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

$$\begin{aligned} \ddot{y} + 7\dot{y} - 3y &= 0 \\ y(0) &= 0 \\ \dot{y}(0) &= 1 \end{aligned}$$

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 govering the production of X and Y:

a.

$$X \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} Y$$

b.

$$X + X \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} Y$$

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

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} C \overset{k_2}{\rightharpoonup} P + E$$

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:

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} C_1 \overset{k_2}{\rightharpoonup} P + E$$

$$I + E \underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}} C_2$$

Table 1.1 Reaction Data for problem 3

Substrate Concentration [mM]	Reaction 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 $V_{max} = -15$[mg/(dl· 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 alhocol 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.

$$V_D = 20 + 0.36m_{BW} - 0.1Y, \quad Men$$
$$V_D = 14 + 0.25m_{BW}, \quad Women$$

Assuming that a 25 year old man of 80 kg consumes a drink containing 2 cl of alcohol (density 800 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.

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 you 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 MATLABs buildt-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

$$\dot{y}_1 = y_2 \tag{0.1}$$
$$\dot{y}_2 = 3y_1(t) - 7y_2 \tag{0.2}$$

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 f as the following function

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

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. `V` is a matrix with two columns, the first column corresponds to $y_1(t) = y(t)$ and the second column corresponds to $y_2(t) = \dot{y}(t)$. `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

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

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 sence.

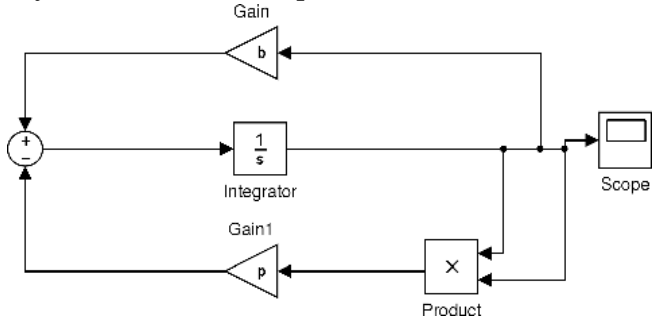
If (0.3) is seen as $S \cdot \begin{pmatrix} a \\ b \end{pmatrix} = T$, the following code will return the values of a and b

```
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.

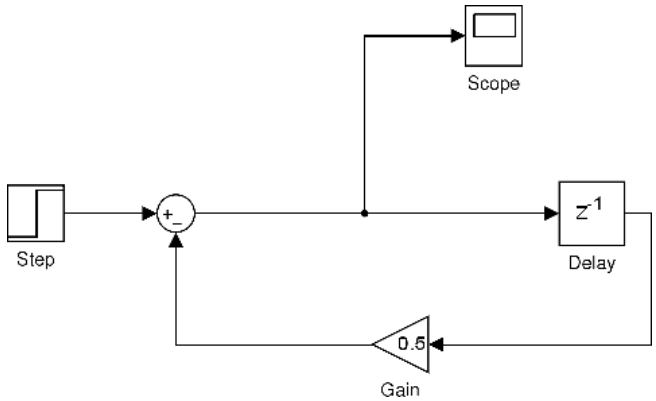
Solve the following exercises using SIMULINK in MATLAB. These exercises are taken from *Exercises in MATLAB/Simulink, Signals and Systems* by Thomas Munther.

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.

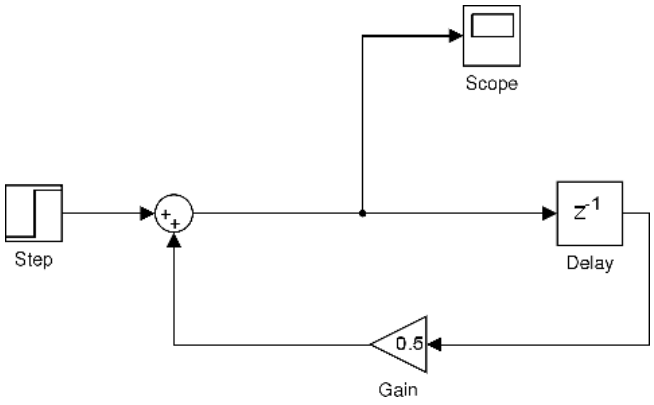


`p` and `b` can be defined in the current workspace. Go to display → blocks and check "Sorted Execution Order". This will numerate the blocks in the order in which they are first activated.

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.


Scope
Step
+
+
z^{-1}
Delay
0.5
Gain

Solutions to Chapter 1. Biochemical Reactions

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

$$\frac{dx}{dt} = -k_1 x + k_{-1} y$$
$$\frac{dy}{dt} = k_1 x - k_{-1} y$$

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

$$\frac{dx}{dt} = -2k_1 x^2 + 2k_{-1} y$$
$$\frac{dy}{dt} = k_1 x^2 - k_{-1} y$$

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 behavoir, see Fig. 1.1. V_{max} and K_m are estimated as shown in the plot.

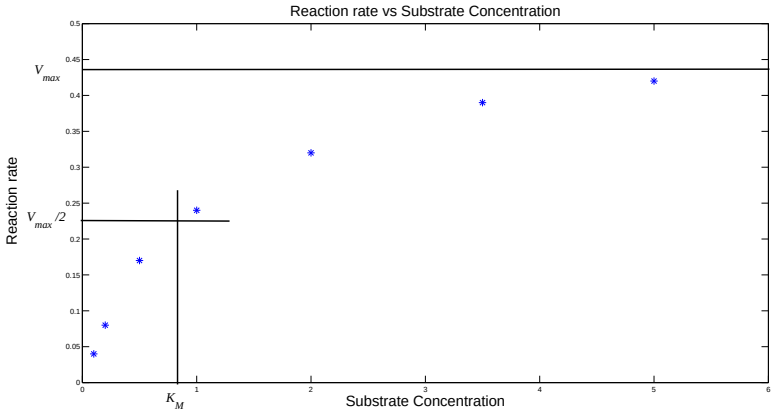


Figure 1.1 Graphical estimation of V_{max} and K_M

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.

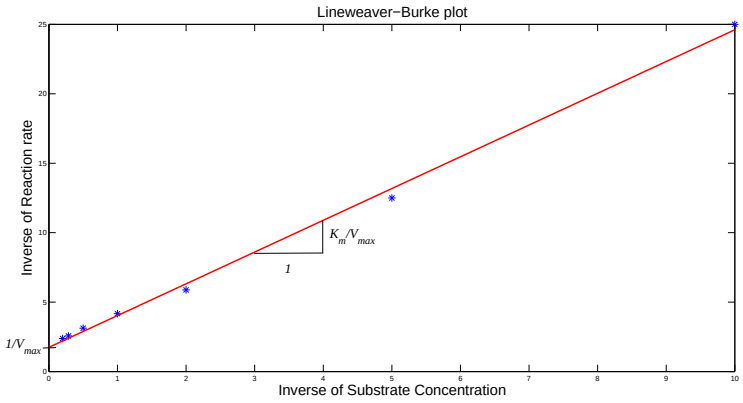


Figure 1.2 Graphical estimation of V_{max} and K_M using the Lineweaver-Burke plot.

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

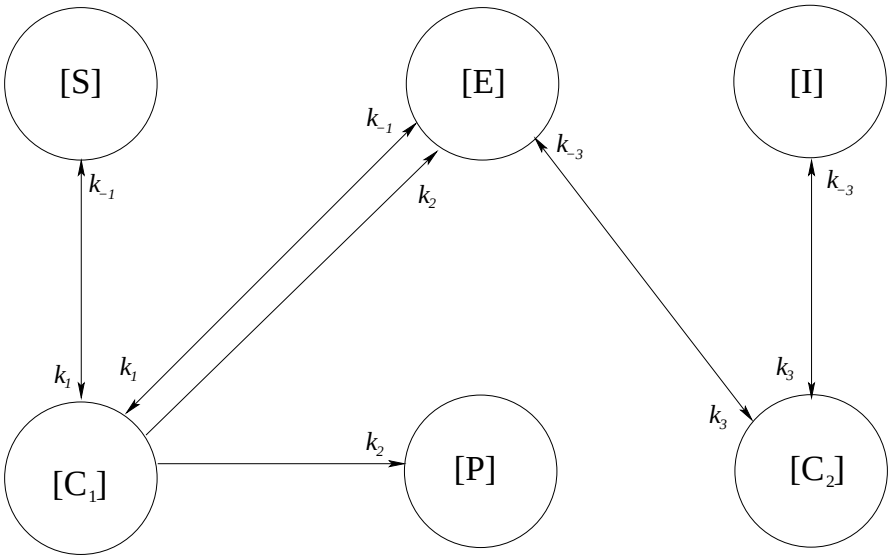


Figure 1.3 Compartment model representation of the enzyme inhibition dynamics.

determine the differential equations governing the reaction dynam-

ics:

$$\frac{d[S]}{dt} = -k_1[S][E] + k_{-1}[C_1] \tag{1.1}$$
$$\frac{d[I]}{dt} = k_{-3}[C_2] - k_3[E][I] \tag{1.2}$$
$$\frac{d[C_1]}{dt} = k_1[S][E] - (k_{-1} + k_2)[C_1] \tag{1.3}$$
$$\frac{d[C_2]}{dt} = k_3[E][I] - k_{-3}[C_2] \tag{1.4}$$
$$\frac{d[E]}{dt} = (k_2 + k_{-1})[C_1] + k_{-3}[C_2] - k_1[S][E] - k_3[E][I] \tag{1.5}$$
$$\frac{d[P]}{dt} = k_2[C_1] \tag{1.6}$$

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

$$[C_1] = \frac{k_1}{k_{-1} + k_2}[S][E] \tag{1.7}$$
$$[C_2] = \frac{k_3}{k_{-3}}[E][I] \tag{1.8}$$

The conservation of enzymatic mass gives

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

Put Eq. (1.6), Eq. (1.8) and Eq. (1.9) together:

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

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)
```

