Exercises

1. (6 points) In this exercise, you will approximate a second-order ODE describing the dynamics of a pendulum using a discrete-time LSSM with two hidden states. The pendulum is governed by the following second-order ODE:

$$\frac{d^2\phi}{dt^2} + c\frac{d\phi}{dt} + \sin\phi = u(t)$$

where $\phi(t)$ is the angle of the pendulum from the vertical at time t, $u(t)$ is an external input torque applied to the pendulum, and c is a friction coefficient. Your goal is to transform this continuous-time system into a discrete-time state-space model using a small discretization step Δt. To do so, follow these steps:

(a) **Define the State Variables**: To convert this second-order ODE into a first-order system, define the following state variables:

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \phi(t) \\ \omega(t) \end{bmatrix},$$

where $\phi(t)$ is the angle of displacement and $\omega(t) = \frac{d\phi}{dt}$ is the angular velocity. Using this state variable, rewrite the pendulum equation as a system of two first-order ODEs:

$$\begin{cases} \frac{d\phi}{dt} = f_1(\omega, \phi, u(t)) = \omega \\ \frac{d\omega}{dt} = f_2(\omega, \phi, u(t)) = ? \end{cases}$$

Find the explicit form of the function $f_2(\omega, \phi, u(t))$.

(b) **Formulate the State Equations**: Now that we have expressed the system as a first-order system of ODEs, write it in the following state-space form:

$$\begin{bmatrix} \frac{d\phi}{dt} \\ \frac{d\omega}{dt} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{bmatrix}}_{\frac{d\mathbf{x}}{dt}} = \underbrace{\begin{bmatrix} f_1(x_2, x_1, u(t)) \\ f_2(x_2, x_1, u(t)) \end{bmatrix}}_{\mathbf{f}(\mathbf{x}, u(t))}.$$

Find the explicit formula for the vector on the right-hand side.

(c) **Discretize the Nonlinear System**: To transform this continuous-time system into a discrete-time model, use Euler's method with a discretization step Δt, i.e.,

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \Delta t \cdot \mathbf{f}(\mathbf{x}(t), u(t)).$$

Define the discrete-time state as $\mathbf{x}_k = \mathbf{x}(k \cdot \Delta t)$ and input $u_k = u(k \cdot \Delta t)$. The state update equation becomes:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot f(\mathbf{x}_k, u_k),$$

which can be written as the following vector equality:

$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} = \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + \Delta t \cdot \begin{bmatrix} ? \\ ? \end{bmatrix},$$

Fill in the entries of the last vector. Your answer should be a function of $x_{1,k}$, $x_{2,k}$, and u_k.

(d) Implement the resulting LSSM in Python and plot the evolution of x_1 as a function of k. In your simulations, set the initial angle to $x_{0,1} = \pi - 0.1$ (close to the vertical position) and the initial angular velocity to zero, i.e., $x_{0,2} = 0$. Set the inputs to be:

```
# Parameters
delta_t = 0.01  # Time step
num_steps = 10000  # Number of time steps to simulate
c = 0.1  # Friction coefficient

# Sinusoidal input
amplitude = 0
frequency = 0.5
u = amplitude*np.sin(frequency*np.arange(num_steps))
```

Provide a physical interpretation of your observations.

(e) Using the same parameters, include a plot where the x-axis is $x_{k,1}$ and the y-axis is $x_{k,2}$. This type of plot is called a **phase space** plot. Provide a physical interpretation of your observations.

(f) In a pendulum, you can induce complex behavior by choosing the right set of parameters. For example, plot the phase space with the same initial conditions and the following set of parameters:

```
# Parameters
delta_t = 0.01  # Time step
num_steps = 100000  # Number of time steps to simulate
c = 0.01  # Friction coefficient

# Sinusoidal input
amplitude = 1
frequency = 0.01
u = amplitude*np.sin(frequency*np.arange(num_steps))
```

What does the pendulum do in physical terms? Does it stabilize to an equilibrium point? Does it oscillate periodically? Does it oscillate irregularly?

2. (5 points) In this exercise, you will build a simple Hidden Markov Model (HMM) to model the operational states of a machine and detect potential faults based on observable signals. Suppose the machine can be in one of three hidden states at any given time: **Normal Operation** (State 1), **Minor Fault** (State 2), or **Severe Fault** (State 3). Although these states are hidden, they influence observable sensor readings. The sensor outputs one of three observable levels of vibration: **Low Vibration** (Observation 1), **Medium Vibration** (Observation 2), or **High Vibration** (Observation 3). Your goal is to analyze an HMM that represents this system, then simulate the model's response to a given observation sequence, and interpret the results. The initial distribution, state transition matrix, and emission matrix of the HMM are as follows:

$$\boldsymbol{\pi} = \begin{bmatrix} 0.8 & 0.15 & 0.05 \end{bmatrix}^\intercal, \ \mathbf{A} = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.05 & 0.15 & 0.8 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 0.9 & 0.08 & 0.02 \\ 0.3 & 0.6 & 0.1 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}.$$

Follow the steps below to construct the model.

