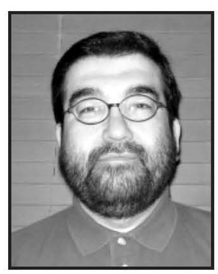
USING SIMULATION MODULE PCLAB
for Steady State Disturbance Sensitivity Analysis in Process Control

Emad Ali and Arimiyawo Idriss
King Saud University • Riyadh 11421, Saudi Arabia

Classical methods of teaching process control have been practiced in the classroom over the past several decades. These methods tend to focus on rigorous solution of differential and/or transfer function equations. The result is that students get caught up in mathematical algorithms rather than conceptualizing what happens in practice. It became evident to academics and practitioners that the way process control is taught to chemical engineers needs updating. It is believed that the strict classical teaching approach needs to be replaced by more practical and concrete approach.[1] To give students insights into the process control courses they take, laboratory courses and simulation tutorials were introduced in most chemical engineering curricula as supplements. This, it was believed, would give students insight and experience into the actual practice of chemical engineering. This issue was raised and discussed by many academic researchers and instructors and by practicing engineers.[1-8]

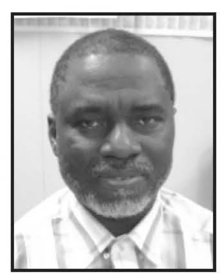
The introduction of simulation software has given students an outlet to follow their imagination. Simulation packages have caught up in all branches of engineering. This is because simulators have now acted as mergers between theory and practice that give students better understanding of processes before they venture into industry. The visualization of the process helps the student to form solid concepts on various aspects of chemical engineering.[9, 10]

Process simulation technology has evolved dramatically over the past 10 years. Many packages are available that allow intuitive visualization with a user-friendly graphical interface that allows rapid control design using click-and-drag operations. Rivera, *et al.*,[11] uses modules incorporated directly in process control computers. Young, et al.,[1] presented workshops based on real-time simulation of industrial processes. Henson and Zhang[2] have integrated simulation experiments based on HYSYS into the undergraduate process control courses. Cooper, *et al.*,[9,10] introduced a training simulator called Control Station. Other software packages for control education include PICLES,[12] and ACS[13]; however, these packages do not, in general, adequately handle large practical-scale problems.[14] Doyle, *et al.*,[15] have developed a process control module (PCM) simulator based on a MATLAB/SIMULINK environment that contains case studies illustrating various process control concepts. Despite the benefits of simulation-based experiments, one main criticism remains the lack of physical process that can be felt by students. It is argued, however, that training simulators can provide students with a broad range of experiences at low cost and in a safe environment. Moreover, students can achieve these experiences conveniently at their own desks.[9]



***Emad Ali** is a faculty member of the Chemical Engineering Department at King Saud University. He received his Ph.D. from the University of Maryland, College Park, in 1996. For the past 12 years he has been teaching process control to undergraduate and graduate students. His main research interest is in the area of process modeling, identification, and control.*

***Arimiyawo Ibn Idriss** is a research scientist with the Department of Chemical Engineering in King Saud University. Having graduated from King Fahd University of Petroleum and Minerals with a Masters in chemical engineering in 1989, Mr. Idriss has taught the Process Control Laboratory course for the past 10 years. His current research interest is in plant energy analysis and catalysis.*



Process Control Laboratory (PCLAB) was developed in the chemical engineering department at King Saud University as an educational tool. General introduction of the tool is given elsewhere.[16] The primary objective of this work is to unveil one of PCLAB's specific features—the steady state disturbance sensitivity analysis (SSDSA).[17] We focus on SSDSA because it is a distinguishing contribution of PCLAB and because its approach is found to be effective in designing the control structure of multi-loops control problems.[18, 19] Specifically, SSDSA will be carried out on a forced-circulation evaporator process, which is one of PCLAB's case studies. The purpose is to explore the disturbance effects on the output and controlled parameters and to conclude from the analysis which manipulated variable could be used to mitigate the effect of the disturbances.

