José David Rojas, Orlando Arrieta, Ramon Vilanova

Industrial PID Controller Tuning

with a multi-objective framework using MATLAB®

January 30, 2021

Springer

Acknowledgements

This work wouldn't be possible without the hard work of many students along many years. The authors want to acknowledge the following students which were part of our research lab and made the subject of optimization and PID control part of their academic career:

- Macarena Céspedes
- Mónica P. Contreras-Leiva
- Joaquín Cordero
- Carlos Gamboa
- Felipe Moya
- Gustavo Montoya
- Francisco Rivas
- Sergio Rodríguez Rojas
- Rosario Ruiz Hernández
- Felipe Sáenz Cortés
- Karen Valverde
- Diana Valverde-Mendez

Also, a very special thanks goes to our mentor, professor Víctor M. Alfaro which showed us to love the strange art of industrial PID control.

Contents

Abbreviations

1DoF One Degree of Freedom.
2DoF Two Degrees of Freedom.
CAD Computer-Aided Design.
CSTH Continuously Stirred Tank Heater.
CSTR Continuously Stirred Tank Reactor.
DPPDT Dual-Pole-Plus-Dead-Time.
ENNC Enhanced Normalized Normal Constraint.
FOPTD First-Order Plus Time Delay.
IAE Integral of the Absolute Value of the Error.
ISE Integral of the Square Error.
ITAE Integral Time Absolute Error.
MIMO Multiple-Input Multiple-Output.
MOO Multiobjective Optimization.
MOOP Multiobjective Optimization Problem.
NBI Normal Boundary Intersection.
NNC Normalized Normal Constraint.
ODSOPTD Overdamped Second-Order Plus Time Delay.
PD Proportional–Derivative.
PI Proportional–Integral.
PID Proportional–Integral–Derivative.
SISO Single-Input Single-Output.
TV Total Variation.
WS Weighted Sum.
WWTP Wastewater treatment plant.

Symbols

$C_r(s, \boldsymbol{\theta})$ Servo component of the controller.
$C_y(s, \boldsymbol{\theta})$ Regulator component of the controller.
K_p Proportional gain.
K Plant gain.
L Time delay.
M_s Maximum sensitivity.
$P(s)$ Plant transfer function.
T_d Derivative time.
T_i Integral time.
T Constant time.
α Filter factor of the derivative part of the controller.
β Weight to the reference signal in the proportional part of the two degrees of freedom controller.
$\boldsymbol{\theta}$ Controller parameters vector.
γ Weight to the reference signal in the derivative part of the two degrees of freedom controller.
$d_i(s)$ Input disturbance signal.
$d_o(s)$ Output disturbance signal.
$r(s)$ Setpoint.
$u(s)$ Control signal.
$y(s)$ Feedback signal.

Chapter 1
Industrial application examples

Abstract In this chapter, different examples are provided to apply the tool presented in this book. The MOOTuning software is used to analyze the temperature control in a Continuously Stirred Tank Heater in Section 8.1 using two a three cost functions. Also a Continuously Stirred Tank Reactor is considered for the control of the concentration of the product in an isothermal case in Section 8.2

1.1 Continuously Stirred Tank Heater

1.1.1 Description of the process

The control of a Continuously Stirred Tank Heater (CSTH) is a common task in industrial processes. In this section, the control of the temperature of the CSTH will be solved as a Multiobjective Optimization Problem (MOOP) using a Two Degrees of Freedom (2DoF) Proportional–Integral–Derivative (PID) controller. The diagram of the process is presented in Figure 8.1. A heat exchanger is installed inside the tank to heat the fluid. The flow rate inside the heat exchanger is controlled with a valve with input variable U_T and the liquid inside the heat exchanger enters with temperature T_{ci} and leaves with temperature T_{co}, the average temperature inside the heat exchanger is T_{ca}. The volume inside the tank is variable, the input flow rate is Q_i with temperature T_i. The output flow rate is Q with temperature T. The output flow rate is controlled with a valve with input variable U_L. The tank is covered with a jacket that prevents any heat loss to the atmosphere.

According to **?**, a possible model for this process is given by the following set of algebraic-differential equations:

- Tank mass balance:
$$A\frac{dH(t)}{dt} = Q_i(t) - Q(t),$$
where A is the transversal area of the tank and $H(t)$ is the liquid level.

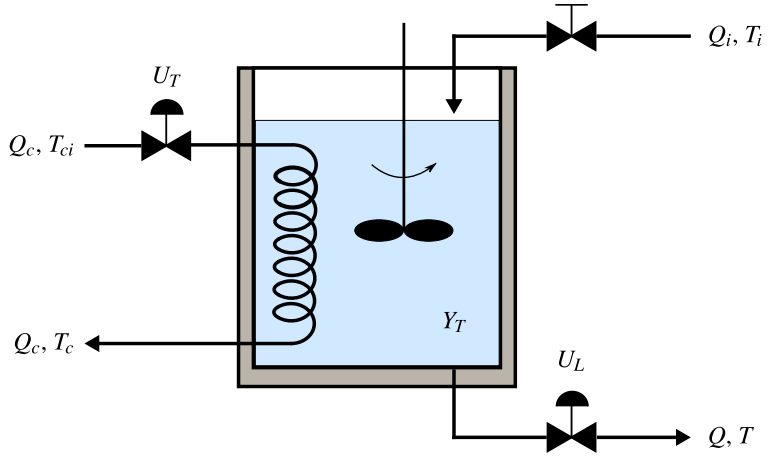


Fig. 1.1: Simplified diagram of a continuously stirred tank heater to be controlled.

- Tank energy balance:

$$\rho C_p A H(t) \frac{T(t)}{dt} = \rho C_p Q_i(t) \left(T_i(t) - T(t)\right) + W(t),$$

where C_p is the heat capacity of the fluid and $W(t)$ is the rate of heat transfer from the heat exchanger to the tank.

- Heat exchanger energy balance:

$$\rho_c C_{pc} V_c \frac{T_{ca}(t)}{dt} = \rho_c C_{pc} Q_c(t) \left(T_{ci}(t) - T_{co}(t)\right) - W(t),$$

where ρ_c is the density of the fluid inside the heat exchanger, C_{pc} is the heat capacity of the fluid inside the heat exchanger and V_c is the volume of the heat exchanger.

- Heat transfer between the heat exchanger and the fluid in the tank:

$$W(t) = U A_c \left(T_{ca}(t) - T(t)\right),$$

where U is overall heat-transfer coefficient, A_c is the area of the heat exchanger, $T_{ca}(t)$ is the average temperature inside the heat exchanger which is related to $T_{co}(t)$ and $T_{ci}(t)$ as:

$$T_{ca}(t) = \frac{T_{ci}(t) + T_{co}(t)}{2}$$

Also, in **?**, the transmitters and the valves are modeled as:

- Level transmitter: it is supposed that the level transmitter is a capacitive type electronic transmitter that has a first order dynamics:

$$T_L \frac{dY_L(t)}{dt} + Y_L(t) = K_L H(t),$$

where T_L is its time constant, Y_L is the level signal and K_L is the transmitter gain.

- Temperature transmitter: It is supposed that a Pt_{100} RTD electronic sensor is installed in a thermowell at the tank outlet pipe. It is supposed that it has a second order dynamic:

$$T_T^2 \frac{d^2Y_T(t)}{dt^2} + 2T_T \frac{dY_T(t)}{dt} + Y_T(t) = K_T T(t),$$

where T_T is its time constant and K_T is its gain.

- Level control valve: it is supposed that a ball valve with an electropneumatic actuator is used. The valve inherent flow characteristics is nearly quadratic and the relationship between the flow $Q(t)$ and the input variable U_L is given by:

$$T_{vL} \frac{dX_L(t)}{dt} + X_L(t) = K_{xL} U_L(t),$$
$$Q(t) = K_{vL} X_L^2(t) \sqrt{\rho g H(t)},$$

where T_{vL} is the level control valve time constant, K_{xL} level control valve stem constant K_{vL} level control valve constant and $X_L(t)$ is the level control valve stem normalized travel.

- Temperature control valve it is also supposed to be a ball valve with an electropneumatic actuator, however, it is supposed that the valve has an equal-percentage inherent flow characteristics given by:

$$T_{vT} \frac{dX_T(t)}{dt} + X_T(t) = K_{xT} U_T(t),$$
$$Q_c(t) = K_{vT} R_{vT}^{(X_T(t)-1)} \sqrt{P_{cp} - (R_c Q_c^2(t) + P_{cr})},$$

where T_{vT} is the temperature control valve time constant, K_{xT} the temperature control valve stem constant, K_{vT} is the temperature control valve constant P_{cp} is the heating fluid pump discharge pressure, P_{cr} is heating fluid system return pressure and $X_T(t)$ is the temperature control valve stem normalized travel.

Taking this model in consideration, it can be said that, from the point of view of the controller, the controlled variables are give by the signals $Y_L(t)$ (which represents the level) and $Y_T(t)$ (which represents the temperature of the fluid of the tank). The manipulated variables are given by $U_L(t)$ (which directly affects Q) and $U_T(t)$ (that directly affects Q_c). $Q_i(t)$, T_i and T_{ci} are considered as disturbances. The state variables of the system are given by $H(t)$, $T_T(t)$, T_{co}, $Y_L(t)$, Y_T, $X_L(t)$ and $X_T(t)$, therefore, this model comprises a seventh order non-linear system for a two-input two-output industrial process. The parameters of the model can be found in Table 8.1. This model was implemented in Simulink® and can be found with the companion software. In Figure 8.2. Each equation of the model was implemented in a subsystem for clarity. For example in Figure 8.3 the Simulink implementation of

Table 1.1: Parameters for the CSTH process

Symbol	Value	Description
	Tank parameters	
ρ	$1200\,\mathrm{kg\,m^{-3}}$	tank fluid density
A	$0.0707\,\mathrm{m^2}$	tank inside section area
C_p	$4190\,\mathrm{J\,kg^{-1}\,°C^{-1}}$	tank fluid heat capacity
g	$9.8\,\mathrm{m\,s^{-2}}$	gravity acceleration
K_T	$2\,\%/°\mathrm{C}$	temperature transmitter gain
K_{vL}	1.25×10^{-5}	level control valve constant
K_{vT}	3×10^{-6}	temperature control valve constant
K_{xL}	$0.01\,/\%$	level control valve stem constant
Q_i	$7\times10^{-4}\,\mathrm{m^3\,s^{-1}}$	normal tank inlet fluid flow rate
T_i	$24\,°\mathrm{C}$	fluid inlet temperature
T_L	$2\,\mathrm{s}$	level transmitter time constant
T_T	$15\,\mathrm{s}$	temperature transmitter time constant
T_{vL}	$3\,\mathrm{s}$	level control valve time constant
T_{vT}	$5\,\mathrm{s}$	temperature control valve time constant
	Heat exchanger parameters	
ρ_c	$800\,\mathrm{kg\,m^{-3}}$	heating fluid density
A_c	$0.6362\,\mathrm{m^2}$	heat exchanger transfer area
C_{pc}	$2400\,\mathrm{J\,kg^{-1}\,°C^{-1}}$	heating fluid heat capacity
K_L	$125\,\%/\mathrm{m}$	level transmitter gain
K_{xT}	$0.01\,/\%$	temperature control valve stem constant
P_{cp}	$4.14\times10^5\,\mathrm{Pa}$	heating fluid pump discharge pressure
P_{cr}	$1.38\times10^5\,\mathrm{Pa}$	heating fluid system return pressure
R_c	$5.5\times10^{10}\,\mathrm{Pa/(m^3/s)^2}$	heating system pipe nominal flow resistance
R_{vT}	50	temperature control valve rangeability
T_{ci}	$320\,°\mathrm{C}$	heating fluid inlet temperature
U	$440\,\mathrm{J\,s^{-1}\,m^{-2}\,°C^{-1}}$	overall heat-transfer coefficient
V_c	$0.0139\,\mathrm{m^3}$	heat exchanger volume

the heat exchanger energy balance is presented. The result of this submodel is the computation of the state variable T_{ca}, which represents the average temperature of the heating fluid. As it can be seen, the parameters of the model are not hard-coded in the Simulink blocks, instead a parameter initialization script is called before the simulation starts. If the user desires to change any value of the parameters, it can be done globally in the script and then automatically called during the simulation.

