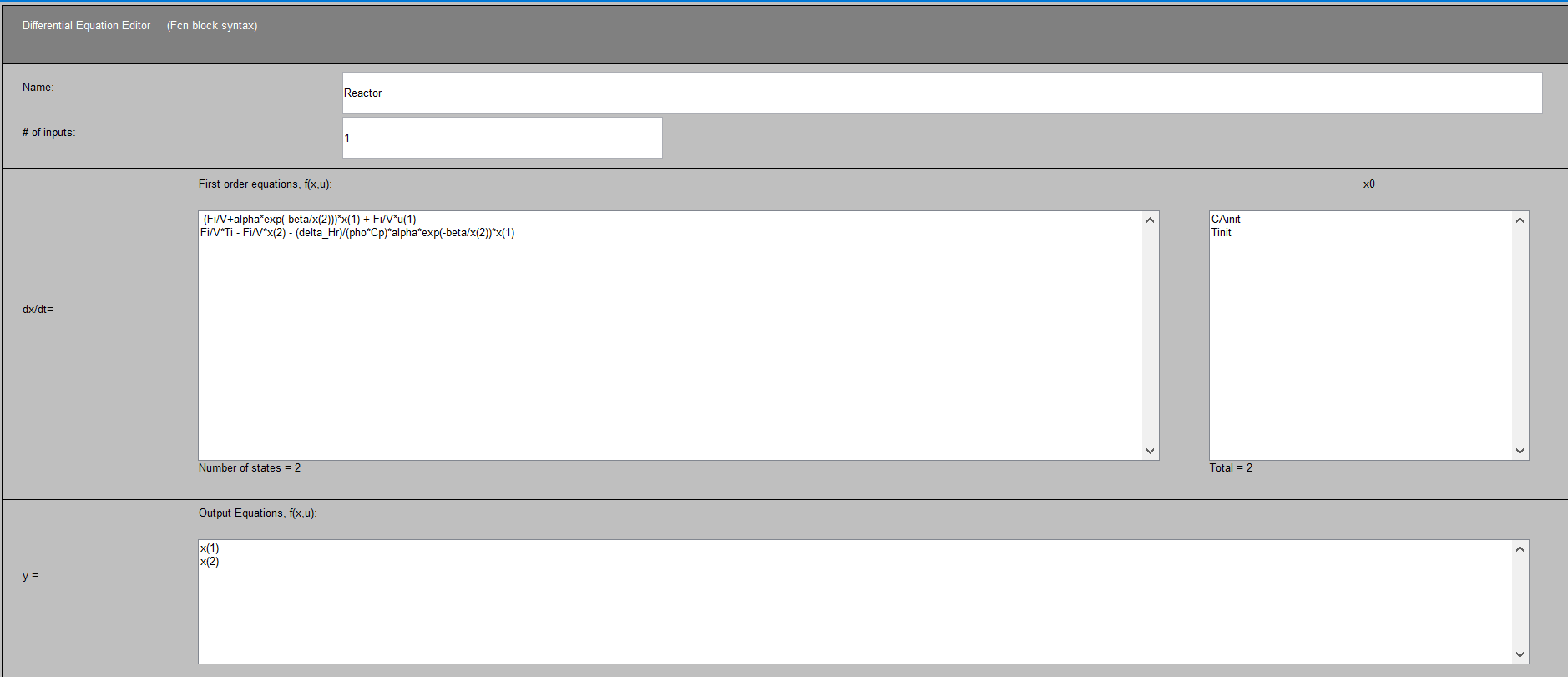
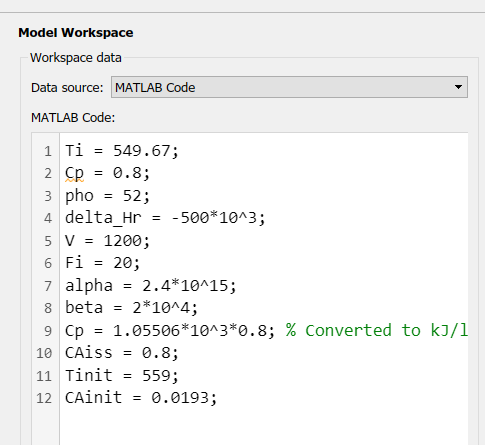
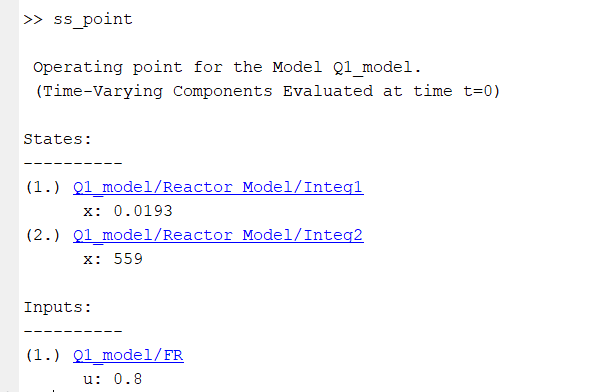
# Question 1 b) Designing Simulink model & Steady-state









