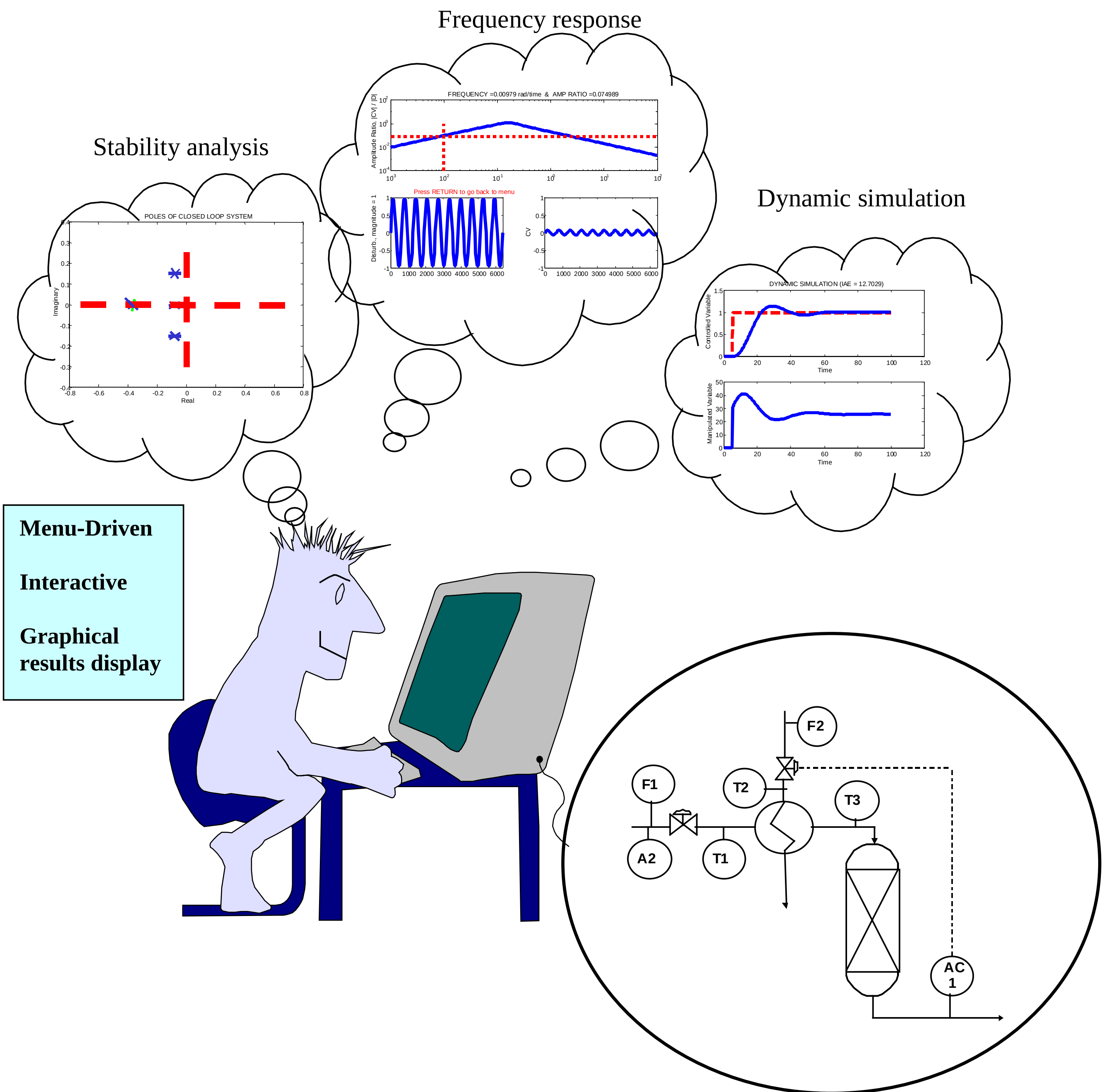
THE SOFTWARE LABORATORY
S_LOOP:
Single-loop Feedback Control System Analysis



Version 3.0 for MATLAB 6.x

S_LOOP: SINGLE-LOOP FEEDBACK CONTROL SYSTEM ANALYSIS

by Michelle Gretzinger, Daniel Zyngier and Thomas Marlin

INTRODUCTION

One of the challenges to the engineer learning process control is relating theoretical concepts to the time-domain behavior of dynamic systems. This **menu-driven, interactive** software package provides learning experiences for single-variable feedback systems, including stability analysis, frequency response, and time-domain performance. Many of the solved examples in the textbook can be repeated and extended using this software.

