NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for naturvitenskap og teknologi
Institutt for kjemisk prosessteknologi

SPECIALIZATION PROJECT 2012

TKP 4550

PROJECT TITLE:

Performance and Robustness of the Smith Predictor Controller

By
Helene Paulsen

Supervisors for the project: Sigurd Skogestad, Vinicius De Oliveira

Date: 7.12.2012

Abstract

The Smith predictor controller and PI controller was simulated with different time delays and the resulting performance and robustness was compared. A P-only controller was tried as the primary controller in the Smith Predictor structure, which resulted in a discontinuous stability domain for a range of time delay values. Robust tuning rules were then used for the Smith Predictor, while SIMC tuning rules were used for the PI controller. This gave better performance for the Smith Predictor, by comparing the integral squared error for the two controllers. Higher controller gains were tried for the Smith predictor, while SIMC tuning rules were still used for the PI controller. This resulted in an unstable Smith Predictor for tight control.

Optimisation of the Smith Predictor and the PI controller in terms of performance and robustness gave the optimal tuning parameters for each controller in two different cases, and a trade-off between performance and robustness was obtained. The Smith Predictor achieved a better performance for all values of M_s in both cases. The optimal tuning parameters from the optimisation were also tested in the model in Simulink. To see the results, the IAE values for the two controllers were plotted against the deviation from the nominal time delay. This resulted in a better performance for the SP controller after a certain value of the time delay was exceeded.

In general the Smith Predictor is better than the PI controller when the time delay is large enough. But if the controller gain in the Smith Predictor is high enough, the controller will have discontinuous stability for the whole time delay range.

Preface

I would like to thank my supervisors Sigurd Skogestad and Vinicius De Oliveira for all their help during this project. In addition I want to thank other students for any input they have had.

Contents

Abstract i

Preface ii

1. Introduction 1

2. Theory 2

  2.1. PI Control 2

  2.2. Smith Predictor Controller 4

    2.2.1. Derivation of Transfer Function for Inner Feedback Loop 5

    2.2.2. Robust Tuning of the Smith Predictor 6

  2.3. SIMC Tuning 7

  2.4. Integral Squared Error 7

  2.5. Trade-off between Performance and Robustness 8

3. Simulation Procedure 9

  3.1. Verification of Example from an Article 9

  3.2. PI Controller for the SP 10

  3.3. Smith Predictor with Variation in Controller Gain 11

  3.4. Robust Tuning of SP and SIMC Tuning of PI 11

  3.5. Optimisation of Performance and Robustness 13

  3.6. Verification of Optimisation in Simulink 13

4. Results 14

  4.1. Verification of Example from an Article 14

  4.2. PI Controller for the SP 16

  4.3. Robust Tuning of SP and SIMC Tuning of PI 17

    4.3.1. Set-point Tracking 17

    4.3.2. Disturbance Rejection 19

  4.4. Smith Predictor with Increasing Controller Gain 21

  4.5. Optimisation of Performance and Robustness 26

    4.5.1. Performance versus Robustness 26

    4.5.2. Optimal Tuning 28

  4.6. Verification of Optimisation in Simulink 30

5. Discussion 32

6. Conclusions 33

7. Symbols and Abbreviations 34
7.1. List of symbols 34
7.2. List of Abbreviations 35
8. References 36

Appendix A – Matlab and Simulink files 37
Appendix B – Calculation of Tuning Values 38
Appendix C – Plots from Verification of Example 39
Appendix D – Plots From the Use of a PI Controller as the Primary Controller 43
Appendix E – Optimal Tuning versus Robustness Plots 47

List of Figures

Figure 1: Block diagram of a control system with a PI controller [6]. 2

Figure 2: Block diagram for a process with the Smith predictor controller [6]..................................... 4

Figure 3: The Smith Predictor block diagram re-drawn as two nested feedback loops [6]. 5

Figure 4: The Smith Predictor block diagram drawn in reduced form [6]... 5

Figure 5: Block diagram of the Smith predictor and PI controller created in Simulink. The top block diagram is the Smith predictor structure and the bottom one is for a process with PI control. 10

Figure 6: Block diagram of the process with the change in set-point and disturbance included. The top block diagram is the Smith Predictor controller, and the bottom one is the PI controller.................. 12

Figure 7: Output of the Smith Predictor and a PI controller where the time delay is equal to θ=1.33. A P-only controller is used as the primary controller in Smith Predictor control structure.................... 15

Figure 8: Output of the Smith Predictor and a PI controller where the time delay is equal to θ=1.78. A P-only controller is used as the primary controller in Smith Predictor control structure.................... 15

Figure 9: Integral squared error plotted against the error in time delay with robust tuning of SP and SIMC tuning of the PI controller for set-point tracking. 18

Figure 10: Integral square error plotted against the error in time delay with robust tuning of SP and SIMC tuning of the PI controller for disturbance rejection. 20

Figure 11: ISE values plotted against the error in delay (θ-θ_n) for set-point tracking when Kc = 2. 21

Figure 12: ISE values plotted against the error in delay (θ-θ_n) for set-point tracking when Kc = 3. 22

Figure 13: ISE values plotted against the error in delay (θ-θ_n) for set-point tracking when Kc = 4. 22

Figure 14: ISE values plotted against the error in delay (θ-θ_n) for disturbance rejection when Kc = 2. 23

Figure 15: ISE values plotted against the error in delay (θ-θ_n) for disturbance rejection when Kc = 3. 24

Figure 16: ISE values plotted against the error in delay (θ-θ_n) for disturbance rejection when Kc = 4. 25

Figure 17: Trade-off between performance (J(c)) and robustness (M_s) for the Smith predictor controller and a PI controller for the process g=exp(-s)/(s+1). The process is optimised in terms of performance and robustness. 26

Figure 18: Trade-off between performance (J(c)) and robustness (M_s) for the Smith predictor controller and a PI controller for the process g=exp(-s)/(s+1). The process is optimised in terms of performance and robustness. 27

Figure 19: IAE plotted against the delay error for case 1 with the process g=exp(-s)/(s+1) when the optimal tuning is applied on both controllers. 30

Figure 20: IAE plotted against the delay error for case 2 with the process g=exp(-s)/(8s+1) when the optimal tuning is applied in both controllers. 31

Figure 21: Optimal tuning for the Smith predictor with the process g = e^{-s}/(s+1). 47

Figure 22: Optimal tuning for the PI controller with the process g = e^{-s}/(s+1). 47
Figure 23: Optimal tuning for the SP controller with the process g = e^{-s}/(8s+1). 48
Figure 24: Optimal tuning for the PI controller with the process g = e^{-s}/(8s+1). 48

List of Tables

Table 1: Description of the different blocks in a control system with a PI controller as shown above. . 2

Table 2: Robust tuning of the Smith predictor when using $\tau_0 = 1.7\Delta\theta_{max}$ [7]. 7

Table 3: Parameters used in the article [10]. 9

Table 4: Tuning values for the Smith Predictor and the PI controller with robust- and SIMC tuning rules, respectively. 11

Table 5: The processes that were optimised in terms of performance and robustness. 13

Table 6: Stability domain made from the simulation of a P-only controller as the primary controller in the Smith predictor control structure. 14

Table 7: Stability table made from the simulation of a PI controller as the primary controller in the Smith predictor control structure. 16

Table 8: Integral squared error results for different time delays for the PI controller and the Smith predictor controller for set-point tracking. SIMC tuning rules were used for the PI controller, while robust tuning rules were used for the SP. 17

Table 9: Integral squared error results for different time delays for the PI controller and the Smith Predictor for disturbance rejection. SIMC tuning rules were used for the PI controller, while robust tuning rules were used for the SP. 19

Table 10: Robustness, controller gain, integral time and performance results from optimisation of both the SP and PI controller for the process g=exp(-s)/(s+1). 28

Table 11: Robustness, controller gain, integral time and performance results from optimisation of both the SP and PI controller for the process g=exp(-s)/(8s+1). 28

Table 12: Optimal tuning parameters for a robustness of M_s = 1.7 for the Smith Predictor and PI controller. 29

1. Introduction

The purpose of this project is to test the robustness and performance of a Smith predictor (SP) controller for processes with time delays. This will be performed by comparing the Smith predictor controller with a PI controller. It is desirable to see how the performance and robustness of the Smith predictor depends on the variation in the time delay. Time delays can occur in industrial processes due to the presence of distance velocity lags, recycle loops and the analysis time associated with composition measurements [1]. The presence of such time delays will reduce the performance of a conventional feedback control system, thus there is a need for time-delay compensation.

Matausek and Micic (1996) tried to control higher order processes with integral action and long dead-time by using a modified Smith predictor [2]. The model consists of an ideal integrator and dead-time, and includes three tuning parameters. These are the dead-time, velocity gain of the model, and the desired time constant of the first order closed-loop set-point response. This was found to give a high possible performance for the set-point response and also for the load disturbance rejection.

Lee and Wang (1996) have also obtained sufficient conditions for the practical stability, robust stability and the robust performance for the Smith predictor controller [3]. A norm-bounded uncertainty is associated with the process model. Lee and Wang (1996) developed a two-step controller design approach based on the discovered conditions.

Lee et al. (1998) have looked at robust PID tuning for the Smith predictor in the presence of model uncertainty [4]. The equivalent gain plus time delay (EGPTD) was used to approximate the tuning method. By using the EGPTD, it was possible to achieve relatively good approximation over the important frequency range. This robust tuning method is simple and easy, but the robustness is still ensured.

Majhi and Atherton (1999) proposed a modified Smith predictor with a simple and effective controller design procedure [5]. This was developed for time delay processes. High performance for the controller was achieved, particularly for an integrating and unstable process. In addition, the predictor was able to successfully control a stable process and an integrator with a long dead-time. Simple tuning formulas were used.

2. Theory

A general plant can be described by a first order plus time delay (FOPTD) transfer function as shown in equation (1).

$$G(s) = \frac{K_p}{\tau s+1} e^{-\theta s} \tag{1}$$

In the transfer function above, K_p is the gain of the process, τ is the process time constant and θ is the time delay or dead time.

2.1. PI Control

A control system with proportional and integral control (PI) is shown in Figure 1.

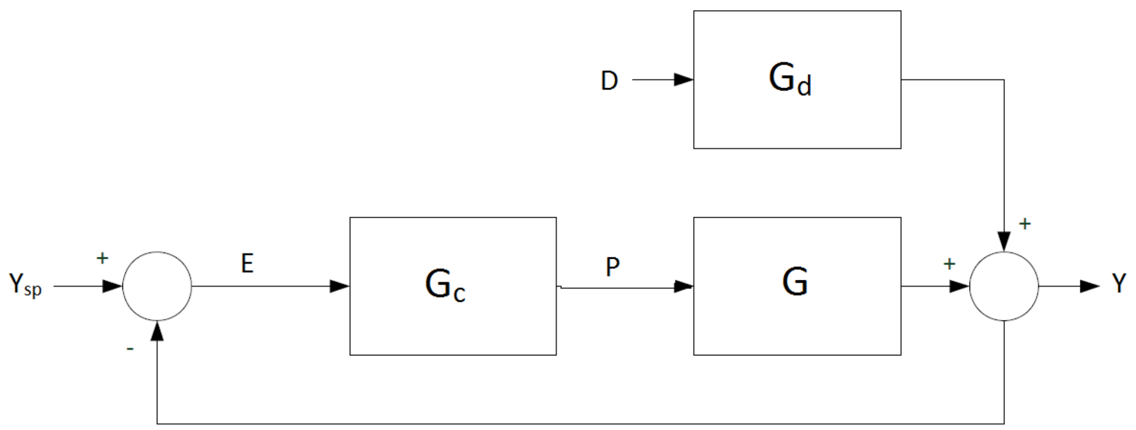


Figure 1: Block diagram of a control system with a PI controller [6].

Table 1 gives an overview of what the different notations in Figure 1 represent.

Table 1: Description of the different blocks in a control system with a PI controller as shown above.

Parameter	Definition
Y_{sp}	Set-point of the process
E	Error between set-point and output of process
G_C	Controller transfer function
P	Output of controller
G	Process transfer function
Y	Output of the process
D	Disturbance
G_d	Disturbance transfer function

For a PI controller, the controller output is given by equation (2) in the time domain [1]:

$$p(t) = \bar{p} + K_c\left(e(t) + \frac{1}{\tau_I}\int_0^t e(t^*)dt^*\right) \tag{2}$$

The corresponding transfer function in the Laplace domain is given by:

$$\frac{P\prime(s)}{E(s)} = K_c\left(1 + \frac{1}{\tau_I s}\right) = K_c\left(\frac{\tau_I s+1}{\tau_I s}\right) \tag{3}$$

The closed-loop set-point transfer function for the feedback control in Figure 1 is obtained from the following derivation, where the disturbance is assumed to be zero:

$$Y = GP \tag{4}$$

$$P = EG_c \tag{5}$$

$$E = Y_{sp} - Y \tag{6}$$

By inserting equation (5) and (6) into equation (4), the equation becomes

$$Y = G_c G(Y_{sp} - Y) \tag{7}$$

Collecting terms with Y on the left side and terms with Y_{sp} on the right side:

$$Y(1 + G_c G) = Y_{sp} G_c G \tag{8}$$

This gives the closed-loop transfer function for set-point changes:

$$\frac{Y}{Y_{sp}} = \frac{G_c G}{1+G_c G} \tag{9}$$

When the process has a time delay, the process transfer function is given by

$$G = G_n e^{-\theta s} \tag{10}$$

Finally the closed-loop set-point transfer function is given as:

$$\frac{Y}{Y_{sp}} = \frac{G_c G_n e^{-\theta s}}{1+G_c G_n e^{-\theta s}} \tag{11}$$

2.2. *Smith Predictor Controller*

Smith predictor control is one of several strategies that have been developed to improve the performance of systems containing time delays [1]. It is desirable to achieve effective time-delay compensation with these methods. Previous experiments have shown that the Smith predictor gives a better performance for set-point changes than a conventional PI controller [1]. This was proved by comparing the integral-squared error criterion. On the other hand, this may not be the case for all types of disturbances.

There are not only benefits associated with the Smith predictor controller. It has a disadvantage due to the fact that a dynamic model of the process is required [1]. This can result in an inaccurate predictive model and reduced controller performance if the process dynamics are considerably changed.

A block diagram of the Smith predictor controller is shown Figure 2.

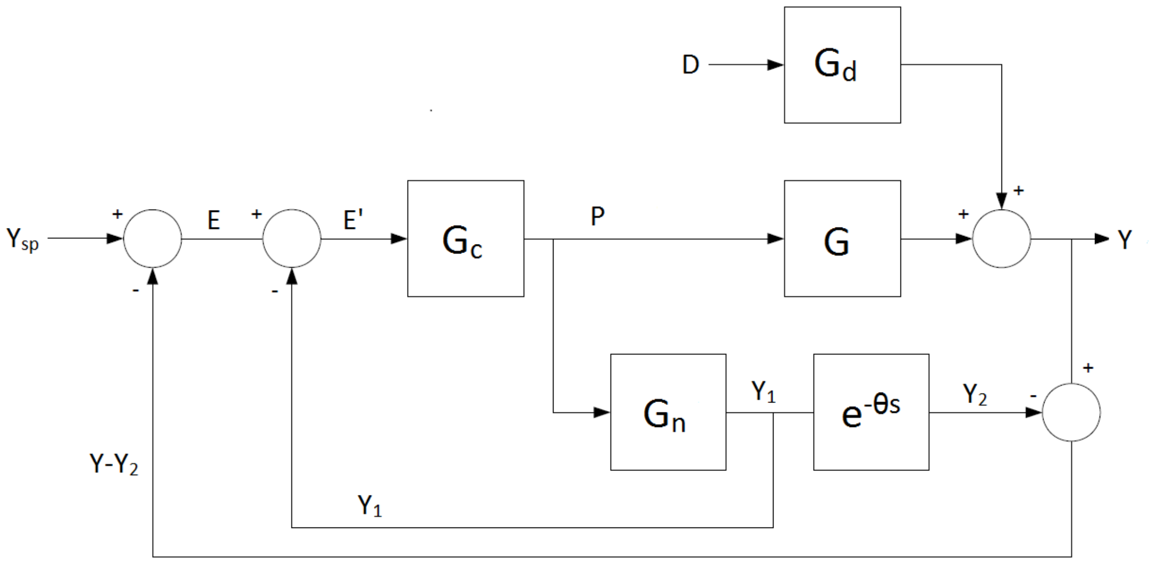


Figure 2: Block diagram for a process with the Smith predictor controller [6].

