ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for Dynamic Systems and Control
IDSC
Institut für Dynamische Systeme und Regelungstechnik

151-0566-00 **Recursive Estimation** (Spring 2019)

Programming Exercise #1 Topic: Kalman Filtering

Issued: May 8, 2019 Due: **June 05, 2019**

Hybrid Extended Kalman Filter for Tracking a Boat

You are driving a motor boat across a big lake and don't want to get lost. You therefore decide to implement a Hybrid Extended Kalman Filter (Hybrid EKF) for tracking the position and orientation of the boat. A schematic diagram of the boat is shown in Figure 1.

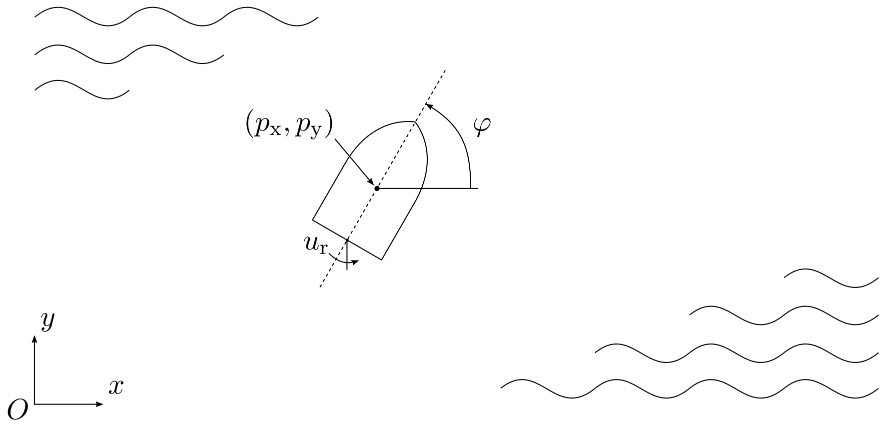

Figure 1: Schematic diagram of the boat and the global coordinate system (x, y) with origin O. The position of the boat is denoted by $(p_{\mathrm{x}}, p_{\mathrm{y}})$, the orientation with respect to the x-axis by φ and the control input to adjust the rudder by u_{r}.

The control inputs to the boat are the thrust command $u_{\mathrm{t}}(t)$ and the rudder command $u_{\mathrm{r}}(t)$. We assume the boat can only move in longitudinal direction. The continuous-time dynamics in the x and y directions can be described by the following equations (note that the time index has been omitted for ease of notation),

$$\dot{p}_{\mathrm{x}} = s_{\mathrm{x}} \tag{1}$$
$$\dot{p}_{\mathrm{y}} = s_{\mathrm{y}} \tag{2}$$
$$\dot{s}_{\mathrm{x}} = \cos(\varphi)\left[\tanh(u_{\mathrm{t}}) - C_{\mathrm{d}}(s_{\mathrm{x}}^2 + s_{\mathrm{y}}^2)(1 + v_{\mathrm{d}})\right] \tag{3}$$
$$\dot{s}_{\mathrm{y}} = \sin(\varphi)\left[\tanh(u_{\mathrm{t}}) - C_{\mathrm{d}}(s_{\mathrm{x}}^2 + s_{\mathrm{y}}^2)(1 + v_{\mathrm{d}})\right]. \tag{4}$$

The $\tanh(u_{\mathrm{t}})$ term in (3), (4) models the generated thrust of the boat, which is saturated for high thrust commands. The last term in (3), (4) models the hydraulic drag term, which is a function of the total speed of the boat and a known drag coefficient C_{d}. The boat can only generate thrust in the positive direction, hence $u_{\mathrm{t}} \geq 0$. The process noise $v_{\mathrm{d}}(t)$ takes model uncertainties of the drag model into account. It is assumed to be continuous-time white noise with

$$E[v_{\mathrm{d}}(t)] = 0, \qquad E[v_{\mathrm{d}}(t)v_{\mathrm{d}}(t+\tau)] = Q_{\mathrm{d}}\delta(\tau), \tag{5}$$

for all times. The angular dynamics are described by

$$\dot{\varphi} = C_\mathrm{r} u_\mathrm{r}(1 + v_\mathrm{r}). \tag{6}$$

The angular velocity is a function of the commanded rudder angle and some given constant C_r. The rudder angle command is limited to $u_\mathrm{r}(t) \in [-\bar{U}_\mathrm{r}, \bar{U}_\mathrm{r}]$. The process noise $v_\mathrm{r}(t)$ takes model uncertainties of the steering into account. It is assumed to be continuous-time white noise with

$$E[v_\mathrm{r}(t)] = 0, \qquad E[v_\mathrm{r}(t) v_\mathrm{r}(t+\tau)] = Q_\mathrm{r}\delta(\tau), \tag{7}$$

for all times.