PCLAB

PCLAB is interactive simulation software for process control analysis and training. It was developed using MATLAB tools and functions including SIMULINK (a graphical simulation toolbox). MATLAB was chosen as the programming platform because it became a standard among academic and industrial users alike for use both in research and education.[14] Moreover, one can easily customize or add to existing modules of MATLAB. The flexibility of this platform allows for migration to many PC and workstation hardware platforms.[14] The PCLAB software is designed in a user-friendly, menu-driven framework such that the process engineer can easily navigate through the various parts of the program, carry out simulation experiments, visualize the results, and draw conclusions on the effects of different parameters and control configurations. This is achieved by using the main-menu shown in Figure 1 which provides a Graphical User Interface (developed in the MATLAB graphics language). The software will run on any platform supported by MATLAB (WIN 95, WIN NT, UNIX). The software consists of several modules that comprise different case studies based on fundamental process models of industrial unit operations. The case studies of the current version of PCLAB shown in Figure 1 include process models adopted from the literature such as Forced Circulation Evaporator,[20] Fluid catalytic Cracking Unit,[21] Double Effect Evaporator,[22] and Two CSTRs in Series.[23] The case studies also include process models that are developed and validated by our research group such Polyethylene Reactor,[18] Ethylene Dimerization Reactor,[19] and Multistage Flash Desalination Plant.[24] The selected modules in addition to the convenient visualization feature of the software provide the student with real-world hands-on experience. PCLAB is available for public use. Interested readers can download the program from the following Web site: <http://faculty.ksu.edu.sa/Emad.Ali/Pages/PCLab.aspx.>

The main menu of the program, as shown in Figure 1, allows the user to choose from different case studies. When a case study is chosen from the main menu, the software will trigger a submenu that contains the available exercises that can be carried out on the chosen case study. The submenu, shown in Figure 2, allows the user to select a specific tutorial from a variety of exercises, such as steady state analysis, process dynamic analysis, process identification, control structure selection and controller tuning for SISO systems, and multiple SISO loop tuning.

In this paper, we will discuss the SSDSA exercise applied on the forced-circulation evaporator case study to explore the versatility of the current version of PCLAB software. Application procedure of SSDSA on other case studies is similar.

FORCED-CIRCULATION EVAPORATOR

The forced-circulation evaporator is a common processing unit in sugar mills, alumina production, and paper manufacture. This process is used to concentrate dilute liquor by evaporating its solvent (usually water), as shown in Figure 3.[20] A feed stream with solute of concentration C_1 (mass percentage) is mixed with high volumetric recycle flow rate and fed to a vertical evaporator (heat exchanger). The solution will pass through the tube. A saturated steam is used to heat the mixture by condensing on the outer surface of the tubes. The liquor, which passes up inside the tube, boils and then passes to a separator vessel. In the separator, the liquid and vapor are separated at constant temperature and pressure. The liquid is recycled with some being drawn off as product with solute concentration of C_2. The vapor is usually condensed with water and used as the coolant.

A description of the process parameters and their values are given elsewhere.[21] For this process we deal with three inputs: the coolant flow rate, F200; the steam pressure, P100; and the steam flow rate, F100. Four disturbances are considered: the feed flow, F1; feed temperature, T1; feed concentration, C1; and the coolant temperature, T200. The process has four outputs: liquid level, L2; output concentration, C2; Column pressure, P2; and outlet flow rate, F2.

PROCESS ANALYSIS

In this section, we discuss how steady state disturbance analysis can be implemented on PCLAB. This procedure is very useful for designing the appropriate control structure.[18] When controlling a plant or a process with many inputs and outputs, it is usually difficult to optimally pair variables into a multi single loops structure. SSDSA is a tool that can help in this regard, although it cannot be implemented on real plant. Instead, simulation of the process can be used to perform the task. Figure 2 illustrates that one can simply click on the steady state disturbance analysis. As a result a new window will pop up. The new window is an SSDSA interface module for the evaporator case study as shown in Figure 4.

Figure 4 shows that the process has three inputs and four possible disturbances, as discussed earlier. The procedure will focus on investigating the static effect of any disturbance or any combination of disturbances on the process outputs in open loop mode. This means that the inputs will remain fixed during the test. This is known as the open loop test. It reveals which process output is affected the most and which one is affected in nonlinear fashion. The test can also be run in closed loop mode. In this case, an output should be selected as the controlled variable and a corresponding input should be selected to be the manipulated variable. The test will then examine the effectiveness of the chosen input to maintain the controlled variable at its nominal value in steady state when the process is under the influence of a range of disturbance values.