As seen in Figure 2, the Smith predictor structure can be divided into two parts. This includes the primary controller, $G_C(s)$, and the predictor structure. The predictor part consists of a model of the plant without time delay ($G_n(s)$), and a model of the time delay ($e^{-\theta s}$) [7]. The parameter θ is the time delay. Thus, the complete process model is given by equation (12).

$$P_n(s) = G_n(s)e^{-\theta s} \tag{12}$$

The model without the dead-time is sometimes called the fast model and is used to compute an open-loop prediction. Thus, the model of the process without time delay (G_n) is used to predict the effect of control actions on the undelayed output [1]. Then the controller uses the predicted response (Y_1) to calculate its output signal (P). The actual undelayed output (Y) is compared with the delayed predicted output (Y_2). In the case of no disturbances or modelling errors, the difference between the process output and the model output will be zero. This means that the output signal (Y-Y_2) will be equal to the output of the plant without any time delay [7].

2.2.1. Derivation of Transfer Function for Inner Feedback Loop

The block diagram for the Smith predictor given in Figure 2 can be re-drawn as two nested feedback loops as shown in Figure 3 [6].

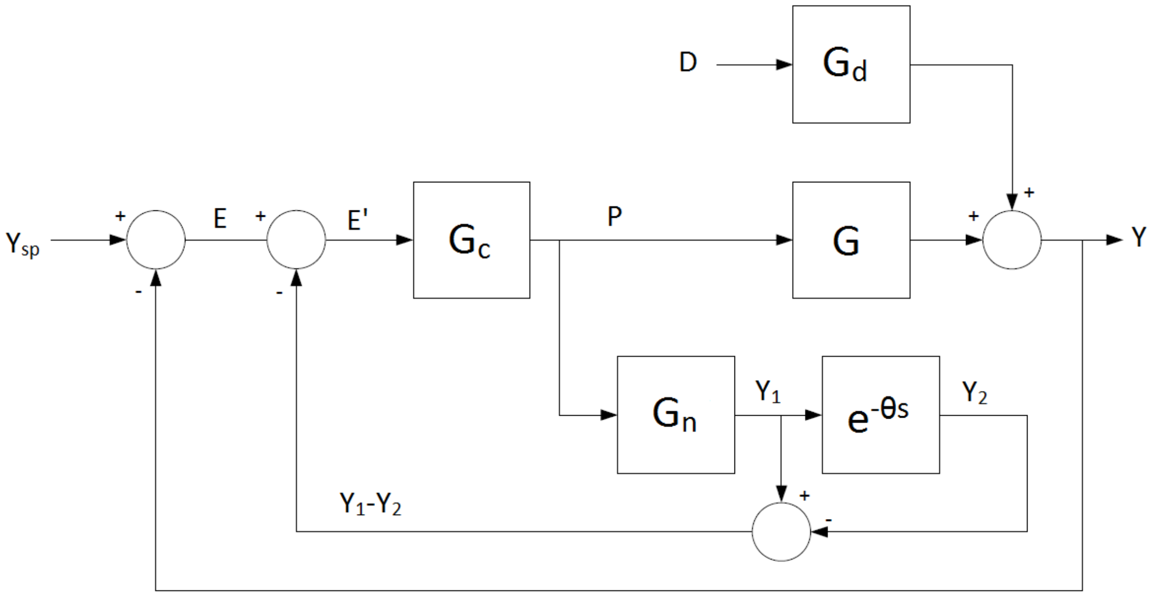


Figure 3: The Smith Predictor block diagram re-drawn as two nested feedback loops [6].

Further on, the inner feedback loop can be reduced as shown in Figure 4.

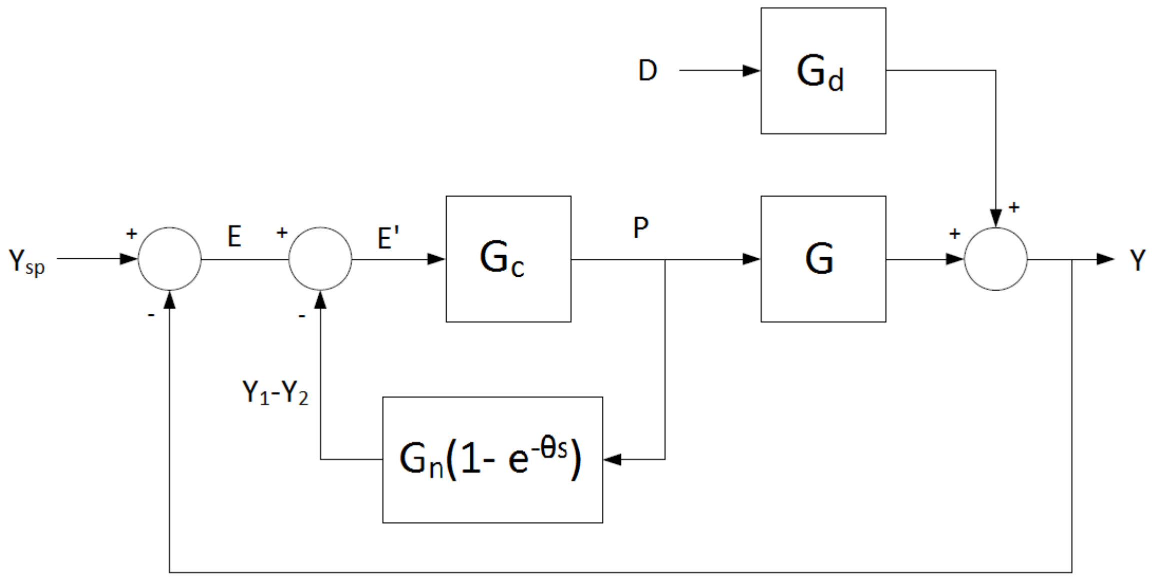


Figure 4: The Smith Predictor block diagram drawn in reduced form [6].

To find the equivalent transfer function for the inner feedback loop in Figure 4, an expression for P/E is needed. This function is derived by using the blocks in Figure 4. The goal is to find G', given by:

$$G' = \frac{P}{E} \tag{13}$$

Investigation of the block diagram in Figure 4 gives the following:

$$P = E'G_c \tag{14}$$

$$E' = E - (Y_1 - Y_2) \tag{15}$$

$$(Y_1 - Y_2) = PG_n(1 - e^{-\theta s}) \tag{16}$$

By inserting equation (15) and (16) into (14), the equation becomes:

$$P = [E - PG_n(1 - e^{-\theta s})]G_c \tag{17}$$

Simplifying equation (17) gives

$$P[1 + G_cG_n(1 - e^{-\theta s})] = EG_c \tag{18}$$

The transfer function for the inner feedback loop in Figure 4 is then obtained:

$$G' = \frac{P}{E} = \frac{G_c}{1+G_cG_n(1-e^{-\theta s})} \tag{19}$$

By doing some rearrangement, the closed-loop set-point transfer function for the Smith predictor is given by:

$$\frac{Y}{Y_{sp}} = \frac{G_cG_ne^{-\theta s}}{1+G_cG_n} \tag{20}$$

By comparing equation (11) and equation (20), it is observed that the time delay part is missing from the characteristic equation of the Smith predictor controller, but it is present in the closed-loop set-point transfer function of the PI controller. The characteristic equation is the denominator part of the closed-loop set-point transfer function for a given controller. This gives a theoretical explanation of the time delay compensating ability of the Smith predictor.

2.2.2. Robust Tuning of the Smith Predictor

Normey-Rico and Camacho (2007) have proposed a robust tuning procedure for the Smith predictor controller [7]. This method is supposed to ensure robust performance of the Smith predictor controller for a certain range of time delays. By considering only dead-time-estimation errors, robust tuning of the Smith predictor can be obtained for processes with dominant dead-time [7].

Under these conditions, the dead-time-error dominates any other errors in the process. A tuning rule for achieving both robust stability and robust performance was found, and the condition for this is given by equation (21):

$$\tau_0 = 1.7\Delta\theta_{max}, \quad (21)$$

where τ_0 is the desired time constant. The suggested tuning rules are given in Table 2.

Table 2: Robust tuning of the Smith predictor when using $\tau_0 = 1.7\Delta\theta_{max}$ [7].

Predictor model	Controller	Filter	K_c	τ_I	τ_1
$\frac{K_p e^{-\theta s}}{1+s\tau}$	$\frac{K_c(1+\tau_I s)}{\tau_I s}$	$\frac{(1+\tau_0 s)}{1+\tau_1 s}$	$\frac{\tau}{K_p\tau_0}$	τ	$\tau_1 \epsilon [\tau, \tau_0]$

In the tuning rules in Table 2, K_p is the process gain, τ is the time constant of the process, θ is the dead time, K_c is the controller gain, τ_I is the integral time, τ_0 is the desired time constant and the nominal set-point response is defined by the parameter τ_1.

2.3. SIMC Tuning

According to Skogestad and Grimholt (2011), the original SIMC rules for a first order model are given by equation (22) and equation (24) for a first order process with a PI controller [8].

$$K_c = \frac{1}{k}\frac{\tau_1}{\tau_c+\theta} = \frac{1}{k'}\frac{1}{\tau_c+\theta} \quad (22)$$

In the above equation, k' is given by

$$k' = \frac{k}{\tau_1} \quad (23)$$

$$\tau_I = \min[\tau_1, 4(\tau_c+\theta)] \quad (24)$$

The only tuning parameter, τ_c, is the desired first-order closed-loop time constant [8]. Further on, K_c is the controller gain, τ_1 is the process time constant, k is the steady-state gain, τ_I is the integral time and θ is the time delay of the process.

2.4. Integral Squared Error

To compare the performance between two controllers, the integral squared error can be calculated. This error is given by equation (25):

$$ISE = \int_0^\infty \left[\left[y(t) - y_{set-point}(t)\right]^2 dt\right], \quad (25)$$

where $y(t)$ is the output of the controller for the time delay i, and $y_{set-point}$ is the set-point of the process.

2.5. Trade-off between Performance and Robustness

To compare the performance and robustness of the two different controllers, there is a need to quantify the performance and robustness. As seen before, the performance can be quantified by the integral squared error (ISE), but the integral absolute error (IAE) can also be used [9]. This is defined as

$$IAE = \int_0^\infty |y(t) - y_s(t)|dt, \quad (26)$$

where $y(t)$ is the output measurement and $y_s(t)$ is the set-point of the output. Other methods to quantify the performance are also available, but the IAE will be used in this project. Grimholt and Skogestad (2011) have used a weighted average of IAE for a step change in the load disturbance d and set-point y_s [9]. This is because the value of IAE depends on which set-point or disturbance change that is occurring. The weighted IAE is given by

$$J(c) = 0.5\left[\frac{IAE_{y_s}(c)}{IAE_{y_s}^0} + \frac{IAE_d(c)}{IAE_d^0}\right], \quad (27)$$

where IAE_{y_s} and IAE_d are the integrated absolute error for a step change in the set-point and load disturbance, respectively.

In this case, J can be interpreted as the weighted cost of the controller, and the goal is to minimize J(c). Also, this cost is independent of the set-point and disturbance magnitudes, the process gain K_p, and the unit used for time [9].

Since the Smith predictor controller is being compared to a PI controller, the weighting factors $IAE_{y_s}^0$ and IAE_d^0 are PI controllers in the Smith predictor structure. They are IAE-optimal for a certain process, and they need to have a given value of the robustness (M_s) to get robust reference controllers [9].

In order to quantify the robustness of the controllers, the sensitivity peak (M_s) will be used. This is given by

$$M_s = \max_\omega \left|\frac{1}{1+gc(j\omega)}\right|, \quad (28)$$

which is the maximum or the peak of the sensitivity function in the frequency domain [9]. When it comes to quantifying the robustness, M_s is the inverse of the closest distance between the critical point at -1 and the loop transfer function gc in the Nyquist plot. Thus, a small value of M_s is desirable to get robustness, because the closest distance between the critical point and the loop transfer function increases when M_s decreases.

3. Simulation Procedure

In all the simulations in this project, the nominal time delay (θ_0) is equal to 1, and the simulations were performed for a FOPTD, as given in equation (1). All the Matlab scripts and Simulink files used in the project are given on a CD in Appendix A.

3.1. *Verification of Example from an Article*

Adam et al. (2000) have performed an experiment regarding the robustness of the Smith predictor with respect to uncertainty in the time-delay parameter [10]. This procedure was reproduced to verify the results that were obtained. The model in equation (1) was used, and the parameters of the undelayed part were as shown in Table 3.

Table 3: Parameters used in the article [10].

K_p	τ	c(s)
1	1	4

This leads to the process transfer function given by equation (29).

$$G(s) = \frac{1}{s+1}e^{-s} \tag{29}$$

As seen from the table above, the controller is a P-only controller with a controller gain equal to K_c=4. The simulation of the example was performed in Matlab and Simulink, and the Matlab script and Simulink file is named *'Robustness_SP_P.m'* and *'SP_and_PI_withP.mdl'*, respectively. The block diagram created in Simulink is shown in Figure 5. In addition to the tuning parameters in Table 3, the solver in the configuration parameters was set to be ode15s (stiff/NDF) and the relative tolerance was set to 1e-6.

After the simulation, the output of the Smith predictor (Y_{SP}) and the output of the PI controller (Y_{PI}) were plotted against time in the same figure for a range of time delays different from the nominal delay, that is $0<\theta<4$. This was done to check if the simulation resulted in the same stability regions for the Smith predictor as in the given example [10]. The output of the PI controller was plotted in order to compare it with the Smith predictor.

SIMC tuning rules were used for the PI controller structure because of simplicity and because they have proven to be good tuning rules. The calculation of the SIMC tuning parameters for the PI controller is given in Appendix B.

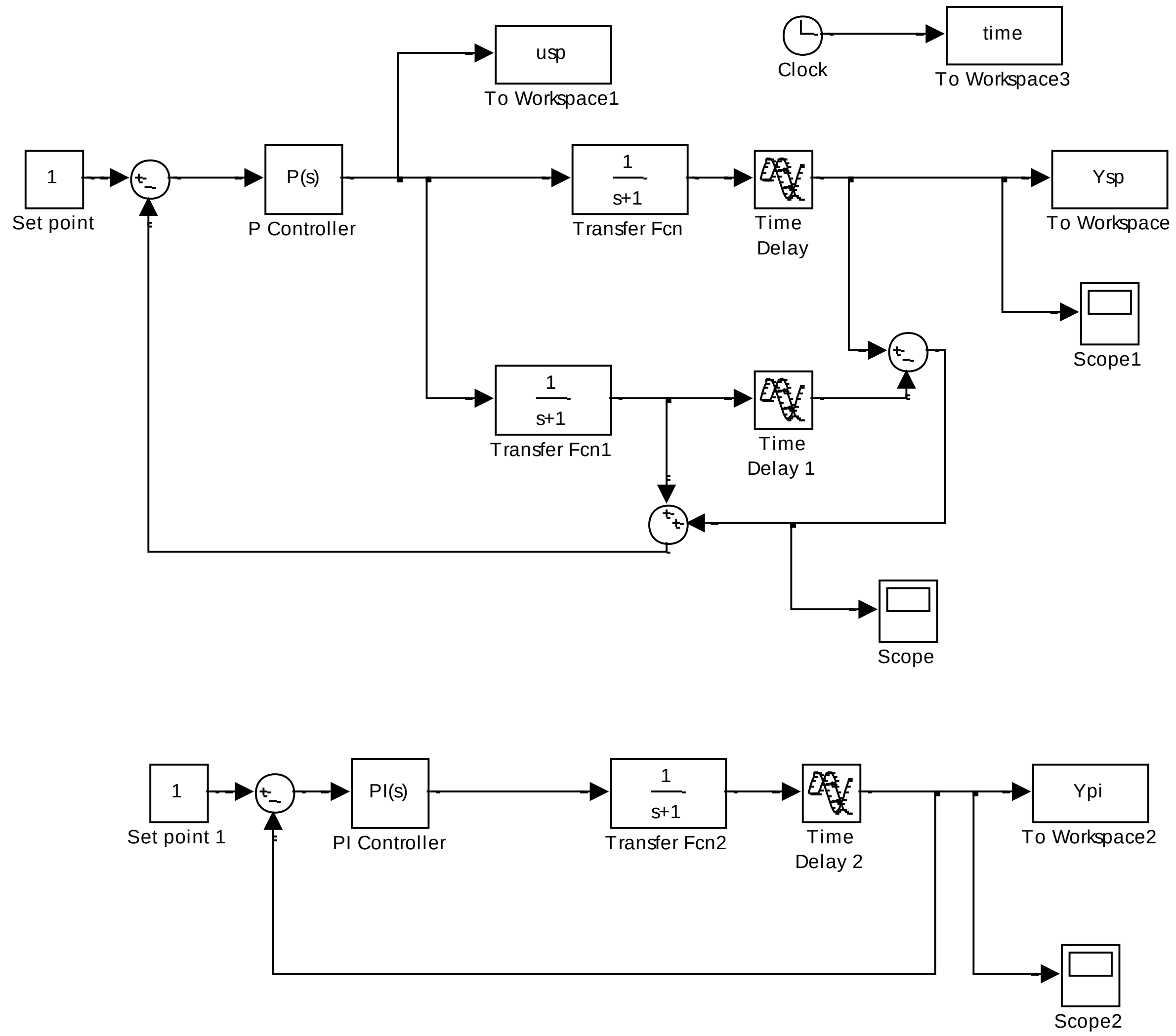


Figure 5: Block diagram of the Smith predictor and PI controller created in Simulink. The top block diagram is the Smith predictor structure and the bottom one is for a process with PI control.

