NAVIGATION SYSTEM FOR A HELICOPTER UAV

By

Chun-Ming (Jimmy) Tsai

A THESIS SUBMITTED IN PARTIAL FULLFILLMENT

OF THE REQUIREMENTS FOR DEGREE OF

BACHELOR OF APPLIED SCIENCE
In the school of Engineering Science

SIMON FRASER UNIVERSITY

Aug 2004

email: jtsai@sfu.ca

APPROVAL

Name :	**Chun-Ming (Jimmy) Tsai**
Degree :	**Bachelor of Applied Science**
Title of thesis:	**Navigation System for a Helicopter UAV**

____

Director,

School of Engineering Science, SFU

Examining Committee:

Chair, Academic
and Technical supervisor

____

Dr. Mehrdad Saif

Director

School of Engineering Science, SFU

Committee Member

____

Dr. Kamal Gupta

Professor

School of Engineering Science, SFU

Date Approved:______________

Abstract

This thesis describes the modeling and implementation of a Kalman navigation filter on a modified model helicopter. The filter described in this thesis was designed to fuse various sensor measurements in order to improve helicopter tracking estimates. Even though Kalman filtering is applicable to any sensor measurement, this thesis focused on combining the complementary properties of the global position system and inertial navigation system. The filter implemented in this thesis helped the Simon Fraser University Aerial Robotics Group achieve fully autonomous hovering and forward flight in their modified model helicopter.

Acknowledgements

Thank you Mom and Dad.
Thank you all members of ARG
Especially Pavel, Marc, Shahin, Michael, and Lawrence
Specially thanks to people in “Autopilot” project for answering a lot of my questions.

Symbols and Acronyms

GPS	global positioning system
INS	inertial navigation system
IMU	inertial measurement unit
KF	Kalman filter
EKF	extended Kalman filter
NED	North East Down (also known as local frame)
ECEF	Earth Center Earth fixed frame
R_{body}^{local}	body to local frame transformation matrix
x	state vector
z	measurement vector
P	error covariance matrix (discrete case) error spectral density matrix (continuous case)
Q	process noise covariance matrix (discrete case) process noise Spectral Density Matrix (continuous case)
R	measurement noise covariance matrix (discrete case) measurement noise Spectral Density Matrix (continuous case)
K	Kalman gain matrix
p, q, r	angular rates around helicopter body axis
a	acceleration
^	superscript denotes estimated values
~	superscript denotes error of the associated variable
(-)	operator denotes a value before measurement update
(+)	operator denotes a value after measurement update
e0~3	quaternion angles (e0, e1, e2, e3)
u, v, w	velocity in the body frame
$x_{s,} y_{s,} z_s$	ECEF position of a satellite
$x_{r,} y_r, z_r$	ECEF position of a GPS receiver
ϕ,λ, h	latitude, longitude, altitude
ϕ,θ, ψ	Roll, Pitch, Yaw

Convention:

All capitalized letters symbolize matrices, while underscored non-capitalized letter are vectors. Others are scalar values. For frames representation, the subscript denotes the current frame and the superscript denotes the frame it is with respect to. Subscript label "org" stands for the origin of a frame

Table of Contents

List of Figures

1 Introduction

1.1 The Purpose and the Scope of This Thesis

International Aerial Robotics Competition (**IARC**) [1] is a collegiate competition sponsored by United States Army. Its goal is to promote the advancement of the unmanned aerial vehicle system. The competition is mission based and awards prize money of 50000 dollars for the team that successfully completes all the missions. All the missions has to be done by the robots under complete computer control.

The missions of 2004 competitions are:

1. Fly 3 kilometer following designated GPS waypoints.
2. Identify a building with a particular sign, and identify a window right underneath the sign.
3. Sending a sub-vehicle through the window found in mission 2 and search for targets inside the building.
4. Complete mission 1 to 3 in 15 minutes.

SFU Aerial Robotics Group (**ARG**) is an organization dedicated to create autonomous aerial vehicle and compete in the annual International Aerial Robotics Competition. Currently, the SFU team has people working on different aspects of the project. In Figure 1, the project is broken down into five portions[2], the navigation system, the mission controller, the ground station, and the vision system. The arrow in the figure indicates information flow. The navigation system tracks helicopter movement and send that information to the flight controller. The vision system sends information to the controller to request further investigation of possible targets. The mission/flight controller commands helicopter hardware and beams telemetry data to the ground station.

This thesis was done in conjunction with the rest of ARG efforts by other teammates, including:

Michael Mierau	PID flight controller
Shahin Roboubi	Neural network controller
Pavel Haintz	Funding & overall system integration
Marc Alfonso	Test flight pilot, electronics
Lawrence Harris	Override board and software support
Eric Leung, Eric Lee, Dimitri Tcaciuc, Saeed Hamid	Subvehicle

[1] For more information about the IARC, you can find it at webpage http://avdil.gtri.gatech.edu/AUVS/IARCLaunchPoint.html

[2] The sub-vehicle is not included in the diagram because its hardware hasn't been finalized yet.

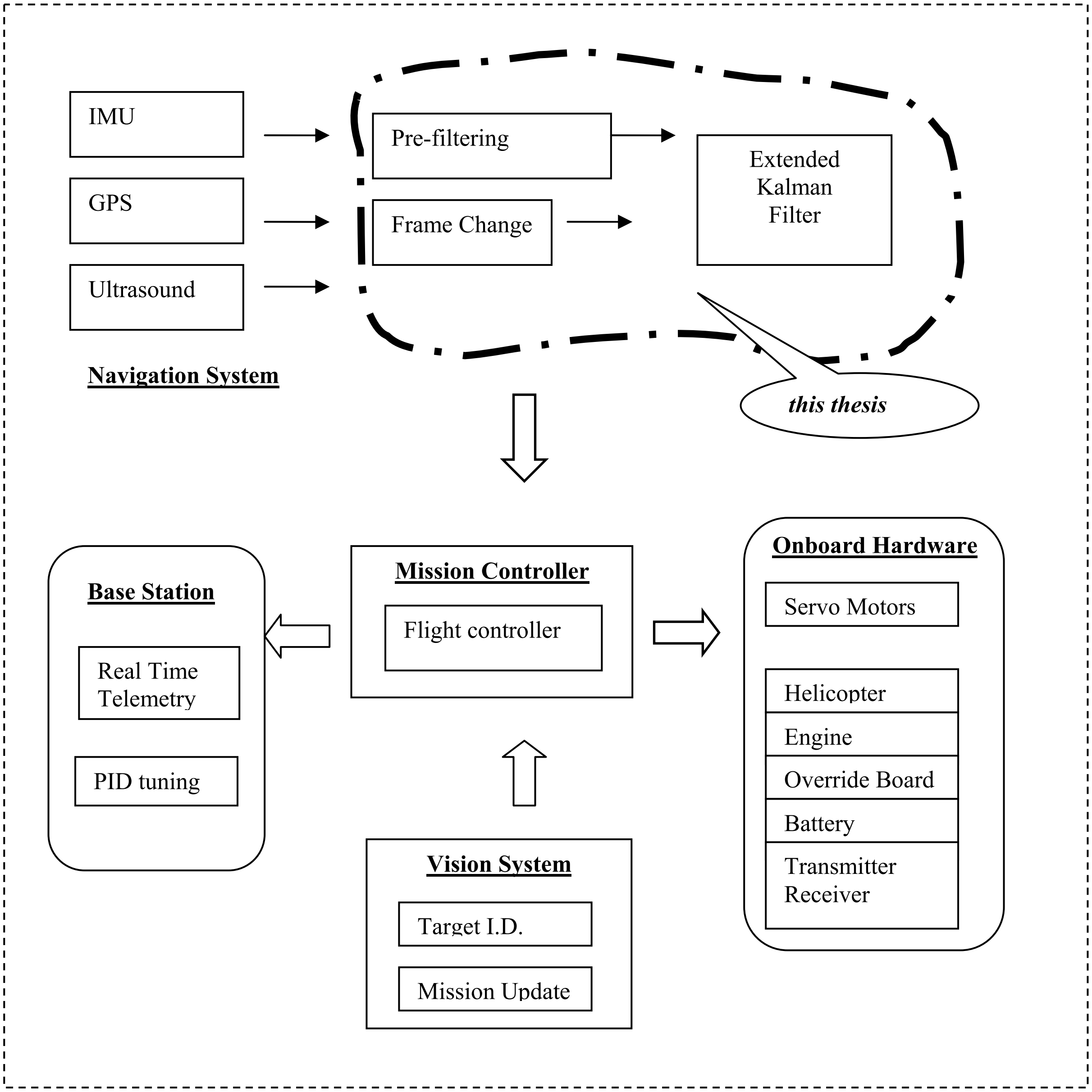


Figure 1: Scope of my thesis in the frame of the ARG project.

The scope of the thesis is the portion of Figure 1 surrounded by the dotted line. It involves conditioning the sensor signals and combining those signals in an estimation algorithm to provide fast and reliable position, velocity, and orientation information.

1.2 Description of the ARG Aerial Platform

All onboard programs ran on a PC104 embedded system. The processor board was Jumptec MOPS with Pentium 166MMX. Sensors on board of the helicopter were CSI Wireless Seres GPS and Microstrain 3dm-G inertial measurement unit. The GPS update rate is 5Hz and the 3dm-G updates at 150Hz through serial port.

Model Helicopter:
The TSK Mystar 60 is a lightweight RC controlled model helicopter powered by a 2-cycle nitro-methane fueled motor. It weights at 4.3 kg and can carry 5kg of payload.

Processor board:
Jumptec MOPS (Minimized Open PC System). This PC104 board contains 166MMX Pentium Processor, graphics controller, 2M video ram, 64M ram and 96M flash disk. The interface contains one parallel, two serial, one floppy, one keyboard and one CRT monitor connection.

Inertial Measurement Unit:
Microstrain 3dm-G Gyro Enhanced Orientation Sensor. At 96 gram, It contains all three orthogonally mounted accelerometers, ceramic rate gyros and magneto-resistive magnetometers.

GPS receiver:
CSI Seres Smart Antenna. Light weight and accurate, this GPS was capable of tracking 12 satellites at the same time. The accuracy came from two channels tracking differential correction from WAAS satellite. The GPS position update rate was five times per second while the horizontal accuracy was 1m 95% of the time.

1.3 Prolog

Since its conception in the 1960s (Kalman, 1960), the Kalman filter has become the most popular choice for navigation and tracking due to its performance, robustness, algorithmic simplicity and computational efficiency. After the members of the ARG completed the hardware for their unmanned helicopter and turned their attention to software, they realized that a Kalman navigational filter was the next natural step.

Before this filter was written, GPS and tilt sensor output were used directly as the input to helicopter controller. Position measurement from GPS was available at 5Hz. The tilt sensor (being it fluid damped or accelerometer based) will produce incorrect angle output in the presence of any non gravitational force.

The filter described in this thesis addresses the above issue and provides state estimates (position, body velocity, and orientation) at a faster rate and better accuracy. Even if the GPS quality fluctuates, the filter will adjust the ratio of GPS/INS data to sustain the best estimate possible.

The section 1.5 explains the complementary nature of the GPS and INS. The background frame transformation required in navigation equations is explained in entire section 2. Section 3 contains a brief derivation of the Kalman filter. Section 4 and 5 deals with the formulation of the state propagation and update equations, while the in section 6 covariance matrix determination is discussed. The next three section (7,8,9) address practical concerns in implementing the filter. The actual results of the filter produced during flight is in the final section.

1.4 Complementary Nature of the GPS/INS

	GPS	INS
Measurement Principle	Distance from time delays	Inertial accelerations
Output variable	Position, Time	Orientation angle, velocity
Long wavelength errors	Low	High
Short wavelength errors	High	Low
Data rate	Low (less than 5Hz)	High (higher than 100 Hz)

Table 1: Difference between GPS and INS measurement.

Global Positioning System (GPS) and Inertial Navigation System (INS) have complementary properties (Jekeli, 2001). Table 1 provides a comparison for the two types of sensor.

Inertial measurement (i.e. gyro and accelerometer) produces angular rate and acceleration with respect to the body frame. It provides high short-term resolution but the measurement can be noisy. In addition, it is necessary to calculate position by integrating the already noisy data. Because of the integration, error (especially bias error) can become unbounded if not corrected. However, it does not require external signal to produce measurements.

On the other hand, GPS describes position in earth frame with a bounded maximum error. Its down side is the slow data output and the possibility of temporary signal loss due to weather and satellite reception. Thus GPS alone can not guarantee quality information for the navigation system on our helicopter.

Our 3dm IMU has a proprietary filter built in for orientation and its filter provide a certain degree of guard against unwanted acceleration induced angle change. However, the filter is limited to only gyro value feedback adjustment for accelerometers.

By combining the two sensors using Kalman filter, we can keep the strength and reduce the shortcoming of both sensors. The first step is to calculate position from the inertial sensors data as a temporary substitute of GPS data during the time that GPS signal is not available or degraded. The second step is to use the difference between GPS and INS data to estimate the error of the INS data. Once the error is given, INS data can be corrected and the integration error can be contained. In summary, the overall aim of the complimentary filter is to create a navigational estimator capable of the speed of the inertial sensors and the stable error bound of the GPS system.

2 Frames

As with robotic applications, navigational computations use frames to describe which platform the movement is with respect to. For example, GPS returns absolute position measurements in Geodetic frame while the gyros returns data in body frame. To effectively combine the data from different frames, we need to know the method of transforming one frame into another. The navigational frames used in this thesis are the NED, ECEF, Geodetic, and body frame.

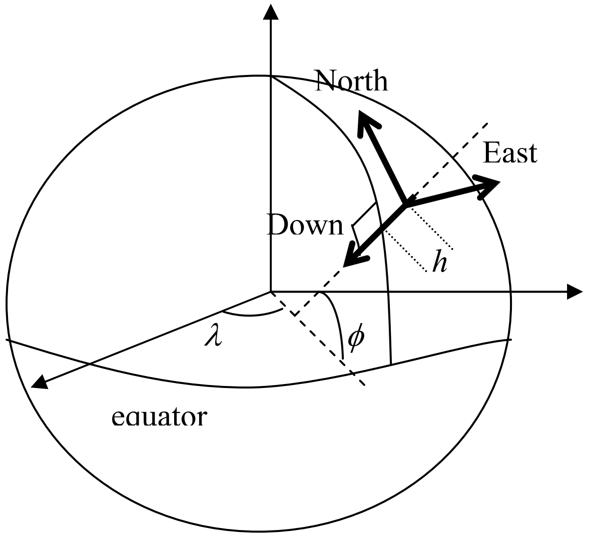


Figure 2: Geodetic and Local Frame

2.1 NED (local) Frame

This frame describes the overall orientation and distance of the vehicle with respect to a fixed origin with axis pointing to earth's true north, east, and along the gravity vector. Also called the North, East, Down (NED) frame, this frame is used when the scope of navigation is small enough so that the curvature of the earth is irrelevant. Since the ARG helicopter did not require long range capability, local frame was the navigation frame of choice.

2.2 Geodetic

Latitude (ϕ), longitude (λ), and altitude (h) are the three parameter of this curvilinear coordinate frame. Latitude is the angle between the Greenwich meridian line and the position vector projected onto the equatorial plane. The angle of longitude is between the plane of equator and the gravity vector. The altitude is the height of the measured location with respect to the average ocean water level. Typically, GPS returns data at this frame. Refer to Figure 2 to see the illustration of the angles.

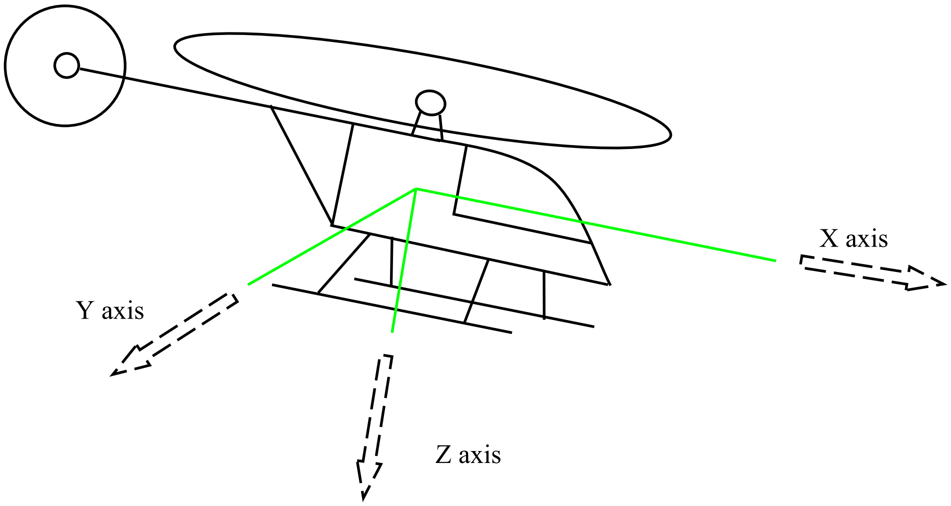


Figure 3: Body coordinate of the helicopter

2.3 Body frame

The body frame is attached to the center of mass of the remote vehicle. This is the frame that the inertial sensors is meant to measure angular rate and acceleration in. However, because it is difficult to put the inertial sensors precisely at the center of mass of the vehicle, unwanted acceleration component needs to be processed out.

2.4 ECEF

“Earth Center Earth Fixed”. This frame has its origin at the center of gravity of the earth. This also has a rectangular coordinate. Its coordinate axes are defined by polar axis, equator, and the Greenwich meridian. This frame is useful because it is an intermediate frame between the local frame and the geodetic frame.

Transformations from local to body frame and geodetic to local frame are described in appendix A. In addition, attitude representation using Euler angle and quaternion are discussed in appendix A as well.

3 Discrete Kalman filter Basics

This section is a very short introduction of Kalman filter. For a much more detail and better explanation, the reader is advised to read R.E. Kalman's famous paper (Kalman, 1960) or many books that deals with the applications of his work. The derivation presented here is from the book "Applied Option Estimation" (A. Gelb, 1974).

Kalman filter is a linear recursive optimum estimator. It models the process and measurement noise as Gaussian white noise.

The definition of optimality here is
"unbiased" - expected value is the same as the quantity being estimated
"minimum variance" – error variance is less then or equal to any other unbiased estimate.
"consistent" – estimate converges to true value as measurement increases.

Moreover, it has been proven that Kalman filter, under Guassian noise system, would be the best filter because a non-linear filter that is better then Kalman filter can not exist (R.G. Brown & Hwang. 1997)

Kalman filter is based on prediction, comparison, and correction using the known stochastic model of the plant and measurement uncertainties. Error covariance matrix is computed every step to provide a measure of accuracy. Section 3.2 and 3.4 attempt to give a simple derivation of the system prediction/propagation and update/correction.

