

Tutorial on IMCTUNE Software

Objectives

- Provide an introduction to IMCTUNE software.
- Describe the tfn and tcf commands for MATLAB that are provided in IMCTUNE to assist in IMC controller selection and to facilitate transfer function analysis.

G.1 INTRODUCTION

IMCTUNE facilitates the design and tuning of the following types of controllers with or without model uncertainty.

- 1DF IMC controllers
- 2DF IMC controllers
- MSF IMC controllers
- 1DF and 2DF PID controllers
- IMC and PID Feedforward and Cascade controllers

The IMCTUNE software package is a collection of MATLAB m-files that can be downloaded from <http://www.phptr.com/brosilow/>. It requires MATLAB 5.3 or higher, and the Control System and Optimization Toolboxes. The software can also compute and display

- single-loop and cascade IMC, MSF, and PID closed-loop responses to step setpoint and disturbance changes,
- the SIMULINK diagrams used to simulate such structures,
- various closed-loop upper and lower-bound frequency responses, and
- individual process closed-loop frequency responses.

We recommend installing IMCTUNE in its own directory and creating a subdirectory called *data* under the IMCTUNE directory to store the data files for IMCTUNE.

G.2 GETTING STARTED ON 1DF SYSTEMS

Either put the IMCTUNE directory in the MATLAB path, or change directories inside MATLAB to the IMCTUNE directory. Start IMCTUNE by typing *imctune* in the MATLAB Command Window. This should open up the IMCTUNE's primary interface shown in Figure G.1.

The primary IMCTUNE interface (c.f. Figure G.1) has been structured to eventually accommodate multi-input, multi-output processes. However, the version of IMCTUNE available on the Prentice Hall website is only for single-input, single-output processes, and that is all that is needed for this text. Thus, the number of inputs and outputs should remain at the default of one. The drop down menu labeled A in Figure G.1 offers the following structure options: one-degree of freedom, two-degree of freedom, and cascade system. We shall focus on the default one-degree of freedom system, and comment only on the major differences in the other two structures. The drop down menu labeled B in Figure G.1 offers a single-term transfer function for the process and model, or a two-term transfer function. The latter allows entry of process descriptions like $(1 + .5e^{-s})/(s + 1)(2s + 1)$. Other inputs are similar to those for the single-term transfer function.

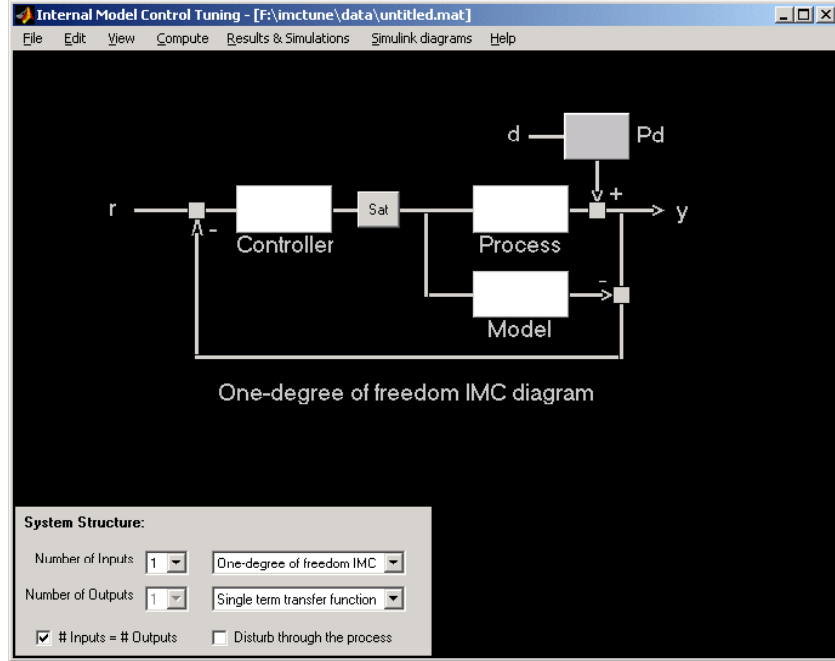


Figure G.1 IMCTUNE 1DF primary interface.

G.2.1 Data Input

Clicking on any block allows the user to enter the transfer function and parameters for that block. For example, clicking on the process block brings up the window shown in Figure G.2.

The coefficients of the numerator polynomial, and denominator polynomial are entered as vectors, and the time delay as a scalar. Consider the process, model and controller given by

$$p(s) = \frac{1}{s+1} e^{-Ts} \quad .5 \leq T \leq 1.5, \quad (\text{G.1})$$

$$\tilde{p}(s) = \frac{1}{s+1} e^{-s}, \quad (\text{G.2})$$

$$q(s) = \frac{(s+1)}{(\epsilon s+1)}. \quad (\text{G.3})$$

The process window for Eq. (G.1) is shown in Figure G.3. Notice that the uncertain dead time is entered as the letter u. This shows up in the *current process* portion of the window as $\exp(-x(1)s)$. The upper and lower-bounds for $x(1)$ are entered as shown in Figure G.3.

Process (edit) =

Disturbance lag (edit): ☐ Disturbance enters through process.

Current process:

Current Disturbance lag:

Upper limits of x:

Lower limits of x:

Figure G.2 Blank process screen.

Process (edit) =

Current process:

Only x [1] required

Upper limits of x:

Lower limits of x:

Figure G.3 Screen after entering the process of Eq. (G.1).

G.2.2 Entering General Numerator and Denominator Polynomials for a Process or Model

A polynomial may be entered in the following two ways.

