MTB Challenge – Summer Term 2017/2018

Rostislav Karásek

March 1, 2018

1 Introduction

In many fields of human interest, it is necessary to precisely estimate the actual state of a system using some noisy sensors. There is more than one method of doing so according to the system properties. It is extremely computationally challenging to solve the arbitrary system with unknown stochastic parameters such as its Joint Probability Distribution ($P_{\mathbf{X}}$) or if it contains memory.

The stochastic systems are usually simplified as a Markov chain or a hidden Markov chain. This process is then called Markov process and it is assumed as a memoryless discrete system. The future state $\boldsymbol{X}_{k+1}$ of this system is defined as a vector function of the actual state $\boldsymbol{X}_k$, and the input of a system $\mathbf{u}_{k+1}$ is given by:

$$\mathbf{X}_{k+1} = \mathbf{f}\left[\mathbf{X}_k, \mathbf{u}_{k+1}\right] + \mathbf{n}_{k+1}, \tag{1}$$

where $\mathbf{f}[.,.]$ is a process model, and $\mathbf{n}_{k+1}$ is an additive process noise. We usually do not observe the state of a system directly, but we use sensors which output is represented by so–called observation vector $\mathbf{z}_{k+1}$ given by:

$$\mathbf{z}_{k+1} = \mathbf{h}\left[\mathbf{X}_{k+1}, \mathbf{u}_{k+1}\right] + \mathbf{w}_{k+1}, \tag{2}$$

where $\mathbf{h}[.,.]$ is an observation model, and $\mathbf{w}_{k+1}$ is an additive measurement noise. The problem is to estimate system state and its error. It is necessary to solve likelihood function of a conditional probability:

$$p\left(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t}\right), \tag{3}$$

which can be used to describe stochastic systems. It is obvious that this description requires the whole history of positions, measurements and control commands. Luckily it is possible to use Markov chain assumption and show that the whole history is contained in previous step and $x_{0:t-1}$ is sufficient statistic of the whole history. This allows us to separate this likelihood function into two parts:

$$\begin{array}{l} p\left(x_t \mid x_{t-1}, u_t\right) \\ p\left(z_t \mid x_t\right), \end{array} \tag{4}$$

where the first part is called state transition probability, and it is describing time evolution given previous state and actual controls. The second part is called measurement probability, and it is describing what measurements z are generated from the state x. One can notice that this is just another description of the same model presented in (1) and (2).

The last important concept is a belief. It is describing robot's internal knowledge of the environment. Belief is a posterior probability given by:

$$bel\left(x_t\right) = p\left(x_t \mid z_{1:t}, u_{1:t}\right). \tag{5}$$

In another word, it describes distribution over the state x_t at the time t, conditioned on all past measurements and controls. But sometimes we need to predict actual state x_t before incorporating measurement z_t. This is usually called prediction, and it is described by:

$$\overline{bel}\left(x_t\right) = p\left(x_t \mid z_{1:t-1}, u_{1:t}\right). \tag{6}$$

This leads us to method how to calculate belief $bel\left(x_t\right)$ from prediction $\overline{bel}\left(x_t\right)$. The most common method is to use so called Bayes Filters. There are several types of Bayes Filter methods depending on the process and observation models. This theory is described in detail in [1].

If the system model is a linear (functions $\mathbf{f}\left[\mathbf{X}_k, \mathbf{u}_{k+1}\right]$ and $\mathbf{h}\left[\mathbf{X}_{k+1}, \mathbf{u}_{k+1}\right]$ are linear) and both n_{k+1} and w_{k+1} are of a Gaussian distribution family, then the Kalman Filter (KF) is the optimal solution [2] and [1]. When the model is a nonlinear function, then KF is not optimal and may even not converge. From the second half of the 20th century, the Extended Kalman Filter (EKF) is mainly used for the solution of nonlinear systems. The problem is that the EKF is the only linearization of the nonlinear model by computation of the Jacobian of a process function. This method can be costly because Jacobian may not be defined in a closed form and iterative estimation of is necessary. Another problem is that if the process model is highly nonlinear the Jacobian is a poor approximation thus the error estimate is high.

For this reason, a new method was developed in the 1990s. It is based on particle filtration which suffers from high computation complexity, but its main advantage is that for certain systems it gives a better precision of the estimate than EKF. This method is called Unscented Kalman Filter (UKF) [2], and it is based on evolving an only low number of correctly chosen particles, called sigma points, through the nonlinear system function and the approximating the evolved sigma points by Gaussian distribution [3], [4].