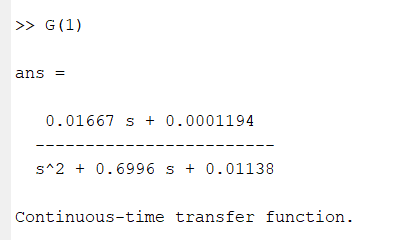
**Steady State** values obtained using **findop**:

**CA,ss = 0.0193 lb/ft3**

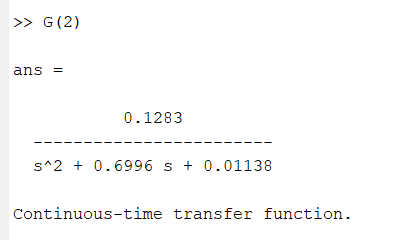
**TSS = 558.564 Rankine = 98.894 Fahrenheit**

# Question 1 c) Transfer function form

## From cAi to cA



## From cAi to Ti



Verified using hand calculations in hand written part

# Question 1d): Step response for 10% step change in CAi

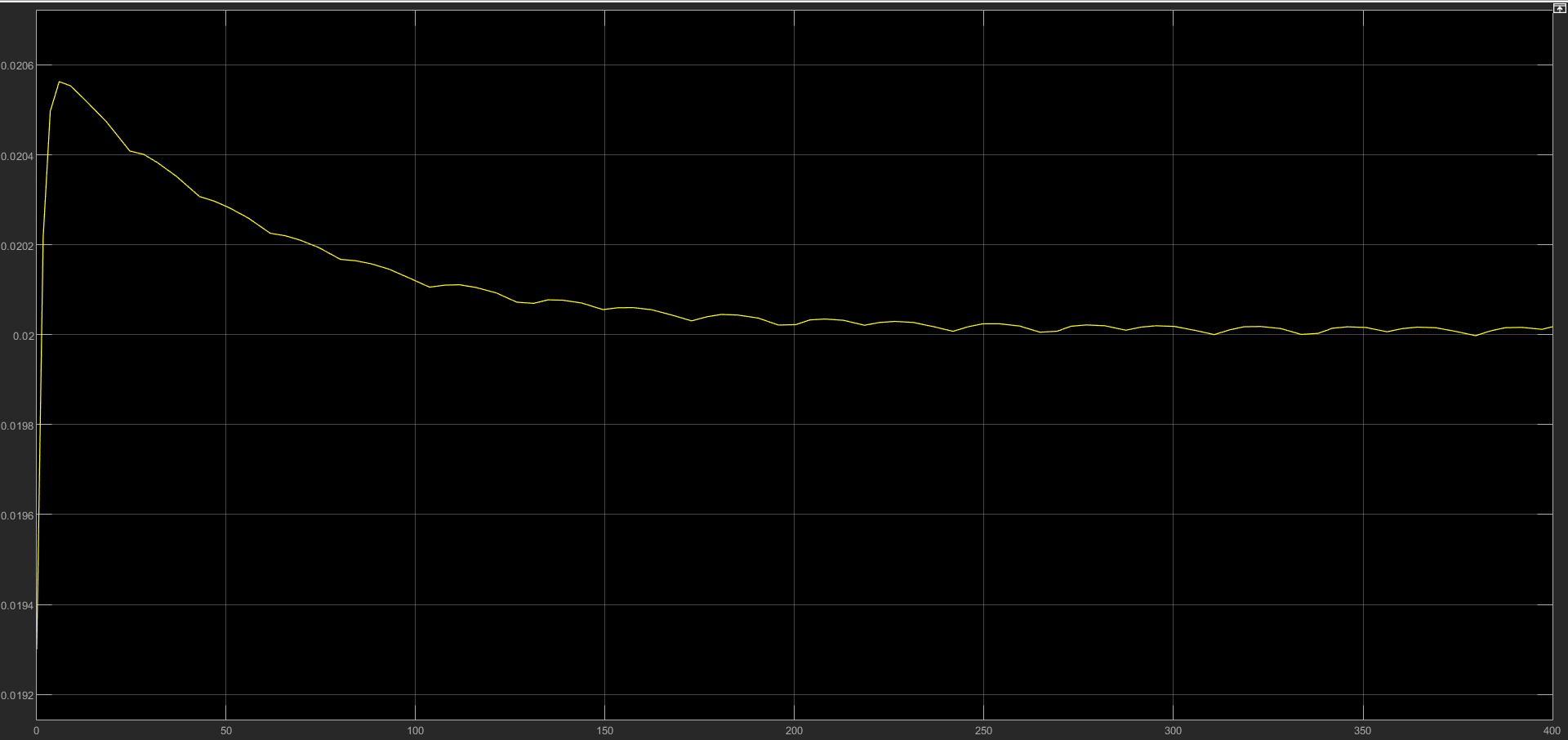


Figure 1: CA response (non linear model)

