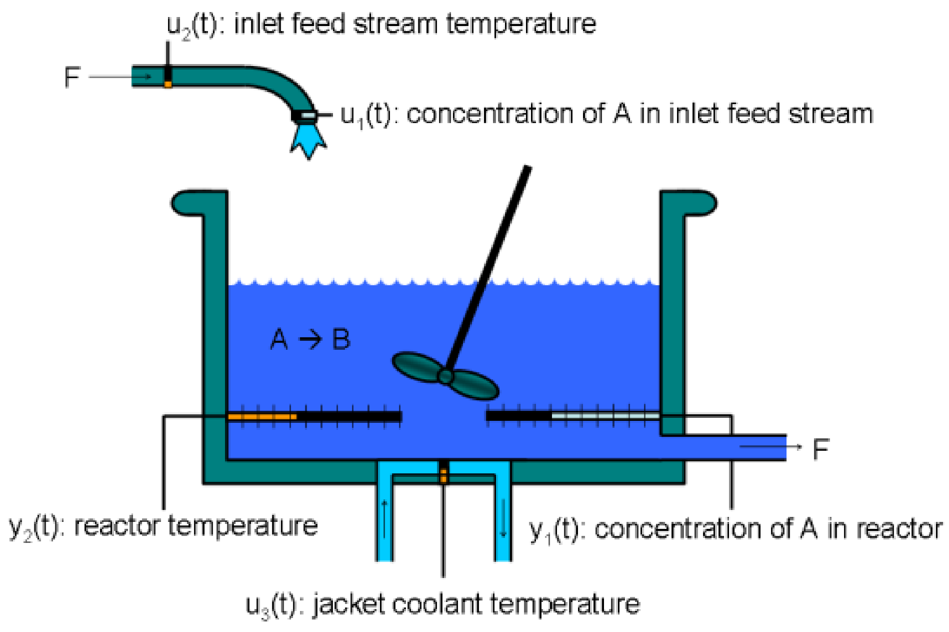
Problem Description


u_2(t): inlet feed stream temperature
F
u_1(t): concentration of A in inlet feed stream
A → B
F
y_2(t): reactor temperature
y_1(t): concentration of A in reactor
u_3(t): jacket coolant temperature

The CSTR system is modeled using basic accounting and energy conservation principles. The change of the concentration of reagent A in the vessel per time unit d C_A(t)/dt (= d y_1(t)/dt) can be modeled as:

```
d C_A(t)

-------- = F/V*(C_Af(t)-C_A(t)) - r(t)

dt
```

where the first term expresses concentration changes due to differences between the concentration of reagent A in the inlet stream and in the vessel, and the second term expresses concentration changes (reaction rate) that occurs due to the chemical reaction in the vessel. The reaction rate per unit volume is described by Arrhenius rate law:

```
r(t) = k_0*exp(-E/(R*T(t)))*C_A(t)
```

which states that the rate of a chemical reaction increases exponentially with the absolute temperature. k_0 is here an unknown nonthermal constant, E is the activation energy, R Boltzmann's ideal gas constant and T(t) (= y_2(t)) the temperature in the reactor.

Similarly, using the energy balance principle (assuming constant volume in the reactor), the temperature change per time unit d T(t)/dt in the reactor can be modeled as:

```
d T(t)

------ = F/V(T_f(t)-T(t)) - (H/c_p*rho)*r(t) - (U*A)/(c_p*rho*V)*(T(t)-T_j(t))

dt
```

where the first and third terms describe changes due to that the feed stream temperature T_f(t) and the jacket coolant temperature T_j(t) differ from the reactor temperature, respectively. The second term is the influence on the reactor temperature caused by the chemical reaction in the vessel. In this equation, H is a heat of reaction parameter, c_p a heat capacity term, rho a density term, U an overall heat transfer coefficient and A the area for the heat exchange (coolant/vessel area).

