

# **Physiological Models and Computations**

## **Exercises**

Department of Automatic Control  
Lund University, Faculty of Engineering  
2013



## 0. Introduction to MATLAB and SIMULINK

---

Solve the following exercises using MATLAB. These exercises are inspired by or fully extracted from *EDA017: Föreläsningsanteckningar, OCTAVE/MATLAB* by Christian Söderberg.

---

### 0.1

- a. Plot  $y(x) = e^{-x/2}\cos(2\pi x)$  when  $-6 \leq x \leq 3$  by using the function handle to create an anonymous function. Give your plot a title as well as labels on the axes. Useful commands: `fplot`, `xlabel`, `ylabel`, `title`.
- b. Modify your code such that you only show values  $-4.5 \leq x \leq -1$  and  $-10 \leq y \leq 10$ . Useful command: `axis`.
- c. Integrate the function for  $-4.5 \leq x \leq -1$ . Useful commands: `integral`, `quad`.
- d. Find the solution to  $f(x) = 0$  when  $f(x) = x^3 + 2x - 1$ . Comment on the answer. Useful command: `fsolve`.

- 0.2** Write a function which for every matrix A gives you the sum of the diagonal elements of that matrix. Useful commands: `diag`, `sum` and `size`.

- 0.3** Solve the differential equation

$$\ddot{y} + 7\dot{y} - 3y = 0$$

$$y(0) = 0$$

$$\dot{y}(0) = 1$$

in the interval  $0 \leq t \leq 5$  by using MATLABs solver `ode45`.

- 0.4** Try to fit a first order polynomial  $ax + b$  to the following measurements

x	y
1	3.9286
2	5.4059
3	6.0771
4	7.7145

---

Solve the following exercises using SIMULINK in MATLAB. These exercises are taken from *Exercises in MATLAB/Simulink, Signals and Systems* by Thomas Munther.

---

- 0.5** Investigate the bacterial growth in a jam pot. Assume that the number of born bacteria is increasing proportional to the existing number of bacteria  $x$  and the number dying is proportional to the existing number in square. This gives the following differential equation

$$\frac{dx}{dt} = bx - px^2$$

where  $b = 1$  [1/hour] is the birth rate constant and  $p = 0.5$  [1/(bacteria·hour)] is the death rate constant. Assume  $x(0) = 100$  [bacteria]. Use SIMULINK to show how the solution to the differential equation looks like.

- 0.6** Some physiological systems are better described in discrete time which gives rise to difference equations. Show the behavior of  $y$  in the two following difference equations

**a.**

$$y_t = -0.5 \cdot y_{t-1} + x_t$$

**b.**

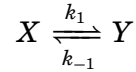
$$y_t = 0.5 \cdot y_{t-1} + x_t$$

where  $x$  is the input signal to the system, in shape of a step starting in  $t = 0$  with amplitude 1 and  $y_{-1} = 1$ .  $y_t$  is the value of  $y$  in time step  $t$ .

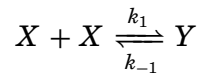
# 1. Biochemical Reactions

- 1.1** Use the law of mass balance to derive the differential equations governing the production of X and Y:

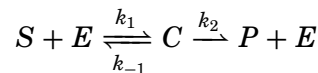
a.



b.



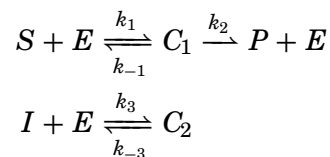
- 1.2** Simulate and plot the concentrations for the substrate  $S$ , enzyme  $E$ , substrate-enzyme complex  $C$  and the end product  $P$  for the basic enzymatic reaction



using the following set of parameters;  $k_1 = 0.1$ ,  $k_{-1} = 0.01$  and  $k_2 = 0.02$ , and with the following initial conditions  $[S]_0 = 0.15$  [mmol/l],  $[E]_0 = 0.01$  [mmol/l],  $[C]_0 = 0$  [mmol/l] and  $[P]_0 = 0$  [mmol/l]. What happens if the initial concentration of the enzyme is doubled? What happens if the initial concentration of the substrate is doubled? How does these results correspond to the Michealis-Menten parameters?

- 1.3** The data in Table 1.1 describes the concentration and reaction rates of a chemical process. Is it an enzymatic reaction following the Michaelis-Menten relationship? Can you give some rough estimates of  $V_{max}$  and  $K_m$  from this graph? Plot the inverse of the concentration versus the inverse of the reaction rate. This plot is commonly reffered to as a Lineweaver-Burk plot. Can you give some rough estimates of  $V_{max}$  and  $K_m$  from this graph as well?

- 1.4** Competetive Inhibition: Some enzymes may bind other substances than the target substrate to the binding site, thereby inhibiting the formation of the intended substrate-enzyme complex and the subsequent end-product. Such a situation is characterized by the following reaction dynamics:



**Table 1.1** Reaction Data for problem 3

Substrate	Reaction
Concentration [mM]	Velocity [mM/s]
0.1	0.04
0.2	0.08
0.5	0.17
1.0	0.24
2.0	0.32
3.5	0.39
5.0	0.42

Derive the following relationship for the reaction velocity of the product reaction, considering steady-state conditions for the enzyme and enzyme complexes and preservation of the total enzyme content:

$$V = \frac{V_{max}[S]}{[S] + K_m(1 + [I]/K_I)}$$

where  $[I]$  is the concentration of the inhibitor,  $K_m = (k_{-1} + k_2)/k_1$  and  $K_I = k_{-3}/k_3$ .

