Introduction to Filtering

Vijay Kumar
University of Pennsylvania

April 13, 2015

1 The Basic Idea

The Kalman filter is a tractable instance of the Bayes filter and allows us to compute an estimate of the state vector x and its distribution at time t from estimate at time $t-1$. In order to make it tractable, we must make three assumptions:

1. The process model, $p(x_t|u_t, x_{t-1})$, is linear with additive noise. In other words,

$$x_t = Ax_{t-1} + B_t u_{t-1} + n_t \tag{1}$$

   where A is a nonsingular $n \times n$ matrix, B is a $n \times m$ matrix, u is a $m \times 1$ vector of inputs that is applied between time $t-1$ and t, the process noise n_t is Gaussian white noise[1] with zero mean and covariance Q. We denote this by $n_t \sim \mathcal{N}(0, Q)$.

2. The measurement model, $p(z_t|x_t)$, is linear with additive noise. In other words,

$$z_t = Cx_t + v_t \tag{2}$$

   where z_t is a $p \times 1$ vector, C is a $p \times n$ matrix, and the measurement noise v_t is Gaussian white noise with zero mean and covariance R, or $v_t \sim \mathcal{N}(0, R)$.

3. The prior on the state x is $p(x_0)$ is normally distributed with a known mean μ_0 and covariance Σ_0, or $p(x_0) \sim \mathcal{N}(\mu_0, \Sigma_0)$.

The filter consists of two steps, the *prediction step* that relates the predicted state, $x_{\bar{t}}$ at time $\bar{t}$ (think of $\bar{t}$ as a time instant just before t) to the estimated state x_{t-1} at time $t-1$, and the *update step* that computes the estimated state x_t at time t.

2 Facts about Gaussian Distributions

Definition 1 *Multivariate normal distribution*

Let X be a vector of random variables. $f_X(x) : \mathbb{R}^n \to \mathbb{R}$ is a multivariate normal distribution (Gaussian) if it takes the form:

$$f_X(x) = \frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{\det \Sigma}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right],$$

where $\mu \in \mathbb{R}^n$ and $\Sigma \in \mathbb{R}^{n\times n}$ is a $n \times n$, symmetric positive definite matrix.

[1] White noise is a signal whose samples are a sequence of serially independent random variables, with each sample having a normal distribution with zero mean.

If the density function of X is $f_X(x)$ as defined above, X is said to be multivariate normal. You should be able to verify[2] that $f_X(x) \geq 0$, $\int_{\mathbb{R}^n} f_X(x)dx = 1$, $\int_{\mathbb{R}^n} xf_X(x)dx = \mu$, and $\Sigma = E((X-\mu)((X-\mu)^T)$.

Definition 2 *Joint Normal random vectors: If X is a n-dimensional multivariate Gaussian, and if*

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix},$$

X_1 and X_2 are both (multivariate) Gaussians and are jointly normally distributed.

Remark 1 *If X_1 and X_2 are Gaussians it does n ot necessarily imply that*

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

is also a Gaussian. However, if

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

is a normal distribution, X_1 and X_2 are both normally distributed.

Fact 1 *If*

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

is a multivariate Gaussian and X_1 and X_2 are uncorrelated, i.e., the covariance matrix has the form:

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}, \quad \Sigma_{12} = \Sigma_{21} = 0,$$

X_1 and X_2 are independent and therefore we can write

$$f_X(x) = f_{X_1}(x_1) f_{X_2}(x_2).$$

Remark 2 *If X_1 and X_2 are independent, they are also uncorrelated. The converse of Fact 1 does not require the Gaussian assumption. Independent random variables are always uncorrelated.*

Fact 2 *The sum of independent Gaussian random variables is also Gaussian. Let X, Y be two independent Gaussian random variables with means μ_X, μ_Y and covariances Σ_X, Σ_Y, respectively. Then their sum, $Z = X + Y$, is also Gaussian with mean $\mu_Z = \mu_X + \mu_Y$ and covariance $\Sigma_Z = \Sigma_X + \Sigma_Y$.*

Fact 3 *Affine transformations of Gaussian distributions are Gaussian. If X is a n-dimensional multivariate Gaussian with the mean μ_X and covariance Σ_X, and if $Y = AX + b$ where A is a $n \times n$ matrix and $b \in \mathbb{R}^n$, then Y is a multivariate Gaussian with mean μ_Y and covariance Σ_Y given by:*

$$\mu_Y = A\mu_X + b, \quad \Sigma_Y = A\Sigma_X A^T.$$

The next two results may not be obvious but are well known. See [1] for example. A particularly nice introduction for the uninitiated is provided in [5].