Put together, the CSTR has three input signals:

```
u_1(t) = C_Af(t)  Concentration of A in inlet feed stream [kgmol/m^3].

u_2(t) = T_f(t)   Inlet feed stream temperature [K].

u_3(t) = T_j(t)   Jacket coolant temperature [K].
```

and two output signals:

```
y_1(t) = C_A(t)   Concentration of A in reactor tank [kgmol/m^3].

y_2(t) = T(t)     Reactor temperature [K]
```

Import Casadi

```
clc, clear all
addpath('C:\dev\casadi-windows-matlabR2016a-v3.5.2');
%addpath('\\home.org.aalto.fi\sliczno1\data\Documents\casadi-windows-matlabR2016a-v3.5.1');
import casadi.*
```

Parameters

```
p(1)   = 1;             % Volumetric flow rate (volume/time) [m^3/h]
p(2)   = 1;             % Volume in reactor [m^3]
p(3)   = 3.5e+07;       % Pre-exponential nonthermal factor [1/h]
p(4)   = 11850;         % Activation energy [kcal/kgmol]
p(5)   = 1.98589;       % Boltzmann's ideal gas constant [kcal/(kgmo..]
p(6)   = -5960;         % Heat of reaction [kcal/kgmol]
p(7)   = 480;           % Heat capacity times density [kcal/(m^3*K)]
p(8)   = 145;           % Overall heat transfer coefficient times tank area [kcal/(K*h)]
```

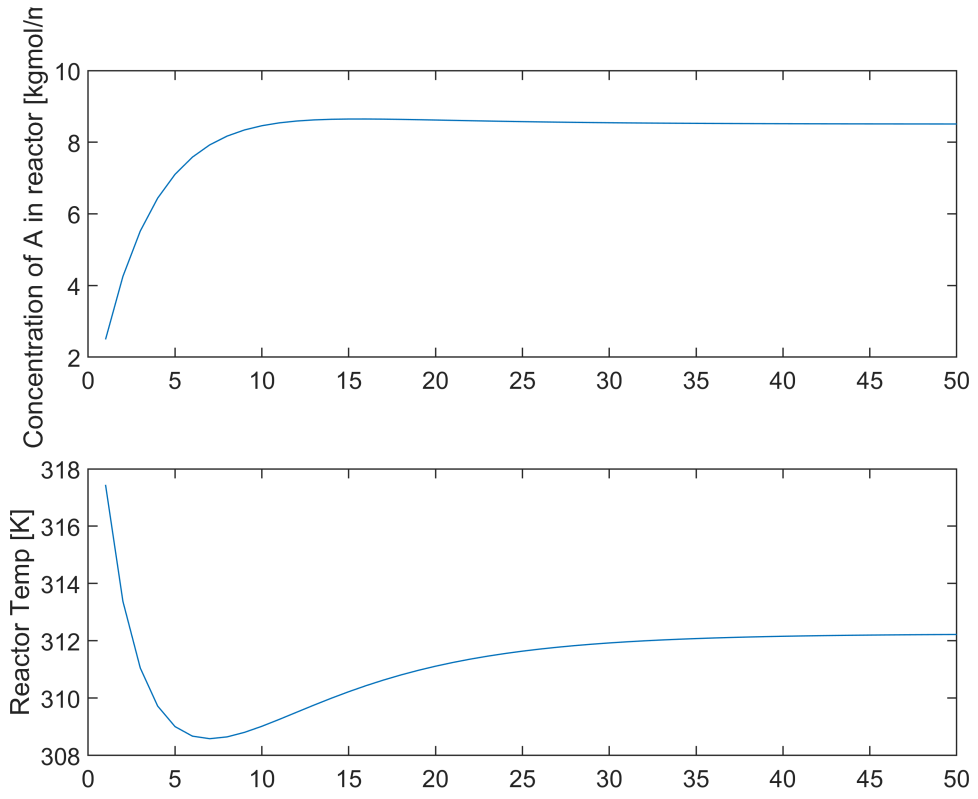
Model and Integrator

```
f = @(x,u) cstr_m(0, x, u, p);
g = @(x)   modelOUT(0, x, p);

%% Build integrator

%Variables
Nx = 2;
Nu = 3;
Ny = 1;

% Time variables
simulationTime       = 1/4;                                          % Minutes
```

```
timeStep              = 1/200;                                       % Minutes

timeStep_in_seconds  = timeStep * 60;                                % Seconds
simulationTime_in_seconds = simulationTime * 60;                     % Seconds
nSteps    = ceil(simulationTime_in_seconds / timeStep_in_seconds);   %

% Integrator
F = buildIntegrator(f, [Nx,Nu] , timeStep_in_seconds);
```

