University of Alberta Computer Process Control Group

Stiction Compensation

CPC Group, University of Alberta

Table of Contents

List of Figures

List of Tables

Introduction

The valve stiction compensation tool was developed by the Computer Process Control Group at the University of Alberta to compensate valve stiction by changing the controller tuning.

A "Quick Start" approach to using this tool will be presented, along with a detailed section of full explanations and examples for using this tool.

System Requirements

MATLAB 2006a (7.1) or better is required to run this tool properly.

Quick Start

Follow the steps below to get quickly started:

1) Extract the files to a desired location from the package.
2) Start MATLAB, and navigate to the directory that the files have been extracted to.
3) At the command prompt, type in "valve_stiction" and press Enter. The GUI shown in Figure 1 should appear.

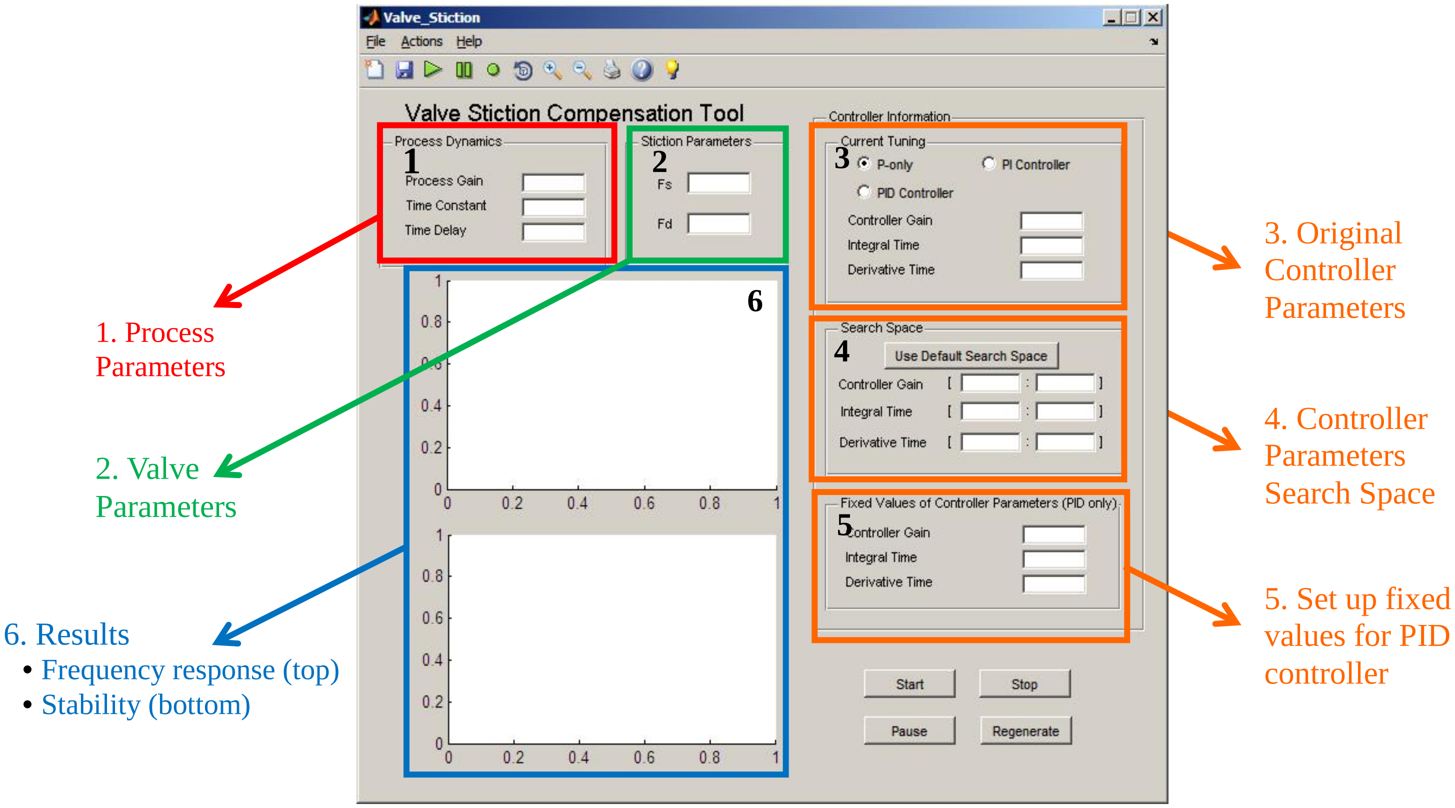


Figure 1: Valve stiction compensation user interface

4) Enter each value of the process parameters in the Process Dynamics block (1) and stiction parameters in the Stiction Parameters block (2).
5) Choose an appropriate Controller Type from the ones shown in the Current Tuning section of the Controller Information block and enter the current controller parameters in the Current Tuning section (3).
6) Select the search space for the controller tuning. Most often, the default values (selected by clicking the "Use Default Search Space" button) are sufficient. If necessary, the search space can be customized by entering the appropriate values in the Search Space block (4).
7) Hit the Start button and the results will be shown in the GUI windows (for PID controller, two larger graphs will be shown in two separate windows) when the progress bar reaches 100%.
8) For the PID controller only, the figures can be regenerated based on the fixed values entered in "Fixed Values of Controller Parameters (PID only)" section (5) after the initial run. Please note, that by default, the initial run uses the original controller parameters as the fixed values for PID controller. These values are required in order to convert the four-dimensional space into a visualisable three-dimensional space.