**CA,ss = 0.0200** (obtained from out.yout)

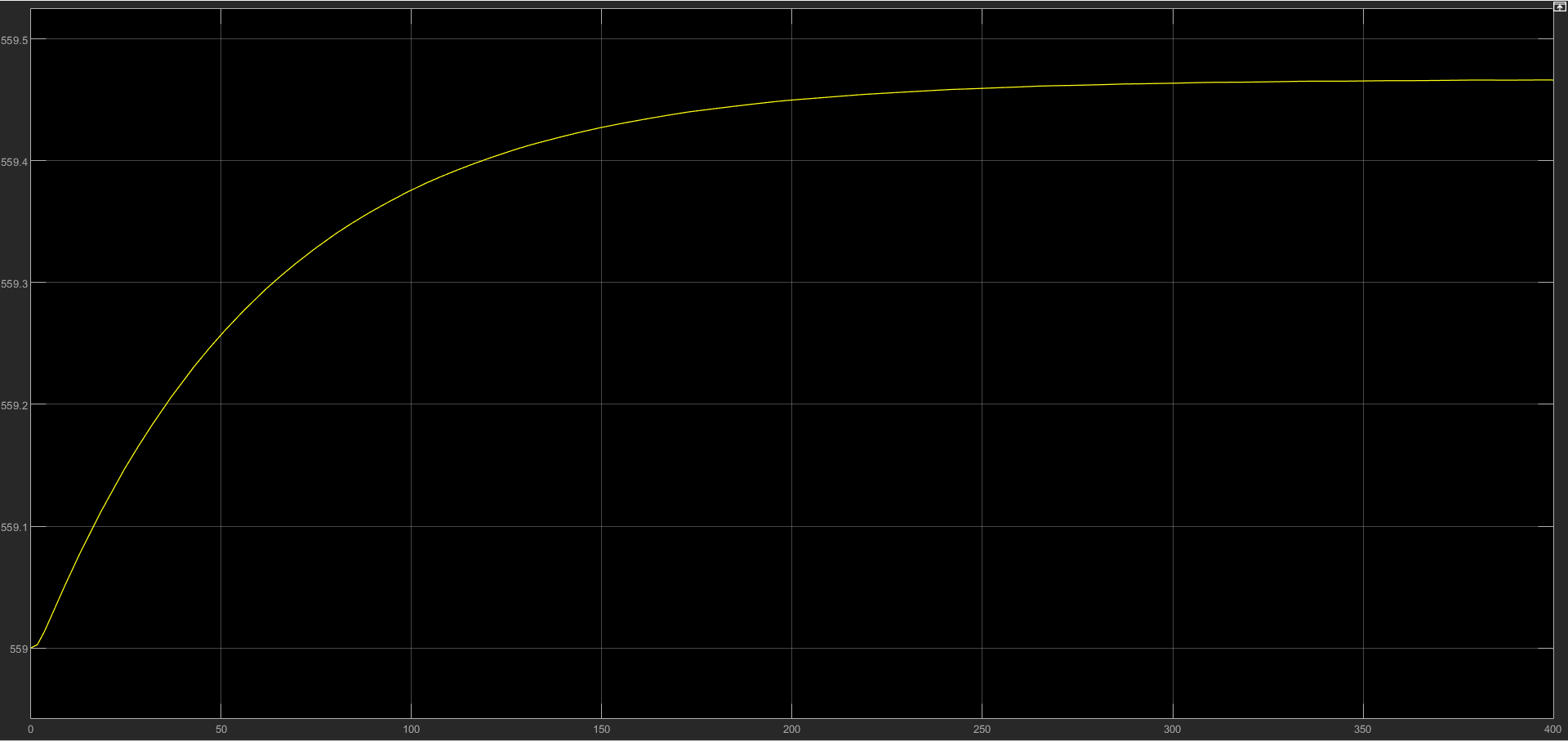


Figure 2: T response (non-linear model)

