EGR102

Final Project Part 5/EC:

*Part 5: Fit Actual Data with All Growth Kinetic Parameters*

*Extra Credit (15 pts): Use Best Parameters Found and then Optimize Production using Best Fin*

Brian Aufderheide

Due: Monday at Midnight May 6th, 2024

The assignment is to simulate a BIOREACTOR WITH SUBSTRATE INHIBITION.

You will be building off this system to

1. Simulate the continuous bioreactor for a time frame with fixed inputs.
2. Simulate the continuous bioreactor with a step response in flow rate.
3. Simulate actual data vs predicted data and calculate mean square error.
4. Fit baseline maximum growth rate, µmax (mu\_max), given set of data.
5. Fit growth kinetics for max growth rate µmax, the Monod Constant, K, and inhibition factor Ki.

EC) Optimize Fin for best production.

Modeling Equations:

Note: growth rate µ(mu) should be in the ODE function file.

% Simulation Parameters

*Obtain from data file: t\_end, delt*

% Initial Conditions:

X(1)= 161.916;% g cells/ L

S(1)= 7.125;% g substrate/ L

P(1)= 80.96;% g protein /L

*Obtain from data file: X\_act*

%Nominal Parameter Values (To Be Fitted):

*replace with max, min, and inc values for for loops to fit mu\_max, K, and Ki*

% Input

*Obtain from data file: Fin\_act*

% Parameters:

V=1000;% bioreactor volume L

YSX = 1.5;% yield g substrate/ g cells

YPX = 0.5;% yield g protein/ g cells

Xin = 0;% feed biomass concentration g cells/L

Sin = 250;% feed substrate concentration g substrate/L

Pin = 0;% feed protein concentration g protein/L

**Flow Diagram for Brute Optimization Parameter Fit**

A diagram of a flowchart

Description automatically generated

1. Build off the part 4 ic file done earlier. Use the same data file final\_proj\_data.mat. Rename part4 ic so part 5.
   1. In part 4 you removed mu\_max= and replace with
      1. mu\_max\_max=0.7
      2. mu\_max\_min=0.4
      3. mu\_max\_inc=0.05

Now do the same for K and Ki variables. Ranges are up to you. I would consider starting so the values given are within your initial range before moving to something much different.

*Note: make sure that best value is not on an extreme value either a minimum or maximum. Not sure then if best value or not.*

* 1. When new MSE is better than Best\_MSE save Best\_MSE=MSE and save the parameter(s) that caused it eg. Best\_mu\_max=mu\_max Best\_K, and Best\_Ki
  2. Keep adjusting ranges and/or increments so find mu\_max, K, and Ki to nearest hundredth for mu\_max and nearest tenths for K and Ki.

1. Generate one figure with 4 subplots:
   1. Title should be: Final Project Part 4 MSE= “MSE” with best mu max= “mu\_max”, best K=”K” , and best Ki=”Ki” NOTE:”” are embedded parameters
   2. t vs X as red crosses

t\_act vs X\_act as black diamonds

legend “Predicted” “Actual”

with yaxis “X (g cells/L)”

* 1. t vs S with yaxis “S (g substrate/L)”
  2. t vs P with yaxis “P (g protein/L)”
  3. t vs Fin with yaxis “Fin (L/hr)” and xaxis “Time (hrs)”

Paste here the following:

1. final\_project\_part5\_ic.m file
2. figure generated

%Bioreactor\_Final\_Project\_Part\_5\_ic

%Ransford Antwi%

%March 5, 2024%

%

%blank slate%

clear

clc

% user assigned%

%loading data file

load final\_proj\_3\_data\_file.mat

% Simulation Parameters

% will obtain from data file instead

% Initial Conditions:

X\_init= 161.916;% g cells/ L

S\_init= 7.125;% g substrate/ L

P\_init= 80.96;% g protein /L

%%%Ki

Ki\_max=53;

Ki\_min=47;

Ki\_inc=1;

%%%K

K\_max = 22;% maximum growth rate h-1

K\_min = 19;% maximum growth rate h-1

K\_inc=1;%increament

%%%mu\_max

mu\_max\_max = 0.59;% maximum growth rate h-1

mu\_max\_min = 0.58;% maximum growth rate h-1

mu\_max\_inc=0.01;%increament

%input added

Fin=Fin\_act;

%Parameters

V=1000;% bioreactor volume L

Ysx = 1.5;% yield g substrate/ g cells

Ypx = 0.5;% yield g protein/ g cells

Xin = 0;% feed biomass concentration g cells/L

Sin = 250;% feed substrate concentration g substrate/L

Pin = 0;% feed protein concentration g protein/L

%main program

%[xdot]=lorenz\_ode\_2(t,x,opts,sigma,r,b)

%intialization

X(1)=X\_init;

S(1)=S\_init;

P(1)=P\_init;

x0=[X(1);S(1);P(1);];