To start the procedure, simply click on the green button or the start button on Figure 4. By doing so, the SSDSA menu

Figure 1. *Main menu showing the main case studies.*

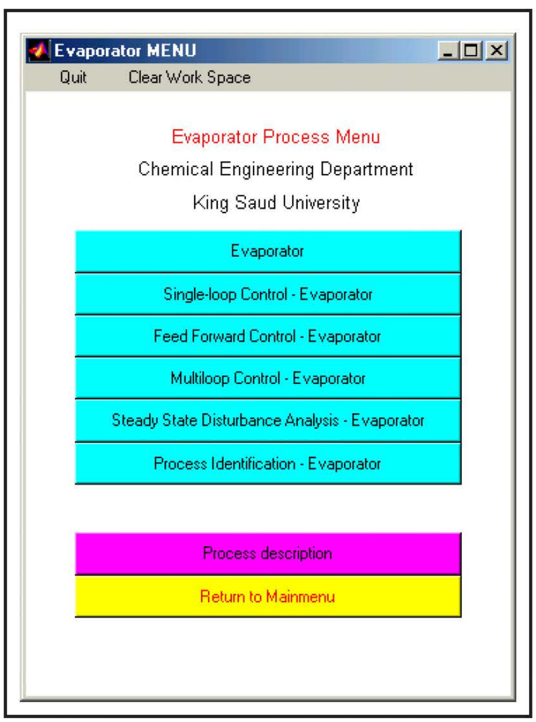


Figure 2. *Sub Menu for Evaporator Case Study showing available control exercises.*

