Localization and control of a mobile robot with range and velocity measurements

I. Introduction

Consider a home robot vacuum cleaner that wants to come back to its docking station. The information that we can obtain from the system is the velocity of the robot, e.g. odometry at 1 KHz with a standard deviation of 0.1 m/s for each velocity component, and the distance of the robot from the docking station, e.g., from a radio chip at 1 Hz with a standard deviation of 0.3 meters.

II. Model

We model the robot as follows

$$\dot{p} = u, \tag{1}$$

where $p, u \in \mathbb{R}^2$ are the position of the robot with respect the docking stations and the control action on the robot respectively. Consequently, the control action u commands the velocity of the robot.

III. Position estimation

The position of the robot will be estimated by a discrete Kalman filter with state vector $x = p$. In particular, if we describe the elements of the state vector x as Gaussian distributions, the dynamics of the mean is given by

$$\hat{x}(k+1) = F\hat{x}(k) + Gu(k), \tag{2}$$

where $F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $G = \begin{bmatrix} \Delta T & 0 \\ 0 & \Delta T \end{bmatrix}$ with ΔT being the time interval between measurements from the odometry. Therefore, the dynamics of the variances of the state vector x is given by

$$P(k+1) = FP(k)F^T + G\sigma_{u(k)}\sigma_{u(k)}^T G^T, [1] \tag{3}$$

where $\sigma_{u(k)} \in \mathbb{R}^2$ is the stacked vector with the standard deviation of the elements (robot's velocity) of the input u.

Everytime the system measures the distance $d = h(p) = \sqrt{p^T p}$, the discrete Kalman filter corrects the state vector x. Since the function $h(\hat{x}) = \sqrt{\hat{x}^T \hat{x}}$ is not linear, the corresponding Jacobian is

$$H = \nabla h(\hat{x}) = \begin{bmatrix} \frac{\partial h(\hat{x})}{\partial \hat{x}_1} & \frac{\partial h(\hat{x})}{\partial \hat{x}_2} \end{bmatrix}, \tag{4}$$

so we can calculate the Kalman gain

$$K = PH^T(HPH^T + H\sigma_r\sigma_r^T H^T)^{-1}, [2] \tag{5}$$

where $\sigma_r \in \mathbb{R}^m$ is the stacked vector of the standard deviations of the elements in the vector of measurements $r \in \mathbb{R}^m$ (in this case $r = d$, so $m = 1$), and we update the state vector of the Kalman filter as follows

$$\hat{x}_u = \hat{x} + K\Big(r - h(\hat{x})\Big) \tag{6}$$

$$P_u = P - KHP, \tag{7}$$

and we continue iterating (2) and (3) with initial values x_u and P_u.

IV. Exercise I

Consider $u = \begin{bmatrix} 5\sin(t) \\ 10\cos(t) \end{bmatrix}$, then implement the Kalman filter from the previous section in the given Python script. Discuss with the teacher when it is a good idea to update the state vector with the range measurements.

V. Exercise II

Design with the Lyapunov approach two different control actions u_1 and u_2 to guide the robot to the docking station. Discuss with the teacher what can go wrong if we use the estimation x from the Kalman filter in the controller.

VI. Exercise III

Consider the following desired path to be tracked by the vehicle

$$f(x) := x_1^2 + x_2^2 - r^2 = 0, \tag{8}$$

where $x_i, i = \{1, 2\}$ are the components of x, and r is the radius os the desired circle. Please, design a guidance vector field to drive the vehicle to travel on the desired trajectory with a speed of 2 m/s. For example, by first normalizing the vector field, so you can have control on the commanded speed. Since the guidance vector field $g(p)$ commands velocity, then $u = g(p)$. Discuss with the teacher why the initial conditions of your Kalman filter must start *close enough* to the actual position of the vehicle.

[1] Remember, this expression holds iff the states and the inputs are independent variables, and $G\sigma_{u(k)}\sigma_{u(k)}^T G^T$ (also known as *process noise* or Q) holds iff all the elements of the input vector are also independent variables.

[2] The expression $H\sigma_r\sigma_r^T H^T$ is also known as *measurement noise* or R.