Continuous Discrete Extended Kalman Filter Documentation

Zhengjie Cui, Greg Miller, Clark Zhang
Michigan Autonomous Aerial Vehicles
University Of Michigan, Ann Arbor

July 2, 2015

Abstract

This contains documentation about research and implementation of our state estimation filter used for the MAAV vehicle 2015-2016.

1 Basic Sensors and definitions overview

For sensors, we have

1. Px4Flow Camera that gives us (the following are in the frame of reference of the camera and not global frame)
   (a) $\frac{dx}{dt}$
   (b) $\frac{dy}{dt}$
   (c) z
2. Microstrain IMU
   (a) roll
   (b) pitch
   (c) yaw

We will use positive Z as upwards direction Looking at a bird's eye view of the field with our line on the bottom. Positive X is towards right, and positive Y is going up. Our Vehicle will have a mass, m.

We are using a DJI Naza Lite that takes as it's input (F_z is also in the frame of the vehicle)

1. F_z
2. ϕ

3. θ

4. $\dot{\psi}$

Our state that we will be estimating is (all states are in global frame)

$$\vec{x} = \begin{Bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \phi \\ \theta \\ \psi \end{Bmatrix} = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{Bmatrix}$$

Our sensor measurement will be in the form (these are in vehicle frame)

$$\vec{y} = \begin{Bmatrix} \phi \\ \theta \\ \psi \\ \dot{x} \\ \dot{y} \\ z \end{Bmatrix} = \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{Bmatrix}$$

our control input will be in the form (vehicle frame)

$$\vec{u} = \begin{Bmatrix} F_z \\ \phi \\ \theta \\ \psi \end{Bmatrix} = \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix}$$

2 Continuous Discrete Extended Kalman Filter

We used the filter as described by Beard in his paper. Here are the predict and update equations. Note: Δt is time since last predict
Prediction:

1. $\hat{x} = \hat{x} + \Delta t f(\vec{x}, \vec{u})$

2. $A = \frac{\Delta f}{\Delta x}$

3. $P = P + \Delta t(AP + PA^T + GQG^T)$

Update

1. $C = \frac{\Delta c}{\Delta x}$

2. $L = PC^T(R + CPC^T)^{-1}$

3. $P = (I - LC)P$

4. $\hat{x} = \hat{x} + L(y - c(\vec{x}))$

Since we don't have direct access to output, we are unable to change the roll, pitch, yaw in the prediction step So we will being doing something unorthodox - just replacing roll pitch yaw with the MicroStrain roll pitch yaw every step.

3 Prediction Model

Our prediction model will continuous

$$\frac{d\vec{x}}{dt} = f(\vec{x}, \vec{u}) = \begin{Bmatrix} x_4 \\ x_5 \\ x_6 \\ -\frac{u_1}{m}\sin(x_8) \\ \frac{u_1}{m}\sin(x_7)\cos(x_8) \\ \frac{u_1}{m}\cos(x_7)\cos(x_8) \\ 0 \\ 0 \\ u_4 \end{Bmatrix}$$

The linearized model of this around a state is (the 'A' matrix)

$$A = \begin{Bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{u_1}{m}\cos(x_8) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{u_1}{m}\cos(x_7)\cos(x_8) & -\frac{u_1}{m}\sin(x_7)\sin(x_8) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{u_1}{m}\sin(x_7)\cos(x_8) & -\frac{u_1}{m}\cos(x_7)\sin(x_8) & 0 \end{Bmatrix}$$

4 Sensor Model

Our sensor model is

$$\hat{y} = c(\vec{x}) = \begin{Bmatrix} x_7 \\ x_8 \\ x_9 \\ x_4\cos(x_9) \\ x_4\sin(x_9) \\ \frac{x_3}{\cos(x_7)\cos(x_8)} \end{Bmatrix}$$

The linearized model of this around a state (the C matrix)

$$
\mathrm{C}= \left\{\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \cos(x_9) & 0 & 0 & 0 & 0 & -x_4\sin(x_9) \\ 0 & 0 & 0 & \sin(x_9) & 0 & 0 & 0 & 0 & x_4\cos(x_9) \\ 0 & 0 & \frac{1}{\cos(x_7)\cos(x_8)} & 0 & 0 & 0 & \frac{x_3\sin(x_7)}{\cos^2(x_7)\cos(x_8)} & \frac{x_3\sin(x_8)}{\cos(x_7)\cos^2(x_8)} & 0 \end{matrix}\right\}
$$