(a) **Simulate the Observation Sequence**: Using these HMM parameters, simulate the hidden states and corresponding observations over time. Write a function in `Python` that generates a sequence of hidden states and observations based on the transition and emission probabilities.

(b) **Decode the Observation Sequence**: Build a function in `Python` that, for each observation, identifies the most likely hidden state by finding the highest emission probability in the emission matrix for each state. For example, if you observe a high vibration level, the most likely hidden state would be **Severe Fault**, since that state has the highest probability of producing this observation.

(c) **Apply the Model to a Given Observation Sequence**: Using the following observation sequence:

```
[1, 1, 2, 3, 3, 2, 1, 1, 2, 3]
```

use your decoding function in `Python` to estimate the hidden state sequence that most likely produced these observations.

(d) **Probability of the Hidden State Sequence**: Using the HMM parameters, calculate in `Python` the probability of the hidden state sequence you estimated in the previous step.

3. (5 points) Consider a dynamical system characterized by the following system

of recursions:

$$\begin{aligned} x_{1,k+1} &= x_{1,k}/2 + x_{2,k}/4 + x_{3,k}/8 + \sin(3k), \\ x_{2,k+1} &= x_{1,k}, \\ x_{3,k+1} &= x_{2,k}, \end{aligned}$$

with the observation equation:

$$y_k = x_{1,k} + x_{2,k} + x_{3,k}.$$

(a) Write the system in matrix form.

(b) Is the system stable? *Hint*: The state matrix is stable if its eigenvalues are inside the unit circle.

(c) Determine whether the LSSM is in controllable or observable canonical form. *Hint*: Refer to Example **??** for guidance on canonical forms.

(d) Simulate the system in Python for 100 time steps, using the initial condition $\mathbf{x}_0 = \mathbf{0}$. Plot the evolution of y_k over time. Verify whether the system is stable.

(e) Write the state and output equations of the system in observable canonical form.

(f) Simulate the transformed system in Python for 100 time steps, using the initial condition $\boldsymbol{\xi}_0 = \mathbf{0}$. Plot the evolution of the output y_k over time.

(g) Compare the simulations of the original and transformed systems. Justify your observations.

4. (4 points) Consider an LSSM with a single hidden state, governed by the following equations:

$$\begin{aligned} x_{k+1} &= x_k/2 + u_k, \\ y_k &= x_k. \end{aligned}$$

Answer the questions below:

(a) Starting with the initial condition $x_0 = 0$, derive an expression for the output y_k in terms of x_0 and the sequence of inputs $\{u_0, u_1, \ldots, u_{k-1}\}$.

(b) Suppose the input u_k is a step function, defined as $u_k = 1$ for all $k \geq 0$ and 0 otherwise. Using your answer from the previous question, derive an expression for the output sequence y_k under this step input.

(c) Compute the sequence of Markov parameters H_i for this system.

(d) Using the Markov parameters $\{H_0, H_1, H_2, \ldots\}$ and the step function as the input sequence, compute the convolution of the Markov parameters with the input. Express the resulting sequence $\{y_0, y_1, y_2, \ldots\}$ as a function of k.

Define a new state variable ξ_k by the invertible transformation $\xi_k = \frac{x_k}{2}$.

(e) Derive the transformed system matrices $\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D}$ and the transformed initial condition ξ_0 for the new state-space model.

(f) Using the step function as an input, compute the output y_k of the transformed LSSM for $k = 0, 1, 2, \ldots$

5. (3 points) Consider an LSSM model with system matrices (A, B, C, D). Prove the following identities using the component-wise definition of tensor Jacobians:

$$\frac{\partial \|\mathbf{y}_k - \widehat{\mathbf{y}}_k\|_2^2}{\partial \widehat{\mathbf{y}}_k} = 2(\widehat{\mathbf{y}}_k - \mathbf{y}_k)^\intercal, \; \frac{\partial \widehat{\mathbf{y}}_k}{\partial \mathbf{x}_k} = C \text{ and } \frac{\partial \mathbf{x}_{j+1}}{\partial \mathbf{x}_j} = A \text{ for all } j \geq 1.$$

6. (3 points) Given 3 matrices U, X and V of appropriate dimensions, prove that:

$$\frac{\partial \left(UXV\right)}{\partial X} = U \otimes V^\intercal$$

7. Consider an LSSM model with state matrix A. Prove that

$$\frac{\partial \mathbf{x}_{k+1}}{\partial A} = \mathbb{I}^{(2)} \otimes \mathbf{x}_k + A{:}\frac{\partial \mathbf{x}_k}{\partial A},$$

where ":" is the symbol for the contraction product using a single index.

8. Explain in your own words the vanishing and exploding gradient problem. What techniques have we covered in this chapter to alleviate this issue?