3.2. PI Controller for the SP

A PI controller was used in the Smith predictor structure instead of the P controller in the previous section. This was performed in order to compare the difference between the two controllers.

The simulation of the example was performed in Matlab and Simulink, and the script and block diagram is given in the files *'Robustness_SP_PI'* and *'SP_and_PI_withPI.mdl'*. The block diagram created in Simulink was the same as the one shown in Figure 5, except that the primary controller in the Smith predictor was a PI controller and not a P-only controller. Because integral control was added, the integral time had to be included in the simulation. The integral time was set to τ_I = 1, which lead to an integral part equal to I=1. In addition the controller gain was changed from K_c=4 to K_c=2.

3.3. *Smith Predictor with Variation in Controller Gain*

After the robust tuning performed in the previous section, the controller gain of the primary controller in the Smith predictor was increased in order to see the effect on the performance. Increasing the controller gain is the same as tightening the control of a process. The simulations were performed for three different controller gains, that is K_c=2, K_c=3 and K_c=4.

3.4. *Robust Tuning of SP and SIMC Tuning of PI*

In this section, the process given in equation (1) was used when tests of the performance and robustness were performed. The Smith predictor was to be compared with a PI controller by using robust tuning rules for the Smith predictor, found in Normey-Rico and Camacho (2007) [7]. Conventional SIMC tuning rules given in section 2.3 was used for the PI controller. The robust tuning rules for the Smith predictor are the ones given in section 2.2.2.

Table 4 summarises the tuning values of the Smith predictor and the PI controller for this comparison when the maximum error in the time delay ($\Delta\theta$) is equal to 1. This means a range of time delays from θ=0 to θ=2 since the nominal time delay is θ_0=1. The calculation of the values in Table 4 is shown in Appendix B.

Table 4: Tuning values for the Smith Predictor and the PI controller with robust- and SIMC tuning rules, respectively.

	$P=K_c$	**$I=1/\tau_I$**
Robust tuning of SP	0.59	1
SIMC tuning of PI controller	0.5	1

To test the performance and robustness of the Smith predictor, this tuning was used for both set-point tracking and disturbance rejection in Simulink. The scripts for this are given in the files named *'SProb_PIsimctest'* and *'SProb_PIsimctest_dist'*. In addition, the corresponding Simulink files are named *'SProb_PIsimc.mdl'* and *'SProb_PIsimc_dist.mdl'*. To compare the performance between the Smith predictor and the PI controller, the integral squared error (ISE) was also calculated for the two control schemes. This was also done for both the set-point tracking and disturbance rejection. To obtain the correct integral square errors, some configuration parameters in Simulink had to be changed. This included changing type in solver options from variable-step to fixed-step, with a step size of 0.01. This was done in order to calculate the integral of the squared error. In addition the solver was set to ode3 (Bogacki-Shampine).

The set-point change was applied at t=10 and the disturbance was applied at t=0 in Simulink. The block diagram of the process in Simulink is given in Figure 6. In this simulation, the set-point change and disturbance was applied one by one. Thus, the controllers were first compared when there was a disturbance, and then when a set-point change was introduced. The blocks named *Dist* and *Dist1* are the disturbances, and the blocks named *Set-p* and *Set-p1* are set-point changes. There are also two blocks named *Dist set-p* and *Dist set-p1*. These blocks represent another type of disturbance which can be treated as a set-point change as well.

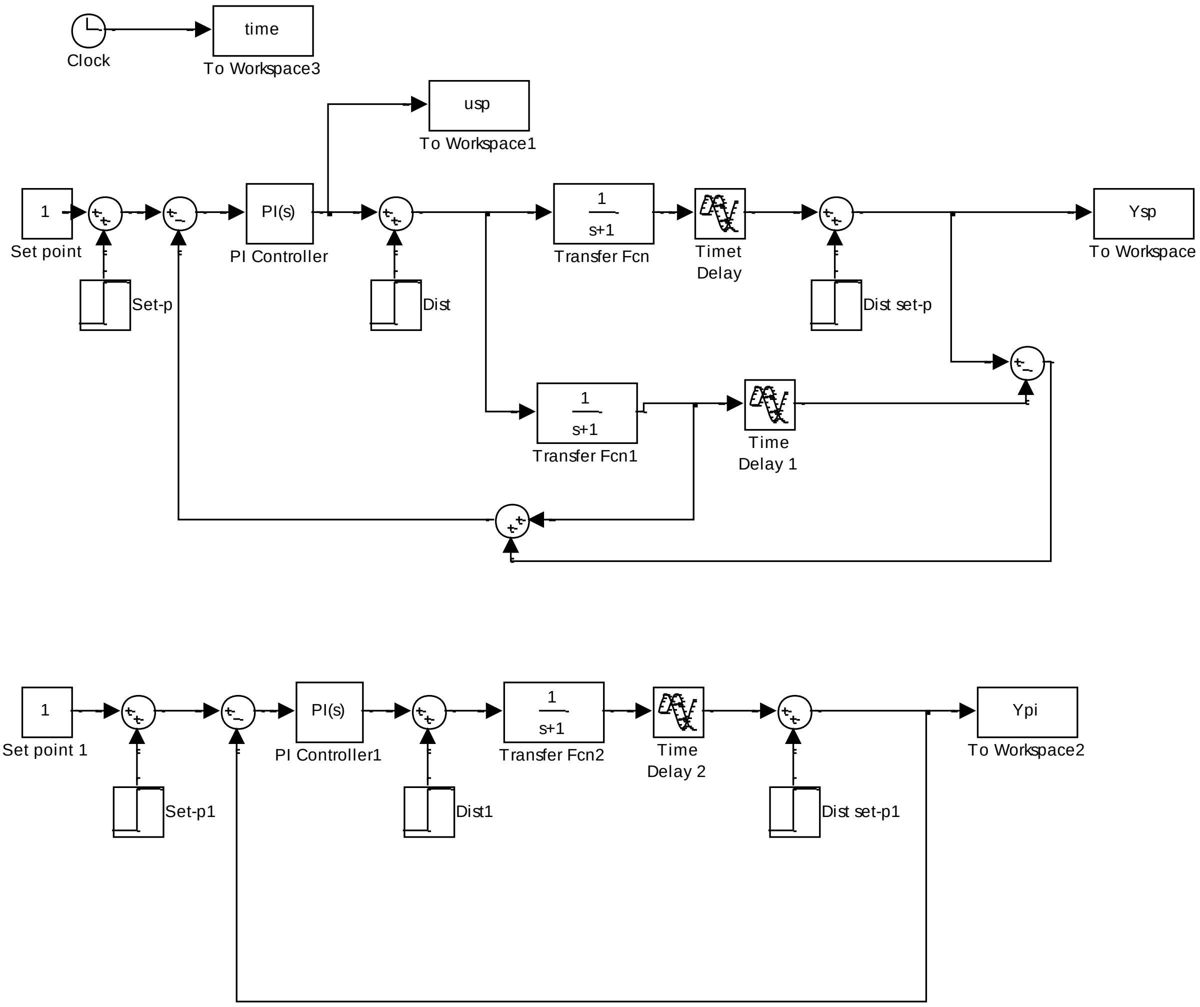


Figure 6: Block diagram of the process with the change in set-point and disturbance included. The top block diagram is the Smith Predictor controller, and the bottom one is the PI controller.

3.5. Optimisation of Performance and Robustness

In order to optimise the controllers in terms of performance and robustness, the functions called *optimset* and *fmincon* in Matlab were used. The latter finds a constrained minimum of a function of several variables. The Matlab scripts used for the optimisation are given in the folder named *'Optimal SP'* on the attached CD in Appendix A. As explained in section 2.5, the optimisation minimises the cost function $J(c)$, which is a function of the IAE values for a given controller. To find the best trade-off between performance ($J(c)$) and robustness (M_s), the cost function was plotted against a range of M_s values. This was done for two different processes, as shown in Table 5.

Table 5: The processes that were optimised in terms of performance and robustness.

	Process
Case 1	$g = \frac{e^{-s}}{s+1}$
Case 2	$g = \frac{e^{-s}}{8s+1}$

Case 1 is a first order plus time delay (FOPTD) process with $\theta = 1$ and $\tau = 1$, and case 2 is also a FOPTD process with $\theta = 1$ and $\tau = 8$. Included in the optimisation in Matlab was also finding the optimal tuning parameters for each controller in both cases.

To be able to compare the Smith predictor with a PI controller, the same IAE weights had to be used for the two controllers. Therefore, the weights of the SP controller were first found, and they were then used for the optimisation of the PI controller.

3.6. Verification of Optimisation in Simulink

To confirm the optimisation results from section 3.5, the optimal tuning for the SP and PI controller was applied to the model in Simulink and the IAE values were plotted for the different cases. This was done for several time delays to see how the controllers responded.

The model in Simulink was modified to be more similar to the model used by Grimholt and Skogestad (2011). In this paper, the set-point change was applied at t = 0 and the disturbance was applied at t = 20 [9]. For convenience, the same was done in this project. So in contrast to the simulations performed in section 3.4, where the disturbance and set-point change was simulated one by one, they were applied in the same simulation in the optimisation because the controllers were optimised for a combination of disturbance and set-point changes. Also, the blocks called *Dist set-p* and *Dist set-p1* in Figure 6 were used to apply the set-point changes because this was done in the paper by Grimholt and Skogestad (2011).

In order to find the best tuning of the two controllers in each case, a value of $M_s = 1.7$ was used because the SIMC rule gives this value. Then the optimal SP controller can be compared with the SIMC tuning rules applied on a PI controller later.

4. Results

4.1. *Verification of Example from an Article*

The plots of the outputs from the simulation of the example in the paper by Adam et al. (2000) are given in Appendix C. Table 6 gives the resulting stability regions achieved for the Smith predictor in this example. As shown here, the stability regions are the same as those found in the article [10].

Table 6: Stability domain made from the simulation of a P-only controller as the primary controller in the Smith predictor control structure.

Time Delay	Stable/Unstable
0 - 0.3462	Stable
0.3462 – 0.5668	Unstable
0.5668 – 1.4425	Stable
1.4425 – 1.8206	Unstable
1.8206 – 2.5320	Stable
2.5320 →	Unstable

Consequently, the first thing that appears is the discontinuity in the stability of the Smith predictor controller. With a time delay $\theta < 2.53$ the SP has three stable regions and two unstable regions of time delays, whereas it is unstable for all time delays above 2.53 in the tested time delay domain ($0<\theta<4$). As this example uses a P-only controller for the primary controller in the SP control structure, it indicates that it can be dangerous to use this if the ideal is a stable process and the time delay parameter is not exactly known. If a designer for example fixes $\theta_0=1$ and verifies that the loop is stable for the case where the actual process has a larger delay, for example $\theta=1.4$, it is incorrect to conclude that the closed loop will be stable for the case where the value of the real delay is $\theta < 1.4$. This is an important consequence of the discontinuous stability domain of the SP.

In order to see the difference between the PI controller and the Smith predictor, some of the output plots are included here. Figure 7 and Figure 8 show the output of the two controllers for time delays equal to $\theta=1.33$ and $\theta=1.78$, respectively. They show how the PI controller is still stable while the SP controller goes unstable at $\theta=1.78$.

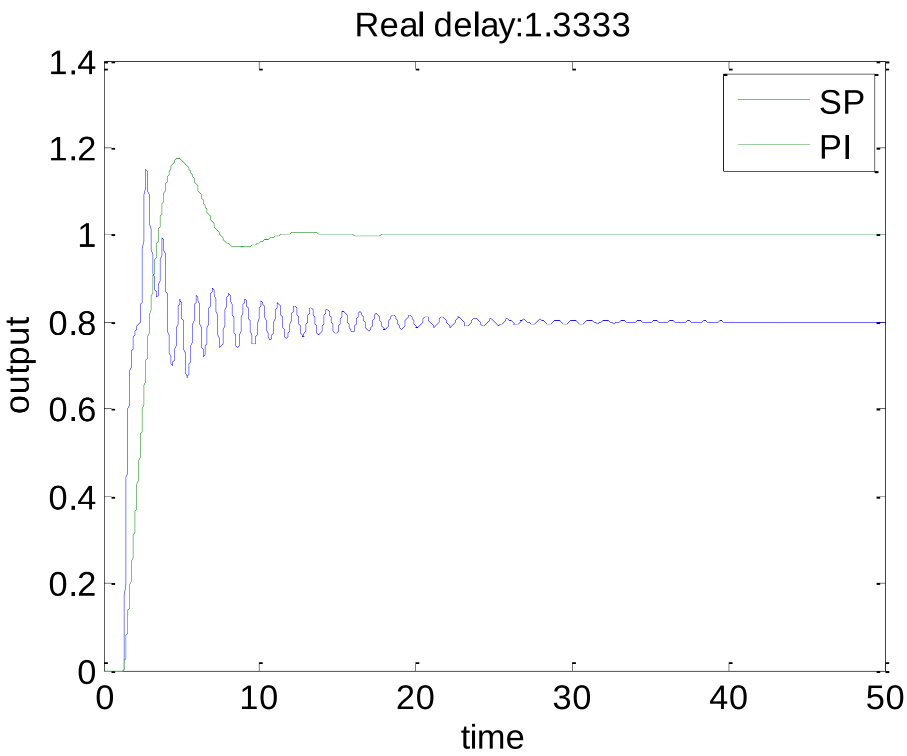


Figure 7: Output of the Smith Predictor and a PI controller where the time delay is equal to θ=1.33. A P-only controller is used as the primary controller in Smith Predictor control structure.

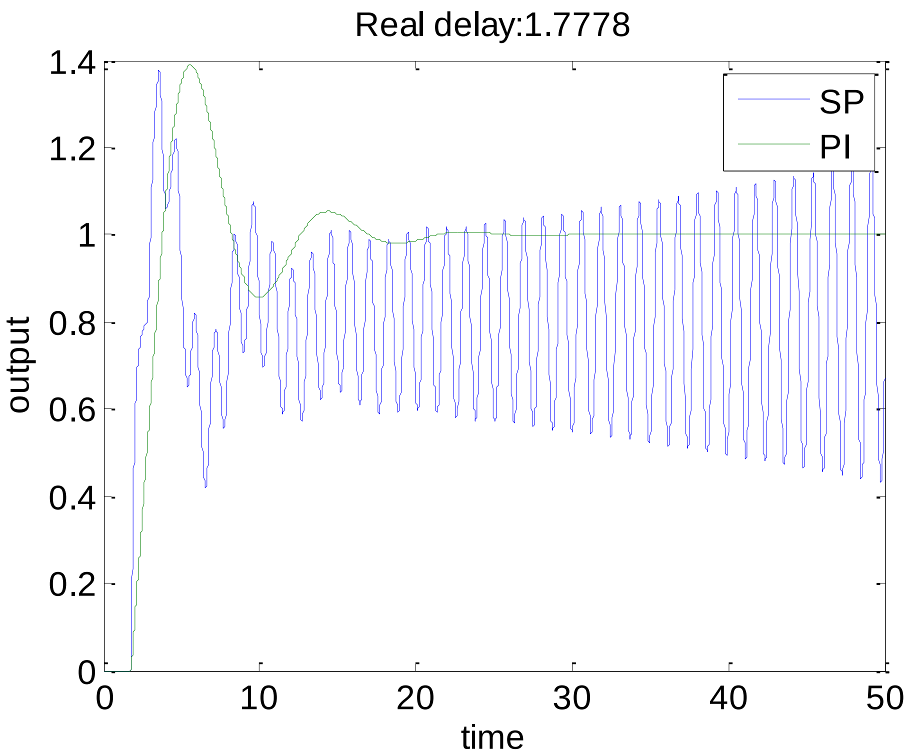


Figure 8: Output of the Smith Predictor and a PI controller where the time delay is equal to θ=1.78. A P-only controller is used as the primary controller in Smith Predictor control structure.