[2] See Wikipedia for the evaluation of the Gaussian integral.

Fact 4 *Conditional density of a multivariate normal distribution. If*

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

is a multivariate Gaussian, with the means μ_1 *and* μ_2 *for the vectors* X_1 *and* X_2 *and the covariance*

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix},$$

the conditional density $f_{X_1|X_2}(x_1|X_2 = x_2)$ *is a multivariate normal distribution for a given* x_2 *with the mean and covariance*

$$\mu_{X_1|X_2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2), \quad \Sigma_{X_1|X_2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}.$$

Fact 5 $X_1 - \mu_{X_1|X_2}$ *and* X_2 *are independent. To prove this first show that* $X_1 - \mu_{X_1|X_2}$ *and* X_2 *are joint normal (the vector is a affine transformations of* $X = [X_1, X_2]$*) and show the covariance is zero.*

3 The Kalman Filter

An overview of the two steps and the final equations are provided in the schematic in Figure 1.

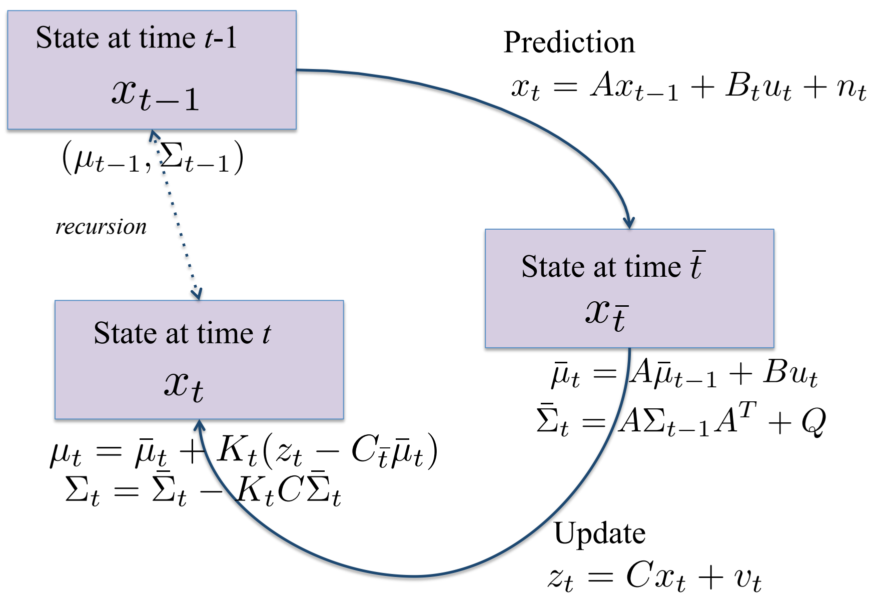

Figure 1: The Kalman Filter

Prediction Step This step computes the predicted state, $x_{\bar{t}}$ (and its distribution) at time $\bar{t}$ (think of $\bar{t}$ as a time instant just before t, but $\bar{t}$ and t have the same value) from the estimated state x_{t-1} (and its distribution) at time $t-1$. Because the process model in (1) is an affine transformation of the prior state and the sum of independent Gaussian random variables (since the noise is independent of the initial state) from Fact 3 and Fact 2,

$$\bar{\mu}_t = A\bar{\mu}_{t-1} + Bu_t \tag{3}$$

$$\bar{\Sigma}_t = A\Sigma_{t-1}A^T + Q \tag{4}$$

Update Step The update step computes the estimated state x_t (and its distribution) at time t from the prediction at time $\bar{t}$ in (3,4). We known that without the measurement z_t, the best we can do is write $x_t = x_{\bar{t}}$. But if we combine this with (2) we can write

$$\begin{aligned} x_t &= x_{\bar{t}} \\ z_t &= Cx_t + v_t \end{aligned}$$

to get

$$\begin{bmatrix} x_t \\ z_t \end{bmatrix} = \begin{bmatrix} I & 0 \\ C & I \end{bmatrix} \begin{bmatrix} x_{\bar{t}} \\ v_t \end{bmatrix} \tag{5}$$

