Problem 1 The continuous-time system

\[ Y(s) = \frac{1}{s+1} U(s) \]

is to be controlled by a PI-controller. For a continuous-time PI-controller,

\[ F^c_{PI}(s) = C_p \frac{C_i}{s}, \]

the corresponding closed loop system will be stable for all \( C_p, C_i > 0 \) (you may verify this). However, the controller must be implemented in a computer, and hence the PI-controller must be converted to a discrete-time controller. One “natural” way to implement a discrete-time PI-controller is

\[
\begin{align*}
    e(kh) &= r(kh) - y(kh), \\
    s(kh) &= s(kh-h) + e(kh), \\
    u(kh) &= K_p e(kh) + hK_i s(kh)
\end{align*}
\]

\[
\begin{align*}
    U(z) &= F^d_{PI}(z) E(z), \\
    F^d_{PI}(z) &= K_p + K_i \frac{h z}{z-1},
\end{align*}
\]

where \( h \) is the sampling period.

(a) Assume that the PI-controller is tuned in continuous time, so that some feasible \( C_p \) and \( C_i \) are found. Convert \( F^d_{PI}(s) \) to \( F^d_{PI}(z) \) using the following approximations of the derivative:

- Forward difference: \( s = \frac{z-1}{h} \)
- Backward difference: \( s = \frac{z+1}{h} \)
- Tustin’s approximation: \( s = \frac{2z-1}{h z + 1} \)

Express \( K_p \) and \( K_i \) in \( F^d_{PI}(z) \) in terms of \( C_p, C_i \) and \( h \) for each case. (1 pt)
Next the stability of the closed loop system will be examined, and for simplicity this analysis will be performed in discrete time.

(b) Determine the sampled version of the system (using zero order hold), and show that the corresponding transfer function can be written as

\[ G^d(z) = \frac{1 - \alpha}{z - \alpha}. \]

Determine \( \alpha \) in terms of the sampling period \( h \).

(c) Analyse the stability of the closed loop system, using \( G^d(z) \) and \( F^d_{\text{PI}}(z) \). Give conditions on \( K_p, K_i \) and \( h \) for stability.

\textbf{Hint:} All roots of \( z^2 + az + b = 0 \) lies within the unit circle if and only if all of the following conditions are fulfilled: (i) \( b < 1 \), (ii) \( a + b > -1 \), (iii) \( a - b < 1 \). (You may verify this.)

(d) Assume that the PI-controller is implemented in Matlab. Fill in the missing lines of code in the following Matlab code snippet, so that the controller works properly:

```matlab
% Initiate controller time
t = GetTime;
while true
  % Increment controller time to next sample instant
  t = t + h;
  % Wait for next sample instant
  WaitUntil(t);
  % Read present setpoint value on A/D-converter, channel 0
  r = ReadAD(0);
  % Read present output value on A/D-converter, channel 1
  y = ReadAD(1);
  % Compute the control input //FILL IN HERE!//
  u = ComputeControlInput(r, y);
  % Write control input u on D/A-converter, channel 0
  WriteDA(0, u);
end % Control loop ends here
```

\textbf{Problem 2} An inverted pendulum has the (linearized) continuous-time model

\[ \dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t), \]
\[ y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(t), \]

where \( u \) is the applied torque about the hinged (sv. "ledade") lower end of the pendulum, and \( y \) is the angular deviation from the vertical line. Your task is to design a control law, \( u(kh) = -L \dot{x}(kh) + mr(kh) \), such that the inverted pendulum is stabilized. The system is sampled with the sampling period \( h = 0.1 \) seconds.

(a) Determine the sampled discrete-time model of the system (using zero order hold). That is, determine \( F, G \) and \( H \) in the state space model

\[ x(kh + h) = F x(kh) + G u(kh), \]
\[ y(kh) = H x(kh). \]
(b) Design an observer-based state feedback controller such that the control error (e.g. for the step response) decays like $e^{-t}$, and that the damping ratio is $\zeta \approx 0.7$. That is, determine $L$ and $m$ in the control law, and the observer gain $K$. A thorough motivation is required.