9) After a few minutes, one frequency response plot and one stability and oscillation plot will be generated as shown in Figure 2.

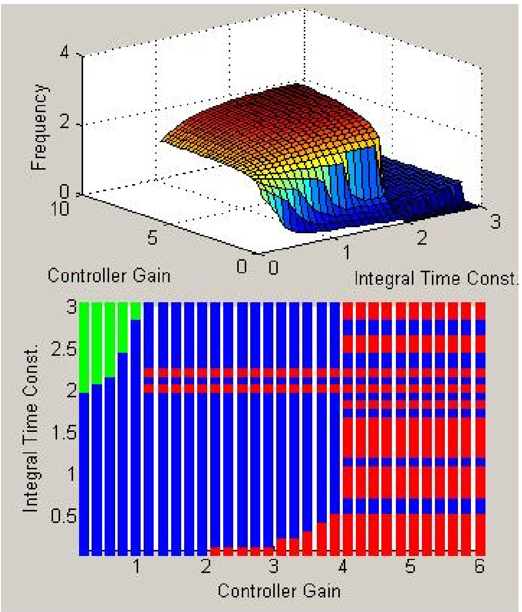


Figure 2: A results of the GUI

The GUI provides two plots, frequency response plot and stability and oscillation plot (shown in Figure 2 from top to bottom). In the frequency response plot, the frequency response is plotted as a function of the controller gain, integral time, and derivative time for PID controllers. For PI controllers, it is only as a function of the controller gain and integral time, while for a proportional controller, then only as a function of the controller gain. Likewise, in the stability and oscillation plots, the stability and oscillation properties are plotted against the controller parameters as labelled on the graph. The interpretation of the stability and oscillation plots is shown in Table 1 below.

Table 1: Stability and oscillation plot interpretation

Colour of the block	Interpretation
Red	Current controller tuning is **unstable**
Blue	Current controller tuning is **stable but oscillatory**
Green	Current controller tuning is **stable and non-oscillatory**

The final controller should be chosen from the green area in the stability and oscillation plot with desired controller gain, integral time and derivative time in order to compensate for valve stiction.

Detailed Instructions

Theory

Valve Stiction, the static friction present in a valve stem, causes problems to the control systems which may affect the product quality, production efficiency, and safety. It can be compensated in various ways, such as vibrating a valve at a certain frequency or replacing a sticky valve. Valve stiction compensation offers a convenient way to resolve the valve stiction problem without interrupting the process or adding any extra costs.

The idea behind valve stiction compensation is to search for a new controller tuning based on the original controller parameters in order to achieve stable and non-oscillatory control over the entire single loop system. A schematic diagram of the single loop system that this tool deals with is shown in Figure 3.

