Exercise 3 - TTK4130 Modeling and Simulation

Camilla Sterud

1 Problem 1

$$\ddot{x} + c\dot{x} + g(1 - (\frac{1}{x})^{\kappa}) = 0, \quad \kappa = 1.40,\ g = 9.81.$$

1.1 a

The system on state-space form:

$$\underline{\underline{\dot{x}_1 = x_2}}$$
$$\underline{\underline{\dot{x}_2 = g(x_1^{-\kappa} - 1) - cx_2}}$$

1.2 b

Assume $x_1(t_0), x_2(t_0)$ known.

$$\underline{\underline{x_{1,n+1} = x_{1,n} + hx_{2,n}}}$$
$$\underline{\underline{x_{2,n+1} = x_{2,n} + h(g(x_{1,n}^{-\kappa} - 1) - cx_{2,n})}}$$

1.3 c

$$\underline{\underline{\mathbf{k_1} = \begin{bmatrix} k_{11} \\ k_{12} \end{bmatrix} = \begin{bmatrix} x_{1,n} \\ g(x_{1,n}^{-\kappa} - 1) - cx_{2,n} \end{bmatrix}}}$$

$$\underline{\underline{\mathbf{k_2} = \begin{bmatrix} k_{21} \\ k_{22} \end{bmatrix} = \begin{bmatrix} x_{2,n} + \frac{h}{2}k_{12} \\ g((x_{1,n} + \frac{h}{2}k_{11})^{-\kappa} - 1) - c(x_{2,n} + \frac{h}{2}k_{12}) \end{bmatrix}}}$$

$$\underline{\underline{x_{1,n+1} = x_{1,n} + hk_{21}}}$$
$$\underline{\underline{x_{2,n+1} = x_{2,n} + hk_{22}}}$$

1.4 d

For $x = 1 = x_1$, $\dot{x}_1 = x_2 = \dot{x}_2 = 0$. When inserted into the state space model this gives us $0 = g(x_1^{-\kappa} - 1)$, which is correct around $x_1 = 1$. This showes that $x_1 = 1$ is, in fact, a stationairy point. Setting $\dot{x}_1 = f_1$ and $\dot{x}_2 = f_2$, we can linearize about this point using:

$$\Delta\dot{x}_1 = \frac{\partial f_1}{\partial x_1}\bigg|_{\substack{x_1 = 1 \\ \dot{x}_2 = 0}} \Delta x_1 + \frac{\partial f_1}{\partial x_2}\bigg|_{\substack{x_1 = 1 \\ \dot{x}_2 = 0}} \Delta x_2,$$

$$\Delta\dot{x}_2 = \frac{\partial f_2}{\partial x_1}\bigg|_{\substack{x_1 = 1 \\ \dot{x}_2 = 0}} \Delta x_1 + \frac{\partial f_2}{\partial x_2}\bigg|_{\substack{x_1 = 1 \\ \dot{x}_2 = 0}} \Delta x_2.$$

This leaves us with the linearization.

$$\underline{\underline{\Delta \dot{x}_1 = \Delta x_2}}$$

$$\underline{\underline{\Delta \dot{x}_2 = -\kappa g \Delta x_1 - c \Delta x_2}}$$

1.5 e

After being linearized, the system can be writtes as

$$\Delta \dot{x} = \mathbf{A}\Delta x = \begin{bmatrix} 0 & 1 \\ -\kappa g & -c \end{bmatrix} \Delta x.$$

Setting up Euler's method for this we get

$$\Delta x_{n+1} = (\mathbf{I} + h\mathbf{A})\Delta x_n.$$

This will be stable if $|\mathbf{I} + h\mathbf{A}| \leq 1$.

$$\left| \begin{bmatrix} 1 & h \\ -\kappa g h & 1 - hc \end{bmatrix} \right| = 1 - hc + \kappa g h^2 \leq 1$$

In the first case, $c = 0$, this will never be stable, as it requires $h = 0$. With $c = 2\sqrt{g\kappa}$ the method is stable for any $\underline{\underline{h \leq \frac{2}{\sqrt{g\kappa}}}}$.

2 Problem 2

$$L\frac{di}{dt} + Ri = u, \quad t > 0,\ i(0) = i_0.$$

2.1 a

The enery stored in the circuit is equal to the enery stored in the inductor. This means that the storage funtion for this system is

$$V = \frac{1}{2}Li^2 \Rightarrow \dot{V} = Li\frac{di}{dt} = iu - Ri^2.$$

The storage function is monotonically decreasing for $u = 0$, which means that the energy stored in the system will end up being zero. The energy can only be zero if the current is zero (given by the equation for the energy stored in an inductor), so the system is clearly asymptotically stable for $u = 0$ (it is in fact stable for all $u < \infty$).
I don't have time to sketch this right now, but the current will always decrease to zero for $u = 0$.

2.2 b

$e(t) = i(t) - i_{ref}$, $\dot{e} = \frac{1}{L}(u - Ri)$.

$$E(t) = \frac{1}{2}Le^2(t) \Rightarrow \dot{E}(t) = Le\dot{e} = ue - Rie.$$

$\lim_{t\to\infty} \dot{E}(t) = 0$, since E must be constant if i is constant. $\lim_{t\to\infty} e(u - Ri) \Rightarrow$ $\underline{\underline{u = Ri_{ref}}}$.

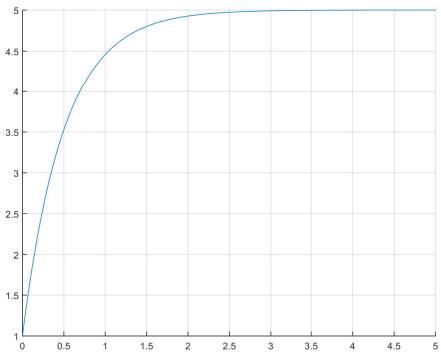
2.3 c

```
clc; clear all; hold on; grid on;

h = 0.01;
t = 0:h:5;

i = zeros (1,length(t));
i(1) = 1;

R = 2;
L = 1;
i_ref = 5;
u = R*i_ref;

f = @(I) (1/L)*(u - R*I);

for j = 1:(length(t) - 1)

    k_1 = f(i(j));
    k_2 = f(i(j) + h*0.5*k_1);
    k_3 = f(i(j) + h*0.5*k_2);
    k_4 = f(i(j) + h*k_3);

    i(j+1) = i(j) + h*((1/6)*k_1 + (1/3)*k_2 + (1/3)*k_3 + (1/6)*k_4);

end

plot(t,i);
```

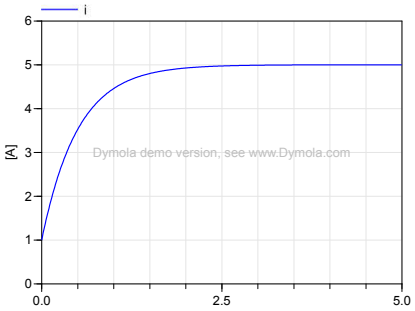
Figur 1: Code for solving the system with Explicit Runge-Kutta of order 4 in MATLAB.

2.4 d

The MATLAB and Dymola plots look suspiciously alike.



Figur 2: The system solved with the MATLAB code.



Figur 3: The system solved with RK4 in Dymola.