4.2. *PI Controller for the SP*

The plots of the outputs from the simulation of the same example, but with a PI controller instead of a P controller as the primary controller, are shown in Appendix D. Table 7 shows the stability region for this controller instead of a P-only controller. As shown here, this controller has only one stable region, and one unstable region in the tested time delay range (0<θ<4). Table 7 summarises the result of this simulation.

Table 7: Stability table made from the simulation of a PI controller as the primary controller in the Smith predictor control structure.

Time delay	Stable/unstable
0 – 2.68	Stable
2.68 →	Unstable

These simulations lead to the assumption that a P-only controller can give a discontinuous stability domain for a certain combination of tuning parameters, while a PI controller leads to a continuous stability domain for a given tuning. Therefore, if the time delay is unknown and has a value of for example θ < 2, using the Smith Predictor controller with a P-only controller includes a high risk of an unstable process or plant.

4.3. *Robust Tuning of SP and SIMC Tuning of PI*

4.3.1. Set-point Tracking

The integral squared error was calculated for different time delays in order to compare the SP with a PI controller and the results for a set-point change are shown in Table 8. In this case, the SP was tuned with robust tuning rules found by Normey-Rico and Camacho (2007) [7]. For the PI controller, conventional SIMC tuning rules were used for simplicity.

Table 8: Integral squared error results for different time delays for the PI controller and the Smith predictor controller for set-point tracking. SIMC tuning rules were used for the PI controller, while robust tuning rules were used for the SP.

Real Time Delay	ISE_{PI}	ISE_{SP}
0	3799,7	3706,1
0.11	3810,9	3717,6
0.21	3822,6	3729,4
0.32	3835,0	3741,5
0.42	3848,0	3754,0
0.53	3861,9	3767,0
0.63	3876,7	3780,3
0.74	3892,6	3794,1
0.84	3909,7	3808,4
0.95	3928,3	3823,3
1.05	3948,6	3838,9
1.16	3970,9	3855,2
1.26	3995,9	3872,3
1.37	4024,2	3890,4
1.47	4056,9	3909,5
1.58	4095,5	3929,8
1.68	4141,7	3951,6
1.79	4198,0	3975,0
1.89	4267,0	4000,2
2.0	4352,4	4027,5

As seen in Table 8, the ISE values of the Smith predictor are lower than the respective values for the PI controller for all time delays between $\theta=0$ and $\theta=2$. This confirms the idea of the Smith Predictor as a time delay compensator. Thus, the performance of the SP is better than for the PI controller in the tested time delay domain for a set-point change.

To see the results in a different way, the ISE values are also plotted in Figure 9 against the deviation from the nominal delay ($\theta-\theta_n$), where $\theta_n=1$.

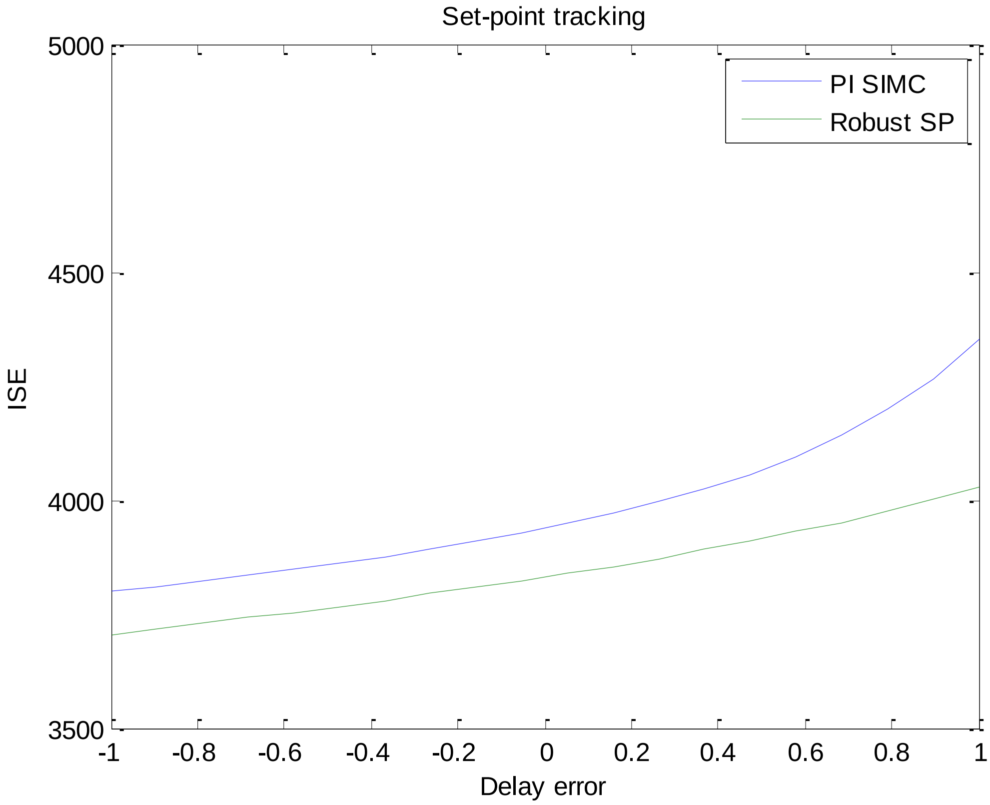


Figure 9: Integral squared error plotted against the error in time delay with robust tuning of SP and SIMC tuning of the PI controller for set-point tracking.

As from the table above, the same conclusions are drawn from Figure 9. The robust tuning of the Smith Predictor gives a better performance than SIMC tuning rules for a PI controller with values of the delay in the range $0<\theta<2$.

4.3.2. Disturbance Rejection

The resulting values of the ISE when a disturbance was applied to the process are given in Table 9. As for the set-point change, robust tuning and SIMC tuning rules were used for the SP and PI, respectively.

Table 9: Integral squared error results for different time delays for the PI controller and the Smith Predictor for disturbance rejection. SIMC tuning rules were used for the PI controller, while robust tuning rules were used for the SP.

Real Time Delay	ISE_{PI}	ISE_{SP}
0	33,8	40,5
0.11	44,2	47,9
0.21	55,5	55,8
0.32	67,9	64,2
0.42	81,5	73,1
0.53	96,3	82,5
0.63	112,6	92,4
0.74	130,5	102,8
0.84	150,2	114,0
0.95	171,9	125,8
1.05	196,0	138,3
1.16	222,8	151,7
1.26	252,7	166,0
1.37	286,2	181,3
1.47	323,9	197,6
1.58	366,8	215,2
1.68	415,8	234,0
1.79	472,5	254,4
1.89	538,6	276,3
2.0	616,8	300,2

For the first three values of the time delay, the PI controller has lower ISE values than the SP. However, the Smith predictor achieved lower values of the ISE for all the remaining time delays in the tested domain. Consequently, the performance of the SP is better than the PI controller for a disturbance in the process when the time delay is between $\theta=0.32$ and $\theta=2.0$, while the PI controller has better performance for time delays between in the range $0< \theta<0.21$.

Figure 10 shows the integral squared error as a function of the delay error, which is the difference from the nominal delay.

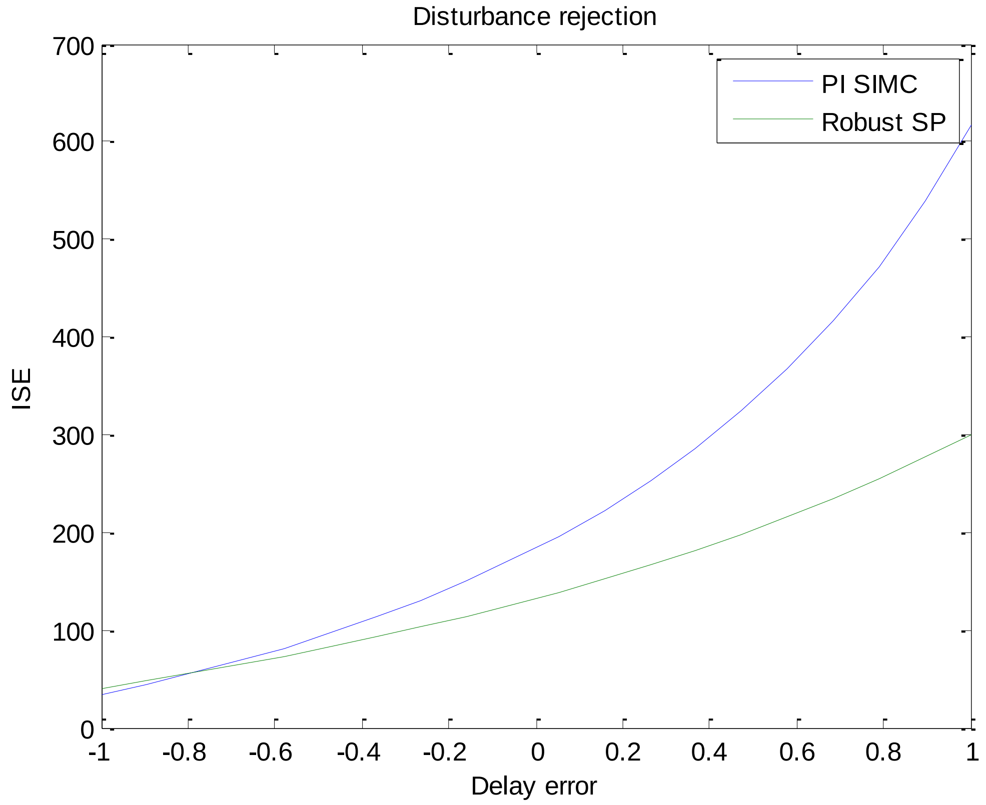


Figure 10: Integral square error plotted against the error in time delay with robust tuning of SP and SIMC tuning of the PI controller for disturbance rejection.

These graphs show the same results as in the previous table. In contrast to the set-point change, the difference in the ISE value between the two controllers is very small for small time delays, and then increases with the time delay value. This means that the improved performance for the SP is more significant at higher time delays than for low time delays when a disturbance is applied.

4.4. *Smith Predictor with Increasing Controller Gain*

In order to see how the Smith Predictor could handle different controller gains, the same simulations as in the previous section with robust tuning rules were performed, but with higher controller gains. This means that the control was tightened. The results are shown below by plotting the ISE values of the two controllers versus time delay. Figure 11, Figure 12 and Figure 13 show the results for set-point tracking when the controller gain has the values K_c = 2, K_c = 3 and K_c = 4, respectively.

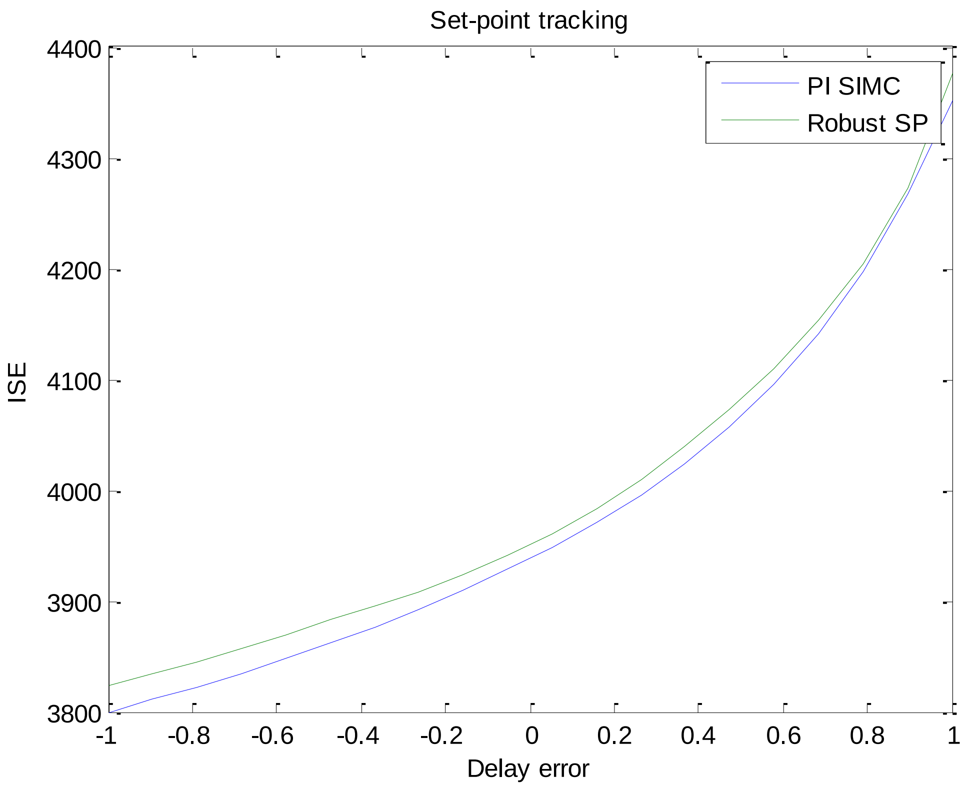


Figure 11: ISE values plotted against the error in delay (θ-θ_n) for set-point tracking when Kc = 2.

With a controller gain of K_c=2 in the primary controller in the SP, the PI controller achieves a higher performance for all the tested time delays, as shown in Figure 11. This indicates that the Smith predictor obtains worse performance for higher controller gains than what the robust tuning rules give. However, the difference between the PI and SP controller is not that evident for this value of the controller gain.

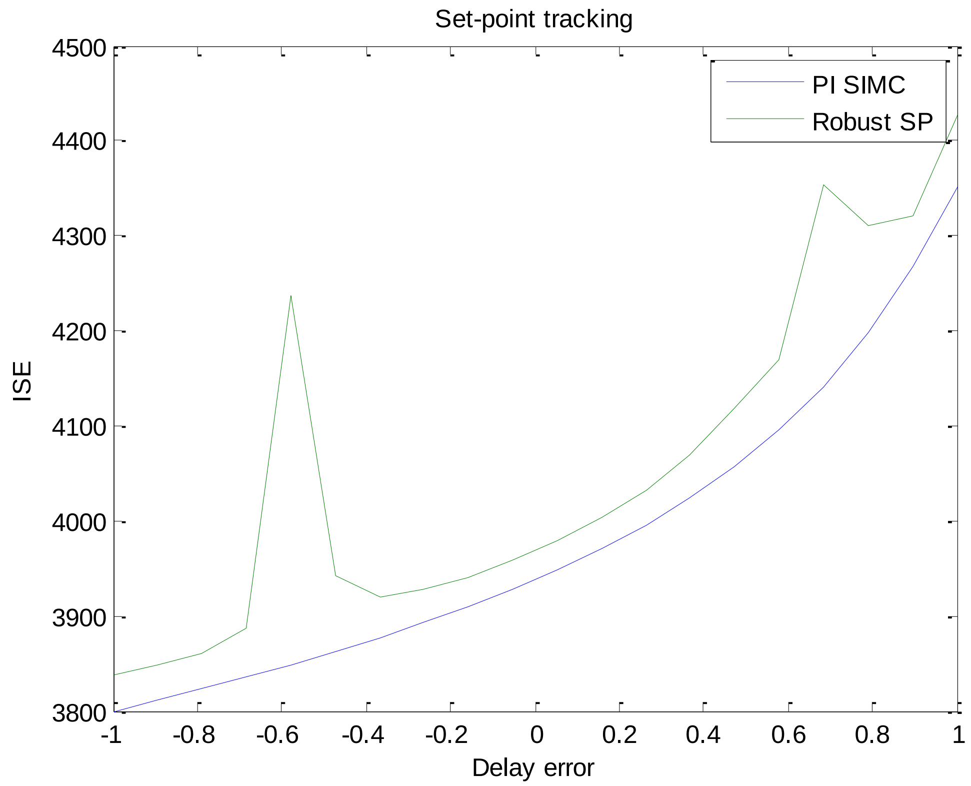


Figure 12: ISE values plotted against the error in delay (θ-θ_n) for set-point tracking when Kc = 3.