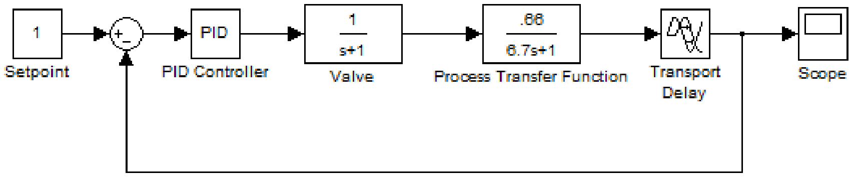


Figure 3: Schematic diagram for a single-loop control system

In the Valve Stiction Compensation Tool, the process is expressed in terms of a first order plus deadtime (FOPDT) model,

$$G_p = \frac{Ke^{-\theta s}}{\tau s + 1} \qquad (1)$$

where K represents the process gain, τ represents the process time constant, and θ represents the deadtime. The proportional, integral, and derivative (PID) controller is of the form

$$G_c = K_c(1 + \frac{1}{\tau_I} + \tau_d s) \qquad (2)$$

where K_c is the controller gain, τ_I is the integral time, and τ_d is the derivative time.

For the purposes of valve stiction compensation, a two-parameter model of valve stiction is used, that is, f_s and f_d are required (Lee, Ren, & Huang, 2008; Choudhury, Shah, Thornhill, & Shook, 2006). In Figure 4, Regions A and B together are referred to as f_s, while Region A is referred to as f_d. The parameters can be obtained using appropriate methods, such as closed-loop valve stiction detection and quantification developed by Lee *et al*. and implemented as a component of the PATS project (Lee, Ren, & Huang, 2008). The PATS project tool is called Closed-loop Valve Stiction Detection and Qualification and can be downloaded from the website.

The default search space of the new controller tuning is defined as 10 times smaller and 3 times greater than the original controller parameters. The search space is partitioned into 30 blocks. For each block, the frequency response and stability of the tuning is generated as the output of this GUI.

Frequency response and describing function are used to study the stability and oscillation of a system with one sticky valve. The describing function is used to explain the relationship between the response of the nonlinear system and its dynamics (in this case, slip-jump and deadband factors for a sticky valve) which is composed of a complex number representing the frequency and magnitude. As well, frequency analysis is also used for determining oscillations and stability of the control loop. More detailed derivations and theory can be found in the thesis *Detection and Compensation For Stiction In Multi-loop Control Systems* (Mohammad, 2011).