From Fact 3, we know that $\begin{bmatrix} x_t \\ z_t \end{bmatrix}$ is joint normal with the mean given by

$$\begin{bmatrix} \bar{\mu}_t \\ C\bar{\mu}_t \end{bmatrix}$$

and covariance given by

$$\begin{bmatrix} I & 0 \\ C & I \end{bmatrix} \begin{bmatrix} \bar{\Sigma}_t & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} I & C^T \\ 0 & I \end{bmatrix} = \begin{bmatrix} \bar{\Sigma}_t & \bar{\Sigma}_t C^T \\ C\bar{\Sigma}_t & C\bar{\Sigma}_t C^T + R \end{bmatrix}.$$

We are interested in the conditional distribution of x_t given the measurement z_t. From Fact 4, we know that $p(x_1|X = x_2)$ is normal. Therefore $p(x_t|Z_t = z_t)$ is normal and the conditional mean and covariance are given by:

$$\mu_{x_t|z_t} = \bar{\mu}_t + \bar{\Sigma}_t C^T (C\bar{\Sigma}_t C^T + R)^{-1}(z_t - C\bar{\mu}_t) \tag{6}$$

$$\Sigma_{x_t|z_t} = \bar{\Sigma}_t - \bar{\Sigma}_t C^T (C\bar{\Sigma}_t C^T + R)^{-1} C\bar{\Sigma}_t. \tag{7}$$

If we denote by K_t the Kalman gain,

$$K_t = \bar{\Sigma}_t C^T (C\bar{\Sigma}_t C^T + R)^{-1},$$

we can write (without denoting explicitly the fact that the expectation and covariance are conditioned on the measurement):

$$\mu_t = \bar{\mu}_t + K_t(z_t - C\bar{\mu}_t) \tag{8}$$

$$\Sigma_t = \bar{\Sigma}_t - K_t C\bar{\Sigma}_t. \tag{9}$$

Thus the Kalman filter is given by Equations (3, 4, 8, 9).

The Kalman gain effectively represents a tradeoff between the prediction and the update. If the confidence in the prediction is high relative to the measurement (Q is small, R is large), K_t is small and the distribution at time t will be more strongly influenced by the prediction.

Remark 3 *If the distribution is not Gaussian but the white noise is uncorrelated with x_0, the Kalman filter represents the minimum variance linear estimator. In other words, if we want a linear estimator of x_t, Equations (3-9) represent the best estimator. See [1].*

4 The Extended Kalman Filter

If the system is nonlinear, the methodology of the previous section can be applied with a linearized model of the system, and updating this model at every time step by linearizing about the current state.

Let the continuous time process model be given by a nonlinear, stochastic differential equation:

$$\dot{x} = f(x, u, n) \tag{10}$$

where x and u are the state and input respectively, and n is Gaussian white noise, $n(t) \sim \mathcal{N}(0, Q)$. The observation model is given by:

$$z = h(x, v) \tag{11}$$

where z is the measurement vector and v is Gaussian white noise, $v(t) \sim \mathcal{N}(0, R)$.

As before, we consider a time interval, $(t', \bar{t})$, where t' is the same as the previous time step $t-1$. The prediction phase requires us to integrate the process model over this time interval $\delta t = \bar{t} - t'$:

$$x_{\bar{t}} = \Phi(t; x_{t-1}, u, n). \tag{12}$$

This integration may be difficult to do analytically. However, any numerical integration scheme can be used to perform this integration. For example, a simple one-step Euler integration yields the update:

$$x_{\bar{t}} = x_{t-1} + f(x_{t-1}, u_t, n_t)\, \delta t. \tag{13}$$

Alternatively, one can linearize this system about the most likely previous pose, *i.e.,* the mean, $(x = \mu_{t-1}, u = u_t, n = 0)$ to write:

$$\begin{aligned} f(x,u,n) &\approx f(\mu_{t-1}, u_t, 0) + \underbrace{\left.\frac{\partial f}{\partial x}\right|_{(\mu_{t-1},u_t,0)}}_{A_t}(x - \mu_{t-1}) + \underbrace{\left.\frac{\partial f}{\partial u}\right|_{(\mu_{t-1},u_t,0)}}_{B_t}(u - u_t) + \underbrace{\left.\frac{\partial f}{\partial n}\right|_{(\mu_{t-1},u_t,0)}}_{U_t}(n - 0) \\ &= f(\mu_{t-1}, u_t, 0) + A_t(x - \mu_{t-1}) + B_t(u - u_t) + U_t(n - 0) \end{aligned} \tag{14}$$

