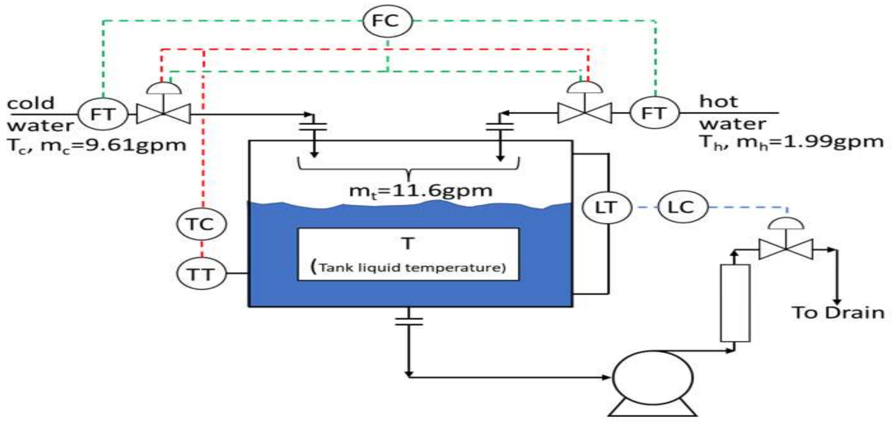


Figure 8.23 Mixing of a hot water stream with a cold water stream in a tank

Here, we control the tank temperature T and total tank liquid flow m_t (GPM) by manipulating the inlet flow rates of cold water m_c (GPM) and hot water m_h (GPM). We assume that the liquid level setpoint corresponds to 50% tank-full level with m_t = 11.6 GPM, and the temperature setpoint is T=24.4°C. The steady-state flow rates are m_c = 9.61 GPM, and m_h = 1.99 GPM.

We wish to find a proper pairing of two control variables (T and m_t) with two manipulated variables (m_c and m_h). In other words, we are interested in the proper pairing of a simple process with two control variables (T and m_t) and two manipulated variables (m_c and m_h). See Figure 8.24 for a simple representation of the current example.

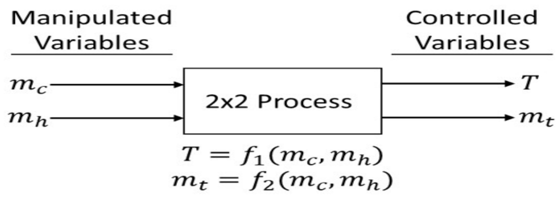


Figure 8.24 A simplified diagram of a 2x2 water-mixing process

The corresponding balance equations are:

$$T = f_1(m_c, m_h) = \frac{m_c T_c + m_h T_h}{m_c + m_h} \qquad (8.23)$$

$$m_t = f_2(m_c, m_h) = m_c + m_h \qquad (8.24)$$

Around steady state, we write ΔT and Δm_t in terms of *open-loop gains* K_{ij}:

$$\Delta T = \left(\frac{\partial T}{\partial m_c}\right)_{m_h} \Delta m_c + \left(\frac{\partial T}{\partial m_h}\right)_{m_c} \Delta m_h = K_{11}\Delta m_c + K_{12}\Delta m_h \qquad (8.25)$$

$$\Delta m_t = \left(\frac{\partial m_t}{\partial m_c}\right)_{m_h} \Delta m_c + = \left(\frac{\partial m_t}{\partial m_h}\right)_{m_c} \Delta m_h = K_{21}\Delta m_c + K_{22}\Delta m_h \qquad (8.26)$$

By substituting Eqs. (8.23) and (8.24) into Eqs. (8.25) and (8.26), we find:

$$K_{11} = [\, m_h (T_c - T_h)\,]/\, m_t^2 \qquad K_{12} = = [\, m_c (T_h - T_c)\,]/\, m_t^2 \qquad K_{21} = K_{22} = 1 \qquad (8.27)$$

In addition to open-loop gains, we introduce closed-loop gains K'_{ij} defined as:

$$K'_{11} = \left(\frac{\partial T}{\partial m_c}\right)_{m_h} \qquad K'_{12} = \left(\frac{\partial T}{\partial m_h}\right)_{m_c} \qquad (8.28)$$

$$K'_{21} = \left(\frac{\partial m_t}{\partial m_c}\right)_{m_h} \qquad K'_{22} = \left(\frac{\partial m_t}{\partial m_h}\right)_{m_c}$$

Basically, K'_{11} represents a measure of how manipulated variable m_c affects control variable T if control variable m_t is held constant ($\Delta m_t = 0$) and under closed-loop control. Specifically, when $\Delta m_t = 0$, Eq. (8.26) gives

$$0 = K_{21}\Delta m_c + K_{22}\Delta m_h \;\text{->}\qquad \Delta m_h = -\frac{K_{21}}{K_{22}}\,\Delta m_c \qquad (8.29)$$

Substituting Eq. (8.29) into Eq. (8.25) gives

$$\Delta T = \left(K_{11} - \frac{K_{11}K_{22} - K_{12}K_{21}}{K_{22}}\right)\Delta m_c \qquad (8.30)$$

When control variable m_t is held constant ($\Delta m_t = 0$)， Eq. (8.30) gives the relationship for the closed-loop gain K'_{11} :

$$K'_{11} = \left(\frac{\partial T}{\partial m_c}\right)_{m_h} = K_{11} - \frac{K_{11}K_{22} - K_{12}K_{21}}{K_{22}} \qquad (8.31)$$

Likewise, we can develop the following expressions [10, pp. 554-556]:

$$K'_{12} = \left(\frac{\partial T}{\partial m_h}\right)_{m_c} = \frac{K_{12}K_{21}-K_{11}K_{22}}{K_{21}} \tag{8.32}$$

$$K'_{21} = \left(\frac{\partial m_t}{\partial m_c}\right)_{m_h} = \frac{K_{12}K_{21}-K_{11}K_{22}}{K_{12}} \tag{8.33}$$

$$K'_{22} = \left(\frac{\partial m_t}{\partial m_h}\right)_{m_c} = \frac{K_{11}K_{22}-K_{12}K_{21}}{K_{12}} \tag{8.34}$$

The ratio of K_{ij} to K'_{ij} is called *a relative gain*, denoted by λ_{ij} :

$$\lambda_{ij} = \frac{K_{ij}}{K'_{ij}} \tag{8.35}$$

Based on Eqs. (8.27), and the defining relationships for Eqs. (8.27) and (8.31) to (8.34), we can write *a relative gain array (RGA)* as follows (with m_h= 1.99GPM, m_c=9.61GPM, and m_t=11.6 GPM):

$$\lambda = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix} = \begin{bmatrix} m_h/m_t & m_c/m_t \\ m_c/m_t & m_h/m_t \end{bmatrix} = \begin{array}{c c c} & m_c & m_h \\ T & 0.172 & 0.828 \\ m_t & 0.828 & 0.172 \end{array} \tag{8.36}$$

RGA has seveal useful properties [8,9,10] that help in choosing a specific manipulated variable that has the most impact on a given control variable.

- *Property 1: The rows and columns of the RGA sum to 1.0.*
- *Property 2: Always pair the manipulated and control variables on positive RGA elements that are closest to 1.0.*
- *Property 3: Pairing on negative RGA elements results in either an unstable system or in an inverse responding system (a system that initially responds in the opposite direction to its final steady-state response when its input is changed).*

In Eq. (8.36), the relative gain element pairing T and m_h, and pairing m_t and m_c, 0.828, is close to 1.0. Property 2 suggests that T (tank liquid temperature) should be controlled by manipulating m_h (hot water flow rate), and m_t (total tank liquid flow rate, initially at 11.6 GPM, and hence the liquid tank level) should be controlled by manipulating m_c (cold water flow rate, 9.61 GPM). This pairing is consistent with physical intuitions, as a higher temperature difference of the hot water mean that hot water flow can change the tank liquid temperature faster than cold water flow, and the larger cold water flow can change the tank liquid level quicker than the smaller hot water flow.

Next, we wish to extend the concept of RGA for application to *the steady-state gain matrix*, such as Eqs. (8.3) and (8.4). In practice, the steady-state gain matrix is typically *not* a square matrix with an equal number of independent (manipulated, and feedforward/disturbance) variables, and dependent (controlled) variables. To handle this situation, DMC3 calculates the RGAs for all 2x2 submatrices in the model and highlights any issues.

Bristol [8], McAvoy [9, pp. 31-33], and Smith and Corripio [10, pp. 561-562] explain how to develop the RGA from an n x n steady-state gain matrix, denoted by **K**, based on matrix operations. Specifically, we find the RGA by obtaining the transpose of the inverse of the steady-state gain matrix [that is, $(\mathbf{K}^{-1})^T$], and multiply each term of the resulting matrix by the corresponding term in the original matrix, **K**. The terms thus obtained are the relative gains. Let us illustrate this calculation procedure by an example from McAvoy [9, pp. 31-33]. We consider the following steady-state gain matrix relationship:

$$\mathbf{CV} = \mathbf{K} * \mathbf{MV} \tag{8.37}$$

$$\begin{bmatrix} CV_1 \\ CV_2 \\ CV_3 \end{bmatrix} = \begin{bmatrix} 2.662 & 8.351 & 8.351 \\ 0.3816 & -0.5586 & -0.5586 \\ 0 & 11.896 & -0.3511 \end{bmatrix} * \begin{bmatrix} MV_1 \\ MV_2 \\ MV_3 \end{bmatrix} \tag{8.38}$$

There are many online matrix calculators (e.g., https://matrixcalc.org/en/) to calculate the inverse and transpose. We find:

$$\mathbf{K}^{-1} = \begin{bmatrix} 0.1195 & 1.787 & 0 \\ 2.341x10^{-3} & -0.01633 & 0.08165 \\ 0.07931 & -0.5532 & -0.08165 \end{bmatrix} \tag{8.39}$$

$$(\mathbf{K}^{-1})^T = \begin{bmatrix} 0.1195 & 2.341x10^{-3} & 0.07931 \\ 1.787 & -0.01633 & -0.5532 \\ 0 & 0.08165 & -0.08165 \end{bmatrix} \tag{8.40}$$

Multiplying each term of the resulting matrix $(\mathbf{K}^{-1})^T$, Eq. (8.40), by the corresponding term in the original matrix, **K,** Eq. (8.38), is a matrix operation called *Hadamard product*, for which online calculators are available (e.g., https://keisan.casio.com/exec/system/15157205321124). We find the resulting RGA as:

$$\lambda = [\,(\mathbf{K}^{-1})^T * \mathbf{K}]_{\text{Hadamard}} = \begin{bmatrix} 0.318 & 0.0195 & 0.663 \\ 0.682 & 0.00913 & 0.309 \\ 0 & 0.971 & 0.0287 \end{bmatrix}$$

We note that the Hadamard product matrix elements result from multiplying each term of Eq. (8.40) by the corresponding term in the original **K** matrix in Eq. (8.38); for example,

$$\lambda_{11} = 2.662 \text{ x } 0.1195 = 0.318 \qquad \lambda_{12} = 8.351 \text{ x } 2.341x10^{-3} = 0.0195$$

8.1.4b Ill-Conditioned Model Matrices and Collinear System

As a part of model evaluation, DMC3 applies a collinearity analysis tool to identify and repair ill-conditioned model matrices [11]. This section introduces the *concepts of conditioned number and collinear system* and relates them to the RGA.

For an n x n steady-state gain matrix **K**, we may decompose it into a product of three matrices:

$$\mathbf{K} = \mathbf{U} * \lambda * \mathbf{V}^T \tag{8.41}$$

where **U** is n x k, λ is k x k, and **V** is n x k. Matrix λ is diagonal, with elements λ_1, λ_1… λ_k being the positive square roots of $\lambda_1^2, \lambda_2^2, \lambda_3^2$ …. , which are nonzero eigenvalues of $\mathbf{K}^T * \mathbf{K}$ or of $\mathbf{K} * \mathbf{K}^T$. The values λ_1, λ_1… λ_k are called the *singular values* of matrix **K**. This matrix decomposition process is called *singular value decomposition (SVD)*. Section A.2.5.3 of Appendix A, “Matrix Algebra in Multivariate Data Analytics and Model-Predictive Control”, gives the details of SVD, including the meaning of matrices **U** and **V**.

Let us use the SVD of three simple matrices to illustrate the concept of *condition number* and an *ill-conditioned system*. We use an online SVD calculator to obtain the numerical results (e.g., https://keisan.casio.com/exec/system/15076953160460).

(1) **K** = $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ = **U** *⋋* **V**^{T} **=** $\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$

The ⋋ matrix is diagonal, and its diagonal elemnts (λ_1, λ_1… λ_k) are singular number. The condition number is the ratio of th largest diagonal element to the smallest diagonal element. For case (1), the condition numer is (1/1) or 1.0.

(2) **K** = $\begin{bmatrix} 1 & 0.96 \\ 0.96 & 1 \end{bmatrix}$ = **U** *⋋* **V**^{T} **=** $\begin{bmatrix} -0.7071 & -0.7071 \\ 0.7071 & -0.7071 \end{bmatrix} * \begin{bmatrix} 0.041 & 0 \\ 0 & 1.96 \end{bmatrix} * \begin{bmatrix} -0.7071 & 0.7071 \\ -0.7071 & -0.7071 \end{bmatrix}$

For case (2), the condition numer is (1.96/0.04) or 49.

(3) **K** = $\begin{bmatrix} 1 & 0.96 \\ 0.96 & 1 \end{bmatrix}$ = **U** *⋋* **V**^{T} **=** $\begin{bmatrix} 0.7071 & -0.7071 \\ -0.7071 & -0.7071 \end{bmatrix} * \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} * \begin{bmatrix} 0.7071 & -0.7071 \\ -0.7071 & -0.7071 \end{bmatrix}$

For case (3), the condition numer is (2/0) or ∞.

McAvoy [9, p. 181] suggests that a gain matrix with *a condition number close to, greater than or equal to 50 indicates the system to be nearly singular or ill-conditioned*. This includes cases (2) and (3) above.

In a patent assigned to AspenTech, Zheng et. al [11] define a process model with an n x m steady-state gain matrix to be collinear by looking at the maximum and minimum singular values resulting from SVD of the gain matrix. In particular, we consider a square (n = m) submatrix of the gain matrix, and define two terms: (1) *Not collinear*: If the submatrix has a rank of m (see Section A.2.1, Appendix A for rank), then the given system has a full rank and the gain matrix is "*not collinear*"; (2) *Collinear or perfectly collinear*: If the submatrix has a rank smaller than m, and the singular value λ_m=0, then the system is "*collinear or perfectly collinear*".

The collinearity analysis tool within DMC3 identifies and repairs ill-conditioned model matrices [11]. When dealing with collinearity, we can focus on square submatrices because, if an n x m matrix is collinear (when n > m), then all its m x m submatrices must be collinear [11]. In particular, the tool calculates the RGA for all 2 x 2 submatrices in the model and highlights possible issues.

DMC3 suggests the following RGA thresholds for each MV-CV pair:

RGA =1	Ideal performance; complete correlation
RGA <5	Good correlation between the MV-CV pair
RGA <8	Reasonable and acceptable correlation between the MV-CV pair
RGA >8	Possible inconsistent gain; collinearity issue
RGA >20	Nearly collinear system; review and repair

Section 8.2.11 gives a hands-on workshop for the model evaluation by collinearity analysis,

8.1.5 Open-Loop Prediction, Prediction Error Filtering and Prediction Update for Steady-State Variables

In Section 8.1.2a(3), we presented an example of the open-loop prediction for a fired heater model and introduced the concept of continuing reconciliation of model-based predictions to the process measurements and feedback correction to update the model predictions to the future that is one of the three sources of benefits of APC. In Section 8.2.12, we demonstrate this open-loop prediction in our hands-on workshop. Here, we look at three types of prediction errors that are calculated on each DMC3 controller execution which we have applied in real industrial projects.

(1) Prediction Error (PREDER)

The prediction error, also called model bias, is the difference between the model prediction and the actual measured CV value at each controller execution:

$$\text{Prediction error (Model bias)} = CV_{pred} - CV_{actual} \tag{8.42}$$

We use the value of this error to shift the future CV prediction vector up or down to match the current CV value, as illustrated previously in Figures 8.10 and 8.11 for the online feedback correction of CV prediction based on measured value.

(2) Accumulated Prediction Error (ACPRER)

Referring to Figure 8.25, we define the accumulated prediction error as the integral of the time-dependent error E(t) from the last prediction initialization to the current time:

$$\text{Accumulated prediction error} = \int_0^t E(t)dt \tag{8.43}$$

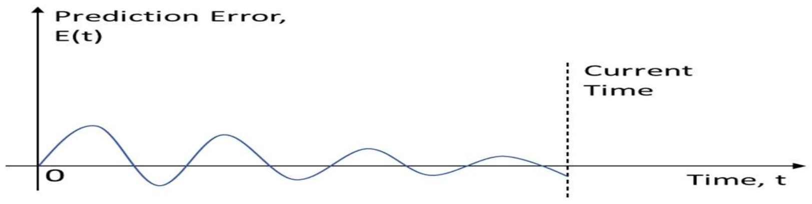


Figure 8.25 An illustration of accumulated prediction error

This integrated prediction error represents the ultimate bias that must be added to the model predicted value of a dependent variable to match its process response. In general, we prefer to monitor the accumulated prediction error, instead of the prediction error, to get a better idea of time-correlated errors.

(3) Average Prediction Error (AVPRER) and Prediction Error Filtering

The concept of average prediction error is related to the noise or model-plant mismatch that affects the controller negatively. Values of average prediction error on the same order of magnitude as the noise band of the CV measurement suggest that prediction errors are caused primarily by measurement noise, which could lead to excessive MV movement and possibly valve wear. Values of average prediction error outside the noise band indicate a potential model-plant mismatch, and significant model-plant mismatch could cause cycling or instability in CV response. Additionally, unmeasured disturbances could lead to unexpected MV moves. To mitigate these effects, we typically apply filtering or smoothing to the prediction error.

The average prediction error is a “filtered” value of the absolute value of the prediction error or model bias. The typical filter used is *a first-order exponential filter* with a filter factor equal to 0.8 to 0.99 (DMC3 use a filter factor of 0.965), which is set to 0.0 upon filter initialization. The reader may refer to [12] for an introduction to exponential filter. We note that an exponential filter is also called an “exponentially weighted moving average (EWMA) filter”, or just “exponential moving average (EMA) filter”.

Nonlinear controllers in DMC3 uses *an extended Kalman filter algorithm* [13] for prediction error or model bias filtering.

(4) Prediction Update for Control Variable Values

In Section 8.1.2a(3), Figures 8.9 to 8.11, we illustrated the “preceding” process for prediction update to control variable values: (1) update the CV predictions from the previous controller execution cycle based on the changes in MVs; (2) compute the prediction errors for CVs; (3) shift the CV predictions by prediction errors to make the current CV predictions match the current CV measurements; and (4) do this online correction for each CV at the beginning of each controller execution cycle.

8.1.6 Concepts and Parameters in Steady-State Economic Optimization and Dynamic Controller Simulation

In Section 8.1.2b, we illustrated the steady-state (SS) economic optimization to determine the MV and CV targets to minimize cost and optimize profit for the copolymerization problem. In this section, we introduce additional concepts and parameters that are relevant to SS optimization and the subsequent dynamic controller simulation steps. These concepts and parameters are key to developing and fine-tuning both linear and nonlinear multivariable model-predictive controllers using Aspen DMCplus and Aspen DMC3. The same concepts and parameters are equally important to Aspen Nonlinear Controller for polyolefin applications.

8.1.6a Variable Limits and Feasible Solution

Figure 8.26 illustrates the concept of variable limits. Take MV as an example. *The upper and lower operating limits*, or simply *the upper and lower limits*, define the control range over which the MV may be moved by the controller. *The upper and lower validity limits* specify the prediction range over which the MV may be used for prediction. If the operating limits are set outside the validity limits, the MV is downgraded to FF status.

The upper and lower engineering limits define the commissioned range and are used to clamp the operating limits if the operating limits are set outside the engineering limits but within the validity limits. Engineering limits outside the validity limits are clamped at the validity limits without downgrading the MV to FF status.

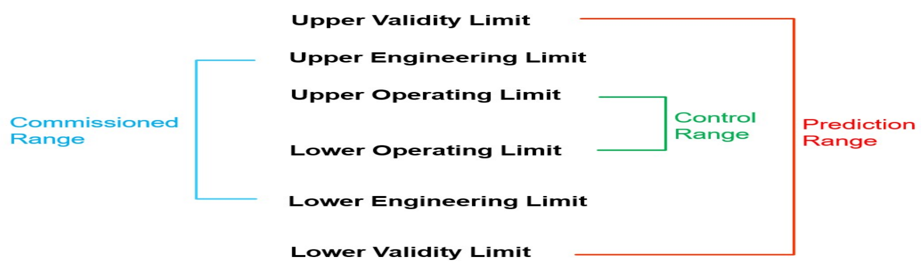


Figure 8.26 An illustration of variable limits [7]. Used with permission from Aspen Technology, Inc.

The SS (steady state) optimizer performs a validity check using the current measurement, limits, tuning parameters, and provides economic optimum MV and CV targets to the path optimizer for dynamic controller simulation. *A feasible solution* is defined as a solution where all CV steady-state targets are at or within their operating limits. We note that MV steady-state targets are always maintained within their operating limits.

How does the SS optimizer know which CV operating limits are the least important and which could be changed slightly, if necessary, to find a feasible solution? The SS optimizer uses two sets of parameters to allow the control engineer to specify the relative importance of CV operating limits: *(1) CV limit ranking,* and *(2) steady-state equal concern error (SS ECE),* which are discussed below.

8.1.6b CV Limit Ranking Method to Handle Steady-State Feasibility

Aspen DMC3 Builder assigns a relative ranking to each CV operating limit to characterize the order of priority of that limit, and the steady-state economic optimization satisfies CV limits in a ranked order. The software checks for the limit for feasibility *in order of increasing rank.* Specifically, a CV limit with a smaller numerical ranking is more important than another CV limit with a larger numerical ranking, e.g., a CV limit with rank 1 is more important than another CV limit with rank 999; and the former CV limit must not be violated, while the latter CV limit could be relaxed if appropriate.

DMC3 Builder recognizes the following possible ranks: (1) rank 0: All CV's have the same rank, have rank 0 and we consider trade-off with MV constraints (not recommended in practice); (2) rank 1-999 (see more below): all CV limits go through a standard feasibility check; (3) rank 1000: a special "soft target" constraint which is solved in the economic optimization only (not in the feasibility calculation); and (4) rank 9999: the CV limit is not used in steady-state economic optimization.

The CV limit ranking results from consulting with experienced plant operators and engineers, who typically know the relative important of each CV limit. When it is not possible to clearly define the relative ranking of a CV limit, we could consider assigning the CV limit into the following suggested rank between 1 and 999 [7]: (1) safety and environmental limits (e.g., stack NO_x emissions; safety valve controller output; heater tubeskin temperature, etc.) : rank 1 to 99; (2) integrating or ramp variable (Section 8.1.1.1c): rank 100 to 199; (3) model validity requirements (e.g., control valve outputs; column flooding limit, etc.) : rank 200 to 299; (4) product quality specifications (fractionator boiling points; product impurity specifications, etc.) : rank 300 to 399; and (5) economic optimization soft targets that cannot be uniquely defined: rank 1000.

Figure 8.27 illustrates *the CV ranking method* to handle steady-state feasibility for CV limits of different ranks. We satisfy the more important constraints (lower numerical ranks) first, while relaxing the less important constraints (larger numerical ranks) to find a feasible solution. In the figure, line B (CV2U, representing CV_2 upper operating limit, of rank 100) and line C (CV1L, representing CV_1 lower operating limit, of rank 200) are both satisfied and intersect at point F. We could find a feasible solution if we could relax the constraint of line A (CV3U, representing CV_3 upper operating limit, of rank 300) by moving line A to line A' which satisfies the feasible solution at point F. The distance between lines A and A' represents the relaxed amount required to make a constraint feasible, for which we call *a constraint give-up (ε).*

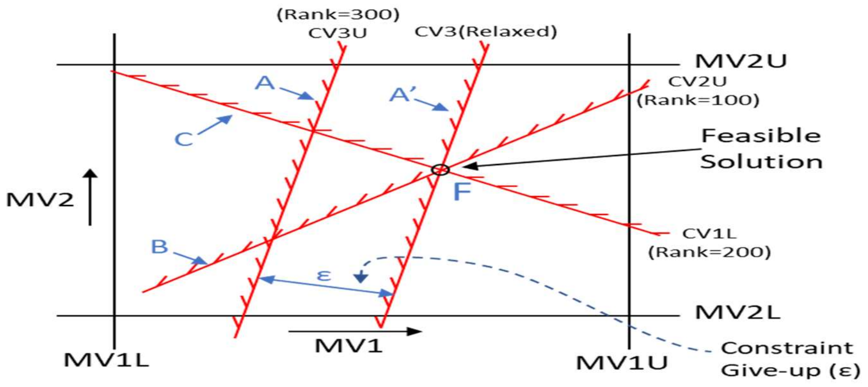


Figure 8.27 Achieving steady-state feasibility for CV limits of different ranks by satisfying constraints CV2U ($CV_{2,}$ upper operating limit of rank 100; line B) and CV1L (CV_1 lower operating limit of rank 200, line C), while relaxing the constraint of CV3U (shifting CV_3 upper operating limit of rank 300 from line A to line A').

Figure 8.28 shows another example about constraint give-up when the CV limit ranks are equal. In the figure, line B (CV2U, representing CV2 upper operating limit) and line C (CV3U, representing CV3 upper operating limit) are both satisfied and intersect at point F. We move the constraint of line A (CV1L, representing CV1 lower operating limit) to line A' which satisfies the feasible solution at point F. The distance between lines A and A' represents the relaxed amount required to make the constraint CV1L feasible. This is the constraint give-up (ε) for CV1L.

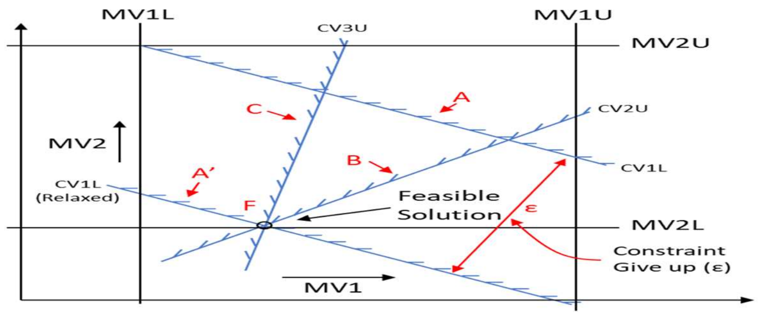


Figure 8.28 Achieving steady-state feasibility for CV limits of the same rank by satisfying constraints CV2U and CV3U, while relaxing the constraint CV1L (shifting CV1 lower operating limit from line A to line A')

8.1.6c Steady-State Equal Concern Error (SS ECE) to Handle Steady-State Feasibility

For the copolymerization problem, assuming the CV limits to be of an equal rank, what magnitude of error in each of the CVs should get an equal level of attention or concern from the control engineer? Let us consider Table 8.2, in which we quantify the CV error that is 10% of the difference between the upper and lower operating limits.

Table 8.2 Magnitude of error in each of the CVs

CV	Measurement (current value)	Lower operating limit (LPL)	Upper operating limit (UPL)	(UPL- LPL)x10% deviation
Polymer, kg/hr	23.3	0	30	3 kg/hr
Mol_Wt.	35000	34500	35500	100
T_Rx, °C	85	70	100	3.0 °C
Conc_MMA, mole fraction	0.56	0.55	0.60	0.005 mole fraction

In terms of engineering units, we see that for each CV, an error above the UPL or below the LPL with a magnitude larger than the value displayed in the last column of Table 8.2 should require the control engineer an equal level of attention or concern to take appropriate corrective action.

DMC3 Builder includes a steady-state parameter, called *steady-state equal concern error (SS ECE)*, to handle the infeasibility of potential violations of multiple CV limits of an equal rank. The SS ECE factors allow the control engineer to specify the “standard” or “reference” amount of error for a given CV. These are then used to balance movement (error) in one CV against movement (error) in another CV. A small SS ECE for a CV means that this CV has a smaller tolerance threshold to any deviation from its upper or lower operating limit, and the control engineer must give sufficient attention or concern to the resulting potential infeasibility. For example, if the SS ECE for CV1 is less than the SS ECE for CV2, and both have the same engineering unit, then satisfying the CV1 limit constraint is more important than satisfying the CV2 limit constraint.

Figure 8.29 gives an example of using SS ECE to resolve a set of infeasible CV limits. In the figure, line A (CV2L, representing CV1 lower operating limit) has a smaller steady-state ECE of SS Low Concern of 0.01, and must be satisfied. Line B (CV2U, representing CV2 upper operating limit) has a larger steady-state ECE of SS High Concern of 1, and could be relaxed. We move the less important constraint of line B to line B’ which satisfies the feasible solution at point F. The distance between lines B and B’ represents the relaxed amount required to make the constraint CV2U feasible. This is the constraint give-up (ε) for CV2L.

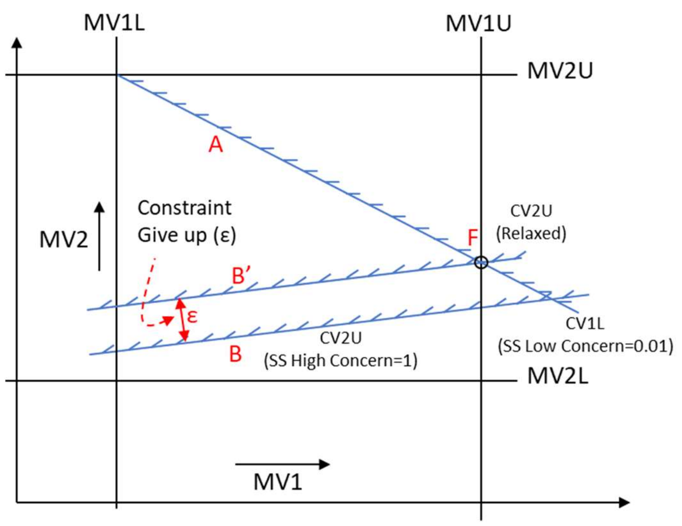


Figure 8.29 Achieving steady-state feasibility for CV limits of the same rank by satisfying constraints CV1L with a smaller SS ECE of 0.01, while relaxing the constraint CV2U of a larger SS ECE of 1 and shifting CV2 upper operating limit from line B to line B'.

To resolve a set of infeasible CV limits of an equal rank, DMC3 Builder provides two algorithms: (1) *LP (linear programming) solution; and (2) QP (quadratic programming) solution.* We consider only the LP solution here. Specifically, assume that ε_1, ε_2, ε_3, …… are the amounts of constraint give-up, illustrated in Figures 8.27 and 8.28, to make a CV limit feasible. We restrict the give-ups, ε_i, to be positive or zero (zero means that a feasible solution exists). For each constraint give-up, we assign a weight or weighting factor W_i indicating the relative importance of satisfying i-th CV limit. The LP solution includes the following linear minimization objective function plus the linear CV limit constraint:

$$\text{Min } \phi = \varepsilon_{1*} W_1 + \varepsilon_{2*} W_2 + \ldots\ldots\ldots \qquad (8.44)$$

subject to the following CV limit constraints in a vector form

$$\mathbf{CV} \le \mathbf{CV_{max}} + \boldsymbol{\varepsilon_1} \qquad (8.45)$$

$$\mathbf{CV} \ge \mathbf{CV_{min}} - \boldsymbol{\varepsilon_2} \qquad (8.46)$$

The weight or weighting factor, W_i, is a positive number, typically from 1 to 10^6; the higher its value, the more important it is to satisfy the upper or lower limit constraint for CV_i.

In applying the LP algorithm to resolve the infeasible CV limits, DMC3 Builder specifically relates the weighting factor W_i (varying from 1 to 10^6) to the corresponding SS ECE_i (varying from 1 to 10^{-6}) by the relationship:

$$\text{SS } ECE_i = 1/W_i \qquad (8.47)$$

Suppose that it is very important to satisfy the i-th CV limit by setting the weighting factor W_i to 10^6, Eq. (8.47) suggests that the corresponding SS ECE for the i-th CV limit, SS ECE_i, is 10^{-6}. We note that in doing a SS Optimizer calculation (simulation), we need only the relative values of ECEs (low concern and high concern) for all CVs, not their specific numerical values in engineering units. As such, we could specify the ECEs for CV limits as 1, 0.1, 0.01, 0.001, ..., with smaller ECE values (higher weighting factors) indicating that it is more important to satisfy the corresponding upper or lower CV limit.

For steady-state economic optimization (SS Optimizer), we need to specify both the limit ranks and the ECEs. These include: *(1) the SS Low Concern, SS Low Rank, SS High Concern and SS High Rank for each CV; and (2) Validity, Engineering and Operator Limits (Low and High) for each MV and each CV.*

8.1.6d Dynamic Equal Concern Errors for CV Limits in Dynamic Controller Simulation

Having completed the steady-state economic optimization via the SS Optimizer and identified the economic optimum, MV and CV targets, DMC3 Builder continues with dynamic controller simulation to determine a series of MV moves to drive the MV and CV to their target values through the Path Optimizer. In this step, a key tuning parameter in dynamic controller simulation is *the dynamic equal concern error, or dynamic ECE*. Basically, dynamic ECEs indicate the level of concern by the control engineer for dynamic deviations from the stead-state CV targets. As with steady-state ECEs, it is the relative values of the dynamic ECEs that determine how tight a CV is controlled to its steady-state target, and not the value of the dynamic ECE itself. We can minimize the deviation of a CV from its steady-state target by reducing the dynamic ECE. This is done at the expense of more errors on the other CVs and more movements for the MVs.

Figure 8.30 illustrates the concept of dynamic ECEs in three different regions: below the lower operating limit (LPL), above the upper operating limit (UPL), and between the LPL and UPL.

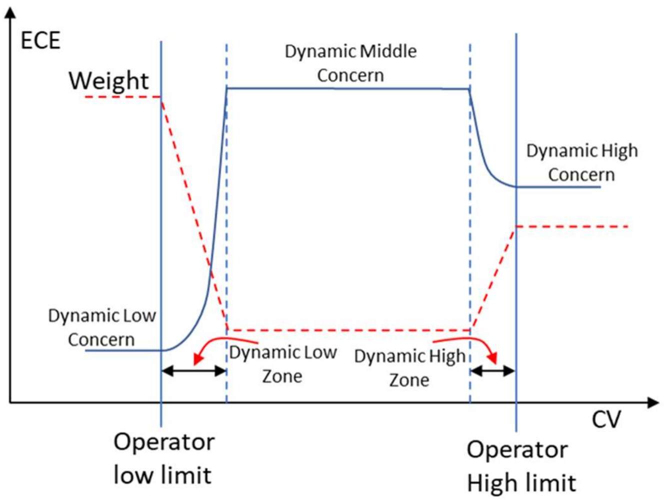


Figure 8.30 Dynamic equal concern errors for CV limits [7]. Used with permission from Aspen Technology, Inc.

First, DMC3 Builder specifies a dynamic ECE called “*Dynamic low concern*” for CV values below the “Operator low limit”, and a dynamic ECE called “*Dynamic high concern*” for CV values above the “Operator high limit”. Next, we see in the figure a narrow transition zone to the right of Operator low limit, called lower transition zone or “*Dynamic low zone*”, in which the weight (dashed line) drops and the Dynamic low concern (solid line) increases; we also see a narrow transition zone to the left of the Operator high limit, called upper transition zone or “*Dynamic high zone*”, in which the weight (dashed line) increases and the Dynamic high concern (solid line) drops. The transition zones help to avoid "chatter" when ECEs are different in the three different regions noted above and displayed in the figure. Thirdly, we see in the figure a middle region between the right boundary line of the lower transition zone or "Dynamic low zone", and the left boundary line of the upper transition zone or "Dynamic high zone". While we see a label "Dynamic middle concern" in this middle region, this ECE has no real significance and is being ignored in the DMC3 dynamic controller simulation. This follows because within this middle region, the CV value is always between its LPL and UPL, and the control engineer sees no chance for the CV to deviate from its limits.

In the DMC3 control structure of Figure 8.18, we see that the path optimizer may determine a series of CV moves to drive the MVs and CVs to their economic optimum, steady-state targets obtained by the SS optimizer. Additionally, the figure shows that the path optimizer may also determine a series of MV moves to drive a specific CV to *an external target value specified by the control engineer* (instead of the target value determined by the SS optimizer, that is, the economic optimum, steady-state target). DMC3 treats *an external target* for a CV the same as a CV constraint and includes it in the feasibility checks by the SS optimizer. Additionally, when doing a dynamic controller simulation through the path optimizer, DMC3 Builder includes a dynamic ECE for the external target, called *Dynamic Target Concern.* This is the concern associated with the dynamic move plan target for a CV. It defines how far the output can drift dynamically from its steady-state target before you get concerned. An increase in this value will allow the output more freedom to deviate dynamically from the steady-state target. A decrease will drive the output closer to the steady-state target dynamically.

To summarize, for dynamic controller simulation, we need to specify both the limit ranks and the ECEs. These include: (1) *SS Low Concern, Dynamic Low Concern, SS High Concern, Dynamic High Concern, and SS Low Rank, SS High Rank, Dynamic Low Zone, and Dynamic High Zone for each CV; (2) Validity, Engineering and Operator Limits (Low and High) for each MV and each CV; and (3) Dynamic Target Concern, if there is an external target for a specific CV*.

8.1.6e Move Suppression for MV

A key parameter for dynamic controller simulation via path optimizer is *the move suppression* for MVs. Move suppression parameter affects how aggressively the controller will move the MVs to achieve control objectives. A larger value means more suppression, i.e., less MV movement.

Figure 8.31 illustrates the trade-off between: (1) minimizing CV error from its steady-state economic optimum target by an aggressive MV movement by specifying a small move suppression; and (2) minimizing MV move size by specifying a large move suppression, resulting in increasing CV error from its SS optimization target. Figure 8.32 compares the impacts of small and large move suppression parameters on the MV move size in response to CV setpoint change from 310°C to 350°C.

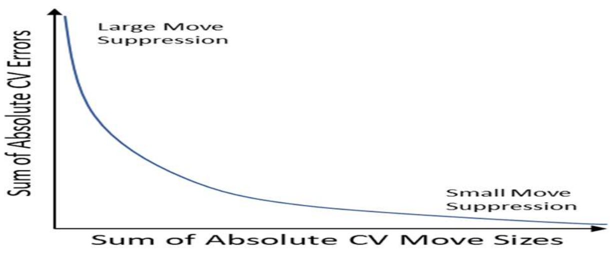


Figure 8.31 The trade-off between minimizing CV error from its steady-state economic optimum target by an aggressive MV movement by specifying a small move suppression, and minimizing MV move size by specifying a large move suppression

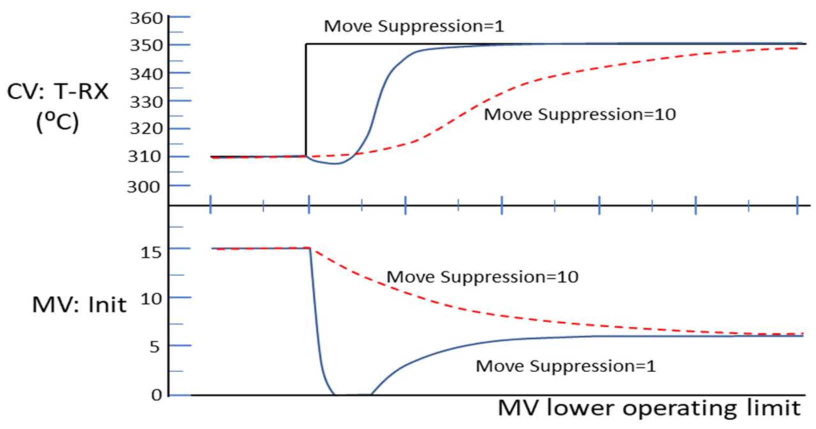


Figure 8.32 Comparing the impacts of small and large move suppression parameters on the MV move size in response to CV setpoint change from 310°C to 350°C.

There are several sources of qualitative information that can help a control engineer determine the appropriate value of move suppression to use in a dynamic controller simulation: (1) experience during step testing to develop the controller model; (2) comfort level of the control engineer for how fast an MV can be moved; (3) the capability of the exiting PID loop to track CV setpoint changes; (4) the type of disturbances for which the MV must compensate; and (5) settings for similar controllers which have demonstrated success.

Additionally, we may apply a multi-level strategy to initialize the move suppression parameters. We start by applying a move suppression value of x (say, 0.1) for a flow setpoint MV. We then specify a move suppression value of 2x (say, 0.2) for a temperature setpoint MV, and of 4x (say, 0.4) for a pressure setpoint MV and a feed rate setpoint MV. The larger values of move suppression for the pressure setpoint MV and feed rate setpoint MV imply that we do not want both the pressure and feed rate setpoints move quickly.

Finally, we note that move suppression is the most straightforward handle on dynamic controller performance. ECE tuning has a relatively narrow range where it affects dynamic controller performance, and it sometimes could give unpredictable results for disturbances affecting more than one CV at a time.

In the next section, we present a hands-on workshop to illustrate all the concepts and parameters introduced so far. We also use the workshop to demonstrate the practical tips in applying DMC3 Builder to the model-predictive control of a copolymerization process.

8.2 Workshop WS8.1 Development and Application of a Predictive Controller Model for a Copolymerization Process

8.2.1 Objective

The objective of this workshop is to teach the reader to use the DMC3 Builder for a multivariable dynamic matrix control (DMC) Project, specifically the development and applications of a predictive controller model for a solution copolymerization reactor based on data from plant step tests. We focus on the identification of a dynamic process model using the modeling tools within DMC3 Builder, and the use of the resulting predictive controller model to optimize the polymer production rate.

8.2.2 A Copolymerization Reactor

Figure 8.33 shows a simplified flowsheet of a solution copolymerization reactor system.

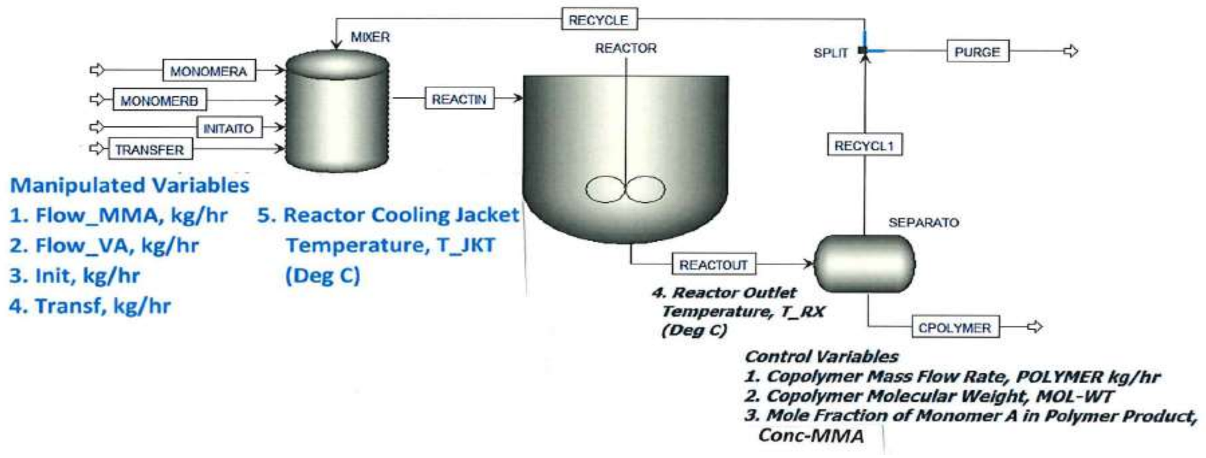
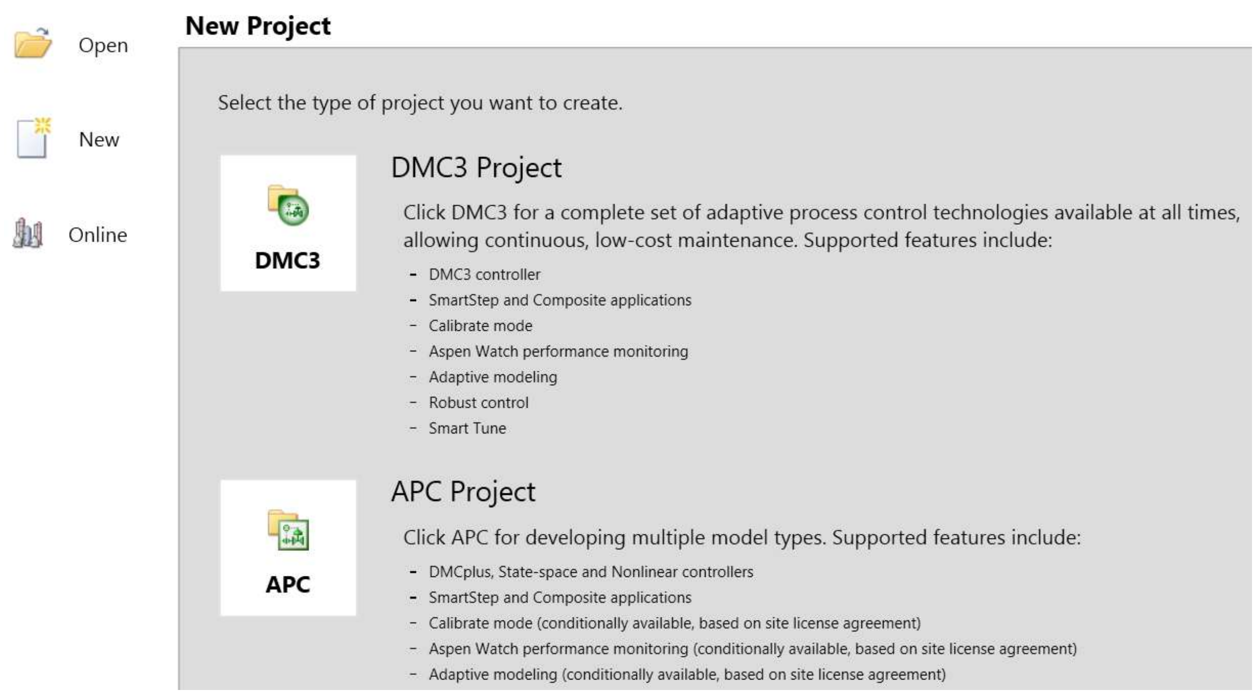

RECYCLE
MIXER
REACTOR
SPLIT
PURGE
MONOMERA
MONOMERB
INITAITO
TRANSFER
REACTIN
RECYCL1
SEPARATO
REACTOUT
CPOLYMER
Manipulated Variables
1. Flow_MMA, kg/hr
2. Flow_VA, kg/hr
3. Init, kg/hr
4. Transf, kg/hr
5. Reactor Cooling Jacket Temperature, T_JKT (Deg C)
4. Reactor Outlet Temperature, T_RX (Deg C)
Control Variables
1. Copolymer Mass Flow Rate, POLYMER kg/hr
2. Copolymer Molecular Weight, MOL-WT
3. Mole Fraction of Monomer A in Polymer Product, Conc-MMA

Figure 8.33 A simplified flowsheet of a solution copolymerization reactor system

There are two monomers, methyl methacrylate (MMA) and vinyl acetate (VA), an initiator (INITIATO), and a chain-transfer agent (TRANSF).The process is define in more detail in [31,33,23,24] Figure 8.33 shows five manipulated (independent) variables, and four control (dependent) variables.

8.2.3 Starting the DMC3 Builder Program: Creating a New Project

Start Aspen DMC3 Builder and choose “New”. Figure 8.34 illustrates the resulting screen to choose one of the two project types: (1) ***DMC project***, which includes ***<u>DMC3 controller</u>***, Aspen Watch for controller performance monitoring, and a complete set of adaptive control tools; and (2) ***APC project***, which includes DMCplus controller and **<u>nonlinear controller</u>**, Aspen Watch, and some adaptive control tools if licensed.


Open

New

Online

New Project

Select the type of project you want to create.

DMC3 Project

Click DMC3 for a complete set of adaptive process control technologies available at all times, allowing continuous, low-cost maintenance. Supported features include:

- DMC3 controller
- SmartStep and Composite applications
- Calibrate mode
- Aspen Watch performance monitoring
- Adaptive modeling
- Robust control
- Smart Tune

APC Project

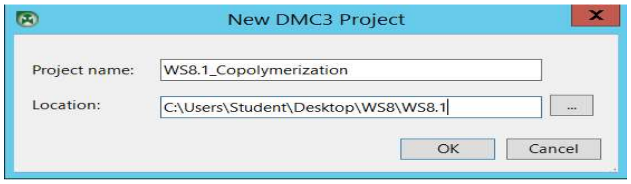
Click APC for developing multiple model types. Supported features include:

- DMCplus, State-space and Nonlinear controllers
- SmartStep and Composite applications
- Calibrate mode (conditionally available, based on site license agreement)
- Aspen Watch performance monitoring (conditionally available, based on site license agreement)
- Adaptive modeling (conditionally available, based on site license agreement)

Figure 8.34 Selection of project type and related software tools

For controlling the production rate, and product concentration and qualities (such as polymer density and melt index) of polyolefins, we recommend choosing APC project and using the *nonlinear* controller, as both polymer density and melt index have noninear dependencies on key manipulated variables.

For controlling the polymer production rate, molecular weight, and concentration of monomer in the polymer, we can use a *linear* multivariable model-predictive control, such as DMCplus controller under APC Project or its newer version, DMC3 controller under DMC3 Project. For now, we choose DMC3 Project, and complete the project name and the working folder location, as in Figure 8.35. After clicking OK, we see the interface layout of Figure 8.36.


New DMC3 Project

Project name: WS8.1_Copolymerization

Location: C:\Users\Student\Desktop\WS8\WS8.1

Figure 8.35 Specifying project name and working folder location

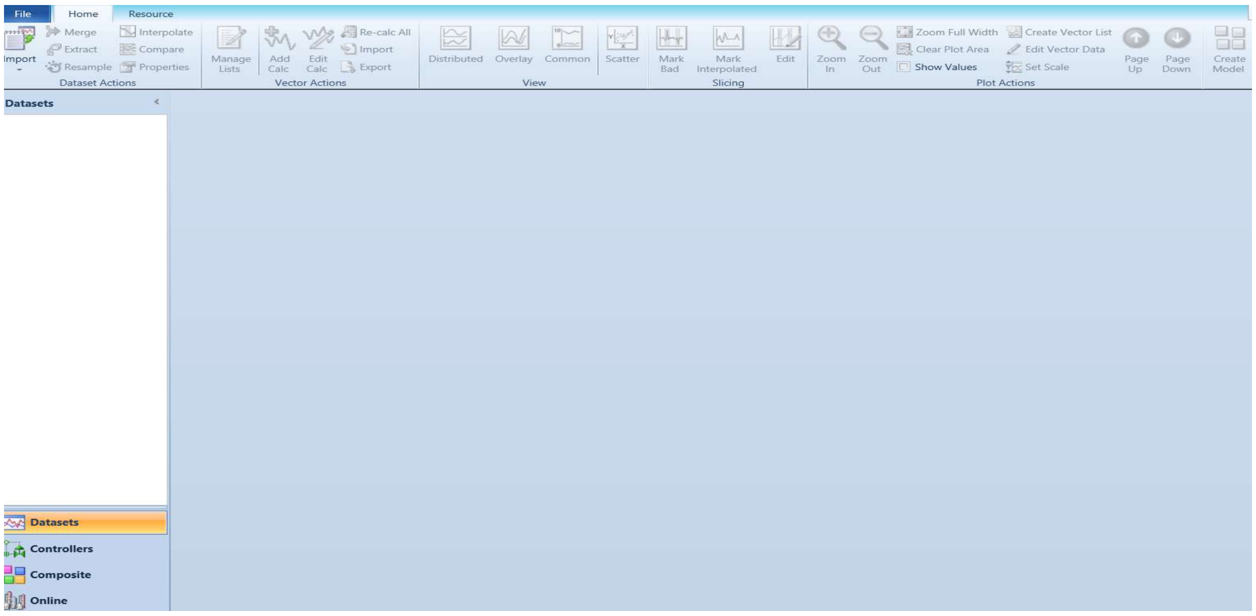
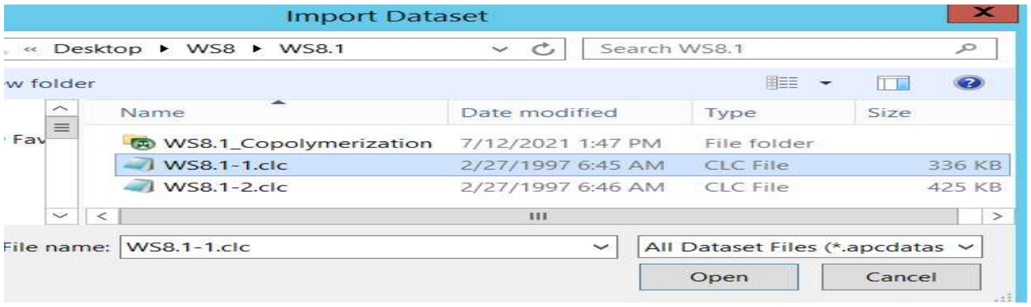


Figure 8.36 DMC3 interface layout: Tool ribbons (top), navigation workflow buttons (bottom left)- datasets, controllers, composite and online; navigation tree area (left "white" column), and workspace (middle).

8.2.4 DMC3 Builder Task One: Data Processing for Developing a Master Model- Import Process Data, Merge the Datasets and Mark and Delete Bad Data Slices

DMC3 Builder can perform six key tasks: ***(1) Master Model:*** *data processing and model identification (ID);* ***(2) Configuration:*** *configuring the steady-state optimizer and dynamic controller;* ***(3) Optimization:*** *performing the steady-state optimization;* ***(4) Simulation:*** *including five types of simulation, namely, controller, optimizer, filter, model, and preview dynamics;* ***(5) Calculations:*** *performing online calculations and transformations; and* ***(6) Deployment****: performing controller deployment*. We begin with task one, data processing to develop the master model, below

From the "Import" tool ribbon on the top left, we choose "Dataset". We then select the collect file ***WS8.1-1.clc*** within our working folder and click on the Open button. See Figures 8.37 and 8.38.

