State and parameter estimation in aerial load manipulation

Vicko Prkačin

University of Dubrovnik

vicko.prkacin@unidu.hr

February 18, 2019

Overview

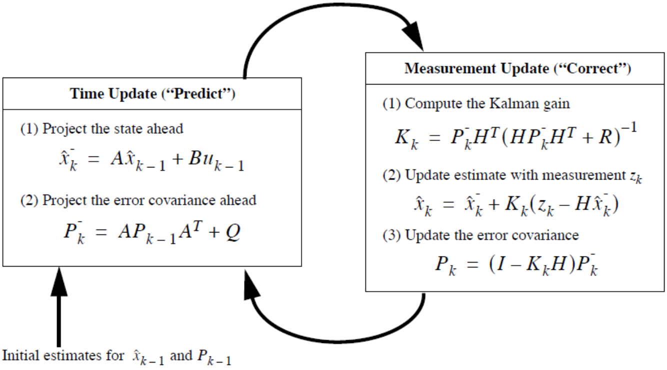
1. Estimation
   - Kalman filter
   - Extended Kalman filter
   - Unscented Kalman filter
   - Least squares
2. Aerial load manipulation
   - Dynamics
   - Kinematics
   - Slung load model
3. Current work
   - Continuous wavelet transform
   - Experimental platform
   - Results

Kalman filter

Addresses general problem of estimating the state $x \in \mathbb{R}^n$ of a discrete time controlled process governed by a linear stochastic differential equation:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$
$$z_k = Hx_k + v_k$$

- $z \in \mathbb{R}^m$ is a vector of measurements
- $u \in \mathbb{R}^l$ is control input vector
- A is $(n \times n)$ system matrix
- B is $(n \times l)$ input matrix
- H is $(m \times n)$ measurement matrix
- w_k, $p(w) \sim N(0, Q)$ is process noise
- v_k, $p(v) \sim N(0, R)$ is measurement noise.



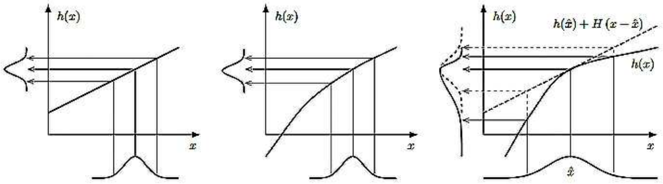
Under the assumption of precisely known linear system (matrices A, H, Q and R are exactly known) and pure white, zero mean, uncorrelated noise sequences w and v, Kalman filter will provide optimal state estimation.

Extended Kalman filter

The process is now governed by the nonlinear stochastic difference equation

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}),$$
$$z_k = h(x_k, v_k)$$

Idea: nonlinear functions can be linearized around the mean and covariance of the current system state estimate.



Linearization:

$$x_k \approx \tilde{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + W w_{k-1},$$
$$z_k \approx \tilde{z}_k + H(x_{k-1} - \hat{x}_{k-1}) + V v_k,$$

where:

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0), \qquad H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\hat{x}_k, 0),$$
$$W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0), \qquad V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}(\hat{x}_k, 0).$$

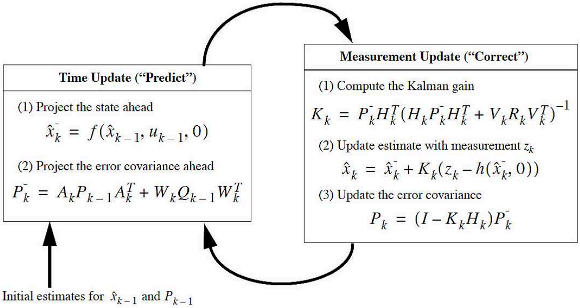
Prediction error and the measurement residual can be defined as:

$$\tilde{e}_{x_k} \equiv x_k - \tilde{x}_k \approx A(x_{k-1} - \hat{x}_{k-1}) + \varepsilon_k,$$
$$\tilde{e}_{z_k} \equiv z_k - \tilde{z}_k \approx H\tilde{e}_{x_k} + \eta_k,$$

where:

$$p(\varepsilon) \sim N(0, WQW^T), \qquad p(\eta) \sim N(0, VRV^T).$$

Operation of the Extended Kalman filter can be presented as:



Optimality for the EKF is not guaranteed. First order linearization of the nonlinear system can introduce large error in the true posterior mean and covariance of the transformed GRV.