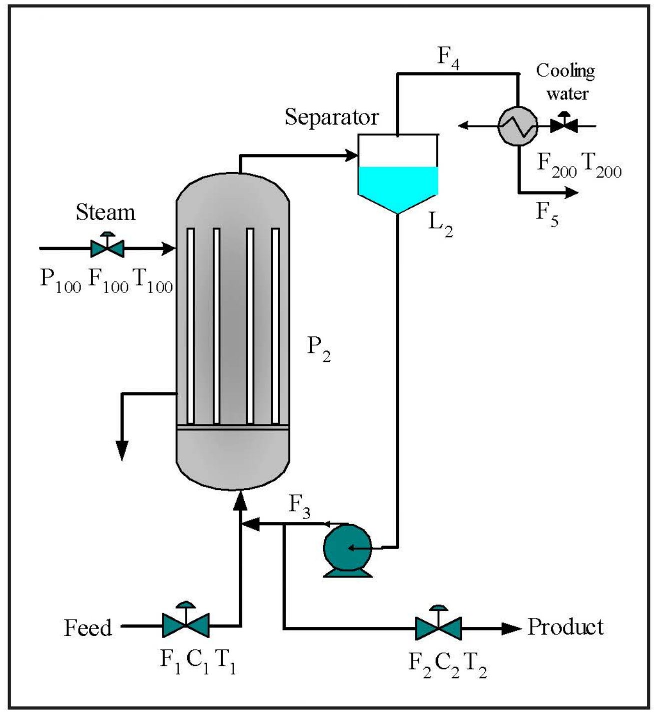


Figure 3. *Flow sheet of Forced circulation Evaporator Process.*

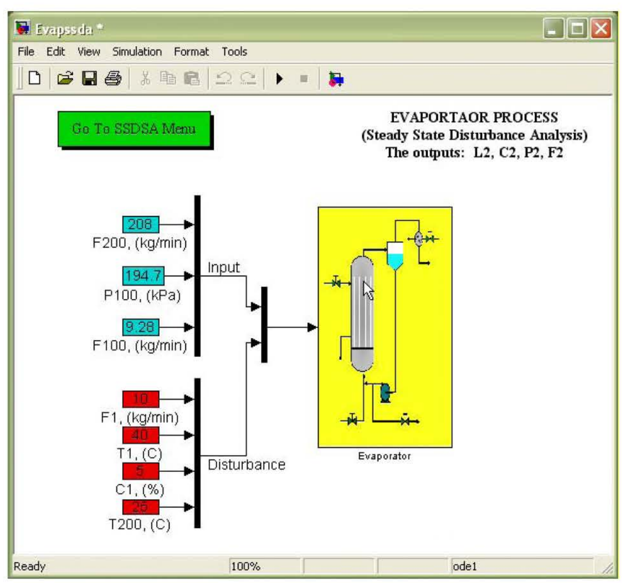


Figure 4. *Steady State Disturbance Module for Evaporator Process.*

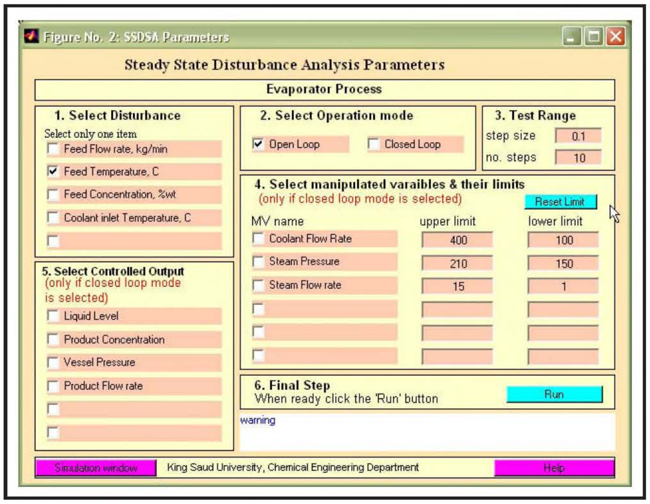


Figure 5. *SSDSA menu for the evaporator process.*

illustrated in Figure 5 pops up:

The menu in Figure 5 allows the user to enter the key parameters that controls the SSDSA analysis. Basically six steps are required to carry out the SSDSA analysis procedure, as discussed in the following two sections.

Open loop mode

First the user should select one of the four possible disturbances by marking the appropriate checkbox shown on Figure 5, for instance, the feed temperature. (Note that the other case studies of PCLAB will have a different list of disturbance variables according to the relevant process.) Next, mark the open loop checkbox. The third step controls the test range. For example, if the nominal value for the feed temperature is $T_0 = 40$ °C and using step size of 0.1 and number of steps of 10, then the disturbance value will have the following range during the test:

$$T \in \left[T_0 + (10/2)*0.1T_0, T_0 - (10/2)*0.1T_0\right] \quad \text{Eq. (1)}$$

Increasing the number of steps at the same step size will increase the temperature range to be covered. The above values for the step size and number of steps cover a ±50 % range, which is good enough from practice point of view. Decreasing the step size will help in producing smoother response curves but it will decrease the overall range. Therefore, if one decreases the step size for better resolution, one should also increase the number of steps to maintain the same operating range. It should be noted, however, that smaller step size requires a higher computational load.

In the open loop mode, steps 4 and 5 are bypassed. (If by mistake the user marks one of the boxes in step 4 or 5, an error message will be displayed in the warning box.) Next, press the run button and look at the results, shown in Figure 6.

Figure 6 illustrates how the four main process outputs respond at steady state to changes in the feed temperature from 20 to 60 °C. The graphs shows the response of the liquid level at the top followed by concentration response, then pressure response and finally the flow response to changes in feed temperature. It is obvious that the liquid level in the separator unit is not affected by this type of disturbance. Thus in the open loop mode, the user can gain information about the directional, magnitude, and nonlinearity effect of a disturbance. For example, as a directional effect, both C_2 and P_2 will increase when the feed temperature increases, while the outlet flow rate F_2 will decrease. One can also observe that the solute concentration (C_2) received the highest (magnitude) impact. Moreover, all outputs are altered linearly with the temperature variation. Thus, the user can learn how the process operation and product quality may be significantly influenced when the feed temperature is changing freely.

