University of Wyoming
Laramie, WY. April 29,2017.

CHE 3900 UNDERGRADUATE RESEARCH
SPRING 2017

Rate Predictive Process Control

Sedona Rockwood
Department of Chemical Engineering

Professor: Dr. David M. Bagley

Professor
Department of Chemical Engineering
College of Engineering and Applied Sciences

Table of Contents

1. Introduction

When dealing with chemical processes, control is needed for safety, and quality. Simple chemical processes are well understood, and can be well controlled. However, as chemical systems become more complex controlling them also becomes complex. Since the 1980s, model predictive control has been the standard for controlling complex chemical processes (Kern 2017a). However, model predictive control has several limitations (Kern 2016).

First, model predictive control is expensive, and few cheaper alternatives have been introduced (Kern 2016). Model predictive control uses highly empirical models (Kern 2016). Therefore, it needs to be retuned every time the process conditions change, making model predictive control both time consuming and difficult to operate. Finally, due to the nature of model predictive control, it is also computationally demanding (Kern 2016).

Allan Kern recently patented a new process control system, called rate predictive control (RPC), that addresses the limitations of model predictive process control (Kern 2017b). Instead of using highly empirical models to predict the process behavior, RPC adjusts the controller output based on the rate of change of the controlled variable (Kern 2017b).

RPC is a novel control strategy that may replace model predictive control. Nevertheless, RPC merits further analysis to answer several key questions.

1. What is the underlying control theory of RPC?
2. What are the stability limits of RPC?
3. How large is the practical operational window for key RPC parameters?

The goal of this paper is to examine the theoretical performance of RPC, and answer the three key questions. To examine the theoretical performance of RPC, it was simulated in Excel, MATLAB, and Simulink. Experiments were run in these simulation environments under different conditions. The results from Simulink and Excel were compared, and the underlying control theory of RPC was determined. The stability of RPC was determined in MATLAB under the different conditions. Finally, the results from MATLAB and Excel were compared to determine the practical operational window for key RPC parameters.

2. Background

Model predictive control (MPC) has been used since the 1980s (Kern 2017a). The theory behind MPC is well documented, and is still taught in control classes. According to Riggs (2016), MPC uses available process measurements, and process models to determine values for the desired manipulated variables. MPC can also provide feedforward compensation for disturbance variables (Riggs 2016). Unlike its predecessors, MPC can also control for a large set of constraints by using a set of tuning parameters (Riggs 2016).

A basic literature search on MPC indicates that MPC is widely used throughout the process control industry. While it may have its limitations, MPC is still considered to be the best solution for controlling complex processes (Henson 1998, Kumar 2012). However, Kern (2016) discusses the shortcomings of MPC in industrial practice. For example, process disturbances often alter the models that are the basis for MPC (Kern 2016). This can require costly, and time-consuming retuning. Additionally, MPC focuses on error minimization instead of practical operating precautions, so precautions may not be built into MPC. In industrial practice, however, practical operating precautions take precedence over model error minimization (Kern 2016). Finally, the computations used in MPC often require large matrices. According to Kern (2016), large-matrix control can make the controller larger and difficult to operate, and may provide worse performance than small-matrix control.

Due to the limitations of MPC discussed in Kern (2016), a new method of process control has been developed. Rate predictive control (RPC), and XMC were developed by Kern to address the problems experienced operationally with MPC. These control methods do not require models because they are based on the rate of change of the controlled variable (Kern 2016). The stability of RPC and XMC is independent of the magnitude of the gain because they have built in functions to accommodate changes in the gain (Kern 2017b). Furthermore, to achieve and maintain stability, RPC and XMC will pause until stability is re-achieved (Kern 2017b). RPC and XMC are also both equipped to accommodate deadtime. They do this by not updating the manipulated variables until the effects of a change are evident (Kern 2017b).

RPC, and control theory have some key features. The controlled variable is the variable that has a desired value within a process. For RPC, the controlled variable is called the indirect controlled variable (ICV), (Kern 2017a). The manipulated variable is the variable that can be changed to alter the controlled variable. For RPC, the manipulated variable is called the direct controlled variable (DCV), (Kern 2017a). Process gain is a constant that is determined by the process, and is denoted as K_1. The move is the rate of change of the controlled variable set by the user, and is denoted as K_2. The RPC Band is a user defined constant within RPC, and is denoted as the inverse of K_3. The process response time is determined by the process, and is denoted as τ_1. Kern (2017a) denotes process response time as Filter1. The time delay, or deadtime, is the time required by the system to respond to changes. In this paper, time delay is denoted as t_d. The predicted response time is the predicted value of τ_1, and it is set by the user. The predicted response time is denoted as θ in this paper, and RPC Time by Kern (2017a).

3. Theoretical Development of RPC

3.1 RPC Transfer Functions

Figure 1 shows the control loop used to describe RPC. Figure 1 includes the transfer functions, constants, and controlled variable.

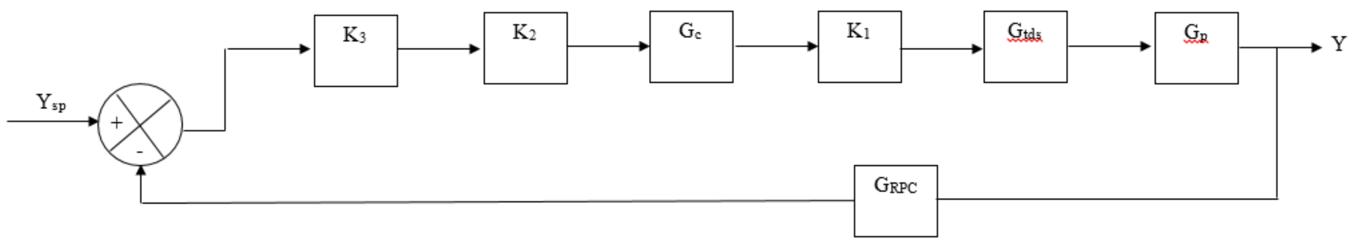

Figure 1: RPC Control Loop

In Figure 1, Y is the controlled variable (CV), Y_{sp} is the set point for Y, G_c is the control transfer function, G_p is the process transfer function, G_{RPC} is the rate predictive transfer function, and G_{tds} is the time delay transfer function. Standard feedback uses the error between Y_{actual} and Y_{sp} as the basis for control action (Equation 1). RPC, however, predicts the next value of Y, $Y_{predicted}$, and uses the error between $Y_{predicted}$ and Y_{sp} as the basis for control action (Equation 2).

$$E = Y_{sp} - Y_{actual} \tag{1}$$

$$E_{predicted} = Y_{sp} - Y_{predicted} \tag{2}$$

Where E is the error, and Y_{actual} is the actual value of the controlled variable. $E_{predicted}$ is the predicted error, and $Y_{predicted}$ is the predicted value of the controlled variable which can be found from Equation 3.

$$Y_{predicted} = G_{RPC} Y_{actual} \tag{3}$$

As indicated in Equation 3, G_{RPC} is the rate predictive transfer function in RPC, and is what makes RPC unique. G_{RPC} is defined in Equation 4.

$$G_{RPC} = \theta s + 1 \tag{4}$$

RPC uses pure integral control, as indicated in Equation 5

$$G_c = \frac{1}{s} \tag{5}$$

3.2 Other Transfer Functions Used for Testing

The other transfer functions, G_{tds} and G_p, were specified as indicated in Equations 6 and 7 to allow for testing.

$$G_{tds} = e^{-t_d s} \tag{6}$$

$$G_p = \frac{1}{\tau_1 s+1} \tag{7}$$

Where Equations 6 and 7 combined are a classic first order plus time delay (FOPTD) transfer function that often represents unknown processes (Seborg 2017). The time constants, t_d and τ_1, are process-specific, and may not be precisely known.

An overall transfer function can now be defined for RPC (Equation 8).

$$\frac{Y}{Y_{sp}} = \frac{K_3 K_2 G_c K_1 G_p}{1+K_3 K_2 G_c K_1 G_p G_{\mathrm{RPC}}} \tag{8}$$

The open loop transfer function (G_{OL}) can be obtained, and is shown in Equation 9.

$$G_{OL} = K_3 K_2 G_c K_1 G_p G_{RPC} \tag{9}$$

By inputting Equations 5, 6, and 7 into Equation 9 and simplifying, Equation 10 can be obtained.

$$G_{OL} = K_3 K_2 K_1 e^{-t_d s} \frac{\theta s+1}{s(\tau_1 s+1)} \tag{10}$$

3.3 Transfer Function Verification

The control loop, and transfer functions were verified by running them in Simulink, and comparing them to an RPC simulator in Excel that was provided by Kern (2017a). Figure 2 shows the comparison between the controlled variables (CV) provided by Mr. Kern's simulator, and Simulink with no time delay, τ_1 equal to 10 minutes, θ equal to 10 minutes, and an inverse K_3 equal to 10 (Kern 2017a). Y_{sp} is a change of 30, i.e., a step change of 30. These conditions were chosen based on values provided in the Excel spreadsheet (Kern 2017a). Figure 3 shows the comparison between CV values from Mr. Kern's simulator, and Simulink with a time delay of 5 minutes, τ_1 equal to 10 minutes, θ equal to 15 minutes, and an inverse K_3 equal to 15. The values for Figure 3 were chosen based on information in the Excel spreadsheet for a time delay of 5 minutes (Kern 2017a).

