University Badji Mokhtar - Annaba, Engineering Faculty Department of Electronics

Section : M1 Auto

2023/2024

Course : Nonlinear systems

Lab 1 : Using S-function blocks in Simulink

I Motivation

With the complexity of medium-size to large-size nonlinear models, it may be more efficient to use a set of differential equations written in an m-file. These m-files will be accessed by Simulink through the S-function block. Thus, this method mixes the advantages of an m-file which can be run directly by solvers such as *ode*45, with the graphical links to other Simulink blocks.

II Example System

$$\frac{dC_a}{dt} = \left(\frac{F}{V}\right) \cdot \left(C_{af} - C_a\right) - k_0 \cdot \exp\left[-\frac{E_a}{R(T+460)}\right] \cdot C_a$$
$$\frac{dT}{dt} = \left(\frac{F}{V}\right) \cdot \left(T_f - T\right) - \frac{\Delta H}{\rho C_p} \cdot \left(k_0 \cdot \exp\left[-\frac{E_a}{R(T+460)}\right] \cdot C_a\right) - \left(\frac{U \cdot A}{\rho \cdot C_p \cdot V}\right) \cdot \left(T - T_j\right)$$

We want to model this system in which we will treat the jacket temperature, T_j , as the input (i.e. manipulated variable). We will also want to monitor concentration and temperature of the liquid in the CSTR as our outputs.

III Write the m-file

Recall that we could model the process by writing an m-file to be used by Matlab solvers such as ode45. One such file, which we will name as **reactor.m**, is shown in Figure 1.

Test the model to make sure it works. For instance, with Tj = 55:

[t,x] = 0de45(@reactor, [0 10], [0.1; 40], [],55);

Note/recall :

The command-line specifies : a simulation-time span of $[0\ 10]$, an initial-value column vector : [0.1;40], a null placeholder, [], for default options, and setting Tj with a value equal to 55.

IV Write an S-function file

This file will also be saved as an m-file such that Simulink can access information from Matlab. Figure 2 is the S-function file reactor_sfcn.m

```
function
             dx = reactor(t,x,Tj)
8
8
    model for reactor
8
                               % lbmol/ft^3
    Ca = x(1)
                           ;
        = x(2)
    т
                               % oF
                           ;
                               % BTU/lbmol
    Ea = 32400
                           ;
       = 15e12
                               % hr^-1
    k0
                           ;
                               % BTU/lbmol
    dH = -45000
                           ;
                               % BTU/hr-ft^2-oF
    U
        = 75
                           ;
    rhocp = 53.25
                               % BTU/ft^3
                           ;
                              % BTU/lbmol-oF
    R
        = 1.987
                           ;
                               % ft<sup>3</sup>
% ft<sup>3</sup>/hr
    v
         = 750
                           ;
    F
        = 3000
                           ;
    Caf = 0.132
                               % lbmol/ft^3
                           ;
    Tf = 60
                               % oF
                           ;
    A = 1221
                               % ft^2
                           ;
       = k0*exp(-Ea/(R*(T+460)))*Ca;
    \mathbf{ra}
    dCa = (F/V) * (Caf-Ca) - ra;
    dT = (F/V) * (Tf-T) - (dH) / (rhocp) * ra...
             -(U*A)/(rhocp*V)*(T-Tj);
    dx = [dCa;dT];
```

FIGURE 1 – File saved as reactor.m

🖳 Edi	tor - C:\MIAII\reactor_sfcn.m*				
File E	dit Text Go Cell Tools Debug Desk	top Wi	ndow	Help	X 5 K
🗅 🗀	🖬 👗 🖿 🛍 🗠 🖓 🎒	(f	🔁 🗶 🖷 🍘 💕 🖨 🕷 Stack: Base 💌	
0 1	⊑ Ç≡ ↓≡ - 1.0 + ÷ 1.1	× %	% %	0	
1	function [sys,x0,str,ts] = r	eactor	_sfc	n(t,x,u,flag, Cinit, Tinit)	
2 -	switch flag				_
3 -	case O % initialize				
4 -	str=[]		:		
5 -	ts = [0 0]		10		
6 -	s = simsizes		:		
7 -	s.NumContStates	= 2	;		
8 -	s.NumDiscStates	= 0	:		
9 -	s.NumOutputs	= 2			
10 -	s.NumInputs	= 1	:		
11 -	s.DirFeedthrough	. = 0	;		
12 -	s.NumSampleTimes	= 1	;		
13 -	sys = simsizes(s)		2		
14 -	xO = [Cinit, Tinit]		;		
15 -	case 1			% derivatives	
16 -	Tj = u;sys = reactor	(t,x,7	Гј);		
17 -	case 3 % output				-
18 -	sys = x;		:		
19 -	case {2 4 9}		20	% 2:discrete,	
20				% 4:calcTimeHit,	
21				% 9:termination	
22 -	sys =[];		;		
23 -	otherwise				
24 -	error(['unhandled fl	ag =',	num2:	str(flag)]) ;	
25 -	end				*
-				reactor_sfcn Lr	1 24 Col 9 OVR
(2)				Bureau * FR 🗸	17.16

FIGURE 2 – File saved as reactor_sfcn.m

V Insert the S-Function block into the Simulink.

In the Simulink Library browser, go to the [User-Define Functions] subdirectory. Then drag-drop the S-Function block (see Figure 3). Double-click on the S-function block and fill in the parameters. Change the Sfunction name to reactor_sfcn. Also, fill in the parameters. In our case, we input 0.1,40 (which is the value for Cinit and Tinit) as shown in Figure 4.





VI Add other Simulink blocks and simulate

Remark : In figure 5, a demux block (demultiplexer) is included to split the output vector to the 2 elements. In other applications where the input vectors has more than one element, we need a mux block (multiplexer). Both mux and demux blocks reside in the Signal Routing subdirectory of the Simulink Library browser.

S-Func	ion
User-del must co passed l specified	nable block. Blocks may be written in M, C, Fortran or Ada and nform to S-function standards. t.x.u and flag are automatically o the S-function by Simulink. "Extra" parameters may be f in the "S-function parameters' field.
Parame S-funct	ters on name:
reacto	_sfcn
S-funct	on parameters:
0.1.40	
1010600	

Figure 4 –



Figure 5 -