3.1 Modeling Assumption

The state vector is denoted as x and the subscript k stands for a discrete time instant k times the sampling time. The state estimate fused with new measurements is denoted with (+) while the estimate before fusion is denoted with (-)

Process:

$$\underline{x}_k = \Phi_{k-1}\underline{x}_{k-1} + \underline{w}_{k-1} \qquad \underline{w}_k \approx N(0, Q_k) \qquad \text{eq. 1}$$

Measurement:

$$\underline{z}_k = H_k\underline{x}_k + \underline{v}_k \qquad \underline{v}_k \approx N(0, R_k) \qquad \text{eq. 2}$$

w and *v* are zero mean Gaussian white noise (N stands for Gaussian/normal distribution) with respective covariance matrix Q_k and R_k. Φ_k and H_k matrices are the state transition and observation matrix. In addition, covariance between *w* and *v* is zero.

$$E[\underline{w}_k\underline{v}_j^T] = 0 \qquad \text{eq. 3}$$

The estimate propagation is computed through multiplication with the state transition matrix.

$$\underline{\hat{x}}_k = \Phi_k\underline{\hat{x}}_{k-1} \qquad \text{eq. 4}$$

3.2 Propagation of error variance

We then define the error of the estimated value. The $\tilde{x}$ is the error, the $\hat{x}$ is estimated value.

$$\tilde{x}_k(+) = \hat{x}_k(+) - x_k \qquad \text{eq. 5}$$

$$\tilde{x}_k(-) = \hat{x}_k(-) - x_k \qquad \text{eq. 6}$$

And then define the error covariance matrix before the observation update

$$P_k(-) = E\left[\tilde{x}_k(-)\tilde{x}_k(-)^T\right] \qquad \text{eq. 7}$$

Substitute in $\tilde{x}_k(-)$ from equation 6, definition of x from equation 1, and simplify the result $P_k(-)$ equation using the fact that white noises are uncorrelated with other parameters, i.e.

$$E\left[\tilde{x}_k w_k{}^T\right] = 0 \qquad \text{eq. 8}$$

The result is that P_k can be computed from the covariance information of the last time slice:

$$P_k = \Phi_{k-1} P_{k-1} \Phi_{k-1}{}^T + Q_{k-1} \qquad \text{eq. 9}$$

3.3 State Estimate Update

We start with a linear combination of the past state and measurement as our estimate. Then we will show that K' depends on *K*.

$$\underline{\hat{x}}_k = K'_k \hat{x}_{k-1} + K_k \underline{z}_k \qquad \text{eq. 10}$$

Having a unbiased estimate means

$$E\left[\underline{\hat{x}}_k - \underline{x}_k\right] = 0 \qquad \text{eq. 11}$$

Substitute $\underline{\hat{x}}_k$ from equation 10 into the error expression,

$$\begin{aligned}
&\underline{\hat{x}}_k - \underline{x}_k \\
&= K'_k \underline{\hat{x}}_{k-1} + K_k \underline{z}_k - \underline{x}_k \qquad \text{eq. 12} \\
&= K'_k\left[\hat{x}_{k-1} + x_{k-1}\right] + \left[K_k H_k \Phi_{k-1} - \Phi_{k-1} + K_k\right] x_{k-1} + \left(K_k H_k - I\right) w_k + K_k v_{k-1}
\end{aligned}$$

Substitute equation 12 into equation 11, and solve for K', we get an expression of K' dependent of K.

$$\begin{aligned}&E[\hat{\underline{x}}_k - \underline{x}_k] = [K_k H_k \Phi_{k-1} - \Phi_{k-1} + K'_k]E[\underline{x}_{k-1}] = 0\\&\Rightarrow K'_k = (I - K_k H_k)\Phi_{k-1}\end{aligned}\qquad \text{eq. 13}$$

By bring in K' of equation 13 back into equation 10, we get

$$\begin{aligned}&\underline{x}_k(+) = \Phi_{k-1}\underline{x}_{k-1} + K_k(\underline{z}_k - H_k \Phi_{k-1}\underline{x}_{k-1})\\&= \underline{x}_k(-) + K_k[\underline{z}_k - H_k \underline{x}_k(-)]\end{aligned}\qquad \text{eq. 14}$$

3.4 Determination of Kalman Gain and Updated Error Covariance Matrix

Now we can use equation 14 and the property of Gaussian noise to determine the Kalman gain and the updated covariance matrix. We start with the updated error expression and show that it is dependent on the Kalman gain K_k, measurement noise, and past error.

$$\begin{aligned}\tilde{\underline{x}}_k(+) &= \hat{\underline{x}}_k(+) - \underline{x}_k\\&= [I - K_k H_k]\hat{\underline{x}}_k(-) + K_k \underline{z}_k - \underline{x}_k\\&= [I - K_k H_k][\hat{\underline{x}}_k(-) - \underline{x}_k] + K_k \underline{v}_k\\&= [I - K_k H_k]\tilde{\underline{x}}_k(-) + K_k \underline{v}_k\end{aligned}\qquad \text{eq. 15}$$

Then the new error covariance matrix after update is shown to be dependent on past error covariance matrix and the measurement covariance matrix R_k.

$$\begin{aligned}P_k(+) &= Cov(\tilde{\underline{x}}_k(+))\\&= E\left[\tilde{\underline{x}}_k(+)(\tilde{\underline{x}}_k(+))^T\right]\\&= (I - K_k H_k)P_k(-)(I - K_k H_k)^T + K_k R_k K_k^T\end{aligned}\qquad \text{eq. 16}$$

To satisfy the minimum variance requirement, the trace of the error covariance matrix has to be minimized. That is the condition we use to determine the Kalman gain matrix.
Take matrix derivative of trace of P with respect to the K_k

$$\frac{\delta Trace(P_k(+))}{\delta K_k} = -2P_k H_k^T + 2K_k H_k P_k(-)H_k^T + 2K_k \qquad \text{eq. 17}$$

Equate the above derivative to zero and then solve for the optimum Kalman gain.

$$K_k = P_k(-)H_k^T[H_k P_k(-)H_k^T + R_k]^{-1} \qquad \text{eq. 18}$$

Now that we know what Kalman gain K_k is, we simplify $P_k(+)$ by substituting in equation 18 back into equation 16, and after much manipulation, the expression for $P_k(+)$ simplifies to

$$P_k(+) = (I - K_k H_k)P_k(-) \qquad \text{eq. 19}$$

The overall Kalman filter computation step are summarized in Table 2 below.

System Model Measurement Model	$\underline{x}_k = \Phi_{k-1}\underline{x}_{k-1} + \underline{w}_{k-1} \qquad \underline{w}_k \approx N(0,Q_k)$ $\underline{z}_k = H_k \underline{x}_k + \underline{v}_k \qquad \underline{v}_k \approx N(0,R_k)$
Initial Conditions Other Assumptions	$E[x(0)] = \hat{x}_0, E[(x(0)-\hat{x}_0)(x(0)-\hat{x}_0)^T] = P_0$ $E[\underline{w}_k \underline{v}_j^T] = 0$ for all j and k
State Estimate Extrapolation Error Covariance Extrapolation	$\hat{\underline{x}}_k = \Phi_k \hat{\underline{x}}_{k-1}$ $P_k = \Phi_{k-1} P_{k-1} \Phi_{k-1}^T + Q_{k-1}$
State Estimate Update Error Covariance Update Kalman Gain Matrix	$\hat{\underline{x}}_k(+) = \hat{\underline{x}}_k(-) + K_k[\underline{z}_k - H_k \hat{\underline{x}}_k(-)]$ $P_k(+) = (I - K_k H_k) P_{k-1}(-)$ $K_k = P_k(-) H_k^T [H_k P_k(-) H_k^T + R_k]^{-1}$

Table 2: Summary of the discrete Kalman filtering equations

3.5 Extended Kalman Filter (Continuous – Discrete)

The filter implemented in this thesis is different then the filter in table 2 because the helicopter system model is not linear.

$$\begin{aligned} &\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t),t) + \underline{w}(t) && \underline{w}(t) \approx N(0,Q(t)) \\ &\underline{z}_k = \underline{h}_k(\underline{x}(t_k)) + \underline{v}_k \quad k = 1,2,3... && \underline{v}_k \approx N(0,R_k) \end{aligned} \qquad \text{eq. 20}$$

In this system model, the vector $\underline{f}$ is a nonlinear function of the state and $\underline{w}(t)$ is zero mean Gaussian noise having spectral density matrix *Q(t)*. In the observation model, the function $\underline{h}_k$ depends upon the state and the sampling time, while $\underline{v}_k$ is a discrete white noise sequence associated with covariance matrix *R$_k$*. In a nutshell, the helicopter filter model is set up as a non-linear system having continuous plant dynamics and discrete time measurement. Even though the helicopter state dynamic is continuous, a discrete system formulation is also appropriate since code implementation requires solving the differential equation in discrete time. Section 4 will contain a detailed description of the plant model.

The primary way to treat a non-linear system model is through the use the so-called "extended Kalman filter". It uses first order approximation of the non-linear system but it nevertheless has been proven in theory and in practice to be adequate. The major problem with extended Kalman filter is as non-linearity increases, the filters performance suffer and the risk of instability increases. Higher order approximations through Taylor series expansion can also be used, albeit at the cost of requiring more computation power. The derivation of the extended Kalman filter can also be found in the (A. Gelb, 1960) and (Brown and Hwang, 1997).

To compute the needed covariance matrix and Kalman gain matrix, approximations of the continuous state transition matrix $F(\hat{\underline{x}}(t),t)$ and discrete observation matrix $H_k(\hat{\underline{x}}(t_k))$ needs to be found. Extended Kalman filter produces the these two matrices by taking the Jacobian of the propagation and update function with respect to the estimated state vectors. i.e. linearization by $\hat{\underline{x}}$.

Equation 21 to 24 contains the Extended Kalman filter algorithm. The basic structure of this filter is the same as the discrete Kalman filter discussed above. The filter performs system and covariance propagation when no measurement is available. Once a measurement becomes available, the filter goes through covariance update, gain matrix update, and system estimate update.

Linearization

$$\begin{aligned} F(\hat{\underline{x}}(t),t) &= \left.\frac{\partial f(\underline{x}(t),t)}{\partial \underline{x}(t)}\right|_{\underline{x}(t)=\hat{\underline{x}}(t)} \\ H_k(\hat{\underline{x}}_k(-)) &= \left.\frac{\partial \underline{h}_k(\underline{x}(t_k)}{\partial \underline{x}(t_k)}\right|_{\underline{x}(t_k)=\hat{\underline{x}}_k(-)} \end{aligned} \tag*{eq. 21}$$

Initial condition and assumption

$$\begin{aligned} & x(0) \approx N(x0,P0) \\ & E\left[w(t){v_k}^T\right] = 0 \end{aligned} \tag*{eq. 22}$$

State Estimate Propagation

$$\begin{aligned} & \dot{\hat{\underline{x}}}(t) = f(\hat{x}(t),t) \\ & \dot{P}(t) = F(\hat{\underline{x}}(t),t)P(t) + P(t)F^T(\hat{\underline{x}}(t),t) + Q(t) \end{aligned} \tag*{eq. 23}$$

State Estimate Update , Error Covariance Update, and Gain Matrix

$$\begin{aligned} & \hat{\underline{x}}_k(+) = \hat{\underline{x}}_k(-) + K_k\left[z_k - h_k(\hat{\underline{x}}_k(-))\right] \\ & P_k(+) = \left[I - K_k H_k(\hat{\underline{x}}_k(-))\right]P_k(-) \\ & K_k = P_k(-){H_k}^T((\hat{\underline{x}}_k(-))\left[H_k(\hat{\underline{x}}_k(-))P(-){H_k}^T(\hat{\underline{x}}_k(-)) + R_k\right]^{-1} \end{aligned} \tag*{eq. 24}$$

3.6 Steady State Kalman Filter

The requirement of computing 13 by 13 state propagation F matrix and P matrix every iteration is the most computationally intensive part of this Kalman filter implementation. Even though the helicopter computer stack seems to be capable of computing through these equations, it would be beneficial to avoid doing such computation. In addition, since first order linearization of the extended Kalman filter can only perform well under limited flight dynamic, a fixed gain Kalman filter might do equally well to replace the filter implemented in this thesis work, especially under steady state conditions such as hovering and simple forward flight. Transition between different mode of flight might still need to be computed using extended Kalman filter algorithm.

Using a steady state F matrix will enable the solution of the P matrix to be computed ahead of time through solving matrix Riccati equation. If a steady state observation H matrix can also be found, then even a suitable Kalman gain can be determined ahead of time. In this thesis work no such attempt was made; but since the ARG project in SFU is on-going, future navigation filter development will definitely consider this possibility.

4 Formulation of the state propagation equations and matrix

4.1 State Vector and Plant Dynamic

The states of the helicopter tracked in this filter are the following

$$\underline{\hat{x}} = \begin{bmatrix} x & y & z & u & v & w & e_0 & e_1 & e_2 & e_3 & p_{bias} & q_{bias} & r_{bias} \end{bmatrix}^T$$

Angular rates ***(p, q, r)*** for roll, pitch, and yaw axis are not estimated in the filter but the gyro biases are. By subtracting the gyro bias from the raw gyro measurement, the helicopter flight controller will get a better angular rates

NED Position	x, y, z
Body linear Velocity	u, v, w
Quaternion Orientation	e_0, e_1, e_2, e_3
Gyro biases	$p_{bias}, q_{bias}, r_{bias}$

In this case, the state propagation equations deal with the dynamic relationship of the onboard sensor and the helicopter. That is, how does the measurement of the gyro and accelerometer, produce the state of the helicopter by means of integration? The next four steps are the answer.

1. Gyro measurement is integrated to provide orientation estimate in the form of 3x3 orientation matrix and 1x4 quaternion vector.

2. Given the orientation estimate, gravity component of acceleration is eliminated from the accelerometer measurement. And then centripetal acceleration and acceleration caused by IMU placement offset from the rotor head is computed and eliminated. The reminding component in the accelerometer measurement is the linear acceleration in body frame.

3. Integrate the body frame linear acceleration to get body velocity. Convert the body velocity to local frame velocity and integrate the local velocity to get local position.

4. The derivative of gyro bias is white noise; so the portion of $\underline{f}(\underline{x}(t), t)$ vector related to gyro biases is zero.

$$\begin{bmatrix} {}^{\partial p_{bias}}\!/\!_{\partial t} \\ {}^{\partial q_{bias}}\!/\!_{\partial t} \\ {}^{\partial r_{bias}}\!/\!_{\partial t} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{eq. 25}$$

This is discussed in more detail in section 4.7.

4.2 Orientation Propagation

We integrate gyro for quaternion angles. Equation 26 and 27 described the relationship between quaternion derivative and gyro angular rates (Rönnbäck, 2000).

$$\begin{bmatrix} \partial e_0 / \partial t \\ \partial e_1 / \partial t \\ \partial e_2 / \partial t \\ \partial e_3 / \partial t \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -e_1 & -e_2 & -e_3 \\ e_0 & -e_3 & e_2 \\ e_3 & e_0 & -e_1 \\ -e_2 & e_1 & e_0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \qquad \text{eq. 26}$$

$$\begin{bmatrix} \partial e_0 / \partial t \\ \partial e_1 / \partial t \\ \partial e_2 / \partial t \\ \partial e_3 / \partial t \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{bmatrix} \qquad \text{eq. 27}$$

4.3 Acceleration Propagation

The accelerometer does not just measure linear acceleration. It measures

4.3.1 Centripetal Acceleration

In 2D case

$$a_n = \frac{v^2}{r} = \omega v \qquad \text{eq. 28}$$

In 3D case

$$\begin{bmatrix} a_{n\_x} \\ a_{n\_y} \\ a_{n\_z} \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \qquad \text{eq. 29}$$

We have the gyro measurements for angular velocity and velocity estimate, therefore we can compute a_n

4.3.2 Gravity

Gravitational force is always pointing downward. Since we are using the body frame velocity as our state vector, we will have to project the gravity accordingly.

$$\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = R_{local}^{body} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} 2g(e_1e_3 - e_0e_2) \\ 2g(e_2e_3 + e_0e_1) \\ g(e_0^2 - e_1^2 - e_2^2 + e_3^2) \end{bmatrix} \qquad \text{eq. 30}$$

4.3.3 Linear Acceleration

This is the component that we want to determine. we will call it $\begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T$

4.3.4 Centripetal Acceleration due to Offset From Rotor Head

This acceleration reading is caused by the fact our IMU was not located at the rotor head of the helicopter. So as the helicopter is yawing, an additional force would be measured. Defining vector r is the distance vector between rotor head and the IMU, we get

$$\begin{bmatrix} a_{imu\_x} \\ a_{imu\_y} \\ a_{imu\_z} \end{bmatrix} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left[\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \right] \qquad \text{eq. 31}$$

4.3.5 All Acceleration Combined

combine all the above equations in section 4.3

$$\begin{bmatrix} \partial u / \partial t \\ \partial v / \partial t \\ \partial w / \partial t \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} - \begin{bmatrix} a_{n\_x} \\ a_{n\_y} \\ a_{n\_z} \end{bmatrix} - \begin{bmatrix} a_{imu\_x} \\ a_{imu\_y} \\ a_{imu\_z} \end{bmatrix} - \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \qquad \text{eq. 32}$$

Expand all the terms, the above equation becomes

$$\begin{bmatrix} \partial u / \partial t \\ \partial v / \partial t \\ \partial w / \partial t \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left[\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \right] - \begin{bmatrix} 2g(e_1e_3 - e_0e_2) \\ 2g(e_2e_3 + e_0e_1) \\ g(e_0^2 - e_1^2 - e_2^2 + e_3^2) \end{bmatrix} \qquad \text{eq. 33}$$

4.4 Velocity Propagation

This differential equation is basically a frame change.

$$\begin{bmatrix} dx/dt \\ dy/dt \\ dz/dt \end{bmatrix} = R_{body}^{local}\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} (e_0^2+e_1^2-e_2^2-e_3^2) & 2(e_1e_2-e_0e_3) & 2(e_1e_3+e_0e_2) \\ 2(e_1e_2+e_0e_3) & (e_0^2-e_1^2+e_2^2-e_3^2) & 2(e_2e_3-e_0e_1) \\ 2(e_1e_3-e_0e_2) & 2(e_2e_3+e_0e_1) & (e_0^2-e_1^2-e_2^2+e_3^2) \end{bmatrix}\begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

eq. 34

4.5 Overall State Propagation Equations

From section 4.1 to 4.4 we get all the differential equations we need to describe $f(\underline{x}(t),t)$

$$\begin{bmatrix} dx/dt \\ dy/dt \\ dz/dt \end{bmatrix} = R_{body}^{local}\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} (e_0^2+e_1^2-e_2^2-e_3^2) & 2(e_1e_2-e_0e_3) & 2(e_1e_3+e_0e_2) \\ 2(e_1e_2+e_0e_3) & (e_0^2-e_1^2+e_2^2-e_3^2) & 2(e_2e_3-e_0e_1) \\ 2(e_1e_3-e_0e_2) & 2(e_2e_3+e_0e_1) & (e_0^2-e_1^2-e_2^2+e_3^2) \end{bmatrix}\begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$\begin{bmatrix} \partial u/\partial t \\ \partial v/\partial t \\ \partial w/\partial t \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left[\begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}\right] - \begin{bmatrix} 2g(e_1e_3-e_0e_2) \\ 2g(e_2e_3+e_0e_1) \\ g(e_0^2-e_1^2-e_2^2+e_3^2) \end{bmatrix}$$

$$\begin{bmatrix} \partial e_0/\partial t \\ \partial e_1/\partial t \\ \partial e_2/\partial t \\ \partial e_3/\partial t \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix}\begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{bmatrix}$$

$$\begin{bmatrix} \partial p_{bias}/\partial t \\ \partial q_{bias}/\partial t \\ \partial r_{bias}/\partial t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