1.1.2 Simplified linear model

In order to find a PID controller using the MOOTuning app, it is necessary to find a linear model of the plant in the operation point. An identification procedure was performed with a change of 10% in the value of U_T to find the transfer function

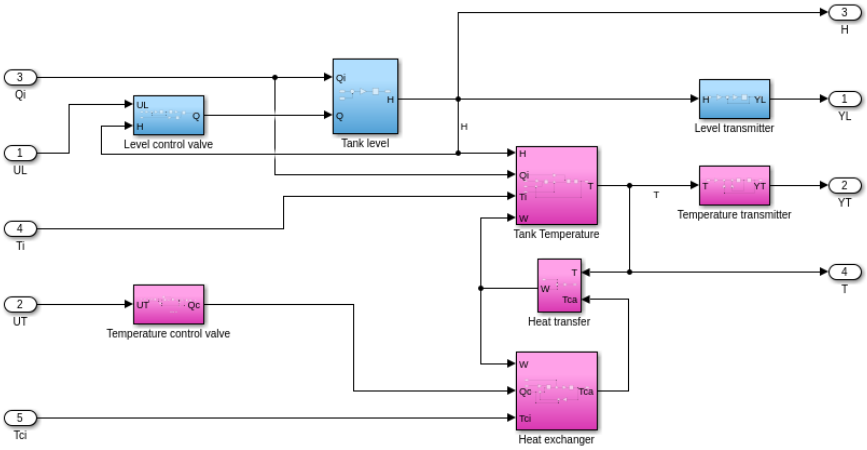


Fig. 1.2: Simulink implementation of the model of the heater.

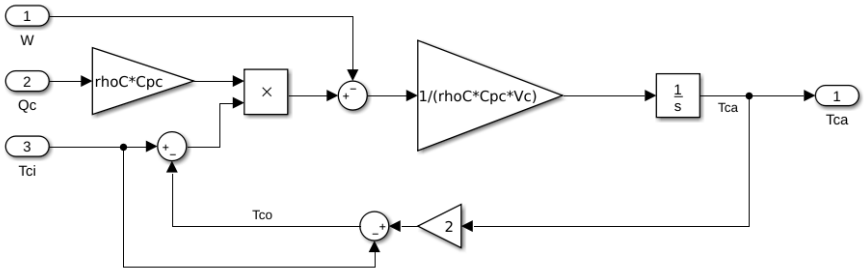


Fig. 1.3: Example of the implementation of the heat exchanger energy balance.

between Y_T and U_T. The response to this change is depicted in Figure 8.4. As it can be seen, the response is overdamped and takes approximately 500 s to reach a new steady state. A change in 10% on the input signal produces a variation of approximately 3.5% in the output signal. It has to be noticed that the presented signals are normalized between 0 and 100% representing the full spam of the transmitter and actuators.

In order to find the model, the process was supposed to have two poles, no zeros and a pure time-delay (also known as dead-time). Of course, if a linearization procedure were performed using the nonlinear model, a seventh order model would be obtained. However, for PID tuning, a first or second order model is usually expected to tune the controller.

Considering the experiment performed with the data as depicted in Figure 8.4, the resulting simplified model is given by:

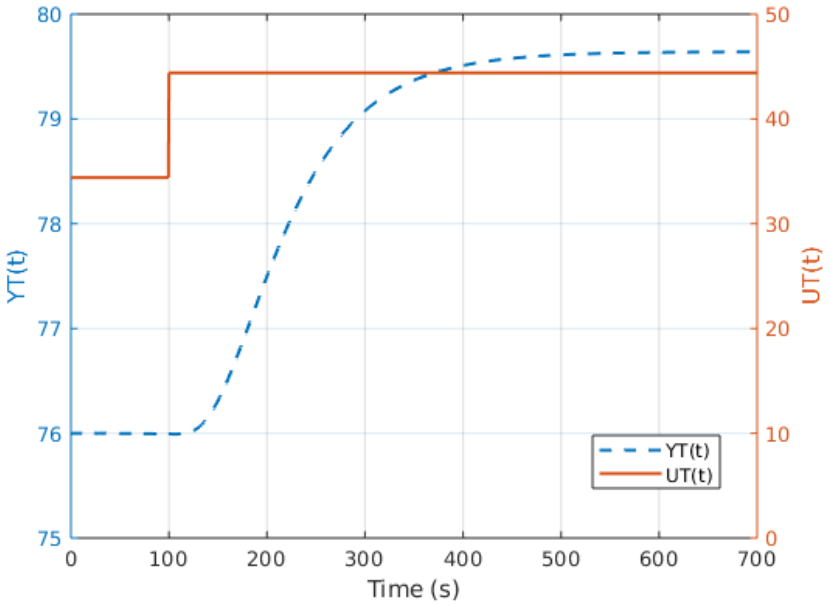


Fig. 1.4: Response of the process to a change of 10% in the $U_T(t)$ input.

$$\frac{Y_T(s)}{U_T(s)} = \frac{0.3658e^{-24.736s}}{(52.861s+1)(52.805s+1)}. \tag{1.1}$$

From this transfer function, it can be deduced that the gain is equal to $K = 0.3658$, the main time constant is given by $T = 52.861$, the ratio between the two time constant is given by $a = 0.9989$ and the dead-time is given by $L = 24.736$, therefore the normalized dead-time is given by $\tau = 0.4679$.

To test the validity of this simplified model, the response of the transfer function is compared against the response of the non-linear model. It was found that the transfer function response is very similar to the response of the non-linear model, as can be seen in Figure 8.5. It is clear that the non-linear model is a good representation of the dynamical response of the process. This is the first step in order to find a suitable PID controller to control the plant. In **?** the level of the tank is also controlled, however the dynamics of the level is simpler (its model can be approximated with a first order model without delay) and in this particular example, only the temperature is going to be controlled, while the level is considered to be constant.

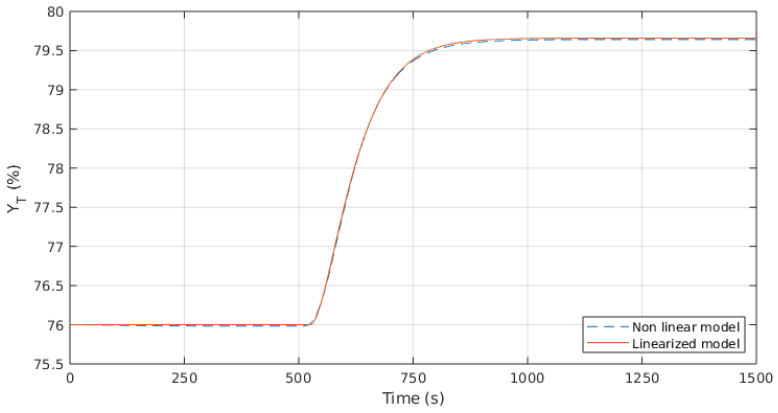


Fig. 1.5: Comparison between the linear and no linear models for the CSTH.

1.1.3 PID control of the CSTH considering two integral cost functions

In this section, the process will be controlled using a PID controller with different tuning methods and compared with the Multiobjective Optimization (MOO) framework used in the book.

Two different tuning rules were considered: the method by **?** and the method by **?**. For the case of the Rovira and Murril method, the model in (8.1) was reduced to a first order model using the Half-Rule in **?**:

$$\frac{Y_T(s)}{U_T(s)} = \frac{0.3658e^{-24.7360s}}{79.2635s+1}. \tag{1.2}$$

The equations were implemented as presented in **?**:

- Supposing a First-Order Plus Time Delay (FOPTD) model given by:

$$P(s) = \frac{Ke^{-Ls}}{Ts+1}$$

The Murrill tuning is given by:

Table 1.2: Comparison of different PID tunings for the CSTH process.

Tuning	K_p	T_i	T_d	β	J_r	J_{di}
Murril	5.87	65.02	23.21	1	82.51	15.34
Rovira	4.34	120.81	18.48	1	76.00	27.79
MOO01	11.3	58.82	29.48	0.43	85.65	7.06
MOO02	9.26	123.85	26.68	0.80	69.96	13.38
MOO03	8.20	185.21	28.14	0.99	67.93	22.54

$$K_p = \frac{1.435}{K}\left(\frac{T}{L}\right)^{0.921}$$
$$T_i = \frac{T}{0.878}\left(\frac{L}{T}\right)^{0.749}$$

$$T_d = 0.482T\left(\frac{L}{T}\right)^{1.137}$$

- Again, supposing a FOPTD as above, the Rovira tuning is given by:

$$K_p = \frac{1.086}{K}\left(\frac{T}{L}\right)^{0.869}$$
$$T_i = \frac{T}{0.740 - 0.13\frac{L}{T}}$$
$$T_d = 0.384T\left(\frac{L}{T}\right)^{0.914}$$

The values of the computed values can be found on Table 8.2, along with its associated values of J_{di} and J_r. The Murril and Rovira methods presented in the table were selected because they are intended to minimize the Integral of the Absolute Value of the Error (IAE). In all cases, the PID tuning is for a one degree of freedom controller (that is the reason why β is equal to one).

The PID that can be found using the data and the framework presented in Chapter 7 is a 2DoF controller, which may do the comparison somehow unfair. However, the idea now is to compare methods that tries to minimize the IAE. Using the MOOTuning Tool that accompanies this book, the Pareto front that was found is given as in Figure 8.6. As it was expected, all the controllers found using the MOO tool present lower values for J_{di} and J_r. If all controllers had the same topology, most certainly both controller would be close to the anchor points. However, what it is important here is the fact that, using the tool, the user has the ability to chose between practically an infinity of possible controllers. From the Pareto front, three different tunings were selected in order to compare the responses using the non-linear plant.

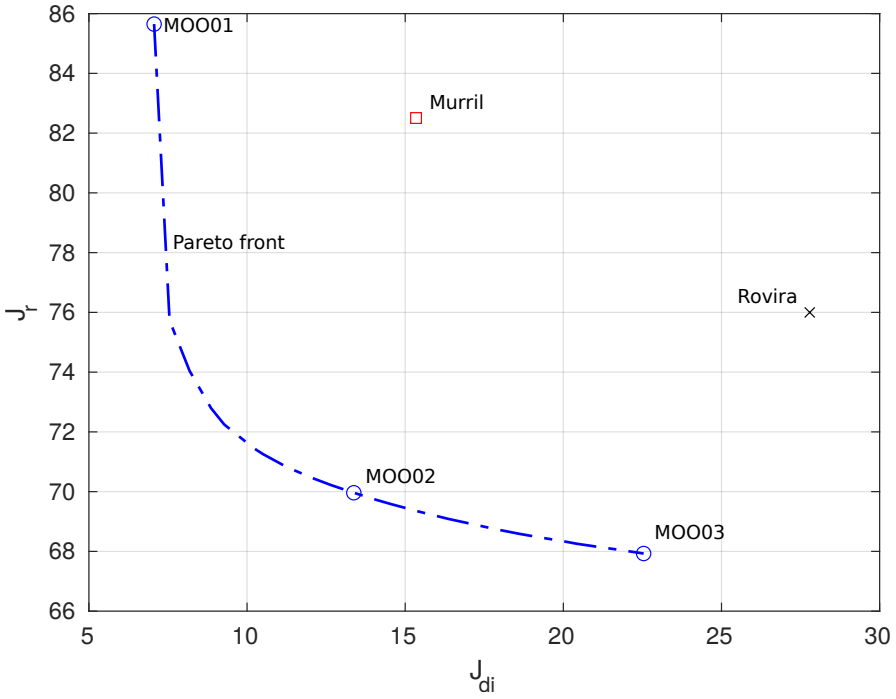


Fig. 1.6: MOOTuning compared to the Murril and Rovira methods that also minimizes IAE.

The values of the parameters are presented also in Table 8.2 and depicted as circles in the Pareto front in Figure 8.6.

In Figures 8.7 and 8.8 the responses to a step change in the setpoint and in the disturbance are presented. The corresponding values of IAE are also presented in the graph. In all cases, the robustness was not considered as a constraint. but it is possible to include it within the MOOTuning software. From Figure 8.6 given the steep slope of the curve for lower values of J_{di} that a small change in J_{di} may improve substantially the performance for J_r. Therefore, one may be more prone to select a controller that may have a little degradation in J_{di} and for this reason, controller MOO01 may not be a good selection as a final solution unless having the minimum value possible of J_{di} is the final goal.