**Tss = 559.4663 Rankine = 99.7963 Fahrenheit**

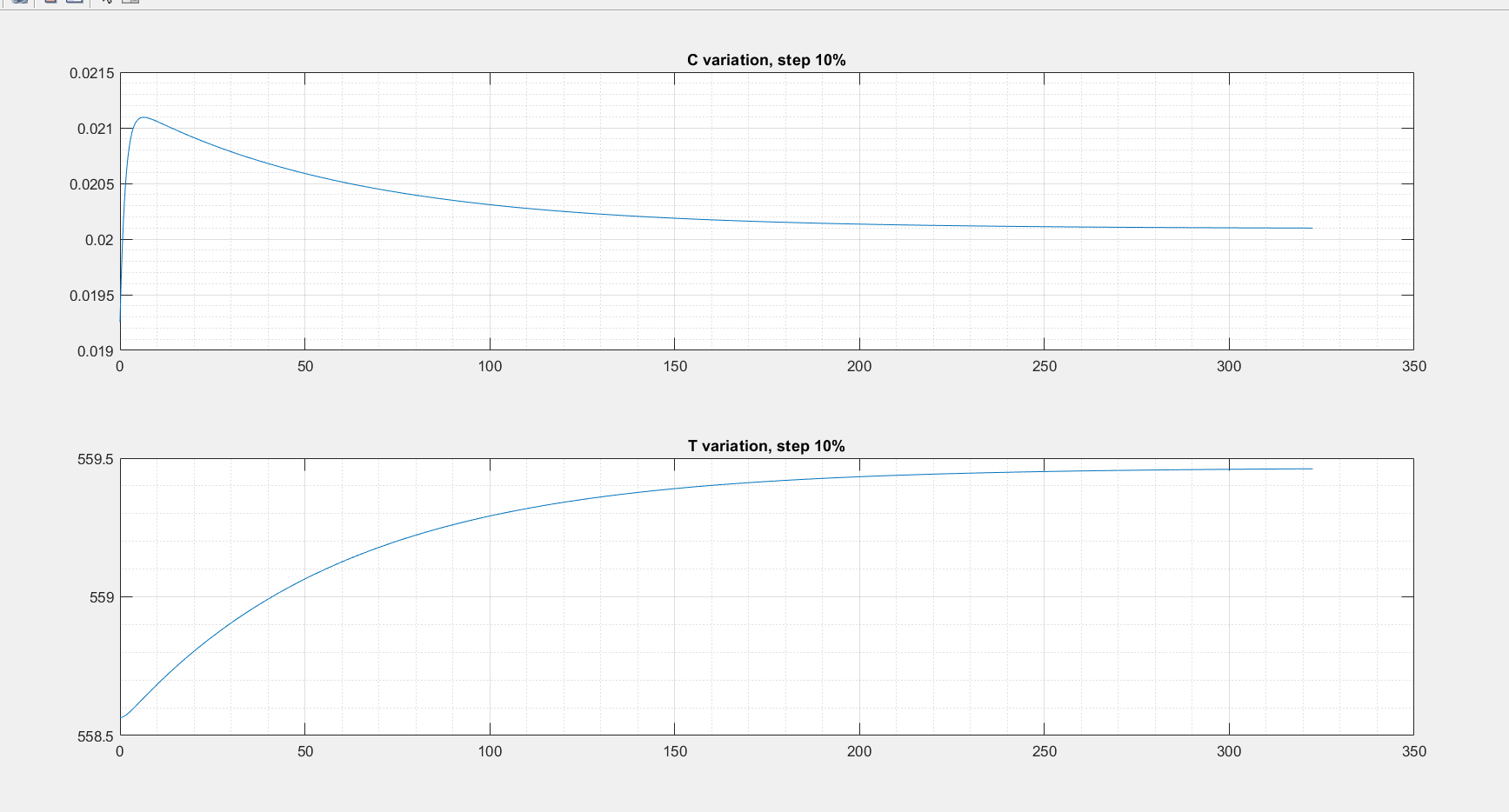


Figure 3: Response from linear models

**CA,ss = 0.0201** and **Tss = 559.462 Rankine = 99.792 Fahrenheit**

% error in **CA,ss**: 0.5 % and % error in **Tss**: 0.004%. We see that the errors are not huge. So for computational simplicity we can adopt a linear model (provided input conditions aren’t altered much)

# Question 1 e)

Gain\_C = (0.2-0.0193)/0.08 = 2.2588

Gain\_T = (559.4663-558.564)/0.08 = 11.278

From the gains we can see that for a unit change in input variable, **Temperature** of reactor is affected more than the **concentration of A** in the reactor.

# MATLAB Code

clear; close all;

%% Part b) Find steady-state and linearise

open\_system('Q1\_model')

% Read the operating conditions into an object

opc = operspec('Q1\_model');

% Operating conditions

opc.Inputs.u = 0.8;

opc.Inputs.Known = 1;

% Constraints

%opc.States(1).Min = 0;opc.States(2).Min = 0;

%opc.States(1).Max = 0.8;

% Find the steady state point

ss\_point = findop('Q1\_model',opc);

% Linearize

