Probabilistic Robotics, BAIPR6, Fall 2011

Partial Exam, Thursday December 22th, 13:00 - 15:00, room G4.15

Arnoud Visser

December 22, 2011

Question 1

Both Extended Kalman Filters and Particle Filters are discussed in this course. Give for both algorithm an example of a situation that ”breaks” the algorithm.

Question 2

Before a mobile robot can perform a serious application, a solution has to be found on the following four tasks:

- Where am I?
- Where have I been?
- Where am I going?
- How can I reach that location?

Unfortunatelly, these problems cannot be solved independently from each other. Each is related to a field of robotics, described with at least a chapter in the textbook [3]. Figure 1 illustrates the overlap between of those robotics fields. The overlapping regions are tagged with Roman numerals.

Give a name to each of the overlapping regions, give a short description of that field of research and explain why an integrated solution is needed.

Give enough detail to demonstrate that you are comfortable with this subject. Keep it concise, because your description can be easily become fishy due a few incorrect associations.

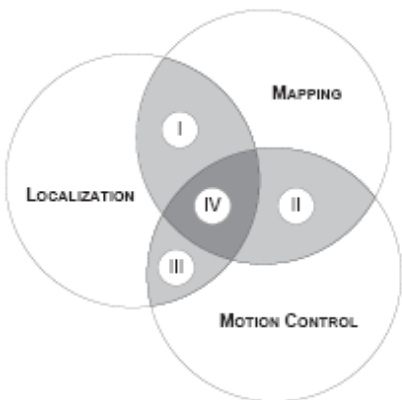

Figure 1: Tasks that need to be solved by a mobile robot. The overlapping regions represent combinations of mapping, localization and motion control. (Courtesy Makarenko *et al.* [2])

Question 3

Bayesian filtering is a general framework for recursively estimating the state of a dynamical system. Key components of each Bayes filter are probabilistic prediction and observation models. The dynamics of the described system is caught in both models, which can have many free parameters. For a blimp the motion model can be described with 16 parameters. In a paper in Autonomous Robots Journal [1], Dieter Fox shows how the free parameters can be learned with a non-parametric Gaussian process (GP) regression method to localize such a blimp (see Fig. 2.

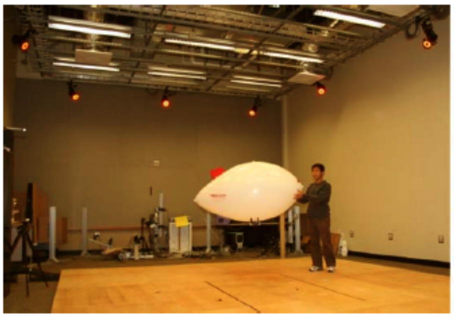

Figure 2: The blimb in an environment with a motion capture system. (Courtesy Ko and Fox [1])

The GP regression method is outside the scope of this course. What is important, is that GP models can be incorporated in three instantiations of Bayes filters that you already know via the definition of four functions:

$$GP_\mu([\mathbf{x}_{k-1}, \mathbf{u}_{k-1}], D_p) \tag{1}$$

$$GP_\Sigma([\mathbf{x}_{k-1}, \mathbf{u}_{k-1}], D_p) \tag{2}$$

$$GP_\mu(\mathbf{x}_k, D_o) \tag{3}$$

$$GP_\Sigma(\mathbf{x}_k, D_o) \tag{4}$$

which give an estimate on the mean μ and covariance Σ of the conditional probabilities $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ and $p(\mathbf{z}_k|\mathbf{x}_k)$ based on the training set D_p and D_o.

The state $\mathbf{x}$ consists of position $\mathbf{p}$, orientation ξ, linear velocity $\mathbf{v}$, and angular velocity ω. The blimb has three fans; two main fans in a gondola below the blimp and one small one in the tail. The control $\mathbf{u}$ consist of three inputs: the power of the blimb's gondola fans, the angle of the gondola fans and the power of the tail fan. The motion capture system in the room is only used for ground truth; the observations $\mathbf{z}$ of the blimp are made with two cameras. Observations are formed by background subtraction followed by the fitting of an ellipse to the remaining (foreground) pixels. The observations are the parameters of the ellipse in image space, parametrized by the position, size, and orientation.

a) Particle filter

Figure 3 illustrates the pseudo code to incorperate a GP model inside a particle filter. Inside figure 3 $GP([\mathbf{x}_{k-1}, \mathbf{u}_{k-1}], D_p)$ is used as short for the Gaussian distribution $\mathcal{N}(GP_\mu([\mathbf{x}_{k-1}, \mathbf{u}_{k-1}], D_p), GP_\Sigma([\mathbf{x}_{k-1}, \mathbf{u}_{k-1}], D_p))$.

Table 1 The GP-PF algorithm

Algorithm GP-PF$(\mathcal{X}_{k-1}, \mathbf{u}_{k-1}, \mathbf{z}_k)$**:**

$\bar{\mathcal{X}}_k = \mathcal{X}_k = \emptyset$
for $m = 1$ *to* M *do*
    sample $\mathbf{x}_k^m \sim \mathbf{x}_{k-1}^m + \mathrm{GP}([\mathbf{x}_{k-1}^m, \mathbf{u}_{k-1}], D_p)$
    $w_k^{[m]} = \mathcal{N}(\mathbf{z}_k; \mathrm{GP}_\mu(\mathbf{x}_k^m, D_o), \mathrm{GP}_\Sigma(\mathbf{x}_k^m, D_o))$
    add $\langle \mathbf{x}_k^m, w_k^{[m]} \rangle$ *to* $\bar{\mathcal{X}}_k$
endfor
for $m = 1$ *to* M *do*
    draw i *with probability* $\propto w_k^{[i]}$
    add $\mathbf{x}_k^{[i]}$ *to* $\mathcal{X}_k$
endfor
return $\mathcal{X}_k$

Figure 3: Gaussian process particle filter algorithm (Courtesy Ko and Fox [1])

The algorithm as shown in figure 3 contains one small mistake. Can you find that mistake? How can the line of pseudo code be corrected?

b) Extended Kalman filter

Figure 4 illustrates the pseudo code to incorperate a GP model inside an extended Kalman filter.

