```
% <---- Main Program for CSTR Open Loop Dynamic Simulation ---->

clear all    % clear work space
clc
close all     %  Close all previously open figures

 % Initailize plotting style

 set(0,'DefaultLineLineWidth', 2)
 set(0,'DefaultaxesLineWidth', 2)
 set(0,'DefaultaxesFontSize', 14)
 set(0,'DefaultTextFontSize', 14)
 set(0,'DefaultaxesFontName', 'arial')

% <---- CSTR related global parameters ---->
% Global variables are shared variables between multiple files

global G_F G_Fc

 % <---- Initialization  of Variables ---->

 % <---- set sampling time & run time ---->

 T = 0.01 ;               % Integration interval = 0.1 min.
 run_count = 2001 ;   % Siumulation time count

 % Define steady state values operating point

 Fs   = 1.0  ;                    % Steady state Inlet reactant flow rate
 Fcs  = 15.0 ;                    % Steady state coolent flow
 Cas = 0.264572 ;                      % steady state concentration
 Ts = 393.952115 ;                   % Steady state reactortemperature
 Xs   = [ Cas Ts ]';       %  Steady state [ Conc. Temp ]
```

```
  % define initial state for dynamic simulation

  G_F = Fs ;                    % Initial Inlet reactant flow rate  (disturbance↙
variable)
  G_Fc = Fcs ;                  % Initail coolent flow  (manipulated input)
  Xp = Xs  ;                    % Initial state for dynamic simulations
  omega = [ 0.75 0.5 ] ;        % Frequency of oscillations introduced in inputs

  %  Variables to save dynamic simulation results

  res = [] ;
  res_dev = [] ;

 for k = 1 : run_count,

   % <----- Generation of inputs at t = k*T ------>

   G_F = Fs +  0.1 * sin(k*T*omega(1)) ;   % Sinusiodal change in reactor inlet↙
flow

   if ( k >  200 )     % Introduce sinusiodal change in Fc after 20 minutes
      G_Fc = Fcs + 2 * cos(k*T*omega(2)+(pi/4)) ;
   end

   % Save results at time t = k*T

   res = [ res ; (k-1)*T Xp' G_Fc G_F ] ;      % save absolute variables

   xp_dev = Xp - Xs ;
   Fc_dev = G_Fc - Fcs ;
   F_dev = G_F - Fs ;
   res_dev =   [ res_dev ; (k-1)*T xp_dev' Fc_dev F_dev ] ;   % save deviation↙
```

```
variables

   % <---- Process simulation from time t = k*T to  t = (k+1)*T
   %       Integrate the model equations for new inputs  ---->

       X0 = Xp ;                % Initial state for integration from t = k*T to (k+1)*T
       time = k*T ;
       dX_by_dt = cstr_dynamics( time, X0) ;
       Xp = X0 + T * dX_by_dt  ;                    % Euler integration to find Xp at t↙
= (k+1)*T

 end

 %  <---- Display simulation results
   figure(1),subplot(211), plot( res(:,1) , res(:,2), '-' ), grid ;
   xlabel('Time (min)'), ylabel('Reactor Conc. (mol/m^3)'), title( 'Process States↙
(Absolute)') ;
   figure(1),subplot(212), plot( res(:,1) , res(:,3), '-' ), grid ;
    xlabel('Time (min)'), ylabel('Reactor Temp. (K)') ;

   figure(2),subplot(211), stairs( res(:,1) , res(:,4) ), grid  ;
   xlabel('Time (min)'), ylabel('Coolent Flow'), title( 'Manipulated Input (Fc) and↙
Disturbance (F) (Absolute)') ;
   figure(2),subplot(212), stairs( res(:,1) , res(:,5) ), grid ;
   xlabel('Time (min)'), ylabel('Reactant Inflow (m^3/s)')

   %  <---- Display simulation results (Perturbation values) ---->

   figure(3),subplot(211), plot( res_dev(:,1) , res_dev(:,2), '-' ), grid ;
   xlabel('Time (min)'), ylabel('Reactor Conc. (mol/m^3)'), title( 'Process States↙
(Deviation)') ;
   figure(3),subplot(212), plot( res_dev(:,1) , res_dev(:,3), '-' ), grid ;
    xlabel('Time (min)'), ylabel('Reactor Temp. (K)') ;
```

```
  figure(4),subplot(211), stairs( res_dev(:,1) , res_dev(:,4) ), grid  ;
  xlabel('Time (min)'), ylabel('Coolent Flow'), title( 'Manipylated Input (Fc) and↙
Disturbance (F) (Deviation)') ;
  figure(4),subplot(212), stairs( res_dev(:,1) , res_dev(:,5) ), grid ;
  xlabel('Time (min)'), ylabel('Reactant Inflow (m^3/s)')

  % <---- End of Main Program ---->

% ------------------------------------------------------------------
% CSTR Problem dynamic model
% The derivatives XDOT = F( X, U ) are calculated and returned
% to the integrator
% ------------------------------------------------------------------
  function [xdot] = cstr_dynamics( t ,X )

% <---- CSTR related global parameters ---->

  global G_F  G_Fc

  Ko = 1.0e10 ;              % ( min-1 )  : Reaction rate parameter
  E  = 8330.1 ;              % ( oK ) : Reaction rate parameter
  V  = 1.0    ;             % ( m3 )  : Reactor volume
  Cp = 1.0    ;             % ( cal / g K ) : Cp value
  rho = 1e6   ;              % ( g / m3 ) : density
  delH = 1.3e8 ;             % ( cal / kmol )  : Heat of reaction
  Cpc  = 1.0   ;            % ( cal / g K ) :  Cp of coolent
  rhoc = 1e6   ;             % ( g / m3 ) : density of coolent
  a = 1.678e6  ;             % ( cal / min K ) : heat transfer eqn parameter
  b  = 0.5     ;            % constant : : heat transfer eqn parameter
  To  = 323.0 ;             % Reactant Inlet temperature
  Tcin = 365;               % K
  Cao = 2.0  ;               % Steady state Inlet concentration
```

```
% Temporary variable used to simplicity of writing ODEs

Ca  = X(1) ;     % Reactor concentration
T   = X(2) ;     %  Reactor temperature

K  = Ko * exp( - E / T ) ;

Gc = a/ ( V * rho * Cp ) ;

B1 = a* G_Fc^b / ( 2 * rhoc * Cpc ) ;
B  = G_Fc^( b+1 ) / ( G_Fc + B1 ) ;

ratio  = delH / ( rho * Cp ) ;

%  expresions for two derivatives

dCa_by_dt = G_F * ( Cao - Ca ) / V - K * Ca ;     % Rate of change of ↙
concentration

dT_by_dt  = G_F * ( To - T ) / V ;              %  rate of change of temperature
dT_by_dt  = dT_by_dt - Gc * B * ( T - Tcin ) ;
dT_by_dt  = dT_by_dt + ratio * K * Ca ;

% Put concentration and temperature derivatives into derivative vector

xdot = [ dCa_by_dt   dT_by_dt ]';
```