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 2022)

Programming Exercise #1 Topic: Kalman Filtering

Issued: May 4, 2022 Due: **June 15, 2022**

Hybrid Extended Kalman Filter for Tracking a Boat

You are driving a motor boat across a big windy 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.

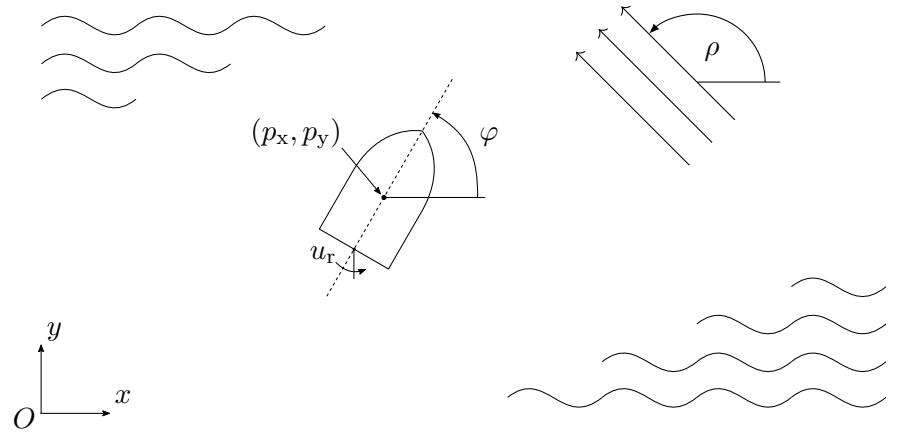

Figure 1: Schematic diagram of the boat and the fixed coordinate system (x, y) with origin O. The position of the boat is denoted by (p_x, p_y), the orientation with respect to the x-axis by φ and the control input to adjust the rudder by u_r. The wind blows at a constant absolute velocity, and its direction with respect to the x-axis is given by ρ.

The control inputs to the boat are the thrust command $u_t(t)$ and the rudder command $u_r(t)$. The boat produces thrust in the longitudinal direction and is affected by hydrodynamic and aerodynamic drag. 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),

$$\dot{p}_x = s_x \tag{1}$$

$$\dot{p}_y = s_y \tag{2}$$

$$\begin{aligned}\dot{s}_x = {} & \cos(\varphi)\left[\tanh(u_t) - C_{d,h}(s_x^2 + s_y^2)(1 + v_d)\right] \\ & - C_{d,a}\left(s_x - C_w\cos(\rho)\right)\sqrt{(s_x - C_w\cos(\rho))^2 + (s_y - C_w\sin(\rho))^2}\end{aligned} \tag{3}$$

$$\begin{aligned}\dot{s}_y = {} & \sin(\varphi)\left[\tanh(u_t) - C_{d,h}(s_x^2 + s_y^2)(1 + v_d)\right] \\ & - C_{d,a}\left(s_y - C_w\sin(\rho)\right)\sqrt{(s_x - C_w\cos(\rho))^2 + (s_y - C_w\sin(\rho))^2}.\end{aligned} \tag{4}$$

The $\tanh(u_t)$ term in (3) and (4) models the generated thrust of the boat, which is saturated for high thrust commands. The second term in the square brackets of (3), and (4) models the

hydrodynamic drag, which is a function of the total speed of the boat and a known fixed drag coefficient $C_{\mathrm{d,h}}$. The last term in (3), and (4) models the aerodynamic forces under the presence of a wind that blows at constant absolute velocity C_{w}. The known constant C_{w} is in the range $[0.5, 1.0]$. Further, the constant aerodynamic drag coefficient $C_{\mathrm{d,a}}$ is known. 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 a 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 dynamics of the direction of the wind are given by

$$\dot{\rho} = v_{\rho} \tag{8}$$

where $v_{\rho}(t)$ is continuous-time white noise with

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

for all times.

The boat receives distance measurements from three ground-based radio stations, placed at the exactly known positions $(x_{\mathrm{a}}, y_{\mathrm{a}})$, $(x_{\mathrm{b}}, y_{\mathrm{b}})$, and $(x_{\mathrm{c}}, y_{\mathrm{c}})$, respectively (see Figure 2). These measurements are affected by noise and 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,

$$\dot{b} = v_{\mathrm{b}} \tag{10}$$

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

$$E[v_{\mathrm{b}}(t)] = 0, \qquad E[v_{\mathrm{b}}(t)v_{\mathrm{b}}(t+\tau)] = Q_{\mathrm{b}}\delta(\tau). \tag{11}$$

In addition, angle measurements are also provided by a compass, which is not affected by drift, but is subject to higher measurement noise. The angle measurements are provided for all times. The measurement equations of the sensors are therefore given by

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

$$z_{\mathrm{b}}[k] = \sqrt{(p_{\mathrm{x}}[k] - x_{\mathrm{b}})^2 + (p_{\mathrm{y}}[k] - y_{\mathrm{b}})^2} + w_{\mathrm{b}}[k] \tag{13}$$

$$z_{\mathrm{c}}[k] = \sqrt{(p_{\mathrm{x}}[k] - x_{\mathrm{c}})^2 + (p_{\mathrm{y}}[k] - y_{\mathrm{c}})^2} + w_{\mathrm{c}}[k] \tag{14}$$

$$z_{\mathrm{g}}[k] = \varphi[k] + b[k] + w_{\mathrm{g}}[k] \tag{15}$$

$$z_{\mathrm{n}}[k] = \varphi[k] + w_{\mathrm{n}}[k] \tag{16}$$

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 values of σ_{c}^2, σ_{g}^2 and σ_{n}^2 are known and fixed, while the known constants

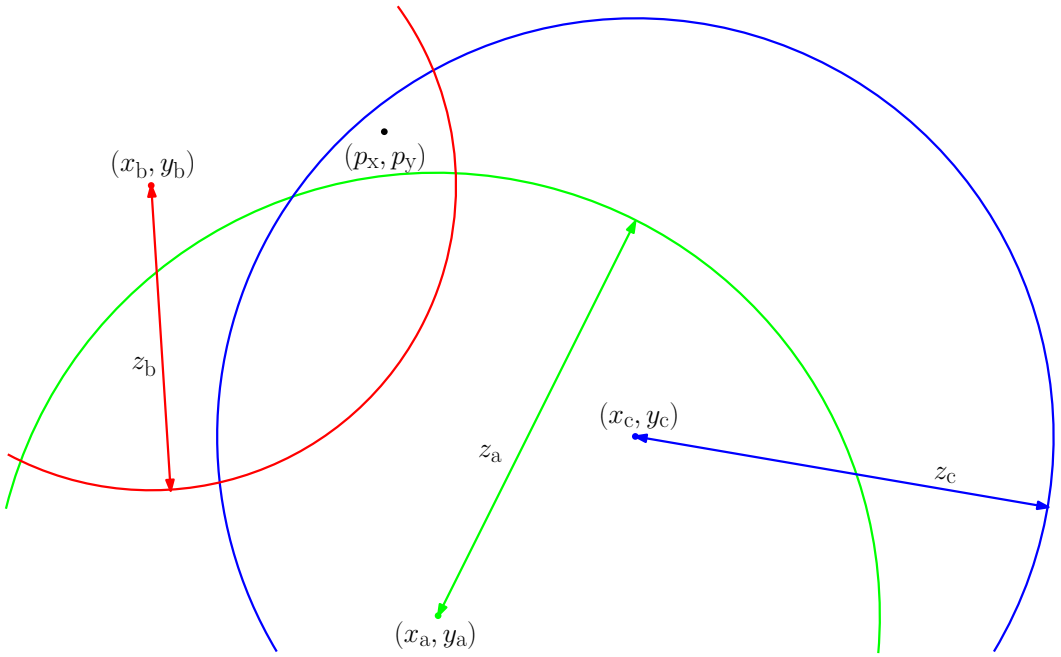

Figure 2: Example of distance measurements: each distance measurement indicates that the boat is positioned on the circumference of a circle centered at the respective radio station. Since the measurements are corrupted by noise, the three circles do not exactly intersect at the boat's position.

σ_a^2 and σ_b^2 are in the range $[15, 30]$. Note that we use $[\cdot]$ to denote discrete-time signals, e.g. $b[k] = b(kT_s)$.

The control inputs to the boat 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 $\varphi(0) = \varphi_0$, while the initial wind direction is given by $\rho(0) = \rho_0$. 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 $\varphi_0 \in [-\bar{\varphi}, \bar{\varphi}]$, while the probability density function of ρ_0 is uniformly distributed with $\rho_0 \in [-\bar{\rho}, \bar{\rho}]$. At $t = 0$ the gyroscope measurement is bias-free.

All random variables φ_0, ρ_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 $v_\mathrm{d}(t)$, $v_\mathrm{r}(t)$, $v_\rho(t)$ and $v_\mathrm{b}(t)$ are similarly independent for all times.

Objective

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 $z_\mathrm{g}[k]$, $z_\mathrm{n}[k]$ and occasionally $z_\mathrm{a}[k]$, $z_\mathrm{b}[k]$, and $z_\mathrm{c}[k]$. Furthermore, the constants Q_d, Q_r, Q_ρ, Q_b, $C_\mathrm{d,h}$, $C_\mathrm{d,a}$, C_w, C_r, σ_a, σ_b, σ_c, σ_g, σ_n, R_0, $\bar{\rho}$, and $\bar{\varphi}$ are known to the estimator. The position, orientation and linear velocity of the boat, the wind direction, and the gyroscope sensor drift are the estimator states. Note that the sensors do not provide direct measurements for all states. Therefore, the estimates of some states may be worse than the estimates of other states.

Provided Matlab Files

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

`run.m`	Matlab function 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 in the file `Estimator.m`. Your code has to run with the Matlab function `run.m`. You *must* use *exactly* the function definition as given in the template `Estimator.m` for the implementation of your estimator.
You are only allowed to use the basic MATLAB installation without any additional toolboxes. The `run.m` script will output an error whenever your estimator does not comply with this rule. In this case, the error message will contain the additional toolboxes on which your code depends. To determine whether a given function (e.g. the function `mean()`) depends on an additional toolbox, you may use the following instructions:

```
[~, pList] = matlab.codetools.requiredFilesAndProducts('mean');
{pList.Name}'
```

In case the function only depends on the basic MATLAB language (as is the case for the function `mean()`), the output should look as follows:

```
ans =
        {'MATLAB'}
```

While the style of your code is not relevant for evaluation, points will be deduced for severe violation of common sense programming techniques, which result for example in considerably increased computation times.
We recommend the use of the `ode45` function to solve differential equations. In order to pass and return matrices to and from `ode45`, you may reshape the matrices to vectors.

Evaluation

To evaluate your solution, we will test your EKF on the given problem data. Moreover, we will make suitable modifications to the parameters in `EstimatorConst.m` and `SimulationConst.m` and test the robustness of your estimator for different values of the constants C_w, σ_a^2 and σ_b^2 for the ranges defined above. The constants Q_d, Q_r, Q_ρ, Q_b, $C_\mathrm{d,h}$, $C_\mathrm{d,a}$, C_r, σ_c^2, σ_g^2, σ_n^2, R_0, $\bar{\rho}$, and $\bar{\varphi}$ are fixed and will not be modified during evaluation.

Deliverables

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