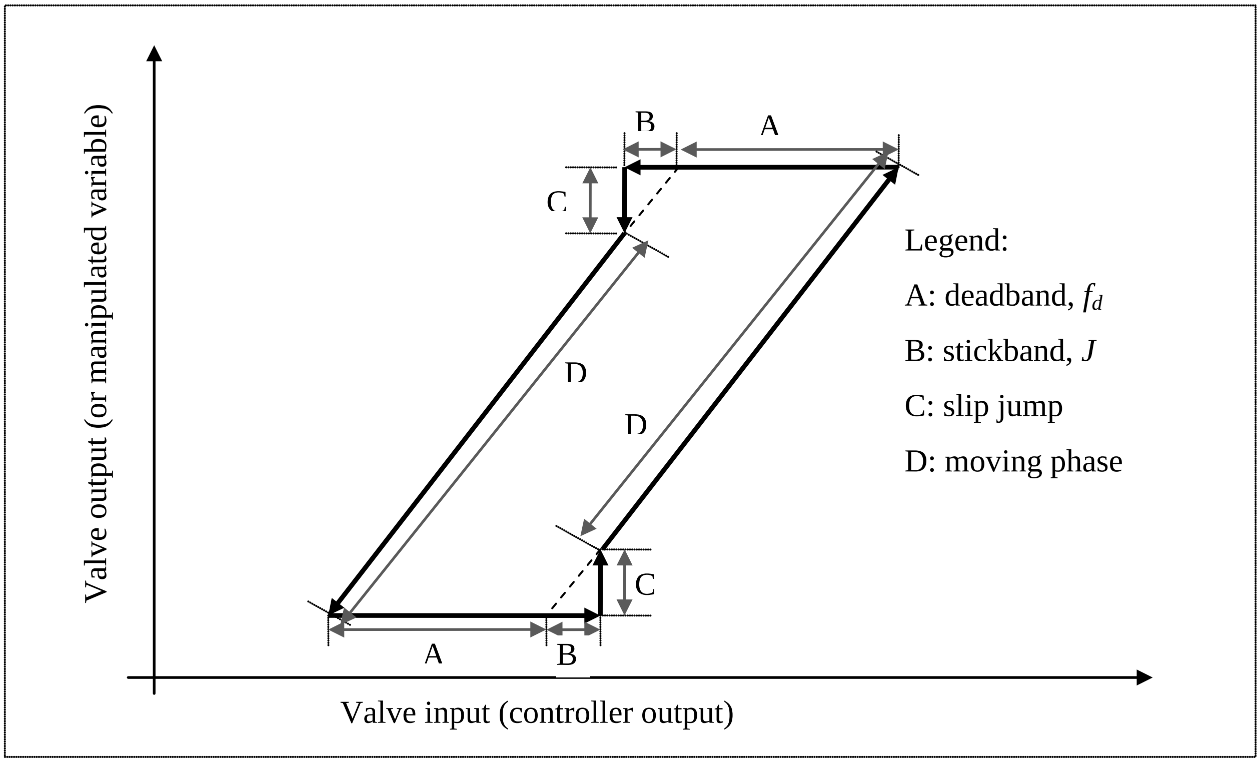


Figure 4: Phase plot for the typical behaviour of valve with stiction (after (Choudhury, Shah, Thornhill, & Shook, 2006))

Using the Tool

Installation

The toolbox can be installed by simply extracting the files to any desired location.

Starting the Tool

The tool is built based on MATLAB GUI, so MATLAB is required to run the GUI properly. In MATLAB main window, navigate to the directory that contains all the extracted files of this tool. At the command prompt, type in “valve_stiction” and press enter. The GUI shown in Figure 5 should appear.

