

2013

Hybrid Projectile Body Angle Estimation for Selectable Range Increase

Christopher J. Gioia
West Virginia University

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Gioia, Christopher J., "Hybrid Projectile Body Angle Estimation for Selectable Range Increase" (2013). *Graduate Theses, Dissertations, and Problem Reports*. 4968.
<https://researchrepository.wvu.edu/etd/4968>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Hybrid Projectile Body Angle Estimation for Selectable Range Increase

Christopher J. Gioia

Thesis submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements for the degree of

Masters of Science
In
Mechanical Engineering

Jay P. Wilhelm, Ph.D., Chair
Wade W. Huebsch, Ph.D.
Mridul Gautam, Ph.D.

Department of Mechanical and Aerospace Engineering
Morgantown, West Virginia
2013

Keywords: Hybrid Projectile, projectile motion, nonlinear estimation, sensor fusion, Extended Kalman filter,
Attitude estimation

ABSTRACT

Hybrid Projectile Body Angle Estimation for Selectable Range Increase

Christopher J. Gioia

A Hybrid Projectile (HP) is a tube launched munition that transforms into a gliding UAV, and is currently being researched at West Virginia University. A simple launch timer was first envisioned to control the transformation point in order to achieve maximum distance. However, this timer would need to be reprogrammed for any distance less than maximum range due to the nominal time to deployment varying with launch angle. A method was sought for automatic wing deployment that would not require reprogramming the round. A body angle estimation system was used to estimate the pitch of the HP relative to the Earth to determine when the HP is properly oriented for the designed glide slope angle. It was also necessary to filter out noise from a simulated inertial measurement unit (IMU), GPS receiver, and magnetometer. An Extended Kalman Filter (EKF) was chosen to estimate the Euler angles, position and velocity of the HP while an algorithm determined when to deploy the wings. A parametric study was done to verify the optimum deployment condition using a Simulink aerodynamic model. Because range is directly related to launch angle, various launch angles were simulated in the model. By fixing the glide slope angle to -10° as a deployment condition for all launch angles, the range differed only by a maximum of 6.1% from the maximum possible range. Based on these findings, the body angle deployment condition provides the most flexible option to maintain maximum distance without the need of reprogramming. Position and velocity estimates were also determined from the EKF using the GPS measurements. Simulations showed that the EKF estimates exhibited low root mean squared error values, corresponding to less than 3% of the total position values. Because the HP was in flight for less than a minute in this experiment, the drift encountered was acceptable.

Acknowledgements

This research project would not have been possible without the help and support of many people. Foremost, I would like to express my sincere gratitude to my advisor Dr. Wilhelm for his patience, guidance, motivation and knowledge while helping me work toward finishing my Master's degree.

I would also like to thank the other members of my thesis committee Dr. Wade Huebsch and Dr. Mridul Gautam for providing suggestions and edits on my thesis.

Last but certainly not least, I would like to thank my mother, father and sister who gave me the confidence and encouragement to continue my education. I would not be where I am today without your love and support.

Sincerely,

Christopher J. Gioia

Table of Contents

ABSTRACT	ii
Acknowledgements	iii
Table of Contents	iv
List of Figures	vi
List of Tables	viii
List of Symbols	ix
1. Problem Statement	1
2. Projectile Flight Dynamics	2
2.1. Euler Angles.....	2
2.2. Direction Cosine Matrix.....	5
2.3. Projectile Motion.....	6
2.4. Ballistics.....	9
3. State Acquisition	11
3.1. Inertial Measurement Unit	11
3.2. Global Positioning System.....	15
3.3. Magnetometer.....	16
4. State Estimation Methods	17
4.1. Linear Kalman Filter	17
4.2. Extended Kalman Filter	21
4.3. Unscented Kalman Filter.....	25
5. Model Construction	30
5.1. Euler Angles 6 DoF Block	31
5.2. Force Calculation Block.....	36
5.3. Sensor State Estimation Block.....	48
5.3.1. Sensors Block.....	49
5.3.1.1. <i>Inertial Measurement Unit Simulation</i>	51
5.3.1.2. <i>GPS Simulation</i>	54
5.3.1.3. <i>Magnetometer Simulation</i>	56
6. Extended Kalman Filter Implementation	58
6.1. State Estimation Process in Matlab.....	60
6.2. Sampling Time Selection	64
7. Results	66
7.1. EKF Error Analysis.....	70

7.2. Hold at -10° Wing Deployment Angle.....	73
7.3. Timer vs. Body Angle Deployment Comparison.....	75
7.4. Head Wind Impact	78
8. Conclusion and Future Work	79
8.1. Conclusion.....	79
8.2. Future Work	81
References.....	82
Appendix A- Matlab Code	86

List of Figures

Figure 1: WVU HP 60mm CAD (<i>Wilhelm 2012</i>).....	1
Figure 2: Rotation example.....	3
Figure 3: Elements of a trajectory as seen in <i>Carlucci 2008</i>	9
Figure 4: Triple Axis Digital Output Gyroscope- ITG-3200.....	13
Figure 5: Kalman filter cycle	19
Figure 6: Kalman filter Process Flow Chart	20
Figure 7: EKF Process Flow Chart.....	22
Figure 8: Hybrid Projectile Simulink Model	31
Figure 9: Reference Frame of the HP (<i>Wilhelm 2012</i>)	32
Figure 10: Subsystem of the 6 DoF Euler Angles block	32
Figure 11: Calculate omega_dot subsystem	34
Figure 12: Calculate DCM & Euler Angles subsystem	35
Figure 13: Calc Forces block subsystem	36
Figure 14: Aerodynamic Body Coefficients for the Hybrid Projectile (<i>Wilhelm 2012</i>).....	37
Figure 15: Body Coefficients Lookup subsystem.....	38
Figure 16: Distance Map for Extended Range of the HP	42
Figure 17: HP Distance Map Confidence Area of 98% (<i>Wilhelm 2012</i>).....	43
Figure 18: Graphing block subsystem	44
Figure 19: Model Calculated Euler Angles of a 1.68 kg HP with a 126 m/s Launch Velocity and Body Trip Angle of 10°	45
Figure 20: Altitude (top) and Distance (bottom) of a 1.68 kg HP with a 126 m/s Launch Velocity and Wing Deployment Body Angle of 10°	46
Figure 21: Trajectories of a HP, with a Wing Deployment Angle of -10°, and a Regular Projectile, both with a mass of 1.68 kg and a Launch Velocity of 126 m/s	Error! Bookmark not defined.
Figure 22: Maximum Distance vs. Launch Angle for a Hybrid Projectile with Wing Deployment Angle of -10° and a Regular Projectile given an Initial Velocity of 126 m/s.....	47
Figure 23: Model Calculated Body Velocities.....	48
Figure 24: Sensor State Estimation Subsystem	49
Figure 25: MPU-6000 IMU (<i>invensense.com</i>).....	50
Figure 26: ArdupilotMega hardware (<i>Anderson 2010</i>)	50
Figure 27: Sensors Subsystem	50
Figure 28: IMU Acceleration (top) and Rotation Rate (bottom) Measurements.....	53
Figure 29: IMU Acceleration (top) and Body Angle Rates (bottom) Measurements with noise .	54
Figure 30: GPS Simulation	55
Figure 31: GPS Simulated Position (top) and Velocity (bottom) with Noise.....	56
Figure 32: Magnetometer Simulation	58
Figure 33: EKF Matlab Function Block in Simulink Model	60
Figure 34: Pitch Error with Varying IMU Noise Power and EKF Sampling Time.....	65
Figure 35: Model and EKF Calculated Position for a Launch angle of 50°, a Body Angle Deployment of -10° and an Initial Velocity of 126 m/s	67
Figure 36: Model and EKF Calculated Velocity for a Launch angle of 50°, a Body Angle Deployment of -10° and an Initial Velocity of 126 m/s	68

Figure 37: Model and EKF Calculated Euler Angles for a Launch angle of 50°, a Body Angle Deployment of -10° and an Initial Velocity of 126 m/s	69
Figure 38: Error between Model and EKF Calculated Position for a Launch Angle of 50°, Wing Deployment Angle of -10°, and an Initial Velocity of 126 m/s.....	70
Figure 39: Error between Model and EKF Calculated Velocity for a Launch Angle of 50°, Wing Deployment Angle of -10°, and an Initial Velocity of 126 m/s.....	71
Figure 40: Error between Model and EKF Calculated Euler Angles for a Launch Angle of 50°, Wing Deployment Angle of -10°, and an Initial Velocity of 126 m/s.....	72
Figure 41: Maximum Range Extension based on Varying Launch and Trip Angles.....	73
Figure 42: Percent Difference between Actual Deployment Angle for Maximum Range and -10° Deployment.....	74
Figure 43: Range Extension Achieved based on Varying Trip Angle Deployment for a Launch Angle of 50°	75
Figure 44: Maximum Range Extension based on Varying Launch Angle and Timer Deployment	76
Figure 45: Range Extension based on Varying Timer Wing Deployment Values at a Launch Angle of 50°	77
Figure 46: Range Extension for Pitch Deployment vs. Timer Deployment	77
Figure 47: Wind Model Subsystem in 'Calc Forces' Block	78
Figure 48: Effect of Head Wind on Body Angle and Timer Wing Deployments at 50° Launch Angle and 126 m/s Initial Velocity.....	79

List of Tables

Table 1: IMU Simulation parameters	51
Table 2: Root Mean Squared Error Values for the EKF State Vector	70

List of Symbols

Symbol	Description	Units
A_b	Body velocity	m/s^2
a_x	Body x acceleration	m/s^2
a_y	Body y acceleration	m/s^2
a_z	Body z acceleration	m/s^2
C_x	Axial force coefficient	--
C_y	Side force coefficient	--
C_z	Normal force coefficient	--
C_l	Roll coefficient	--
C_m	Pitch coefficient	--
C_n	Yaw coefficient	--
C_L	Lift coefficient	--
C_D	Drag coefficient	--
CG	Center of gravity	m
CP	Center of pressure	m
f	Nonlinear function of the states	--
g	Acceleration due to gravity	m/s^2
GPS	Denotes GPS measurement	--
h	Measurement function	--
H	Observation matrix	--
I_{xx}, I_{yy}, I_{zz}	Moments of inertia about roll, pitch, and yaw axes	$kg \cdot m^2$

k	Time step	--
K	Kalman gain matrix	--
m_x, m_y, m_z	Magnetometer x measurement	
p, q, r	Roll rate	rad/s
P_k^-	Error covariance prediction at time step k	--
P_k	Error covariance at time step k	--
Q	Process noise covariance matrix	--
R	Measurement noise covariance matrix	--
u, v, w	Body velocity components	m/s
T_s	Sampling time	sec
U, V, W	Body velocity components	m/s
x, y, z	Global position components	m
V_x, V_y, V_z	Global velocity components	m/s
\hat{x}_k^-	State vector prediction at time k	$\begin{bmatrix} m \\ m \\ m \\ m/s \\ m/s \\ m/s \\ rad \\ rad \\ rad \end{bmatrix}$
\hat{x}_k	State vector at time k	$\begin{bmatrix} m \\ m \\ m \\ m/s \\ m/s \\ m/s \\ rad \\ rad \\ rad \end{bmatrix}$

z_k	Measurement vector at time step k	$\begin{bmatrix} m \\ m \\ m \\ m/s \\ m/s \\ m/s \\ rad \\ rad \\ rad \end{bmatrix}$
α	Incidence angle	rad
φ	Euler angle- roll	rad or deg
θ	Euler angle- pitch	rad or deg
ψ	Euler angle- yaw	rad or deg
ω	Body angular velocity	rad/s
$\frac{d\omega}{dt}$	Body angular acceleration	rad/s ²

1. Problem Statement

Traditional mortar rounds are launched from a tube and follow a parabolic trajectory toward their target without control. A Hybrid Projectile (HP), seen in Figure 1, is a tube launched munition that begins its flight as a simple projectile and at a predetermined point in flight; wings stowed in the body are deployed, transforming it into a glider (*Manole 2012*.) This enables the round to travel farther than a standard projectile. A simple launch timer was first envisioned to control transformation point for maximum distance. An optimal deployment time was calculated using various launch angle simulations for extended range. This method however, does not always extend range to its maximum because the deployment time depends on the launch angle. The maximum range, in turn, is affected by the launch angle. This means that in order to achieve a desired range, the timer would need to be reprogrammed before launch.

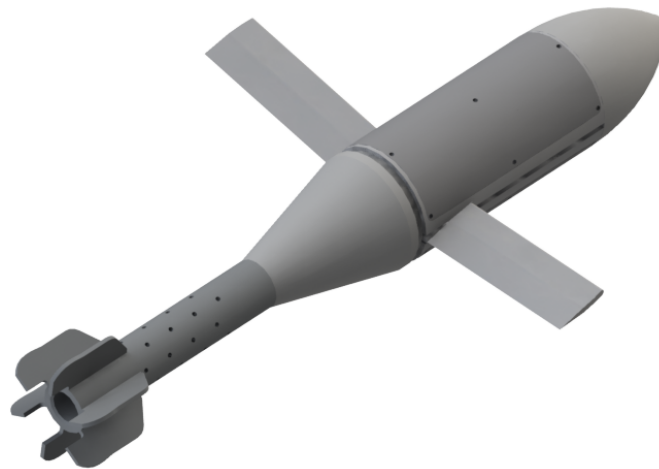


Figure 1: WVU HP 60mm CAD (*Wilhelm 2012*)

Due to this need, a method was sought for automatic wing deployment that would not require reprogramming the round when launching at various angles. A system was envisioned to estimate the pitch of the HP relative to the Earth to determine when the HP is properly oriented

for the designed glide slope angle, or the angle relative to the horizontal that the projectile maintains after wing deployment. An inertial measurement unit (IMU) could be used to estimate accelerations and body angle rates, and the estimation system provided a method of converting the body angle rates to useable body angle measurements. An extended Kalman filter (EKF) was used to estimate body angles and position of the HP, and also filter out noise from simulated IMU measurements, while an algorithm determined when to deploy the wings. An aerodynamic model was constructed to simulate the performance of the HP. In the next few sections, the model is broken down and each section is explained in detail.

2. Projectile Flight Dynamics

Hybrid projectiles, defined in *Manole 2012*, are designed to exhibit dynamics of both conventional projectiles and a fixed winged aircraft. The first portion of the flight is accurately represented by projectile motion of a rigid body with a fixed mass, and its equations. At some predetermined instance, wings stowed in the body of the hybrid projectile are deployed and its motion takes on the characteristics of a fixed winged aircraft. The body dynamics change and Euler angles must be defined to fully represent the equations of motion, position and attitude of the hybrid projectile. Ballistics can also be used to describe the flight and will be addressed in this section as well.

2.1. Euler Angles

In order to effectively transition between projectile and hybrid dynamics, the body angles, most importantly pitch, must be measured. Body angles are the orientation of a rigid object in space relative to three axes. They can be measured using either Euler angles or Quaternions. Both have

their place in aerodynamics, and each of their advantages will be explored. In the figure below, a simple rotation with respect to the y-axis is shown. Proper definition of the orientation depends on the order of the rotations. For instance, rotating about the x-axis then the y-axis will not yield the same orientation as rotating about the y-axis then the x-axis.

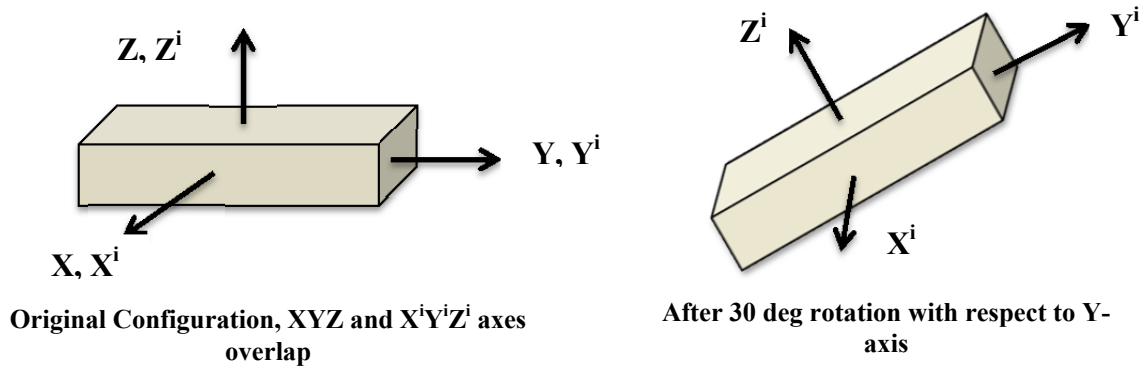


Figure 2: Rotation example

Euler angles are three independent angles that describe the orientation of some object with respect to some defined reference frame. Euler angles involve three successive rotations about three axes that are not, in general, perpendicular. By performing these three successive rotations in the proper sequence, a coordinate system can reach any orientation.

These rotations can be represented by matrices. Given the initial coordinate system XⁱYⁱZⁱ, the angle of rotation about the current Zⁱ axis is represented by ϕ^i , and the transformation matrix given by this rotation is:

$$A_1^i = \begin{bmatrix} \cos\phi^i & -\sin\phi^i & 0 \\ \sin\phi^i & \cos\phi^i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-1)$$

The coordinate system $X^i Y^i Z^i$ is then rotated about the current X^i axis by an angle θ^i , and the change of the orientation of the coordinate system can be described by the matrix:

$$A_2^i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta^i & -\sin\theta^i \\ 0 & \sin\theta^i & \cos\theta^i \end{bmatrix} \quad (2-2)$$

Finally, the $X^i Y^i Z^i$ coordinate system is rotated about the Y^i axis by an angle ψ^i and the change in orientation is represented by:

$$A_3^i = \begin{bmatrix} \cos\psi^i & 0 & -\sin\psi^i \\ 0 & 1 & 0 \\ \sin\psi^i & 0 & \cos\psi^i \end{bmatrix} \quad (2-3)$$

The overall transformation can be represented by the product of all three, and the final orientation of the coordinate system $X^i Y^i Z^i$ can be represented by:

$$A^i = A_1^i A_2^i A_3^i \quad (2-4)$$

Equation 2-4 can be expanded in terms of φ^i , θ^i and ψ^i and written as:

$$A^i = \begin{bmatrix} \cos\varphi^i \cos\psi^i + \sin\varphi^i \sin\psi^i \sin\theta^i & -\cos\theta^i \sin\varphi^i & \cos\psi^i \sin\varphi^i \sin\theta^i - \cos\varphi^i \sin\psi^i \\ \cos\psi^i \sin\varphi^i - \cos\varphi^i \sin\psi^i \sin\theta^i & \cos\varphi^i \cos\theta^i & -\sin\varphi^i \sin\psi^i - \cos\varphi^i \cos\psi^i \sin\theta^i \\ \cos\theta^i \sin\psi^i & \sin\theta^i & \cos\psi^i \cos\theta^i \end{bmatrix} \quad (2-5)$$

The three angles used, φ , θ , and ψ , are known as the Euler angles. The matrix A^i above can be constructed using these angles and three rotations about any axis. This section covers a Z-X-Y rotation, but the transformation matrix is not limited to any combination or any order. The process of representing three rotations is the same as listed above. (*Shabana 1994*) There is one drawback to using Euler angles: gimbal lock. Gimbal lock occurs when a set of rotations causes two axes to align, and one rotation has no effect. (*Lepetit 2005*) This singularity can result in ill-conditioned optimization problems. The use of quaternions helps to avoid this problem, but the risks of a singularity occurring were low in this project. Therefore, quaternions were not used.

2.2. Direction Cosine Matrix

The Direction Cosine Matrix (DCM) allows for a vector that is defined in the body (local) coordinate system to be converted to the global coordinate system. The multiplication of all three individual transformation matrices shown in matrix A^i above is the DCM. The nine elements of the matrix A^i above can be used to represent the direction cosines of the X^i , Y^i and Z^i axes in terms of φ , θ , and ψ . This means that the elements of columns 1, 2 and 3 are the direction cosines of the X^i , Y^i , and Z^i axes respectively. It appears that all nine elements are independent parameters, but there are really only three independent ones. Because of the six orthogonality (or normalization) conditions, the three column vectors are mutually perpendicular and the magnitude of each column vector is equal to one. (*Premerlani 2009*)

Using an example, the DCM of a Z^i - X^i - Z^i rotation would look like the matrix B^i show below:

$$B^i = \begin{bmatrix} \cos\psi^i \cos\varphi^i - \cos\theta^i \sin\psi^i \sin\varphi^i & -\cos\varphi^i \sin\psi^i - \sin\varphi^i \cos\psi^i \cos\theta^i & \sin\theta^i \cos\varphi^i \\ \cos\psi^i \sin\varphi^i + \cos\varphi^i \sin\psi^i \cos\theta^i & -\sin\varphi^i \sin\psi^i + \cos\varphi^i \cos\psi^i \cos\theta^i & -\sin\theta^i \cos\varphi^i \\ \sin\theta^i \cos\psi^i & \sin\theta^i \cos\psi^i & \cos\theta^i \end{bmatrix} \quad (2-6)$$

To get the Euler angles using direction cosines, the last row and column can be used by:

$$\theta^i = \cos^{-1} B_{33} \quad (2-7)$$

$$\varphi^i = \cos^{-1} \left(\frac{B_{32}}{\sin\theta^i} \right) \quad (2-8)$$

$$\psi^i = \cos^{-1} \left(\frac{B_{23}}{\sin\theta^i} \right) \quad (2-9)$$

Several rotation matrices can be multiplied together and get a rotation matrix that is equivalent to applying all of the rotations in succession. Because rotation matrices depend on the order of rotations, the order of the matrix multiplication matters as well. That is to say, matrix multiplication is not commutative and it must be left multiplied when converting from one orientation to a final orientation. (*Shabana 1994*)

2.3. Projectile Motion

Projectile motion, as seen in *Walker (2008)*, in its simplest sense is motion in a two-dimensional plane that moves in the vertical plane with some initial velocity \vec{v}_o and its acceleration is always the free fall acceleration \vec{g} , acting downward. In projectile motion air resistance is neglected. The horizontal and vertical components of projectile motion are independent of each other. Thus, the initial velocity of a projectile can be written as

$$(2-10)$$

$$\vec{v}_o = v_{o_x} \hat{i} + v_{o_y} \hat{j}.$$

The horizontal and vertical components v_{o_x} and v_{o_y} can be found if the launch angle θ_o between \vec{v}_o and the positive x-direction is known:

$$v_{o_x} = v_o \cos \theta_o \text{ and } v_{o_y} = v_o \sin \theta_o. \quad (2-11)$$

This enables the two-dimensional motion to be broken down into two one-dimensional motions, one for the horizontal (with zero acceleration) and one for the vertical motion (with constant downward acceleration.) Because there is no acceleration in the horizontal direction, the horizontal component of the projectile's velocity, v_x , does not change throughout the entire motion. The projectile's horizontal displacement $x - x_o$ from an initial position x_o at any time, t , can be expressed by

$$x - x_o = v_{o_x} t \quad (2-12)$$

Since $v_{o_x} = v_o \cos \theta_o$,

$$x - x_o = (v_o \cos \theta_o) t \quad (2-13)$$

The vertical motion is the motion of a particle in free fall with a constant acceleration \vec{g} , so the vertical displacement can be found by:

$$y - y_o = (v_o \sin \theta_o)t - \frac{1}{2}gt^2 \quad (2-14)$$

Likewise, the vertical velocity can be expressed by the equations:

$$v_y = v_o \sin \theta_o - gt \quad (2-15)$$

$$v_y^2 = (v_o \sin \theta_o)^2 - 2g(y - y_o) \quad (2-16)$$

As seen in Equation 2-15, the vertical velocity components behaves exactly like a projectile launched straight up. It starts from initial velocity, and its speed steadily decreases until it stops at maximum height. Its direction then reverses and gains speed with time. An equation of the projectile's path, or trajectory, can be found by eliminating t in equations (2-13) and (2-14).

Assuming x_o and y_o to be zero, the equation for the projectile's path is:

$$y = (\tan \theta_o)x - \frac{gx^2}{2(v_o \cos \theta_o)^2} \quad (2-17)$$

It can be noted from equation 2-17 that the motion is parabolic. The horizontal range, R , of a projectile is the horizontal distance the projectile has travelled when it returns to its initial height and can be expressed as:

$$R = \frac{v_o^2}{g} \sin 2\theta_o \quad (2-18)$$

The maximum range for a projectile occurs at a launch angle of 45° . However, equation 2-18 did not give the distance travelled for a projectile that lands at a different height than the launch height. This analysis also did not include drag forces acting on the projectile, which were considered in ballistic analysis.