The basic design of the software enables the user to use simple menus to enter parameters, select specific calculations and display results. **No programming is required**. To provide a simple yet flexible system, the process dynamics are represented by a general transfer function model, fourth order with lead and dead time, and the controller is limited to a proportional-integral-derivative (PID) feedback algorithm. The user can tailor this system by setting selected parameters to zero.

SYSTEM STRUCTURE

The overall structure of the system is given in Figure 1, and the main Main Menu is shown in Figure 2. The Main Menu appears when the program is first executed and remains the principle interface throughout the session. In a typical session, the user first defines the process and controller through Submenus 1-3; then, he/she selects calculations through the remaining submenus. An important feature is the use of the ***same models*** for all calculations, so that the user can clearly relate the frequency response analysis to time-domain behavior.

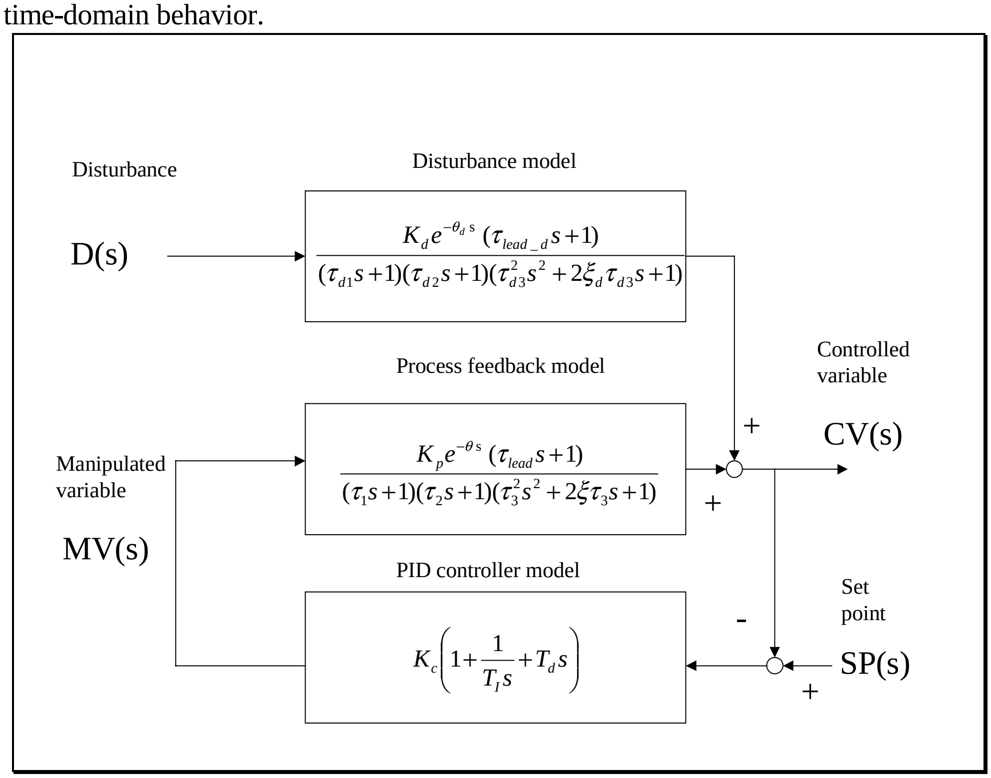


Figure 1. Block diagram of the dynamic system in S_LOOP.

```
************************************************************
*        S_LOOP:   SINGLE LOOP CONTROL SYSTEM ANALYSIS      *
*                                                           *
*          McMASTER UNIVERSITY CHEMICAL ENGINEERING         *
*          ========================================         *
*                         Version 3.0                       *
*                                                           *
*                         MAIN MENU                         *
************************************************************
SELECT THE APPROPRIATE MENU ITEM
1) Modify Feedback Process Model Parameters
2) Modify Disturbance Process Model Parameters
3) Modify PID Controller Tuning Constants
4) Plot Poles of Closed Loop System
5) Bode Plot for GOL for Stability Analysis
6) Bode Plot for Process, Gp
7) Bode Plot for Closed Loop Responses
8) Dynamic Simulation of System
0) Quit to Matlab
Please enter the desired selection:
```

Figure 2. Main Menu for the S_LOOP software.