Closed loop mode

In this test mode, the student should unmark the open loop checkbox and mark the closed-loop checkbox instead. Furthermore, there is a need to specify a controlled variable. Let us choose, for example, the output concentration, C_2. In addition, the user should select one of the manipulated variables listed in Step-Box 4 as the candidate. Let the candidate-manipulated variable be the coolant flow rate. See Figure 7 for example. The user then has the choice to either change the upper and lower permissible values for the candidate-manipulated variable or leave them at their default values. Note that the default values for the upper and lower limit can be restored at any time by simply clicking the "reset limit" button. It should be noted that in this simulation mode no typical control system is involved. The control objective, *i.e.*, the output deviation from its set point, is formulated as an algebraic constraint. These constraints along with the model algebraic equations are solved at steady state using the manipulated variables as the design parameters.

After the user finishes marking the required checkboxes in the SSDSA menu, he can simply click the run button in Step-Box 6. The result for the above specification is shown in Figure 8.

By inspecting the output response in Figure 8a, one observes that the controlled variable C_2 is well maintained at the nominal value. Not visible here, but on the computer screen, a red color is used for the controlled variable to distinguish it from the other uncontrolled outputs. Because the evaporator pressure is not controlled, it increases as expected with disturbance but this time with a larger magnitude. On the other hand, the feedback control caused the output flow rate to change slightly with disturbance.

More important is the response of the manipulated variable. In the right-hand plot, Figure 8b, red lines would be seen, showing the upper and lower limits for the coolant flow rate, which is set at 400 and 100, respectively. A white line represents the response of the coolant flow rate to disturbances,

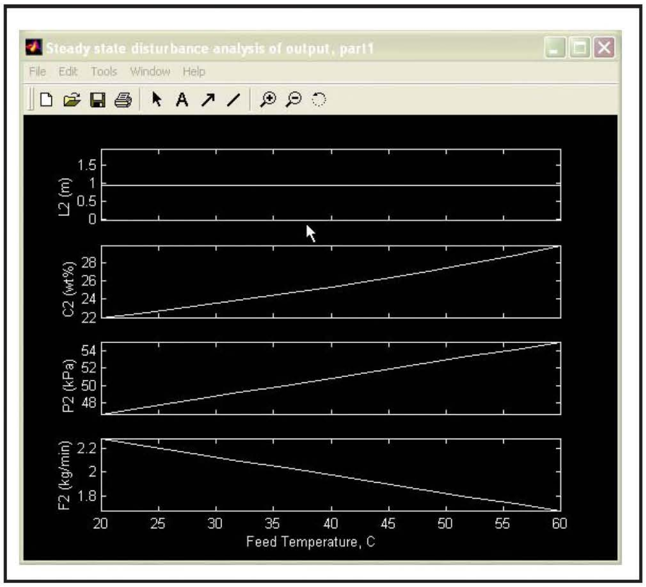


Figure 6. *SSDSA results for open loop test showing the effect of feed temperature.*

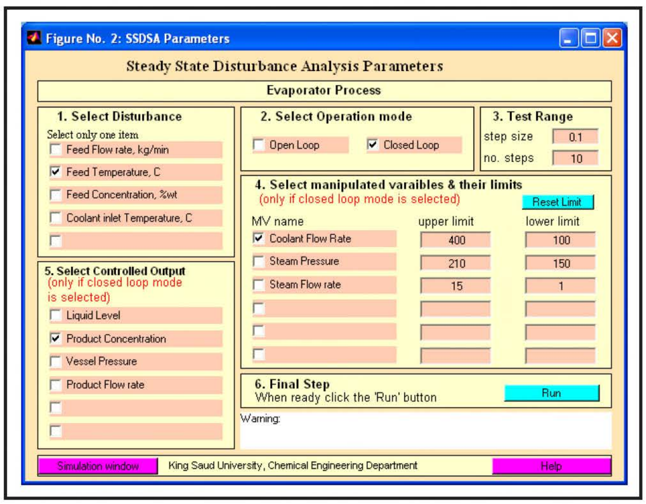
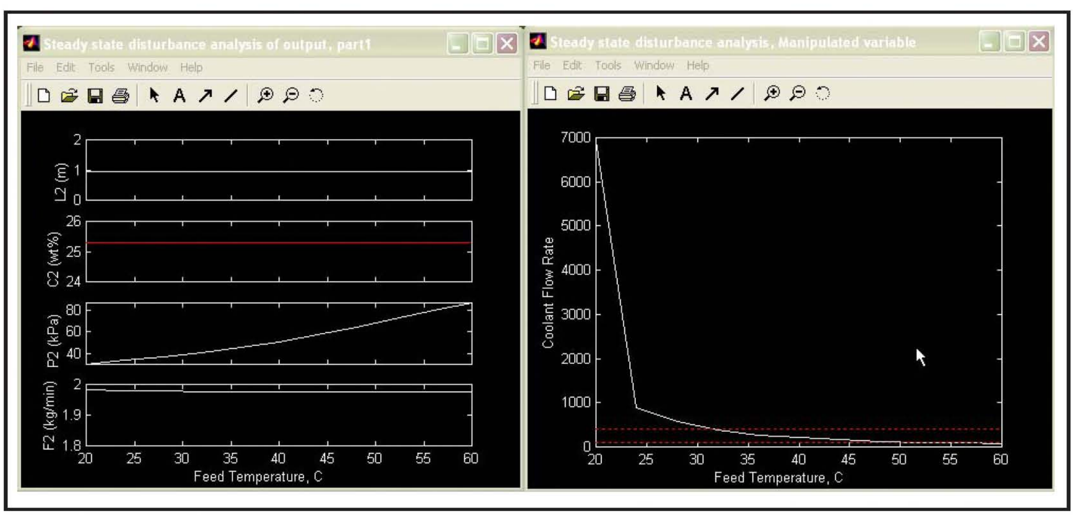