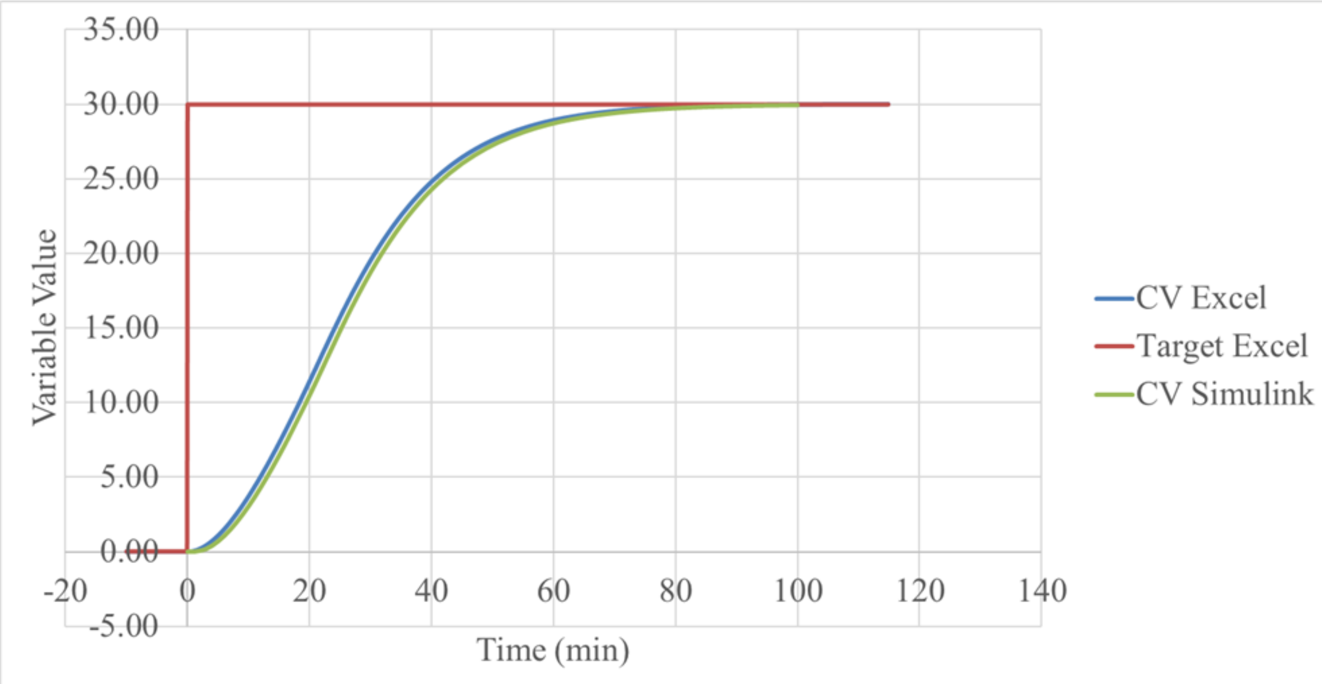

Figure 2: Comparison of RPC Simulator and Simulink for a First Order Process with no Time Delay, τ_1 Equal to 10 Minutes, θ Equal to 10 Minutes, and Inverse K_3 Equal to 10

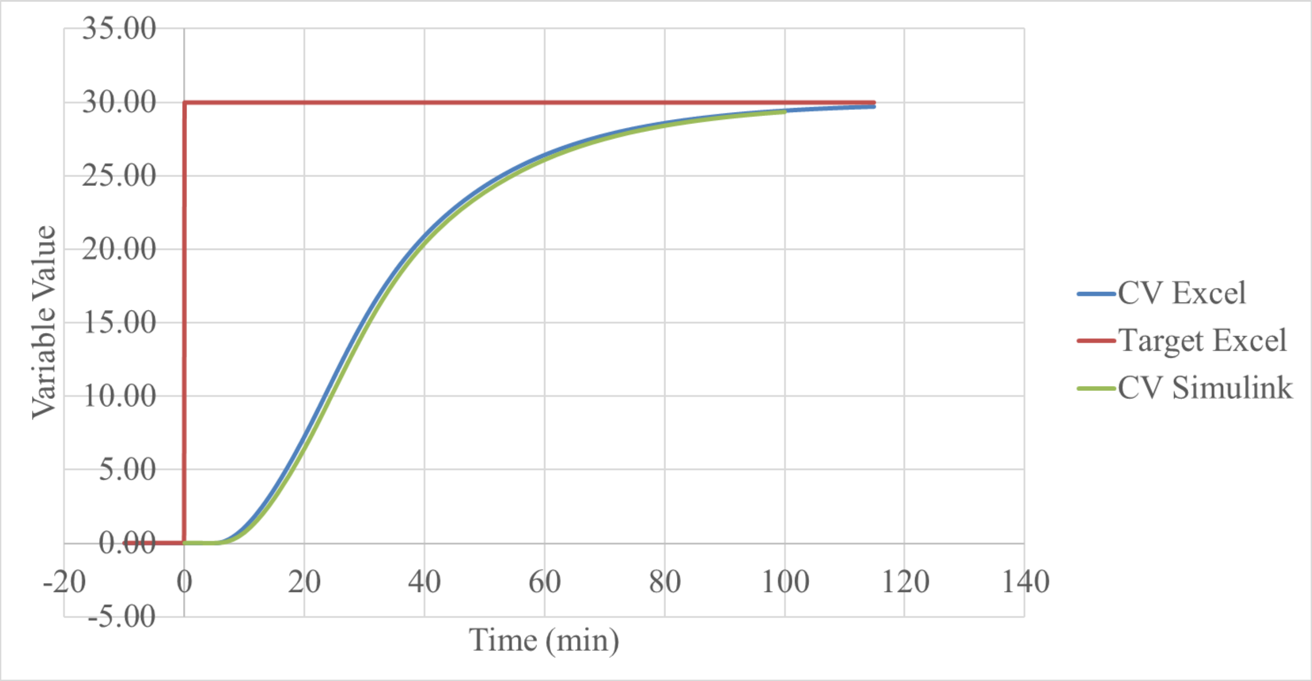

Figure 3: Comparison of RPC Simulator and Simulink for a First Order Process with a Time Delay of 5 Minutes, τ_1 Equal to 10 Minutes, θ Equal to 15 Minutes, and Inverse K_3 Equal to 15

The CV output from Excel increases, and then levels out as it approaches the target value. The CV output from the Simulink, also increases, and then levels out as it approaches the target value. The CV output from Simulink is less than the CV output from the Excel from time 0 minutes to about 60 minutes for Figure 2, and 5 minutes to about 80 minutes for Figure 3. Then the CV output from Simulink merges with the Excel CV output. To model RPC in Simulink, RPC Band was included for the entire time. In contrast, RPC Band is not included until about time 60 minutes for Figure 2, and time 80 minutes for Figure 3. This difference in the modeling explains the difference in the CV outputs before the specified times.

4. Stability Analysis

Stability is an important consideration for examining a control loop. The stability of RPC was tested in MATLAB for a first order system without time delay. Figure 4 shows the Bode diagram for RPC applied to a first order process with no time delay when θ and τ_1 are equal. The equation for G_{OL} simplifies to give Equation 11.