The user enters the process and controller parameters in the first three Submenus.

PROCESS AND CONTROLLER DATA

Submenu

1) **Process Model -** The process model, $G_p(s)$, parameters are entered using Submenu 1. Recall that the process model defines the dynamics between the manipulated variable and the controlled variable. Any parameter can be set to zero without causing numerical errors.

2) **Disturbance Model -** The disturbance model, $G_d(s)$, parameters are entered through Submenu 2. Recall that the disturbance model defines the dynamics between the disturbance and the controlled variable. Any parameter can be set to zero without causing numerical errors.

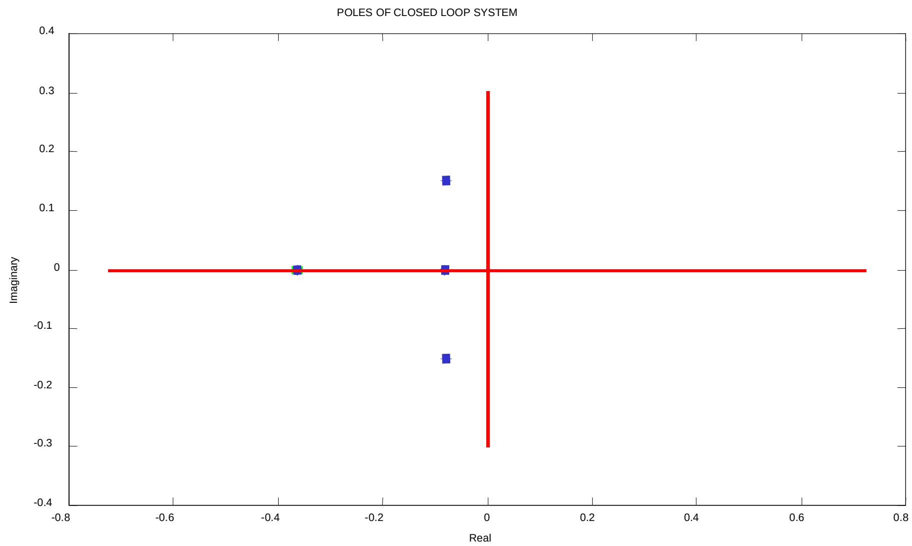
3) **Submenu 3: PID Controller -** The control algorithm is ***Old Faithful***, the proportional-integral-derivative or PID controller. This is by far the most widely used controller in industry. It has three adjustable tuning parameters, which are defined in Submenu 3: the gain (K_c), the integral time (T_I), and the derivative time (T_d). The user can turn off the controller by setting $K_c = 0$. If the user sets the integral time to zero, the integral mode is removed from the algorithm (a divide by zero does not occur).

ANALYSIS AND SIMULATION

The remaining Submenus enable the user to select from the following analysis calculations for the process and control system defined through Submenus 1-3.