Figure 7. *SSDA menu for closed loop case option.*

allowing the user to maintain C_2 at nominal value. For large disturbances, *i.e.*, when the feed temperature exceeds 55 °C, the coolant flow should be reduced slightly below the lower limit in order to reject the effect of the disturbance. At feed temperature below 30 °C, however, the coolant flow rate must be increased multifold, especially below 25 °C, to maintain the required operation. Low feed temperature requires higher steam pressure to provide enough heat of vaporization, which in turn increases the process temperature. As a result a large amount of coolant flow is needed to absorb the extra heat and to cool down the vapor. The high demand on coolant flow may not be physically possible, however. Therefore, one can conclude that the coolant flow rate is not a good manipulated variable for negative disturbance in the feed temperature.

The user can also carry out a multivariable SSDSA. For example, one can add another controlled variable to the control structure, such as the column pressure. For this case, the user should consider a suitable manipulated variable, such as the steam flow rate. Rerunning the SSDSA as before, we obtain the results shown in Figures 9 and 10 (next page). Figure 9 shows clearly that the controlled variable and consequently the remaining outputs are well maintained at their targets. (Note that, on-screen, the controlled variables are distinguished by the red color.) On the other hand, Figure 10 shows how the selected manipulated variables change to counteract the effect of the disturbance and ultimately regulate the output at their set points. Notably, the first manipulated variable (*e.g.*, coolant flow rate) has not changed, while the second manipulated variable (*e.g.*, steam flow) decreased slightly as the feed temperature increased. One can accept this control structure because for a range of ±50% changes in the disturbance, the product concentration and column pressure are well regulated with minimum change in the manipulated variable. Moreover, this is achieved without violating the physical bound of the manipulated variables.

Although Yi and Luyben[17] have used SSDSA for determining basic control structure, its outcome can help in building



◄ ***Figure 8.*** *Output responses to disturbance in feed temperature when the coolant flow rate is used as manipulated variable.*

an appropriate input-output pairing. In fact, the user can test various scenarios by examining other candidate-manipulated variable and by repeating the procedure for the other controlled variables. At the end, the user can build up a satisfactory control structure for the process, *i.e.*, can select the appropriate input-output pairing configuration.

Similar studies can be carried out on the remaining disturbances and find their effects on the controlled parameter. Furthermore the case study that was used here can be replaced by another case study and similar SSDSA analysis can be carried out.