The controller MOO02 may be seen as an intermediate solution between MOO1 and MOO03 in case both J_{di} and J_r are equally important for the decision maker. The power of the multiobjective framework is evident, and giving that the computational work is done offline, it becomes a good tool for the tuning of PID controllers in an industrial setting.

To check how the tuning performs with the nonlinear model, a simulation was performed using the tuning of the controller MOO02. The setpoint was increased by 5% at $t = 100\,\mathrm{s}$ and the temperature of the steam was increased by $10\,^{\circ}\mathrm{C}$ at $t = 700\,\mathrm{s}$.

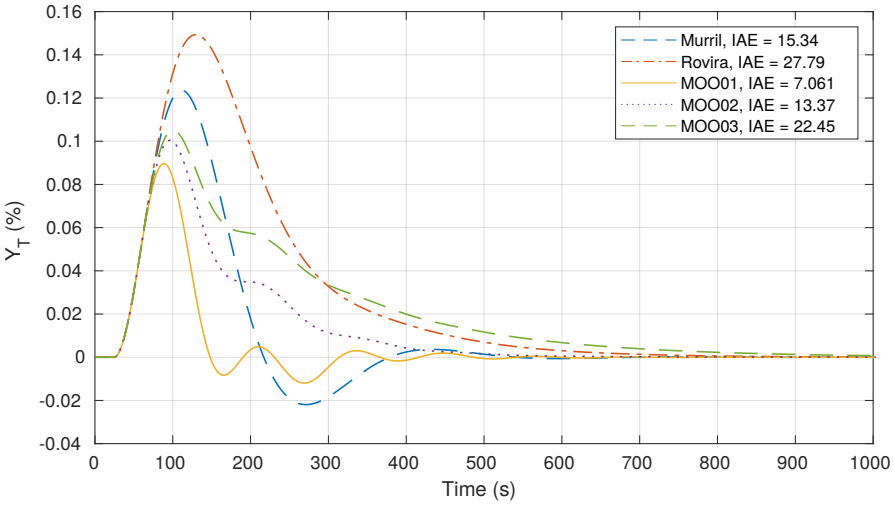


Fig. 1.7: Regulator response comparison for minimum IAE.

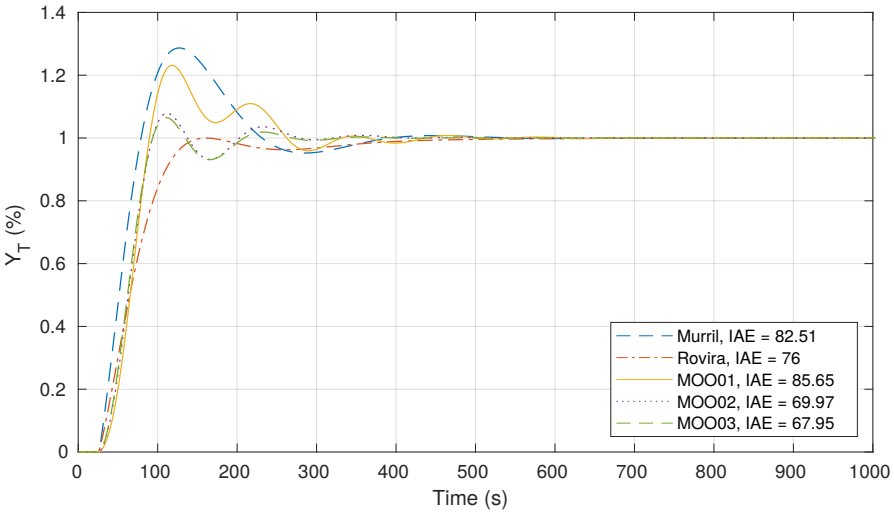


Fig. 1.8: Servo response comparison for minimum IAE.

The response is presented in Figure 8.9. As it can be seen, the servo response is very close to the one presented in Figure 8.8, which is a clear indicator that the linear model was a good approximation of the plant at the given operation point. The response to the change in the stem temperature cannot be compared with the regulation presented in Figure 8.7, because the disturbance was not applied directly at the input of the plant. However, it is interesting to note that the controller was able to respond with a good dynamic even though it was not optimized for this case.

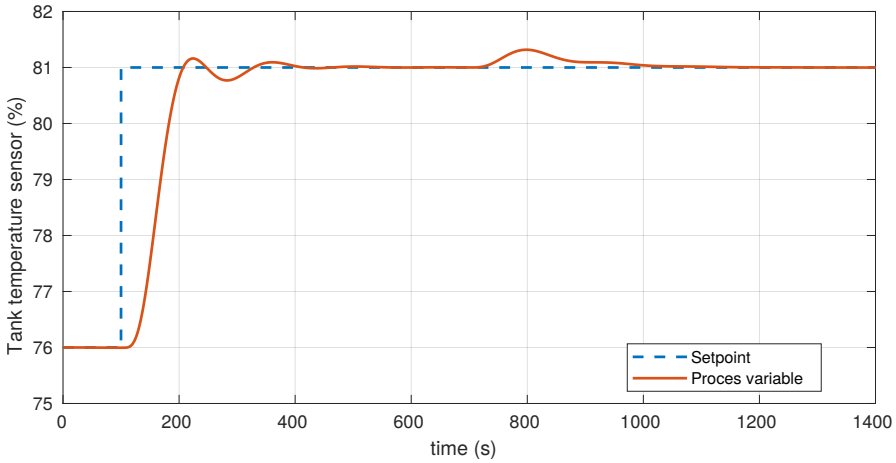


Fig. 1.9: Response of the controlled system using the nonlinear model for the CSTH.

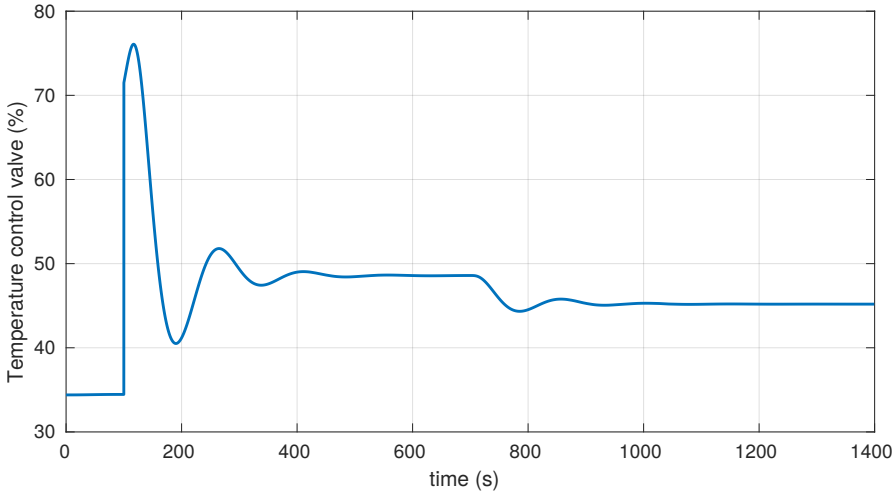


Fig. 1.10: Response of the controlled system using the nonlinear model for the CSTH.

The controlled variable is presented in Figure 8.10. When the setpoint changes, the response of the controller is abrupt (more than double its original value), but then the value rapidly reaches the new setpoint. The change produced by the disturbance has a milder response, and in less than 200 s reach again a new steady state.

The case presented here was intended to show the steps to use the Pareto front as the methodology to find the controller tuning more appropriate to the task. The example is a simple plant, but very representative of the dynamics that can be found in an industry environment. Many of the plants can be modeled as a second order

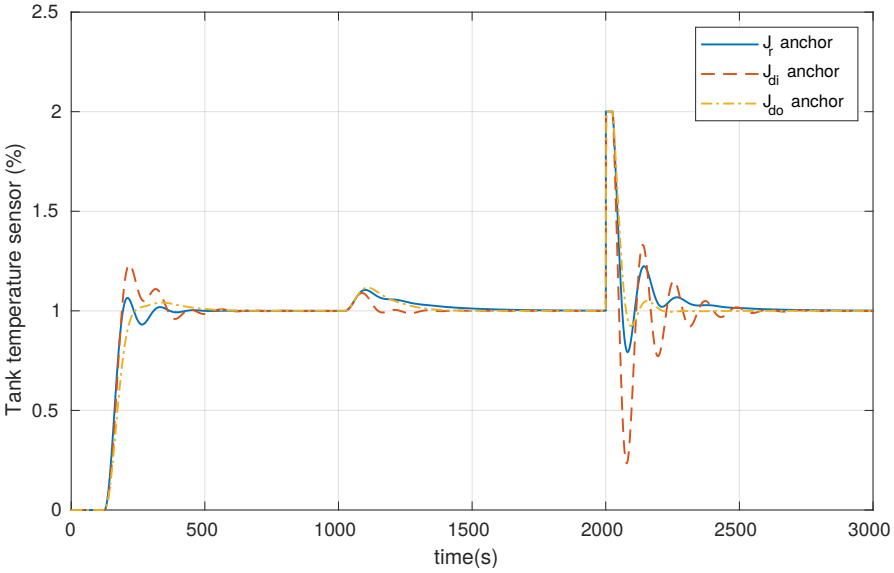


Fig. 1.11: Response of the CSTH process in the three anchor points of the Pareto front for $M_s \leq 2.0$.

overdamped process, and therefore, the tool and the data used in this book are readily applied in many cases.

1.1.4 PID control of the CSTH considering three integral cost functions

The MOOTuning software is also able to find the optimal parameters of a PID controllers considering three cost functions as presented in Section 7.2.2. Of course, it is not necessary to use this software since the data base with all the values is also part of the companion software, but the MATLAB® app has the advantage to be a simple interface between the user and the data.

As an example, the tuning tool is used to find the parameters of the controller that has the lowest J_{do} value. This case is interesting because the anchor point where J_{do} has the lowest value, neither the value of J_r nor J_{di} have their maximum value. On the other hand, when J_r is set to be the lowest possible value, J_{do} becomes the function that has an intermediate value but J_{di} has its maximum value from the Pareto. Only for the anchor point where J_{di} is minimum, both J_r and J_{do} get their maximum value. The responses for the three anchor points are depicted in Figure 8.11 for the case where $M_s \leq 2.0$. As it can be seen, the response is quite different among the three anchor points. First a change in the reference value is

performed at $t = 100\,\text{s}$, an input step disturbance is present at $t = 1000\,\text{s}$ and an output step disturbance is introduced in the system at $t = 2000\,\text{s}$. The values of the cost functions are presented in Table 8.3. The percentage increment is reported for

Table 1.3: Cost functions for the three cost functions case scenario.

	J_r anchor point	J_{di} anchor point	J_{do} anchor point
J_r	67.95 (minimal)	85.63 (+26.02%)	82.46(+21.35%)
J_{di}	22.46 (+218.13%)	7.06 (minimal)	17.63 (+149.72%)
J_{do}	74.25 (+38.01%)	102.43 (+90.39%)	53.80 (minimal)
Parameters	$K_p = 8.19$ $T_i = 185.18$ $T_d = 28.14$ $\beta = 0.99$	$K_p = 11.29$ $T_i = 58.8306$ $T_d = 29.48$ $\beta = 0.43$	$K_p = 6.18$ $T_i = 108.76$ $T_d = 29.93$ $\beta = 0.82$

each cost function in each case.

Using Figure 8.11 and Table 8.3, it can be confirmed that the tuning with the best parameters for J_{di} produces the worst responses for the other two functions. With these results one may be prone to select the tuning for the lowest value of J_{di} as the final choice because it does not have the worst values of the other cost functions and it has much better response for the output disturbance case. Observe for example the response for the J_{di} anchor point where the response for the output disturbance is bad (it has a J_{do} increment of $+90.39\%$ with respect to its lowest value). In some sense, it could be seen as the best compromise between the cost functions. Only if the control engineer is heavily invested in minimize the J_{di} cost function, it may select the J_{di} anchor point as the final response, however, using the multiobjective framework presented here, he or she has to be fully aware that is selecting the worst response for the other functions.