$$G_{OL} = \frac{K_3 K_2 K_1}{s} \tag{11}$$

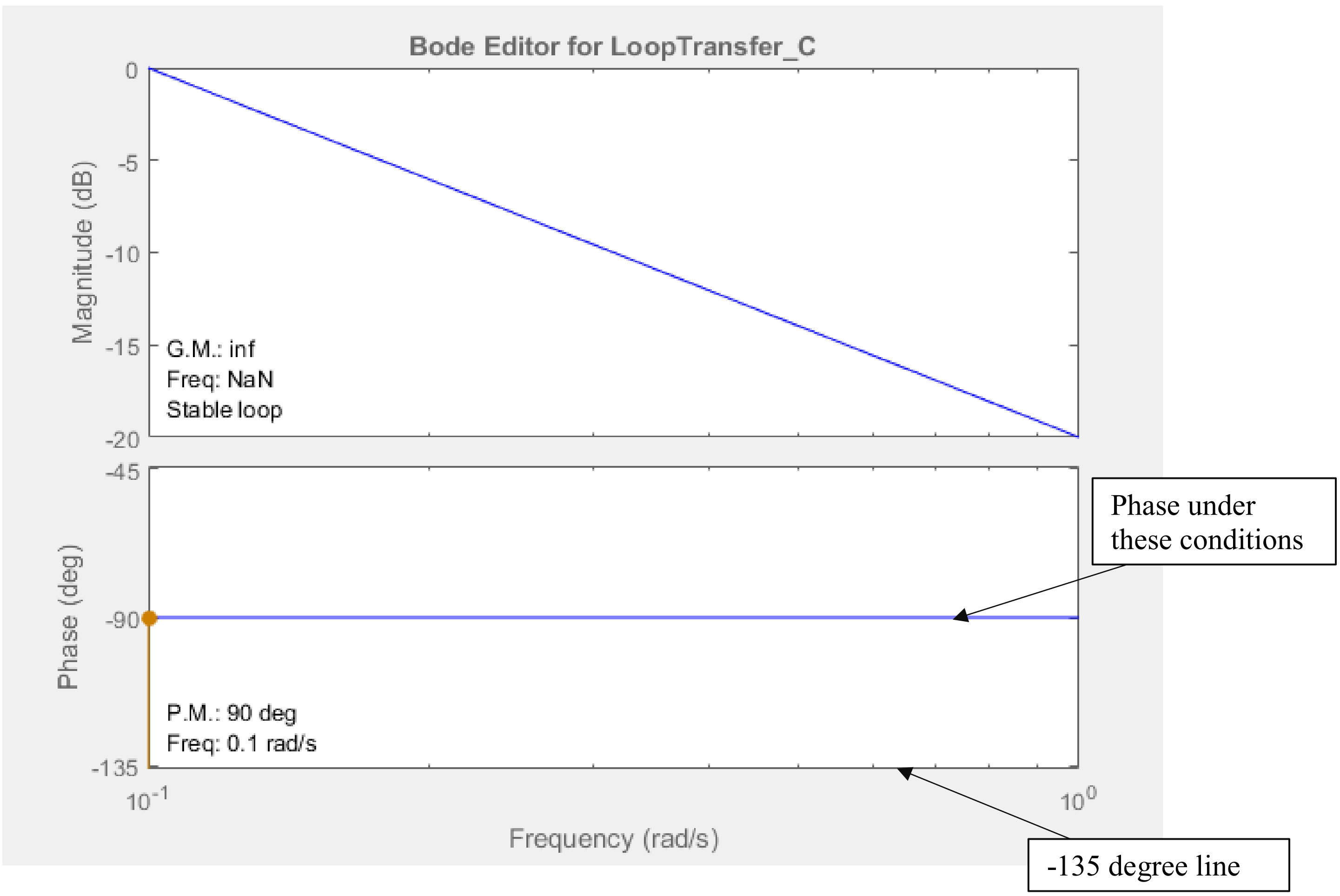

Figure 4: Bode Diagram for RPC with a First Order Function without Time Delay when θ Equals τ_1

As indicated in Figure 4, RPC is always stable when G_p is a first order function with no time delay when θ is equal to τ_1 because the phase is always -90°, and never goes below -180°. It is important to note that this case is only true when G_p is a first order function. If G_p is second order or higher, RPC may not be stable under these conditions.

The stability of RPC was tested in MATLAB for a first order function without time delay when θ and τ_1 are not equal. When θ and τ_1 are not equal, G_{OL} simplifies to Equation 12.

$$G_{OL} = K_3 K_2 K_1 \frac{\theta s+1}{s(\tau_1 s+1)} \tag{12}$$

Figure 5 shows the Bode diagram of RPC for Equation 12 when τ_1 equals 10 minutes, and θ equals 5 minutes.

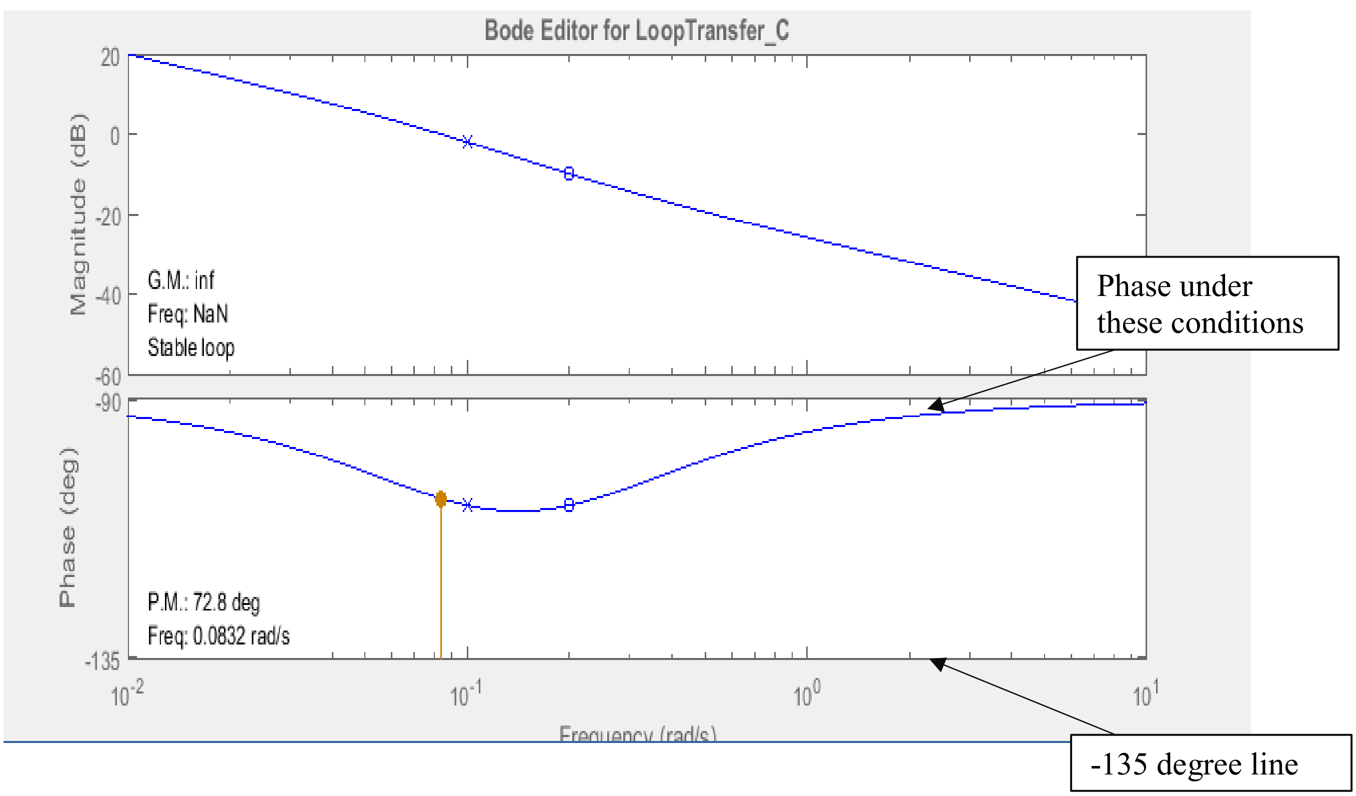

Figure 5: Bode Diagram of RPC for a First Order Process without Time Delay when τ_1 Equals 10 Minutes, and θ Equals 5 Minutes

As indicated in Figure 5, RPC is always stable when G_p is a first order function without time delay when τ_1 equals 10 minutes, and θ equals 5 minutes because the phase never goes below -180°.

Figure 6 shows a Bode diagram of RPC for Equation 12 when τ_1 equals 5 minutes, and θ equals 10 minutes.

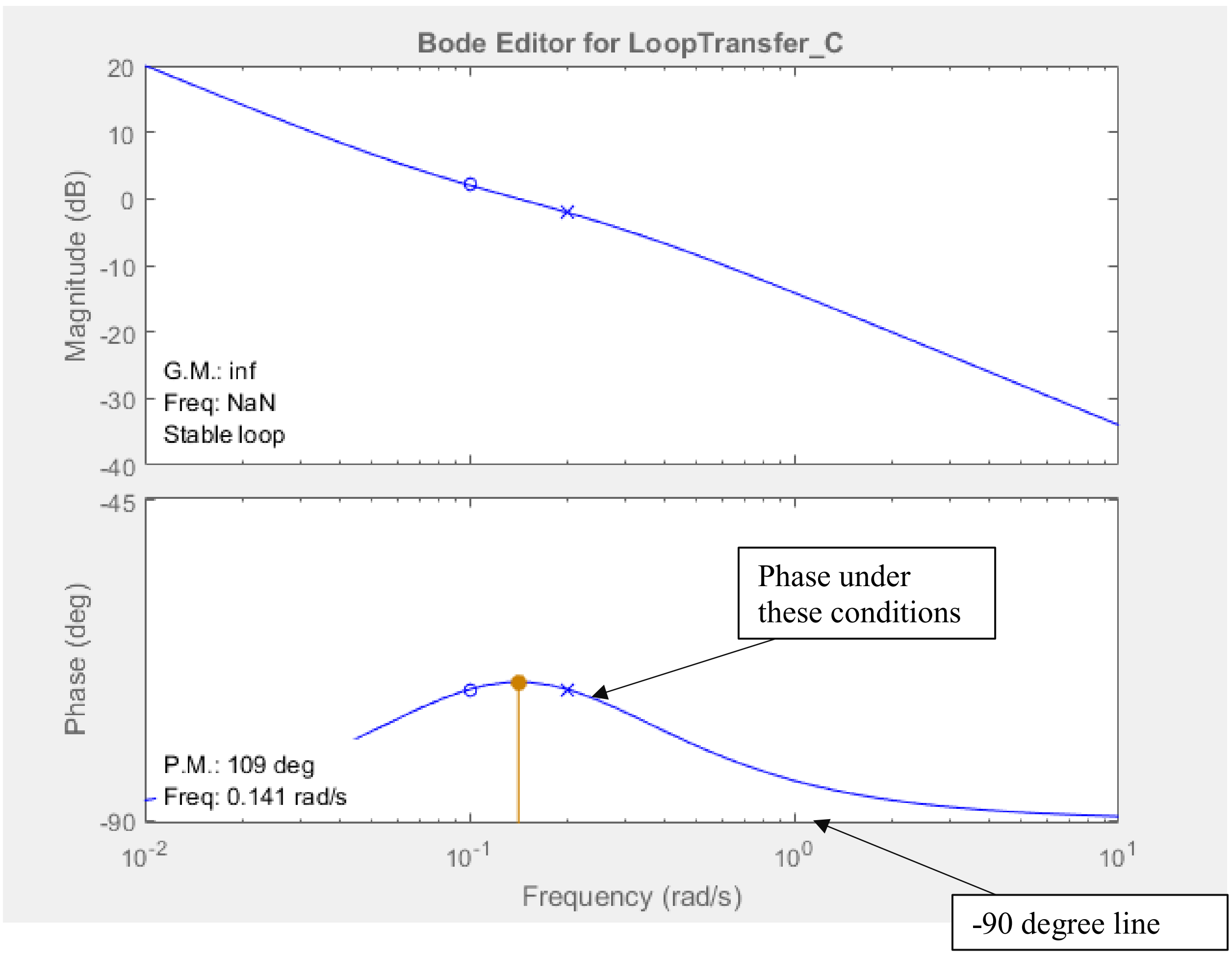

Figure 6: Bode Diagram of RPC for a First Order Process without Time Delay when τ_1 Equals 5 Minutes, and θ Equals 10 Minutes.