4.6 Linearization for the State Propagation Matrix

The formulation of the extended Kalman filter require us to linearize the $f(\bar{x}(t),t)$ by the present estimated values, and then use it compute the state transition matrix. Here we only use the linearization to produce a 10x10 sub matrix. The portion for gyro biases is omitted because it will be treated differently and the overall 13x13 state matrix discussed in section 4.7. So the $f(\bar{x}(t),t)$ in equation 35 does not contain the three rows of bias equations.

$$F_{10x10}\Big|_{\underline{x}(t)=\hat{\underline{x}}(t)} = \left[\frac{\partial f}{\partial \overline{x}}\right]_{\underline{x}(t)=\hat{\underline{x}}(t)} = \left[\frac{\partial f}{\partial x},\frac{\partial f}{\partial y},\frac{\partial f}{\partial z},\frac{\partial f}{\partial u},\frac{\partial f}{\partial v},\frac{\partial f}{\partial w},\frac{\partial f}{\partial e_0},\frac{\partial f}{\partial e_1},\frac{\partial f}{\partial e_2},\frac{\partial f}{\partial e_3}\right]_{\underline{x}(t)=\hat{\underline{x}}(t)} \qquad \text{eq. 35}$$

The function $f(\overline{x}(t),t)$ contains no position dependent terms. Therefore the derivative with respect to position is zero. The derivative with respect to velocity contains the body-to-navigation transformation matrix and skew-symmetric matrix of angular rates.

$$\left[\frac{\partial f}{\partial x},\frac{\partial f}{\partial y},\frac{\partial f}{\partial z}\right] = 0^{9x3} \quad \text{eq. 36} \qquad \left[\frac{\partial f}{\partial u},\frac{\partial f}{\partial v},\frac{\partial f}{\partial w}\right] = \begin{bmatrix} R_{body}^{local} \\ \begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} \\ 0^{4x3} \end{bmatrix} \qquad \text{eq. 37}$$

The derivative with respect to the quaternion is more complicated. The first three row is derivative from body to navigation transformation matrix.

$$\left[\frac{\partial f}{\partial \overline{e}}\right]_{rows1-3} = 2*\begin{bmatrix} e_0u-e_3v+e_2w & e_1u+e_2v+e_3w & -e_2u+e_1v+e_0w & -e_3u-e_0v+e_1w \\ e_3u+e_0v-e_1w & e_2u-e_1v-e_0w & e_1u+e_2v+e_3w & e_0u-e_3v+e_2w \\ -e_2u+e_1v+e_0w & e_3u+e_0v-e_1w & -e_0u+e_3v-e_2w & e_1u+e_2v+e_3w \end{bmatrix}$$
$$\text{eq. 38}$$

Rows four to six are found by taking derivative of the projected gravity vector. Rows seven to ten are direct results from taking derivative of equation 27.

$$\left[\frac{\partial f}{\partial \overline{e}}\right]_{rows4-6} = 2g\begin{bmatrix} -e_2 & e_3 & -e_0 & e_1 \\ e_1 & e_0 & e_3 & e_2 \\ e_0 & -e_1 & -e_2 & e_3 \end{bmatrix} \qquad \text{eq. 39}$$

$$\left[\frac{\partial f}{\partial \overline{e}}\right]_{rows7-10} = \frac{1}{2}\begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \qquad \text{eq. 40}$$

There is a potential problem with taking derivative with respect to all four quaternion angles. Since the quaternion angles are not independent of each other (only three values are required to represent orientation as in use of Euler angles). However, in extended Kalman filter formulation, there is no requirement for state propagation F matrix to meet, and actual result in section 9 has proven that linearization has been sufficient.

4.7 Gyro Bias Tracking

Due to the nature of integration, any gyro bias will cause the orientation estimate to become unstable. To improve the performance and the stability of the filter, all three gyro biases were estimated as part of the helicopter states.

The propagation equations and the error covariance parameters for gyro biases were modeled as “random walk”. The intuitive realization of a “random walk” process is to imagine a person making n steps, equal likelihood for walking forward or backward. After n steps, the average distance that was traveled and the variance of the distance traveled are the variables of interest.

The state variable differential equation for the random walk process is

$$\frac{\partial X(t)}{\partial t} = W(t) \qquad or \qquad X(t) = \int_{t0}^{t} W(t^{'})dt^{'}; \qquad X(t_0) = x_0 \qquad \text{eq. 41}$$

where W is a zero mean, white noise process with covariance function

$$E\left[W(t_1)\cdot W(t_2)^T\right] = q\delta(t_1, t_2) \qquad \text{eq. 42}$$

The state propagation equation for the three gyro biases is therefore:

$$\begin{bmatrix} \partial p_{bias} / \partial t \\ \partial q_{bias} / \partial t \\ \partial r_{bias} / \partial t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The formulation of state transition matrix, however, will contain an additional matrix related to gyro biases

$$F_{bias} = \frac{1}{2}\begin{bmatrix} -e_1 & -e_2 & -e_3 \\ e_0 & -e_3 & e_2 \\ e_3 & e_0 & -e_1 \\ -e_2 & e_1 & e_0 \end{bmatrix} \qquad \text{eq. 43}$$

the matrix in equation 43 is exactly the same matrix that convert gyro angular rate reading into quaternion angle derivatives in equation 26.

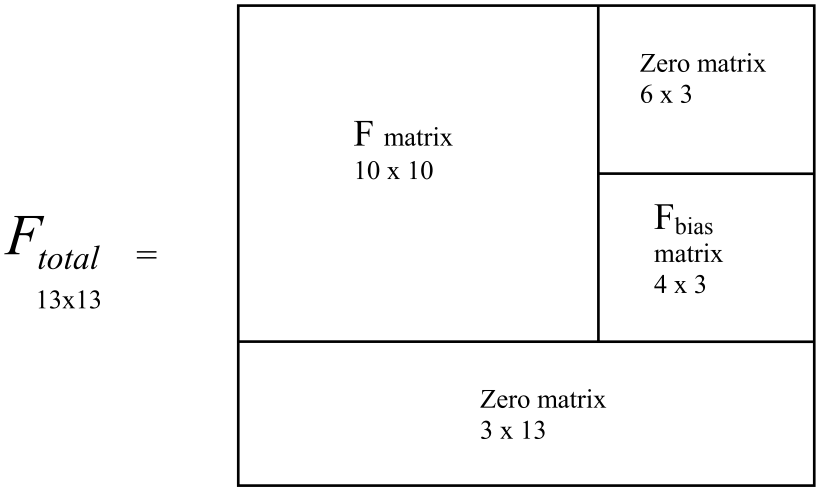


Figure 4: The composition of the overall state transition matrix F_{total}

This shows how Euler integration updates error covariance matrix with

$$P_{k+1} = P_k + \Delta t\left[F_{total} P_k + P_k F_{total}{}^T - P_K H_k{}^T R^{-1} H_K P_K + Q\right] \qquad \text{eq. 44}$$

This will force the Kalman filter to estimate gyro biases every time observation (i.e. GPS or magnetometer) comes in.

5 Formulation of the Observation Equations

The normal measurement updates come from two sources: the GPS and the magnetometer. The GPS provides speed, heading, and geodetic position (latitude, longitude, and altitude). This information is converted to local frame position and local frame horizontal velocity. The magnetometer gives the earth's magnetic vector with respect to body frame. It is used to compute yaw.

$$\underline{z}_{GPS} = \begin{bmatrix} x & y & z & v_N & v_E \end{bmatrix}^T$$
$$\underline{z}_{magnetometer} = [\psi]$$

Local Position	x, y, z
Local Velocity (horizontal only)	v_N, v_E
Yaw	ψ

In the case of prolonged GPS signal loss, the filter is programmed to estimate only the orientation. Under this condition, orientation errors from gyro integration will be corrected through update from accelerometer derived roll and pitch. The roll and pitch derived through accelerometer is long term stable but could be contaminated by short term non-gravitational force.

$$\underline{z}_{IMU} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$$

Roll, Pitch, Yaw	$\phi \quad \theta \quad \psi$

During a Kalman update, the difference (called innovation) between the measurement and the current state vector is multiplied by the Kalman gain and added to the current state vector to produce an updated estimate.

5.1 Position and Velocity Observation

Our Seres GPS provides direct measurement position, true heading and horizontal speed[3]. The position is given in the geodetic coordinate, but we can transform it into local coordinate using the equations in appendix A. By using the true heading and ground speed, we can know the helicopter velocity and position in the NED frame. However, there is an offset between the rotor head and location of the GPS antenna. So the precise location of the rotor head need to be computed using orientation information. In addition, NED velocity needs to be converted into body velocity so that we can compare the measurement and the propagated state.

[3] These information are found through GPS logs with header $GPGGA and $GPVTG

5.1.1 GPS offset compensation

By knowing the orientation, the geodetic coordinate, the location of the antenna with respect to the body coordinate, the location of the rotor head in the NED coordinate can be derived. The GPS antenna is situated on the tail boom of the helicopter nearing the tail stabilizer fins. So it is right on the x axis of the helicopter body coordinate.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{rotorhead} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}^{local}_{antenna} - R^{local}_{body} \begin{bmatrix} gps\_offset \\ 0 \\ 0 \end{bmatrix} \qquad \text{eq. 45}$$

The velocity detected by the GPS also needs to eliminate the yaw rate component, since the fuselage could be spinning and the rotor head is not moving at all.

5.2 Orientation Observation

The accelerometers can provide good roll and pitch measurement when only gravitational force is present. The MicroStrain 3dm IMU provides both normalized magnetic north vector and gravitational vector with respect to its inertia frame. The Accelerometer alone can be used for roll and pitch angle approximation

$$accel\_roll = \arcsin(ay, -\sqrt{ax^2 + ay^2 + az^2}) \qquad \text{eq. 46}$$

$$accel\_pitch = \arctan(ax, -az) \qquad \text{eq. 47}$$

Magnetometer provides the earth’s magnetic field in terms of body coordinate. Two things has to be done before the proper yaw can be derived from our magnetometer vector.

1. Compensate for “Magnetic Inclination/Dip”. The magnetic field vector has a component pointing towards the ground roughly at 70 degree[4]. If this component is not taken into account, then when the helicopter is rolling or pitching, the z portion will go into different axis and produce an incorrect yaw data. The proper way of getting yaw is to first compute estimates of the roll and pitch angle, then transform the magentic vector back to level ground and then take only the x-y component to compute heading. Equation 48 illustrates the math involved in this.

[4] Michael J. Caruso Application to Applications of Magnetoresistive Sensors in Navigation Systems Honeywell Inc.

2. Compensate for "Magentic Declination". ,i.e., convert magnetic heading to true heading. Our frame of navigation is with respect to true north instead of magnetic north. The deviation varies greatly in different parts of the world. In Vancouver area, we need to add 21 degree in the yaw.

$$\begin{aligned} heading = \arctan\{ & (-magx \cdot \sin(\phi) \cdot \sin(\theta) + magy \cdot \cos(\theta) - magz \cdot \cos(\phi) \cdot \sin(\theta), \\ & (\cos(\phi) \cdot magx + \sin(\phi) \cdot magz)\} + meg\_declination\_angle \end{aligned}$$

eq. 48

ϕ ,θ, and ψ denote roll, pitch and yaw in radians

6 Determination of the Covariance

6.1 Process Noise

6.1.1 Using Power Spectral Density

The angular rates and accelerometer is used for state propagation. Intuitively, a major portion of the state error covariance comes from the error of the 3dm sensor itself. Because this filter models its state propagation as continuous, the value of interest is the power spectral density of the 3dm gyro and accelerometer.

The matrix Q in equation 23 is the power spectral density of the 3dm gyro and accelerometer, which can be found experimentally. The problem is that linearization and computational issues such as round-off errors also contributes to the process noise of the system. Those effects are not easily quantized and should be determined experimentally.

6.1.2 Using Simulation Result from Flight Test Data

We filmed helicopter trajectory and logged sensor values during flight tests. By running simulation multiple times with different coefficients and compare the outputs, a covariance matrix could be determined. This was the main approach in this project because determining a power spectral density under general circumstance was too complicated and time consuming.

6.2 Observation Covariance

6.2.1 GPS Covariance

GPS measures pseudo-ranges to four or more satellites which are broadcasting their positions. By comparing the incoming signal with the GPS receiver signal, we can determine four things: latitude, longitude, altitude, and receiver clock offset.

The GPS errors covariance matrix diagonal elements are given from the GPS receiver logs under NMEA[5] standard.

For addition information about GPS error source, and how differential correction and wide area augmentation improves GPS accuracy, see appendix D.

6.2.2 IMU Covariance

The orientation update from the accelerometer is in long term stable but can be temporarily contaminated with non-gravitational acceleration. The IMU orientation update was only used

[5] NMEA stands for National Marine Electronics Association

when GPS measurement is not available or during a filter reset. Even so, a scheme had to be devised to determine how reliable to accelerometer roll and pitch were.

There are a few things that can be done:
1. Low passing the accelerometer data. Since linear accelerations are short term, by digitally lowpassing the signal, we can reduce the magnitude of the acceleration.

2. Checking the overall accelerometer magnitude. If there is linear acceleration, the overall magnitude of the accelerometer will be bigger or less then 9.8 meter/s. The problem is even a constant magnitude does not mean the helicopter is not accelerating. For example, the motion can be forward and downward, thus the accelerometer magnitude detected is roughly constant.

Vibration Isolation

Being able to have reliable orientation information is of paramount importance for helicopter control. However, the vibration caused by various moving components and their interactions with the frames can be as much as +/- 5m/s^2 and 0.5 rad/s. If no steps were taken to reduce the vibration, we risk the sensors saturating and signals being swamped under the noise floor. In addition, high frequency vibration data would alias into low frequency data and become mistaken for actual measurement data.

6.3 Helicopter Vibration Components

Professor E. Feron in MIT described the vibration sources and their corresponding frequency in his conference paper (Feron, 2000). We verified the noise components ourselves by measuring the accelerations and gyro rate noise during hovering. Since during hovering the inertial sensors should ideally record very little movement, most of what we measured during hovering should be noise caused by vibration. Of course, the pilot were not able to do a perfect hover, but the scale of noise dominated the raw measurement.

The major source of vibration comes from the main rotor (27~28 Hz), engine (250 Hz), and tail rotor (100 Hz) . The Table below is the experimentally determined vibration amplitude during hovering and what we aimed to achieve with hardware and software filtering. The magnetometers on board the 3dm-G does not depend on acceleration or rate input, and should not be less affected by the vibration. Our experiment result confirmed that.

Sensor	vibration amplitude	Acceptable Value[6]
X accelerometer	+/- 5 m/s^2	+/- 0.8 m/s^2
Y accelerometer	+/- 4 m/s^2	+/- 0.8 m/s^2
Z accelerometer	+/- 7 m/s^2	+/- 1.5 m/s^2
Roll gyro	+/- 43 deg/s	+/- 8 deg/s
Pitch gyro	+/- 34 deg/s	+/- 8 deg/s
Yaw gyro	+/- 14 deg/s	+/- 8 deg/s

Table 3: Magnitude of uncompensated vibration compared to the acceptable value.

[6] According to Trammel Hudson, administrator of project "autopilot: UAV Command and Control", Email discussion "Engine Noise on the inertial sensors." Wed, 29 Oct 2003. The autopilot project has successfully hovered a similar scale helicopter.

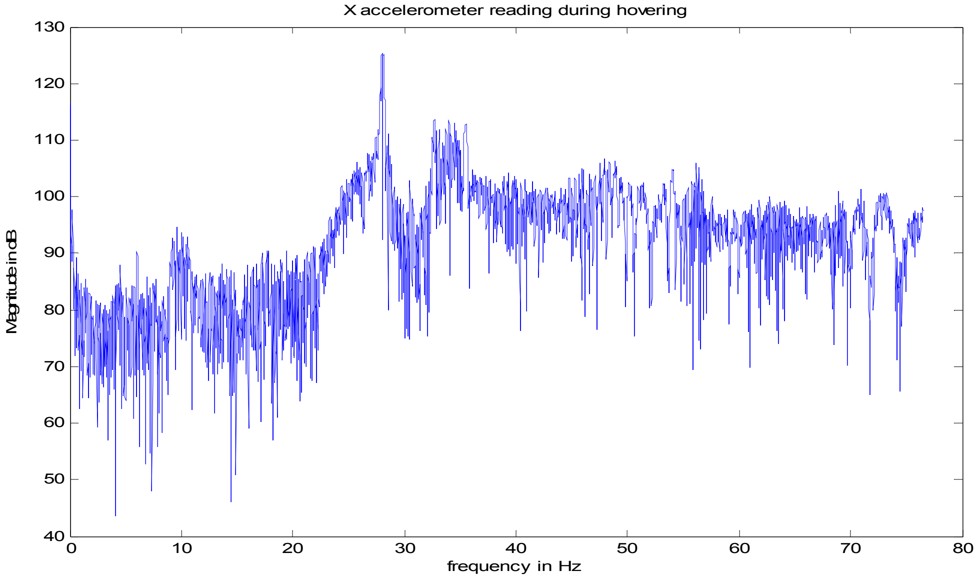


Figure 5: FFT of the X direction accelerometer during our helicopter hovering

From the X accelerometer FFT plot, it is clear that the biggest noise source comes from 27~30Hz at the rotor head frequency. This noise is caused by unbalanced rotor head motion as the blades hitting the air at high velocity. There are peaks at frequency (50~ 70). However, they are most likely noises aliased from higher frequency vibration (200 and above) source such as the engine and tail rotor. Our sensor is only capable of sampling at maximum of 150Hz; thus the vibrations such as tail rotor noise will alias into 50 ~ 70 range. There is a peak at 6~7 Hz that is structural vibration caused by our RC pilot compensating helicopter in roll and pitch to keep the vehicle steady in the air.

6.4 Solutions for Vibration Problem

We ran the sensor at its highest speed (150Hz) by forgoing the build-in Microstrain proprietary filter . By running at the maximum rate, the aliasing effect is reduced. Then we put on vibration mount and damp the sensor using lead weight. The last step was to apply digital low pass filtering. The details are recorded in appendix C.

6.5 Result of Counter Measures

Sensor	acceptable vibration magnitude	vibration magnitude after hardware and software filtering
X accelerometer	+/- 0.8 m/s^2	+/- 0.05 m/s^2
Y accelerometer	+/- 0.8 m/s^2	+/- 0.4 m/s^2
Z accelerometer	+/- 1.5 m/s^2	+/- 0.5 m/s^2
Roll gyro	+/- 8 deg/s	+/- 6 deg/s
Pitch gyro	+/- 8 deg/s	+/- 7 deg/s
Yaw gyro	+/- 8 deg/s	+/- 6 deg/s

Table 4: Comparison of acceptable vibration and the our result.

