# Extended Kalman Filter Derivations

Rik Bähnemann

January 22, 2016

## 1 Introduction

Out of our simulated experiment trajectories we want to pick the trajectory which is most informative with respect to MAV parameter identification. Thus we need a measure of information that each experiment gives. We pick the overall uncertainty in parameter estimates as our information measure of choice.

In order to make a prediction about the uncertainty in parameter estimation we will perform an extended Kalman filter (EKF) estimation along the simulated trajectories. Due to its recursive fashion the EKF is well suited for computationally demanding estimations. Further it has been proven to converge with a hexacopter's non-linear dynamics by Burri [1].

In the following we will first summarize the EKF routine. Then we will state how each single component is derived. This includes stating the MAV dynamics model and deriving the EKF propagation and update equations. We will close the section by showing how different trajectories influence the simulated parameter uncertainty.

## 2 Extended Kalman Filter

Propagating a belief from one node to another node in the tree is divided into three phases: initialize, propagate, update. The three steps are summarized in table 1.

At the starting node the Kalman filter is initialized with the MAV state (position, velocity, attitude, angular velocity, constant parameters) and its covariance matrix.

This initial state is then propagated using the discretized nominal non-linear MAV dynamics $f_k(x_{k-1}, u_k, w_k = 0)$. We use rotor speeds to drive the system. These rotor speeds are derived from the minimum-snap (jerk) trajectories which connect two nodes. Meanwhile the a priori estimate covariance is updated using the error dynamics $F_k$ which are linearized around the current maximum likelihood estimate $\hat{x}_{k|k-1}$. Propagation is run at $100\,\mathrm{Hz}$.

Measurement updates are performed at a rate of $20\,\mathrm{Hz}$. We simulate position and attitude measurements which coincide with the propagated maximum likelihood estimates. Consequently this leads to zero innovations $\tilde{y}_k$ and only the covariance, not the state is updated. We do this because we want the parameter estimates to stay constant. The innovation covariance $S_k$, Kalman gain $K_k$ and updated a posteriori estimate covariance $P_{k|k}$ calculates according to the current a priori estimate and linearized measurement model equations.

| Phase | Equations | |
|---|---|---|
| Initialization | $\hat{x}_{0\|0} = x_0$ | (1a) |
| | $P_{0\|0} = P_0$ | (1b) |
| Propagate | $\hat{x}_{k\|k-1} = f_k(\hat{x}_{k-1}, u_k, 0)$ | (2a) |
| | $P_{k\|k-1} = F_k P_{k-1} F_k^T + Q_k$ | (2b) |
| Update | $\tilde{y}_k = z_k - H_k \hat{x}_{k\|k-1} = 0$ | (3a) |
| | $S_k = H_k P_{k\|k-1} H_k^T + R_k$ | (3b) |
| | $K_k = P_{k\|k-1} H_k^T S_k^{-1}$ | (3c) |
| | $\hat{x}_{k\|k} = \hat{x}_{k\|k-1} + K_k \tilde{y}_k = \hat{x}_{k\|k-1}$ | (3d) |
| | $P_{k\|k} = (I - K_k H_k) P_{k\|k-1}$ | (3e) |

Table 1: Extended Kalman filter predict and update equations.

# 3 MAV Dynamics Model

The dominant forces and moments acting on a MAV origin from the aerodynamics forces and moments acting on each rotor and the gravitational force. Summing up these, we formulate the translational and rotational dynamics using Newton's respectively Euler's equations. We do not model motor dynamics, as they are much faster than the remaining and thus can be considered instantaneous.

## Assumptions

For simplification and because we consider their effects small in the experiment setup we neglect several effects. These are

- fuselage drag (small velocities),
- motor dynamics (very fast, considered instantaneous)

from the system perspective and on rotor dynamics

- rotor drag,
- blade flapping (stiff rotors),
- high order linear and angular velocity terms (small at hovering compared to blade tip speed),
- linear and angular acceleration of propellers (low mass),
- angular acceleration of motors (small at hovering),
- friction torque due to rotational motion.

