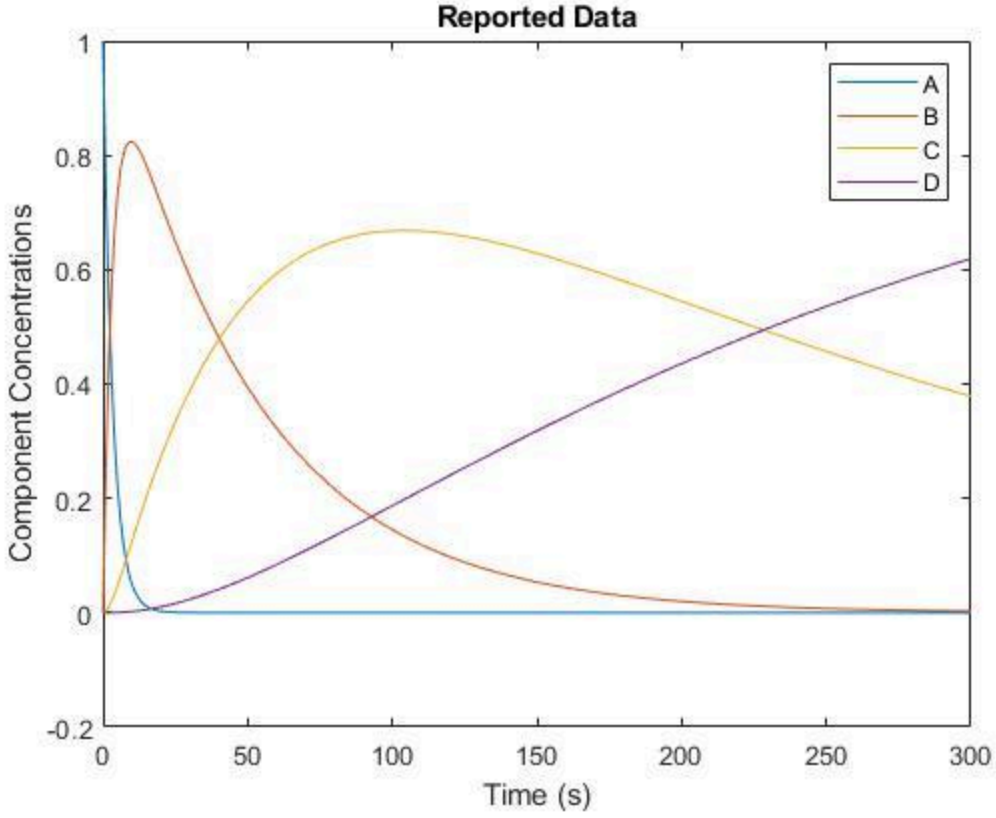
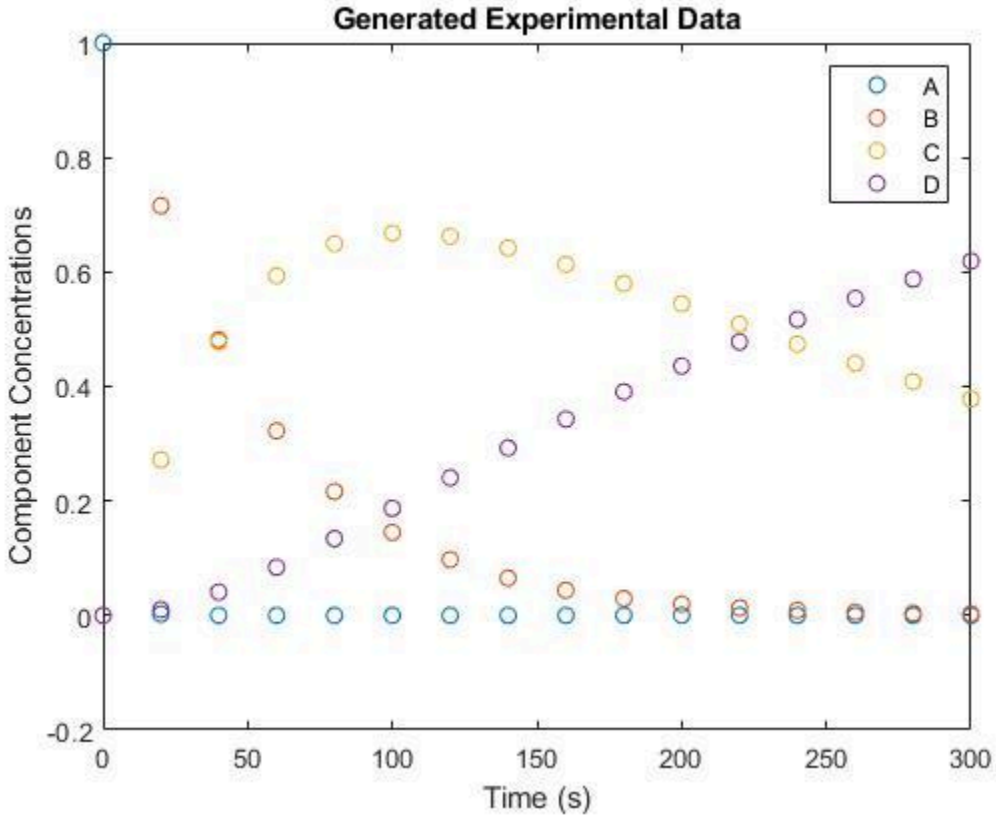
```
clear; clc;
%Plotting some data from the Paper
tspan = 0:300;
y0 = [1;0;0;0];
k0 = [0.3;0.02;0.0040];
[t,y] = ode15s(@(t,y) ODE_system_x(k0,y), tspan,y0);
%ax1 = subplot(3,3,2);
figure(1)
plot(t,y)
title('Reported Data')
xlabel('Time (s)')
ylabel('Component Concentrations')
legend('A','B','C','D')

%Preparing the experimental Data:
    %We prepare the experimental data as the paper reports nothing
 other
    %than the specific rates.
    %We pick 15 points
    %This will work for a timespan of 300 only
l = 1;
z = zeros(100,1);
while l <= length(y)
    z(l,1) = y(l,1);
    z(l,2) = y(l,2);
    z(l,3) = y(l,3);
    z(l,4) = y(l,4);
    l = l+20;
end
l = 1;
i = 1;
z0 = zeros(15,4);
xa = zeros(15,1);
while l <= length(z)
    if z(l)~= 0
        z0(i,1) = z(l,1);
        z0(i,2) = z(l,2);
        z0(i,3) = z(l,3);
        z0(i,4) = z(l,4);
        xa(i) = t(l);
        i = i+1;
    end
    l = l+1;
end

%Begin with x = 0
%use increment x if z ! = 0
%store that value of z in vairable z'
%plot(x0,y0);
% hold on

%ax2 = subplot(3,3,5);
```