Unscented Kalman filter

Addresses the approximation issues of the EKF and in general yields better results

- The state distribution is again represented by a GRV specified using a minimal set of carefully chosen sample points. These sample points completely capture the true mean and covariance of the GRV.
- When propagated through the true nonlinear system, the posterior mean and covariance is captured accurately.
- This process is defined by **unscented transformation**.

Unscented transformation

Suppose a $(nx1)$ vector x has the known mean $\bar{x}$ and covariance P.
$2n$ sigma point vectors $x^{(i)}$ are formed as:

$$
\begin{aligned}
x^{(i)} &= \bar{x} + \tilde{x}^{(i)}, & i = 1, ..., 2n \\
\tilde{x}^{(i)} &= \left(\sqrt{nP}\right)_i^T, & i = 1, ..., n \\
\tilde{x}^{(n+i)} &= -\left(\sqrt{nP}\right)_i^T, & i = 1, ..., n
\end{aligned}
$$

where $\sqrt{nP}$ is the matrix square root of nP such that $\left(\sqrt{nP}\right)^T \sqrt{nP} = nP$ and $\left(\sqrt{nP}\right)_i$ is the i-th row of $\sqrt{nP}$. Ensemble mean and covariance of the set of sigma point vectors are equal to $\hat{x}$ and P.

To approximate the mean and covariance of a nonlinear function $y = h(x)$:

- individual sigma points are transformed:

$$y^{(i)} = h(x^{(i)}), i = 1, ..., 2n$$

- weighted sum of the transformed sigma points is calculated:

$$\bar{y}_u = \frac{1}{2n}\sum_{i=1}^{2n}(y^{(i)}), i = 1, ..., 2n$$

$$P_u = \frac{1}{2n}\sum_{i=1}^{2n}(y^{(i)} - y_u)(y^{(i)} - y_u)^T$$

UKF algorithm

Time update:

The UKF is initialized as:

$$\hat{x}^{+} = E(x_0)$$
$$P_0^{+} = E\left[(x_0 - \hat{x}_0^{+})(x_0 - \hat{x}_0^{+})^T\right].$$

Time update equations are:

- Sigma points $x_{k-1}^{(i)}$ are chosen as:

$$\hat{x}_{k-1}^{(i)} = \hat{x}_{k-1}^{+} + \tilde{x}^{(i)}, \qquad i = 1, ..., 2n$$
$$\tilde{x}^{(i)} = \left(\sqrt{nP_{k-1}^{+}}\right)_i^T, \qquad i = 1, ..., n$$
$$\tilde{x}^{(n+i)} = -\left(\sqrt{nP_{k-1}^{+}}\right)_i^T, \qquad i = 1, ..., n$$

- Sigma points are transformed into $\hat{x}_k^{(i)}$ vectors using:

$$\hat{x}_k^{(i)} = f(\hat{x}_{k-1}^{(i)}, u_k, t_k)$$

- Vectors $\hat{x}_k^{(i)}$ are combined to obtain the *a priori* state estimate at time k:

$$\hat{x}_k^{-} = \frac{1}{2n}\sum_{i=1}^{2n}(\hat{x}_k^{(i)})$$

- *A priori* covariance is estimated as:

$$P_k^{-} = \frac{1}{2n}\sum_{i=1}^{2n}(\hat{x}_k^{(i)} - \hat{x}_k^{-})(\hat{x}_k^{(i)} - \hat{x}_k^{-})^T + Q_{k-1}$$

UKF algorithm

Measurement update:

Measurement update equations are:

- Sigma points $x_k^{(i)}$ are chosen as:

$$\hat{x}_k^{(i)} = \hat{x}_k^- + \tilde{x}^{(i)}, \qquad i = 1, \dots, 2n$$

$$\tilde{x}^{(i)} = \left(\sqrt{nP_k^-}\right)_i^T, \qquad i = 1, \dots, n$$

$$\tilde{x}^{(n+i)} = -\left(\sqrt{nP_k^-}\right)_i^T, \qquad i = 1, \dots, n$$

- Transform the sigma points into vectors of predicted measurements $\hat{y}_k^{(i)}$:

$$\hat{y}_k^{(i)} = h(\hat{x}_k^{(i)}, t_k)$$

These vectors are combined to obtain the predicted measurement at time k:

$$\hat{y}_k = \frac{1}{2n} \sum_{i=1}^{2n} (\hat{y}_k^{(i)})$$

- Covariance of the predicted measurement is estimated:

$$P_y = \frac{1}{2n} \sum_{i=1}^{2n} (\hat{y}_k^{(i)} - \hat{y}_k)(\hat{y}_k^{(i)} - \hat{y}_k)^T + R_k$$

- Cross covariance between the *a priori* state estimate $\hat{x}_k^-$ and the predicted measurement $\hat{y}_k$ is:

$$P_{xy} = \frac{1}{2n} \sum_{i=1}^{2n} (\hat{x}_k^{(i)} - \hat{x}_k^-)(\hat{y}_k^{(i)} - \hat{y}_k)^T$$

Least squares

Unknown parameter of a mathematical model should be chosen such that the sum of the squares of the differences between the actually observed and the computed values, multiplied by numbers that measure the degree of precision, is a minimum.

$$y = Hx + v$$

- $y \in \mathbb{R}^k$ is a vector of measurements corrupted by noise v
- $x \in \mathbb{R}^n$ is a unknown vector to be estimated
- H is $(k \times n)$ measurement matrix.

Measurement residual and cost function for estimated value of x denoted $\hat{x}$ are:

$$\epsilon_y = y - H\hat{x}$$
$$J = \epsilon_{y1}^2 + ... + \epsilon_{yk}^2 = \epsilon_y^T \epsilon_y = (y - H\hat{x})^T (y - H\hat{x}) \,.$$

To minimize J with respect to $\hat{x}$, $\frac{\partial J}{\partial \hat{x}}$ is calculated and set equal to zero, which yields:

$$\hat{x} = (H^T H)^{-1} H^T y$$
$$\hat{x} = H^L y.$$

H^L is the left pseudo inverse of H, which exists if $k \geqslant n$ and H is full rank.

Weighted least squares

Incorporates the variance of the measurement noise as the measure of uncertainty:

$$E(v_i^2) = \sigma_i^2 (i = 1, ..., k)$$

$$R = E(vv^T) = \begin{bmatrix} \sigma_1^2 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & \sigma_k^2 \end{bmatrix}.$$

New cost function and estimated value of x are:

$$\begin{aligned} J &= \epsilon_{y1}^2/\sigma_1^2 + ... + \epsilon_{yk}^2/\sigma_k^2 \\ &= \epsilon_y^T R^{-1} \epsilon_y \\ &= (y - H\hat{x})^T R^{-1} (y - H\hat{x}) \end{aligned}$$

$$\hat{x} = \left(H^T R^{-1} H\right)^{-1} H^T R^{-1} y$$

Recursive least squares

In most of the applications we obtain measurements sequentially and want to update estimate of x after every measurement:

$$y_k = H_k x + v_k$$
$$\hat{x}_k = \hat{x}_{k-1} + K_k \left(y_k - H_k \hat{x}_{k-1} \right),$$

where K_k is the estimator gain matrix to be determined. The estimation error mean and covariance can be computed as:

$$E(\epsilon_{x,k}) = E(x - \hat{x}_k) = (I - K_k H_k) E(\epsilon_{x,k-1}) - K_k E(v_k)$$

$$P_k = (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T$$

Optimality criterion for the calculation of K_k is to minimize the sum of variances of the estimation error at time k, defined by the cost function J_k:

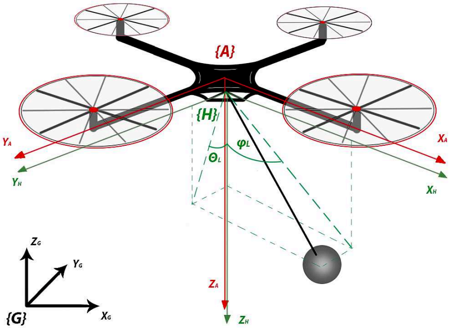
$$\begin{aligned} J_k &= E[(x_1 - \hat{x}_1)^2] + ... + E[(x_n - \hat{x}_n)^2] \\ &= E(\epsilon_{x1,k}^2 + ... + \epsilon_{xn,k}^2). \end{aligned}$$

When partial derivative with respect K_k is calculated and set equal to zero, expression for K_k is obtained:

$$K_k = P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1}.$$

Aerial load manipulation

To be able to understand and implement described estimation techniques in the context of aerial load manipulation, one must be familiar with the system kinematics and dynamics. Quadcopters are one of the most common used UAVs for this application.



System description

Two coordinate frames are defined:

- $\{A\}$ aircraft-fixed reference frame
- $\{G\}$ ground-fixed frame, considered to be the inertial frame.

The variables are:

- $\eta_1 = \begin{bmatrix} x & y & z \end{bmatrix}^T$ - position of the origin of $\{A\}$ measured in $\{G\}$,
- $\eta_2 = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ - roll, pitch and yaw angles that describe the orientation of $\{A\}$ with respect to $\{G\}$,
- $\nu_1 = \begin{bmatrix} u & v & w \end{bmatrix}^T$ - linear velocity of $\{A\}$ relative to $\{G\}$, expressed in $\{A\}$,
- $\nu_2 = \begin{bmatrix} p & q & r \end{bmatrix}^T$ - angular velocity of $\{A\}$ relative to $\{G\}$, expressed in $\{A\}$.

Dynamics

UAV's 6 degree of freedom nonlinear dynamics equation is expressed as:

$$M\dot{\nu} + C(\nu)\nu + D\nu + G(\eta) = \tau + \tau_L$$

- $\eta = \begin{bmatrix} \eta_1 & \eta_2 \end{bmatrix}^T$ is the vector of position and orientation
- $\nu = \begin{bmatrix} \nu_1 & \nu_2 \end{bmatrix}^T$ is the vector of linear and angular velocities
- M is the mass and inertia matrix of the UAV
- $C(\nu)$ is the matrix of Coriolis and centripetal terms
- $D\nu$ represents dissipative force and torque vector, where D is the damping matrix.
- $G(\nu)$ is the vector of gravitational forces and moments

- Control inputs are given as:

$$f_\tau = {}^G_A R^{-1}(\eta_2) \begin{bmatrix} 0 \\ 0 \\ U_1 \end{bmatrix}, \tau(\eta_2, U) = \begin{bmatrix} f_\tau(\eta_2) \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}$$

where U_1, U_2, U_3, U_4 are control forces generated by rotors.

- Forces and torques that the load exerts on the UAV are:

$$\tau_L = \begin{bmatrix} F_H & T_H \end{bmatrix}.$$

Kinematics

System kinematics is given as:

$$\begin{bmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \end{bmatrix} = \begin{bmatrix} {}^G_A R(\nu_2) & 0 \\ 0 & Q(\nu_2) \end{bmatrix} \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix},$$
$$\dot{\eta} = J_R(\eta)\nu$$

where:

- ${}^G_A R(\eta_2)$ is the transformation matrix between the two reference frames
- $Q(\eta_2) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix}$.

Slung load model

The external slung load is modelled as a point mass pendulum suspended from a single point.

- $\{H\}$ is defined at the suspension point with unit vectors parallel to $\{A\}$ and position with respect to UAV CoG given by $\rho_H = \begin{bmatrix} x_H & y_H & z_H \end{bmatrix}^T$
- Position vector ρ_L of the load with respect to the suspension point is:

$$\rho_L = R_{Y_H}(\theta_L) R_{X_H}(\phi_L) \begin{bmatrix} 0 \\ 0 \\ l_L \end{bmatrix}$$

Position of the load, as described here represents the state, and the length of the cable (l_L) is parameter for estimation in the focus of our research.