The success of our vibration isolation effort could best be verified through a comparison between pilot input and the onboard gyro recording. During a hover, the RC transmitter stick position was proportional to absolute angle of the helicopter swash plate. When compared these inputs with the gyro sensor readings, the gyro rates should be similar to the negative of the pilot servo commands.

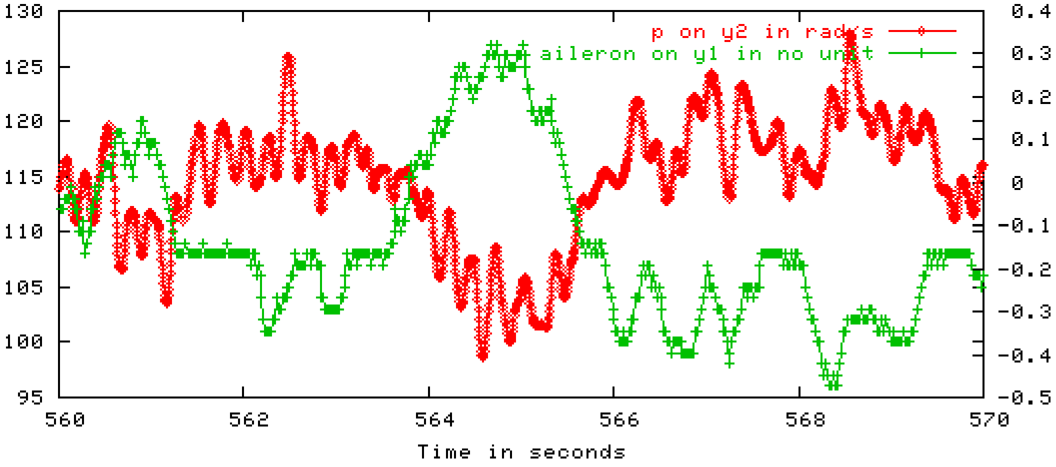


Figure 6 Pilot aileron stick position compared to x axis gyro measurement

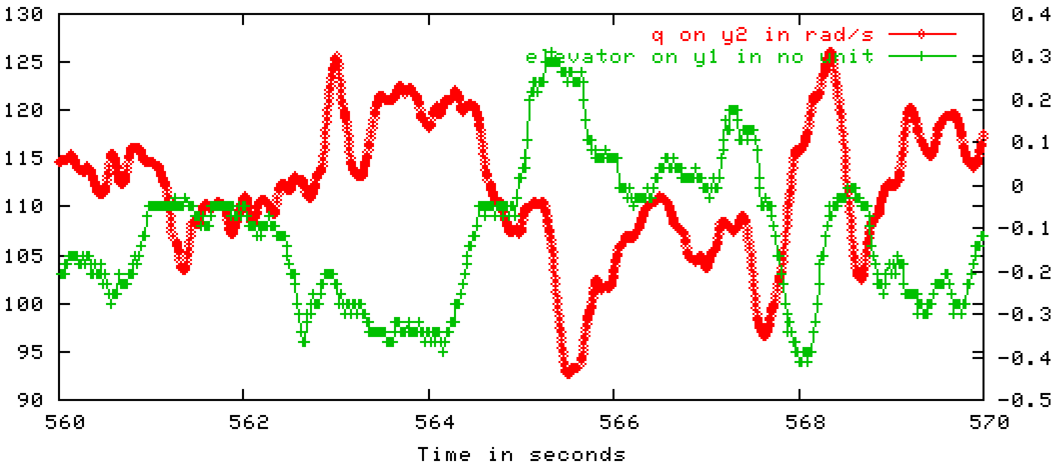


Figure 7: Pilot elevator stick position compared to y axis gyro measurement.

The result of combination of hardware and software lowpass filtering results in sufficient reduction of the vibration noise for the Kalman filter to perform sensor fusion.

7 Synchronization Issues

7.1 Latency Problem and Solution

The GPS sensor has delays due to its computational and signal transmission time. In fact, the initial heuristic estimate is around 0.17 to 0.2 second of delay between the time measurement taken and the time samples are read by our filter program. Comparing to the GPS, the 3dm IMU not only runs at a much faster rate (150Hz) but also under much lower latency.

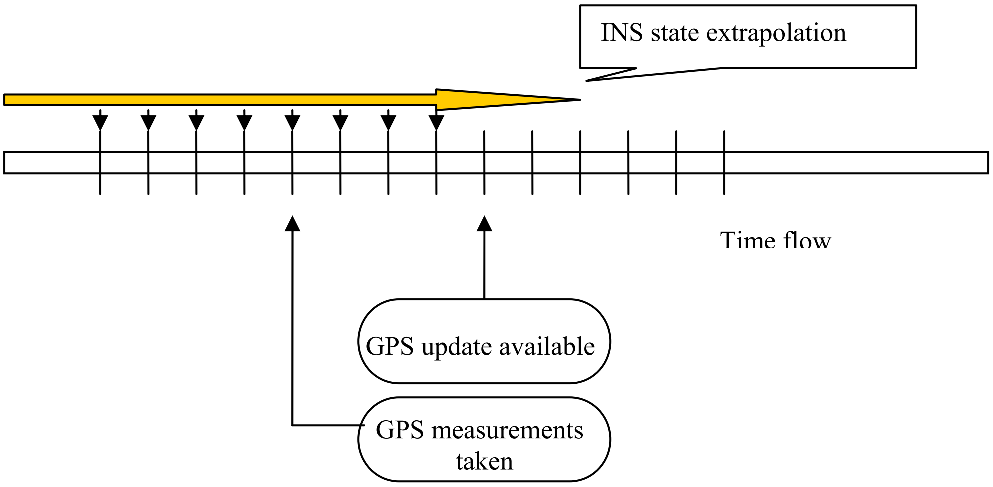


Figure 8: Illustration of the latency problem of the GPS.

During the sensor fusion, the Kalman filter model requires the measurement and the estimate to be synchronized. The consequence of not addressing the GPS latency will reduce the performance and cause instability under higher dynamic conditions.

To address this issue properly, the following steps need to be taken

1. Determine when the GPS measurement is taken.
2. The moment GPS measurement is taken, the state estimate and the error covariance matrix is buffered.
3. Buffer all the inertial measurement afterward until the GPS logs arrive.
4. The movement GPS measurement arrives from the serial port, use the buffered covariance matrix and estimate got in step 2 to produce a Kalman update.
5. Then use the buffered inertial measurement to propagate the filter estimate back to the current time. In our case, 0.2 s * 150 Hz = 30 logs buffered means that we have to iterate the estimation 30 times.

7.2 Pulse Per Second Signal

A quality GPS receiver would provide a pulse per second (PPS) signal in its output. One purpose of the PPS signal was to provide a reliable pulse every second synchronized with the atomic clocks on satellites. In addition, once every second GPS took its measurement in correspondence with the PPS signal rising edge[7]. Thus by polling the serial port PPS pin for voltage change, we could determine when a GPS measurement was taken.

The only complication was that our GPS was running at 5Hz and the PPS signal only came in once per second, so we had to estimate the measurement time within that second by adding a computer timer to the last PPS time. In fact, estimating the next PPS time this way was necessary because sometimes our software program took too long to execute and missed the PPS rising edge all together. If that happened, we could still use the estimated PPS time for latency compensation.

7.3 Computational Load

The biggest computational load came from computing the state transition matrix and the error covariance matrix. These two matrices are both 13 by 13 and had to be generated each iteration using the state estimate and buffered inertial measurements. By averaging the inertial measurements and thus increasing the step size of the integration, we were able to lower the impact of latency compensation algorithm to the overall computational load.

7.4 Experimental Latency Determination

During our test, the helicopter was subjected to an sudden push (mimicking an impulse), which took the sensors 1 meter away from its original position. The time difference between when the sensors started reporting value change was the latency.

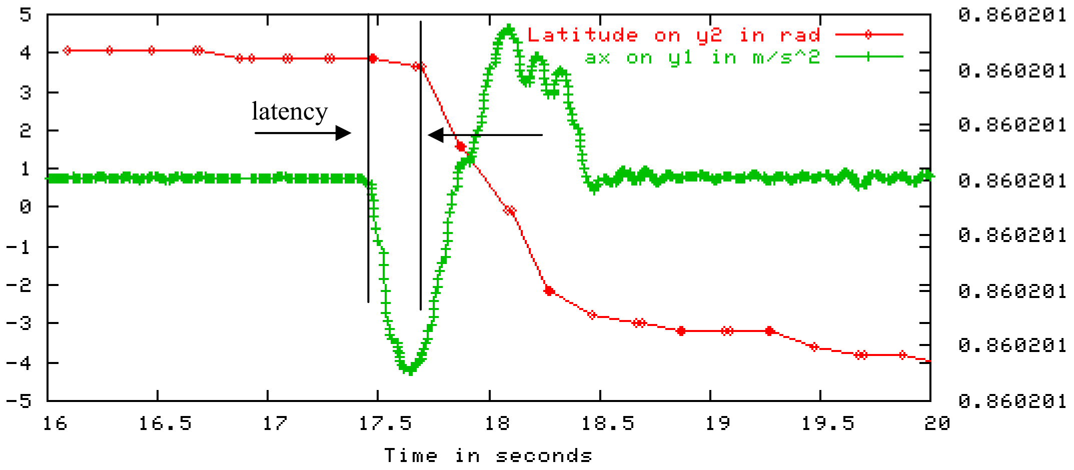


Figure 9: Response time difference for GPS and inertial sensor

[7] This information come from email change with Steve Lawrence from CSI-Wireless

8 Software Implementation

8.1 Onboard Software Configuration

The helicopter onboard computer was running a distribution of Linux operating system called "White Dwarf Linux"[8], which was specifically designed for the Jumptec PC104 board that we used. This operating system, however, did not have real time scheduling capability. Increasingly we realized this was a significant deficiency as we had trouble to utilize CPU cycle effectively.

8.2 Fail Safe Measures

The Kalman filter could reach numerical instability if the error was too large (such as when GPS was permanently lost and position error increased without bound.) or when the linearization was not close enough to the actual states. During takeoff or landing, the low frequency vibration was large and cause sensors to report unreasonable values. Since the linearized state matrix contained corrupted gyro measurement values, Kalman filter would likely diverge. Therefore, fail safe measures were needed in improve the robustness of the navigation system.

8.2.1 Measures Taken to Guard Against Filter Instability:

The health of the filter was best monitored through the position error covariance term. Basically, if the predicted value were close enough to the observation values, then the error covariance terms were held in check. If not, then the filter health could quickly deteriorate. Resetting the filter would reinitialize the position and orientation values through the gravity vector and GPS observation. If the trace of error covariance matrix P became either too big or becomes negative, then the filter went into reset.

What would happen if GPS reception was permanently lost? The Kalman filter implemented had two modes: Thirteen states and Seven states. The seven state filter was orientation only and used accelerometer derived roll and pitch angle for Kalman update. The thirteen state tracked all the states and used only GPS for error correction. Usually, the integration error for the orientation was held in check by GPS update, but if GPS was lost, then direct roll and pitch update from accelerometer would be a good choice. There was a timer which reset to zero whenever new GPS measurement arrived. If the timer value was too large, then the filter would reset to be a seven state filter because position data could not be observed.

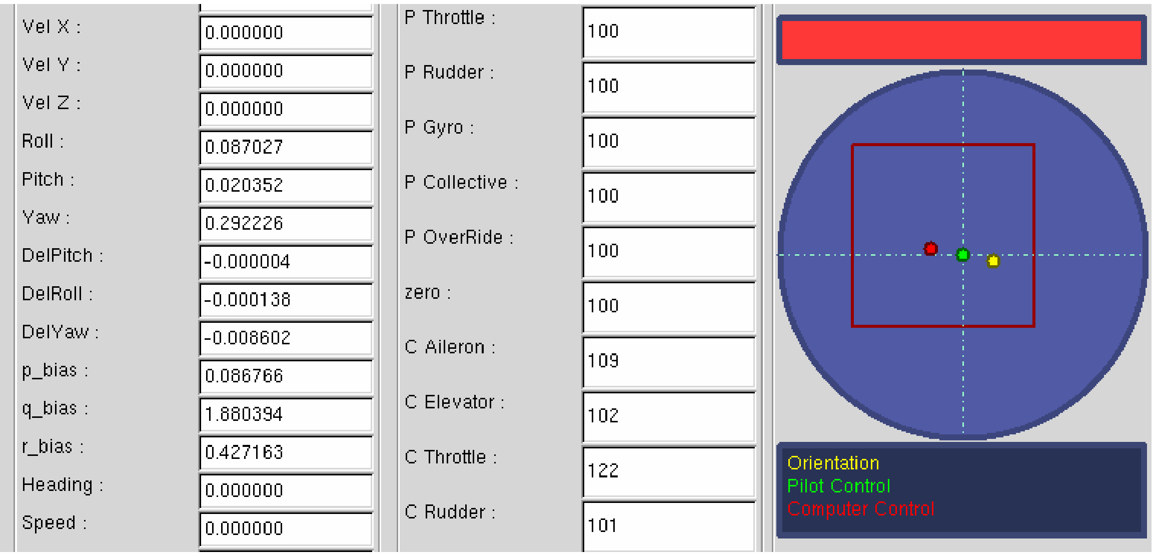
Additionally, the filter was coded to reset when the pilot flips a switch on his RC transmitter after the helicopter takes off. Doing so would reinitialize the filter so that the measurements would not be contaminated with vibration.

[8] You can find this flavor of Linux at http://www.whitedwarflinux.org/. At the time of this thesis release, the current version is 2.1.0

8.3 Kalman Filter Software Flow Chart

Appendix E contains the flow chart of Kalman filter implementation. Once the helicopter controller called the Kalman compute state function, it went through the logic shown on this flow chart. This chart only dealt with filter normal operation and error detection. Thus the details of filter initialization and reset are omitted. For example, the averaging of the sensor bias values during initialization is not shown.

8.4 Real Time Telemetry Monitoring



Vel X :	0.000000	P Throttle :	100
Vel Y :	0.000000	P Rudder :	100
Vel Z :	0.000000	P Gyro :	100
Roll :	0.087027	P Collective :	100
Pitch :	0.020352	P OverRide :	100
Yaw :	0.292226	zero :	100
DelPitch :	-0.000004	C Aileron :	109
DelRoll :	-0.000138	C Elevator :	102
DelYaw :	-0.008602	C Throttle :	122
p_bias :	0.086766	C Rudder :	101
q_bias :	1.880394		
r_bias :	0.427163		
Heading :	0.000000		
Speed :	0.000000		

Figure 10: Real Time Monitoring Software Screen

The hardware or software on board the helicopter could have glitches. For added security, there was a wireless link between the helicopter and a laptop. We used the real time telemetry transmitted via this link to monitor orientation and position. Instead of looking at numbers, which was less intuitive, we used a circle display with position of the yellow dot representing the roll and pitch estimate of the helicopter (Figure 10). The more the helicopter rolled or pitched, the further away the yellow dot would be from the center of the circle display. Should the estimate ever be grossly erroneous, the RC pilot would be asked to land the helicopter immediately.

By the same token, controller was monitored continuously as the red dot. The further away the red dot was from the center, the larger servo movement the computer commended. The red square inside the circle display was the software limit for the computer servo value. If the computer output was not within a reasonable range, the test flight would be stopped.

9 Results

Measurement of accuracy is very difficult to come by because the truth value of the helicopter state is very difficult to obtain. However, this thesis work has enabled the ARG helicopter to hover autonomously using both a PID controller (Michael Mierau, 2004) and a neural network controller (Shahin Roboubi, 2004). According to our test pilot, Marc Alfonso, precise angle control from helicopter controller (as precise a one degree of error margin) is required to hover a helicopter properly.

The goals of this thesis:

1. Proper gyro bias tracking
2. Stable and reasonably accurate orientation estimate
3. Improved position estimate from 5Hz to 150Hz by integrating accelerometer
4. Enable controller algorithm to fly the helicopter in the IARC competition

9.1 Gyro Bias Tracking

Figure 11 demonstrates the Kalman filter tracking the gyro bias on the pitch gyro. The changes in gyro measurement between 67 second to 100 second was actual movement. For the rest of the time, the helicopter was stationary thus the change in gyro measurement was caused by bias change. Thus this plot shows that the filter is tracking the bias properly.

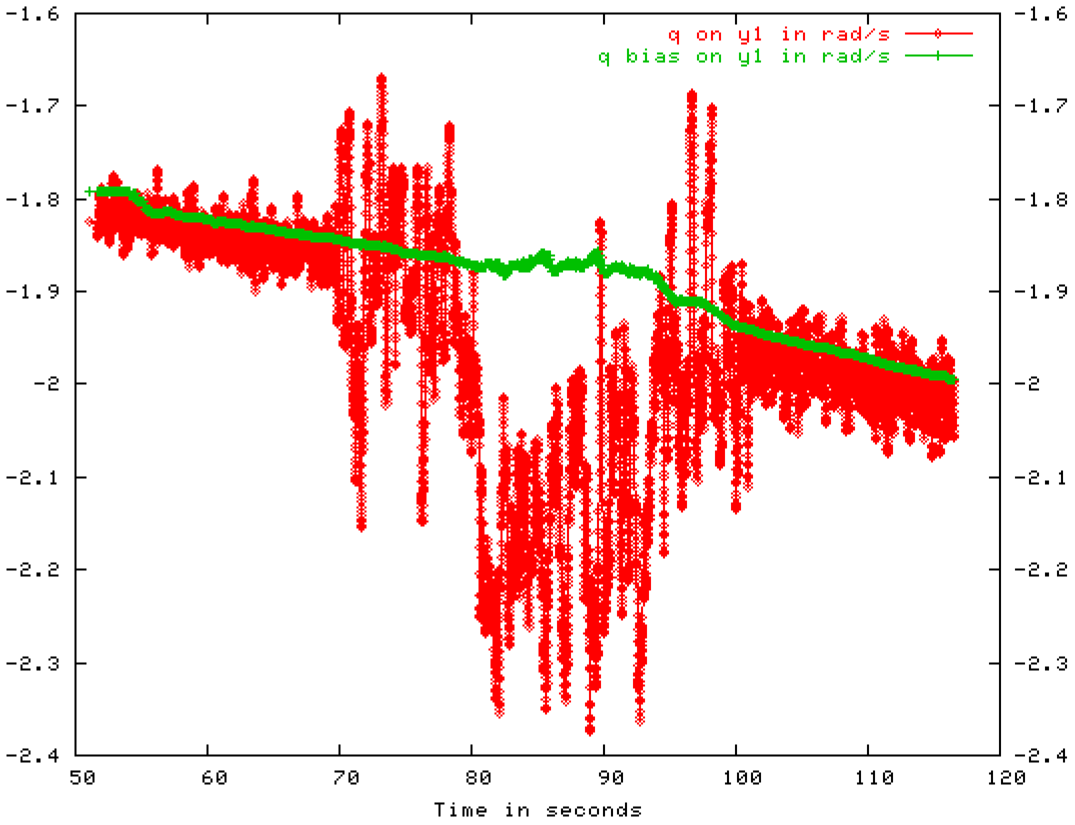


Figure 11: Raw gyro output and the estimated gyro bias with respect to time.