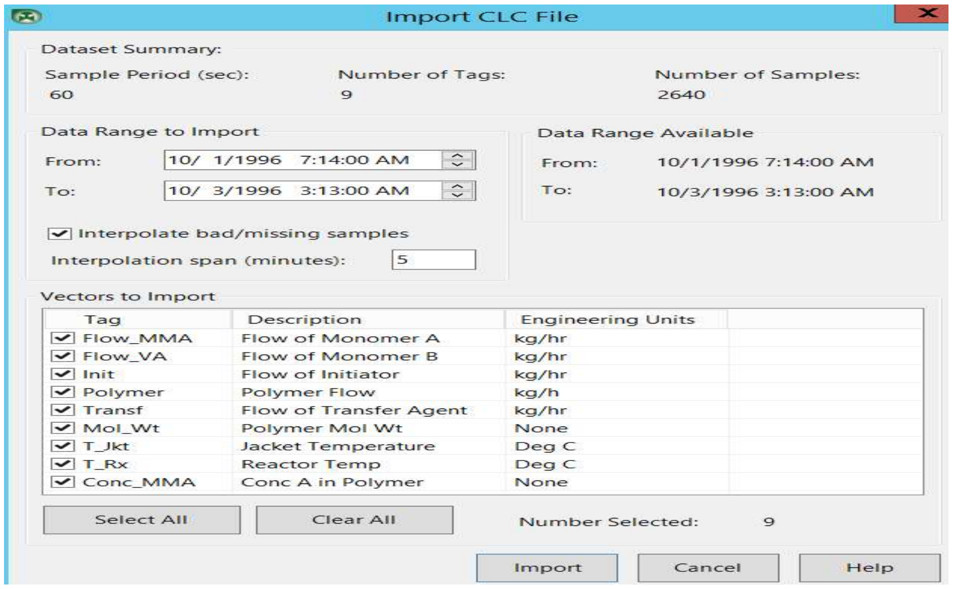


Name	Date modified	Type	Size
WS8.1_Copolymerization	7/12/2021 1:47 PM	File folder	
WS8.1-1.clc	2/27/1997 6:45 AM	CLC File	336 KB
WS8.1-2.clc	2/27/1997 6:46 AM	CLC File	425 KB

File name: WS8.1-1.clc — All Dataset Files (*.apcdatas

Figure 8.37 Important dataset, collect file ***WS8.1-1.clc***.

We see in Figure 8.38 that the first collect file has 9 tags, a sample period of 60 sec, an interpolation span of 5 min and a total of 2640 samples collected from 10/1/1996 07:14:00 to 10/3/1996 03:13:00. We click "Import" to upload the data into the project. In general, an interpolation span of 5 to 10 minutes would be sufficient for most problems.



Dataset Summary:

Sample Period (sec):	Number of Tags:	Number of Samples:
60	9	2640

Data Range to Import

From: 10/ 1/1996 7:14:00 AM

To: 10/ 3/1996 3:13:00 AM

Data Range Available

From: 10/1/1996 7:14:00 AM

To: 10/3/1996 3:13:00 AM

✓ Interpolate bad/missing samples

Interpolation span (minutes): 5

Vectors to Import

	Tag	Description	Engineering Units
✓	Flow_MMA	Flow of Monomer A	kg/hr
✓	Flow_VA	Flow of Monomer B	kg/hr
✓	Init	Flow of Initiator	kg/hr
✓	Polymer	Polymer Flow	kg/h
✓	Transf	Flow of Transfer Agent	kg/hr
✓	Mol_Wt	Polymer Mol Wt	None
✓	T_Jkt	Jacket Temperature	Deg C
✓	T_Rx	Reactor Temp	Deg C
✓	Conc_MMA	Conc A in Polymer	None

Number Selected: 9

Figure 8.38 Contents of the first collect file

Before the software import the data into the project, an “Interpolate Dataset” window shows up. We click on “Start” button to interpolate any bad and missing data slices longer than 5 minutes in duration. We then click on “Close” button to conclude the interpolation analysis. The analysis results in the message “0 of 9 vectors (variables) have been interpolated”. We do not show the screen images of this straightforward step.

Figure 8.39 displays the first dataset in the Datasets view and trend plots. The software will automatically show the first three in the view (which happen to be all manipulated variables, Flow_MMA, Flow_VA, and Init), but we choose to add the remaining two manipulated variables (Transf and T_JKT) to display. Of particular significance in the displayed plot are *the stepwise changes in all five manipulated variables* within the total duration of step tests.

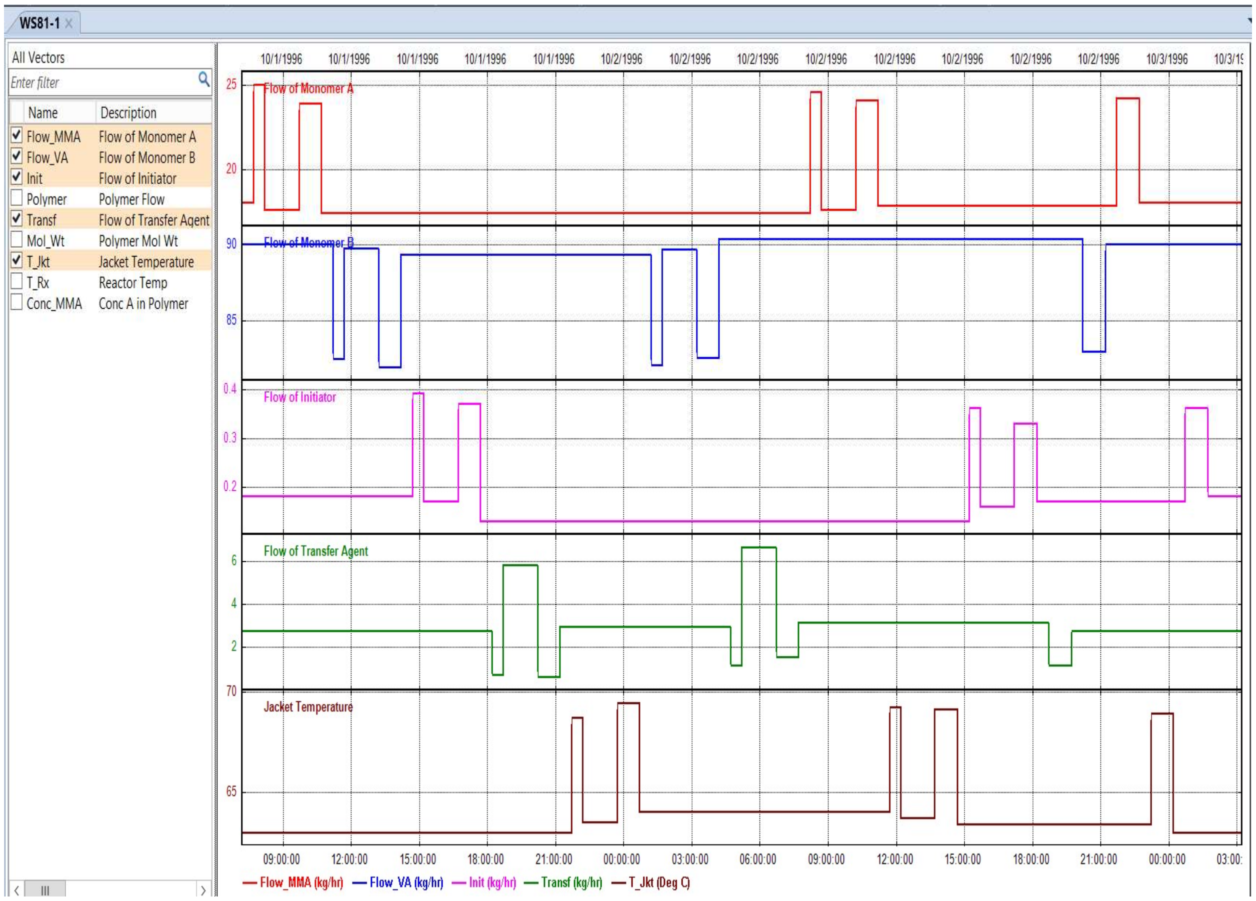
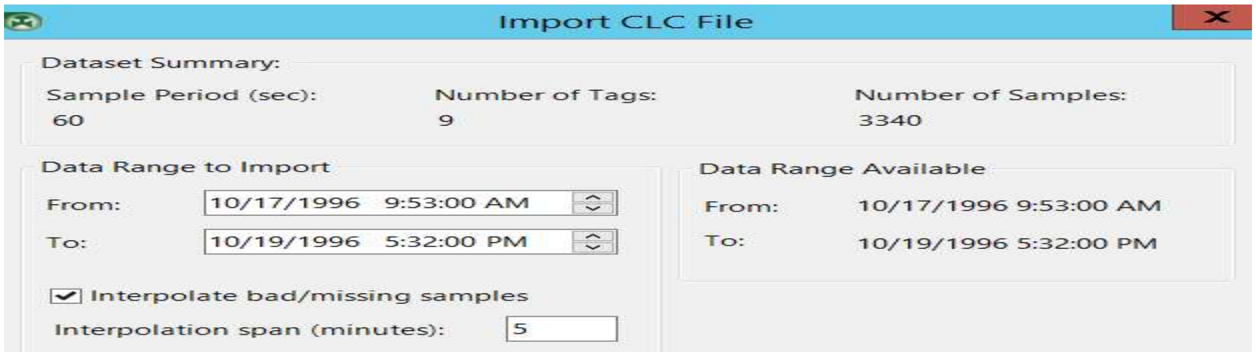


Figure 8.39 A display of the first dataset for stepwise changes in five manipulated variables during step tests

We repeat the same process to import the second collect file, ***WS8.1-2.clc***. Figure 8.40 illustrates the key features of the second collect file. Its list of vectors to import is identical to that in Figure 8.38. A display of the five manipulated variables is similar to Figure 8.39. Figure 8.41 displays *the continuous changes in all four controlled (dependent) variables within the duration of step tests.*


Import CLC File

Dataset Summary:

Sample Period (sec): 60

Number of Tags: 9

Number of Samples: 3340

Data Range to Import

From: 10/17/1996 9:53:00 AM

To: 10/19/1996 5:32:00 PM

[✓] Interpolate bad/missing samples

Interpolation span (minutes): 5

Data Range Available

From: 10/17/1996 9:53:00 AM

To: 10/19/1996 5:32:00 PM

Figure 8.40 Contents of second collect file.

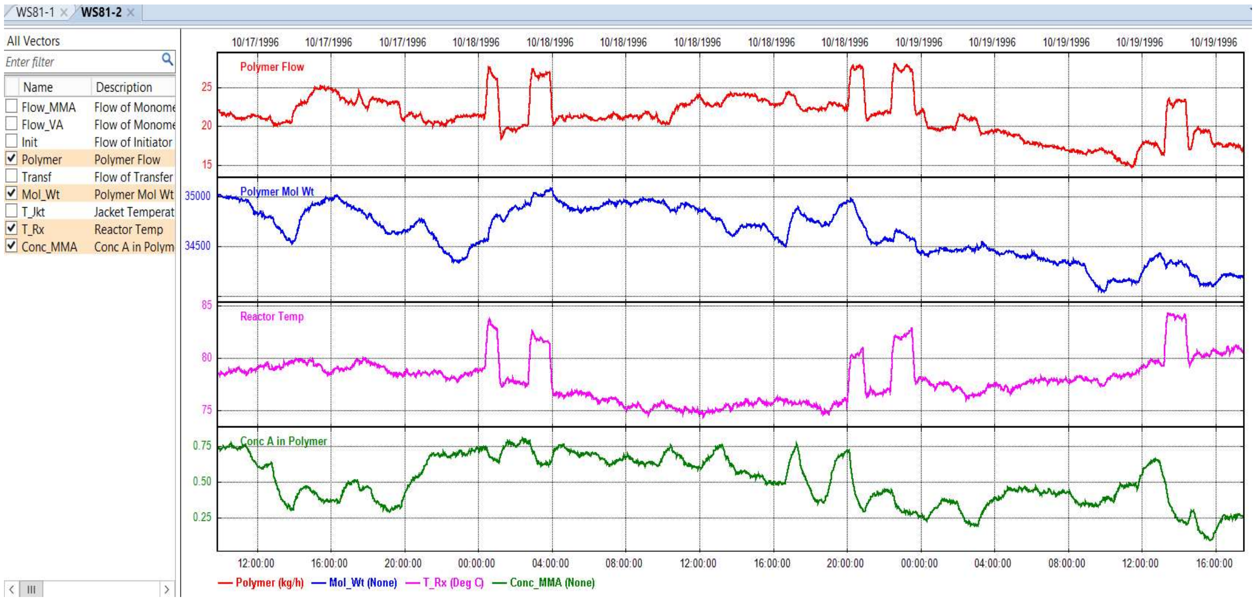
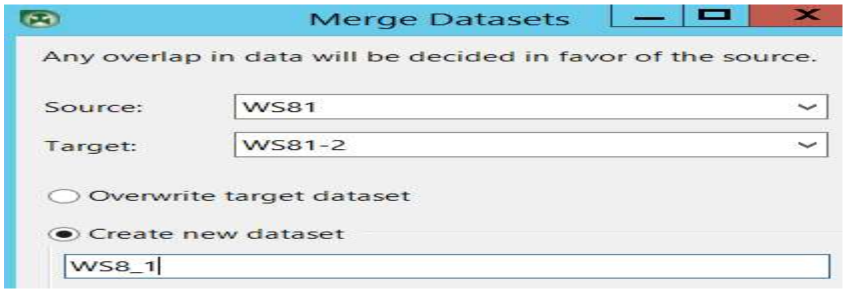


Figure 8.41 A display of the second dataset for continuous changes in four control variables during step tests

Next, we follow the path: tool ribbons -> dataset actions -> merge -> create new dataset: name - WS8_1 -> OK. See Figure 8.42.


Merge Datasets

Any overlap in data will be decided in favor of the source.

Source: WS81

Target: WS81-2

Overwrite target dataset

[✓ Create new dataset]

WS8_1

Figure 8.42 Merge two datasets into a new dataset, WS8_1

Figure 8.43 illustrates the merged datasets. We note that the software has automatically highlighted in grey the section of date and time within the duration of the dataset that contains bad/missing values. When we choose to use our mouse to highlight the grey section, it will become green and activate the data slicing tools on the top ribbon buttons. See Figure 8.44.

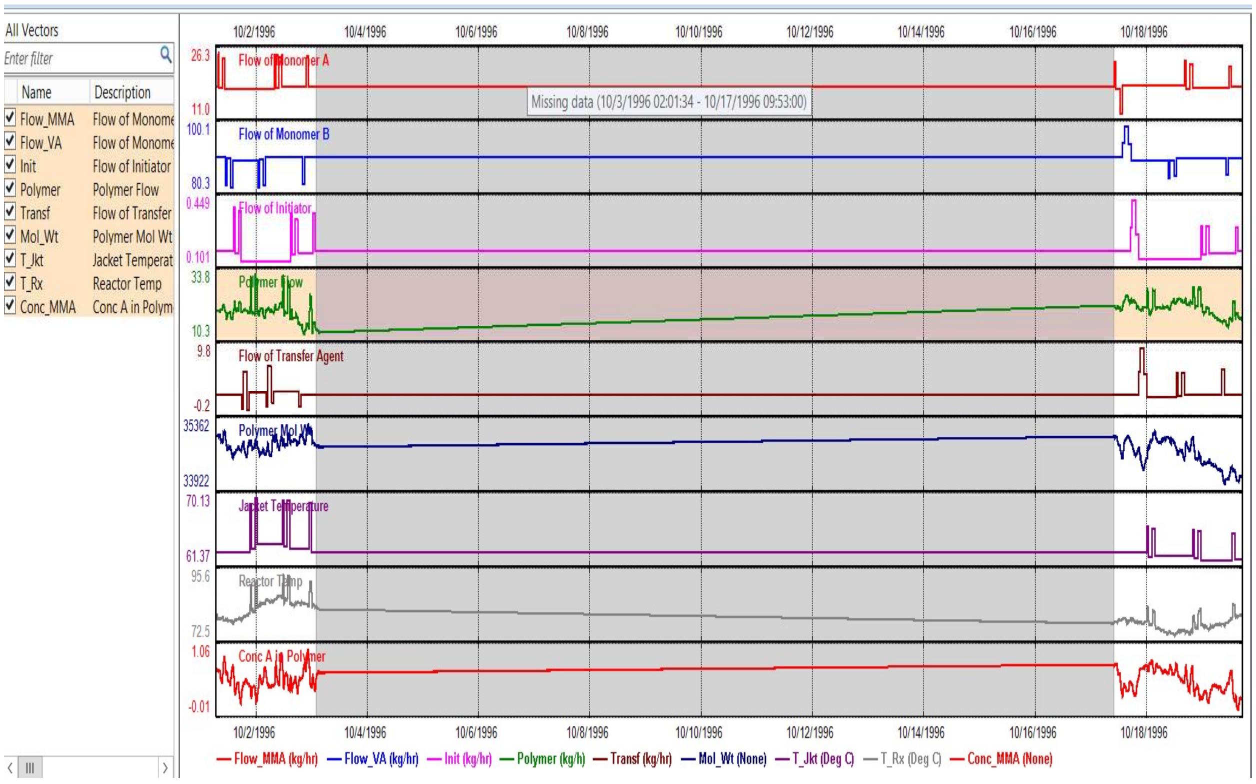


Figure 8.43 Merged dataset WS8_1

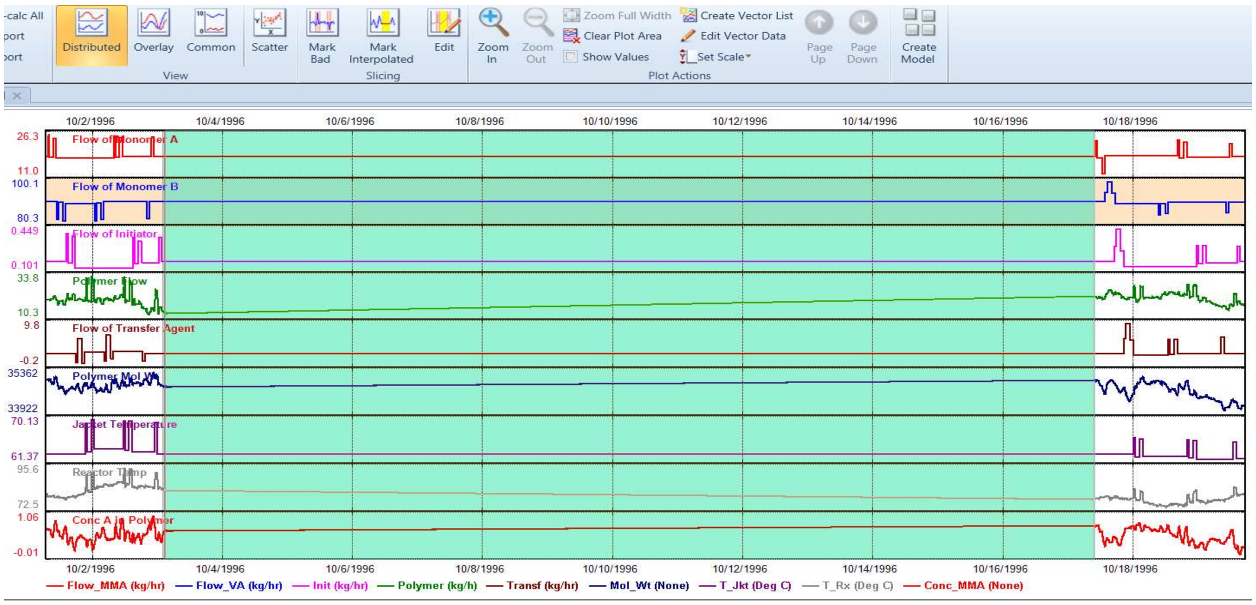
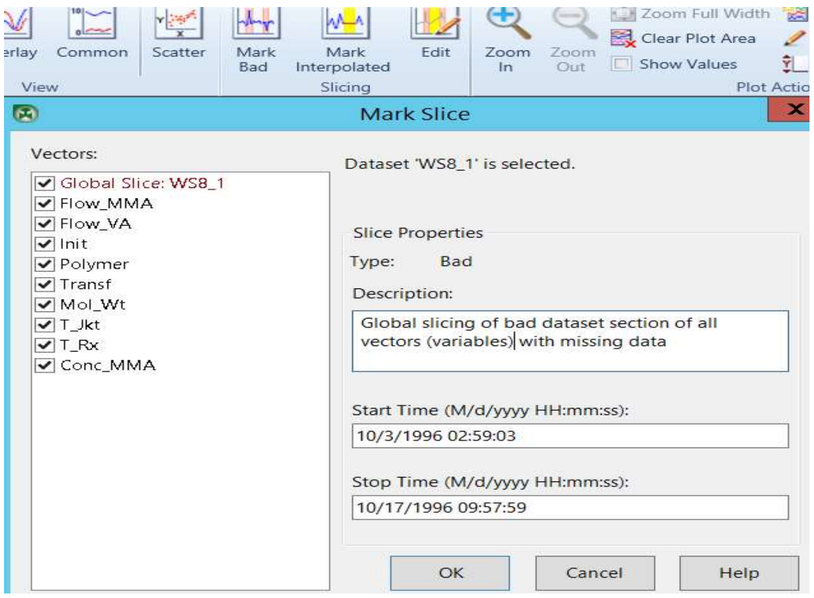


Figure 8.44 Using a mouse to highlight the bad/missing data section to activate the data slicing tools in the ribbon buttons

We then click on the “Mark Bad” ribbon button, and see the input window of Figure 8.45, in which we apply global slicing tool to remove bad dataset section of all vectors (variables) with missing data and click the OK button.



Mark Slice

Vectors:

- ☑ Global Slice: WS8_1
- ☑ Flow_MMA
- ☑ Flow_VA
- ☑ Init
- ☑ Polymer
- ☑ Transf
- ☑ Mol_Wt
- ☑ T_Jkt
- ☑ T_Rx
- ☑ Conc_MMA

Dataset 'WS8_1' is selected.

Slice Properties

Type: Bad

Description:

Global slicing of bad dataset section of all vectors (variables) with missing data

Start Time (M/d/yyyy HH:mm:ss):

10/3/1996 02:59:03

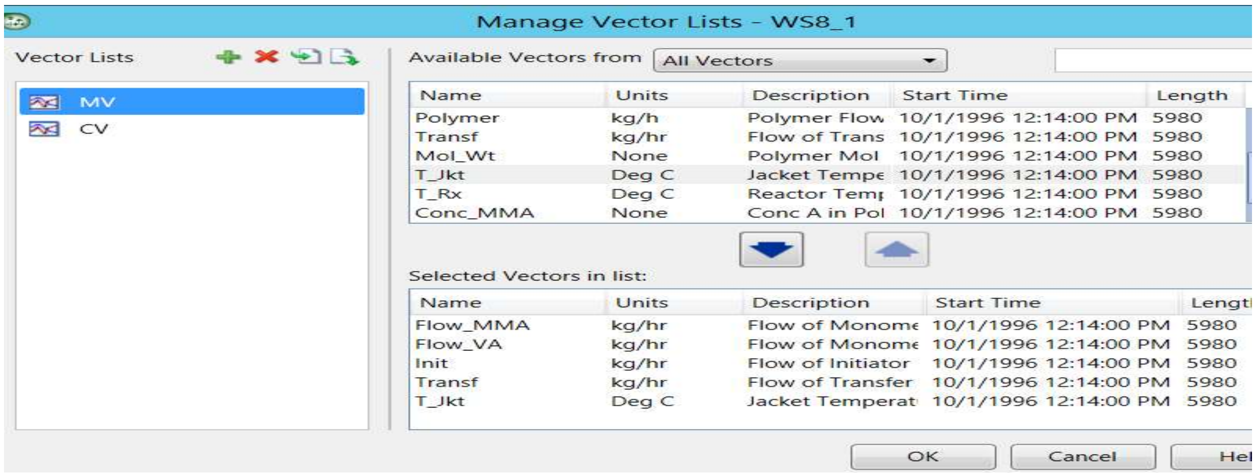
Stop Time (M/d/yyyy HH:mm:ss):

10/17/1996 09:57:59

Figure 8.45 Global slicing of bad dataset section of all vectors (variables) with missing data

8.2.5 Create Manipulated Variable (MV) and Control Variable (CV) Lists

On the top tool ribbon, we choose Manage Lists to build: (1) MV (manipulated variable) list – Flow_MMA, Flow_VA, Init, Transf, and T_Jkt; and (2) CV (control variable) list - Polymer, Mol_Wt, T_Rx and Conc_MMA. Use the Add (+) and Delete (-) buttons to create a new list or delete an existing list, respectively. After creating a list, choose the desired variable (vector) from the right-top list and use the arrow key to move it to the list on the right-button section. See Figures 8.46 and 8.47 for the MV and CV lists.



Manage Vector Lists - WS8_1

Vector Lists

- MV
- CV

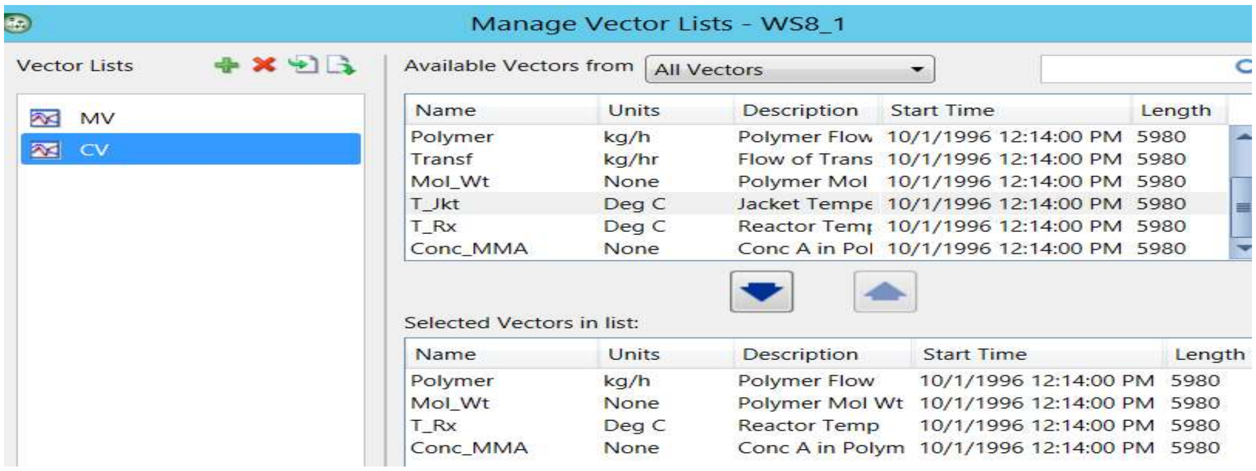
Available Vectors from: All Vectors

Name	Units	Description	Start Time	Length
Polymer	kg/h	Polymer Flow	10/1/1996 12:14:00 PM	5980
Transf	kg/hr	Flow of Trans	10/1/1996 12:14:00 PM	5980
Mol_Wt	None	Polymer Mol	10/1/1996 12:14:00 PM	5980
T_Jkt	Deg C	Jacket Tempe	10/1/1996 12:14:00 PM	5980
T_Rx	Deg C	Reactor Tem	10/1/1996 12:14:00 PM	5980
Conc_MMA	None	Conc A in Pol	10/1/1996 12:14:00 PM	5980

Selected Vectors in list:

Name	Units	Description	Start Time	Lengt
Flow_MMA	kg/hr	Flow of Monome	10/1/1996 12:14:00 PM	5980
Flow_VA	kg/hr	Flow of Monome	10/1/1996 12:14:00 PM	5980
Init	kg/hr	Flow of Initiator	10/1/1996 12:14:00 PM	5980
Transf	kg/hr	Flow of Transfer	10/1/1996 12:14:00 PM	5980
T_Jkt	Deg C	Jacket Temperat	10/1/1996 12:14:00 PM	5980

Figure 8.46 Manipulated variable list, MV


Manage Vector Lists - WS8_1

Vector Lists: MV, CV

Available Vectors from: All Vectors

Name	Units	Description	Start Time	Length
Polymer	kg/h	Polymer Flow	10/1/1996 12:14:00 PM	5980
Transf	kg/hr	Flow of Trans	10/1/1996 12:14:00 PM	5980
Mol_Wt	None	Polymer Mol	10/1/1996 12:14:00 PM	5980
T_Jkt	Deg C	Jacket Tempe	10/1/1996 12:14:00 PM	5980
T_Rx	Deg C	Reactor Temp	10/1/1996 12:14:00 PM	5980
Conc_MMA	None	Conc A in Pol	10/1/1996 12:14:00 PM	5980

Selected Vectors in list:

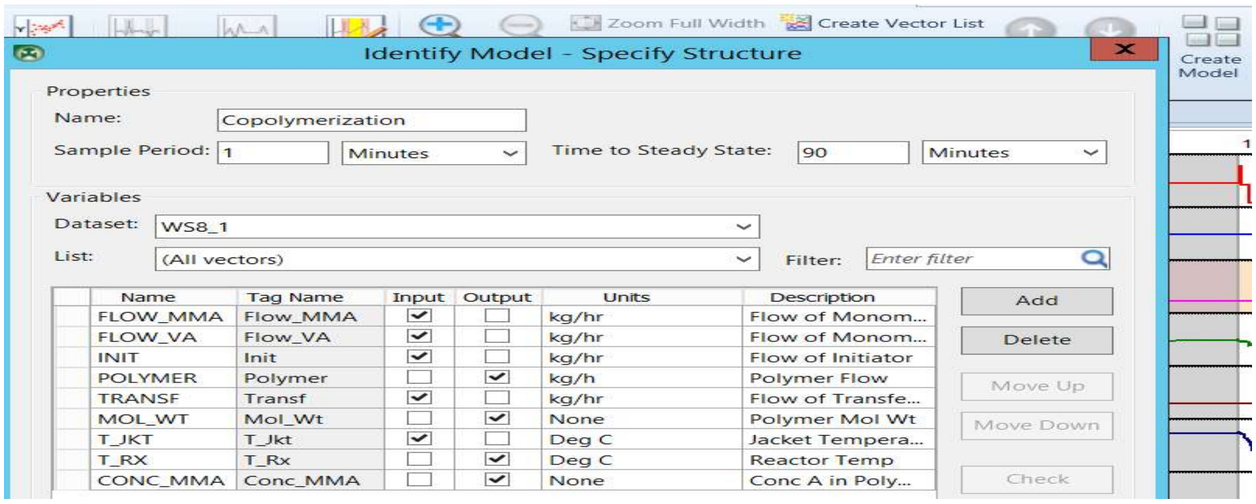
Name	Units	Description	Start Time	Length
Polymer	kg/h	Polymer Flow	10/1/1996 12:14:00 PM	5980
Mol_Wt	None	Polymer Mol Wt	10/1/1996 12:14:00 PM	5980
T_Rx	Deg C	Reactor Temp	10/1/1996 12:14:00 PM	5980
Conc_MMA	None	Conc A in Polym	10/1/1996 12:14:00 PM	5980

Figure 8.47 Control variable list, CV.

We pause to present an important note. Our manipulated variables are independent variables that the operator can change. A control problem may include additional independent variables, called *feedforward variables (FF),* that impact the process, but the operator cannot change them directly. If our dataset from plant step tests includes the time-dependent change profile of feedforward vectors (variables), we should include those FF vectors *to the end of the independent variable list*. For our current problem, we would include any FF variables to the end of the MV list in Figure 8.46 and place those FF variables after the reactor cooling jacket temperature variable, T_Jkt.

8.2.6 DMC3 Builder Task One: Model Identification (ID) for Developing Master Model - Setting up the Model ID

We click on Create Model button located at the far right of the tool ribbons to start building the dynamic controller model using the dataset, ***WS8.1_1.*** In the Identify Model-Specify Structure input form, we enter model name, Copolymerization, specify a Time to Steady State of 90 minutes, and choose the five MVs as input variables, and the four CVs as output variables. See Figure 8.48. Click OK to see the Case Editor Screen of Figure 8.49.



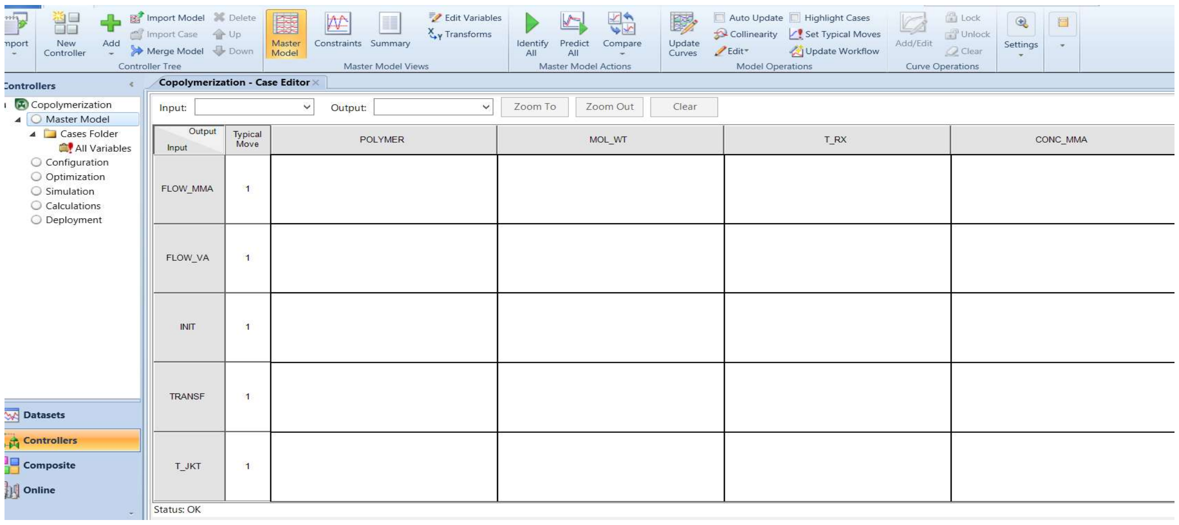
Identify Model - Specify Structure

Properties
Name: Copolymerization
Sample Period: 1 Minutes
Time to Steady State: 90 Minutes

Variables
Dataset: WS8_1
List: (All vectors)
Filter: Enter filter

Name	Tag Name	Input	Output	Units	Description
FLOW_MMA	Flow_MMA	✓		kg/hr	Flow of Monom...
FLOW_VA	Flow_VA	✓		kg/hr	Flow of Monom...
INIT	Init	✓		kg/hr	Flow of Initiator
POLYMER	Polymer		✓	kg/h	Polymer Flow
TRANSF	Transf	✓		kg/hr	Flow of Transfe...
MOL_WT	Mol_Wt		✓	None	Polymer Mol Wt
T_JKT	T_Jkt	✓		Deg C	Jacket Tempera...
T_RX	T_Rx		✓	Deg C	Reactor Temp
CONC_MMA	Conc_MMA		✓	None	Conc A in Poly...

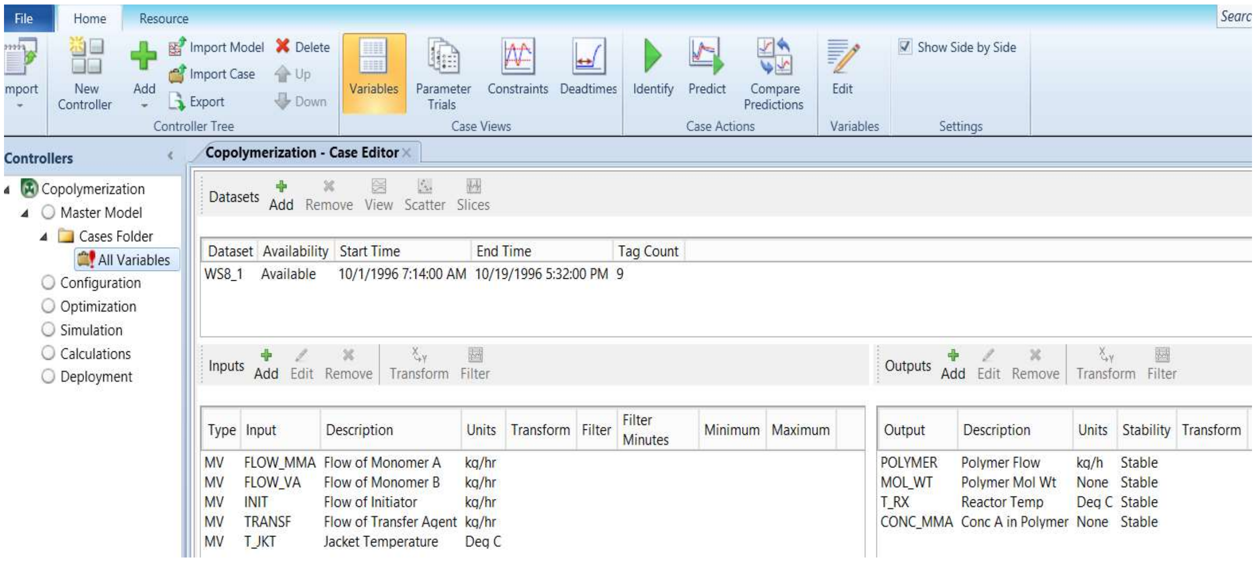
Figure 8.48 Controller model specification



Output / Input	Typical Move	POLYMER	MOL_WT	T_RX	CONC_MMA
FLOW_MMA	1				
FLOW_VA	1				
INIT	1				
TRANSF	1				
T_JKT	1				

Figure 8.49 Case editor screen.

On the left column of the navigation tree for Copolymerization controller, we click on All Variables within Master Model to see a listing of dataset and input and output variables, as illustrated in Figure 8.50.



Dataset	Availability	Start Time	End Time	Tag Count
WS8_1	Available	10/1/1996 7:14:00 AM	10/19/1996 5:32:00 PM	9

Type	Input	Description	Units	Transform	Filter	Filter Minutes	Minimum	Maximum
MV	FLOW_MMA	Flow of Monomer A	kg/hr					
MV	FLOW_VA	Flow of Monomer B	kg/hr					
MV	INIT	Flow of Initiator	kg/hr					
MV	TRANSF	Flow of Transfer Agent	kg/hr					
MV	T_JKT	Jacket Temperature	Deg C					

Output	Description	Units	Stability	Transform
POLYMER	Polymer Flow	kg/h	Stable	
MOL_WT	Polymer Mol Wt	None	Stable	
T_RX	Reactor Temp	Deg C	Stable	
CONC_MMA	Conc A in Polymer	None	Stable	

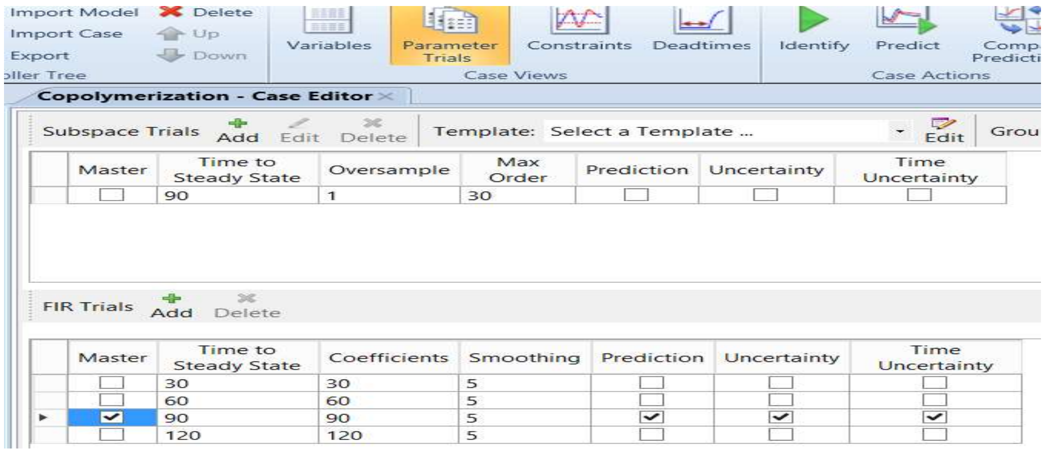
Figure 8.50 Dataset and input and output variables

Next, we click on Parameter Trials of the Case Views on the tool ribbons (see Figure 8.50) to start specifying the trial cases, focusing on FIR trials (simulation runs) with the parameters listed in Table 8.3.

Table 8.3 Parameters for FIR trial cases, WS8.1

Time to Steady State, min	Number of Coefficients	Smoothing Factor
30	30	5
60	60	5
90	90	5
120	120	5

When we click on Parameter Trials, the software automatically creates the cases with "***Time to Steady State (TTSS)***" equal to 30, 60 and 90 min. *Make sure to check the boxes for Master, Prediction, Uncertainty and Time Uncertainty for the 90-min case.* We also need to click on the "+" button next to FIR Trials to add the new case with a TTSS of 120 min. See Figure 8.51.



Master	Time to Steady State	Oversample	Max Order	Prediction	Uncertainty	Time Uncertainty
☐	90	1	30	☐	☐	☐

Master	Time to Steady State	Coefficients	Smoothing	Prediction	Uncertainty	Time Uncertainty
☐	30	30	5	☐	☐	☐
☐	60	60	5	☐	☐	☐
✓	90	90	5	✓	✓	✓
☐	120	120	5	☐	☐	☐

Figure 8.51 List of trial cases

As our sampling period (data collection interval) is 1 min (Figure 8.48), the software assumes a controller execution interval of 1 min and gives ***the Number of Model Coefficients*** equal to the TTSS according to Eq. (8.48):

$$\begin{bmatrix} Controller \\ Execution \\ Interval, min \end{bmatrix} = \frac{Time\ to\ Steady\ State, min}{Number\ of\ Moel\ Coefficients} \times \begin{bmatrix} Sampling \\ Period, min \end{bmatrix} \qquad (8.48)$$

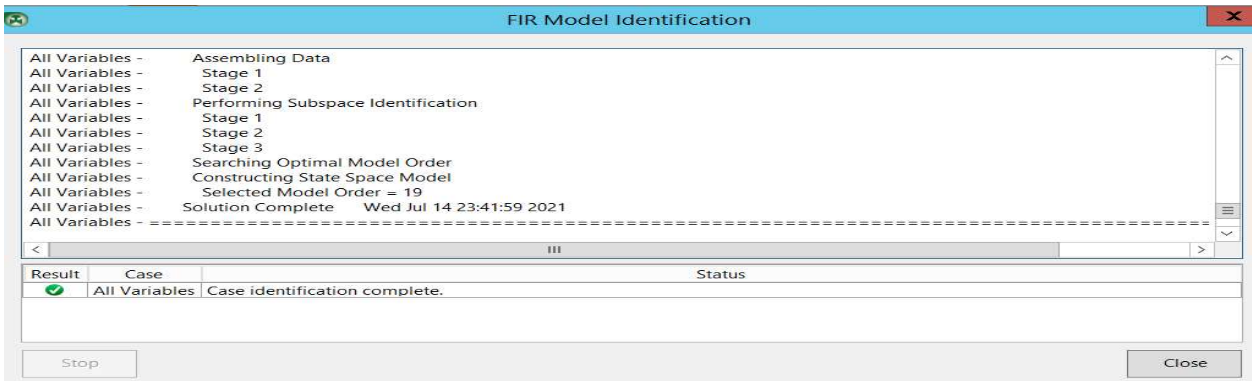
Model coefficients are required to model faster responses. For example, if the controller execution interval is 0.5 min, with the TTSS (= 90 min) and sampling period (= 1 min) remain the same, Eq. (8.48) means the number of model coefficients is 180. The number of model coefficients also determines *the number of future control moves* being calculated by DMC3 Builder. See Table 8.4. We note that the larger the number of model coefficients, the smaller the controller execution interval and the larger the number of control moves calculated. Understanding this relationship is key to applying control to respond to fast-changing independent disturbances in the system.

Table 8.4 Relationship between the number of model coefficients and the number of control moves calculated

Number of model coefficients	30	45	60	75	90	105	120
Number of control moves calculated	8	9	10	11	12	13	14

The third parameter, ***Smoothing Factor***, is used to smooth the data and apply the penalty for change between successive FIR model coefficients. The default value of 5 is acceptable in most cases.

We now move to Case Actions within the tool buttons and click on the Identify button to run the model identification (ID). Figure 8.52 shows the window display of the progress of the FIR model identification. We click "Close" when seeing the message of "Solution Complete". Figure 8.53 shows the identified smoothed and unsmoothed model curves for each TTSS



FIR Model Identification

```
All Variables -      Assembling Data
All Variables -        Stage 1
All Variables -        Stage 2
All Variables -      Performing Subspace Identification
All Variables -        Stage 1
All Variables -        Stage 2
All Variables -        Stage 3
All Variables -      Searching Optimal Model Order
All Variables -      Constructing State Space Model
All Variables -        Selected Model Order = 19
All Variables -    Solution Complete      Wed Jul 14 23:41:59 2021
All Variables - ==========================================================================
```

Result	Case	Status
✓	All Variables	Case identification complete.

Figure 8.52 Window display of the progress of the FIR model identification

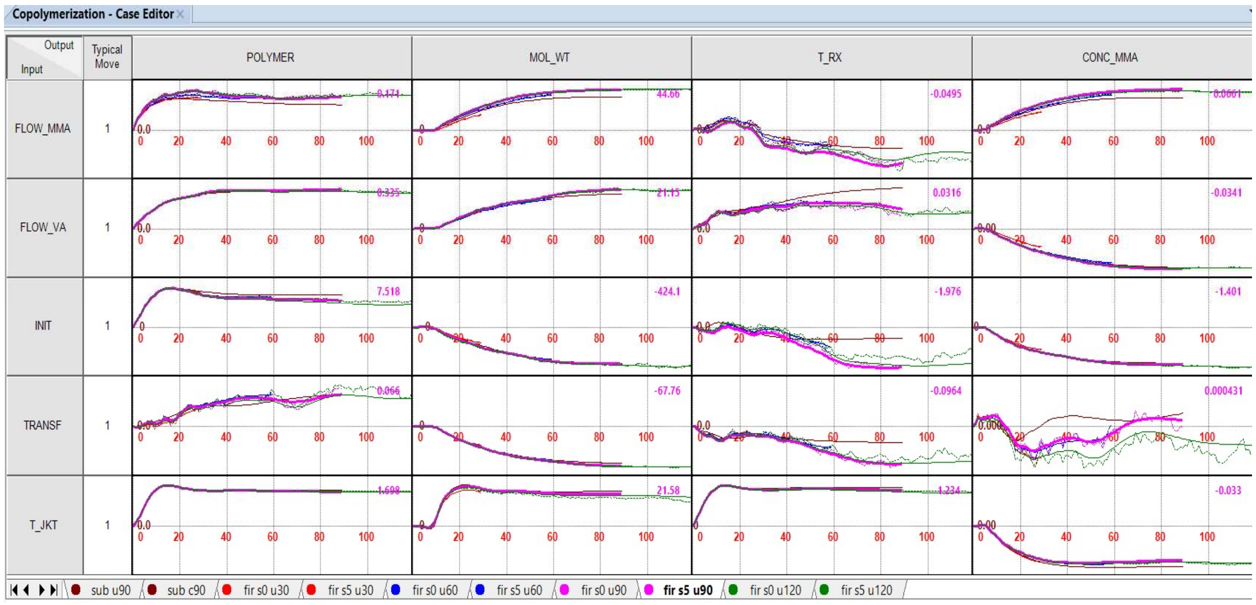


Figure 8.53 Identified smoothed and unsmoothed model response curves:
"fir S5, U90" means finite-impulse response; smoothing factor = 5; TTSS = 90 min.

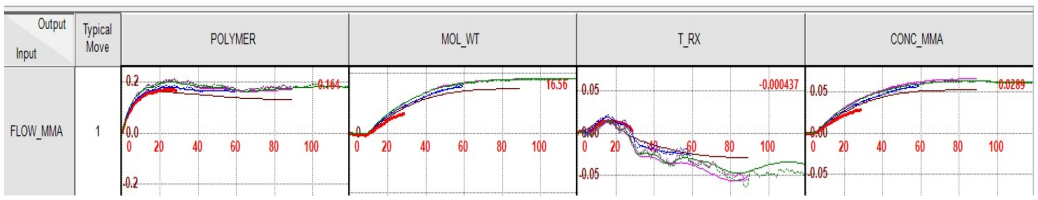
8.2.7 Guidelines for Selecting Model Parameters

In Section 8.2.6, we have previously discussed *the number of model coefficients* in relation to Eq. (8.48), the sampling period and the controller execution interval. We indicate that the use of *a smoothing factor* of 5 is always a good practice. How do we then choose the third parameter, the time to steady state (TTSS)?

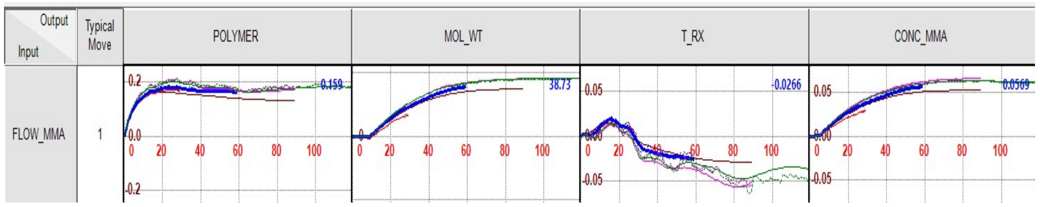
We choose the TTSS based on the slowest responses in the model, and all model responses should reach steady state at the selected TTSS. We extend faster response curves to match the selected TTSS.

Figures 8.54a to 8.54d compare the finite-impulse response curves at TTSS = 30, 60, 90 and 120 min and with a smoothing factor of 5. *The comparison confirms our selection of master model with a TTSS of 90 min for all control variables (Polymer, Mol_Wt, T_Rx and Conc_MA) to reach their steady-state values.* In general, a control variable will continue to change past a chosen TTSS value that is too short; and a large

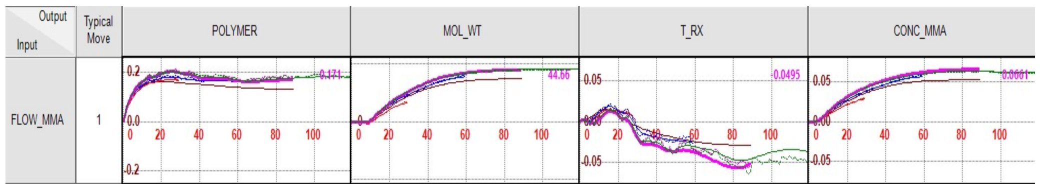
TTSS will cause the smoothed and unsmoothed response curves of a control variable to drift apart at the end.



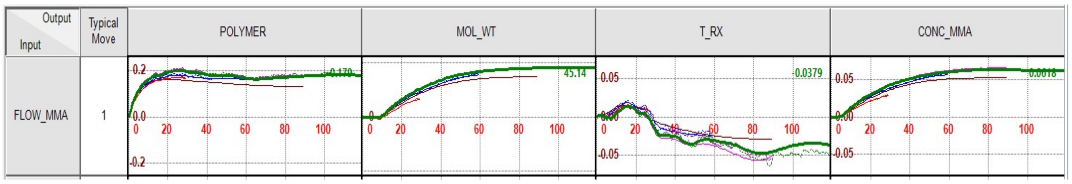
(a) fir S5 U30: finite-impulse response -> S5 =smoothing factor of 5, U30 = TTSS of 30 min. MOL_WT and CONC_MMA (y-axis) continue to increase past TTSS of 30 min (x-axis). T_RX continues to drop past TTSS of 30 min. Pay attention to the red curve that ends at 30 min.



(b) fir S5 U60: TTSS of 60 min. MOL_WT and CONC_MMA continue to increase past TTSS of 60 min. T_RX continues to drop past TTSS of 60 min.



(c) fir S5 U90: TTSS of 90 min. POLYMER, MOL_WT, T_RX and CONC_MMA appear to reach their steady-state values and do not change much past TTSS of 90 min.



(d) fir S5 U120: TTSS of 120 min appears to be too long as all dependent variables have already reached their steady-state values around TTSS of 90 min.

Figure 8.54(a)-(d) Evaluation of the selection of time to steady state (TTSS).

8.2.8 Uncertainty and Correlation Plots of the Master Model

We display the master model with a TTSS of 90 min by following the path on the navigation tree: Copolymerization Model ->Master Model -> Cases Folder ->All Variables and click on the "Frequency Uncertainty" button on the tool ribbons. Figure 8.55 shows the resulting *frequency-domain uncertainty plot*. In each response plot, the shaded area above and below the dark average response curve indicates a two-sigma confidence region that includes 95.4% of all data points. The narrower the shaded area, the more accurate the average response curve is. For example, the model response plots of Mol_Wt and

Conc_MMA to changes in Flow_MMA contain very narrow two-sigma confidence region and the quality of these model response plots is graded “excellent” or “A”. By contrast, the wide shaded two-sigma region for the model response plot of T_Rx to changes in Flow_MMA and in Transf indicates a plot of poor quality with a grade of “C”. We see plots of very poor quality with large shaded two-sigma region for the model response plot of Polymer to changes in Transf with a grade of D.

Likewise, Figure 8.56 shows *the time-domain uncertainty plot.* For each input/output variable pair, we see the shaded two-sigma confidence region, and the corresponding model grade from A to D. Both frequency-domain and time-domain uncertainty plots result in essentially identical model grades.