G.2.2.1 Expanded Form

The polynomial $\alpha_n s^n + \alpha_{n-1} s^{n-1} + \dots \alpha_1 s + \alpha_0$ is entered using only its coefficients as $\alpha_n \alpha_{n-1} \dots \alpha_1 \alpha_0$. That is, the coefficients are entered separated by spaces. Any uncertain coefficient is entered as u and its limits are given in the rows labeled upper and lower limits. IMCTUNE will assign a variable $x(i)$ to each uncertain coefficient you enter. For example, if α_n and α_1 are uncertain coefficients, the polynomial would be entered as $u \alpha_{n-1} \dots u \alpha_0$. If any uncertain coefficients are the same they are entered using $u1, u2$, etc. For example, if the coefficients $\alpha_n, \alpha_{n-1}, \alpha_1$, and α_0 of the polynomial $\alpha_n s^n + \alpha_{n-1} s^{n-1} + \dots \alpha_1 s + \alpha_0$ are uncertain, and α_{n-1} and α_0 are the same, but different from α_n and α_1 , which are also the same, then the polynomial is entered as $u1 \ u2 \ \alpha_{n-2} \dots \alpha_2 \ u1 \ u2$.

G.2.2.2 Factored Form

The polynomial $(s + \alpha_1)(s + \alpha_2)(s^2 + \alpha_3 s + \alpha_4)$ can be entered as $1 \ \alpha_1; 1 \ \alpha_2; 1 \ \alpha_3 \ \alpha_4$. Again, each independent uncertain coefficient is entered as u .

If the disturbance passes through a lag, this lag can be entered in either of two ways. First, it can be entered just like the process by clicking on the Pd block in Figure G.1, which brings up Figure G.4.

Figure G.4 Disturbance lag screen.

Numerator and denominator polynomials are entered as described above. There is no deadtime element since an unmeasured delayed disturbance cannot be distinguished from one that simply enters later without passing through a deadtime.

Second, if the disturbance passes through the process, check the box labeled *disturbance through the process* in the primary screen (Figure G.1).

G.2.1.3 Parallel Processes

If the process is modeled as a sum of two transfer functions, it is entered by opening the pull down menu labeled B in Figure G.1, and selecting *two-term transfer function*. Clicking on the process block, and entering the process given by Eq. (G.4), results in the screen of Figure G.5.

Process 1 : (1,1)

Process (edit) = $\frac{2u_1}{u_1 1} + \frac{u}{1 1}$ (Deadtime: 0)

Process (result) = $\frac{2x(1)}{(x(1)s+1)} + \frac{x(2)}{(s+1)} \exp(-1s)$

Current Process String: $2x(1)/((x(1)s+1))+x(2)*exp(-1s)/(s+1)$

Only x [1 2] required

Upper limits of x: 1.2 1.3

Lower limits of x: .8 .7

OK Help

Figure G.5 Screen after entering the process of Eq. (G.4).

In this process we have entered the process model

$$p(s) = \frac{2x(1)}{x(1)s + 1} + \frac{x(2)}{s + 1} e^{-s}, \quad 0.8 \leq x(1) \leq 1.2, \quad 0.7 \leq x(2) \leq 1.3 \quad (G.4)$$

Note that the uncertainty in gain and time constant of the first term are correlated. This correlation is entered using $u1$ rather than u for the uncertain variable. If you have more correlated uncertain variables, use $u1$, $u2$, and so on. IMCTUNE automatically

assigns variable names $x(1)$, $x(2)$, and so on, for each independent uncertain variable, as shown in Figure G.5.

G.2.1.4 Uncertainty Bounds

Upper and lower-bounds must be provided for each uncertain parameter (entered as u or $u\#$) in the transfer function. See Figures G.3 and G.5. Such bounds are entered as arrays of numbers separated by spaces just like the coefficient vectors.

G.2.1.5 Entering the Model

By clicking once in the block labeled *Model*, you can enter the model. The model transfer function is entered using the same format as described previously, either as the coefficients of a single polynomial or as the coefficients of factored polynomials separated by semicolons. Since a future implementation of IMCTUNE may find optimal values for model parameters, the current IMCTUNE interface accepts uncertain parameters for the model, which become variables $y(1)$, $y(2)$, ..., $y(n)$. However, we recommend that in using the current version, only constant values should be entered for model parameters. Figure G.6 shows the Model window for the model of Eq. (G.2) using a mid-range deadline.

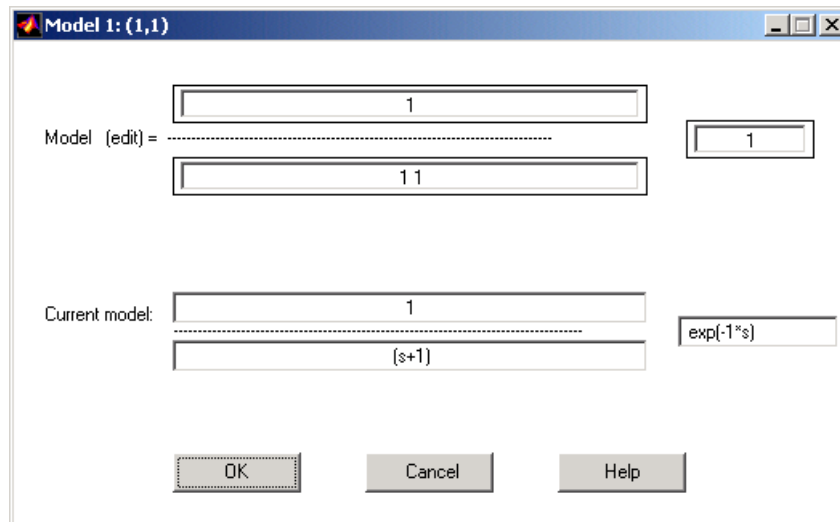


Figure G.6 Model window for Eq. (G.2).

