P1 Consider the third order differential equation below:

\[ \frac{d^3y}{dt^3} + 1.2 \frac{d^2y}{dt^2} - 2.7 \frac{dy}{dt} + 15.1y = 0 \]

(a) Write it as a system of coupled ODEs by introducing new variables \( w = \frac{d^2y}{dt^2} \), and \( x = \frac{dy}{dt} \).

(b) The resulting coupled ODE is a linear system:

\[
\frac{d}{dt} \begin{pmatrix} w \\ x \\ y \end{pmatrix} = A \begin{pmatrix} w \\ x \\ y \end{pmatrix} + \vec{b}
\]

What are the matrices \( A, \vec{b} \)?

(c) Calculate the matrix exponentials \( e^A, e^{2A} \) and \( e^{3A} \).

(d) For initial conditions given by \( w = 0.2, x = -0.1, \) and \( y = 1, \) write down the value of \( y \) at times \( t = 1, 2 \) and 3 second.

(e) Check whether the system is stable by computing its eigenvalues.

Note: You are free to use MATLAB (tm), Octave or Python to compute matrix exponentials and/or the eigenvalues. The Matlab/Octave function for matrix exponential is \texttt{expm} (and not \texttt{exp}).

\begin{verbatim}
(a) The coupled ODEs are
\[
\frac{dw}{dt} = -1.2w + 2.7x - 15.1y
\]
\[
\frac{dx}{dt} = w
\]
\[
\frac{dy}{dt} = x
\]

(b) The matrices are
\[
A = \begin{bmatrix}
-1.2 & 2.7 & -15.1 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix}
\]
\[
b = \begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix}
\]

(c) Using matlab we obtain
\[
>> \text{expm}(A)
\]
\[
ans =
\begin{bmatrix}
-0.5359 & -3.8605 & -6.5257 \\
0.4322 & -0.0173 & -5.0274 \\
0.3329 & 0.8317 & -0.9162
\end{bmatrix}
\]
\[
>> \text{expm}(2*A)
\]
\[
ans =
\begin{bmatrix}
-0.5359 & -3.8605 & -6.5257 \\
0.4322 & -0.0173 & -5.0274 \\
0.3329 & 0.8317 & -0.9162
\end{bmatrix}
\]
\end{verbatim}
\[ A = \begin{bmatrix} -3.5539 & -3.2922 & 28.8841 \\ -1.9129 & -5.8494 & 1.8725 \\ -0.1240 & -2.0617 & -5.5145 \end{bmatrix} \]

\[ \expm(3A) \]

\text{ans} =

\[ \begin{bmatrix} 10.0983 & 37.7997 & 13.2798 \\ -0.8795 & 9.0429 & 40.1743 \\ -2.6605 & -4.0721 & 16.2264 \end{bmatrix} \]

\[ \expm(3A) \]

\text{(c) Once again, we use matlab to compute the states of the system, to obtain } y(1) = -0.9328, y(2) = -5.3332, y(3) = 16.1015.

\[ p0 = [0.2; -0.1; 1] \]

\text{p0} =

\[ \begin{bmatrix} 0.2000 \\ -0.1000 \\ 1.0000 \end{bmatrix} \]

\[ \expm(A) \ast p0 \]

\text{ans} =

\[ \begin{bmatrix} -6.2469 \\ -4.9392 \\ -0.9328 \end{bmatrix} \]

\[ \expm(2A) \ast p0 \]

\text{ans} =

\[ \begin{bmatrix} 28.5025 \\ 2.0749 \\ -5.3332 \end{bmatrix} \]

\[ \expm(3A) \ast p0 \]

\text{ans} =

\[ \begin{bmatrix} 11.5195 \\ 39.0941 \\ 16.1015 \end{bmatrix} \]
The eigenvalues of $A$ are $-3.3507, 1.0754 + 1.8303i, 1.0754 - 1.8303i$. Two eigenvalues lie on the right half of the complex plane and thus, the system is unstable.

P2 Consider the Vanderpol oscillator given by the system of coupled ODEs:

\[
\begin{align*}
\frac{dx}{dt} &= y \\
\frac{dy}{dt} &= (1 - x^2)y - x
\end{align*}
\]

1. Find all the possible equilibria of this system. (**Hint:** set the RHS of the ODE to 0 and solve for $x, y$).

2. Using a Runge-Kutta solver (**ode23** function in MATLAB and equivalents) solve the ODE for various initial conditions randomly chosen inside the box $x \in [-1, 1]$ and $y \in [-1, 1]$ for time $t \in [0, 100]$ units. Plot the resulting trajectories.

3. Use the trajectories to decide if the system is stable or unstable at each of the equilibria found.

4. Draw a Simulink diagram for the Vanderpol system. Allow the simulation to set various values for $x(0), y(0)$ and be able to plot the result.

1. The equilibria of the system are found by setting the RHS of the ODEs to 0.

\[
y = 0, \quad (1 - x^2)y - x = 0
\]

Solving, we obtain the only equilibrium as $x = y = 0$.

2. Attached file **vdpSimulation.m** generates the trajectories shown below.

3. The system is unstable.

4. See the attached diagram **vdpModel.slx** and script **simulateVdpModel.m** that runs this model for various initial conditions for the integrators. The diagram below shows time plots for $x(t)$ and $y(t)$ for various initial values through simulating the Simulink diagram.
P3 Consider the following control system:

\[
\frac{d\vec{x}}{dt} = \begin{pmatrix} 1 & -1 \\ 1 & -2 \end{pmatrix} \vec{x} + \begin{pmatrix} 1 \\ 0.2 \end{pmatrix} u,
\]

wherein \( A = \begin{pmatrix} 1 & -1 \\ 1 & -2 \end{pmatrix} \) and \( B = \begin{pmatrix} 1 \\ 0.2 \end{pmatrix} \) with \( u \) as a single input.