Further we assume that the center of gravity (CoG) coincides with the geometric center in xy-direction with some offset in z-direction. And the directions of thrust at each rotor are perpendicular to the rotor planes and coincide.

For parameter estimation we assume to know the vector from the CoG to each rotor plane center $A_i$ in base coordinates $\mathbf{r}$ and the mass $m$.

## Model

Due to our assumptions we only consider thrust acting on each rotor blade.

$$\mathbf{T}_i = c_T n_i^2 \mathbf{e}_z + \mathbf{w}_{T,i} \tag{4}$$

The constant $c_T$ describes our thrust constant. $n_i$ is the angular velocity of the $i$-th rotor blade. $\mathbf{e}_z$ is a unit vector in z-direction in base coordinates. $\mathbf{w}_{T,i}$ is zero mean Gaussian noise with covariance $\boldsymbol{\sigma}_T^2 = \sigma_T^2 \mathbf{I}_{3\times3}$. It is accounting for the unstructured modeling errors in blade dynamics.

Our state vector consists of the vehicles position in inertial frame $\mathbf{p}$, its velocity in base coordinates $\mathbf{v}$, its rotation quaternion between base and inertial frame $\mathbf{q}$ and its angular velocity in base frame $\boldsymbol{\omega}$. Further the state is augmented by the the parameters for which we wish to estimate the estimation uncertainty. These are the thrust constant $c_T$, the moment constant $c_M$ and the three moments of inertia $\mathbf{j}$ which form the moments of inertia matrix $\mathbf{J} = diag(\mathbf{j})$. Thus the state vector contains 18 states.

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}^T & \mathbf{v}^T & \mathbf{q}^T & \boldsymbol{\omega}^T & c_T & c_M & \mathbf{j}^T \end{bmatrix}^T \tag{5}$$

Using Newton's and Euler's equations, kinematic relations and the rotation matrix $\mathbf{C}(\mathbf{q})$ from base to inertial frame, we derive the non-linear state differential equations.

$$\dot{\mathbf{p}} = \mathbf{C}(\mathbf{q}) \cdot \mathbf{v} \tag{6a}$$

$$\dot{\mathbf{v}} = \frac{1}{m} \sum_{i=1}^{k} \mathbf{T}_i - \mathbf{C}^T(\mathbf{q}) \cdot \mathbf{g} - \boldsymbol{\omega} \times \mathbf{v} + \mathbf{w}_A \tag{6b}$$

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \tag{6c}$$

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1} \left( \sum_{i=1}^{k} \left( -\epsilon_i c_m \mathbf{T}_i + \mathbf{T}_i \times \mathbf{r} \right) - \boldsymbol{\omega} \times \mathbf{J} \boldsymbol{\omega} \right) + \mathbf{w}_M \tag{6d}$$

$$\dot{c}_T = 0 \tag{6e}$$

$$\dot{c}_M = 0 \tag{6f}$$

$$\dot{\mathbf{j}} = \mathbf{0} \tag{6g}$$

$\otimes$ denotes quaternion multiplication. $\mathbf{w}_A$ and $\mathbf{w}_M$ are zero mean Gaussian process noise with variance $\boldsymbol{\sigma}_A^2 = \sigma_A^2 \mathbf{I}_{3\times3}$ and $\boldsymbol{\sigma}_M^2 = \sigma_M^2 \mathbf{I}_{3\times3}$ accounting for modelling error in linear acceleration or angular acceleration respectively. $\mathbf{g} = [0\ 0\ g]^T$ is the gravity vector where $g$ is the earth gravity constant. $\epsilon_i = \pm 1$ is the turning direction of each rotor. The process noise vector $\mathbf{w}$ contains $3 \cdot k + 6$ components.