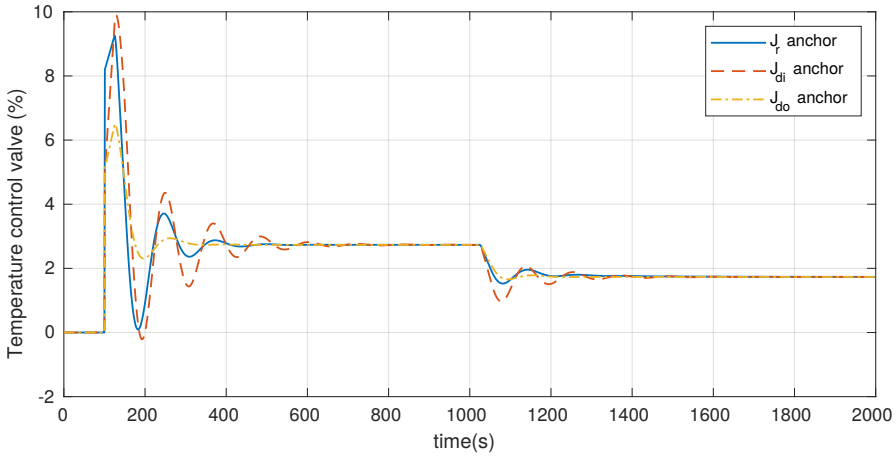
Another reason to select the J_{do} anchor point is related to the control effort. When the Total Variation (TV) computed as:

$$TV = \sum_{i=0}^{N-1} \left(u(i+1) - u(i) \right),$$

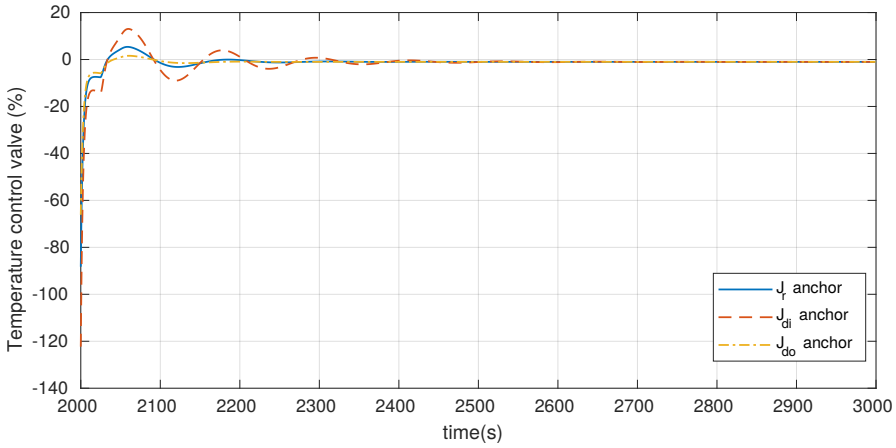
is compared between the three responses, it is found that $TV_{J_r} = 223.82$, $TV_{J_{di}} = 352.94$ and $TV_{J_{do}} = 152.81$ using a step size of $0.01\,\text{s}$. It is clear that the response given by the J_{do} anchor point is a very good choice among all the possible values. The control signal is plotted on Figure 8.12. Since the control signal has a larger magnitude for an output disturbance rejection, it was plotted in different axis.

As it can be seen from the response, effectively the response of the J_{do} anchor point is smoother than the other two. Even if the control engineer is not looking for the best response to the output disturbance, the obtained tuning may be a good compromise between servo and regulation responses with a mild control signal.

In section 8.1.3, the system was controlled considering only two sources of disturbances. When adding another dimension to the problem, certainly the selection



(a) Reference and input disturbance step changes.



(b) Output disturbance step change.

Fig. 1.12: Control signal for the CSTH using three cost functions.

of the final controller may be more difficult because another degree of freedom is added. However, the insight that was gathered from rethinking the problem from this other point of view can be seen as beneficial, because the tuning found using the 3 dimensional Pareto could be a better solution (from a physical point of view) that may not be part of the front with only two functions.

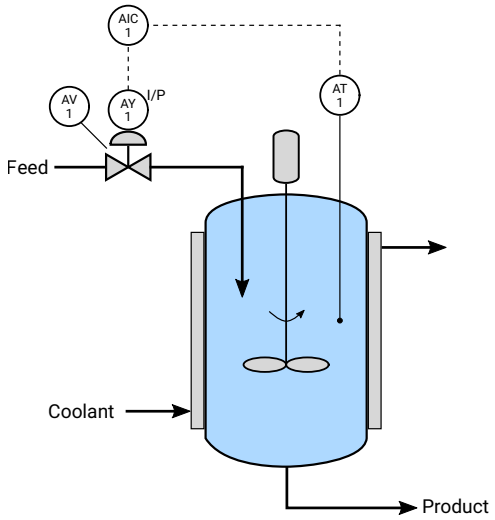


Fig. 1.13: CSTR process using the Van de Vusse reaction model.

1.2 Continuously Stirred Tank Reactor

1.2.1 Description of the process

The Continuously Stirred Tank Reactor (CSTR) with the Van de Vusse reaction (**?**) is a common benchmark plant for testing control algorithms given its different dynamics depending on the operating point and is depicted in Figure 8.13. For this particular case, the isothermal process is considered and both the concentration sensor as the valve actuator will be modeled as well. The objective is to control the feed flow to obtain the desired concentration of a product.

The Van de Vusse reaction models a process where desired product B is obtained from A, but at the same time, both A and B are degraded to D and C respectively. The chemical equation that represent this reaction is given by:

$$\begin{aligned} A &\xrightarrow{k_1} B \xrightarrow{k_2} C, \\ 2A &\xrightarrow{k_3} D, \end{aligned}$$

where k_i are the rate constants of the formation rates of A and product B as given by (**?**):

$$\begin{aligned} r_A &= -k_1 A - k_3 A^2, \\ r_B &= k_1 A - k_2 B. \end{aligned} \tag{1.3}$$

When performing a mass balance, the model becomes (**?**):

Table 1.4: Parameters values for the CSTR model.

$k_1 = 0.833\,\text{min}^{-1}$	$k_2 = 1.667\,\text{min}^{-1}$
$k_3 = 0.167\,\text{min}^{-1}$	$C_{Ai} = 10\,\text{mol}\,\text{L}^{-1}$
$V = 700\,\text{L}$	

$$\begin{aligned}\frac{dC_A(t)}{dt} &= \frac{F_r(t)}{V}(C_{Ai} - C_A(t)) - k_1 C_A(t) - k_3 C_A^2(t)\\ \frac{dC_B(t)}{dt} &= -\frac{F_r(t)}{V}C_B(t) + k_1 C_A(t) - k_2 C_B(t)\end{aligned} \tag{1.4}$$

where C_A and C_B represents the reactants concentrations in $\text{mol}\,\text{L}^{-1}$, C_{Ai} is the concentration of A in the feed flow in $\text{mol}\,\text{L}^{-1}$, F_r is the input flow in $\text{L}\,\text{min}^{-1}$, and V is the volume of the CSTR in L. The nominal values of the parameters are presented in

The range of the sensor for the product is supposed to be in the range 0 to $1.5714\,\text{mol}\,\text{L}^{-1}$ and the maximum flow that is allowed by the valve is given by $634.1719\,\text{L}\,\text{min}^{-1}$. Given these values, the model for the sensor-transmitter is given by:

$$y(t) = \left(\frac{100}{1.5714}\right) C_B(t), \tag{1.5}$$

while the transmitter is modeled as:

$$F_r(t) = \left(\frac{634.1719}{100}\right) u(t), \tag{1.6}$$

where $u(t)$ and $y(t)$ are the normalized input and output signal, respectively. The operation point of the plant is given by $u_0 = 60\%$ and $y_0 = 70\%$ which represents concentration of $C_{A0} = 2.9175\,\text{mol}\,\text{L}^{-1}$ and $C_{B0} = 1.10\,\text{mol}\,\text{L}^{-1}$ with the input concentration given by $C_{Ai0} = 10\,\text{mol}\,\text{L}^{-1}$.

These parameters give the system a wide range of operation. In Figure 8.14 it can be seen that with the value of C_{Ai0} given and the input at 60%, the output yield 70%. In the figure the dashed lines represents the variation due to $\pm 10\%$ variation of C_{Ai}. It can be seen that, with a 100% value for u and a variation of 10% of C_{Ai}, the output is less than 100% of the capacity of the sensor.

The concentration of C_A and C_B are both plotted in Figure 8.15. The C_A concentration is on the horizontal axis is while the C_B concentration is on the vertical axis. The selected operation point is represented with a circle in the curve and also the curves with the variation on the value of C_{Ai} are also presented. As it can be seen, in the operation point selected, the curves start to diverge, which means that the model has a larger dependency on the variation of the input concentration. This has to be taken into account when designing the feedback controller.

When considering the transient response to a change in the input u, the response is as given in Figure 8.16. It is interesting to note that the system has an inverse response. This characteristic limits the possible performance of the controlled loop,

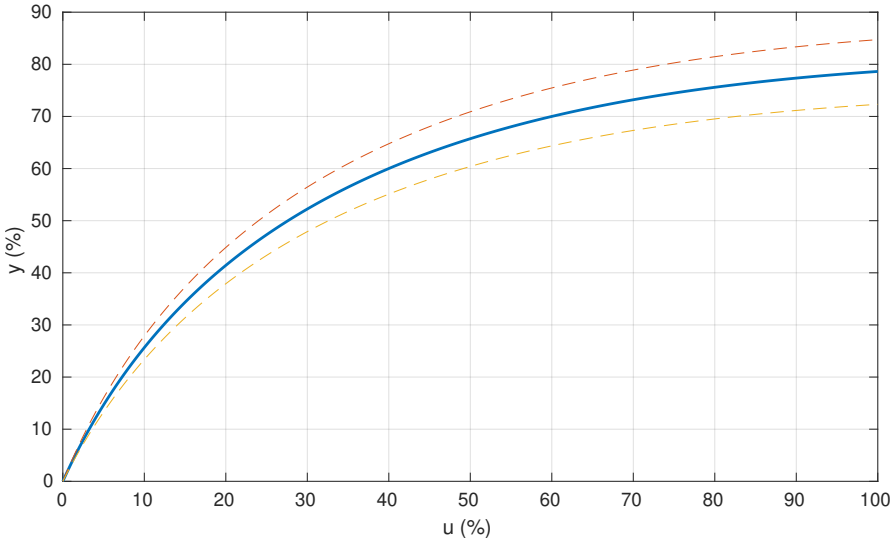


Fig. 1.14: Operation points of the CSTR process.

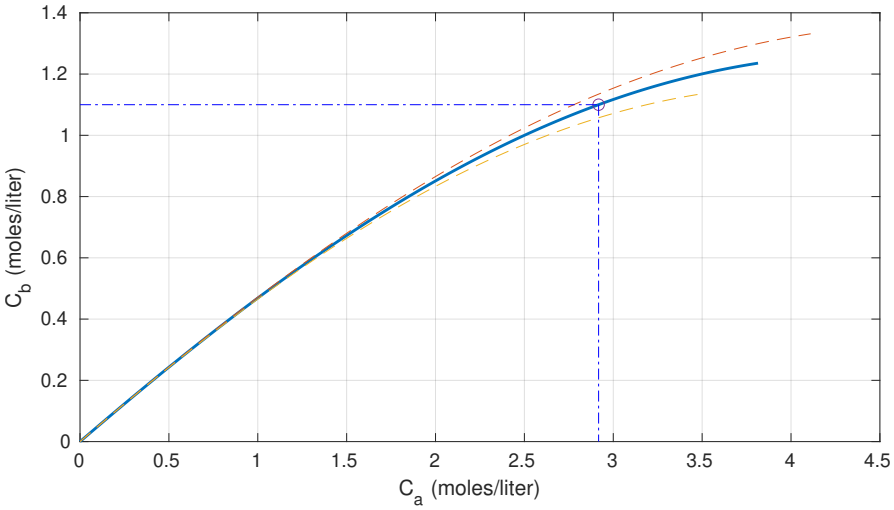


Fig. 1.15: Concentration of the reactants for all possible operating points.