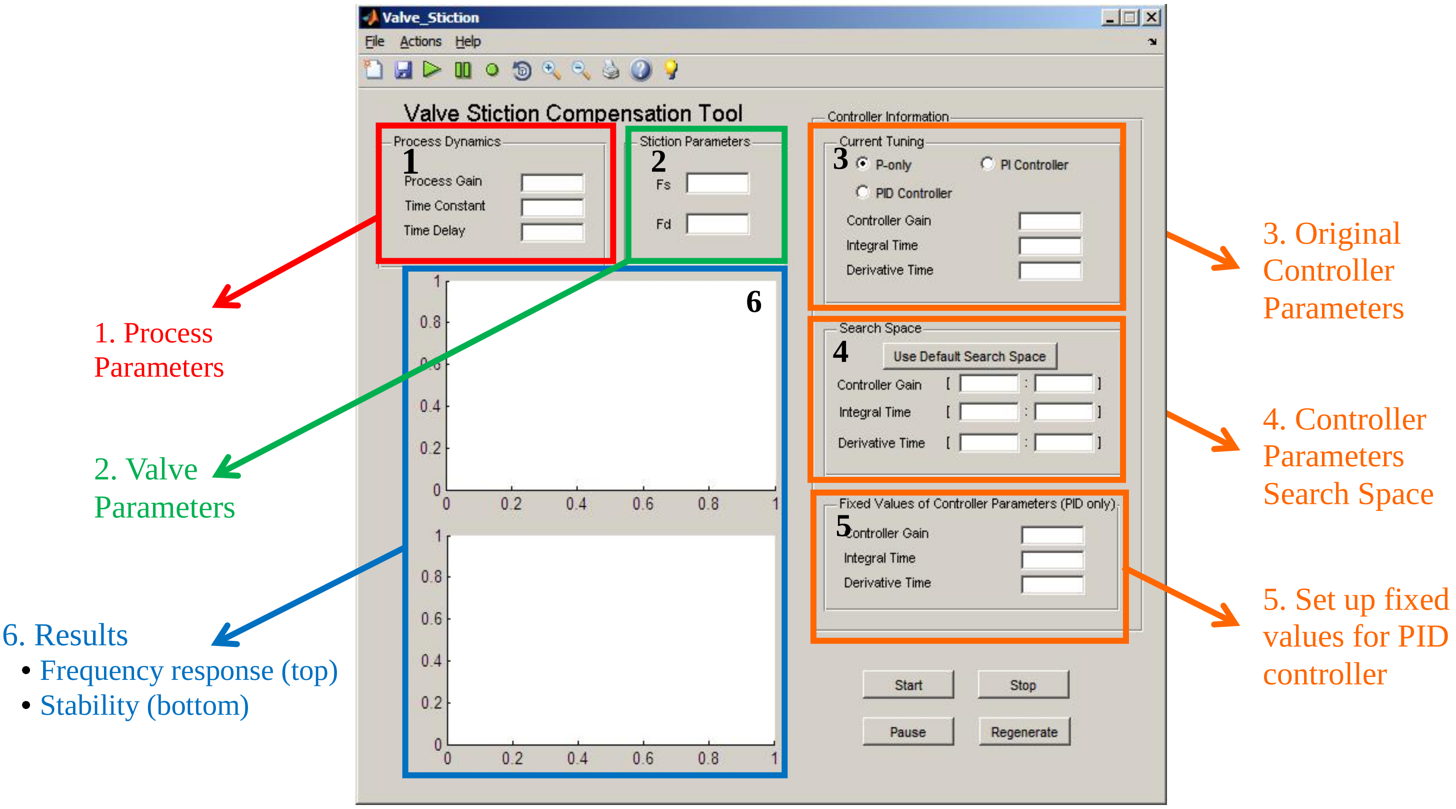


Figure 5: Valve stiction compensation user interface

In-depth Discussion of the GUI

Menu Section

The Menus on the top of the GUI contain three drop-down lists:

1) **File**:

   a. **New**: All values present in the input fields of the GUI and any selections that have already been made will be cleared.
   b. **Save All**[*]: This will save all the figures generated by the GUI.
   c. **Exit**: This will terminate the programme.

2) **Actions**:
   a. **Start**: This will start the calculation.
   b. **Stop**[*]: This will terminate all ongoing calculations. All the calculated data will be lost. Should anything unexpected happen in the programme or MATLAB, the programme can always be terminated by pressing Ctrl + C. In this case, the programme will not be usable and should be closed and restarted.
   c. **Pause/Continue**[*]: This will temporarily stop the calculation if one is in progress, in case the computer is needed for other use. If an ongoing process is paused, this button will also resume the calculation.
   d. **Regenerate Figures for PID Controller**: Since the results of this GUI are plotted against the controller parameters, for PID controller, there will be two four-dimensional plots which cannot easily be plotted in MATLAB. In this case, for PID controller, based on the selected values for controller gain, integral time, and derivative time are used as fixed values which must be within the defined search space. Thus, 3 three-dimensional plots for the frequency response and stability (six figures in total) will be generated.

3) **Help**:
   a. **Help**: This will display the help file.
   b. **About**: This will display the developers and copyright information.

[*] Not yet implemented due to the limitations of MATLAB

Process Dynamics Section

The values in this section are entered based on Equation (1). If an integrating process is present, the set $\tau = 0$.

Stiction Parameters Section

All information related to the valve stiction modeling is entered here. It can be noted that f_s should always be greater than f_d.

Controller Information Section

The original controller tuning, search space and fixed values for PID controllers are entered here.

1) **Current Tuning:** The current controller tuning parameters are entered here along with the controller type. The values are entered based on the controller tuning rule given as Equation (2).
2) **Search Spaces:** The search space of the new controller tuning can be enter here. Please note for each controller parameter, the value on the left should always be smaller than the one on the right. The default search space can also be used for a quick search. In this case, the default search space of the new controller tuning is defined as 10 times smaller and 3 times greater than the original controller parameters.
3) **Fixed Values of Controller Parameters (PID only)**: Selected values for controller gain, integral time and derivative time are used as fixed values which must be within the defined search space.