- 1.5** Alcohol metabolism: Clearance of the blood alcohol level (BAL)  $[A]$  [mg/dl] from the liver is metabolized by more than 20 different enzymes. From experimental data the total clearance effect of these enzymes has been lumped into a common Michaelis-Menten relationship with population average  $V_{max} = -15[\text{mg}/(\text{dl} \cdot \text{h})]$  and a  $K_m = 5$  [mg/dl].

$$\frac{d[A]}{dt} = \frac{V_{max}[A]}{K_m + [A]}$$

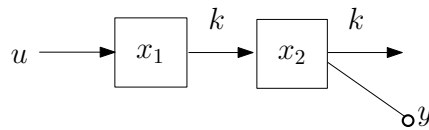
To calculate the BAL, the total distribution volume of the body for alcohol has to be known. The following relationship between the total water volume, representing this distribution volume  $V_D$  [l], and the weight  $m_{BW}$  [kg], gender and age  $Y$  [years] of the person has been suggested.

$$\begin{aligned} V_D &= 20 + 0.36m_{BW} - 0.1Y, & \text{Men} \\ V_D &= 14 + 0.25m_{BW}, & \text{Women} \end{aligned}$$

Assuming that a 25 year old man of 80 kg consumes a drink containing 2 cl of alcohol (density  $800 \text{ kg/m}^3$ ) at a fasting state. Digestion of alcohol is very rapid on an empty stomach, and you may assume that the total alcohol content has reached the blood stream after 20 minutes whereafter metabolization is considered to start. Simulate and plot the BAL level for the four hours following the drink.

## 2. Model Building and Linearization

**2.1** Given the compartment model below



assume that  $x_1$  and  $x_2$  represent quantities of a substance subject to conservation.  $y$  is a measurement of  $x_2$ .

- Give the balance equations when  $k = 1$ . What are the states, the input and the output of the system?
- From the balance equations derive the state space representation for the system.
- Determine the transfer function of the system analytically and by using functions from the control toolbox in MATLAB.

**2.2** Give the state-space representation of the system

$$\ddot{y} + 3\dot{y} + 2y = u$$

where  $u(t)$  and  $y(t)$  are the input and output, respectively. Choose states  $x_1 = y$ ,  $x_2 = \dot{y}$  and  $x_3 = \ddot{y}$ .

**2.3** A process with output  $y(t)$  and input  $u(t)$  is described by the differential equation

$$\dot{y} + \sqrt{y} + y\dot{y} = u^2$$

- Introduce states  $x_1 = y$ ,  $x_2 = \dot{y}$  and give the state space representation of the system.
- Find all stationary points  $(x_1^0, x_2^0, u^0)$  of the system.
- Linearize the system around the stationary point corresponding to  $u^0 = 1$ .

**2.4** Linearize the system

$$\begin{aligned} \dot{x}_1 &= x_1^2 x_2 + \sqrt{2} \sin u & (= f_1(x_1, x_2, u)) \\ \dot{x}_2 &= x_1 x_2^2 + \sqrt{2} \cos u & (= f_2(x_1, x_2, u)) \\ y &= \arctan \frac{x_2}{x_1} + 2u^2 & (= g(x_1, x_2, u)) \end{aligned}$$

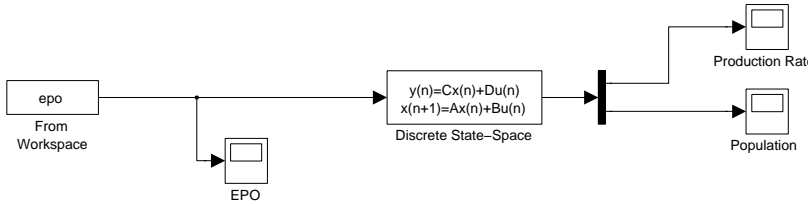
around the stationary point  $u^0 = \pi/4$ .

**2.5** Blood Doping: Everyday about  $2.5 \cdot 10^{11}$  (0.25 trillion) new red blood cells (RBCs) are released from the bone marrow into the peripheral circulation, and in steady-state the same number of depleted RCB:s

are cleared by the spleen. Assume that the average lifespan of a RCB is 120 days, and the cleared amount between two days  $k$  and  $k - 1$  is a constant fraction  $f$  of the total cell population  $R(k - 1)$  at day  $k - 1$ . The cell population  $R(k)$  is  $R_{ref}$  [trillion cells] at steady state. Furthermore, the rate of production  $r(k)$  [trillion cells/day] is controlled by the level of erythropoietin EPO  $u(k)$  [Units/ml] according to the outlined dynamics below (changes in the EPO level do not fully affect the production rate directly, but the production rate  $r(k)$  is partly dependent on the production rate the previous day  $r(k - 1)$ ):

$$r(k) = 0.9 \cdot r(k - 1) + u(k - 1), \quad r(0) = f \cdot R_{ref}, u(0) = 0.025 \quad (2.1)$$

Set up the difference equations for the red blood cell population  $R(k)$  and the production rate  $r(k)$ . Assume that we are at steady state with a total cell population  $R_{ref}$  of  $120 \cdot 0.25$  trillion cells. Create a Simulink model according to Fig. 2.1 and simulate the system for 100 days. Assume that the level of EPO normally is constant at 0.025 Units/ml, but that it is artificially elevated to the double normal level by injections for 20 consecutive days between day 21 and 40.



