MTE-546 Multi-sensor Data Fusion
Final Exam

Release time: Thursday April 11, 2024 12:00 pm
Deadline Thursday April 18, 2024 11:59 pm

Mechanical and Mechatronics Engineering Department
University of Waterloo, **Instructor: Arash Arami**

Instruction:

The following questions are the final take-home exam for the Winter 2024 MTE 546 Multi-sensor Data Fusion offering. Each student is expected to complete the required analysis, simulation, and discussion for each question. Ensure that all assumptions and conclusions are explained with engineering judgment. If you use ideas from other papers, you need to cite them in your exam report properly. Using another student's solutions or communication with third parties will constitute an academic offense and be addressed according to departmental guidelines. In addition, any communication with someone other than the course instructor will be viewed as an academic offense under UW Police 71.

Make sure all plots are legible (use fonts of 12 or larger), numbered, have a caption and their axis are defined and labeled properly.

If you feel a required assumption is .missing, make that assumption, state it clearly in your solution, and continue with it.

The problem set and its accompanied data are confidential and should not be shared with others even after the exam period.

3 Problems, one week

Problem		Mark	Page
1		**40 (+10)**	**2-4**
2		**30**	**5-6**
3		**30(+10)**	**7-8**
Total:		**100(+20)**	

Problem 1: Extended Kalman Filter to fuse noisy measurement and process model (40pt)

Some fishermen want to know where to place their nets in the water to catch fish. Their boat is equipped with a sonar mapping device that locates fish in the water. The sonar sensor estimates the distance between the sensor and fish in the water and the angle between the sensor's line of sight and the fish. Design a tracking system that uses an Extended Kalman Filter to estimate the location of every fish in the surrounding water.

The reference coordinate frame, shown in Figure 1, is defined as follows:

- The sensor is located at the origin of the coordinate frame.
- The x-axis is vertical with respect to the water, positive x-values are into the water.
- The y-axis is horizontal with respect to the water, positive y-values are to the right of the boat.
- The sensor points along the +x-axis such that an angle of 0 deg is directly below the boat.
- Positive sensor angles correspond to counterclockwise rotation.

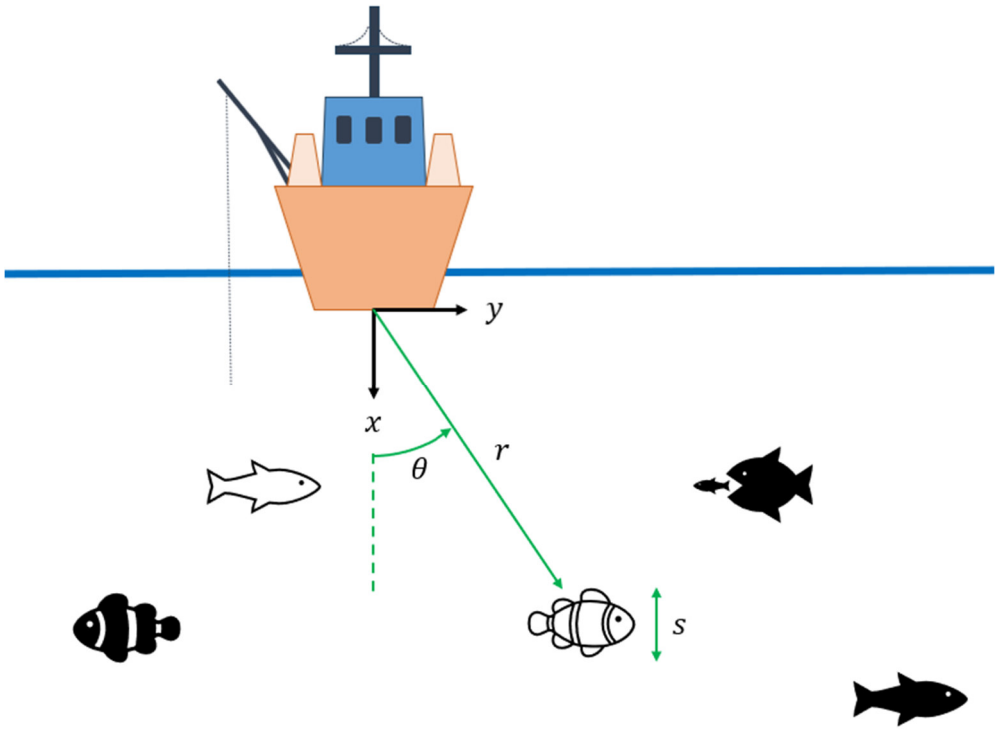

Figure 1: Diagram of the Fishing Boat and Sensor Coordinate Frame

The sensor scans the water, sweeping through a range of angles from -180 deg to +180 deg. If any fish are detected, the angle between the fish and the x-axis, the distance to the fish, and the size of the fish are recorded. The sensor reports the data of all the fish detected in each sweep after a full sweep through the range of angles. This means the number of measurements reported at each time varies depending on the number of fish detected.