Table 2 The GP-EKF algorithm

Algorithm GP-EKF$(\mu_{k-1}, \Sigma_{k-1}, \mathbf{u}_{k-1}, \mathbf{z}_k)$**:**

$\hat{\mu}_k = \mu_{k-1} + \mathrm{GP}_\mu([\mu_{k-1}, \mathbf{u}_{k-1}], D_p)$
$Q_k = \mathrm{GP}_\Sigma([\mu_{k-1}, \mathbf{u}_{k-1}], D_p)$
$G_k = I + \frac{\partial \mathrm{GP}_\mu([\mu_{k-1}, \mathbf{u}_{k-1}], D_p)}{\partial \mathbf{x}_{k-1}}$
$\hat{\Sigma}_k = G_k \, \Sigma_{k-1} \, G_k^T + Q_k$
$\hat{\mathbf{z}}_k = \mathrm{GP}_\mu(\hat{\mu}_k, D_o)$
$R_k = \mathrm{GP}_\Sigma(\hat{\mu}_k, D_o)$
$H_k = \frac{\partial \mathrm{GP}_\mu(\hat{\mu}_k, D_o)}{\partial \mathbf{x}_k}$
$K_k = \hat{\Sigma}_k \, H_k^T (H_k \, \hat{\Sigma}_k \, H_k^T + R_k)^{-1}$
$\mu_k = \hat{\mu}_k + K_k(\mathbf{z}_k - \hat{\mathbf{z}}_k)$
$\Sigma_k = (I - K_k \, H_k) \, \hat{\Sigma}_k$
return μ_k, Σ_k

Figure 4: Gaussian process extended Kalman filter algorithm (Courtesy Ko and Fox [1])

Give the definition of the motion function $g()$ and measurement function $h()$ based on the pseudo code in figure 4.

c) Unscented Kalman filter

Figure 5 illustrates the pseudo code to incorperate a GP model inside an unscented Kalman filter.

Table 3 The GP-UKF algorithm

Algorithm GP-UKF$(\mu_{k-1}, \Sigma_{k-1}, \mathbf{u}_{k-1}, \mathbf{z}_k)$**:**

$\mathcal{X}_{k-1} = (\mu_{k-1} \quad \mu_{k-1} + \gamma\sqrt{\Sigma_{k-1}} \quad \mu_{k-1} - \gamma\sqrt{\Sigma_{k-1}})$

for $i = 0 \dots 2n$: $\quad \bar{\mathcal{X}}_k^{[i]} = \mathcal{X}_{k-1}^{[i]} + \mathrm{GP}_\mu([\mathcal{X}_{k-1}^{[i]}, \mathbf{u}_{k-1}], D_p)$

$Q_k = \mathrm{GP}_\Sigma([\mu_{k-1}, \mathbf{u}_{k-1}], D_p)$

$\hat{\mu}_k = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{X}}_k^{[i]}$

$\hat{\Sigma}_k = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_k^{[i]} - \hat{\mu}_k)(\bar{\mathcal{X}}_k^{[i]} - \hat{\mu}_k)^T + Q_k$

$\hat{\mathcal{X}}_k = (\hat{\mu}_t \quad \hat{\mu}_t + \gamma\sqrt{\hat{\Sigma}_k} \quad \hat{\mu}_t - \gamma\sqrt{\hat{\Sigma}_k})$

for $i = 0 \dots 2n$: $\quad \hat{\mathcal{Z}}_k^{[i]} = \mathrm{GP}_\mu(\hat{\mathcal{X}}_k^{[i]}, D_o)$

$R_k = \mathrm{GP}_\Sigma(\hat{\mu}_k, D_o)$

$\hat{z}_k = \sum_{i=0}^{2n} w_m^{[i]} \hat{\mathcal{Z}}_k^{[i]}$

$S_k = \sum_{i=0}^{2n} w_c^{[i]} (\hat{\mathcal{Z}}_k^{[i]} - \hat{z}_k)(\hat{\mathcal{Z}}_k^{[i]} - \hat{z}_k)^T + R_k$

$\hat{\Sigma}_k^{x,z} = \sum_{i=0}^{2n} w_c^{[i]} (\hat{\mathcal{X}}_k^{[i]} - \hat{\mu}_k)(\hat{\mathcal{Z}}_k^{[i]} - \hat{z}_k)^T$

$K_k = \hat{\Sigma}_k^{x,z} S_k^{-1}$

$\mu_k = \hat{\mu}_k + K_k(z_k - \hat{z}_k)$

$\Sigma_k = \hat{\Sigma}_k - K_k S_k K_k^T$

return μ_k, Σ_k

Figure 5: Gaussian process extended Kalman filter algorithm (Courtesy Ko and Fox [1])

How many sigma points are used in this algorithm (a blimp floats in 3 dimensions)?

References

[1] J. Ko and D. Fox, "GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models", *Autonomous Robots*, volume 27(1):pp. 75–90, 2009, ISSN 0929-5593, 10.1007/s10514-009-9119-x.

[2] A. Makarenko, S. Williams, F. Bourgault and H. Durrant-Whyte, "An Experiment in Integrated Exploration", in "Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)", 2002.

[3] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*, The MIT Press, September 2005, ISBN 0-262-20162-3.