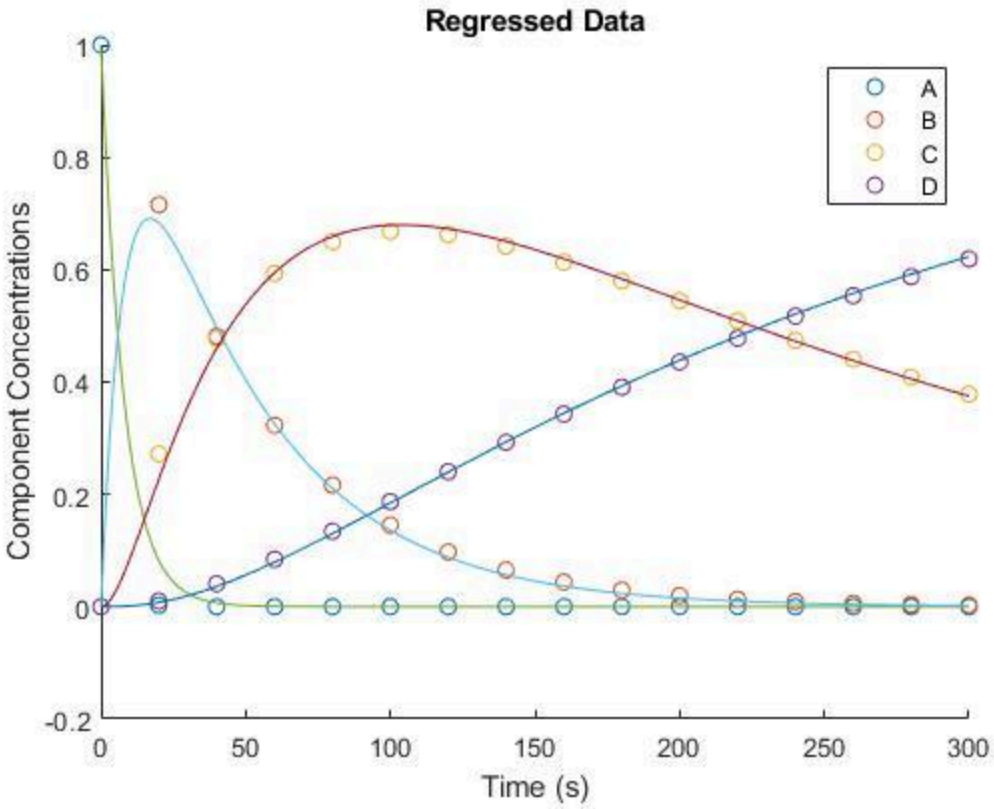
```
figure(2)
%hold on
plot(xa,z0,'o');
%plot(xa,z0(:,3),'o');
title('Generated Experimental Data')
ylabel('Component Concentrations')
xlabel('Time (s)')
legend('A','B','C','D');
%hold off

%Now we have experimental data. for A and B

%Splining the data
ya = spline(xa,z0(:,1),t);
yb = spline(xa,z0(:,2),t);
yc = spline(xa,z0(:,3),t);
yd = spline(xa,z0(:,4),t);
yy = [ya, yb, yc, yd];

%Next Step, take guess values of K to fit the initial data
lb = zeros(3,1);
ub = ones(3,1);
k0_2 = [0.5;0.5;0.5];
opt_k = lsqnonlin(@(k) difference_func(k,yy), k0_2, lb, ub);
[tf,yf] = ode15s(@(t,y)ODE_system_x(opt_k, y), tspan, y0);
figure(3)
%ax3 = subplot(3,3,8);
hold on;
plot(xa,z0,'o');
%plot(xa,z0(:,3),'o');
plot(tf,yf,'-');
%plot(tf,yf(:,4),'-');
title('Regressed Data');
ylabel('Component Concentrations')
xlabel('Time (s)')
legend('A','B','C','D');
hold off

function out_sys = ODE_system_x(k,y)
r = zeros(3,1);
out_sys = zeros(4,1);
%out_sys = d[y1]/dt
% y(1) = OH.
% y(1) = A
% y(2) = B
% y(3) = C
% y(4) = D
% k1 - B forms
% k2 - C forms
% k3 - D forms

%system of reactions:
r(1)= -k(1)*y(1);
r(2)= -k(2)*y(2);