9.2 Orientation Tracking

Figure 12 shows the filter orientation output during a test flight. During that flight, helicopter was under computer control with the exception of collective (height) control. The wind was strong but the computer controller (running neural network algorithm) was able to hold the horizontal position under five by five meter square. The overall flight lasted about 200 seconds.

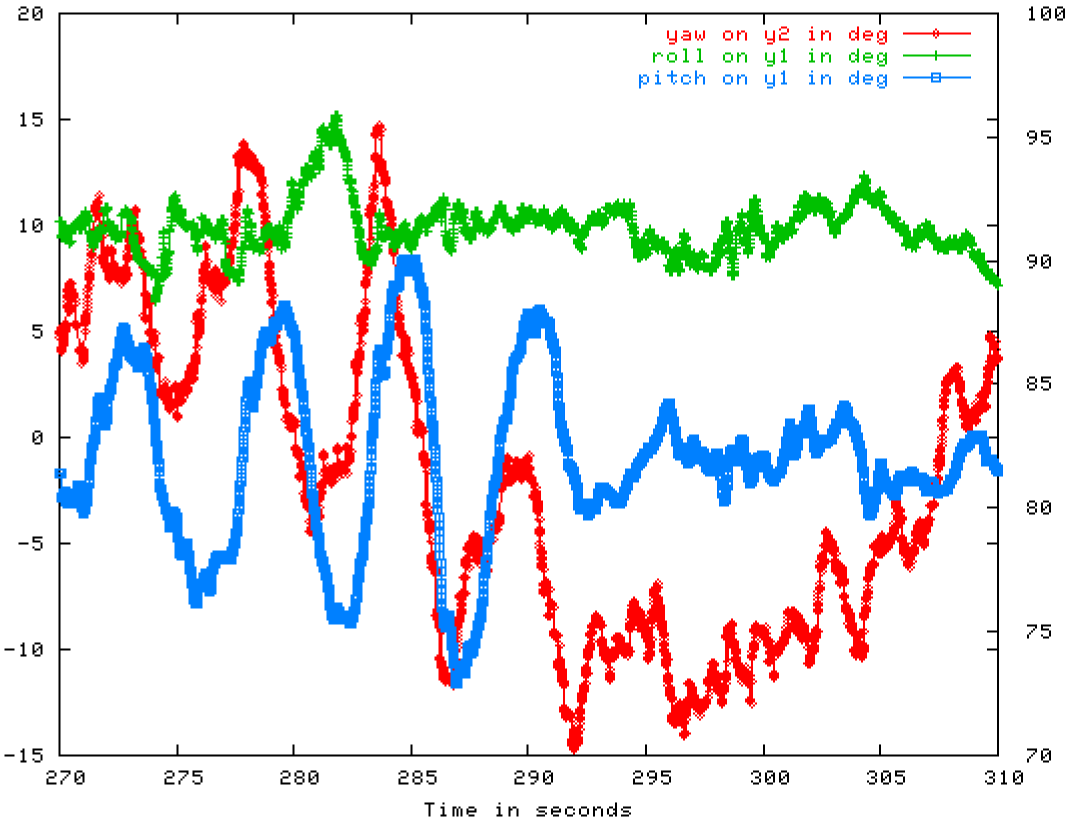


Figure 12: Orientation estimate computed onboard the helicopter during test flight.

9.2.1 Yaw Tracking Under Aggressive Tail Spin

The ARG model helicopter because of its weight, was not capable of performing aggressive maneuvers in roll and pitch axis. However, the pilot once tried increase the rudder servo output to cause strong tail spin. In Figure 13, a strong rudder input for 10 seconds caused the yaw to undergo six 180 degree turns. The filter remained stable on orientation and position under this condition.

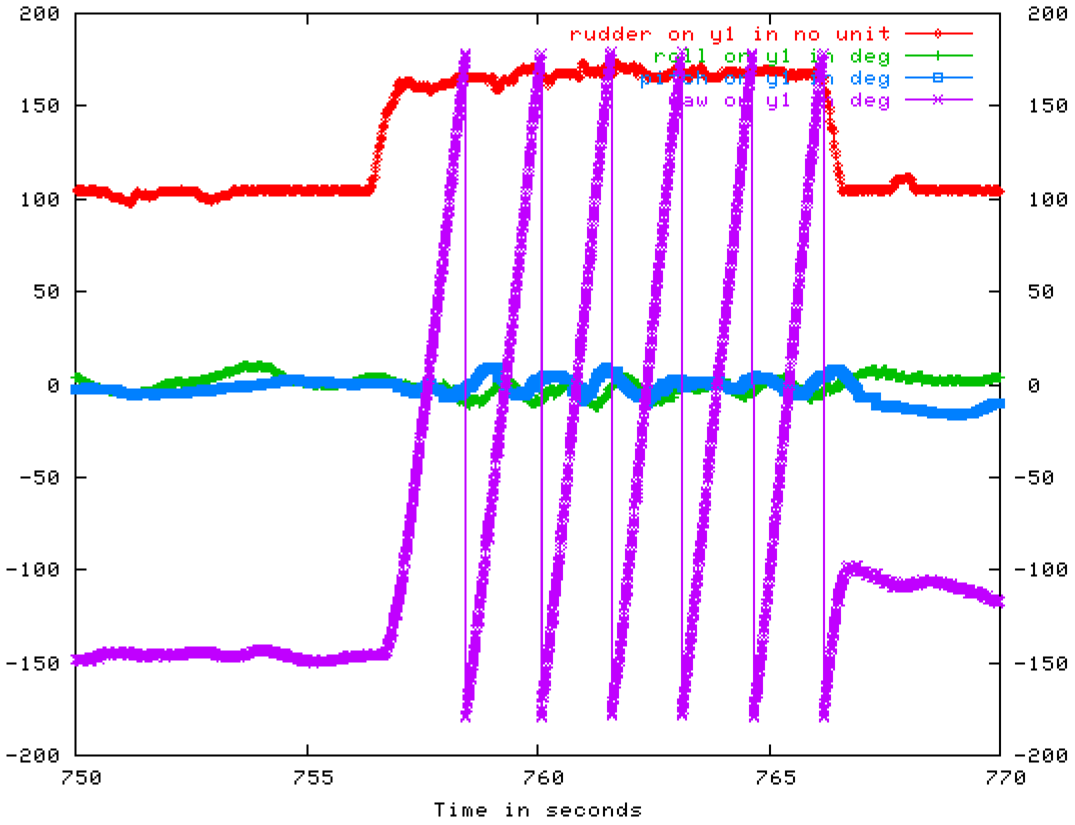


Figure 13: Orientation estimate during aggressive tail spin.

9.3 Position Estimation

The smoothness of the position was a good indicator of the filter performance because if the orientation were not right, the gravity force would be mistaken as part of the linear acceleration, thus forced the integrated position estimate to be off. When the GPS position measurement came in, the position would be corrected and thus made the position line to look jagged.

Figure 14 and Figure 17 are from the same flight as Figure 12. Figure 14 shows the position estimate from the filter output as compared to the GPS position. There is an offset between them because the GPS was on the tail while the filter estimated the rotor head position. The line was smooth and the position estimate was provided at 150 Hz compared to 5Hz from the GPS. Figure 17 is the XY plot of the flight.

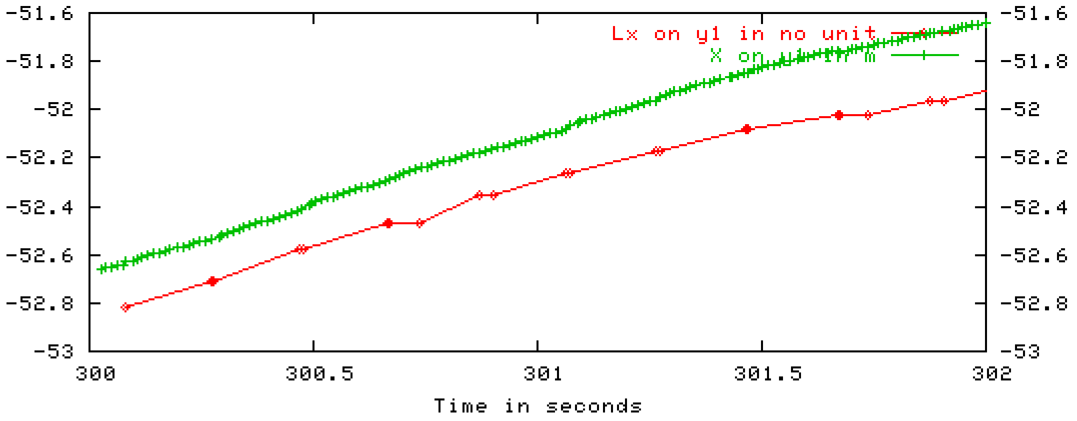


Figure 14: Position with respect to time.

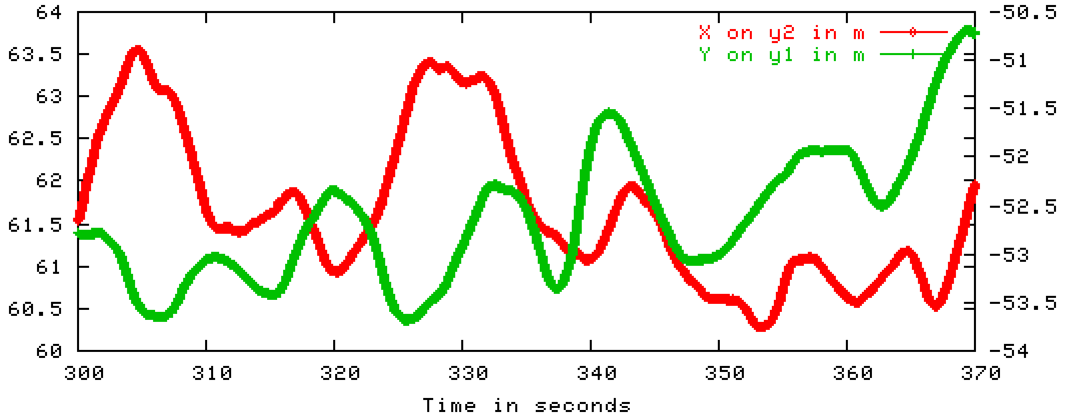


Figure 15: position X, Y, with respect to time during a flight

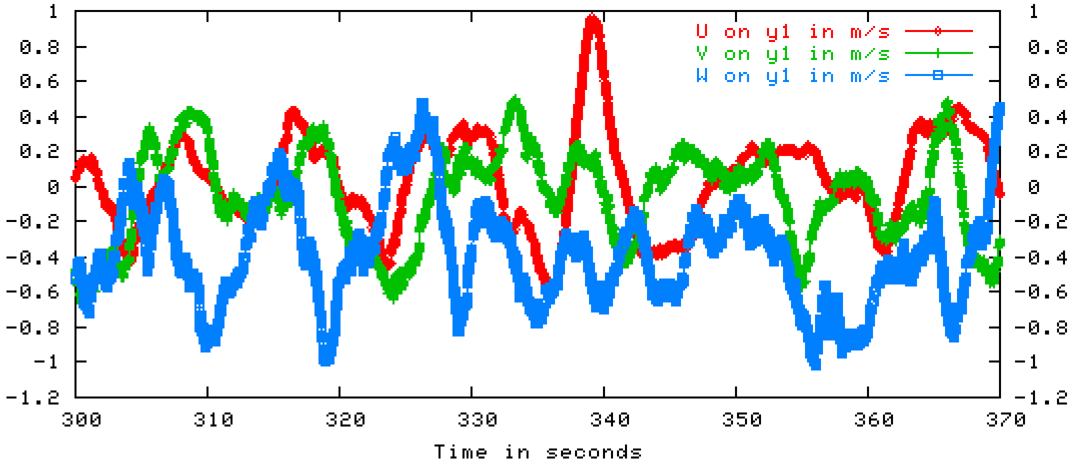


Figure 16: Velocity with respect to body axis during a flight

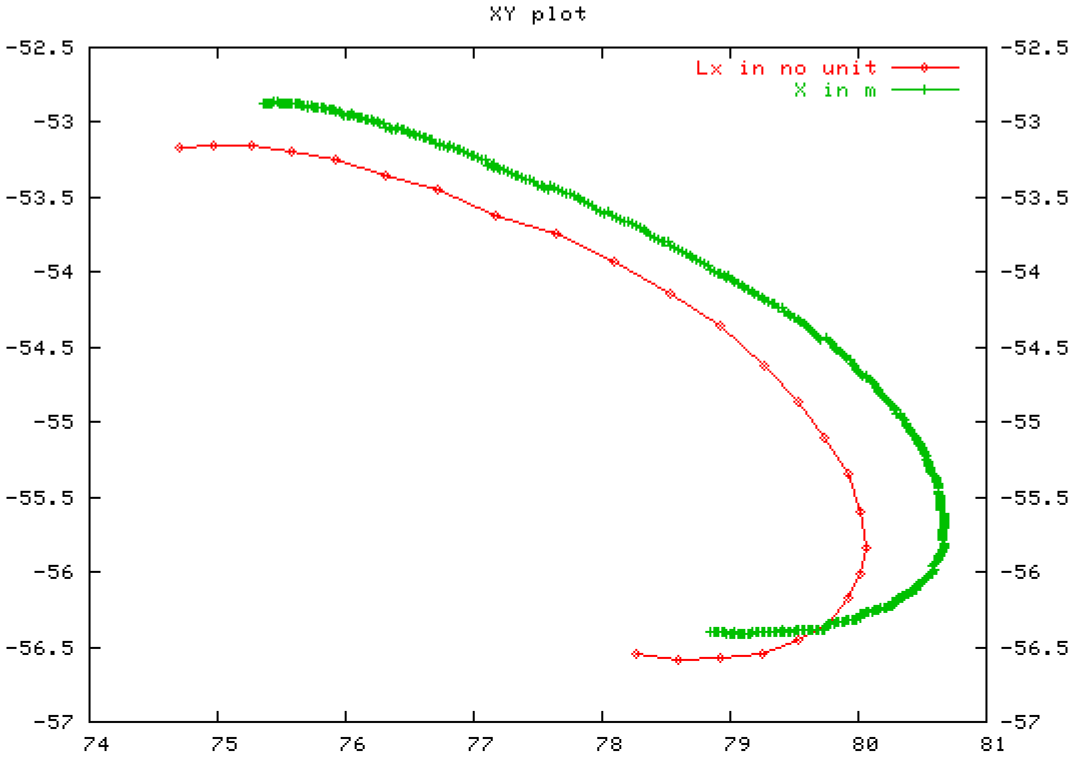


Figure 17: XY plot for position estimate computed during flight

9.4 Loss of GPS signal

The provision on the Kalman filter was that even if we lost satellite coverage during flight, the filter would still be able to estimate orientation (albeit at a lower performance).

Because we always conducted flight tests at places that has large empty space (safety reason) and our flights are limited by fuel (< 15mins), we never had problem involved in low satellite count or poor reception. However, we did encounter hardware problems that caused our serial port to fail for 10 seconds during flight.

Figure 18 contains the roll and pitch estimate during the loss of GPS measurement. The red like is the UTC time generated by the GPS receiver. When the UTC time stopped incrementing, no new position correction was received. In Figure 18, the computer stopped receiving GPS data a little bit past 292 second, timeout warning happened 1 second after and the filter reseted itself as an orientation only filter.

During the transition, the model helicopter under computer control still maintains a stable hover even through no more position estimate became available.

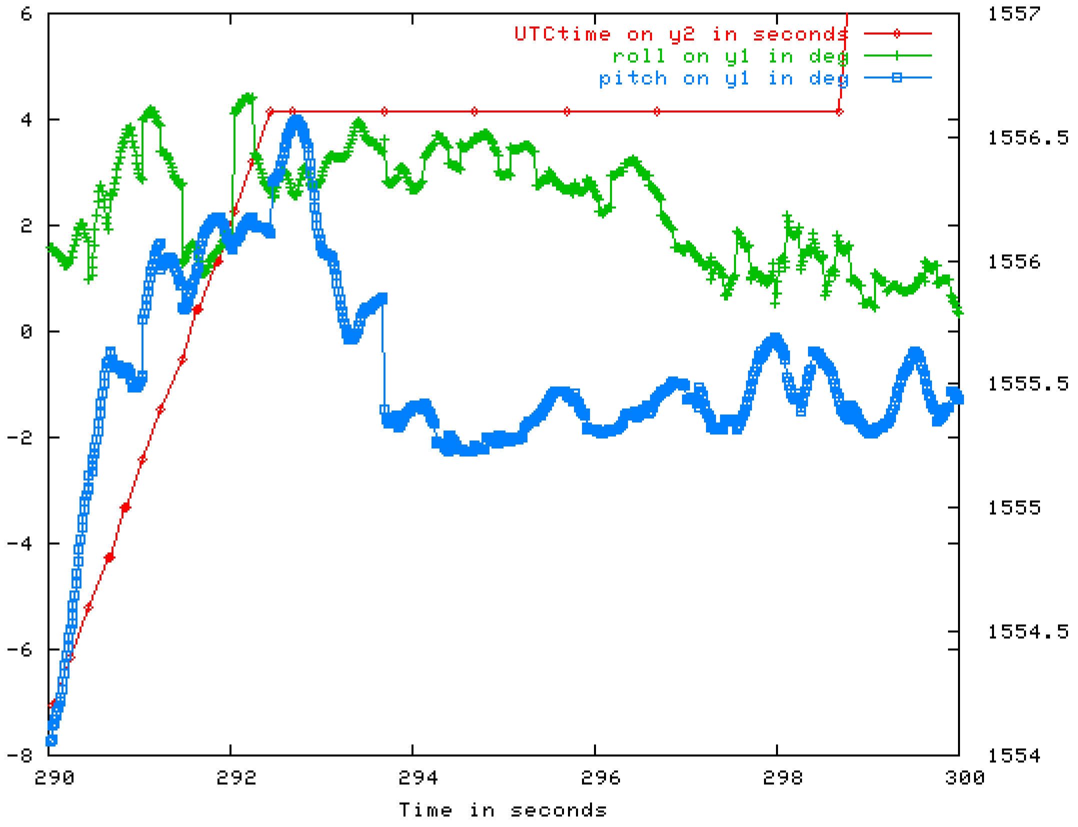


Figure 18: Roll and pitch angle estimate with respect to time during loss of GPS signal.