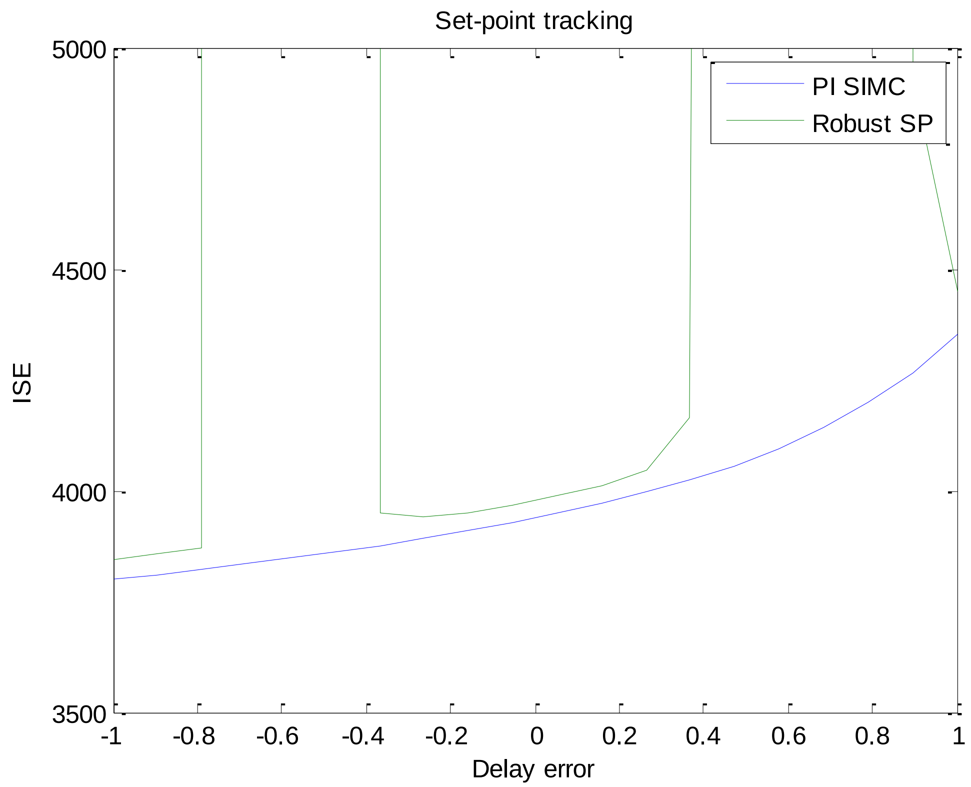


Figure 13: ISE values plotted against the error in delay (θ-θ_n) for set-point tracking when Kc = 4.

The plots in Figure 12 and Figure 13 show a strange behaviour of the ISE values of the Smith Predictor controller compared to the PI controller. First of all, the PI controller has a better performance than the SP for all time delays both when K_c=3 and when K_c=4. But it is also seen that the SP has two major jumps in the ISE values in both cases, where it becomes unstable. Between the two jumps, the ISE values decreases to the level it was before the peak. These results indicate that the Smith Predictor is very unstable for higher controller gains, especially if the time delay parameter is uncertain. Because of this, the SP may not be used that much when the time delay is not known and the model is not well understood.

Figure 14, Figure 15 and Figure 16 show the results for disturbance rejection when the controller gain has the values K_c = 2, K_c = 3 and K_c = 4, respectively.

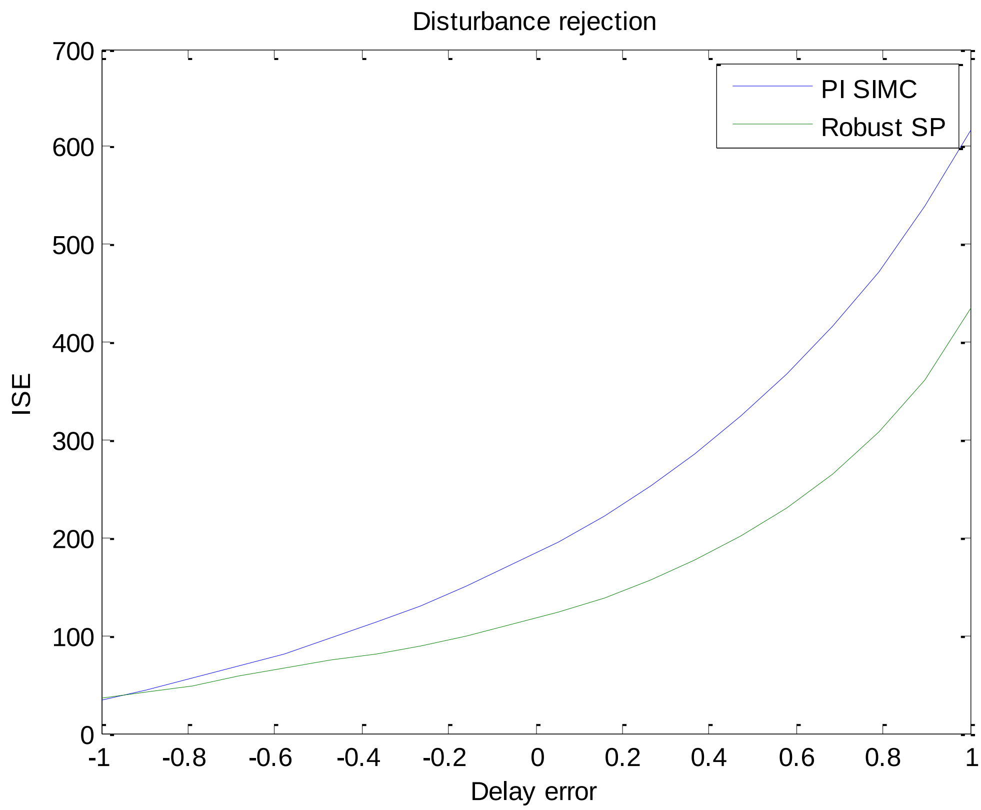


Figure 14: ISE values plotted against the error in delay (θ-θ_n) for disturbance rejection when Kc = 2.

When K_c=2, the SP still has better performance for the whole time delay domain. However, the performances of the PI and SP for small time delays are almost identical.

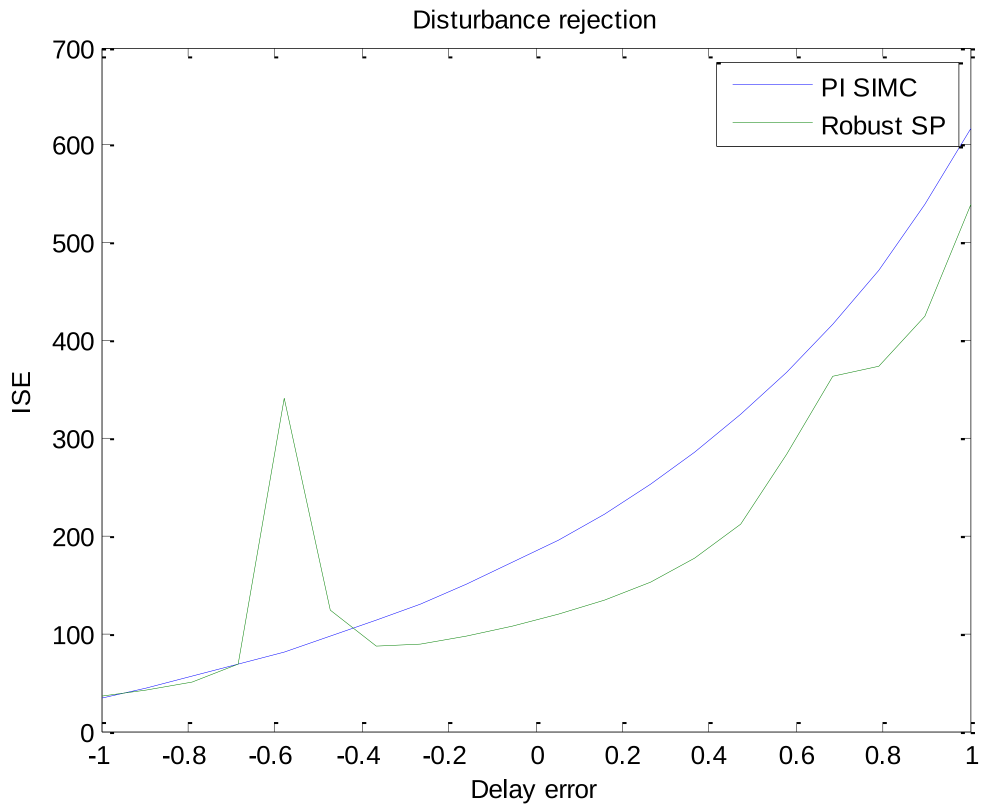


Figure 15: ISE values plotted against the error in delay (θ-θ_n) for disturbance rejection when Kc = 3.

The ISE values of the SP for a controller gain of K_c=3, show a strange behaviour. At a low time delay, the ISE values suddenly jump and peak at a delay error around -0.6, and the values decrease again after that. This result indicates that the SP may be unstable for certain time delays with this controller gain.

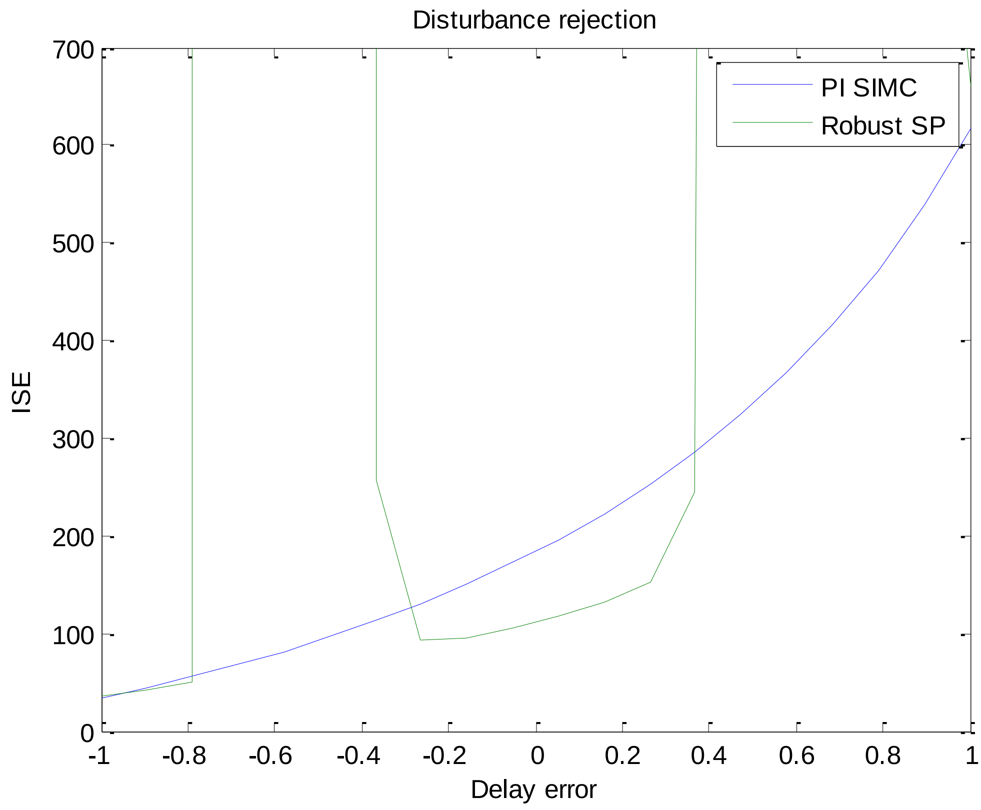


Figure 16: ISE values plotted against the error in delay (θ-θ_n) for disturbance rejection when Kc = 4.

For a controller gain of K_c = 4, the Smith Predictor is even more unstable than for K_c=3. This is seen in Figure 16, where the SP goes crazy at two different time delays, and then decreases again after a while. Thus, the PI controller has much better performance than the SP at the time delay values where the SP increases remarkably. When the ISE values of the SP are at "normal" values, they are a little below the values of the PI controller, so here the SP has better performance. A conclusion from this is that the Smith Predictor gets unstable when the control is tight enough (high enough K_c). Accordingly, it can be dangerous to control a process like this with the Smith predictor controller when the time delay is not exactly known. Thus, the decision of using a Smith Predictor or another controller will depend on the importance of a stable process and if the time delay is known.

To sum up, the Smith Predictor controller goes unstable and have two large peaks in the ISE values for both set-point changes and disturbances when the controller gain is increased to K_c = 4.

4.5. Optimisation of Performance and Robustness

The two processes g=exp(-s)/(s+1) and g=exp(-s)/(8s+1) were optimised in Matlab in terms of performance and robustness, and the results of this are given in the following sections. As a result of the optimisation, the optimal tuning for the Smith Predictor and the PI controller was also found.

4.5.1. Performance versus Robustness

Figure 17 and Figure 18 compares the trade-off between performance and robustness for the SP and PI controller for case 1 and case 2, respectively. It is widely known that there exists a trade-off between the performance and the robustness of a controller because it is not possible to have the best possible performance simultaneously as the best possible robustness. Therefore the selection of tuning parameters will depend on what of the two objectives is the most important. It is desired to find the best combination of performance and robustness.

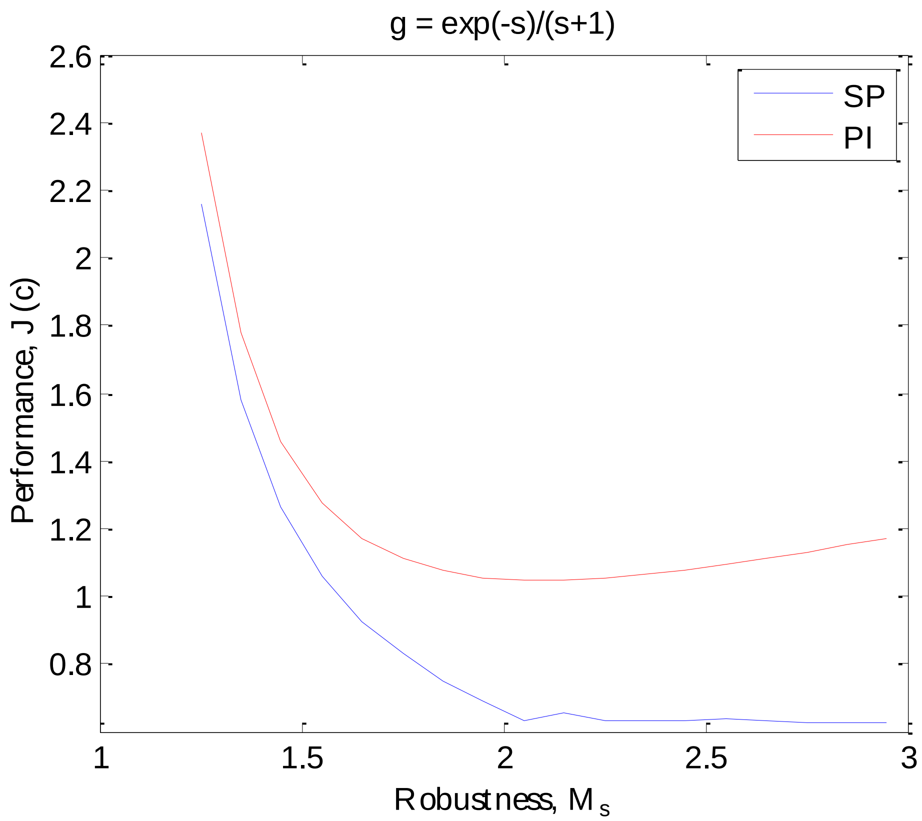


Figure 17: Trade-off between performance (J(c)) and robustness (M_s) for the Smith predictor controller and a PI controller for the process g=exp(-s)/(s+1). The process is optimised in terms of performance and robustness.

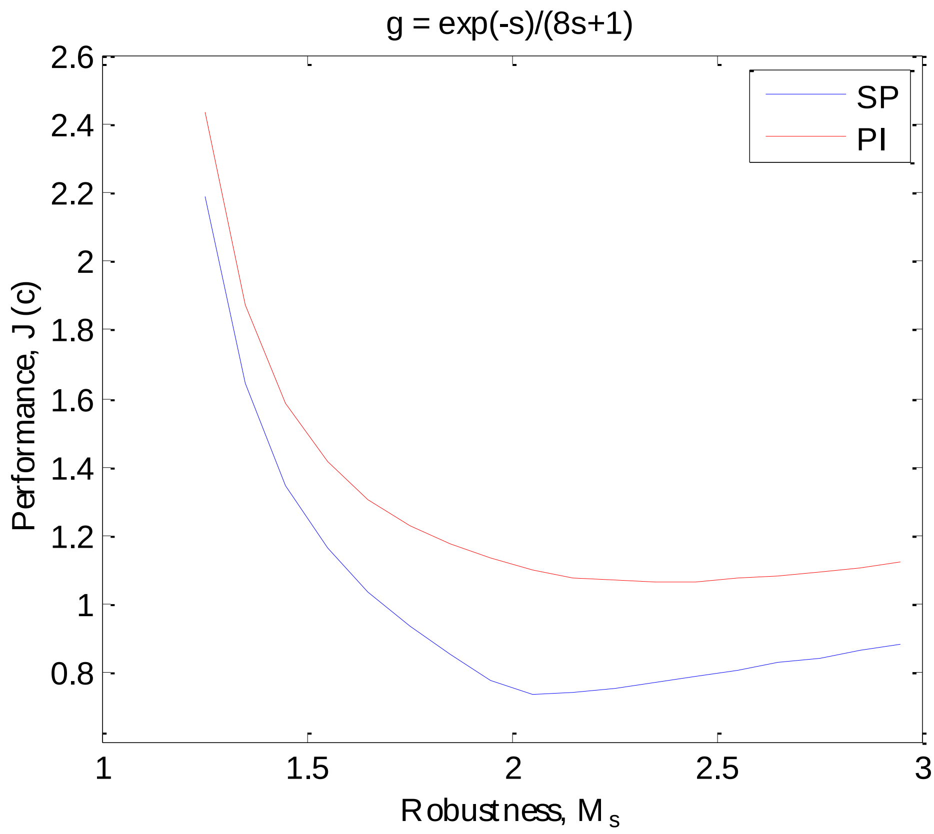


Figure 18: Trade-off between performance (J(c)) and robustness (M_s) for the Smith predictor controller and a PI controller for the process g=exp(-s)/(s+1). The process is optimised in terms of performance and robustness.