2.4. Ballistics

Ballistics provides another approach to analyzing hybrid projectiles. It expands on projectile motion analysis to include other factors such as force due to drag, its spin as well as gravity. (Carlucci 2008) Carlucci provides a diagram of a more specific projectile motion analysis, by expanding on the trajectory of the round, and it can be seen below in Figure 2.

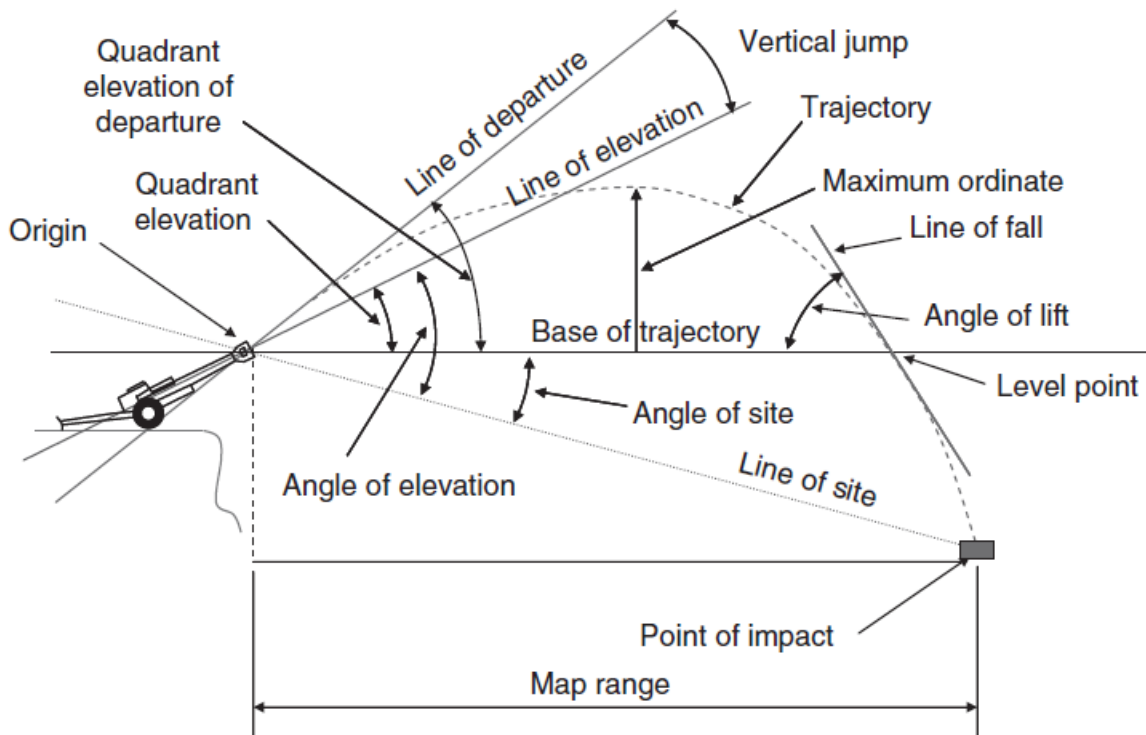


Figure 3: Elements of a trajectory as seen in Carlucci 2008

It should be noticed that the range (map range) defined above is different from the projectile motion range defined in Section 2.3 because it is defined below the height of launch (base of trajectory). The range and base of trajectory are important for proper trajectory definition. In this thesis, it was assumed that the launch height was negligible since the range is much greater than the launch tube height. Another important characteristic seen in Figure 2 is the difference between the line of departure and the line of elevation. It is common to assume that the projectile exits the barrel along the axis of the barrel, however, due to aerodynamic effects and the dynamics of the round this is not truly the case. Later in this thesis, the assumption will be made that the round leaves the barrel along the axis of the barrel, but the fact that this does not accurately represent a real world launch should be considered when examining the findings in later sections.

Exterior ballistic analysis takes into account forces, moments and coefficients acting on the projectile. The drag on a projectile is the force exerted on it as it moves through the air. In general, two types of drag exist: pressure drag and skin friction drag, which are normal to and along the surface of a projectile respectively. The equation for the drag force is defined in terms of the drag coefficient, density of the fluid, surface area, and speed which the projectile is travelling:

$$F_D = \frac{1}{2} \rho V^2 A C_D \quad (2-19)$$

Lift is defined as the aerodynamic force that acts perpendicular to the velocity vector. This is the force that gives an object flight and can be defined as:

$$F_L = \frac{1}{2} \rho V^2 A C_L \quad (2-20)$$

These forces are taken into account in the simulation, and how they were used will be explained later in Section 5.2.

3. State Acquisition

It was necessary to measure the states of the hybrid projectile to be used in 3D localization. There are various sensors that can be used, but the most applicable are the inertial measurement unit (IMU), a Global Positioning System (GPS) receiver, and a magnetometer. Each are discussed more in depth below, covering how they work and why they are the most beneficial for this project.

3.1. Inertial Measurement Unit

An IMU consists of a triad of orthogonal accelerometers and rate gyroscopes that are continuously updated using a computer to calculate the acceleration and body angle rates of a body in space over time. (*Gross 2011*) By combining linear accelerometers with rate gyro measurements it is possible to monitor both the translational and rotational movements and completely define the trajectory of a projectile from its origin of motion, also known as dead reckoning. (*Titterton 2004*)

The first type of IMU that was developed was the gimballed system. (*Walchko 2002*) The accelerometers were mounted on a motorized gimballed platform which always kept in alignment

with the navigation frame. Pickups are located on the outer and inner gimbals which keep track of the attitude of the stabilized platform relative to the vehicle on which the INS is mounted. There are several factors that make this system undesirable: bearings are non-frictionless, motors are not perfect, power is consumed to keep the platform aligned with the navigational frame, cost is high due to the need for high quality motors, slip rings and other mechanical parts, and recalibration is difficult and requires regular maintenance by certified personnel.

In order to reduce these drawbacks, the Strap-down INS was created. In a strap-down system, referenced from *Rogers 2003*, three accelerometers and three gyroscopes are mounted in orthogonal triads and rigidly attached to the vehicle body. Motions sensed by the gyros, i.e., $\omega_{i/b}^b$ are in coordinates fixed to the body. The body referenced accelerometer outputs, f^b , are transformed from the body to the navigation frame in the navigation computer using the C_b^k transformation matrix. This method overcomes the problems from the gimbaled system by reducing the size, cost, power consumption and complexity of the system, making IMUs ideal for this project.

Sources of error, such as bias and drift, present difficulties when using IMUs. These are the most devastating effects on accuracy; drift rate for the gyros and the accelerometer bias are small offsets that the IMU improperly reads and propagates through the measurements. The bias has a quadratic effect on the position derived from the IMU's accelerometer ($error = \frac{1}{2}bias * t^2$) and must be accounted for. The drift rate has a similar and equally substantial impact on the position of a system. If a drift is not properly accounted for, the IMU thinks it is rotating and the navigation equations will not properly account for gravity. (*Rogers 2003*)

Micro-electrical mechanical system (MEMS) gyroscopes, an example can be seen below in Figure 3, are used as part of the IMUs and have many advantages. (Woodman 2007) They are light weight, small in size and have a low power consumption to name a few advantages over mechanical and optical gyros. MEMS gyros have sources of error that need to be addressed, however.

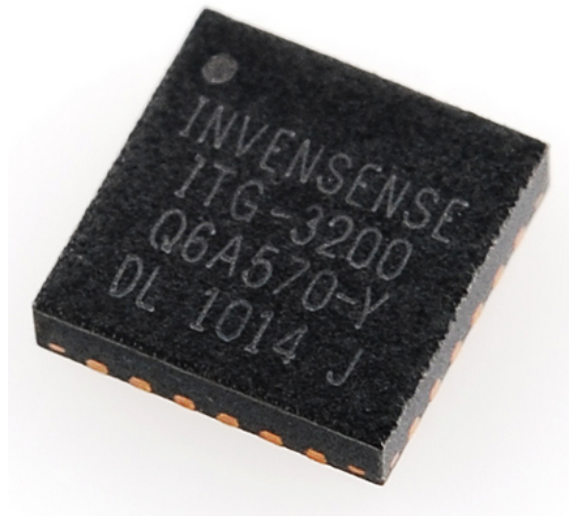


Figure 4: Triple Axis Digital Output Gyroscope- ITG-3200

The bias of a rate gyro is the average output from the gyroscope when it is not undergoing any rotation. A constant bias error, when integrated, causes an angular error which grows linearly with time ($error = bias * t.$) Thermo-mechanical white noise will also affect the output of the gyro sensor at a rate greater than the sampling rate of the sensor. As a result, the samples obtained from the sensor are perturbed by a white noise sequence which is simply a sequence of non-zero mean uncorrelated random variables. To see what effect the white noise has on the

integrated signal, a simple analysis can be done assuming the rectangular rule is used for integration. The result of this on the white noise signal, ε , is:

$$\int_0^t \varepsilon(\tau) d\tau = \delta t \sum_{i=1}^n N_i \quad (3-1)$$

where N_i is the i^{th} random variable in the white noise sequence, n is the number of samples received from the device during the period and δt is the time between successive samples. Calculating the mean, E , and variance, Var , of the equation above yields:

$$E\left(\int_0^t \varepsilon(\tau) d\tau\right) = \delta t * n * E(N) = 0 \quad (3-2)$$

$$\text{Var}\left(\int_0^t \varepsilon(\tau) d\tau\right) = \delta t^2 * n * \text{Var}(N) = \delta t * t * \sigma^2 \quad (3-3)$$

Hence, the noise introduces a zero-mean random walk error into the integrated signal, whose standard deviation is:

$$\sigma_{\theta}(t) = \sigma * \sqrt{\delta t * t} \quad (3-4)$$

which grows proportionally to the square root of time.

The drift rates and accelerometer biases also tend to change each time the unit is switched on. Typically there is a low pass filter used to remove some of this noise before the measurements

are used in the navigation equations. (Rogers 2003) Also realistically, there tends to be low pass filtering somewhere in the system due to hardware limitation because not everything has infinite bandwidth. As explained above, when random noise is filtered, this produces a random walk. Again, the integration of this random walk will result in velocity and positions moving at different rates during different runs even though the IMU and vehicle are in the same orientation and experiencing the same accelerations during each run. In this project the noise experienced from the IMU was filtered out through Kalman filtering. An introduction to Kalman filtering can be found in Chapter 4, and the specific process of the noise filtering will be explained in Chapter 6.

3.2. Global Positioning System

The Global Positioning System (GPS) system utilizes the concept of one-way time of arrival (TOA) ranging. (Kaplan 2006) The navigation data from satellites provides the means for the receiver to determine the location of the satellite at the time of signal transmission, whereas the ranging code enables the user's receiver to determine the transit time of the signal and thereby determine the satellite-to-user range. In order for this to work properly, this technique requires that the user receiver also contain a clock. Range information from multiple GPS satellites (at least four) provides information that is used to calculate the three-dimensional position of the GPS receiver within a Cartesian coordinate system that is within either a rotating frame or an inertial frame. (Gross 2011) The common designation of a Cartesian frame that rotates with the Earth's rotation is Earth-Centered Earth Fixed (ECEF), while an inertial frame is referred to as Earth-Centered Inertial (ECI).

Since the GPS receiver senses the position and velocity states directly, it does not need information about previous states in order to produce a solution. (*Theil*) The GPS receiver is considered an absolute sensor with bounded error in time. This feature gives long operation time stability, which a dead reckoning sensor like the IMU does not experience. The drawback is that it cannot provide a fast navigation solution, unlike the IMU, because of its low sample rate and the requirement that at least 4 GPS signals are available. Therefore, although the advantage of being a stand-alone sensor with long operation time stability, it also has associated errors due to the path of the satellite signals to the receiver.

Primary GPS error sources consist of measurement noise, propagation delay and non-synchronous clocks. Of these sources, the largest source of error is clock offset, or bias attributed to actual clock bias and ionosphere delays. Specifically, satellite and receiver clock offsets directly translate into pseudo range and carrier-phase errors. A GPS signal was simulated in Simulink using the model calculated position and velocities and noise was also added to the signal. This will be expanded on in Section 5.3.1.2.

3.3. Magnetometer

Magnetometers are devices that use the Earth's magnetic field to measure the orientation of a body in space. Using this data, the azimuth (heading) and spin rate of an aerial vehicle can be determined. *Jagadish (2007)* outlines how IMU measurements can be coupled with magnetometer measurements by using their projections of gravity and magnetic field of the Earth on an aerial vehicle to derive its Euler angles.

4. State Estimation Methods

Various methods can be used to estimate the states of a system that are provided by measurements. One method is linear Kalman filtering, which as the name suggests can only be applied to linear systems. For nonlinear systems, extended Kalman filtering and Unscented Kalman filtering can be applied to estimate the states.

4.1. Linear Kalman Filter

In *Kalman 1960*, the recursive solution to the discrete data linear filtering problem is introduced. The Kalman Filter (KF) has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation (*Welch 2006*). The KF, commonly employed by control engineers and other physical scientists has been successfully used in such diverse areas as the processing of signals in aerospace tracking and underwater sonar, and the statistical control of quality (*Meinhold 1983*). As mentioned earlier with IMU and GPS components, noise is a factor when measuring the system; Kalman filters provide engineers and scientists with a way to reduce or remove the noise. In order to use a KF to remove noise from a signal, the process that we are measuring must be able to be described by a linear system (*Simon 2001*).

The Kalman filter addresses the general problem of trying to estimate the state $x \in R^n$ of a discrete time controlled process that is governed by the linear stochastic difference equation

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (4-1)$$

With a measurement

$$z_k = Hx_k + v_k \quad (4-2)$$

The random variables w_k and v_k represent the process and measurement noise respectively. They are assumed to be independent of each other white noise with normal probability distributions.

$$p(w) \sim N(0, Q) \quad (4-3)$$

$$p(v) \sim N(0, R) \quad (4-4)$$

In practice, the process noise covariance Q and measurement noise covariance R matrices might change with each time step or measurement.

According to the work of Welch and Bishop, the KF estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of noisy measurements. As such, the equations for the KF fall into two groups: time update equations and measurement update equations. The time update equations are responsible for projecting forward in time the current state and error covariance estimates to obtain the *a priori* estimates for the next time step. The measurement update equations are responsible for the feedback. The time update equations can also be thought of as predictor equations, while the measurement update equations can be thought of as corrector equations. The final estimation algorithm resembles that of a *predictor-corrector* algorithm for solving numerical problems as shown in Figure 5 below.

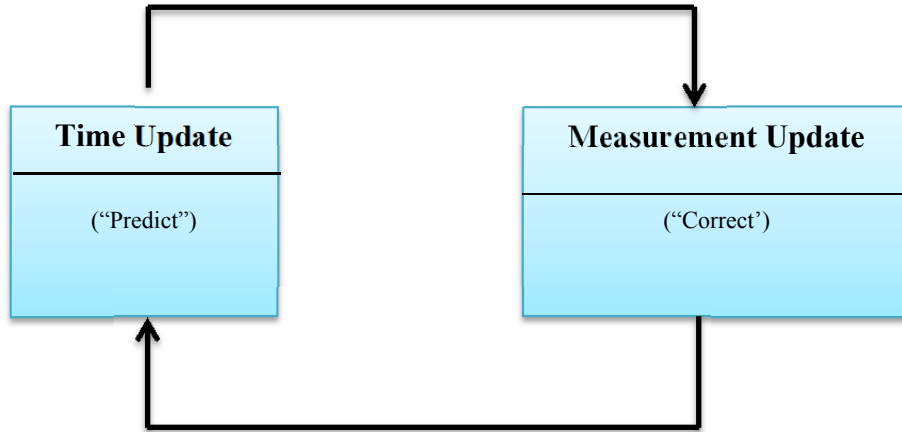


Figure 5: Kalman filter cycle

The specific equations for the time and measurement updates are:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (4-5)$$

$$P_k^- = A_k P_{k-1} A_k^T + Q \quad (4-6)$$

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (4-7)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-) \quad (4-8)$$

$$P_k = (I - K_k H) P_k^- \quad (4-9)$$

The first task during the measurement update is to compute the Kalman gain K_k . The next step is to actually measure the process to obtain z_k , and then to generate an *a posteriori* state estimate by incorporating the measurement as seen in Equation 4-8. The final step is to obtain an *a posteriori* error covariance estimate, seen in Equation 4-9. After each time and measurement update pair, the process is repeated with the previous *a posteriori* estimates used to project or predict the new *a priori* estimates. This recursive nature is one of the very appealing features of the KF.

In the actual implementation of the filter, the measurement noise covariance R is usually measured prior to operation of the filter. Measuring the measurement error covariance R is possible in general because the process is measured while the filter is in operation. Offline samples can be taken in order to determine the measurement noise variance. The determination of the process noise covariance Q is generally more difficult because it is not always possible to directly observe the process being estimated. Superior filter performance can be obtained by tuning the filter parameters Q and R . The tuning process can be seen below in Figure 6.

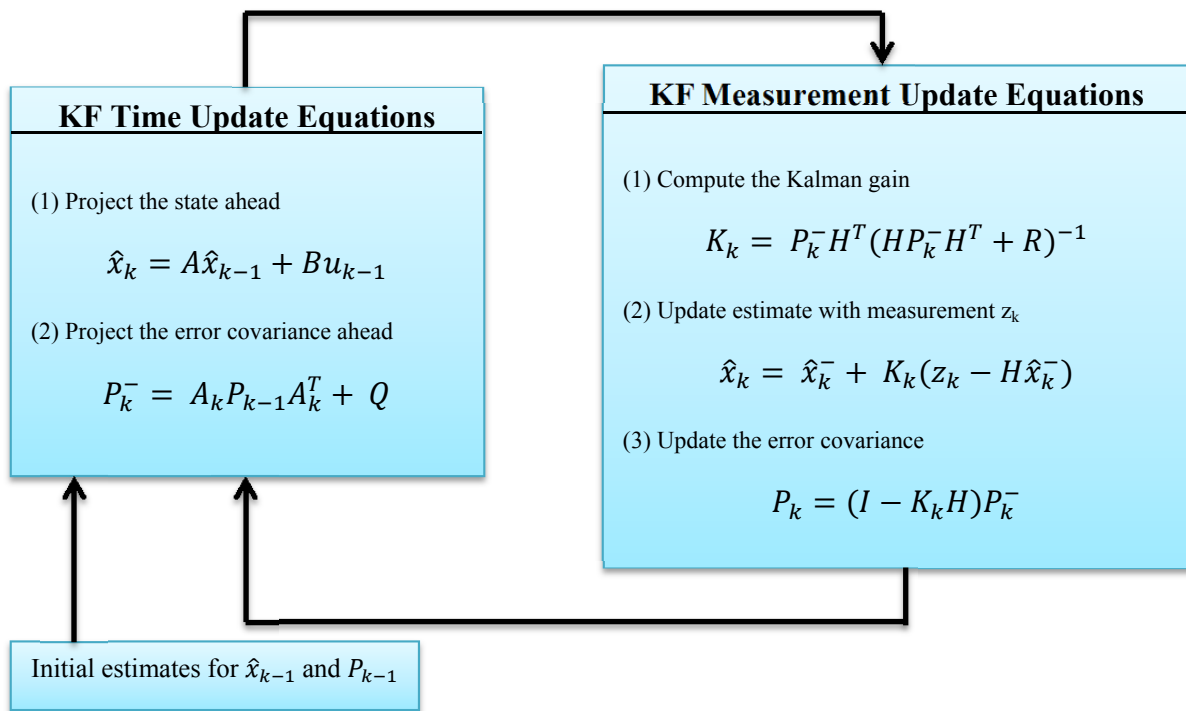


Figure 6: Kalman filter Process Flow Chart

It should be noted that under conditions where R and Q are constant, both the estimation error covariance P_k and the Kalman gain K_k will stabilize quickly and then remain constant. It is frequently the case, however, that the measurement error does not remain constant. Also, the

process noise Q is sometimes changed dynamically during filter operation, becoming Q_k in order to adjust for different dynamics. In such cases, Q_k might be chosen to account for uncertainty about the user's intentions and uncertainty in the model.

A linear Kalman filter was not used because as the name implies, it can only be applied to linear systems. A 6 degree of freedom body in space has equations of motion that are nonlinear. According to *Ribeiro (2004)*, if either the system state dynamics or the observation dynamics is nonlinear, the conditional probability density functions that provide the minimum mean-square estimate are no longer Gaussian. The optimal non-linear filter propagates these non-Gaussian functions and evaluates their mean, which represents a high computational burden. In order to avoid this pitfall, an Extended Kalman filter was used and is explained in the next section.

4.2. Extended Kalman Filter

An Extended Kalman filter (EKF) can be used to estimate state values of a nonlinear system, such the equations of motion of a flight system (*Welch 2006*). The EKF works by predicting the current states and then comparing them to actual measurements. Based on the error between the two, the Kalman gain adjusts the predictions accordingly to match the true values of the state. This is done by using the Jacobian matrices of the equations of motion to predict the states and measurements from the previous estimate. The basic function of the EKF can be seen in the equations below where x represents the states, Q and R represent noise covariance, P represents the error covariance, K represents the Kalman gain and A and H represent the Jacobian matrices. (*Welch 2006*)

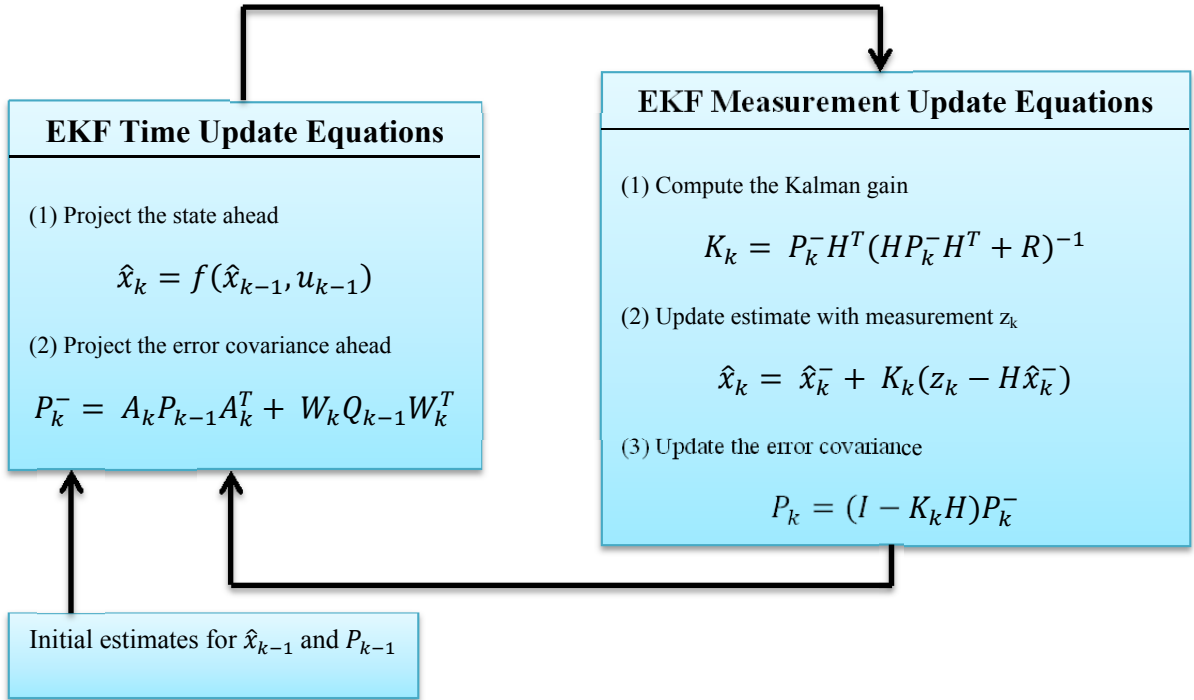


Figure 7: EKF Process Flow Chart

The general process equations that were used for the EKF can be found in *Lozano 2010*. In order to apply the EKF technique to the system, it must first be represented in state space form but a set of first-order nonlinear equations:

$$\dot{x} = f(x, w) \quad (4-10)$$

Where $x \in \mathbb{R}^n$ denotes the system states vector, $f(x)$ is the nonlinear function of the states, and $w \in \mathbb{R}^n$ represents a zero mean random process. For this project the nonlinear function of the states is the set of motion equations of the body angle rates seen above. The matrix of the process noise $Q \in \mathbb{R}^{n \times n}$ is given by

$$Q = E(ww^T) \quad (4-11)$$

The measurement equation is considered to be a nonlinear function of the states according to

$$z = h(x, v) \quad (4-12)$$

Where $v \in \mathbb{R}^m$ is a zero mean random process described by a matrix of measurement noise $R \in \mathbb{R}^{m \times m}$ as:

$$R = E(vv^T) \quad (4-13)$$

For systems with discrete-time measurements, it is possible to rewrite the nonlinear equation of measurements as:

$$z_k = h(x_k, v_k) \quad (4-14)$$

As the system and measurement equations are nonlinear, then it is necessary to linearize them by a first order approach to obtain the dynamic matrix of system F and the measurement matrix H . These matrices are related to the nonlinear equations according to

$$F = \frac{\partial f(x)}{\partial x} \quad (4-15)$$

$$H = \frac{\partial h(x)}{dx} \quad (4-16)$$

The fundamental matrix can be approximated by the Taylor expansion series as follows:

$$\Phi_k = I + FT_s + \frac{F^2T_s^2}{2!} + \frac{F^3T_s^3}{3!} + \dots \quad (4-17)$$

Where T_s is the sampling time and I is the identity matrix. The series often are approximated by the first two terms, that is,

$$\Phi_k \approx I + FT_s \quad (4-18)$$

For liner systems, the matrix F , H and Φ are linear, but, in the EKF, these matrices can be nonlinear. However, the Riccati equations for calculating the Kalman gain are identical to those seen in Figure 7.