Buttons

1) **Start**: This will start the calculation.
2) **Stop**[*]: This will terminate all the ongoing calculation. All the calculated data will be lost.
3) **Pause/Continue**[*]: This will temporarily stop the calculation if one is already in progress, in case the computer is needed for other use. If the process is paused, this

[*] Not yet implemented due to the limitations of MATLAB

button will also resume the calculation. Should anything unexpected happen in the programme or MATLAB, the programme can always be terminated by pressing Ctrl + C. In this case, the programme will not be usable and should be closed and restarted.

4) **Regenerate**: Since the results of this GUI are plotted against the controller parameters, for PID controller, there will be two four-dimensional plots which would be confusing if plotted in MATLAB. In this case, for PID controller, a selected value for controller gain, integral time and derivative time are used as fixed values which must be within the defined search space.

Examples

The following example illustrates how Valve Stiction Compensation Tool can be used to remove the oscillations from a single loop process. The process schematic is shown in Figure 6.

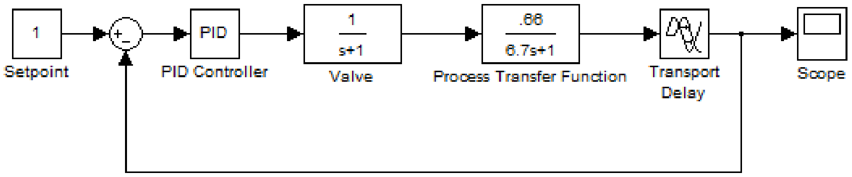


Figure 6: Diagram for Example 1

The original process and controller parameters are shown in Table 2 and Table 3.

Table 2: Process parameters for Example 1

	Process Gain, K	Time Constant, τ (unit time)	Time Delay, θ (unit time)
Parameters	0.66	6.7	2.6

Table 3: Original controller parameters for Example 1

	Controller Gain	Integral Time (unit time)
Parameters	0.1	1.11

Figure 7 shows that the system reaches steady state quickly without inducing any oscillations. Valve stiction (f_s = 1.2 and f_d = 1) is then added to this process for demonstrative purpose. The result is shown in Figure 8. Figure 8 shows that after adding valve stiction, the system response is oscillatory.

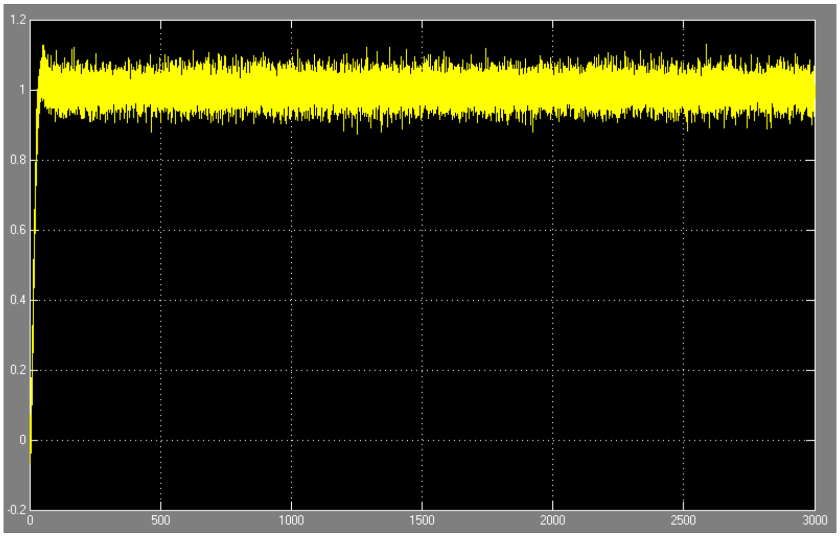


Figure 7: MATLAB process response simulation without valve stiction for Example 1