By comparing Figure 17 and Figure 18, it is observed that they look very similar. The difference between the two processes is the value of the process time constant. It has a value of τ=1 in the first case and a value of τ=8 in the second case. As explained before, a small value of M_s is desirable because the process is more robust the smaller value of M_s. At the same time, a small value of J(c) is also desirable because this is the weighted IAE values for set-point change and disturbances. And the optimal solution has the smallest possible error. In both the plots above, the part of the graphs beyond M_s = 2 is uninteresting because both the performance and the robustness decreases at the same time when M_s is larger than approximately 2. Somewhere in the region to the left of this value is therefore where the optimal trade-off will be located.

Also, the graphs show that the Smith Predictor controller has a better performance than the PI controller for all values of the robustness in both case 1 and case 2. This is for a combination of a set-point change and an applied disturbance to the processes. In addition, the increased performance for the SP is higher for higher values of M_s. One thing to notice is that this optimisation is for the processes with a time delay equal to θ=1, so the time delay is a known parameter for the optimisation. The performance of the controllers may not be the same when the real time delay is different from the nominal time delay.

4.5.2. Optimal Tuning

The robustness, tuning parameters and the performance function obtained from the optimisation is given in Table 10 and Table 11 for case 1 and case 2, respectively. The results are shown for both the Smith Predictor and the PI controller in each case.

Table 10: Robustness, controller gain, integral time and performance results from optimisation of both the SP and PI controller for the process g=exp(-s)/(s+1).

Case 1: $g = \frac{e^{-s}}{s+1}$								
	Smith Predictor				PI Controller			
M_s	$K_c = P$	K_c/τ_I	$I = 1/\tau_I$	J(c)	$K_c = P$	K_c/τ_I	$I = 1/\tau_I$	J(c)
1,25	3,63	3,63	1,00	2,16	3,98	3,98	1,00	2,37
1,35	2,66	2,65	1,00	1,58	2,99	2,99	1,00	1,78
1,45	2,13	2,12	1,00	1,26	2,45	2,45	1,00	1,46
1,55	1,79	1,78	0,99	1,06	2,15	2,13	0,99	1,27
1,65	1,56	1,55	0,99	0,93	2,01	1,92	0,95	1,17
1,75	1,40	1,39	0,99	0,83	1,97	1,77	0,90	1,11
1,85	1,26	1,25	0,99	0,75	1,94	1,67	0,86	1,07
1,95	1,19	1,13	0,95	0,69	1,94	1,61	0,83	1,06
2,05	1,18	1,07	0,91	0,67	1,96	1,57	0,80	1,05
2,15	1,21	1,06	0,88	0,67	1,98	1,55	0,78	1,05
2,25	1,26	1,05	0,84	0,69	2,02	1,53	0,76	1,06
2,35	1,34	1,06	0,79	0,71	2,06	1,52	0,74	1,07
2,45	1,43	1,06	0,74	0,74	2,11	1,52	0,72	1,08
2,55	1,52	1,07	0,70	0,77	2,15	1,53	0,71	1,09
2,65	1,62	1,07	0,66	0,80	2,21	1,54	0,70	1,11
2,75	1,69	1,08	0,64	0,82	2,26	1,55	0,69	1,13
2,85	1,76	1,08	0,61	0,84	2,31	1,56	0,68	1,15
2,95	1,82	1,08	0,59	0,86	2,36	1,58	0,67	1,17

Table 11: Robustness, controller gain, integral time and performance results from optimisation of both the SP and PI controller for the process g=exp(-s)/(8s+1).

Case 2: $g = \frac{e^{-s}}{8s+1}$								
	Smith Predictor controller				PI Controller			
M_s	$K_c = P$	K_c/τ_I	$I = 1/\tau_I$	J(c)	$K_c = P$	K_c/τ_I	$I = 1/\tau_I$	J(c)
1,25	3,86	2,67	0,69	2,19	4,30	2,96	0,69	2,44
1,35	2,99	1,93	0,65	1,64	3,56	2,09	0,59	1,87
1,45	2,46	1,58	0,64	1,35	3,23	1,60	0,50	1,58
1,55	2,09	1,39	0,66	1,16	3,02	1,33	0,44	1,42
1,65	1,81	1,27	0,70	1,03	2,88	1,16	0,40	1,31
1,75	1,58	1,20	0,76	0,94	2,79	1,04	0,37	1,23
1,85	1,37	1,15	0,84	0,86	2,74	0,94	0,34	1,18
1,95	1,17	1,10	0,94	0,78	2,68	0,87	0,33	1,13
2,05	1,17	1,01	0,86	0,74	2,66	0,81	0,30	1,10
2,15	1,18	1,00	0,85	0,74	2,67	0,74	0,28	1,08
2,25	1,21	1,00	0,83	0,75	2,69	0,70	0,26	1,07
2,35	1,27	1,00	0,79	0,77	2,72	0,67	0,25	1,07
2,45	1,38	1,00	0,73	0,80	2,74	0,66	0,24	1,07
2,55	1,53	1,00	0,65	0,85	2,80	0,63	0,23	1,07
2,65	1,56	1,00	0,64	0,86	2,85	0,62	0,22	1,08
2,75	1,42	1,00	0,70	0,81	2,88	0,62	0,21	1,09
2,85	1,44	1,00	0,69	0,82	2,97	0,59	0,20	1,11
2,95	1,47	1,00	0,68	0,83	3,02	0,59	0,19	1,12

As explained in section 3.6, a robustness of $M_S = 1.7$ was used as the best optimal tuning for the two controllers. In order to find the exact optimal tuning values for this robustness, the optimal tuning was also plotted as a function of the robustness in Matlab, and these graphs are given in Appendix E. The best optimal tuning for this robustness is shown in Table 12 for both cases.

Table 12: Optimal tuning parameters for a robustness of M_s = 1.7 for the Smith Predictor and PI controller.

Robustness, M_s = 1.7				
	$P = K_c$		$I = 1/\tau_I$	
	SP	PI	SP	PI
Case 1: $g = \frac{e^{-s}}{s+1}$	1.48	1.99	0.99	0.93
Case 2: $g = \frac{e^{-s}}{8s+1}$	1.70	2.83	0.73	0.39

The optimal tuning parameters found from the optimisation are different from the tuning parameters found by the robust tuning of SP and the SIMC tuning of the PI controller. Case 2 has different optimal tuning parameters than case 1 and that is probably because of the different time constant of the two processes. Case 2 has a tighter control for both the SP and PI controller when it comes to controller gain. Also, case 2 has higher values of integral time for the two controllers.

4.6. Verification of Optimisation in Simulink

In order to verify the optimisation results from the previous section, the simulations in Simulink were performed with the obtained optimal tuning parameters for case 1 and case 2. However, the simulations were performed for a combination of a set-point change and a disturbance in order to compare it with the optimisation results. The true time delay was varied, and the IAE values were plotted for each case, as shown in Figure 19 and Figure 20.

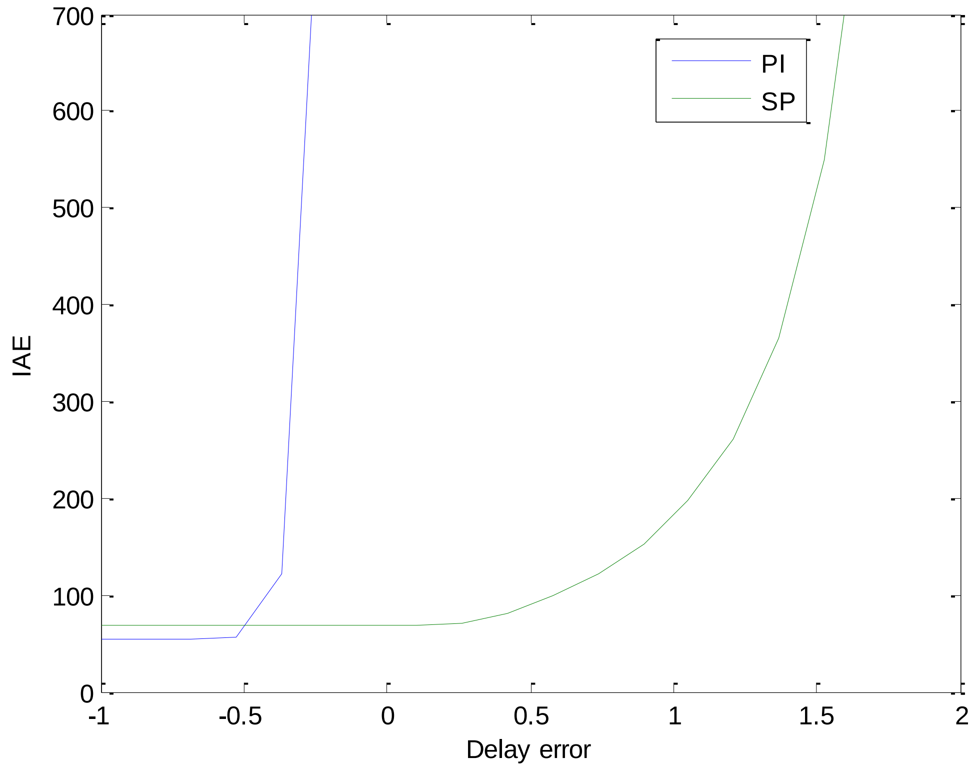


Figure 19: IAE plotted against the delay error for case 1 with the process g=exp(-s)/(s+1) when the optimal tuning is applied on both controllers.

With a delay error from -1 to -0.5 in case 1, the PI controller actually has lower IAE values than the Smith Predictor controller. With further increase in the time delay, the SP has the best performance. However, both controllers IAE values increase rapidly when at two different time delays.

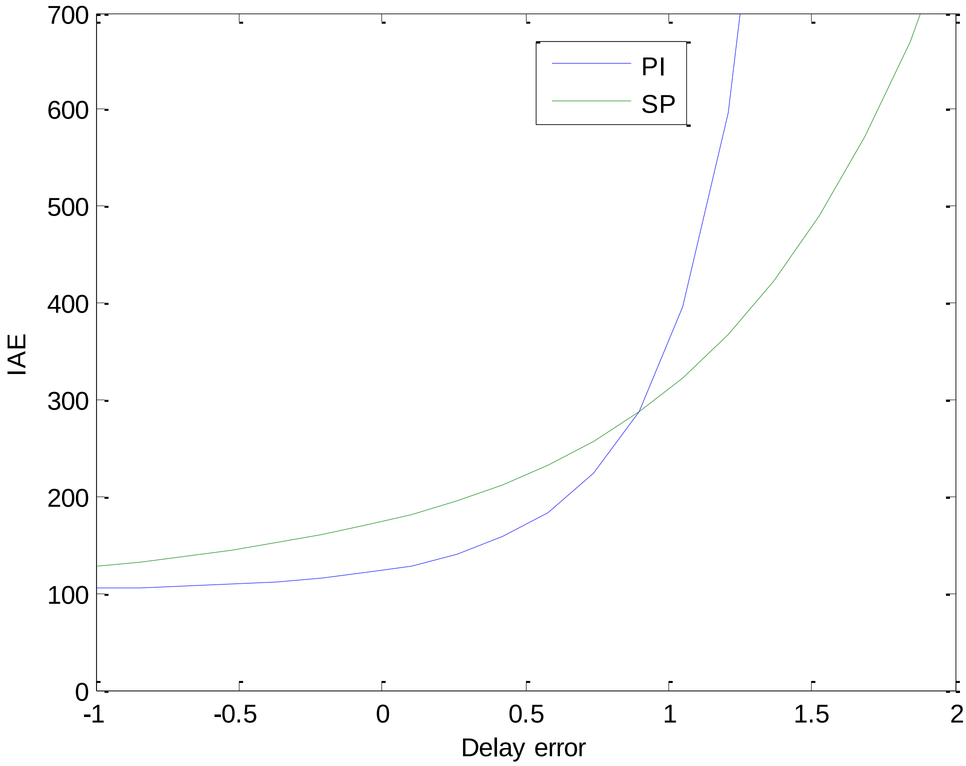


Figure 20: IAE plotted against the delay error for case 2 with the process g=exp(-s)/(8s+1) when the optimal tuning is applied in both controllers.

A similar behaviour is observed for case 2, but the PI controller has the best performance for delay errors up to around 0.9.

Based on these results, it looks like even though the optimal tuning is applied for both controllers, the Smith Predictor does not have the best performance for all values of the time delay. But at the same time, the optimal tuning parameters were found as a trade-off between performance and robustness. In addition, the magnitude of the difference in performance is not very large at the time delays where the PI controller has best performance.

5. Discussion

After evaluating all the results, it can be concluded that the stability of the Smith predictor depends strongly on the chosen tuning parameters for a given process. For example, the stability was drastically altered when the controller gain was increased. Another issue is that the PI control system was tuned using the SIMC tuning rules in all of the simulations. The results and comparisons with the Smith predictor could have been different if the tuning parameters of the PI controller were changed as well.

In addition to the Smith Predictor controller, there exist other types of controllers that compensate for the time delay parameters. These could be better or worse than the Smith Predictor. The Smith Predictor has also been modified for certain processes, and this could be tested to see if the performance and robustness would increase.

In the optimisation part, a third case with the transfer function g=exp(-s) was also supposed to be optimised. However, there were problems with achieving convergence in the optimisation of this process (pure time delay). Several initial guesses were tried, but none of them gave convergence for all the iterations. Thus, only case 1 and case 2 was optimised and focused on.

Further work from this project could be to optimise the third process (pure time delay) in terms of performance and robustness in order to see how this kind of process would behave. In addition, modified versions of the Smith Predictor controller could be tested in order to improve the standard Smith Predictor. A wider range of time delays could also have been tested to see how the controllers respond to very large time delays.

Another thing that can be tried is to add a reference filter to the Smith Predictor structure in order to improve the set-point response [11]. A predictor filter could also be used to improve the properties of the predictor, as proposed by Normey-Rico and Camacho (2009).

6. Conclusions

In this project, the performance and robustness of a Smith Predictor controller has been tested for processes with time delays. First order processes were tested and compared with the performance and robustness of a PI controller. The SP and PI controllers were also optimised in terms of performance and robustness for two different process transfer functions, and the optimal tuning parameters were found.

From the optimisation, the Smith Predictor controller turned out to have better performance than the PI controller for all values of the robustness M_s, as shown in Figure 17 and Figure 18. Also, the magnitude of the performance difference between the two controllers increases with increasing M_s.

When the time delay was varied for the optimal tuning of the SP and PI controller, the SP controller had better performance when the time delay reached a certain value, as shown in Figure 19 and Figure 20. However, the Smith predictor controller had discontinuous stability domains and unstable behaviour for certain combinations of the controller gain and other tuning parameters.