The EKF will be applied to estimate the Euler angles of the hybrid projectile, while reducing noise. Equations of motion are required to measure the states of the hybrid projectile at any moment in time. The equations of motion, from *Roskam (2003)*, used to represent the body angle rates can be seen below where ϕ is roll, θ is pitch, ψ is yaw, and p , q and r are the respective rates of each body angle.

$$\dot{\phi} = p + (q * \sin\phi + r * \cos\phi)\tan\theta \quad (4-19)$$

$$\dot{\theta} = q * \cos\phi - r * \sin\phi \quad (4-20)$$

$$\dot{\psi} = (q * \sin\phi + r * \cos\phi) \sec\theta \quad (4-21)$$

The above equations can be used to represent the state at the next time step in a discretized setting by taking the previous state and adding the change of the state at that particular time.

$$\phi_{k+1} = \phi_k + p_k + (q_k * \sin\phi_k + r * \cos\phi_k) \tan\theta_k \quad (4-22)$$

$$\theta_{k+1} = \theta_k + q_k * \cos\phi_k - r_k * \sin\phi_k \quad (4-23)$$

$$\psi_{k+1} = \psi_k + (q_k * \sin\phi_k + r_k * \cos\phi_k) \sec\theta_k \quad (4-24)$$

4.3. Unscented Kalman Filter

Other types of linear Kalman filters can be used for estimation of nonlinear systems. While the EKF is effective as a nonlinear state estimator, the Unscented Kalman filter (UKF) was developed as an improvement. In reference to *Wan (2000)*, several flaws are pointed out with the EKF and the UKF process is explained in detail. The EKF calculates the prior state estimates exactly in the linear case, and can be viewed as an efficient method for analytically propagating a Gaussian random variable (GRV) through linear system dynamics. For nonlinear models, however, the EKF approximates the optimal terms, where the predictions are approximated as simply the function of the prior mean value for estimates. The covariance is determined by linearizing the dynamic equations and then determining the posterior covariance matrices analytically for the linear system. In other words, in the EKF the state distribution is approximated by a GRV which is propagated analytically through the first order linearization of the nonlinear system, and the EKF can be viewed as providing “first-order” approximations to the optimal terms. These approximations, however, can introduce large errors in the true

posterior mean and covariance of the transformed GRV, which may lead to sub-optimal performance and sometimes divergence of the filter. The UKF avoids these pitfalls by approximating the probability density resulting from the non-linear transformation of a random variable instead of approximating the nonlinear functions with a Taylor series expansion. (*St-Pierre 2004*)

According to St-Pierre, the approximation is done by evaluating the nonlinear function with a minimal setoff carefully chosen sample points. The posterior mean and covariance estimated from the sample points are accurate to the second order for any nonlinearity. (*Van der Merwe 2000*) If the priori random variable is Gaussian, the posterior mean and covariance are accurate to the third order. (*Wan, 2000*) Wan defines an unscented transformation (UT) as a method for calculating statistics of a random variable which undergoes a nonlinear transformation. Consider propagating a random variable x (dimension L) through a nonlinear function, $y = g(x)$. Assume x has a mean \bar{x} and covariance P_x . To calculate the statistics of y , we form matrix χ of $2L + 1$ sigma vectors χ_i (with corresponding weights W_i), according to the following:

$$\chi_0 = \bar{x} \quad (4-25)$$

$$\chi_i = \bar{x} + (\sqrt{(L + \lambda P_x)})_i \quad i = 1, \dots, L \quad (4-26)$$

$$\chi_i = \bar{x} - (\sqrt{(L + \lambda P_x)})_{i-L} \quad i = L + 1, \dots, 2L \quad (4-27)$$

$$W_0^{(m)} = \lambda / (L + \lambda) \quad (4-28)$$

$$W_0^{(c)} = \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta) \quad (4-29)$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{\{2(L + \lambda)\}} \quad i = 1, \dots, 2L \quad (4-30)$$

Where $\lambda = \alpha^2 (L + \kappa) - L$ is a scaling parameter. α determines the spread of the sigma points around \bar{x} and is usually set to a small positive value (e.g., $1 \text{ e-}3$). κ is a secondary scaling parameter which is usually set to 0, and β is used to incorporate prior knowledge of the distribution of x (for Gaussian distributions, $\beta = 2$ is optimal). $(\sqrt{(L + \lambda P_x)})_i$ is the i th row of the matrix square root. These sigma vectors are propagated through the nonlinear function,

$$Y_i = g(\chi_i) \quad i = 0, \dots, 2L \quad (4-31)$$

And the mean and covariance for y are approximated using a weighted sample mean and covariance of the posterior sigma points,

$$\bar{y} \approx \sum_{i=0}^{2L} W_i^{(m)} Y_i \quad (4-32)$$

$$P_y \approx \sum_{i=0}^{2L} W_i^{(c)} \{Y_i - \bar{y}\} \{Y_i - \bar{y}\}^T \quad (4-33)$$

Note that this method differs substantially from general “sampling” methods (e.g., Monte-Carlo methods such as particle filters (*de Freitas 1998*) which require orders of magnitude more sample points in an attempt to propagate an accurate (possibly non-Gaussian) distribution of the state. The simple approach taken with the UT results in approximations that are accurate to the third order for Gaussian inputs for all nonlinearities. The Unscented Kalman Filter (UKF) is a straightforward extension of the UT to the recursive estimation $\hat{x}_k = (\text{prediction of } x_k) + \kappa_k *$

$[y_k - (\text{prediction of } y_k)]$, where the state RV is redefined as the concatenation of the original state and noise variables: $x_k^a = [x_k^T v_k^T n_k^T]^T$. The UT sigma point selection scheme (Equations 34-39) is applied to this new augmented state RV to calculate the corresponding sigma matrix, χ_k^a . The equations used by the UKF are:

Initialize with:

$$\hat{x}_o = E[x_o] \quad (4-34)$$

$$P_o = E[(x_o - \hat{x}_o)(x_o - \hat{x}_o)^T] \quad (4-35)$$

$$\hat{x}_o^a = E[x^a] = [\hat{x}_o^T \ 0 \ 0]^T \quad (4-36)$$

$$P_o^a = E[(x_o^a - \hat{x}_o^a)(x_o^a - \hat{x}_o^a)^T] = \begin{bmatrix} P_o & 0 & 0 \\ 0 & P_v & 0 \\ 0 & 0 & P_n \end{bmatrix} \quad (4-37)$$

For $k \in \{1, \dots, \infty\}$,

Calculate the sigma points:

$$\chi_{k-1}^a = [\hat{x}_{k-1}^a \ \hat{x}_{k-1}^a \pm \sqrt{(L + \lambda)P_{k-1}^a}] \quad (4-38)$$

The time update equations are:

$$\chi_{k|k-1}^x = F[\chi_{k-1}^x, \chi_{k-1}^v] \quad (4-39)$$

$$\hat{x}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \chi_{i,k|k-1}^x \quad (4-40)$$

$$P_k^- = \sum_{i=0}^{2L} W_i^{(c)} [\chi_{i,k|k-1}^x - \hat{x}_k^-][\chi_{i,k|k-1}^x - \hat{x}_k^-]^T \quad (4-41)$$

$$Y_{k|k-1} = H[\chi_{k|k-1}^x, \chi_{k-1}^n] \quad (4-42)$$

$$\hat{y}_k^- = \sum_{i=0}^{2L} W_i^{(m)} Y_{i,k|k-1} \quad (4-43)$$

And the measurement update equations are:

$$P_{\bar{y}_k \bar{y}_k} = \sum_{i=0}^{2L} W_i^{(c)} [Y_{i,k|k-1}^x - \hat{y}_k^-][Y_{i,k|k-1}^x - \hat{y}_k^-]^T \quad (4-44)$$

$$P_{x_k y_k} = \sum_{i=0}^{2L} W_i^{(c)} [\chi_{i,k|k-1}^x - \hat{x}_k^-][Y_{i,k|k-1}^x - \hat{y}_k^-]^T \quad (4-45)$$

$$\kappa = P_{x_k y_k} P_{\bar{y}_k \bar{y}_k}^{-1} \quad (4-46)$$

$$\hat{x}_k = \hat{x}_k^- + \kappa(y_k - \hat{y}_k^-) \quad (4-47)$$

$$P_k = P_k^- - \kappa P_{\bar{y}_k \bar{y}_k} \kappa^T \quad (4-48)$$

Where $x^a = [x^T \ v^T \ n^T]^T$, $\chi^a = [(\chi^x)^T \ (\chi^v)^T \ (\chi^n)^T]^T$, $\lambda =$ composite scaling parameter, $L =$ dimension of augmented state, $P_v =$ process noise covariance, $P_v =$ measurement noise covariance, $W_i =$ weights as calculated in Equations 4-25 through 4-30. In this project a UKF was not used because it is too computationally expensive and therefore not as the EFK for the desired hardware. According to Van der Merwe, the most computationally expensive operation in the UKF corresponds to calculating the new set of sigma points at each time step update. At the speed that the hybrid projectile travels, faster computations are more accurate and provide better performance.

5. Model Construction

In order to determine whether a body angle was a viable option over the timer deployment condition, a model was constructed for simulation. The projectile aerodynamics model was built around the ‘6DoF (Euler Angles)’ block, the first option in the 6DoF equations of motion from the aerodynamics toolbox in Simulink. It uses aerodynamic coefficients and mass properties to simulate flight (*Wilhelm 2012*). In this thesis, assumptions were made that eliminated two degrees of freedom, which will be outlined in Section 5.2. Mass, aerodynamic and physical properties of the round first have to be initialized in order for the model to be able to function properly. The highest level of the model, seen in Figure 8 output, all of the speed and angle simulated data for the HP. The forces and moments acting on the HP were calculated and updated with each time step using the ‘Calc Forces’ block. The forces and moments were then fed into the 6DoF Euler Angles block, which calculated the states of the HP. The states, earth reference frame velocity, earth reference frame position, Euler angles, Direct Cosine Matrix (DCM), and body velocity, were then fed into both the ‘Graphing’ and ‘Sensor State Estimation’ blocks. The ‘Graphing’ block was used to create graphs, visualize the velocity, trajectory and body angles of the HP, and output the Euler angles to the ‘Sensor State Estimation’ block. It also checked for a condition that determined when to terminate the simulation. In the ‘Sensor State Estimation’ block and IMU is simulated and EKF is applied to the HP.

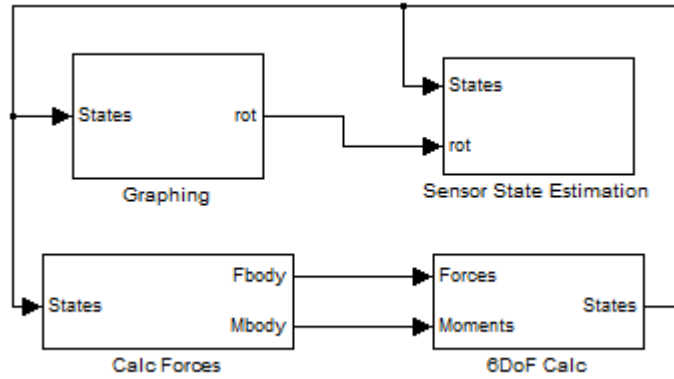


Figure 8: Hybrid Projectile Simulink Model

5.1. Euler Angles 6 DoF Block

The 6 DoF (Euler Angles) block was used from the aerodynamics toolbox in Simulink. It outputs the earth velocity, V_e (m/s), and position, X_e (m), of the HP as well as body angles, the body-to-Earth Direct Cosine matrix DCM_{be} , body velocity, V_b (m/s), body angle rates, ω (rad/s), body angular accelerations, $\frac{d\omega}{dt}$ (rad/s²), and body accelerations, A_b (m/s²). Inputs to the block were the body forces and moments in the XYZ reference frame, seen below in Figure 9. Positive axis orientation can be thought of as “North-East-Down” relative to the HP. It should be noted that the reference frame in the figure below is a generalization; the actual reference frame is placed at the Center of Gravity (CG).

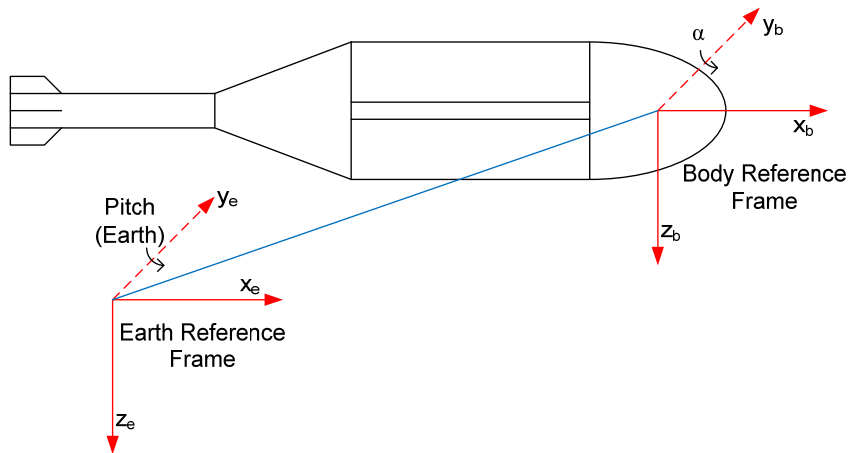


Figure 9: Reference Frame of the HP (Wilhelm 2012)

The process of calculating all of the states that were listed above can be seen below in the 6 DoF subsystem block diagram in Figure 10. This is the default subsystem for the 6DoF (Euler Angles) block in Simulink, and cannot be modified. Each subsystem within the block will be described and the process of finding each of the outputs will be explained.

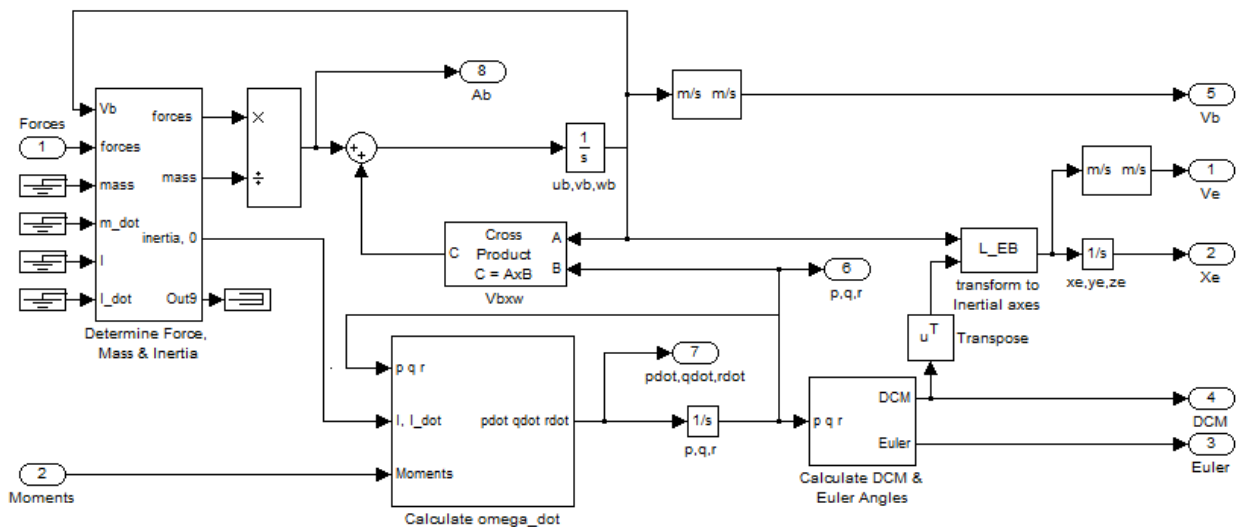


Figure 10: Subsystem of the 6 DoF Euler Angles block

The equations of motion of the 6DoF block that were defined can be found in the ‘Help’ section of Simulink and they can be seen below. The applied forces $[F_x F_y F_z]^T$ are in the body-fixed frame, and the mass of the body m is assumed constant.

$$\bar{F}_b = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m(\bar{V}_b + \bar{\omega} \times \bar{V}_b) \quad 5-1$$

$$\bar{V}_b = \begin{bmatrix} u_b \\ v_b \\ w_b \end{bmatrix}, \bar{\omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad 5-2$$

The applied moments are $[L M N]^T$, and the inertia tensor I is with respect to the origin O .

$$\bar{M}_B = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = I\bar{\omega} + \bar{\omega} \times (I\bar{\omega}) \quad 5-3$$

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \quad 5-4$$

The relationship between the body-fixed angular velocity vector, $[p q r]^T$, and the rate of change of the Euler angles, $[\dot{\phi} \dot{\theta} \dot{\psi}]^T$ can be determined by resolving the Euler angle rates into the body-fixed coordinate frame.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad 5-5$$

$$\stackrel{\text{def}}{=} J^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

Inverting J then gives the required relationship to determine the Euler angle rate vector:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = J \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad 5-6$$

The body accelerations, A_b (m/s^2), were calculated by dividing the forces by the mass, which was extracted from the initializing .m file. From here, the body accelerations were integrated and then output as body velocities, V_b (m/s). The body angle rates \dot{p} , \dot{q} , and \dot{r} (rad/s^2) were calculated in the ‘Calculate omega_dot’ block, and the process can be seen below in Figure 11. These rates are identical to the 6 DoF output $\frac{d\omega}{dt}$ (rad/s^2). Two moments due to the body angle rates were calculated by multiplying them by the inertia and rate of change of inertia. These two moments were then subtracted from the moments calculated in the ‘Calc Forces’ block. This resultant moment vector was divided by the inertia and reshaped to output the body angle accelerations. The body angle accelerations were then integrated to determine the body angle rates p , q and r (rad/s).

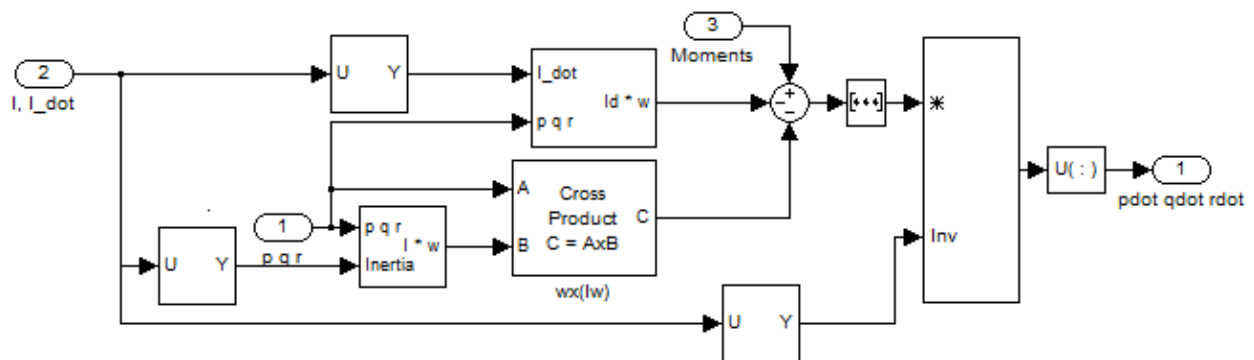


Figure 11: Calculate omega_dot subsystem

Next, the body-to-Earth DCM (DCM_{be}) and Euler angles φ , θ and ψ were calculated in the ‘Calculate DCM & Euler Angles’ block seen in Figure 12. The basic function of this block was to apply the equations of the DCM, using p , q , r , φ , θ and ψ . Initial Euler angles were defined in the .m file to allow the DCM to be calculated on the first time step. The equations for the DCM are:

$$DCM = \begin{bmatrix} \cos\varphi\cos\psi + \sin\varphi\sin\psi\sin\theta & -\cos\theta\sin\varphi & \cos\psi\sin\varphi\sin\theta - \cos\varphi\sin\psi \\ \cos\psi\sin\varphi - \cos\varphi\sin\psi\sin\theta & \cos\varphi\cos\theta & -\sin\varphi\sin\psi - \cos\varphi\cos\psi\sin\theta \\ \cos\theta\sin\psi & \sin\theta & \cos\psi\cos\theta \end{bmatrix} \quad 5-7$$

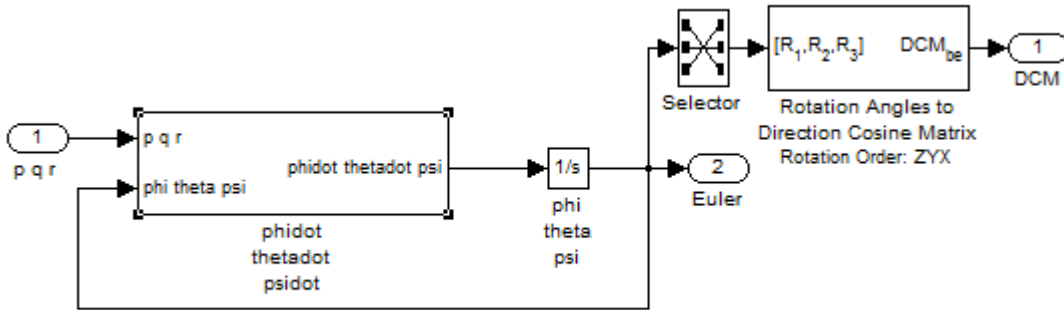


Figure 12: Calculate DCM & Euler Angles subsystem

The values for p , q and r were fed into the ‘phidot, thetadot, psidot’ block which calculated the equations of motion as laid out in (Eq. whatever it is). They were then integrated to obtain the Euler angles, and the DCM was calculated and also output. A Selector was used reorder the Euler angle vector from φ , θ , ψ to facilitate a ZYX rotation. The global velocity vector was then found by taking the transpose of the newly found DCM. This was due to the DCM being calculated as a ZYX rotation, and the body to global transformation is calculated as an XYZ rotation. Once the velocity has been calculated, integration yields the global position vector.

5.2. Force Calculation Block

The ‘Calc Forces’ block, seen below in Figure 13, determined the forces and moments that act on the body. The inputs to the subsystem are the body velocity vector, body-to-Earth DCM, body angles, and global position of the HP. Determining whether or not wings are deployed was the most important function of this block. Choosing between body angle or timer deployment methods was carried out here, as well as modeling head wind later in the simulation. The CG and Center of Pressure (CP) were important aerodynamic properties that were defined in this block; they were located at $[0 \ 0 \ 0]$, or the exact center of the body. Air density was defined as 1.2 kg/m^3 , which corresponds to a height of 500 ft (152.4m) above sea-level.

