# Robotics
# Exercise 9

Marc Toussaint
Lecturer: Peter Englert
TAs: Matt Bernstein, Danny Driess, Hung Ngo
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

January 18, 2017

## 1 Bayes Smoothing

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 our knowledge of all subsequent controls) of Bayes smoothing on slide 08:32. Explain in each step which rule/transformation you applied.

## 2 Kalman Localization

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

Download the code framework here:
https://ipvs.informatik.uni-stuttgart.de/mlr/16-Robotics/e09-code.tbz2

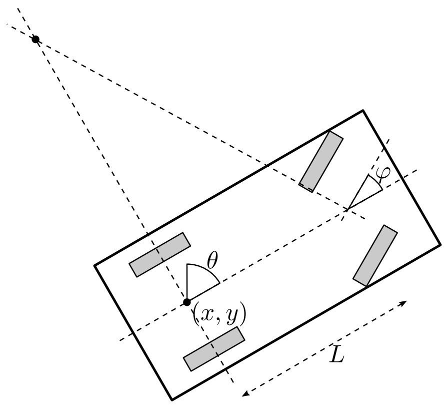
Unpack the code such that the folder structure is 'robotics15/share/teaching/RoboticsCourse/kalman_filter/main.cpp'

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

$\dot{x} = v\ \cos\theta$
$\dot{y} = v\ \sin\theta$
$\dot{\theta} = (v/L)\ \tan\varphi$
$|\varphi| < \Phi$




(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 08:29) 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?

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

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?

*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 08:29) to track the car (this does not require a solution to question part c). Note that $c(x_t) = \widehat{y}_t - C(x_t)x_t$, where $\widehat{y}_t$ is the mean observation in state $x_t$. The variables $C, A, Q, W, \widehat{y}_t$ of the Kalman filter are already provided in the code.

**Tips:**

– `~X` transposes the matrix X

– `inverse(X)` calculates the matrix inverse of matrix X