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_{T,1}^T & \cdots & \mathbf{w}_{T,k}^T & \mathbf{w}_A^T & \mathbf{w}_M^T \end{bmatrix}^T \tag{7}$$

## Discretization

The EKF propagation equations 2a require the continuous time system equations 6 in discretized form. We attain the discretized non-linear nominal system's equations using zero order hold discretization and setting the process noise to zero.

$$f_k(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{0}) = \begin{bmatrix} \mathbf{p}_{k-1} + \mathbf{C}(\mathbf{q_{k-1}}) \cdot \mathbf{v}_{k-1} \cdot \Delta t + \frac{1}{2}\mathbf{C}(\mathbf{q_{k-1}}) \cdot \mathbf{a}_{k-1} \cdot \Delta t^2 \\ \mathbf{v}_{k-1} + \mathbf{a}_{k-1} \cdot \Delta t \\ \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\omega}_{k-1}\Delta t + \frac{1}{4}\dot{\boldsymbol{\omega}}_{k-1} \cdot \Delta t^2 \end{bmatrix} \otimes \mathbf{q}_{k-1} \\ \boldsymbol{\omega}_{k-1} + \dot{\boldsymbol{\omega}}_{k-1} \cdot \Delta t \\ c_{T,k-1} \\ c_{M,k-1} \\ \mathbf{j}_{k-1} \end{bmatrix}, \tag{8}$$

where

$$\mathbf{T}_{i,k-1} = c_T \left( \frac{n_{i,k-1} + n_{i,k}}{2} \right)^2 \mathbf{e}_z, \tag{9}$$

$$\mathbf{a}_{k-1} = \frac{1}{m} \sum_{i=1}^{k} \mathbf{T}_{i,k-1} - \mathbf{C}^T(\mathbf{q}_{k-1}) \cdot \mathbf{g} - \boldsymbol{\omega}_{k-1} \times \mathbf{v}_{k-1}, \tag{10}$$

$$\dot{\boldsymbol{\omega}}_{k-1} = \mathbf{J}^{-1} \left( \sum_{i=1}^{k} \left( -\epsilon_i c_m \mathbf{T}_{i,k-1} + \mathbf{T}_{i,k-1} \times \mathbf{r} \right) - \boldsymbol{\omega}_{k-1} \times \mathbf{J}\boldsymbol{\omega}_{k-1} \right). \tag{11}$$

## Error dynamics

Deriving the state and noise transition matrix $\mathbf{F}_k$ and $\mathbf{Q}_k$ for the a priori covariance propagation step 2b requires linearizing the system about the current maximum likelihood estimate. This will give us the linear continuous time system

$$\dot{\tilde{\mathbf{x}}} = \mathbf{F}_c \tilde{\mathbf{x}} + \mathbf{G}_c \mathbf{w}, \tag{12}$$

where $\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$ is the error state vector.

As we are using quaternions to describe the attitude, we can not just calculate the difference between the true attitude and estimated attitude. This would not conserve unit length and would lead to numerical issues due to its non-minimal representation. So instead it is common to use an error angle vector $\delta\boldsymbol{\Theta}$ to express changes in attitude. We follow the derivations by Weiss [2] and Achtelik [3] whos work is based on Trawny and Roumeliotis [4]. The error angle vector is defined as

$$\delta\boldsymbol{\Theta} = \hat{\mathbf{q}}^{-1} \otimes \mathbf{q} \qquad\qquad \delta\mathbf{q} \approx \begin{bmatrix} 1 & \frac{1}{2}\delta\boldsymbol{\Theta}^T \end{bmatrix}^T. \tag{13}$$

From this definition we can derive the error rotation matrix. $\lfloor \cdot_\times \rfloor$ defines the skew symmetric matrix.

$$\mathbf{C}(\mathbf{q}) = \mathbf{C}(\hat{\mathbf{q}}) \cdot \Delta\mathbf{C} \approx \mathbf{C}(\hat{\mathbf{q}}) \left( \mathbf{I}_{3\times3} + \lfloor \delta\boldsymbol{\Theta}_\times \rfloor \right) \tag{14}$$

The error state vector now describes as

$$\tilde{\mathbf{x}} = \begin{bmatrix} \Delta\mathbf{p}^T & \Delta\mathbf{v}^T & \delta\boldsymbol{\Theta}^T & \Delta\boldsymbol{\omega} & \Delta c_T & \Delta c_M & \Delta\mathbf{j}^T \end{bmatrix}^T \tag{15}$$

and we can calculate the error state differential equations as the difference between the real differential equations 6 and its nominal estimates. Using the small angle approximations from equation 14, neglecting higher order terms and linearizing around zero expected error and noise we attain the linearized dynamics.

$$\mathbf{F}_c = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{C}(\hat{\mathbf{q}}) & -\mathbf{C}(\hat{\mathbf{q}})\cdot\lfloor\hat{\mathbf{v}}_\times\rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\lfloor\hat{\boldsymbol{\omega}}_\times\rfloor & -\lfloor\mathbf{C}^T(\hat{\mathbf{q}})\cdot\mathbf{g}_\times\rfloor & \lfloor\hat{\mathbf{v}}_\times\rfloor & \mathbf{A} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\lfloor\hat{\boldsymbol{\omega}}_\times\rfloor & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{B} & \mathbf{C} & \mathbf{D} & \mathbf{E} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 0 & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 0 & \mathbf{0}_{1\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{16}$$

$$\mathbf{G}_c = \begin{bmatrix} \mathbf{0}_{3\times3} & \cdots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \frac{1}{m}\mathbf{I}_{3\times3} & \cdots & \frac{1}{m}\mathbf{I}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \cdots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{F}_1 & \cdots & \mathbf{F}_k & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \mathbf{0}_{1\times3} & \cdots & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & \cdots & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \\ \mathbf{0}_{3\times3} & \cdots & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{17}$$

with

$$\mathbf{A} = \frac{1}{m}\sum_{i=1}^{k} n_i^2\mathbf{e}_z, \tag{18}$$

$$\mathbf{B} = \hat{\mathbf{J}}^{-1}\left(\lfloor\hat{\mathbf{J}}\cdot\hat{\boldsymbol{\omega}}_\times\rfloor - \lfloor\boldsymbol{\omega}_\times\rfloor\cdot\hat{\mathbf{J}}\right), \tag{19}$$

$$\mathbf{C} = \hat{\mathbf{J}}^{-1}\left(\sum_{i=1}^{k} -\epsilon_i\hat{c}_M n_i^2\mathbf{e}_z + n_i^2\mathbf{e}_z\times\mathbf{r}_i\right), \tag{20}$$

$$\mathbf{D} = \hat{\mathbf{J}}^{-1}\left(\sum_{i=1}^{k} -\epsilon_i\hat{c}_T n_i^2\mathbf{e}_z\right), \tag{21}$$

$$\mathbf{E} = -\hat{\mathbf{J}}^{-1}\begin{bmatrix} 0 & -\hat{\omega}_2\hat{\omega}_3 & \hat{\omega}_2\hat{\omega}_3 \\ \hat{\omega}_1\hat{\omega}_3 & 0 & -\hat{\omega}_1\hat{\omega}_3 \\ -\hat{\omega}_1\hat{\omega}_2 & \hat{\omega}_1\hat{\omega}_2 & 0 \end{bmatrix} - \dots \tag{22}$$

$$\operatorname{diag}\left(\hat{\mathbf{J}}^{-2}\left(\sum_{i=1}^{k}\left(-\epsilon_i\hat{c}_M\hat{c}_T n_i^2\mathbf{e}_z + \hat{c}_T n_i^2\mathbf{e}_z\times\mathbf{r}_i\right) - \lfloor\hat{\boldsymbol{\omega}}_\times\rfloor\hat{\mathbf{J}}\hat{\boldsymbol{\omega}}\right)\right), \tag{23}$$