9.5 IARC Competition

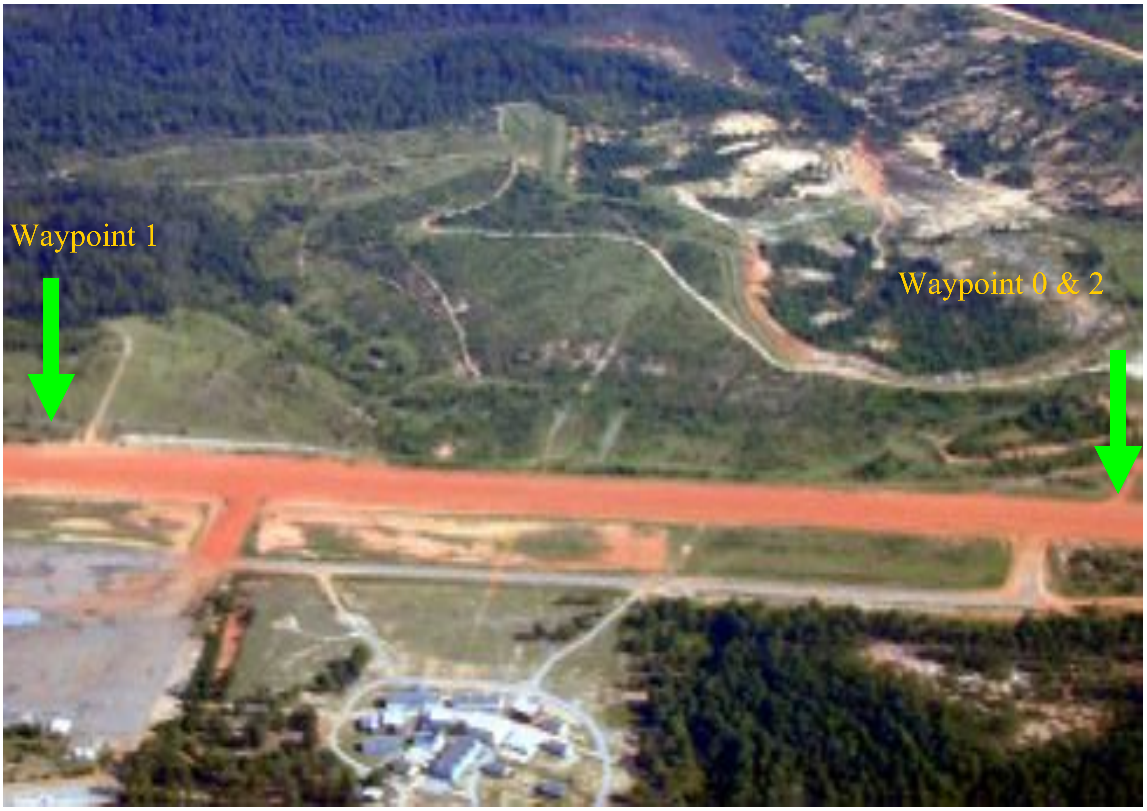


Figure 19: IARC Venue- U.S. Army Soldier Battle Lab McKenna Urban Operations Site at Fort Benning, Georgia

During the 2004 International Aerial Robotics Competition, the SFU team demonstrated autonomous way point flying in Fort Benning, Georgia. There were two demonstrations. Our first demonstration was a square waypoint pattern in which the waypoints are less then 100 meter apart. We've replicated such demonstration back in Vancouver so a successful result was more or less expected. The second demonstration was a long distance flight totaling two kilometer between the ends of the gravel road. The flight went from waypoint zero to waypoint two in Figure 19. The second flight was much more significant because we could not attempt such a feat due to the flying field length limitation back in Vancouver. We wanted to know if our helicopter could achieve stability in a long flight.

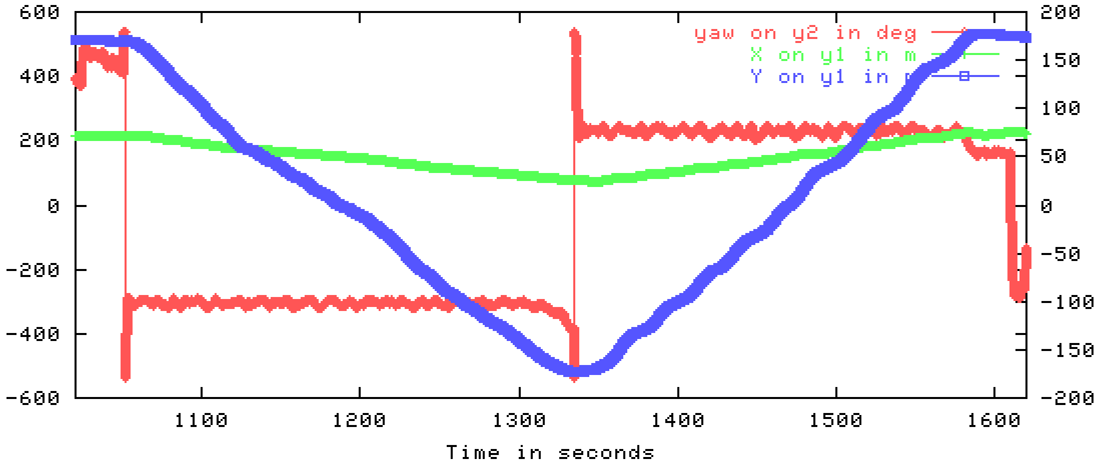


Figure 20: Competition Long Distance Flight Demonstration (Position & Yaw)

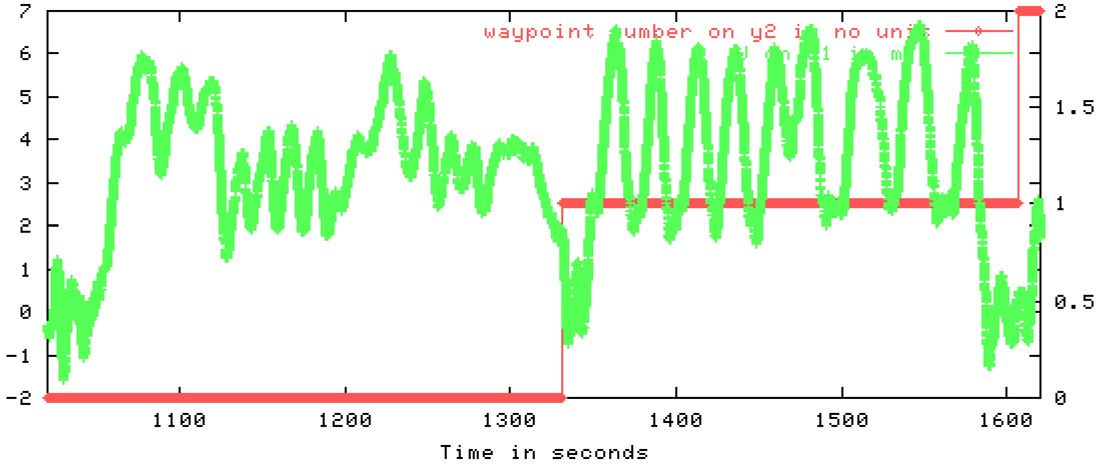


Figure 21: Competition Long Distance Flight Demonstration (Forward Velocity & Waypoint Numbers)

The result showed that our estimator/controller loop worked properly. Figure 20 shows helicopter followed a straight line between the two waypoints and the helicopter turned 180 degree at the end of waypoint one to head toward waypoint two. Figure 21 shows the forward velocity of the helicopter under computer control. The oscillation in speed was caused by controller detecting deviation from the course and slowing down the helicopter to correct the heading.

10 Conclusion

This thesis discussed the implementation of a navigation system for a model helicopter. By combining the advantages of inertial sensors and Global Positioning System, accurate and fast tracking of the helicopter position and orientation was achieved.

This report started with the formulation of important equations and then went on the describe the practical issue that needed to be addressed during filter implementation. Kalman state propagation and observation equations for the system was discussed first. Then gyro bias tracking was done through random walk and covariance determination was done through simulations from the flight test data. Vibration isolation was implemented through hardware mount and digital filter and tested successfully during flight. Latency compensation for the GPS was done through buffering and using pulse per second pin from the receiver.

Due to the limitation of the onboard computing power, sensor modeling is kept to a minimal. Only the gyro bias is tracked though a simple random walk function. If more power computer and larger memory is available in the future, a more complicated filter can be used to compensate for sensor nonlinearity, biases, and scaling error.

Adding more sensors on board will also improve the estimate. Currently the height is not as accurate compared to XY position, but a ultrasound or a altimeter can change that. Vision processing can be used to recognize terrain and distinguish targets, if a camera and a frame grabber board are onboard. All these information can be added into Kalman filter as measurement updates.

Another important issue yet to be explored by this thesis is the comparison between true and estimated values. The performance of the system is only measured by the fact that it successfully aide the autopilot to control the helicopter. If the system is observable, then we can intentionally omit one particular measurement and create estimator so that it also estimate the measurement as one of its state. Then we can have a evaluation of filter's performance by comparing the sensor values and estimated states. However, to do that several Kalman filter implementation each with a different sensor measurement omitted would be required. In addition, depending on the sensor quality, the sensor measurement might not be even close to the true value. Should future funding be sufficient enough for the ARG to acquire additional high grade sensors, this approach to evaluate filter performance could be adopted.

Last but not least, there needs to be a better understanding about the conditions that can caused the filter to become unstable. Under computer control, because the dynamics was limited by software, the filter output had always been stable. On the other hand, our test pilot could make the helicopter do aggressive rolling, pitching, and coordinated turns. Under those conditions, the filter was far more likely to become unstable because the linearization could fail. Therefore, in the future it would be beneficial to have a study done for the filter's sensitivity to disturbance and ability to track aggressive maneuvers.

The successful result of ARG project puts SFU into a prestigious few universities (Carnegie Mellon, Georgia Tech, MIT, Southern Polytechnic) in the world that has demonstrated successful navigation and control a model helicopter. In addition, SFU's autonomous helicopter is the first operational Canadian team helicopter entry for the IARC competition. Hopefully, the work done on this thesis will be a foundation for the future SFU aerial robotic projects and help the next generation SFU UAVs soar into the sky.

References:

Brown, R.G. and Hwang, Y. C. 1997, *Introduction to random signals and applied Kalman filtering.* New York: Wiley & Sons

Gelb, A. et al. 1974, *Applied Optimal Estimation*,. Cambridge Massachusetts: The M.I.T. Press.

Gavrilets V., A. Shterenberg, M. Dahleh and E. Feron., "Avionics System for a Small Unmanned Helicopter Performing Aggressive Maneuvers", Presented at 2000 Digital Avionics Systems Conference

Jekeli, C. 2001, *Inertial Navigation System*, New York: Walter de Gruyter.

Kalman, R.E., 1960, "A New Approach to Linear Filtering and Prediction Problems", *ASME-Journal of Basic Engineering,* Vol. 82, 35-45

Mierau, M. July 2004, *Control System on an UAV*, B.ASc thesis, Simon Fraser Univ. Burnaby, BC, Canada

Roboubi, S. July 2004, *Comparison of Artificial Neural Networks and their Application*, B.ASc thesis, Simon Fraser Univ. Burnaby, BC, Canada

Rönnbäck, S. February 2000, *Development of a INS/GPS navigation loop for an UAV.* M.Eng. thesis, Luleå University of Technology, Luleå, Sweden

Appendix A: Frames and Transformation

Transformations

The transformations used extensively in this thesis are body to local frame, local to body frame, Geodetic to ECEF, ECEF to local frame. The conversion between local and body frame is needed for inertial sensor data, while the Geodetic to local frame is needed to convert GPS position.

Body (Helicopter) to Local Frame

The transformation of two frames requires two things to be transferred: orientation and position. The orientation matrix is a 3x3 matrix composed of body frame axis represented with respect to the local frame. The column vector is the dot product of the body frame axis with the three axis of the local frame (As long as both the body frame axis and the local frame axis is expressed in the same frame).

$$ {}^{L}_{B}R = DCM = \begin{bmatrix} \vdots & \vdots & \vdots \\ {}^{L}\hat{X}_B & {}^{L}\hat{Y}_B & {}^{L}\hat{Z}_B \\ \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_L & \hat{Y}_B \cdot \hat{X}_L & \hat{Z}_B \cdot \hat{X}_L \\ \hat{X}_B \cdot \hat{Y}_L & \hat{Y}_B \cdot \hat{Y}_L & \hat{Z}_B \cdot \hat{Y}_L \\ \hat{X}_B \cdot \hat{Z}_L & \hat{Y}_B \cdot \hat{Z}_L & \hat{Z}_B \cdot \hat{Z}_L \end{bmatrix} \qquad \text{eq. 49} $$

To transform a point in the body frame transformed into the local frame, first left multiply the orientation matrix and then add the offset between the body frame origin and the local frame origin.

$$ {}^{L}P = {}^{L}_{B}R\,{}^{B}P + {}^{L}P_{BORG} \qquad \text{eq. 50} $$

(Euler angle)

The 3x3 orientation matrix is not the most concise way of expressing orientation because the column vector needs to be orthogonal. People use Euler angle or Quaternion to represent orientation. Euler angle contains three angles: roll, pitch, and yaw.

Each Euler angle is associated with a rotation along one axis on the current frame. So the orientation matrix is the result of pre-multiplication of three rotation matrices.

$$
{}^{L}_{B}R(\phi,\theta,\psi) = R_Z(\phi)R_Y(\theta)R_X(\psi)
$$
$$
\Rightarrow \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix}
$$
$$
\Rightarrow \begin{bmatrix} \cos\theta\cos\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\varphi \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \cos\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix}
$$

eq. 51

Quaternion Angles serves as the same purpose as Euler angles. But the mathematics involved in its operations is very different. The quaternion to orientation matrix is defined by:

$$
{}^{L}_{B}R(\phi,\theta,\psi) = \begin{bmatrix} (e_0^2 + e_1^2 - e_2^2 - e_3^2) & 2(e_1e_2 - e_0e_3) & 2(e_1e_3 + e_0e_2) \\ 2(e_1e_2 + e_0e_3) & (e_0^2 - e_1^2 + e_2^2 - e_3^2) & 2(e_2e_3 - e_0e_1) \\ 2(e_1e_3 - e_0e_2) & 2(e_2e_3 + e_0e_1) & (e_0^2 - e_1^2 - e_2^2 + e_3^2) \end{bmatrix}
$$

eq. 52

Quaternion is also the representation of choice in the subsequent work of this thesis. It has two advantages over Euler angles:

1. It does not contain trigonometric calculations in calculating orientation matrix. Therefore it is more computationally efficient and symbolically concise, especially when the Jacobian matrix needs to be computed.

2. Unlike Euler angles, it will not run into singularity problems when the angles are nearing 90 degree.

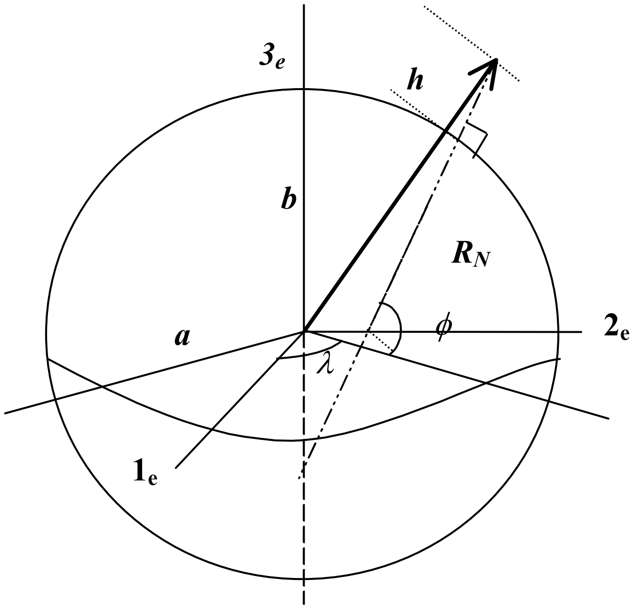


Figure 22: Parameters for Geodetic to ECEF transformation

Geodetic to ECEF

The Earth-approximating ellipsoid parameters we use are the ones adopted by US department of World Geodetic System 1984.

Semi-minor axis (b)
Semi-major axis (a) =6378137
Flattening (f) = (a-b)/a =1/298.257222101
R_N is the radius of curvature of the ellipsoid in the prime vertical plane.

$$R_N(\phi) = \frac{a^2}{\sqrt{a^2 \cos^2 \phi + b^2 \sin^2 \phi}} \qquad \text{eq. 53}$$

Since ECEF and Geodetic coordinate is concentric, i.e. both origins are at the center of the earth, the transformation is simply the following:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ECEF} = \begin{bmatrix} (R_N + h)\cos\phi\cos\lambda \\ (R_N + h)\cos\phi\sin\lambda \\ (\frac{a^2}{b^2} R_N + h)\sin\phi \end{bmatrix} \qquad \text{eq. 54}$$

ECEF to NED

To transfer the ECEF coordinate to the local coordinate, we need to do two plane rotations. First, a plane rotation about ECEF z axis to align the rotated y axis with the local plane east;

$$R_z(\lambda) = \begin{bmatrix} \cos\lambda & \sin\lambda & 0 \\ -\sin\lambda & \cos\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{eq. 55}$$

then we rotate along the new y axis to align the new z axis with the local plane “down” vector.

$$R_Y\left(-\phi - \frac{\pi}{2}\right) = \begin{bmatrix} \cos\left(-\phi - \frac{\pi}{2}\right) & 0 & \sin\left(-\phi - \frac{\pi}{2}\right) \\ 0 & 1 & 0 \\ -\sin\left(-\phi - \frac{\pi}{2}\right) & 0 & \cos\left(-\phi - \frac{\pi}{2}\right) \end{bmatrix} = \begin{bmatrix} -\sin\phi & 0 & \cos\phi \\ 0 & 1 & 0 \\ -\cos\phi & 0 & -\sin\phi \end{bmatrix} \qquad \text{eq. 56}$$

Multiplying the two together completes the orientation transformation.

$${}^{local}_{ECEF}R = R_Y\left(-\phi - \frac{\pi}{2}\right)R_z(\lambda) = \begin{bmatrix} -\sin\phi\cos\lambda & -\sin\phi\sin\lambda & \cos\phi \\ -\sin\lambda & \cos\lambda & 0 \\ -\cos\phi\cos\lambda & -\cos\phi\sin\lambda & -\sin\phi \end{bmatrix} \qquad \text{eq. 57}$$

Because the origin of the ECEF frame is not the same as the local frame, we will have to subtract the offset. The e-org vector is the vector from local frame origin pointing to the ECEF origin, and expressed in terms of the local frame.

$$\begin{bmatrix} a' \\ b' \\ c' \end{bmatrix}_{local} = R^{local}_{ECEF} \begin{bmatrix} a \\ b \\ c \end{bmatrix}_{ECEF} - \begin{bmatrix} x_{e-org} \\ y_{e-org} \\ z_{e-org} \end{bmatrix}_{local} \qquad \text{eq. 58}$$

Appendix B: Important Aspects of Kalman filter

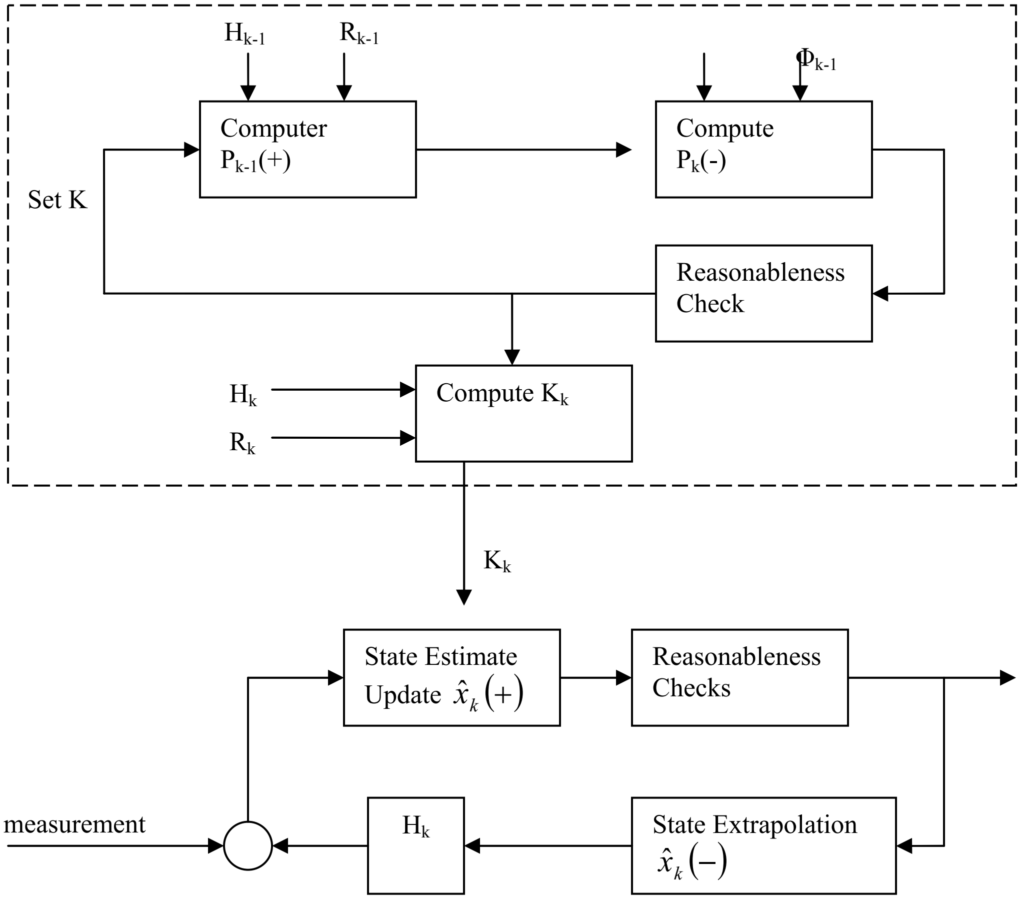


Figure 23: Information flow diagram of the Kalman filter

The type of observation is unlimited.
As long as the measurement matrix H_k and the covariance matrix R_k are modeled properly, Kalman filter will accept any measurements and use it to improve the overall state estimate. For our purpose, the measurement can be from different sensors such as altimeter, radar, ultrasound, or GPS.