7. Symbols and Abbreviations

7.1. *List of symbols*

Symbol	Unit	Description
θ	min	Process time delay
θ_0	min	Nominal time delay (=1)
τ	min	Time constant
τ_0	min	Desired time constant
τ_1	min	Process time constant
τ_c	min	Desired first order closed-loop time constant
τ_I	min	Integral time
c	-	Transfer function for a given controller
D	-	Disturbance to the process
e(t)	-	Error between set-point and output in time domain
E	-	Error between set-point and output in Laplace domain
g	-	Transfer function for a given process
G	-	Transfer function in Laplace domain
G_c	-	Controller transfer function
G_d	-	Disturbance transfer function
G_n	-	Transfer function for process without time delay
G_p	-	Process transfer function
I (tuning)	-	Integral part
J(c)	-	Weighted integral absolute error for disturbance and set-point change
k	-	Steady-state gain
k'	s^{-1}	Steady-state gain divided by process time constant
K_c	-	Controller gain
M_s	-	Robustness
p(t)	-	Output of controller in time domain
P	-	Output of controller/input to process
P(tuning)	-	Proportional part
$P_n(s)$	-	Total process in the predictor part
s	-	Laplace domain
t	min	Time
y(t)	-	Process output in time domain
Y	-	Process output
Y_1	-	Output of process without time delay
Y_2	-	Output of process with time delay in Smith predictor structure
Y_S	-	Process set-point

7.2. List of Abbreviations

Abbreviation	Stands for
EGPTD	Equivalent gain plus time delay
FOPTD	First order plus time delay
IAE	Integral absolute error
ISE	Integral squared error
PI	Proportional and Integral
PID	Proportional, Integral and Derivative
SP	Smith Predictor

8. References

1. Seborg, D.E., Edgar, T. F., Mellichamp, D. A., Doyle III, F. J., *Process Dynamics and Control*2011, Asia: John Wiley & Sons, Inc.
2. Matausek, M.R., Micic, A. D., *A Modified Smith Predictor for Controlling a Process with an Integrator and Long Dead-Time.* 1996. **41**(8).
3. Lee, T.H., Wang, Q. G., *Robust Smith-Predictor Controller for Uncertain Delay Systems.* 1996. **42**(4).
4. Lee, D., Lee, M., Sung, S., Lee, I., *Robust PID tuning for Smith predictor in the presence of model uncertainty.* 1999.
5. Majhi, S., Atherton, D. P., *Modified Smith predictor and controller for processes with time delay.* 1999. **146**(5).
6. da Silva, G., *CHEN30009, Process Dynamics and Control, Course Notes 2011*, 2011, Department of Chemical and Biomolecular Engineering: The University of Melbourne.
7. Normey-Rico, J.E., Camacho, E. F., *Control of Dead-time Processes*. Advanced Textbooks in Control and Signal Processing 2007, London: Springer-Verlag.
8. Skogestad, S., Grimholt, C., *The SIMC method for smooth PID controller tuning.* 2011.
9. Grimholt, C., Skogestad, S., *Optimal PI-Control and Verification of the SIMC Tuning Rule.* 2011.
10. Adam, E.J., Latchman, H. A., Crisalle, O. D., *Robustness of the Smith Predictor with respect to Uncertainty in the Time-Delay Parameter.* 2000: p. 1452-1457.
11. Normey-Rico, J.E., Camacho, E. F., *Unified approach for robust dead-time compensator design.* 2009.

Appendix A – Matlab and Simulink files

Appendix B – Calculation of Tuning Values

The calculation of the tuning parameters with SIMC tuning rules is shown below for the PI controller:

$$K_c = \frac{1}{k}\frac{\tau_1}{\tau_c+\theta} = \frac{1}{1}\frac{1}{1+1} = 0.5$$

$$\tau_I = \min[\tau_1, 4(\tau_c + \theta)]$$

$$\tau_1 = 1$$

$$4(\tau_c + \theta)] = 4(1 + 1) = 8$$

$$=> \tau_I = 1$$

The calculation of the tuning parameters for the robust tuning of the Smith predictor is given here:

$$\tau_0 = 1.7\Delta\theta_{max}$$

Where $\Delta\theta_{max} = 1$

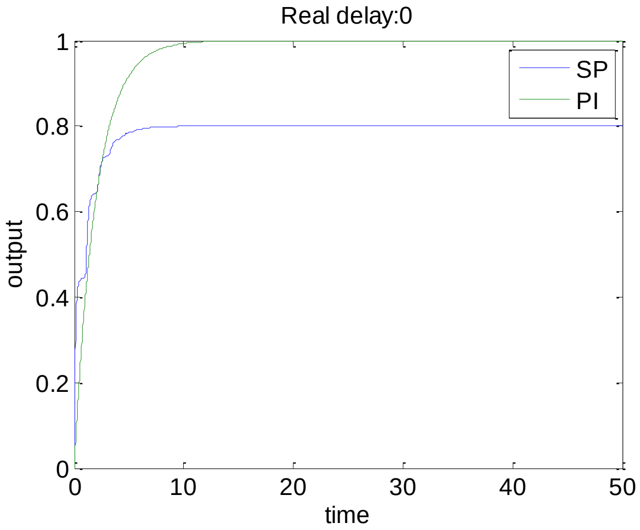
$$=> \tau_0 = 1.7$$

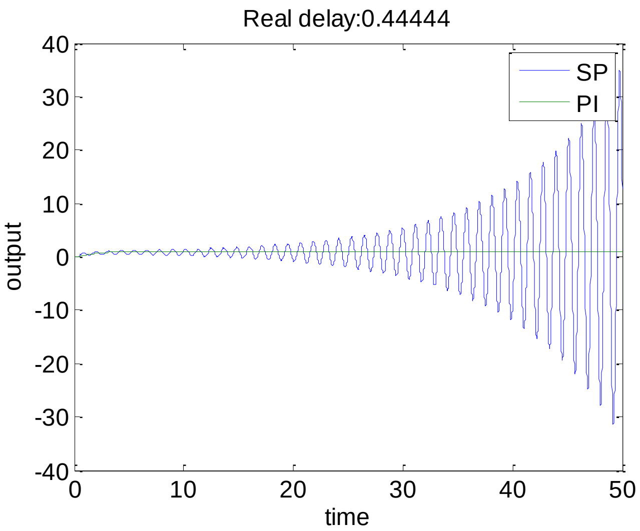
$$\tau_I = \tau = 1$$

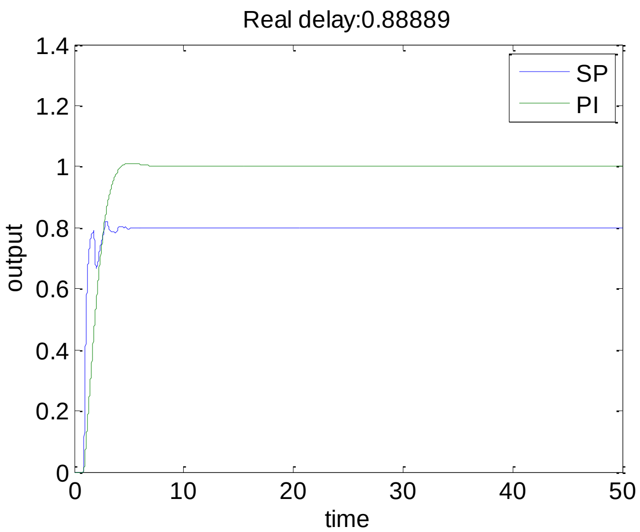
$$K_c = \frac{\tau}{K_p\tau_0} = \frac{1}{1*1.7} = 0.59$$

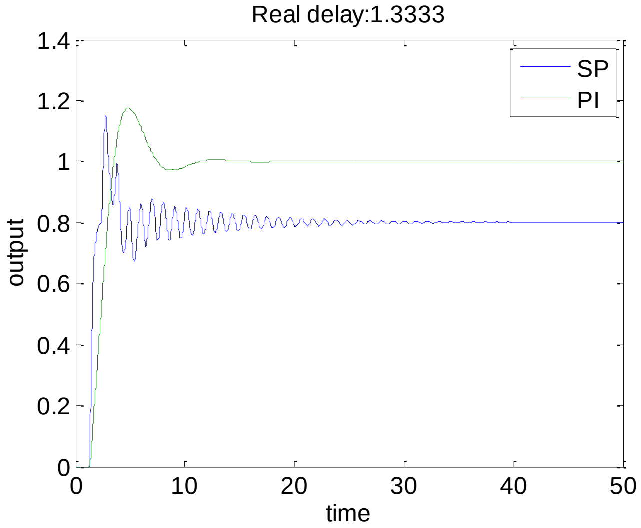
Appendix C – Plots from Verification of Example

The plots made from this script are shown in the following figures for a time delay range of 0 < θ < 4.

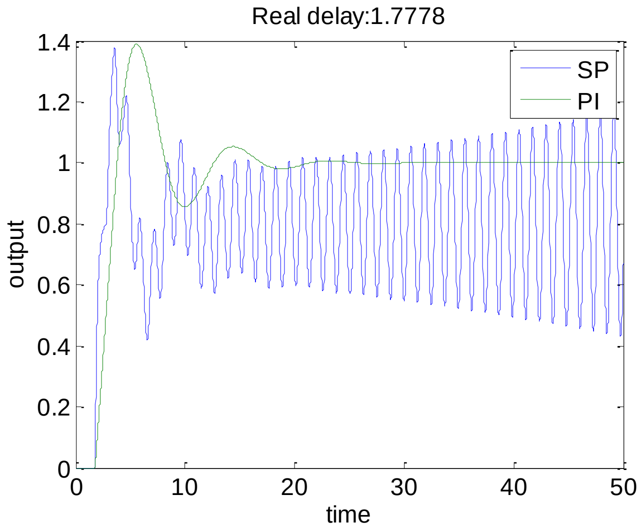




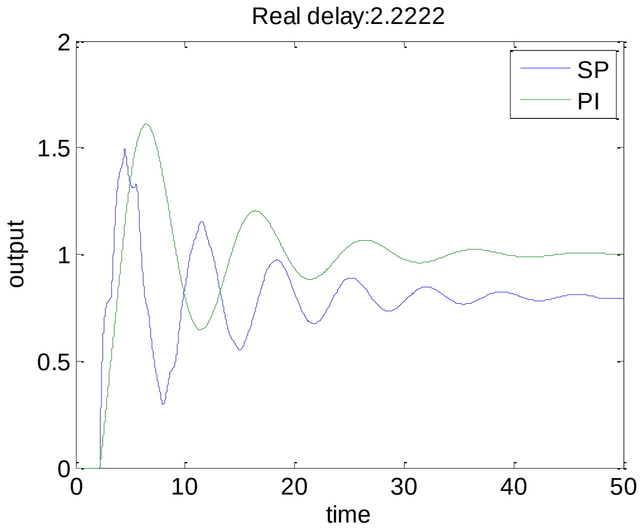




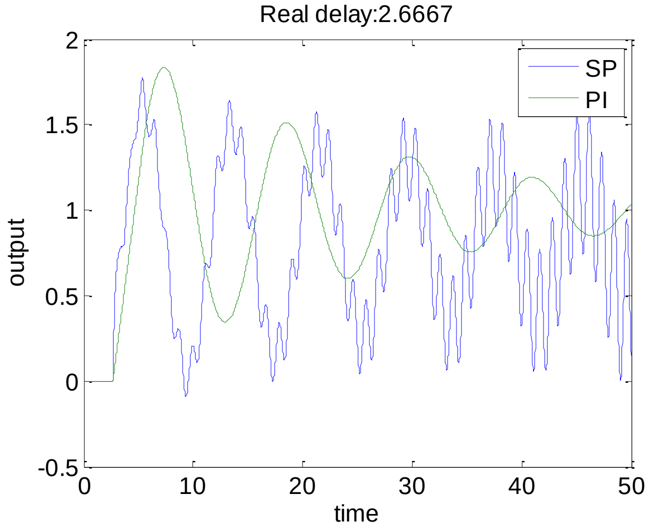
Real delay:1.3333
SP
PI
output
time

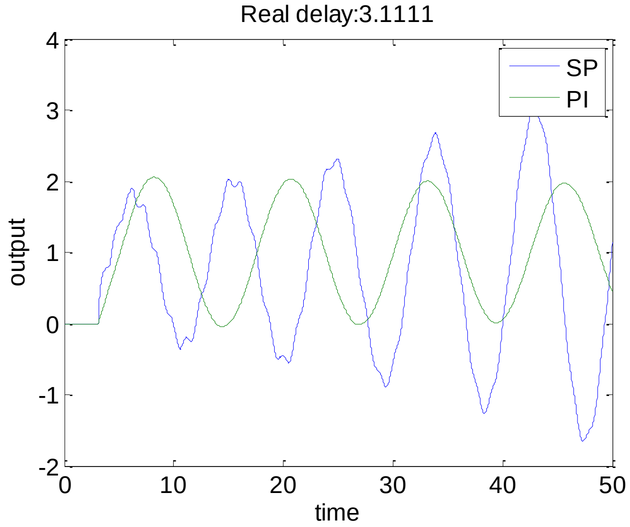


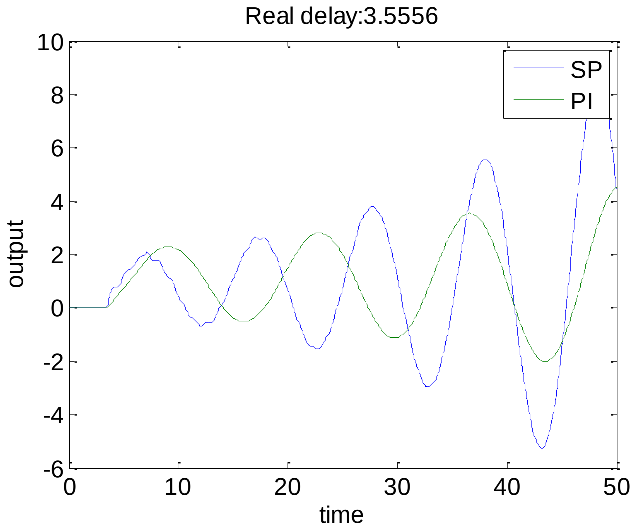
Real delay:1.7778
SP
PI
output
time

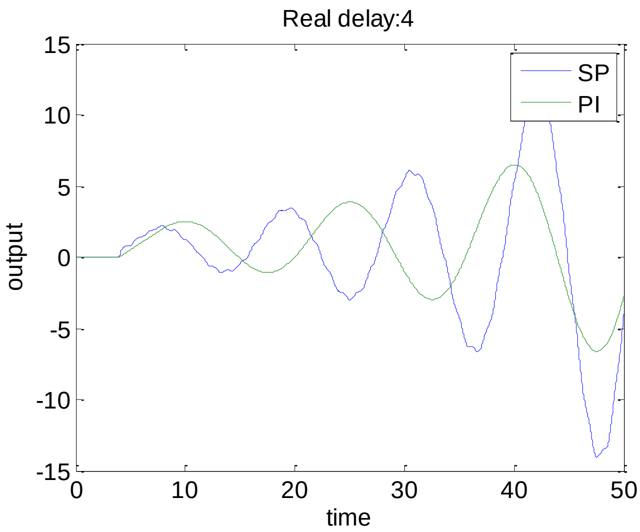


Real delay:2.2222
SP
PI
output
time


Real delay:2.6667
output
time
SP
PI
2
1.5
1
0.5
0
-0.5
0
10
20
30
40
50

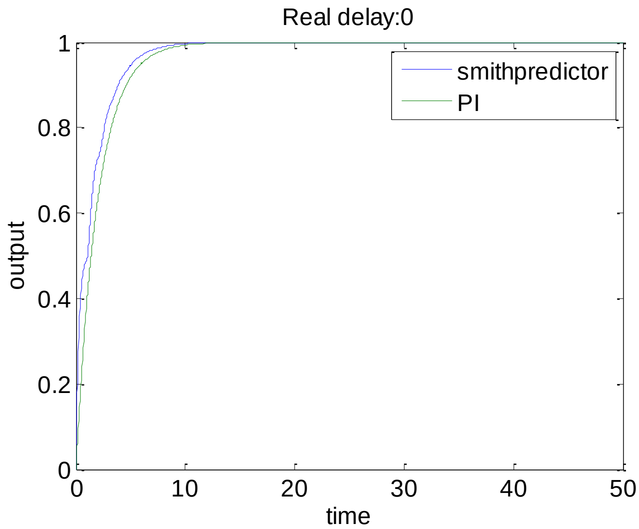

Real delay:3.1111
output
time
SP
PI
4
3
2
1
0
-1
-2
0
10
20
30
40
50

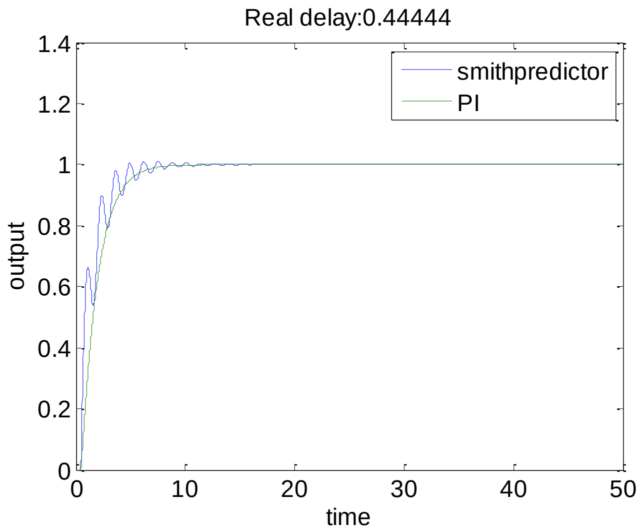

Real delay:3.5556
output
time
SP
PI
10
8
6
4
2
0
-2
-4
-6
0
10
20
30
40
50

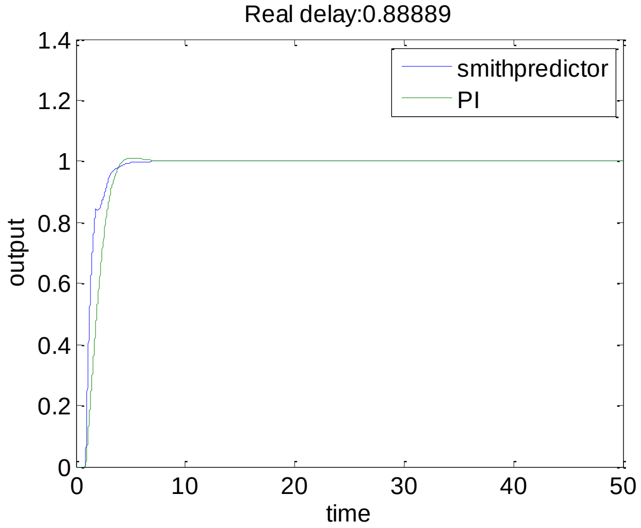

Real delay:4
15
10
5
0
-5
-10
-15
output
0
10
20
30
40
50
time
SP
PI

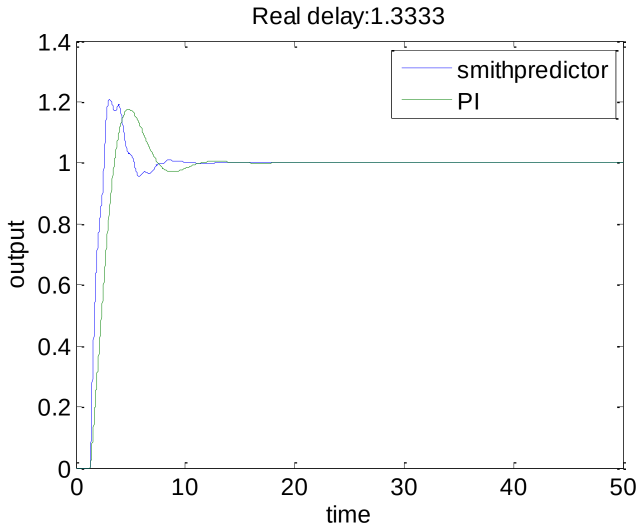
Appendix D – Plots From the Use of a PI Controller as the Primary Controller