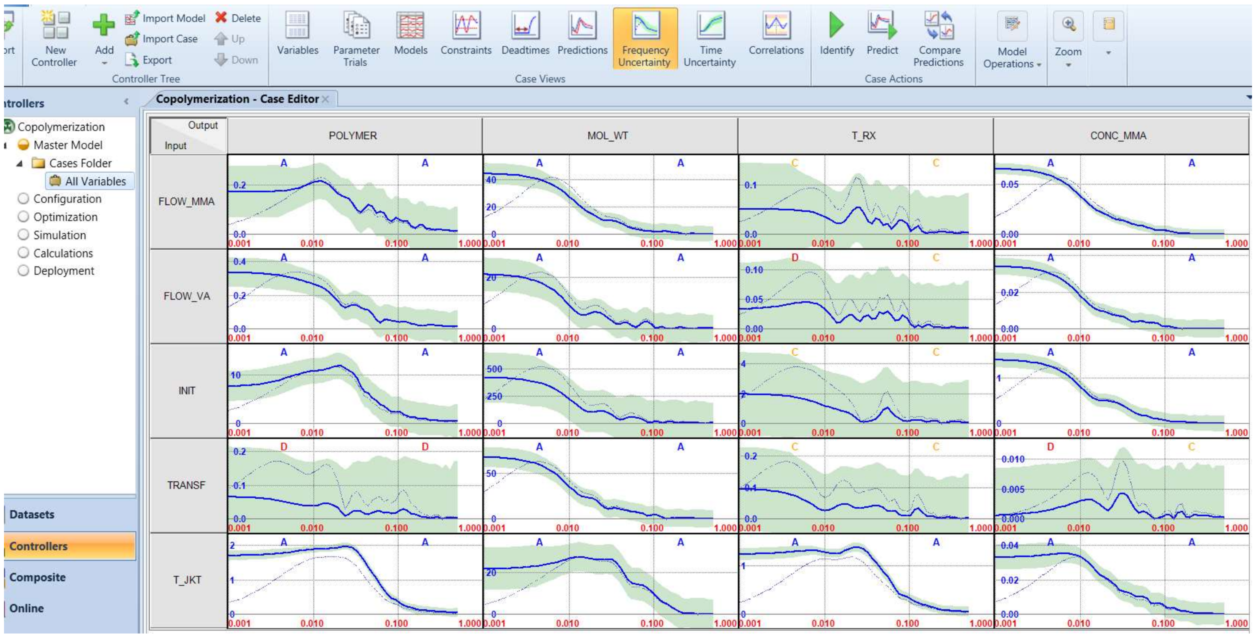


Figure 8.55 Frequency domain uncertainty plot for the master model at a TTSS of 90 min

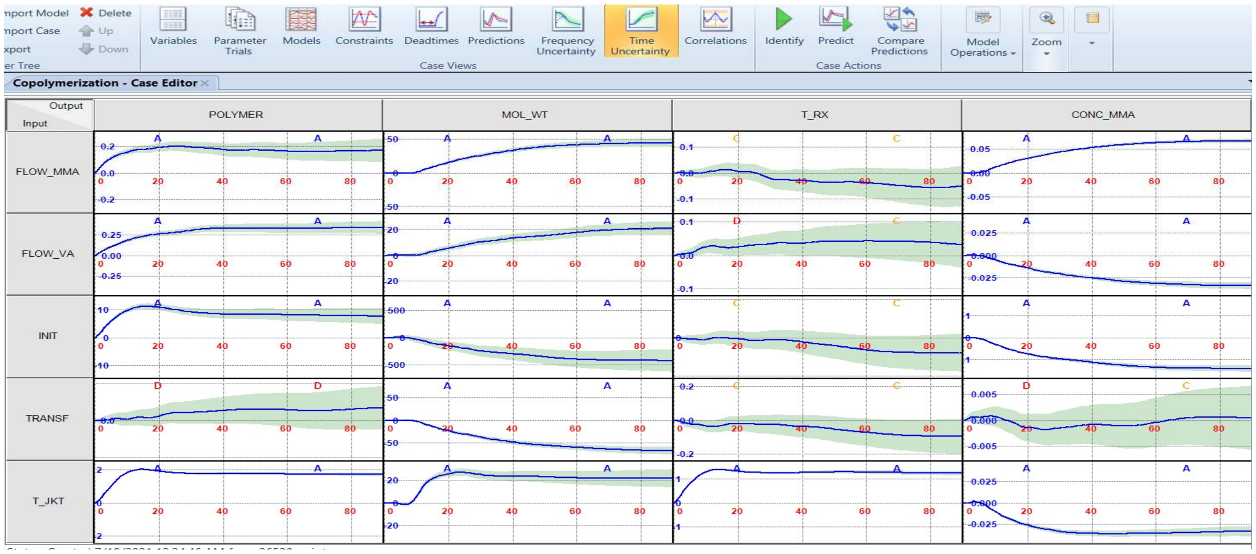
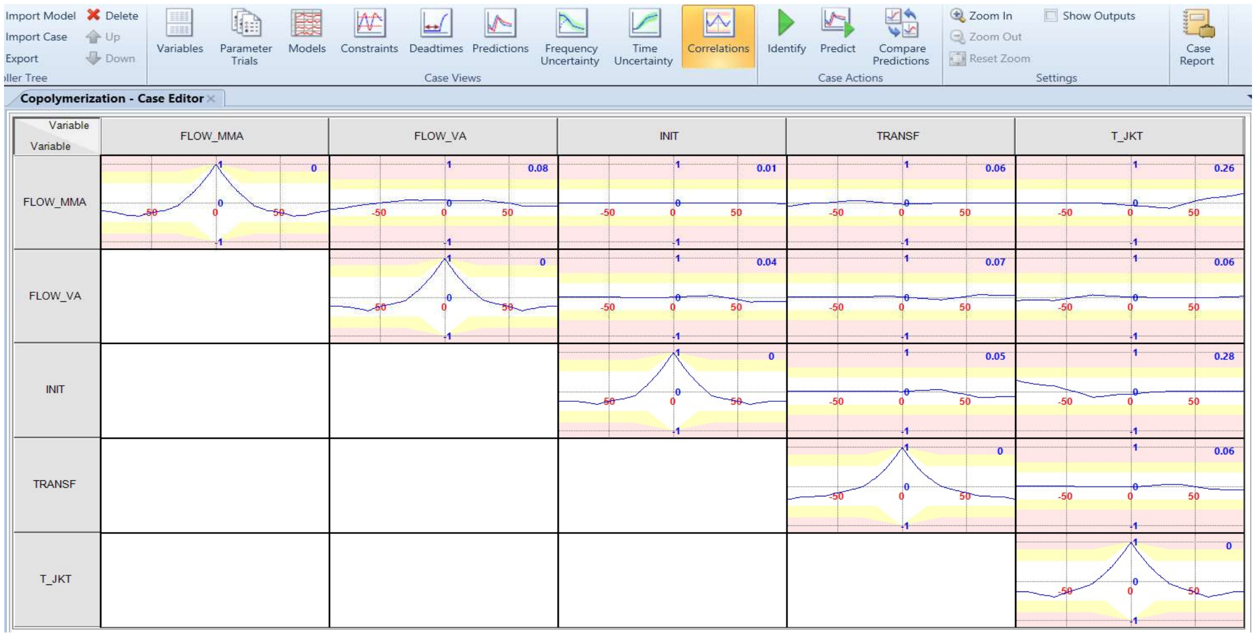


Figure 8.56 Time domain uncertainty plot for the master model with a TTSS of 90 min

The correlation plots show how much an input variable or MV correlates with another input variable. The correlation coefficient is a statistical measure of the strength of the relationship between the relative movements of two variables. The values range between -1.0 and 1.0. ... A correlation of -1.0 shows a perfect negative correlation, while a correlation of 1.0 shows a perfect positive correlation. In Figure 8.57, both x-axis and y-axis represent input variables or MVs. We see the value of the correlation coefficient between two input variables on the upper right corner of each plot, with values between 0 and 0.28, indicating a relatively minor positive correlation.

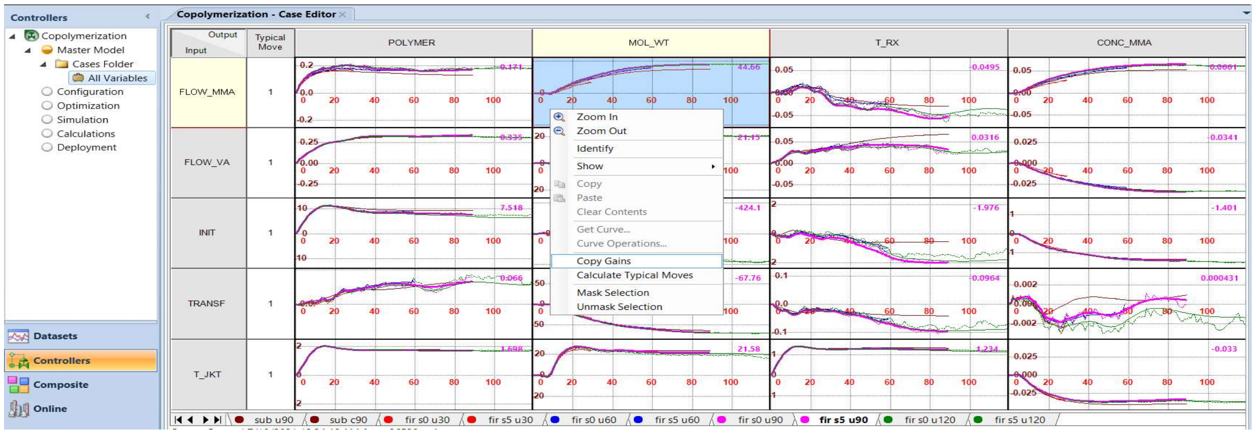


Variable	FLOW_MMA	FLOW_VA	INIT	TRANSF	T_JKT
FLOW_MMA	0	0.08	0.01	0.06	0.26
FLOW_VA		0	0.04	0.07	0.06
INIT			0	0.05	0.28
TRANSF				0	0.06
T_JKT					0

Figure 8.57 Correlation plot for the master model at a TTSS of 90 min

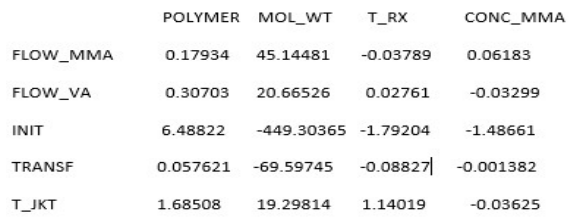
8.2.9 DMC3 Builder Task One: Building the Controller Model for Developing the Master Model

Before creating the final controller model, we need to check the steady-state gain of each input-output pair and investigate which changes in an input variable have a notable impact on a specific output variable. Specifically, we copy the steady-state gain values of all input-output pairs by following the path: Controllers ->Copolymerization -> Master Model ->Case Folder -> All Variables -> Right-click within the workspace of model response curves ->Copy gains. See Figures 8.58 and 8.59.



Input \ Output	Typical Move	POLYMER	MOL_WT	T_RX	CONC_MMA
FLOW_MMA	1	0.171	44.66	0.0495	0.0601
FLOW_VA	1	0.335	21.15	0.0316	-0.0341
INIT	1	7.518	-424.1	-1.976	-1.401
TRANSF	1	0.066	-67.76	0.0964	0.000431
T_JKT	1	1.698	21.58	1.234	-0.033

Figure 8.58 Copying steady-state gains of model response curves.



	POLYMER	MOL_WT	T_RX	CONC_MMA
FLOW_MMA	0.17934	45.14481	-0.03789	0.06183
FLOW_VA	0.30703	20.66526	0.02761	-0.03299
INIT	6.48822	-449.30365	-1.79204	-1.48661
TRANSF	0.057621	-69.59745	-0.08827	-0.001382
T_JKT	1.68508	19.29814	1.14019	-0.03625

Figure 8.59 Steady-state gains of model response curves with a TTSS of 90 min

Figure 8.60 gives the explicit ratio elements of the steady-state gain matrix of Figure 8.23.

$$\begin{array}{cccc}
\frac{\Delta(\text{Polymer})}{\Delta(\text{Flow\_MMA})} & \frac{\Delta(\text{Mol\_Wt})}{\Delta(\text{Flow\_MMA})} & \frac{\Delta(\text{T\_Rx})}{\Delta(\text{Flow\_MMA})} & \frac{\Delta\ (\text{Conc\_MMA})}{\Delta(\text{Flow\_MMA})} \\
\frac{\Delta(\text{Polymer})}{\Delta(\text{Flow\_VA})} & \frac{\Delta(\text{Mol\_Wt})}{\Delta(\text{Flow\_VA})} & \frac{\Delta(\text{T\_Rx})}{\Delta(\text{Flow\_VA})} & \frac{\Delta(\text{Conc\_MMA})}{\Delta(\text{Flow\_VA})} \\
\frac{\Delta(\text{Polymer})}{\Delta(\text{Init})} & \frac{\Delta(\text{Mol\_Wt})}{\Delta(\text{Init})} & \frac{\Delta(\text{T\_Rx})}{\Delta(\text{Init})} & \frac{\Delta(\text{Conc\_MMA})}{\Delta(\text{Init})} \\
\frac{\Delta(\text{Polymer})}{\Delta(\text{Transf})} & \frac{\Delta(\text{Mol\_Wt})}{\Delta(\text{Transf})} & \frac{\Delta(\text{T\_Rx})}{\Delta(\text{Transf})} & \frac{\Delta(\text{Conc\_MMA})}{\Delta(\text{Transf})} \\
\frac{\Delta(\text{Polymer})}{\Delta(\text{T\_Jkt})} & \frac{\Delta(\text{Mol\_Wt})}{\Delta(\text{T\_Jkt})} & \frac{\Delta(\text{T\_Rx})}{\Delta(\text{T\_Jkt})} & \frac{\Delta(\text{Conc\_MMA})}{\Delta(\text{T\_kJt})}
\end{array}$$

Figure 8.60 Explicit ratio elements of the steady-state gain matrix

Comparing the magnitudes of the steady-state gain values in each CV column, we conclude that all input variables or MV rows have a notable influence on the “polymer molecular weight” column. Additionally, we note the following:

- The mass flow rate of the chain-transfer agent, Transf, has the least influence on the polymer mass flow rate, Polymer, with a gain of 0.05762.
- Both mass flow rates of MMA and VA, that is, Flow_MMA and Flow_VA, have the smallest influences on the reactor exit temperature, T_Rx with gains of -0.03789 and 0.02761.
- The mass flow rate of the chain-transfer agent, Transf, has the least influence on the concentration or mole fraction of monomer MMA in the polymer product, Conc_MMA with a gain of -0.00138.

We eliminate those MVs that have the least influence on a specific CV with smallest steady-state gain values and click on “Mask Selection” to disable copying those response curves over to the final master model. For example, we mask the mass flow rate of transfer agent, Transf, as a MV for controlling the mole fraction of MMA in the polymer product, Conc_MMA, by highlighting the corresponding response curve, right-click to show the selection menu, and then choose “Mask Selection”. See Figures 8.61 and 8.62.

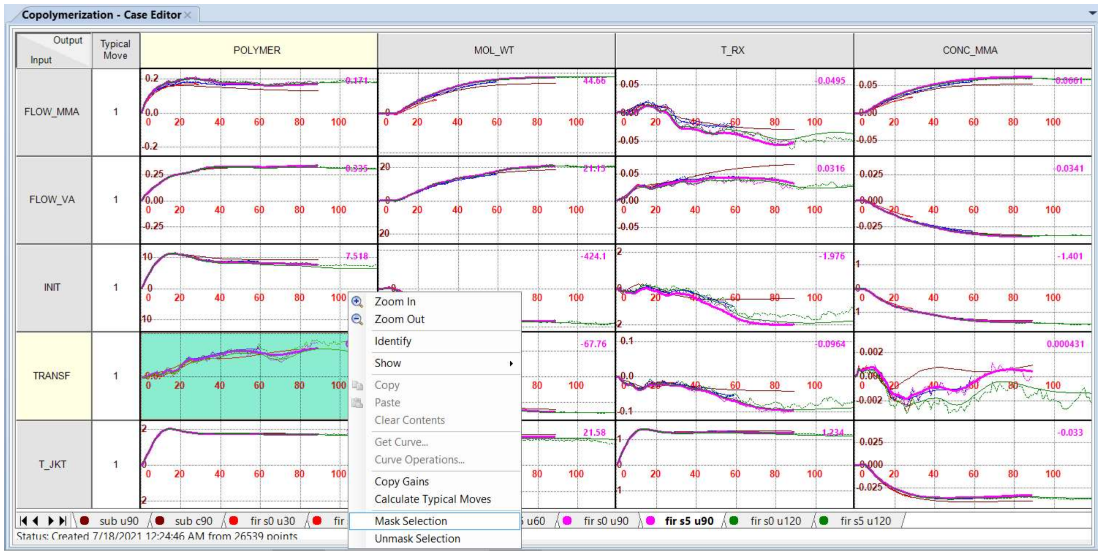


Figure 8.61 Eliminating the model response curve of a selected input-output pair from the final master model by Mark Selection.

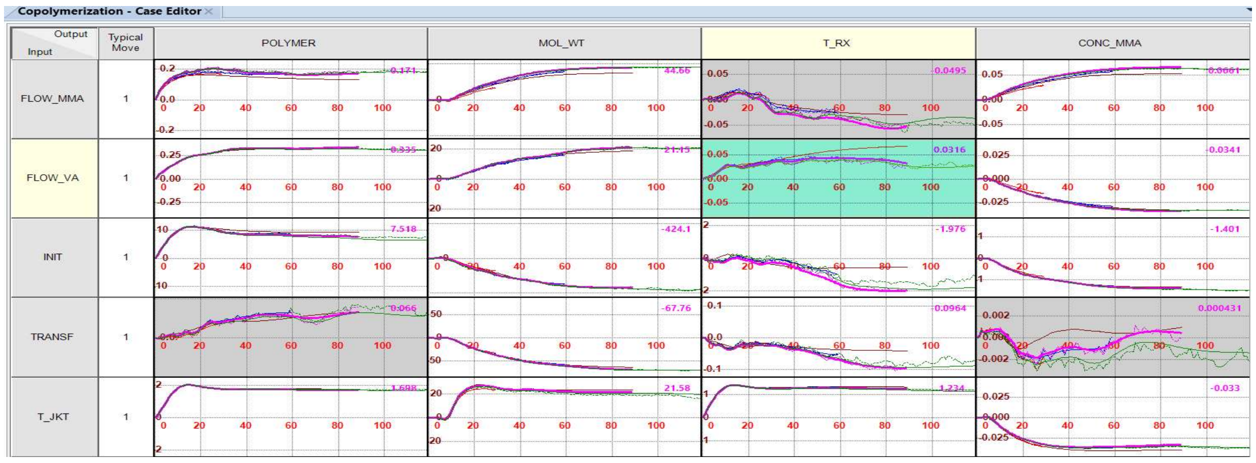
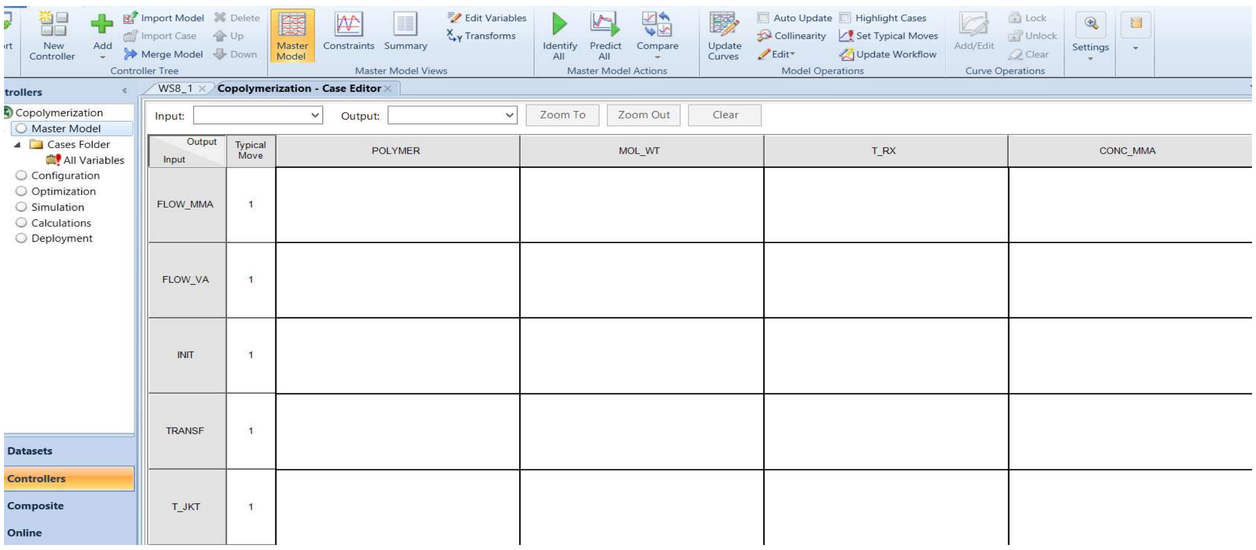


Figure 8.62 A display of four highlighted response curves that are eliminated in the final maser model after Mask Selection

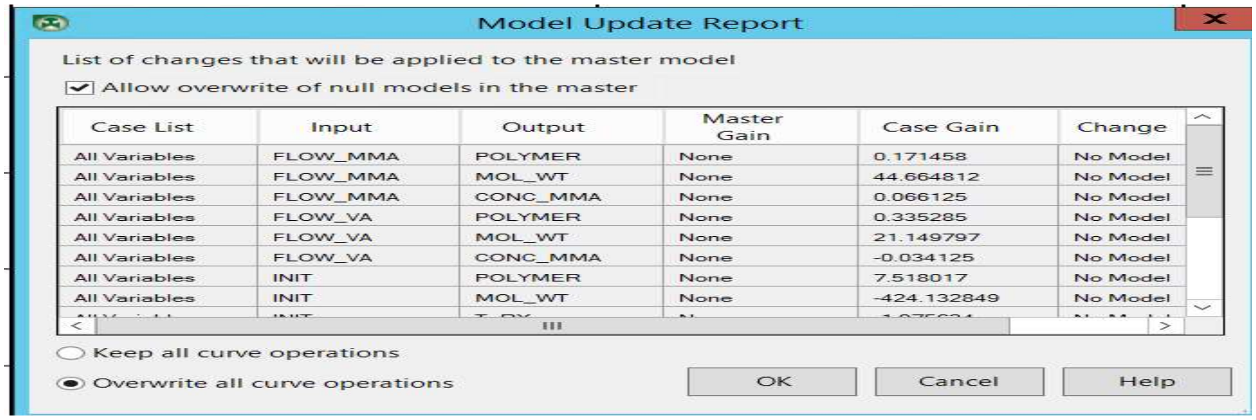
8.2.10 DMC3 Builder Task One: Creating a Controller Model

The first step to create a controller model is to define the various trial runs as the Master trial. We confirm our previous selection illustrated in Figure 8.53. We click on Master Model in the navigation tree and see the Model views in the tool ribbons. Click on Update Curve within Model Operations in the tool ribbons. See Figure 8.63. After clicking on Update Curves, we see a model update report. Be sure to choose “Allow overwrite of all null models in the master”, and “Overwrite all curve operations” (see Figure 8.64). Clicking OK will generate the Master Model response curves of Figure 8.65.



Output / Input	Typical Move	POLYMER	MOL_WT	T_RX	CONC_MMA
FLOW_MMA	1				
FLOW_VA	1				
INIT	1				
TRANSF	1				
T_JKT	1				

Figure 8.63 Master Model view waiting for Update Curves to copy the Master Case response curves with a TTSS of 90 min to the empty model panel


Model Update Report

List of changes that will be applied to the master model

[✓] Allow overwrite of null models in the master

Case List	Input	Output	Master Gain	Case Gain	Change
All Variables	FLOW_MMA	POLYMER	None	0.171458	No Model
All Variables	FLOW_MMA	MOL_WT	None	44.664812	No Model
All Variables	FLOW_MMA	CONC_MMA	None	0.066125	No Model
All Variables	FLOW_VA	POLYMER	None	0.335285	No Model
All Variables	FLOW_VA	MOL_WT	None	21.149797	No Model
All Variables	FLOW_VA	CONC_MMA	None	-0.034125	No Model
All Variables	INIT	POLYMER	None	7.518017	No Model
All Variables	INIT	MOL_WT	None	-424.132849	No Model

Keep all curve operations
[✓ Overwrite all curve operations]

Figure 8.64 Model update report after clicking on Update Curve.

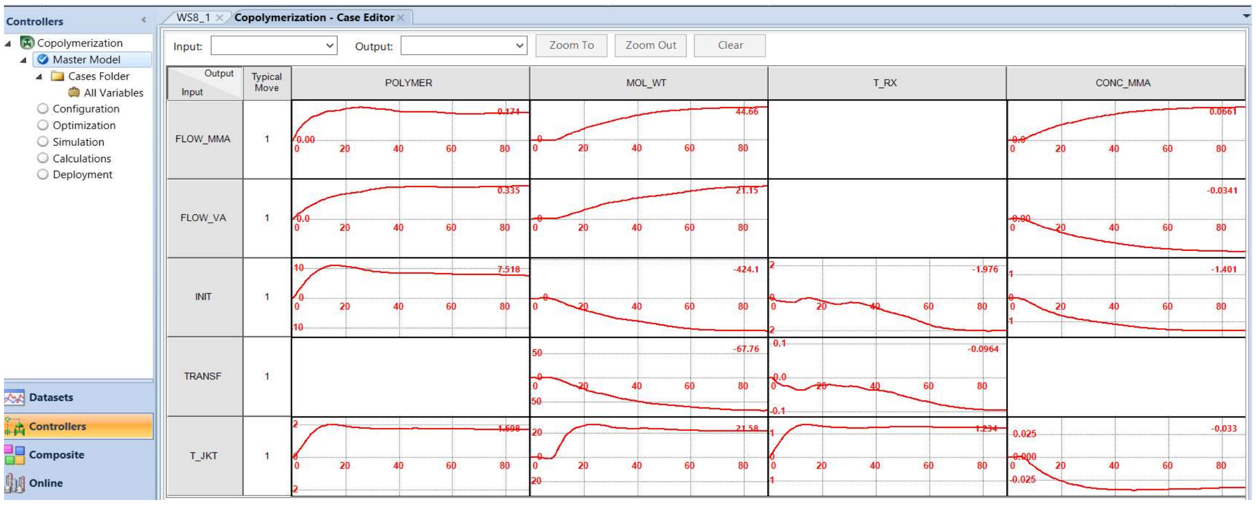


Figure 8.65 Updated final master model with a TTSS of 90 min.

8.2.11 Identification of Dead Time in Model Response Curves

We enlarge the model response curve between Flow_VA and Mol_Wt by double-clicking on the curve. The enlarged curve appears to show a dead time of about 9 min before Mol_Wt begins to change after Flow_VA changes. We identify the dead time by following the path: Highlight the model curve -> Right-click to open the menu for curve operations -> Click on Curve Operations (see Figure 8.66) -> Shift the curve by -9 min and update chart (see Figure 8.67)-> Then shift the curve by +9 min and update chart (see Figure 8.68) -> Update curve.

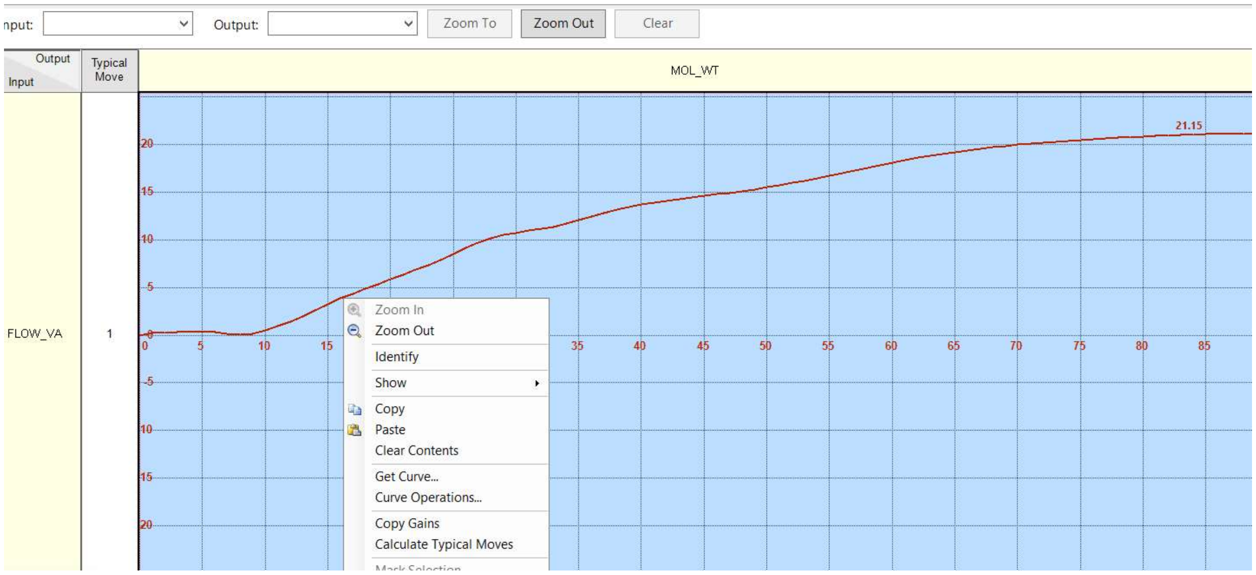
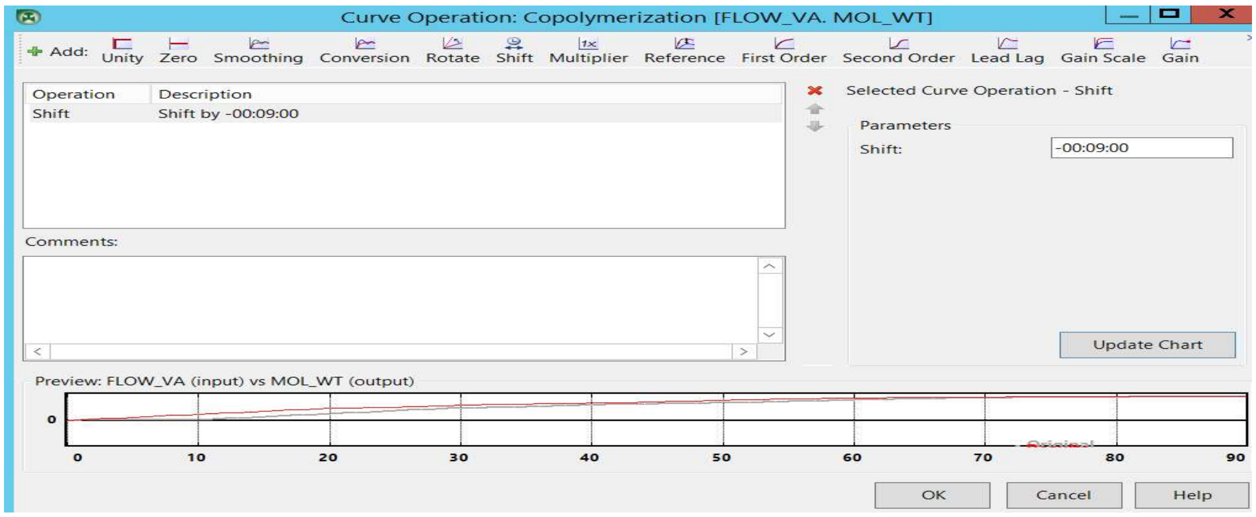


Figure 8.66 Access the menu for curve operations



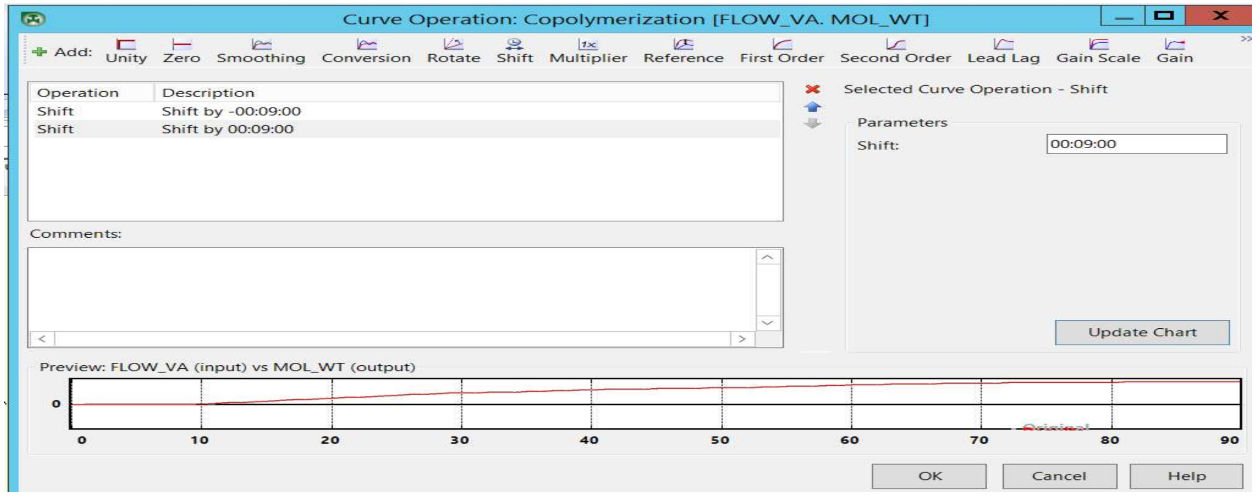
Operation	Description
Shift	Shift by -00:09:00

Selected Curve Operation - Shift

Parameters

Shift: -00:09:00

Figure 8.67 Shifting the model curve by -9 min and updating chart.



Curve Operation: Copolymerization [FLOW_VA. MOL_WT]

Operation	Description
Shift	Shift by -00:09:00
Shift	Shift by 00:09:00

Selected Curve Operation - Shift

Parameters

Shift: 00:09:00

Figure 8.68 Shifting the model curve by 9 min and updating chart.

Figure 8.69 shows the model response curves resulting from this curve operation to identify the dead time between output variable Mol_Wt and input variable Flow_VA. Note the dark blue triangular on the lower right corner of the response curve plot between these variables.

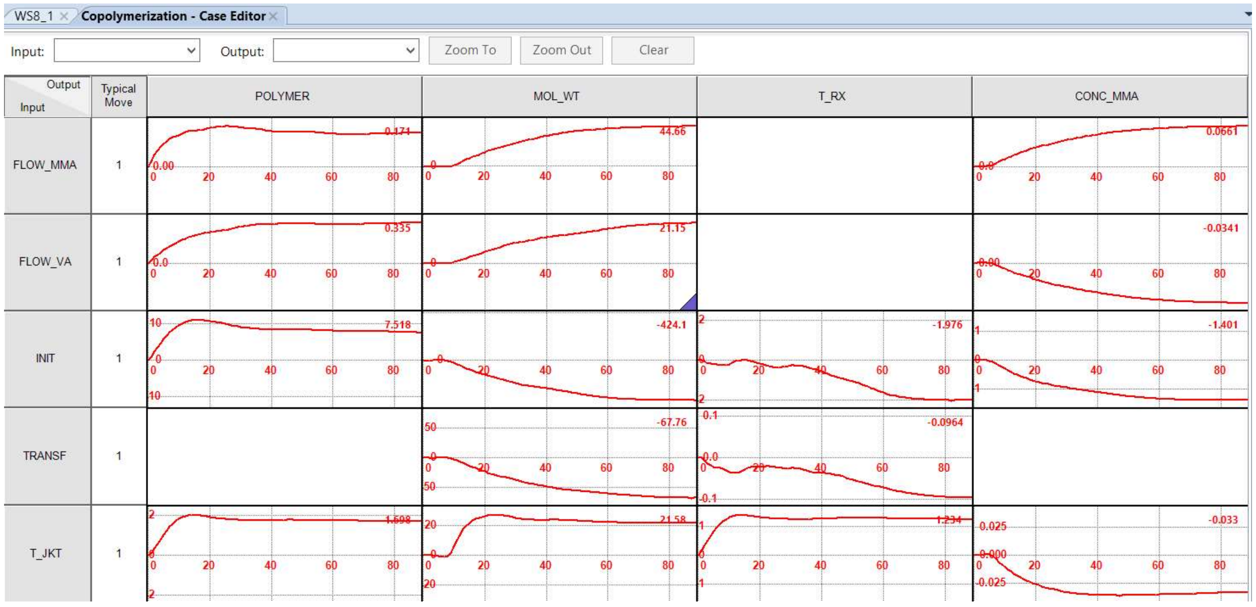


Figure 8.69 Model response plot with a blue triangular on the lower right corner of a plot indicating having completed a curve operation

Following the same procedure, we identify the dead times of other input-output pairs. Figure 8.70 shows the resulting model response curves.

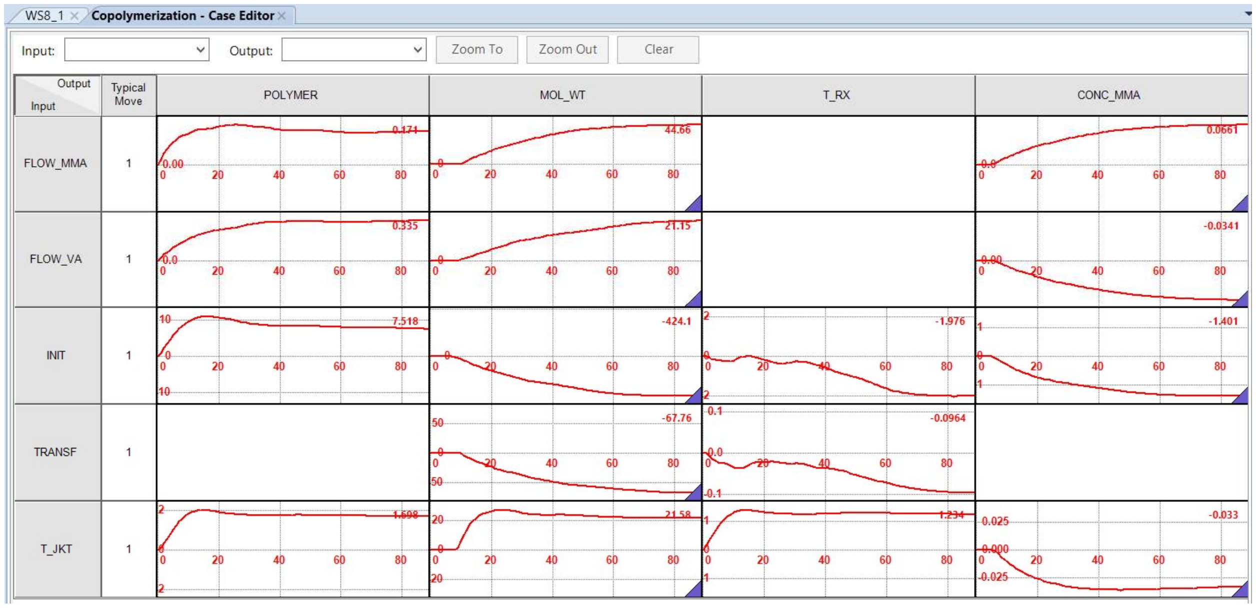


Figure 8.70 Model response curves after curve operations for dead times.

8.2.12 Collinearity Analysis

Always perform collinearity analysis (discussed in Sections 8.1.4a and 8.1.4b) on the master model before its deployment. The collinearity analysis identifies and repairs ill-conditioned model matrices. It can identify sub-models from a model matrix that are nearly collinear or highly nonlinear. A collinearity analysis includes the following four steps:

Step 1. Select variables and specify options – We choose MVs and CVs to be analyzed and specify the gain analysis options, such as RGA (relative gain analysis) threshold, singular value analysis, allowable gain changes, etc.

Step 2. Analyze and determine relationships – We analyze the model and determine which CV-MV pairs have collinearity trouble and specify confidence limits for the gains on individual model response curves.

Step 3. Create groups – We create groups using MV-CV curves that do not have square relationships (2x2, 3x3, etc.).

Step 4. Repair groups and update model – We repair gains for the square and non-square groups by either collinearizing or un-collinearizing the groups.

We explain and demonstrate the details of each step below.

Step 1. Select Variables and Specify Options.

We begin by going to the Controller navigation tree and choose Copolymerization -> Master Model. On the top tool buttons, we click on “Collinearity” within “Model Operations”. See Figure 8.71. We immediately see a dialog: “Do you want to use collinearity repair wizard?” We choose “No” in order to use the collinearity repair dialog. See Figure 8.72.

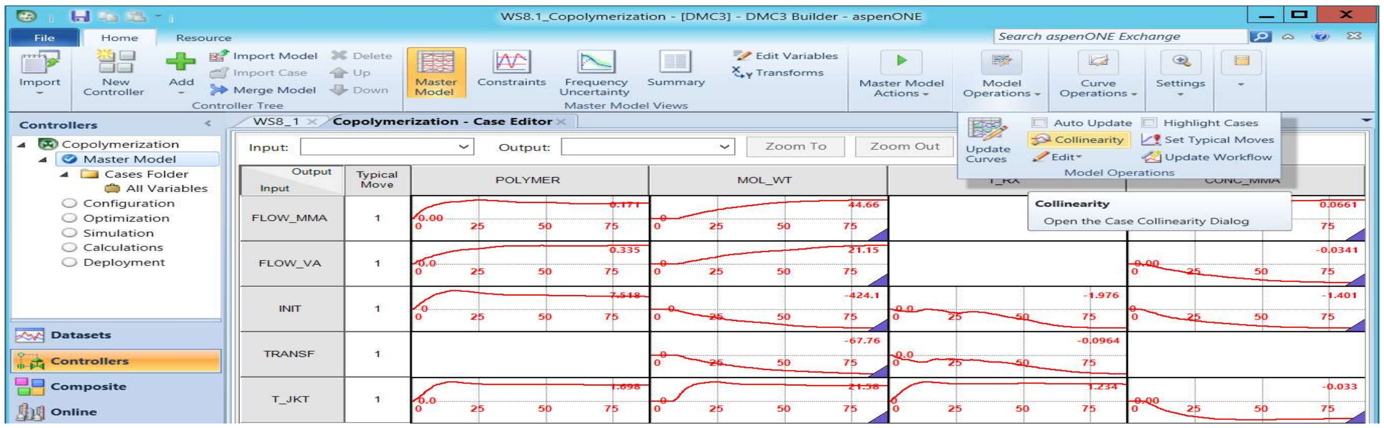
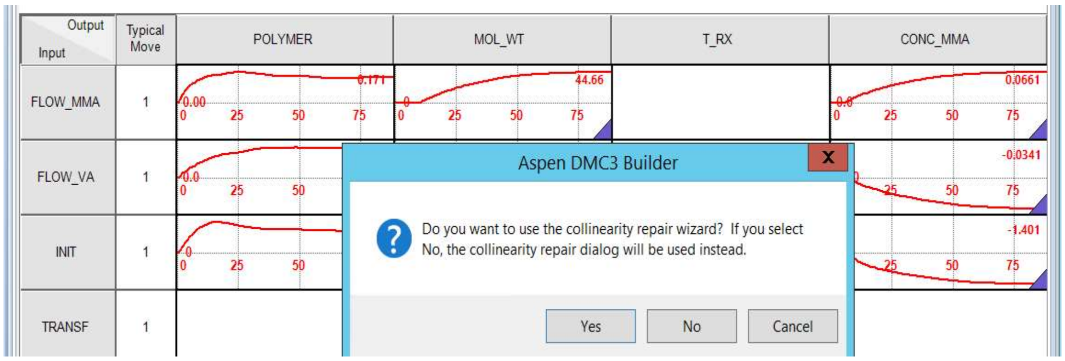


Figure 8.71 Activate the collinearity analysis within model operations.



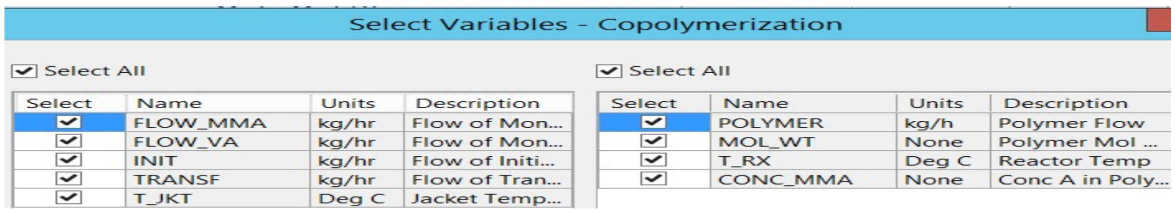
Aspen DMC3 Builder

Do you want to use the collinearity repair wizard? If you select No, the collinearity repair dialog will be used instead.

Yes No Cancel

Figure 8.72 Choose the collinearity repair dialog by clicking on “No”.

We then see “Select Variables –Copolymerization” and select all MVs and CVs for analysis. Click OK to proceed to the Collinearity Analysis window. See Figure 8.73. We note that if our list of MVs includes feedforward variables, we do not choose them in the collinearity analysis.


Select Variables - Copolymerization

✓ Select All

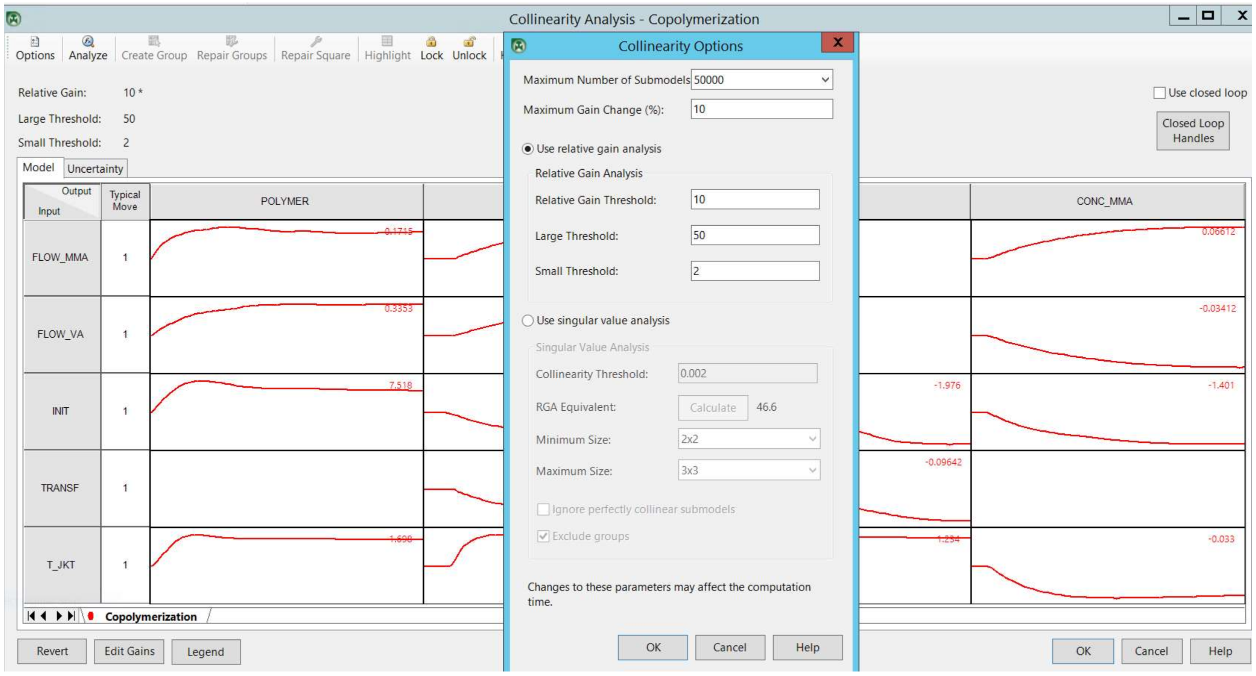
Select	Name	Units	Description
✓	FLOW_MMA	kg/hr	Flow of Mon...
✓	FLOW_VA	kg/hr	Flow of Mon...
✓	INIT	kg/hr	Flow of Initi...
✓	TRANSF	kg/hr	Flow of Tran...
✓	T_JKT	Deg C	Jacket Temp...

✓ Select All

Select	Name	Units	Description
✓	POLYMER	kg/h	Polymer Flow
✓	MOL_WT	None	Polymer Mol ...
✓	T_RX	Deg C	Reactor Temp
✓	CONC_MMA	None	Conc A in Poly...

Figure 8.73 Select variables for collinearity analysis.

Next, we see the “Collinearity Analysis –Copolymerization” window, displaying the top toolbar for collinearity analysis buttons. We click on “Options” and see the window of “Collinearity Options”, as shown in Figure 8.74. We note that the default settings specify the use of RGA with a relative gain threshold of 10, a large threshold of 50 and a small threshold of 1. These default settings will suffice for our model. We click OK on the “Collinearity Options” window to accept these settings, and then see the display of the results of the collinearity analysis of Figure 8.75. In the figure, we see the total number of submatrices as 1. This follows because we have 5 MVs and 4 CVs, and a 5x4 gain matrix; and RGA applies only to a single 4x4 square submatrix of the 5x4 gain matrix.



Relative Gain: 10 *
Large Threshold: 50
Small Threshold: 2

Use closed loop

Closed Loop Handles

Collinearity Options

Maximum Number of Submodels: 50000
Maximum Gain Change (%): 10

(●) Use relative gain analysis

Relative Gain Analysis
Relative Gain Threshold: 10
Large Threshold: 50
Small Threshold: 2

( ) Use singular value analysis

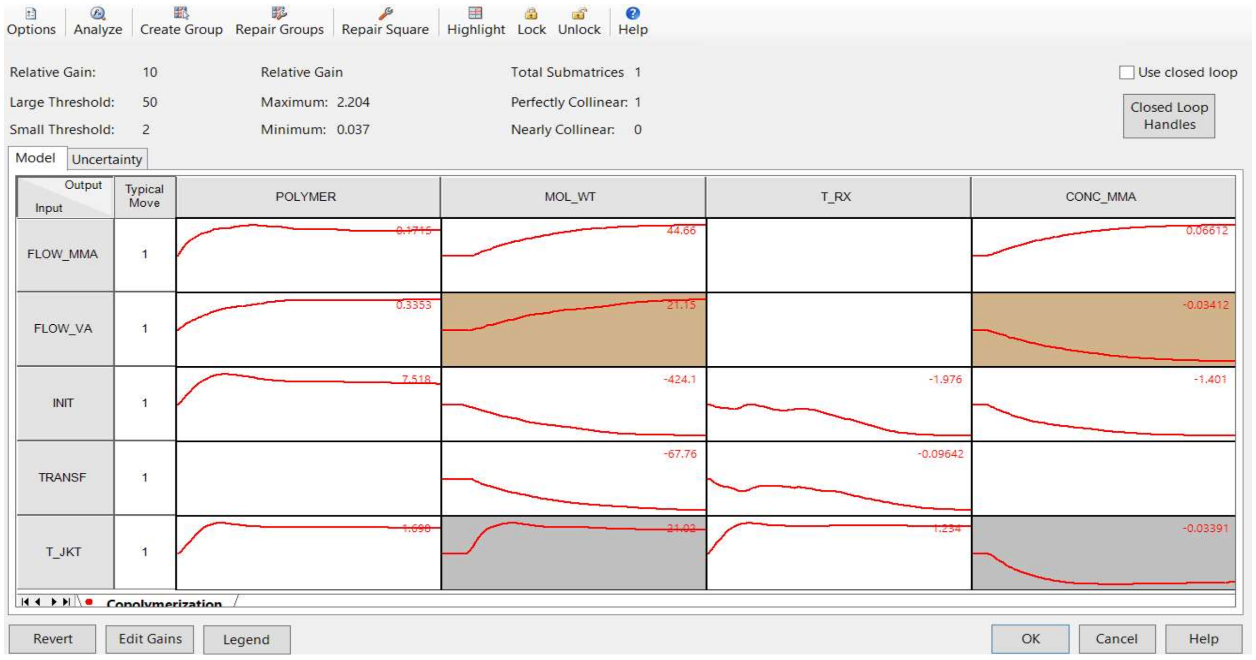
Singular Value Analysis
Collinearity Threshold: 0.002
RGA Equivalent: Calculate 46.6
Minimum Size: 2x2
Maximum Size: 3x3
☐ Ignore perfectly collinear submodels
☑ Exclude groups

Changes to these parameters may affect the computation time.

OK Cancel Help

Figure 8.74 Collinearity analysis options with default settings:
Relative gain analysis (RGA) or singular value analysis (SVA).

Step 2. Analyze and Determine Relationships



Relative Gain: 10
Large Threshold: 50
Small Threshold: 2

Relative Gain
Maximum: 2.204
Minimum: 0.037

Total Submatrices 1
Perfectly Collinear: 1
Nearly Collinear: 0

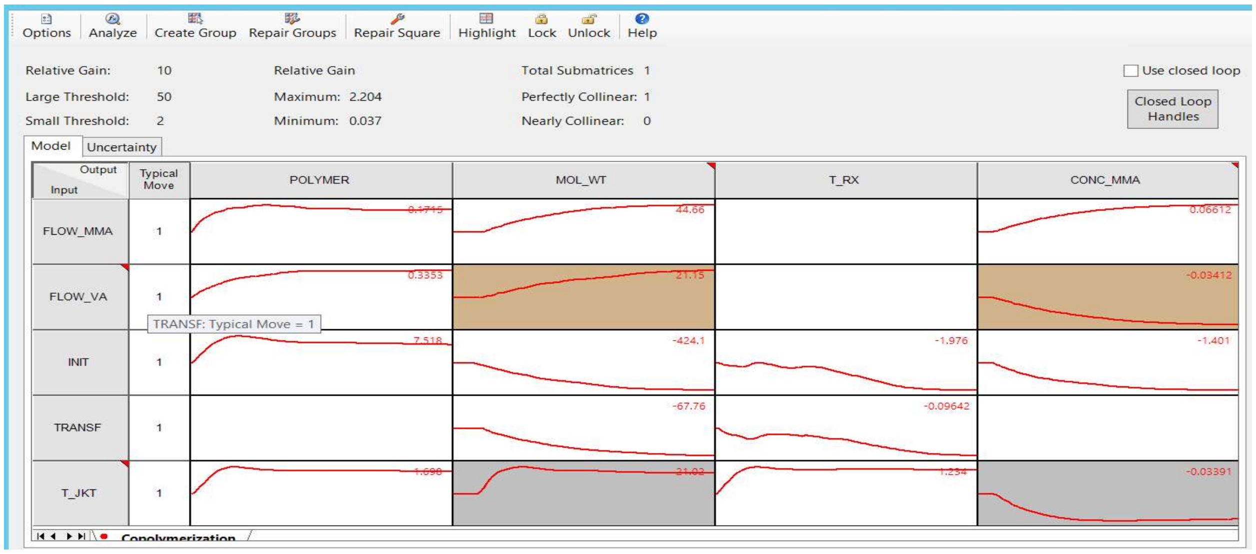
Input \ Output	Typical Move	POLYMER	MOL_WT	T_RX	CONC_MMA
FLOW_MMA	1	0.1715	44.66		0.06612
FLOW_VA	1	0.3353	21.15		-0.03412
INIT	1	7.518	-424.1	-1.976	-1.401
TRANSF	1		-67.76	-0.09642	
T_JKT	1	1.698	21.03	1.234	-0.03391

Figure 8.75 Screen display of the results of collinearity analysis

Figure 8.75 shows that there are two collinear systems, as indicated by the shaded MV-CV pairs: (Flow_VA)-(MOL-WT) with a gain of 21.15 and (T_JKT)-(MOL_WT) with a gain of 21.03; and (FLOW_VA)-(CONC_MMA) with a gain of -0.03412, and (T_JKT)-(CONC_MMA) with a gain of -0.03391.

Step 3. Create Groups

Next, we click on CV names, MOL_WT and CONC_MMA, and on MV names, FLOW_VA and T_JKT to select these variables to form "Parallel Groups". This results in a red triangle on the top right corner of the selected variable name. See Figure 8.76.