CONCLUSION

In this paper, the process control laboratory PCLAB is introduced. PCLAB is based on MATLAB platform and designed in a user-friendly environment by using a convenient graphical interface. The current version of PCLAB thus far includes seven modules (case studies) that reproduce basic chemical processes. PCLAB has a number of control design problems (exercises) that can be applied to each of the seven case studies. One of the PCLAB exercises is the steady state disturbance analysis. The exercise is illustrated through a tutorial using the evaporator case study. The examples illustrate how the SSDSA can be carried out in either open loop or closed loop mode without tedious programming. The analysis helps in illustrating which disturbance has a detrimental impact on the process. Furthermore, different control structure configurations can be screened off line to determine the most appropriate input-output pairing. In addition to this exercise, there are other control projects that can be studied to enhance student learning. These exercises can help students to practice and visualize the theoretical concepts taught in the classrooms.

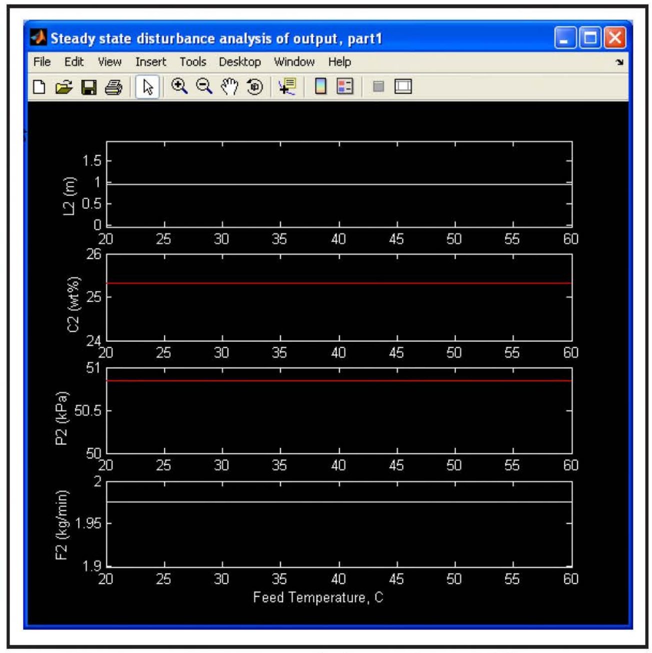


***Figure 9.** SSDSA result for multivariable case, output response.*

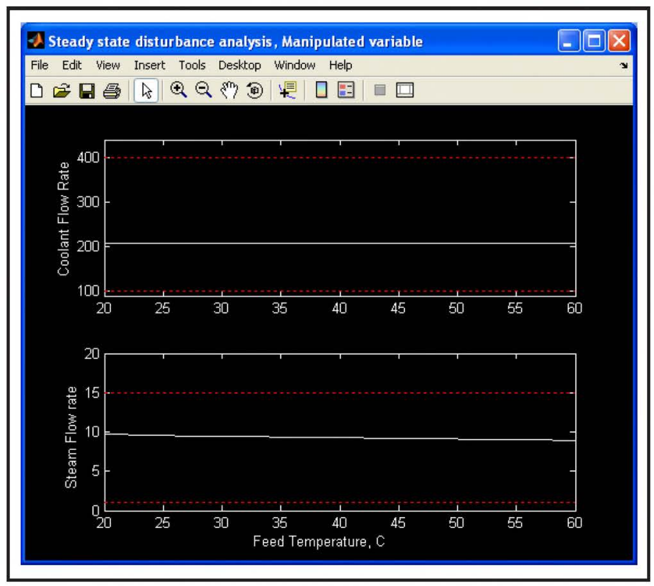


***Figure 10.** SSDSA result for multivariable case, the input response.*

REFERENCES

1. Young, B., D. Mahoney, and W. Svrcek, "Real-Time Computer Simulation Workshops for the Process Control Education of Undergraduate Chemical Engineering," *Comput. Appl. Eng. Ed.*, **9**(1), 57 (2001)
2. Henson, A.H., and Y. Zhang, "Integration of Commercial Dynamic Simulators into the Undergraduate Process Control Curriculum," presented at AIChE annual meeting (2000)
3. Bequette, B.W., K.D. Schott, V. Prasad, V. Natarajan, and R.R. Rao "Case Study Projects in an Undergraduate Process Control Course," *Chem. Eng. Educ.*, **32**(3), 214 (1998)
4. Bequette, B.W., J.H. Chow, C.J. Li, E. Maby, J. Newell and G. Buckbee "An Interdisciplinary Control Education Studio," in *Proceedings of the Conference on Decision and Control*, Phoenix, 370-374 (1999)
5. Brisk, M., and R.B. Newell, "Current Issues and Future Directions in Process Control," *Chemical Engineering in Australia*, **14**(3), 8 (1989)
6. Edgar, T.F., "Process Control Education in the Year 2000," *Chem. Eng. Ed.*, **24**(2) 7 (1990)
7. Ramaker, B.L., H.K. Lau, and E. Hernandez, "Control Technology Challenges for the Future," in *Proceedings of the Chemical Process Control V Conference*, Tahoe City, CA, J (1996); Kantor, J.C., C.E. Garcia, and B. Carnahan (Eds.), published by CACHE, AIChE, New York, AIChE Symposium Series No. 316, **93**, 1-7 (1997)
