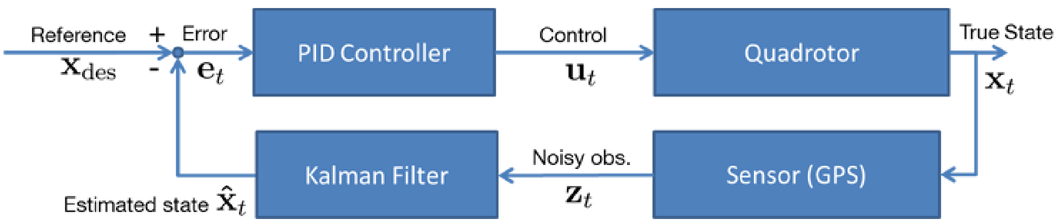
In this exercise you will implement your own Kalman filter to correct noisy sensor measurements. The output of the Kalman filter will be fed into a PD controller to guide the quadrotor along a square trajectory. The data flow of the whole system is illustrated in the following figure:



Help

Your task is to implement the prediction and correction step of the Kalman filter.

For sake of simplicity, the quadrotor is restricted to motion in the x-y-plane and its heading is neglected. Its state is defined by:

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix}$$

If we moreover assume constant velocities $\dot{x}$ and $\dot{y}$, we can predict the next state of the quadrotor after a time step Δt as:

$$\mathbf{x}_{t+1} = \begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \dot{x}_{t+1} \\ \dot{y}_{t+1} \end{pmatrix} = \begin{pmatrix} x_t + \dot{x}_t \cdot \Delta t \\ y_t + \dot{y}_t \cdot \Delta t \\ \dot{x}_t \\ \dot{y}_t \end{pmatrix}$$

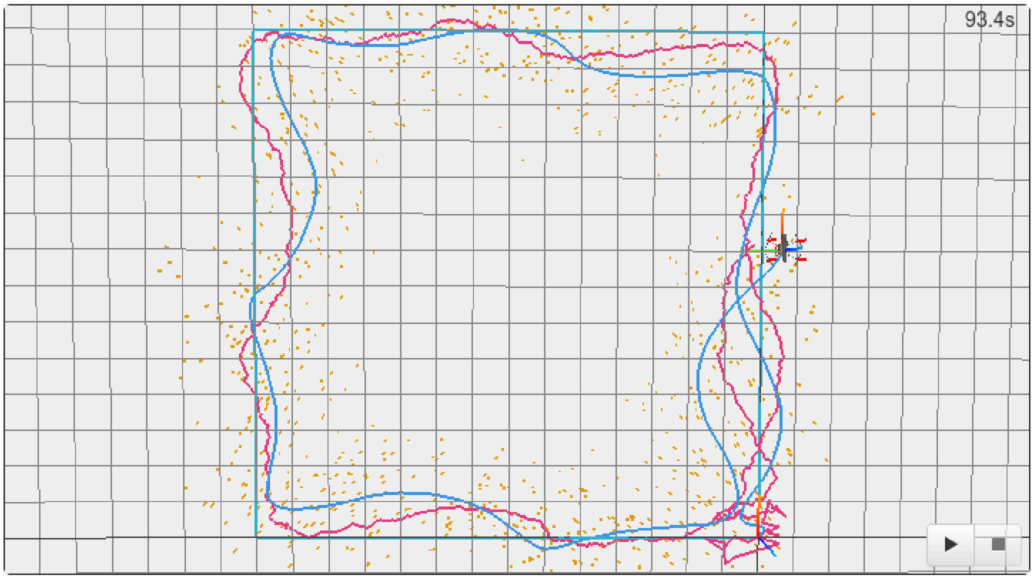
The prediction can be expressed as a matrix multiplication of the state transition matrix $\mathbf{A}$ with the state vector $\mathbf{x}_t$, i.e., $\mathbf{x}_{t+1} = \mathbf{A} \cdot \mathbf{x}_t$ with

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The code below implements a linear Kalman filter. The state_callback method is invoked in every simulation step and performs one Kalman filter prediction step (200Hz). The measurement_callback method is called when a new GPS sensor reading arrives and calculates the Kalman filter correction step (13Hz). The GPS reading gives a noisy measurement of the absolute position in world coordinates.

You have to **implement** the predictState and the correctState method. The covariance prediction, correction and the Kalman gain calculation are already implemented.

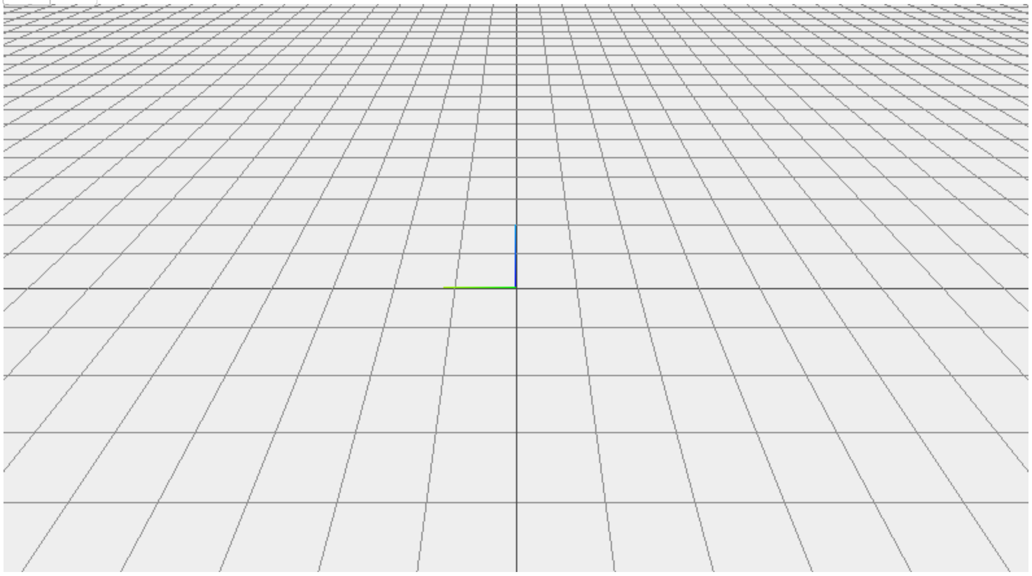
A valid solution will give you a trajectory which looks similar to the following example (top-down view, zoomed out):



The noisy GPS measurements are visualized with orange flares. Initially, the quadrotor is located at the world origin. From these measurements, your task is to estimate the pose of the quadrotor using the Kalman filter. The estimated trajectory is visualized in pink. This estimated pose is then fed into a PD controller to generate the control commands to move the quadrotor along a square trajectory. The actual trajectory of the quadrotor is then visualized with the blue line. The goal of this exercise is to finish the Kalman filter such that the estimated trajectory (pink) matches the true trajectory (blue) as good as possible. Note that the match will not be perfect, because of noise of the measurements and time delays of the estimation. The grader will check whether the estimated trajectory is sufficiently close to the true trajectory.

The PD controller will access the self.x member variable to get the current state! Please do not modify the measurement_callback and state_callback method.

KALMAN FILTER IN 2D (15/15 points)



Check | Save | Show Answer *You have used 1 of 6 submissions*

STATE AND SENSOR NOISE (2/2 points)

What happens to the covariance ellipse if the values in the state noise matrix Q are decreased?

- it grows
- it shrinks ✔
- it stays the same

What happens to the covariance ellipse if the values in the sensor noise matrix R are increased?

- it grows ✔
- it shrinks
- it stays the same

Final Check | Save | Show Answer | *You have used 1 of 2 submissions*

Show Discussion

New Post