Simulation without control

```

InitialStates = [0 325];                 % Initial value of the initial states.

feedConc    = 10  * ones(1,nSteps);   % kgmol/m^3
feedTemp    = 298 * ones(1,nSteps);   % Kelvin
jacketTemp  = 298 * ones(1,nSteps);   % Kelvin

u = [feedConc', feedTemp', jacketTemp'];

x0 = [InitialStates(1)*ones(1,1);
      InitialStates(2)*ones(1,1)];

[yout,~,xout] = simulateSystem(F, g, x0, u);

subplot(2,1,1)
plot(yout(1,:)')
ylabel('Concentration of A in reactor [kgmol/m^3]')
subplot(2,1,2)
plot(xout(2,:)')
ylabel('Reactor Temp [K]')
```



Simulation with control

$$\min_{u[t_i,\, t_f]} \quad J := \frac{1}{2}\Delta z^T(t_i) P\, \Delta z(t_i) + \frac{1}{2}\int_{t_i}^{t_f} u^T(t) R u(t)\, dt$$

$$\text{s.t.} \quad \dot{x}(t) = F(x(t), u(t)), \quad x(t_i) = \hat{x}(t_i),$$

$$z_{\text{pred}}(t_i) = Z(x[0, t_f], u[0, t_f]),$$

$$x(t_f) \in \mathcal{X},$$

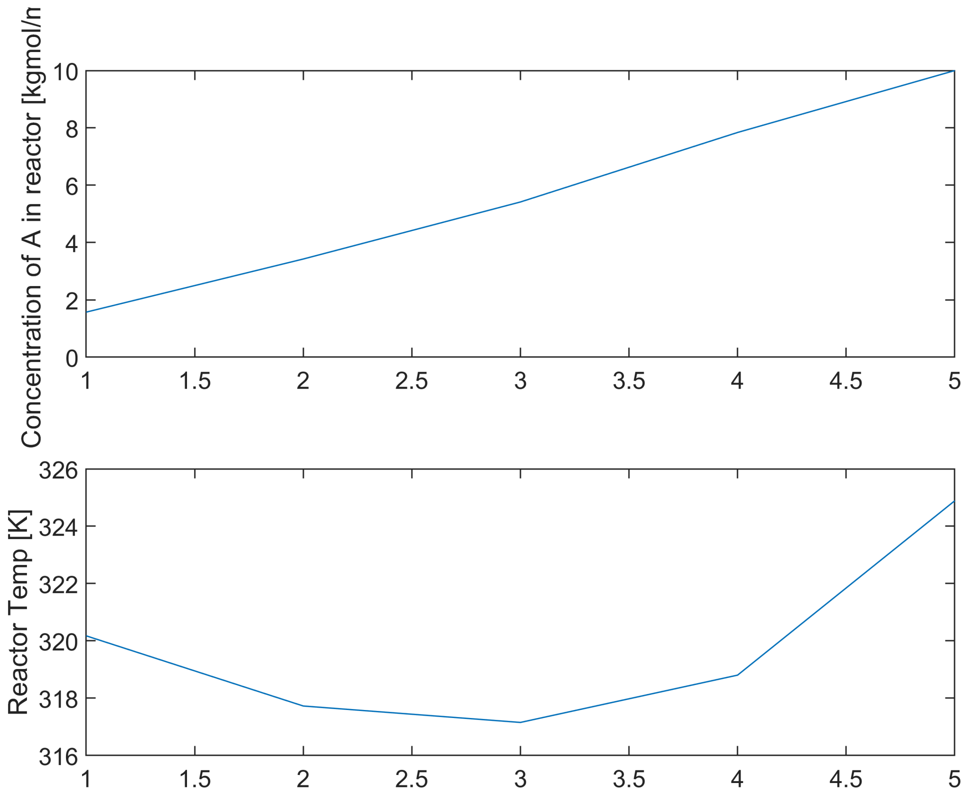
```
target = 10;

OCP = struct('Nx', Nx, 'Nu', Nu, 'Ny',Ny, 'N', 5, ...
    'x_lu', [0*ones(Nx,1) inf*ones(Nx,1)],  ...
    'u_lu', [1*[0; 300; 273] [20; 400; 400]], ...
    'x_eq',[], ...
    'u_eq',[], ...
    'F',F, ...
    'L',@(u) 0.5*(u)'*diag([1E-16, 1E-16, 1E-16])*(u), ...
    'Lf',@(x,yd) 0.5*(g(x)-yd)'*diag([1])*(g(x)-yd) );

[xout, uout]  = singleShooting(OCP, x0, target);
```

```
This is Ipopt version 3.12.3, running with linear solver mumps.
NOTE: Other linear solvers might be more efficient (see Ipopt documentation).

Number of nonzeros in equality constraint Jacobian...:        0
Number of nonzeros in inequality constraint Jacobian.:       15
Number of nonzeros in Lagrangian Hessian.............:      120

Total number of variables............................:       15
                     variables with only lower bounds:        0
                variables with lower and upper bounds:        0
                     variables with only upper bounds:        0
Total number of equality constraints.................:        0
Total number of inequality constraints...............:       15
        inequality constraints with only lower bounds:        0
   inequality constraints with lower and upper bounds:       15
        inequality constraints with only upper bounds:        0

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0 5.0000000e+001 0.00e+000 1.28e+000  -1.0 0.00e+000    -  0.00e+000 0.00e+000   0
   1 4.9085503e+001 0.00e+000 1.82e+000  -1.0 1.10e+000    -  2.82e-001 1.00e+000f  1
