Robot Mapping (SLAM)

|  |
| --- |
| Topics:   * Simultaneous Localization and Mapping * Kalman filter, EKF, UKF * Information filter * Partile filter * Graph-based approaches * Least squares error minimization * Robust optimization approches * Hierarchical approaches * Data association * SLAM front-ends * Apperance-based approaches * Long term operation * Semantic mapping   **Mixture of**:   * Introduction to robot mapping & SLAM * Key milestones of the past 20 years * Relevant state-of-the-art approaches for robot mapping * Hands on the problems – priactical work   **Prerequisites**:   * Basic math skills (LA, probabilistic concepts) * Basic programming skills * Useful: having attended the Introduction to Mobile Robotics course |

|  |
| --- |
| **Introduction to Robot Mapping**  What is Robot Mapping?   * Robot – a device, that moves through the environment * Mapping – modeling the environment   Related Terms:   * State Estimation * Mapping * Navigation * Localization * SLAM * Motion Planning   What is SLAM?   * Computing the robot’s poses (tư thế) and the map of the environment at the same time * Localization: estimating the robot’s location * Mapping: building a map * SLAM: building a map and localizing the robot simultaneously   SLAM Problem:   * SLAM is a checken-or-egg problem:   + A map is needed for localization and   + A pose estimate is needed for mapping   SLAM is Relevant   * It is considered a fundamental problem for truly autonomous robots * SLAM is the basis for most navigation systems   SLAM Applications:   * SLAM is central to a range of indoor, outdoor, air and underwater applications for moth manned autonomous vehicles.   Examples:   * At home: vacuum cleaner, lawn mower * Air: surveilance with unmanned air vehicels * Underwater: reef monitoring * Underground: exploration of mines * Space: terrain mapping for localization   Definition of the SLAM problem  Given:   * The robot’s controls   + U1:T = {u1, u2, u3,..., uT} * Observations   + Z1:T = {z1, z2, z3,..., zT}   Wanted:   * Map of the environment   + m * Path of the robot:   + X0:T = {x0, x1, x2, ..., xT} |