8. Downs, J.J. ,and J.E. Doss, "Present Status & Future needs—A View From North American Industry," *Proceedings of Chemical Process Control IV*, Arkun, Y., and W.H. Ray, (Eds.), AIChE., New York (1991)
9. Cooper, D.J., and D. Dougherty, "Enhancing Process Control Education with the Control Station Training Simulator," *Computer Applications in Engineering Education*, **7**, 203 -212 (1999)
10. Cooper, D.J., D. Dougherty, and R. Rice, "Building Multivariable Process Control Intuition Using Control Station," *Chem. Eng. Ed.*, **37**,100 (2003)
11. Rivera, D., K. Jun, V. Sater, and M. Shetty, "Teaching Process Dynamics and Control Using an Industrial-Scale Real-Time Computing Environment," *Comput. Appl. Eng. Ed.*, **4**(3), 191 (1996)

12. Cooper, D.J., "Picles: A simulator for teaching the real-world of process control," *Chem. Eng. Educ.*, **27**, 176 (1993)
13. Koppel, L.B., and Sullivan, G.R., "Use of IBM's Advanced Control System in Undergraduate Process Control Education," *Chem. Eng. Ed.*, **20**, 70 (1986)
14. Doyle III, F.J., and F. Kayihan, "Experiences Using MATLAB/Simulink for Dynamic 'Real-time' Process Simulation in an Undergraduate Process Control Course," American Society for Engineering Education Conference, Session 3613 (1998)
15. Doyle III, F.J., E. Gatzke, and R. Parker., *Process Control Modules, A Software Laboratory for Control Design*, Prentice-Hall, NJ (2000)
16. Ali, E., and A. Idriss, "An Overview of Simulation Module, PCLAB, For Undergraduate Chemical Engineers in Process Control," accepted by *Computer Applications in Eng. Ed* (2008)
17. Yi, C., and W. Luyben, "Evaluation of Plant-Wide Control Structures by Steady- State Disturbance Sensitivity Analysis," *Ind. Eng. Chem. Res.*, **34**, 2393 (1995)
18. Ali, E., K. Al-Humaizi, and A. Ajbar, "Multivariable Control of a Simulated Industrial Gas-Phase Polyethylene Reactor," *Ind. Eng. Chem. Res.*, **42**, 2349 (2003)
19. Ali, E., and K. Alhumaizi, "Temperature Control of Ethylene to Bueten-1 Dimerization Reactor," *Ind. Eng. Chem. Res.*, **39**, 1320 (2000)
20. Newell, R.B., and P.L. Lee, *Applied Process Control —A Case Study*, Prentice- Hall, Sydney (1989)
21. McFarlane, R.C., R.C. Reineman, J. Bartee, and C. Georgakis, "Dynamic Simulator for a Model IV Fluid Catalytic Cracking Unit," *Comp Chem. Eng.*, **17**, 275 (1993)
22. Daoutidis, P., and A. Kumar, "Structural Analysis and Output Feedback Control of Nonlinear Multivariable Processes," *AIChE*, **40**, 647 (1994)
23. Cao, Y., and D. Biss, "An Extension of Singular Value Analysis for Assessing Manipulated Variable Constraints," *J. Process Control*, **6**, 34 (1996)
24. Ali, E., A. Ajbar, and K. Al-humaizi, "Robust Control of Industrial Multi Stage Flash Desalination Processes," *Desalination*, **114**, 289 (1997) ❒