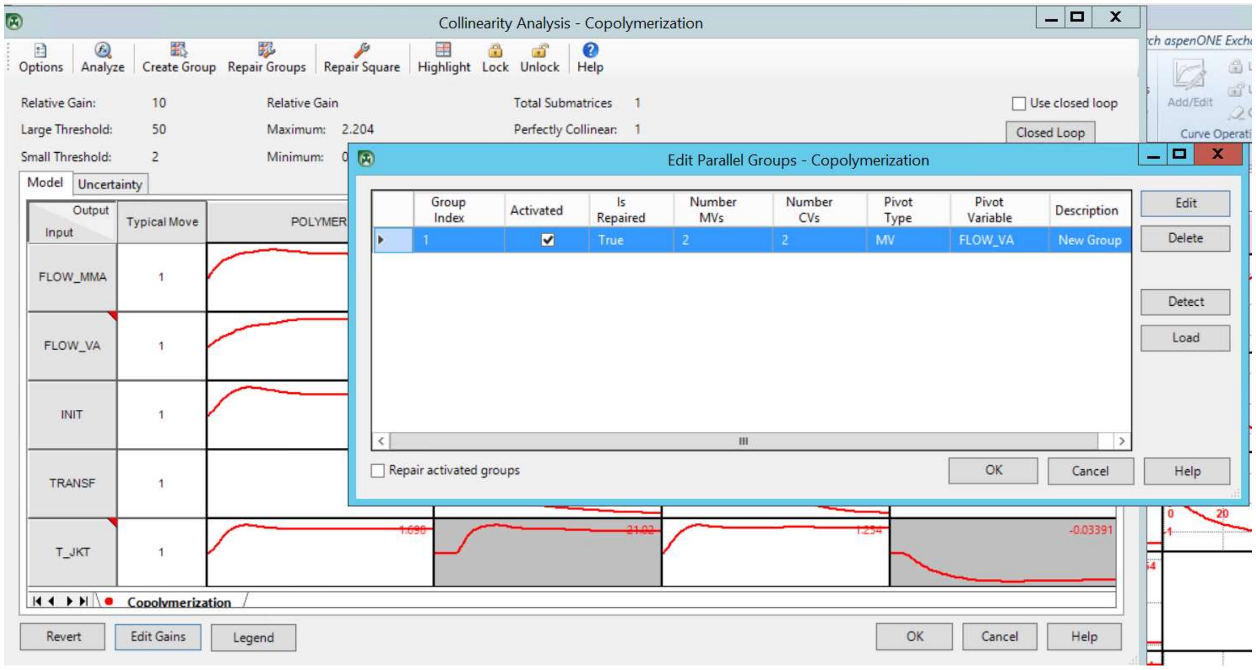


Relative Gain:	10	Relative Gain		Total Submatrices	1
Large Threshold:	50	Maximum:	2.204	Perfectly Collinear:	1
Small Threshold:	2	Minimum:	0.037	Nearly Collinear:	0

Output / Input	Typical Move	POLYMER	MOL_WT	T_RX	CONC_MMA
FLOW_MMA	1	0.1715	44.66		0.06612
FLOW_VA	1	0.3353	21.15		-0.03412
INIT	1	7.518	-424.1	-1.976	-1.401
TRANSF	1		-67.76	-0.09642	
T_JKT	1	1.698	21.02	1.234	-0.03391

Figure 8.76 Choosing the MV and CV to form parallel groups and the resulting red triangles on the top right corners of the variable names

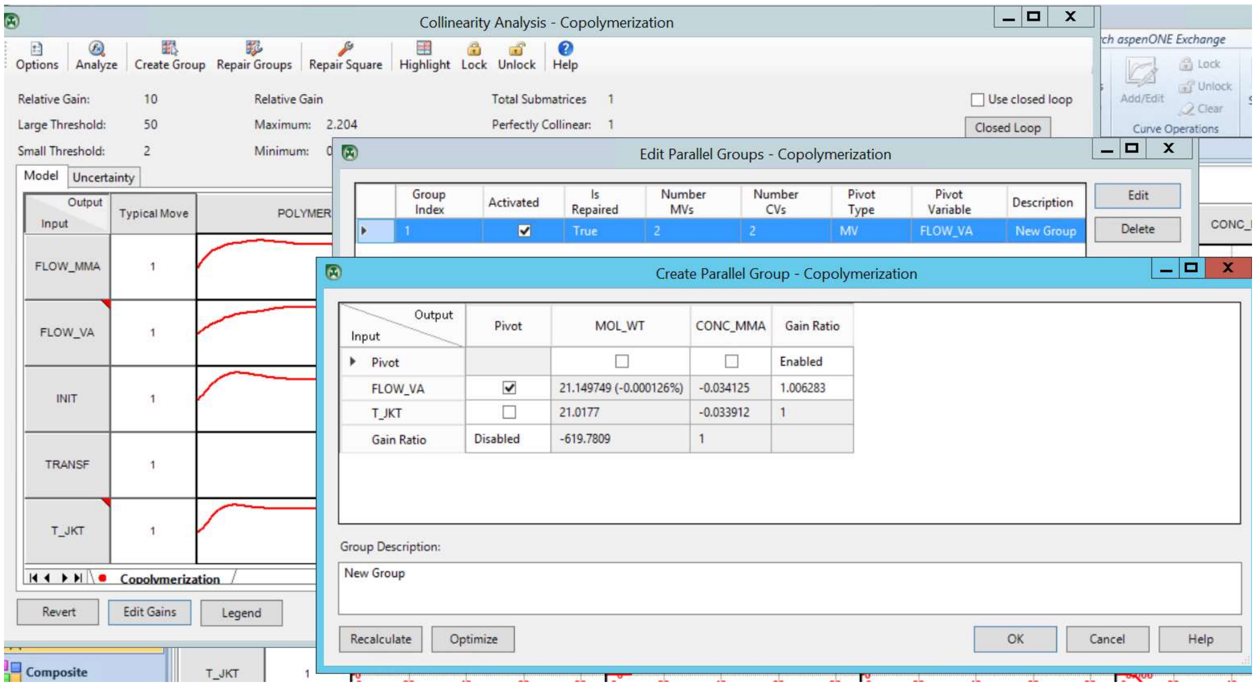
We then click on the "Create Group" button to the top tool bar for collinearity analysis and see the display of "Edit Parallel Groups" screen of Figure 8.77. Clicking on the "Edit" button displayed in Figure 8.77 will show the screen of Figure 8.78.



Relative Gain:	10	Relative Gain		Total Submatrices	1
Large Threshold:	50	Maximum:	2.204	Perfectly Collinear:	1
Small Threshold:	2	Minimum:	0		

Group Index	Activated	Is Repaired	Number MVs	Number CVs	Pivot Type	Pivot Variable	Description
1	✓	True	2	2	MV	FLOW_VA	New Group

Figure 8.77 Display of the "Edit Parallel Groups" screen



Relative Gain: 10
Large Threshold: 50
Small Threshold: 2
Relative Gain Maximum: 2.204
Total Submatrices 1
Perfectly Collinear: 1

Edit Parallel Groups - Copolymerization

	Group Index	Activated	Is Repaired	Number MVs	Number CVs	Pivot Type	Pivot Variable	Description
▸	1	✓	True	2	2	MV	FLOW_VA	New Group

Create Parallel Group - Copolymerization

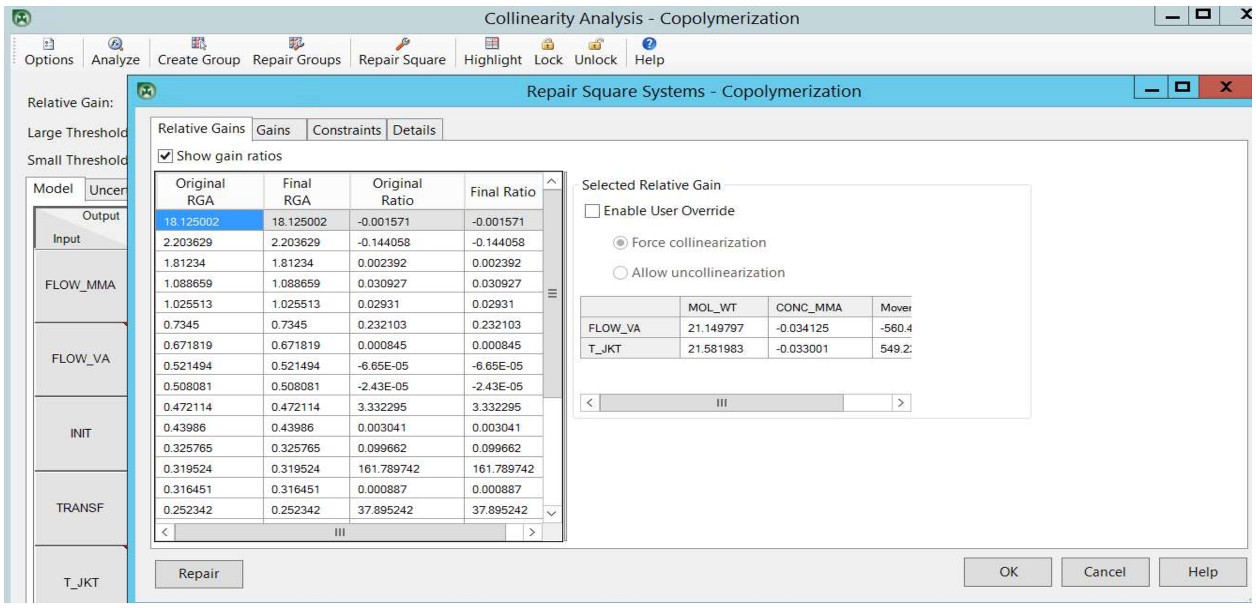
Input \ Output	Pivot	MOL_WT	CONC_MMA	Gain Ratio
Pivot		☐	☐	Enabled
FLOW_VA	✓	21.149749 (-0.000126%)	-0.034125	1.006283
T_JKT	☐	21.0177	-0.033912	1
Gain Ratio	Disabled	-619.7809	1	

Group Description:
New Group

Figure 8.78 Creating parallel groups, choosing the default, FLOW_VA, as the pivot, and clicking on "Recalculate" to determine the required gain changes to collinearize the MVs.

Step 4. Repair Groups and Update Model

We click "OK" in the "Create Parallel Group" folder in Figure 8.78, followed by clicking on "Repair Square" in the top tool bar for collinearity analysis to fix the gain matrix for both remaining square submatrix groups, and the parallel group defined in the previous task. This leads to Figures 8.79 and 8.80, which show the relative gains and the (model) gains, respectively.



Repair Square Systems - Copolymerization

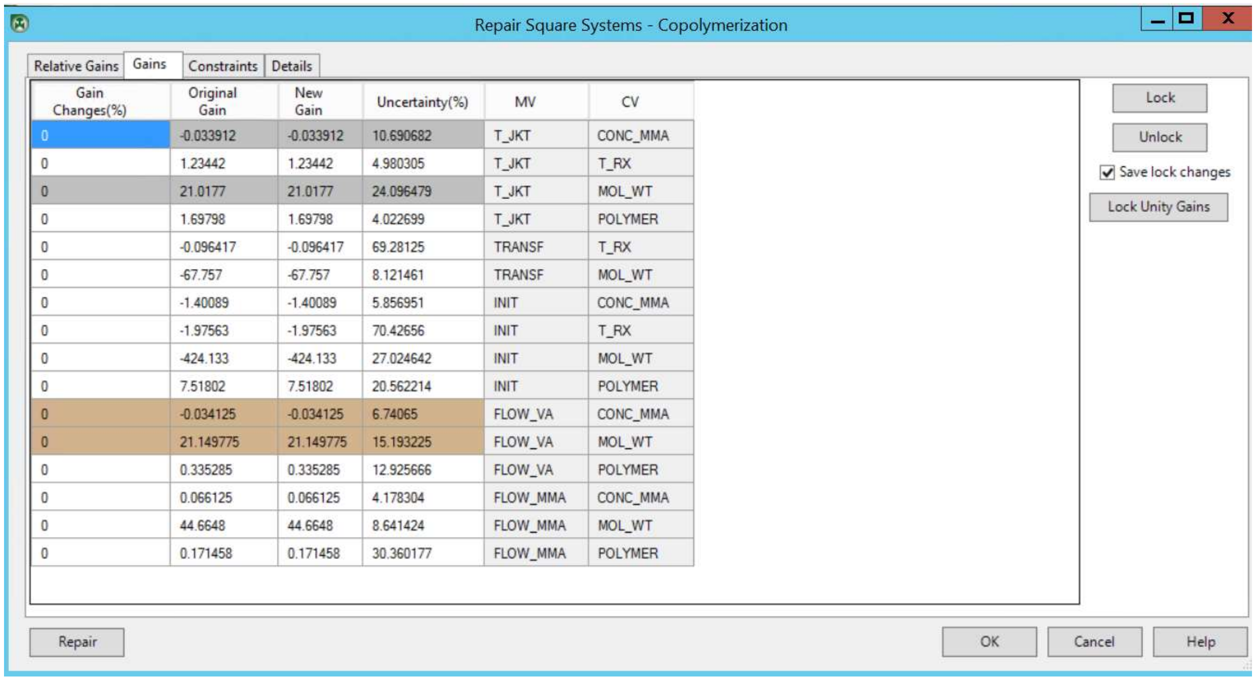
☑ Show gain ratios

Original RGA	Final RGA	Original Ratio	Final Ratio
18.125002	18.125002	-0.001571	-0.001571
2.203629	2.203629	-0.144058	-0.144058
1.81234	1.81234	0.002392	0.002392
1.088659	1.088659	0.030927	0.030927
1.025513	1.025513	0.02931	0.02931
0.7345	0.7345	0.232103	0.232103
0.671819	0.671819	0.000845	0.000845
0.521494	0.521494	-6.65E-05	-6.65E-05
0.508081	0.508081	-2.43E-05	-2.43E-05
0.472114	0.472114	3.332295	3.332295
0.43986	0.43986	0.003041	0.003041
0.325765	0.325765	0.099662	0.099662
0.319524	0.319524	161.789742	161.789742
0.316451	0.316451	0.000887	0.000887
0.252342	0.252342	37.895242	37.895242

Selected Relative Gain
☐ Enable User Override
(●) Force collinearization
( ) Allow uncollinearization

	MOL_WT	CONC_MMA	Mover
FLOW_VA	21.149797	-0.034125	-560.4
T_JKT	21.581983	-0.033001	549.2

Figure 8.79 Display of relative gains after clicking on "Repair Square"


Repair Square Systems - Copolymerization

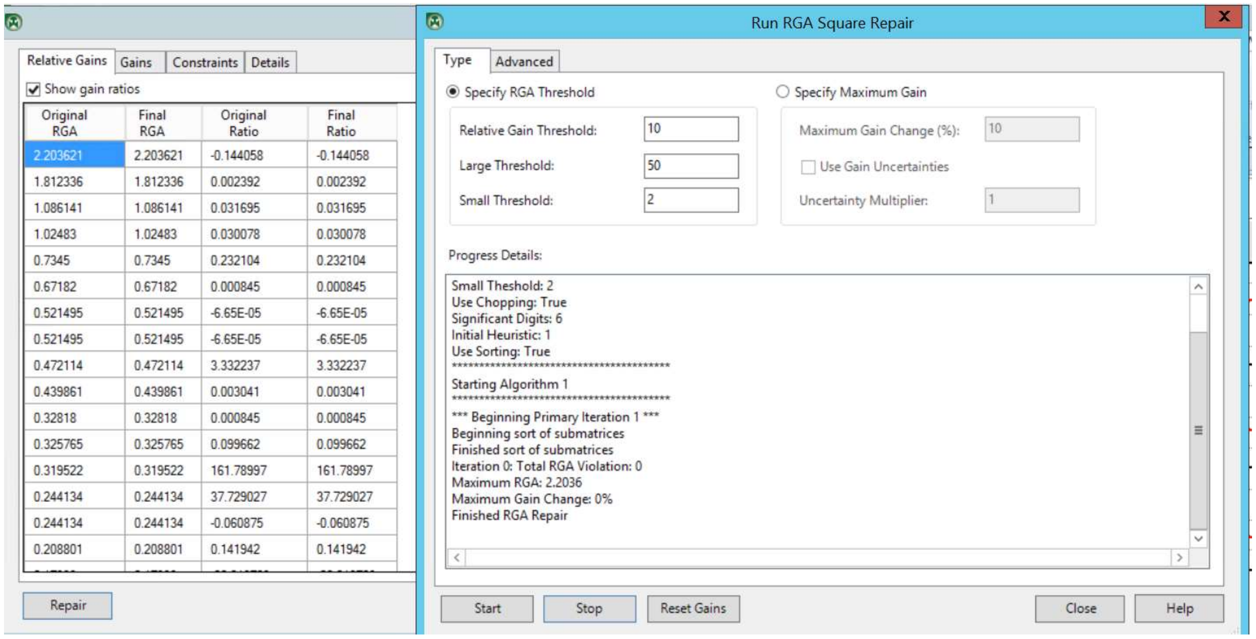
Relative Gains | Gains | Constraints | Details

Gain Changes(%)	Original Gain	New Gain	Uncertainty(%)	MV	CV
0	-0.033912	-0.033912	10.690682	T_JKT	CONC_MMA
0	1.23442	1.23442	4.980305	T_JKT	T_RX
0	21.0177	21.0177	24.096479	T_JKT	MOL_WT
0	1.69798	1.69798	4.022699	T_JKT	POLYMER
0	-0.096417	-0.096417	69.28125	TRANSF	T_RX
0	-67.757	-67.757	8.121461	TRANSF	MOL_WT
0	-1.40089	-1.40089	5.856951	INIT	CONC_MMA
0	-1.97563	-1.97563	70.42656	INIT	T_RX
0	-424.133	-424.133	27.024642	INIT	MOL_WT
0	7.51802	7.51802	20.562214	INIT	POLYMER
0	-0.034125	-0.034125	6.74065	FLOW_VA	CONC_MMA
0	21.149775	21.149775	15.193225	FLOW_VA	MOL_WT
0	0.335285	0.335285	12.925666	FLOW_VA	POLYMER
0	0.066125	0.066125	4.178304	FLOW_MMA	CONC_MMA
0	44.6648	44.6648	8.641424	FLOW_MMA	MOL_WT
0	0.171458	0.171458	30.360177	FLOW_MMA	POLYMER

☑ Save lock changes

Figure 8.80 Display of (model) gains after clicking on "Repair Square"

We then click on "Repair" followed by "Start". Figure 8.81 shows the Start and Finish of RGA (relative gain array) repair.


Relative Gains | Gains | Constraints | Details

☑ Show gain ratios

Original RGA	Final RGA	Original Ratio	Final Ratio
2.203621	2.203621	-0.144058	-0.144058
1.812336	1.812336	0.002392	0.002392
1.086141	1.086141	0.031695	0.031695
1.02483	1.02483	0.030078	0.030078
0.7345	0.7345	0.232104	0.232104
0.67182	0.67182	0.000845	0.000845
0.521495	0.521495	-6.65E-05	-6.65E-05
0.521495	0.521495	-6.65E-05	-6.65E-05
0.472114	0.472114	3.332237	3.332237
0.439861	0.439861	0.003041	0.003041
0.32818	0.32818	0.000845	0.000845
0.325765	0.325765	0.099662	0.099662
0.319522	0.319522	161.78997	161.78997
0.244134	0.244134	37.729027	37.729027
0.244134	0.244134	-0.060875	-0.060875
0.208801	0.208801	0.141942	0.141942

Run RGA Square Repair

Type | Advanced

(● Specify RGA Threshold) ○ Specify Maximum Gain

Relative Gain Threshold: 10

Large Threshold: 50

Small Threshold: 2

Maximum Gain Change (%): 10

☐ Use Gain Uncertainties

Uncertainty Multiplier: 1

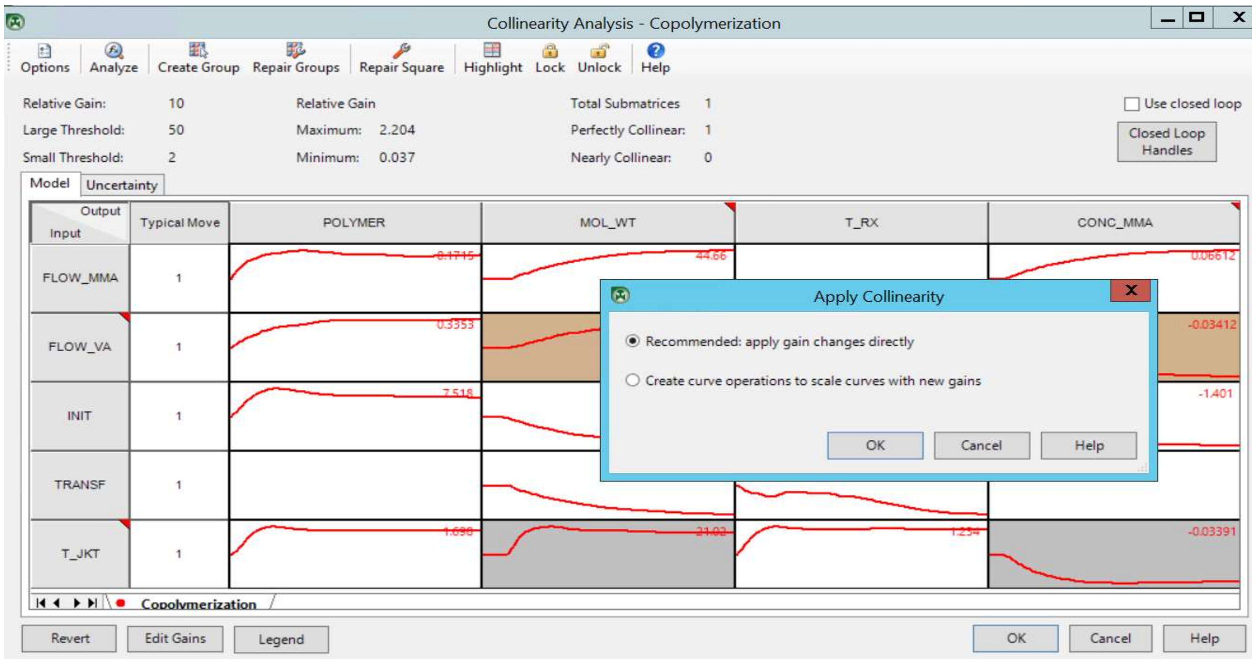
Progress Details:

```
Small Theshold: 2
Use Chopping: True
Significant Digits: 6
Initial Heuristic: 1
Use Sorting: True
****************************************
Starting Algorithm 1
****************************************
*** Beginning Primary Iteration 1 ***
Beginning sort of submatrices
Finished sort of submatrices
Iteration 0: Total RGA Violation: 0
Maximum RGA: 2.2036
Maximum Gain Change: 0%
Finished RGA Repair
```

Figure 8.81 Run RGA square repair.

Step 5. Review and Save Gains to the Master Model

Figure 8.82 asks us to apply the recommended changes. Click OK. This leads to Figure 8.83. We place the mouse inside the figure, right-click to open the options, and choose "Copy Gains".


Collinearity Analysis - Copolymerization

Relative Gain: 10
Large Threshold: 50
Small Threshold: 2

Relative Gain
Maximum: 2.204
Minimum: 0.037

Total Submatrices 1
Perfectly Collinear: 1
Nearly Collinear: 0

Use closed loop

Apply Collinearity

(●) Recommended: apply gain changes directly
( ) Create curve operations to scale curves with new gains

Figure 8.82 Display of “Apply Collinearity” and apply gain changes directly.

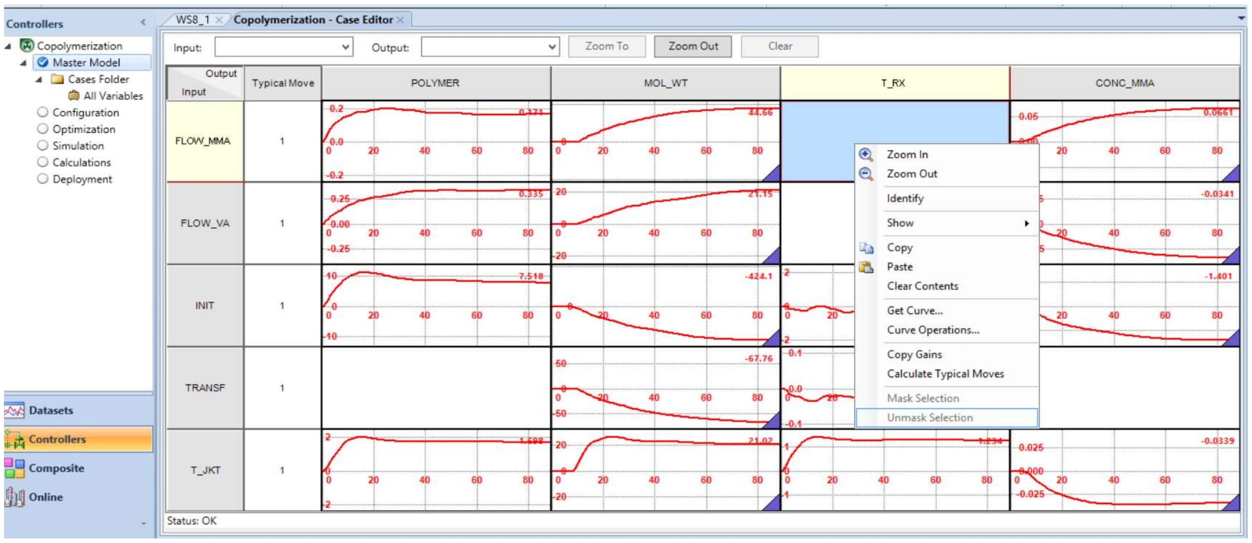
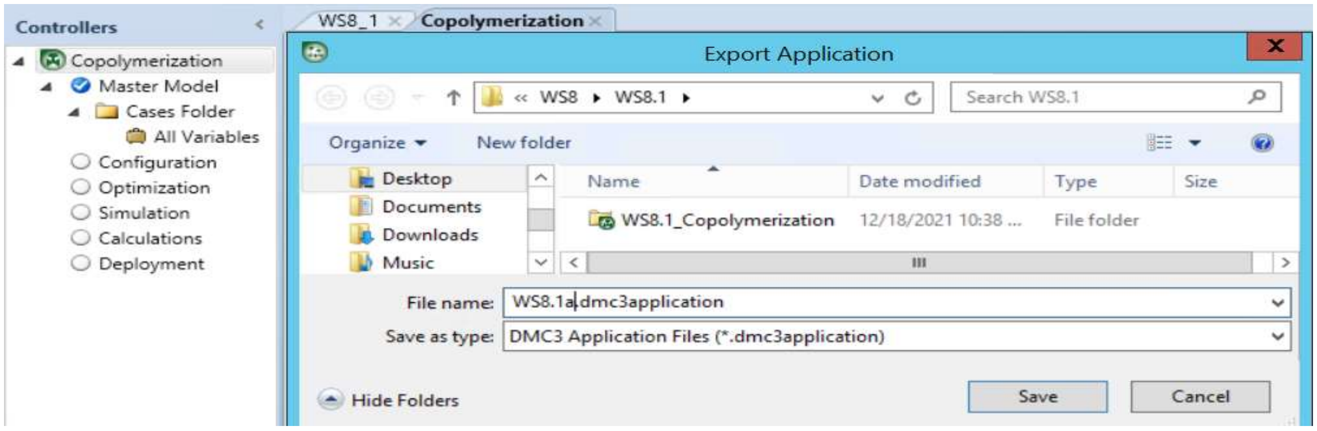


Figure 8.83 “Copy Gains” of the model after collinearity analysis

The final model gains are as follows:

	POLYMER	MOL_WT	T_RX	CONC_MMA
FLOW_MMA	0.171458	44.6648	0	0.066125
FLOW_VA	0.335285	21.14978	0	-0.034125
INIT	7.51802	-424.133	-1.97563	-1.40089
TRANSF	0	-67.757	0.096417	0
T_JKT	1.69798	21.0177	1.23442	-0.033912

Before we continue further, we export the current controller application and save it according to the following path: Controller -> Copolymerization ->Right-click: Export -> Save as ***WS8.1a.dmc3application*** (see Figure 8.84).



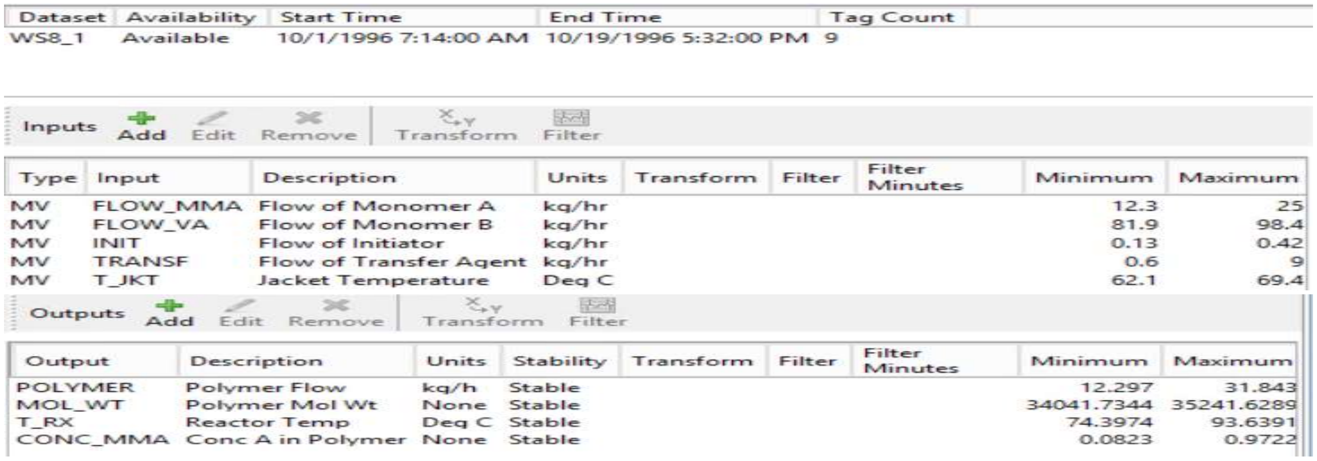
File name: WS8.1a.dmc3application

Save as type: DMC3 Application Files (*.dmc3application)

Figure 8.84 Export and save the controller model as ***WS8.1a.dmc3application.***

8.2.13 Open-Loop Prediction and Prediction Error (Model Bias)

We first identify the units and ranges of MVs and CVs in our dataset before continuing with open-loop prediction. Following the path: Controllers ->Copolymerization -> Master Model -> Cases Folder ->All variables, we see the units and ranges of MVs and CVs displayed in Figure 8.85.



Dataset	Availability	Start Time	End Time	Tag Count
WS8_1	Available	10/1/1996 7:14:00 AM	10/19/1996 5:32:00 PM	9

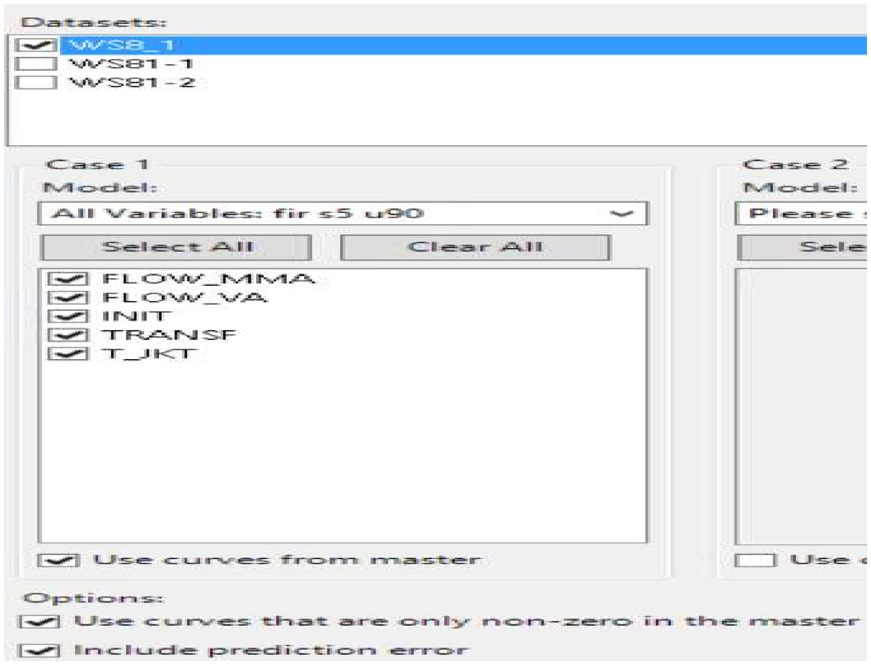
Type	Input	Description	Units	Transform	Filter	Filter Minutes	Minimum	Maximum
MV	FLOW_MMA	Flow of Monomer A	kg/hr				12.3	25
MV	FLOW_VA	Flow of Monomer B	kg/hr				81.9	98.4
MV	INIT	Flow of Initiator	kg/hr				0.13	0.42
MV	TRANSF	Flow of Transfer Agent	kg/hr				0.6	9
MV	T_JKT	Jacket Temperature	Deg C				62.1	69.4

Output	Description	Units	Stability	Transform	Filter	Filter Minutes	Minimum	Maximum
POLYMER	Polymer Flow	kg/h	Stable				12.297	31.843
MOL_WT	Polymer Mol Wt	None	Stable				34041.7344	35241.6289
T_RX	Reactor Temp	Deg C	Stable				74.3974	93.6391
CONC_MMA	Conc A in Polymer	None	Stable				0.0823	0.9722

Figure 8.85 Units and ranges of MVs and CVs in copolymerization controller model.

To proceed with predictions, we follow the path: Controller->Copolymerization ->Master Model ->Top ribbons: Master Model Actions ->Compare -> Compare predictions -> See Figure 8.86 -> Generate predictions-> Close -> Top ribbons: Zoom-In ->Figure 8.87. We note from Figure 8.85 from the the dataset, the polymer production rate, POLYMER, varies from 12.297 to 31.843 kg/hr. This is the range of POLYMER in Figure 8.87. To understand Figure 8.87, we note the difference between the prediction (blue) and measurement (red) gives the prediction error (pink). In the figure, *we should read the positive and negative values for the prediction error beginning from the baseline of zero prediction error at 20 kg/hr*. Figure 8.88 shows the prediction plot for all four CVs.

A significant result from the prediction analysis is the scatter plot. Predictions should be unbiased over the entire dataset range. It is important to review the scatter plot. Figure 8.89 illustrates that the scatter plots for all four CVs in our copolymerization controller appear to be acceptable.



Datasets:

- [✓] WS8_1
- [ ] WS81-1
- [ ] WS81-2

Case 1

Model: All Variables: fir s5 u90

Select All | Clear All

- [✓] FLOW_MMA
- [✓] FLOW_VA
- [✓] INIT
- [✓] TRANSF
- [✓] T_JKT

[✓] Use curves from master

Case 2

Model: Please

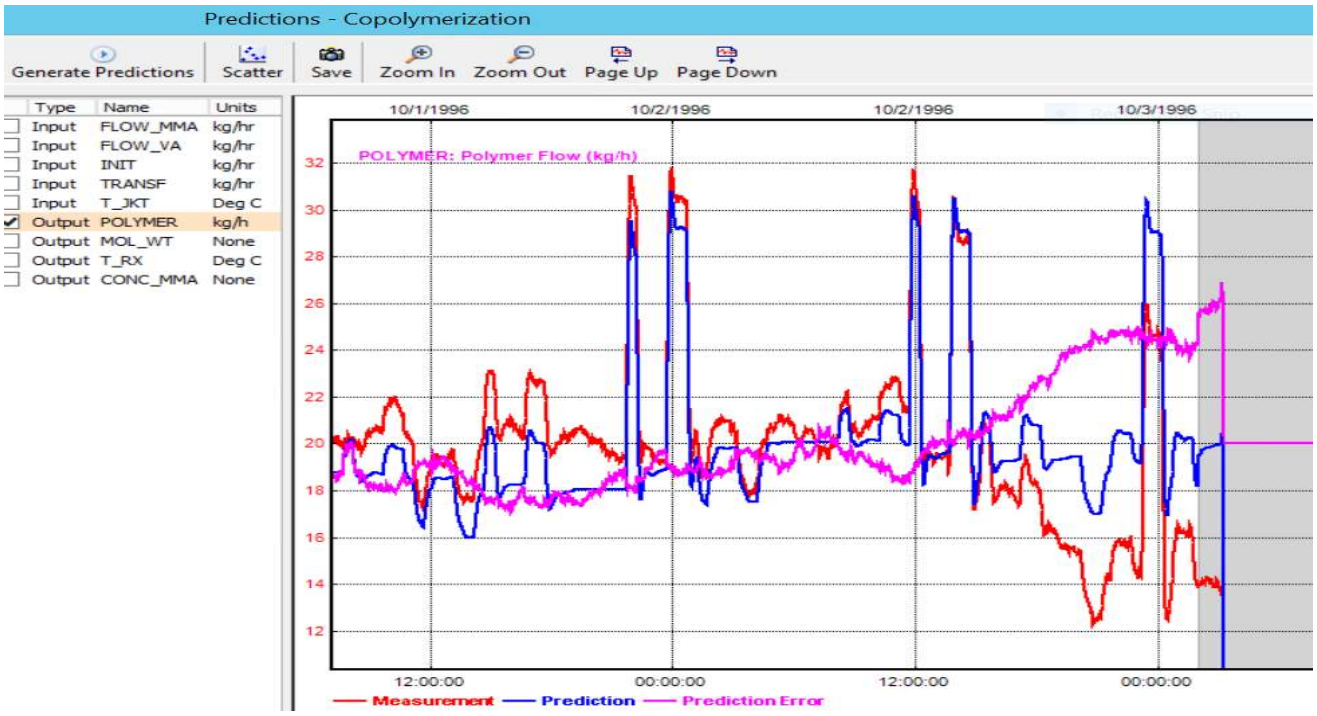
Sele

[ ] Use

Options:

- [✓] Use curves that are only non-zero in the master
- [✓] Include prediction error

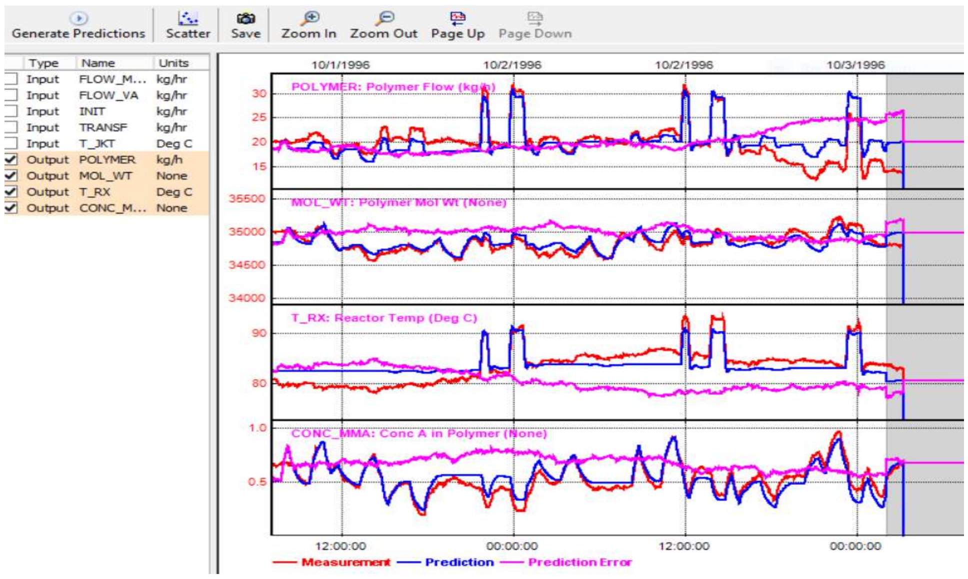
Figure 9.86 Setup for prediction run



Predictions - Copolymerization

	Type	Name	Units
[ ]	Input	FLOW_MMA	kg/hr
[ ]	Input	FLOW_VA	kg/hr
[ ]	Input	INIT	kg/hr
[ ]	Input	TRANSF	kg/hr
[ ]	Input	T_JKT	Deg C
[✓]	Output	POLYMER	kg/h
[ ]	Output	MOL_WT	None
[ ]	Output	T_RX	Deg C
[ ]	Output	CONC_MMA	None

Figure 8.87 Comparison of CV prediction with measurement and illustration of prediction error



	Type	Name	Units
☐	Input	FLOW_M…	kg/hr
☐	Input	FLOW_VA	kg/hr
☐	Input	INIT	kg/hr
☐	Input	TRANSF	kg/hr
☐	Input	T_JKT	Deg C
✓	Output	POLYMER	kg/h
✓	Output	MOL_WT	None
✓	Output	T_RX	Deg C
✓	Output	CONC_M…	None

Figure 8.88 Prediction plots for all four CVs

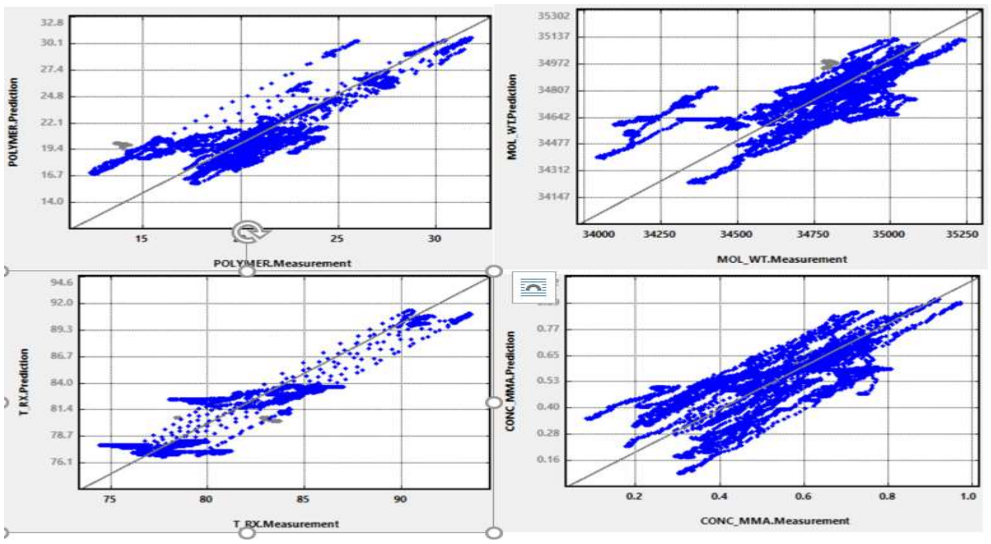
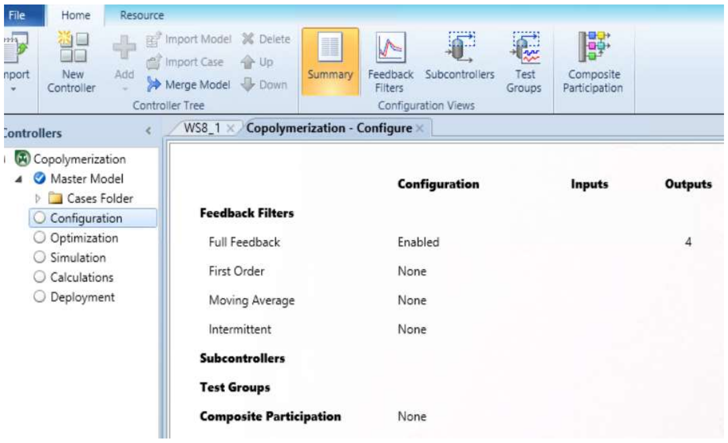


Figure 8.89 Scatter plots for all four CVs

8.2.14 DMC3 Builder Task 2: Configuration – Model Configuration

The model configuration task involves the specifications of: (1) feedback filters for prediction errors (discussed previously in Section 8.1.5 for prediction error filtering); (2) subcontrollers; (3) test groups; and (4) composite participation. See Figure 8.90.

Aspen DMC3 allows a controller to be subdivided into multiple units of MVs and CVs for operational convenience in turning multiple variables ON or OFF at the same time. These units of MVs and CVs are known as *subcontrollers*. For example, we may classify a large DMC3 controller for an ethylene production train to have the following subcontrollers: (1) ethylene cracked gas compressor and quench; (2) cold-box and demethanizer, refrigeration compressors; and (3) de-ethanizer and C2 splitter. If subcontrollers are used, every MV in the controller must be a member of one and only one subcontroller. Every CV in the controller must be a member of at least one subcontroller, although a CV may belong to more than one subcontroller. Feedfirwards do n0t belong to subcontrollers. Our current woekshop deals with a small controller and does not have subcontrollers.



	Configuration	Inputs	Outputs
Feedback Filters			
Full Feedback	Enabled		4
First Order	None		
Moving Average	None		
Intermittent	None		
Subcontrollers			
Test Groups			
Composite Participation	None		

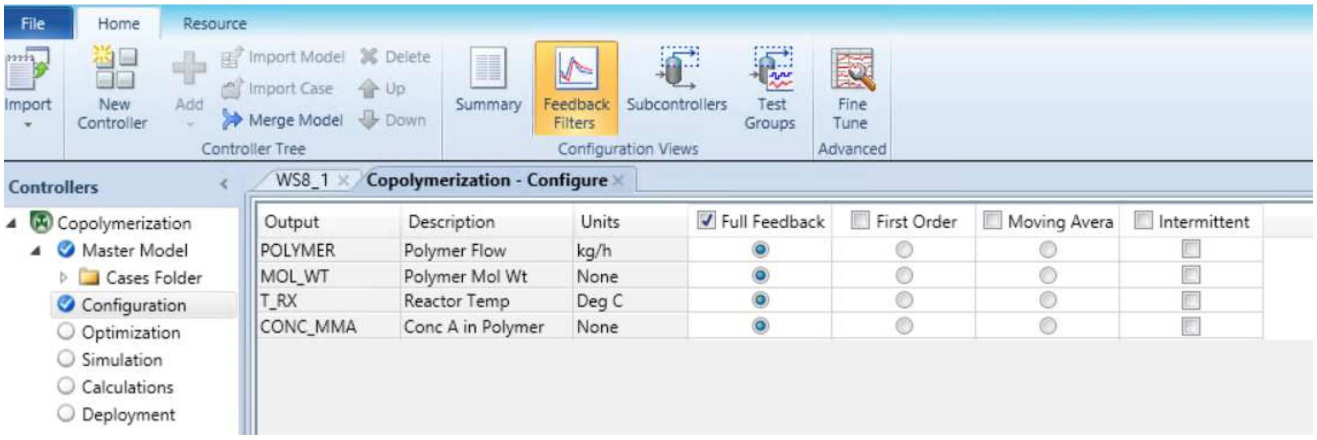
Figure 8.90 Model configuration task in DMC3.

must be a member of at least one subcontroller, although a CV may belong to more than one subcontroller. Feedforwards do not belong to subcontrollers. Our current workshop deals with a small controller, and does not have subcontrollers.

Aspen DMC3 SmartStep application uses primitive process models to predict the behavior of the tested process. When the tester application automatically steps an independent variable, it also keeps dependent variables within their prescribed limits. The result is a constrained step test where all constraints are honored. A SmartStep application uses the concept of test groups to help maximize testing efficiency. *A test group* consists of MVs and CVs for which step tests are performed. The current workshop does not involve a SmartStep application with test groups.

The last DMC3 model configuration application involves *composite controllers*. An Aspen DMC3 composite application facilitates the coordinated action of multiple DMC3 controller applications. It works by providing consistently calculated steady-state MV and CV targets to participating controllers. A Composite application is typically used in the following scenarios: (1) a large part of the unit is under the control of several controller applications; and (2) controllers on separate processes, with significantly different times to steady-state, are linked by common constraints. A DMC3 composite application utilizes the same steady-state optimization technology that is embedded in FIR controller applications. The composite suite variable set is a superset of all MVs, FFs and CVs in all participating controllers. The steady-state solution obtained from the DMC3 composite application, therefore, honors the constraints and utilizes the MVs of all the participating controllers. Our current workshop does not include composite application.

Figure 8.91 illustrates that we specify the default option of “full feedback “of prediction error (model bias), in which we calculate the difference between the current measurement and the current prediction to calculate a bias that is applied to each element of the prediction error. This is exactly what we previously demonstrated in Figures 8.9 to 8.11, Section 8.1.2a. Figure 8.91 also shows the options of “First order” and “Moving average” filters, which were previously explained in Section 8.1.5 for prediction error filtering. Lastly, the check boxes in the option of “Intermittent” in Figure 8.91 refers to those CVs for which a new measurement is not available in each controller execution cycle. This is typically the case of a discretely sampled variable, such as composition from a stream analyzer.

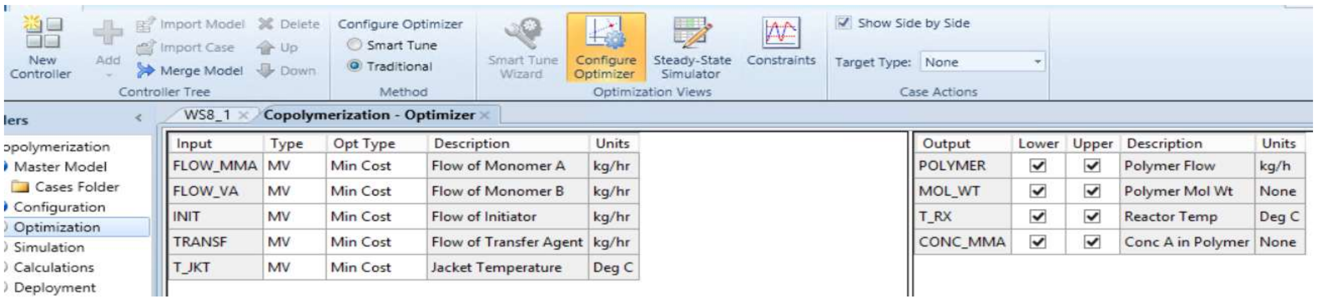


Output	Description	Units	☑ Full Feedback	☐ First Order	☐ Moving Avera	☐ Intermittent
POLYMER	Polymer Flow	kg/h	●	○	○	☐
MOL_WT	Polymer Mol Wt	None	●	○	○	☐
T_RX	Reactor Temp	Deg C	●	○	○	☐
CONC_MMA	Conc A in Polymer	None	●	○	○	☐

Figure 8.91 Specification of "full feedback" option for prediction error feedback in model configuration.

8.2.15 DMC3 Builder Task 2: Configuration – Configuring the Steady-State Optimization

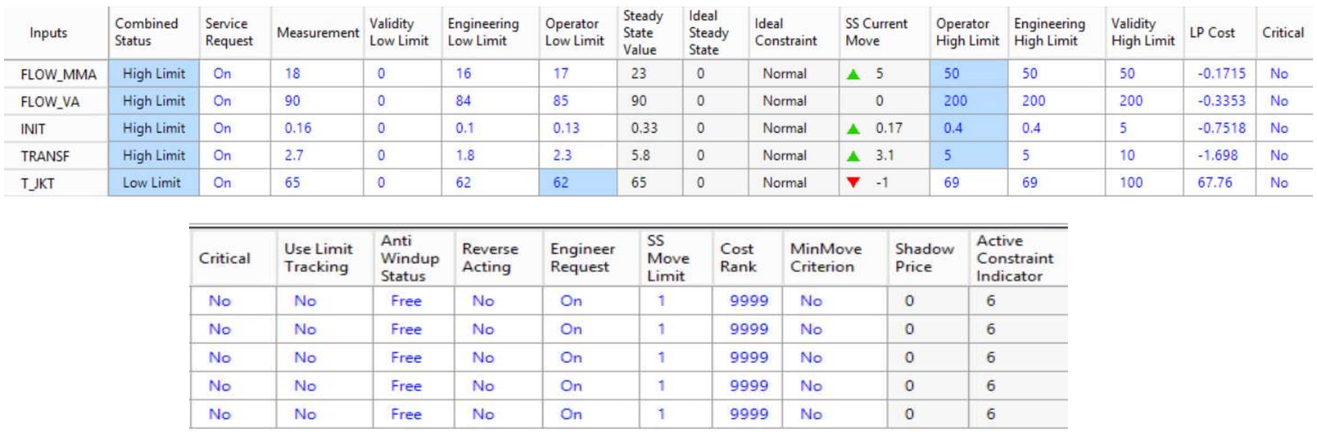
We follow Section 8.1.6a-b to configure the steady-state optimizer. Figure 8.92 illustrates the interface to configure the SS optimizer. Figures 8.93a-b show the input entries for MVs and CVs for the steady-state simulator, respectively.



Input	Type	Opt Type	Description	Units
FLOW_MMA	MV	Min Cost	Flow of Monomer A	kg/hr
FLOW_VA	MV	Min Cost	Flow of Monomer B	kg/hr
INIT	MV	Min Cost	Flow of Initiator	kg/hr
TRANSF	MV	Min Cost	Flow of Transfer Agent	kg/hr
T_JKT	MV	Min Cost	Jacket Temperature	Deg C

Output	Lower	Upper	Description	Units
POLYMER	✓	✓	Polymer Flow	kg/h
MOL_WT	✓	✓	Polymer Mol Wt	None
T_RX	✓	✓	Reactor Temp	Deg C
CONC_MMA	✓	✓	Conc A in Polymer	None

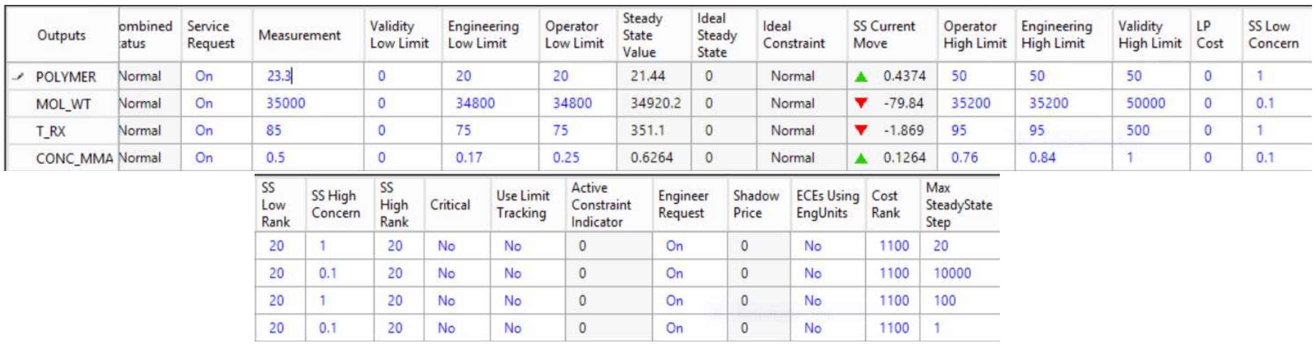
Figure 8.92 Configuration of the steady-state optimizer



Inputs	Combined Status	Service Request	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Ideal Steady State	Ideal Constraint	SS Current Move	Operator High Limit	Engineering High Limit	Validity High Limit	LP Cost	Critical
FLOW_MMA	High Limit	On	18	0	16	17	23	0	Normal	▲ 5	50	50	50	-0.1715	No
FLOW_VA	High Limit	On	90	0	84	85	90	0	Normal	0	200	200	200	-0.3353	No
INIT	High Limit	On	0.16	0	0.1	0.13	0.33	0	Normal	▲ 0.17	0.4	0.4	5	-0.7518	No
TRANSF	High Limit	On	2.7	0	1.8	2.3	5.8	0	Normal	▲ 3.1	5	5	10	-1.698	No
T_JKT	Low Limit	On	65	0	62	62	65	0	Normal	▼ -1	69	69	100	67.76	No

Critical	Use Limit Tracking	Anti Windup Status	Reverse Acting	Engineer Request	SS Move Limit	Cost Rank	MinMove Criterion	Shadow Price	Active Constraint Indicator
No	No	Free	No	On	1	9999	No	0	6
No	No	Free	No	On	1	9999	No	0	6
No	No	Free	No	On	1	9999	No	0	6
No	No	Free	No	On	1	9999	No	0	6
No	No	Free	No	On	1	9999	No	0	6

Figure 8.93a Input entries for MVs for steady-state simulator



Outputs	ombined atus	Service Request	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Ideal Steady State	Ideal Constraint	SS Current Move	Operator High Limit	Engineering High Limit	Validity High Limit	LP Cost	SS Low Concern
POLYMER	Normal	On	23.3	0	20	20	21.44	0	Normal	▲ 0.4374	50	50	50	0	1
MOL_WT	Normal	On	35000	0	34800	34800	34920.2	0	Normal	▼ -79.84	35200	35200	50000	0	0.1
T_RX	Normal	On	85	0	75	75	351.1	0	Normal	▼ -1.869	95	95	500	0	1
CONC_MMA	Normal	On	0.5	0	0.17	0.25	0.6264	0	Normal	▲ 0.1264	0.76	0.84	1	0	0.1

SS Low Rank	SS High Concern	SS High Rank	Critical	Use Limit Tracking	Active Constraint Indicator	Engineer Request	Shadow Price	ECEs Using EngUnits	Cost Rank	Max SteadyState Step
20	1	20	No	No	0	On	0	No	1100	20
20	0.1	20	No	No	0	On	0	No	1100	10000
20	1	20	No	No	0	On	0	No	1100	100
20	0.1	20	No	No	0	On	0	No	1100	1

Figure 8.93b Input entries for CVs for steady-state simulator

8.2.16 DMC3 Builder Task 3: Optimization – Performing the Steady-State Optimization

We initialize the SS optimizer tuning by clicking on "Initialize Tuning" button. We choose the dataset WS8_1, uncheck "initialize dynamic tuning", and click "OK", following by clicking on "Calculate" button (see Figure 8.94). This results of CV targets from the SS optimization appear in Figure 8.95.