**Figure 2.1** Simulink model for the red blood cell system

**2.6** Infection; Bacteria-Leukocytes Predator-Prey System: Neutrophils are specialised white blood cells (leukocytes), specialising in defending against bacterial infections. Let  $B(t)$  denote the number of bacteria in a wound and  $N(t)$  the number of neutrophils. The bacterial growth factor is  $\alpha$  [bacteria/hour] and the killing factor of the neutrophils  $\beta$  [bacteria/hour] and assume that the entry rate of new neutrophils is  $u(t)$  [neutrophils/hour].

$$\frac{dB}{dt} = \alpha B(t) - \beta \cdot B(t) \cdot N(t) \quad (2.2)$$

$$\frac{dN}{dt} = -\gamma N(t) + u(t) \quad (2.3)$$

Simulate the system in Simulink with  $\alpha = 3, \beta = 1.1, \gamma = 1.5$ , and with initial conditions  $B(0) = 100, N(0) = 0$  and let  $u(t)$  be a step with magnitude 10. What happens if  $\alpha$  becomes large ( $> 8$ )?



# Solutions to Chapter 0. Introduction to MATLAB and SIMULINK

---

Solve the following exercises using MATLAB. These exercises are inspired by or fully extracted from *EDA017: Föreläsningsanteckningar, OCTAVE/MATLAB* by Christian Söderberg.

---

- 0.1 a.** Create an anonymous function using the function handle. This function is only saved in your workspace until you close MATLAB (or clear your workspace by the `clear all` command). In case you would like to save your function as a file in your current folder (from where you can reach it at another time), use a function m-file (go to new → function).

```
y = @(x) exp(-x/2)*cos(2*pi*x);

figure
fplot(y, [-6 3])
title('My fancy plot')
xlabel('x')
ylabel('y')
```

`figure` is a command which is useful when you want to create several plots in the same script. Use the `help`-command whenever you need information about one of MATLAB's built-in functions. In this case you would write `help figure` in the command window and the description of the function should appear.

**b.**

```
axis([-4.5 -1 -10 10])
```

- c.** % Rewrite `y` to be accepted by `quad/integral` (read in the % description of `quad/integral` to understand why).

```
y = @(x) exp(-x/2).*cos(2*pi*x);

integral(y, -4.5, -1)
% or
quad(y, -4.5, -1)
```

- d.** `f = @(x) x^3+2*x-1;`  
`solution = fsolve(f, 0)`

The answer is 0.4534. Write `format long` in the command window (then use the `fsolve` command) to get more decimals in the answer. Due to it being numerically calculated  $f(0.4534)$  is approximately zero.

- 0.2** Go to new → function. A file with a function-shell will appear. The function shell looks like:

```
function [ output_args ] = untitled( input_args )
%UNTITLED Summary of this function goes here
% Detailed explanation goes here

end
```

Replace `untitled` with the name of your function, `input_args` with the input your function needs and `output_args` with the output your function will give. Between the `function`-row and the `end` you should write the code for the function.

For the particular function of this exercise, it will look as follows

```
function sumOfDiag = sumOfDiagonal(A)
[n,m] = size(A);

if n ~= m
    error('A is not a square matrix')
end

sumOfDiag = sum(diag(A));
end
```

Where  $\neq$  is written as `~=` in MATLAB. Save your function as an m-file in your current folder, by the name of your function. In this case it would be "sumOfDiagonal.m". Now you can use your function directly from the command window or from a script which is saved in the same folder as your function.

To create a matrix in MATLAB use the following principle

```
my_matrix = [1 2; 3 4];
```

[ and ] begins and ends the matrix. Elements are separated by space (or comma) and rows are separated by ;. The resulting matrix is

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

- 0.3** Introduce  $y_1(t) = y(t)$  and  $y_2(t) = \dot{y}(t)$  in order to rewrite the initial second-order differential equation into two first-order differential equations as follows

$$\dot{y}_1 = y_2 \quad (0.1)$$

$$\dot{y}_2 = 3y_1(t) - 7y_2 \quad (0.2)$$

The initial conditions for  $y_1(t)$  and  $y_2(t)$  are

$$y_1(0) = y(0) = 0$$

$$y_2(0) = \dot{y}(0) = 1$$

(0.1) and (0.2) can be written together on matrix form as follows

$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} y_2 \\ 3y_1 - 7y_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 3 & -7 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

Define  $\mathbf{v} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$ . Then, define  $\mathbf{f}$  as the following function

$$\mathbf{f}(t, \mathbf{v}) = \mathbf{f}\left(t, \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\right) = \begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} y_2 \\ 3y_1 - 7y_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 3 & -7 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

In MATLAB this can be written as

```
f = @(t,v) [v(2); 3*v(1)-7*v(2)];
```

Or by matrix multiplication

```
f = @(t,v) [0 1; 3 -7]*v;
```

To solve the differential equation write the following code

```
[t_ode V] = ode45(f,[0 5],[0 1]);
```

The first input to `ode45` is the right part of the differential equation, the second input is the time span of the solution while the third is the initial condition of the differential equation.  $\mathbf{V}$  is a matrix with two columns, the first column corresponds to  $y_1(t) = y(t)$  and the second column corresponds to  $y_2(t) = \dot{y}(t)$ .  $\mathbf{t\_ode}$  is the times between 0 and 5 at which `ode45` has calculated  $y_1$  and  $y_2$ . Use the following code to plot  $y(t)$  over  $0 \leq t \leq 5$

```
plot(t_ode,V(:,1))
```

#### 0.4 The first order polynomial means that

$$b + a = 3.9286$$

$$b + 2a = 5.4059$$

$$b + 3a = 6.0771$$

$$b + 4a = 7.7145$$

In matrix form this becomes

$$\begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 3.9286 \\ 5.4059 \\ 6.0771 \\ 7.7145 \end{pmatrix} \quad (0.3)$$

We have two unknowns and four equations. Therefore, we need to approximate  $a$  and  $b$  such that the distance between the line  $ax + b$  and the points is minimized in some sense.