The force F_H that the load exerts on the vehicle and the torque T_H are respectively given by:

$$F_H = -m_L G_L,$$
$$T_H = \rho_H \times F_H$$

where:

$$G_L = R_{X_A}(\phi)^{-1} R_{Y_A}(\theta)^{-1} \begin{bmatrix} 0 \\ 0 \\ -m_L g \end{bmatrix}$$

represents the vector of gravitational forces and moments, ϕ and θ are the roll and pitch angles of the UAV respectively, and m_L is the mass of the load.

Both F_H and T_H are functions of ϕ_L and θ_L, as well as of UAV states.

Current objective:

- Online state estimation of the slung load without introduction of the additional sensors.
- Model-free on-line state estimation of the slung load utilizing only standard sensors available on the UAV using the signal processing of the sensor data.

Continuous wavelet transform

Wavelet is a function with a zero average, normalized $||\psi|| = 1$, and centred in the neighbourhood of $t = 0$:

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0$$

The continuous wavelet transform (CWT) of a signal $x(t) \in L^2(\mathbb{R})$ is a sequence of projections onto rescaled and translated versions of the wavelet $\psi(t)$:

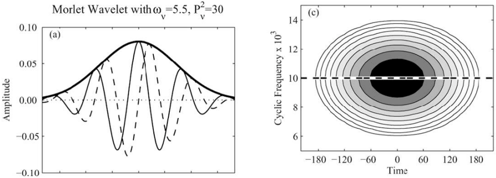
$$W(t,s) = \int_{-\infty}^{+\infty} \frac{1}{s}\psi^*\left(\frac{u-t}{s}\right)x(u)du$$

where (*) symbol denotes the complex conjugate, scaling factor is denoted by s, and translating factor is denoted by u.

CWT is a 2D representation of a 1D signal. For a real valued function the result of a transform is an $(N_a x N)$ matrix, where N_a is the number of scales and N is the number of samples. If the wavelet is complex-valued, the coefficients are complex-valued.

Morlet Wavelet - complex-valued wavelet, esssentially a Gaussian envelope modulated by a complex-valued carrier wave at radian frequency ν:

$$\psi_\nu(t) = a_\nu e^{-(1/2)t^2} \left[e^{i\nu t} - e^{-(1/2)\nu^2} \right]$$

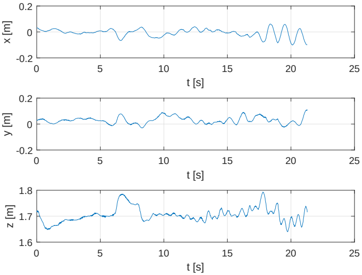


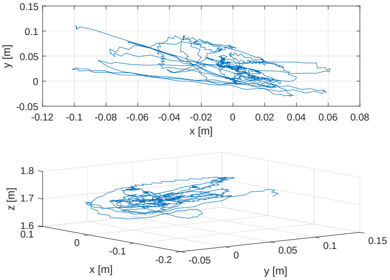
Experimental platform

- Crazyflie 2.0 quadcopter
- Loco UWB positioning system
- Crazyradio radio link

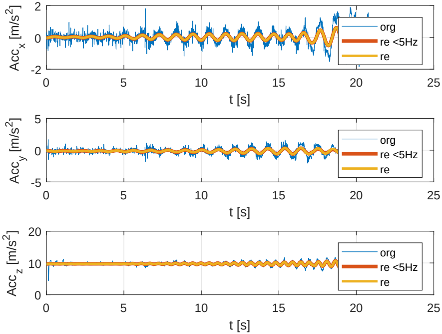
Results

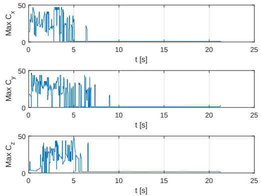
Position:

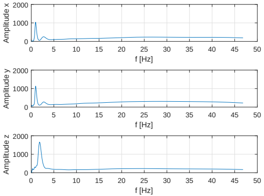




Disturbance reconstruction:







The End