G.2.1.6 Controller

The IMC controller is entered in the controller window by specifying the part of the process model to be inverted by the controller. The filter order is generally chosen to be the relative degree of the transfer function to be inverted. Any filter time constant may be entered here, but it is usually calculated by IMCTUNE either to satisfy an M_p criterion for an uncertain process, or to satisfy a high frequency maximum noise amplification specification for a perfect model. Figure G.7 shows the IMC controller window for the controller given by Eq. (G.3) using a filter time constant of 0.05 for a maximum noise amplification of 20.

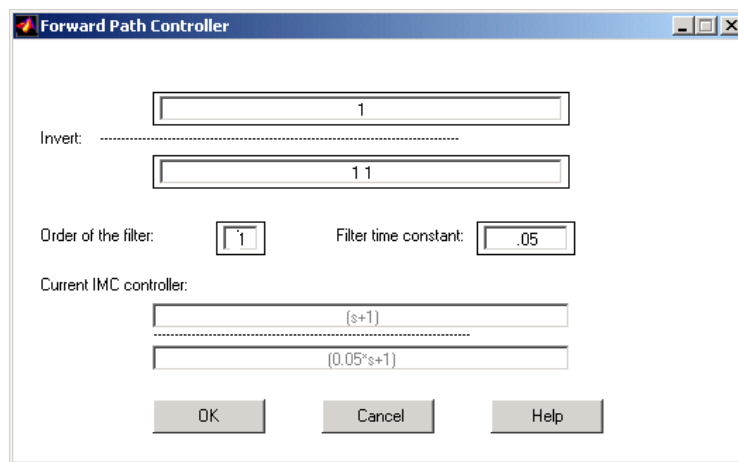


Figure G.7 IMC controller window for the process model of Figure G.6.

G.3 MENU BAR FOR 1DF SYSTEMS

The menu bar has the following options:

File

- New: Start a new design.
- Load: Load a file containing data describing a previous problem and controller design.
- Save: Save current state of IMCTUNE as a *mat* file in the data subdirectory of the current directory.
- Save as: Save current state of IMCTUNE with a file name, and in a directory as specified by the user.
- Exit: Quit IMCTUNE.

Edit

- Saturation bounds: For entering upper and lower limits of the manipulated variable (same as clicking on the *sat* block).
- Default values: You can enter values for maximum allowed noise amplification, frequency range for closed loop frequency response calculations, the number of points per decade used in the calculation, and the desired accuracy of the calculation. Any number larger 10^5 is treated as infinity during calculations. The default values are shown in Figure G.8. Any of these may be changed. The frequency range is in powers of 10 (e.g., 0.1 to 10) for Figure G.8.

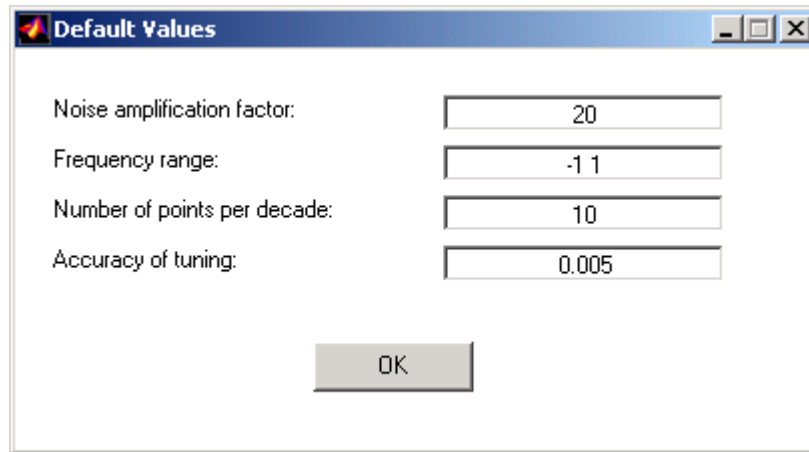


Figure G.8 Default values.

View

View brings up the summary of all parameters entered so far as in Figure G.9 for the process of Eq. (G.1), and the model and controller of Equations (G.2) and (G.3).

The screenshot shows the 'System' window in IMCTUNE software. It contains the following fields and sections:

- System structure:** One degree of Freedom
- Process:** $1/[(s+1)] \cdot \exp(-x(1) \cdot s)$
- Model:** $1/[(s+1)] \cdot \exp(-1 \cdot s)$
- Controller:** $(s+1)/[(0.05 \cdot s+1)]$
- Disturbance lag, model:** Two input fields, both containing '1'.
- Saturation bounds:** Two input fields, containing '10000' and '-10000'.
- Uncertainty on process parameters:** A box containing the text $0.5 \leq x(1) \leq 1.5$.
- Default variables:** A table with the following values:

Noise amplification factor	20
Frequency Range (rad/unit time)	1×10^{-2} to 1×10^2
# of frequency points per decade	10
Tolerance	0.005
Infinity	Inf
- OK** button at the bottom center.

Figure G.9 View window for Equations (G.1), (G.2) and (G.3) with $\varepsilon = .05$.

Compute

The choices under the Compute menu option are

- **Tuning:** Computes filter time constant via Mp Tuning (see Chapter 7). Figure G.10 shows the tuning results for an Mp specification of 1.05 for Eq. (G.1).
- **Noise amplification filter:** Computes filter time constant that satisfies the noise amplification set in the default values window (see Chapter 3).
- **MSF K, K_{sp} :** Computes the MSF coefficients (see Chapter 5).
- **Find uncertainty bounds:** Given a model, a filter time constant, and a set of nominal values of the uncertain parameters, IMCTUNE computes the fractional variation of the uncertain parameters around the nominal values for which the specified Mp will be met (see Chapter 3).
- **Tuning for lower-bound saturation:** Computes a safe lower-bound for the IMC filter time constant for MSF (see Chapter 5).
- **PID controller:** Computes the parameters of an ideal PID controller, and PID controllers cascaded with 1st and 2nd order lags (see Chapter 6).
- **Frequency response:** Computes upper- and/or lower-bounds for the sensitivity or the complementary sensitivity functions. Individual process frequency responses can be added to the upper and/or lower-bounds results via the add button (see Chapter 7 and Figure G.10).