linsys = linearize('Q1\_model',ss\_point); %Using lin mod: linmod('Q3\_model',x\_ss,[80 100])

[NUM, DEN] = ss2tf(linsys.A,linsys.B,linsys.C,linsys.D);

NUM = {NUM(1,:) NUM(2,:)};

G = tf(NUM,DEN);

%% Hand calculations

Css=ss\_point.States(1).x;

Tss=ss\_point.States(2).x;

Ti = 549.67;

Cp = 0.8;

pho = 52;

delta\_Hr = -500\*10^3;

V = 1200;

Fi = 20;

alpha = 2.4\*10^15;

beta = 2\*10^4;

Cp = 1.05506\*10^3\*0.8; % Converted to kJ/lb

CAiss = 0.8;

Tinit = 559;

CAinit = 0.0193;

A = zeros(2);

A(1,:) = [-(Fi/V + alpha\*exp(-beta/Tss)) -alpha\*exp(-beta/Tss)\*beta/Tss^2\*Css];

A(2,:) = [-delta\_Hr/(pho\*Cp)\*alpha\*exp(-beta/Tss) -delta\_Hr/(pho\*Cp)\*alpha\*exp(-beta/Tss)\*beta/Tss^2\*Css-Fi/V];

B = [Fi/V;0];

C = eye(2);

%% Part d): Computing response

% Since linear system, changes in input and output are proportional

[Y,T,X]=step(linsys);

figure();

subplot(2,1,1);plot(T,Y(:,1)\*0.1\*0.8+Css); title('C variation, step 10%');

grid on; grid minor;

subplot(2,1,2);plot(T,Y(:,2)\*0.1\*0.8+Tss); title('T variation, step 10%');

grid on; grid minor;

%% Part e): Comparing gains

Gain\_T = 0.4663/0.08;

Gain\_C = (0.2-0.0193)/0.08;

# Simulink model:

