AIAA Infotech@Aerospace 2010
20 - 22 April 2010, Atlanta, Georgia

AIAA 2010-3509

The Design of Rapidly Reconfigurable Filters for Attitude and Position Determination

Renwick E. Curry,* Mariano Lizarraga, †
and Gabriel Hugh Elkaim ‡

University of California-Santa Cruz, Santa Cruz, CA 95060

Attitude and position estimation are essential components of any Unmanned Aerial Vehicle. Linear and Extended Kalman Filters as well as complementary filters have traditionally played a major role in these calculations. In order to successfully field these estimators, a design process requiring many iterations is required. We describe the design process using a new and flexible platform developed for rapid prototyping of UAV Guidance, Navigation, and Control algorithms. The evolution of the design begins with a hybrid model consisting of combined Extended Kalman and complementary filters and ends with complementary filters alone. The resulting complementary filters compare favorably with traditional fifteen-state Extended Kalman Filters previously reported in the literature and are thus attractive candidates where computational resources are limited. The filter results are shown for actual flight test data.

*Adjunct Professor, Computer Engineering Department, Senior Member.

†Research Fellow, Computer Engineering Department, Senior Member.

‡Associate Professor, Computer Engineering Department, Senior Member.

Copyright © 2010 by Renwick E. Curry. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

Nomenclature

a_i	Inertial acceleration) (m/s/s)
a_r	Ideal accelerometer reading, or specific force (m/s/s)
g	acceleration due to gravity (m/s/s)
V	Inertial velocity (m/s)
$\hat{u}_g$	Estimated direction of gravity
A_{sens}	Sensed acceleration (m/s/s)
A_{bias}	Accelerometer bias (m/s/s)
G_{sens}	Sensed angular rate (rad/s)
G_{bias}	Gyro bias (rad/s)
q	Quaternion from navigation to body coordinates
DCM	Direction cosine matrix from navigation to body coordinates
ψ, θ, ϕ	Euler angles from navigation to body coordinates (rad)
$\hat{\Omega}$	Angular input to quaternion
K_P	Loop gain (1/s)
K_Y	Gain on heading reference
K_I	Gain on gyro bias estimates (1/s/s)
ψ_{ref}	Unit heading reference vector
$\hat{P}(t)$	Estimated position (m)
$\hat{V}(t)$	Estimated velocity (m/s)
$P_{GPS}(t_k)$	Horizontal GPS position vector at time t_k (m)
$V_{GPS}(t_k)$	Horizontal GPS velocity vector at time t_k (m/s)
a_k	Estimated acceleration in $[t_{k-1}, t_k]$
α	Acceleration weight

I. Introduction

In the recent years there has been an exponential growth in the use of (and interest in) Unmanned Aerial Vehicles (UAV). This growth has been more notable in military operations, but surveillance applications such as observation of marine fisheries, forest fire monitoring, and border patrol have also seen a considerable growth.[1–3] Regardless of its application, the UAVs require accurate onboard determination of attitude and position. In recent years, the most cost-efficient solutions have involved low-cost inertial sensors coupled with external information sources such as GPS receivers and barometric sensors.[4,5] Traditionally, the Extended Kalman Filter (EKF) and its many variants (see References[6–10]), has been used as the *de facto* tool for integrating these information sources.[11–16] The use of the EKF has been so widely adopted that Crassidis,[17] in his survey of nonlinear attitude determination methods, calls the EKF *"the workhorse of real-time spacecraft attitude determination"*. In general, the advantages of the EKF are near-optimal estimation but requiring extensive computational resources, typically either a CPU with a floating point unit (FPU) or a fixed-point implementation. The implementation of the EKF also requires a detailed model of system dynamics, process noise and observation noise. In practice, these may be difficult to determine, or may have characteristics ill-suited to the model framework. Furthermore, instability and divergence of the EKF may occur if the initial conditions are far from the true state.[17]

Observers and Complementary Filters (CF) are suboptimal solutions to the position and attitude estimation problem, with the advantage that they require fewer assumptions about noise and system dynamics. In contrast to the EKF, CFs have lower computational demands, albeit sacrificing convergence time and response to initial conditions. The essence of the CF is to combine low frequency information from one source and high frequency information from another (see References[12,18] for a thorough discussion of Complementary Filters). Brown[18] also compares the two approaches for inertial navigation system (INS) with external updates. He points out the difficulty of determining noise models, and shows that the all-pass complementary constraint is only slightly suboptimal in high signal-to-noise environments. Provable stability is an important feature of any filter to avoid the divergence problems that can arise in the EKF. In Reference[19] the authors develop a nonlinear CF for estimating position and velocity of a rigid body with position and angular measurements, and prove its exponential stability. Ref.[20] contains a detailed proof showing almost global stability for several classes of complementary filters.

The filters presented here were implemented within SLUGS (**S**anta Cruz **L**ow-cost **U**nmanned Aerial Vehicle **G**uidance, Navigation & Control **S**ystem). The SLUGS platform is an open-source system which is designed from the outset to be a very flexible environment for research in guidance, navigation, and control.[21] See the Appendix for a description of some of the features of the SLUGS environment. This flexibility allowed major changes to the filter designs to be implemented, verified, validated in simulation, and downloaded to the UAV in hours, not days. The novelty of this paper is in demonstrating how the SLUGS environment allowed rapid changes in filter design to be cycled from algorithm design through to flight test. The complementary filters of the final design show comparable results to those of a full-state EKF with the advantage of being implemented and flight tested on a very low-cost and relatively low processing-power 16-bit digital signal controller with no FPU.

The remainder of the paper is composed of the following: Section II describes the initial design of the attitude and position filters; Section III discusses calibration, and ultimate redesign of the position filters based on flight test telemetry data; Section IV discusses the performance of the filters in flight tests and in comparison with a 15 state Extended Kalman Filter replaying actual flight data from another source (including a truth reference).

II. Initial Filter Design

A. Attitude Filter

There are many different approaches to estimating attitude from gyroscope, accelerometer and GPS measurements. One method involves integrating the differential equations for Euler angles with GPS to update estimates of position, velocity, and biases in gyros and accelerometers.[22] Another technique is that of an attitude and heading reference system (AHRS). This approach uses the fact that a body's attitude can be determined from two known, non-collinear inertial directions. Accelerometers are used to determine the direction of gravity, a low frequency measurement, and integrated gyro rates for changes in attitude, thus "leveling", or determining the *down* direction. Another measurement, typically the earth's magnetic field, is used for a heading reference.[4]

The design of the attitude filter begins with a complementary filter[23] but used a magnetometer triad for a heading reference. The output of the attitude filter consisted of information of the estimated attitude (Euler angles and direction cosine matrix) obtained from the integrated quaternion, and gyro bias estimates. The attitude (and position) filters were created in Simulink, and these models were driven by a Simulink simulation of a 6 degree-of-freedom (DOF) UAV flying a closed course defined by waypoints. The waypoints required changes in altitude as well as changes in track. The filter tracked the Euler angles well in the presence of realistic noise except in pitch when the UAV descended to a lower altitude. This behavior was traced to the original implementation of the leveling logic. The leveling process estimates the direction of gravity in body coordinates based on accelerometer readings and an independent estimate of inertial acceleration. Accelerometers measure specific force:

$$a_r = a_i - g, \tag{1}$$

where a_r is the vector of accelerometer readings, a_i is the inertial acceleration, and g is the acceleration due to gravity. The direction of the gravity vector is estimated from

$$\hat{u}_g = \frac{a_i - a_r}{\mid a_i - a_r \mid}. \tag{2}$$

If a_i is computed as seen in body coordinates, the full expression is:

$$a_i = \frac{dV}{dt} + \omega \times V, \tag{3}$$

where V is inertial velocity of the UAV, dV/dt is the longitudinal acceleration, or rate of change of inertial velocity as seen in body coordinates, and ω is the angular velocity of the body relative to inertial coordinates.

Many designs use the second term only,[23,24] and it is common to hear practitioners talk about the "omega cross V" term since most papers are concerned with a UAV flying in level flight at constant airspeed.

However, longitudinal acceleration turned out to be important for our application because it was significant when changing altitudes. The descent profile was a reduction of thrust to a minimum with a maximum pitch down attitude. This resulted in an increase in airspeed and inertial velocity ($dV/dt > 0$), then a deceleration back to commanded airspeed ($dV/dt < 0$) upon leveling off, both of which caused errors in the estimated pitch angle. The impact of longitudinal acceleration on estimated pitch angle is shown in Figure 1. In this simulation both maximum longitudinal acceleration and deceleration were obtained in level flight using thrust only. Although the longitudinal accelerations experienced in the waypoint-following simulation were not as large as these, they were enough to cause significant pitch error.

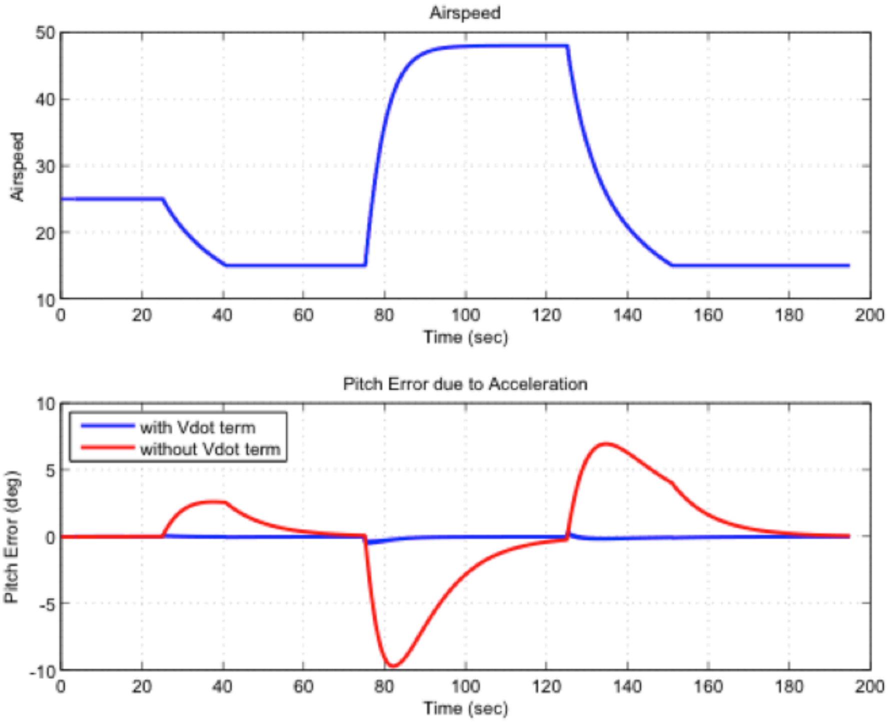


Figure 1: Pitch Error Induced by Acceleration.

Several candidates for computing the dV/dt term were compared on the basis of tracking error before choosing a high pass filter to compute the approximate numerical derivative. Rather than use the high pass filter to calculate dV/dt directly in body coordinates for equation 3, the high pass filter was applied to the inertial velocity estimates in inertial coordinates, and this was transformed to body coordinates (using the estimated attitude) and used as an estimate for a_i in equation 2.

The major components of this design are shown in Fig. 2. The inputs to the filter are the sensed acceleration, A_{sens}, estimated accelerometer bias, A_{bias}, sensed gyro angular rates, G_{sens}, inertial velocity V, and a unit heading reference vector ψ_{ref}. Note that the inertial velocity goes through a high pass filter mentioned above. The outputs of the filter are: estimates of gyro bias, G_{bias}, direction cosine matrix (DCM) from local north-east-down to body coordinates, and, for convenience, the Euler angles (ϕ, θ, ψ).

The attitude parametrization used internally in the filter is a unit quaternion. The quaternion, q is used to compute the direction cosine matrix (DCM), and Euler angles ψ, θ, ϕ. The inputs to the quaternion rate of change are the angular rates from the gyros, G_{sens}, and the error signals from the estimate of gravity, and the estimate of heading error. The equation for the quaternion rate of change is:

$$\dot{q} = \frac{1}{2} q \otimes (0, \hat{\Omega})^\top, \tag{4}$$

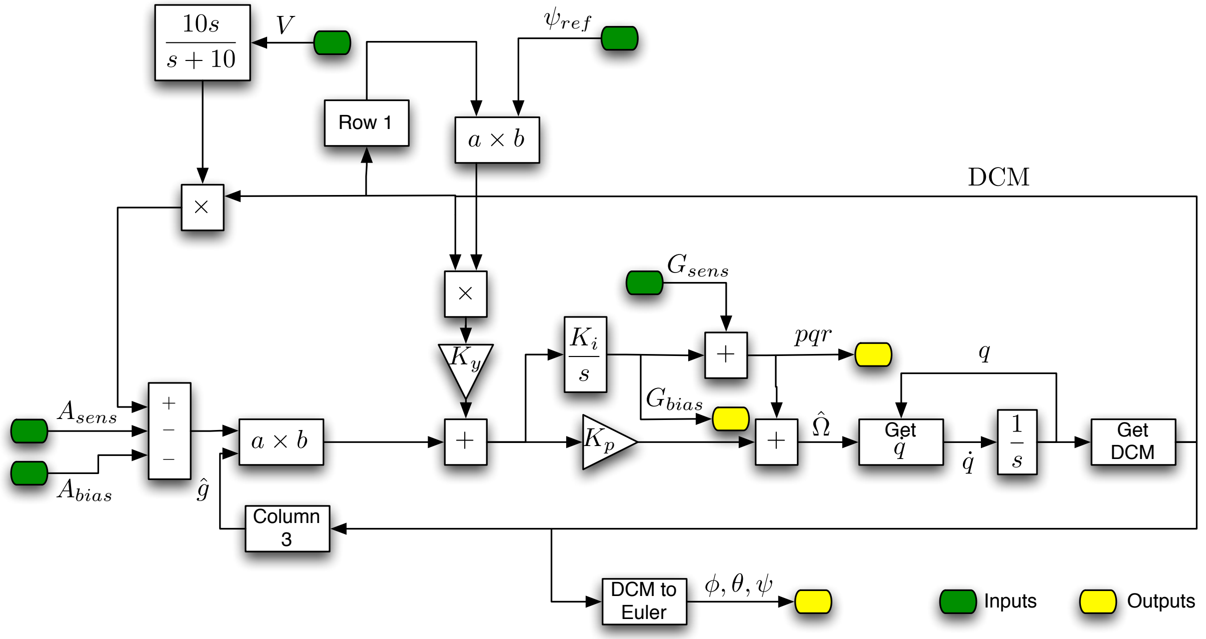


Figure 2: Block Diagram of Attitude Filter

where $\otimes$ represents the quaternion product, and the angular rate $\hat{\Omega}$ is:

$$\hat{\Omega} = G_{sens} + K_p(\epsilon_g + K_Y \epsilon_Y) + G_{bias} \tag{5}$$

where ϵ_g is the error signal derived from misalignment of the estimated Z axis with estimated gravity vector, and ϵ_Y is the error signal derived from the misalignment of the estimated direction of the heading reference.

One of the advantages of the complementary filter is the small number of tuning parameters that need to be chosen. Useful guidelines appear in.[23] In our design K_p was 0.1sec^{-1}, and K_i was 0.001sec^{-2}. The gain K_y controls the ratio of the yaw reference error and the leveling error and a value of 3 was used throughout.

Two attitude filters were designed and flight tested: one used magnetometers for the heading reference ψ_{ref}, the other used GPS course-over-ground (*COG*). The *COG* design was created because of the possibility of unreliable magnetometer readings. The *COG* reference ignores differences between heading and ground track. As such, it can create errors in estimates of actual heading angle as well as transient errors when the relative wind changes. However, flight tests were conducted in relatively low wind conditions, thus minimizing these effects.

B. Position Filter

The initial design for the position filter was a 9-state extended Kalman Filter with state variables consisting of position and velocity in north-east-down (NED) coordinates, and accelerometer biases. Measurements for the position filter were horizontal GPS position and velocity, barometric altitude, accelerometer readings, and the direction cosine matrix from the attitude filter. The filter performed strap down navigation calculations for the state and covariance matrix at 100Hz between GPS updates by integrating accelerometer readings after they were transformed to NED coordinates. The state variables were updated with the GPS readings according to the assumed GPS noise covariance and the propagated *a priori* state covariance. This filter performed very well in the 6DOF simulations with simulated noise whose characteristics matched the sensor specifications, and this filter was downloaded to the autopilot for further flight testing.

The combined attitude and position filters are shown at a high level in Figure 3 which shows the signal flows between these systems.

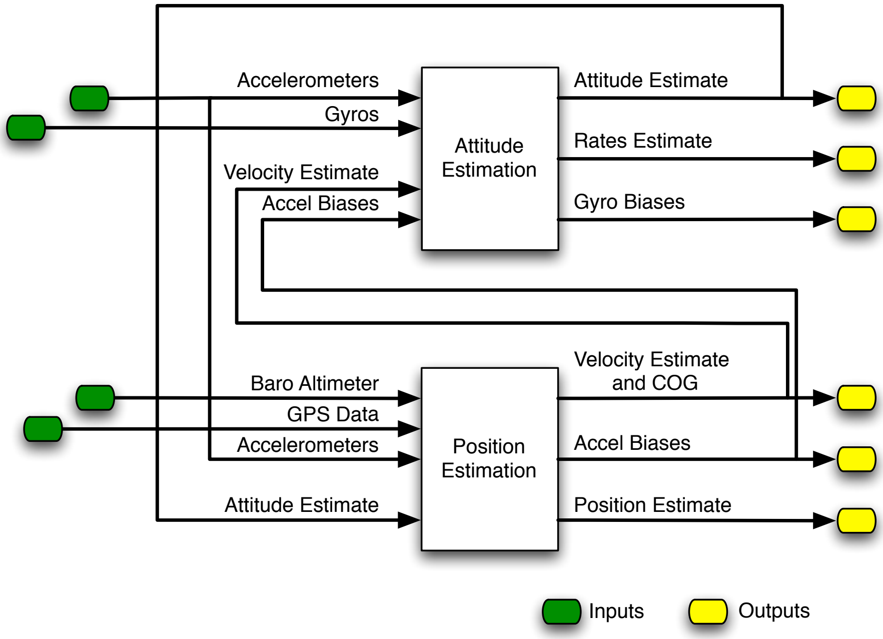


Figure 3: Block Diagram of Position and Attitude Filter

III. Flight Test Calibration and Filter Redesign

The objective of the first few flight tests was to log sensor readings while the UAV was manually controlled by a pilot on the ground to validate attitude and position estimates before using these outputs for closed-loop guidance and control. Magnetometer readings were taken in the field during static ground test before the first flight, and they were deemed unreliable due to inconsistent static readings and high noise levels due to the UAV's electric motor. The machine code for both the magnetometer and the *COG* attitude filters had been created beforehand, so the code for the attitude filter with *COG* heading reference was downloaded to the autopilot in the field and used for the remainder of the flights. (The MATLAB/Simulink models and supporting code resided on the ground station. If necessary, any Simulink model could be modified, recompiled, and downloaded to the autopilot in the field.

Sensor readings during manually controlled flight were recorded in the ground station via RF telemetry. These data were then replayed, post flight, though the Simulink position and attitude models. The redesign process is shown in Figure 4. The telemetry data from the flights are fed into the Simulink models of the position and attitude filters; the response of these filters is examined and design changes can be made based on these results. Finally the Simulink model is compiled and downloaded to the autopilot for validation via flight tests.

This replay turned out to be one of the most valuable steps in the process because it allowed the models to operate on real, in-flight, sensor readings, not pseudo random data with standard distribution functions parameterized from specification sheets. The SLUGS ground station software prepared the flight data in real time so that there was very little delay between flight and the replay.

The replay from the manually controlled flight showed that the accelerometers' in-flight readings were corrupted with a significant amount of noise. This had a direct impact on the performance of the EKF which had been implemented with a more optimistic noise model (extracted from the data sheet and valid for static conditions). An attempt to identify the source of the noise (e.g.: vibration, electromagnetic interference), proved unsuccessful. The telemetry data also showed that the GPS readings were updating at an average of 3Hz and had little noise except for the Z-axis (altitude). The Z readings showed several hundred feet of drift during a 10-minute flight, as well as *a periodic*, and very large abrupt changes in Z position readings associated with satellites dropping in and out of the solution.

Because of the relatively noiseless GPS data, its reasonably fast update rate, and looming project dead-

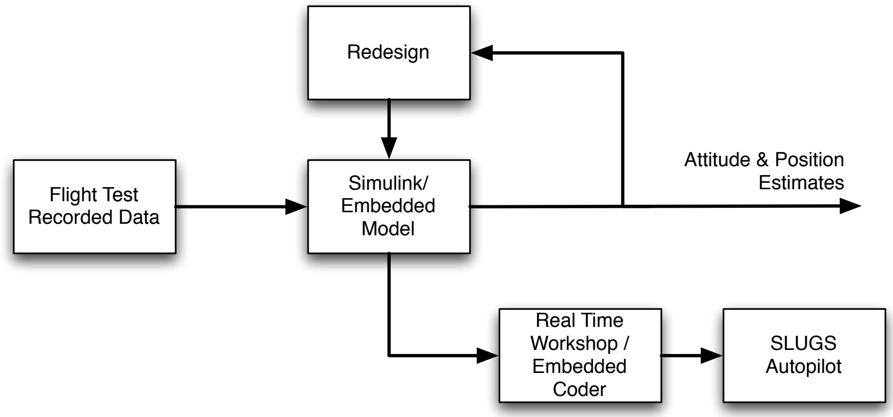

Redesign
Flight Test Recorded Data
Simulink/ Embedded Model
Attitude & Position Estimates
Real Time Workshop / Embedded Coder
SLUGS Autopilot

Figure 4: Replay and Iterative Design Process

lines, it was decided to replace the EKF in the horizontal channel with complementary filters that blended GPS position and velocity. In fact, a satisfactory result was obtained by predicting the X-Y positions between GPS updates using the last GPS position plus the last estimated velocity of the last GPS reading. The change in X-Y velocities between GPS updates were predicted using a version of a first-order-hold but with a weighted average of the last two changes as shown in equations 6–9.

$$\hat{P}(t) = P_{GPS}(t_k) + V_{GPS}(t_k)(t - t_k), \quad t >= t_k \tag{6}$$

$$\hat{V}(t) = V_{GPS}(t_k) + (\alpha a_k + (1-\alpha)a_{k-1})(t - t_k), \quad t >= t_k \tag{7}$$

$$a_k = (V_{GPS}(t_k) - V_{GPS}(t_{k-1}))/(t_k - t_{k-1}) \tag{8}$$

$$a_{k-1} = (V_{GPS}(t_{k-1}) - V_{GPS}(t_{k-2}))/(t_{k-1} - t_{k-2}), \tag{9}$$

where $\hat{P}(t), \hat{V}(t)$ are the estimated horizontal position and velocity, $P_{GPS}(t_k), V_{GPS}(t_k)$ are the GPS readings of horizontal position and velocity at time t_k, and a_k, a_{k-1} are estimates of horizontal acceleration over the last two GPS update intervals. α is a weighting coefficient between the last two accelerations and a factor of 0.8 was used. This finite memory filter estimating acceleration proved to be very effective, more so than a conventional Kalman Filter for this problem due of it's finite memory.

The outlier changes in GPS Z readings were trapped with a threshold test such that the sample-to-sample change in GPS Z reading passed to the CF was either (1) the actual change or (2) zero if the change from the last reading was too large. These jump-free readings were blended with the barometric altitude as the low frequency signal to eliminate drift.

Note that using the actual flight sensor data from test flight, these new designs were proposed, implemented in Simulink models, simulated, compiled, and downloaded to the autopilot in less than 24 hours, demonstrating the flexibility of SLUGS environment. Comparable changes using high level languages such as C/C++ would have taken several days, and would have been more prone to error, thus requiring a much longer validation cycle. The ability to switch not only the sensor details, but the entire structure (EKF to CF) within the Simulink modeling environment makes such code revisions relatively easy and free of implementation errors. Validation against actual flight data using the replay function is an additional step that is easy with the SLUGS environment.

Not only did these changes result in improved filter performance, but the change from EKF to CF reduced the computational load on the sensor DSC from nearly 100% to 40%. This is especially interesting as the DSC is emulating floating point operations, and would be able to see an additional large performance increase by moving to a fixed point implementation of the CF code (this too, can be accomplished within the Simulink environment by using the Fixed Point blockset tools).

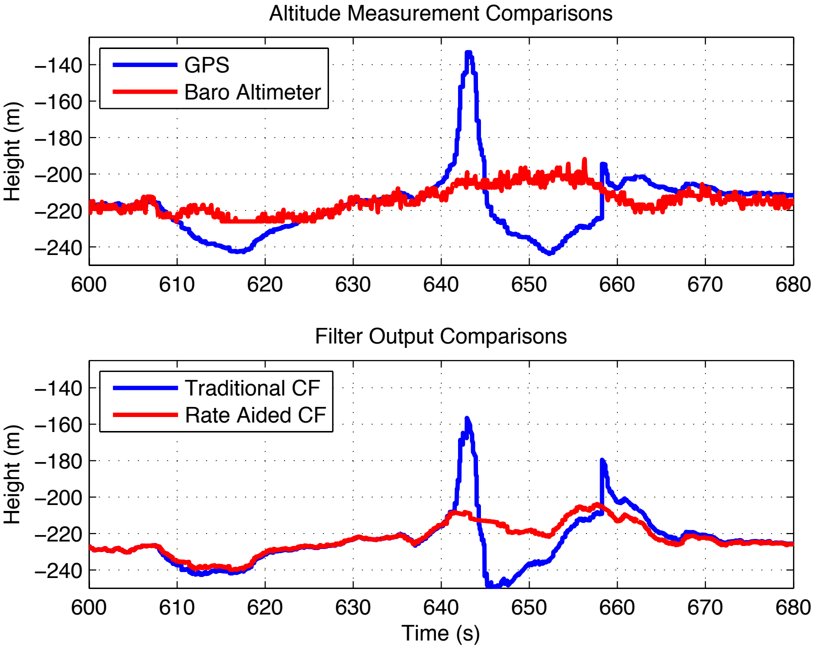


Figure 5: Typical CF and Rate-Aided Estimates of Height. The top plot compares the altitude as reported from the GPS (blue) and the barometric height (red). Note how, according to the GPS, the UAV climbed 100 meters in a period of 5 seconds [645 - 650sec]. This is clearly wrong. The bottom plot shows the output of a traditional CF (blue) and compares it with the rate-aided height (red). Note how once the abrupt changes (reported by the GPS in Z) fade out, both altitudes coincide.

IV. Results

A. UCSC Flight Tests

The main objective of the extensive flight tests with the Mentor UAV was to explore various control schemes and navigation architectures. In the case of the flight tests, a non-linear flight guidance law was used to keep the aircraft on the trajectories between waypoints.[25] The trajectory of a nominal flight in our test field shown in Fig. 6. The overshoot in the lower-left and "short-cutting" in the upper left are artifacts of the lateral guidance and its response to the prevailing top-to-bottom wind.

Since there were no independent "truth" measurements of position and attitude, no detailed evaluation is available for these flights. However, all the functions related to attitude and position estimates, such as lateral guidance and airframe stabilization, performed quite well to control the aircraft trajectory.

B. University of Minnisota Flight Tests

UAV flight test sensor data is available in the DVD distributed in the book edited by Gleason and Gebre-Egziabher.[22] The data consist of IMU data (gyro and accelerometer readings) and GPS readings from Micro Nav, a low cost IMU/GNSS system made by Crossbow. Also flown on the same UAV was a MIDG II, an integrated IMU/GNSS system that provided higher quality position and attitude estimates. Current MIDG II specifications are 0.4° (1-σ) in pitch and roll, and 2° (1-σ) in heading. The complete data set was collected over 13.5 minutes period of a UAV flying in circular patterns (Figure 7).

The Crossbow IMU and GPS data were played through the complementary filters in the SLUGS replay mode. Fig. 8 shows 2 minutes of traces of the CF Euler angles and their references with a GPS update rate of 1Hz. At this scale, there is good agreement with the more accurate MIDG II estimates.

For a more detailed evaluation, error statistics were computed during a 5-minute flight interval. The data were used to drive the CF as well as a 15-state EKF whose source code is provided on the DVD.[22] Both