The filter comes with its own accuracy estimate.
The covariance matrix P can be used as the indicator of the filter health and performance. After every measurement coming in, the P matrix should decrease. If the P matrix increase out of bound, it can mean problems with modeling or numeric processing.

Kalman gain computation requires a matrix inverse.

The optimum gain has a matrix inverse component. $K_k = P_k(-)H_k{}^T[H_k P_k(-)H_k{}^T + R_k]^{-1}$ This matrix inversion can cause program exceptions due to singularity and can add substantially to the computational load. It also makes the filter much more susceptible to numerical problems such as round-off or truncation errors. The health of the Kalman filter will have to be monitored continuously. Usually, the common practice is to look at the trace of the error covariance matrix P. If the trace value spikes, it can mean the filter health is seriously compromised and a filter has to be reseted.

Modeling problem: The Kalman filter we described above expects a discrete linear system with Gaussian white noises. For the case of ARG helicopter, the sensor dynamics are not linear; furthermore the some noises such as gyro bias are time correlated.

Appendix C: Vibration Isolation Counter Measures

Effect of Aliasing

Initially, we polled the micro-strain sensor at around 50Hz and used the result for a frequency analysis (see top plot of Figure 24). The FFT showed that there is high noise floor with no particular source of noise frequency. Compare that picture with bottom plot , which is under 150Hz sampling rate, we find that the noise floor and the noise peaks are clearly separated when you sample at higher rate

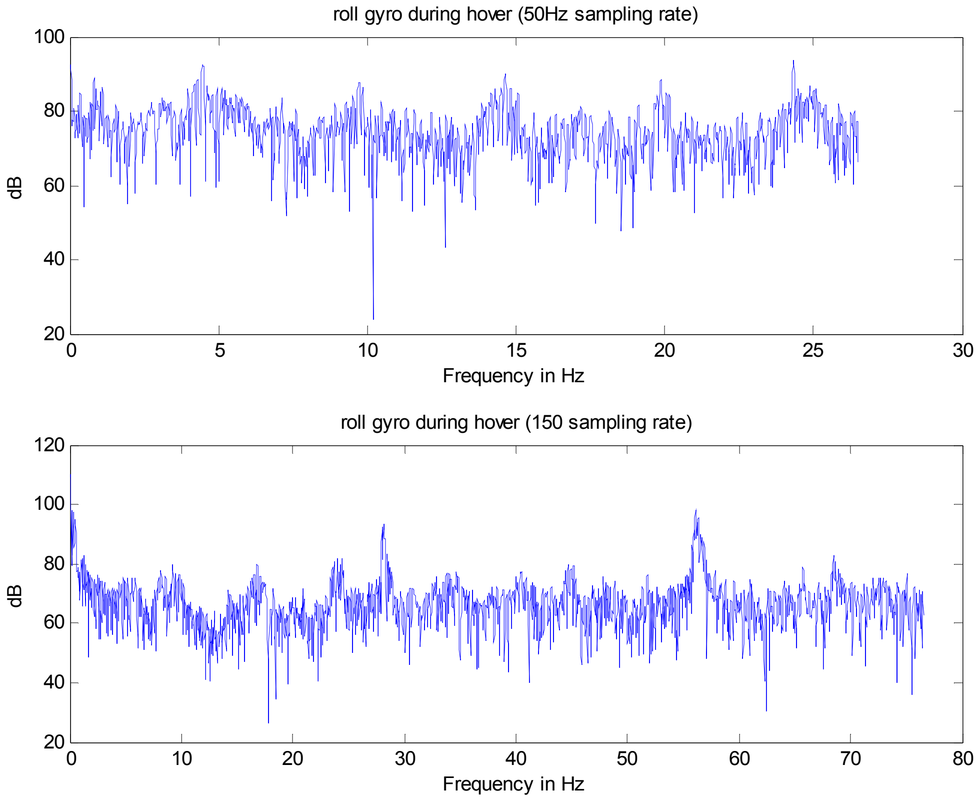


Figure 24: The comparison between gyro noise spectrum seen at different sampling rate.

Then we applied low pass digital filtering on both results. The 50Hz time samples shows magnitude varying from 1m/s^2 to -1.5m/s^2, which the 150Hz samples shows magnitude ranging from 0 to -0.4 m/s^2. That is 6 times of noise reduction just because of the difference of higher sampling frequency.

However, sampling at higher frequency in combination with digital filtering does not solve all our problems. The accelerometer has range of +/- 2G acceleration and the Gyro only has +/- 5.23 rad/s. Sensor saturation under intense vibration will not be prevented. In addition 150Hz is not enough to prevent aliasing caused by the engine noise at 250Hz. The solution is a combination of mechanical and digital filtering.

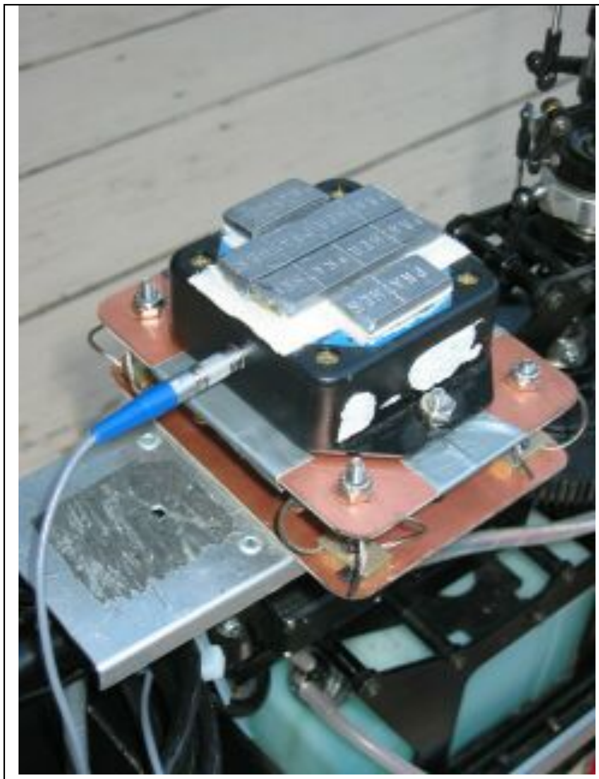


Figure 25: 3DM-G and vibration isolation mount (isometric view)

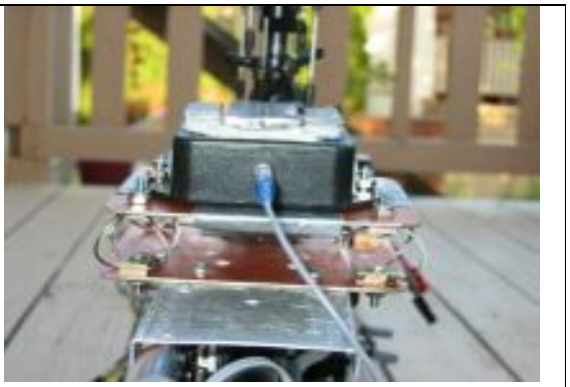


Figure 26: 3DM-G and vibration isolation mount (front view)

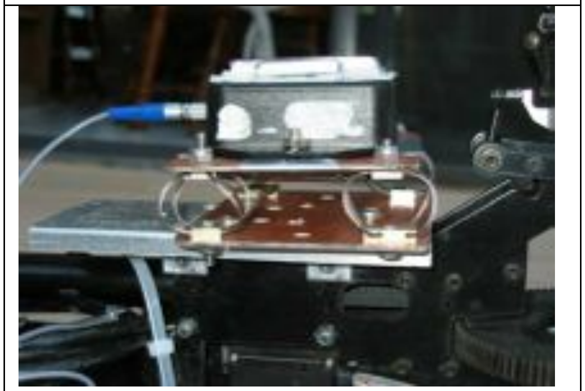


Figure 27: 3DM-G and vibration isolation mount (side view)

Isolation Mount

Initially, we start with elastomer isolators found in CD-ROM drives and used them to create a vibration mount. They worked but they were fragile and very difficult to come by, not to mention the high resonant amplitude. Later we brought wire-rope isolators from Enidine. The wire rope isolators were expansive (18 Cdn a piece) but they were durable and has a low resonant amplification factor. The silver metal on top of the sensor is lead. We used lead strips to add weight to the 3dm sensor because higher inertia means less movement during vibration and higher damping on the elastomer isolators. Lead is the ideal choice because it is cheap and has a high mass density of 4lb/inch^3. The added weight of lead increase the 3dm-G sensor weight from 100 gram to 300 gram. Experimental result shows that without sufficient weight, the vibration isolators is not effective in tuning out vibration.

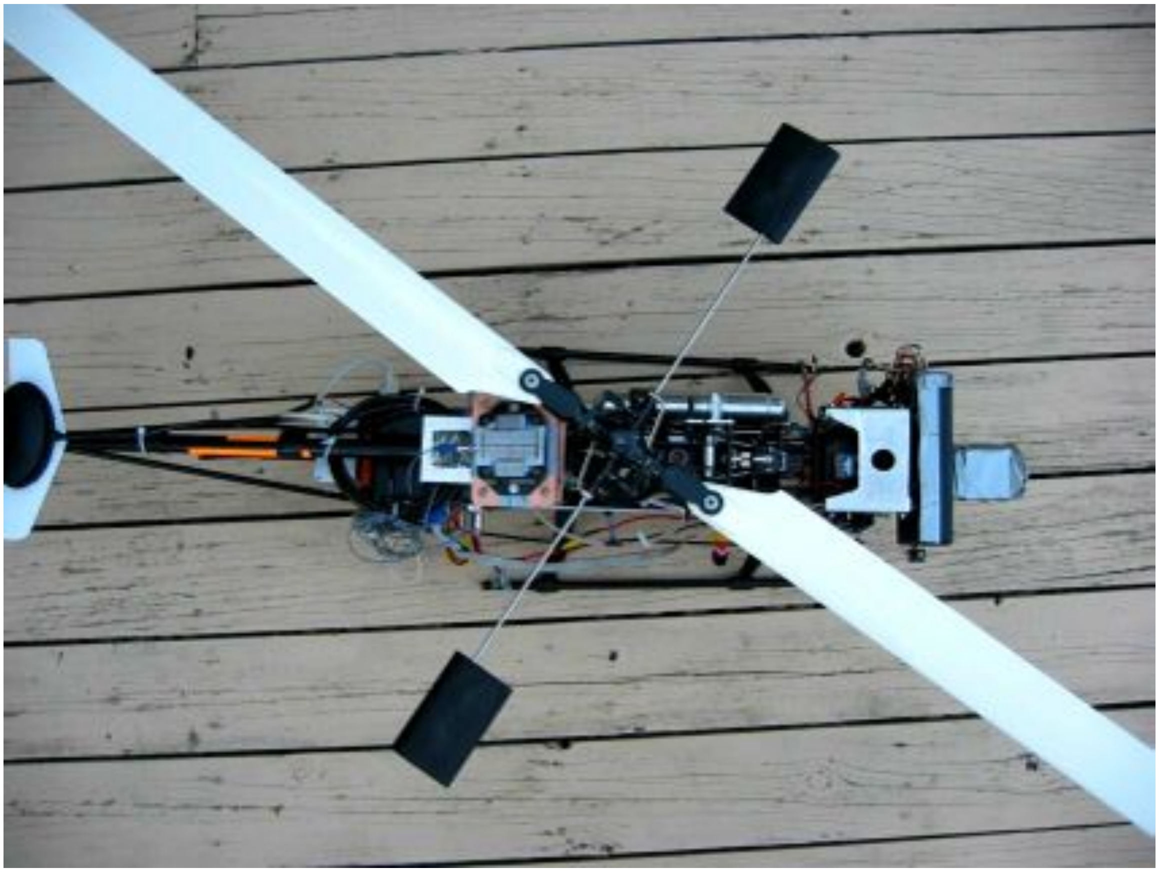


Figure 28: Location of the 3dm-G inertial unit on the helicopter

The 3dm-G IMU was placed near the rotor head of the helicopter. The benefit of this location was that it reduced vibration since it was close from the axis of helicopter rotation.
We measured the sensor output during hovering after placing the 3dm-g on the of the mount . The result is significant reduction of the noise in X and Y accelerometer. The Z accelerometer shows improvement but because the Z direction is constrained in the compression configuration, the mount is less effective.

Sensor	hard mount	with anti-vibration mount
X accelerometer	+/- 5 m/s^2	+/- 1 m/s^2
Y accelerometer	+/- 4 m/s^2	+/- 2 m/s^2
Z accelerometer	+/- 7 m/s^2	+/- 5 m/s^2
Roll gyro	+/- 43 deg/s	+/- 23 deg/s
Pitch gyro	+/- 34 deg/s	+/- 23 deg/s
Yaw gyro	+/- 14 deg/s	+/- 11 deg/s

Table 5: Sensor reading with and without vibration mount

Digital Filtering

Because the higher vibration has been reduced by hardware filtering and there is less aliasing, the software filtering on the sensor values can be much more effective. The low pass filter we picked is a 10th order IIR Butterworth filter with 10Hz cut off.

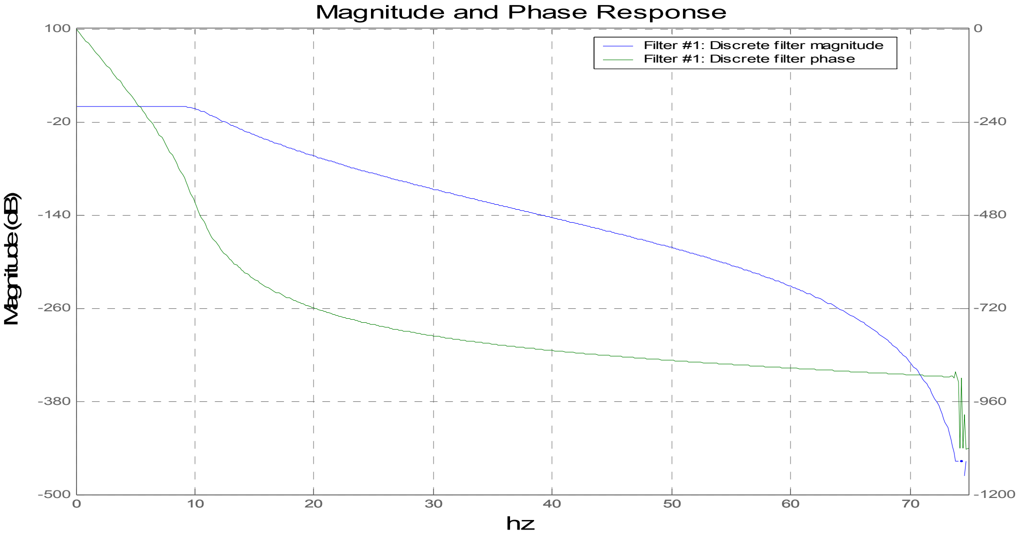


Figure 29: Magnitude and phase response of the low pass IIR filter.

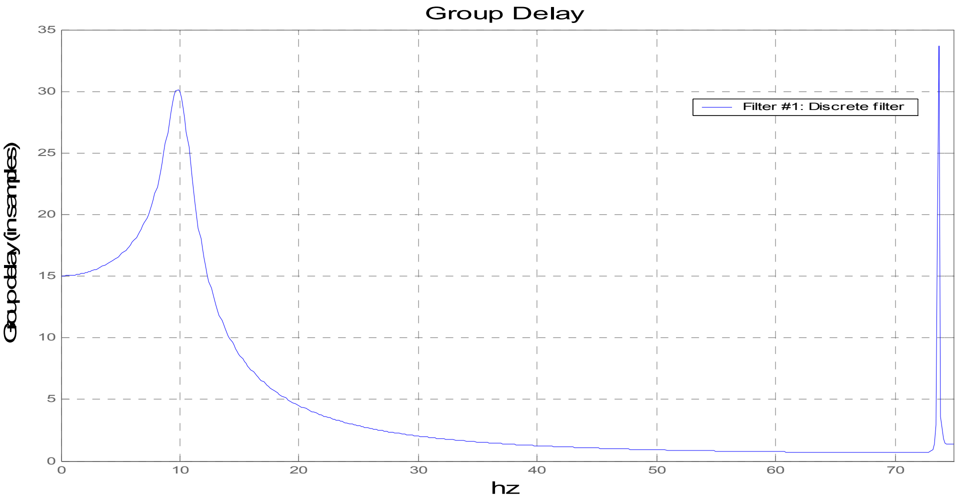


Figure 30: Group delay of the low pass IIR filter

We sacrificed latency to achieve steep cut off slope. The group delay peak at 10 Hz is 0.2 seconds. Even at 0 to 5 Hz, there is 0.1 second of delay. However, this delay is tolerable because the feedback loop of a human model helicopter pilot is slower then 0.5 second and the response is less then 5Hz. So if a human pilot can hold a model helicopter steady in the air, then our navigation system should be sufficiently quick.