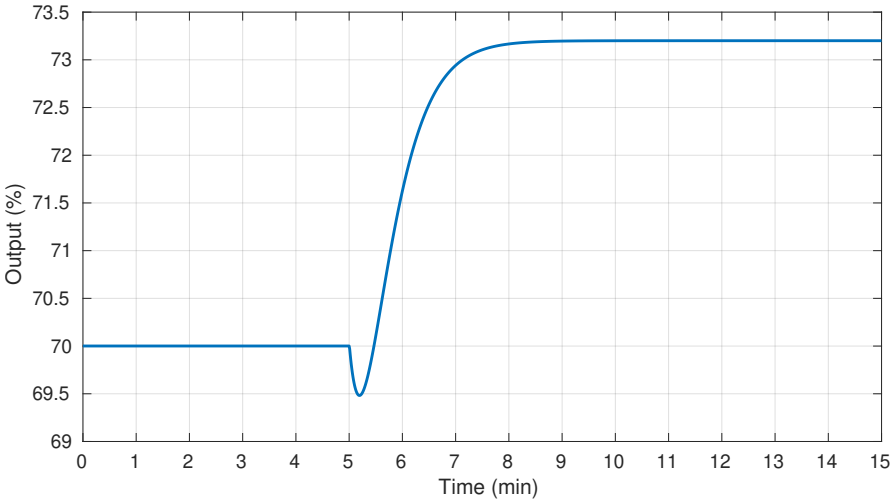


Fig. 1.16: Open loop response to a step change in the input signal of the CSTR.

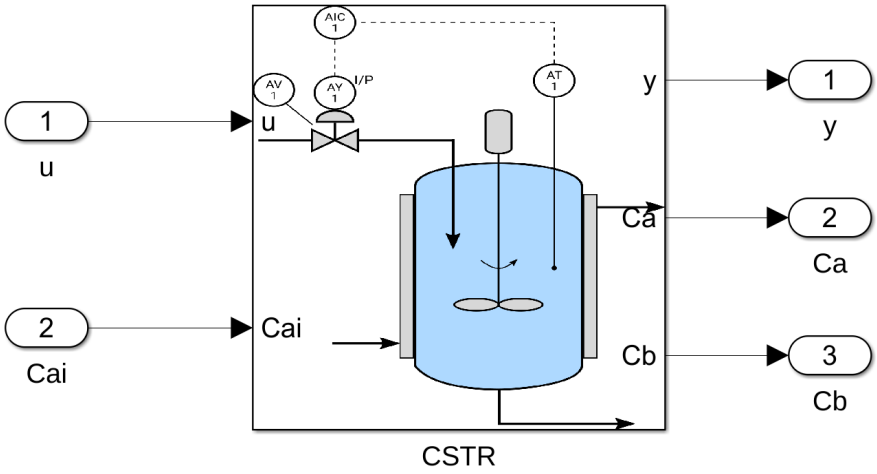


Fig. 1.17: CSTR implemented as an S-function in Simulink.

making unfeasible to increase the gain of the controller to achieve a faster response without instability. This is also another point that has to be considered when designing the controller.

This model was implemented as an S-Function in MATLAB/ Simulink and can be found in the companion software. In Figure 8.17, the basic block with the model is presented. It has u and C_{Ai} as inputs and y, C_A and C_B as outputs. With the intention to facilitate the characterization of the model, a mask was designed to enter the parameters and the initial value of the states. This mask can be found in Figure 8.18.

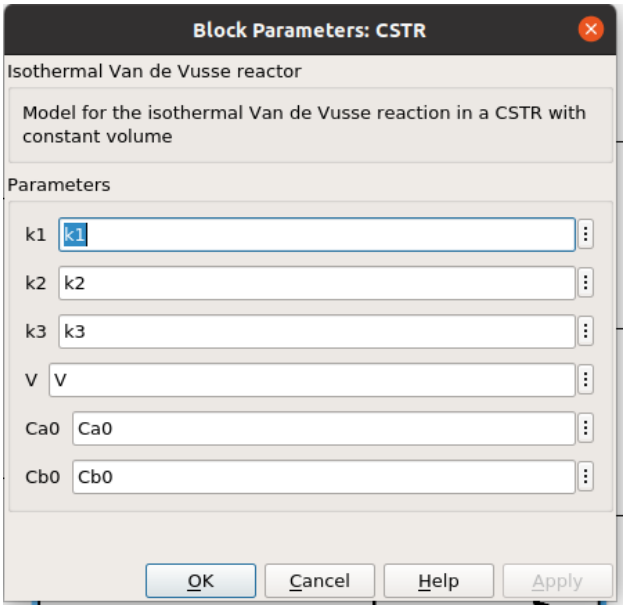


Fig. 1.18: Simulink mask to enter parameters and initial values for the CSTR system.

All the values of the parameters can be set on the MATLAB workspace and use directly on the mask. This is and advantage in case the user desires to use another set of parameters or compare the response of the system with different values.

The objective of this example is to control the reactor using the multiobjective approach. In order to use the framework and the MATLAB app, it is necessary to find a suitable linear second order model. Next, the procedure to find this model is presented.

1.2.2 Linearization

In order to use the MOOTuning software, it is necessary to have a linear model of the process. In this case, the linearization is done by taking the first order approximation of the model in (8.4) near the operation point.

When defining the incremental variables $\delta\mathbf{x}$, $\delta\mathbf{u}$ and δy as:

$$\delta\mathbf{x} = \begin{bmatrix} C_A - C_{A0} \\ C_B - C_{B0} \end{bmatrix},$$
$$\delta\mathbf{u} = \begin{bmatrix} u - u_0 \\ C_{Ai} - C_{Ai0} \end{bmatrix},$$
$$\delta y = y - y_0.$$

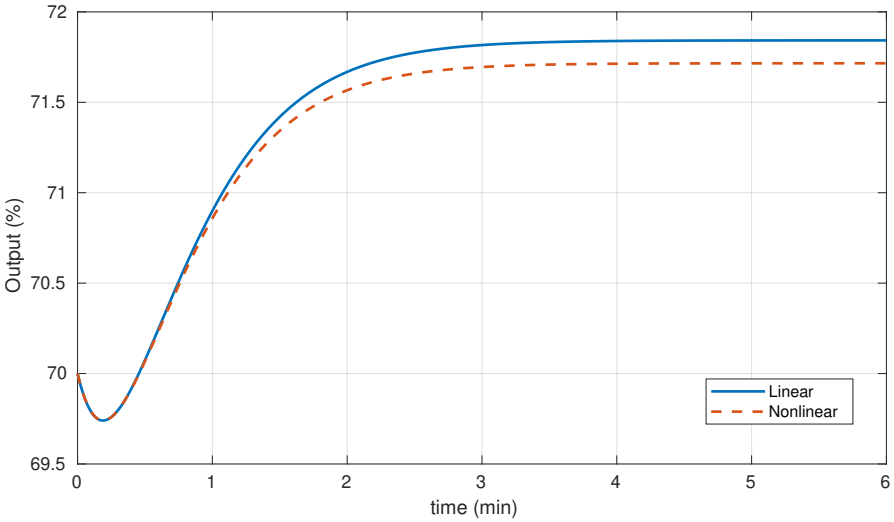


Fig. 1.19: Comparison of the nonlinear and linear model for the CSTR for a 5% change in the input.

The process dynamics can be approximated by the first order model given by:

$$\dot{\delta \mathbf{x}} = \mathbf{A}\delta\mathbf{x} + \mathbf{B}\delta\mathbf{u}, \tag{1.7}$$

$$\delta y = \mathbf{C}\delta\mathbf{x}, \tag{1.8}$$

where,

$$\mathbf{A} = \begin{bmatrix} \frac{6.341719u_0}{V} - k1 - 2k_3C_{A0} & 0 \\ k1 & \frac{-6.341719u_0}{V} - k_2 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} \frac{6.341719(C_{Ai0} - C_{A0})}{V} & \frac{6.341719u_0}{V} \\ \frac{-6.341719C_{B0}}{V} & 0 \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} 0 & \frac{100}{1.5714} \end{bmatrix}.$$

Using the parameters of Table 8.4 and the operation point values, the resulting transfer function between input u and output y is found to be:

$$F(s) = \frac{-0.6342s + 1.913}{s^2 + 4.56s + 5.193} \tag{1.9}$$

It has to be clear that this transfer function is only an approximation of the model, and therefore it is valid only around the operating point. When a step change of 5% is entered in the system, the response of the nonlinear model and the response of the transfer function start to differ as presented in Figure 8.19. As it can be seen,

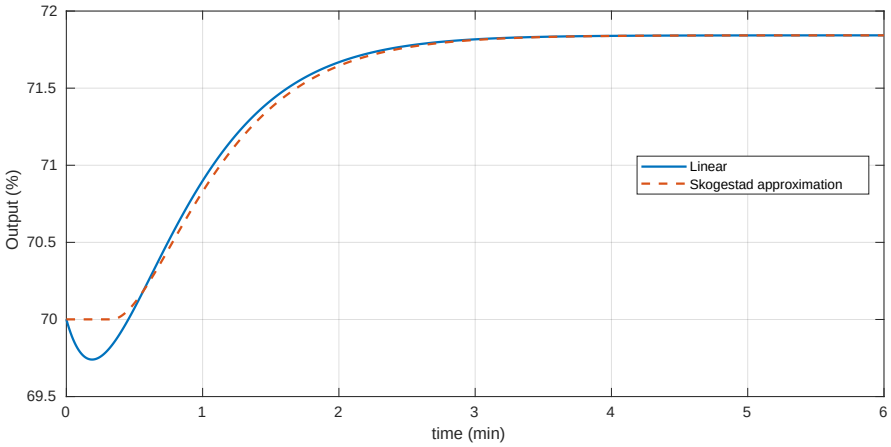


Fig. 1.20: Approximation of the non-minimum phase zero with a time delay for the linear model of the CSTR.

the linear model represents the transient response well, however it fails to predict the steady state. Of course, the smaller the change in the input, the better is the approximation for both the transient and the steady state response.

However, the model that is expected in the MATLAB app does not contemplate a non-minimum phase zero in the model. Using the model reduction rules of **?**, the Overdamped Second-Order Plus Time Delay (ODSOPTD) model ends as:

$$F_{aprox} = \frac{0.368e^{-0.331}}{0.193s^2 + 0.878s + 1}. \tag{1.10}$$

which corresponds to approximate the non-minimum phase zero by a time delay. The gain is given by $K = 0.3684$, the time constant is given by $\tau = 0.4256\,\text{min}$, the time delay is $L = 0.3315\,\text{min}$ and $a = 0.9408$. This new transfer function gives the response presented in Figure 8.20. Of course, the dynamic model is quite different from the original nonlinear model, both in the transient and steady states. Since the controller is going to be designed with the linearized time-delayed transfer function, it is necessary to consider some constraint on the performance in order to give certain robustness to the design.

1.2.3 Controller design

For this particular example, the design of the controller parameters will contemplate three different robustness cases: without any constraint to the value of M_s, $M_s = 2.0$ and $M_s = 1.8$. The MOOTuning app is able to introduce the robustness as a

Table 1.5: Different tunings for the CSTR obtained with MOOTuning.

Design	K_p	T_i	T_d	β	J_{di}	J_r
			$M_s \geq 2.0$			
C1	6.00	0.61	0.29	0.49	0.13	0.98
C2	4.54	1.24	0.26	0.99	0.27	0.80
C3	5.54	0.85	0.26	0.66	0.14	0.85
C4	5.33	0.91	0.26	0.73	0.17	0.84
			$M_s = 2.0$			
C1	4.43	0.76	0.21	0.66	0.17	0.92
C2	4.29	1.19	0.25	0.99	0.28	0.80
C3	4.44	0.86	0.22	0.73	0.19	0.86
C4	4.40	1.00	0.24	0.87	0.23	0.83
			$M_s = 1.8$			
C1	3.88	0.72	0.22	0.72	0.20	0.98
C2	3.86	1.13	0.24	0.99	0.29	0.82
C3	3.91	0.85	0.21	0.75	0.22	0.89
C4	3.92	0.96	0.22	0.87	0.24	0.85

constraint in the design. However, it is limited to certain values of M_s which are considered to be reasonable for an industrial control applications.

In this particular example, only J_r and J_{di} are going to be considered. The design is based on the model in (8.10) and will be tested on the nonlinear model as well. Several points of the Pareto are going to be tested: the two anchor points (C1 for best regulator and C2 for best servo), the best regulator possible with a 20% degradation on J_r (C3) and the best servo possible with a 20% degradation on J_{di} (C4).

All the tuning values for the cases studied are presented in Table 8.5. In general, it can be seen that the proportional gain is heavily dependent of the robustness value (more robustness implies lower values of K_p). On the other hand it seems that the value of T_i is more dependent of the degradation of J_{di} while the variation of T_d is small across all cases. The value of β seems like a combination of the degradation of J_{di} and the value of the robustness.

