UKF with 7D State Vector

November 21, 2023

1 Design

1.1 State Vector

$$\mathbf{x}_t = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \psi \end{bmatrix} \tag{1}$$

1.2 Prediction Step

1.2.1 Control Input

We define a control input $\mathbf{u}$ populated by linear accelerations from the IMU:

$$\mathbf{u}_t = \begin{bmatrix} \ddot{x}^b \\ \ddot{y}^b \\ \ddot{z}^b \end{bmatrix} \tag{2}$$

The linear accelerations are in the drone's body frame, so we need to rotate these vectors into the global frame based on the yaw variable that we are tracking and the roll and pitch values from the IMU as well. This transformation will occur in the state transition function.

1.2.2 State Transition Function

We first note that the rotation from the body frame to the global frame can be achieved by a rotation matrix $R(\psi)$ that is dependent on the yaw angle ψ. For simplicity, let us consider only the yaw angle so that the rotation matrix looks like something along the lines of this:

$$R(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

At this point, we can apply this rotation in the acceleration in the body frame to get the acceleration in the global frame:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = R(\psi)\mathbf{u_t} \tag{4}$$

. At this point, we can use kinematics to construct the state transition function considering that the change in position is the velocity times the change in Δt, and the change in velocity is the change in acceleration times the change in Δt. We get the yaw rate from the camera. So, knowing that our state transition function g predicts the next state $\mathbf{x_t}$ based on the previous state, control inputs, and elapsed time, we can write the state transition function

$$g(\mathbf{x}_{t-\Delta t}, \mathbf{u}_t, \Delta t) \tag{5}$$

as:

$$\mathbf{x_t} = \mathbf{x_t} - \mathbf{\Delta t} + \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \psi_{camera} \end{bmatrix} \tag{6}$$

1.3 Measurement Update Step

1.3.1 Measurement Vector

$$\mathbf{z}_t = \begin{bmatrix} r \\ x \\ y \\ \dot{x} \\ \dot{y} \\ \psi_{\text{camera}} \end{bmatrix} \tag{7}$$

1.3.2 Measurement Function

We know from section **1.2** that our state vector is defined as:

$$\mathbf{x_t} = \mathbf{x_t} - \mathbf{\Delta t} + \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \psi_{camera} \end{bmatrix} \tag{8}$$

. Using our state vector x_t, we can define $h(\mathbf{x_t})$ to map from the state space to the measurement space as follows:

$$h(\mathbf{x_t}) = \begin{bmatrix} h_1(\mathbf{x_t}) \\ h_2(\mathbf{x_t}) \\ h_3(\mathbf{x_t}) \\ h_4(\mathbf{x_t}) \\ h_5(\mathbf{x_t}) \\ h_6(\mathbf{x_t}) \end{bmatrix} \tag{9}$$

where $h_1(x_t)$ is the slant range r, which can be calculated from the altitude z and the angles ϕ and θ, knowing that the ToF sensor is oriented downwards: $h_1(x_t) = \sqrt{z^2/(cos(\theta)^2 cos(\phi)^2)}$, $h_2(x_t)$ and $h_3(x_t)$ are the x and y positions from the state vector, $h_4(x_t)$ and $h_5(x_t)$ are the velocities $\dot{x}, \dot{y}$, and $h_6(x_t)$ is the yaw estimate from the drone's camera ψ_{camera}.

As a result, this means that our measurement function is:

$$h(\bar{\mathbf{x}}_t) = \begin{bmatrix} \sqrt{z^2/(cos(\theta)^2 cos(\phi)^2)} \\ x \\ y \\ \dot{x} \\ \dot{y} \\ \psi \end{bmatrix} \tag{10}$$

1.3.3 Measurement Covariance Matrix

Let us assume that we have the following measurements: ToF slant range reading variance σ_r^2, x and y planar position estimates variances σ_x^2 and σ_y^2, variances of the x and y velocities $\sigma_{\dot{x}}^2$ and $\sigma_{\dot{y}}^2$, and the variance of the yaw estimate from the camera $\sigma_{psi_{camera}}^2$. Then, our measurement covariance matrix R_t would look like:

$$\mathbf{R_t} = \begin{bmatrix} \sigma_r^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_x^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_y^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{x}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{\dot{y}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\psi_{\text{camera}}}^2 \end{bmatrix} \tag{11}$$