$$= -\hat{\mathbf{J}}^{-1}\left(\begin{bmatrix} 0 & -\hat{\omega}_2\hat{\omega}_3 & \hat{\omega}_2\hat{\omega}_3 \\ \hat{\omega}_1\hat{\omega}_3 & 0 & -\hat{\omega}_1\hat{\omega}_3 \\ -\hat{\omega}_1\hat{\omega}_2 & \hat{\omega}_1\hat{\omega}_2 & 0 \end{bmatrix} + \operatorname{diag}\left(\hat{c}_T\mathbf{C} - \hat{\mathbf{J}}^{-1}\lfloor\hat{\boldsymbol{\omega}}_\times\rfloor\hat{\mathbf{J}}\hat{\boldsymbol{\omega}}\right)\right) \tag{24}$$

$$\mathbf{F}_i = \hat{\mathbf{J}}^{-1}\left(-\epsilon_i\hat{c}_M\mathbf{I}_{3\times3} - \lfloor\mathbf{r}_{i\times}\rfloor\right) \tag{25}$$

These matrices can be generated quickly as they only consist of matrix multiplications.

Discretization is done via zero order hold. Note that following Weiss' work one may also find an exact solution [2].

We define $\boldsymbol{\varepsilon}(k)$ as a white noise vector with unit normal distribution of appropriate size. Accordingly, $\mathbf{Q}_c$ is the noise covariance matrix and defined as $\mathbf{Q}_c = \operatorname{diag}({\boldsymbol{\sigma}_T^2}^T, \dots, {\boldsymbol{\sigma}_T^2}^T, {\boldsymbol{\sigma}_A^2}^T, {\boldsymbol{\sigma}_M^2}^T)$.

$$\tilde{\mathbf{x}}(k+1) \approx (\mathbf{I} + \mathbf{F}_c \Delta t)\tilde{\mathbf{x}}(k) + \underbrace{\Delta t \mathbf{G}_c \sqrt{\mathbf{Q}_c}}_{\sqrt{\mathbf{Q}_k}} \boldsymbol{\varepsilon}(k) \tag{26}$$

We get

$$\mathbf{F}_k = \mathbf{I} + \mathbf{F}_c \Delta t, \tag{27}$$

$$\mathbf{Q}_k = \mathbf{G}_c \mathbf{Q}_c \mathbf{G}_c^T \Delta t^2. \tag{28}$$

The noise transition matrix explicitly expands to

$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \frac{k}{m^2}\boldsymbol{\sigma}_T^2 + \boldsymbol{\sigma}_A^2 & \mathbf{0}_{3\times3} & \frac{1}{m}\boldsymbol{\sigma}_T^2 \sum_{i=1}^k \mathbf{F}_i^T & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \frac{1}{m}\boldsymbol{\sigma}_T^2 \sum_{i=1}^k \mathbf{F}_i & \mathbf{0}_{3\times3} & \boldsymbol{\sigma}_M^2 + \sum_{i=1}^k \mathbf{F}_i \boldsymbol{\sigma}_T^2 \mathbf{F}_i^T & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 0 & \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 0 & 0 & \mathbf{0}_{1\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \end{bmatrix} \Delta t^2, \tag{29}$$

where

$$\mathbf{F}_i \boldsymbol{\sigma}_T^2 \mathbf{F}_i^T = \hat{\mathbf{J}}^{-2} \boldsymbol{\sigma}_T^2 \left( \hat{c}_M^2 \mathbf{I} - \lfloor \mathbf{r}_{i\times} \rfloor^2 \right) \tag{30}$$

The derived transition matrix are sparse and the non-constant elements can be generated quickly at every propagation step as they only consist of matrix multiplication and addition. This allows the propagation step to run fast. We now need to derive the measurement update equations.

## Measurements

We assume to receive position and attitude measurements at 20 Hz from a Vicon motion capture system.

$$\mathbf{z}_p = \mathbf{p} + \mathbf{v}_p, \qquad \boldsymbol{v}_p \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_p), \tag{31}$$

$$\mathbf{z}_q = \mathbf{q} \otimes \mathbf{v}_q, \tag{32}$$

$$\mathbf{v}_q = \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\Phi} \end{bmatrix}, \qquad \boldsymbol{\Phi} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_q), \tag{33}$$

$$\mathbf{R}_p = \sigma_p^2 \mathbf{I}_{3\times3}, \qquad \mathbf{R}_q = \sigma_q^2 \mathbf{I}_{3\times3}. \tag{34}$$