As previously mentioned the UKF is an approximation of the real problem using Gaussian distribution. This will cause a high error in highly nonlinear systems. The example of this problem is a navigation of the robot using ultrasound sensors with a limited range on a long hall with opened doors. The map is known, and the robot is trying to find its position on this map. The main problem is that robot is quite sure that it is passing by doors, but it cannot decide which door. This leads to uncertainty that there is the equal probability that it is located next to any of the door in a hallway. This is something that cannot be correctly approximated by Gaussian since it has only one maximum. We need Probability Density Function (PDF) with several maxims. This problem is modeled as an arbitrary density function, and since we cannot describe this function by a single equation as a Gaussian, we are performing so–called sampling of a PDF. This is the base idea of so–called particle filter, where particles are in fact samples of a PDF.

To find more details we encourage you to use book Probabilistic Robotics [1] and also, handy is SLAM course by Cyrill Stachniss.

This project aims to implement Bayesian Filter and use it to filter data obtained from noisy and inaccurate sensors to obtain filtered state of a nonlinear system which represents tank–like four–wheel robot position and orientation. It is necessary to optimize assigned stochastic sensor parameters to minimize Mean Square Error (MSE) of the estimated robot position. More information and details will be specified in the following section.

Figure 1: Example of the four–wheel robot.

2 Competition setting

The task is to realize Bayesian Filer for estimation of the position and the heading H of a differential drive four–wheel robot, which can be modelled as a unicycle by its velocity v in the direction of H and angular velocity ω. The heading is defined as an orientation angle of the robot in used 2D Cartesian coordinate system. Details about differential drive can be found here.

The problem is that unicycle model is not correct since the four–wheel robot movement is more "tank–like." This fact is causing nonlinearity in the system model because wheels are slipping while turning. At the same time when the set velocity is low and angular velocity is not set to zero robot is not able to overcome friction between wheels and surface and will not move at all.

Usually, wheeled robots are navigated using mainly Global Navigation Satellite System (GNSS) receiver and Inertial Measurement Unit (IMU). In this task, the available sensors are Real Time Kinematics (RTK) Global Positioning System (GPS), hall sensors attached to motors and magnetometer. The RTK GPS is sending data using so–called National Marine Electronics Association (NMEA) protocol. For simplicity, the GPS data are given as a vector of 2D Cartesian position, velocity, and the heading. Assume white Gaussian noise of the position with zero mean $\mu_{xy} = 0$ m and standard deviation $\sigma_{xy} = 0.3$ m. Statistical parameters of measured velocity and the heading with GPS are tricky because the precision of velocity decrease with decreasing velocity. Heading measurement depends on velocity if a two antenna GPS receiver solution is not used. For this reason, the low pass filter is used and when velocity is a lower than $v < 0.2$ ms^{-1} then the measuring of velocity and the heading is stopped, and value in NMEA is zero. Again, for simplicity assume Additive white Gaussian noise (AWGN) if $v \geq 0.2$ ms^{-1} with following parameters: $\mu_v = 0$ ms^{-1}, $\sigma_v = 0.03$ ms^{-1}, $\mu_H = 0$ rad, $\sigma_H = 0.01$ rad. Another problem is that GPS receiver sampling rate is 1 Hz, but the system is aimed for sampling period given by constant `TS` in the `UKFdata.mat`. Also, a short outage of the GPS signal can occur (seconds).

Hall sensors work with excellent precision at submillimetre level, but it cannot be properly used if wheels are slipping during turns the following speed according to differential drive model. The distance between robot right and left wheel is L=0.44 m. Wheel radius is 0.13 m. Used Hall sensor measure distance in revolutions with 2000 pulses per revolution with $\sigma = 1$ pulse. The output format is stored separate from the right $\mathbf{\Delta r}_R$ and left $\mathbf{\Delta r}_L$ motor as a change of distance travelled over one sampling period in a number of revolutions.

Finally, the magnetometer is used for measuring absolute heading value of the robot. The problem is that magnetic field measured by the magnetometer can be easily influenced by near electromagnetic field produced by motors and by the magnetic materials near the sensor. But it is possible to correct the error and assume zero mean. High precision magnetometers can achieve heading precision around 0.5° RMS if the tilt is lover than ±15°. Assume an only low–cost sensors are used, the heading precision is $\sigma = 0.1$ rad.

Assume independent observation $\mathbf{z}_{k+1}$ and state $\boldsymbol{X}_k$ vector. In other words, every used covariant matrix Σ is diagonal. The system function is given by:

$$\begin{aligned} H_{k+1} &= H_k + T_S\omega_{k+1} + n_{H}{}_{k+1}, \\ x_{k+1} &= x_k + T_S v_{x}{}_{k+1} \mathrm{e}^{-2T_S\omega_{k+1}^2} \left(1 - \frac{\left|\omega_{k+1}^{0.6}\right|}{\mathrm{e}^{5T_S v_{k+1}}}\right)^{10} + n_{x}{}_{k+1}, \\ y_{k+1} &= y_k + T_S v_{y}{}_{k+1} \mathrm{e}^{-2T_S\omega_{k+1}^2} \left(1 - \frac{\left|\omega_{k+1}^{0.6}\right|}{\mathrm{e}^{5T_S v_{k+1}}}\right)^{10} + n_{y}{}_{k+1}, \end{aligned} \tag{7}$$

where v and ω are inputs of the system $\mathbf{u}$, T_S is time step between system samples in seconds, v_x and v_y are given by:

$$\begin{aligned} v_x &= v \cos H, \\ v_y &= v \sin H. \end{aligned} \tag{8}$$

The nonlinearity is located only in position increment, and this function is dependent on control signals of the system v and ω. This nonlinear increment function of the position Δx in x–axis for $H = 0$ rad is shown in Fig. 2. The wheel slipping rate increase with increasing angular velocity ω. For small values of v and higher values of ω robot will not even move.

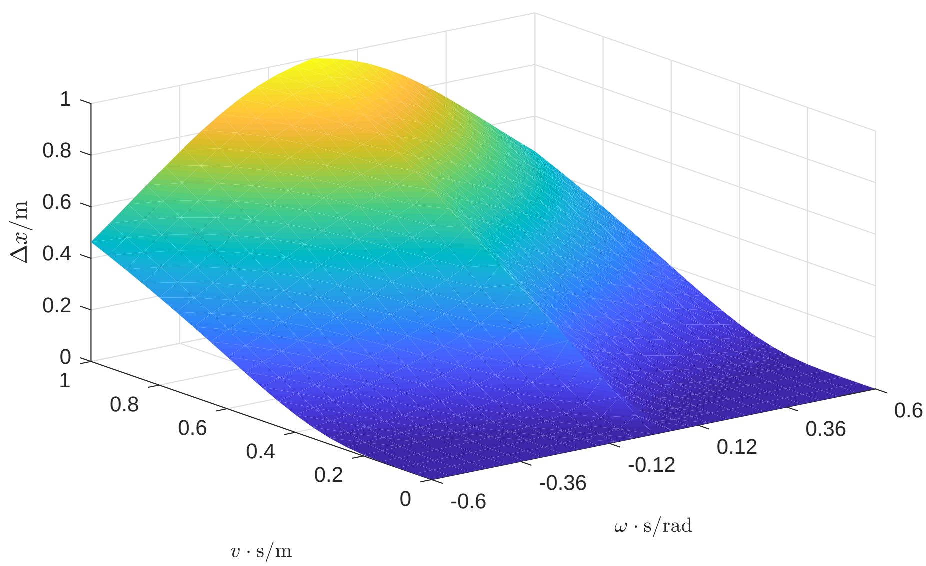

Figure 2: Nonlinear system function used for prediction of the system state.

The robot trajectory is obtained according to specified system model using randomly generated set of controlling signals v and ω. Example trajectory of the robot is shown in Fig. 3. The measuring is generated according to specified sensors from this trajectory. In Fig. 3 is shown position measuring obtained from GPS.

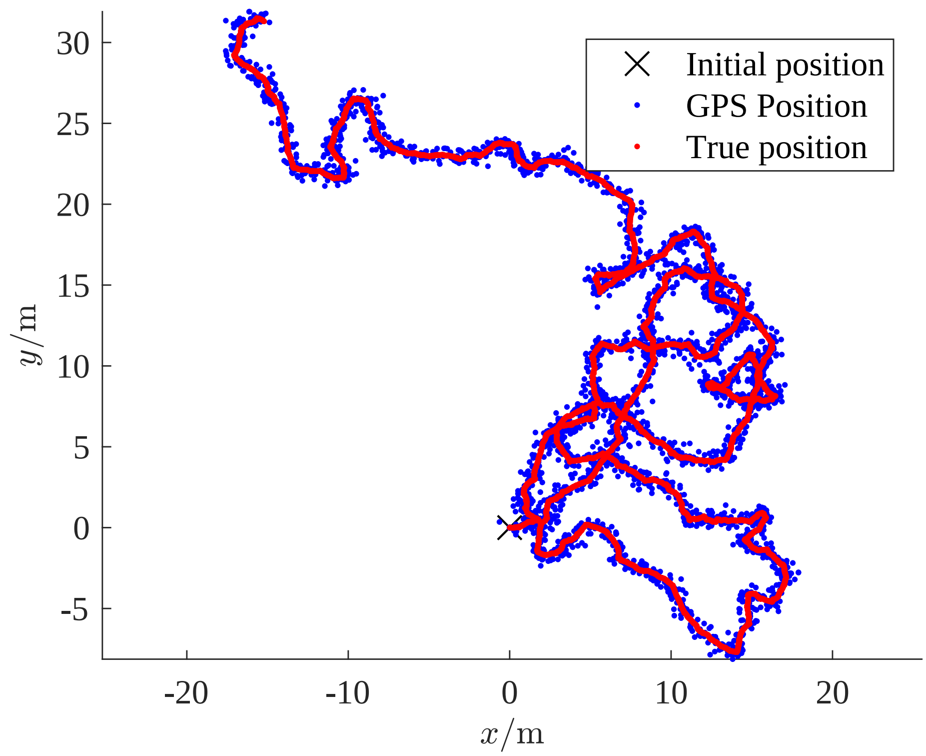

Figure 3: Example of robot trajectory and measured position.

The task is to fuse all measured data and obtain actual robot state. The real trajectory can be used to optimize covariant matrices to achieve minimal MSE of the estimate. The attached input data in `UKFdata.mat` contain Matlab variables necessary for solving this task. The input data are defined according to usual notation for UKF, where matrix $\mathbf{u}$ is containing N inputs of the system in a following format:

$$\mathbf{u} = \begin{bmatrix} \mathbf{v} & \omega \end{bmatrix}^{\mathrm{T}}, \tag{9}$$

and the observation $\mathbf{z}$ is containing N measurements stored in format:

$$\mathbf{z} = \begin{bmatrix} \mathbf{x}_{\mathrm{GPS}} & \mathbf{y}_{\mathrm{GPS}} & \mathbf{v}_{\mathrm{GPS}} & \mathbf{H}_{\mathrm{GPS}} & \mathbf{\Delta r}_{R} & \mathbf{\Delta r}_{\mathrm{L}} & \mathbf{H}_{\mathrm{mag}} \end{bmatrix}^{\mathrm{T}}. \tag{10}$$

Finally, the correct robot state is stored in a struct `RobotTrue`. The solution should contain computation of a MSE to compare output of a created UKF with the robot true state stored in a struct `RobotTrue`. The output MSE value should be separated for a each coordinate and a heading.

3 Criteria

- This project can be selected by a unlimited number of students. However, no collaboration between students is expected.
- The project should be submitted including short documentation describing how the algorithm works.
- Like for regular projects, a short presentation (a couple of minutes) is expected.
- To be awarded with credits it is necessary to implement arbitrary type of a Bayesian Filter to extrapolate observation to sampling period given by T_S.
- In competition, the minimal MSE is judged, and the first n students achieving the minimal MSE will be awarded with prices.
- No toolboxes or external codes and libraries (dll, mex) are allowed.
- It is possible to always withdraw from the competition and select of regular projects. This decision should be discussed with lecturers, and their approval is required.

4 Consultation

- Consultation is every Thursday 15:00 – 16:00 in T2:B2-818.
- It is possible to arrange individual consultation if the consultations are in your lecture time

References

[1] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics.* MIT press, 2005.

[2] S. M. Kay. *Fundamentals of statistical signal processing, volume I: estimation theory.* Prentice Hall Signal Processing Series. Prentice-Hall PTR, 1993.

[3] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *American Control Conference, Proceedings of the 1995*, volume 3, pages 1628–1632 vol.3, Jun 1995.

[4] E. A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, pages 153–158, 2000.