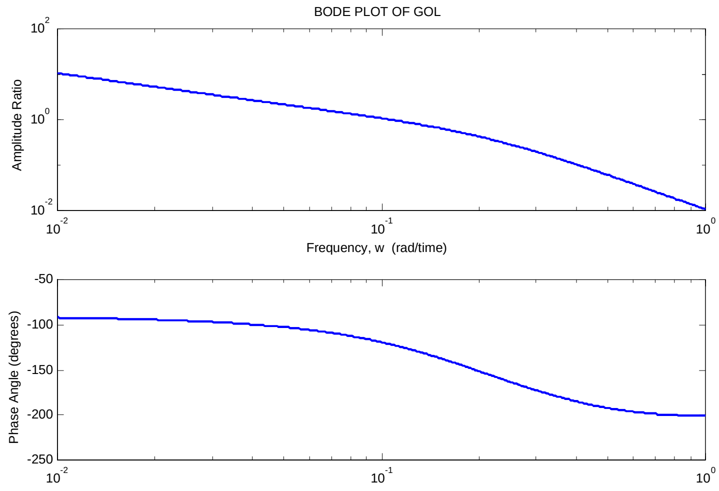
Submenu

4) **Poles -** Solve for and plot the roots of the characteristic equation (i.e., the poles) of the closed-loop system to evaluate stability. This can only be done if the characteristic equation is a polynomial in "s" and does not contain a dead time. If a dead time occurs in the feedback process model, a message is provided to the user explaining that the calculations cannot be performed.



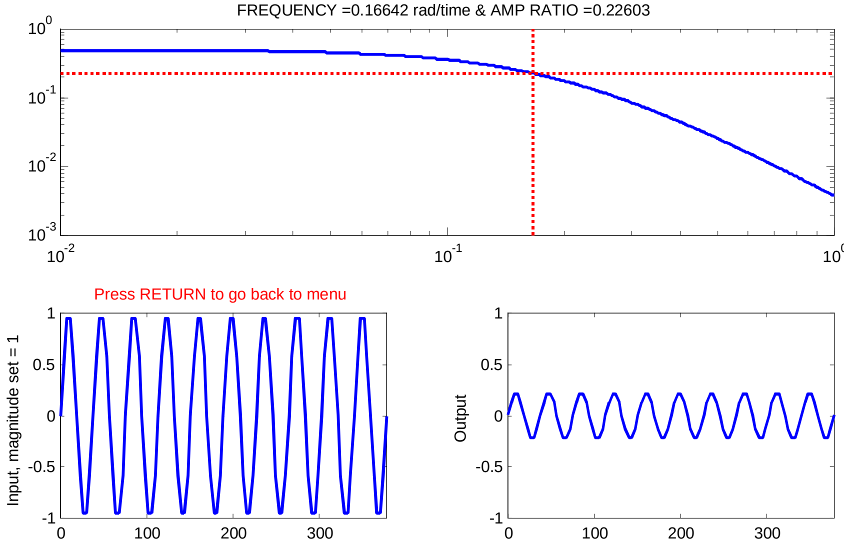
The poles are:
-0.3626
-0.0778 + 0.1504i
-0.0778 - 0.1504i
-0.0818

5) **Bode Stability** - Calculate the frequency response displayed as a Bode plot of $G_{OL}(j\omega)=G_p(\omega j)G_v(\omega j)G_c(\omega j)G_s(\omega j)$. This can be used to determine the stability and apply the Zeigler-Nichols tuning.

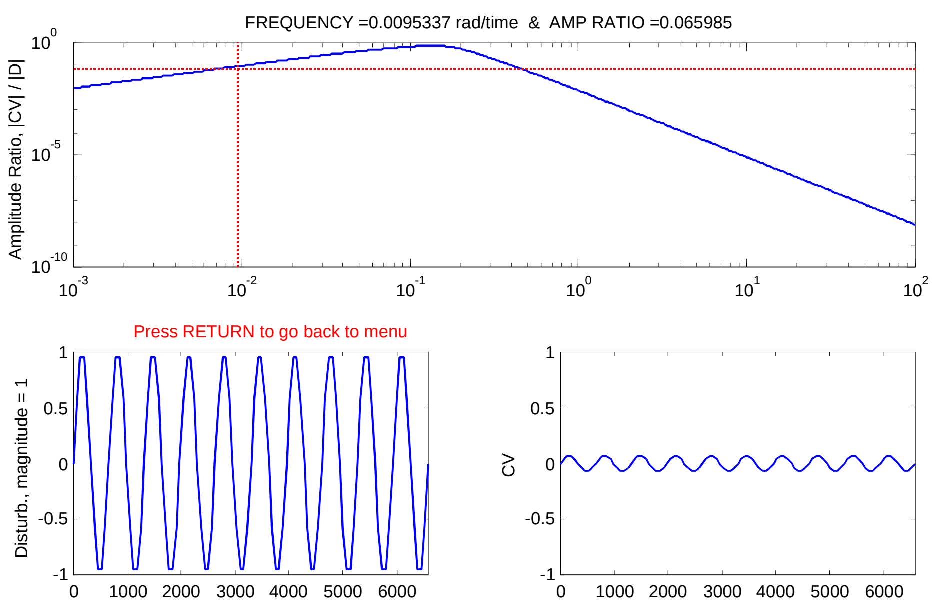



```
**************************************************
     Critical frequency and amplitude ratio
             from Bode plot of GOL
**************************************************

Caution:  1) cross check with plot because of possible MATLAB error
             in calculating the phase angle
          2) the program finds the first crossing of -180

The critical frequency is between          0.353
                                       and 0.35381

The amplitude ratio at the critical frequency is 0.1398
```