The training and test datasets, ekf_training_data.csv and ekf_test_data.csv, respectively, report the data as follows:

- The first column is the timestamp of the sweep.
- Each sweep corresponds to its own row.
- For each sweep (i.e. each row), the columns alternate between distances in meters, angles in degrees, and sizes in meters for every fish that was detected in the sweep.
- Note that measurements in the same column do not necessarily correspond to the same entities in the water. Instead, they correspond to the first, second, third etc. fish that were detected in the sweep.

Table 1 shows an example of how data is laid out in the csv files. Note that 1) the number of columns in each row is different 2) the first fish in column 1 (blue) at time 0.1 was missed in the measurement at time 0.2, so the second and third fish (green and orange) are all shifted over by 1 column.

Table 1:Example Format of Sensor Data

Timestamp [sec]	Dist. 1	Angle 1	Size 1	Dist. 2	Angle 2	Size 2	Dist. 3	Angle 3	Size 3
0.1	**1.03**	**0.587**	**1.04**	*2.06*	*0.804*	*0.24*	8.92	1.21	0.82
0.2	*2.05*	*0.811*	*0.25*	9.02	1.20	0.81			

1. Design a tracking system that uses an Extended Kalman Filter (EKF) to estimate the position of fish in the water. Note that your system must not rely on future measurements to estimate the current positions (only past measurements and process models should be used to estimate current position). In your report:
    a. Define the states in the EKF.
    b. Define the process model of the EKF and briefly explain how it was chosen. Explain how each parameter in the process model were set.
    c. Define the sensor model equations.
    d. Explain how the process covariance, Q, and sensor covariance, R, were selected.
    e. Explain how the unknown number of fish was handled by the tracking system.

f. Explain why an Extended Kalman Filter is needed for this problem rather than a conventional Kalman Filter.

2. Tune the tracking system using the training data, ekf_training_data.csv. The actual positions of the fish are provided in ekf_training_labels.csv.
    a. Each distinct fish in ekf_training_labels.csv corresponds to its own triplet of columns denoting the x- and y-coordinates, and the size of the fish.
    b. If fish is not in view at a particular time, the x- and y-coordinates, as well as the size are empty cells in the csv file.
3. Apply the tracking system to the test data, ekf_test_data.csv. Estimate the position of the fish in the water in the Cartesian coordinates of the reference frame defined in Figure 1.
    a. Report the position and covariance estimates of all the fish at the last timestep.
    b. Plot the estimated x- and y-position of each fish using a 2D line plot and overlay the position measurements from the sensors as a scatter plot.
4. Estimate the number of fish at every timestep in the test set. Note that the number of fish does not necessarily correspond to the number of measurements in each sensor sweep.
    a. Save the timeseries as a csv file in a single column with the count estimates for each timestep. **Do not include any headers** or miscellaneous data in other cells since the file will be evaluated with a script.

Note1. The top 3 networks based on their performance on the test data will receive 10% bonus to their final exam mark.

Problem 2: Bayesian fusion for Vorlon's football.

The famous Vorlons recently developed a game of football (aka soccer). Despite the common intergalactic belief on Volrons have telepathic power and complex sensory apparatus, they have a big problem with the invented (or maybe copied) game. They cannot see where the ball is!
Some scientists believe that this issue might have something to do with the Vorlons not having eyes!

Figure 2. left: a Vorlon (one of typical outer space species); right: a Volron (top scorer Ulkesh Naranek) is kicking the ball during an intergalactic match.

As a commission to earth you are tasked with designing a Bayesian fusion algorithm to predict where the ball goes during a penalty kick.
Determine the minimum error estimation of ball location with respect to the goal after a penalty kick.
Plot a portrait of each likelihood function, and their combined (posterior) probability of where the ball would go.

This is what we know:

The human spectators observed and collected data of 100 penalty kicks (Penalty_kick.csv). This includes:

- where the ball goes after the penalty kick with respect to a coordinate system attached to one of the goal posts (see figure 3), noted as column x, y, and z; the measurements are in meters, and a negative or out of range (y of goal is $y \in [0\ \ 7.32]$ and z o goal is $z \in [0\ \ 2.44]$) measure means that the ball was not on target (goal).