As indicated in Figure 6, RPC is always stable when G_p is a first order function without time delay when τ_1 equals 5 minutes, and θ equals 10 minutes because the phase never drops below -180°. Control theory states that open loop transfer functions that are second order overall without a time delay should always be stable. For a first order process without time delay, RPC produces a second order open loop transfer function, and should always be stable. This is confirmed in Figures 4, 5, and 6, which show the effect of changing θ relative to τ_1.

Instability is introduced into RPC for a first order process when a time delay is added to the system. Figure 7 shows the Bode diagram for RPC applied to a first order process with a time delay of 5 minutes when θ equals τ_1.

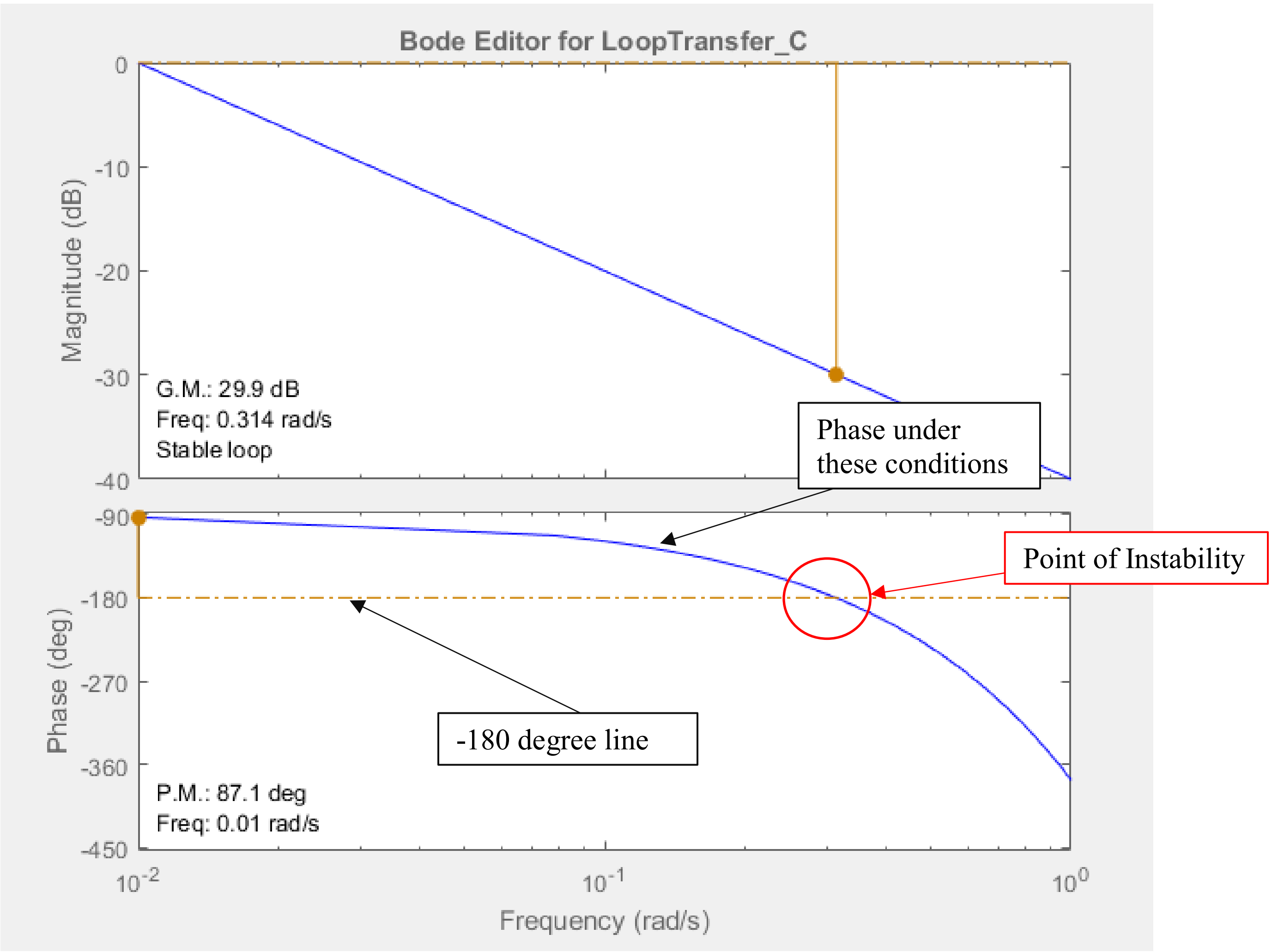

Figure 7: Bode Diagram of RPC for a First Order Process with a Time Delay of 5 Minutes, and when θ Equals τ_1

In Figure 7, the system is stable until the phase is less than or equal to -180°. For the specific conditions in Figure 7, RPC becomes unstable when $K_1K_2K_3$ is greater than or equal to 3. Figure 8 shows the CV response for RPC applied to a first order process with a time delay of 5 minutes, θ equal to τ_1, and $K_1K_2K_3$ equal to 0.0667, which is below the limit of 3. The CV in Figure 8 reaches the target without increasing oscillation, so it is stable. Figure 9 shows the CV for RPC applied to a first order process with a time delay of 5 minutes, θ equal to τ_1, and $K_1K_2K_3$ equal to 10, which is above the limit of 3. The CV in Figure 9 reaches the target, but begins to oscillate. The oscillations in Figure 9 are increasing, but they are increasing slowly, so the increase in oscillation cannot be seen in Figure 9.

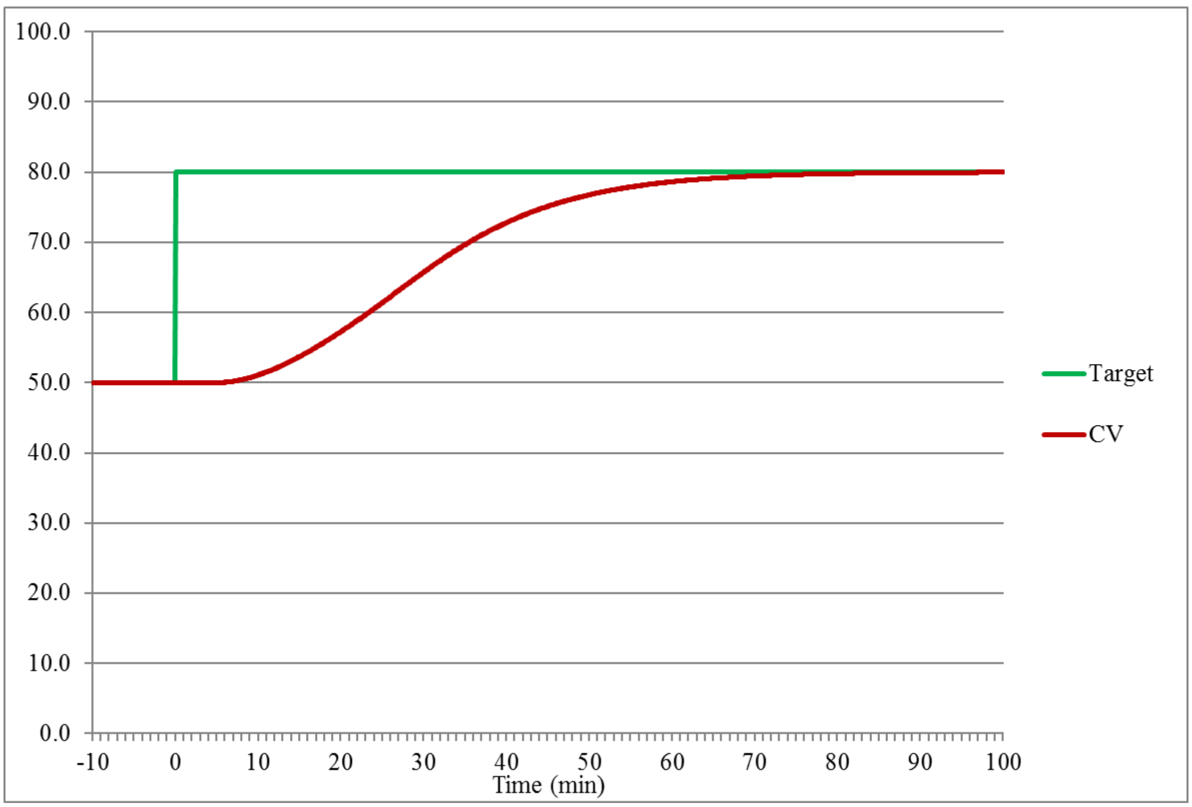

Figure 8: CV with Respect to Time when RPC is Stable for a First Order Process with a 5 Minute Time Delay when θ Equals τ_1, and $K_1K_2K_3$ is 0.0667