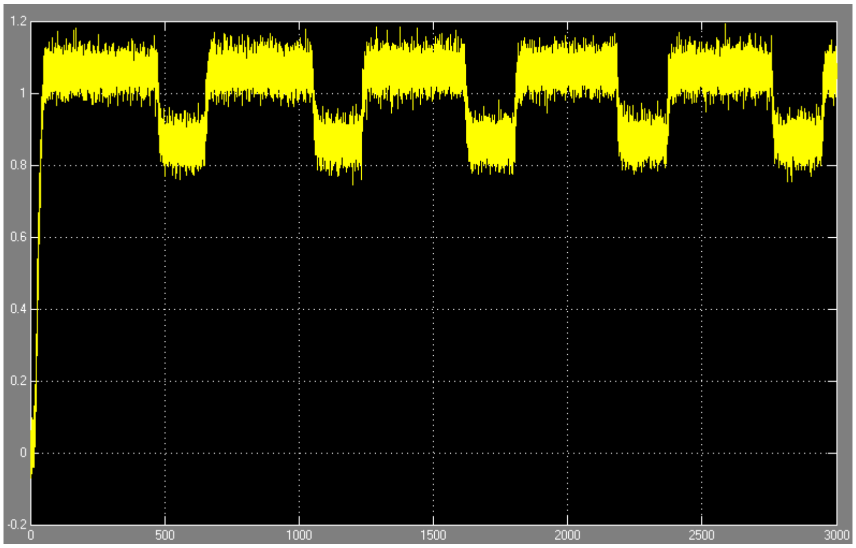
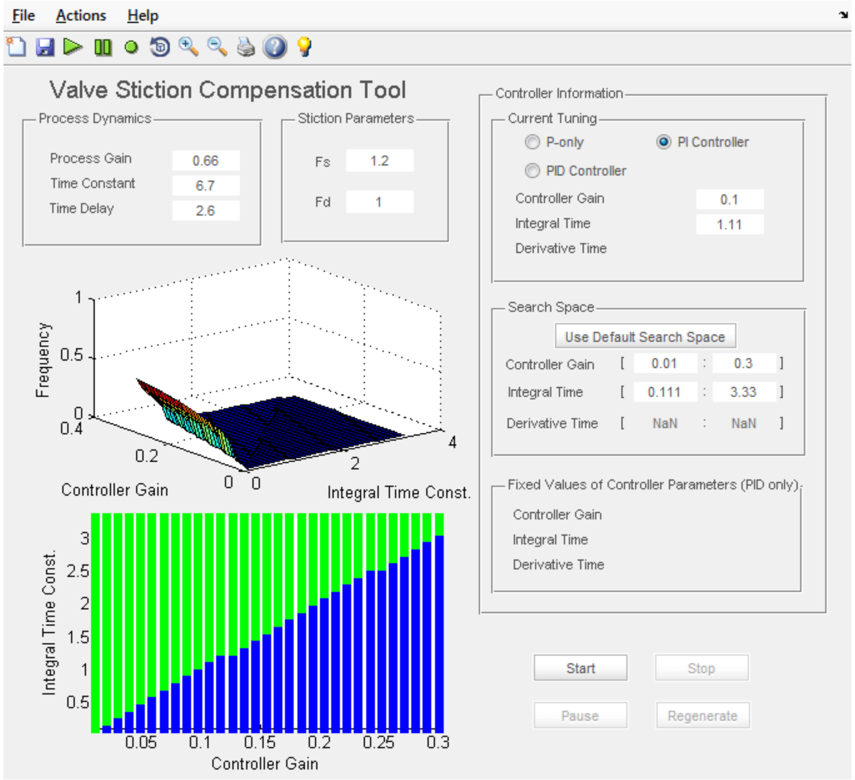


Figure 8: MATLAB process response simulation with valve stiction for Example 1

Using the Valve Stiction Compensation Tool, the oscillation can be removed by changing the controller tuning. After entering all the values into the GUI in the appropriate places, using the default search space, and running the calculation, the GUI window should appear as shown in Figure 9.