If (0.3) is seen as  $S \cdot \begin{pmatrix} a \\ b \end{pmatrix} = T$ , the following code will return the values of  $a$  and  $b$

```
x = S\T;
```

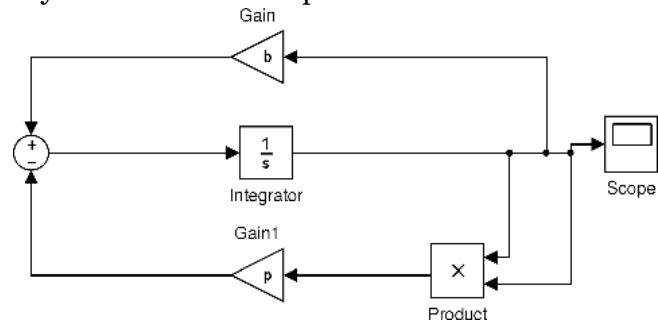
Where  $a = x(1)$  and  $b = x(2)$ . This uses the least squares method to fit  $ax + b$  to the points. Plot the points and the line in the same plot to see the fit.

---

Solve the following exercises using SIMULINK in MATLAB. These exercises are taken from *Exercises in MATLAB/Simulink, Signals and Systems* by Thomas Munther.

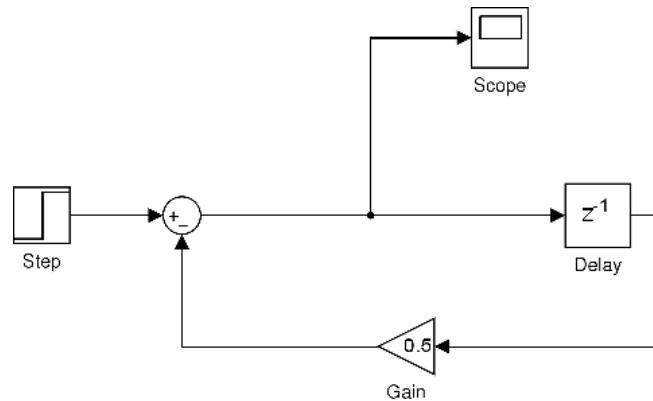
---

- 0.5** Start SIMULINK by writing `simulink` in the MATLAB command window. This makes the SIMULINK Library Browser window pop up. Go to File → New → Model. In this window you can start to create your SIMULINK model. Use the Library Browser to find appropriate blocks and drag them into the model sheet. You can connect two blocks by their connection spots.

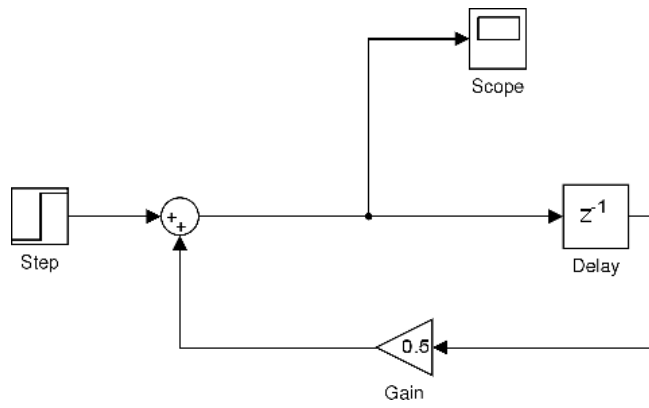


`p` and `b` can be defined in the current workspace. Go to display → blocks and check "Sorted Execution Order". This will numerate the blocks in the order in which they are first activated.

- 0.6 a.** Before running the simulation go to Simulation → Configuration Parameters. In Solver Options choose Fixed-step and Solver → Discrete. Set the sample time in each block to 1 [sec].



- b.** The only difference from the previous model is that the minus sign in the sum-block is changed to a plus sign.