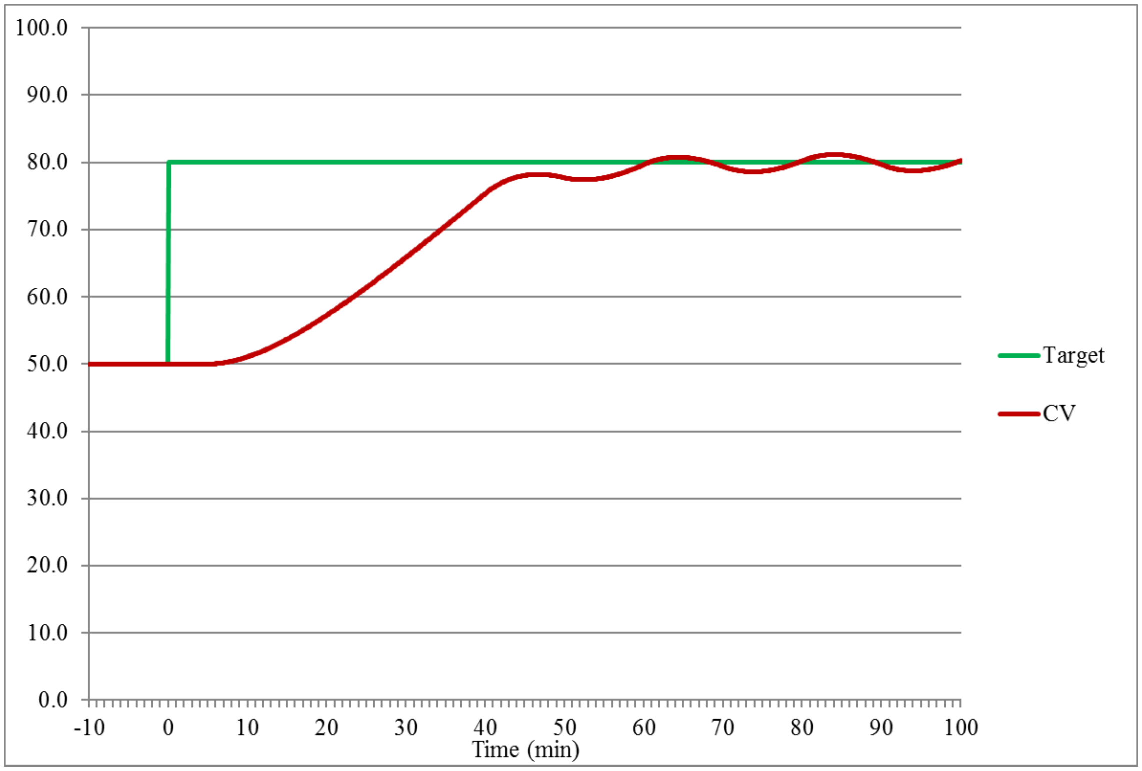

Figure 9: CV with Respect to Time when RPC is Unstable for a First Order Process with a 5 Minute Time Delay when θ Equals τ_1, and $K_1K_2K_3$ is 10

The stability range for $K_1K_2K_3$ can be determined as a function of time delay (t_d). Figure 10 shows the maximum $K_1K_2K_3$ value allowable for stability of RPC applied to a first order process with respect to t_d when θ and τ_1 are equal. A best-fit curve was fit to the plot in Figure 10. The equation of the best-fit curve is given in Equation 13.

$$y = 14.471x^{-0.998} \qquad R^2 = 0.9987 \qquad (13)$$

Where y is $K_1K_2K_3$, and x is t_d.

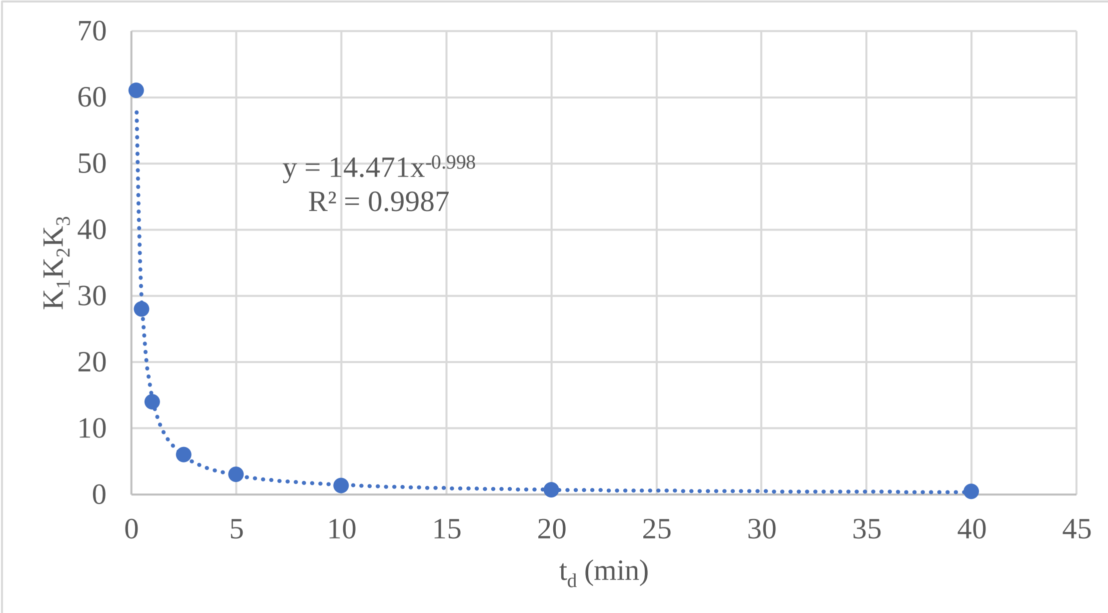

Figure 10: Maximum Allowable $K_1K_2K_3$ for Stability in RPC for a First Order Process with Respect to t_d when θ and τ_1 are Equal

A similar stability analysis can be done for $\theta < \tau_1$. Figure 11 shows the plot of the maximum $K_1K_2K_3$ value allowable for stability of RPC applied to a first order process when τ_1 equals 10 minutes, and θ equals 5 minutes. The best-fit curve is provided in Equation 14.

$$y = 25.79x^{-1.16} \qquad R^2 = 0.9975 \tag{14}$$

Where y is $K_1K_2K_3$, and y is t_d.

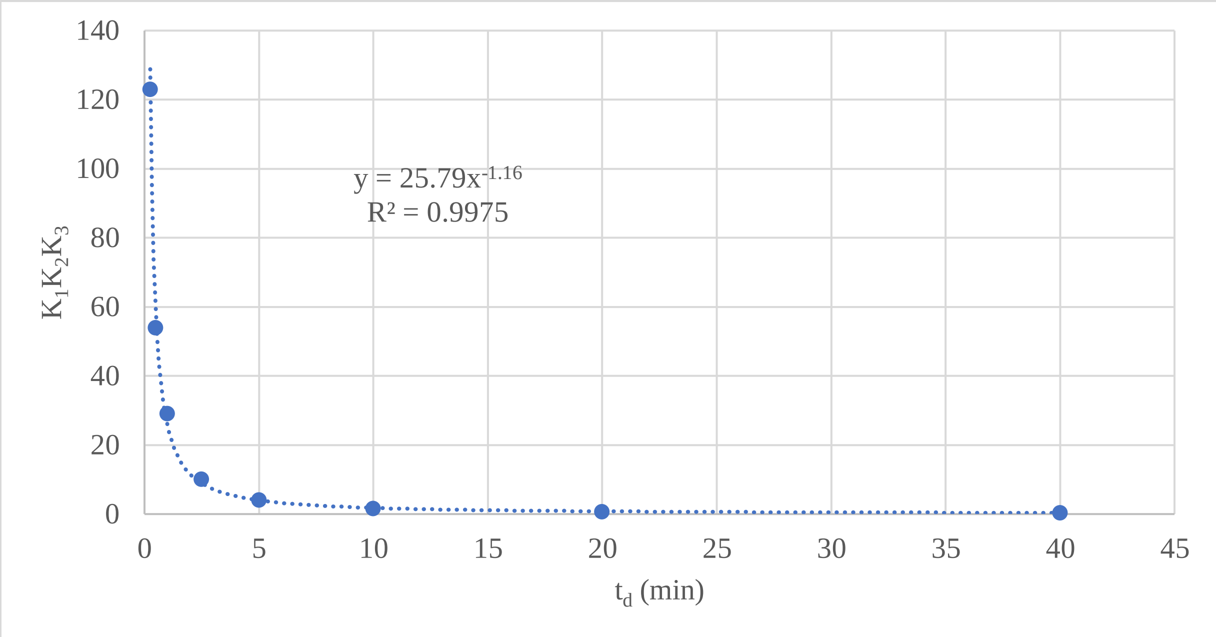

Figure 11: Maximum Allowable $K_1K_2K_3$ for Stability in RPC for a First Order Process with Respect to t_d when τ_1 Equals 10 minutes, and θ Equals 5 Minutes

A similar stability analysis can also be done for $\theta > \tau_1$. Figure 12 shows the maximum $K_1K_2K_3$ values allowable for stability of RPC applied to a first order process when τ_1 equals 5 minutes, and θ equals 10 minutes. The best-fit curve is shown in Equation 15.

$$y = 8.4108x^{-0.854} \qquad R^2 = 0.9942 \tag{15}$$

Where y is $K_1K_2K_3$, and x is t_d.