Instead of measuring the position and attitude directly, we measure the error states instead. Thus, the discrete time output equations evolve as

$$\mathbf{z}(k) = \begin{bmatrix} \mathbf{z}_p(k) \\ \mathbf{z}_q(k) \end{bmatrix} = \mathbf{H}_k \tilde{\mathbf{x}}(k) + \mathbf{v}(k), \qquad \mathbf{v}(k) \sim \mathbf{N}(\mathbf{0}, \mathbf{R}_k), \tag{35}$$

where

$$\mathbf{R}_k = \frac{1}{\Delta t}\mathbf{R}_c, \tag{36}$$

$$\mathbf{R}_c = \operatorname{diag}\left(\mathbf{R}_p, \mathbf{R}_q\right), \tag{37}$$

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3} \end{bmatrix}. \tag{38}$$

# 4 Linearization Points from Flat States

In the previous section we derived the system dynamics and measurement equations. The system matrix $\mathbf{F}_l$ is linearized about the state and rotor speeds at every time step. Approximately the rotor speeds, attitude and angular velocity can be derived from the flat states and its derivatives following Mellinger [5]. From the trajectories we have the state in world coordinates.

$$\mathbf{x}_f = \begin{bmatrix} \mathbf{p}^T & \mathbf{v}^T & \mathbf{a}^T & \mathbf{j}^T & \mathbf{s}^T & \psi & \dot{\psi} & \ddot{\psi} \end{bmatrix}^T \tag{39}$$

The full attitude $\mathbf{q}$ is described as

$$\mathbf{z}_B = \frac{\mathbf{t}}{|\mathbf{t}|}, \qquad \mathbf{t} = \mathbf{a} + \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T, \tag{40}$$

$$\mathbf{y}_B = \frac{\mathbf{z}_B \times \mathbf{x}_C}{||\mathbf{z}_B \times \mathbf{x}_C||}, \qquad \mathbf{x}_C = \begin{bmatrix} \cos\psi & \sin\psi & 0 \end{bmatrix}^T, \tag{41}$$

$$\mathbf{x}_B = \mathbf{y}_B \times \mathbf{z}_B, \tag{42}$$

$$\mathbf{C}(\mathbf{q}) = \begin{bmatrix} \mathbf{x}_B & \mathbf{y}_B & \mathbf{z}_B \end{bmatrix}, \qquad ||\mathbf{z}_B \times \mathbf{x}_C|| \neq 0. \tag{43}$$

The angular velocity $\boldsymbol{\omega}$ is

$$\mathbf{h}_\omega = \frac{1}{|\mathbf{t}|}\left(\mathbf{j} - \left(\mathbf{z}_B^T \cdot \mathbf{j}\right)\mathbf{z}_B\right), \tag{44}$$

$$\omega_1 = -\mathbf{h}_\omega^T \cdot \mathbf{y}_B, \tag{45}$$

$$\omega_2 = \mathbf{h}_\omega^T \cdot \mathbf{x}_B, \tag{46}$$

$$\omega_3 = \dot{\psi} \cdot \mathbf{z}_W^T \cdot \mathbf{z}_B. \tag{47}$$

The angular acceleration $\boldsymbol{\alpha}$ is

$$\mathbf{h}_\alpha = \left(\frac{1}{|\mathbf{t}|}\left(\mathbf{s} - \left(\mathbf{z}_B^T \cdot \mathbf{s}\right)\mathbf{z}_B\right) - 2\left(\boldsymbol{\omega} \times \mathbf{z}_B\right) - \boldsymbol{\omega} \times \left(\boldsymbol{\omega} \times \mathbf{z}_B\right) + \mathbf{z}_B^T\left(\boldsymbol{\omega} \times \left(\boldsymbol{\omega} \times \mathbf{z}_B\right)\right)\mathbf{z}_B\right), \tag{48}$$

$$\alpha_1 = -\mathbf{h}_\alpha^T \cdot \mathbf{y}_B, \tag{49}$$

$$\alpha_2 = \mathbf{h}_\alpha \cdot \mathbf{x}_B, \tag{50}$$

$$\alpha_3 = \ddot{\psi}\mathbf{z}_W^T \cdot \mathbf{z}_B. \tag{51}$$

Finally, a linear mapping $\mathbf{A}$ between thrust $T$ and torques $\boldsymbol{\tau}$ and rotor speeds $\mathbf{n}$ exists, which allows us calculating the rotor speeds.