To compare the responses, it is useful to plot the step response of C1 controllers across all robustness values as presented in Figure 8.21 and for C2 controllers as in Figure 8.22. It is interesting to note that C1 controllers presents more variability in the response with respect to the controllers of the C2 family. In both figures it is clear that there is a compromise between the servo and the regulator responses, but this compromise becomes less important when the robustness is considered. Consider Figure 8.21, the response for $M_s \geq 2.0$ does not take into account any constraint on the robustness and as it can be seen this response is very different than the cases where $M_s = 2.0$ and $M_s = 1.8$ are forced. It has to be noticed that adding the robustness constraint greatly limits the possible values of the controllers.

On the other hand, it was found that for the C2 family of controllers the responses are very similar among all robustness, as can be seen in Figure 8.22. The reason for

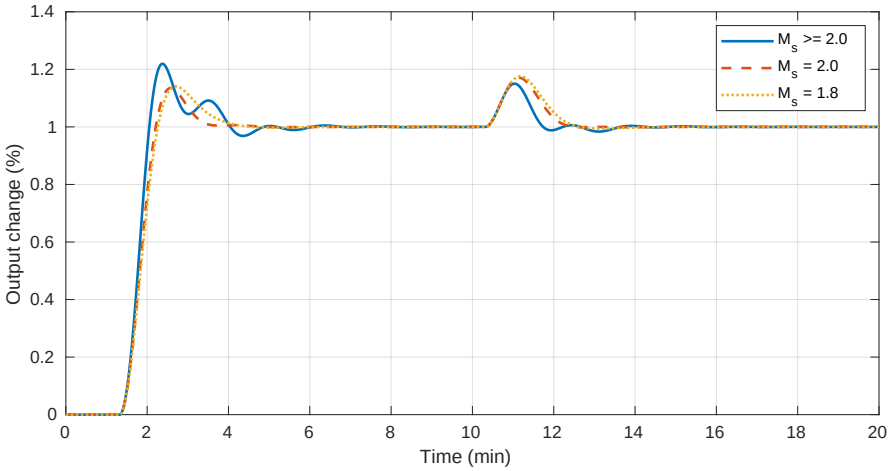


Fig. 1.21: Response for C1 controllers for all M_s values.

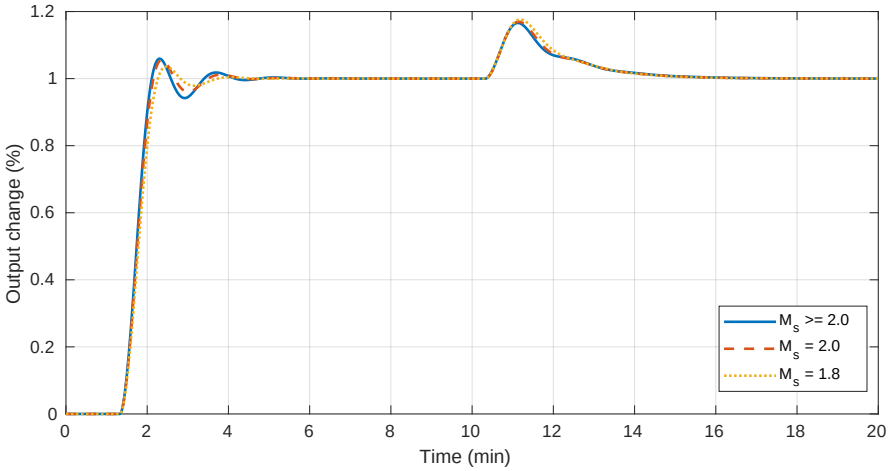


Fig. 1.22: Response for C2 controllers for all M_s values.

this is the parameter β. This parameter does not affect the robustness value of the controlled system nor the regulator response. Therefore, the optimization tend to find a low value for K_p, which gives better robustness, but then compensates with a high value of β.

The control signals for the controllers are presented in Figures 8.23 and 8.24. For the case of C1 controllers, the control signal varies significantly according to the robustness value. The case without any constraint has a larger peak and more oscillatory response than any of the other controllers. It is interesting, however, to note that the control signal for the C2 controllers are very similar and again, this

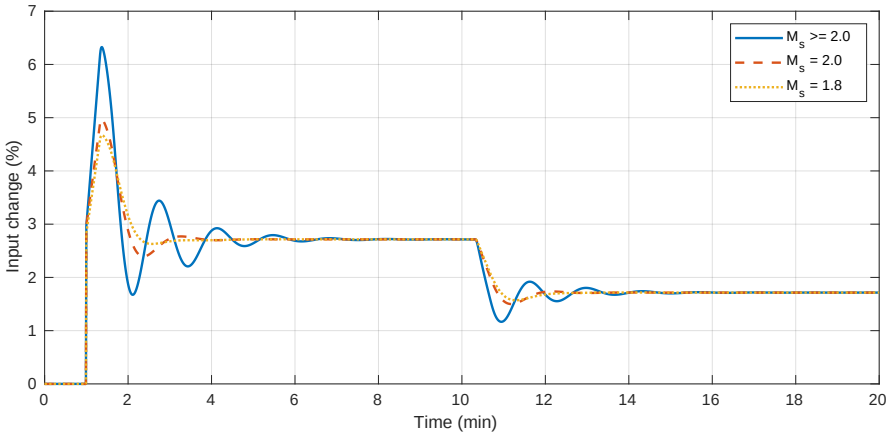


Fig. 1.23: Control signal for C1 controllers for all M_s values.

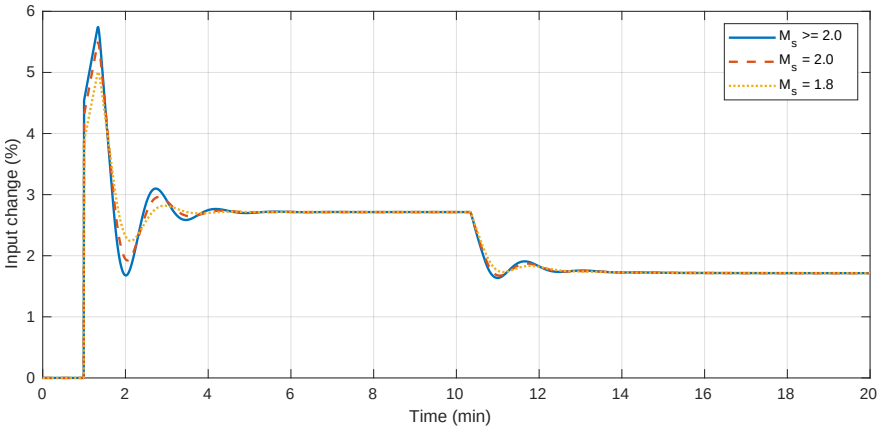


Fig. 1.24: Control signal for C2 controllers for all M_s values.

is due to the presence of the β parameter. The fact that an unconstrained controller may be applied to the controlled system has to be taken with caution, because in this design, the model used for the controller tuning is known to be different than the system. Therefore, applying an aggressive control signal may lead to unwanted oscillatory behavior and even instability.

Now, the responses of the different controllers families can be compared for the same given robustness, for example $M_s = 2.0$ as in Figure 8.25. For all this controllers, the robustness is near $M_s = 2.0$, but the performance is varied from J_r and J_{di}. Here, the compromise between both responses is clearer since the best servo is at the same time the worst regulator and the best regulator is the worst servo.

However, given that all controllers are constrained to fulfill $M_s = 2.0$, the difference between them are not very notorious. It can be seen on Figure 8.26 that the

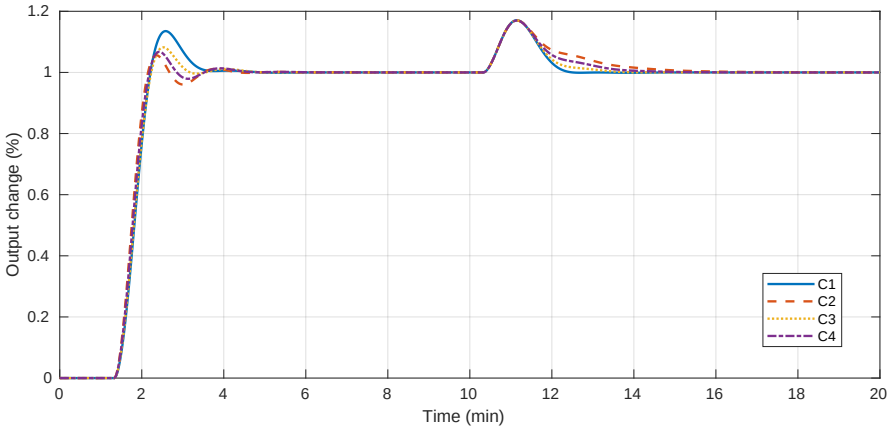


Fig. 1.25: Response for all controller families with $M_s = 2.0$.

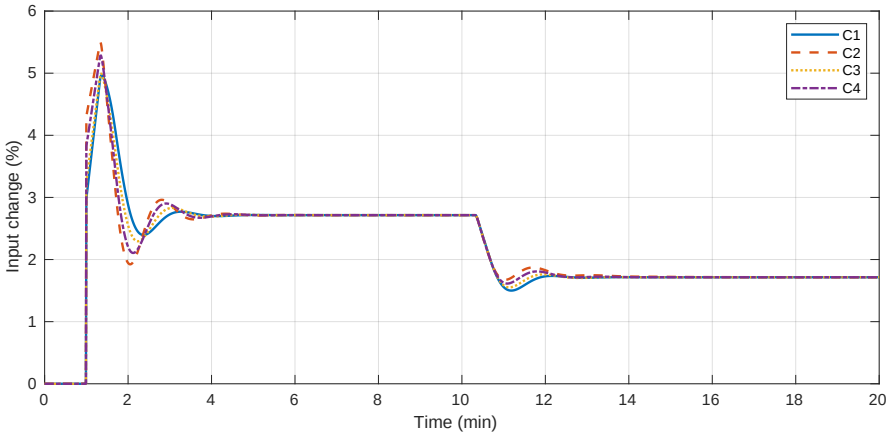


Fig. 1.26: Control signal for all controller families with $M_s = 2.0$.

peaks and oscillatory behavior of all controllers are practically the same. When the robustness became too tight (for example values below 1.4) the controllers for servo and regulator practically become the same and the degree of freedom to select the dynamic behavior is practically non-existent.

1.2.4 Validation of the controller designs

The designed controllers were tested using the nonlinear model of the CSTR. It is important to note that the controllers were tuned for a plant model that is different than the "real" plant. In this particular case, several approximations where made

Table 1.6: IAE values for the controllers applied to the nonlinear model as servo controllers

Robustness	Controller			
	C1	C2	C3	C4
$M_s = 1.8$	0.94	0.83	0.87	0.84
$M_s = 2.0$	0.86	0.79	0.83	0.81
$M_s \geq 2.0$	11.20	0.78	1.02	0.83

Table 1.7: TV values for the controllers applied to the nonlinear model as servo controllers

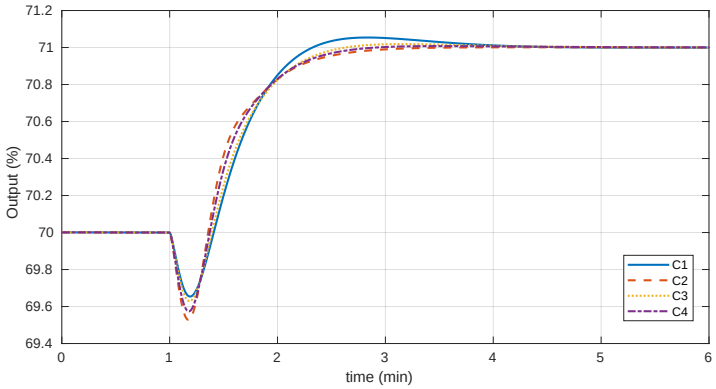
Robustness	Controller			
	C1	C2	C3	C4
$M_s = 1.8$	9.40	13.20	9.14	11.10
$M_s = 2.0$	11.34	20.01	12.75	16.37
$M_s \geq 2.0$	1936.10	29.10	128.30	73.20

which were necessary in order to use the framework presented in the other chapters. Because of this, it is important to look for a solution that takes into account this possible sources of uncertainty, therefore instability. In Table 8.6, the IAE for the servo response to a step change in the reference at $t = 1\,\mathrm{s}$ is presented, with the response plotted in Figure 8.27. On the other hand the total variation is presented in Table 8.7 and the control signal in Figure 8.28. All controllers where tested for all possible robustness measures.