and explicitly integrate this matrix differential equation over the time interval (t', t) to get $\Phi(t; x_{t-1}, u, n)$. We can also use the linear model to approximate the next pose using (13) and (14)

$$\begin{aligned} x_{\bar{t}} &\approx x_{t-1} + f(x_{t-1}, u_t, n_t)\, \delta t \\ &\approx x_{t-1} + \big(f(\mu_{t-1}, u_t, 0) + A_t(x_{t-1} - \mu_{t-1}) + B_t\cancelto{0}{(u_t - u_t)} + V_t(n_t - 0)\big)\, \delta t \\ &= \underbrace{(I + A_t\, \delta t)}_{F_t} x_{t-1} + \underbrace{(U_t\, \delta t)}_{V_t} n_t + \underbrace{\big(f(\mu_{t-1}, u_t, 0) - A_t \mu_{t-1}\big)\, \delta t}_{b_t} \\ &= F_t x_{t-1} + b_t + V_t n_t \end{aligned} \tag{15}$$

Using the linearized models from (13)–(15) and Facts 2–3, we are able to find the predicted mean and covariance of the state. The mean is

$$\begin{aligned} \bar{\mu}_t &= F_t \mu_{t-1} + b_t \\ &= (I + A_t\, \delta t)\mu_{t-1} + \big(f(\mu_{t-1}, u_t, 0) - A_t \mu_{t-1}\big)\, \delta t \\ &= \mu_{t-1} + f(\mu_{t-1}, u_t, 0)\, \delta t, \end{aligned} \tag{16}$$

and the covariance is given by

$$\bar{\Sigma}_t = F_t \Sigma_{t-1} F_t^T + V_t Q_t V_t^T. \tag{17}$$

For the update phase, we similarly linearize the observation model $z = h(x, v)$ around the most likely predicted state $(x = \bar{\mu}_t, v = 0)$ to be able to use the results of the previous section and write:

$$\begin{aligned} z_t &\approx h(\bar{\mu}_t, 0) + \underbrace{\left.\frac{\partial h}{\partial x}\right|_{(\bar{\mu}_t,0)}}_{C_t} (x - \bar{\mu}_t) + \underbrace{\left.\frac{\partial h}{\partial v}\right|_{(\bar{\mu}_t,0)}}_{W_t} (v - 0) \\ &= h(\bar{\mu}_t, 0) + C_t(x - \bar{\mu}_t) + W_t(v - 0) \end{aligned} \tag{18}$$

The Kalman Filter was based on the linear model in (1) and (2). We are now in a position to write the equations for the *Extended Kalman Filter* using either (12) or (14) instead of (1), and (18) instead of (2).

Prediction Step We compute the predicted state, $x_{\bar{t}}$ (and its distribution) at time $\bar{t}$ from the estimated state x_{t-1} (and its distribution) at time $t-1$. The expected state, $\bar{\mu}_t$, is obtained either by integration over the time interval $(t-1, \bar{t})$:

$$\bar{\mu}_t = \Phi(\bar{t}; \mu_{t-1}, u_t, 0) \tag{19}$$

or from the linear approximation at time $t-1$:

$$\bar{\mu}_t = \mu_{t-1} + f(\mu_{t-1}, u_t, 0)\, \delta t. \tag{20}$$

Either way we can invoke the affine process model in (20) to find the covariance using Fact 3:

$$\bar{\Sigma}_t = F_t \Sigma_{t-1} F_t^T + V_t Q_t V_t^T \tag{21}$$

For the update step we use the linear approximation using the state $\bar{\mu}_t$ at time $\bar{t}$ to write:

$$z_t = h(\bar{\mu}_t, 0) + C_t(x_{\bar{t}} - \bar{\mu}_t) + W_t(v_t - 0) \tag{22}$$

Following the same derivation as for the Kalman Filter we get:

$$\begin{bmatrix} x_t \\ z_t \end{bmatrix} = \begin{bmatrix} I & 0 \\ C_t & W_t \end{bmatrix} \begin{bmatrix} x_{\bar{t}} \\ v_t \end{bmatrix} + \begin{bmatrix} 0 \\ h(\bar{\mu}_t, 0) - C_t \bar{\mu}_t \end{bmatrix} \tag{23}$$

which is an affine transformation and using Fact 3 we get:

$$\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t, 0)) \tag{24}$$

$$\Sigma_t = \bar{\Sigma}_t - K_t C_t \bar{\Sigma}_t. \tag{25}$$

where

$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + W_t R_t W_t^T)^{-1}.$$

5 Application to Quadrotors

In our approach we do not explicitly model the inputs (the propellor speeds or the motor currents) since often these inputs can be quite different from the commanded inputs and difficult to measure in small flying robots. We instead define our state so that their derivatives can be written in terms of sensed quantities obtained from the Inertial Measurement Unit (IMU). Said another way, we will abstract out the dynamics so we can write the model in terms of kinematic quantities.

There are two approaches to doing this. First, we can assume that we have a very good velocity sensor that gives you excellent estimates of the linear velocity of the vehicle. This allows us to eliminate the velocity (and the acceleration) from the state vector as we will see below in Section 5.1. Alternatively, we can relax this assumption and instead use the acceleration information obtained from the IMU. See Fig. 2. This forces us to retain the velocity in the state vector but eliminate the motor torques and the propellor speeds from the process model. This is explained in Section 5.2.

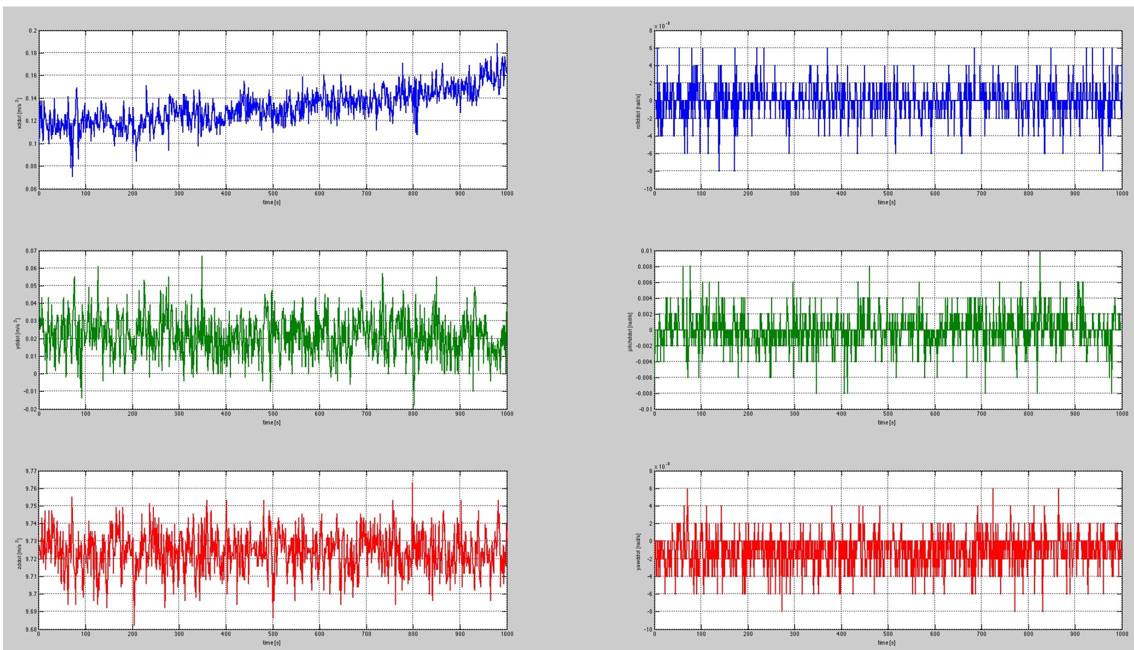

Figure 2: A sample data set from the IMU used in the lab. The measured accelerations (left) and the angular velocity (right) are modeled by a bias and an additive noise.

5.1 Model A: Quadrotor with a good velocity sensor

Process Model We write the state vector, $\mathbf{x} \in \mathbb{R}^9$, as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \mathbf{b}_g \end{bmatrix}$$

where $\mathbf{p}$ is the position, $\mathbf{q}$ is a 3×1 vector parametrization of $SO(3)$, and $\mathbf{b}_g$ is the gyro bias. In what follows we use the Euler angle parameterization (despite its limitations when the rotation is significantly different from the identity element):

$$\mathbf{q} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix},$$

where ψ is the yaw angle, ϕ is the roll angle, and θ is the pitch following the $Z-X-Y$ Euler angle notation. Recall that the rotation matrix is given by

$$R(\mathbf{q}) = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix}.$$

and the components of the angular velocity in the body frame (aligned with the axes of the gyro) are given by:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

We will call the coefficient matrix above G and we can write:

$$\omega = G(\mathbf{q})\dot{\mathbf{q}} \tag{26}$$

We assume we have good velocity measurements $\mathbf{v}_m$ in the inertial frame from the VICON motion capture camera system. The gyro measurements ω_m are in the body-fixed frame and have a bias $\mathbf{b}_g$ that drifts over time. For the gyro measurements we write:

$$\omega_m = \omega + \mathbf{b}_g + \mathbf{n}_g \tag{27}$$

where $\mathbf{n}_g$ is an additive measurement noise modeled by a Gaussian, white noise. The velocity measurement $\mathbf{v}_m$ is given by:

$$\mathbf{v}_m = \dot{\mathbf{p}} + \mathbf{n}_v \tag{28}$$

where $\mathbf{n}_v$ is an additive Gaussian, white noise. We assume that the drift in the bias vector is described by Gaussian, white noise random processes:

$$\dot{\mathbf{b}}_g = \mathbf{n}_{bg}(t) \tag{29}$$

where $\mathbf{n}_{bg} \sim \mathcal{N}(0, Q_g)$.

Without explicitly modeling the inputs (the propellor speeds or the motor currents), we develop the process model from Equations (26-35) as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{v}_m - \mathbf{n}_v \\ G(\mathbf{x}_2)^{-1}(\omega_m - \mathbf{b}_g - \mathbf{n}_g) \\ \mathbf{n}_{bg} \end{bmatrix}. \tag{30}$$

Measurement model We will use a camera to measure the pose of the robot in an inertial frame. We use the theory of projective transformations and solutions of the 2D-3D pose discussed in the computer vision segment of the class. Specifically we saw how to estimate the pose of the robot using a minimum of four features (*e.g.*, April Tags) on the ground plane. In other words, given the projection equations for four features ($i = 1, \ldots, 4$):

$$\begin{bmatrix} u_i \\ v_i \\ w \end{bmatrix} \sim K \left[R(\mathbf{q}) \;\; \mathbf{p}\right]_{3\times 4} \begin{bmatrix} X_i \\ Y_i \\ 0 \\ W \end{bmatrix},$$

or

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = H_{3\times 3} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix}, \quad H \sim K \left[\bar{R}(\mathbf{q}) \;\; \mathbf{p}\right]_{3\times 3}$$

where $\bar{R}$ is formed by deleting the third column of the rotation matrix. With four features we can estimate H up to scale, and with a known K we can recover R by solving for $K^{-1}H$, constructing H' by replacing the last column of $K^{-1}H$ with the cross product of the first two rows, and then finding R:

$$R = \underset{SO(3)}{\arg\min} \|R - H'\|^2.$$

Once R is known, we can solve for $\mathbf{p}$ from H.

Thus our measurement model is linear because we can obtain the rotation matrix R, and therefore the vector $\mathbf{q}$ as well as the position $\mathbf{p}$:

$$\mathbf{z} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} = C\mathbf{x} \tag{31}$$

where C is a 6×9 matrix:

$$C = \begin{bmatrix} I_{3\times 3} & 0 & 0 \\ 0 & I_{3\times 3} & 0 \end{bmatrix}$$

Extended Kalman Filter Equations (30) and (31) should be sufficient to write the EKF. Note that (30) is nonlinear and the linearization of the equations might require the use of a symbolic manipulator like Mathematica or Matlab.