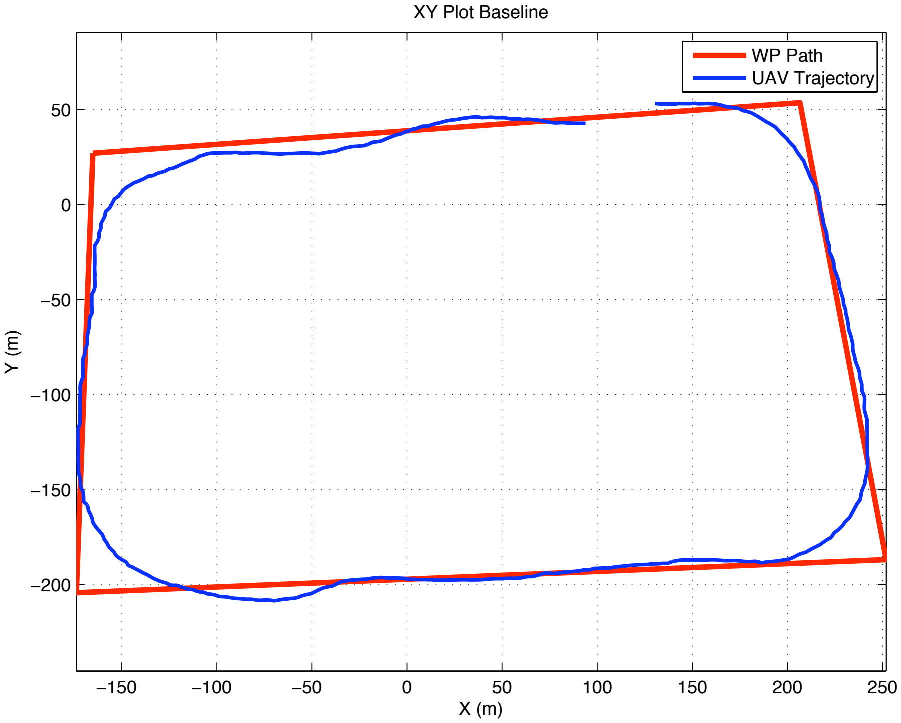


Figure 6: UCSC UAV Ground Track

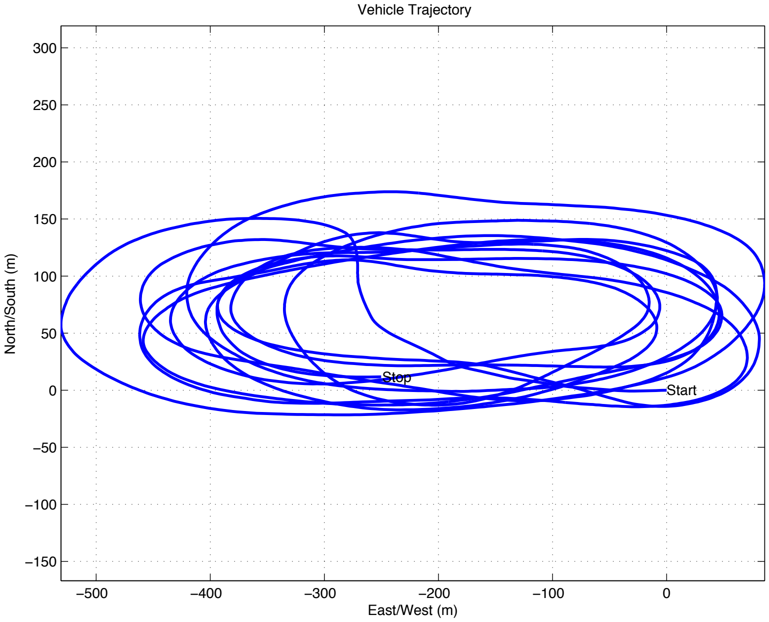


Figure 7: UAV Trajectory t = [7, 13.5] minutes.

filters were compared to the MIDG II data that was regarded as truth for pitch and roll. The yaw output of the CF was compared with the *COG* from the GPS since that was the horizontal reference for the CF. The

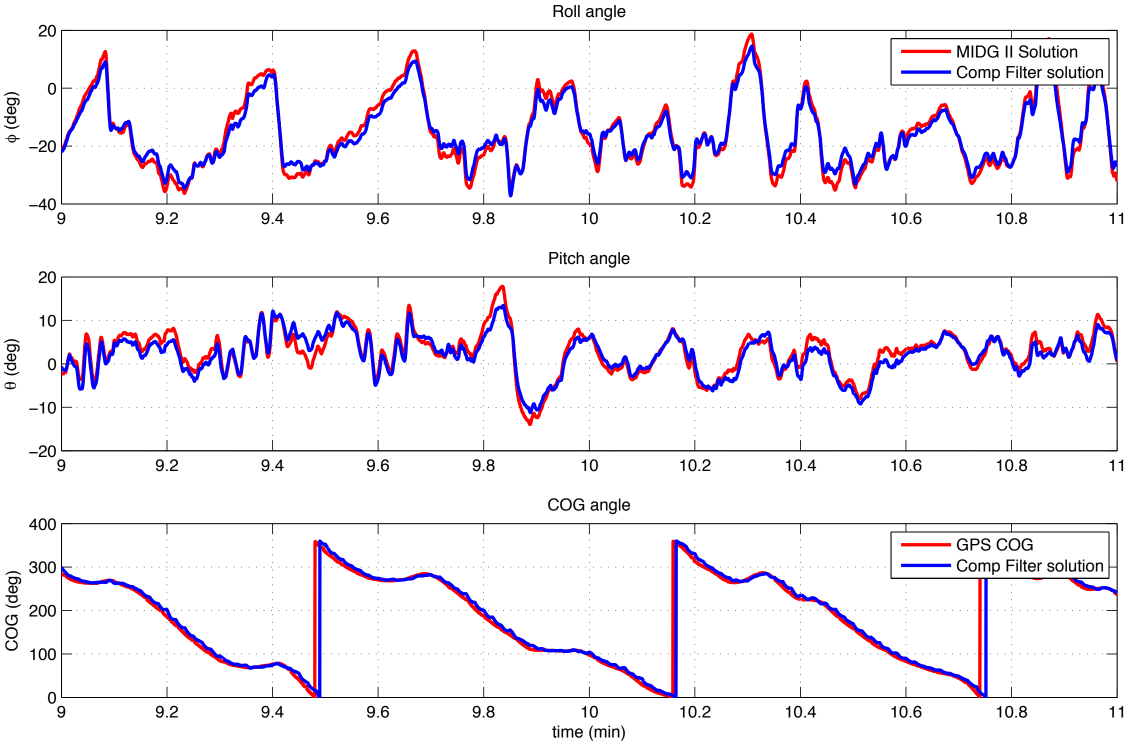


Figure 8: Complementary Filter Euler Angles (blue) and Their References (red). GPS Update Rate 1Hz.

heading output of the EKF was compared to the MIDG heading.

The GPS data on the DVD is updated at 10Hz. This allowed testing of the CF and EKF at different GPS update rates, and the following results are presented for GPS updates at 1 Hz and 3 Hz which are more realistic update rates for low-cost off-the-shelf GPSs. The comparisons are displayed in Figure 9 and Table 1 and Table 2. Figure 9a shows the mean error and Figure 9b shows the standard deviation.

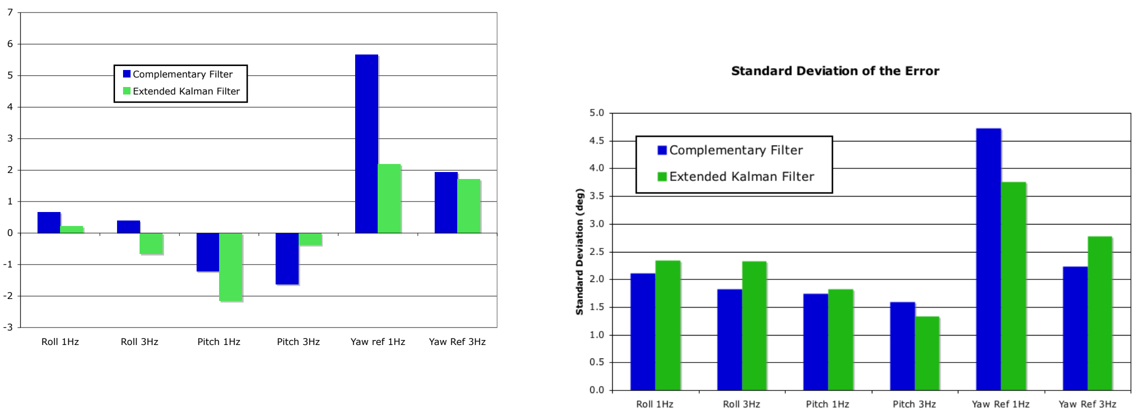
A note of caution: comparing the Yaw Reference performance of the CF and EKF is not strictly proper because the EKF is tracking heading, and the CF is tracking *COG*. Although the low frequency content of these values is similar (varying only by the heading correction for wind), the high frequency content could be quite different. Therefore the relative performance of the errors in yaw reference should be considered as representative.

