Estimation Techniques in Dynamic Systems and Machine Learning

Objective:

The objective of this project is to explore and implement advanced estimation techniques for dynamic systems or in machine learning contexts. The project aims to understand how estimation can be used to predict states in dynamic systems or to improve the accuracy and reliability of predictions in machine learning models.

Project Description:

The focus of this project is on developing and applying estimation methods in the context of dynamic systems or machine learning applications. The goal is to estimate the hidden or unknown parameters of a system, or predict the future behavior of a time-dependent model based on historical data. The project can be tailored towards real-world problems in fields like robotics, economics, weather forecasting, or even state estimation in neural networks.

The project will be divided into the following tasks:

1. **Problem Definition and System Modelling:**

- Choose a dynamical system or machine learning problem that requires state estimation or parameter estimation. For a dynamical system, this could be a robotic arm, vehicle motion, or weather system. In the machine learning context, this could involve time series prediction, latent variable estimation, or training Bayesian models.
- Define a state-space model for the system or a probabilistic model for the machine learning task. Identify key variables, system states, measurements, and sources of noise or uncertainty.

2. Estimation Technique Selection:

Based on the nature of the system problem, select appropriate estimation methods such as:

- **Kalman Filter (KF) and its variants:** Standard KF for linear systems, Extended Kalman Filter (EKF) for mildly nonlinear systems, and Unscented Kalman Filter (UKF) for more complex systems.
- **Particle Filters (PF)**: To handle non-Gaussian noise and highly nonlinear systems.
- **Bayesian Inference Methods**: For probabilistic estimation tasks in machine learning, using techniques like Variational Inference, Expectation-Maximization, or Markov chain Monte Carlo.
- **Deep Learning-based Estimation**: Applying neural networks like recurrent neural networks (RNN), long short-term memory (LSTM) networks, or Transformers for dynamic systems where states evolve over time.

3. **Implementation of Estimation Algorithms:**

- Develop a simulation environment to model the chosen dynamic system or gather data for the machine learning problem. Implement the selected estimation algorithms using MATLAB, Python, TensorFlow, or PyTorch.
- Generate synthetic or use real-world data to evaluate and test the estimation methods.

4. **Estimation Analysis and Evaluation:**

- Evaluate the accuracy of the estimators by comparing estimated states/parameters with ground truth values using error metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), or Negative Log-Likelihood (NLL).
- Perform sensitivity analysis by varying noise levels or system parameters. For machine learning tasks, analyze model convergence, stability, and the effect of hyperparameters.

5. **Application to Real-world Data (optional):**

- Choose a relevant real-world problem like navigation for autonomous robots, financial market predictions, health monitoring, or even image and video processing.
- Apply the developed estimation techniques to solve the selected problem and analyze the results in real-world scenarios.

6. **Deliverables:**

- A comprehensive project report (written in Latex) describing the mathematical modeling, estimation algorithms, implementation details, results, and analysis.
- Plots, figures, and tables illustrating system behavior, estimation performance, and the impact of noise or uncertainty.
- Source code for the implemented estimation techniques and simulation/real-world problem solutions.
- A presentation summarizing the project and key findings.

7. **Skills and Tools Required:**

- Mathematics and Statistics: Probability, linear algebra, differential equations, and Bayesian statistics.
- Programming: MATLAB, Python, TensorFlow, or PyTorch for simulations and model training.
- Estimation Theory and Machine Learning: Understanding of state estimation methods and machine learning algorithms like neural networks or Bayesian methods.

Student Grouping:

- **Group Size:** 2-3 students per group.
- **Group Assignment:** Students are free to form their own groups, but must submit group information to the Teaching Assistants by the end of Week 13.
- **Role Assignment:** Each group should designate a project manager, who will be responsible for coordinating tasks and submissions. Other roles (e.g., programmer, analyst, documenter) should also be assigned within the group.

Note: All reports and code must be submitted to the Teaching Assitants by 11:59 PM on January 5th. Each submission should include a cover page with the group number, member names, and assigned roles. The presentation is scheduled for December 19th, during the final class session.