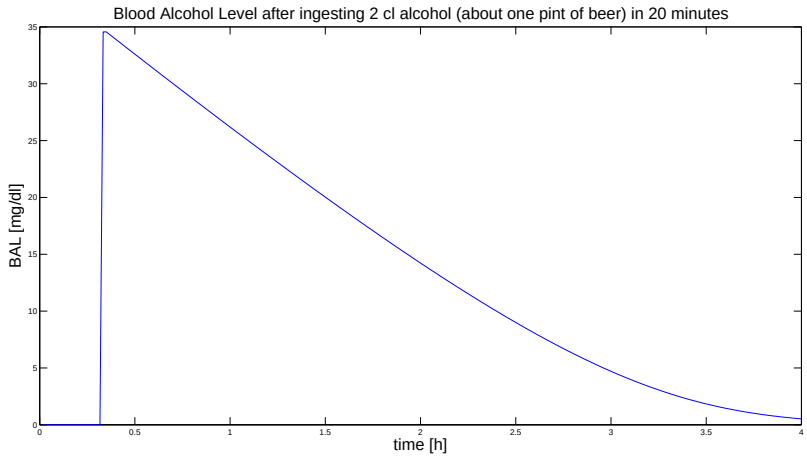


Figure 1.4 Blood alcohol content according to the simulation example.

Running the code generates the plot in Fig. 1.4.

Another possibility is to use MATLABs built in solvers for odrinary 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 metabolization
% 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)
```