Fig. 9a and Table 1 show that the mean errors of the EKF and the CF are comparable, except for the yaw reference at a GPS update rate of 1 Hz. At the 3 Hz GPS update rate the errors are very similar. This means the performance of the CF and EKF are essentially the same since a GPS update rate of 3 Hz is readily available in low-cost GPS receivers.

	Mean Error (degrees)					
	Roll (1Hz)	Roll (3Hz)	Pitch (1Hz)	Pitch (3Hz)	Yaw (1Hz)	Yaw (3Hz)
Complementary Filter	0.67	0.40	-1.20	-1.60	5.67	1.94
Extended Kalman Filter	0.22	-0.65	-2.15	-0.36	2.20	1.71

Table 1: Mean Angular Errors (degrees)

The standard deviation of both EKF and CF errors, shown graphically in Fig. 9b and numerically in Table 2, indicate that there is no practical difference between the CF and the EKF, especially at the higher



(a) (b)

Figure 9: CF and EKF Error Statistics. (a) Mean Error of CF Error (blue) and EKF Error (green). GPS Update Rate, 1Hz and 3Hz. (b) Standard Deviation of CF Error (blue) and EKF Error (green). GPS Update Rate 1Hz and 3Hz

	Error Standard Deviation (degrees)					
	Roll (1Hz)	Roll (3Hz)	Pitch (1Hz)	Pitch (3Hz)	Yaw (1Hz)	Yaw (3Hz)
Complementary Filter	2.11	1.83	1.75	1.60	4.73	2.24
Extended Kalman Filter	2.34	2.33	1.82	1.33	3.76	2.78

Table 2: Error Standard Deviation (degrees)

update GPS rate of 3Hz. Given the greater ease of tuning the parameters for the CF than the EKF, more robust performance to larger errors, and lower computational burden, the CF demonstrates an excellent candidate for an embedded attitude and position filter for small UAVs.

V. Conclusions

The process of design and development of the attitude and position filters for the UCSC UAV was a series of iterative changes, driven both by theory and the results on experimental data. This is just like any other design that goes from concept, to simulation, to validation, and finally to flight tests. There were many unanticipated conditions which were discovered, evaluated, mitigated, and overcome. In this regard the SLUGS environment turned out to be very useful and shortened the time from start to finish by months. The complementary filters in the final design performed well when compared to an Extended Kalman Filter. Both the CF and EKF seem to be more sensitive to GPS update in following the yaw reference, and at 3 Hz, the performance of the CF can be considered very similar to the EKF. It is well known that higher GPS rates can vastly improve performance of low cost IMUs. Unfortunately, the matrix computations demands of the EKF are especially burdensome at higher update rates. For the systems evaluated here, the readily-achieved GPS update rate of 3 Hz, and the lower computational requirements of the CF, make then attractive alternatives. Flight tests demonstrated that the CF performance is adequate to stabilize the platform while delivering excellent waypoint and trajectory tracking.

Appendix: The SLUGS System

The filters presented here were implemented within SLUGS (**S**anta Cruz **L**ow-cost **U**nmanned Aerial Vehicle **G**uidance, Navigation & Control **S**ystem) The SLUGS platform is an open-source system which is a very flexible environment for research in guidance, navigation, and control. The essential SLUGS components for this discussion are:

1. A low-cost, light-weight, low-power autopilot (Shown in Fig. 10), with a ADIS16355 Tri-axis inertial sensor; a San Jose Navigation EB-85 GPS receiver with asynchronous updates averaging 3Hz; a HMC5843 magnetometer triad; and pressure transducers for air data computations. Computations are handled with two Microchip dsPIC33FJ256MC710 digital signal controllers (DSC) one for sensor calculations, and one for guidance and control tasks.

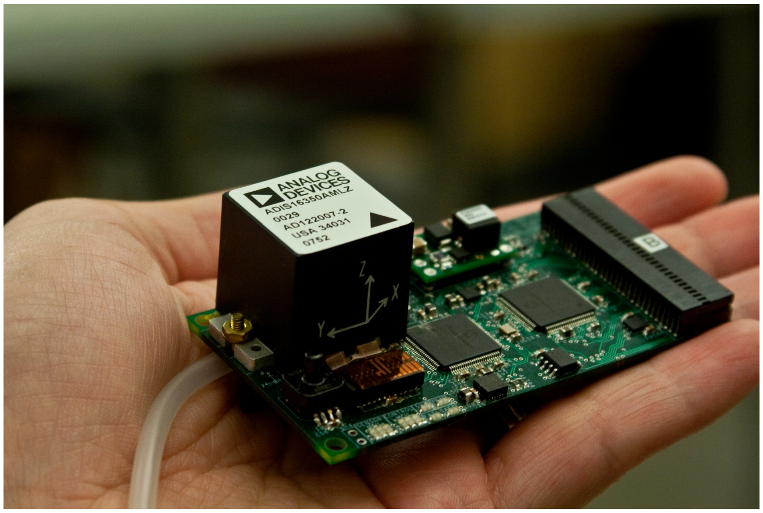


Figure 10: SLUGS Autopilot Board

2. The design of all the GNC algorithms is accomplished in Simulink using block diagrams and Embedded MATLAB functions. Once designs are verified through simulation, they are compiled directly into DSC executable code using Simulink"s Real Time Workshop auto code generation. This executable firmware is downloaded directly to the autopilot for hardware-in-the-loop simulation and flight tests.

3. Ground station software, for sending commands, gains, and waypoint information to the autopilot; and, in turn, receiving telemetry of system parameters, system health, sensor readings, and other logged data.

4. An efficient way to record inflight sensor data and other variables via telemetry, convert to a MATLAB format, and drive attitude and position filters in the Simulink environment. This "Replay" capability allows rapid design change to actual signal and noise characteristics.

The aircraft used for flight tests was a small off-the-shelf radio-controlled (RC) aircraft with a wingspan of 1.65 meters weighing 2.01 kilograms with the the autopilot mounted on the rear part of the "cockpit"(see Fig. 11). The motors, batteries, radio modems, autopilot, and supporting hardware added approximately 1.2 kilograms of payload.