```

```
r(3)= -k(3)*y(3);

%system of concentrations
out_sys(1) = r(1);
%out_sys(2) = r(1);
out_sys(2) = r(2) - r(1);
out_sys(3) = r(3) - r(2);
out_sys(4) = -r(3);
end
function diff_func = difference_func(k,y_exp)
%This function takes in the experimental concentration values and
 compares
%them with the theoretically generated values, for changing k matrix.

%What we get in the end is known as a difference matrix, which is to
 be
%optimized.

%Using the same timespan and initial concentrations
tspan = 0:300;
y0 = [1;0;0;0];
[t,y] = ode15s(@(t,y) ODE_system_x(k,y),tspan,y0);
diff_func = (abs(y_exp(:,1)-y(:,1)))+(abs(y_exp(:,2)-
y(:,2)))+(abs(y_exp(:,3)-y(:,3)))+(abs(y_exp(:,4)-y(:,4)));
end
```

```
Local minimum possible.

lsqnonlin stopped because the final change in the sum of squares
 relative to
its initial value is less than the value of the function tolerance.
```


Reported Data
Component Concentrations
Time (s)
A
B
C
D
1
0.8
0.6
0.4
0.2
0
-0.2
0
50
100
150
200
250
300


Generated Experimental Data
Component Concentrations
Time (s)
A
B
C
D
1
0.8
0.6
0.4
0.2
0
-0.2
0
50
100
150
200
250
300



Published with MATLAB® R2019a