6) Process Frequency Response - Calculate the frequency response of the process model alone, i.e., $G_p(j\omega)$, displayed as a Bode plot. Also, you can plot the sine input and output at any frequency.



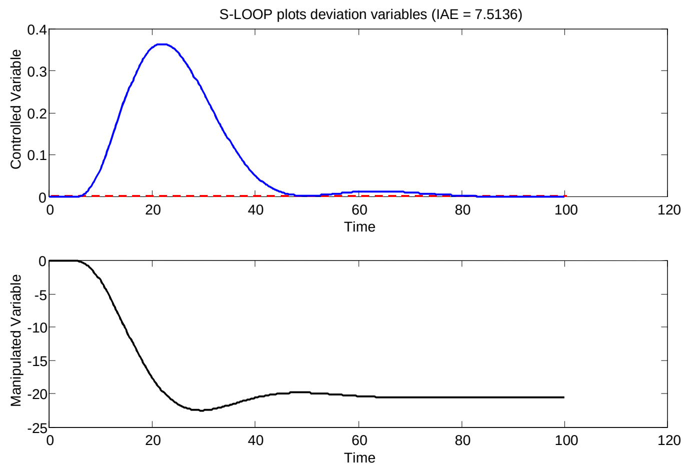
7) Closed-loop Frequency Response - Calculate the amplitude ratio of the closed-loop control system to evaluate the control performance as a function of input frequency. The user can select the input to be a disturbance or set point. . Also, you can plot the sine input and output at any frequency.



8) **Dynamic Simulation -** Calculate the closed-loop dynamic response in the time-domain to a step input, which is introduced as a step at 5% of the total simulation time. The user can select the size of the desired input step, and set all others to zero. The type of input can be a
a) process reaction curve (step in the manipulated variable with the controller off),
b) disturbance, or
c) set point change.

The controller can be selected to be a continuous PID (executed every simulation time step) or a digital PID (executed at the period selected by the user).

The variables are plotted in deviation variables from their initial values of 0.0. The manipulated variable does not observe any bound, although all final elements have limits in a real plant. The IAE of the controlled variable is reported in the figure.




```
                                    PRESENT VALUES
1) Total simulation time              100.00
2) Time step for simulation           0.200

3) Set point change                   0.00
4) Disturbance change                 0.80
5) Process reaction curve MV input    0.00

6) Select continuous/digital controller, currently continuous
   (Controller executed every simulation time step)

7) Execute dynamic simulation
8) Return to main menu
```

For Submenus 5-7, the user has the ability to define the frequency range for the Bode plot. This choice determines the clarity of the Bode plot. For the Stability analysis in Submenu 5, the frequency range must include the frequency at which the phase angle crosses –180 degrees. If the range does not include the critical frequency, the program provides a warning to the user.

For Submenu 8, the user enters the total time and the time step used for the numerical simulation. To provide reasonable accuracy, the time step should be a small fraction of the smallest non-zero time constant. If the combination of the total time and time step, Δt, yields a large number of steps, the simulation would be unnecessarily time consuming. Therefore, the program checks these inputs automatically and ensures that the number of steps is less than the maximum allowed, which is currently set at 20000 (Usually, about (total time)/(step size) $\approx$ 1000 steps gives adequate accuracy.) If the program resets the time step, the software displays a message to the user and changes the value displayed in the menu.

SAMPLE SESSION

After entering MATLAB, the user must define the MATLAB path where the `s_loop` m-files are stored; naturally, the path will be different for every installation. After the correct directory has been defined, the user types `s_loop` and strikes enter in the MATLAB window to begin the program. The Main Menu will appear on the screen, indicating that the S_LOOP program is running.