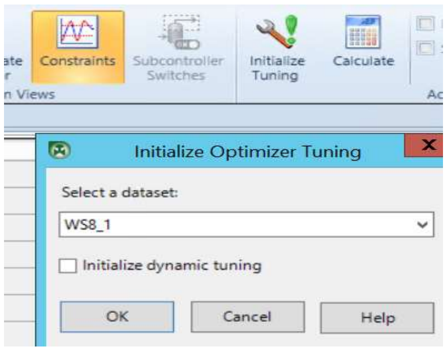
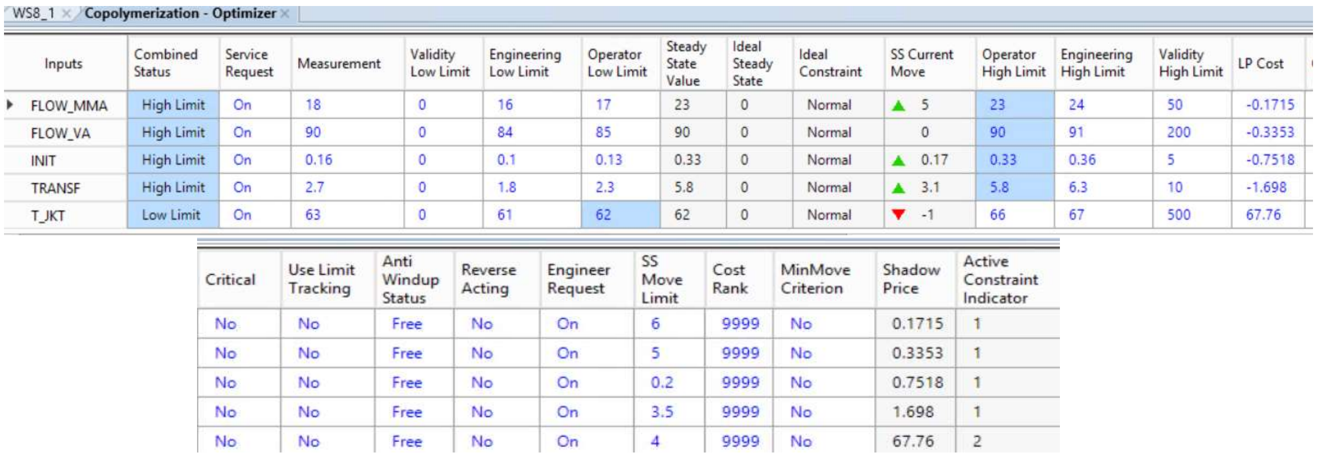


Figure 8.94 Initialize optimizer tuning and calculate.

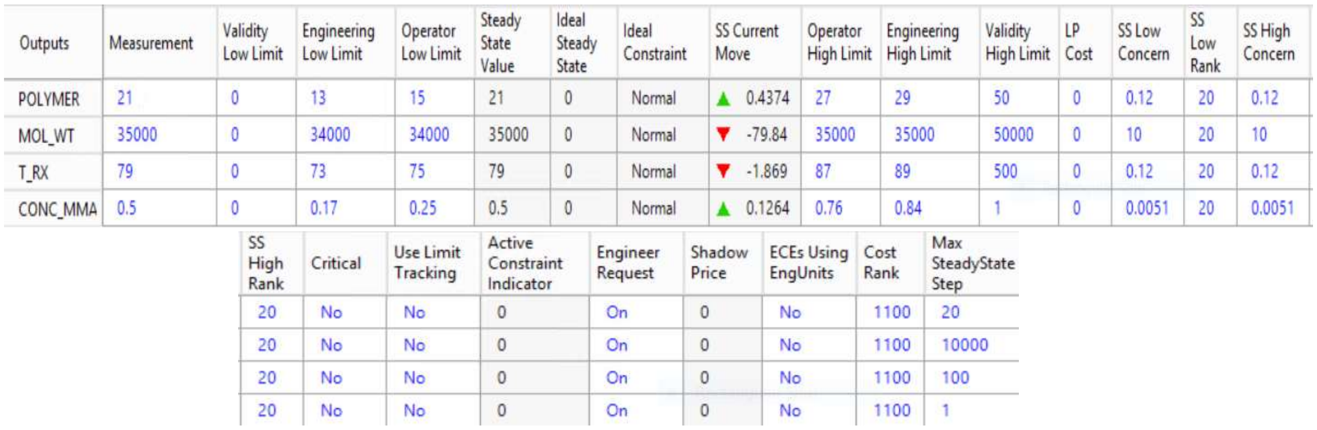


WS8_1 × Copolymerization - Optimizer ×

Inputs	Combined Status	Service Request	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Ideal Steady State	Ideal Constraint	SS Current Move	Operator High Limit	Engineering High Limit	Validity High Limit	LP Cost
FLOW_MMA	High Limit	On	18	0	16	17	23	0	Normal	▲ 5	23	24	50	-0.1715
FLOW_VA	High Limit	On	90	0	84	85	90	0	Normal	0	90	91	200	-0.3353
INIT	High Limit	On	0.16	0	0.1	0.13	0.33	0	Normal	▲ 0.17	0.33	0.36	5	-0.7518
TRANSF	High Limit	On	2.7	0	1.8	2.3	5.8	0	Normal	▲ 3.1	5.8	6.3	10	-1.698
T_JKT	Low Limit	On	63	0	61	62	62	0	Normal	▼ -1	66	67	500	67.76

Critical	Use Limit Tracking	Anti Windup Status	Reverse Acting	Engineer Request	SS Move Limit	Cost Rank	MinMove Criterion	Shadow Price	Active Constraint Indicator
No	No	Free	No	On	6	9999	No	0.1715	1
No	No	Free	No	On	5	9999	No	0.3353	1
No	No	Free	No	On	0.2	9999	No	0.7518	1
No	No	Free	No	On	3.5	9999	No	1.698	1
No	No	Free	No	On	4	9999	No	67.76	2

Figure 8.95 MV results of steady-state optimization using the current configuration and tuning



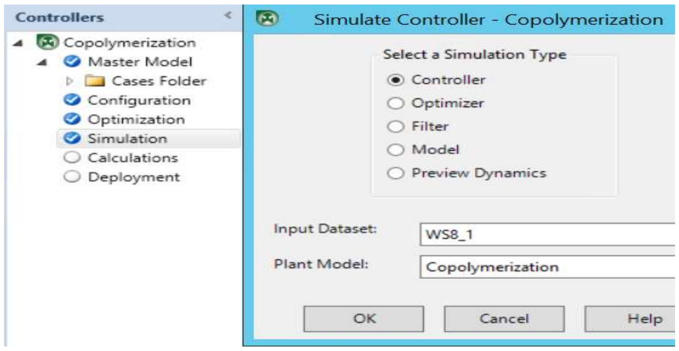
Outputs	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Ideal Steady State	Ideal Constraint	SS Current Move	Operator High Limit	Engineering High Limit	Validity High Limit	LP Cost	SS Low Concern	SS Low Rank	SS High Concern
POLYMER	21	0	13	15	21	0	Normal	▲ 0.4374	27	29	50	0	0.12	20	0.12
MOL_WT	35000	0	34000	34000	35000	0	Normal	▼ -79.84	35000	35000	50000	0	10	20	10
T_RX	79	0	73	75	79	0	Normal	▼ -1.869	87	89	500	0	0.12	20	0.12
CONC_MMA	0.5	0	0.17	0.25	0.5	0	Normal	▲ 0.1264	0.76	0.84	1	0	0.0051	20	0.0051

SS High Rank	Critical	Use Limit Tracking	Active Constraint Indicator	Engineer Request	Shadow Price	ECEs Using EngUnits	Cost Rank	Max SteadyState Step
20	No	No	0	On	0	No	1100	20
20	No	No	0	On	0	No	1100	10000
20	No	No	0	On	0	No	1100	100
20	No	No	0	On	0	No	1100	1

Figure 8.96 CV results of steady-state optimization using the current configuration and tuning

8.2.17 DMC3 Builder Task 4: Simulation – Configuring and Simulating the Dynamic Controller

We follow Section 8.1.6c-e to configure the dynamic controller. Figures 8.97a-c shows how to initialize the controller simulation. Figures 8.98a-c show the input entries for MVs and CVs, including operating values and tuning values. After completing the entries displayed in Figures 8.97a-c, we save the simulation file as ***WS8.1_BaseCase.dmc3application.***

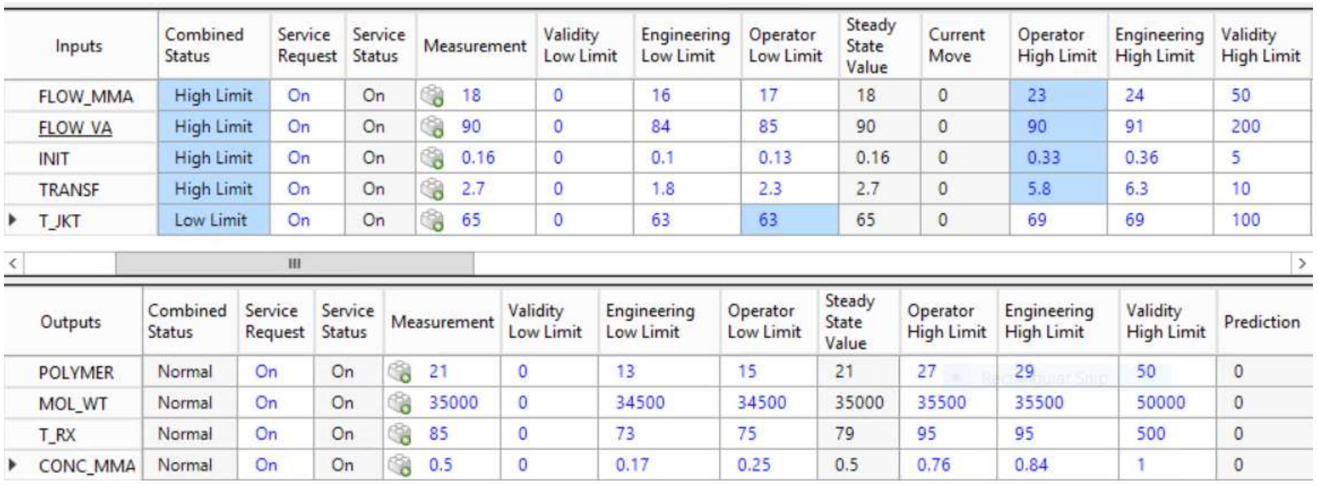


Select a Simulation Type
- [✓ Controller]
- Optimizer
- Filter
- Model
- Preview Dynamics

Input Dataset: WS8_1

Plant Model: Copolymerization

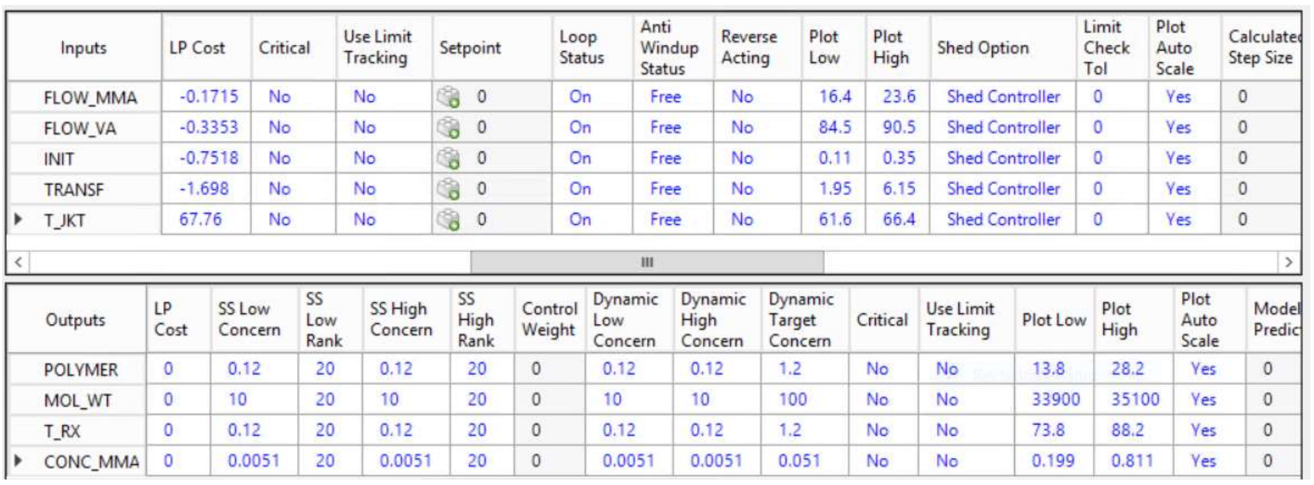
Figure 8.97 Initialize the controller simulation



Inputs	Combined Status	Service Request	Service Status	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Current Move	Operator High Limit	Engineering High Limit	Validity High Limit
FLOW_MMA	High Limit	On	On	18	0	16	17	18	0	23	24	50
FLOW_VA	High Limit	On	On	90	0	84	85	90	0	90	91	200
INIT	High Limit	On	On	0.16	0	0.1	0.13	0.16	0	0.33	0.36	5
TRANSF	High Limit	On	On	2.7	0	1.8	2.3	2.7	0	5.8	6.3	10
T_JKT	Low Limit	On	On	65	0	63	63	65	0	69	69	100

Outputs	Combined Status	Service Request	Service Status	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Operator High Limit	Engineering High Limit	Validity High Limit	Prediction
POLYMER	Normal	On	On	21	0	13	15	21	27	29	50	0
MOL_WT	Normal	On	On	35000	0	34500	34500	35000	35500	35500	50000	0
T_RX	Normal	On	On	85	0	73	75	79	95	95	500	0
CONC_MMA	Normal	On	On	0.5	0	0.17	0.25	0.5	0.76	0.84	1	0

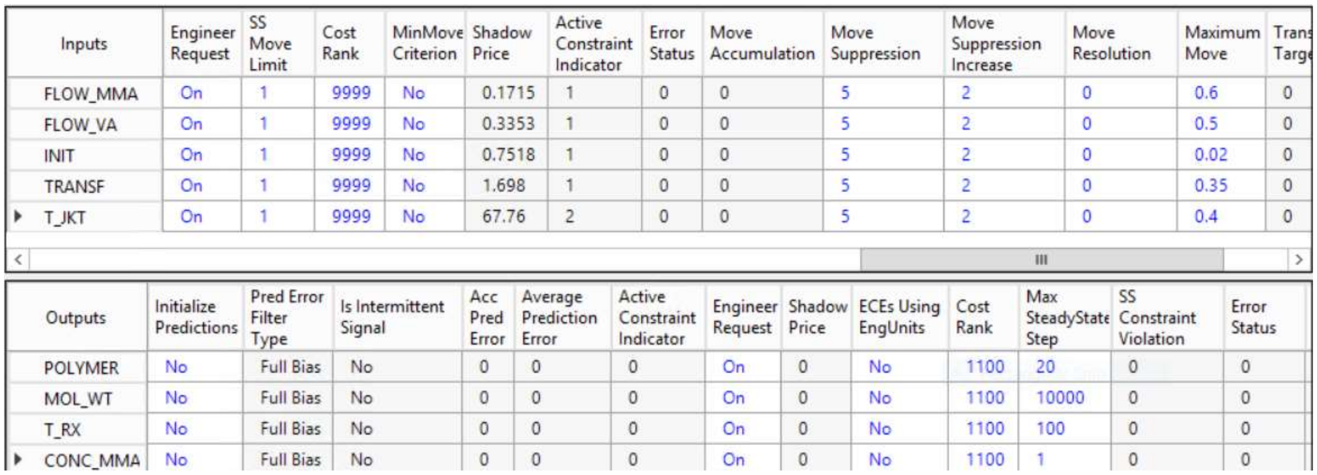
Figure 8.97a Input entries for controller simulation –part 1.



Inputs	LP Cost	Critical	Use Limit Tracking	Setpoint	Loop Status	Anti Windup Status	Reverse Acting	Plot Low	Plot High	Shed Option	Limit Check Tol	Plot Auto Scale	Calculate Step Size
FLOW_MMA	-0.1715	No	No	0	On	Free	No	16.4	23.6	Shed Controller	0	Yes	0
FLOW_VA	-0.3353	No	No	0	On	Free	No	84.5	90.5	Shed Controller	0	Yes	0
INIT	-0.7518	No	No	0	On	Free	No	0.11	0.35	Shed Controller	0	Yes	0
TRANSF	-1.698	No	No	0	On	Free	No	1.95	6.15	Shed Controller	0	Yes	0
T_JKT	67.76	No	No	0	On	Free	No	61.6	66.4	Shed Controller	0	Yes	0

Outputs	LP Cost	SS Low Concern	SS Low Rank	SS High Concern	SS High Rank	Control Weight	Dynamic Low Concern	Dynamic High Concern	Dynamic Target Concern	Critical	Use Limit Tracking	Plot Low	Plot High	Plot Auto Scale	Model Predic
POLYMER	0	0.12	20	0.12	20	0	0.12	0.12	1.2	No	No	13.8	28.2	Yes	0
MOL_WT	0	10	20	10	20	0	10	10	100	No	No	33900	35100	Yes	0
T_RX	0	0.12	20	0.12	20	0	0.12	0.12	1.2	No	No	73.8	88.2	Yes	0
CONC_MMA	0	0.0051	20	0.0051	20	0	0.0051	0.0051	0.051	No	No	0.199	0.811	Yes	0

Figure 8.97b Input entries for controller simulation –part 2.



Inputs	Engineer Request	SS Move Limit	Cost Rank	MinMove Criterion	Shadow Price	Active Constraint Indicator	Error Status	Move Accumulation	Move Suppression	Move Suppression Increase	Move Resolution	Maximum Move	Trans Targ
FLOW_MMA	On	1	9999	No	0.1715	1	0	0	5	2	0	0.6	0
FLOW_VA	On	1	9999	No	0.3353	1	0	0	5	2	0	0.5	0
INIT	On	1	9999	No	0.7518	1	0	0	5	2	0	0.02	0
TRANSF	On	1	9999	No	1.698	1	0	0	5	2	0	0.35	0
T_JKT	On	1	9999	No	67.76	2	0	0	5	2	0	0.4	0

Outputs	Initialize Predictions	Pred Error Filter Type	Is Intermittent Signal	Acc Pred Error	Average Prediction Error	Active Constraint Indicator	Engineer Request	Shadow Price	ECEs Using EngUnits	Cost Rank	Max SteadyState Step	SS Constraint Violation	Error Status
POLYMER	No	Full Bias	No	0	0	0	On	0	No	1100	20	0	0
MOL_WT	No	Full Bias	No	0	0	0	On	0	No	1100	10000	0	0
T_RX	No	Full Bias	No	0	0	0	On	0	No	1100	100	0	0
CONC_MMA	No	Full Bias	No	0	0	0	On	0	No	1100	1	0	0

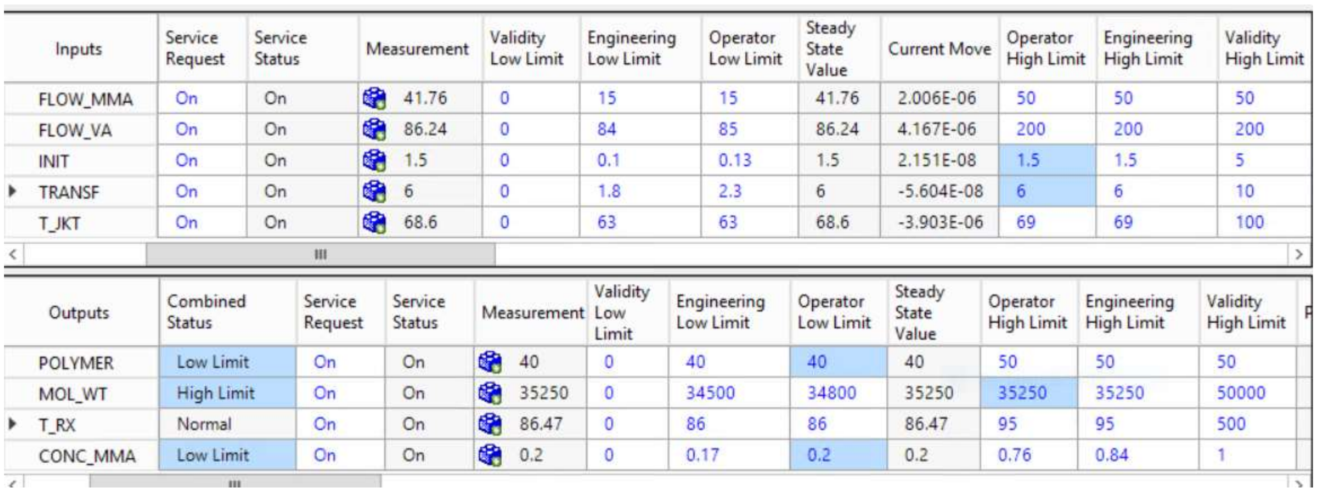
Figure 8.97c Input entries for controller simulation –part 3.

Before we make the controller simulation, we save this simulation file as ***WS8.1_BaseCase.dmc3application.*** Saving the input entries for the base case is essential, as it allows us to return to these initial specifications later if necessary. We note that when running the controller simulation forward in time, DMC Builder has no provision to let the controller rewind in time to its initial specifications.

With this controller model, we can proceed to fine-tune the controller to optimize the polymer production, improve product quality, and compensate for disturbances and setpoint changes, etc.

8.2.18 DMC3 Builder Task 4: Simulation –Dynamic Controller Applications to Polymer Production and Setpoint Changes

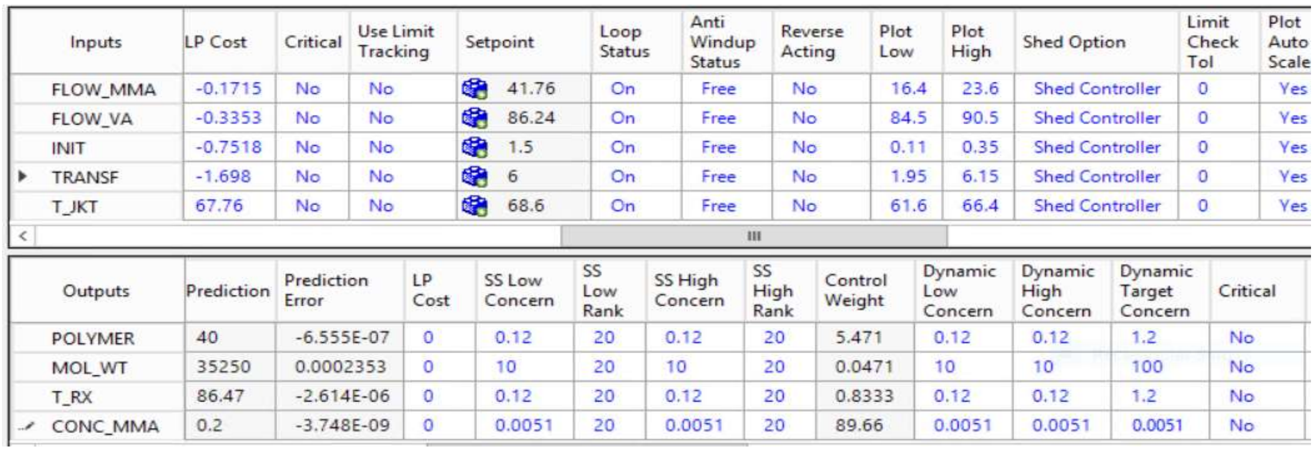
To increase the polymer production, we save the base case as a new file, ***WS8.1-1.dmc3application.*** There are many ways to increase the polymer production from its current value. We illustrate an approach in Figures 8.98a-b, which show the controller input entries to raise the polymer production to 40 kg/hr, while satisfying all the constraints. This involves setting the higher engineering and validity limits of initiator mass flow INIT to 1.5 kg/hr, the higher engineering and validity limits of chain transfer agent mass flow TRANSF to 6 kg/hr, and the lower engineering and validity limits of polymer mass flow POLYMER to 40 kg/hr.



Inputs	Service Request	Service Status	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Current Move	Operator High Limit	Engineering High Limit	Validity High Limit
FLOW_MMA	On	On	41.76	0	15	15	41.76	2.006E-06	50	50	50
FLOW_VA	On	On	86.24	0	84	85	86.24	4.167E-06	200	200	200
INIT	On	On	1.5	0	0.1	0.13	1.5	2.151E-08	1.5	1.5	5
TRANSF	On	On	6	0	1.8	2.3	6	-5.604E-08	6	6	10
T_JKT	On	On	68.6	0	63	63	68.6	-3.903E-06	69	69	100

Outputs	Combined Status	Service Request	Service Status	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Operator High Limit	Engineering High Limit	Validity High Limit
POLYMER	Low Limit	On	On	40	0	40	40	40	50	50	50
MOL_WT	High Limit	On	On	35250	0	34500	34800	35250	35250	35250	50000
T_RX	Normal	On	On	86.47	0	86	86	86.47	95	95	500
CONC_MMA	Low Limit	On	On	0.2	0	0.17	0.2	0.2	0.76	0.84	1

Figure 8.98a Input entries for controller simulation increasing polymer production to 40 kg/hr – part 1.



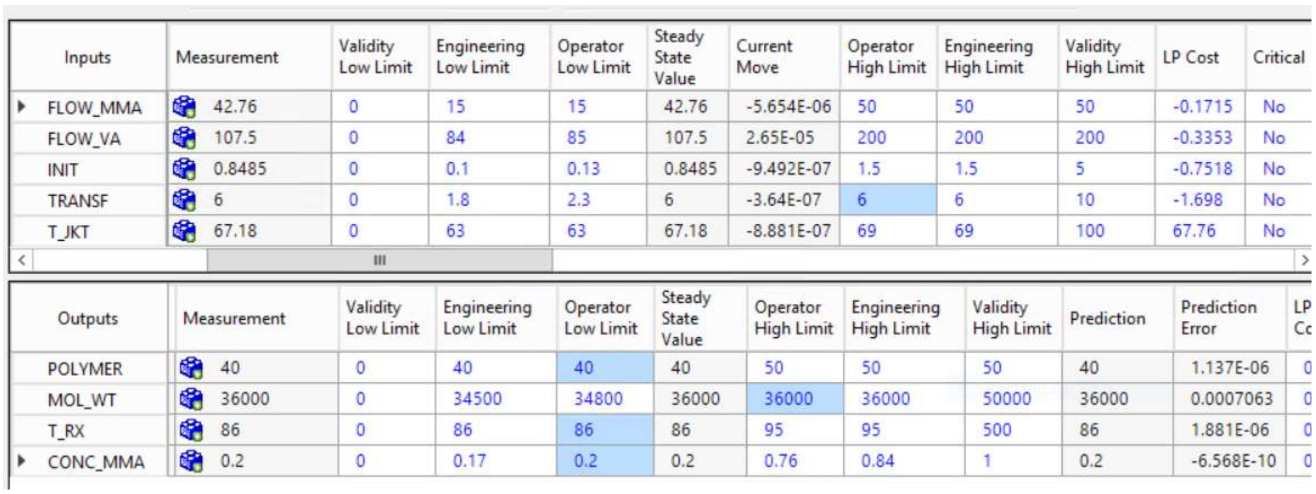
Inputs	LP Cost	Critical	Use Limit Tracking	Setpoint	Loop Status	Anti Windup Status	Reverse Acting	Plot Low	Plot High	Shed Option	Limit Check Tol	Plot Auto Scale
FLOW_MMA	-0.1715	No	No	41.76	On	Free	No	16.4	23.6	Shed Controller	0	Yes
FLOW_VA	-0.3353	No	No	86.24	On	Free	No	84.5	90.5	Shed Controller	0	Yes
INIT	-0.7518	No	No	1.5	On	Free	No	0.11	0.35	Shed Controller	0	Yes
TRANSF	-1.698	No	No	6	On	Free	No	1.95	6.15	Shed Controller	0	Yes
T_JKT	67.76	No	No	68.6	On	Free	No	61.6	66.4	Shed Controller	0	Yes

Outputs	Prediction	Prediction Error	LP Cost	SS Low Concern	SS Low Rank	SS High Concern	SS High Rank	Control Weight	Dynamic Low Concern	Dynamic High Concern	Dynamic Target Concern	Critical
POLYMER	40	-6.555E-07	0	0.12	20	0.12	20	5.471	0.12	0.12	1.2	No
MOL_WT	35250	0.0002353	0	10	20	10	20	0.0471	10	10	100	No
T_RX	86.47	-2.614E-06	0	0.12	20	0.12	20	0.8333	0.12	0.12	1.2	No
CONC_MMA	0.2	-3.748E-09	0	0.0051	20	0.0051	20	89.66	0.0051	0.0051	0.0051	No

Figure 8.98b Input entries for controller simulation increasing polymer production to 40 kg/hr – part 2.

We save the converged simulation file as ***WS8.1-2.dmc3application.***

Next, we wish to raise the polymer molecular weight to 36,000, while keeping the polymer production to 40 kg/hr, and controlling the concentration of MMA in the polymer product to 0.2. We can achieve this setpoint change by referring to the input entries of Figure 8.89a, and raising the operator and validity high limits of polymer molecular weight MOL_WT to 36,000, while keeping other input entries unchanged. Running the simulation quickly reaches the new polymer molecular weight target value of 36,000. See Figure 8.99. We save the resulting simulation file as **WS8.1-3.dmc3application.**



Inputs	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Current Move	Operator High Limit	Engineering High Limit	Validity High Limit	LP Cost	Critical
FLOW_MMA	42.76	0	15	15	42.76	-5.654E-06	50	50	50	-0.1715	No
FLOW_VA	107.5	0	84	85	107.5	2.65E-05	200	200	200	-0.3353	No
INIT	0.8485	0	0.1	0.13	0.8485	-9.492E-07	1.5	1.5	5	-0.7518	No
TRANSF	6	0	1.8	2.3	6	-3.64E-07	6	6	10	-1.698	No
T_JKT	67.18	0	63	63	67.18	-8.881E-07	69	69	100	67.76	No

Outputs	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Operator High Limit	Engineering High Limit	Validity High Limit	Prediction	Prediction Error	LP Co
POLYMER	40	0	40	40	40	50	50	50	40	1.137E-06	0
MOL_WT	36000	0	34500	34800	36000	36000	36000	50000	36000	0.0007063	0
T_RX	86	0	86	86	86	95	95	500	86	1.881E-06	0
CONC_MMA	0.2	0	0.17	0.2	0.2	0.76	0.84	1	0.2	-6.568E-10	0

Figure 8.99 Input entries for controller simulation increasing the polymer molecular weight MOL_WT to 36,000 while keeping the concentration of MMA in the product CONC_MMA at 0.2

This concludes the current "long" workshop of introducing DMC3 for a copolymerization problem. We covered the DMC3 tasks of (1) master model, (2) configuration, (3) optimization, and (4) simulation. Interested readers may refer to training courses offered by Aspen Technology, Inc. for an introduction to the additional tasks of (5) calculations (performing online calculations and variable transformations), and (6) deployment (performing controller deployment).

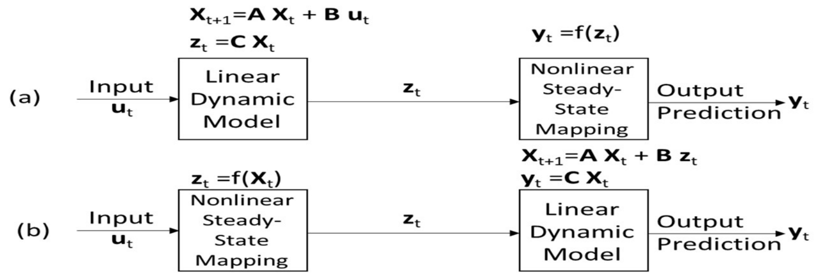
8.3 Model-Predictive Control of Nonlinear Polyolefin Processes

8.3.1 Challenges of Developing Nonlinear Predictive Modeling for Polyolefin Process Control

In Section 1.4.2, we review the observations by Turner and his colleagues [14,15] of the significant deficiencies in applying the conventional neural networks to model-predictive control of polymer processes, particularly with grade changes. Specifically, we mention that: (1) Conventional neural network architectures intrinsically contain regions where the partial derivative of a dependent variable (a process variable, PV) with respect to an independent variable (a manipulative variable, MV) becomes zero, and the resulting zero model gain would lead to an infinite controller gain; and (2) conventional neural network models cannot cope with the extrapolative demands of predictive control during polymer grade transitions. These two deficiencies are only two of the ten reasons that Turner and his colleagues [14,15] speak against applying conventional neural network models to model-predictive control of polymer processes. In a 2020 article, Bindlish [18] has demonstrated a controller output variable that has a steady-state gain inversion (changing signs from positive to negative, or from negative to positive) in a nonlinear model-predictive control of a DOW chemical process.

In analyzing what must be done for model-based control of polymer processes, Bausa [19] describes that the nonlinearities in a polymer process occur mainly during the grade change. A model that was identified for a special grade often does not predict the steady-state gains correctly when considering other polymer qualities such as melt index, which typically varies nonlinearly with process independent variables.

Bausa [19] says that it is a logical step to extend the linear model-predictive control algorithms gradually with nonlinear model characteristics. Figure 8.100 illustrates two approaches to do this. The Wiener approach multiplies the linear dynamic model output with a nonlinear steady-state function or mapping to yield the output prediction; the Hammerstein approach connects the output from a nonlinear steady-state function or mapping with a linear dynamic model to produce the output prediction. Jeong et. al [20] have demonstrated a nonlinear model-predictive controller using a Wiener model for an experimental continuous methyl methacrylate polymerization reactor. We note that in applying a Wiener model, the linear dynamic model is typically a multi-input and multi-output (MIMO) model, while the nonlinear steady-state function or mapping is typically a multi-input single-output (MISO) model. For example, the single output could be the melt index of a polyolefin product, while the multiple inputs could be the hydrogen mass flow rate, flow rate ratios of ethylene to hydrogen, and butane to hydrogen, etc.



(a)
Input $\mathbf{u}_t$
$\mathbf{X}_{t+1}=\mathbf{A}\,\mathbf{X}_t + \mathbf{B}\,\mathbf{u}_t$
$\mathbf{z}_t = \mathbf{C}\,\mathbf{X}_t$
Linear Dynamic Model
$\mathbf{z}_t$
$\mathbf{y}_t = f(\mathbf{z}_t)$
Nonlinear Steady-State Mapping
Output Prediction $\mathbf{y}_t$
(b)
Input $\mathbf{u}_t$
$\mathbf{z}_t = f(\mathbf{X}_t)$
Nonlinear Steady-State Mapping
$\mathbf{z}_t$
$\mathbf{X}_{t+1}=\mathbf{A}\,\mathbf{X}_t + \mathbf{B}\,\mathbf{z}_t$
$\mathbf{y}_t = \mathbf{C}\,\mathbf{X}_t$
Linear Dynamic Model
Output Prediction $\mathbf{y}_t$

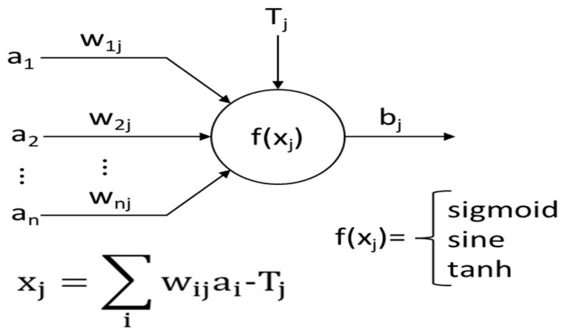
Figure 8.100 (a) The Wiener model; (b) The Hammerstein model.

8.3.2 Nonlinear Steady-State Mapping by State-Space Bounded Derivative Network (SS-BDN)

8.3.2a Possible Gain Inversion and Non-Monotonic Behavior of Conventional Neural Networks

We refer the reader to reference [17] and many online tutorials about conventional neural networks, and will not repeat those readily available, basic materials in this text. We briefly review the relevant features of a conventional neural network that is essential to demonstrating its deficiencies for polymer process control applications.

Figure 8.101 illustrates the foundation of a neural network, the neuron, or node (sometimes called a processing element). We represent the inputs to the j-th node as an input vector, **a**, with components a_i (i = 1 to n). The node manipulates these inputs, or activities, to give the output, b_j, which can then form a part of the input to other nodes. In the figure, we see that the j-th node transfers the i-th input a_i to the j-th output b_j through a weight factor w_{ij} and a transfer function $f(x_j)$. T_j is the internal threshold for node j.



a_1
w_{1j}
a_2
w_{2j}
a_n
w_{nj}
T_j
$f(x_j)$
b_j
$x_j = \sum_i w_{ij} a_i - T_j$
$f(x_j) = \begin{cases} \text{sigmoid} \\ \text{sine} \\ \text{tanh} \end{cases}$

Figure 8.101 The processing element (neuron or node) of a neural network.

In polymer process control using neural network models, input components a_i could represent the independent variables such as hydrogen mass flow rate, flow rate ratios of ethylene to hydrogen, and of butane to hydrogen, etc.; while the output b_j could be a dependent variable, such as the polymer melt index. Depending on the type of transfer function $f(x_j)$ being used, we may find that the partial derivative of an output or a dependent variable b_j with respect to an input component or an independent variable a_i may change sign from positive to negative, or from negative to positive. According to Eqs. (8.3) and (8.4), these partial derivatives represent elements of the steady-state gain matrix. We call this sign change as *a steady-state gain inversion*.

Consider, for example, a popular transfer function, $f(x_j) = \tanh(x_j)$, the hyperbolic tangent function. We review some basic calculus for the hyperbolic tangent function here.

Define:

$$\cosh x = \frac{e^x + e^{-x}}{2} \qquad \sinh x = \frac{e^x - e^{-x}}{2}$$

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad \operatorname{sech} x = \frac{1}{\cosh x} = \frac{2}{e^x + e^{-x}} \qquad \operatorname{csch} x = \frac{1}{\sinh x} = \frac{2}{e^x - e^{-x}}$$

Let u = f(x), we write the derivatives as follows:

$$\frac{d}{dx}(\sinh u) = \cosh u \frac{du}{dx} \qquad \frac{d}{dx}(\cosh u) = \sinh u \frac{du}{dx} \qquad \frac{d}{dx}(\tanh u) = (\operatorname{sech} u)^2 \frac{du}{dx} \quad (8.49)$$

Figure 8.102 illustrates the hyperbolic tangent function and its derivative. While the hyperbolic tangent function monotonically increases with increasing x_j, its derivative value changes from monotonically positive when x_j is negative to monotonically negative when x_j is positive. Therefore, using the hyperbolic tangent transfer function could lead to a change in the sign of the partial derivative of dependent variable b_j with respect to independent variable x_j, resulting in a gain inversion.

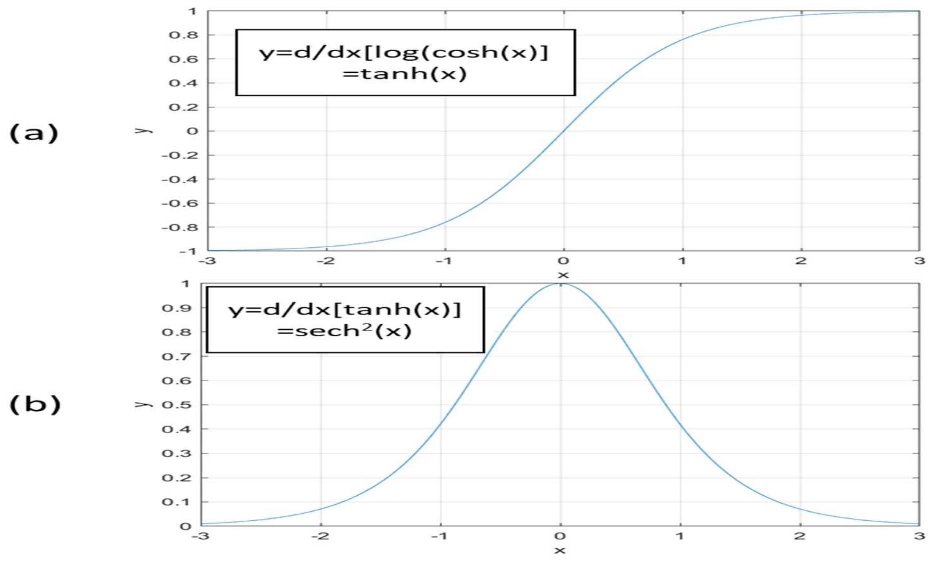


Figure 8.102 The hyperbolic tangent transfer function and its derivative.

What type of transfer function do we need to avoid possible gain inversion? We want to choose a transfer function whose derivative varies monotonically. Consider, for example, the analytical integral of a standard hyperbolic tangent transfer function as our new transfer function [14,15]:

$$\int \tanh(ax)\,dx = \frac{1}{a}\ln[\cosh(ax)] + c \Longrightarrow \begin{matrix}(Nonlinear\ transer\ function)\\ \ln(\cosh u)\end{matrix} \qquad (8.50)$$

$$\frac{\mathrm{d}}{\mathrm{dx}}[\ln(\cosh \mathrm{u})] = \frac{1}{\cosh \mathrm{u}}\sinh \mathrm{u}\frac{\mathrm{du}}{\mathrm{dx}} = \tanh \mathrm{u}\frac{\mathrm{du}}{\mathrm{dx}} \qquad (8.51)$$

Figure 8.103 illustrates the transfer function log(cosh u), Eq. (8.50) and its derivative, Eq. (8.51). While the function itself is always positive in value, its derivative monotonically increases with increasing independent variable $x_{j.}$ Therefore, there is no concern for possible gain inversion.

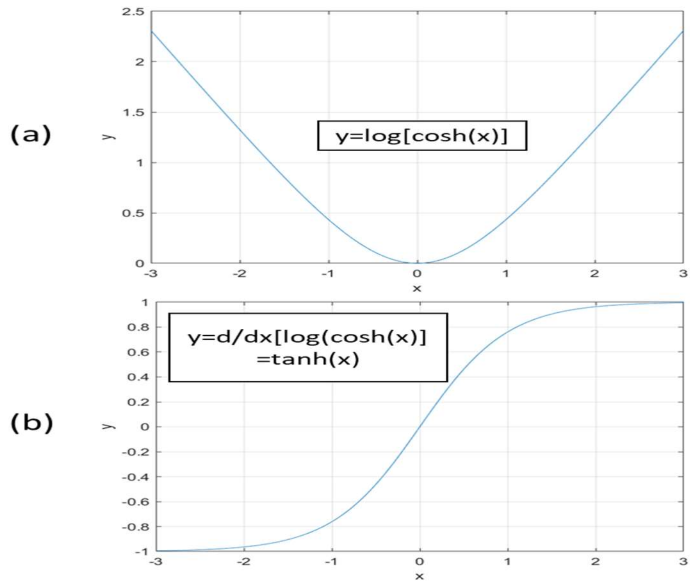


Figure 8.103 The transfer function log(coshu) and its derivative

8.3.2b State-Space Bounded Derivative Network (SS-BDN)

As discussed in [15,16], the SS-BDN is essentially the analytical integral of a neural network. Based on Eq. (8.50) - (8.51), we illustrate the general model architecture of a SS-BDN based on the analytical integral of a neural network based on a hyperbolic tangent transfer function. Figure 8.104 shows the model architecture.

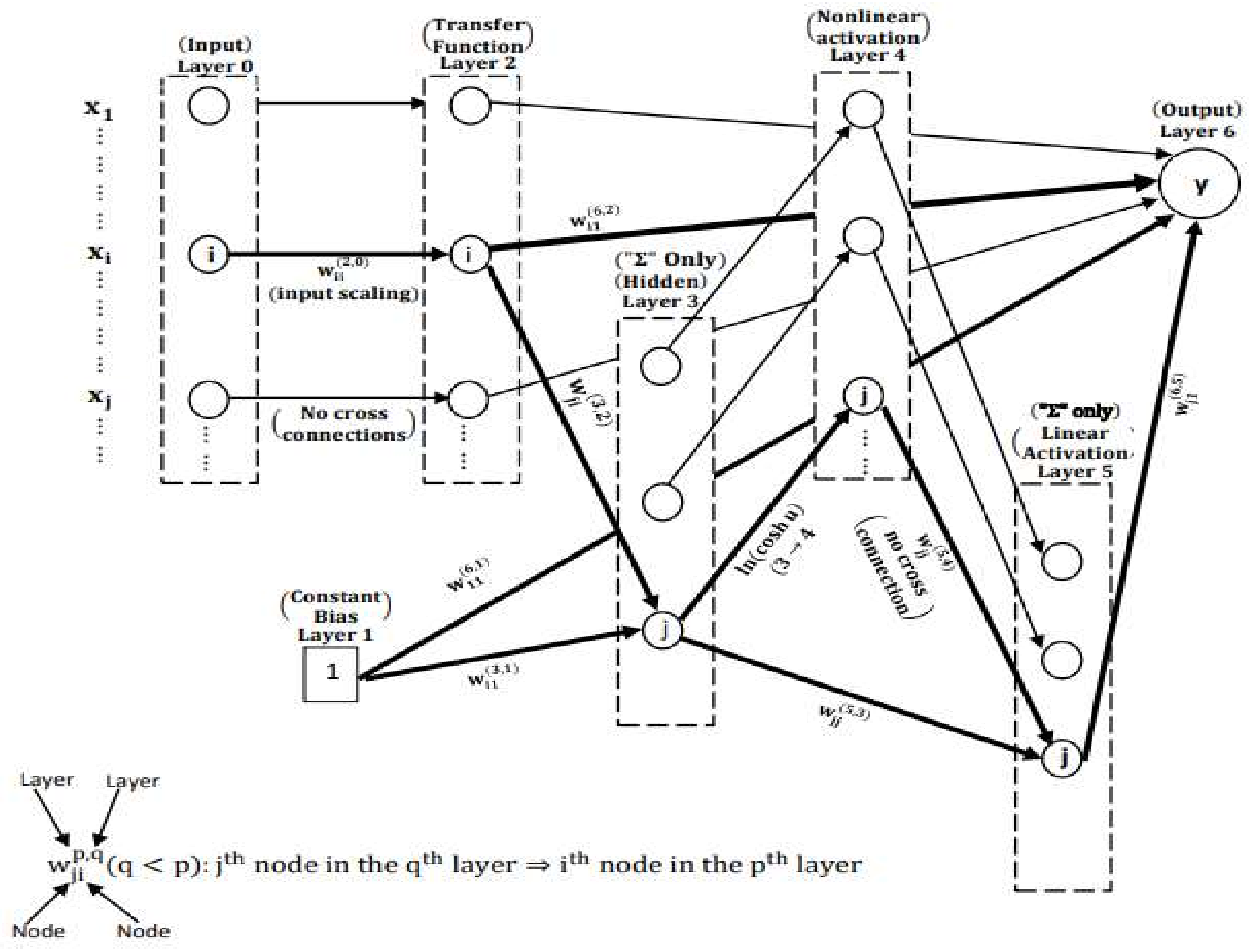

(Input) Layer 0
(Transfer Function) Layer 2
(Nonlinear activation) Layer 4
(Output) Layer 6
("Σ" Only) (Hidden) Layer 3
("Σ" only) (Linear Activation) Layer 5
(Constant Bias) Layer 1
x1
xi
xj
y
(input scaling)
No cross connections
ln(cosh u) (3 → 4)
no cross connection
Layer
Layer
Node
Node
$w_{ji}^{p,q}(q < p)$: jth node in the qth layer ⇒ ith node in the pth layer

Figure 8.104 Architecture of the state-space bounded derivative network

We wish to use this architecture to demonstrate that the partial derivative of the dependent variable y with respect to independent variables x_k is always bounded (hence the name of bounded derivative network). Based on Figure 8.104, we write:

$$y = \underset{(1\to 6)}{w_{11}^{(6,1)}} + \sum_i \underset{(0\to 2\to 6)}{w_{i1}^{(6,2)} . w_{ii}^{(2,0)}} . x_i$$

$$+\sum_j \underset{(5\to 6)}{w_{j1}^{(6,5)}} \left\{ \underset{(4\to 5)}{w_{jj}^{(5,4)}} \left[\underbrace{\log\Big(\cosh}_{(3\to 4)} \Big(\underset{(1\to 3)}{w_{j1}^{(3,1)}} + \sum_i \underset{(2\to 3)}{w_{ji}^{(3,2)}} \Big(\underset{(0\to 2)}{w_{ii}^{(2,0)}} x_i \Big)\Big)\Big) \right] \right\}$$

$$+\sum_j \underset{(5\to 6)}{w_{j1}^{(6,5)}} \left\{ \underset{(3\to 5)}{w_{jj}^{(5,3)}} \left[\underset{(1\to 3)}{w_{j1}^{3,1)}} + \sum_i \underset{(2\to 3)}{w_{ji}^{(3,2)}} \Big(\underset{(0\to 2)}{w_{ii}^{(2,0)}} x_i \Big) \right] \right\}$$

Applying $\frac{d}{dx}$ln[cosh(ax)] = atanh(ax) and setting i = k, we can write:

$$\frac{\partial y}{\partial x_k} = \underset{(0\to 2\to 6)}{w_{k1}^{(6,2)} . w_{kk}^{(2,0)}}$$

$$+ \underset{(0\to 2)}{w_{kk}^{(2,0)}} \sum_j \underset{(5\to 6)}{w_{j1}^{(6,5)}} . \underset{(4\to 5)}{w_{jj}^{(5,4)}} . \underset{(2\to 3)}{w_{jk}^{(3,2)}} \underbrace{\underset{(3\to 4)}{\tanh}\left(\underset{(1\to 3)}{w_{j1}^{(3,1)}} + \sum_i \underset{(2\to 3)}{w_{ji}^{(3,2)}} \underset{(0\to 2)}{w_{ii}^{(2,0)}} x_i \right)}_{-1 \le \tanh u \le 1}$$

$$+ \underset{(0\to 2)}{w_{kk}^{(2,0)}} \sum_j \underset{(5\to 6)}{w_{j1}^{(6,5)}} . \underset{(3\to 5)}{w_{jj}^{(5,3)}} . \underset{(2\to 3)}{w_{jk}^{(3,2)}}$$

$$\frac{\partial y}{\partial x_k} = \underset{(0\to 2)}{w_{kk}^{(2,0)}} \left[\sum_j \underset{(5\to 6)}{w_{j1}^{(6,5)}} . \underset{(3\to 5)}{w_{jj}^{(5,3)}} . \underset{(2\to 3)}{w_{jk}^{(3,2)}} \right.$$

$$\left. + \sum_j \underset{(5\to 6)}{w_{j1}^{(6,5)}} . \underset{(4\to 5)}{w_{jj}^{(5,4)}} . \underset{(2\to 3)}{w_{jk}^{(3,2)}} . \underset{(3\to 4)}{\tanh}\left(\underset{(1\to 3)}{w_{j1}^{(3,1)}} + \sum_i \underset{(2\to 3)}{w_{ji}^{(3,2)}} \underset{(0\to 2)}{w_{ii}^{(2,0)}} x_i \right) + \underset{(2\to 6)}{w_{k1}^{(6,2)}} \right]$$

$$= \begin{cases} w_{kk}^{(2,0)} \left[\sum_j w_{j1}^{(6,5)} . w_{jj}^{(5,3)} . w_{jk}^{(3,2)} - \sum_j \left| w_{j1}^{(6,5)} . w_{jj}^{(5,4)} . w_{jk}^{(3,2)} \right| + w_{k1}^{(6,2)} \right] \left(\begin{matrix} \text{lower bound} \\ \text{when } \tanh u = -1 \end{matrix} \right) & (8.51) \\ w_{kk}^{(2,0)} \left[\sum_j w_{j1}^{(6,5)} . w_{jj}^{(5,3)} . w_{jk}^{(3,2)} + \sum_j \left| w_{j1}^{(6,5)} . w_{jj}^{(5,4)} . w_{jk}^{(3,2)} \right| + w_{k1}^{(6,2)} \right] \left(\begin{matrix} \text{upper bound} \\ \text{when } \tanh u = 1 \end{matrix} \right) & (8.52) \end{cases}$$

Eqs. (8.51) and (8.52) show the key features of the SS-BDN for nonlinear steady-state mapping that ensures the partial derivative of the dependent variable with respect to independent variables remain bounded. Additionally, as demonstrated in Figure 8.103, the choice of the transfer function within the network makes the values of the partial derivative monotonically increasing with increasing values of the independent variable. Both features are essential to the success of applying the Wiener model, Figure 8.100a, to polymer process control [15,16].

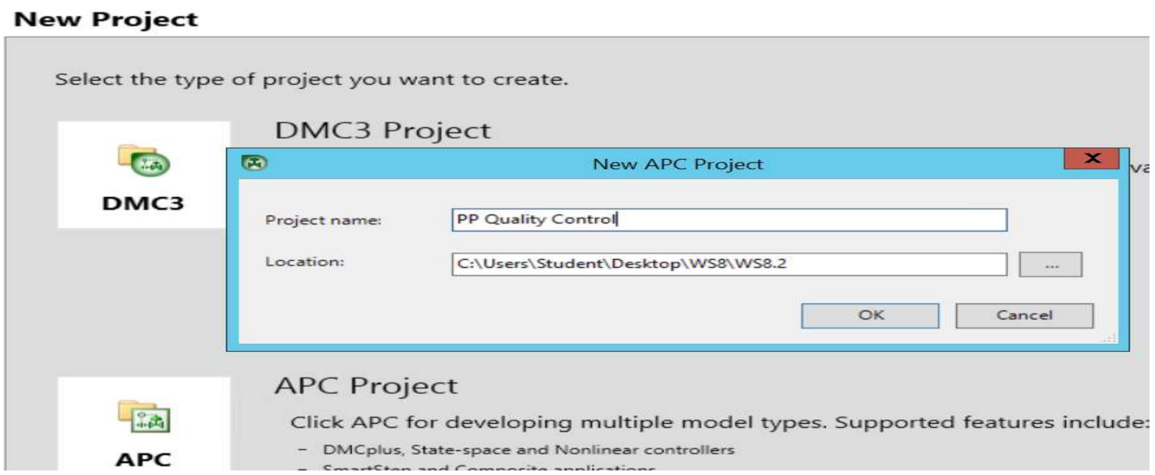
8.4 WS8.2 Development of a Nonlinear Predictive Controller Model for a Polypropylene Process

8.4.1 Objective

The objective of this workshop is to demonstrate how to use the DMC3 Builder to develop a nonlinear model-predictive controller for a polypropylene process based on the Wiener model of Figure 100a. This model consists of a linear state-space dynamic model for the process dynamics integrated with a nonlinear state-space bounded derivative network (SS-BDN) for polymer quality control. The goal of the controller is to control the polymer melt index and density. We also simulate the controller performing a transition from a melt index of 1 to a melt index of 10 at a constant density of 920 kg/m^3.

8.4.2 Starting an APC Project and Choosing Nonlinear Controllers, and Data Preprocessing

Figure 8.105 shows the selection of APC Project and DMCplus, State-Space and Nonlinear Controllers. We save the project as “PP Quality Control”. From the “Import” tool ribbon on the top left, we choose “Dataset”, and select the text file, WS8.2.txt within our working folder. We then click on the Open button. See Figures 8.106 and 8.107.


New Project

Select the type of project you want to create.