$$\begin{bmatrix} \boldsymbol{\tau} \\ T \end{bmatrix} = \mathbf{K} \cdot \mathbf{A} \cdot \mathbf{n}^2, \tag{52}$$

$$\boldsymbol{\tau} = \mathbf{J}\boldsymbol{\alpha} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}, \tag{53}$$

$$T = m|\mathbf{t}|. \tag{54}$$

For a hexacopter we get

$$\mathbf{A} = \begin{bmatrix} s & 1 & s & -s & -1 & -s \\ -c & 0 & c & c & 0 & -c \\ -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \tag{55}$$

$$\mathbf{K} = \text{diag}\left(\begin{bmatrix} lc_T & lc_T & c_T c_M & c_T \end{bmatrix}\right), \tag{56}$$

where $s = \sin(30°)$, $c = \cos(30°)$ and $l$ is the rotor arm length. The squared rotor speeds, which are required for linearization, can now be obtained from known quantities.

$$\mathbf{n}^2 = A^{\dagger} \cdot K^{-1} \cdot \begin{bmatrix} \boldsymbol{\tau} \\ T \end{bmatrix} \tag{57}$$

# 5 Discussion

## Covariance

We derived all components necessary to drive the Kalman filter equations 1 – 3. We are particularly interested in how the covariance evolves, as this gives us a measure of how the parameter estimate uncertainty changes over the course of the trajectory.

During the propagation step 2b the covariance grows through the system matrix $\mathbf{F}_k$ and noise input matrix $\mathbf{Q}_k$ from equations 27 and 28. As one can see from equation 29, the process noise matrix only grows the uncertainty in velocity and angular velocity. The other states and parameters are not affected by process noise. However, the system matrix 27 leads to some coupling which also affect covariance growth in position and attitude as well as parameter uncertainty. The parameter uncertainty grows in the correlated terms due to the coupling terms **A**, **C**, **D**, **E** from equations 18, 20, 21, 23. The magnitude which the covariance grows with is depending on the magnitude of $\mathbf{F}_k$ which is depending on the current linearization point which itself is depending on the trajectory. This correlation finally leads to the parameter estimate covariance shrinking during measurement updates.

## Measurement generation

As we only care about the propagation of the covariance we may consider leaving out propagating the state and updating it. We could assume that the MAV stays close to the given trajectory and generate the linearization point from the polynomial only. Then we would only update the a priori and a posteriori covariance. We would neglect the discrete state propagation and state estimate update.

# References


[1] M. Burri, M. Datwiler, M. W. Achtelik, and R. Siegwart, “Robust state estimation for Micro Aerial Vehicles based on system dynamics,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, IEEE, 2015, pp. 5278–5283.

[2] S. M. Weiss, “Vision based navigation for micro helicopters,” PhD thesis, Citeseer, 2012.

[3] M. W. Achtelik, “Advanced closed loop visual navigation for micro aerial vehicles,” PhD thesis, 2014. DOI: 10.3929/ethz-a-010147198. [Online]. Available: http://e-collection.library.ethz.ch/view/eth:8596.

[4] N. Trawny and S. I. Roumeliotis, “Indirect Kalman filter for 3D Attitude Estimation,” *… , Dept. of Comp. Sci. & Eng., …*, no. 2005-002, pp. 1–25, 2005. [Online]. Available: http://www-users.cs.umn.edu/%7B~%7Dtrawny/Publications/Quaternions%7B%5C_%7D3D.pdf.

[5] D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 2520–2525.