For this sample session, entries for Example 9.2 from the textbook are discussed. The essential data are summarized in the following.

$$\frac{CV(s)}{MV(s)} = G_p(s) = \frac{K_p}{(\tau_p\ s + 1)^3} = \frac{0.039}{(5\ s + 1)^3}$$

$$\frac{CV(s)}{D(s)} = G_d(s) = \frac{K_d}{(\tau_d\ s + 1)^3} = \frac{1}{(5\ s + 1)^3}$$

First, the user must enter the data from the attached table into the program. This is done for the process parameters (Submenu 1), the disturbance (Submenu 2) , and the controller (Submenu 3). Note that program returns to the Main Menu simply by striking the "enter" with a blank entry from any Submenu.

Submenu

1) The process model defined in Submenu 1 includes all process and instrumentation elements in the feedback loop. The tacit assumption is that the single model between the manipulated and controlled variables includes the process and final element, i.e., $G_p(s)=G'_p(s)G_v(s)$, and that the sensor contributes negligible dynamics, i.e., $G_s(s)=1.0$. The third order mixing system can be represented in the S_LOOP format using the following

$$\frac{0.039}{(5\ s + 1)^3} = \frac{0.039}{(5\ s + 1)\ (5\ s + 1)^2} = \frac{0.039}{(5\ s + 1)\ (25\ s^2 + 10\ s + 1)}$$
$$= \frac{K_p}{(\tau_1\ s + 1)\ (\tau_2\ s + 1)\ (\tau_3^2\ s^2 + 2\ \tau_3\ \xi\ s + 1)}$$

Thus, $K_p = 0.039$, $\tau_{lead} = 0$, $\tau_1 = 0$, $\tau_2 = 5$, $\tau_3 = 5$ and $\xi = 1.0$.

2) The disturbance model is entered via Submenu 2. In this case, only the steady-state gain is different from the process model, although in other cases many parameters may be different.

Thus, $K_d = 1.0$, $\tau_{lead\_d} = 0$, $\tau_{d1} = 0$, $\tau_{d2} = 5$, $\tau_{d3} = 5$ and $\xi_d = 1.0$.

3) The controller tuning is of $K_c=30$, $T_I=11$ and $T_d = 0.80$ is entered through Submenu 3. At any time the user wishes to turn off the controller, the controller gain can be set to zero ($K_c = 0$).

8) The purpose of the exercise is to evaluate the dynamic response; thus, Submenu 8 is selected. The user may select the total time, here 200 to be consistent with Figure 9.6. The time step is selected to give good accuracy without exceeding the limit of 20000 steps; thus, 0.20 is selected. This gives $\Delta t/\tau = 0.04$, which should provide reasonable accuracy. Note that for systems with dead time, which this example does not have, the dead time is approximated as an ***integer multiple*** of the step size.

Continuing in Submenu 8, the user has the option of a unit step in the set point, disturbance, or manipulated variable (a process reaction curve). The user selects the desired input by entering a non-zero value for the changing input and zero values for the other inputs. If the user enters a non-zero value for the manipulated variable, the feedback controller is automatically turned off. The simulation is started by selecting Submenu item 7. The results are automatically plotted on the screen and can be compared with the answers in Example 9.2 and Figure 9.6.

Although Example 9.2 does not deal with other aspects of the system behavior, they are discussed here to demonstrate all program features.

Submenu

4) The roots of the characteristic equation are the exponents of the time-domain solution for the dynamics. For this system, the characteristic equation is

$$0 = 1 + G_p(s)\,G_c(s) = 1 + \frac{.039}{(1+5s)^3}\,30\left(1 + \frac{1}{11s} + 0.8s\right)$$

First, the program rearranges this equation into a polynomial form; the user ***does not*** have to perform this reformulation. The roots for this equation are -0.3626, -0.0778 + 0.1504i, -0.0778 - 0.1504i, and -0.0818. Since all real parts are negative, the system is (bounded input-output) stable. Also, since two roots are complex the system is underdamped, as can be seen in the dynamic response.

If the controller gain is set to zero, the roots of the **process** transfer function, $G_p(s)$, are evaluated.