# Solutions to Chapter 1. Biochemical Reactions

1.1 a. Denote the concentrations  $x = [X]$  and  $y = [Y]$

$$\frac{dx}{dt} = -k_1x + k_{-1}y$$
$$\frac{dy}{dt} = k_1x - k_{-1}y$$

b. Denote the concentrations  $x = [X]$  and  $y = [Y]$

$$\frac{dx}{dt} = -2k_1x^2 + 2k_{-1}y$$
$$\frac{dy}{dt} = k_1x^2 - k_{-1}y$$

1.2 A matlab script may look as follows:

```
% Simulation of the substrate, enzyme and product concentrations in a MM
% example
% ds/dt = -k_1 *(se) + k_{-1}*c
% de/dt = -k_1 *(se) + (k_{-1} + k_2)*c
% dc/dt = k_1 *(se) - (k_{-1} + k_2)*c
% dp/dt = k_2 c
%-----
% Initial conditions
s(1) = 0.15; % mmol/L
e(1) = 1e-2; % mmol/L
c(1) = 0; % mmol/L
p(1) = 0; % mmol/L
%-----
% Parameters
k1 = 0.1;
k3 = 0.01; % k_{-1}
k2= 0.02;
%-----

% Run discretized simulation
for k = 2:10000
    s(k) = s(k-1) + k3*c(k-1) - k1*s(k-1)*e(k-1);
    e(k) = e(k-1) + (k3+k2)*c(k-1) - k1*s(k-1)*e(k-1);
    c(k) = c(k-1) - (k3+k2)*c(k-1) + k1*s(k-1)*e(k-1);
    p(k) = p(k-1) + k2*c(k-1);
end

figure(1)
[ax,h1,h2] = plotyy(1:10000,[s' p'],1:10000,[e' c'])
legend('Substrate','Product','Enzyme','Complex')
xlabel('time [s]')
ylabel(ax(1),'Substrate/Product Concentration [mmol/L]')
ylabel(ax(2),'Enzyme/Complex Concentration [mmol/L]')
title('Simulation of enzymatic reaction')
```

```
% Run ode-solver simulation
% y = [S E C P]

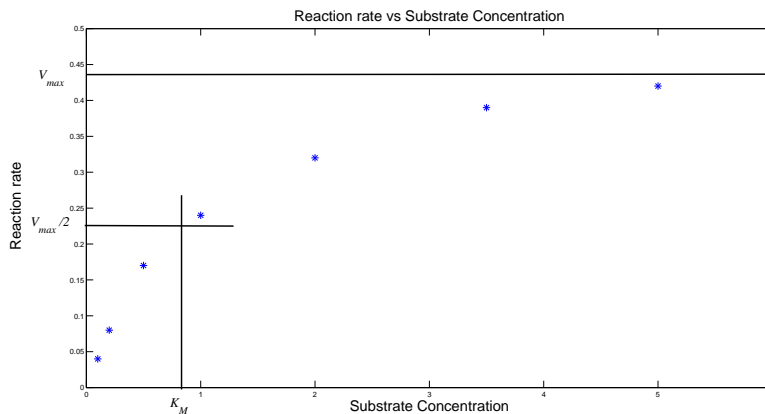
dAll = @(t,y) [-k1*y(1)*y(2)+k3*y(3); ...
    -k1*y(1)*y(2)+(k3+k2)*y(3); ...
    k1*y(1)*y(2)-(k3+k2)*y(3); ...
    k2*y(3)];

[t Y] = ode45(dAll,[0 10000],[0.15 1e-2 0 0])

figure(2)
[ax,h1,h2] = plotyy(t,[Y(:,1) Y(:,4)],t,[Y(:,2) Y(:,3)])
legend('Substrate','Product','Enzyme','Complex')
xlabel('time [s]')
ylabel(ax(1),'Substrate/Product Concentration [mmol/L]')
ylabel(ax(2),'Enzyme/Complex Concentration [mmol/L]')
title('Simulation of enzymatic reaction')
```

Doubling the enzymatic concentration doubles the production rate since  $V_{max} = k_2 \cdot e_0$ . Likewise since  $K_m = (k_2 + k_{-1})/k_1 = 0.3$  and  $V = V_{max}s/(K_m + s)$ , a doubling of  $s_0$  from  $K_m/2$  to  $K_m$  means that the initial reaction rate will become 1.5 times greater.

- 1.3** The plot indicates that the relationship between the reaction rate and the substrate concentration goes to saturation in a M-M-like behaviour, see Fig. 1.1.  $V_{max}$  and  $K_m$  are estimated as shown in the plot.



**Figure 1.1** Graphical estimation of  $V_{max}$  and  $K_M$

Lineweaver-Burke plot: The Michaelis-Menten relationship between substrate concentrations  $[S]$  states that:

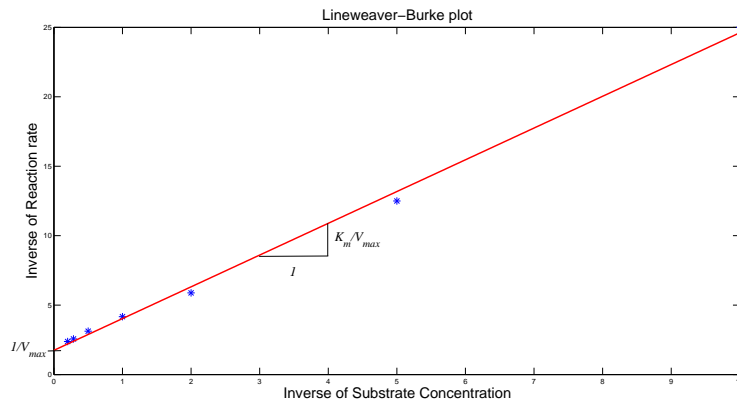
$$v = \frac{V_{max}[S]}{K_m + [S]}$$

Taking the inverse yields:

$$\frac{1}{v} = \frac{K_m}{V_{max}} \frac{1}{[S]} + \frac{1}{V_{max}}$$

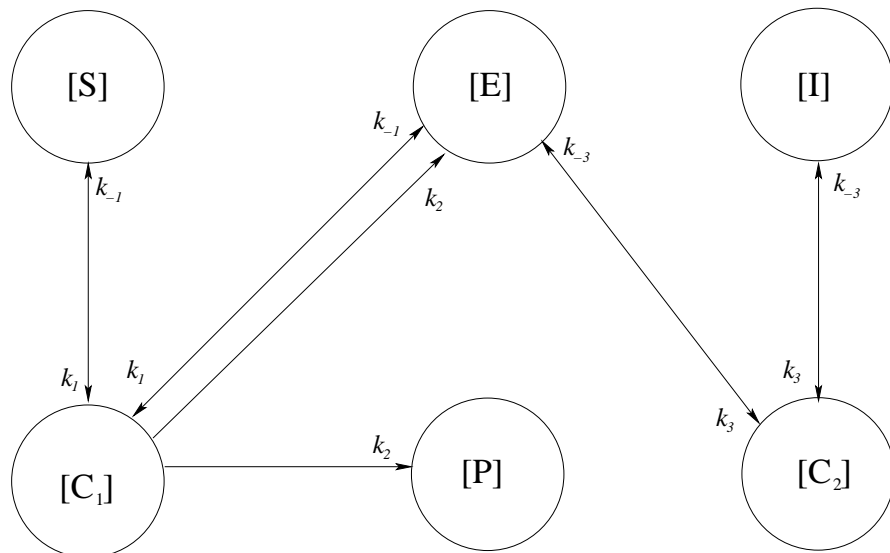
Now, the parameters  $K_m/V_{max}$  and  $1/V_{max}$  for this linear relationship may be estimated from the plot as seen in Fig. 1.2.





**Figure 1.2** Graphical estimation of  $V_{max}$  and  $K_M$  using the Lineweaver-Burke plot.

**1.4** Draw a graph of the compartment representation, see Fig 1.3. Next,



**Figure 1.3** Compartment model representation of the enzyme inhibition dynamics.

determine the differential equations governing the reaction dynam-

ics:

$$\frac{d[S]}{dt} = -k_1[S][E] + k_{-1}[C_1] \quad (1.1)$$

$$\frac{d[I]}{dt} = k_{-3}[C_2] - k_3[E][I] \quad (1.2)$$

$$\frac{d[C_1]}{dt} = k_1[S][E] - (k_{-1} + k_2)[C_1] \quad (1.3)$$

$$\frac{d[C_2]}{dt} = k_3[E][I] - k_{-3}[C_2] \quad (1.4)$$

$$\frac{d[E]}{dt} = (k_2 + k_{-1})[C_1] + k_{-3}[C_2] - k_1[S][E] - k_3[E][I] \quad (1.5)$$

$$\frac{d[P]}{dt} = k_2[C_1] \quad (1.6)$$

Next, use the steady-state assumptions;  $d[C_1]/dt = d[C_2]/dt = 0$  to get

$$[C_1] = \frac{k_1}{k_{-1} + k_2}[S][E] \quad (1.7)$$

$$[C_2] = \frac{k_3}{k_{-3}}[E][I] \quad (1.8)$$

The conservation of enzymatic mass gives

$$[E_0] = [E] + [C_1] + [C_2] = [E]\left(1 + \frac{k_1}{k_{-1} + k_2}[S] + \frac{k_3}{k_{-3}}[I]\right) \quad (1.9)$$

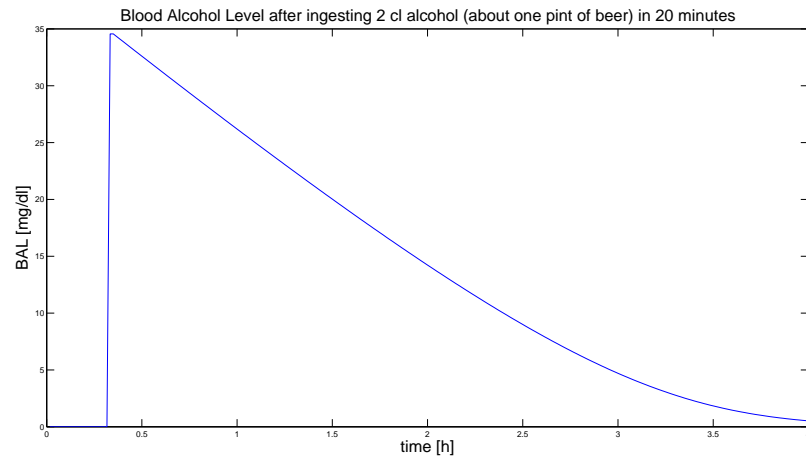
Put Eq. (1.6), Eq. (1.7) and Eq. (1.9) together:

$$V = \frac{d[P]}{dt} = \frac{k_2[E_0][S]}{[S] + \frac{k_1}{k_{-1} + k_2}\left(1 + \frac{k_3}{k_{-3}}[I]\right)} \quad (1.10)$$

## 1.5 Blood alcohol level

A matlab script may look as follows:

```
% BAL simulation
V = -15;% mg/(l*h)
K_m = 5;% mg/dl
%-----
VD = 10*(20 + 0.36*80-0.1*25); %dl
BAL(1:20) = zeros(20,1);
BAL(20) = 0.02*1000*0.8*1000/VD; % mg/dl
der = 0;
for k=21:1:240
    BAL(k) = BAL(k-1) + der;
    der = V/60*BAL(k)/(K_m + BAL(k));
end
plot([1:length(BAL)]/60,BAL)
title('Blood Alcohol Level after ingesting 2 cl alcohol ...
(about one pint of beer) in 20 minutes','FontSize',10)
ylabel('BAL [mg/dl]','FontSize',10)
xlabel('time [h]','FontSize',10)
```



**Figure 1.4** Blood alcohol content according to the simulation example.

Running the code generates the plot in Fig. 1.4.