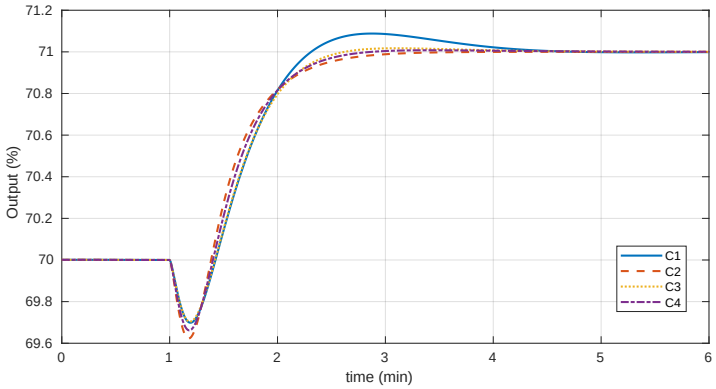
The first thing to notice is that for the case of $M_s \geq 2.0$, presented in Figure 8.27c, the response becomes practically unstable for the C1 tuning. The PID controller that was implemented in Simulink has an antiwindup loop that prevents the integral part from becoming infinite. For the other cases the response presents important oscillations. The problem with C1 is that the gain is relatively high, which produces the controller to saturate as can be seen in Figure 8.28c.

For this particular case, controller C2 (which was expected to be the "best" servo) yields a better IAE when comparing the $M_s \geq 2.0$ against $M_s = 1.8$ and $M_s = 2.0$ as it was expected knowing Table 8.5. However this is not the case for C4 controllers, because the nonlinearity of the plant and the approximations made start to take a toll on the design of the controllers.

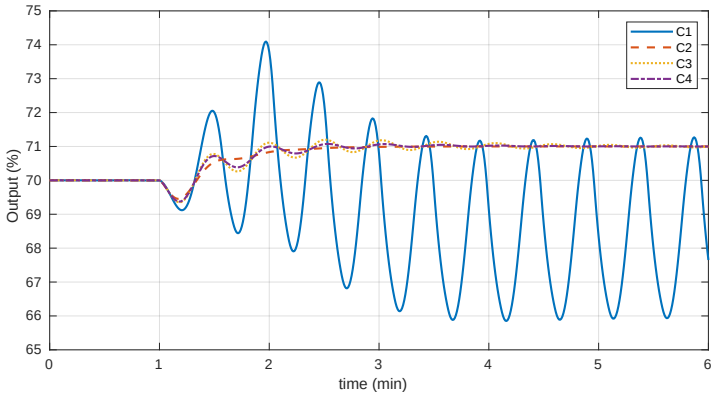
Now consider the controllers that took into account the robustness as a constraint in the optimization. For all cases, the controller is able to control the plant without oscillation (except for controller C1). As expected, controllers C2 and C4 had the best performance, but also the were more expensive (higher values of TV). However, an interesting option is controller C3. This controller was designed by allowing a 20% degradation of J_{di}. Its servo response may not be the best in terms of performance but it has an interesting compromise between the control effort and the



(a) $M_s = 2.0$

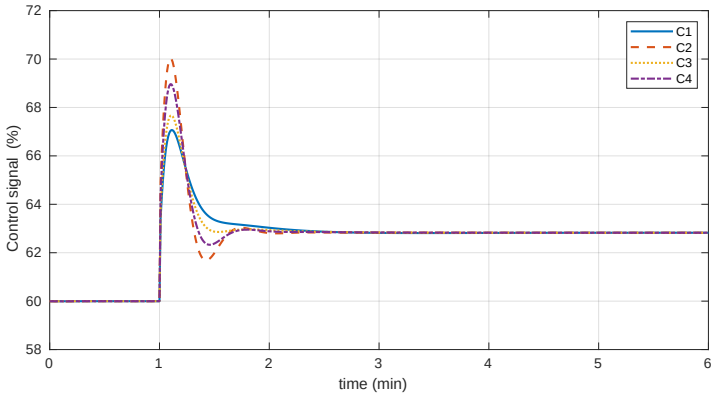


(b) $M_s = 1.8$

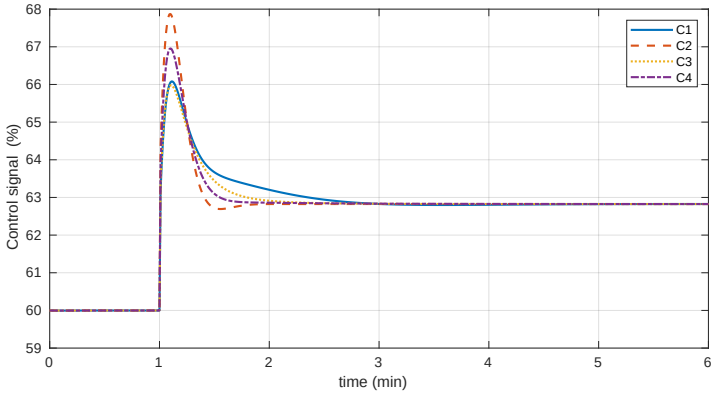


(c) $M_s \geq 2$

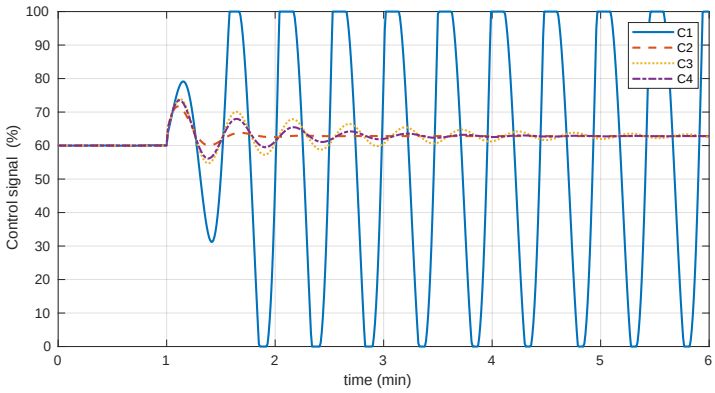
Fig. 1.27: Response of the controlled system with the nonlinear model for several robustness levels serving as servo.



(a) $M_s = 2.0$



(b) $M_s = 1.8$



(c) $M_s \geq 2$

Fig. 1.28: Control signal of the controlled system with the nonlinear model for several robustness levels serving as servo.

Table 1.8: IAE values for the controllers applied to the nonlinear model as regulator controllers

Robustness	Controller			
	C1	C2	C3	C4
$M_s = 1.8$	2.29	3.40	2.54	2.85
$M_s = 2.0$	2.03	3.24	2.26	2.65
$M_s \geq 2.0$	11.62	3.18	1.97	2.00

Table 1.9: TV values for the controllers applied to the nonlinear model as regulator controllers

Robustness	Controller			
	C1	C2	C3	C4
$M_s = 1.8$	13.30	11.85	12.50	11.65
$M_s = 2.0$	13.68	15.54	13.16	14.10
$M_s \geq 2.0$	3034.50	22.00	183.90	84.00

IAE value. If this degradation does not affect the regulator response too much, this controller may be considered as the final tuning.

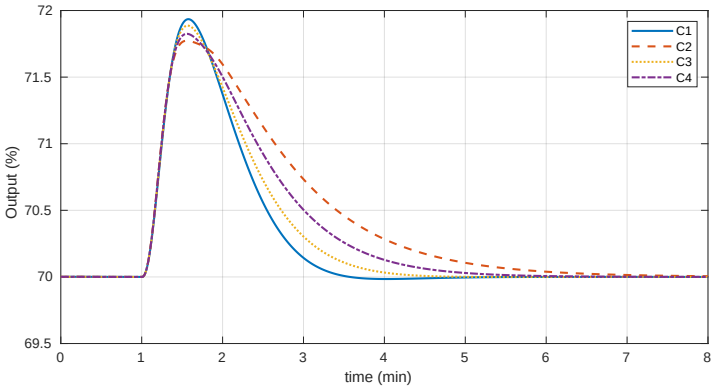
The next step is then to validate the response as a regulator. In Table 8.8 the values of IAE are presented for a step change of $1\,\mathrm{mol\,L^{-1}}$ in C_{ai} for all controller cases and robustness values. The corresponding values of TV are presented in Table 8.9. The plots of the responses and the control signal are shown in Figure 8.29 and Figure 8.30 respectively. As in the case of the servo response, the design for $M_s \geq 2.0$ is not useful in the C1 case, because the response is practically unstable again as depicted in Figure 8.29c. It has to be noticed that the response of the plant between C_{ai} and the output does not have a non-minimum phase zero, and therefore an inverse response is not present.

For the other cases ($M_s = 2.0$ and $M_s = 1.8$) the best controllers were the C1 family, followed by C3. This is expected since these controllers were found as the best regulators. Controller C2 has the worst regulator response but at the same time, it as also the most expensive cost signal for the case $M_s = 2.0$.

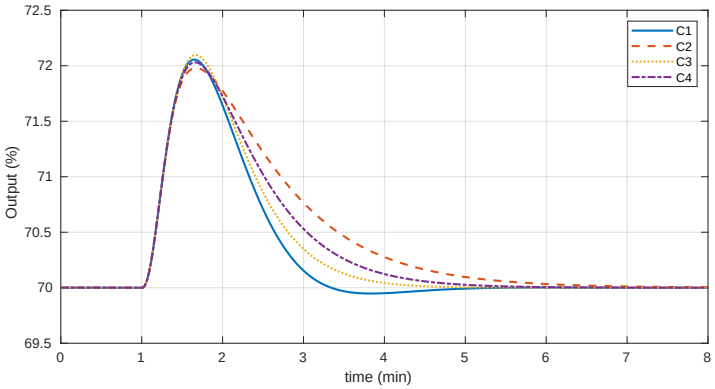
Let's examine C3 controllers. The performance of these controllers are worse than C1, but just by approximately 11%, and with a less aggressive control signal for the regulator response (Table 8.9). Taking into account that C3 was also a family of controllers that had a relatively good performance for servo control, so far it represents a good candidate to become the final controller.

As a final validation test, let the setpoint change to be 5%. In that case the response is as given in Figure 8.31 for all controllers families and $M_s = 2.0$.

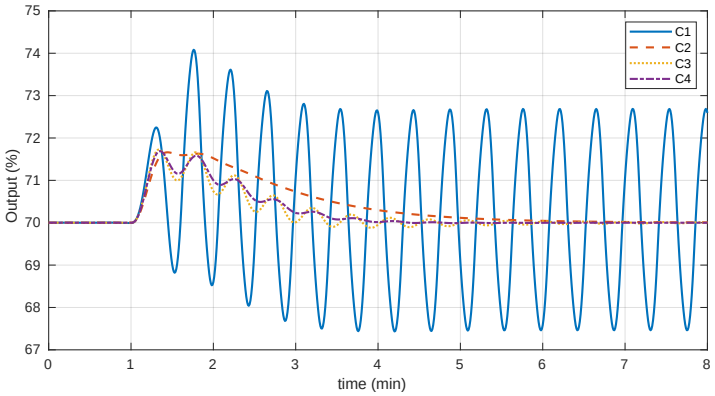
The values of IAE and TV are presented in Table 8.10. The PID was implemented to be limited in the range between 0 and 100%, with the corresponding antiwindup. Since controllers C2 and C4 have a control signal which is more aggressive for set-



