SES/RAS 598: Space Robotics and AI

Lecture 1: Course Introduction & State Estimation Overview

Dr. Jnaneshwar Das

Arizona State University
School of Earth and Space Exploration

Spring 2025

Lecture Outline

Course Structure

- **Meeting Times:** Tu/Th 10:30-11:45am

Course Structure

- **Meeting Times:** Tu/Th 10:30-11:45am
- **Location:** PSF 647

Course Structure

- **Meeting Times:** Tu/Th 10:30-11:45am
- **Location:** PSF 647
- **Course Components:**
  - Assignments (20%)
  - Midterm Project (20%)
  - Final Project (50%)
  - Class Participation (10%)

Course Structure

- **Meeting Times:** Tu/Th 10:30-11:45am
- **Location:** PSF 647
- **Course Components:**
    - Assignments (20%)
    - Midterm Project (20%)
    - Final Project (50%)
    - Class Participation (10%)
- **Prerequisites:**
    - Linear algebra, calculus, probability theory
    - Python programming with NumPy, SciPy
    - Basic computer vision concepts
    - Linux/Unix systems experience

Course Resources

- **Recommended Books:**
  - Probabilistic Robotics (Thrun, Burgard, Fox)
  - Optimal State Estimation (Simon)
  - Pattern Recognition and Machine Learning (Bishop)

Course Resources

- **Recommended Books:**
  - Probabilistic Robotics (Thrun, Burgard, Fox)
  - Optimal State Estimation (Simon)
  - Pattern Recognition and Machine Learning (Bishop)
- **Interactive Tutorials:**
  - Sensor Fusion
  - Parameter Estimation
  - Gaussian Processes

Course Resources

- **Recommended Books:**
  - Probabilistic Robotics (Thrun, Burgard, Fox)
  - Optimal State Estimation (Simon)
  - Pattern Recognition and Machine Learning (Bishop)
- **Interactive Tutorials:**
  - Sensor Fusion
  - Parameter Estimation
  - Gaussian Processes
- **Required Software:**
  - Linux OS
  - ROS2
  - Python with scientific computing libraries

Why State Estimation?

- **Real-World Applications:**
  - Mars rover navigation
  - Drone flight control
  - Satellite attitude determination

Why State Estimation?

- **Real-World Applications:**
  - Mars rover navigation
  - Drone flight control
  - Satellite attitude determination
- **Key Challenges:**
  - Sensor noise and uncertainty
  - Environmental dynamics
  - Resource constraints

Why State Estimation?

- **Real-World Applications:**
  - Mars rover navigation
  - Drone flight control
  - Satellite attitude determination
- **Key Challenges:**
  - Sensor noise and uncertainty
  - Environmental dynamics
  - Resource constraints
- **Impact on Space Exploration:**
  - Autonomous navigation
  - Precision landing
  - Sample collection

Least Squares Estimation

- **Mathematical Foundation:**

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} (y_i - h(\theta))^2$$

Least Squares Estimation

- **Mathematical Foundation:**

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} (y_i - h(\theta))^2$$

- **Key Properties:**
  - Minimizes squared error
  - Optimal for Gaussian noise
  - Computationally efficient

Least Squares Estimation

- **Mathematical Foundation:**

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} (y_i - h(\theta))^2$$

- **Key Properties:**
  - Minimizes squared error
  - Optimal for Gaussian noise
  - Computationally efficient
- **Applications:**
  - Sensor calibration
  - Trajectory estimation
  - Parameter identification

Maximum Likelihood Estimation

- **Principle:**

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta} \prod_{i=1}^{n} p(y_i|\theta)$$

Maximum Likelihood Estimation

- **Principle:**

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta} \prod_{i=1}^{n} p(y_i|\theta)$$

- **Connection to Least Squares:**
  - Equivalent under Gaussian assumptions
  - More general framework
  - Handles different noise models

Maximum Likelihood Estimation

- **Principle:**

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta} \prod_{i=1}^{n} p(y_i|\theta)$$

- **Connection to Least Squares:**
  - Equivalent under Gaussian assumptions
  - More general framework
  - Handles different noise models
- **Space Applications:**
  - Orbit determination
  - Attitude estimation
  - Sensor fusion

State-Space Models

- **System Dynamics:**

$$
\begin{aligned}
x_{k+1} &= Ax_k + Bu_k + w_k \\
y_k &= Cx_k + v_k
\end{aligned}
$$

State-Space Models

- **System Dynamics:**

$$x_{k+1} = Ax_k + Bu_k + w_k$$
$$y_k = Cx_k + v_k$$

- **Components:**
  - State vector x_k
  - Input vector u_k
  - Measurement vector y_k
  - Process noise w_k
  - Measurement noise v_k

Case Study: Mars Rover Navigation

- **State Variables:**
  - Position (x, y, z)
  - Orientation (roll, pitch, yaw)
  - Velocities

Case Study: Mars Rover Navigation

- **State Variables:**
  - Position (x, y, z)
  - Orientation (roll, pitch, yaw)
  - Velocities
- **Sensors:**
  - Visual odometry
  - Inertial measurement unit (IMU)
  - Sun sensors

Case Study: Mars Rover Navigation

- **State Variables:**
  - Position (x, y, z)
  - Orientation (roll, pitch, yaw)
  - Velocities
- **Sensors:**
  - Visual odometry
  - Inertial measurement unit (IMU)
  - Sun sensors
- **Challenges:**
  - Wheel slippage
  - Varying terrain
  - Limited computational resources

Preparation for Next Lecture

- **Review:**
  - Matrix operations
  - Probability concepts
  - Basic Python programming

Preparation for Next Lecture

- **Review:**
  - Matrix operations
  - Probability concepts
  - Basic Python programming
- **Setup:**
  - Install Linux if needed
  - Configure ROS2 environment
  - Test Python scientific libraries

Preparation for Next Lecture

- **Review:**
  - Matrix operations
  - Probability concepts
  - Basic Python programming
- **Setup:**
  - Install Linux if needed
  - Configure ROS2 environment
  - Test Python scientific libraries
- **Reading:**
  - Skim Kalman filter basics
  - Review assigned papers
  - Explore interactive tutorials

Questions?

Thank you!

Contact: jdas5@asu.edu