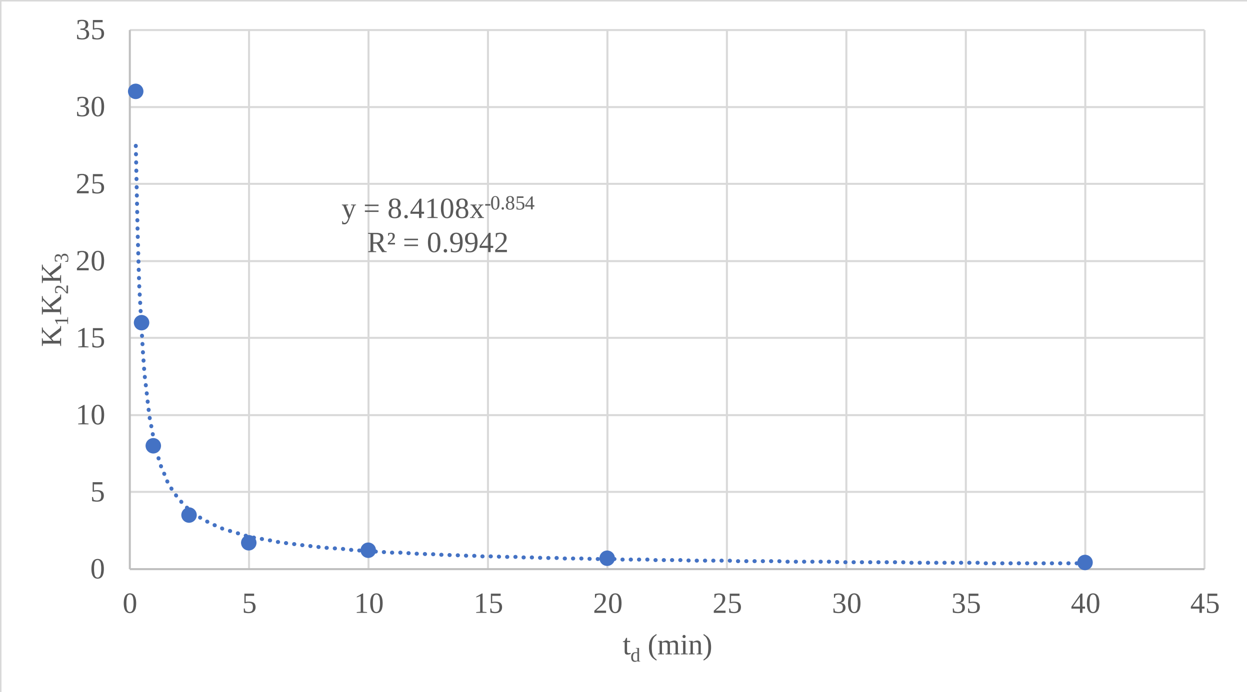

Figure 12: Maximum Allowable $K_1K_2K_3$ for Stability in RPC for a First Order Process with Respect to t_d when τ_1 Equals 5 Minutes, and θ Equals 10 Minutes

Table 1 summarizes the maximum allowable $K_1K_2K_3$ for stability of RPC under the different conditions tested.

Table 1: Maximum Allowable $K_1K_2K_3$ for Stability of RPC for a First Order Process with Respect to t_d for all Tested Conditions

$\theta=\tau_1$		$\theta=5$ $\tau_1=10$		$\theta=10$ $\tau_1=5$	
t_d (min)	$K_1K_2K_3$	t_d (min)	$K_1K_2K_3$	t_d (min)	$K_1K_2K_3$
0.25	61	0.25	123	0.25	31
0.5	28	0.5	54	0.5	16
1	14	1	29	1	8
2.5	6	2.5	10	2.5	3.5
5	3	5	4	5	1.7
10	1.3	10	1.6	10	1.2
20	0.7	20	0.7	20	0.7
40	0.4	40	0.4	40	0.4

As seen in Table 1, when t_d is 0.25 minutes, the maximum allowable value of $K_1K_2K_3$ for stability changes between the conditions. When τ_1 equals 10 minutes, and θ equals 5 minutes, the maximum allowable value of $K_1K_2K_3$ for stability at t_d of 0.25 minutes is about twice as large as when $\theta = \tau_1$. In contrast, when τ_1 equals 5 minutes, and θ equals 10 minutes, the maximum allowable value of $K_1K_2K_3$ for stability at t_d of 0.25 minutes is about two times smaller than when $\theta = \tau_1$ Furthermore, when t_d increases to 40 minutes, the maximum allowable $K_1K_2K_3$ for stability under all of the conditions tested converges to 0.4. It is important to note that K_2 and K_3 are both user defined functions, whereas, K_1 is a process defined function.

5. Operational Issues

While stability in process control systems is important, a system can be stable, but still be undesirable from an operational stand point. For example, a system may be stable when it oscillates, but oscillation is operationally undesirable. Kern (2017a) provides operational guidelines for RPC that are intended to keep RPC within operational limits. Kern's guidelines are shown in Equations 16, and 17.

$$\theta \geq t_d + \tau_1 \tag{16}$$

$$\frac{1}{K_3} \geq K_1 K_2 \theta \tag{17}$$

Because there is likely to be a time delay while operating RPC, the operational response of RPC was only tested with the presence of a deadtime. Also, for testing purposes, K_1 and K_2 were assumed to be equal to 1.

In Figure 13, the manipulated variable (MV) and the controlled variable (CV) are plotted with respect to time for a time delay of 5 minutes, a τ_1 of 10 minutes, a θ set to 15 minutes, and an inverse K_3 set to 15 (which follows the guidelines shown above). Remember that for stability in this case, the K_3 should be no greater than 3.

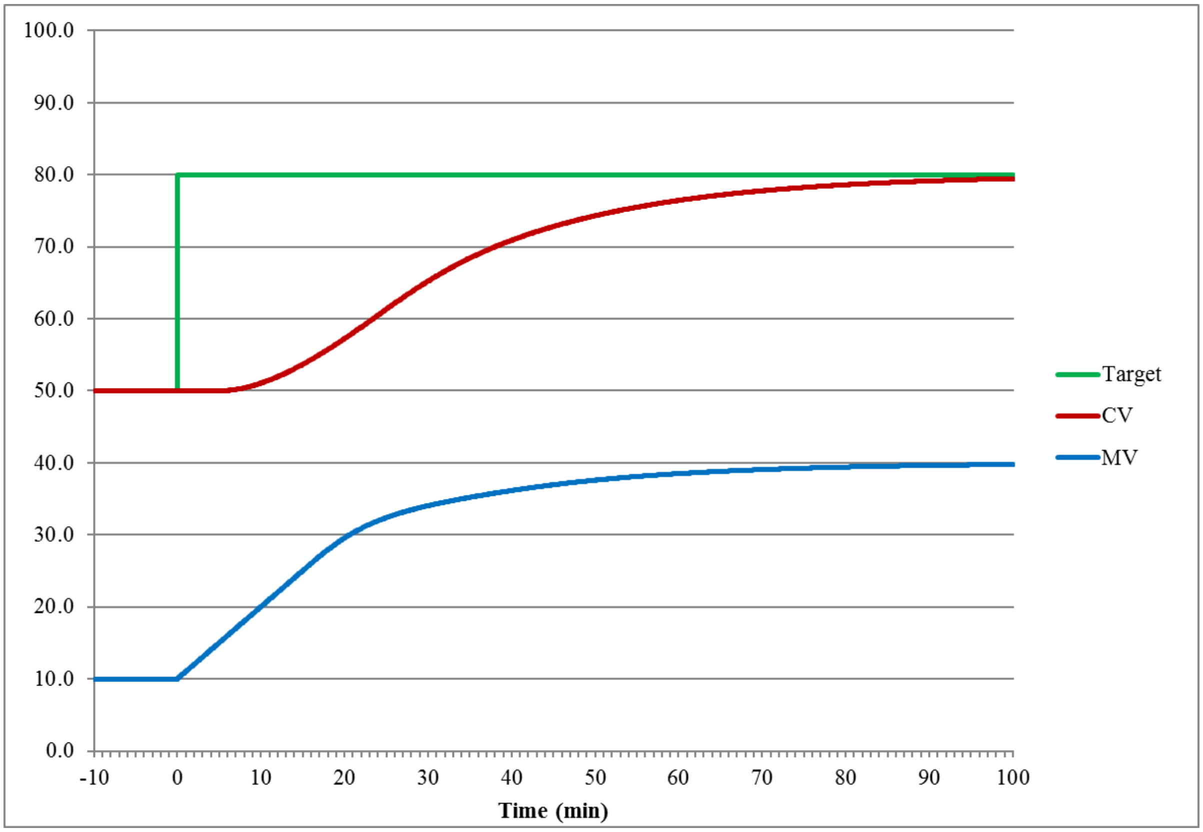

Figure 13: CV and MV with Respect to Time for a Time Delay of 5 minutes, a τ_1 of 10 Minutes, a θ of 15 Minutes, and an Inverse K_3 of 15

The MV changes smoothly toward a new operating condition as does the CV, which also does not overshoot the new target, or oscillate. For this case, Kern's guidelines are operationally sound.