(a) $M_s = 2.0$

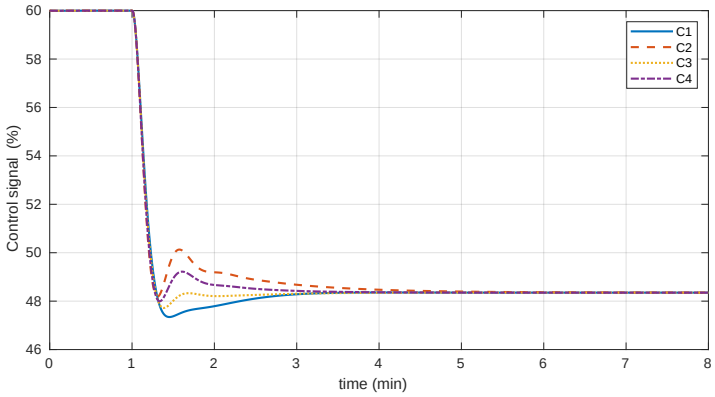


(b) $M_s = 1.8$

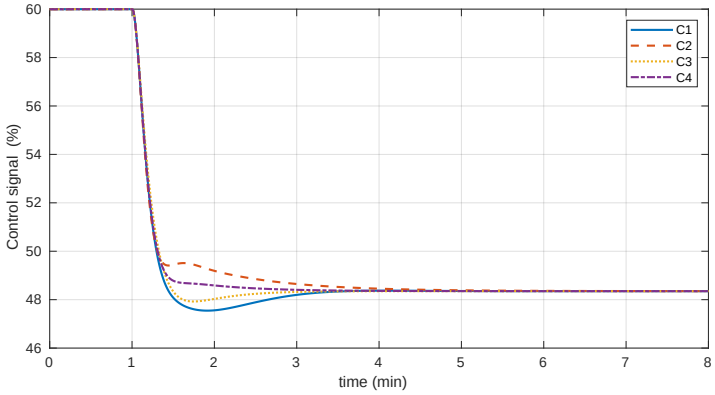


(c) $M_s \geq 2$

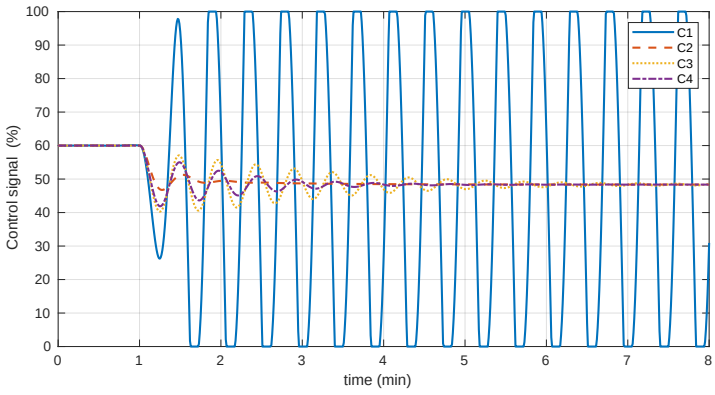
Fig. 1.29: Response of the controlled system with the nonlinear model for several robustness levels serving as regulator.



(a) $M_s = 2.0$

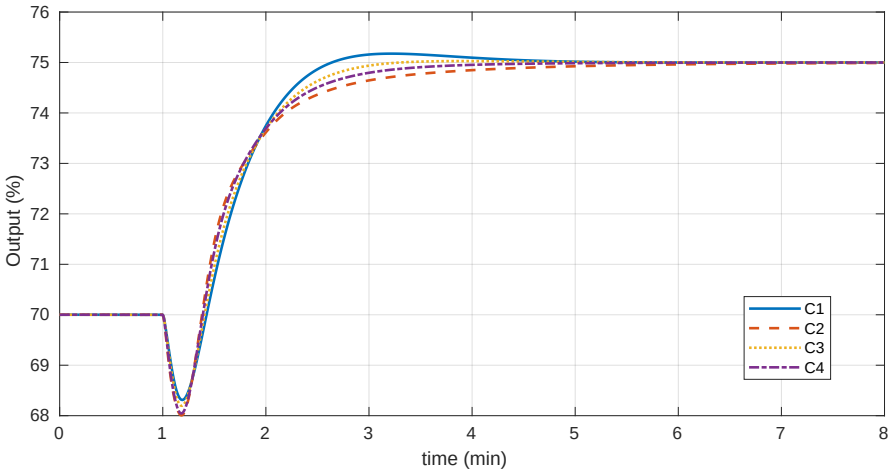


(b) $M_s = 1.8$

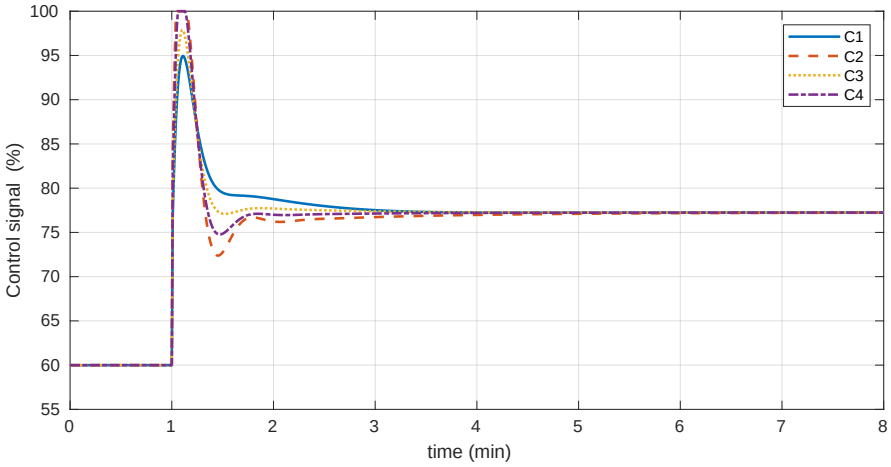


(c) $M_s \geq 2$

Fig. 1.30: Control signal of the controlled system with the nonlinear model for several robustness levels serving as regulator.



(a) Output signal



(b) Control signal

Fig. 1.31: Response to a 5% step change in the setpoint signal.

point changes, they saturate and produces larger values of IAE than the controllers intended for regulation. Interestingly, C3 controller has a lower value of IAE than any other controller while its TV value lies in between C1 and C4 controllers. Again, this test indicates that C3 controller can be a good candidate for the final tuning of the PID.

Lastly, let us again set the change in the setpoint to 5%, but lets compare the difference when C3 is selected as the controller and the M_s is varied. In Figure 8.32 the output of the controlled loop and the control signal is presented and in Table 8.11, the values of IAE and TV are presented for both $M_s = 1.8$ and $M_s = 2.0$. As it

Table 1.10: IAE values for the controllers applied to the nonlinear model as servo controller for a step change of 5% and $M_s = 2.0$

Cost function	Controller			
	C1	C2	C3	C4
IAE	4.69	5.07	4.60	4.71
TV	52.70	73.44	59.81	68.01

Table 1.11: IAE values for the C3 controllers applied to the nonlinear model as servo controller for a step change of 5%

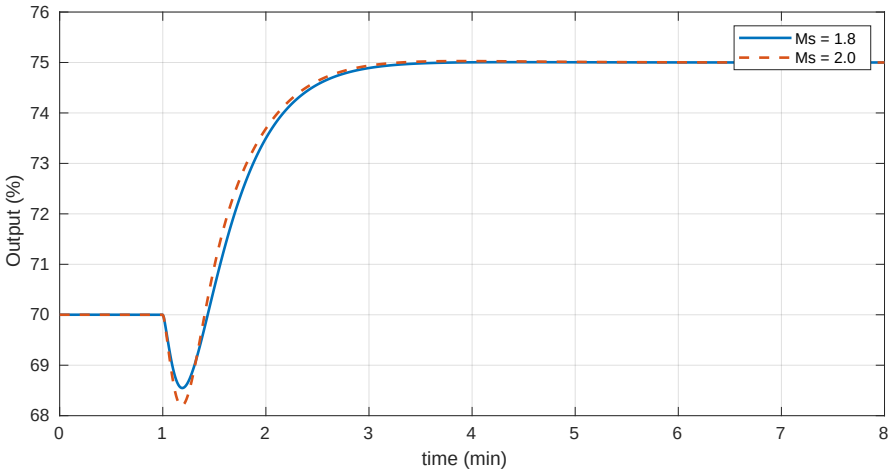
Cost function	M_s	
	1.8	2.0
IAE	4.83	4.61
TV	42.11	59.81

was expected, since neither of those controllers saturates, the response with a less restrictive robustness constraint has a better IAE. However it certainly has a more aggressive control signal. The IAE for the case $M_s = 2.0$ is 4.55% lower than the $M_s = 1.8$ case however it comes to a cost of having a 42.03% higher TV. Considering all the analysis done to this point and having into account that the tuning were made with several approximations from the original model, it seems that the C3 tuning (best servo allowing a 20% degradation on J_{di}) with a robustness constraint of $M_s = 1.8$ is the most sensible choice as the final tuning.

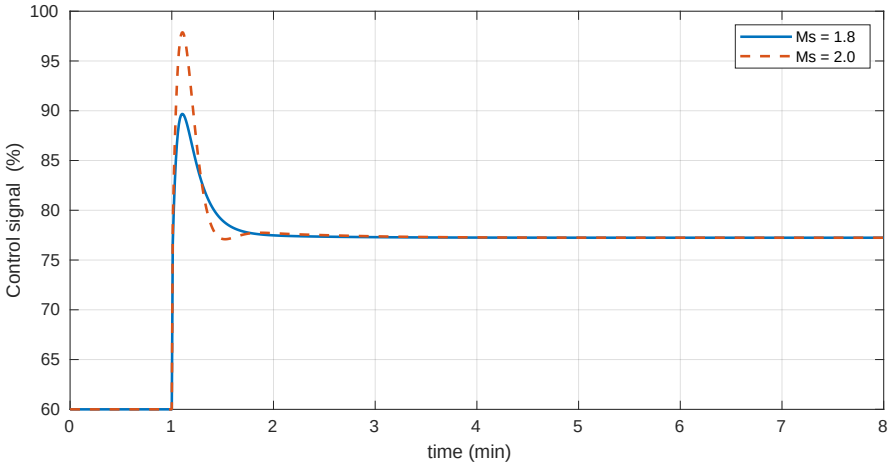
1.3 Final remarks

The motivation of the examples presented in this chapter was to show all the advantages that can be derived from using a multiobjective approach when tuning a PID controller for industrial applications. As has been shown,, the final selection of the parameters was defined not only by its optimal value, but also, according to the robustness needs and the level of compromise between the cost functions.

However, it has to be noticed that the cost functions selected for these studies are totally arbitrary and other authors may choose to optimize the tuning of the parameters with respect to other criteria. For example, in Section 5.1.2, the total variation was selected as one of the cost functions and the Pareto found was considerably different from the ones found using J_{di}, J_{do} and J_r. But it is known that the IAE is a practical measure of the optimality of the control in industry (**?**), and for this reason was the selected cost function for the tool presented in this book and consequently the examples examined in this chapter.



(a) Output signal



(b) Control signal

Fig. 1.32: Response to a 5% step change in the setpoint signal for C3 controllers.

Apart from the theoretical contribution of this book, the MATLAB tool that is included along with the complete set of data, represents an interesting starting point for further studies. Of course the methodology presented in 7.1 is completely general and can (and is encouraged to) be changed to the needs of the decision maker.

One of the most interesting characteristics of control systems is that some kind of compromise is always involved. The relationship between servo and regulation control, or between performance and aggressiveness is always something that has to be taken into account along with the need to have a robust control system that is able to keep working despite the difference between the model and the actual

plant. It is the desire of the authors to have helped in the development of a deeper understanding on this issues, and to open the door to more research in the field of optimization applied to industrial control.

References

Alfaro VM, Vilanova R (2016) Model-Reference Robust Tuning of PID Controllers. Advances in Industrial Control, Springer International Publishing, Cham, DOI 10.1007/978-3-319-28213-8, URL http://link.springer.com/10.1007/978-3-319-28213-8

Arrieta O, Visioli A, Vilanova R (2010) PID autotuning for weighted servo/regulation control operation. Journal of Process Control 20(4):472–480, DOI 10.1016/j.jprocont.2010.01.002

Murrill P (1967) Automatic control of processes. International Textbook Co.

O'Dwyer A (2009) Handbook of PI and PID Controller Tuning Rules, 3rd edn. Imperial College Press, London, UK

Rovira AJA, Murrill P, Smith CL (1969) Tuning controllers for setpoint changes. Instruments and Control Systems 42:67–69

Shinskey FG (2002) Process Control: As Taught vs as Practiced. Industrial & Engineering Chemistry Research 41(16):3745–3750, DOI 10.1021/ie010645n, URL https://pubs.acs.org/doi/10.1021/ie010645n

Skogestad S (2003) Simple analytic rules for model reduction and PID controller tuning. Journal of Process Control 13(4):291–309, DOI 10.1016/S0959-1524(02)00062-8, URL https://www.sciencedirect.com/science/article/pii/S0959152402000628 https://linkinghub.elsevier.com/retrieve/pii/S0959152402000628

Van de Vusse J (1964) Plug-flow type reactor versus tank reactor. Chemical Engineering Science 19(12):994–996, DOI 10.1016/0009-2509(64)85109-5, URL https://linkinghub.elsevier.com/retrieve/pii/0009250964851095