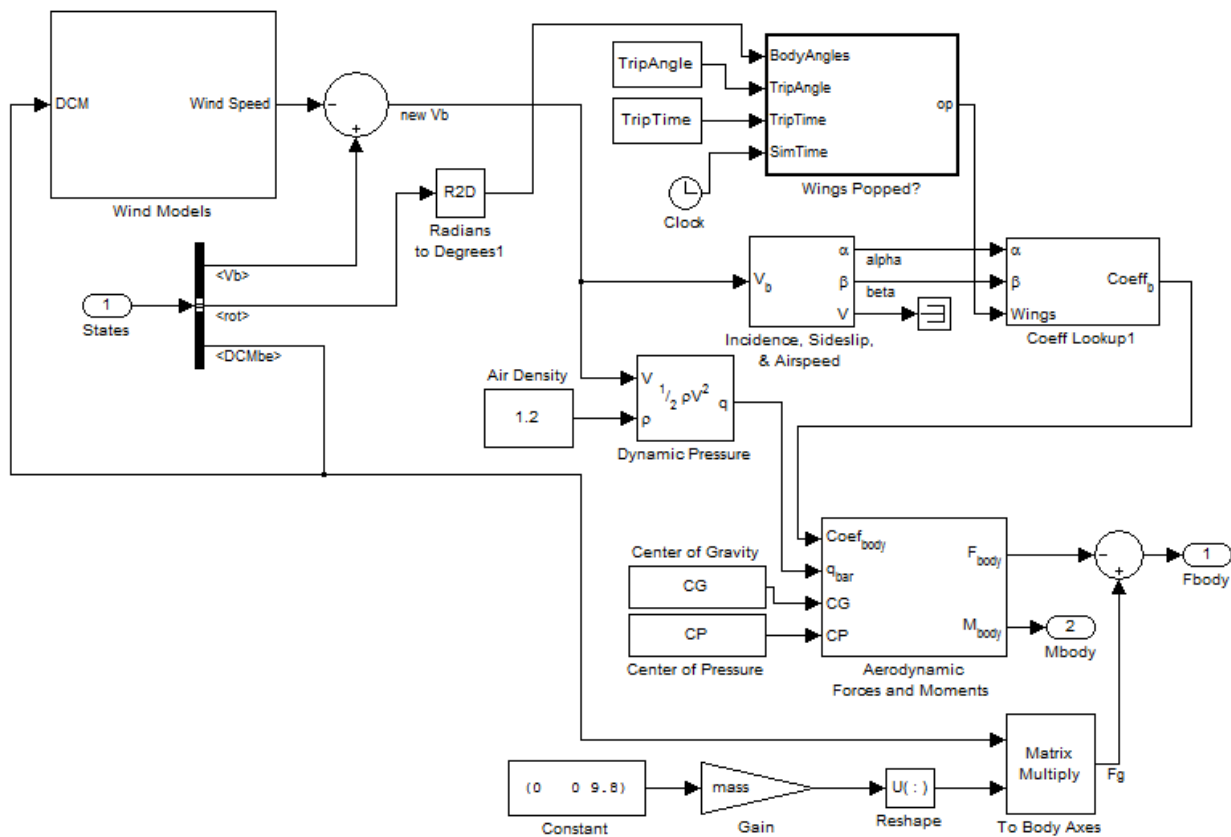


Figure 13: Calc Forces block subsystem

Forces and moments acting on the HP were found by using the ‘Aerodynamic Forces and Moments’ block from the aerodynamics toolbox in Simulink. The inputs to the block were the body coefficients, dynamic pressure, center of gravity (CG), and the center of pressure (CP). In order to determine the body coefficients, the model first had to determine if the wings deployed. Before determining which deployment condition was more effective, the model calculated if the wings were deployed using a body angle condition. Wind tunnel test data was collected for both conditions of deployed and non-deployed flight (*Browning 2011*), and the coefficients can be seen in Figure 14. However, the data was reference to the nose in this experiment, not the traditional chord and span of airplane aerodynamic data.

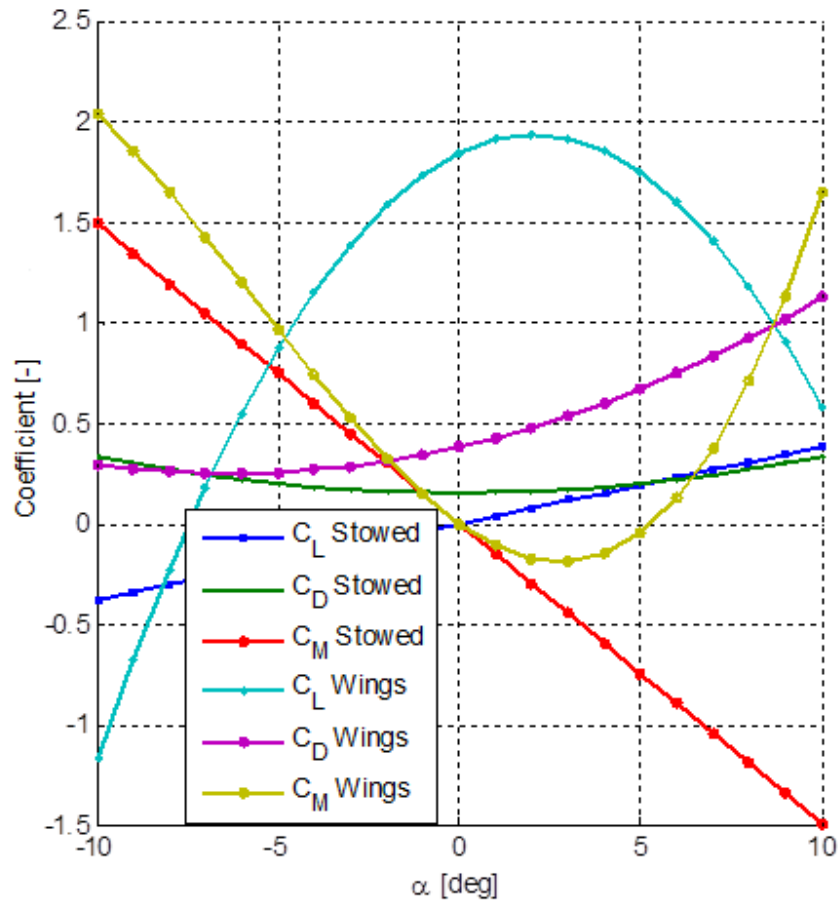


Figure 14: Aerodynamic Body Coefficients for the Hybrid Projectile (*Wilhelm 2012*)

Figure 15 below shows the process the model took to determine the body coefficients at each time step. Dynamic pressure was calculated at each time step, while the CG and CP are fixed values that were defined at the beginning of the simulation. Conditions of head wind during flight were also tested in this block; however, the effect of head wind will be discussed later in the Results section.

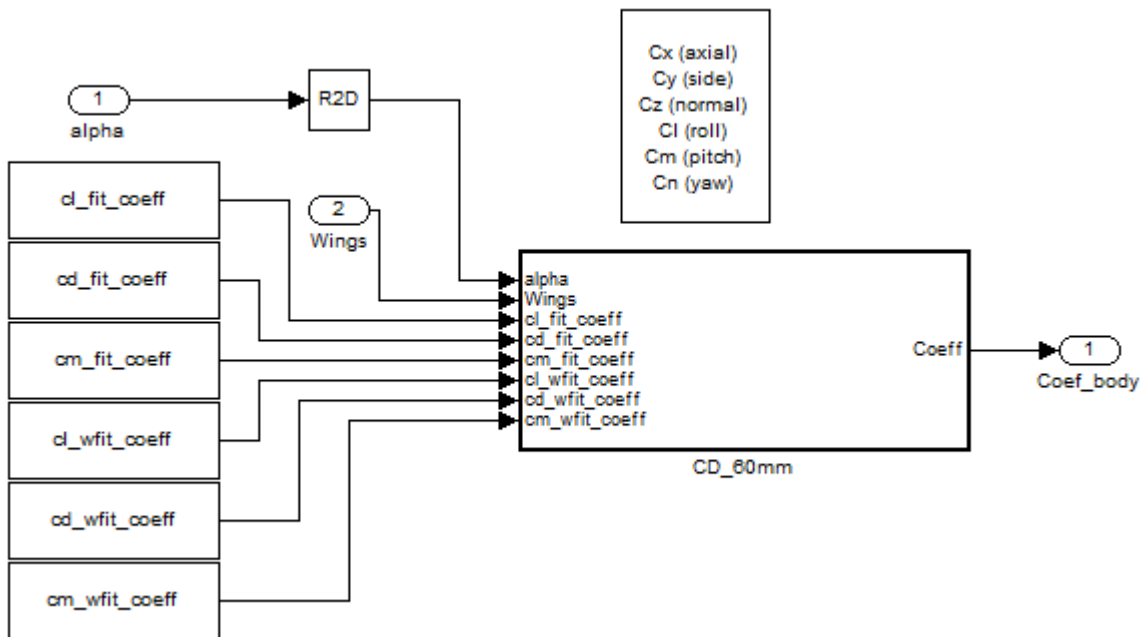


Figure 15: Body Coefficients Lookup subsystem

In order to determine the aerodynamic forces on the HP, aerodynamic coefficients had to be known at each instance during flight. Based on the wind tunnel data from *Browning (2011)*, fixed coefficients were used in this study according to the condition of the wings if they were stowed or deployed. The Aerodynamic Forces and Moments block, which provided the option to define the system axes with respect to wind, stability or body axes, determined which set of coefficients had to be used. The ‘body axes’ option was chosen, meaning that the axial (C_x), side

(C_y), normal (C_z), roll (C_l), pitch (C_m) and yaw (C_n) coefficients had to be defined. An important aerodynamic property involved in these calculations was the incidence angle, α and it was used to calculate C_x and C_z using the equations:

$$C_x = C_L * \sin(\alpha) + C_D * \cos(\alpha) \quad 5-8$$

$$C_z = C_L * \cos(\alpha) + C_D * \sin(\alpha) \quad 5-9$$

Where α is in radians and C_D and C_L were found by:

$$C_D = C_{D1} * \alpha^2 + C_{D2} * \alpha^2 + C_{D3} \quad 5-10$$

$$C_L = C_{L1} * \alpha^2 + C_{L2} * \alpha^2 + C_{L3} \quad 5-11$$

The lift and drag coefficients seen in Equations 5-10 and 5-11 above were determined based on the condition if the wings are stowed or deployed. The values for both conditions were:

$$C_{D,wings\ stowed} = \begin{bmatrix} 0.0018 \\ 0 \\ 0.152 \end{bmatrix} \quad 5-12$$

$$C_{D,wings\ deployed} = \begin{bmatrix} 0.0033 \\ 0.0414 \\ 0.379 \end{bmatrix} \quad 5-13$$

$$C_{L,wings\ stowed} = \begin{bmatrix} 0 \\ 0 \\ 0.038 \end{bmatrix} \quad 5-14$$

$$C_{L,wings\ deployed} = \begin{bmatrix} -0.0214 \\ 0.0876 \\ 1.843 \end{bmatrix} \quad 5-15$$

The pitch coefficient, C_m , was calculated in a similar manner using the equation:

$$C_m = C_{m_1} * \alpha^3 + C_{m_2} * \alpha^2 + C_{m_3} * \alpha + C_{m_4} \quad 5-16$$

where

$$C_{m,wings\ stowed} = \begin{bmatrix} 0 \\ 0 \\ -0.25 \\ 0 \end{bmatrix} \quad 5-17$$

$$C_{m,wings\ deployed} = \begin{bmatrix} 0.0011 \\ 0.01841 \\ -0.25 \\ 0 \end{bmatrix} \quad 5-18$$

Assumptions were made in the simulation that the round was not subjected to side forces, nor did it experience rolling or yawing moments. This was done to simplify the model, since pitch was the major body angle being analyzed in this thesis. Because of these assumptions, C_y , C_l , and C_n were set equal to zero. Realistically this would not be the case as the HP could leave the barrel of the launching tube already in a roll, but such a case will not be addressed in this thesis. The overall coefficient vector could then be represented by:

$$C_{overall} = \begin{bmatrix} C_x \\ 0 \\ C_z \\ 0 \\ C_m \\ 0 \end{bmatrix} \quad 5-19$$

Based on these coefficients, the performance of the HP can be seen below in Figure 16.

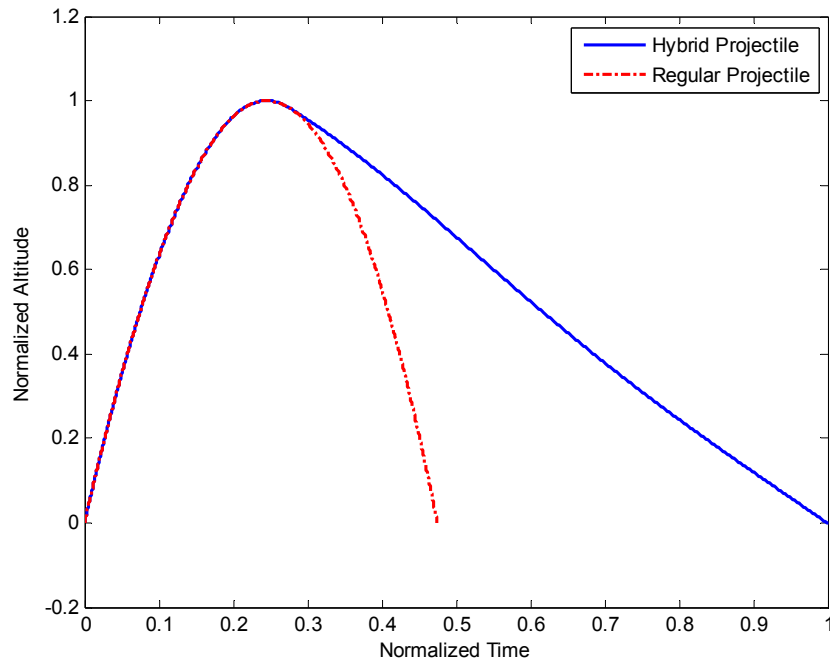


Figure 16: Trajectories of a HP, with a Wing Deployment Angle of -10° , and a Regular Projectile, both with a mass of 1.68 kg and a Launch Velocity of 126 m/s

Once the model was constructed, a simulation was conducted to determine the best conditions for maximum extended range. A wing deployment angle of -10° relative to the horizontal was used because the HP was designed to have a glide ratio of 10:1. An appropriate launch angle had to be determined that allowed maximum range extension to be achieved. A distance map of the HP with varying launch and wing deployment angles can be seen in Figure 17.

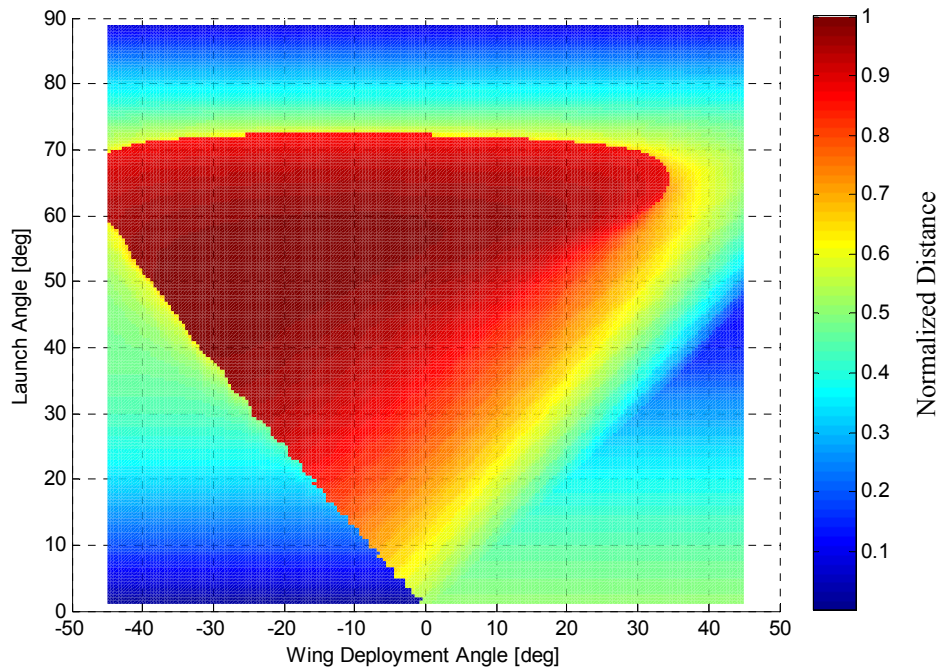


Figure 17: Distance Map for Extended Range of the HP

The region in dark red shows the possible combinations of launch and wing deployment angles for optimal extended range. A launch angle of 50° was chosen with the -10° wing deployment angle, which corresponded to a 98.77% maximum range extension. There was a region in the distance map where at least 98% range extension was achieved, which can be seen in Figure 18. This provides flexibility in both the wing deployment angle and launch angle should there need to be any changes in the design or manner of operation of the HP.

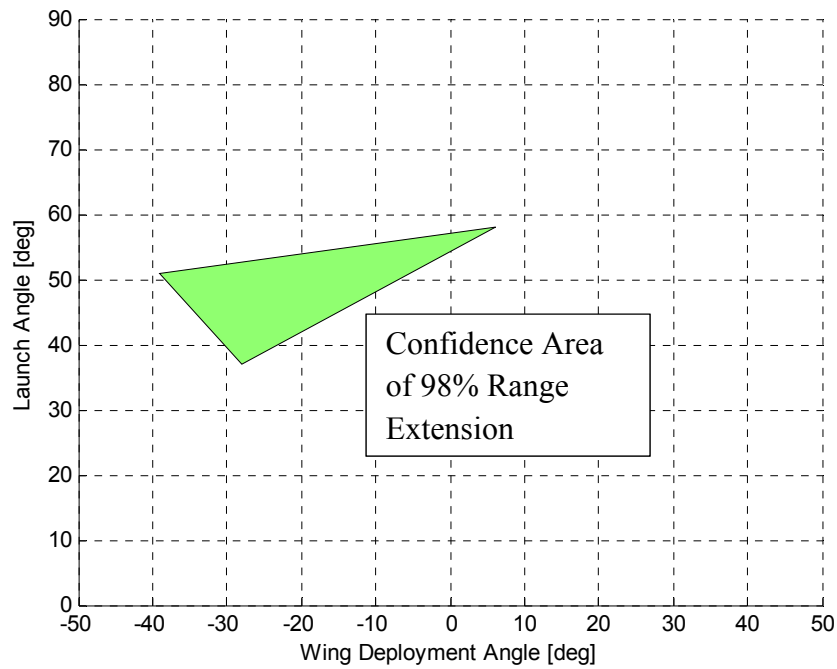


Figure 18: HP Distance Map Confidence Area of 98% (Wilhelm 2012)

Once the forces calculations were defined, the model was able to accurately determine the states of the HP. In order to better understand what the model was doing, a graphing block was created to visualize body angle, position and velocity values. The ‘Graphing’ block was created as a way to visualize the velocity, trajectory and Euler angles of the HP within the model, and can be seen below in Figure 19. Plotting velocity, distance and altitude were the main functions of the block as well as determining when to end the simulation. Once the altitude calculated by the model reached zero, Simulink terminated the simulation.

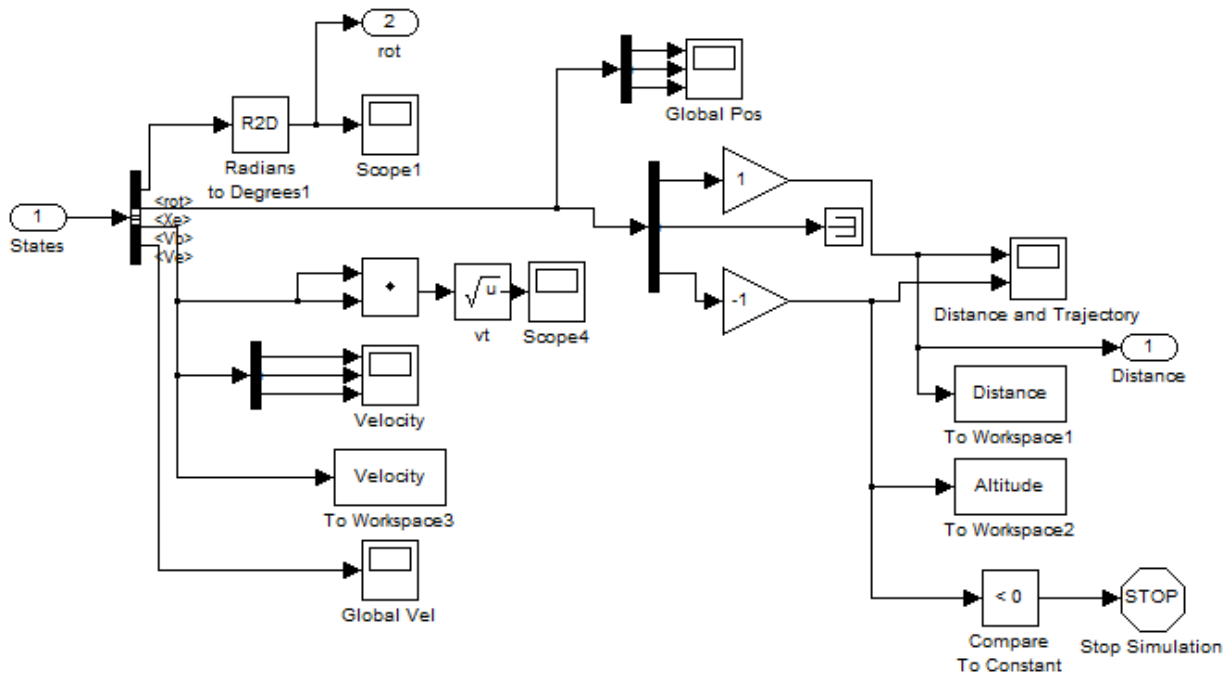


Figure 19: Graphing block subsystem

Its inputs were the Euler angles in radians (rot), global position in m (X_e), body velocity in m/s (V_b) and global velocity in m/s (V_e). In order to use a body angle wing deployment condition, the pitch had to be known during the HP's flights. The Euler angles were visualized by converting them from radians to degrees as seen in Figure 20. This was used as a benchmark for comparison later when the body angles were estimated by the EKF, and also provides an early visualization of how the HP is oriented during its flight.

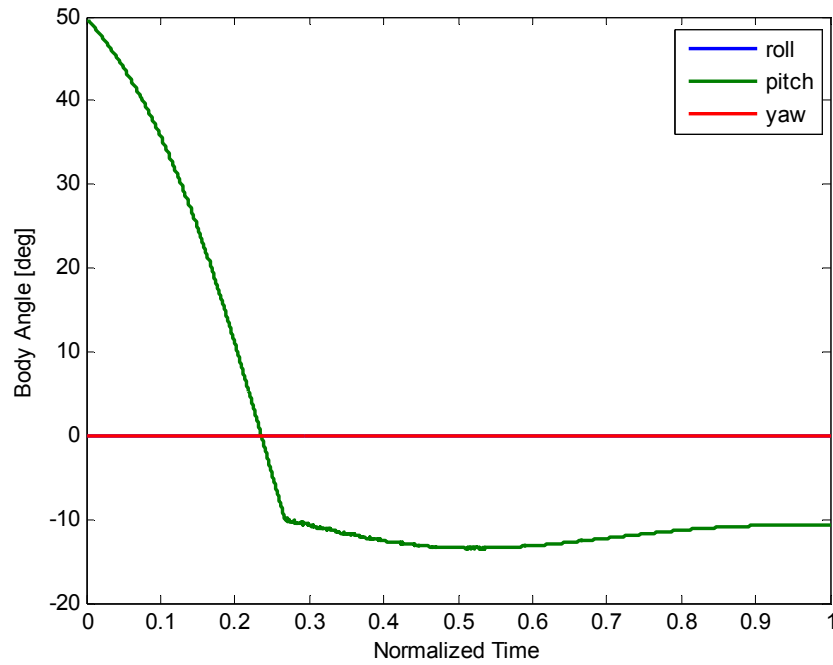


Figure 20: Model Calculated Euler Angles of a 1.68 kg HP with a 126 m/s Launch Velocity and Body Trip Angle of 10°

The distance and altitude plot can be seen below in Figure 21; however, the altitude signal had to be multiplied by -1 due to the reference frame definition of the positive z-direction as down, seen in Figure 9. The moment of wing deployment, around 0.3 on the x-axis, can clearly be detected on the plot of the HP’s trajectory in the top section of Figure 21, at which time the graph becomes almost linear.

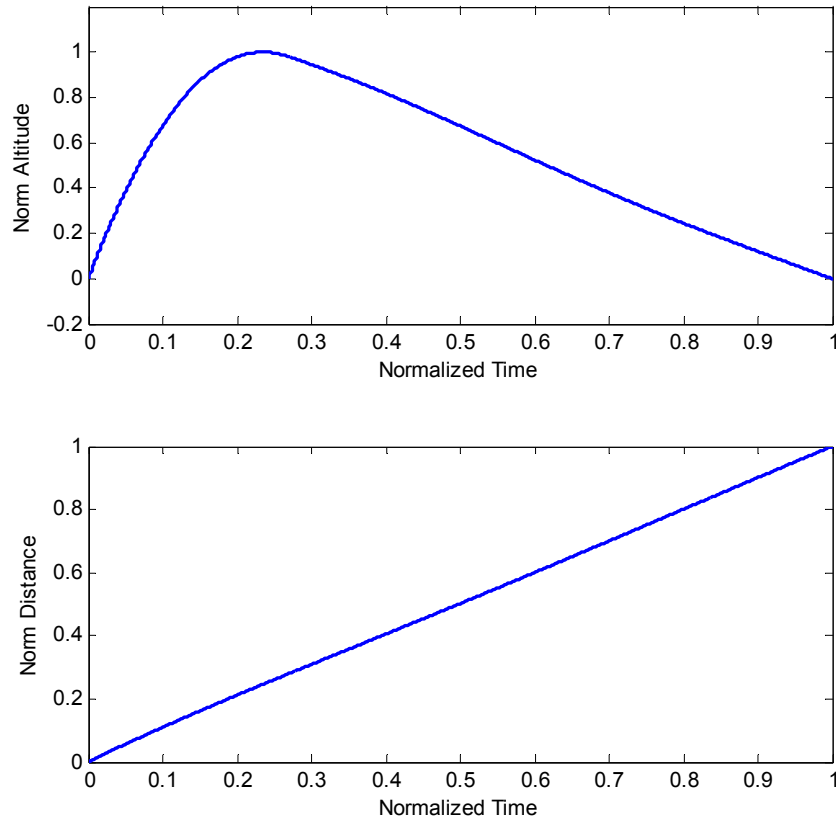


Figure 21: Altitude (top) and Distance (bottom) of a 1.68 kg HP with a 126 m/s Launch Velocity and Wing Deployment Body Angle of 10°

In order to better understand the effect of the HP's range extension capabilities, the maximum distances of a regular projectile and a hybrid projectile were plotted against each other. Launch angles between 0 and 90° were then chosen to test the HP's maximum range extension capabilities over regular projectiles, and its performance can be seen below in Figure 22.

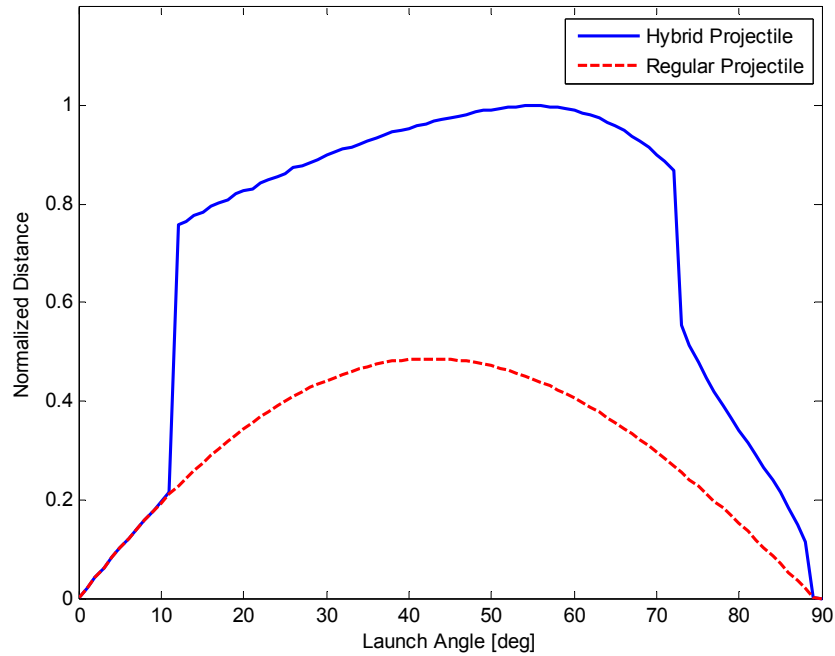


Figure 22: Maximum Distance vs. Launch Angle for a Hybrid Projectile with Wing Deployment Angle of -10° and a Regular Projectile given an Initial Velocity of 126 m/s

According to simulation results, the maximum range is achieved at a launch angle of 56° . The maximum distance between both projectile types is the same up until a launch angle of 10° because the wing deployment condition is never met. A significant increase in maximum distance occurs after a 10° launch angle, showing the effectiveness of the HP's glide capabilities. However, between launch angles of 72° and 73° the performance of the HP declined by 31.4%. The performance continued to drop significantly for launch angles greater than 73° , which means that at these conditions the wings are not as effective at extending range.

In aerospace systems equations of motion are often expressed in terms of the body velocities. They are expressed using the variables u , v , and w (m/s) and can be transformed using the Direct Cosine Matrix (DCM) to calculate global velocity. The global velocity vector can then be

integrated to calculate global position. Body velocities calculated by the model can be seen below in Figure 23. Because it was assumed that the HP did not experience any side forces, there was no y- velocity, v.

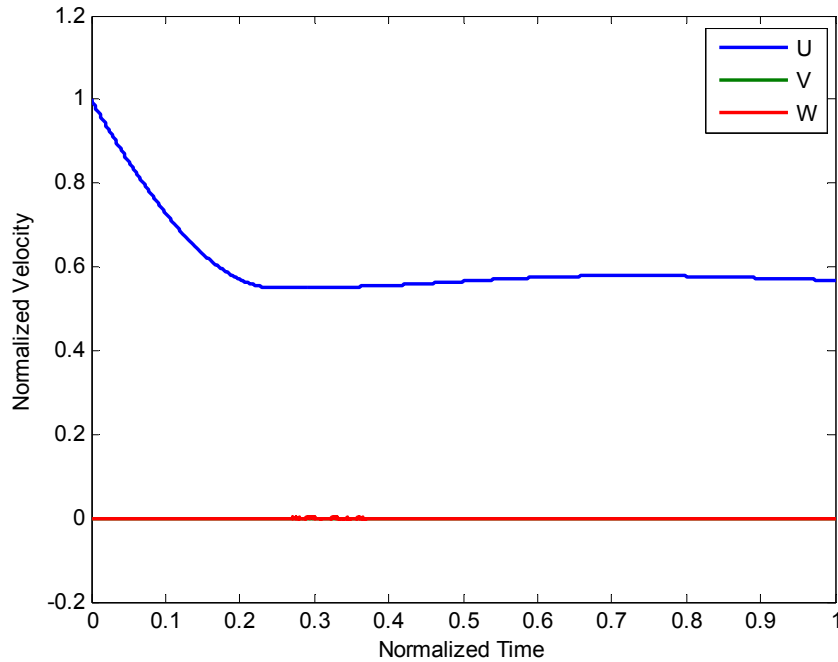


Figure 23: Model Calculated Body Velocities

The model calculated values for position, velocity, acceleration, body angles and body angle rates were then used as inputs to a block that simulated sensors on the IMU.

5.3. Sensor State Estimation Block

The ‘Sensor State Estimation’ block was where the majority of testing was done. Starting with the states already calculated from the 6 DoF (Euler Angles) block, the real values of the body accelerations and body velocity were stored in variables for comparison. They were then fed into the Sensors block, where the data for the IMU and GPS was simulated. Also present in this block

was the process involved with the Extended Kalman Filter, and the overall subsystem can be seen below in Figure 24.

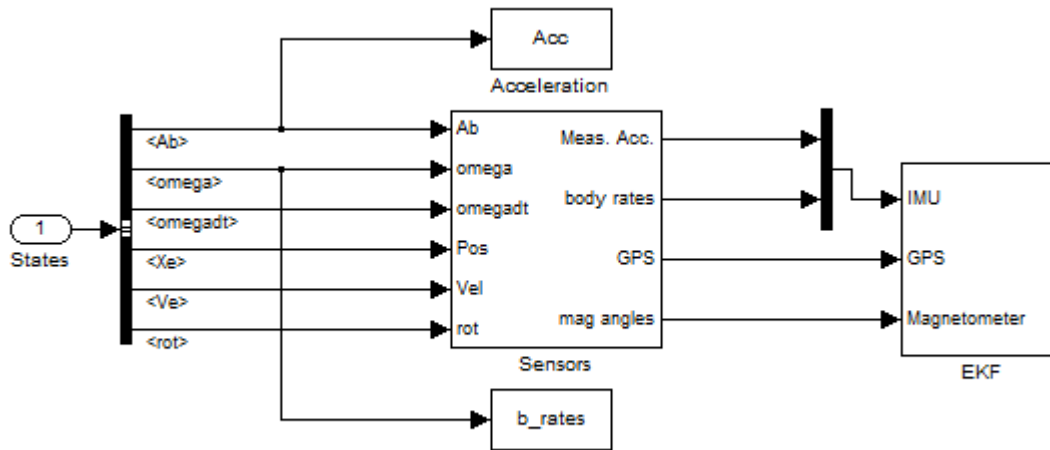


Figure 24: Sensor State Estimation Subsystem

The estimated pitch value from the EKF was also compared to the model simulated value here. Error was calculated in the ‘Pitch Error Calculation’ block, and a test to determine the time to deployment for various launch angles was conducted in the ‘Time to Deployment Test’ block. These results will be discussed later in the thesis.

5.3.1. Sensors Block

In the ‘Sensors’ block, the outputs of the IMU placed on the HP are simulated and analyzed along with GPS and magnetometer signals. Actual hardware was implemented for test purposes on the WVU HP, specifically an ArduPilotMega APM2 autopilot (seen in Figure 26) which has a built in MPU-6000 IMU (seen below in Figure 25), GPS unit and magnetometer (Anderson 2010).



Figure 25: MPU-6000 IMU (*invensense.com*)



Figure 26: ArdupilotMega hardware (*Anderson 2010*)

The process flow of the ‘Sensors’ block can be seen below in Figure 27. Simulated outputs from the sensors were then able to be used in the EKF state estimation process.

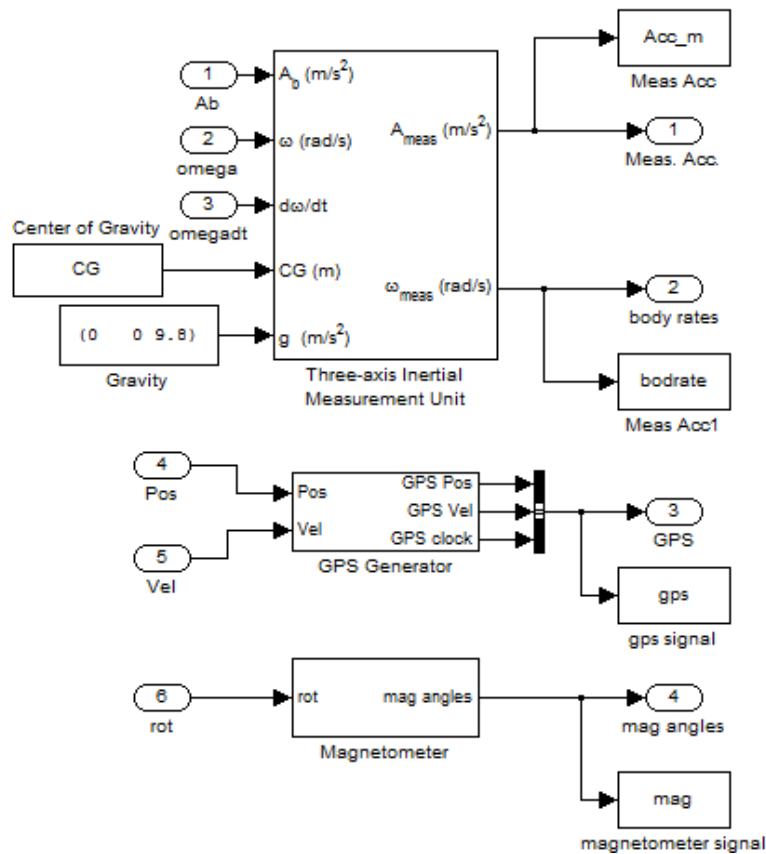


Figure 27: Sensors Subsystem

The GPS and magnetometer sensors were simulated in Simulink using the model calculated state values as a basis. The resultant measurement signals were verified based on available sensor specifications. The IMU was simulated using the block in the Aerospace block set, and block parameters were chosen using real sensor specs.

5.3.1.1. Inertial Measurement Unit Simulation

The IMU block’s configuration can be seen below in Table 1; second order dynamics were assumed for both the accelerometer and gyroscope in this situation. The inputs to the IMU block include body accelerations, A_b (m/s²), body angle rates, ω (rad/s), body angular accelerations, $\frac{d\omega}{dt}$ (rad/s²) center of gravity, CG (m) and gravity, g (m/s²).

Table 1: IMU Simulation parameters

Units	Metric (MKS)
IMU Location	[-10 0 0]
Update Rate	0.001 sec
Accelerometer natural frequency (rad/sec)	190
Accelerometer damping ratio	1
Accelerometer scale factor and cross coupling	[1 0 0; 0 1 0; 0 0 1]
Accelerometer measurement bias	[0 0 9.8]
Gyro natural frequency (rad/sec)	190
Gyro damping ratio	0.707
Gyro scale factors and cross-coupling	[1 0 0; 0 1 0; 0 0 1]
Gyro measurement bias	[0 0 0]
G-sensitive bias	[0 0 0]
Noise seeds	[23093 23094 23095 23096 23097 23098]
Noise power	[2e-4 2e-4 2e-4 1e-7 1e-7 1e-7]

Realistically, the accelerometer and gyroscope upper and lower bounds would have some finite bounds, but for the sake of testing they were allowed to go to infinity to avoid clipping. Gravity was accounted for in the accelerometer as a bias. The accelerometer was assumed to be critically damped; the gyroscope however was assumed to be underdamped. Noise was generated in the IMU to simulate real world measurement error using the starting seeds for the random number generator for each IMU output seen above in Table 1.

The outputs of the IMU block were the three body accelerations and the three body angle rates and they are stored in variables for comparison. In order to verify that the IMU estimated accelerations and body angle rates accurately, noise was omitted and the outputs were analyzed; they can be seen below in Figure 28.

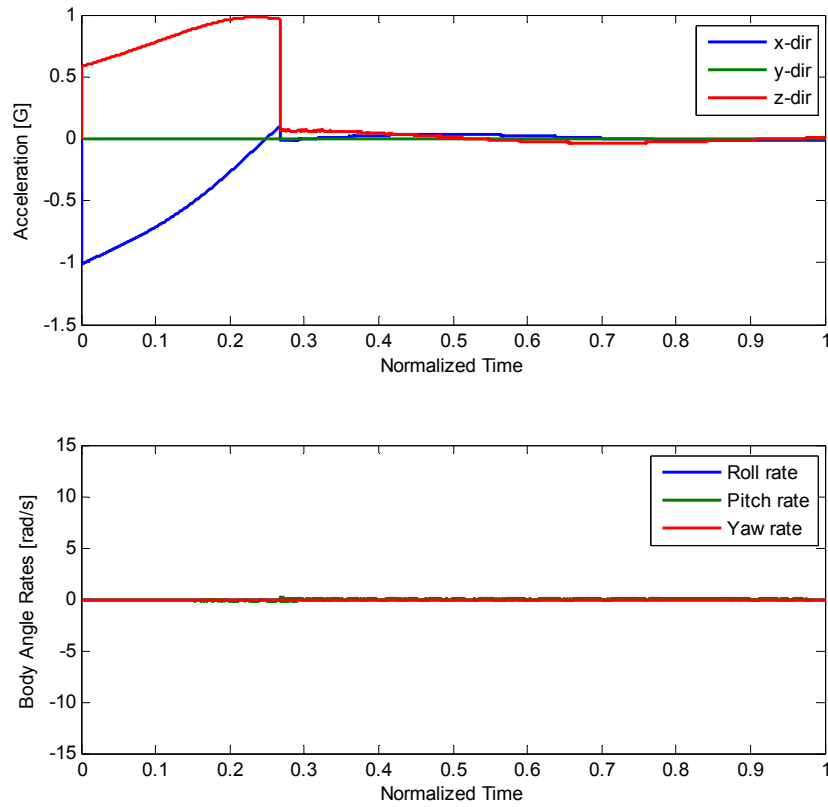


Figure 28: IMU Acceleration (top) and Rotation Rate (bottom) Measurements

Since a real world IMU would have a noisy signal, noise was added and the new signal can be seen below in Figure 29. Noise values were added based on selected sensor specifications.

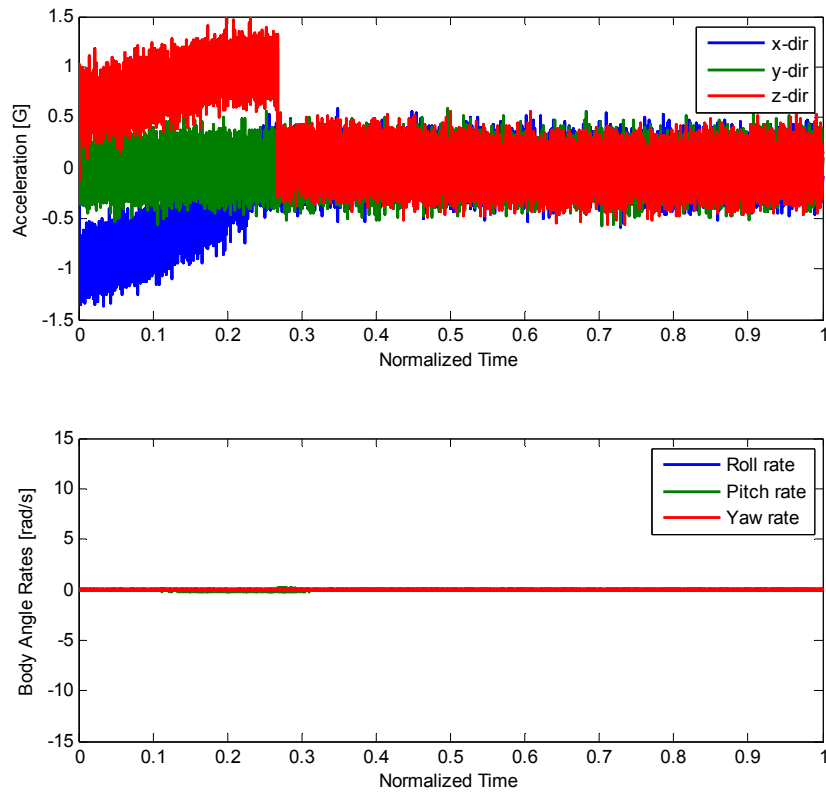


Figure 29: IMU Acceleration (top) and Body Angle Rates (bottom) Measurements with noise

5.3.1.2. *GPS Simulation*

GPS outputs were simulated by using the model calculated position and velocity signals and passing them through a rate transition block where they were discretized at a sampling rate of 10 Hz. A clock was also discretized so when the GPS data was used by the EKF, it would be able to tell when a new measurement was received so the update equations could be used. Gaussian noise was added to the signals with a variance of 100 m^2 for position and $10 \text{ m}^2/\text{s}^2$ for velocity. This method, seen in Figure 30, operated under the assumption that the GPS measurements were already converted from geodetic latitude, longitude, and altitude (LLA) to a flat Earth reference frame.

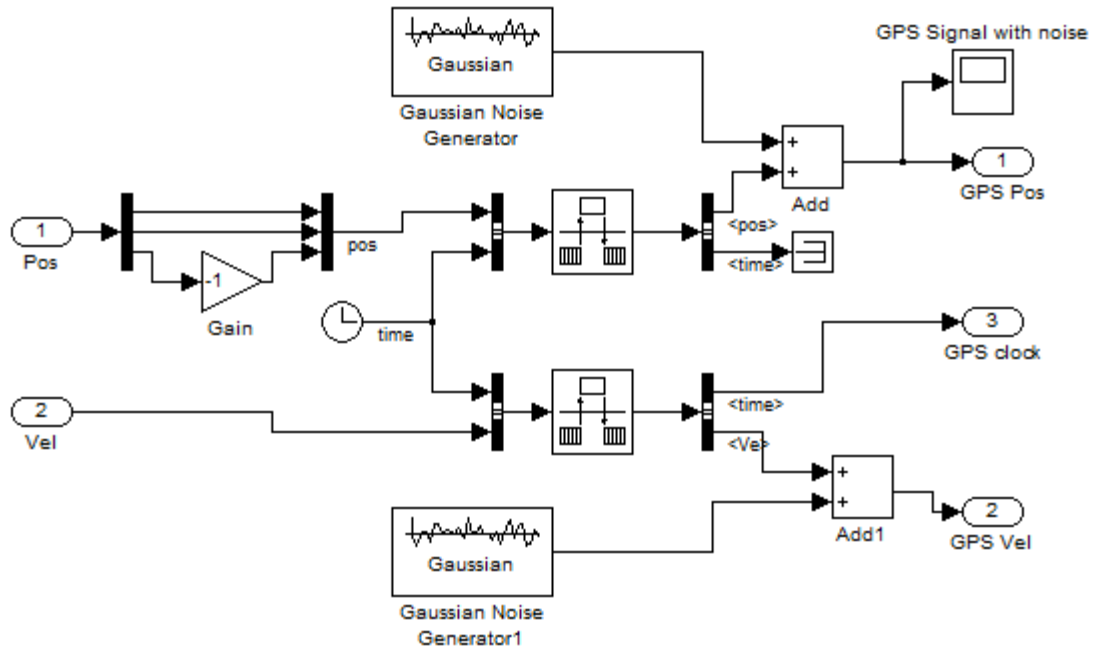


Figure 30: GPS Simulation

Velocity can be measured directly from GPS receivers by analyzing the relativistic Doppler Effect between the receiver and transmitter (*Zhang, et al 2006*). The Doppler Effect is experienced when there is relative motion between the source and observer of a wave signal. For a GPS signal, the Doppler Effect can be expressed as (*Mirsa and Enge 2001*):

$$f_r = f_s \left(1 - \frac{\dot{\vec{r}}_r^s}{c}\right) \quad 5-20$$

Where f_r is the received frequency, f_s is the original frequency of the transmitter, \vec{r}_r^s is the receiver-satellite line of sight vector (LOS), and $\dot{\vec{r}}_r^s$ is the relative velocity of the receiver-satellite LOS vector. With this knowledge the assumption that the model calculated velocities were used is justified, and the simulated position and velocity outputs can be seen below in Figure 31.

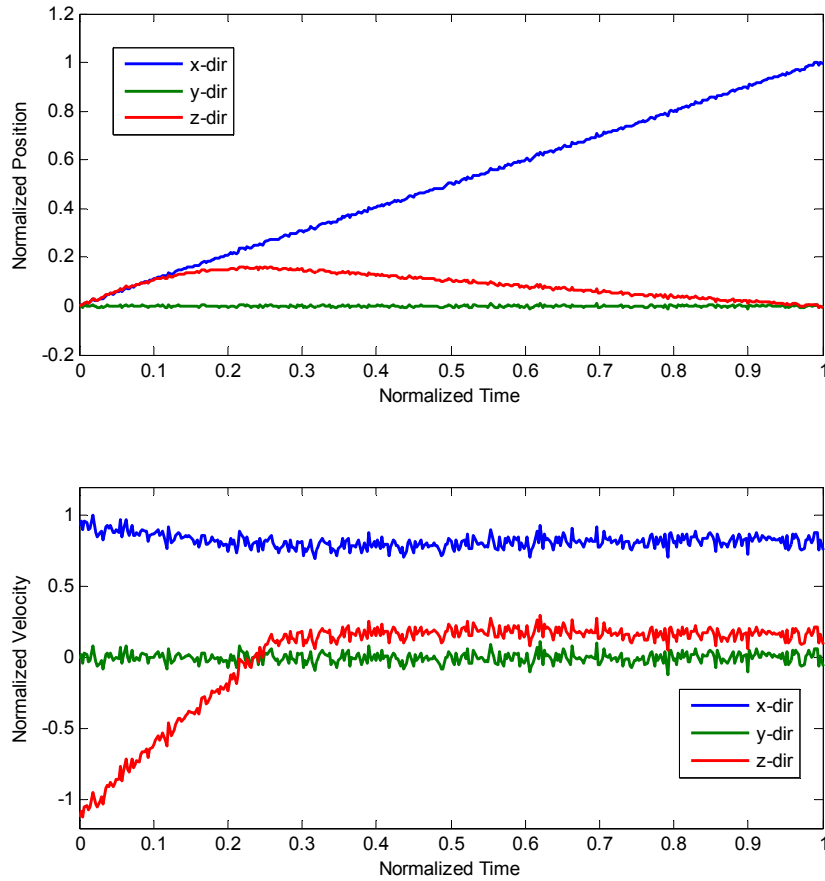


Figure 31: GPS Simulated Position (top) and Velocity (bottom) with Noise

5.3.1.3. *Magnetometer Simulation*

Magnetometers measure the strength and direction of magnetic fields, and they can be used to provide measurements of a body's orientation in space. The work of *Jagadish, et al (2007)* outlines various methods of using magnetometer and accelerometer data to find Euler angles. The Roll-Pitch-Yaw sequence was chosen because it uses the measurements a_z , m_x , m_y and m_z . Other methods use measurements that account for side forces and would not accurately represent the Euler angles in this case where side forces are assumed to be zero. The equations used to calculate the Euler angles can be seen below, where 'c' represents cosine and 's' represents sine.

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} c\psi_1 c\theta_1 & s\psi_1 c\varphi_1 + c\psi_1 s\theta_1 s\varphi_1 & s\psi_1 s\varphi_1 - c\psi_1 s\theta_1 c\varphi_1 \\ -s\psi_1 c\theta_1 & c\psi_1 c\varphi_1 - s\psi_1 s\theta_1 s\varphi_1 & c\psi_1 s\varphi_1 + s\psi_1 s\theta_1 c\varphi_1 \\ s\theta_1 & -c\theta_1 s\varphi_1 & c\theta_1 c\varphi_1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad 5-21$$

$$\begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = \begin{bmatrix} c\psi_1 c\theta_1 & s\psi_1 c\varphi_1 + c\psi_1 s\theta_1 s\varphi_1 & s\psi_1 s\varphi_1 - c\psi_1 s\theta_1 c\varphi_1 \\ -s\psi_1 c\theta_1 & c\psi_1 c\varphi_1 - s\psi_1 s\theta_1 s\varphi_1 & c\psi_1 s\varphi_1 + s\psi_1 s\theta_1 c\varphi_1 \\ s\theta_1 & -c\theta_1 s\varphi_1 & c\theta_1 c\varphi_1 \end{bmatrix} \begin{bmatrix} c\alpha \\ 0 \\ s\alpha \end{bmatrix} \quad 5-22$$

$$\theta_1 = s^{-1}\left(\frac{[m_z - a_z s\alpha]}{c\cos\alpha}\right) \quad 5-23$$

$$\varphi_1 = c^{-1}\left(\frac{a_z}{c\theta_1}\right) \quad 5-24$$

$$\psi_1 = s^{-1}\left(\frac{[Bm_z - Am_y]}{[A^2 + B^2]}\right) \quad 5-25$$

where α is the is the inclination angle of the earth magnetic field (NGDC 2007) and

$$A = c\theta_1 c\alpha - s\theta_1 c\varphi_1 s\alpha \quad 5-26$$

$$B = s\varphi_1 s\alpha \quad 5-27$$

The rotation angles found from Equations 2-12 through 2-14 are then converted to standard Euler angles using the equations below.

$$\theta = s^{-1}(c\varphi_1 s\theta_1 c\psi_1 - s\varphi_1 s\psi_1) \quad 5-28$$

$$\varphi = c^{-1}\left(\frac{c\varphi_1 c\theta_1}{c\theta}\right) \quad 5-29$$

$$\psi = c^{-1}\left(\frac{c\theta_1 c\psi_1}{c\theta}\right) \quad 5-30$$

The equations validated the assumption of using the model calculated Euler angles with added noise as magnetometer, and the Simulink modeling can be seen below in Figure 32.

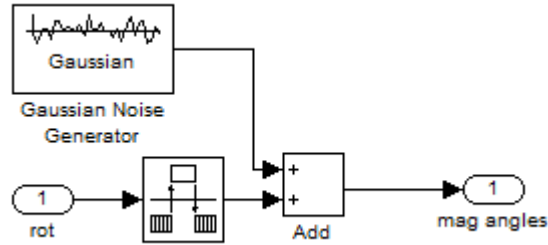


Figure 32: Magnetometer Simulation

After acquiring the sensor data, a method was sought to reduce noise and provide accurate estimates of the states; based on the nonlinear nature of the system, the Extended Kalman Filter was chosen.

6. Extended Kalman Filter Implementation

Accurate readings of the states were essential in order for the HP to effectively extend its range. However, noise propagating through integration steps provided excessive error. An Extended Kalman Filter was used as a method to reduce the noise from the measurements while simultaneously estimating the proper states.

The EKF process equations from *Lozano (2010)* can be seen below:

$$\dot{x} = f(x, w) \quad 6-1$$

$$Q = E(ww^T) \quad 6-2$$

$$z = h(x, v) \quad 6-3$$

$$R = E(vv^T) \quad 6-4$$

$$F = \frac{\partial f(x)}{\partial x} \quad 6-5$$

$$H = \frac{\partial h(x)}{\partial x} \quad 6-6$$

$$\Phi_k \approx I + FT_s \quad 6-7$$

where x is the system states vector, f is the nonlinear function of the states, and w is a zero mean random process. The process noise matrix is represented by Q in Equation 6-2, and the nonlinear measurement function can be seen by Equation 6-3. The measurement noise R , Equation 6-4, is a function of a zero-mean random noise process, v . Since the function of the states is nonlinear, it was necessary to linearize them using a first-order approach to obtain the dynamic matrix of the system F and the measurement matrix H , seen in Equations 6-5 and 6-6. The fundamental matrix Φ can be approximated by using a Taylor Series expansion; however, the series is often expressed using only the first two terms of the expansion, seen in Equation 6-7, where I is the identity matrix and T_s is the sampling time. For both linear and nonlinear systems, the Riccati equations for calculating the Kalman gain are identical and can be seen below:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad 6-8$$

$$P_k^- = A_k P_{k-1} A_k^T + Q \quad 6-9$$

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad 6-10$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \quad 6-11$$

$$P_k = (I - K_k H) P_k^- \quad 6-12$$

6.1. State Estimation Process in Matlab

Simulation of the EKF was done using a Matlab function block within the Simulink model. The inputs to the block, seen below in Figure 33, were the IMU inputs (u), the initial state vector (x_0), initial covariance matrix (P_0), EKF sampling time (T_s), GPS measurements (gps), GPS clock ($gpsTs$), and magnetometer measurements (mag .)

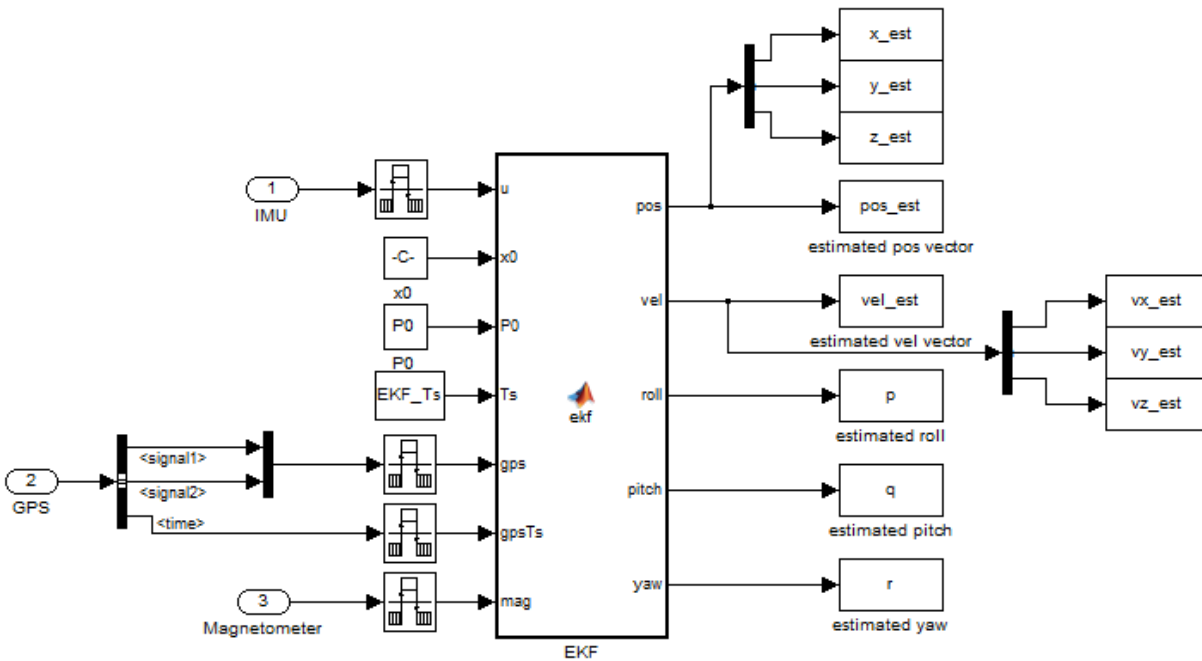


Figure 33: EKF Matlab Function Block in Simulink Model

The outputs of the block seen in Figure 33, global position, global velocity and body angles, were defined as the state vector within the EKF function block. The prediction of the state vector seen below in Equation 6-13 (*Gross 2010*) was done in the beginning of the algorithm and corresponds to Equation 6-8. The subscript k denotes the state at the current time step, x , y , and z are the global distance states, V_x , V_y , and V_z are the global velocity states, and φ , θ , and ψ are the Euler angles of the HP.

$$\begin{bmatrix} x_k \\ y_k \\ z_k \\ V_{x_k} \\ V_{y_k} \\ V_{z_k} \\ \varphi_k \\ \theta_k \\ \psi_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + T_s V_{x_{k-1}} \\ y_{k-1} + T_s V_{y_{k-1}} \\ z_{k-1} + T_s V_{z_{k-1}} \\ V_{x_k} + T_s (c\psi_{k-1}c\theta_{k-1}a_{x_{k-1}} + (-s\psi_{k-1}c\varphi_{k-1} + c\psi_{k-1}s\theta_{k-1}s\varphi_{k-1})a_{y_{k-1}} + (s\psi_{k-1}s\varphi_{k-1} + c\psi_{k-1}s\theta_{k-1}c\varphi_{k-1})a_{z_{k-1}}) \\ V_{y_k} + T_s (s\psi_{k-1}c\theta_{k-1}a_{x_{k-1}} + (c\psi_{k-1}c\varphi_{k-1} + s\psi_{k-1}s\theta_{k-1}s\varphi_{k-1})a_{y_{k-1}} + (-c\psi_{k-1}s\varphi_{k-1} + s\psi_{k-1}s\theta_{k-1}c\varphi_{k-1})a_{z_{k-1}}) \\ V_{y_k} + T_s (-s\theta_{k-1}a_{x_{k-1}} + (c\theta_{k-1}s\varphi_{k-1})a_{y_{k-1}} + (c\theta_{k-1}c\varphi_{k-1})a_{z_{k-1}} - g) \\ \varphi_{k-1} + T_s (p_{k-1} + q_{k-1}s\varphi_{k-1}t\theta_{k-1} + r_{k-1}c\varphi_{k-1}t\theta_{k-1}) \\ \theta_{k-1} + T_s (q_{k-1}c\varphi_{k-1} - r_{k-1}s\varphi_{k-1}) \\ \psi_{k-1} + T_s (q_{k-1}s\varphi_{k-1} + r_{k-1}c\varphi_{k-1})\sec\theta_{k-1} \end{bmatrix} \quad 6-13$$

The next step was to predict the covariance P from Equation 6-9, where F , Equation 6-15, is linearized state relationship matrix and Q is the process noise covariance matrix. Q can be measured offline based on sensor error analysis, but was approximated based on the work of *Gross (2010)* in addition to error analysis; it was defined as:

$$\begin{aligned} Q &= \text{diag}(0,0,0, \sigma^2 \text{ of } a_x, \sigma^2 \text{ of } a_y, \sigma^2 \text{ of } a_z, \sigma^2 \text{ of } p, \sigma^2 \text{ of } q, \sigma^2 \text{ of } r) = \\ &= \text{diag}(0,0,0, 2e-4, 2e-4, 1e-7, 1e-7, 1e-7) \end{aligned} \quad 6-14$$

$$F_k = \frac{\partial f(x)}{\partial x}$$

$$= \begin{bmatrix} 1 & 0 & 0 & T_s & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & T_s & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & T_s & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & T_s \begin{pmatrix} (s\psi s\varphi + c\psi s\theta c\varphi)a_y \\ + (s\psi c\varphi - c\psi s\theta s\varphi)a_z \end{pmatrix} & T_s \begin{pmatrix} -c\psi s\theta a_x + c\psi c\theta s\varphi a_y \\ + c\psi c\theta c\varphi a_z \end{pmatrix} & T_s \begin{pmatrix} -s\psi c\theta a_x - (c\psi c\varphi + s\psi s\theta s\varphi)a_y \\ + (c\psi s\varphi - s\psi s\theta c\varphi)a_z \end{pmatrix} \\ 0 & 0 & 0 & 0 & 1 & 0 & T_s \begin{pmatrix} (-c\psi s\varphi + s\psi s\theta c\varphi)a_y \\ - (c\psi c\varphi + s\psi s\theta s\varphi)a_z \end{pmatrix} & T_s \begin{pmatrix} -s\psi s\theta a_x + s\psi c\theta s\varphi a_y \\ + s\psi c\theta c\varphi a_z \end{pmatrix} & T_s \begin{pmatrix} c\psi c\theta a_x + (-s\psi c\varphi + c\psi s\theta s\varphi)a_y \\ + (s\psi s\varphi + c\psi s\theta c\varphi)a_z \end{pmatrix} \\ 0 & 0 & 0 & 0 & 0 & 1 & T_s(c\theta c\varphi a_y - c\theta s\varphi a_z) & -T_s \begin{pmatrix} c\theta a_x + s\theta s\varphi a_y \\ + s\theta c\varphi a_z \end{pmatrix} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 + T_s t\theta(q c \varphi - r s \varphi) & T_s \sec^2 \theta_{k-1}(q s \varphi_{k-1} + r c \varphi_{k-1}) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -T_s(q s \varphi + r c \varphi) & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & T_s(q c \varphi - r s \varphi) & T_s \sec \theta_{k-1} t \theta_{k-1}(q s \varphi_{k-1} + r c \varphi_{k-1}) & 1 \end{bmatrix} \quad 6-15$$

The update process was only executed when GPS and magnetometer readings were taken. GPS and magnetometer data were compiled into a single vector, which became the measurement vector, z seen below in Equation 6-16.

$$z = [x_{GPS} \quad y_{GPS} \quad z_{GPS} \quad V_{x_{GPS}} \quad V_{y_{GPS}} \quad V_{z_{GPS}} \quad \varphi_{mag} \quad \theta_{mag} \quad \psi_{mag}]^T \quad 6-16$$

According to Equation 6-3, z can be expressed as a function of the state vector and some noise value, specifically:

$$z = Hx + v \quad 6-17$$

where H , which can be found by taking the derivative of the measurement function z with respect to the state vector, is the observation matrix. The next step in the EKF process was calculating the Kalman gain, K , using Equation 6-10. Measurement noise covariance values, R , and the observation matrix H were necessary to calculate K , and the equations used to find them were:

$$R = \text{diag}(\sigma^2 \text{ of } x_{GPS}, \sigma^2 \text{ of } y_{GPS}, \sigma^2 \text{ of } z_{GPS}, \sigma^2 \text{ of } V_{x_{GPS}}, \sigma^2 \text{ of } V_{y_{GPS}}, \sigma^2 \text{ of } V_{z_{GPS}}, \sigma^2 \text{ of } \varphi_{mag}, \sigma^2 \text{ of } \theta_{mag}, \sigma^2 \text{ of } \psi_{mag}) \quad 6-17$$

$$R = \text{diag}(0.01, 0.01, 0.01, 1e-4, 1e-4, 1e-4, 1e-7, 1e-7, 1e-7)$$

$$H = \frac{\partial h(x)}{\partial x} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad 6-18$$

Residual error between the measurement and the initial prediction was then calculated and multiplied by the Kalman gain K . The prediction of the state vector was then updated by adding this value to the previous state vector prediction. This process was repeated for each time step.

6.2. Sampling Time Selection

For discrete time systems, a sampling time must be defined. In this thesis, there were four necessary sampling times that were defined: the sampling times for the IMU, EKF, GPS and magnetometer. The hardware that was implemented on the HP, the ArduPilotMega (*Anderson 2010*), has a sampling rate of 1 kHz, so that was chosen for use in this thesis. In between measurement updates, dead reckoning was used to determine the states of the HP. Dead reckoning uses previously known values to predict where the HP is at the current time step. Once a new measurement was taken, these predictions were corrected using the EKF process.

A simulation was then done to determine the sensitivity of the EKF to noise based on the sampling rate chosen. Figure 34 below shows the relationship between pitch error, EKF sampling time and IMU noise power. Noise power is defined in Simulink as the height of the power spectral density (PSD). As expected, there is higher error in pitch estimation as the IMU noise power becomes larger and the EKF sampling rate gets slower. Since the IMU was the input to the EKF, its sampling rate of 1 kHz was assumed to be the sampling rate of the EKF as well. This assumption was validated for any noise power, with the highest average pitch error magnitude being 0.341° at a noise power of 0.5, which is much larger than the chosen noise power of $1e-7$. The highest average pitch error magnitude of 9.91° occurred at a sampling time of 0.5s and a noise power of 0.5.

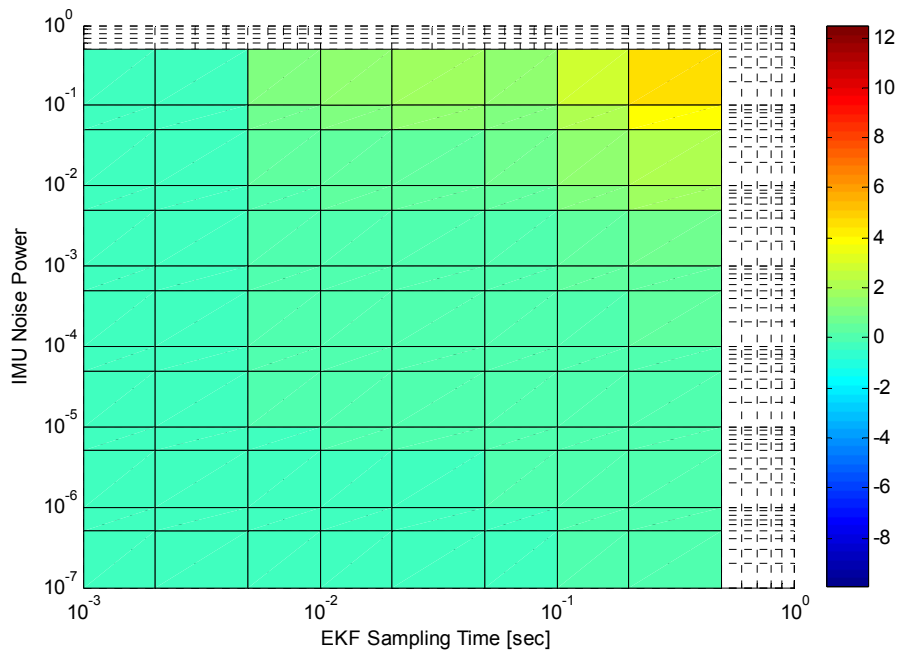


Figure 34: Pitch Error with Varying IMU Noise Power and EKF Sampling Time

Selecting the sampling rates of the measurements, the GPS and magnetometer, was dependent on the specifications of the ArduPilotMega; it is available with a GPS module that operates at 5 or 10 Hz. To ensure the best possible position estimation, a sampling rate of 10 Hz was selected for the GPS. The magnetometer that was used on the HP was a Honeywell HMC5883L 3-Axis Digital Compass IC, features a sampling rate of 160 Hz. This would allow for the Euler angle update equations to run more frequently than the position equations. However, the sampling time of the magnetometer was assumed to be equal to that of the GPS so the entire state vector could be updated at the same time. Error analysis in the next section validated this procedure. Once sampling times and noise had been defined, the EKF was run and the results were compared to that of the model calculated values.

7. Results

Once all parameters of the simulation were defined, the effectiveness of the EKF for state estimation was tested with noisy IMU, GPS and magnetometer data. In Figure 35 the graphs of the model calculated and EKF estimated position values can be seen. Distance and Altitude, x and z position respectively, are estimated very well, but the y position estimates deviate from the true value of zero. At the end of the simulation, there is a noticeable difference in true and estimated z position; this will be expanded on in Section 7.1. Velocity was next to be analyzed, and Figure 36 below shows the graphs of the model calculated and EKF estimated velocity. The estimated values adhered well to the true values calculated by the model. Finally, the Euler angles were analyzed, and they can be seen below in Figure 37. Overall, the Euler angle estimation was the best as shown by the graphs.

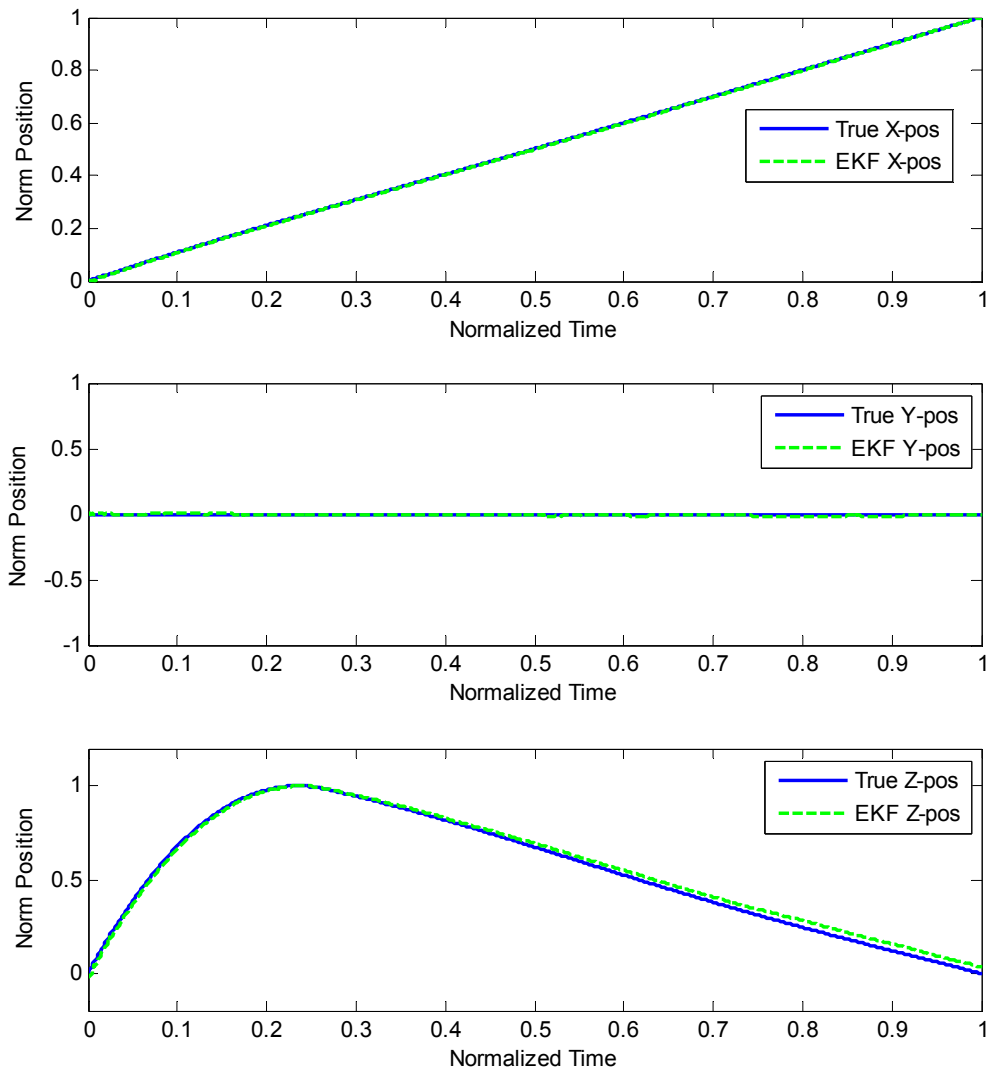


Figure 35: Model and EKF Calculated Position for a Launch angle of 50° , a Body Angle Deployment of -10° and an Initial Velocity of 126 m/s

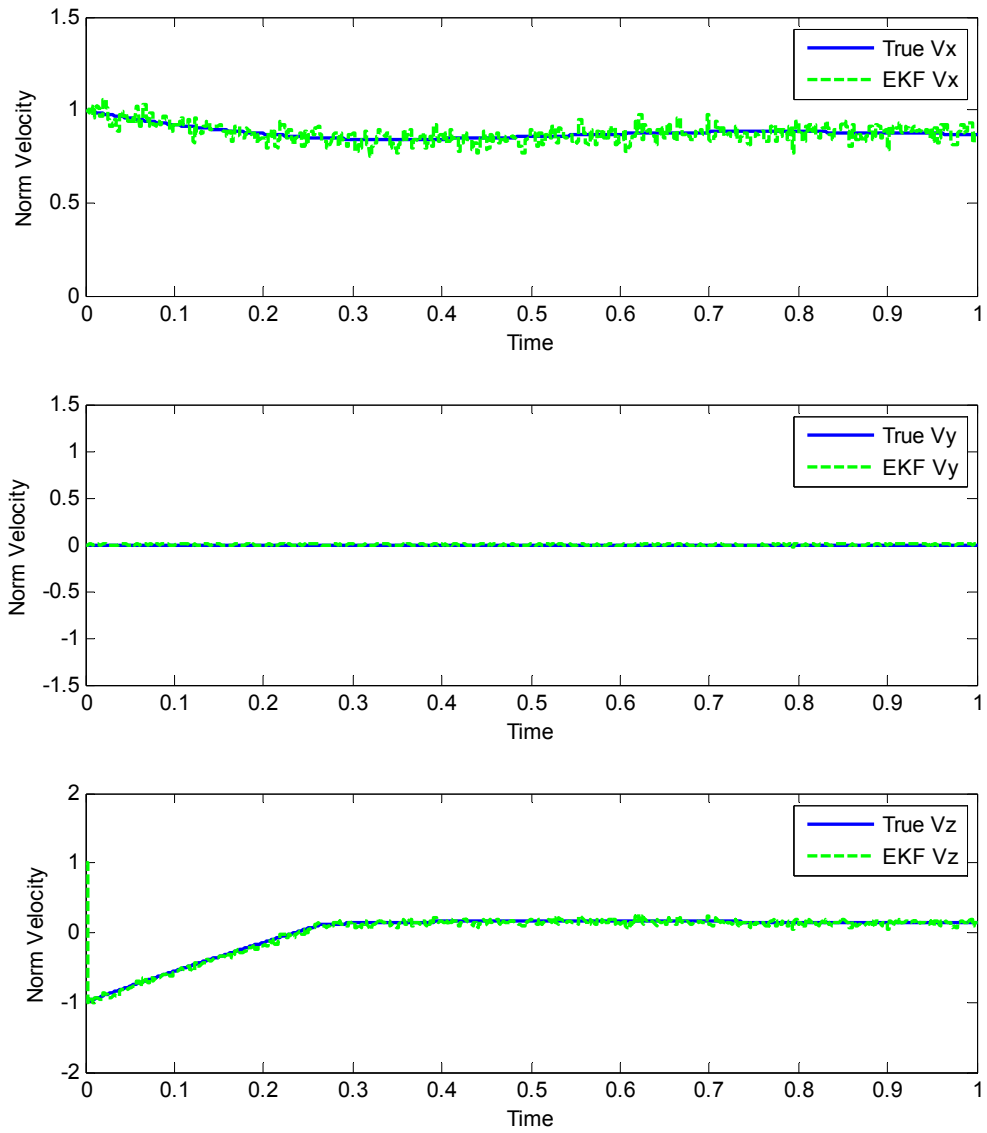


Figure 36: Model and EKF Calculated Velocity for a Launch angle of 50° , a Body Angle Deployment of -10° and an Initial Velocity of 126 m/s

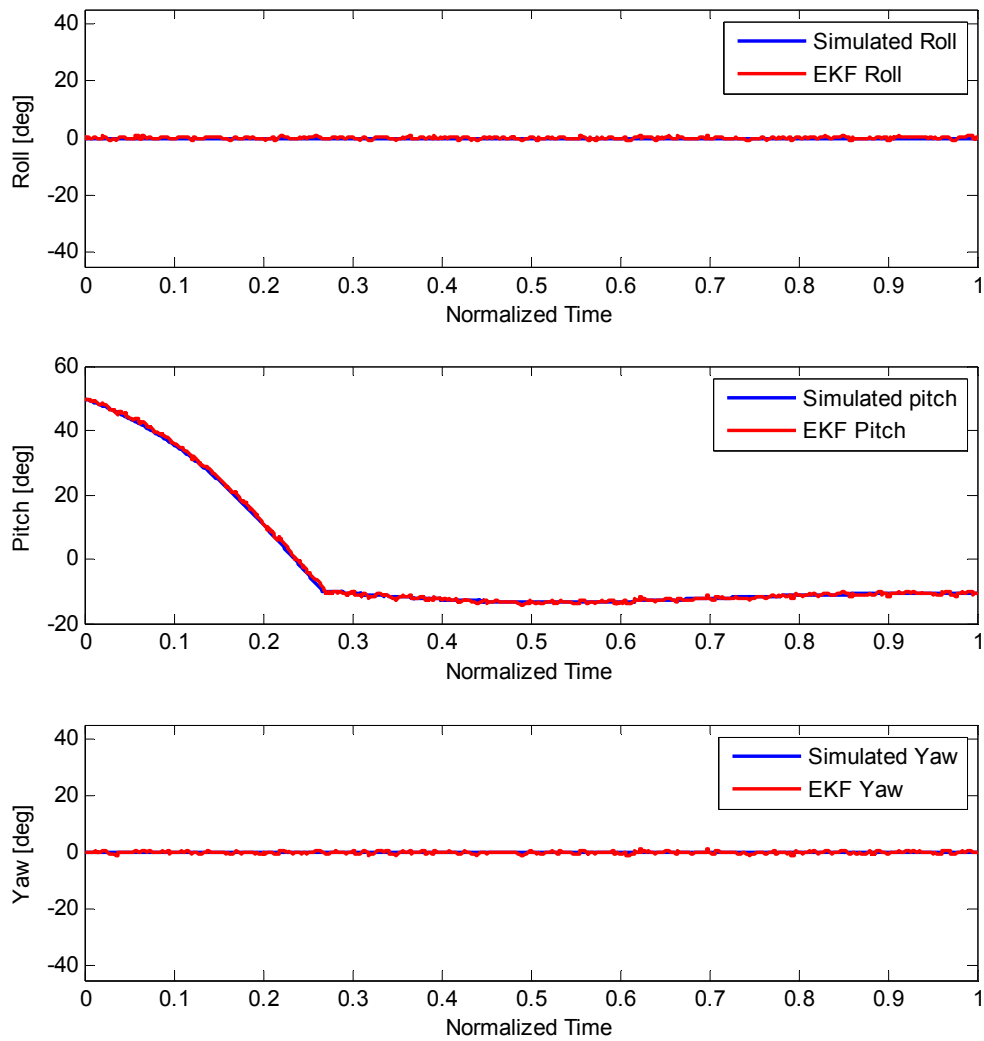


Figure 37: Model and EKF Calculated Euler Angles for a Launch angle of 50°, a Body Angle Deployment of -10° and an Initial Velocity of 126 m/s

After estimated the states through the EKF and comparing them to model calculated values, an error analysis was done to better determine the overall effectiveness of the filter.

7.1. EKF Error Analysis

Error between the EKF estimated state and model state vectors was calculated over time and can be seen below in Figure 38. From these values, the root mean squared (RMS) error was then calculated for each state, as seen in Table 2.

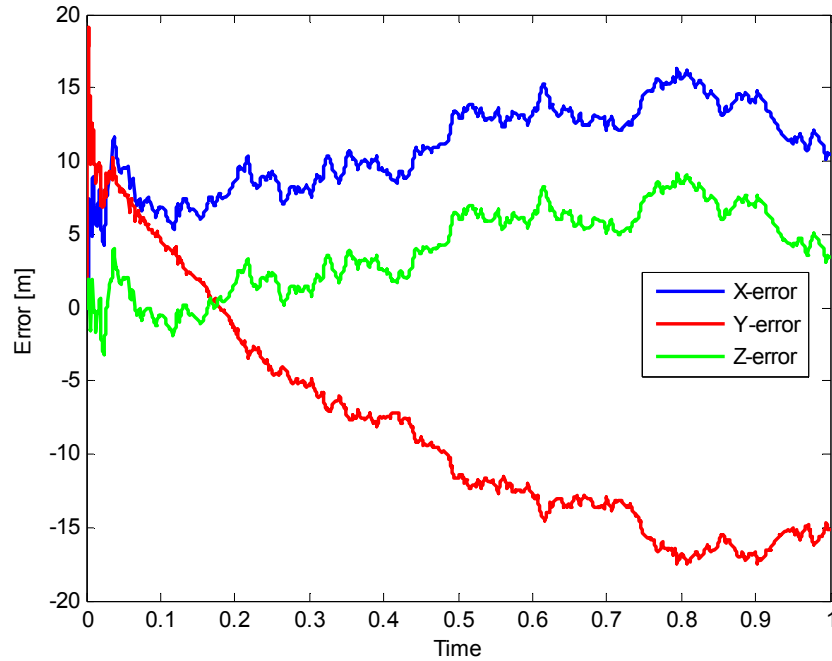


Figure 38: Error between Model and EKF Calculated Position for a Launch Angle of 50° , Wing Deployment Angle of -10° , and an Initial Velocity of 126 m/s

Table 2: Root Mean Squared Error Values for the EKF State Vector

State	RMS Value	Units
x	11.372	m
y	4.896	m
z	11.383	m
V_x	2.891	m/s
V_y	2.924	m/s
V_z	10.141	m/s
ϕ	0.391	deg
θ	0.490	deg
ψ	0.391	deg

It should be noted that over time the position error values increase, meaning that the filter diverges. According to *Perea, et. Al (2007)* there is divergence issues in EKFs when the dynamic models, measurement models, or both are not linear functions of the state vector. While the dynamic model for each state is nonlinear, the position prediction vector is calculated using the equations for velocity, allowing velocity error to propagate through a time step. The error plots for velocity and Euler angles can be seen in Figure 39 and Figure 40, respectively.

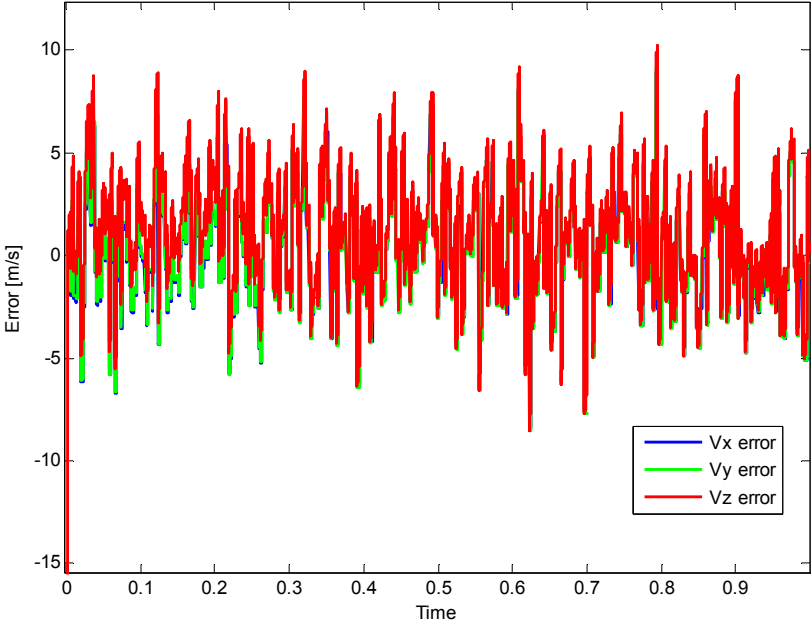


Figure 39: Error between Model and EKF Calculated Velocity for a Launch Angle of 50°, Wing Deployment Angle of -10°, and an Initial Velocity of 126 m/s

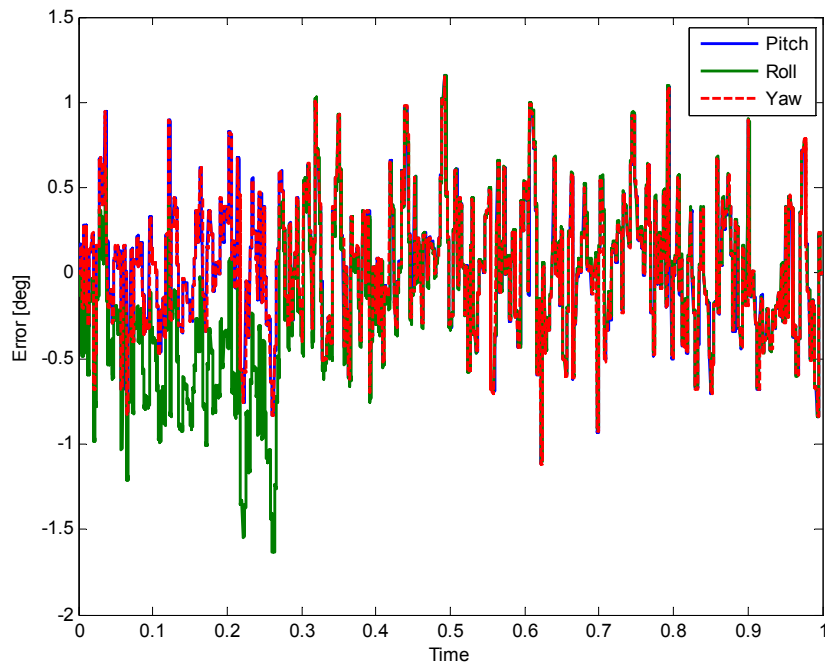


Figure 40: Error between Model and EKF Calculated Euler Angles for a Launch Angle of 50° , Wing Deployment Angle of -10° , and an Initial Velocity of 126 m/s

The error values for velocity and Euler angles, unlike position, stayed within a finite bandwidth. Based on these results, it should be noticed that the EKF only diverges over time for position estimation. Because the HP is only in flight for a short amount of time and does not cover significant distances, the divergence of position estimation is acceptable. If the EKF update equations were run faster than 10 Hz, it could potentially prevent any such divergence, but the effect of modifying the update rate was not investigated in this thesis. After concluding that there were acceptable error values, the model was then analyzed to see if the assumption that a glide angle of -10° provided maximum range extension to the HP.

7.2. Hold at -10° Wing Deployment Angle

Since the HP was designed to glide at a -10° angle, it was assumed throughout the project that this was the deployment condition that provided maximum extended range. With the distance map already outlining HP performance based on launch and body deployment angle in Figure 17, further tests were run to support this claim. Figure 41 provides a close up of possible launch angles and body deployment angles that provide extended range. The dark red region of the graph shows the possible combinations of launch angle and body angle deployment criteria where maximum range extension was achieved. At a body angle of -10° , performance begins around a launch angle of 45° .

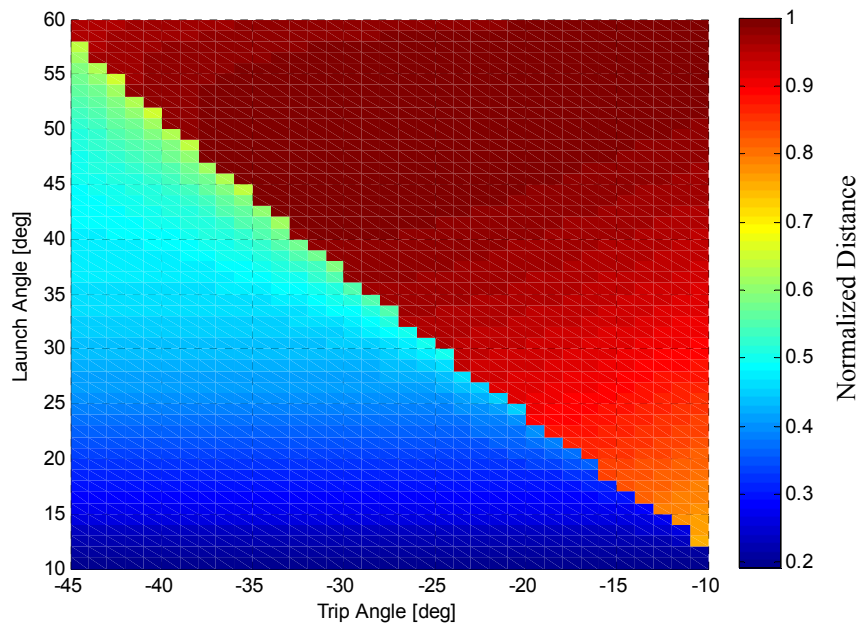


Figure 41: Maximum Range Extension based on Varying Launch and Trip Angles

The -10° body angle deployment condition was tested for various launch angle conditions to see if it was in fact the best possible angle to deploy the wings at. If another body angle deployment condition performed better its range was recorded, and the percent difference between the range

of the -10° condition and the absolute maximum achievable range was calculated. This can be seen below in Figure 42. There are, however, sharp jumps in percent difference for launch angles between 10° and around 35° which can be attributed to confidence issues. After this region, there is a steady decline in difference between the true maximum and -10° condition's ranges. The difference in range for a launch angle of 50° is only 1%, which shows near optimal performance for the criteria used in the model.

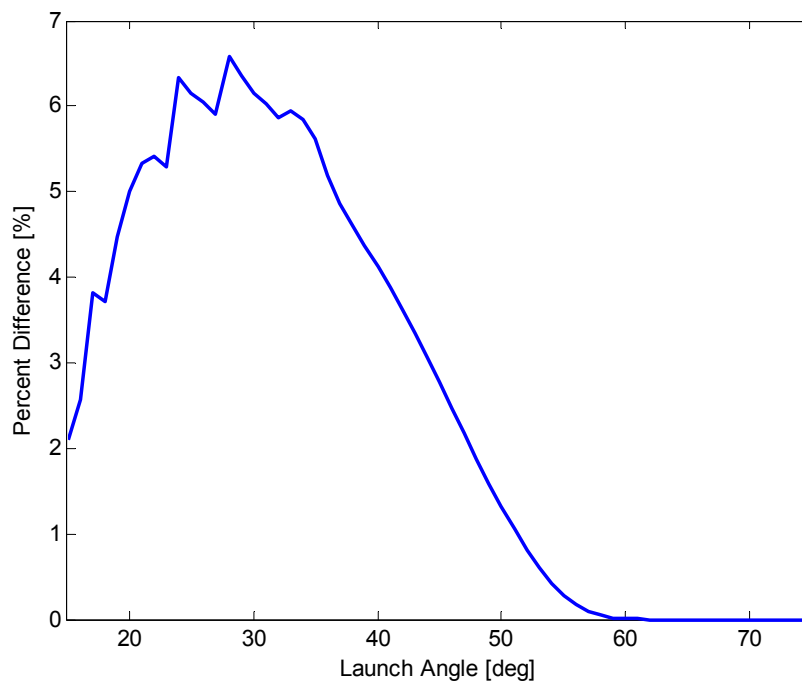


Figure 42: Percent Difference between Actual Deployment Angle for Maximum Range and -10° Deployment

Next, launch angle was held constant at 50° , and a test was run to determine what body angle deployment condition provided maximum range extension of the HP, shown by Figure 43. There was no major difference between 10° and 39° , with the best overall range extension occurring at angles between 25° and 27° . The next sample at 40° showed a sharp decline in performance, a drop of 37.4%.

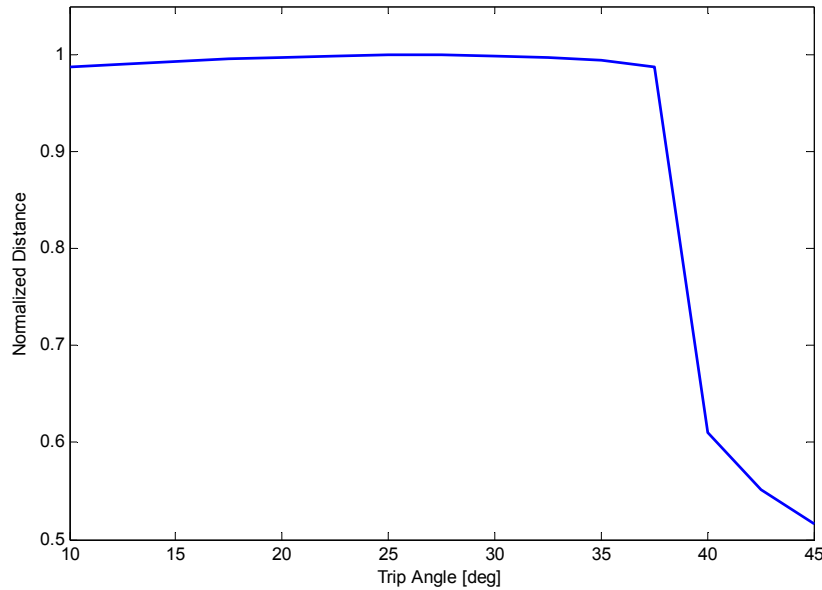


Figure 43: Range Extension Achieved based on Varying Trip Angle Deployment for a Launch Angle of 50°

Differences between the absolute maximum range extension and that achieved by the assumed conditions are small. Based on these tests, the use of a launch angle of 50° and a body angle deployment condition of -10° is supported. The possibility of using a timer condition to deploy wings was then investigated to determine if there were any advantages over using the body angle deployment condition.

7.3. Timer vs. Body Angle Deployment Comparison

Another proposed condition of wing deployment was using a timer started after launch. An experiment was conducted that determined if a timer wing deployment method would achieve better range extension than a body angle wing deployment method. In Figure 44 below, a graph of maximum range extension based on the launch angle and time of wing deployment can be seen. Optimal range extension occurs at a launch angle between 40° and 60° and for a timer value between 11 and 15 seconds.

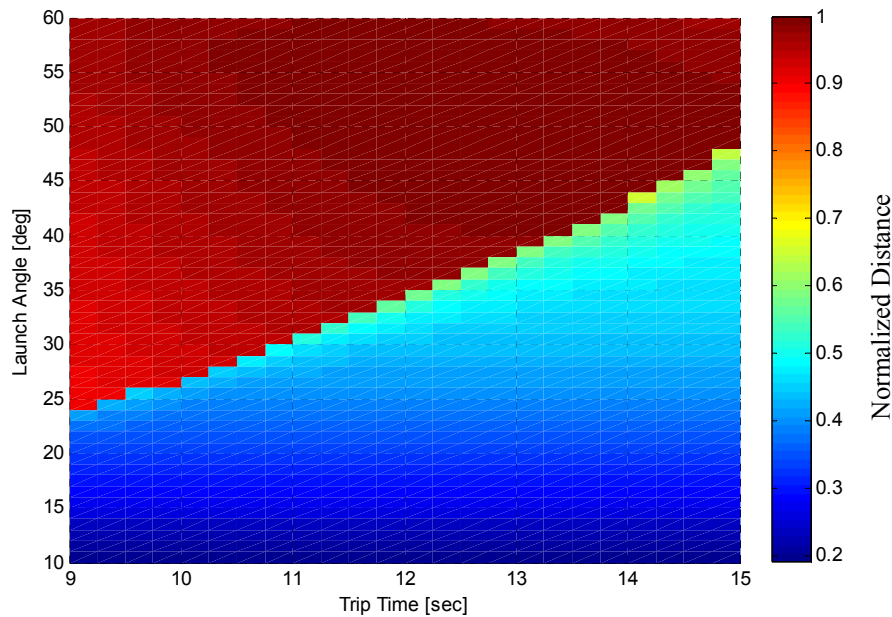


Figure 44: Maximum Range Extension based on Varying Launch Angle and Timer Deployment

Since the body angle wing deployment condition was launched at an angle of 50° , the HP was simulated using a timer condition at the same launch angle. Figure 45 below shows that using a timer value of 11.5 seconds achieved the best range extension; however, any timer value between 9 and 15 seconds will achieve at worst 2% lower range extension. Additionally, the case was considered where launch angle was varied and the performance of the timer deployment method was compared to the performance of the body angle deployment method. In Figure 46 below, the timer value required for maximum range extension was determined for each launch angle. It was assumed that a body angle deployment condition of -10° provided maximum range extension for any launch angle. Based on Figure 42, where the maximum difference between the actual maximum deployment condition and -10° was 6.17% at a launch angle of 28° , and the difference for a launch angle of 50° was 1%, the assumption was validated.