5) The Bode plot of the transfer function including all elements in the feedback path, $G_{OL}(s)$, is evaluated via this Submenu. Recall that this transfer function can be used to i) evaluate the tuning by Zeigler-Nichols method by setting $G_c(s)=1.0$ ($K_c=1.0$, $T_I=0$, and $T_d=0$) or 2) testing the stability of the system by checking the amplitude ratio at the critical frequency with any controller, $G_c(s)$. Since the controller tuning has been entered, $G_{OL}(s)$ is given in the following.

$$G_{OL}(s) = G_p(s)\,G_c(s) = \frac{.039}{(1+5s)^3}\,30\left(1 + \frac{1}{11s} + 0.80s\right)$$

For the parameters in the example, the Bode plot demonstrates that $|G_{OL}(j\omega_c)|<1.0$ at the critical frequency, i.e., the frequency at which the phase angle equals -180°. The program provides the user with values of the critical frequency (0.35 rad/min) and amplitude ratio (0.139) in the MATLAB window. Therefore, the system is stable.

6) The Bode plot of the process, $G_p(j\omega)$, is evaluated. This can be used in determining how the process without control would attenuate a sine disturbance.

7) The closed-loop Bode diagram is evaluated in this Submenu. Recall that this frequency response gives the amplitude of the controlled variable relative to the input for a range of frequencies. The user must enter the choice of set point or disturbance input to obtain the appropriate result.

$$\textbf{setpoint:}\quad \frac{CV(s)}{SP(s)} = \frac{G_p(s)G_c(s)}{1+G_p(s)G_c(s)} = \frac{\dfrac{.039}{(1+5s)^3}\,30\left(1+\dfrac{1}{11s}+0.80s\right)}{1+\dfrac{.039}{(1+5s)^3}\,30\left(1+\dfrac{1}{11s}+0.80s\right)}$$

$$\textbf{disturbance:}\quad \frac{CV(s)}{D(s)} = \frac{G_d(s)}{1+G_p(s)G_c(s)} = \frac{\dfrac{1}{(1+5s)^3}}{1+\dfrac{.039}{(1+5s)^3}\,30\left(1+\dfrac{1}{11s}+0.80s\right)}$$

The disturbance is selected through the Submenu item 3 defining parameter “d”, and the result is plotted in a Bode plot. The result for the disturbance shows that $|CV(j\omega)|/|D(j\omega)|$ has

- a small magnitude (small error) at low frequencies (because of effective feedback control),
- a small magnitude (small error) at high frequencies (because of the process filtering the disturbance)
- a resonance peak at a frequency of about 1 rad/min where feedback control is not very effective.

The user can evaluate the set point Bode plot by selecting the “s” input and running the program again. The Bode plot shows the excellent set point tracking at low frequencies and poor tracking at high frequencies.

The value of the program is that the user can evaluate control performance and stability using various frequency- domain and time-domain methods, and s/he may then compare the results and change process and controller parameters to evaluate the effects.

LIMITATIONS

S_LOOP should provide excellent experience with frequency-dependent and time-domain behavior of single-loop feedback control. The major limitations are noted below.

1) The number of time steps, determined from (total time)/(step size), is not allowed to be greater than a maximum limit. This limit is currently set at 20000 time steps. Typically, the time duration (t_{end}) divided by the step size (Δt) less than 1000 provides adequate accuracy for dynamic systems in the textbook.

The dead time used in the dynamic simulations of the time-domain dynamic response (but not the frequency response calculations) are rounded to an *integer multiple* of the simulation step size.

The controller execution period is rounded to an *integer multiple* of the simulation time step.

2) The calculation of the frequency response involves the phase angle that may exceed -180°. MATLAB has a feature (the `UNWRAP` function) which ***usually*** correctly recognizes a change of quadrant and accounts for this fact in calculating the `arctangent`; however, this feature is not foolproof. Spurious results, phase angles larger than their proper values, can be obtained; therefore, the Bode plot should be scutinized when performing the Bode stability analysis. Narrowing the range of the frequencies often corrects the problem, should it occur.

3) The process and disturbance transfer functions are limited to fourth order. No higher orders can be simulated.