DMC3 Project

New APC Project

Project name: PP Quality Control

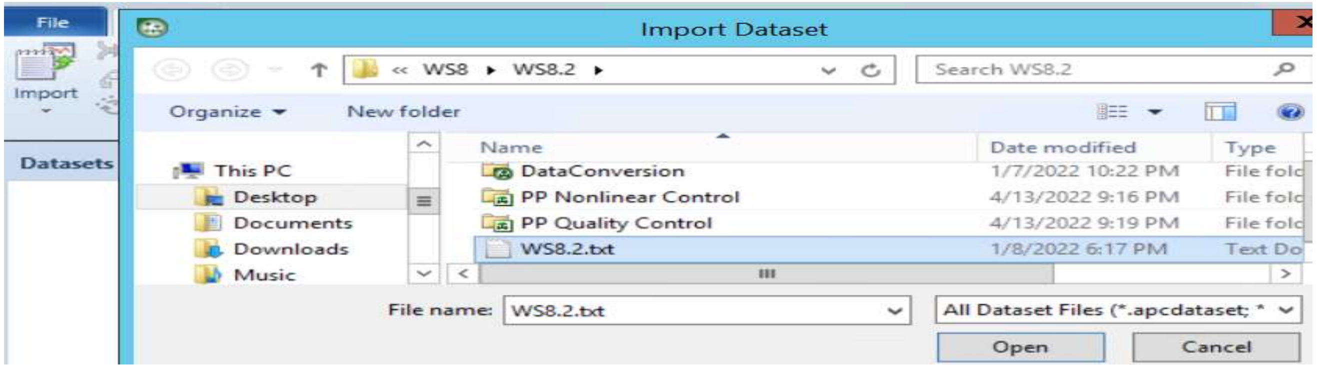
Location: C:\Users\Student\Desktop\WS8\WS8.2

APC Project

Click APC for developing multiple model types. Supported features include:

- DMCplus, State-space and Nonlinear controllers

Figure 8.105 Selection of APC Project and DMCplus, State-Space and Nonlinear Controllers.

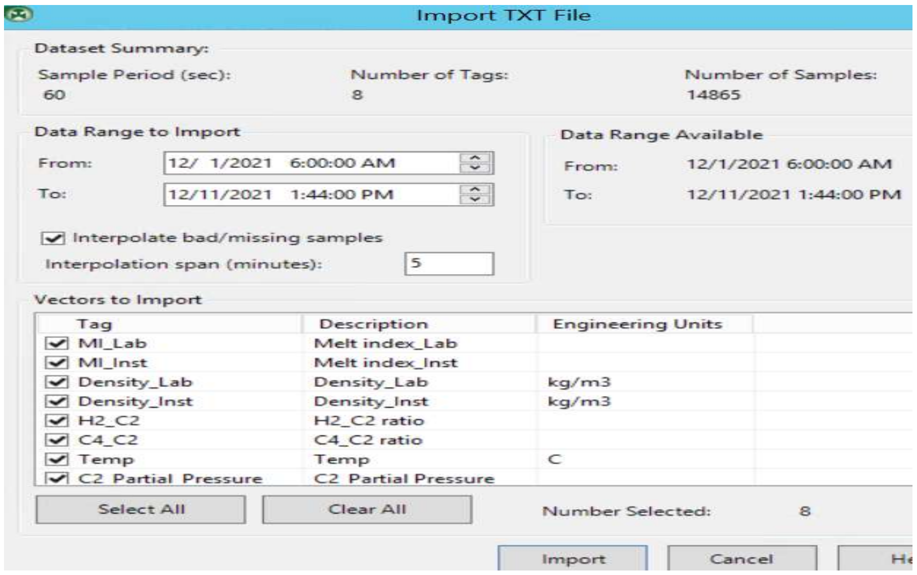

Import Dataset

Name	Date modified	Type
DataConversion	1/7/2022 10:22 PM	File fold
PP Nonlinear Control	4/13/2022 9:16 PM	File fold
PP Quality Control	4/13/2022 9:19 PM	File fold
WS8.2.txt	1/8/2022 6:17 PM	Text Do

File name: WS8.2.txt

All Dataset Files (*.apcdataset; *

Figure 8.106 Import dataset, ***WS8.2.txt***


Import TXT File

Dataset Summary:

Sample Period (sec): 60

Number of Tags: 8

Number of Samples: 14865

Data Range to Import

From: 12/ 1/2021 6:00:00 AM

To: 12/11/2021 1:44:00 PM

Data Range Available

From: 12/1/2021 6:00:00 AM

To: 12/11/2021 1:44:00 PM

[✓] Interpolate bad/missing samples

Interpolation span (minutes): 5

Vectors to Import

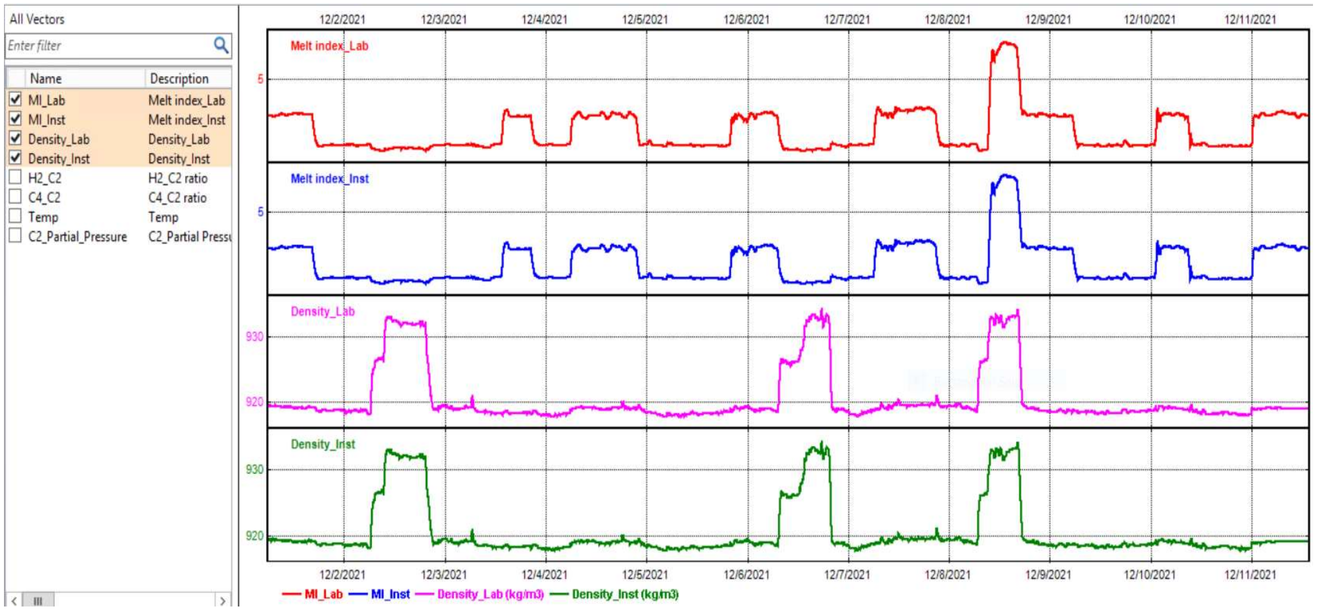
Tag	Description	Engineering Units
✓ MI_Lab	Melt index_Lab	
✓ MI_Inst	Melt index_Inst	
✓ Density_Lab	Density_Lab	kg/m3
✓ Density_Inst	Density_Inst	kg/m3
✓ H2_C2	H2_C2 ratio	
✓ C4_C2	C4_C2 ratio	
✓ Temp	Temp	C
✓ C2 Partial Pressure	C2 Partial Pressure	

Number Selected: 8

Figure 8.107 Contents of imported dataset

In Figure 8.107, we see the following variables: (1) CVs – MI_Lab, MI_Inst, Density_Lab and Density_Inst; (2) MVs – H2_C2 and C4_C2; and (3) DVs: Temp and C2_Partial_Pressure. We click on Import displayed in Figure 8.107, and see an "Interpolate Dataset" window. We click on "Start" button to interpolate any bad and missing data slices longer than 5 minutes in duration. We then clock the "Close" button to complete the interpolation analysis. We see the message that "0 of 8 vectors (variables) have been interpolated". We do not show the screen images of these simple steps.

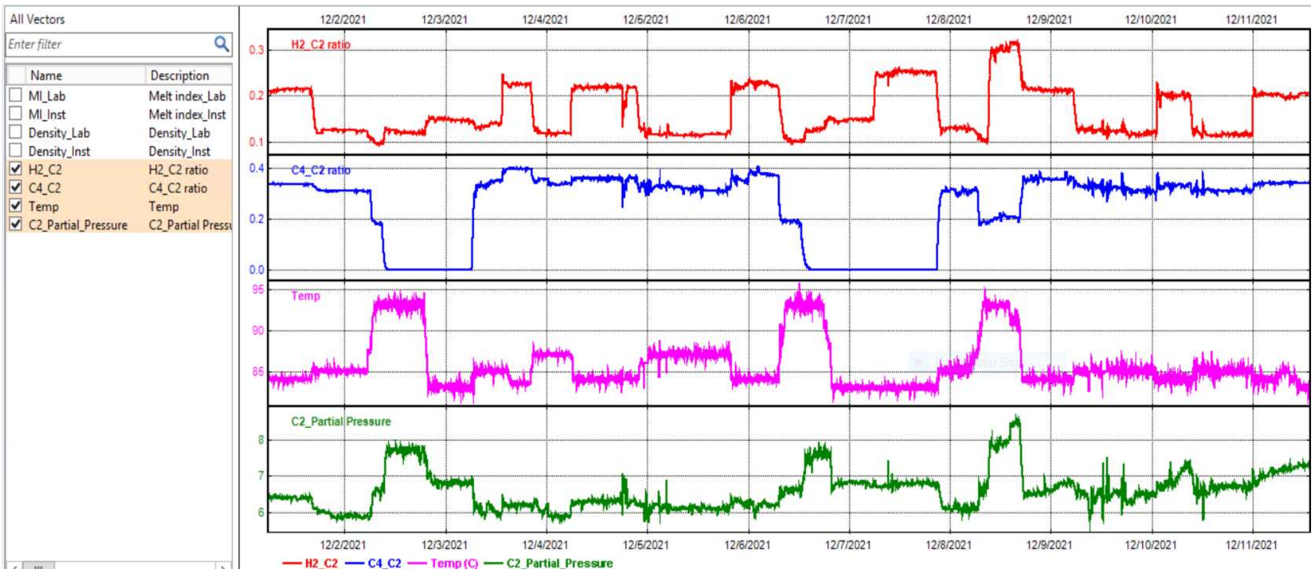
Following the interpolation step, we see the vector (variable) summary and the corresponding trend plot. The software automatically shows the first three CVs (MI_Lab, MI_Int, and Density_Lab), and we choose the fourth CV (Density_Inst). See Figure 8.108. We also display the MVs (H2_C2 and C4_C2) and DVs (Temp and C2_Partial_Pressure). See Figure 8.109.


All Vectors

Enter filter

	Name	Description
✓	MI_Lab	Melt index_Lab
✓	MI_Inst	Melt index_Inst
✓	Density_Lab	Density_Lab
✓	Density_Inst	Density_Inst
	H2_C2	H2_C2 ratio
	C4_C2	C4_C2 ratio
	Temp	Temp
	C2_Partial_Pressure	C2_Partial Pressu

Figure 8.108 A display of the CVs


All Vectors

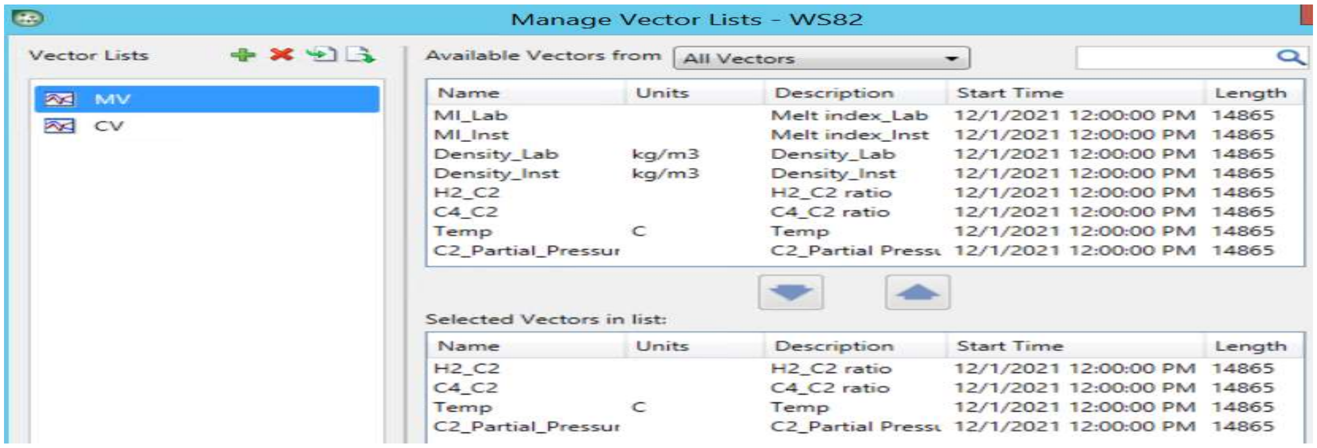
Enter filter

	Name	Description
	MI_Lab	Melt index_Lab
	MI_Inst	Melt index_Inst
	Density_Lab	Density_Lab
	Density_Inst	Density_Inst
✓	H2_C2	H2_C2 ratio
✓	C4_C2	C4_C2 ratio
✓	Temp	Temp
✓	C2_Partial_Pressure	C2_Partial Pressu

Figure 8.109 A display of the MVs and DVs

Reviewing the trend plots of Figures 8.108 and 8.109, we see no need to do data slicing, as there is no bad data slice.

Next, we click on "Manage Lists" button on the top tool ribbon and follow Figures 8.46 and 8.47 to create the MV and CV lists. See Figures 8.110 and 8.111.

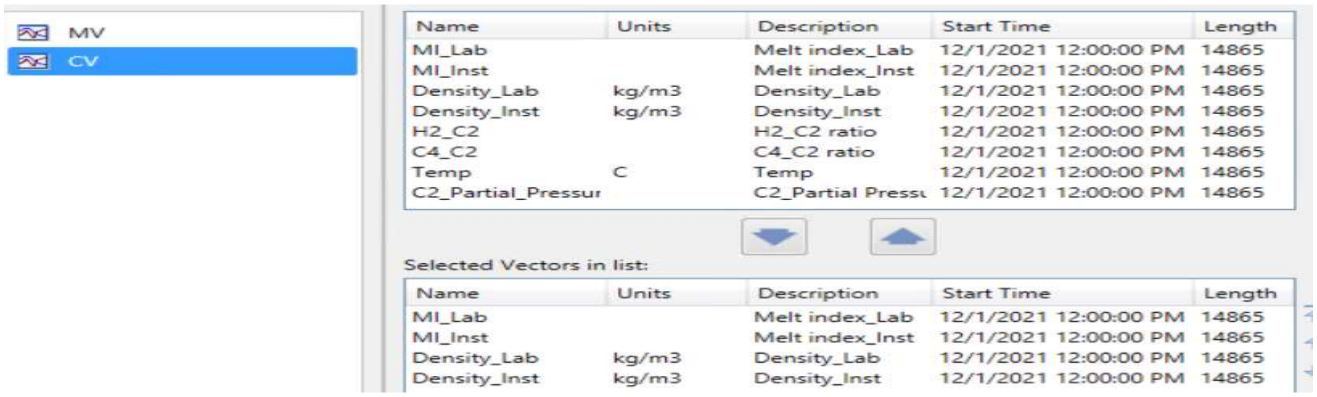


Name	Units	Description	Start Time	Length
MI_Lab		Melt index_Lab	12/1/2021 12:00:00 PM	14865
MI_Inst		Melt index_Inst	12/1/2021 12:00:00 PM	14865
Density_Lab	kg/m3	Density_Lab	12/1/2021 12:00:00 PM	14865
Density_Inst	kg/m3	Density_Inst	12/1/2021 12:00:00 PM	14865
H2_C2		H2_C2 ratio	12/1/2021 12:00:00 PM	14865
C4_C2		C4_C2 ratio	12/1/2021 12:00:00 PM	14865
Temp	C	Temp	12/1/2021 12:00:00 PM	14865
C2_Partial_Pressur		C2_Partial_Pressu	12/1/2021 12:00:00 PM	14865

Selected Vectors in list:

Name	Units	Description	Start Time	Length
H2_C2		H2_C2 ratio	12/1/2021 12:00:00 PM	14865
C4_C2		C4_C2 ratio	12/1/2021 12:00:00 PM	14865
Temp	C	Temp	12/1/2021 12:00:00 PM	14865
C2_Partial_Pressur		C2_Partial_Pressu	12/1/2021 12:00:00 PM	14865

Figure 8.110 Manipulated variables, MVs



Name	Units	Description	Start Time	Length
MI_Lab		Melt index_Lab	12/1/2021 12:00:00 PM	14865
MI_Inst		Melt index_Inst	12/1/2021 12:00:00 PM	14865
Density_Lab	kg/m3	Density_Lab	12/1/2021 12:00:00 PM	14865
Density_Inst	kg/m3	Density_Inst	12/1/2021 12:00:00 PM	14865
H2_C2		H2_C2 ratio	12/1/2021 12:00:00 PM	14865
C4_C2		C4_C2 ratio	12/1/2021 12:00:00 PM	14865
Temp	C	Temp	12/1/2021 12:00:00 PM	14865
C2_Partial_Pressur		C2_Partial_Pressu	12/1/2021 12:00:00 PM	14865

Selected Vectors in list:

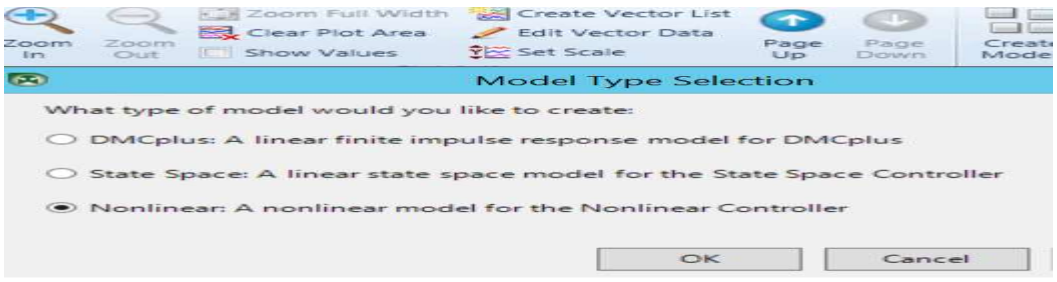
Name	Units	Description	Start Time	Length
MI_Lab		Melt index_Lab	12/1/2021 12:00:00 PM	14865
MI_Inst		Melt index_Inst	12/1/2021 12:00:00 PM	14865
Density_Lab	kg/m3	Density_Lab	12/1/2021 12:00:00 PM	14865
Density_Inst	kg/m3	Density_Inst	12/1/2021 12:00:00 PM	14865

Figure 8.111 Controlled variables, CVs

8.4.3 Aspen Nonlinear Controller: Task 1 - Model Identification

8.4.3a Step-Response Plot

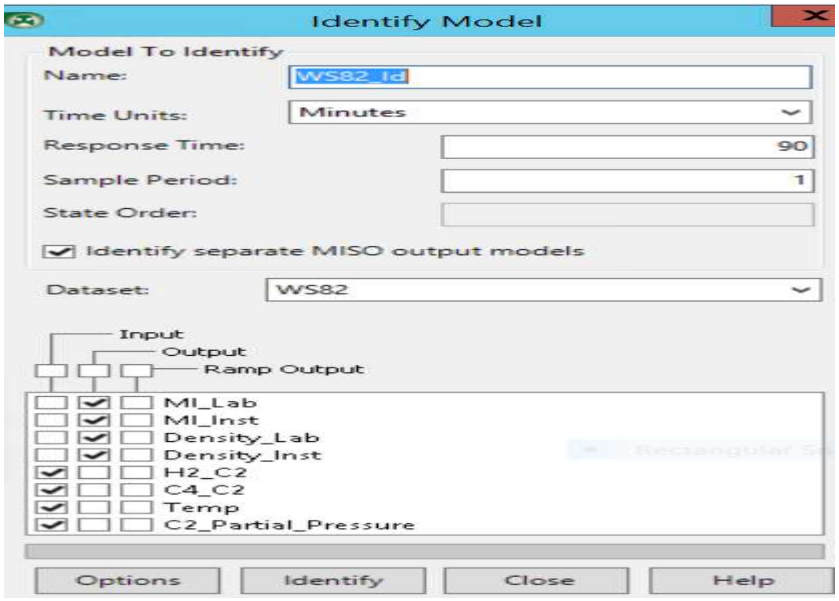
At the far right of the tool ribbons, we click on "Create Model" button. In "Model Type Selection", we choose "Nonlinear". See Figure 8.112. Clicking on OK gives the "Identify Model" inputs. See Figure 8.113.



What type of model would you like to create:

- ○ DMCplus: A linear finite impulse response model for DMCplus
- ○ State Space: A linear state space model for the State Space Controller
- ● Nonlinear: A nonlinear model for the Nonlinear Controller

Figure 8.112 Selection of model type


Identify Model

Model To Identify

Name: WS82_Id

Time Units: Minutes

Response Time: 90

Sample Period: 1

State Order:

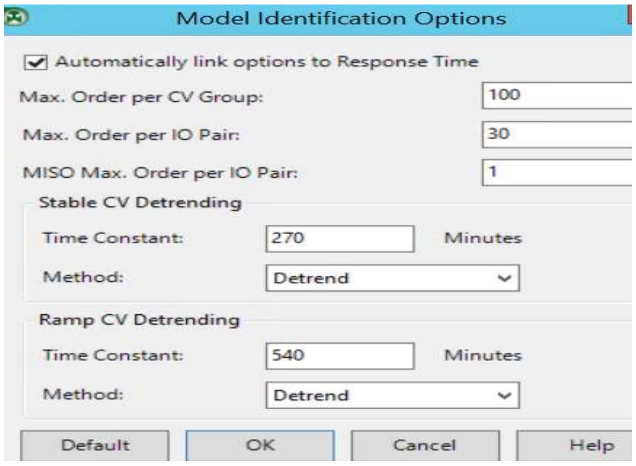
[✓] Identify separate MISO output models

Dataset: WS82

Input	Output	Ramp Output	
	✓		MI_Lab
	✓		MI_Inst
	✓		Density_Lab
	✓		Density_Inst
✓			H2_C2
✓			C4_C2
✓			Temp
✓			C2_Partial_Pressure

Figure 8.113 “identify Model” inputs

Clicking on “Options” displayed in Figure 8.113 gives the default specifications of Figure 8.114. We accept these specifications and click on “OK”. We then click on “Identify” shown in Figure 8.113. This results in the step-response plot of Figure 8.115, which represents the first type of plot under “model views” button on the top tool ribbon.


Model Identification Options

[✓] Automatically link options to Response Time

Max. Order per CV Group: 100

Max. Order per IO Pair: 30

MISO Max. Order per IO Pair: 1

Stable CV Detrending

Time Constant: 270 Minutes

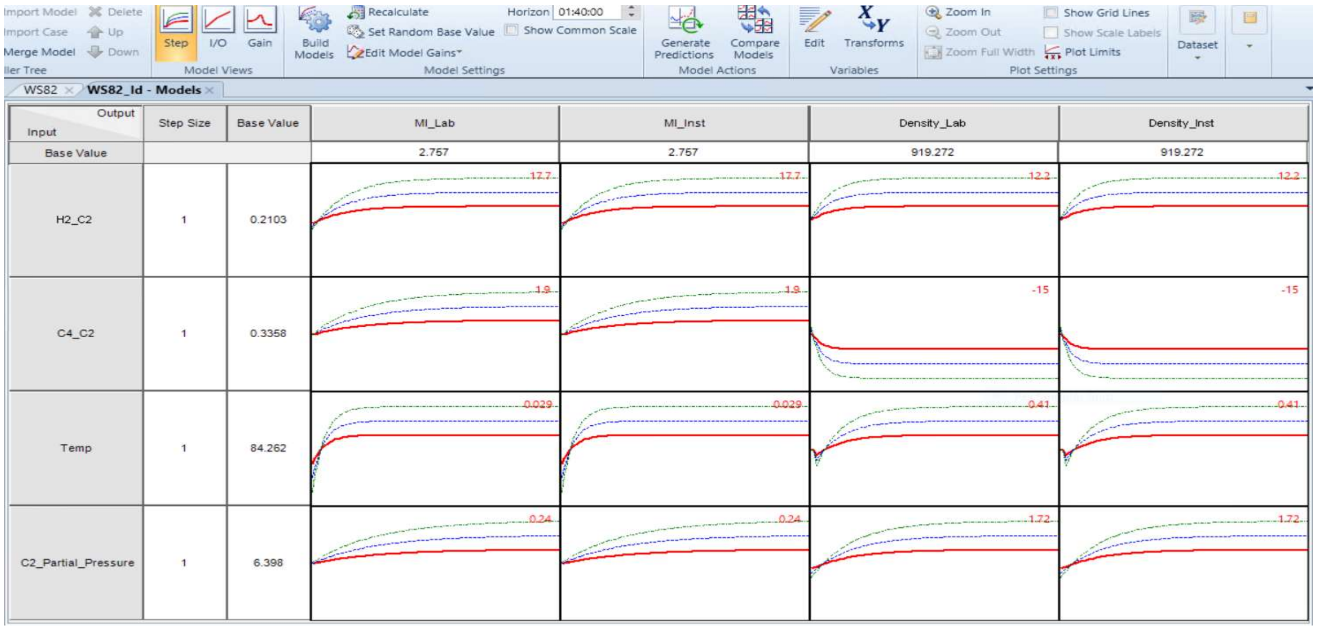
Method: Detrend

Ramp CV Detrending

Time Constant: 540 Minutes

Method: Detrend

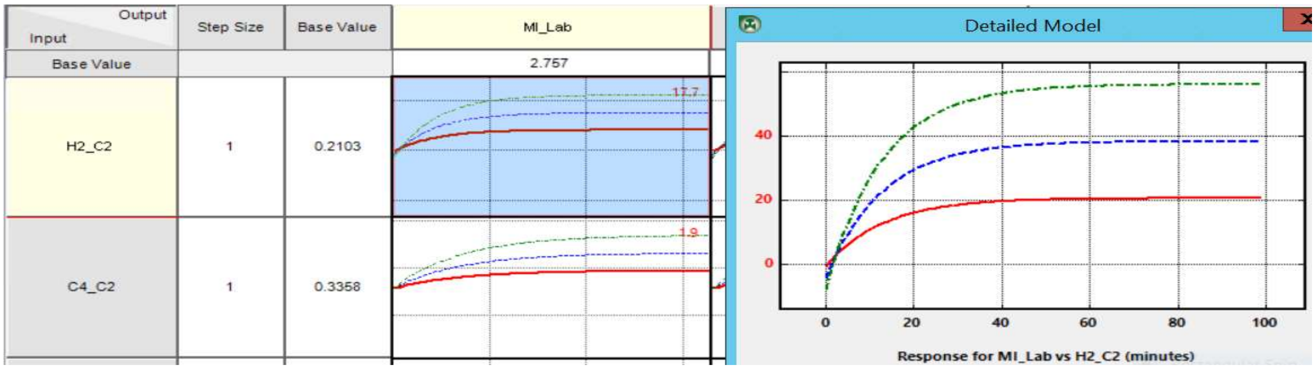
Figure 8.114 Model identification options



Output / Input	Step Size	Base Value	MI_Lab	MI_Inst	Density_Lab	Density_Inst
Base Value			2.757	2.757	919.272	919.272
H2_C2	1	0.2103	17.7	17.7	12.2	12.2
C4_C2	1	0.3358	1.9	1.9	-15	-15
Temp	1	84.262	0.029	0.029	0.41	0.41
C2_Partial_Pressure	1	6.398	0.24	0.24	1.72	1.72

Figure 8.115 Step response curves of a nonlinear polypropylene process

The step-response curve of a nonlinear polymer process is quite different from that of a linear process (e.g., Figure 8.53). For example, when H2_C2, a MV, increases, the affected MI_Lab, a CV, also increases, but it displays three response curves of increasing values from colors in red to blue and then to green. By contrast, when MV C4_C2 increases, the affected CVs, Density_Lab and Density_Inst, show three response curves of decreasing values from colors in red to blue and then to green. These responses depend on the operating point or values of the MVs, the direction of the change and the step size of the change in MV. In particular, the three response curves in the figure represent the time-dependent change of a chosen CV to a change of a chosen MV with its magnititue equal to the default step size of 1 (red curve), two times the step size (blue curve) and three times the step size (green curve). Additionally, putting the mouse inside the response curve box for a selected CV-MV pair, right-click to open the menu and select "details", we see a detailed plot of the MI_Lab-H2_C2 response curve in Figure 8.116.

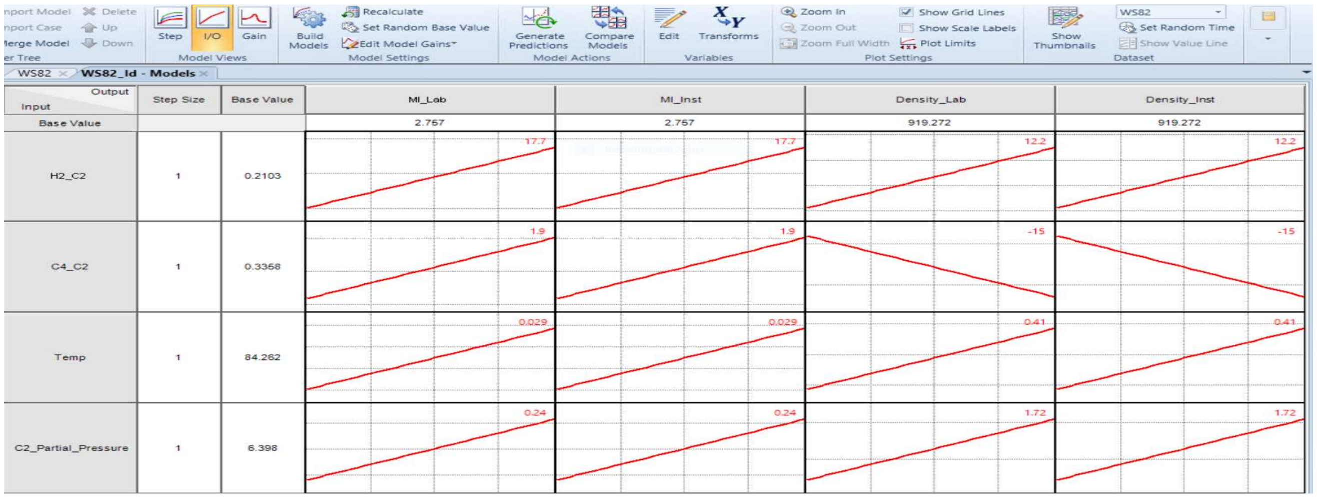


Output / Input	Step Size	Base Value	MI_Lab
Base Value			2.757
H2_C2	1	0.2103	17.7
C4_C2	1	0.3358	1.9

Figure 8.116 Showing the details of the MI_Lab-H2_C2 step response curve

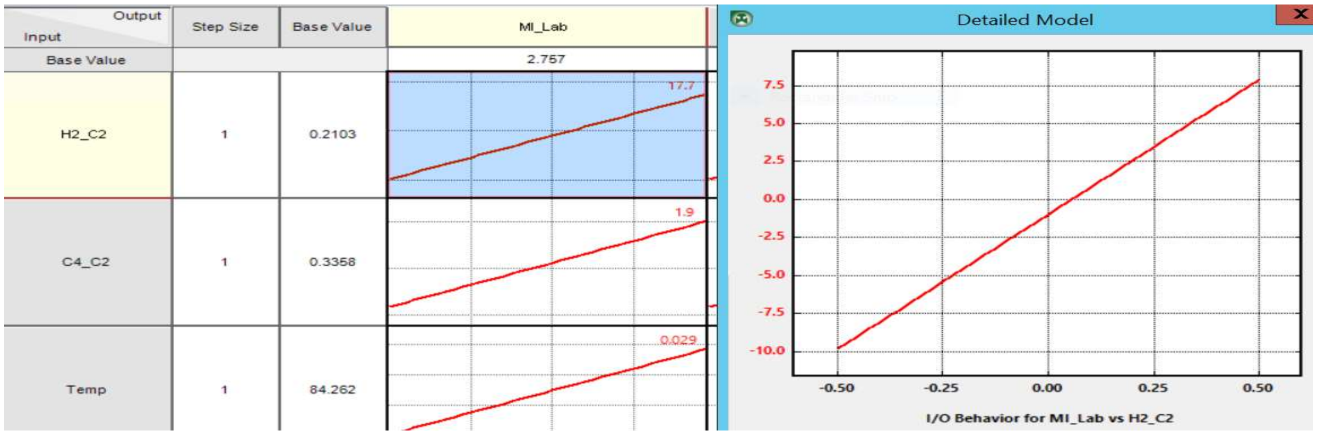
8.4.3b I/O Response Plot

Next, we click on the I/O button on the top tool ribbon to generate an I/O plot which represents the response of each output as each input is moved from its lower limit to its upper limit. See Figure 9.117 for the resulting I/O plot and Figure 8.118 for a detailed MI_Lab-H2_C2 I/O plot. In the I/O plot, the limits are either the validity limits, or the minimum and maximum values for that input from the current dataset. We see in Figure 8.117 that as MV C4_C2 increases, both Density_Lab and Density_Inst decrease. Table 8.5 summarizes the positive or negative sign of ΔCV/ΔMV or ΔCV/ΔDV for CVs (MI_Lab, MI_Inst, Density_Lab and Density_Inst) and MVs (H2_C2, C4_C2) or DVs (Temp, C2_Partial_Pressure).



Output / Input	Step Size	Base Value	MI_Lab	MI_Inst	Density_Lab	Density_Inst
Base Value			2.757	2.757	919.272	919.272
H2_C2	1	0.2103	17.7	17.7	12.2	12.2
C4_C2	1	0.3358	1.9	1.9	-15	-15
Temp	1	84.262	0.029	0.029	0.41	0.41
C2_Partial_Pressure	1	6.398	0.24	0.24	1.72	1.72

Figure 8.117 The I/O response curve of a nonlinear polypropylene process



Output / Input	Step Size	Base Value	MI_Lab
Base Value			2.757
H2_C2	1	0.2103	17.7
C4_C2	1	0.3358	1.9
Temp	1	84.262	0.029

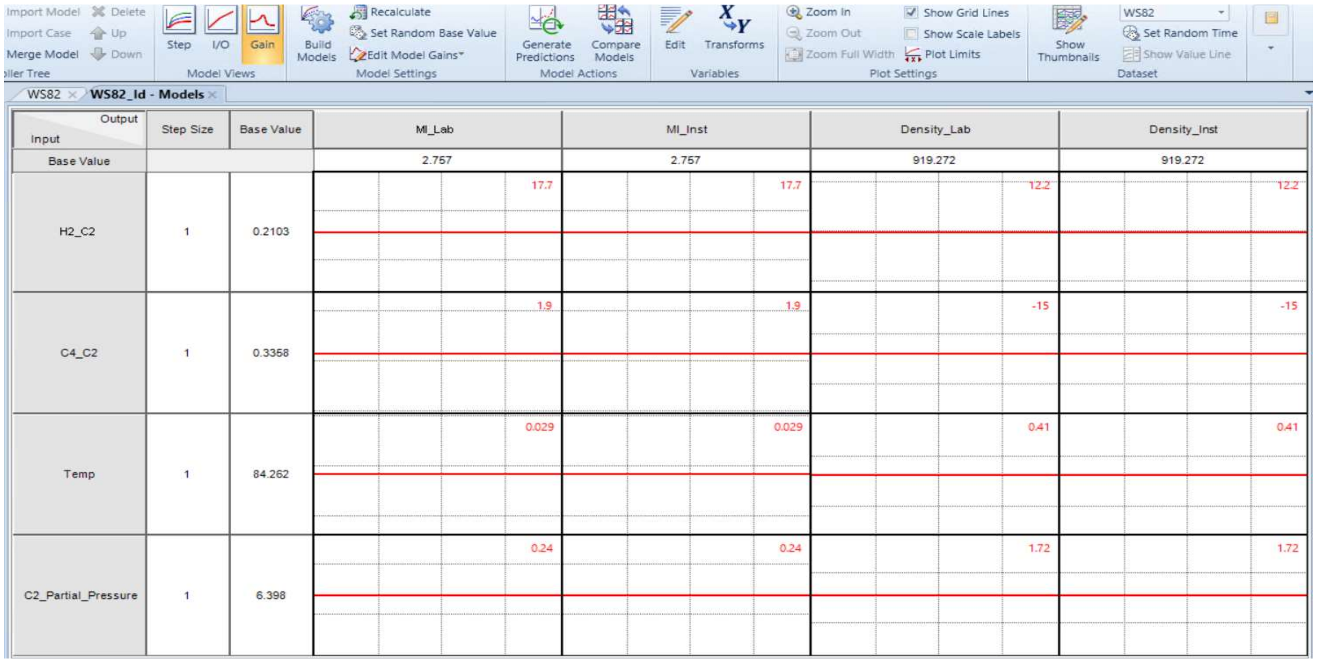
Figure 8.118 Showing the details of the MI_Lab-H2_C2 I/O response curve

Table 8.5 Positive or negative sign of ΔCV/ΔMV or ΔCV/ΔDV

MV or DV	CV – 1. MI	CV – 2. Density
1. H2_C2	ΔCV/ΔMV > 0	ΔCV/ΔMV > 0
2. H2_C4	ΔCV/ΔMV > 0	ΔCV/ΔMV < 0
3. Temp	ΔCV/ΔDV > 0	ΔCV/ΔDV > 0
4. H2_partial_pressure	ΔCV/ΔDV > 0	ΔCV/ΔDV > 0

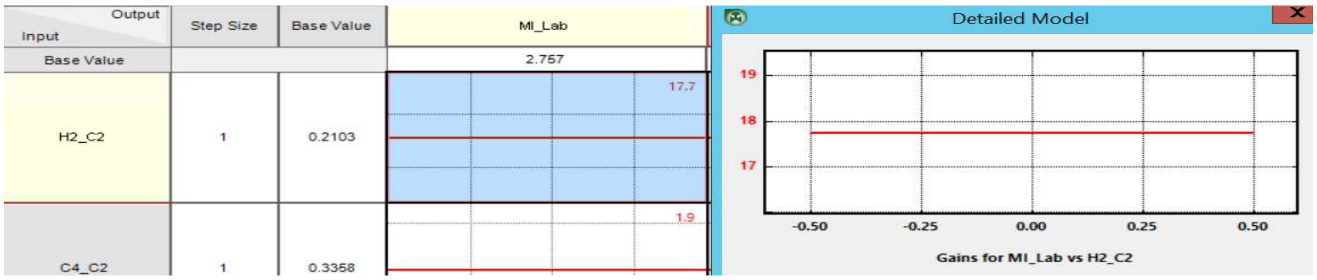
8.4.3c Gain Plot

Finally, we click on the “Gain” button on the top tool ribbon to generate a gain plot of Figure 8.119 which represents the amount of gain for each input/output pair as each input increases from its lower limit to its upper limit. Figure 8.120 presents a detailed MI_Lab-H2_C2 I/O plot.



Output / Input	Step Size	Base Value	MI_Lab	MI_Inst	Density_Lab	Density_Inst
Base Value			2.757	2.757	919.272	919.272
H2_C2	1	0.2103	17.7	17.7	12.2	12.2
C4_C2	1	0.3358	1.9	1.9	-15	-15
Temp	1	84.262	0.029	0.029	0.41	0.41
C2_Partial_Pressure	1	6.398	0.24	0.24	1.72	1.72

Figure 8.119 The gain plot of a nonlinear polypropylene plot.



Output / Input	Step Size	Base Value	MI_Lab
Base Value			2.757
H2_C2	1	0.2103	17.7
C4_C2	1	0.3358	1.9

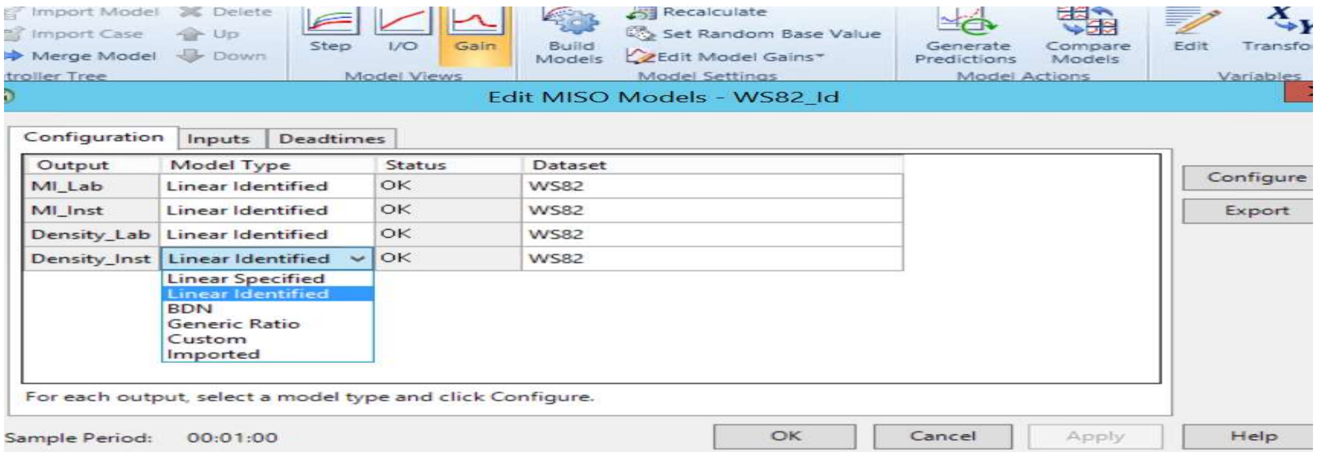
Figure 8.120 Showing the details of the MI_Lab-H2_C2 gain curve

8.4.4 Aspen Nonlinear Controller: Task 1 – Model Identification; Building the Nonlinear State-Space Bounded Derivative Network (SS-BDN)

8.4.4a Configure Dynamics and Output States

On the top tool ribbon, we click on “Build Models” button. We see an “Edit MISO Models” window and hit the “Configuration” button. This results in Figure 8.121, displaying the default model type, “Model Identified”. We replace each model type of our output variable (CV) by BDN through the drop-down menu. This leads to Figure 8.122. We see the “Inputs” button in Figure 8.122. Clicking on the “Inputs and specifying the inputs affecting each output, we see Figure 8.123. Next, we hit the “Deadtimes” and specify the initial Deadtimes in number of sample periods. These dead times are to model nonlinearity in the initial process response during changes. See Figure 8.124. We click on each output variable, followed by hitting the “Identify Deadtimes” button and keeping the default parameters, and then click on “Identify” to identify Deadtimes. See Figure 8.125. Repeat this step for all four output variables (MI_Lab, MI_Inst, Density_Lab and Density_Inst).

Having identified the Deadtimes, we click on “Configure” button displayed in Figures 8.121 to 125. This leads to Figure 8.126, in which we configure within the “Dynamics” tab, the filter time constants for input variables (He_C2, C4_C2, Temp, C2_Partial_Pressure) for output variable MI_Lab. We repeat this step for output variables MI-Inst, Density_Lab, and Density_Inst. Next, we switch to the “Output States” tab within “Configure” step, and configure one output states, one for each filter, as illustrated in Figure 8.127. We repeat this step for output variables MI-Inst, Density_Lab, and Density_Inst.



Edit MISO Models - WS82_Id

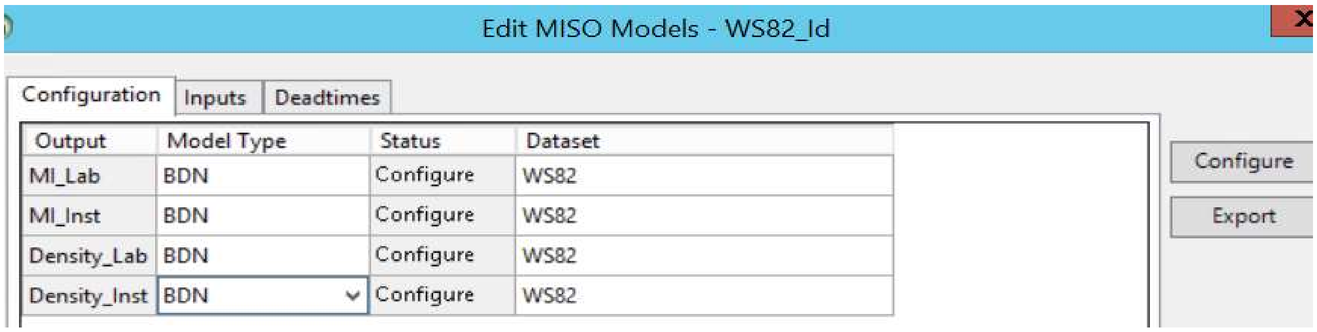
Output	Model Type	Status	Dataset
MI_Lab	Linear Identified	OK	WS82
MI_Inst	Linear Identified	OK	WS82
Density_Lab	Linear Identified	OK	WS82
Density_Inst	Linear Identified	OK	WS82

Linear Specified
Linear Identified
BDN
Generic Ratio
Custom
Imported

For each output, select a model type and click Configure.

Sample Period: 00:01:00

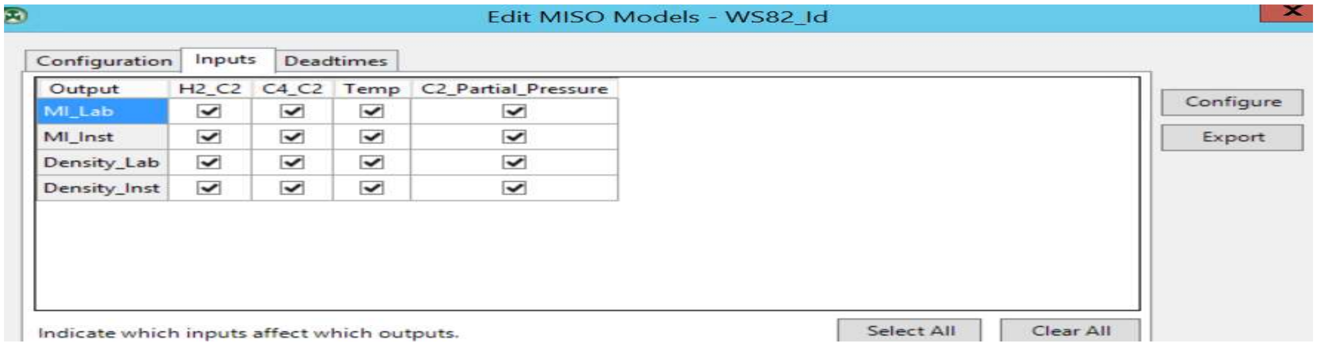
Figure 8.121 Displaying available model types



Edit MISO Models - WS82_Id

Output	Model Type	Status	Dataset
MI_Lab	BDN	Configure	WS82
MI_Inst	BDN	Configure	WS82
Density_Lab	BDN	Configure	WS82
Density_Inst	BDN	Configure	WS82

Figure 8.122 Choosing the model type, bounded derivative network (BDN)

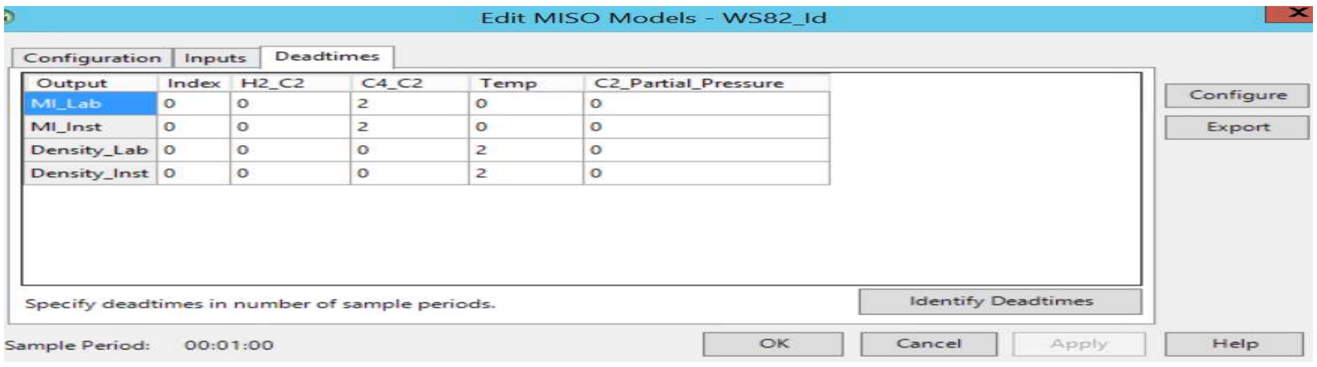


Edit MISO Models - WS82_Id

Output	H2_C2	C4_C2	Temp	C2_Partial_Pressure
MI_Lab	✓	✓	✓	✓
MI_Inst	✓	✓	✓	✓
Density_Lab	✓	✓	✓	✓
Density_Inst	✓	✓	✓	✓

Indicate which inputs affect which outputs.

Figure 8.123 Specifying the input variables affecting each output



Edit MISO Models - WS82_Id

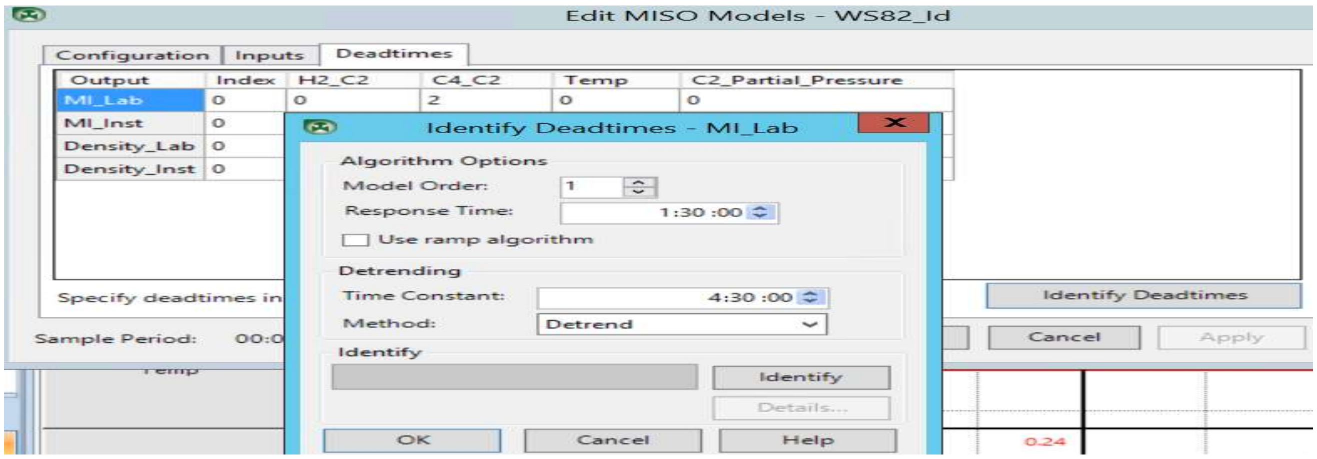
Configuration | Inputs | Deadtimes

Output	Index	H2_C2	C4_C2	Temp	C2_Partial_Pressure
MI_Lab	0	0	2	0	0
MI_Inst	0	0	2	0	0
Density_Lab	0	0	0	2	0
Density_Inst	0	0	0	2	0

Specify deadtimes in number of sample periods.

Sample Period: 00:01:00

Figure 8.124 Specifying the dead times: “2” refers to the number of sample periods (called index value)



Identify Deadtimes - MI_Lab

Algorithm Options

Model Order: 1

Response Time: 1:30 :00

☐ Use ramp algorithm

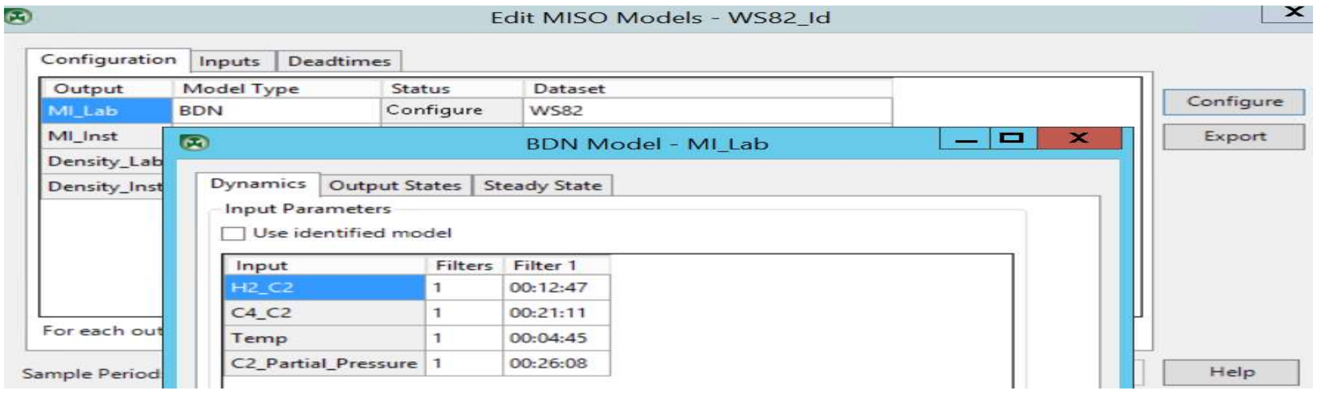
Detrending

Time Constant: 4:30 :00

Method: Detrend

Identify

Figure 8.125 Identify for Deadtimes for MI_Lab. Repeat this step for all four output variables.