5.2 Model B Quadrotor with a good acceleration sensor

Process Model In this model, we assume that we cannot directly measure velocity but can measure acceleration using an accelerometer. Accordingly, the state vector, $\mathbf{x} \in \mathbb{R}^{15}$ is written as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \end{bmatrix} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \dot{\mathbf{p}} \\ \mathbf{b}_g \\ \mathbf{b}_a \end{bmatrix}$$

where $\mathbf{p}$ is the position and $\dot{\mathbf{p}}$ is the velocity of the center of mass, $\mathbf{q}$ is a 3×1 vector parametrization of $SO(3)$, $\mathbf{b}_g$ is the gyro bias, and $\mathbf{b}_a$ is the accelerometer bias. As in Section 5.1, we use the Euler angle parameterization and write the components of the angular velocity in the body frame which are aligned with the axes of the accelerometer and the gyro. The gyro measurements ω_m and the accelerometer measurements $\mathbf{a}_m$ are in the body-fixed frame and have biases $\mathbf{b}_g$ and $\mathbf{b}_a$ that drift over time. For the gyro measurements we write:

$$\omega_m = \omega + \mathbf{b}_g + \mathbf{n}_g \tag{32}$$

where $\mathbf{n}_g$ is an additive measurement noise modeled by a Gaussian, white noise. The accelerometer measurement $\mathbf{a}_m$ is given by:

$$\mathbf{a}_m = R(\mathbf{q})^T(\ddot{\mathbf{p}} - \mathbf{g}) + \mathbf{b}_a + \mathbf{n}_a \tag{33}$$

where $\mathbf{g}$ is the acceleration due to gravity in the inertial frame (the $[0, 0, 1]^T$ vector), and $\mathbf{n}_a$ is an additive Gaussian, white noise. We assume that the drift in the bias vectors are described by Gaussian, white noise random processes:

$$\dot{\mathbf{b}}_g = \mathbf{n}_{bg}(t) \tag{34}$$

$$\dot{\mathbf{b}}_a = \mathbf{n}_{ba}(t) \tag{35}$$

where $\mathbf{n}_{bg} \sim \mathcal{N}(0, Q_g)$ and $\mathbf{n}_{ba} \sim \mathcal{N}(0, Q_a)$.

Without explicitly modeling the inputs (the propellor speeds or the motor currents), we develop the process model from Equations (26-35) as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{x}_3 \\ G(\mathbf{x}_2)^{-1}(\omega_m - \mathbf{b}_g - \mathbf{n}_g) \\ \mathbf{g} + R(\mathbf{q})(\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a) \\ \mathbf{n}_{bg} \\ \mathbf{n}_{ba} \end{bmatrix}. \tag{36}$$

Measurement model As in Section 5.1, we use a camera to measure the pose of the robot in an inertial frame. We can also use the onboard camera to estimate the velocity of the robot from optical flow. Alternatively the VICON camera measurements can also be used to provide a measurement of the velocity.

Thus our measurement model is linear because we can obtain the rotation matrix R, and therefore the vector $\mathbf{q}$ as well as the position $\mathbf{p}$ and velocity $\dot{\mathbf{p}}$:

$$\mathbf{z} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \\ \dot{\mathbf{p}} \end{bmatrix} = C\mathbf{x} \tag{37}$$

where C is a 9×15 matrix:

$$C = \begin{bmatrix} I_{3\times3} & 0 & 0 & 0 & 0 \\ 0 & I_{3\times3} & 0 & 0 & 0 \\ 0 & 0 & I_{3\times3} & 0 & 0 \end{bmatrix}$$

Extended Kalman Filter Equations (36) and (37) should be sufficient to write the EKF. Note that (36) is nonlinear and the linearization of the equations might require the use of a symbolic manipulator like Mathematica or Matlab.

5.3 Extensions

There are two simplifying assumptions in this treatment. First, we have used the Euler angle parameterization in our state. It would have been better to use either quaternions or exponential coordinates in the derivation. Second, we have assumed that the camera is a sensor that yields the pose as a measurement. In reality, the camera is a noisy sensor and the tracking of the features is also a noisy process. Thus the estimation of the pose should include a model of this noise. A more rigorous treatment of the Vision-aided Inertial Navigation (VIN) problem that addresses both these limitations is available in [4].

References

[1] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Dover Publications, 2005.

[2] H. Khalil, *Nonlinear Systems*. Prentice Hall, 1996.

[3] D. E. Kirk, *Optimal Control Theory*. Dover Publications, 2004.

[4] D. G. Kottas and J. A. Hesch and S. L. Bowman and S. I. Roumeliotis, On the Consistency of Vision-aided Inertial Navigation, *International Symposium on Experimental Robotics*, Quebec, Canada, June 2012.

[5] M. Mintz, Notes on the Multivariate Normal Distribution and the Discrete-Time Kalman Filter, University of Pennsylvania, 2006.