*Hint:* Use pole placement; choose feasible continuous-time poles, and then “translate” these to corresponding discrete-time poles. (2 pts)

(c) What is the transfer function from $r(kh)$ to $y(kh)$ with your controller? (1 pt)

(d) Simulate and plot the step response in Matlab. Verify that the requirements are met. (1 pt)
**Problem 1** Consider the continuous-time system

\[
\dot{x}(t) = \begin{bmatrix} -\alpha & 0 \\ 1 & -\alpha \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} v(t), \quad \alpha > 0
\]

\[
y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(t),
\]

where \( v(t) \) is white noise with intensity \( \Phi_v(\omega) = R_v = 1 \).

(a) Determine the covariance matrix for the state vector, \( \Pi_x = Ex(t)x(t)^T \). Also determine the spectrum \( \Phi_y(\omega) \) for the output \( y \). (2 pts)

(b) If the system above is sampled with the sampling period \( h \), the discrete-time system

\[
x(kh + h) = \begin{bmatrix} \beta & 0 \\ h \beta & \beta \end{bmatrix} x(kh) + w(kh), \quad Ew(kh)w(kh)^T = R_w = \begin{bmatrix} r_1 & r_{12} \\ r_{12} & r_2 \end{bmatrix},
\]

\[
y(kh) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(kh),
\]

is obtained, where \( \beta = e^{-\alpha h} \). Determine the covariance matrix for the sampled state vector, \( \Pi_x = Ex(kh)x(kh)^T \), expressed in \( \beta, h, r_1, r_{12} \) and \( r_2 \). (1 pt)

(You may compare with the result in (a) — use

\[
r_1 = \frac{1 - \beta^2}{2\alpha}, \quad r_{12} = \frac{1 - (1 + 2\alpha h)\beta^2}{4\alpha^2}, \quad r_2 = \frac{1 - (1 + 2\alpha h + 2\alpha^2 h^2)\beta^2}{4\alpha^3}
\]

— the results should be identical.)

**Problem 2** The block diagram below represents a certain industrial process.

The control objective is to keep the output \( y \) as close as possible to the constant setpoint \( r \). The output must not exceed a critical level for more than 1% of the time on average — this is a strict requirement. On the other hand, the lower the output is kept, the more expensive is the operation of the process. Thus, the setpoint is chosen as close to the critical level as possible, without violating the 1% of the time requirement.

The system is modeled as

\[
\dot{x}(t) = \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} (u(t) + d(t)),
\]

\[
y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(t),
\]
where $d$ is a resonant random disturbance. It is found that the disturbance can be modeled as

$$
\dot{z}(t) = \begin{bmatrix} -0.4 & -4.04 \\ 1 & 0 \end{bmatrix} z(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} v(t),
$$

$$
d(t) = \begin{bmatrix} 1 & 0.2 \end{bmatrix} z(t),
$$

where $v$ is white noise with the intensity $\Phi_v(\omega) = R_v = 1$.

An initial design utilizes proportional control in order to stabilize the system (the control law is $u = r - y$, chosen to give reasonable stability margins). It is then found that the setpoint must be 0.07 units below the critical level — see the figure below, showing a simulation of this case.

Your task is to design a controller that performs better than the proportional controller above, and thereby save money for your company (every 0.01 unit the setpoint gets closer to the critical level reduces the cost by SEK 100 000 monthly). Use the LQG design technique:

- Minimize the output $y$ (assuming $r = 0$)
- The constraint $|u(t)| \leq 0.30 \ \forall t$ must not be violated (due to physical limitations)

Your choices of the design parameters ($Q_1, Q_2$ etc) should be well motivated.

(a) Combine the models of the system and the disturbance into an augmented state space model, with $u$ and $v$ as inputs and $y$ as output. \hspace{1cm} (1 \text{ pt})

(b) Design the controller, that is find a state feedback gain $L$ and a Kalman filter gain $K$ such that the requirements are fulfilled. \hspace{1cm} (4 \text{ pts})

(c) Simulate the obtained closed loop system and plot the results to verify that the requirements are met. Use the pre-defined realizations $d1$, $d2$ and $d3$ of the disturbance $d$ in the simulations — these are available in a .mat-file at Studentportalen. \hspace{1cm} (2 \text{ pts})

How much closer to the critical level can the setpoint be with your controller? The distance should be reduced with at least 50% compared to the proportional control above.