Edit MISO Models - WS82_Id

Configuration | Inputs | Deadtimes

Output	Model Type	Status	Dataset
MI_Lab	BDN	Configure	WS82
MI_Inst			
Density_Lab			
Density_Inst			

BDN Model - MI_Lab

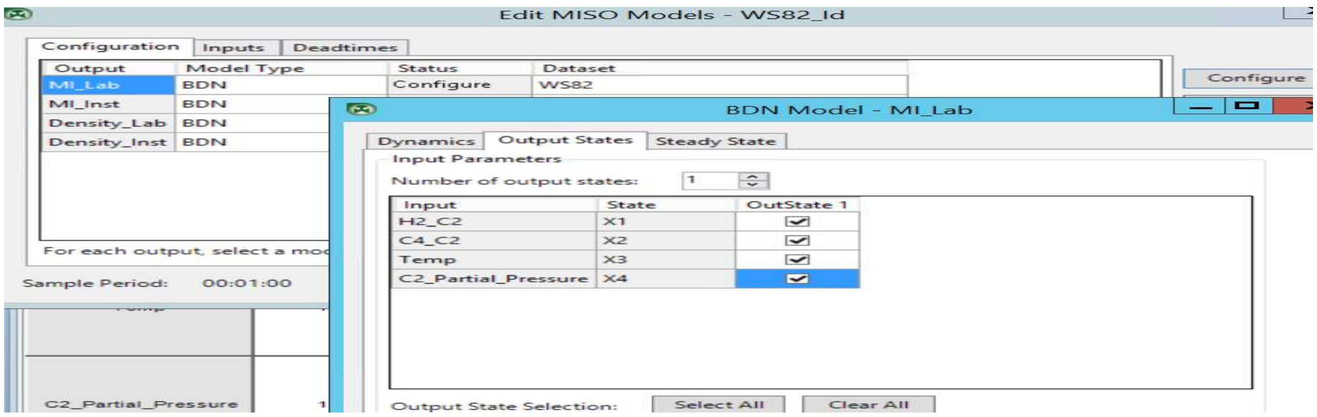
Dynamics | Output States | Steady State

Input Parameters

☐ Use identified model

Input	Filters	Filter 1
H2_C2	1	00:12:47
C4_C2	1	00:21:11
Temp	1	00:04:45
C2_Partial_Pressure	1	00:26:08

Figure 8.126 Configure the filter time constants for input variables (He_C2, C4_C2, Temp, C2_Partial_Pressure) for output variable MI_Lab



Output	Model Type	Status	Dataset
MI_Lab	BDN	Configure	WS82
MI_Inst	BDN		
Density_Lab	BDN		
Density_Inst	BDN		

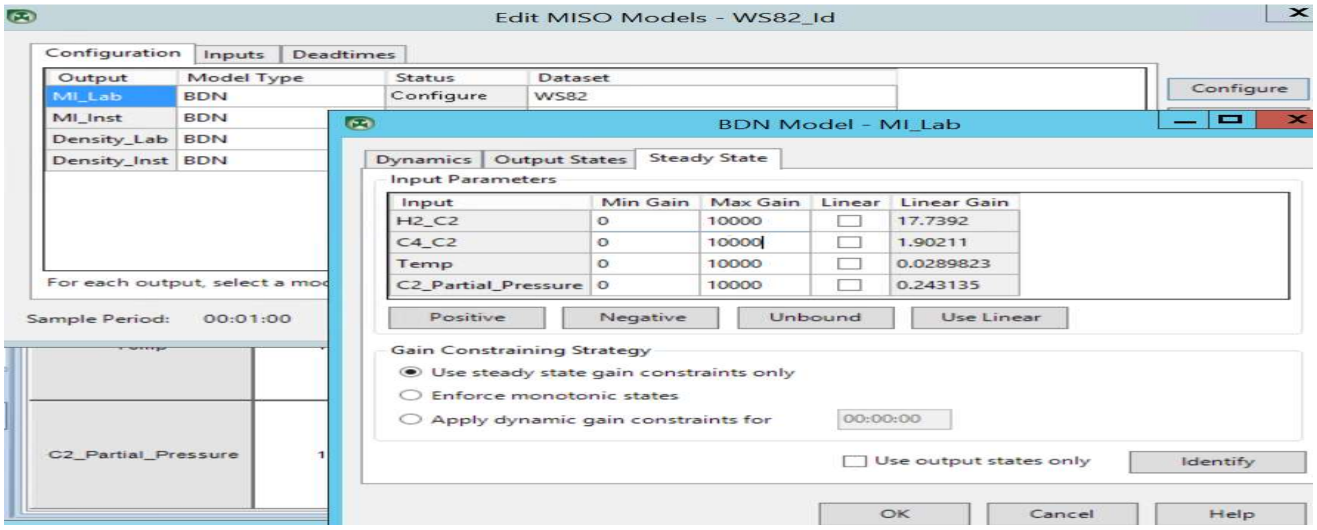
Input	State	OutState 1
H2_C2	X1	✓
C4_C2	X2	✓
Temp	X3	✓
C2_Partial_Pressure	X4	✓

Figure 8.127 Configure the output states for each filter for input variables (He_C2, C4_C2, Temp, C2_Partial_Pressure) for output variable MI_Lab

8.4.4b Build Model with Gain Constraints

This step features a significant ability of the BDN to build a model based on specified gain constraints to avoid incorrect gain inversion discussed in Section 8.3.2a. Based on Figure 8.117 and Table 8.5 in Section 8.4.3b, we can specify the corresponding gain constraints.

We specify the steady-state gain constraints by continuing the “Configure” step, and clicking on “Steady State tab displayed in Figure 8.127, corresponding to output or CV, MI_Lab. This gives Figure 8.128, in which we specify a Min Gain of 0 and a Max Gain of 10000 for a positive gain; and a Min Gain of -10000 and a Max Gain of 0 for a negative gain following Table 8.5. We then select “Identify” to build the BDN model for the MI-Lab. Figure 8.129 shows the resulting comparison between the nonlinear BDN model prediction and plant data for MI-Lab. For an average MI_Lab value of 4, the root-mean-squared error (RMSE) between the model prediction and plant data is only 0.0221, or approximately 0.55%.



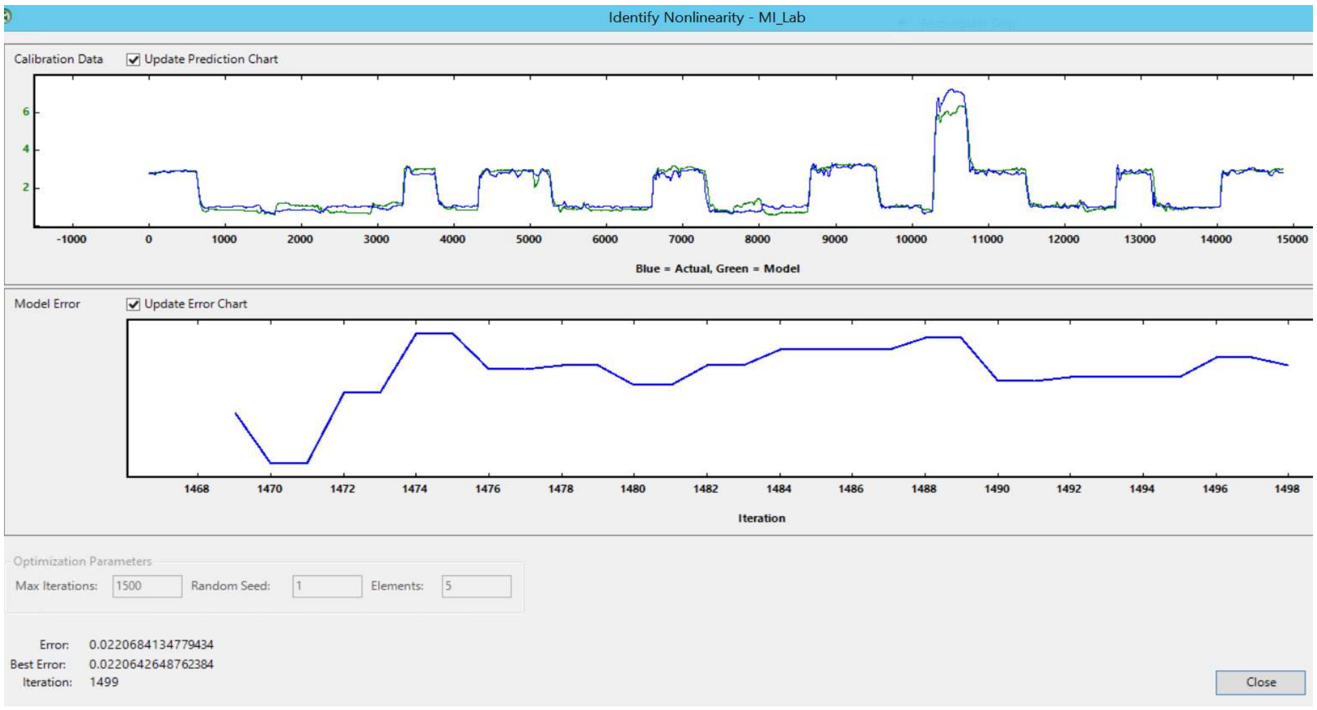
Output	Model Type	Status	Dataset
MI_Lab	BDN	Configure	WS82
MI_Inst	BDN		
Density_Lab	BDN		
Density_Inst	BDN		

Input	Min Gain	Max Gain	Linear	Linear Gain
H2_C2	0	10000	☐	17.7392
C4_C2	0	10000	☐	1.90211
Temp	0	10000	☐	0.0289823
C2_Partial_Pressure	0	10000	☐	0.243135

Gain Constraining Strategy
- ● Use steady state gain constraints only
- ○ Enforce monotonic states
- ○ Apply dynamic gain constraints for 00:00:00

☐ Use output states only

Figure 8.128 Specify the steady-state gain constraints for MI-Lab following Table 8.5



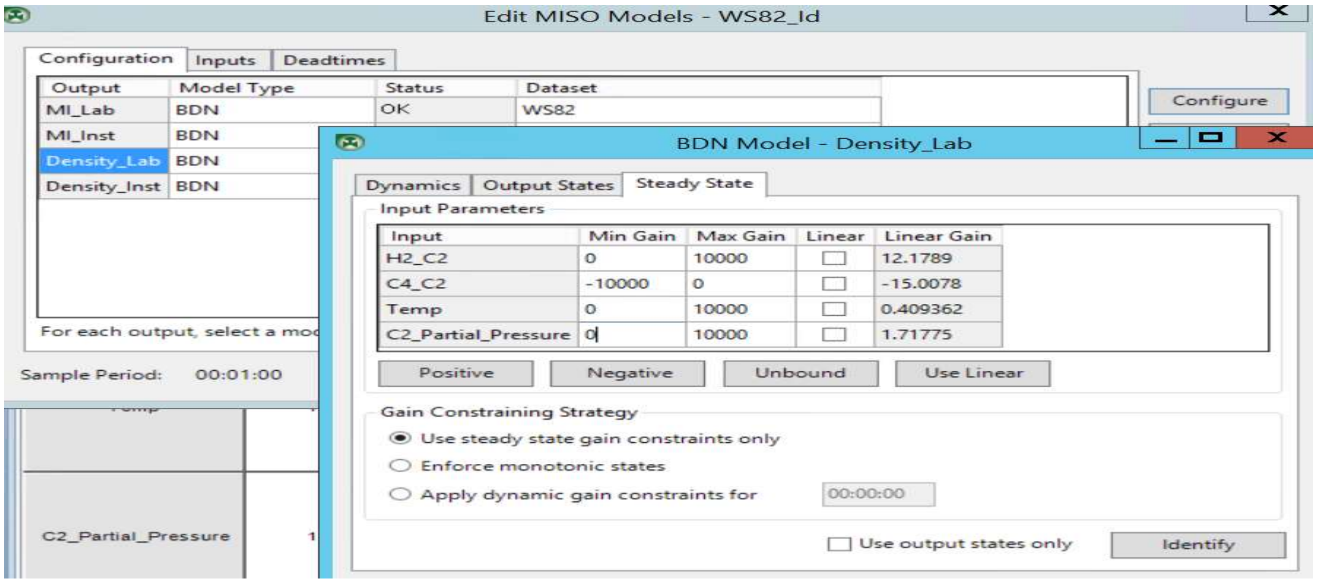
Optimization Parameters

Max Iterations: 1500 Random Seed: 1 Elements: 5

```
     Error:  0.0220684134779434
Best Error:  0.0220642648762384
 Iteration:  1499
```

Figure 8.129 Comparison between nonlinear BDN model prediction with plant data for MI_Lab with a "Max Gain" of 10000

Figure 8.130 shows the specification of steady-state gain constraints for Density_Lab, following Table 8.5, and Figure 8.131 compares the predicted Density_Lab values from the nonlinear BDN model with plant data. For an average value of Density_Lab of 918, the RMSE between the model prediction and plant data is only 0.007485, or approximately 0.008154%.



Output	Model Type	Status	Dataset
MI_Lab	BDN	OK	WS82
MI_Inst	BDN		
Density_Lab	BDN		
Density_Inst	BDN		

Sample Period: 00:01:00

Input Parameters

Input	Min Gain	Max Gain	Linear	Linear Gain
H2_C2	0	10000	☐	12.1789
C4_C2	-10000	0	☐	-15.0078
Temp	0	10000	☐	0.409362
C2_Partial_Pressure	0	10000	☐	1.71775

Gain Constraining Strategy

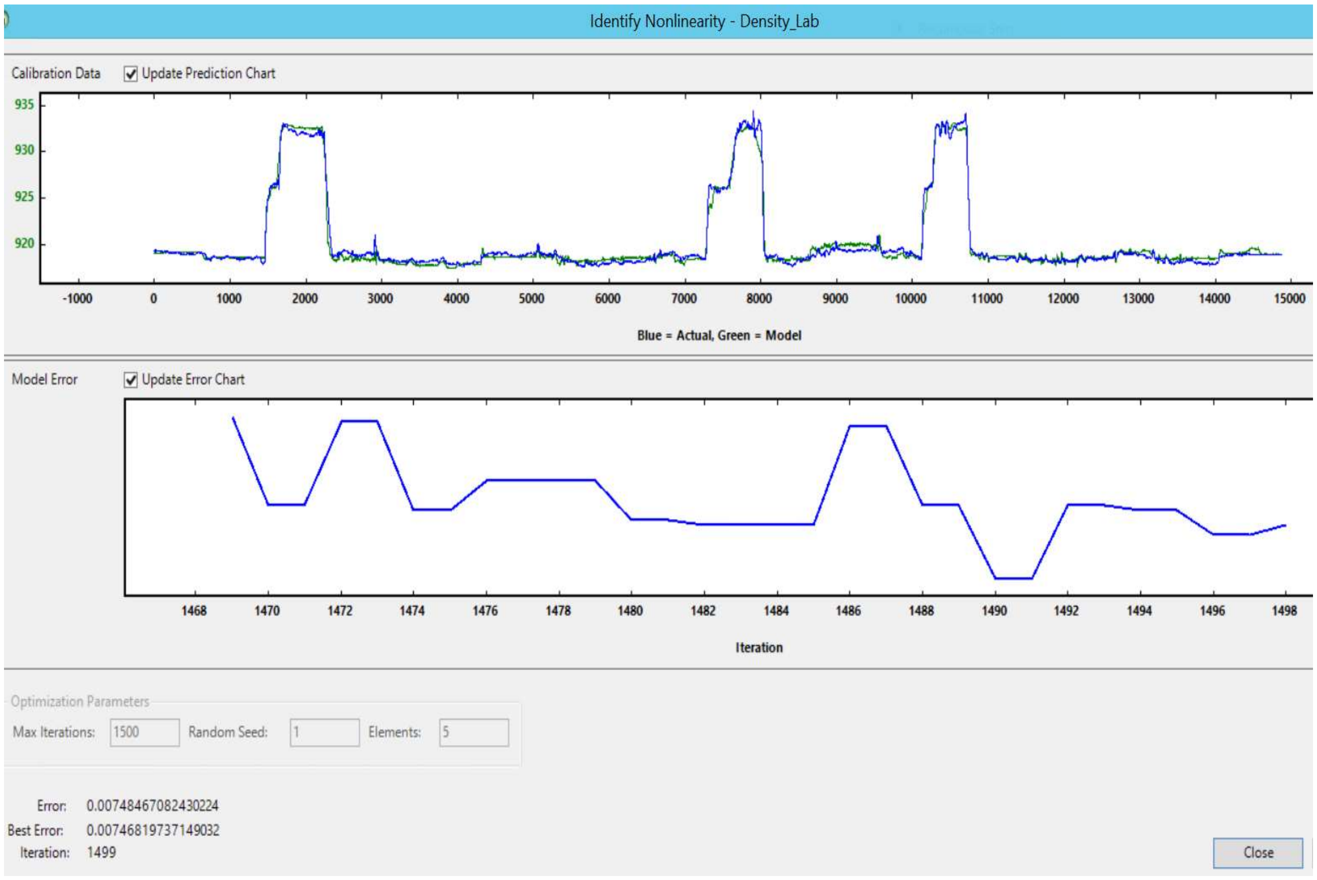
● Use steady state gain constraints only

○ Enforce monotonic states

○ Apply dynamic gain constraints for 00:00:00

☐ Use output states only

Figure 8.130 Specify the steady-state gain constraints for Density-Lab following Table 8.5



Optimization Parameters

Max Iterations: 1500 Random Seed: 1 Elements: 5

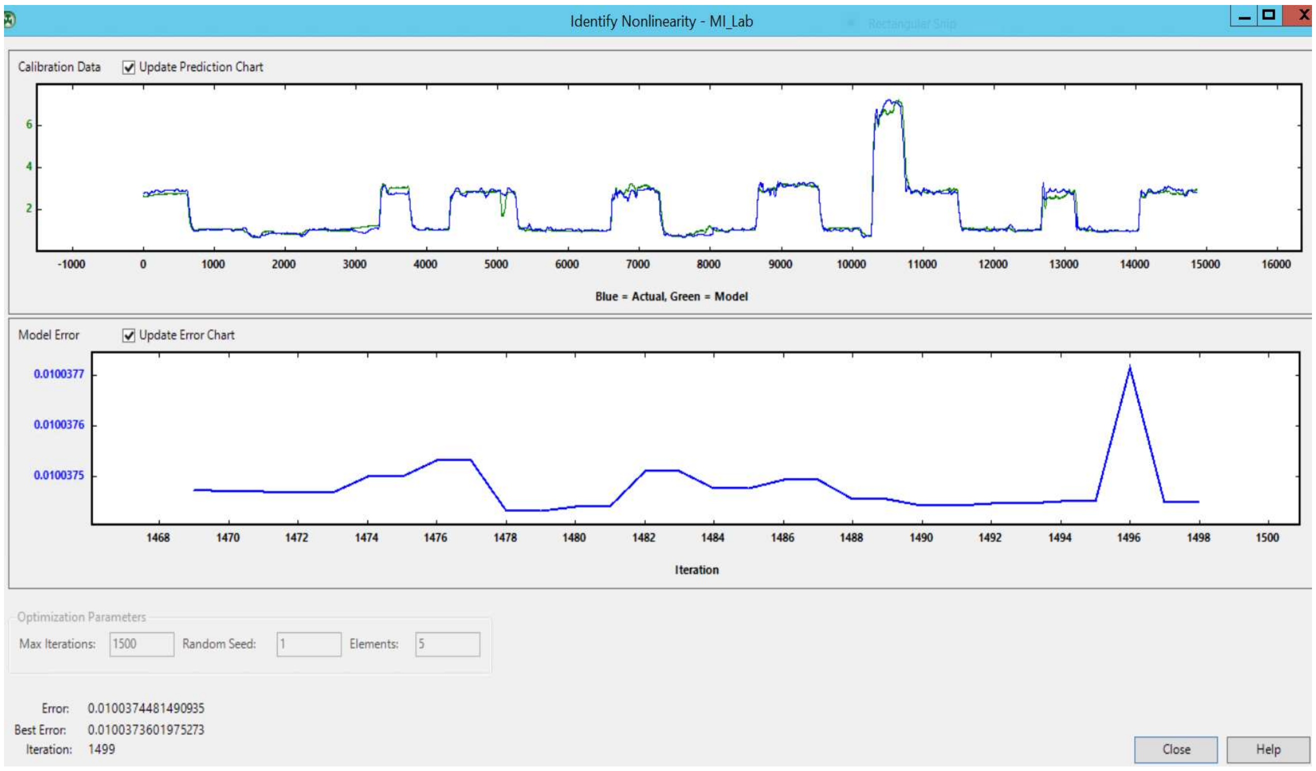
Error: 0.00748467082430224
Best Error: 0.00746819737149032
Iteration: 1499

Figure 8.131 Comparison between nonlinear BDN model prediction with plant data for Density_Lab with a "Min Gain" for C4_C2 of -10000, and a "Max Gain" for the remaining inputs of +10000.

Following the steady-state gain constraints of Table 8.5 and repeating the same procedure to identify the models for MI_Inst and Density_Inst, we get essentially identical comparison curves as in Figures 8.129 for MI and 8.131 for Density.

8.4.4c Fine-Tune Steady-State BDN Gains

Referring to Figure 8.128, we narrow the range of the steady-state BDN gain by lowering the "Max Gain" for all four inputs from 10000 to 100 and run the BDN regression again. This results in Figure 8.132, in which the error between the model prediction and plant data of MI_Lab drops from 0.022068413 to 0.010037448. Likewise, referring to Figure 8.130 for Density_Lab, we change the "Min Gain" for C4_C2 to -100, and the "Max Gain" for the remaining three inputs to 100, and run the BDN regression. We find that the resulting error between the model prediction and plant data of Density_Lab shows no improvement.

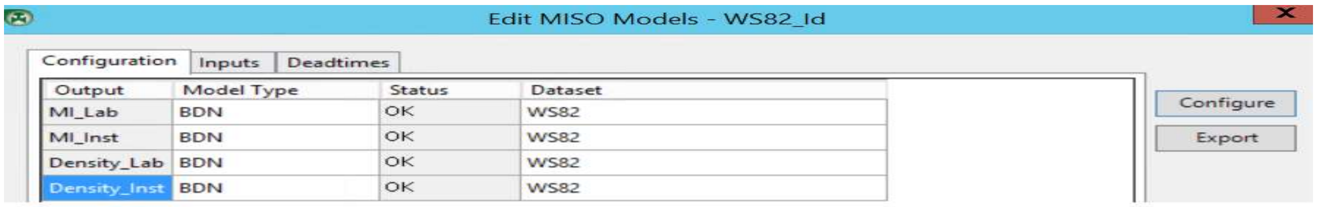


Max Iterations: 1500 Random Seed: 1 Elements: 5

Error: 0.0100374481490935
Best Error: 0.0100373601975273
Iteration: 1499

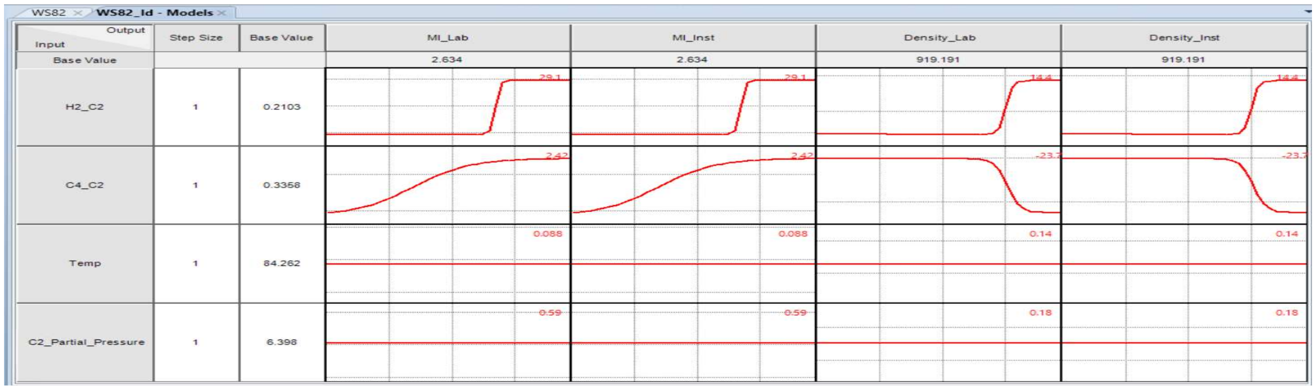
Figure 8.132 Comparison between nonlinear BDN model prediction with plant data for MI_Lab with a "Max Gain" of 100

After configuring and identifying the SS-BDN model, we see the "OK" status of the model identification, as seen in Figure 8.133. We also see the resulting steady-state gain plot of Figure 8.134.



Output	Model Type	Status	Dataset
MI_Lab	BDN	OK	WS82
MI_Inst	BDN	OK	WS82
Density_Lab	BDN	OK	WS82
Density_Inst	BDN	OK	WS82

Figure 8.133 Status" OK" indicating completion of the SS-BDN model identification



Output / Input	Step Size	Base Value	MI_Lab	MI_Inst	Density_Lab	Density_Inst
Base Value			2.634	2.634	919.191	919.191
H2_C2	1	0.2103	29.1	29.1	14.4	14.4
C4_C2	1	0.3358	2.42	2.42	-23.7	-23.7
Temp	1	84.262	0.088	0.088	0.14	0.14
C2_Partial_Pressure	1	6.398	0.59	0.59	0.18	0.18

Figure 8.134 Steady-state gain plot of the SS-BDN model

Table 8.6 shows the resulting steady-state gain for the SS-BDN model. In practice, we only pay attention to the columns of MI_Lab and Density_Lab. We do not need to develop the model for MI_Inst and Density_Inst.

Table 8.6 Steady-state gains of the SS-BDN model

MV/DV↓ CV→	MI_Lab	MI_Inst	Density_Lab	Density_Inst
H2_C2	29.1	29.1	14.4	14.4
C4_C2	2.42	2.42	-23.7	-23.7
Temp	0.088	0.088	0.44	0.44
C2_Partial_Pressure	0.59	0.59	0.18	0.18

8.4.4d Generate Model Predictions

Next, we apply the SS-BDN model to predict the MI_Lab and Density_Lab, and compare the predictions with plant data. We click on "Generate Predictions" on the top tool ribbons, and choose dataset WS82. See Figure 8.135. The resulting comparison appears in Figure 8.136.

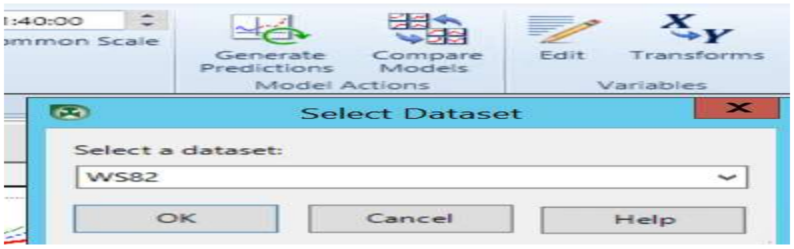


Figure 8.135 Select dataset to compare with model predictions

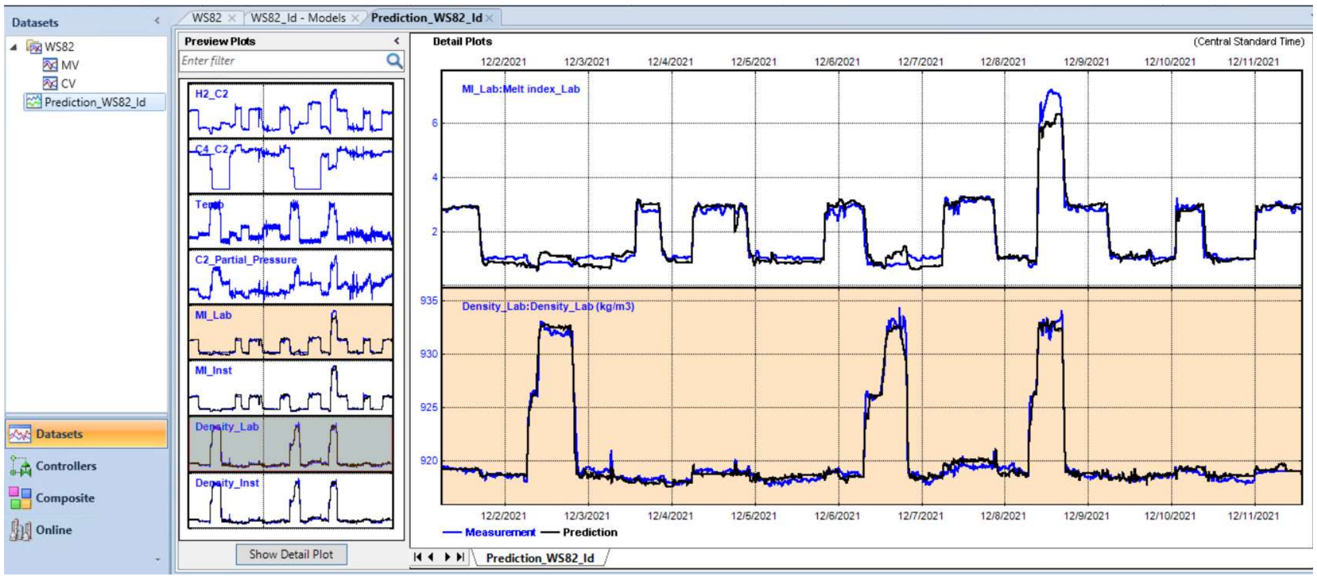
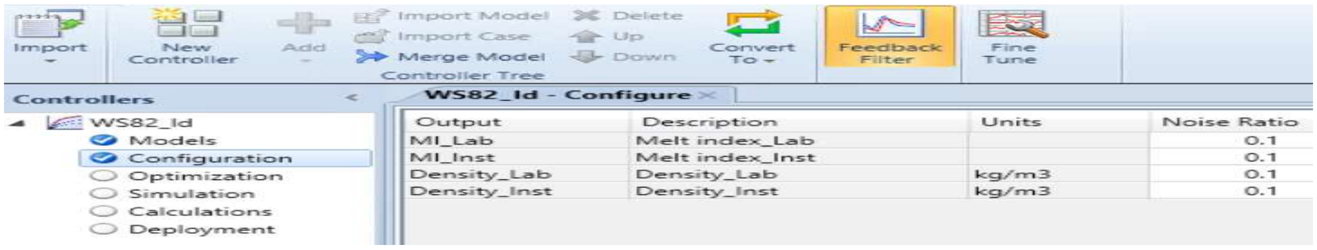


Figure 8.136 Generated model predictions of MI_Lab and Density_Lab.

8.4.5 Aspen Nonlinear Controller: Task 2 – Configuration – Model Configuration

As discussed in Section 8.2.14, the model configuration task involves the specifications of feedback filters for prediction errors, based on prediction error filtering covered in Section 8.1.5. We click on the "Feedback Filter" button within the top tool ribbons, followed by "Fine Tune" button to fine-tune the feedback filter. See Figures 8.137 and 8.138.



Output	Description	Units	Noise Ratio
MI_Lab	Melt index_Lab		0.1
MI_Inst	Melt index_Inst		0.1
Density_Lab	Density_Lab	kg/m3	0.1
Density_Inst	Density_Inst	kg/m3	0.1

Figure 8.137 Choosing the default feedback filter

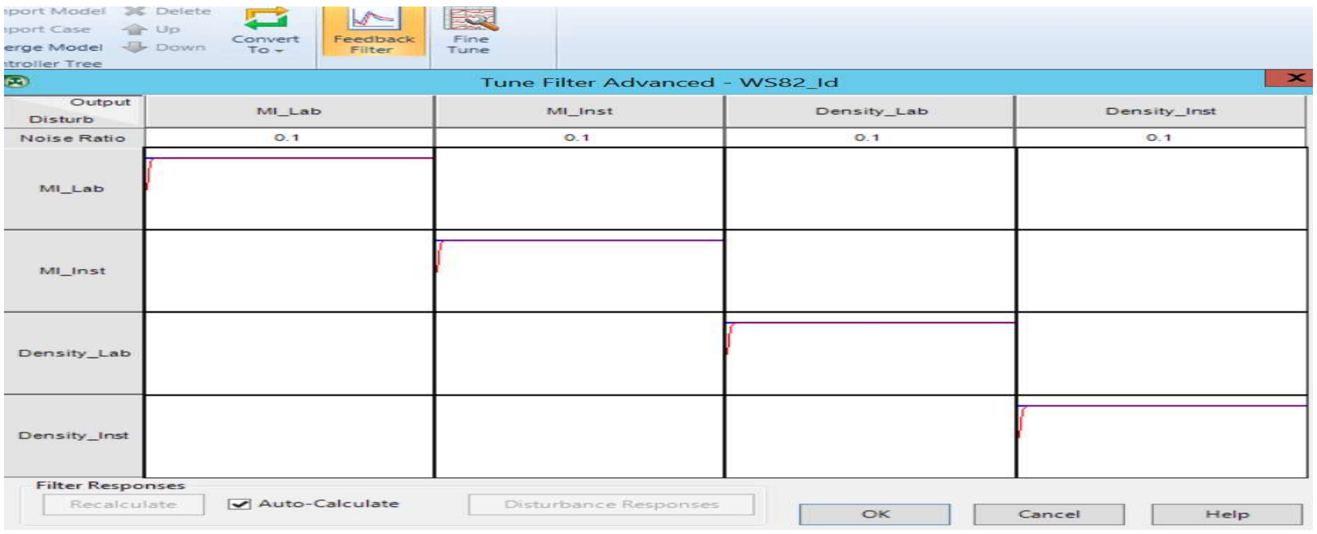
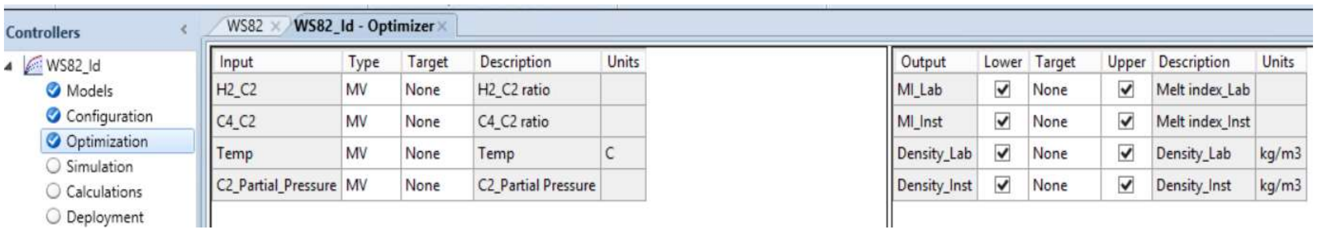


Figure 8.138 Fine-tune the feedback filter

8.4.6 Aspen Nonlinear Controller: Task 2 – Configuring and Runnng the Steady-State Optimiation

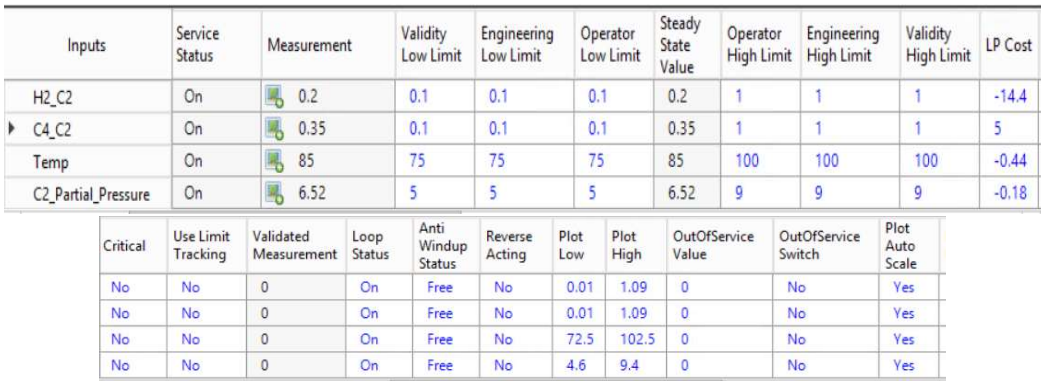
We follow Figure 8.92 to configure the steady-state optimizer. Figure 8.139 specifies the inputs and outputs to configure the optimizer. Figures 8.140a-b show the input entries for MVs and CVs for the steady-state simulator, respectvely. In Figure 8.140a, we set the initially LP costs for MVs based on the negative values of the steady-state gains reported in Table 8.6. We do this by following the example illustrated in Figure 8.12 and Table 8.1. We note that in Table 8.6, in the MI_Lab column, all gains are positive; in the Density_Lab column, the gain Δ(Density_Lab)/Δ(C4_C2) is negative. In the following, we choose the negative values of the steady-state gains in the Density_Lab column of Table 8.6 as our initial LP costs, except that we change the gain value for Δ(Density_Lab)/Δ(C4_C2) from -23.7 to -5 (hence, the LP cost becomes +5 for MV or input C4_C2 in Figure 8.140a). As discussed in Section 8.1.2b and Table 8.1, an input or a MV with a positive LP cost means that to minimize cost and maximize profit, we tend to move the MV towards its lower operating limit. By contrast, we tend to move a MV with a negative LP cost towards its upper operating limit. We will explore the impact of having different initial LP costs on the resulting steady-state targets of MVs and CVs.



Input	Type	Target	Description	Units
H2_C2	MV	None	H2_C2 ratio	
C4_C2	MV	None	C4_C2 ratio	
Temp	MV	None	Temp	C
C2_Partial_Pressure	MV	None	C2_Partial Pressure	

Output	Lower	Target	Upper	Description	Units
MI_Lab	✓	None	✓	Melt index_Lab	
MI_Inst	✓	None	✓	Melt index_Inst	
Density_Lab	✓	None	✓	Density_Lab	kg/m3
Density_Inst	✓	None	✓	Density_Inst	kg/m3

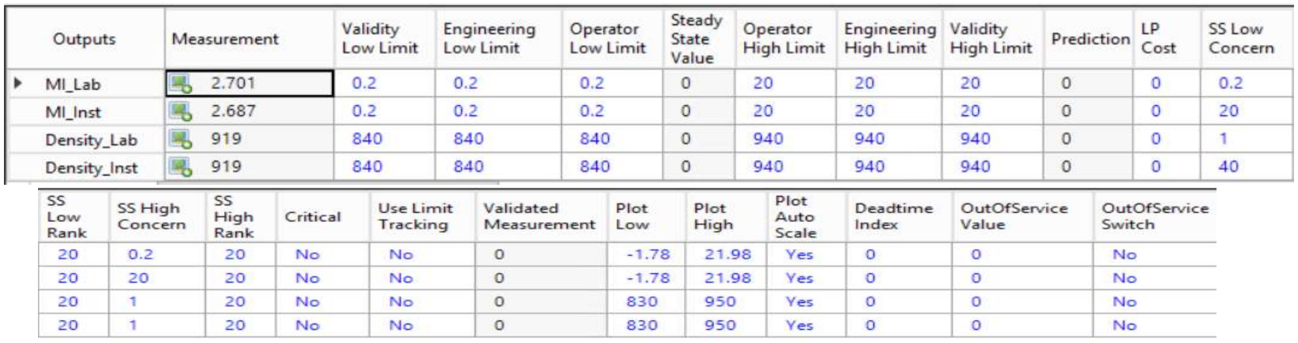
Figure 8.139 Inputs and outputs for steady-state optimization



Inputs	Service Status	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Operator High Limit	Engineering High Limit	Validity High Limit	LP Cost
H2_C2	On	0.2	0.1	0.1	0.1	0.2	1	1	1	-14.4
C4_C2	On	0.35	0.1	0.1	0.1	0.35	1	1	1	5
Temp	On	85	75	75	75	85	100	100	100	-0.44
C2_Partial_Pressure	On	6.52	5	5	5	6.52	9	9	9	-0.18

Critical	Use Limit Tracking	Validated Measurement	Loop Status	Anti Windup Status	Reverse Acting	Plot Low	Plot High	OutOfService Value	OutOfService Switch	Plot Auto Scale
No	No	0	On	Free	No	0.01	1.09	0	No	Yes
No	No	0	On	Free	No	0.01	1.09	0	No	Yes
No	No	0	On	Free	No	72.5	102.5	0	No	Yes
No	No	0	On	Free	No	4.6	9.4	0	No	Yes

Figure 8.140a Specifications of MVs for steady-state simulator.

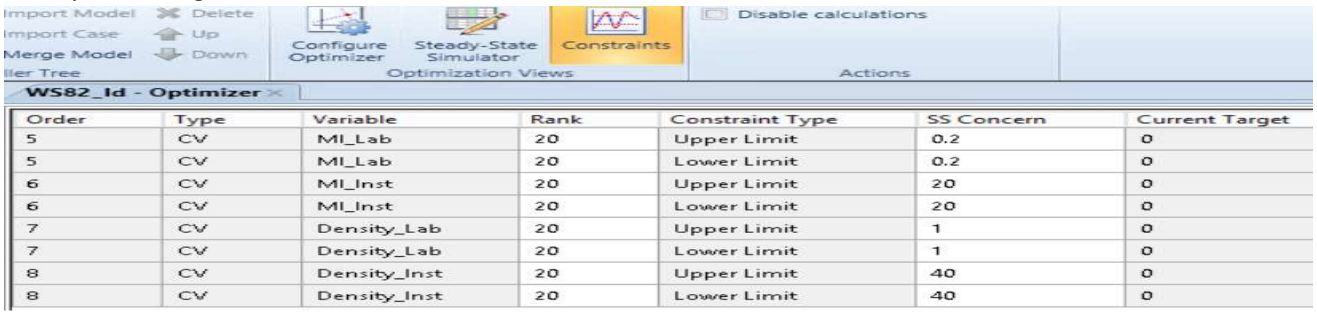


Outputs	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Operator High Limit	Engineering High Limit	Validity High Limit	Prediction	LP Cost	SS Low Concern
MI_Lab	2.701	0.2	0.2	0.2	0	20	20	20	0	0	0.2
MI_Inst	2.687	0.2	0.2	0.2	0	20	20	20	0	0	20
Density_Lab	919	840	840	840	0	940	940	940	0	0	1
Density_Inst	919	840	840	840	0	940	940	940	0	0	40

SS Low Rank	SS High Concern	SS High Rank	Critical	Use Limit Tracking	Validated Measurement	Plot Low	Plot High	Plot Auto Scale	Deadtime Index	OutOfService Value	OutOfService Switch
20	0.2	20	No	No	0	-1.78	21.98	Yes	0	0	No
20	20	20	No	No	0	-1.78	21.98	Yes	0	0	No
20	1	20	No	No	0	830	950	Yes	0	0	No
20	1	20	No	No	0	830	950	Yes	0	0	No

Figure 8.140b Specifications of CVs for steady-state simulator

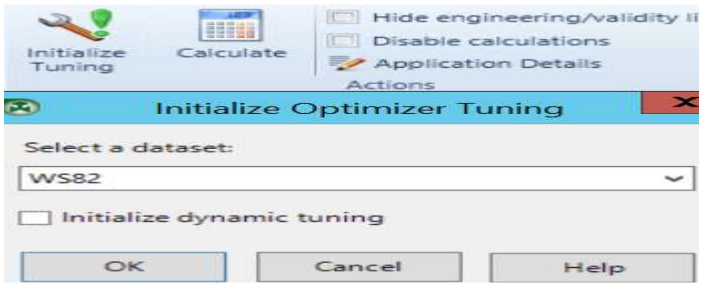
Next, we click on the “Constraints” button, and see the display of Figure 8.141. We are to calculate the steady-state targets of CVs and MVs.



Order	Type	Variable	Rank	Constraint Type	SS Concern	Current Target
5	CV	MI_Lab	20	Upper Limit	0.2	0
5	CV	MI_Lab	20	Lower Limit	0.2	0
6	CV	MI_Inst	20	Upper Limit	20	0
6	CV	MI_Inst	20	Lower Limit	20	0
7	CV	Density_Lab	20	Upper Limit	1	0
7	CV	Density_Lab	20	Lower Limit	1	0
8	CV	Density_Inst	20	Upper Limit	40	0
8	CV	Density_Inst	20	Lower Limit	40	0

Figure 8.141 Current constraints of MVs and CVs for steady-state optimization

We initialize the steady-state optimizer calculation by specifying th dataset WS82 and cancelling the initialization of the dynamic tuning. See Figure 8.142.

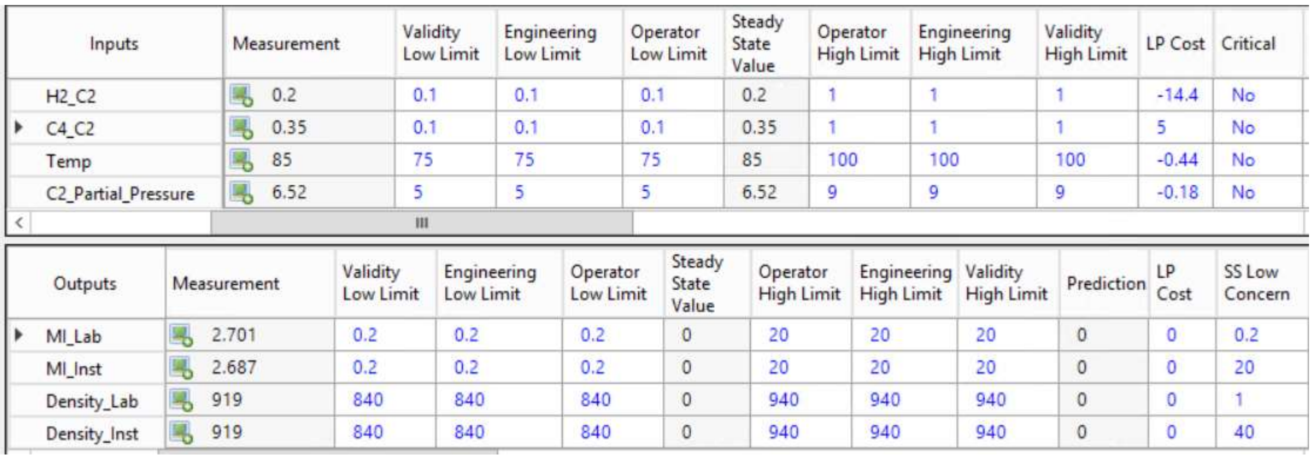


Select a dataset:

WS82

☐ Initialize dynamic tuning

Figure 8.142 Initialize steady-state optimizer by specifying the dataset



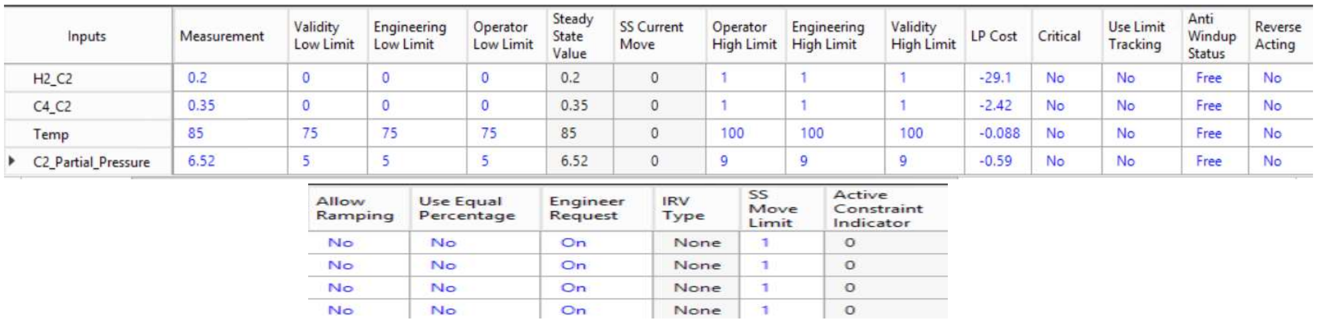
Inputs	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Operator High Limit	Engineering High Limit	Validity High Limit	LP Cost	Critical
H2_C2	0.2	0.1	0.1	0.1	0.2	1	1	1	-14.4	No
C4_C2	0.35	0.1	0.1	0.1	0.35	1	1	1	5	No
Temp	85	75	75	75	85	100	100	100	-0.44	No
C2_Partial_Pressure	6.52	5	5	5	6.52	9	9	9	-0.18	No

Outputs	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Operator High Limit	Engineering High Limit	Validity High Limit	Prediction	LP Cost	SS Low Concern
MI_Lab	2.701	0.2	0.2	0.2	0	20	20	20	0	0	0.2
MI_Inst	2.687	0.2	0.2	0.2	0	20	20	20	0	0	20
Density_Lab	919	840	840	840	0	940	940	940	0	0	1
Density_Inst	919	840	840	840	0	940	940	940	0	0	40

Figure 8.143 Steady-state values obtained by the steady-state optimizer

We now explore the impact of using the negative values of the steady-state gains in the MI_Lab column of Table 8.6 as our initial LP costs. Figure 8.144 shows the specifications MVs for steady-state simulator. The specifications for the CV are identical to those displayed in Figure 8.140b. Following the same procedure as in Figures 8.141 to 8.142, we find the results of steady-state values obtained by the steady-state optimizer in Figure 8.145, which are different from those displayed in Figure 8.143. This comparison demonstrates that the initial LP cost specifications affect the resulting steady-state targets for both MVs and CVs.

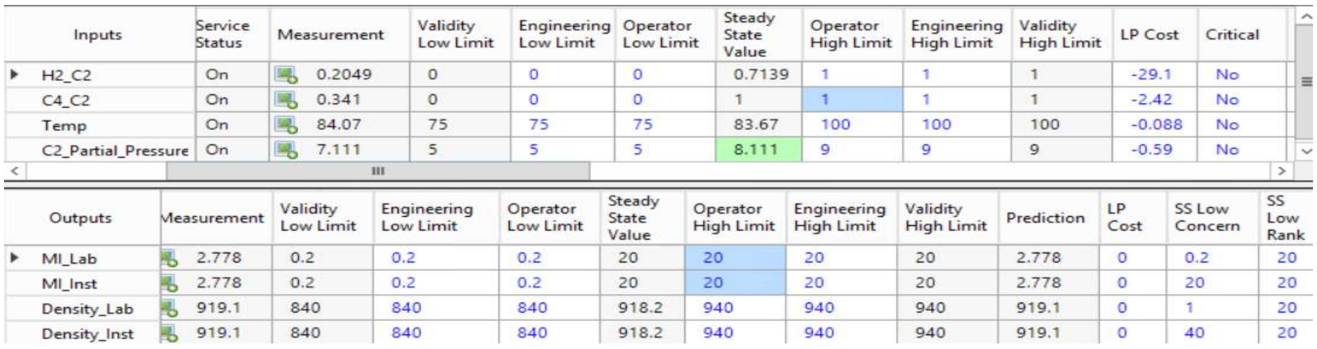
In Table 8,6, between the negative values of Density_Lab column and of MI_Lab column, which set of values should we use as initial LP costs displayed in Figures 8.143 and 8.145? We suggest choosing the set of initial LP costs that gives us the steady-state target values of CV that are close to our intended controller operation.