   2 1.2453272e+001 0.00e+000 5.48e-001  -1.0 1.82e+001    -  1.61e-002 1.00e+000f  1
   3 7.5465508e+000 0.00e+000 2.12e-001  -1.0 3.48e+000    -  2.70e-001 8.18e-001f  1
   4 1.6268750e+000 0.00e+000 1.69e-001  -1.0 9.58e+000    -  4.76e-001 1.00e+000f  1
   5 2.5235049e-001 0.00e+000 9.89e-002  -1.0 5.24e+000    -  7.79e-001 1.00e+000f  1
   6 1.4159845e-002 0.00e+000 1.27e-002  -1.0 4.89e+000    -  1.00e+000 1.00e+000f  1
   7 6.4603580e-003 0.00e+000 1.53e-002  -1.7 5.74e+000    -  1.00e+000 1.00e+000f  1
   8 8.7701590e-005 0.00e+000 1.18e-003  -2.5 1.59e+000    -  1.00e+000 1.00e+000f  1
   9 3.6186639e-007 0.00e+000 1.14e-004  -3.8 4.48e-001    -  1.00e+000 1.00e+000f  1
iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
  10 3.3829282e-010 0.00e+000 4.79e-006  -5.7 1.18e-001    -  1.00e+000 1.00e+000f  1
  11 1.0223463e-010 0.00e+000 2.57e-006  -8.6 5.03e-001    -  9.76e-001 1.00e+000f  1
  12 4.6835994e-011 0.00e+000 2.87e-009  -8.6 2.87e-005  -4.0 1.00e+000 1.00e+000f  1

Number of Iterations....: 12

                                   (scaled)                 (unscaled)
Objective...............:  4.6835994126536414e-011   4.6835994126536414e-011
Dual infeasibility......:  2.8732214730545500e-009   2.8732214730545500e-009
Constraint violation....:  0.0000000000000000e+000   0.0000000000000000e+000
Complementarity.........:  2.5070133821822639e-009   2.5070133821822639e-009
Overall NLP error.......:  2.8732214730545500e-009   2.8732214730545500e-009


Number of objective function evaluations             = 13
Number of objective gradient evaluations             = 13
Number of equality constraint evaluations            = 0
Number of inequality constraint evaluations          = 13
Number of equality constraint Jacobian evaluations   = 0
Number of inequality constraint Jacobian evaluations = 13
Number of Lagrangian Hessian evaluations             = 12
Total CPU secs in IPOPT (w/o function evaluations)   =      0.179
Total CPU secs in NLP function evaluations           =      0.318

EXIT: Optimal Solution Found.
      solver  :   t_proc      (avg)   t_wall      (avg)    n_eval
       nlp_f  |   5.00ms (384.62us)   4.99ms (384.23us)        13
       nlp_g  |        0 (       0)        0 (       0)        13
  nlp_grad_f  |  31.00ms (  2.21ms)  30.04ms (  2.15ms)        14
  nlp_hess_l  | 283.00ms ( 23.58ms) 283.98ms ( 23.67ms)        12
   nlp_jac_g  |        0 (       0)        0 (       0)        14
       total  | 501.00ms (501.00ms) 500.05ms (500.05ms)         1
```