Figure 14 shows a plot of the CV and MV with respect to time under the same conditions as Figure 13, except the inverse of K_3 is 10, which is less than the inverse K_3 recommended by the guideline, but is still greater than the stability limit.

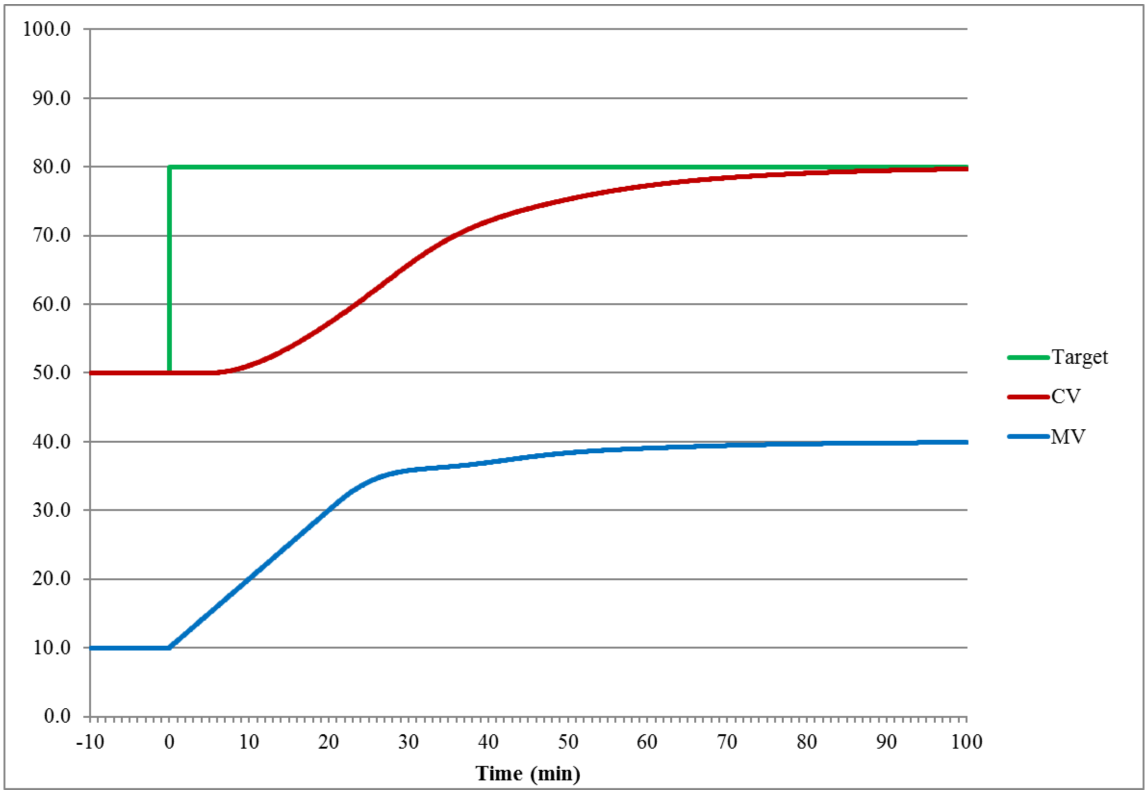

Figure 14: CV and MV with Respect to Time for a Time Delay of 5 minutes, a τ_1 of 10 Minutes, a θ of 15 Minutes, and an Inverse K_3 of 10

Figure 14 shows that even when the inverse of K_3 is under the recommended value, CV still reaches the target, and does not oscillate or overshoot. MV is also smooth. Therefore, for a typical operational time delay of 5 minutes, the guideline is conservative. In Figure 13, CV reaches 78.5 (95% of the target) in 78.5 minutes. In Figure 14, CV reaches 78.5 in 71.2 minutes. Therefore, under these conditions, CV reaches the target faster when the inverse of K_3 equals 10.

Figure 15 is a plot of the CV and MV with respect to time under the same conditions as Figures 13 and 14, except the inverse of K_3 is set to 1, which is less than the guideline, but still within the stability limit.

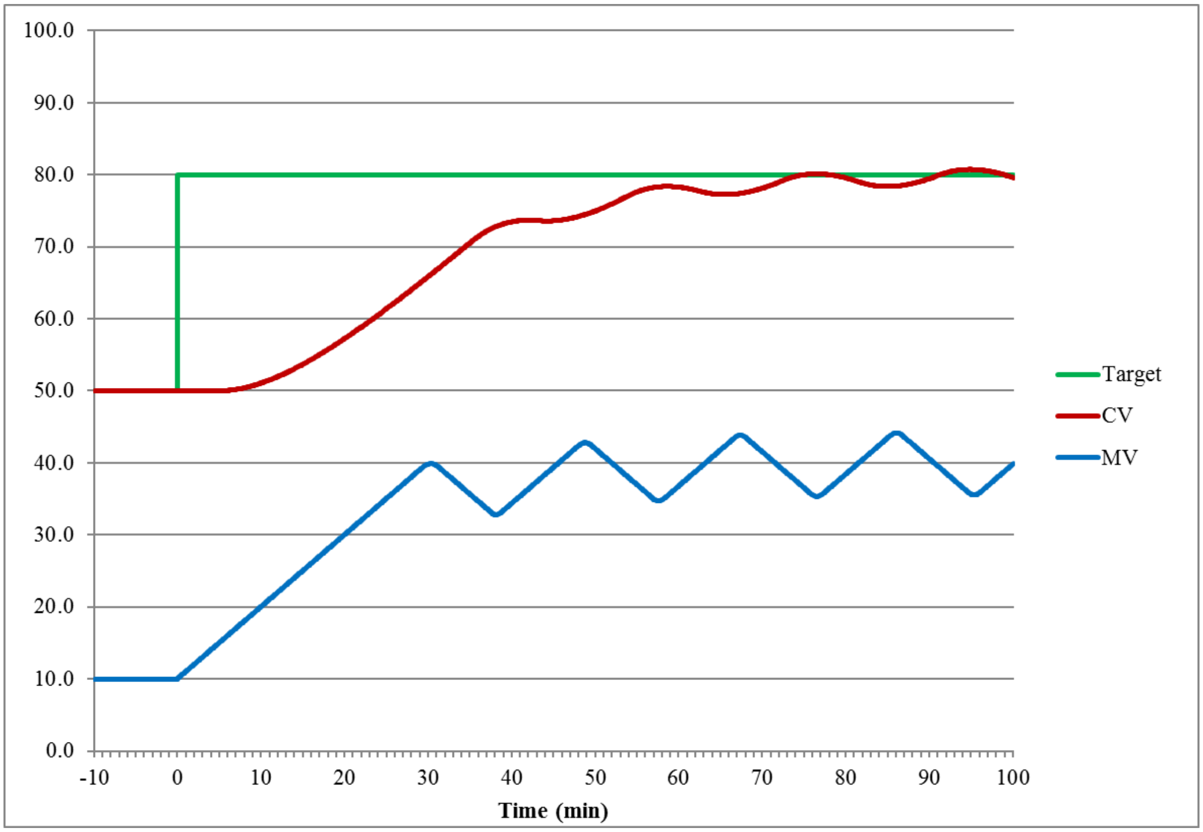

Figure 15: CV and MV with Respect to Time for a Time Delay of 5 minutes, a τ_1 of 10 Minutes, a θ of 15 Minutes, and an Inverse K_3 of 1

The CV, in Figure 15, has oscillation and overshoots the target, both of which are operationally undesirable. The MV oscillates sharply in Figure 15, which is also operationally undesirable. However, RPC is stable under these conditions because CV does not have increasing oscillations. In Figure 15, CV reaches 78.5 in 86.6 minutes, so under these conditions the CV reaches the target slower than in both of the previous conditions.

Changes in θ can also cause operational issues. Figure 16 shows a plot of the CV and MV output of RPC for a first order system with a 5 minute time delay, τ_1 equal to 10 minutes, θ equal to 10 minutes (which is less than the value recommended by the guideline), and the inverse of K_3 equal to 15.