The boat receives distance measurements from two ground-based radio stations, respectively placed at the exactly known positions $(x_\mathrm{a}, y_\mathrm{a})$ and $(x_\mathrm{b}, y_\mathrm{b})$. These measurements are affected by noise. A third radio station (see Figure 2), placed exactly at $(x_\mathrm{c}, y_\mathrm{c})$, provides more accurate distance measurements. However, the latter are not available at all sampling instants kT_s, where T_s is the constant sampling time in seconds and k the time index. Angle measurements are provided by a gyroscope which is affected by sensor drift,

$$\dot{b} = v_\mathrm{b} \tag{8}$$

where $v_b(t)$ is continuous-time white noise with

$$E[v_\mathrm{b}(t)] = 0, \qquad E[v_\mathrm{b}(t) v_\mathrm{b}(t+\tau)] = Q_\mathrm{b}\delta(\tau), \tag{9}$$

for all times. In addition, angle measurements are also provided by a compass, which is not affected by drift, but subject to higher measurement noise. With $b[k] = b(kT_s)$ the measurement equations of the sensors are therefore given by

$$z_\mathrm{a}[k] = \sqrt{(p_\mathrm{x}[k] - x_\mathrm{a})^2 + (p_\mathrm{y}[k] - y_\mathrm{a})^2} + w_\mathrm{a}[k] \tag{10}$$

$$z_\mathrm{b}[k] = \sqrt{(p_\mathrm{x}[k] - x_\mathrm{b})^2 + (p_\mathrm{y}[k] - y_\mathrm{b})^2} + w_\mathrm{b}[k] \tag{11}$$

$$z_\mathrm{c}[k] = \sqrt{(p_\mathrm{x}[k] - x_\mathrm{c})^2 + (p_\mathrm{y}[k] - y_\mathrm{c})^2} + w_\mathrm{c}[k] \tag{12}$$

$$z_\mathrm{g}[k] = \varphi[k] + b[k] + w_\mathrm{g}[k] \tag{13}$$

$$z_\mathrm{n}[k] = \varphi[k] + w_\mathrm{n}[k] \tag{14}$$

where $w_\mathrm{a}[k] \sim \mathcal{N}\left(0, \sigma_\mathrm{a}^2\right)$, $w_\mathrm{b}[k] \sim \mathcal{N}\left(0, \sigma_\mathrm{b}^2\right)$ and $w_\mathrm{c}[k] \sim \mathcal{N}\left(0, \sigma_\mathrm{c}^2\right)$, $w_\mathrm{g}[k] \sim \mathcal{N}\left(0, \sigma_\mathrm{g}^2\right)$, and $w_\mathrm{n}[k] \sim \mathcal{N}\left(0, \sigma_\mathrm{n}^2\right)$.

The boat's commands are given at discrete time instants $t_0 = 0, t_1 = T_s, t_2 = 2T_s, \ldots$ and are kept constant over the sampling interval.

At the initial time $t = 0$, the boat has zero velocity and is located at $(p_\mathrm{x}(0), p_\mathrm{y}(0)) = (x_0, y_0)$ with orientation $\varphi(0) = \varphi_0$. The initial position (x_0, y_0) is equally likely to be anywhere inside a circle of radius R_0, centered at the origin. The probability density function of φ_0 is uniformly distributed with $\varphi_0 \in [-\bar{\varphi}, \bar{\varphi}]$. At $t = 0$ the gyroscope measurement is bias free.

All random variables φ_0, x_0, y_0, $\{w_\mathrm{a}[\cdot]\}$, $\{w_\mathrm{b}[\cdot]\}$, $\{w_\mathrm{c}[\cdot]\}$, $\{w_\mathrm{g}[\cdot]\}$, $\{w_\mathrm{n}[\cdot]\}$ are mutually independent. The continuous time process noises $v_\mathrm{d}(t)$, $v_\mathrm{r}(t)$ and $v_\mathrm{b}(t)$ are similarly independent for all times.

Estimator Design

The objective is to design a hybrid EKF to estimate the full state of the boat. The estimator is executed at the time instants $t_0 = 0, t_1 = T_s, t_2 = 2T_s, \ldots$. At time $t_k = kT_s$, the estimator has access to the time t_k, the control inputs $u_\mathrm{t}[k-1]$ and $u_\mathrm{r}[k-1]$, and the measurements $z_\mathrm{a}[k]$, $z_\mathrm{b}[k]$, $z_\mathrm{g}[k]$, $z_\mathrm{n}[k]$ and possibly $z_\mathrm{c}[k]$. Furthermore, the constants Q_d, Q_r, Q_b, C_d, C_r, σ_a, σ_b, σ_c, σ_g, σ_n, R_0, and $\bar{\varphi}$ are known to the estimator. The position, the orientation, the linear velocity and the gyroscope sensor drift are the estimator states.

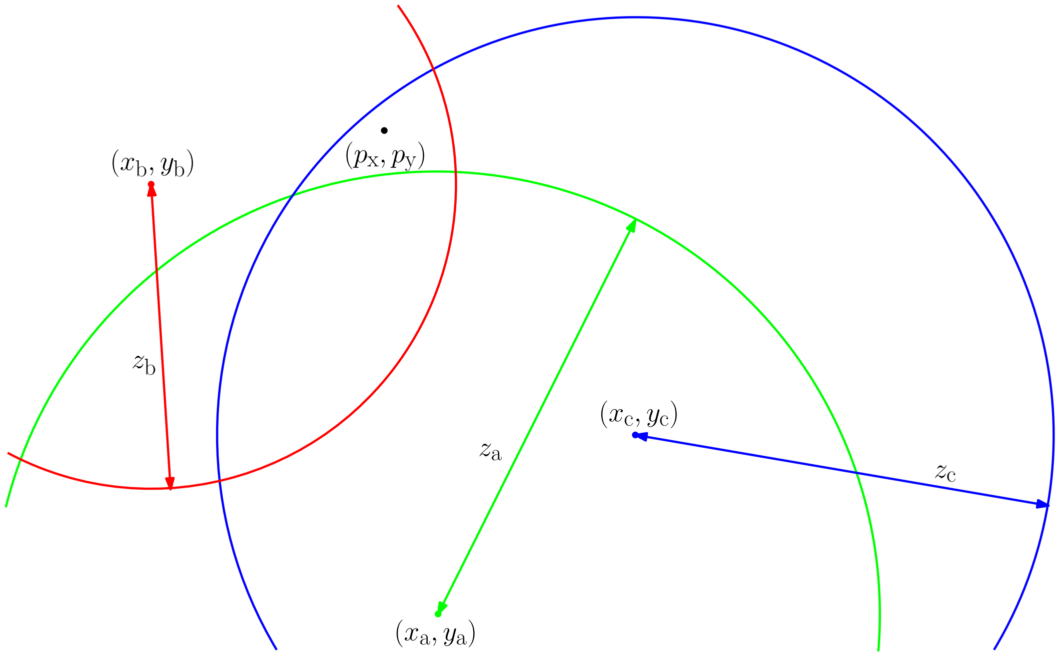

Figure 2: Example of distance measurements: each distance measurement indicates that the boat is positioned along a circle centered at the respective radio station. Since the measurements are corrupted by noise, the three circles do not exactly intersect at the boat's position.

Provided Matlab Files

A set of Matlab files is provided on the class website. Please use them for solving the exercise.

`run.m`	Matlab function template that is used to execute a simulation of the true system, run the estimator, plot the results, and report the root-mean squared tracking error.
`Estimator.m`	Matlab function template to be used for your implementation of the EKF.
`Simulator.p`	Matlab function used to simulate the motion of the boat and measurements. This function is called by `run.m`, and is obfuscated (i.e., its source code is not readable).
`EstimatorConst.m`	Constants known to the estimator.
`SimulationConst.m`	Constants used for the simulation. These constants are *not* known to the estimator.

Task

Implement your solutions for the estimator design in the file `Estimator.m`. Your code has to run with the Matlab function `run.m`. Use exactly the function definition as given in the template `Estimator.m` for the implementation of your estimator. Your implementation should run with an average computation time for a single update below 0.01 seconds on the TA's laptop[1]. Points can be deducted for exceeding this value.
You are only allowed to use the basic MATLAB installation without any additional toolboxes. While the style of your code is not relevant for evaluation, points can be deduced for severe violation of common sense programming techniques, which result for example in considerable increased computation times.

[1] Core i7 CPU running at 2.8GHz, with 16GB of RAM.

Evaluation

For evaluating your solution, we will test your PF on the given problem data. Moreover, we will make suitable modifications to the parameters in `EstimatorConst.m` and `SimulationConst.m` and also test the robustness of your estimator on those variations and on different scenarios.

Deliverables

Your submission has to follow the instructions reported in the `Deliverables.pdf` ***file provided, as grading is automated. Submissions that do not conform will have points deducted.***