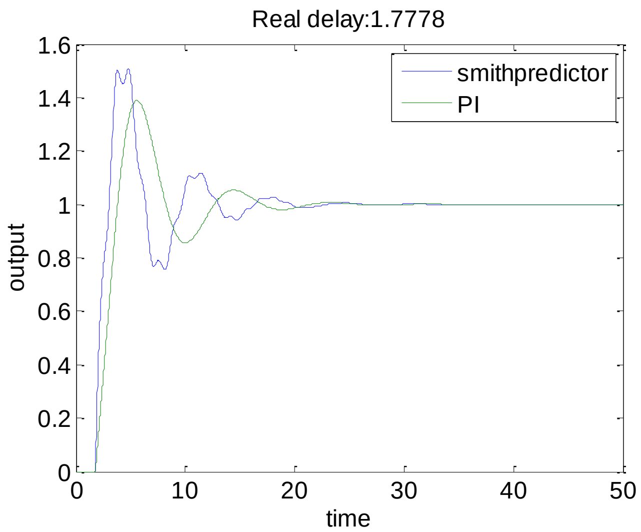
The plots made from this script are shown in the following figures and the values of the time delay are 0 < θ < 4.

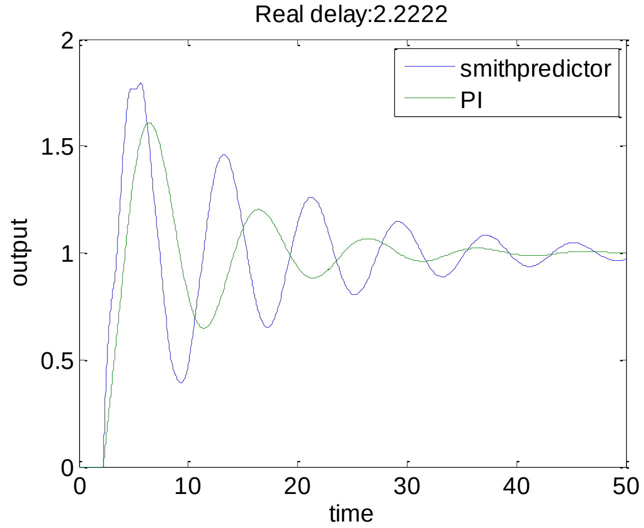


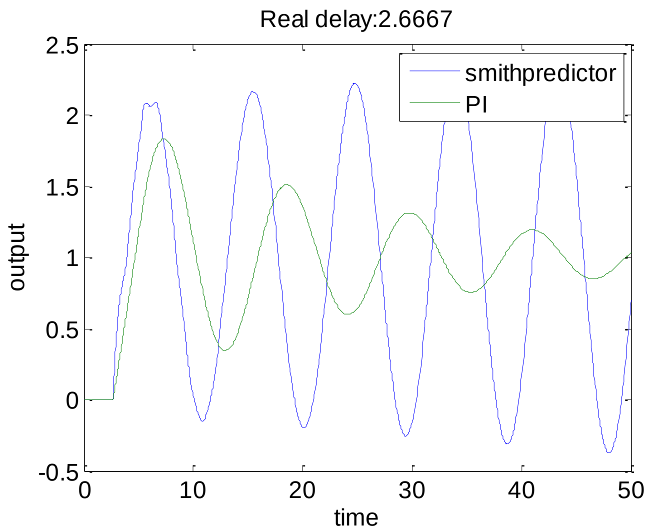


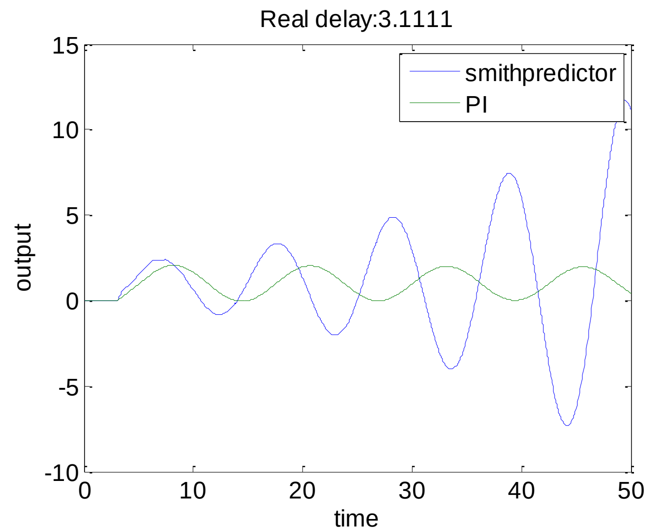

Real delay:0.88889
smithpredictor
PI
output
time
0
0.2
0.4
0.6
0.8
1
1.2
1.4
10
20
30
40
50

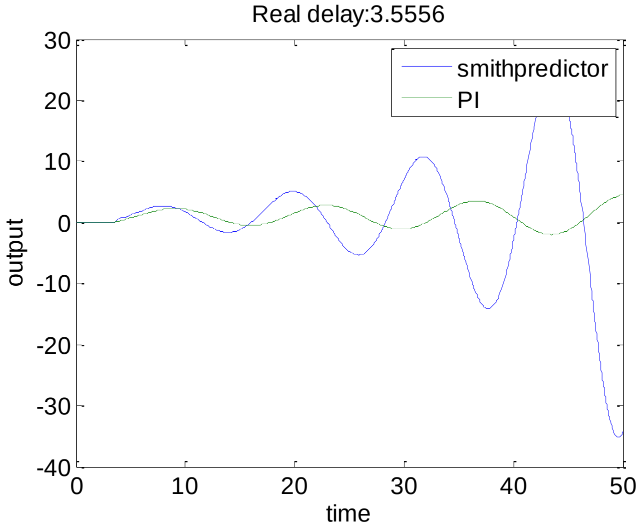

Real delay:1.3333
smithpredictor
PI
output
time
0
0.2
0.4
0.6
0.8
1
1.2
1.4
10
20
30
40
50

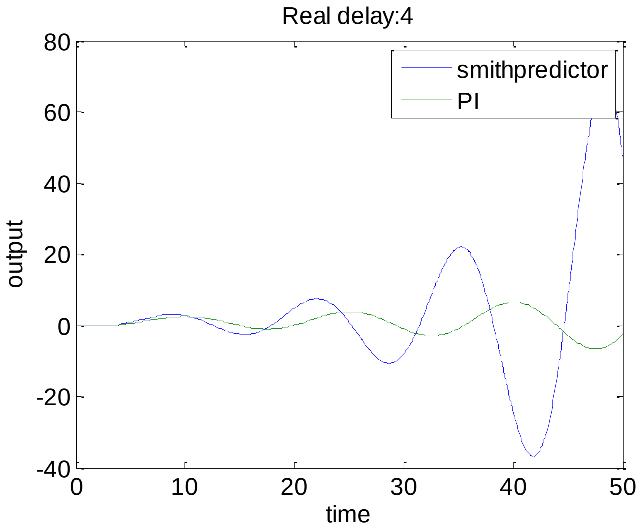

Real delay:1.7778
smithpredictor
PI
output
time
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
10
20
30
40
50


Real delay:2.2222
smithpredictor
PI
output
time
0
10
20
30
40
50
0
0.5
1
1.5
2


Real delay:2.6667
smithpredictor
PI
output
time
0
10
20
30
40
50
-0.5
0
0.5
1
1.5
2
2.5


Real delay:3.1111
smithpredictor
PI
output
time
0
10
20
30
40
50
-10
-5
0
5
10
15


Real delay:3.5556
smithpredictor
PI
output
time
30
20
10
0
-10
-20
-30
-40
0
10
20
30
40
50


Real delay:4
smithpredictor
PI
output
time
80
60
40
20
0
-20
-40
0
10
20
30
40
50

Appendix E – Optimal Tuning versus Robustness Plots

Figure 21 and Figure 22 show the optimal tuning plotted against the robustness in case 1 for the SP and PI controller, respectively.

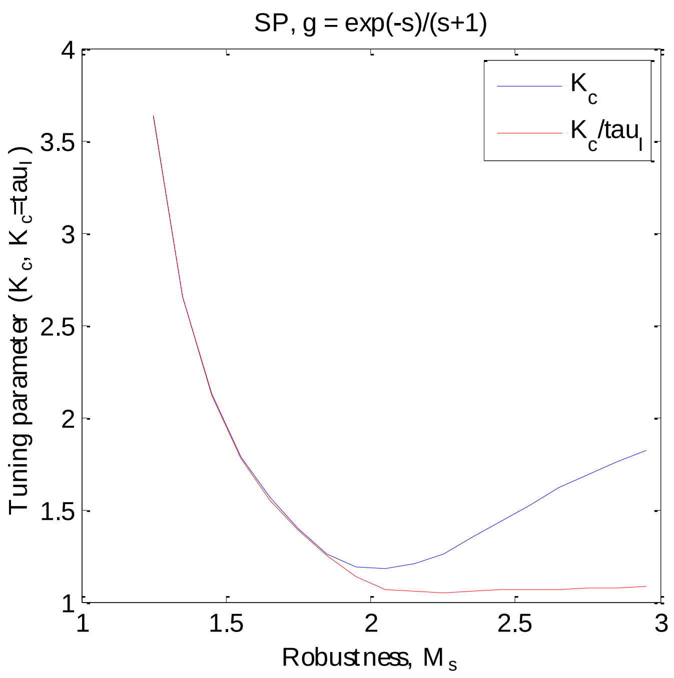


Figure 21: Optimal tuning for the Smith predictor with the process g = e^{-s}/(s+1).

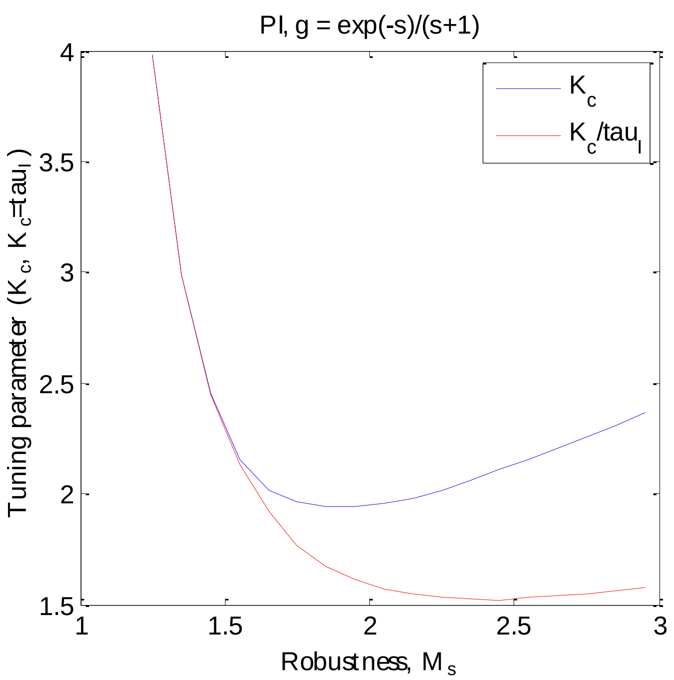


Figure 22: Optimal tuning for the PI controller with the process g = e^{-s}/(s+1).

Figure 23 and Figure 24 show the optimal tuning plotted against the robustness in case 2 for the SP and PI controller, respectively.

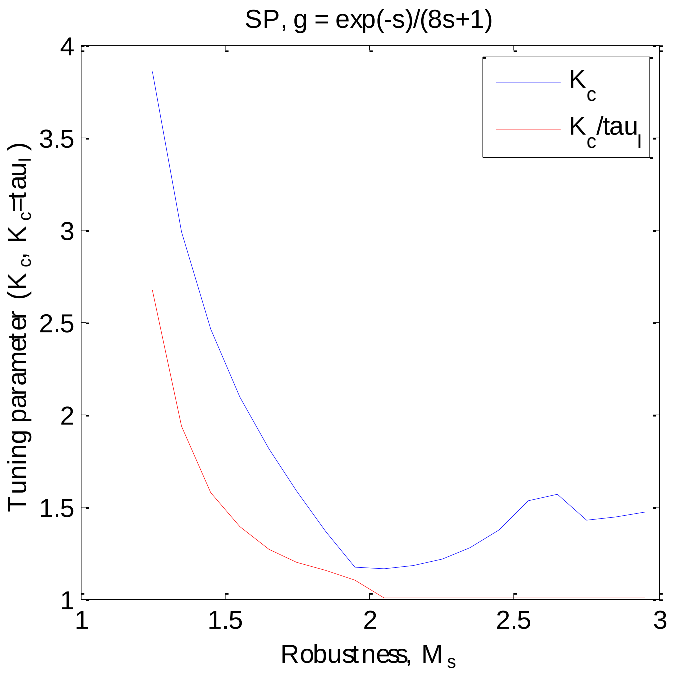


Figure 23: Optimal tuning for the SP controller with the process $g = e^{-s}/(8s+1)$.

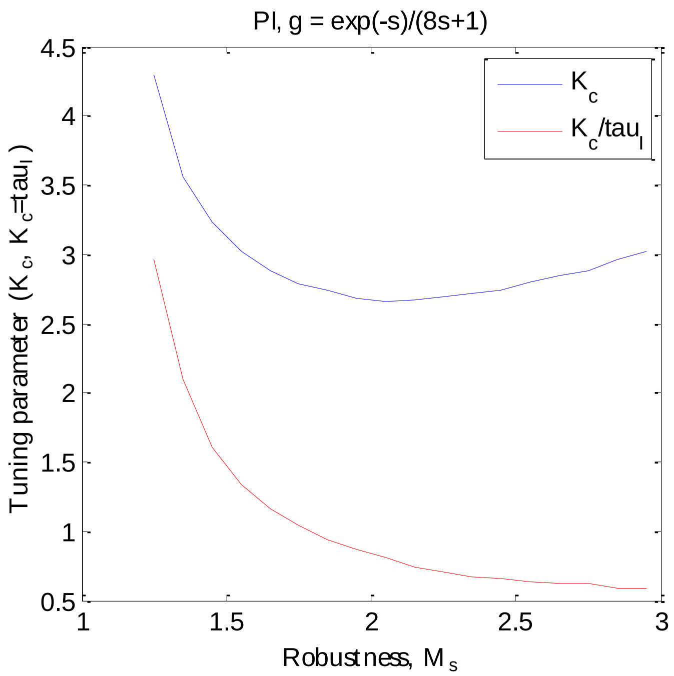


Figure 24: Optimal tuning for the PI controller with the process $g = e^{-s}/(8s+1)$.