%unpacking to various parameters

param(1)=V;

param(2)=Ysx;

param(3)=Ypx;

param(4)=Xin;

param(5)=Sin;

param(6)=Pin;

%initial time

t(1)=0;

%number of samples in for loop

N=ceil(t\_end/delt);

%%%%%ECP1

%best MSE

Best\_MSE= 1E10;

%a loop fo K,Ki,mumax

for Ki=Ki\_min:Ki\_inc:Ki\_max

for K=K\_min:K\_inc:K\_max

for mu\_max=mu\_max\_min:mu\_max\_inc:mu\_max\_max

% Initial Sum of Square Error

SSE=(X\_act(1)-X(1))^2;

%This is my experimental loop!!!!!

for i=2:N

t(i)=t(i-1)+delt;

%[xdot]=vdv\_fxn(t,x,opts,param,u)

[~,Xout]=ode45('bioreactor\_ode\_EC',[t(i-1) ,t(i)],x0,[],param,Fin(i-1),mu\_max,K,Ki);

%unpack Xout to common notation for states

X(i)=Xout(end,1);

S(i)=Xout(end,2);

P(i)=Xout(end,3);

% ba reinitialization vector

x0=[X(i);S(i);P(i)];

%BA call sum of sq error

SSE=SSE+(X\_act(i)-X(i))^2;

end

%Mean square error divide total by numebr sample

MSE=SSE/N;

if MSE<Best\_MSE

Best\_MSE=MSE; % save current MSE as the new best one

Best\_mu\_max=mu\_max; % save the current paramater used as the new best one.

Best\_K=K;

Best\_Ki=Ki;

end

% Clean up and Reset Experiment 7. BA

clear t X S P % always clear the time vector and states

% Reinitialize

t(1)=0; % reinitialize time vector

X(1)=X\_init; % 2. set first CA and T to init values

S(1)=S\_init;

P(1)=P\_init;

x0=[X(1);S(1);P(1)];% ba initialization vector

end %end of loop for Mu\_max

end %end of loop for K

end %end of loop for Ki

% Set parameter(s) to Best ones found! and rerun simulation 8. BA

mu\_max=Best\_mu\_max;

K=Best\_K;

Ki=Best\_Ki;

%experimental loop

% Initial Sum of Square Error

SSE=(X\_act(1)-X(1))^2;

for i=2:N

t(i)=t(i-1)+delt;% BA time vector

%[xdot]=vdv\_fxn(t,x,opts,param,u)

[~,Xout]=ode45('bioreactor\_ode\_EC',[t(i-1) ,t(i)],x0,[],param,Fin(i-1),mu\_max,K,Ki);

%unpack Xout to common notation for states

X(i)=Xout(end,1);

S(i)=Xout(end,2);

P(i)=Xout(end,3);

% ba reinitialization vector

x0=[X(i);S(i);P(i)];

%BA call sum of sq error

SSE=SSE+(X\_act(i)-X(i))^2;

end

MSE=SSE/N; % mean square error divide total by number samples

%Graphing

%ploting subplot

figure(1);

subplot(4,1,1);plot(t,X,'rx',t\_act,X\_act,'kd')

ylabel('X(g cells/L')

legend('Predicted','Actual')

title(['Final Project Part 4 MSE=',num2str(MSE),' with the best mu max=',num2str(mu\_max),'best K=',num2str(K),'and best Ki=',num2str(Ki)])

subplot(4,1,2);plot(t,S)

ylabel('S(g substrate/L')

subplot(4,1,3);plot(t,P)

ylabel('P(g Protein/L')

subplot(4,1,4);plot(t,Fin)

ylabel('Fin(g L/hr)')

xlabel('Time(hrs)')

1. Figure Obtained
2. 

**Extra Credit: Optimize Production by Fin**

**Objective Function for Production:**

Production here is the sum of protein product produced, P, over the whole run. The formula to calculate this is straightforward the volumetric flowrate, Fin, times the exit protein concentration at that sample time P(i) times the duration of sample, delt, all summed over the run. Goal is to maximize this amount.

1. Use the best set of parameters found in Part 5 as your base model to be optimized for production. Here only maximizing a fixed flowrate, Fin so no need for a graph of Fin just embed the best production and Fin in the title. So could use best parameters from part 5 and start with Part 4 ic since it has one for loop for trying different combinations for different combinations of mu\_max which you can hijack and switch to different combinations of Fin.
2. Not loading any actual data here! Optimizing your model to find best set of conditions which then will get tested by actual experiments later.
3. Place in parameters the best values for mu\_max, K and Ki found from part 5 into the user section. No mu\_max\_min, etc… remove from user section and replace with mu\_max with given best value.
4. For input remove Fin and replace with Fin\_min, Fin\_inc and Fin\_max
   1. Make sure Best\_Prod=0 as initial value so easily beaten for first experiment production.
   2. Initialize Prod=0 for sum inside for loop later.
   3. Modify the ode45 call so no longer Fin(i-1) since Fin is fixed value here just put Fin with no indexing.
   4. When new Prod is better than Best\_Prod save Best\_Prod=Prod and save the input Fin that caused it eg. Best\_Fin=Fin.