Another possibility is to use MATLABs built in solvers for ordinary differential equations, such as `ode45`. The MATLAB-script would then look something like

```
% BAL simulation
V = -15;% mg/(l*h)
K_m = 5;% mg/dl
VD = 10*(20 + 0.36*80-0.1*25); % dl

% The 'initial value' of the concentration [A] is actually
% the concentration in t = 20 min when the metabolism
% of the alcohol starts.

initial_value_A = 0.02*1000*0.8*1000/VD; % mg/dl

% Define the differential equation y(t) = [A](t)
dAdt = @(t,y) V/60*y/(K_m+y);

% Solve the differential equation
[t, Y] = ode45(dAdt, [0 220], initial_value_A);

t = (t+20)/60; % Shifting the time vector 20 min, and changing into
% hours instead of minutes.
Y = [zeros(size(0:0.1:(t(1)-0.01))) Y']; % Adding zeros to the
% value-vector for time 0-20 min.
t = [0:0.1:(t(1)-0.01) t']; % Adding the time between 0-20 minutes
% to the time vector.

plot(t,Y)
title('Blood Alcohol Level after ingesting 2 cl alcohol ...
      (about one pint of beer) in 20 minutes','FontSize',10)
ylabel('BAL [mg/dl]','FontSize',10)
xlabel('time [h]','FontSize',10)
```

# Solutions to Chapter 2. Model Building and Linearization

## 2.1

a. By concentration of substrate, we have

$$\begin{aligned}\frac{dx_1}{dt} &= -x_1 + u \\ \frac{dx_2}{dt} &= x_1 - x_2 \\ y &= x_2\end{aligned}$$

The states are  $x_1$  and  $x_2$ . The input is  $u$  and the output is  $y$ .

b.

$$\begin{aligned}\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} &= \begin{pmatrix} -1 & 0 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u \\ y &= \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\end{aligned}$$

c.

$$\begin{aligned}G(s) &= C(sI - A)^{-1}B + D \\ &= \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} s+1 & 0 \\ -1 & s+1 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \frac{1}{(s+1)^2}.\end{aligned}$$

```
% State the state space matrices
A = [-1 0 ; 1 -1];
B = [1 ; 0];
C = [0 1];
D = []; % Empty matrix

% Construct the state space system
system = ss(A,B,C,D);

% Construct the transfer function
G = tf(system)

% OR after having decided the transfer function
% analytically use
s = tf('s'); % To create the Laplace variable
G = 1/(s+1)^2;
```

**2.2**

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u$$

$$y = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

**2.3 a.**

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\sqrt{x_1} - x_1 x_2 + u^2 \\ y &= x_1 \end{aligned}$$

- b.** A stationary point implies  $\dot{x}_1 = \dot{x}_2 = 0$ . From the first equation we directly obtain  $x_2 = 0$ . Subsequently, the second equation yields  $\sqrt{x_1} = u^2$ . Hence there are infinitely many stationary points and they can be parametrized through  $t$  as  $(x_1^0, x_2^0, u^0) = (t^4, 0, t)$ .
- c.**  $u^0 = 1$  gives the stationary point  $(x_1^0, x_2^0, u^0) = (1, 0, 1)$ . We let

$$\begin{aligned} f_1(x_1, x_2, u) &= x_2 \\ f_2(x_1, x_2, u) &= -\sqrt{x_1} - x_1 x_2 + u^2 \\ g(x_1, x_2, u) &= x_1 \end{aligned}$$

and compute the partial derivatives

$$\begin{aligned} \frac{\partial f_1}{\partial x_1} &= 0 & \frac{\partial f_1}{\partial x_2} &= 1 & \frac{\partial f_1}{\partial u} &= 0 \\ \frac{\partial f_2}{\partial x_1} &= -\frac{1}{2\sqrt{x_1}} - x_2 & \frac{\partial f_2}{\partial x_2} &= -x_1 & \frac{\partial f_2}{\partial u} &= 2u \\ \frac{\partial g}{\partial x_1} &= 1 & \frac{\partial g}{\partial x_2} &= 0 & \frac{\partial g}{\partial u} &= 0 \end{aligned}$$

At the stationary point we have

$$\begin{aligned} \frac{\partial f_1}{\partial x_1}(x_1^0, x_2^0, u^0) &= 0 & \frac{\partial f_1}{\partial x_2}(x_1^0, x_2^0, u^0) &= 1 & \frac{\partial f_1}{\partial u}(x_1^0, x_2^0, u^0) &= 0 \\ \frac{\partial f_2}{\partial x_1}(x_1^0, x_2^0, u^0) &= -\frac{1}{2} & \frac{\partial f_2}{\partial x_2}(x_1^0, x_2^0, u^0) &= -1 & \frac{\partial f_2}{\partial u}(x_1^0, x_2^0, u^0) &= 2 \\ \frac{\partial g}{\partial x_1}(x_1^0, x_2^0, u^0) &= 1 & \frac{\partial g}{\partial x_2}(x_1^0, x_2^0, u^0) &= 0 & \frac{\partial g}{\partial u}(x_1^0, x_2^0, u^0) &= 0 \end{aligned}$$

Use the following variable substitution

$$\begin{aligned} \Delta x_1 &= x_1 - x_1^0 \\ \Delta x_2 &= x_2 - x_2^0 \\ \Delta u &= u - u^0 \end{aligned}$$

Taylor expansion in the stationary point, where only the linear terms are used, then gives the linearized system

$$\begin{pmatrix} \Delta \dot{x}_1 \\ \Delta \dot{x}_2 \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x_1^0, x_2^0, u^0) & \frac{\partial f_1}{\partial x_2}(x_1^0, x_2^0, u^0) \\ \frac{\partial f_2}{\partial x_1}(x_1^0, x_2^0, u^0) & \frac{\partial f_2}{\partial x_2}(x_1^0, x_2^0, u^0) \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} + \begin{pmatrix} \frac{\partial f_1}{\partial u}(x_1^0, x_2^0, u^0) \\ \frac{\partial f_2}{\partial u}(x_1^0, x_2^0, u^0) \end{pmatrix} \Delta u$$

$$\Delta y = \begin{pmatrix} \frac{\partial g}{\partial x_1}(x_1^0, x_2^0, u^0) & \frac{\partial g}{\partial x_2}(x_1^0, x_2^0, u^0) \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} + \frac{\partial g}{\partial u}(x_1^0, x_2^0, u^0) \Delta u$$