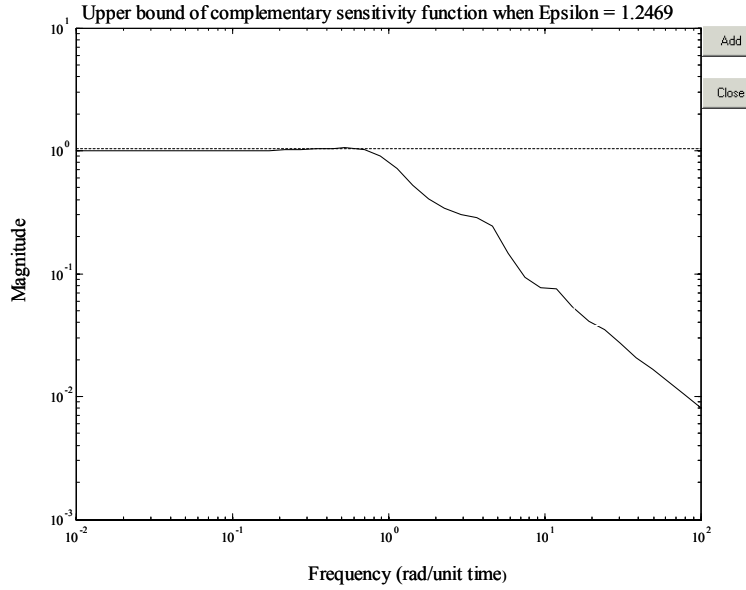


Figure G.10 Results from Mp Tuning of Eq. (G.1) for Mp = 1.05.

Results & Simulations

- PID controller: Shows the results of the most recent calculations of PID parameters.
- MSF gain, K, K_{sp} : Computes the MSF feedback parameters (see Chapter 5).
- IMC & MSF step responses: Provides, and allows comparison of, IMC and MSF step setpoint and disturbance changes as shown in Figures G.11a and G.11b.
- IMC & PID step responses: Provides, and allows comparison of, IMC and MSF step setpoint and disturbance changes as shown in Figure G.12.

The responses in Figures G.11a, G.11b, and G.12 are for the following process, model, and IMC controller

$$\tilde{p}(s) = p(s) = \frac{e^{-s}}{(s+1)}, \quad q(s) = \frac{(s+1)}{(\epsilon s + 1)} \quad 0 \leq u(t) \leq 1.1. \quad (G.5)$$

The model and IMC controller in Eq. (G.5) are the same as for Equations (G.2) and (G.3), but the process has no uncertainty so as to show up the difference between IMC and MSF responses. Also, the control effort is constrained as shown. The button windows in Figure G.12 are the same as for Figure G.11b, except for the *add and remove* window, which is shown on the figure.

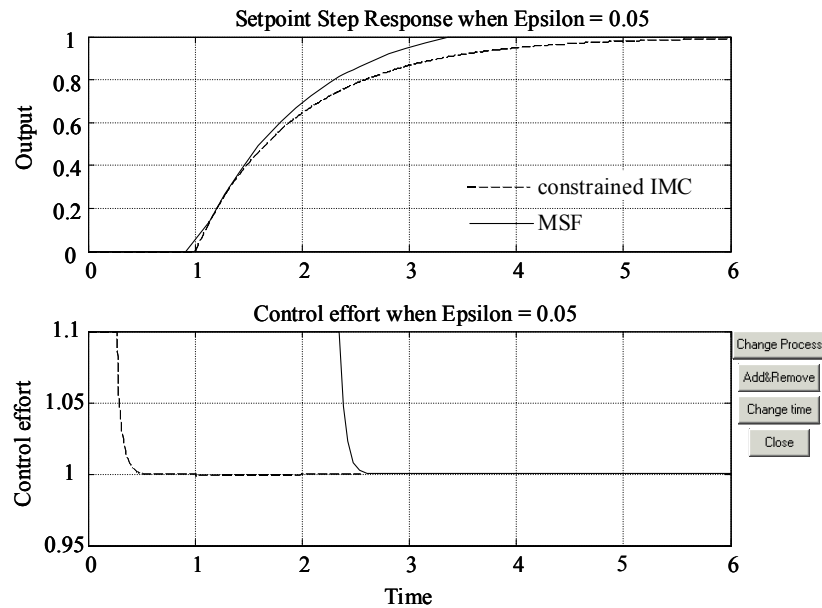


Figure G.11a IMC & MSF step responses.

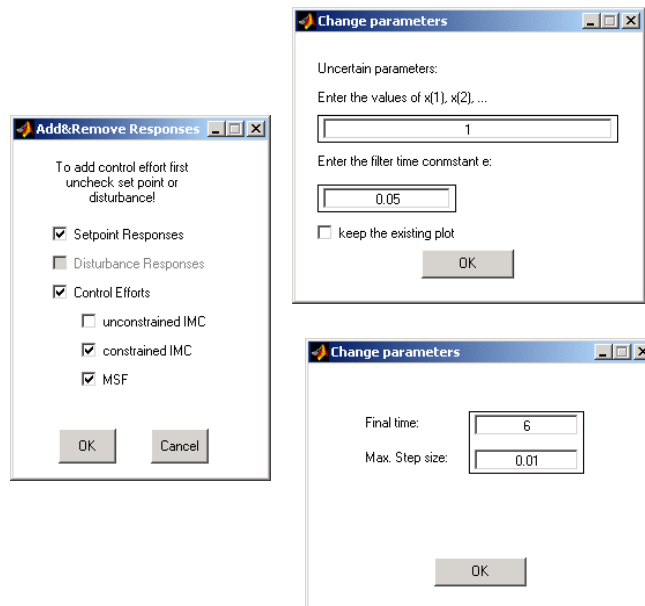


Figure G.11b Screens associated with the buttons of Figure G.11a.

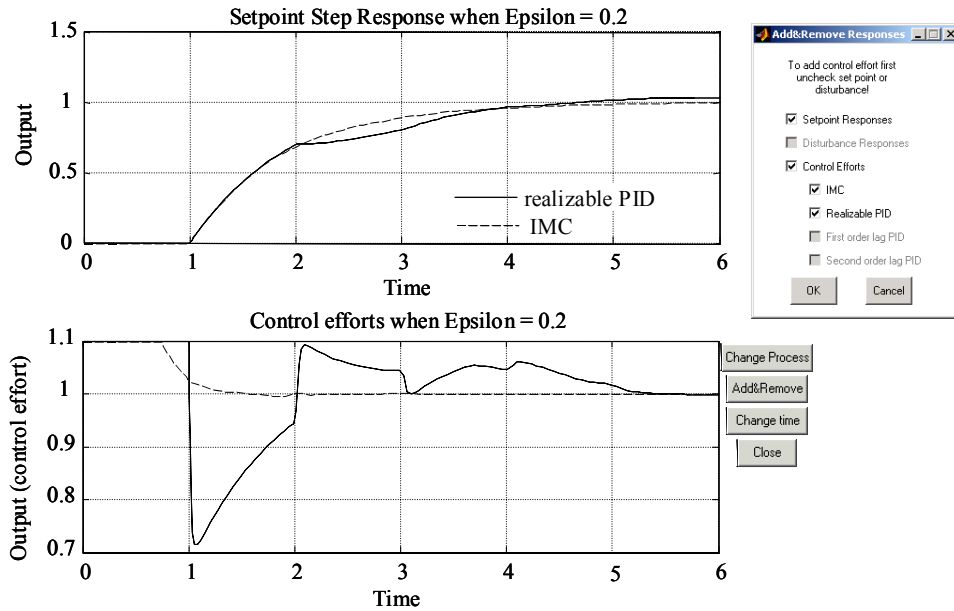


Figure G.12 IMC and PID step responses.

SIMULINK diagrams

The user has access to the SIMULINK programs that carry out the IMC, MSF, and PID simulations, and can modify these diagrams as desired. However, to maintain the integrity of the IMCTUNE software, all the diagrams can be restored to their original form.

G.4 GETTING STARTED ON 2DF SYSTEMS

Figure G.13 shows the primary IMCTUNE interface for 2DF control systems for the case where the disturbance passes through the process. As the reader can see from Figure G.13, the controller is now split into two parts: a forward path part and a feedback path part. While the menu bar for 1DF and 2DF systems are the same, the items under View and Compute are different, as are the model and controller windows. This section describes the differences in the controllers and the model. The next section describes the differences in View and Compute menus.

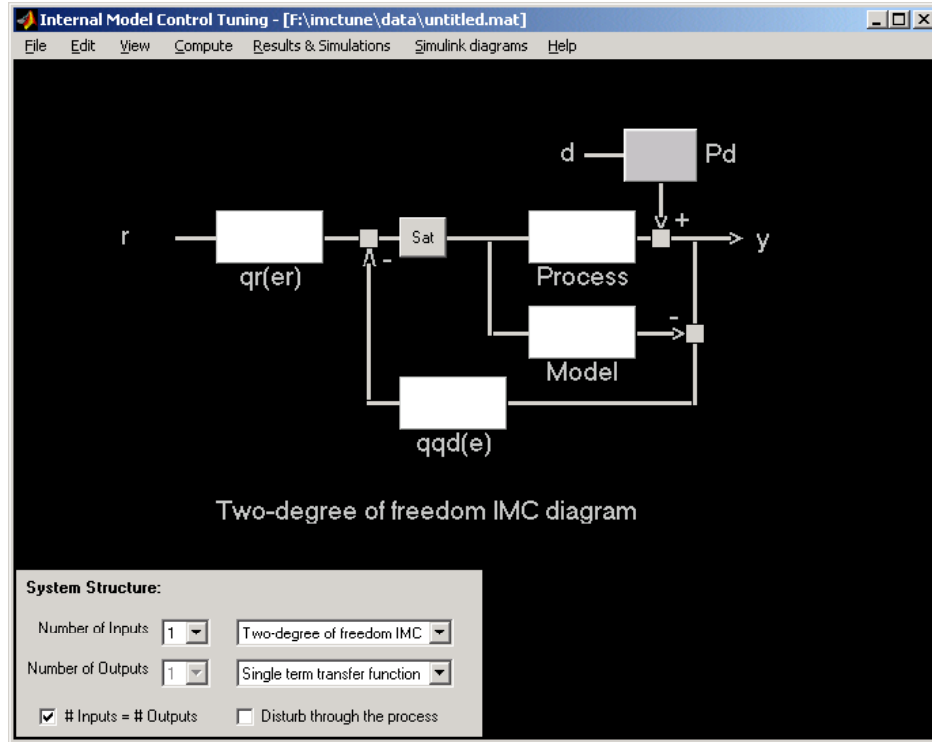


Figure G.13 IMCTUNE 2DF primary interface.

G.4.1 The Model

Figure G.14 shows the model window for a 2DF system using the same model as in Eq. (G.2). There is one very important difference with the model window for 1DF systems. There is now a model *disturbance lag*. This lag actually has nothing to do with the model, but rather establishes the form of the $q_d(s)$ part of the feedback path controller. Future versions of IMCTUNE will therefore have the *disturbance lag* moved to the feedback path controller window. Its location in the model is due to the history of the development of IMCTUNE. The $q_d(s)$ part of the feedback path controller is selected so as to have the zeros of $(1 - \tilde{p}q_d(s))$ the same as the roots of the disturbance lag.

Figure G.15 shows the feedback path controller window for the model of Figure G.14 for a filter time constant of .7. Figure G.16 shows the forward path controller. Figure G.17 shows the process, model, and controllers for all windows in Figures G.14 through G.16.

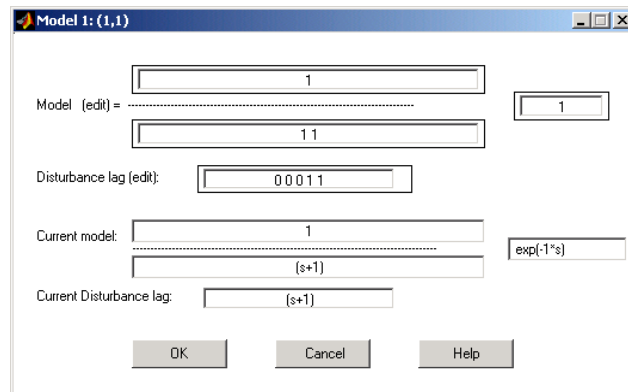


Figure G.14 2DF model window.

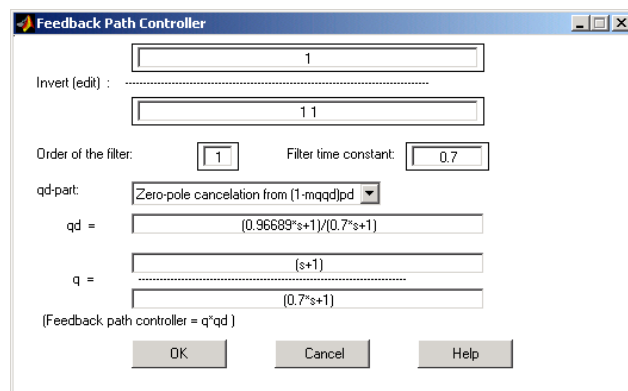


Figure G.15 2DF feedback path controller window.

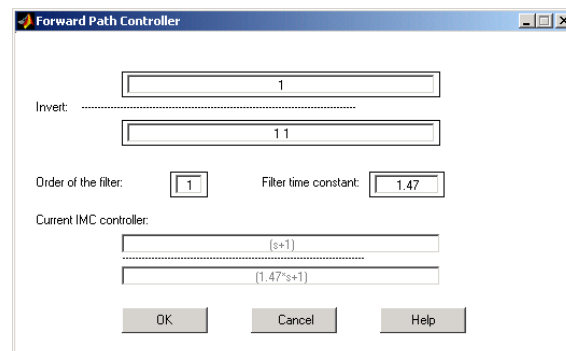


Figure G.16 2DF forward path controller window.

G.5 MENU BAR FOR 2DF SYSTEMS

View

Figure G.17 shows the current system. Notice that the feedback controller has two parts, q , and q_d . Also, the filter time constants for the forward path and feedback path controllers are not the same.

The screenshot shows the 'System' window with the following configuration:

System structure:		Two degree of Freedom	
Process:	$1/[(s+1)]^2 \exp(-x(1)s)$		
Model:	$1/[(s+1)]^2 \exp(-1s)$		
Forward path controller:	$(s+1)/(1.47s+1)$		
Disturbance lag, model:	"Through Process"	$[s+1]$	
Feedback controller =	$(s+1)/(0.7s+1)$	×	$(0.96689s+1)/(0.7s+1)$
Saturation bounds:	10000	-10000	

Uncertainty on process parameters	Default variables
$0.5 \leq x(1) \leq 1.5$	Noise amplification factor: 20
	Frequency Range (rad/unit time): 1×10^{-2} to 1×10^2
	# of frequency points per decade: 10
	Tolerance: 0.005
	Infinity: Inf

OK

Figure G.17 View screen for 2DF system.

Compute

The choices under the Compute menu option are similar to those for 1DF systems. Therefore, below we emphasize the differences

- 2DF tuning: Has both feedback path controller tuning (inner loop tuning) via the partial sensitivity function and forward path controller tuning via the complementary sensitivity function (see Chapter 8).
- 2DF PID controller: Provides the setpoint filter as well as the PID controller approximation for $c(s)$ in Figure 6.7 of Chapter 6.
- Frequency response: For the response of the output to the disturbance, computes the sensitivity function, the integrated sensitivity function, the normalized integrated sensitivity function, and the partial sensitivity function.

G.6 GETTING STARTED ON CASCADE CONTROL SYSTEMS

Figure G.18 shows the primary window for cascade control systems.

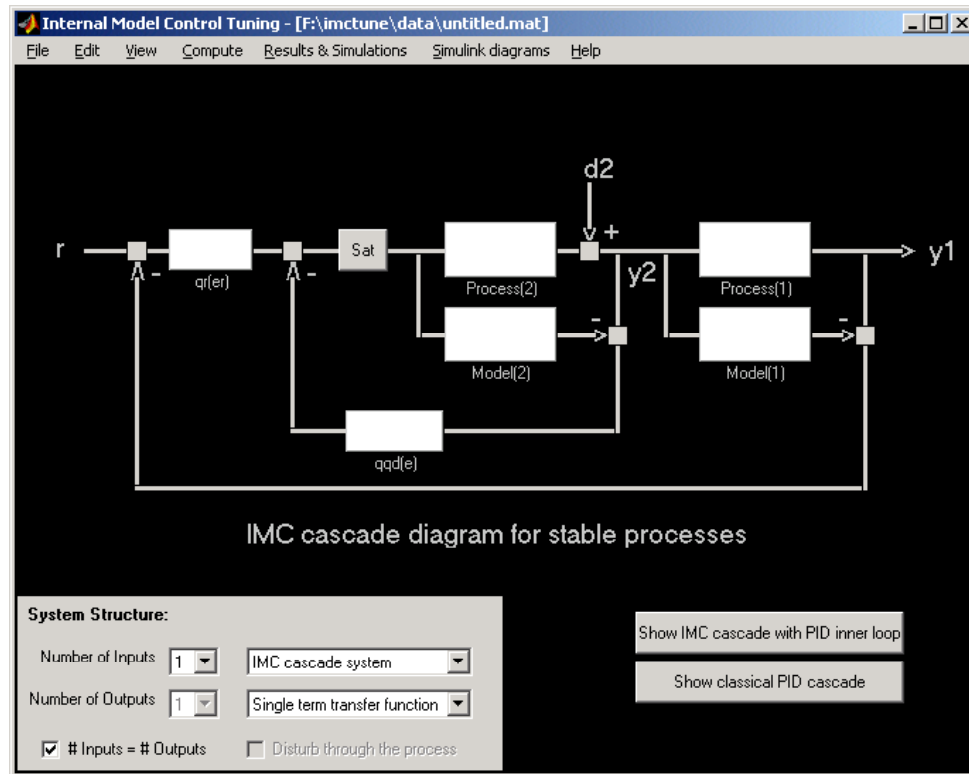


Figure G.18 Cascade control system primary interface.

Cascade control system design and tuning starts with the IMC cascade structure shown in Figure G.18. Once the controllers $qq_d(s, \epsilon)$ and $q_r(s, \epsilon_r)$ have been designed and tuned, the user can select two other cascade configurations: (1) an IMC cascade with a PID inner loop as in Figure G.19 or (2) a classical PID cascade control system as in Figure G.20. Clicking on either of the *Show* buttons on the lower right of the primary interface brings up the diagrams in Figures G.19 and G.20. However, neither of these diagrams is active until the computations under the compute menu are activated as described in the next section.

After computation of the various PID controllers, clicking on a PID controller icon brings up a screen containing descriptions of the various possible PID controllers. Clicking inside the screen allows scrolling up and down within the screen.

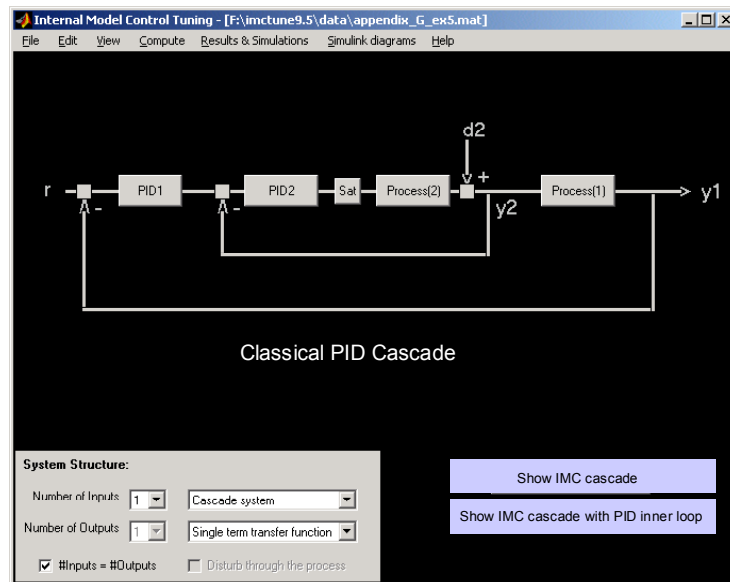


Figure G.19 IMC cascade with PID inner loop.

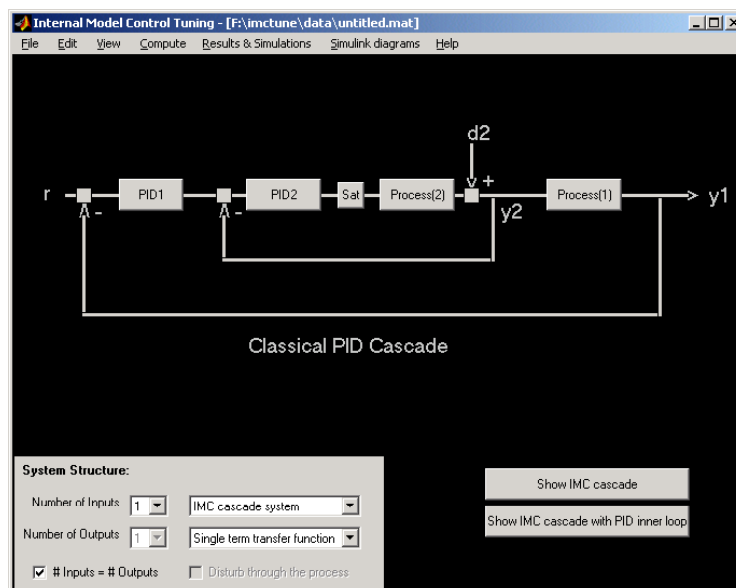


Figure G.20 Traditional PID cascade.

G.7 MENU BAR FOR CASCADE SYSTEMS

Here again, we review only those items in the menu bar that differ significantly from the menu bar of 1DF systems.

View

Figure G.21 shows a typical view window for uncertain inner loop and outer loop processes. The deadtime in the outer loop process is uncertain, while the gain is the uncertain element in the inner loop.