*Note: cleaning up experiment and reinitializing are all done same way as before.*

1. Generate one figure with 3 subplots:
2. Title should be: Final Project EC Best Prod= “Prod” with best mu max= “mu\_max”, best K=”K” , and best Ki=”Ki” NOTE:”” are embedded parameters
3. t vs X

with yaxis “X (g cells/L)”

1. t vs S with yaxis “S (g substrate/L)”
2. t vs P with yaxis “P (g protein/L)”

with xaxis “Time (hrs)”

Paste here the following:

1. final\_project\_EC\_ic.m file
2. figure generated

%Bioreactor\_Opt\_Production\_EC\_ic%

%Ransford Antwi%

%March 5, 2024%

%Final\_Project\_Part\_EC\_main

%blank slate%

clear

clc

% user assigned%

%sample time

t\_end=20;

delt=0.1;

% Initial Conditions:

X\_init= 161.916;% g cells/ L

S\_init= 7.125;% g substrate/ L

P\_init= 80.96;% g protein /L

%Fitted Parameter Values (To Be Fitted):

K= 20;% best K

Ki= 51;% best Ki

mu\_max=0.59; %best mu\_max

%input added

Fin\_min=10;% initial feed flow rate L/hr

Fin\_inc=5; %increment

Fin\_max=100;% final feed flow rate L/hr

%Parameters

V=1000;% bioreactor volume L

Ysx = 1.5;% yield g substrate/ g cells

Ypx = 0.5;% yield g protein/ g cells

Xin = 0;% feed biomass concentration g cells/L

Sin = 250;% feed substrate concentration g substrate/L

Pin = 0;% feed protein concentration g protein/L

%main program

%[xdot]=lorenz\_ode\_2(t,x,opts,sigma,r,b)

%intialization

X(1)=X\_init;

S(1)=S\_init;

P(1)=P\_init;

x0=[X(1);S(1);P(1);];

%unpacking to various parameters

param(1)=V;

param(2)=Ysx;

param(3)=Ypx;

param(4)=Xin;

param(5)=Sin;

param(6)=Pin;

param(7)=K;

param(8)=Ki;

%initial time

t(1)=0;

%number of samples in for loop

N=ceil(t\_end/delt);

%best production being maximized

Best\_Prod= 0;

for Fin=Fin\_min:Fin\_inc:Fin\_max

% Initial Sum of prod

Prod=0;

%This is my experimental loop!!!!!

for i=2:N

t(i)=t(i-1)+delt;

%[xdot]=vdv\_fxn(t,x,opts,param,u)

[~,Xout]=ode45('bioreactor\_ode\_EC',[t(i-1) ,t(i)],x0,[],param,Fin,mu\_max,K,Ki);

%unpack Xout to common notation for states

X(i)=Xout(end,1);

S(i)=Xout(end,2);

P(i)=Xout(end,3);

% ba reinitialization vector

x0=[X(i);S(i);P(i)];

%BA calculate sum of production

Prod=Prod+Fin\*P(i)\*delt;

end

%compare to the previous ones to see if this one is better

if Prod>Best\_Prod

Best\_Prod=Prod;

Best\_Fin=Fin;

end

% Clean up and Reset Experiment 7. BA

clear t X S P % always clear the time vector and states

% Reinitialize

t(1)=0; % reinitialize time vector

X(1)=X\_init; % 2. set first CA and T to init values

S(1)=S\_init;

P(1)=P\_init;

x0=[X(1);S(1);P(1)];% ba initialization vector

end %end o loop for Mu\_max

% Set parameter(s) to Best ones found! and rerun simulation 8. BA

Fin=Best\_Fin;

%experimental loop

% Initial Sum of production

Prod=0;

for i=2:N

t(i)=t(i-1)+delt;% BA time vector

%[xdot]=vdv\_fxn(t,x,opts,param,u)

[~,Xout]=ode45('bioreactor\_ode\_EC',[t(i-1) ,t(i)],x0,[],param,Fin,mu\_max,K,Ki);

%unpack Xout to common notation for states

X(i)=Xout(end,1);

S(i)=Xout(end,2);

P(i)=Xout(end,3);

% ba reinitialization vector

x0=[X(i);S(i);P(i)];

%BA calculate sum of production

Prod=Prod+Fin\*P(i)\*delt;

end

%Graphing

%ploting subplot

figure(1);

subplot(3,1,1);plot(t,X,'rx')

ylabel('X(g cells/L')

title([' Final Project EC Best Prod= ',num2str(Prod),' with the best mu max=',num2str(mu\_max),'Best K=',num2str(K),'and Best Ki=',num2str(Ki)])

subplot(3,1,2);plot(t,S)

ylabel('S(g substrate/L')

subplot(3,1,3);plot(t,P)

ylabel('P(g Protein/L')

xlabel('Time(hrs)')