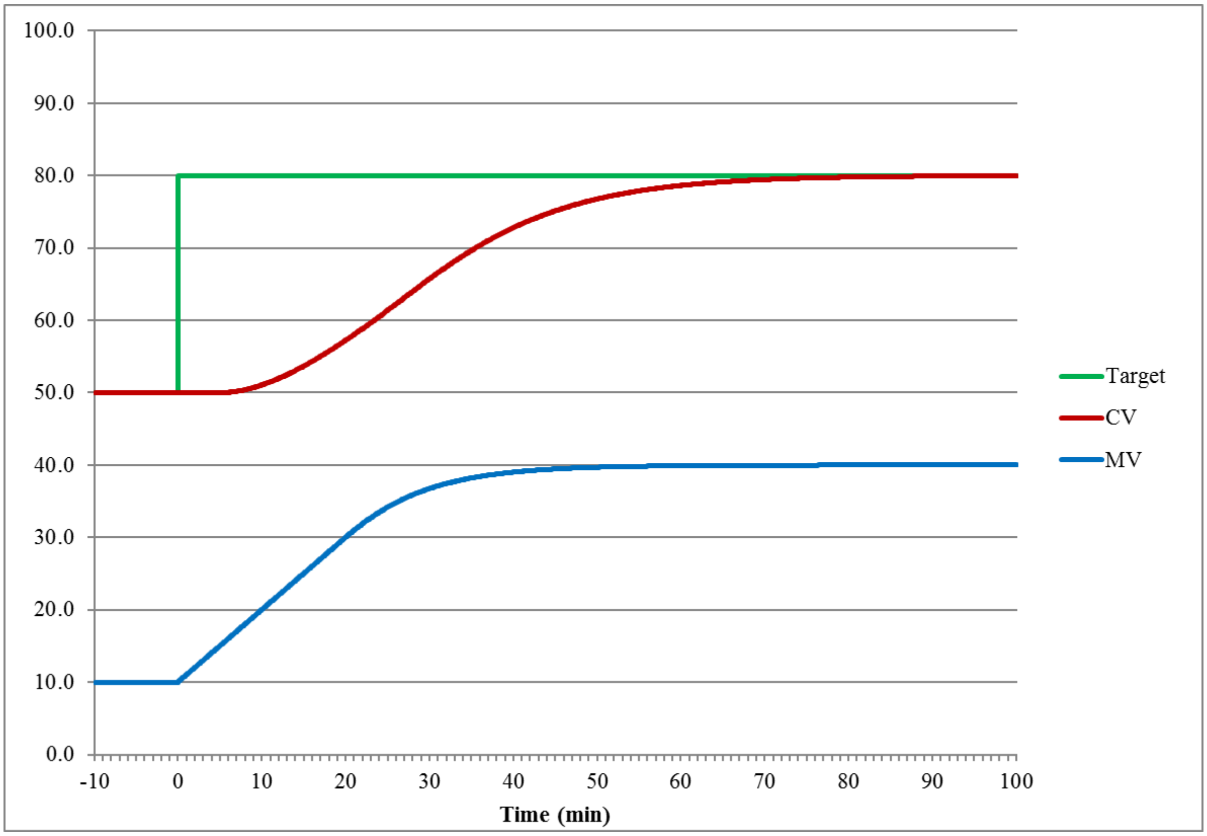

Figure 16: CV and MV with Respect to Time for a Time Delay of 5 minutes, τ_1 of 10 Minutes, a θ of 10 Minutes, and an Inverse K_3 of 15

The CV in Figure 16 is still operationally desirable because there is no overshoot or oscillation. The MV in Figure 16 is also operationally desirable because it is smooth. In Figure 16, CV reaches 78.5 in 59.0 minutes, compared to Figure 13 where CV reaches 78.5 in 78.5 minutes. Therefore, under these conditions, CV reaches the target faster when θ equals 10 than when θ equals 15 minutes.

Figure 17 Shows RPC for a first order system with the same conditions as Figure 13 and 16, except θ is equal to 1 minute.

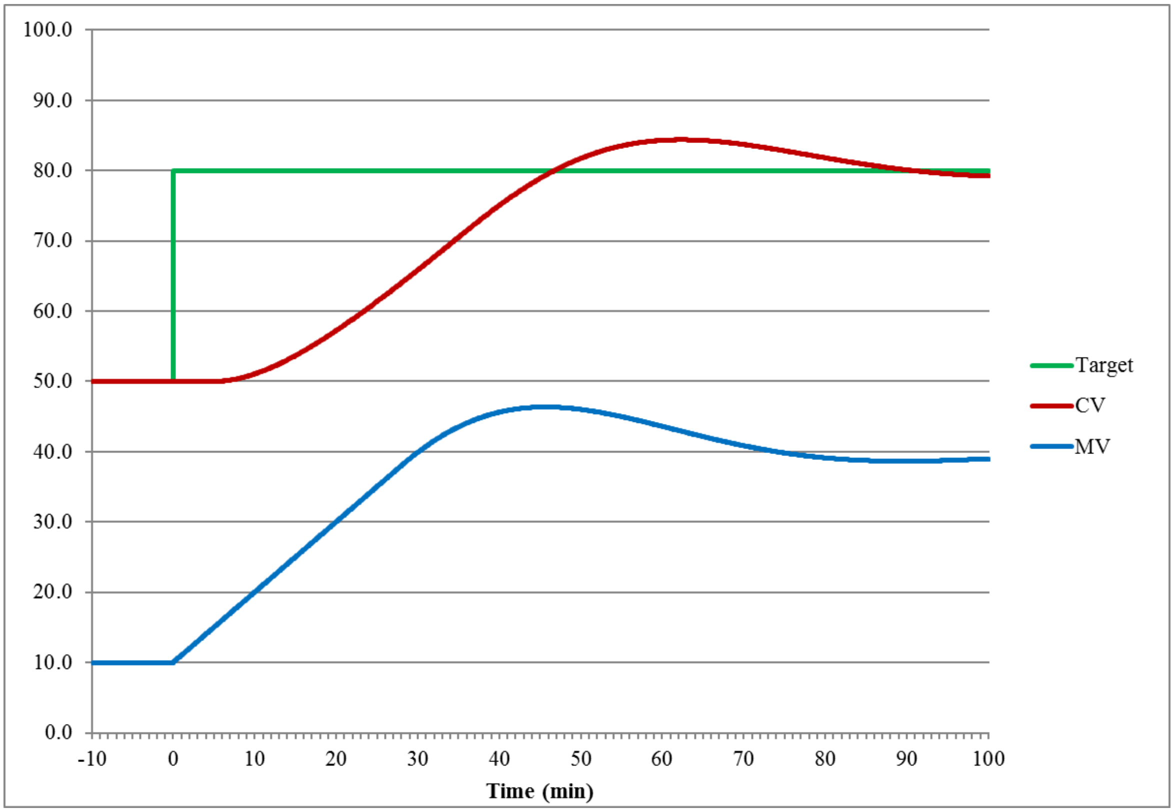

Figure 17: CV and MV with Respect to Time for a Time Delay of 5 minutes, τ_1 of 10 Minutes, a θ of 1 Minute, and an Inverse of K_3 of 15

In Figure 17, the CV overshoots the target, which is operationally undesirable. The CV in Figure 17 reaches 78.5 in 44.4 minutes. Therefore, with a θ of 1 minute, CV reaches the target faster than it does when θ is within the guidelines for these conditions.

6. Summary, Conclusions, and Recommendations

6.1 Summary

The overall control loop, and transfer functions for RPC were determined by simulating RPC in Simulink. Once the overall control loop and transfer functions for RPC were determined, they were verified by plotting the control variable output in Excel and Simulink. The plots were compared. Once the overall control loop and transfer functions were verified, the stability of RPC was analyzed in MATLAB. The maximum values of $K_1K_2K_3$ for stability were determined with respect to time delay under different operating conditions. The operational issues of RPC were addressed by running simulations in Excel. Operational guidelines provided by Mr. Kern were tested for operational stability.

6.2 Conclusions

The overall control loop and transfer functions simulated in Simulink seem to match the Excel results closely. For a first order process without time delay, RPC is inherently stable. For a first order process with time delay, RPC is stable within a certain range of constants. The range of constants for which RPC is stable depends on the amount of time delay (a process-specified variable), the process response time (τ_1) (a process-specified variable), and the RPC time (θ) (a user-specified variable). Even if RPC is stable, it may not be operationally ideal. The guidelines for K_3 and θ are appropriate within operational limits, if K_2 is assumed to be 1. The guidelines are conservative, but provide a good basis for setting the constant values.

6.3 Recommendations

In the future, RPC should be tested for higher order processes. Testing RPC for first order processes is a good starting point, but the behavior of RPC may change for higher order processes. The theory of RPC should be tested using a second order process and possibly a third order process.

A more in depth comparison of RPC and MPC should also be conducted in the future. Now that some of theory behind RPC has been determined, real industrial process data could be used in RPC and MPC, and the theoretical performance of both could be compared. RPC could also be implemented into an existing system that has previously used MPC. The results between the two could then be compared.

Resources

Henson, M. A. (1998). *Nonlinear model predictive control: current status and future directions [Abstract]*. Computers & Chemical Engineering, 23(2), 187-202. doi:10.1016/s0098-1354(98)00260-9

Kern, A. (2016). *The history and possible future of model-less multivariable control.* Hydrocarbon Processing.

Kern, A. (2017)a. Personal Communications.

Kern, A. (2017)b. U.S. Patent No. 9563184 B2. *Rate-Based Multivariable Control with Stability Assurance.* Washington, DC: U.S. Patent and Trademark Office.

Kumar, A. S., & Ahmad, Z. (2012). *Model Predictive Control (MPC) And Its Current Issues In Chemical Engineering.* Chemical Engineering Communications, 199(4), 472-511. doi:10.1080/00986445.2011.592446

Riggs, J. B., & Karim, M. N. (2016). *Chemical and bio-process control*. Austin, TX: Ferret Publishing.

Seborg, D. E., Edgar, T. F., Mellichamp, D. A., & Doyle, F. J. (2017). *Process dynamics and control.* Hoboken, NJ: John Wiley & Sons, Inc.