References

[1] Kingston, D. and Beard, R., "Real-time attitude and position estimation for small UAVs using low-cost sensors," *American Institute of Aeronautics and Astronautics*, 2000.

[2] Beard, R., Kingston, D., Quigley, M., Snyder, D., Christiansen, R., Johnson, W., McLain, T., and Goodrich, M., "Autonomous vehicle technologies for small fixed wing UAVs," *AIAA Journal of Aerospace Computing, Information, and Communication*, Vol. 2, No. 1, 2005, pp. 92–108.

[3] Nonami, K., "Prospect and Recent Research & Development for Civil Use of Autonomous Unmanned Aircraft as UAV and MAV," Vol. 1 of *Journal of System Design and Dynamics*, 2007.

[4] Gebre-Egziabher, D. and Elkaim, G., "MAV attitude determination by vector matching," *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 44, No. 3, july 2008, pp. 1012 –1028.

[5] Wang, M., Yang, Y., Hatch, R., and Zhang, Y., "Adaptive filter for a miniature MEMS based attitude and heading reference system," *Position Location and Navigation Symposium, 2004. PLANS 2004*, 2004, pp. 193–200.

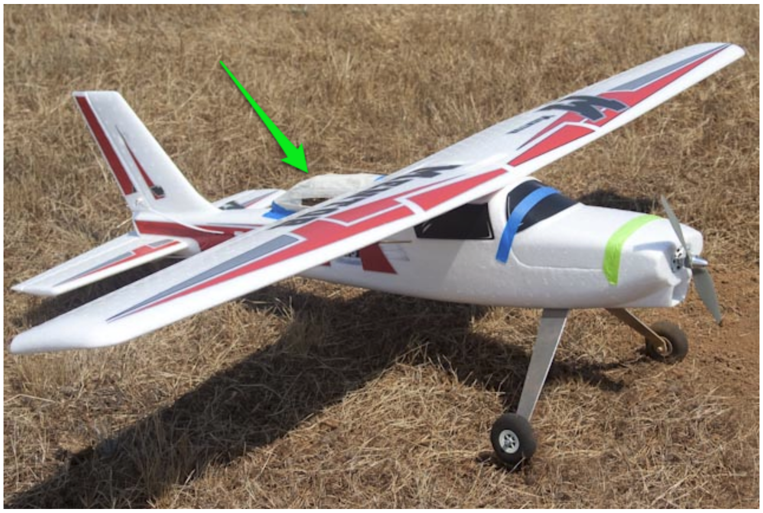


Figure 11: Multiplex Mentor RC Aircraft. Green arrow shows the autopilot mount.

[6] Julier, S., Uhlmann, J., and Durrant-Whyte, H., "A new approach for filtering nonlinear systems," *Proceedings of the American Control Conference*, Vol. 3, American Automatic Control Council, Evanston, IL, 1995, pp. 1628–1632.

[7] Wan, E. and Van Der Merwe, R., "The unscented Kalman filter for nonlinear estimation," *The IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC*, 2000, pp. 153–158.

[8] Sasiadek, J., Wang, Q., Johnson, R., Sun, L., and Zalewski, J., "UAV Navigation Based On Parallel Extended Kalman Filter," *AIAA Guidance, Navigation and Control Conference Denver, Colorado*, 2000.

[9] Van Der Merwe, R., Wan, E., and Julier, S., "Sigma-point Kalman filters for nonlinear estimation and sensor-fusion: Applications to integrated navigation," *Proceedings of the AIAA Guidance, Navigation & Control Conference*, 2004.

[10] Shin, E., *Estimation techniques for low-cost inertial navigation*, Ph.D. thesis, University of Calgary, 2005.

[11] Roberts, R., "Applied Mathematics in Integrated Navigation Systems," 2007.

[12] Merhav, S., "Aerospace Sensor Systems and Applications," 1998.

[13] Britting, K., *Inertial Navigation System Analysis*, Wiley-Interscience, 1971.

[14] Kayton, M. and Fried, W., editors, *Avionics Navigation Systems*, Wiley, 1996.

[15] Savage, P., "Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms," *Journal of guidance control and dynamics*, Vol. 21, 1998, pp. 19–28.

[16] Savage, P., "Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms," *Journal of guidance control and dynamics*, Vol. 21, 1998, pp. 208–221.

[17] Crassidis, J., Markley, F., and Cheng, Y., "A survey of nonlinear attitude estimations methods," *Journal of Guidance, Control, and Dynamics*, 2007.

[18] Brown, R., "Integrated Navigations Systems and Kalman Filtering: A Perspective," *Journal of the Institue of Navigation*, Vol. 19, No. 4, 1972, pp. 355–382.

[19] Baldwin, G., Mahony, R., Trumpf, J., Hamel, T., and Cheviron, T., "Complementary filter design ont e Special Euclidean group SE(3)," *Proceedings of the European Control Conference*, 2007.

[20] Mahony, R., Hamel, T., and Pflimlin, J., "Nonlinear complementary Filters on the Special Orthogonal Group," *IEEE Transactions on Automatic Control*, Vol. 55, No. 5, S 2008, pp. 1203–1218.

[21] Lizarraga, M. I., Elkaim, G. H., Horn, G. M., Curry, R., Dobrokhodov, V., and Kaminer, I., "Low Cost Rapidly Reconfigurable UAV Autopilot for Research and Development of Guidance, Navigation and Control Algorithms," *ASME/IEEE International Conference on Mechatronic Embedded Systems and Applications*, 2009.

[22] Gebre-Egziabher, D., Petovello, M., and Bevly, D., *Integration of GNSS and INS: Part 1 & 2*, chap. 6 & 7, Artech House Publishers, 1st ed., 2009.

[23] Euston, M., Coote, P., Mahony, R., Kim, J., and Hamel, T., "A Complementary Filter for Attitude Estimatin of a Fixed-Wing UAV," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 340–345.

[24] Vasconcelos, J. F., Calvario, P., Oliveira, P., and Silvestre, C., "GPS Aided IMU for Unmanned Air Vehicles," *5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, 2004.

[25] Park, S., Deyst, J., and How, J., "Performance and Lyapunov Stability of a Nonlinear Path-Following Guidance Method," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 6, 2007.