Valve Stiction Compensation Tool

Process Dynamics

Process Gain 0.66

Time Constant 6.7

Time Delay 2.6

Stiction Parameters

Fs 1.2

Fd 1

Controller Information

Current Tuning

○ P-only ● PI Controller

○ PID Controller

Controller Gain 0.1

Integral Time 1.11

Derivative Time

Search Space

Controller Gain [0.01 : 0.3]

Integral Time [0.111 : 3.33]

Derivative Time [NaN : NaN]

Fixed Values of Controller Parameters (PID only)

Controller Gain

Integral Time

Derivative Time

Figure 9: MATLAB main window for Example 1

The new controller tuning should be chosen from green area in bottom left figure of the main window. The frequency response is only given as an aid in understanding the behaviour of the system for different controller parameters. For the purposes of this example, the set of parameters shown in Table 4 will be used. The simulation results using this controller setting are shown in Figure 10. Figure 10 shows that valve stiction can be successfully compensated by changing the controller tuning to give a non-oscillatory and stable response.

Table 4: New controller tuning parameters for Example 1

	Controller Gain	Integral Time (unit time)
Parameters	0.1	2

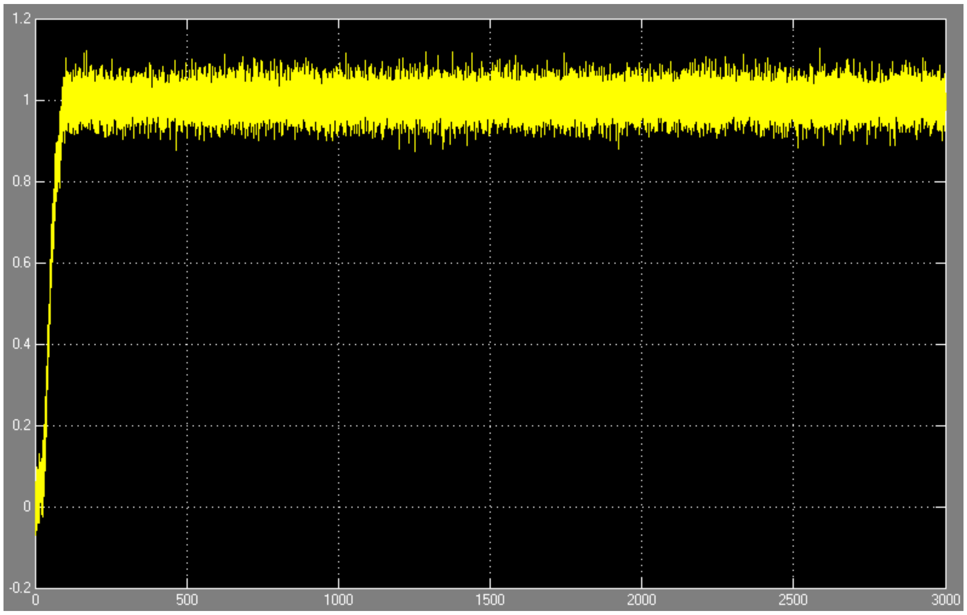


Figure 10: MATLAB process response simulation after valve stiction compensation for Example 1

Additional Information

Performance

Due to the limitation of MATLAB, Version 1.0 of this programme cannot use all the resources of multi-core processor or a multi-processor system. The results for the analysis of a P-only, PI- and PID-controller are expected to take up to 10 seconds, 1 minute, and 25 minutes respectively on a computer with Intel Core 2 Duo E6550 at 2.33 GHz and 4 GB of DDR2-SDRAM running Windows Server 2003 R2 and MATLAB 2010a (Version 7.10.0).

References

Choudhury, S., Shah, S. L., Thornhill, N. F., & Shook, D. S. (2006). Automatic detection and qunatification of stiction in control valves. *Control Engineering Practice , 14*, 1395-1412.

Lee, K. H., Ren, Z., & Huang, B. (2008). Novel Closed-loop Stiction Detection and Quantification Method via System Identification. *Advanced Control of Industrial Processes (ADCONIP).* Jasper, Alberta, Canada.

Mohammad, M. A. (2011). *Detection and Compensation For Stiction In Multi-loop Control System.* Edmonton.