The 'System' window displays the following parameters:

Process(1):	$1/((s+1)) \cdot \exp(-x(1) \cdot s)$	
Process(2):	$x(2)/((s+1)) \cdot \exp(-1 \cdot s)$	
Model(1):	$1/((s+1)) \cdot \exp(-1 \cdot s)$	
Model(2):	$1.5/((s+1)) \cdot \exp(-1 \cdot s)$	
Forward path controller:	$(s+1)^2 / (1.5 \cdot (1.0218 \cdot s + 1)^2)$	
Process(1) lag, model:	$1/((s+1))$	$(s+1)$
Feedback controller =	$(s+1) / (1.5 \cdot (0.7 \cdot s + 1))$	$\times (0.96689 \cdot s + 1) / (0.7 \cdot s + 1)$
Saturation bounds:	10000	-10000

Uncertainty on process parameters

$0.5 \leq x(1) \leq 1.5$
$0.5 \leq x(2) \leq 1.5$

Default variables

Noise amplification factor	20
Frequency Range (rad/unit time)	1×10^{-3} to 1×10^1
# of frequency points per decade	10
Tolerance	0.005
Infinity	Inf

OK

Figure G.21 Typical view window.

Compute

- 2DF tuning: Allows design and tuning of the inner loop controller as a 2DF controller based on the lag of the outer loop process, as well as the design and tuning of the outer loop controller as a 1DF controller.
- Noise amplification: Computes the minimum filter time constant for the inner loop to achieve the desired maximum noise amplification.

- IMC controller with PID inner loop: Converts the IMC cascade of Figure G.18 to the diagram in Figure G.19. Clicking on the controller icons shows their transfer functions.
- Classical PID cascade: Converts the IMC cascade of Figure G.18 to the classical cascade structure of Figure G.20. Clicking on the controller icons shows their transfer functions.
- Frequency response: Computes the frequency response of the transfer functions (1) between the setpoint r and the primary output y_1 , and (2) from the disturbance d_2 and the output of the inner loop y_2 , with the outer loop open.

Results and Simulations

- Step responses for cascade with outer loop IMC controller: Provides, and allows comparison of, IMC cascade and IMC cascade with PID inner loop. The *Add & Remove* menu permits selection of just IMC cascade or just IMC cascade with a PID inner loop or both. It also permits selection of different controllers for the inner loop (e.g., a PID controller cascaded with a first or second order lag). The uncertain parameters in both inner and outer loops can also be changed via the change process button.
- Step responses for cascade with outer loop PID controller: Provides, and allows comparison of, IMC cascade and classical PID cascade. The *Add & Remove* menu permits selection of just IMC cascade or just PID cascade or both. It also permits selection of different controllers for the inner loop (e.g., a PID controller cascaded with a first or second order lag). The uncertain parameters in both inner and outer loops can also be changed via the change process button.

SIMULINK Diagrams

Allows access to all the SIMULINK diagrams used to generate the time responses obtained from the Results and Simulations menu.

G.8 OTHER USEFUL .m FILES INCLUDED WITH IMCTUNE

The files *tfn.m* and *tcf.m* were created in order to facilitate entering into, and manipulating transfer functions in MATLAB. The file, *tcf.m* puts a transfer function into time constant form so that right half plane zeros can be conveniently reflected around the imaginary axis. It also cancels common factors in the numerator and denominator and allows linear and quadratic factors to be easily modified. We recommend copying *tfn.m* and *tcf.m* into the Control System toolbox so that they are always available. As is usual in MATLAB, the commands *help tfn* and *help tcf* bring up instructions on how to use the commands.

G.8.1 TFN.m: Create Transfer Functions as Products of Polynomials Cascaded with a Deadtime

The MATLAB m-file *tfn* takes a scalar gain k , matrices n and d , and dead time td to form a SISO transfer function, using the rows of n and d to form the numerator and denominator polynomials, respectively. If the dead time is omitted, it is taken as zero. For example,

```
num=[1 3]
den=[1 2 5; 0 1 2]
td=6
g= tfn(5,num,den,td)
```

forms:

$$g(s) = \frac{5(s+3)e^{-6s}}{(s^2+2s+5)(s+2)}$$

MATLAB returns the transfer function in three forms as:

Transfer Function as Entered

$$\frac{5*(s+3)}{(s^2+2s+5)*(s+2)}$$

input delay: 6

Time Constant Form:

$$\frac{1.5*(0.33333s+1)}{(0.2s^2+0.4s+1)*(0.5s+1)}$$

input delay: 6

Transfer function:

$$\exp(-6*s) * \frac{5s+15}{s^3+4s^2+9s+10}$$

Notice that each row of *num* or *den* must have the same number of elements. Therefore, lower order polynomials must have explicit zero coefficients for the higher order terms.

G.8.2 TCF.m Time Constant Form

tcf(g) is used to display a transfer function g in time constant form. For example, if

$$g = \frac{5s + 15}{s^3 + 4s^2 + 9s + 10},$$

then, *tcf*(g) returns:

$$\frac{1.5*(0.33333 s + 1)}{(0.2 s^2 + 0.4 s + 1)*(0.5 s + 1)}$$

The command [k, n, d] = *tcf*(g) displays the time constant form of g, prints the dead time associated with g, if any, and returns the transfer function gain as k and its numerator and denominator as the polynomial matrices n and d.

The time constant form will also cancel common factors, if any. Each row of num and den contains the coefficients of a polynomial in the numerator and denominator of g, respectively, with the common factors removed. For example, [k,n,d] = *tcf*(g) for the above transfer function, g, returns:

$$\frac{1.5*(0.33333 s + 1)}{(0.2 s^2 + 0.4 s + 1)*(0.5 s + 1)}$$

input delay: 0

k = 1.5000

n = 0 0.3333 1.0000

d = 0.2000 0.4000 1.0000
 0 0.5000 1.0000