Albert-Ludwigs-Universität Freiburg
Lecture: Introduction to Mobile Robotics
Summer term 2007

Institut für Informatik

Prof. Dr. W. Burgard
MSc AI J. Sturm
Dipl.-Inf. B. Lau

Sheet 5

Topic: Extended Kalman Filter I

Submission deadline: Fri 25.05.2007, 11:00 a.m. (before class)

Introduction

This exercise deals with the prediction step of the Extended Kalman Filter (EKF). A pose estimate in time step t is represented by a Gaussian with mean μ_t and a covariance matrix Σ_t. These are the corresponding update equations:

$$\begin{aligned} \mu_t &:= g(\mu_{t-1}, u_t) \\ \Sigma_t &:= G\Sigma G^T + VMV^T \end{aligned}$$

Exercise 1:

As with the odometry model introduced in the lecture and Ex. 1 of the last sheet, a control vector u consists of an initial rotation angle δ_{rot_1}, a straight forward motion distance δ_{trans}, and a final rotation angle δ_{rot_2}. The mean poses (or states) μ_t are defined as usual:

$$u_t = \begin{pmatrix} \delta^t_{rot_1} \\ \delta^t_{trans} \\ \delta^t_{rot_2} \end{pmatrix} \text{ and } \mu_t = \begin{pmatrix} x_t \\ y_t \\ \theta_t \end{pmatrix} .$$

Given a pair of absolute odometry readings $s_1 = \langle x_1, y_1, \theta_1 \rangle$ and $s_2 = \langle x_2, y_2, \theta_2 \rangle$ as in the CARMEN log files, compute the relative pose like `s2.subtract(s1)` would do. Determine the equivalent control vector $u = \langle \delta_{rot_1}, \delta_{trans}, \delta_{rot_2} \rangle$. Note that this is the inverse transformation of Sheet 4, Ex. 1 (a).

Exercise 2:

The function $g(\mu_{t-1}, u_t)$ defines how the new pose mean estimate μ_t depends on the last pose μ_{t-1} and the control vector u_t. Determine g for the motion and control model of Ex. 1.

Exercise 3:

To perform the linearization of the previously defined function g, derive the Jacobians

$$G = \frac{\partial g(\mu_{t-1}, u_t)}{\partial \mu_{t-1}} \text{ and } V = \frac{\partial g(\mu_{t-1}, u_t)}{\partial u_t} .$$

If you are not sure about the form of Jacobians in general, please look it up (e.g. Wikipedia).

Exercise 4:

M is the motion noise matrix in control space. With VMV^T it is linearized and mapped into state space:

$$M = \begin{pmatrix} \sigma^2_{\delta_{rot_1}} & 0 & 0 \\ 0 & \sigma^2_{\delta_{trans}} & 0 \\ 0 & 0 & \sigma^2_{\delta_{rot_2}} \end{pmatrix}.$$

Assume $\sigma_{\delta_{rot_1}} = 5°$, $\sigma_{\delta_{trans}} = 0.5m$, and $\sigma_{\delta_{rot_2}} = 10°$. Given the initial estimates μ_0 and Σ_0, and the control vectors u_1 and u_2, compute the resulting estimates μ_1, Σ_1 and μ_2, Σ_2 using the Kalman filter equations.

$$\mu_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \Sigma_0 = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{pmatrix}, \quad u_1 = \begin{pmatrix} 10° \\ 3m \\ 10° \end{pmatrix}, \quad u_2 = \begin{pmatrix} -20° \\ 10m \\ -10° \end{pmatrix}.$$