Inputs	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	SS Current Move	Operator High Limit	Engineering High Limit	Validity High Limit	LP Cost	Critical	Use Limit Tracking	Anti Windup Status	Reverse Acting
H2_C2	0.2	0	0	0	0.2	0	1	1	1	-29.1	No	No	Free	No
C4_C2	0.35	0	0	0	0.35	0	1	1	1	-2.42	No	No	Free	No
Temp	85	75	75	75	85	0	100	100	100	-0.088	No	No	Free	No
C2_Partial_Pressure	6.52	5	5	5	6.52	0	9	9	9	-0.59	No	No	Free	No

Allow Ramping	Use Equal Percentage	Engineer Request	IRV Type	SS Move Limit	Active Constraint Indicator
No	No	On	None	1	0
No	No	On	None	1	0
No	No	On	None	1	0
No	No	On	None	1	0

Figure 8.142 Specifications of MVs for steady-state simulator



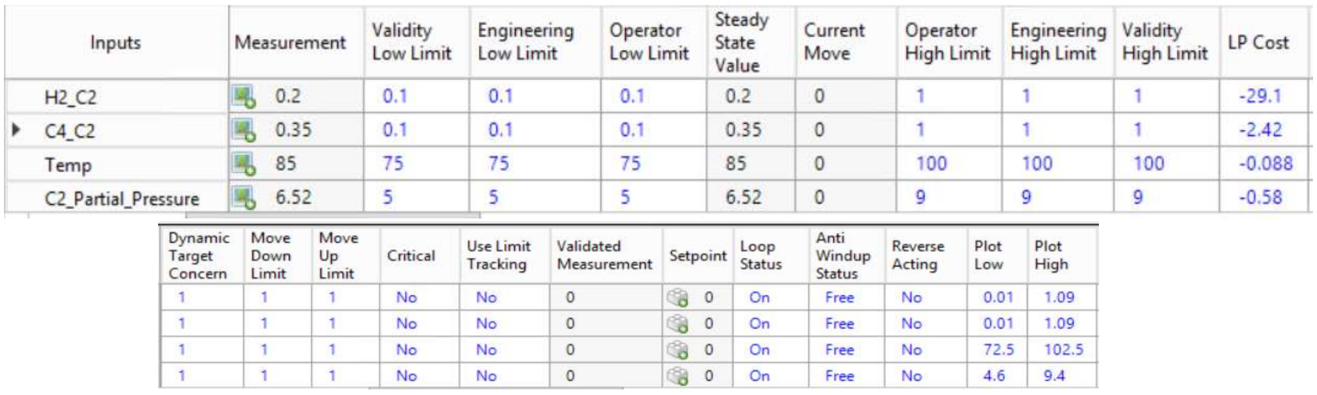
Inputs	Service Status	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Operator High Limit	Engineering High Limit	Validity High Limit	LP Cost	Critical
H2_C2	On	0.2049	0	0	0	0.7139	1	1	1	-29.1	No
C4_C2	On	0.341	0	0	0	1	1	1	1	-2.42	No
Temp	On	84.07	75	75	75	83.67	100	100	100	-0.088	No
C2_Partial_Pressure	On	7.111	5	5	5	8.111	9	9	9	-0.59	No

Outputs	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Operator High Limit	Engineering High Limit	Validity High Limit	Prediction	LP Cost	SS Low Concern	SS Low Rank
MI_Lab	2.778	0.2	0.2	0.2	20	20	20	20	2.778	0	0.2	20
MI_Inst	2.778	0.2	0.2	0.2	20	20	20	20	2.778	0	20	20
Density_Lab	919.1	840	840	840	918.2	940	940	940	919.1	0	1	20
Density_Inst	919.1	840	840	840	918.2	940	940	940	919.1	0	40	20

Figure 8.143 Steady-state values obtained by the steady-state optimizer

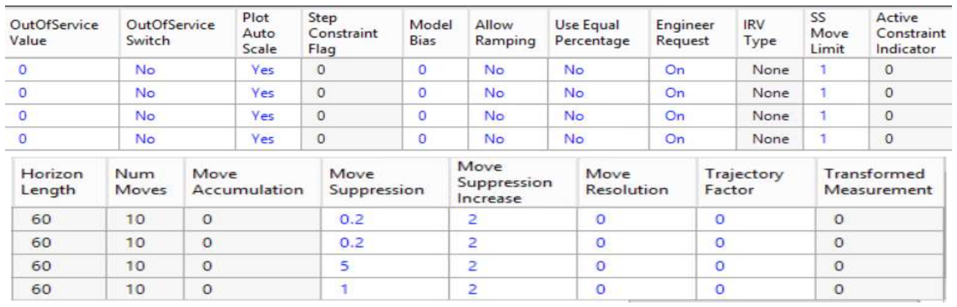
8.4.7 Aspen Nonlinear Controller: Task 3 – Configuring and Simulating the Dynamic Controller with Setpoint Changes

We follow Figure 8.97 to initialize the controller simulation. Figure 8.144a-b show the inputs for MVs and CVs, including the operating values and tuning values. We save the resulting simulation file as ***WS8.2_BaseCase_BDN.dmc3application.***



Inputs	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Current Move	Operator High Limit	Engineering High Limit	Validity High Limit	LP Cost
H2_C2	0.2	0.1	0.1	0.1	0.2	0	1	1	1	-29.1
C4_C2	0.35	0.1	0.1	0.1	0.35	0	1	1	1	-2.42
Temp	85	75	75	75	85	0	100	100	100	-0.088
C2_Partial_Pressure	6.52	5	5	5	6.52	0	9	9	9	-0.58

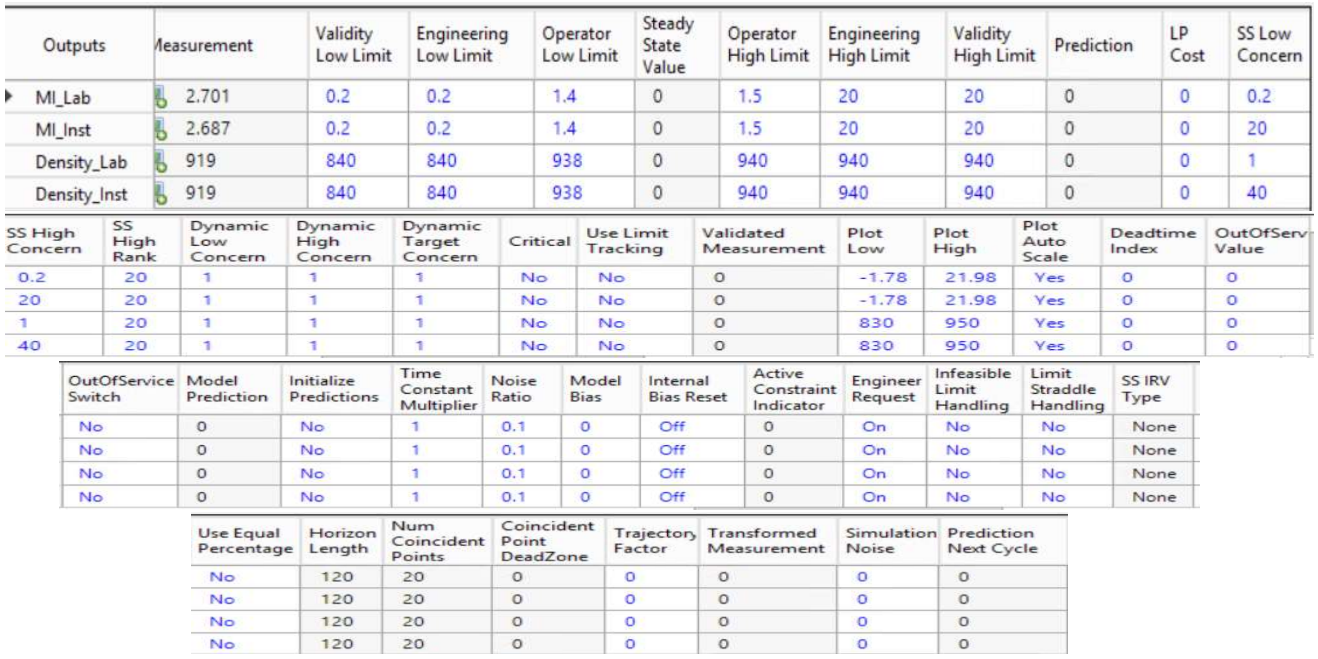
Dynamic Target Concern	Move Down Limit	Move Up Limit	Critical	Use Limit Tracking	Validated Measurement	Setpoint	Loop Status	Anti Windup Status	Reverse Acting	Plot Low	Plot High
1	1	1	No	No	0	0	On	Free	No	0.01	1.09
1	1	1	No	No	0	0	On	Free	No	0.01	1.09
1	1	1	No	No	0	0	On	Free	No	72.5	102.5
1	1	1	No	No	0	0	On	Free	No	4.6	9.4



OutOfService Value	OutOfService Switch	Plot Auto Scale	Step Constraint Flag	Model Bias	Allow Ramping	Use Equal Percentage	Engineer Request	IRV Type	SS Move Limit	Active Constraint Indicator
0	No	Yes	0	0	No	No	On	None	1	0
0	No	Yes	0	0	No	No	On	None	1	0
0	No	Yes	0	0	No	No	On	None	1	0
0	No	Yes	0	0	No	No	On	None	1	0

Horizon Length	Num Moves	Move Accumulation	Move Suppression	Move Suppression Increase	Move Resolution	Trajectory Factor	Transformed Measurement
60	10	0	0.2	2	0	0	0
60	10	0	0.2	2	0	0	0
60	10	0	5	2	0	0	0
60	10	0	1	2	0	0	0

Figure 8.144a Initial MV specifications of controller simulation



Outputs	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Operator High Limit	Engineering High Limit	Validity High Limit	Prediction	LP Cost	SS Low Concern
MI_Lab	2.701	0.2	0.2	1.4	0	1.5	20	20	0	0	0.2
MI_Inst	2.687	0.2	0.2	1.4	0	1.5	20	20	0	0	20
Density_Lab	919	840	840	938	0	940	940	940	0	0	1
Density_Inst	919	840	840	938	0	940	940	940	0	0	40

SS High Concern	SS High Rank	Dynamic Low Concern	Dynamic High Concern	Dynamic Target Concern	Critical	Use Limit Tracking	Validated Measurement	Plot Low	Plot High	Plot Auto Scale	Deadtime Index	OutOfServ Value
0.2	20	1	1	1	No	No	0	-1.78	21.98	Yes	0	0
20	20	1	1	1	No	No	0	-1.78	21.98	Yes	0	0
1	20	1	1	1	No	No	0	830	950	Yes	0	0
40	20	1	1	1	No	No	0	830	950	Yes	0	0

OutOfService Switch	Model Prediction	Initialize Predictions	Time Constant Multiplier	Noise Ratio	Model Bias	Internal Bias Reset	Active Constraint Indicator	Engineer Request	Infeasible Limit Handling	Limit Straddle Handling	SS IRV Type
No	0	No	1	0.1	0	Off	0	On	No	No	None
No	0	No	1	0.1	0	Off	0	On	No	No	None
No	0	No	1	0.1	0	Off	0	On	No	No	None
No	0	No	1	0.1	0	Off	0	On	No	No	None

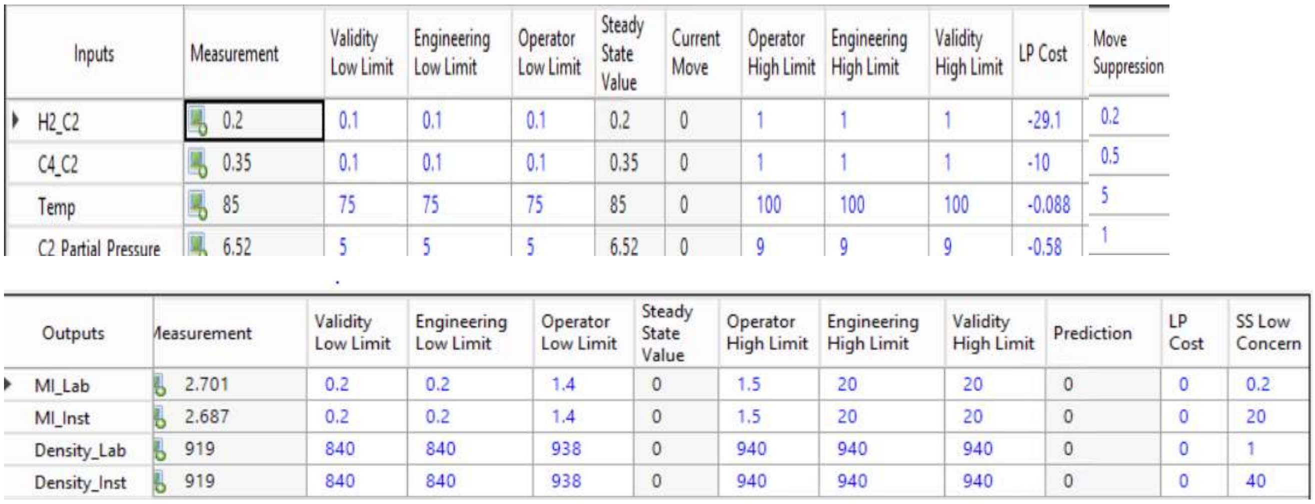
Use Equal Percentage	Horizon Length	Num Coincident Points	Coincident Point DeadZone	Trajectory Factor	Transformed Measurement	Simulation Noise	Prediction Next Cycle
No	120	20	0	0	0	0	0
No	120	20	0	0	0	0	0
No	120	20	0	0	0	0	0
No	120	20	0	0	0	0	0

Figure 8.144b Initial CV specifications of controller simulation

We wish to simulate the transition control of CV values of MI_Lab and MI_Inst from 2.7 to 1.5, while keeping both Density_Lab and Density_Inst between a lower operating limit of 938 kg/m^3 and an upper operating limit of 940 kg/m^3. Based on Figure 8.134 and Table 8.6, we expect the following changes to the MVs: C2_H2 and C4_C2 values to increase toward their upper operating limit, and temp and C2_Partial_Pressure values remain essentially unchanged.

We lower the initial move suppression of both MVs, C2_H2 and C4_C2, from 1 to 0.2, to speed up the increase of both MVs. We also increase the initial move suppression of MV, Temp, from 1 to 5, to slow down the change in Temp.

Figure 8.145 shows the changes of our MV and CV specifications.

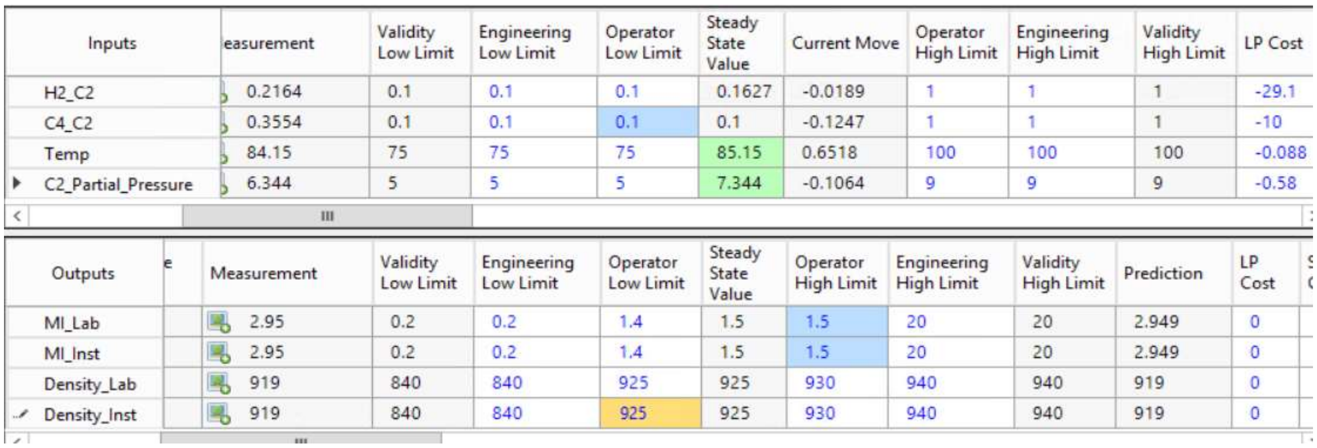


Inputs	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Current Move	Operator High Limit	Engineering High Limit	Validity High Limit	LP Cost	Move Suppression
H2_C2	0.2	0.1	0.1	0.1	0.2	0	1	1	1	-29.1	0.2
C4_C2	0.35	0.1	0.1	0.1	0.35	0	1	1	1	-10	0.5
Temp	85	75	75	75	85	0	100	100	100	-0.088	5
C2 Partial Pressure	6.52	5	5	5	6.52	0	9	9	9	-0.58	1

Outputs	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Operator High Limit	Engineering High Limit	Validity High Limit	Prediction	LP Cost	SS Low Concern
MI_Lab	2.701	0.2	0.2	1.4	0	1.5	20	20	0	0	0.2
MI_Inst	2.687	0.2	0.2	1.4	0	1.5	20	20	0	0	20
Density_Lab	919	840	840	938	0	940	940	940	0	0	1
Density_Inst	919	840	840	938	0	940	940	940	0	0	40

Figure 8.145 Changes to selected MV and CV tuning parameters for
MI and Density transition control

Figure 8.146 shows the results after CV stead-state values reach their operating limits, that is, at a MI-Lab value of 1.5, and a Density-Lab value of 925 kg/m^3. We note that during the simulation, the controller runs in a true closed-loop fashion, with measurement data received as follows: (1) For MVs, the setpoint calculated by the move plan is transferred to the measurement value; and (2) For CVs, the prediction for the next cycle is transferred to the measurement value. Therefore, the measurements of all variables do not become stale.



Inputs	easurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Current Move	Operator High Limit	Engineering High Limit	Validity High Limit	LP Cost
H2_C2	0.2164	0.1	0.1	0.1	0.1627	-0.0189	1	1	1	-29.1
C4_C2	0.3554	0.1	0.1	0.1	0.1	-0.1247	1	1	1	-10
Temp	84.15	75	75	75	85.15	0.6518	100	100	100	-0.088
C2_Partial_Pressure	6.344	5	5	5	7.344	-0.1064	9	9	9	-0.58

Outputs	e	Measurement	Validity Low Limit	Engineering Low Limit	Operator Low Limit	Steady State Value	Operator High Limit	Engineering High Limit	Validity High Limit	Prediction	LP Cost
MI_Lab		2.95	0.2	0.2	1.4	1.5	1.5	20	20	2.949	0
MI_Inst		2.95	0.2	0.2	1.4	1.5	1.5	20	20	2.949	0
Density_Lab		919	840	840	925	925	930	940	940	919	0
Density_Inst		919	840	840	925	925	930	940	940	919	0

Figure 9.146 Results after CV steady-state values reach their operating limits

The top plot in Figure 8.147 shows that the closed-loop MI_Lab prediction (in red) continues to decrease downward and approach the calculated steady-state target (the upper operating limit) of 1.5 (in green); the bottom plot in Figure 8.147 shows that the closed-loop Density_Lab prediction (in red) continues to increase upward and approach the calculated steady-state target (the lower operating limit) of 925 kg/m^3. We will not show the remaining simulation cycle in which the closed-loop prediction values match the calculated steady-state targets.

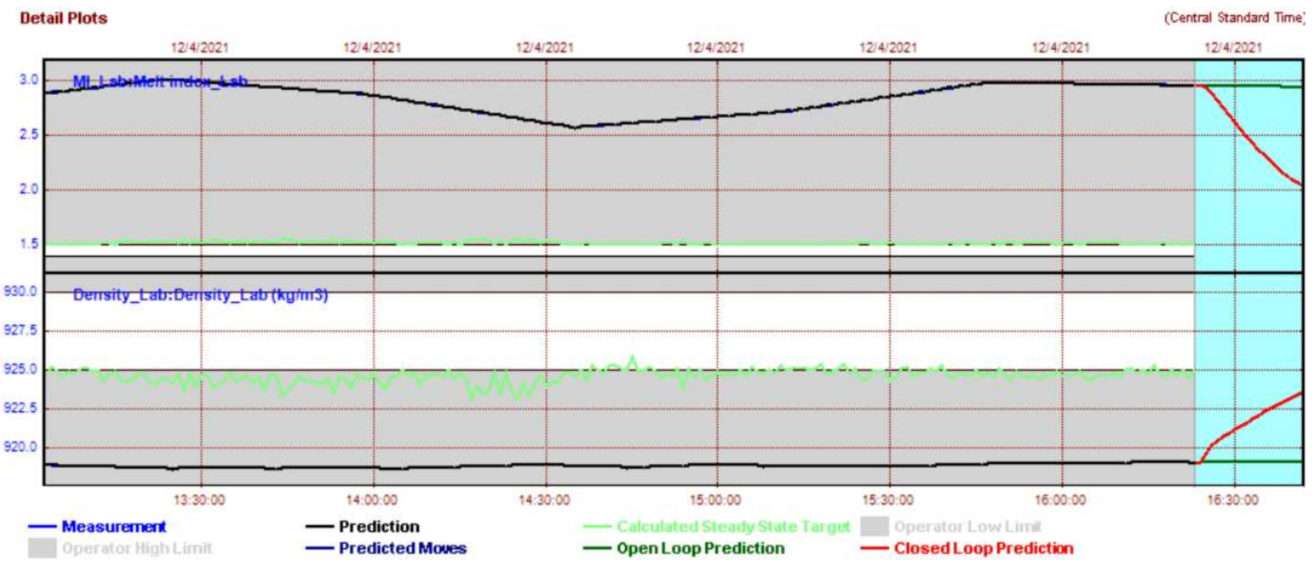


Figure 8.147 Controller simulation plot showing the closed-loop precitions (in red) approach the claculated steady-state targets (in greed) of CVs.

This concludes the current workshop. We save the project as ***PP Density and MI Control_Final***.

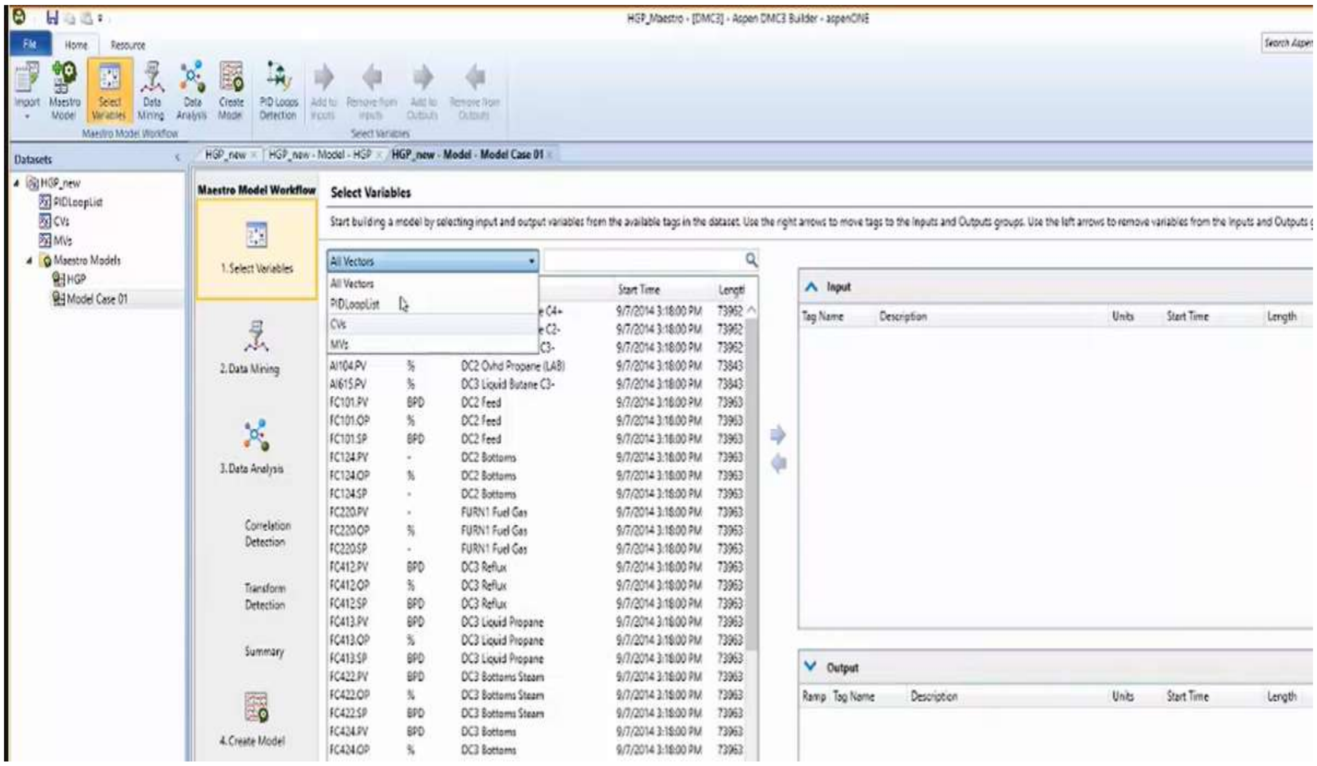
8.5 Aspen Maestro for Automating the Model-Building Workflow

Aspen DMC3 V12 has added a powerful tool to automate the model-building process for model-predictive control. We recommend the reader to take time to view the on-demand webinar by Kalafatis and Reis [21] to see how embedding AI into DMC3 can greatly speed up the model-building process and improve the model- prediction accuracy. However, we emphasize that to truly understand the concepts and know-hows behind each step of this automated model-building process, the reader should first become familiar with the fundamentals and practice we cover in Sections 8.1 and 8.2.

Figures 8.148a-d [21] show the screen image of the four steps of the automated model-building process using Aspen Maestro which is an integrated part of DMC3 V12 and later versions. Note that we have purposed removed a part of the step-response curves on the right side of the figure to clearly show the Aspen Maestro workflow steps.

Step 1. <u>Select variables</u>: Figure 8.148a; follow Sections 8.2.2 to 8.2.4. Note the new “Maestro Model” button next to the “Select Variables” button in DMC3 V12 on the left of top tool buttons.

Step 2. <u>Data mining</u>: Figure 8.148b; automate Section 8.2.5, data slicing - this step explores data slices used to create the model. Select one of the four available options in the sensitivity scale (PID, low, medium and high) and view the data slicing results. A high sensitivity scale tends to include the best independent moves available to each input or manipulated variable. Note the new “Data Mining” button in DMC3 V12 on the left of top tool buttons.

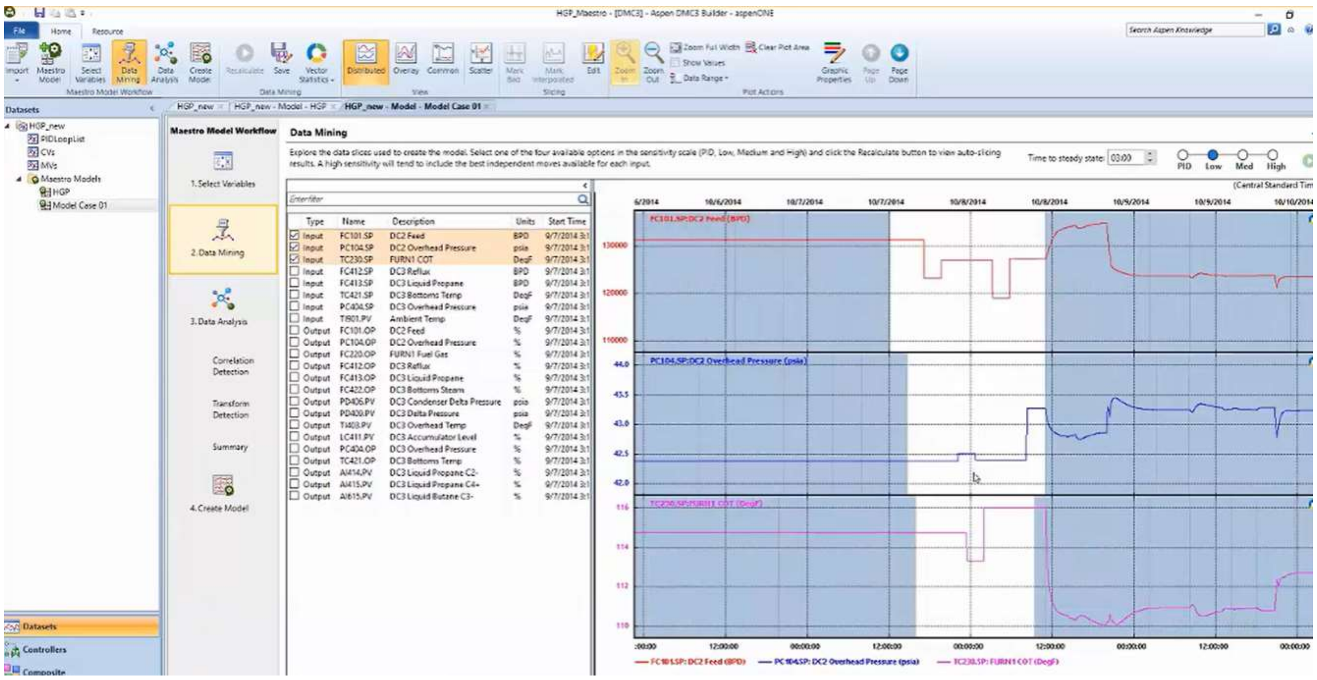


Select Variables

Start building a model by selecting input and output variables from the available tags in the dataset. Use the right arrows to move tags to the Inputs and Outputs groups. Use the left arrows to remove variables from the Inputs and Outputs

			Start Time	Length
		C4+	9/7/2014 3:18:00 PM	73962
		C2-	9/7/2014 3:18:00 PM	73962
		C3-	9/7/2014 3:18:00 PM	73962
AI104.PV	%	DC2 Ovhd Propane (LAB)	9/7/2014 3:18:00 PM	73843
AI615.PV	%	DC3 Liquid Butane C3-	9/7/2014 3:18:00 PM	73843
FC101.PV	BPD	DC2 Feed	9/7/2014 3:18:00 PM	73963
FC101.OP	%	DC2 Feed	9/7/2014 3:18:00 PM	73963
FC101.SP	BPD	DC2 Feed	9/7/2014 3:18:00 PM	73963
FC124.PV	-	DC2 Bottoms	9/7/2014 3:18:00 PM	73963
FC124.OP	%	DC2 Bottoms	9/7/2014 3:18:00 PM	73963
FC124.SP	-	DC2 Bottoms	9/7/2014 3:18:00 PM	73963
FC220.PV	-	FURN1 Fuel Gas	9/7/2014 3:18:00 PM	73963
FC220.OP	%	FURN1 Fuel Gas	9/7/2014 3:18:00 PM	73963
FC220.SP	-	FURN1 Fuel Gas	9/7/2014 3:18:00 PM	73963
FC412.PV	BPD	DC3 Reflux	9/7/2014 3:18:00 PM	73963
FC412.OP	%	DC3 Reflux	9/7/2014 3:18:00 PM	73963
FC412.SP	BPD	DC3 Reflux	9/7/2014 3:18:00 PM	73963
FC413.PV	BPD	DC3 Liquid Propane	9/7/2014 3:18:00 PM	73963
FC413.OP	%	DC3 Liquid Propane	9/7/2014 3:18:00 PM	73963
FC413.SP	BPD	DC3 Liquid Propane	9/7/2014 3:18:00 PM	73963
FC422.PV	BPD	DC3 Bottoms Steam	9/7/2014 3:18:00 PM	73963
FC422.OP	%	DC3 Bottoms Steam	9/7/2014 3:18:00 PM	73963
FC422.SP	BPD	DC3 Bottoms Steam	9/7/2014 3:18:00 PM	73963
FC424.PV	BPD	DC3 Bottoms	9/7/2014 3:18:00 PM	73963
FC424.OP	%	DC3 Bottoms	9/7/2014 3:18:00 PM	73963

Figure 9.148a Step 1 of Aspen Maestro model workflow for DMC3 – Select variables



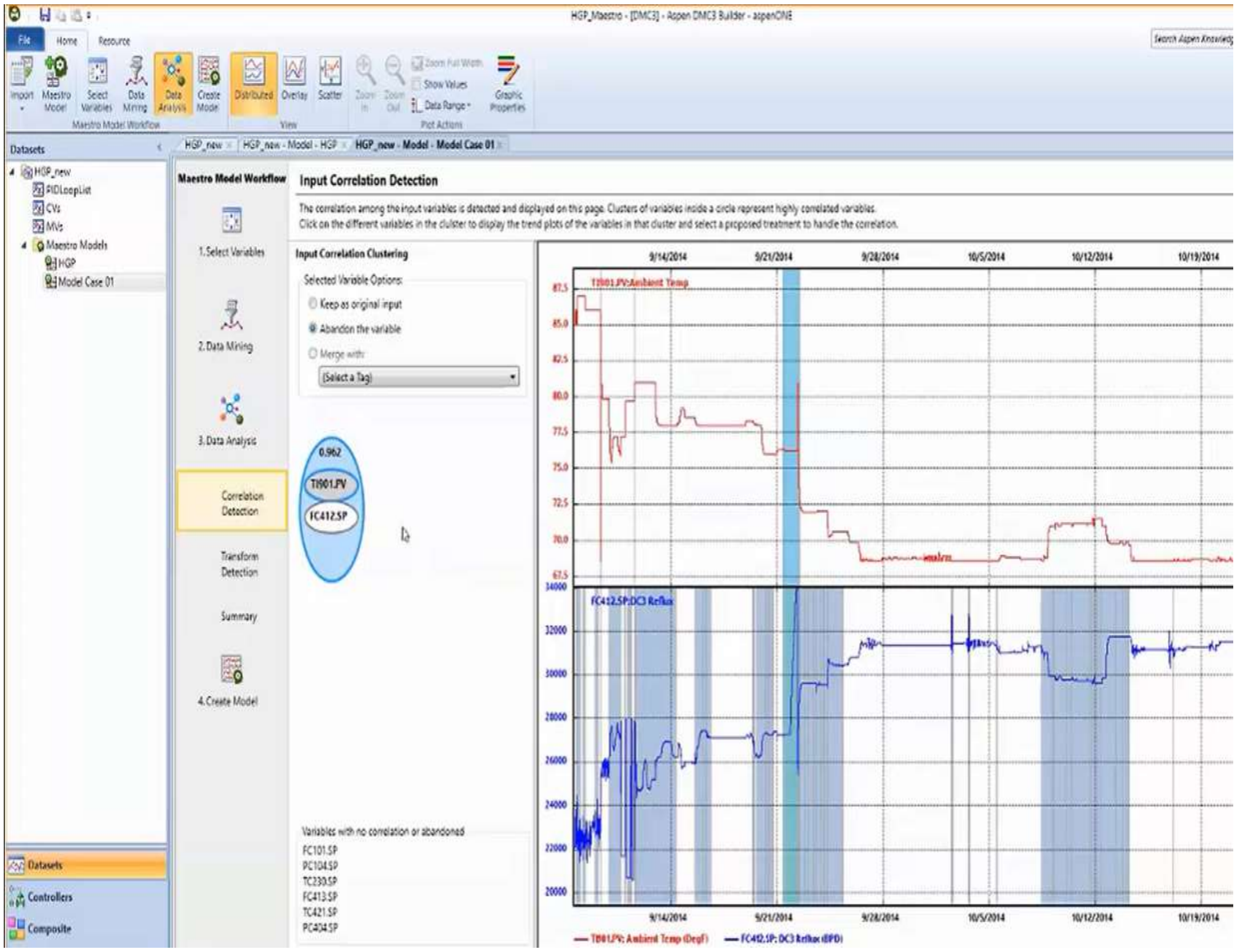
Data Mining

Explore the data slices used to create the model. Select one of the four available options in the sensitivity scale (PID, Low, Medium and High) and click the Recalculate button to view auto-slicing results. A high sensitivity will tend to include the best independent moves available for each input.

	Type	Name	Description	Units	Start Time
☑	Input	FC101.SP	DC2 Feed	BPD	9/7/2014 3:1
☑	Input	PC104.SP	DC2 Overhead Pressure	psia	9/7/2014 3:1
☑	Input	TC230.SP	FURN1 COT	DegF	9/7/2014 3:1
☐	Input	FC412.SP	DC3 Reflux	BPD	9/7/2014 3:1
☐	Input	FC413.SP	DC3 Liquid Propane	BPD	9/7/2014 3:1
☐	Input	TC421.SP	DC3 Bottoms Temp	DegF	9/7/2014 3:1
☐	Input	PC404.SP	DC3 Overhead Pressure	psia	9/7/2014 3:1
☐	Input	TI801.PV	Ambient Temp	DegF	9/7/2014 3:1
☐	Output	FC101.OP	DC2 Feed	%	9/7/2014 3:1
☐	Output	PC104.OP	DC2 Overhead Pressure	%	9/7/2014 3:1
☐	Output	FC220.OP	FURN1 Fuel Gas	%	9/7/2014 3:1
☐	Output	FC412.OP	DC3 Reflux	%	9/7/2014 3:1
☐	Output	FC413.OP	DC3 Liquid Propane	%	9/7/2014 3:1
☐	Output	FC422.OP	DC3 Bottoms Steam	%	9/7/2014 3:1
☐	Output	PD406.PV	DC3 Condenser Delta Pressure	psia	9/7/2014 3:1
☐	Output	PD400.PV	DC3 Delta Pressure	psia	9/7/2014 3:1
☐	Output	TI403.PV	DC3 Overhead Temp	DegF	9/7/2014 3:1
☐	Output	LC411.PV	DC3 Accumulator Level	%	9/7/2014 3:1
☐	Output	PC404.OP	DC3 Overhead Pressure	%	9/7/2014 3:1
☐	Output	TC421.OP	DC3 Bottoms Temp	%	9/7/2014 3:1
☐	Output	AI414.PV	DC3 Liquid Propane C2-	%	9/7/2014 3:1
☐	Output	AI415.PV	DC3 Liquid Propane C4+	%	9/7/2014 3:1
☐	Output	AI615.PV	DC3 Liquid Butane C3-	%	9/7/2014 3:1

Figure 9.148b Step 2 pf Aspen Maestro model workflow for DMC3 –Data mining.

Step 3a. Data analysis – Correlation detection Figure 8.148c – input correlation detection. This step quantifies how much an input variable or a MV correlates with another input variable. Clusters of input variables inside a circle represent highly correlated variables with correlation coefficients close to -1.0 to 1.0. A correlation of -1.0 shows a perfect negative correlation, while a correlation of 1.0 shows a perfect positive correlation. See Section 8.2.8 and Figure 8.57. The plot also identifies input variables with no or minimum correlation.



Input Correlation Detection

The correlation among the input variables is detected and displayed on this page. Clusters of variables inside a circle represent highly correlated variables.
Click on the different variables in the cluster to display the trend plots of the variables in that cluster and select a proposed treatment to handle the correlation.

Input Correlation Clustering

Selected Variable Options:

( ) Keep as original input
(●) Abandon the variable
( ) Merge with: (Select a Tag)

Variables with no correlation or abandoned

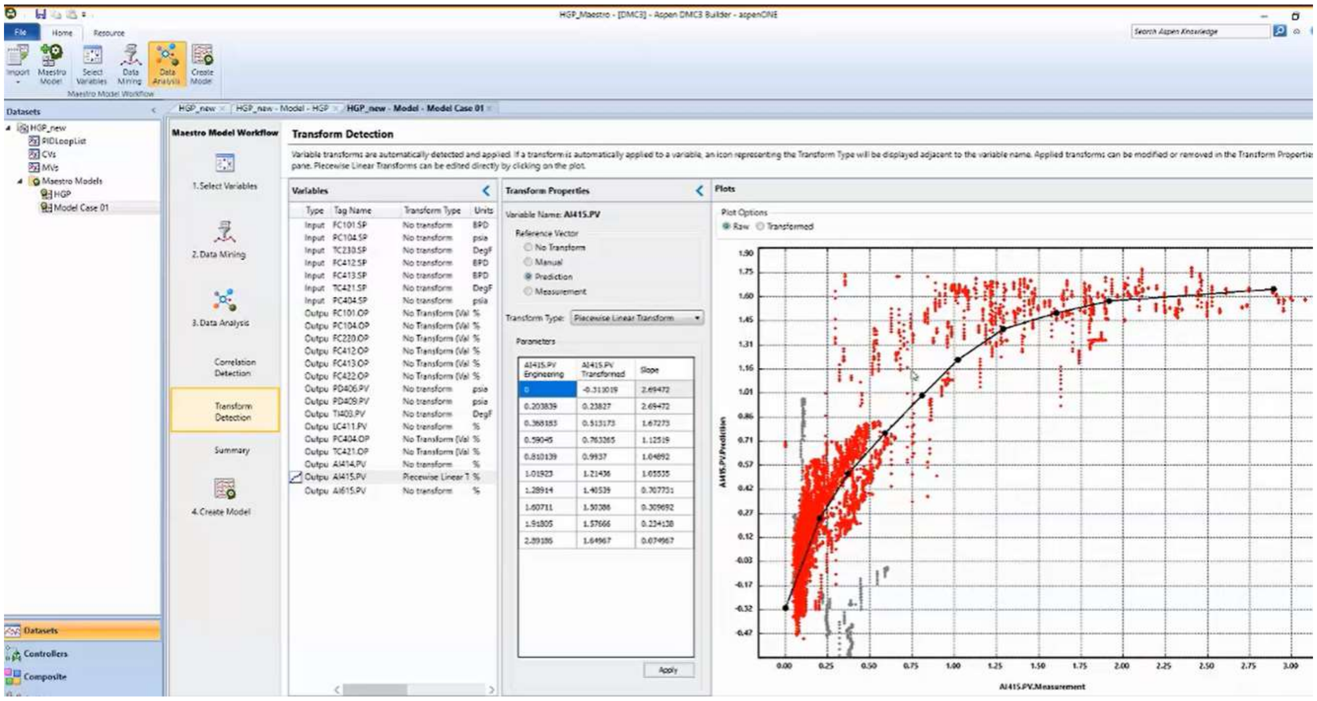
FC101.SP
PC104.SP
TC230.SP
FC413.SP
TC421.SP
PC404.SP

Figure 8.148c Step 3a pf Aspen Maestro model workflow for DMC3 –Input correlation detection.

Step 3b. Transform detection. Figure 8.148d. This figure shows an example of transforming dependent variable measurements into *a piecewise linear representation*, that is, correlating the measurement data into connected multiple straight-line segments with different slopes. Aspen Maestro automates the development of transforms in DMC3 to deal with nonlinear dependent variable measurements and configures transforms to re-scale the data. For example, Aspen Maestro includes the well-known *linear valve output transform* and *parabolic valve output transform* introduced in *Perry's Chemical Engineers' Handbook*, 5th edition, that relates the fraction of maximum flow rate, Q, to fraction of valve stem travel, L, with a valve transform parameter α (0 < α ≤ 1) according to Eqs. (8.53) and (8.54):

$$Q = \frac{L}{\sqrt{\alpha + (1-\alpha)L^2}} \quad \text{(linear)} \qquad (8.53)$$

$$Q = \frac{L^2}{\sqrt{\alpha + (1-\alpha)L^4}} \text{ (parabolic)} \tag{8.54}$$



Transform Detection

Variable transforms are automatically detected and applied. If a transform is automatically applied to a variable, an icon representing the Transform Type will be displayed adjacent to the variable name. Applied transforms can be modified or removed in the Transform Properties pane. Piecewise Linear Transforms can be edited directly by clicking on the plot.

Variables

Type	Tag Name	Transform Type	Units
Input	FC101.SP	No transform	BPD
Input	PC104.SP	No transform	psia
Input	TC230.SP	No transform	DegF
Input	FC412.SP	No transform	BPD
Input	FC413.SP	No transform	BPD
Input	TC421.SP	No transform	DegF
Input	PC404.SP	No transform	psia
Outpu	FC101.OP	No Transform (Val	%
Outpu	PC104.OP	No Transform (Val	%
Outpu	FC220.OP	No Transform (Val	%
Outpu	FC412.OP	No Transform (Val	%
Outpu	FC413.OP	No Transform (Val	%
Outpu	FC422.OP	No Transform (Val	%
Outpu	PD406.PV	No transform	psia
Outpu	PD409.PV	No transform	psia
Outpu	TI403.PV	No transform	DegF
Outpu	LC411.PV	No transform	%
Outpu	PC404.OP	No Transform (Val	%
Outpu	TC421.OP	No Transform (Val	%
Outpu	AI414.PV	No transform	%
Outpu	AI415.PV	Piecewise Linear T	%
Outpu	AI615.PV	No transform	%

Transform Properties

Variable Name: **AI415.PV**

Reference Vector: No Transform / Manual / [✓ Prediction] / Measurement

Transform Type: Piecewise Linear Transform

Parameters

AI415.PV Engineering	AI415.PV Transformed	Slope
0	-0.311019	2.69472
0.203839	0.23827	2.69472
0.368183	0.513173	1.67273
0.59045	0.763265	1.12519
0.810139	0.9937	1.04892
1.01923	1.21436	1.05535
1.28914	1.40539	0.707731
1.60711	1.50386	0.309692
1.91805	1.57666	0.234138
2.89186	1.64967	0.074967

Figure 8.148d Step 3b of Aspen Maestro model workflow for DMC3 –Transform detection.

Figure 8.149 shows a plot of valve output transforms, displaying both linear and parabolic valves, Eqs. (8.53) and (8.54).

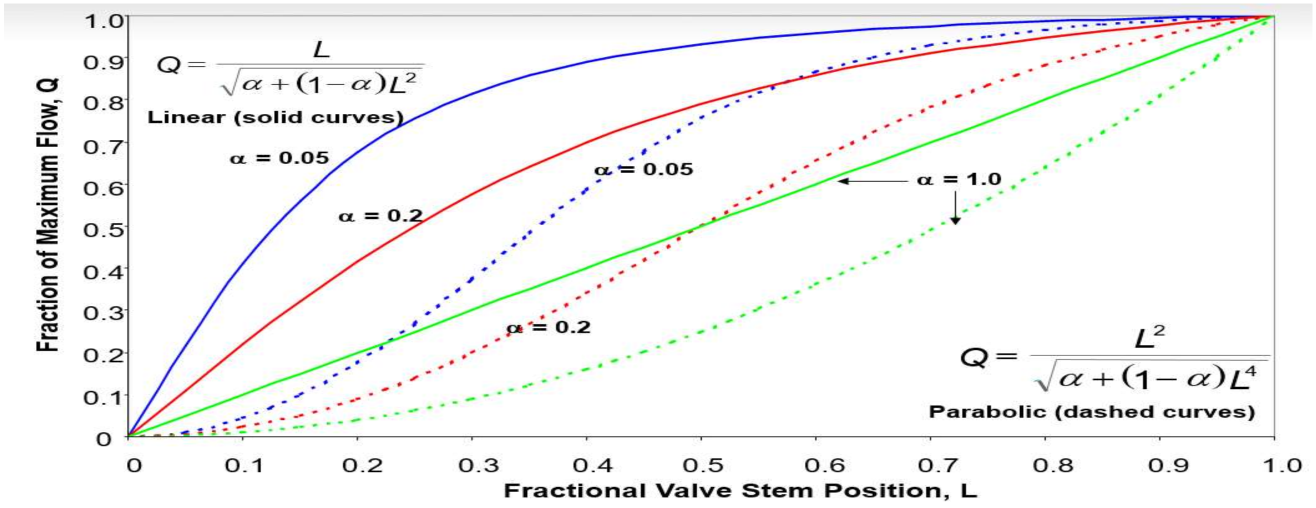
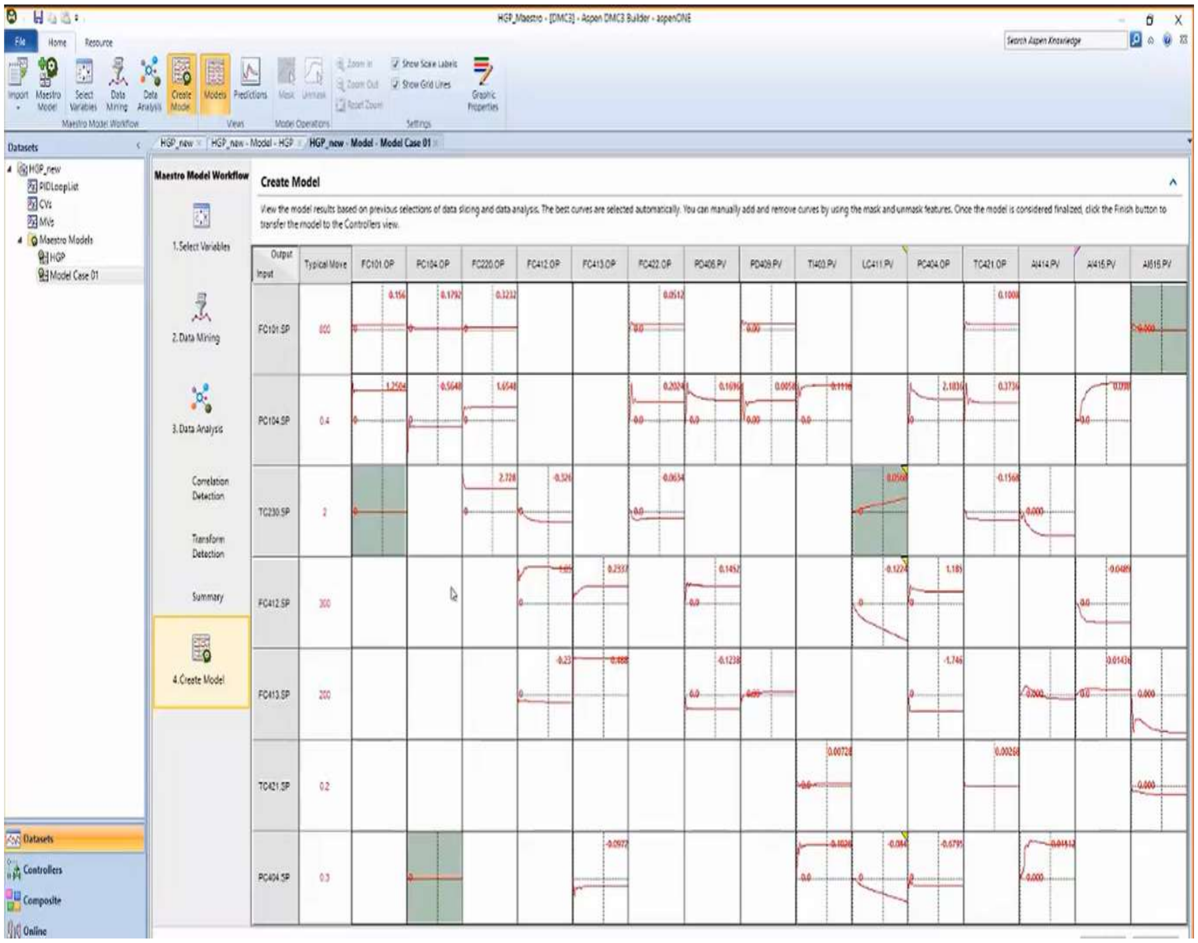


Figure 8.149 An illustration of linear and parabolic valve output transformations included in Aspen Maestro. Used with permission from Aspen Tehnology. Inc.

Step 4. Create Model. Figure 8.150 shows the model results based on previous selections of data mining (data slicing) and data analysis. Aspen Maestro automatically selects the best model curves to generate the final model, and we can transfer the resulting model to the controller view.



View the model results based on previous selections of data slicing and data analysis. The best curves are selected automatically. You can manually add and remove curves by using the mask and unmask features. Once the model is considered finalized, click the Finish button to transfer the model to the Controllers view.

Figure 8.150 Step 4 of Aspen Maestro model workflow for DMC3 –create model.

This concludes our illustration of Aspen Maestro for automating the model-building process. In Chapter 10, we will further illustrate embedding AI into DMC3 by using deep learning neural networks (such as LSTM (long short-term memory) recurrent networks (see Section 10.4.2b), and GRU (gated recurrent networks) (see Section 10.4.2c) to develop soft sensors or IQ inferential for process and product quality variables that are not measured frequently.

8.6 Conclusion

In conclusion, this chapter systematically unpacks the complexities and potentials of advanced process control (APC) and model-predictive control (MPC) within polyolefin manufacturing. Through the careful delineation of APC concepts, the exploration of dynamic and nonlinear control models, and the integration of AI technologies, it lays a foundational and advanced understanding essential for both newcomers and seasoned experts in the field. The practical workshops and discussions provided not only illuminate the path toward developing sophisticated control systems but also highlight the significant impact such technologies can have on improving process efficiencies, optimization, and sustainability in the chemical processing industry.For future work stemming from this study, it would be advantageous to explore the integration of first principles with model predictive control to enhance the robustness and efficiency of the system [24]. Ultimately, this chapter advocates for the embracement of APC and MPC as transformative tools in the polyolefin production process, promising a future of enhanced industrial performance and innovation.

This chapter is published with Wiley publication in the book *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing by Liu & Sharma.* [26-38]

8.7 Bibliography

1. Camacho, E. F.; Bordons, C. (2007). *Model-predictive control*. 2nd edition, Springer-Verlag, London, United Kingdom.

2. Lahiri, S. K. (2017). *Multivariable Predictive Control: Applications in Industry*. Wiley, New York.

3. Stephanopoulos, G. (1983). *Chemical Process Control: An Introduction to Theory and Practice*, Prentice-Hall, Englewood Cliffs, New Jersey.

4. Cutler, C. R.; Ramaker, B. L. (1979). Dynamic Matrix Control – A Computer Control Algorithm. *AIChE National Meeting*, Houston, Texas.

5. Liu, Y. A.; Chang, A. F.; Pashikanti, K. (2018). *Petroleum Refinery Process Modeling: Integrated Optimization Tools and Applications*. Wiley-VCH, Weinheim, Germany.

6. Sadeghbeigi, R. (2000). *Fluid Catalytic Cracking Handbook: Design, Operation and Troubleshooting of FCC Facilities.* 2nd edition, Gulf Publishing Company, Houston, TX.

7. Aspen Technology, Inc. (2016), Training course APC125, "Introduction to Aspen DMC3 Builder: Modeling and Building Controllers for Industrial Processes".

8. Bristol, E. (1966)." On a Measure of Interaction for Multivariable Process Control", *IEEE Trans. Autom. Control*. **AC-11**, 133.

9. McAvoy, T. J. (1983). *Interaction Analysis*. Instrument Society of America, Research Triangle Park, NC.

10. Smith, C. A.; Corripio, A. B. (1997). *Principles and Practice of Automatic Process Control*. 2nd edition, Wiley, New York, NY.

11. Zhang, Q.; Harmse, M. J.; Rasmussen, K.; McIntyre, B. (2007). "Methods and Articles for Detecting, Verifying, and Repairing Collinearity in a Model or Subsets of a Model". U. S. patent no. 7,231,264 B2, assigned to Aspen Technology, Cambridge, MA.

12. Stanley, G., "Exponential Filter", Greg Stanley and Associates, (https://gregstanleyandassociates.com/whitepapers/FaultDiagnosis/Filtering/Exponential-Filter/exponential-filter.htm), accessed December 16, 2021.

13. Becker, A., "Kalman Filter Tutorial, https://www.kalmanfilter.net/default.aspx, accessed December 16, 2021.

14. Turner, P.; Guiver, J.; Lines, B. (2003). Introducing the Bounded Derivative Network for Commercial Transition Control. *Proceedings of American Control Conference*, Denver, Colorado, June 4-6, p. 5400.

15. Turner, P.; Guiver, J. (2005). Introducing the Bounded Derivative Network - Superceding the Application of Neural Networks in Control. *Intern. J. Control,* **15**, 407.

16. Donat, J. S.; Bhat, N.; McAvoy (1991). Neural-Net Based Model-Predictive Control. *Intern. J. Control*, **54**, 1453.

17. Baughman, D. R.; Liu, Y. A. (1995). Chapter 5, "Forecasting, Modeling and Control" in *Neural Networks in Bioprocessing and Chemical Engineering*, Academic Press, Inc., San Diego, CA.

18. Bindlish, R. (2020). Nonlinear Model-predictive control of an Industrial Process with Steady-State Gain Inversion. *Comput. Chem. Eng.*, **135**, 106739.

19. Bausa, J. (2007). Model-Based Operation of Polymer Processes – What Has to Be Done? *Macromol Symp.*, **259**, 42.

20. Jeong, B.-G.; Yoo, Y.-K.; Rhee, H.-K. (2001). Nonlinear Model-predictive control Using a Wiener Model for a Continuous Methyl Methacrylate Polymerization Reactor. *Ind. Eng. Chem. Res.*, **40**, 5968.

21. Kalafatis, A.; Reis, L. Revolutionize APC Model Building and Make More Accurate Predictions with Embedded AI. AspenTech on-Demand Webinar, December 10, 2020. https://www.aspentech.com/en/resources/on-demand-webinars/revolutionize-apc-model-building-and-make-more-accurate-predictions-with-embedded-ai, accessed May 22, 2022.

22. Kalafatis, A.; Embedding AI in APC – Current Capabilities, Direction and Roadmap. AspenTech Optimize Conference 21 (Virtual)- The Future Starts with Industrial AI. May 21, 2021.

23. Sharma, N., & Liu, Y. A. (2019). 110th anniversary: an effective methodology for kinetic parameter estimation for modeling commercial polyolefin processes from plant data using efficient simulation software tools. *Industrial & Engineering Chemistry Research*, *58*(31), 14209-14226. https://doi.org/10.1021/acs.iecr.9b02277

24. Sharma, N., & Liu, Y. A. (2022). A hybrid science-guided machine learning approach for modeling chemical processes: A review. *AIChE Journal*, *68*(5), e17609. https://doi.org/10.1002/aic.17609

25. Nguyen, X. D. J., Sharma, N., Liu, Y. A., Lee, Y., & McDowell, C. C. (2023). Analyzing the occurrence of foaming in batch fermentation processes using multiway partial least square approaches. *AIChE Journal*, *69*(12), e18250. https://doi.org/10.1002/aic.18250

26. Liu, Y. A., & Sharma, N. (2023). *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing*. Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831

27. 28. Liu, Y. A., & Sharma, N. (2023). Introduction to Integrated Process Modeling, Advanced Control, and Data Analytics in Optimizing Polyolefin Manufacturing. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing* (Chapter 1, pp. 1-40). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch1

29. Liu, Y. A., & Sharma, N. (2023). Selection of Property Methods and Estimation of Physical Properties for Polymer Process Modeling. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing* (Chapter 2, pp. 41-86). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch2

30. Liu, Y. A., & Sharma, N. (2023). Reactor Modeling, Convergence Tips, and Data-Fit Tool. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing* (Chapter 3, pp. 87-114). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch3

31. Liu, Y. A., & Sharma, N. (2023). Free Radical Polymerizations: LDPE and EVA. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing* (Chapter 4, pp. 115-162). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch4

32. Liu, Y. A., & Sharma, N. (2023). Ziegler–Natta Polymerization: HDPE , PP , LLDPE, and EPDM. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 5, pp. 163-265). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch5

33. Liu, Y. A., & Sharma, N. (2023). Free Radical and Ionic Polymerizations: PS and SBS Rubber. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 6, pp. 267-319). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch6

34. Liu, Y. A., & Sharma, N. (2023). Improved Polymer Process Operability and Control Through Steady-State and Dynamic Simulation Models. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 7, pp. 321-379). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch7

35. Liu, Y. A., & Sharma, N. (2023). Model-Predictive Control of Polyolefin Processes. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 8, pp. 381-476). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch8

36. Liu, Y. A., & Sharma, N. .2023. Application of Multivariate Statistics to Optimizing Polyolefin Manufacturing. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing* (Chapter 9, pp. 477-531). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch9

37. Liu, Y. A., & Sharma, N. (2023). Applications of Machine Learning to Optimizing Polyolefin Manufacturing. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 10, pp. 553-650). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch10.

38. Liu, Y. A., & Sharma, N. (2023). A Hybrid Science-Guided Machine Learning Approach for Modeling Chemical and Polymer Processes. In *Integrated Process Modeling, Advanced Control and Data Analytics for Optimizing Polyolefin Manufacturing.* (Chapter 11, pp. 651-698). Wiley-VCH GmbH. https://doi.org/10.1002/9783527843831.ch11

39. Sharma, N. and Liu, Y., 2019, November. Polyolefin Process Modeling and Monitoring. In *2019 AIChE Annual Meeting*. AIChE.

40. Sharma, N. and Liu, Y., 2020, November. Polyolefin Process Improvement Using Machine Learning. In *2020 Virtual AIChE Annual Meeting*. AIChE

41. Sharma, N., 2022, November. Polyolefin Property Estimation using Process Modeling and Machine Learning in Industry. In *2022 AIChE Annual Meeting*. AIChE.

42. Lee, W. and Weekman Jr, V.W., 1976. Advanced control practice in the chemical process industry: A view from industry. *AIChE Journal*, *22*(1), pp.27-38.

43. Alford, J.S., 2006. Bioprocess control: Advances and challenges. *Computers & Chemical Engineering*, *30*(10-12), pp.1464-1475.

44. Sharma, N., SANDESH, D.S., Chowdhury, R., Pant, A. and Mediratta, G., SHPP Global Technologies BV, 2023. *Process and apparatus for precipitation of poly (phenylene ether)*. U.S. Patent 11,802,184.