Robotics
Exercise 10

Marc Toussaint
Lecturer: Duy Nguyen-Tuong
TAs: Philipp Kratzer, Janik Hager, Yoojin Oh
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

January 15, 2020

1 Kalman Localization (8 Points)

We consider the same car example as for the last exercise, but track the car using a Kalman filter.
Access the code as usual:

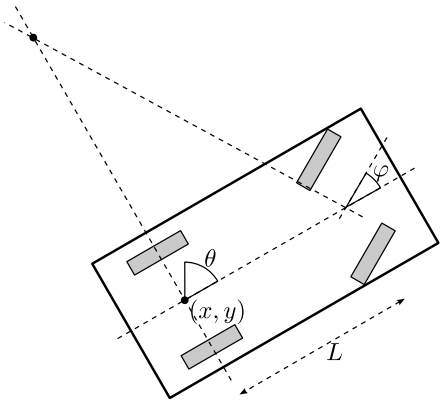
1. To make sure you have an updated version of the repository, run '*git pull*' and '*git submodule update*'
2. For python run: *'jupyter-notebook py/08-kalman/08-kalman.ipynb'*
3. For C++ run: *'cd cpp/08-kalman'*, *'make'*, *'./x.exe'*

The motion of the car is described by the following:

State $q = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$ **Controls** $u = \begin{pmatrix} v \\ \varphi \end{pmatrix}$

Sytem equation $\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = f(q,u) = \begin{pmatrix} v\ \cos\theta \\ v\ \sin\theta \\ (v/L)\ \tan\varphi \end{pmatrix}$

$|\varphi| < \Phi$



(In the following, we will denote the state with x_t (instead of q_t), such that it is consistent with the lecture slides.)

a) To apply a Kalman filter (slide 06:29) we need Gaussian models for $P(x_t \,|\, x_{t\text{-}1}, u_{t\text{-}1})$ as well as $P(y_t|x_t)$. We assume that the dynamics model is given as a local Gaussian of the form

$$P(x_{t+1} \,|\, x_t, u_t) = \mathcal{N}(x_{t+1}|x_t + B(x_t)u_t), \sigma_{\text{dynamics}})$$

where the matrix $B(x_t) = \frac{\partial f(x_t,u_t)}{\partial u_t}$ gives the local linearization of the car dynamics. What is $B(x_t)$ (the Jacobian of the state change w.r.t. u) for the car dynamics? (2P)

b) Implement the linearized dynamics model in the function 'getControlJacobian()'. (2P)

c) Concerning the observation likelihood $P(y_t|x_t)$ we assume

$$P(y_t|x_t, \theta_{1:N}) = \mathcal{N}(y_t \,|\, C(x_t)x_t + c(x_t), \sigma_{\text{observation}})$$

What is the matrix $C(x_t)$ (the Jacobian of the landmark positions w.r.t. the car state) in our example? (2P)

Hints:
- Assume there is only one landmark in the world.
- The car observes this landmark in its own coordinate frame, $y = l^C \in \mathbb{R}^2$.
- Write down the transformations between world and car coordinates $T_{W\to C}$ and the inverse $T_{C\to W}$.
- Use them to define a mapping $y = g(x_t)$ that maps the state x_t to the observation y.
- Compute the local linearization $C = \frac{\partial g(x_t)}{\partial x_t}$.

d) Implement the Kalman filter (slide 06:29) to track the car (this does not require a solution to question part c). Note that $c(s) = \widehat{y}_t - C(s)s$, where $\widehat{y}_t$ is the mean observation in the estimated state s. The variables $C, A, Q, W, \widehat{y}_t$ of the Kalman filter are already provided in the code. (2P)

2 Bayes Smoothing (4 points)

In the lecture we derived the Bayesian filter: given information on the past (observations $y_{0:t}$ and controls $u_{0:t\text{-}1}$) it estimates the current state x_t. However, we can use the available information on $y_{0:T}$ and $u_{0:T}$ also to get a Bayes-optimal estimate of a past state x_t at a previous time $t < T$. This estimate should be "better" than the forward filtered $P(x_t \,|\, y_{0:t}, u_{0:t\text{-}1})$ because it uses the additional information on $y_{t+1:T}$ and $u_{t:T}$. This is called Bayes smoothing (slides 06:32 – 06:33).

Derive the backward recursion $\beta_t(x_t) := P(y_{t+1:T}|x_t, u_{t:T})$ (the likelihood of all future observations given x_t and knowledge of all subsequent controls) of Bayes smoothing on slide 06:33. Explain in each step which rule/transformation you applied.