Short Tutorial on Matlab

(©2004 by Tomas Co)

Part 6. Finding Steady States and Linearization via Simulink®

1. Consider the biochemical system,

(based on process developed in B. Wayne Bequette, "*Process Control: Modeling, Design and Simulation*", Prentice Hall, 2003, pp. 631-634.)

$$\frac{dx_1}{dt} = (\mu - D)x_1$$

$$\frac{dx_2}{dt} = D\left(x_{2f} - x_2\right) - \frac{\mu x_1}{Y}$$

$$\mu = \frac{\mu_{\max} x_2}{k_m + x_2 + k_1 x_2^{\,2}}$$

2. Build a Simulink model

```
function    dx = bioreactor(t,x,D,x2f)
%
%   model for bioreactor process
%   where,  x1 = biomass concentration
%           x2 = substrate concentration
%           D  = dilution rate
%           x2f= substrate feed

    x1 = x(1)                             ;
    x2 = x(2)                             ;

    mumax   = 0.53          ;   % hr^(-1)
    km      = 0.12          ;   % g/liter
    k1      = 0.454         ;   % liter/g
    Y       = 0.4           ;   %

    mu  = mumax*x2/(km+x2+k1*x2^2)      ;

    dx1 = (mu - D)*x1                     ;
    dx2 = D*(x2f-x2)-mu*x1/Y              ;

    dx = [dx1;dx2]                        ;
```

Figure 1. m-File for Bioprocess System

```
function  [sys,x0,str,ts]=        ...
      bioreactor_sfcn(t,x,u,flag, ...
      x1_init,x2_init)

switch flag

    case 0  % initialize

        str=[]                     ;
        ts = [0 0]                 ;

        s = simsizes               ;

            s.NumContStates  = 2   ;
            s.NumDiscStates  = 0   ;
            s.NumOutputs     = 2   ;
            s.NumInputs      = 2   ;
            s.DirFeedthrough = 0   ;
            s.NumSampleTimes = 1   ;

        sys = simsizes(s)          ;

        x0 = [x1_init, x2_init]    ;

    case 1  % derivatives

        D   = u(1)                 ;
        x2f = u(2)                 ;
        sys = bioreactor(t,x,D,x2f);

    case 3  % output

        sys = x                    ;

    case {2 4 9}    % 2:discrete,
                    % 4:calcTimeHit,
                    % 9:termination

        sys =[]                    ;

    otherwise
        error(...
           ['unhandled flag =',...
            num2str(flag)])        ;

end
```

Figure 2. Code for S-function

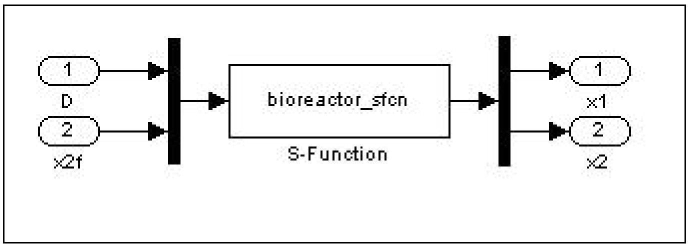

Figure 3. Simulink model: bioreactor_sys.mdl

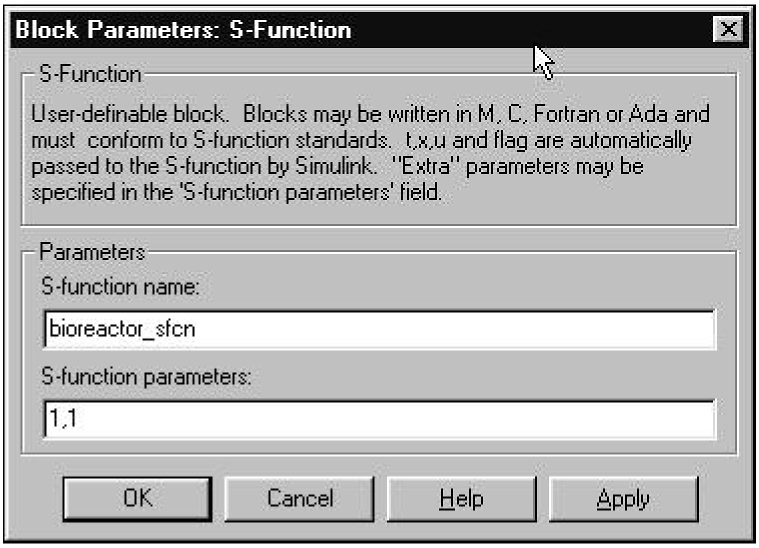

Figure 4. Set Parameters (in our case, initial conditions)

3. Use the **`trim`** function to get steady state.

   a) set some more parameters: i.e. we want to fix the input values of D=0.3 and x2f=4.0

```
>> X0=[];
>> U0=[0.3;4];
>> Y0=[];
>> IX=[];
>> IU=[1;2];
>> IY=[];
```

Remarks:

i) **`X0`**, **`U0`** and **`Y0`** are the state vector, input vector and output vector that will be fixed while the function attempts to find the steady state. Note that since we will not constrain the states and outputs, we can set these to null.

ii) **`IX, IU`** and **`IY`** indicates which values of **`X0, U0`** and **`Y0`** will be held fixed. In our case, we want both input vector to be fixed, so we set **`IU = [1;2]`**.

b) run the trim function:

```
>> [X,U,Y,DX]=TRIM('bioreactor_sys',X0,U0,Y0,IX,IU,IY);
>> X

X =

    0.9943
    1.5141
```

Thus, the function found X=(0.9943,1.5141) as the steady state.

4. Linearization via the **`linmod`** function.

```
>> [A,B,C,D]=linmod('bioreactor_sys',X,U0);
```

which results in the following:

```
A =
    0.0000   -0.0678
   -0.7500   -0.1304

B =
   -0.9943         0
    2.4859    0.3000

C =
    1.0000         0
         0    1.0000

D =
     0     0
     0     0
```