- where was the contact point on the ball at the kick (using a sensor embedded in the ball) indicated by the column ball_contact_area.

Figure 3. Description of the coordinate system attached to the goal post. Also regions on the ball (the sensors measurement on where the player kicked the ball, 1,2,3,4,5,6).

1) Compute the prior probability for the ball position as a mixture of three 2D Gaussians (defined on Y-Z plane) on the goal (hint: use only columns y and z of the data). **When you obtain the prior probability model, plot a color map of it on Y-Z plane. Print the mean vector and covariance matrix of each Gaussian in the mixture model.**
   Show details of the formula used and how it was normalized to be a probability measure in your report.
   For your ease the ball locations were sorted based on their y-axis values.
   Hint: use a scatter plot to visualize all the ball locations, to decide how to group them for each Gaussian function. Feel free to manually group the points.
   Otherwise, if you are using a different technique to find the mixture of Gaussian detail it in the report and make sure to include the code.

2) Unfortunately, the embedded sensor itself is not great! Compute **the most likely *y* and *z* of ball with respect to goal frame** given that ball sensors showed contact at either region 4 or 5 (on the ball).
   Plot a contour map or a heat map.
   You can interpret this reading as with 46% probability kick happened at region 4, with 46% at region 5, and with 2% on region 1, and similarly 2% on regions 2,3 or 6.

3) compute the *y* and *z* of ball in the previous condition using the posterior probability.

*You may assume that the combined probability is the product of the probabilities given by the independent sources.

Please note that besides the correct solution, the quality of your visualization and presentation is important.

Problem 3: Feedforward Neural Network to fuse multiple sensors data.

When we walk, our interaction with the ground generates a force that propels us forward and maintains our stability. This force, known as the Ground Reaction Force (GRF), plays a pivotal role in both gait analysis and control of assistive robots. GRF is measured using force plates which are essentially large load cells. However, due to their cost, integration complexity with mobile devices, and limited applicability outside clinical settings, alternative methods for GRF monitoring are necessary.
In this problem, you are tasked with developing an alternative solution for estimating GRF using previously collected data from participants walking on a treadmill equipped with load cells. The objective is to design, train, and test a feedforward neural network to achieve this goal.

Dataset:
The dataset of this problem is confidential. Even after the exam is completed don't share it with anyone. The dataset includes information from 10 participants, labeled as "sub_x_input.csv" and "sub_x_output.csv" where x denotes the participant number (1-10). Note: "sub_10_output.csv" is withheld for final testing of your model's performance.
Input data features include joint angles (q), joint speeds (dq), joint torques (u) of the hip and knee, gyroscope (gyro) and acceleration (acc) data across the x, y, and z axes on the thigh, shank, and foot of left and right legs, and a binary stance flag (sf) indicating when foot is in contact with the ground (sf=1 → that foot is in contact with the ground).
Output data consists of the vertical (z) component of GRF as well as the anterior-posterior (y) component, which is horizontal in the direction of movement, for both legs.

Tasks:

1- Design your Model:
   a. Decide on the structure and inputs of the feedforward neural network. Justify your choice briefly in the report.
      Note that you can select a subset of the data attributes to be fed into your neural network, as well as time delayed version of those inputs.
   b. Choose an appropriate training algorithm and justify your selection.
2- Training and Validation:
   a. Train your model using data from 8 participants and validate it on 1 participant (numbered 9).

b. Use data shuffling in the training but avoid random data shuffling. Instead, divide the training data into chunks of 500 samples each, and shuffle the chunks during different training iterations.
c. Aim for a Root Mean Squared Error (**RMSE**) of **less than or equal to 0.15** and a Coefficient of Determination ($\mathbf{R^2}$) **above 0.80** in validation of horizontal force (f_y). Also aim for an **RMSE** of **less than or equal to 0.3** and a Coefficient of Determination ($\mathbf{R^2}$) above **0.90** in validation of vertical force (f_z).
d. Provide error histograms for each network output, along with overall RMSE and R^2 for both training and validation phases.

Testing and Reporting:

1- Test your model on the 10th participant data.
2- Submit your network's output in a ".csv" file, along with a detailed report and the code used for this project.

Note1. The top 3 networks based on their performance on the test data will receive 10% bonus to their final exam mark.

Note 2. While we did not ask for a multi-fold analysis here, to keep it simple, you might want to try that in case of aiming for the top 3 performance.

This question challenges you to apply neural network methodologies to estimate GRF, a critical parameter in gait analysis and exoskeleton control. Your solution should demonstrate an understanding of neural network design, training, validation, and testing principles, as well as the ability to interpret and report on your model's performance accurately.