Where the derivatives are given as their value in the stationary point. Using the specific values gives

$$\begin{pmatrix} \Delta \dot{x}_1 \\ \Delta \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\frac{1}{2} & -1 \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 2 \end{pmatrix} \Delta u$$

$$\Delta y = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix}$$

**2.4** At the sought operating point it holds that

$$\begin{aligned} 0 &= x_1^2 x_2 + 1 \\ 0 &= x_1 x_2^2 + 1 \\ y &= \arctan \frac{x_2}{x_1} + \frac{\pi^2}{8} \end{aligned}$$

which yields  $x_1^0 = -1$ ,  $x_2^0 = -1$  and  $y^0 = \frac{\pi}{4} + \frac{\pi^2}{8}$ . Computation of the partial derivatives now yields

$$\begin{aligned} \frac{\partial f_1}{\partial x_1} &= 2x_1 x_2 & \frac{\partial f_1}{\partial x_2} &= x_1^2 & \frac{\partial f_1}{\partial u} &= \sqrt{2} \cos u \\ \frac{\partial f_2}{\partial x_1} &= x_2^2 & \frac{\partial f_2}{\partial x_2} &= 2x_1 x_2 & \frac{\partial f_2}{\partial u} &= -\sqrt{2} \sin u \\ \frac{\partial g}{\partial x_1} &= \frac{-x_2}{x_1^2 + x_2^2} & \frac{\partial g}{\partial x_2} &= \frac{x_1}{x_1^2 + x_2^2} & \frac{\partial g}{\partial u} &= 4u \end{aligned}$$

With the variable substitution

$$\begin{aligned} \Delta u &= u - \frac{\pi}{4} \\ \Delta x_1 &= x_1 + 1 \\ \Delta x_2 &= x_2 + 1 \\ \Delta y &= y - \frac{\pi}{4} - \frac{\pi^2}{8}. \end{aligned}$$

the linearized system becomes

$$\begin{pmatrix} \Delta \dot{x}_1 \\ \Delta \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \Delta u$$

$$\Delta y = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} + \pi \Delta u.$$

## 2.5 Blood Doping

The system dynamics are:

$$\begin{aligned} R(k) &= (1 - f) \cdot R(k - 1) + r(k - 1), \quad R(0) = R_{ref} \\ r(k) &= 0.9 \cdot r(k - 1) + u(k - 1), \quad r(0) = f \cdot R_{ref} \\ u(k) &= \begin{cases} 0.025 & \text{if } k = [1 - 19, 41 - 100] \\ 0.05 & \text{if } k = [21 - 40] \end{cases} \end{aligned}$$

The matrices in the Simulink discrete state space block thus are:

$$\begin{aligned} A &= \begin{bmatrix} (1 - 1/120) & 1 \\ 0 & 0.9 \end{bmatrix} \\ B &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ C &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ D &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{aligned}$$

The initial conditions are:

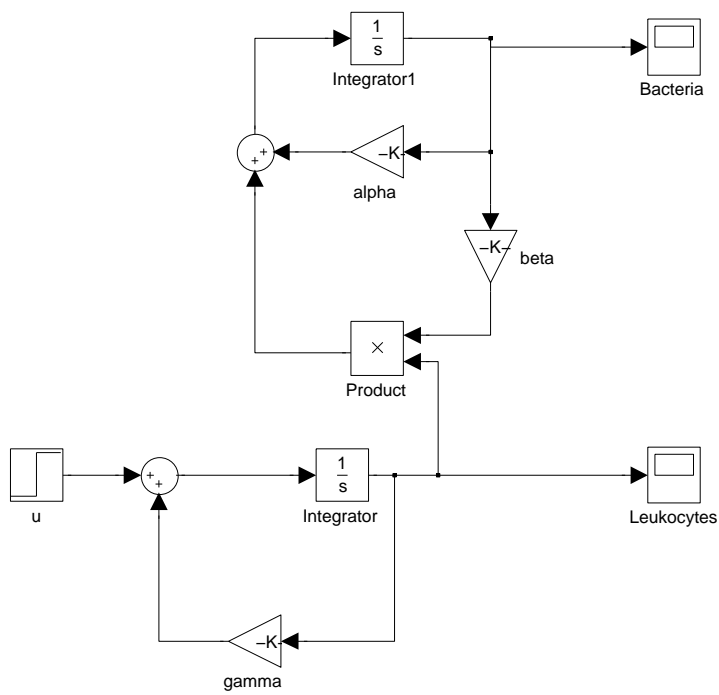
$$x_0 = \begin{bmatrix} 0.25 \cdot 120 \\ 0.25 \end{bmatrix}$$

Define epo in the Matlab workspace as:

```
>> epo(:,1) = 1:100;
>> epo(:,2) = 0.025*ones(100,1);
>> epo(20:40,2) = 0.05;
```

## 2.6 The Simulink model can be seen in Fig. 2.1.

If  $\alpha$  becomes large the bacteria outgrow the neutrophils and uncontrolled bacterial growth occurs.



**Figure 2.1** Simulink model for the Predator-Prey system