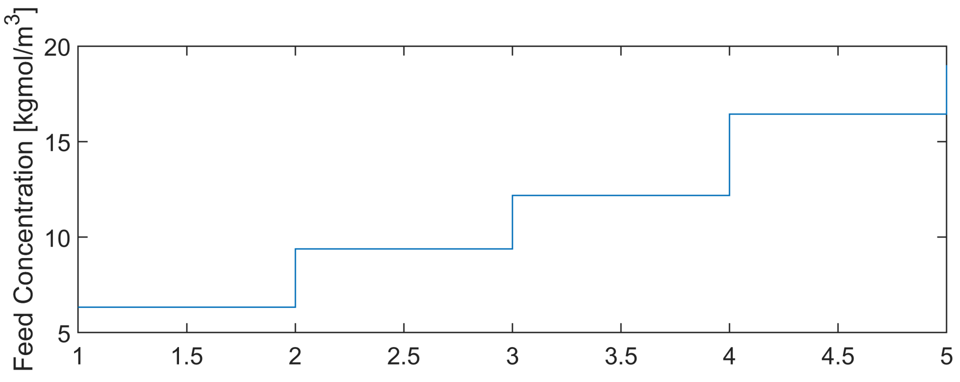
```
[yopt,~,xopt] = simulateSystem(F, g, x0, uout');
```

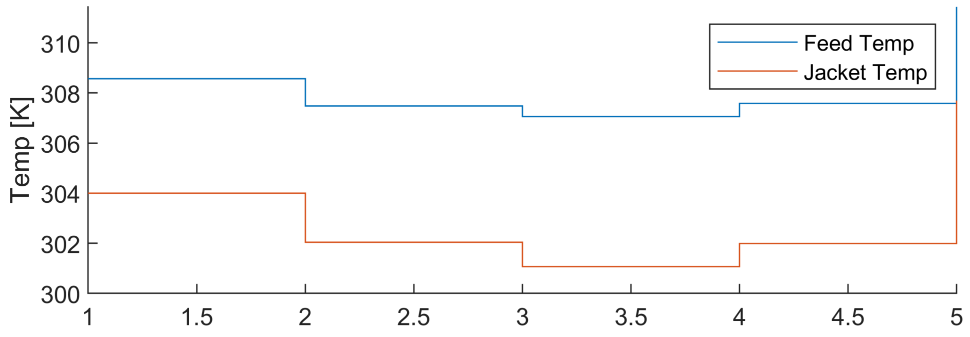
Plotting

```
subplot(2,1,1)
plot(xopt(1,:)')
ylabel('Concentration of A in reactor [kgmol/m^3]')
subplot(2,1,2)
plot(xopt(2,:)')
ylabel('Reactor Temp [K]')
```



```

figure()
subplot(2,1,1)
stairs(uout(1,:)')
ylabel('Feed Concentration [kgmol/m^3]')
subplot(2,1,2)
hold on
stairs(uout(2,:)'); stairs(uout(3,:)')
legend('Feed Temp','Jacket Temp');
ylabel('Temp [K]');  hold off
```


Feed Concentration [kgmol/m^3]
20
15
10
5
1
1.5
2
2.5
3
3.5
4
4.5
5


Feed Temp
Jacket Temp
Temp [K]
310
308
306
304
302
300
1
1.5
2
2.5
3
3.5
4
4.5
5