1. Find a static state feedback stabilizing control law of the form \( u = K\vec{x} \) using the idea of placing "eigenvalues" of the state space system as shown in the class. For this problem, the eigenvalues of the closed loop should be at \( \lambda_1 = -1, \lambda_2 = -2 \).

2. Implement a PID controller that will attempt to stabilize the value of \( x_1 \) to a reference point \( x_1 = 3 \). For this controller, do not use the values of \( x_2 \) in your feedback loop. Report the values of the various gains for this controller and show a plot of the closed loop simulation in Simulink.

1. With a feedback law \( u = K\vec{x} \), the closed loop dynamics become

\[
\frac{d\vec{x}}{dt} = (A + BK)\vec{x}.
\]

Writing \( K = [k_1 \ k_2] \), we obtain:

\[
A + BK = \begin{pmatrix} 1 + k_1 & -1 + k_2 \\ 1 + 0.2k_1 & -2 + 0.2k_2 \end{pmatrix}
\]

We wish the eigenvalues to be \( \lambda = -1, -2 \). Therefore, we equate the sum of eigenvalues to the trace of the matrix.

\[
1 + k_1 - 2 + 0.2k_2 = -1 + -2 = -3, \text{ simplifies to } k_1 = -2 - 0.2k_2.
\]

Likewise, we have the determinant to be the product of eigenvalues.

\[
(1 + k_1)(-2 + 0.2k_2) - (1 + 0.2k_1)(-1 + k_2) = -2 \times -1 = 2
\]

Simplifying, we obtain

\[
1.8k_1 + 0.8 = -3
\]

Solving, we obtain \( k_1 = -2.2727 \) and \( k_2 = 1.3636 \).

2. Simulink model is attached prob3assign5.slx.
In this assignment you will model the different pieces of a simple artificial pancreas setup that controls blood glucose levels in people with type-1 diabetes.

The human insulin-glucose response is modeled by the Bergman minimal model with three state variables \((G, I, X)\) wherein \(G\) is plasma glucose, concentration above the basal value \(G_B\) (units: mmol/L), and \(I\) is the plasma insulin concentration above the basal value \(I_B\) (units: U/L). \(X\) is the insulin concentration in an interstitial chamber. Note that time is measured in minutes for this model. The ODEs are

\[
\begin{align*}
\frac{dG}{dt} &= -p_1 G - X(G + G_B) + u_g(t) \\
\frac{dX}{dt} &= -p_2 X + p_3 I \\
\frac{dI}{dt} &= -n(I + I_B) + \frac{1}{V_I} u_i(t).
\end{align*}
\]

Typical parameter values are \(p_1 = 0.01, p_2 = 0.025, p_3 = 1.3 \times 10^{-5}, V_I = 12, n = 0.093, G_B = 4.5, I_B = 15\).

The functions \(u_g(t)\) and \(u_i(t)\) model the infusion of glucose and insulin into the bloodstream. Specifically, \(u_g(t)\) is the rate at which glucose is appearing, while \(u_i\) is the rate at which insulin is appearing.

The initial values are

\[
G(0) = 0, X(0) = 0, I(0) = 0.05
\]

(A) Draw a Simulink subsystem with two inputs: \(u_g, u_i\) for the meal glucose and meal insulin, respectively and one output \(G(t)\).

(B) The control logic is a switched feedback controller with the following control law for the rate at which insulin is infused.

\[
u_i(t) = \begin{cases} 
\frac{25}{3} & G(t) \leq 4 \\
\frac{25}{3} (G(t) - 3) & G(t) \in [4, 8] \\
\frac{125}{3} & G(t) \geq 8
\end{cases}
\]

Model this in a control logic subsystem.

(C) The meal glucose model captures the rate at which the carbohydrates in a meal appear in the blood stream of the patient. A typical rate of appearance curve that is measured using trace-meal studies looks as follows:
<table>
<thead>
<tr>
<th>Time Interval after meal (mins)</th>
<th>% of glucose appearing in interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 10</td>
<td>0 %</td>
</tr>
<tr>
<td>10 - 20</td>
<td>7 %</td>
</tr>
<tr>
<td>20 - 30</td>
<td>14 %</td>
</tr>
<tr>
<td>30 - 40</td>
<td>21 %</td>
</tr>
<tr>
<td>40 - 50</td>
<td>18 %</td>
</tr>
<tr>
<td>50 - 60</td>
<td>7 %</td>
</tr>
<tr>
<td>60 - 70</td>
<td>3 %</td>
</tr>
<tr>
<td>70 +</td>
<td>0 %</td>
</tr>
</tbody>
</table>

For instance, suppose a patient eats a meal with 110 gms of carbs at time $T$, then we can say that the value of $u_g(t)$ at time $T + 55$ is given by $\frac{7\% \times 110}{10} = 0.77 gms/min$.

Given a meal specified by gms of carbs + time of meal (minute after simulation start), implement a meal glucose module that generates the value of $u_g(t)$ for that meal using the table above.

**D** Close the loop and simulate the closed loop system for different meal sizes at time $t = 20$. The meal sizes to be tried include \{10gms, 20gms, 40gms, 80gms, 110gms, 125gms\}. Simulate each scenario for $t \in [0, 240]$ mins.

For each of the meal scenarios, compute the maximum and minimum values achieved for $G(t)$ from simulation, the glucose output.

See the file `igModel.slx` and attached script `runIGModel.m`. We obtain the following plot: