Combined Complementary Filter For Inertial Navigation System

M.K. Filiashkin
M.V.Novik

October 9, 2012

Introduction

Unmanned Aerial Vehicles (UAVs) have been under development since the beginning of flight because they can eliminate the risk to a pilot's life and they are generally inexpensive to procure. Most UAVs are controlled remotely from a ground control station but automation systems are being progressively integrated for autonomous operation. GPS (Global Positioning System) and other sophisticated navigation systems make it possible for UAVs to take over high-risk aerial missions that required manned aircraft.

Some UAV Applications

1. Border interdiction. Patrol of the borders by aerial platforms.
2. Search and rescue. Looking for survivors from shipwrecks, aircraft accidents etc.
3. Wild fire suppression. UAVs equipped with infrared sensors can detect fire in forests and notify the fire brigade on time.
4. Law enforcement. VTOL UAVs can take the role of police helicopters in a more cost effective way.
5. Industrial applications. Such applications can be crops spraying, nuclear factory surveillance, surveillance of pipelines etc.

Kalman Filter Disadvantages

Noise Whiteness

It is known from the theory, that the Kalman filter is optimal in case that a) the model perfectly matches the real system, b) the entering noise is white and c) the covariances of the noise are exactly known. In practice it is difficult to estimate process covariance noise and whiteness of noise should be investigated.

Linearization

In case of nonlinear systems Extended Kalman Filter is adopted where by means of Taylor series expansion, nonlinear systems is linearized and approximated around each current state estimate. When large deviations between the estimated state trajectory and the nominal trajectory exist, the nonlinear model is weakly approximated bye Taylor series expansion around the conditional mean.

Covariance Calculations

The problem of Kalman filter divergence caused by covariance calculations, which has been with system developers since the 1960s, is still with the aerospace community. Basically, the covariance matrix $P_k(+)$ becomes too small resulting in a small K and thus eliminates the weighting on new measurements.

Complementary filter

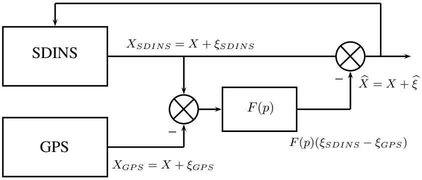

Figure: Complementary filter

$$F(p) = \begin{cases} \dfrac{1}{T_1 p + 1} & \text{if } t_{up} \leq T_1; \\ \dfrac{T_2 p + 1}{(T_2 p + 1)(T_2 p + 1)(T_2 p + 1)} & \text{if } T_1 < t_{up} \leq T_2; \\ \dfrac{T_3 p + 1}{(T_3 p + 1)(T_3 p + 1)(T_3 p + 1)} & \text{if } T_2 < t_{up}. \end{cases} \tag{1}$$

Simple INS

Equation of Motion

$$\dot{\varphi} = \frac{V_N}{R_E + h}; \dot{h} = V_U; \dot{\vartheta} = \omega - \frac{V_N}{R_E + h};$$
$$\dot{V}_N = a_N - \frac{V_N}{R_E + h} V; \dot{V}_U = a_U + \frac{V_N}{R_E + h} V_N - g; \tag{2}$$
$$a_N = a_y \cos\vartheta - a_z \sin\vartheta; a_U = a_y \sin\vartheta + a_z \cos\vartheta;$$

φ,h- latitude and height;
V_N, V_U- North and Up velocity;
a_N, a_U - North and Up acceleration;
a_y, a_z - acceleration in body frame (accelerometers output);
ϑ - pitch angle;
ω- angular velocity in body frame (gyro output);

R_E- Earth radius;

Equation of Motion

$$\delta\dot{\vartheta} = -\frac{1}{R_E + h}\delta V_N + \frac{V_N}{(R_E + h)^2}\delta h + \varepsilon \tag{3}$$

δh- INS height error; δV_N- INS North velocity error; $\delta\vartheta$- attitude error; ε- gyro bias.

Sensors Parameters

Simulation Parameters

Parameter	Value
Gyro bias	$100^\circ/hr$
Angular random walk	$1.2^\circ/\sqrt{hr}$
Accelerometer bias	$10^{-2}g$
Velocity random walk	$0.18m/s/\sqrt{hr}$
GNSS position precision	$7m(1\sigma)$
GNSS velocity precision	$0.05m/s(1\sigma)$

Estimation error of positon and velocity

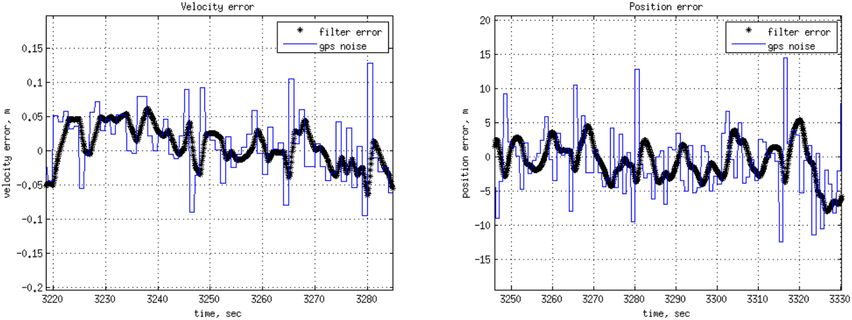

Figure: Estimation error of positon and velocity

INS error of positon and velocity

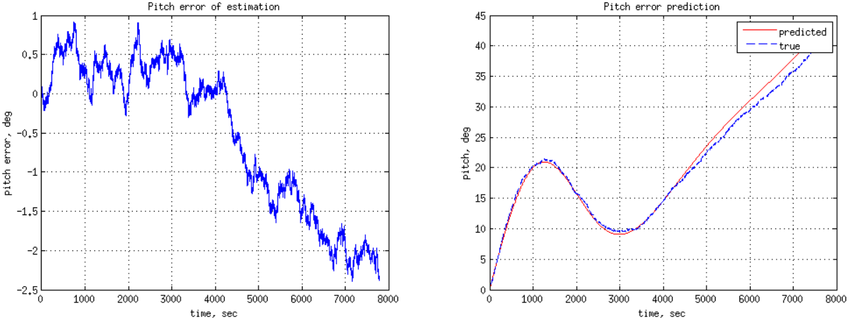

Figure: INS error of positon and velocity

INS attitude error in case of 10% error in initial condition

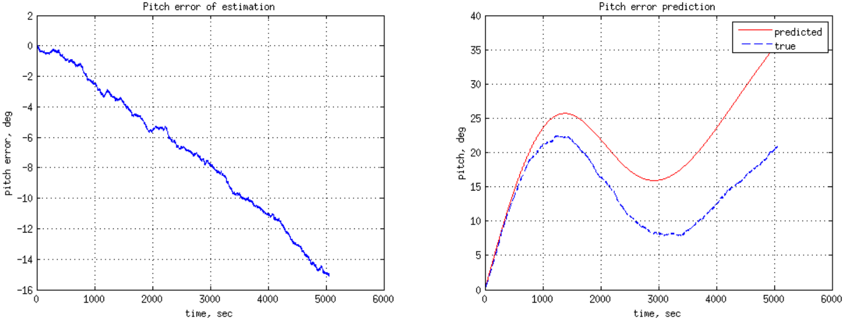

Figure: Prediction of INS attitude error in case of 10% error in initial condition for gyro bias

sudo rm -rf /

Cheers!

@isinf

https://github.com/phen0m/compf