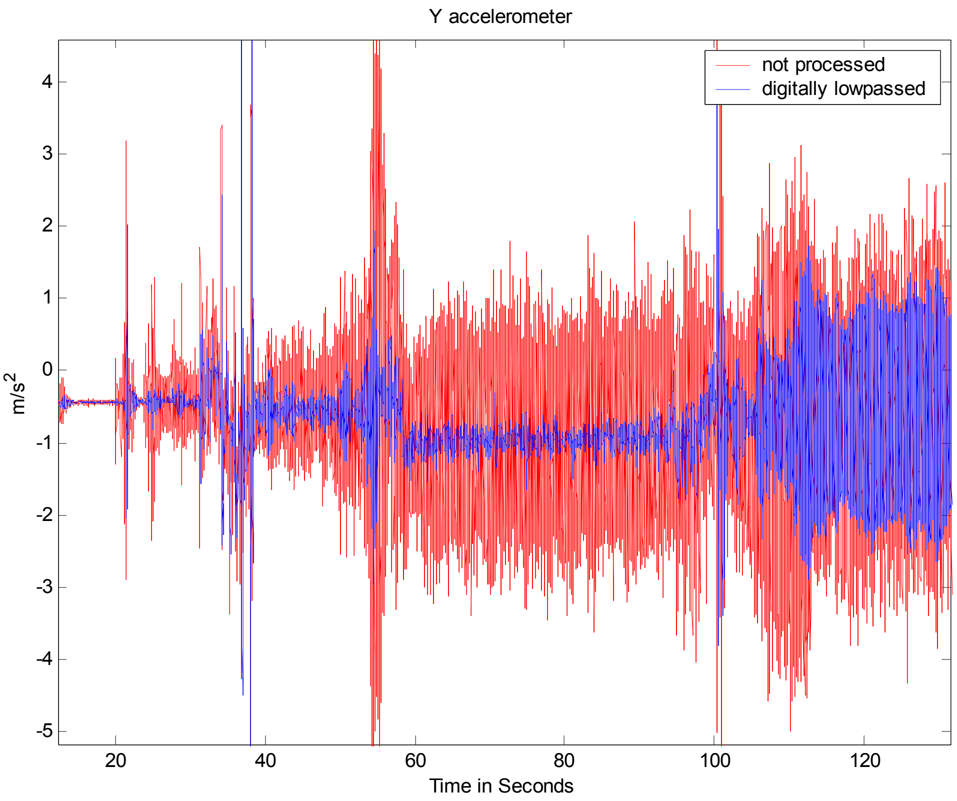


Figure 31: Comparison between digitally lowpass-ed and un-filtered flight data.

In Figure 31, between 60th ~90th seconds the helicopter was hovering. The filter attenuate out the +/-2 m/s^2 of vibration to +/- 0.5 m/s^2. After the helicopter has landed in 100th second, the engine and the rotor head was not shut off immediately. This causes a low frequency vibration because the helicopter landing strut bounced around the uneven surface of the air field. Our lowpass filtering should not filter those vibration out. In addition, the spikes at 70th to 100th second was caused by taking off and landing of the model helicopter.

Appendix D: Global Positioning System Error

GPS system is a radio-navigation system using 24 satellite and their ground stations. By measuring the delay of no less then four satellite signal from space to the ground, and triangulate the four distances, the GPS receiver can determine its location on the earth.

Since GPS system relies on timing, any additional delays caused by atmosphere condition or reflection will degrade the its performance. Therefore, the accuracy of a GPS system is related to how well the errors have been modeled and compensated for.

Multi-path: when GPS signal received was not directly from satellite but was bounced from another surface. The result would be a messy signal- first the direct one and the a bunch of delayed reflected signal.

Atmospheric: GPS signal can be slowed down significantly in two different layers of atmosphere: ionosphere and troposphere. Troposphere has high humidity that will refract GPS signal and cause distortion. Charged particle from the sun ionize the ionosphere and cause it to be electrically active. Troposphere delay can be easily modeled but the ionosphere electrical activity changes day to day due to Sun's solar cycle and thus is much harder to predict

Ephemeris : Satellite trajectory is closely monitored but it is not perfect. Therefore any deviation form its trajectory can cause timing to change. Orbit error is called Ephemeris errors.

Geometric Dilution of Position (GDOP): Errors from triangulation is related to the geometry of the satellite location. If the satellites GPS receiver tracks are close together, it would not be as good as if they are apart. Therefore good receivers will determine which satellites has lowest GDOP.

DGPS and WAAS correction

The D in DGPS stands for Differential. Differential correction requires a pair of receiver with one of them being stationary. Because the stationary receiver and remote receiver are close together, their signal should go through the same slice of atmosphere and their errors should be the same. If we have an accurate position of the stationary receiver, we can use it to compute all the timing error parameters and feed it to the remote receiver. Therefore, instead of using the timing to compute position, a known position is used to produce a timing reference and deduce the errors correction factor. This way, significant boost of GPS accuracy can be acquired.

If the two receivers becomes too far apart, then the errors can be de-correlated. The bigger problem would be the requirement of having two receivers and the need of maintaining a radio link between the two receivers. For the aerial robotics project, it increases cost and makes the model helicopter navigation dependent on the ground station, thus making the system less robust.

WAAS is the acronym for “wide area augmentation system”. It is a new (since year 2000) differential correction service developed by the United States Federal Aviation Administration (FAA). It is broadcasted at the same frequency as the GPS signal, so no additional antenna is required to receive such correction. Instead of one stationary receiver, the FAA provides a network of geostationary satellites, ground reference centers and processing centers. This effectively removes the problem of spatial de-correlation problem and the need for second receiver and radio link. In addition, WAAS transmits a timely map of ionosphere condition that can be used for compensating signal delay errors.

WAAS is not perfect either. First of all it is only available in North America. Secondly, it requires line of sight the same way GPS system does. Since there are currently only two WAAS satellites compared to twenty four GPS satellites, WAAS reception degrade noticeably when line of sight is lost. For British Columbia, WAAS satellites are low on horizon and the WAAS signal can sometimes be masked by terrain or buildings. Last but not least, WAAS is still relatively new and FAA does not guarantee the accuracy of WAAS. Users will have to assume its own risk.

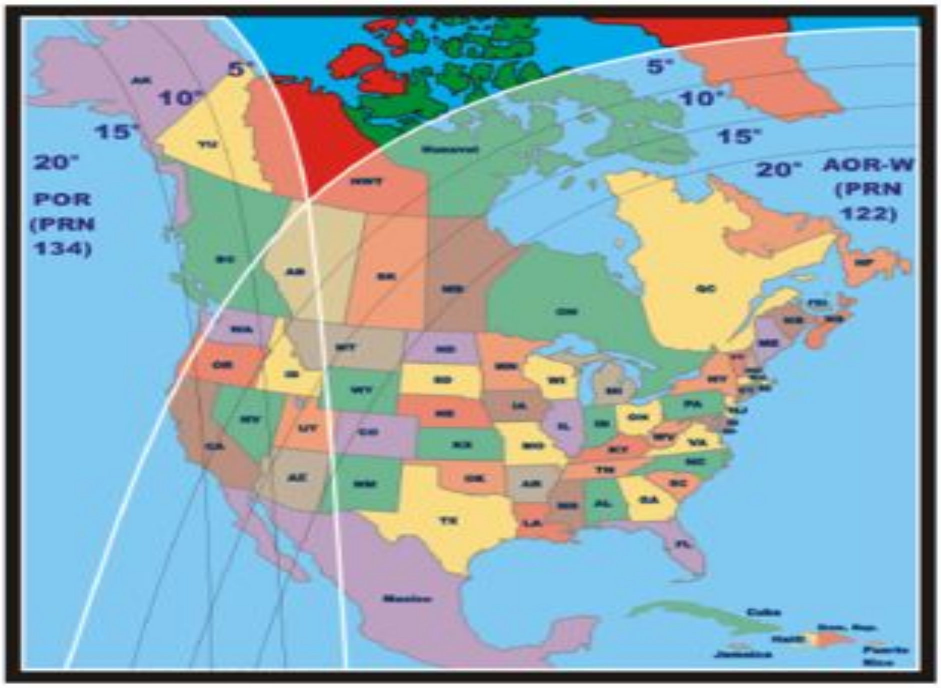


Figure 32: WAAS coverage in North America

The lower the horizon angles, the harder it is to receive WAAS signals. PRN 134 and PRN 122 are the names of WAAS satellites.

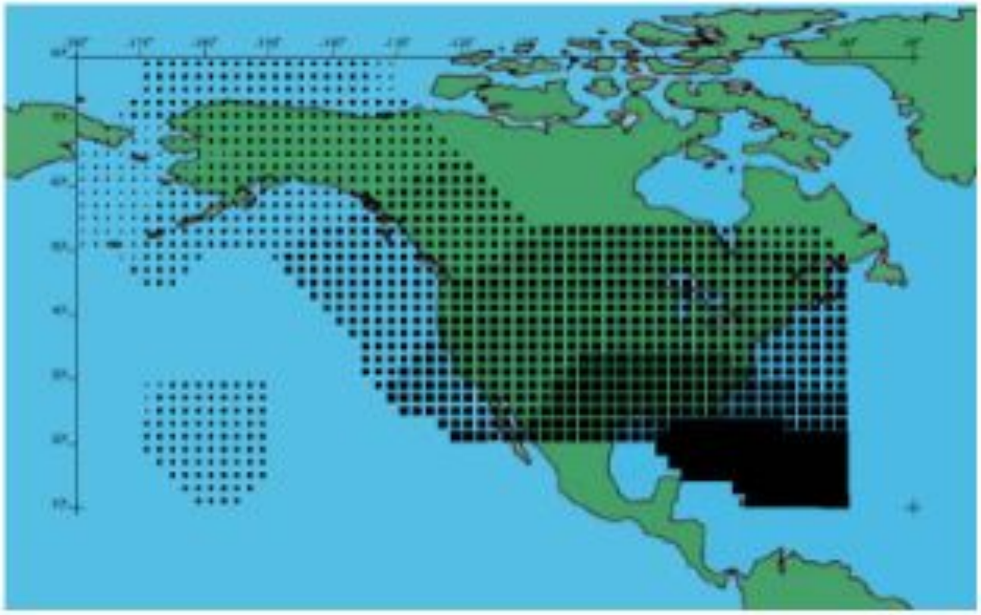


Figure 33: The availability of ionosphere map from WAAS

Timely correction for the ionosphere delay provided by WAAS can improve GPS accuracy.

CSI-wireless Seres

Our GPS (the CSI-wireless Seres Smart antenna) uses WAAS to achieve sub meter accuracy with out the need of a base station transmitting differential signals. It also comes with the proprietary "coast" technology that enables user to maintain differential accuracy even during loss of correction data for up to 30 minutes. This capability increases the robustness of our navigation system.

Appendix E: Flow Chart

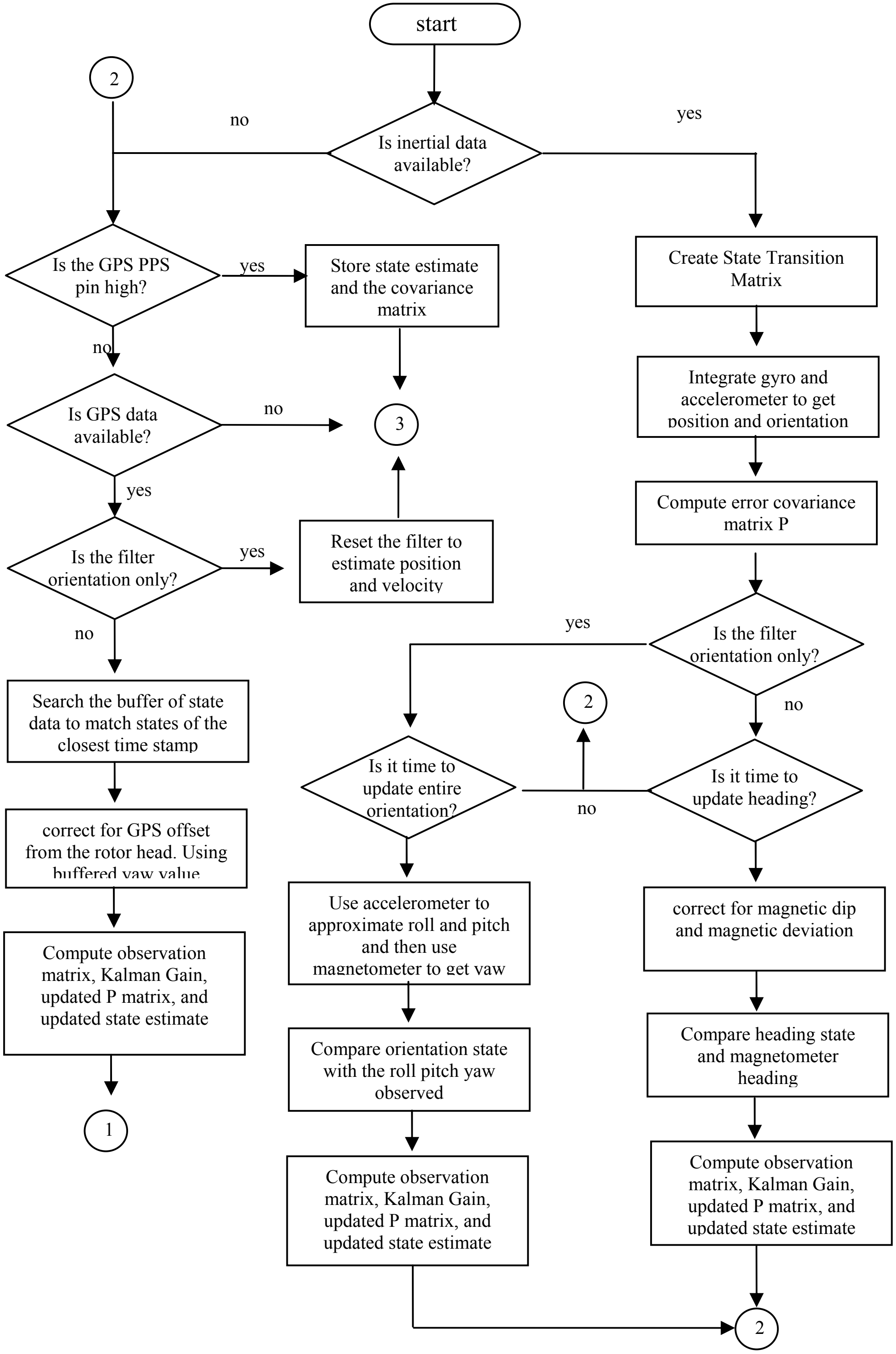

start
2
no
Is inertial data available?
yes
Is the GPS PPS pin high?
yes
Store state estimate and the covariance matrix
Create State Transition Matrix
no
Is GPS data available?
no
3
Integrate gyro and accelerometer to get position and orientation
yes
Compute error covariance matrix P
Is the filter orientation only?
yes
Reset the filter to estimate position and velocity
no
yes
Is the filter orientation only?
Search the buffer of state data to match states of the closest time stamp
2
no
Is it time to update entire orientation?
Is it time to update heading?
no
correct for GPS offset from the rotor head. Using buffered yaw value
Use accelerometer to approximate roll and pitch and then use magnetometer to get yaw
correct for magnetic dip and magnetic deviation
Compute observation matrix, Kalman Gain, updated P matrix, and updated state estimate
Compare orientation state with the roll pitch yaw observed
Compare heading state and magnetometer heading
1
Compute observation matrix, Kalman Gain, updated P matrix, and updated state estimate
Compute observation matrix, Kalman Gain, updated P matrix, and updated state estimate
2

Figure 34: Steps and the decisions made after measurements become available.

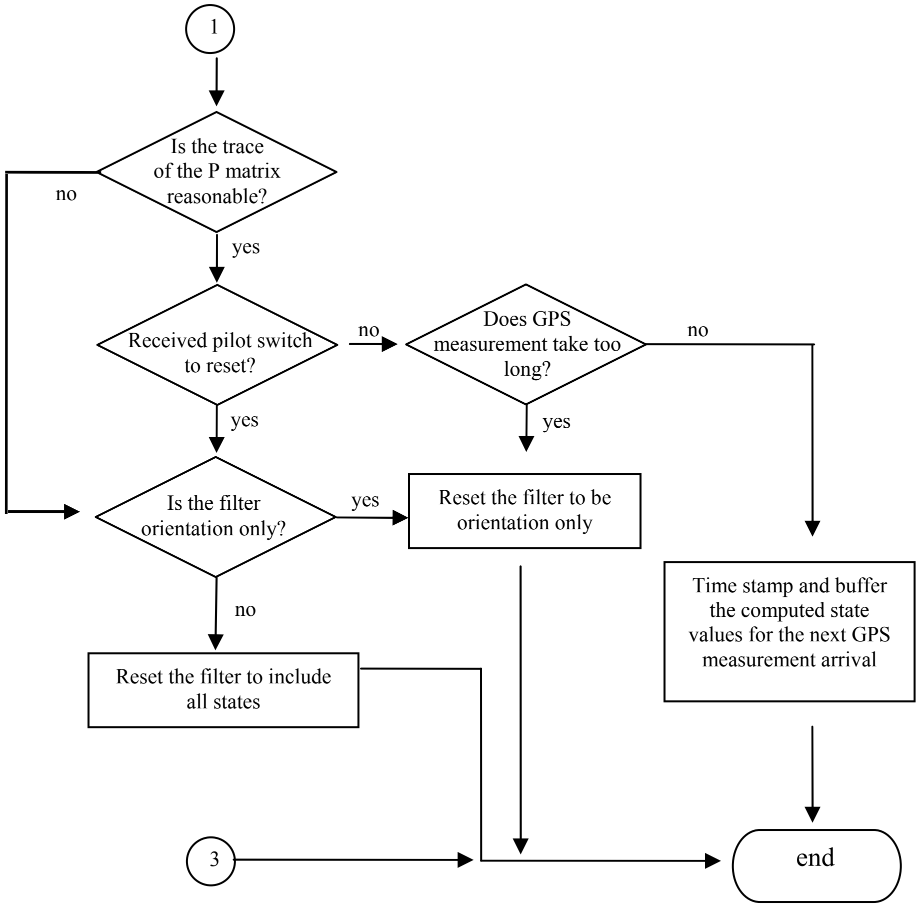

1
Is the trace of the P matrix reasonable?
no
yes
Received pilot switch to reset?
no
Does GPS measurement take too long?
no
yes
yes
Is the filter orientation only?
yes
Reset the filter to be orientation only
no
Time stamp and buffer the computed state values for the next GPS measurement arrival
Reset the filter to include all states
3
end

Figure 35: Decision process to reset filter