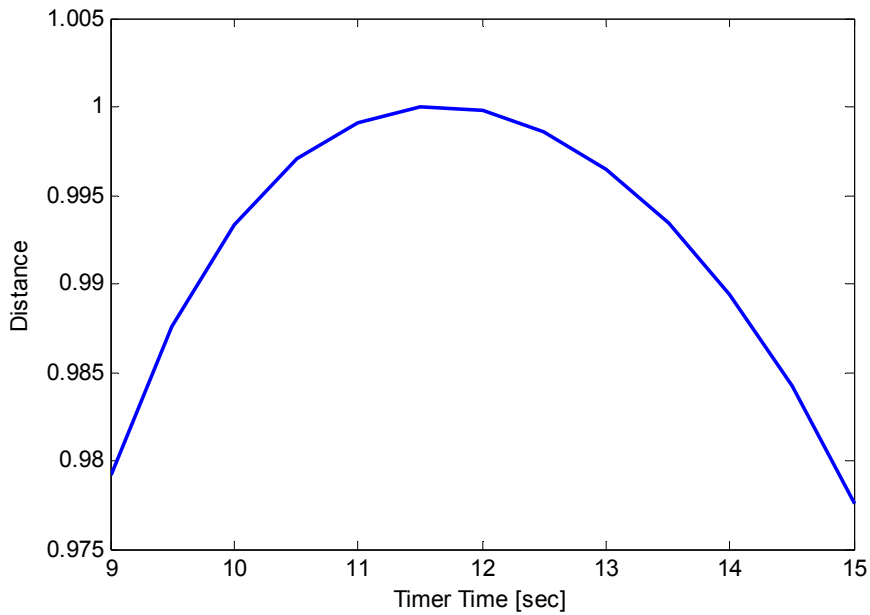


Figure 45: Range Extension based on Varying Timer Wing Deployment Values at a Launch Angle of 50°

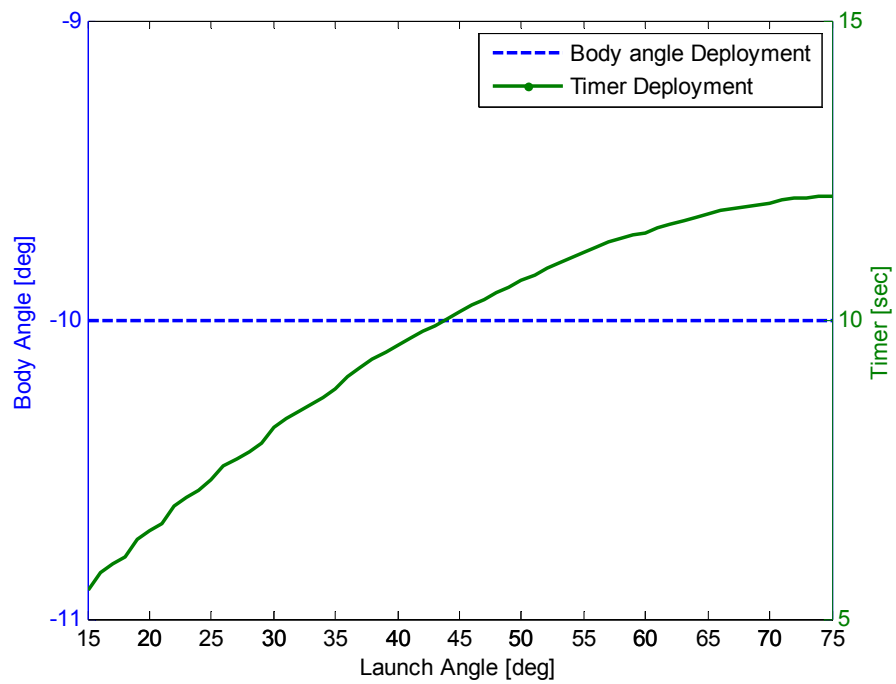


Figure 46: Range Extension for Pitch Deployment vs. Timer Deployment

Figure 46 shows that when the launch angle changes, the timer value required to achieve maximum range extension varies. In order to achieve maximum range extension with the timer

deployment condition, the HP's electronics would have to be reprogrammed before being launched. The body angle deployment condition does not require reprogramming and can be launched at any launch angle, providing control on both the trajectory and maximum distance that the HP would travel. Because of these results, the timer deployment condition was not chosen over the body angle deployment condition. Another concern was the possibility that a strong head wind could push the HP using a timer deployment condition off course significantly enough to reduce overall range extension.

7.4. Head Wind Impact

If the HP were to use a timer deployment method, timing would be critical to ensure that maximum range extension was achieved. It was assumed that a strong head wind could possibly push the HP's trajectory enough off course to even push it backwards if the wings deployed at the incorrect time. A wind model, seen in Figure 47, was constructed in Simulink by using the model calculated DCM and performing matrix multiplication with a wind velocity vector (m/s) to transform the wind velocity to the body reference frame. The wind vector was then subtracted from the model calculated V_b and the new body velocity was propagated through the HP model.

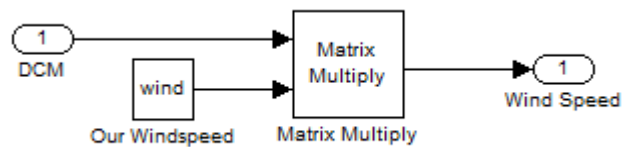


Figure 47: Wind Model Subsystem in 'Calc Forces' Block

Varying wind speeds were then added to the simulation for both body angle deployment and timer deployment conditions. The maximum distance achieved for each wind speed was recorded and can be seen below in Figure 48. Winds between 0 and 15 m/s (a high wind, 7 on the

Beaufort scale) were tested with negligible difference in maximum range extension between the two cases. This shows that non-storm condition winds have little effect on which condition is used to deploy the wings of the HP.

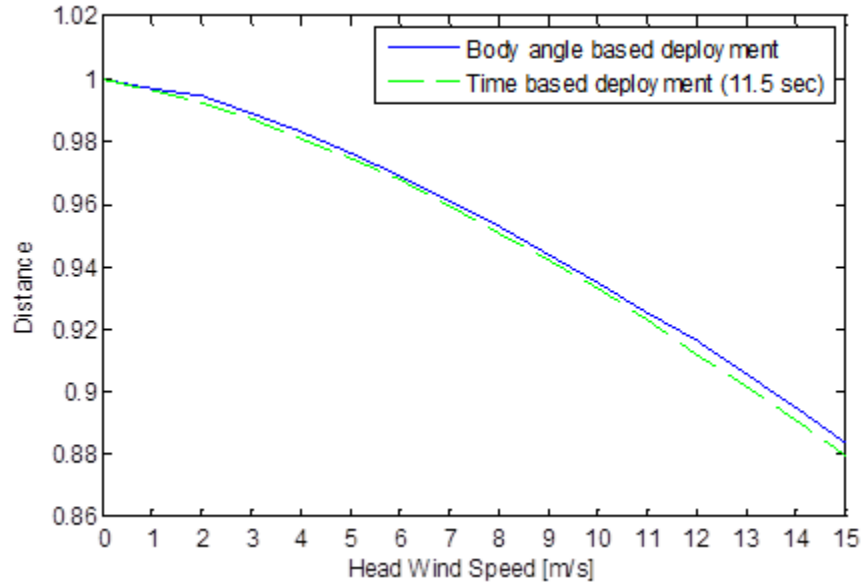


Figure 48: Effect of Head Wind on Body Angle and Timer Wing Deployments at 50° Launch Angle and 126 m/s Initial Velocity

8. Conclusion and Future Work

8.1. Conclusion

In this thesis, a method of wing deployment of a 60mm Hybrid Projectile was analyzed to determine which provided maximum extended range over a regular projectile. A Simulink model was constructed to simulate the dynamics of the HP by calculating the forces acting on it at each time step using predetermined aerodynamic data. At a predetermined point in flight, wings stowed in the body of the HP were deployed. The HP was designed to deploy wings when a pitch angle of -10° was achieved for maximum range extension. Simulations showed that there were

other combinations of body angle wing deployment conditions that provided extended range, but they did not differ by much in regard to the designed parameters. The use of a timer was initially investigated and showed that in situations where the range desired depended on the launch angle the timer would need to be reprogrammed to achieve the desired range. Because of this using a body angle deployment method was considered, which would not require reprogramming before launch. The use of an IMU and a magnetometer would allow body angles to be measured, so it was chosen over the use of a timer.

Sensors were simulated by using model determined states with additive noise. Regarding actual implementation of hardware, determining Euler angles, velocity and position would be done using an Inertial Measurement Unit, a GPS module, and a magnetometer. An estimation process had to be used to remove noise, and an Extended Kalman filter was chosen for its ability to linearize nonlinear systems. The EKF was simulated using a Matlab function block in the Simulink model that ran the update equations whenever measurements were received by the GPS and magnetometer. There was minimal difference between the true states and the EKF estimates as seen by RMS values of [11.4, 4.9, 11.4]m for position, [2.3, 2.9, 10.1]m/s for velocity and [0.4, 0.5, 0.4]deg for Euler angles. However, position was estimated using velocity estimation equations and as a result error was propagated. This caused position estimates to diverge over time, but because the HP was only airborne for duration of less than a minute, the final error values were acceptable for this application of a low-cost expendable projectile.

8.2. Future Work

In addition to the research already conducted in this thesis, the following cases should be considered to benefit the HP project and provide more accurate estimation.

- Side forces, rolling and yawing moments should be added to the model, as the HP would be subjected to them in a real application.
- EKF divergence that was noticed in this thesis should be investigated to determine if a GPS outage would be catastrophic to position and velocity estimation. Studies have been done on EKF divergence and should be considered moving forward. Measurement hardware drift and bias could also be the cause of divergence in this case.
- The implementation of a UKF could be more beneficial if there is enough computing power in the HP hardware. Error values could be reduced and EKF divergence could be avoided altogether.
- Real world sensor calibration techniques and launch forces could cause huge initial error that the EKF may not be able to recover from. The launch effects on the sensors should be studied in order to filter out unwanted disturbances.
- Combine position estimation with control surfaces to add guidance and tracking capabilities.

References

- (Anderson 2012) Anderson, Chris. "ArduPilot - DIY Drones." *ArduPilot*. Chris Anderson, 4 Dec 2012. Web. 14 June 2013.
- (Anderson 2010) Anderson, Chris. "ArduPlane Home Page." DIY Drones, 1 May 2010. Web. 24 June 2013.
- (Browning 2011) Browning P., Cain R., LaBarbara, K., Huebsch, W., Wilhelm, J., "An Experimental Investigation of the Transient Effects Associated with Wing Deployment During Ballistic Flight," SAE Int. Journal of Aerospace, Volume 4, Issue 2, November 2011.
- (Carlucci 2008) Carlucci, Donald E., and Sidney S. Jacobson. "Exterior Ballistics." *Ballistics: Theory and Design of Guns and Ammunition*. Boca Raton: CRC/Taylor & Francis Group, 2008. Print.
- (D. R. Merwe 2000) D. R. Merwe, N. Freitas, E. Wan, "*The Unscented Particle Filter, Tech report CUED/FINFENG/TR-380*", Cambridge University Engineering Department, Cambridge, England August 2000.
- (Gross 2011) Gross, Jason N. "Sensor Fusion Based Fault-Tolerant Attitude Estimation Solutions for Small Unmanned Aerial Vehicles." Diss. West Virginia University, 2011. Print.
- (Gross 2010) Gross, Jason, et al. "A comparison of extended Kalman filter, sigma-point Kalman filter, and particle filter in GPS/INS sensor fusion." *AIAA Guidance, Navigation, and Control Conference*. 2010.
- (Durrant-Whyte 1988) H.F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Trans. Robotics and Automation*, 4(1):23-31, 1988.
- (Hoag 1963) Hoag, David. "Apollo Guidance and Navigation: Considerations of Apollo IMU Gimbal Lock." *Apollo Lunar Surface J* (1963).
- (de Freitas 1998) J. de Freitas, M. Niranjan, A. Gee, and A. Doucet. Sequential Monte-Carlo methods for optimization of neural network models. Technical Report CUESF-INFENGITR-Dept. of Engineering, University of Cambridge, Nov 1998.
- (Jagadish 2007) Jagadish, Chirag, and Bor-Chin Chang. "Diversified redundancy in the measurement of Euler angles using accelerometers and magnetometers." *Decision and Control, 2007 46th IEEE Conference on*. IEEE, 2007.

- (Kalman 1960) Kalman, Rudolph Emil. "A new approach to linear filtering and prediction problems." *Journal of basic Engineering* 82.1 (1960): 35-45.
- (Kaplan 2006) Kaplan, Elliott D., and Christopher J. Hegarty. *Understanding GPS: principles and applications*. Artech House Publishers, 2006.
- (Kuipers 1999) Kuipers, Jack B. *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton, NJ: Princeton UP, 1999.
- (Lepetit 2005) Lepetit, Vincent, and Pascal Fua. *Monocular model-based 3D tracking of rigid objects*. New Publishers Inc, 2005.
- (Liu 1999) Liu, Qingchong. "Doppler measurement and compensation in mobile satellite communications systems." *Military Communications Conference Proceedings, 1999. MILCOM 1999. IEEE*. Vol. 1. IEEE, 1999.
- (Lozano 2010) Lozano, R. "3D Localization." *Unmanned Aerial Vehicles: Embedded Control*. London: ISTE, 2010. 269-70. Print.
- (Manole 2012) Manole, Leon R., Ernest L. Logsdon, Jr., Mohan J. Palathingal, and Anthony J. Sebasto. Gun Launched Hybrid Projectile. The United States of America as Represented by the Secretary of the Army, assignee. Patent 8,115,149. 14 Feb. 2012.
- (Meinhold 1983) Meinhold, Richard J., and Nozer D. Singpurwalla. "Understanding the Kalman filter." *The American Statistician* 37.2 (1983): 123-127.
- (NGDC 2007) NGDC, "Geomagnetism," <http://www.ngdc.noaa.gov/seg/geomag/geomag.shtml>", 2007.
- (Perea 2007) Perea, L., et al. "Nonlinearity in sensor fusion: Divergence issues in EKF, modified truncated SOF, and UKF." *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*. 2007.
- (Premerlani 1987) Premerlani, William, and Paul Bizard. "Direction cosine matrix IMU: Theory." gentlenav.googlecode.com/files/DCMDraft2.pdf, No Date Given, Viewed 26 (2009).
R. Smith and P. Cheesman. On the representation of spatial uncertainty. *Int. J. Robotics Research*, 5(4):56-68, 1987.
- (Smith 1990) R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I.J. Cox and G.T. Wilfon, editors, *Autonomous Robot Vehicles*, pages 167-193. Springer-Verlag, 1990.

- (Ribeiro 2004) Ribeiro, Maria Isabel. "Kalman and extended kalman filters: Concept, derivation and properties." *Institute for Systems and Robotics* (2004): 1-43.
- (Rogers 2003) Rogers, Robert M. *Applied mathematics in integrated navigation systems*. AIAA, 2003.
- (Roskam 2003) Roskam, Jan. "Equations of Motion and Axis Systems." *Airplane Flight Dynamics and Automatic Flight Controls*. Ottawa, Kan.: Roskam Aviation and Engineering, 2003. 19. Print.
- (Shabana 1994) Shabana, Ahmed A. "Euler Angles." *Computational Dynamics*. New York: Wiley, 1994. 388-91. Print.
- (Simon 2001) Simon, Dan. "Kalman filtering." *Embedded Systems Programming* 14.6 (2001): 72-79.
- (St-Pierre 2004) St-Pierre, Mathieu, and Denis Gingras. "Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system." *Intelligent Vehicles Symposium, 2004 IEEE*. IEEE, 2004.
- (Theil) Theil, Stephan. "GPS-IMU-EKF Navigation."
- (Titterton 2004) Titterton, David. *Strapdown inertial navigation technology*. Vol. 17. IET, 2004.
- (Sparkfun 2013) *Triple Axis Accelerometer Breakout- ADXL345t*. Sparkfun Electronics, n.d. Web. 25 Jan. 2013.
- (Walchko 2002) Walchko, Kevin J., and Paul AC Mason. "Inertial navigation." *Proceedings of Florida Conference on Recent Advances in Robotics*. 2002.
- (Walker 2008) Walker, Jearl, David Halliday, and Robert Resnick. "Projectile Motion." *Fundamentals of Physics*. Hoboken, NJ: Wiley, 2008. 65-67. Print.
- (Wan 2000) Wan, Eric A., and Rudolph Van Der Merwe. "The unscented Kalman filter for nonlinear estimation." *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. IEEE, 2000.
- (Welch 2006) Welch, Greg, and Gary Bishop. "An Introduction to the Kalman Filter." *An Introduction to the Kalman Filter*. University of North Carolina at Chapel Hill, 24 July 2006. Web. 27 Nov. 2012. <<http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html>>.

(Wendel 2006) Wendel, J., O. Meister, C. Schlaile, and G. Trommer. "An Integrated GPS/MEMS-IMU Navigation System for an Autonomous Helicopter." *Aerospace Science and Technology* 10.6 (2006): 527-33.

(Wilhelm 2012) Wilhelm, J. P., Jackson, E.R., Browning, P., & Huebsch, W. "Flight Simulation of a Hybrid Projectile to Estimate the Impact of Launch Angle on Range Extension." *Proceedings of the ASME 2012 International Mechanical Engineering Congress & Exposition*. Proceedings of the ASME 2012 International Mechanical Engineering Congress & Exposition IMECE 2012, Houston.

(Woodman 2007) Woodman, Oliver J. "An introduction to inertial navigation." *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696* (2007).

(Zhang 2006) Zhang, J, Zhang, K, Grenfell, R and Deakin, R 2006, 'On the relativistic Doppler effect for precise velocity determination using GPS', *Journal Of Geodesy*, vol. 80, pp. 104-110.

Appendix A- Matlab Code

The following code was written in the Matlab function block that was implemented in the Sensor State Estimation/EKF block to. It carried out the EKF update equations when a GPS and magnetometer measurement was received.

```
%The purpose of this function block is to design an EKF to filter
%the results of the IMU measurements, estimate the state values
%reduce the effects of noise.

function [pos,vel,roll,pitch,yaw] = ekf(u,x0,P0,Ts,gps,gpsTs,mag)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Inputs:
% u- IMU input
% x0- initial state values [acc, bodyang, pos]
% P0- initial error covariance matrix
% Ts- EKF sample time
% gps- gps measurements
% gpsTs- gps clock
% mag- magnetometer measurements

%Outputs:
% pos- estimated position
% vel- estimated velocity
% roll- estimated roll
% pitch- estimated pitch
% yaw- estimated yaw

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-----Initialize parameters-----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

m= 1.68;          %mass of projectile in kg
g= 9.81;

persistent P Q R
persistent xhat oldt
persistent p q r
persistent theta phi psi
persistent ax ay az
persistent vxe vye vze
persistent xe ye ze

if isempty(P)
    P= P0;
    Q= diag([0, 0, 0, 7.29e-4, 8.47e-4, 9.36e-4, 1e-7, 1e-7, 1e-7]);
    R= diag([0.01, 0.01, 0.01, 0.0001, 0.0001, 0.0001, 1e-7 1e-7 1e-7]);
    xe= x0(1);
    ye= x0(2);
    ze= x0(3);
    vxe= x0(4);
    vye= x0(5);
```

```

    vze= x0(6);
    phi=  x0(7);
    theta= x0(8);
    psi=  x0(9);
    oldt= 0;
end

%Assign inputs to variables
ax= u(1);
ay= u(2);
az= u(3);
p  = u(4);
q  = u(5);
r  = u(6);

%When GPS measurement comes in
if gpsTs > oldt

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-----EKF Filtering Equations-----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%3x3 zero matrix
C= zeros(3,3);

%Position Jacobian wrt pos
Pos1= eye(3,3);

%Position Jacobian wrt vel
Pos2= [Ts, 0, 0;
       0, Ts, 0;
       0, 0, Ts];

%Velocity Jacobian wrt vel
Vel2= eye(3,3);

%Velocity Jacobian wrt body angles
Vel3= [ Ts*(( sin(psi)*sin(phi) + cos(psi)*sin(theta)*cos(phi))*ay +
          (sin(psi)*cos(phi)-cos(psi)*sin(theta)*sin(phi))*az), Ts*(-
          cos(psi)*sin(theta)*ax + cos(psi)*cos(theta)*sin(phi)*ay +
          cos(psi)*cos(theta)*cos(phi)*az), Ts*(-sin(psi)*cos(theta)*ax - (
          cos(psi)*cos(phi) + sin(psi)*sin(theta)*sin(phi))*ay + (cos(psi)*sin(phi) -
          sin(psi)*sin(theta)*cos(phi))*az);
       Ts*((-cos(psi)*sin(phi) + sin(psi)*sin(theta)*cos(phi))*ay -
          (cos(psi)*cos(phi)+sin(psi)*sin(theta)*sin(phi))*az), Ts*(-
          sin(psi)*sin(theta)*ax + sin(psi)*cos(theta)*sin(phi)*ay +
          sin(psi)*cos(theta)*cos(phi)*az), Ts*( cos(psi)*cos(theta)*ax + (-
          sin(psi)*cos(phi) + cos(psi)*sin(theta)*sin(phi))*ay + (sin(psi)*sin(phi) +
          cos(psi)*sin(theta)*cos(phi))*az);
       Ts*(( cos(theta)*cos(phi))*ay - (cos(theta)*sin(phi))*az)
, -Ts*(cos(theta)*ax + sin(theta)*sin(phi)*ay + sin(theta)*cos(phi)*az)
,
0
];

%Body angles portion of the Jacobian

```

```

Body= [(1+(q*cos(phi))*tan(theta)- r*sin(phi))*tan(theta))*Ts,
((q*sin(phi))*sec(theta)^2+ r*cos(phi))*sec(theta)^2)*Ts, 0;
      (-q*sin(phi)-r*cos(phi))*Ts,
0;
      ((q*cos(phi)- r*sin(phi))*sec(theta))*Ts, ((q*sin(phi) +
r*cos(phi))*sec(theta))*tan(theta))*Ts,
      1];

%Final Jacobian matrix
F= [ Pos1, Pos2, C ;
      C , Vel2, Vel3
      C , C, Body];

%Measurement matrix
H= eye(9);

%Propogate the state and covariance matrices.
xhat= [ xe + Ts*vxe;
      ye + Ts*vye;
      ze - Ts*vze;
      vxe + Ts*(cos(psi)*cos(theta)*ax + (-sin(psi)*cos(phi) +
cos(psi)*sin(theta)*sin(phi))*ay + ( sin(psi)*sin(phi) +
cos(psi)*sin(theta)*cos(phi))*az);
      vye + Ts*(sin(psi)*cos(theta)*ax + ( cos(psi)*cos(phi) +
sin(psi)*sin(theta)*sin(phi))*ay + (-cos(psi)*sin(phi) +
sin(psi)*sin(theta)*cos(phi))*az);
      vze + Ts*(
cos(theta)*sin(phi))*ay + (
cos(theta)*cos(phi))*az - g
);
      phi + Ts*(p + q*sin(phi))*tan(theta) + r*cos(phi))*tan(theta));
      theta + Ts*(q*cos(phi) - r*sin(phi));
      psi + Ts*(q*sin(phi) + r*cos(phi))*sec(theta)];

%Numerical Error avoidance
P= F*P*F' + Q;

%Find Kalman Gain
K= P*H'*inv(H*P*H' + R);

%Calculate the measurement residual
yhat= xhat;
meas= [gps; mag];
resid= meas - yhat;

%Update state and covariance matrices
xhat= xhat + K*resid;
P= (eye(size(K,1))-K*H)*P;

else

      xhat= [ xe + Ts*vxe;
      ye + Ts*vye;
      ze - Ts*vze;

```

```

        vxe + Ts*(cos(psi)*cos(theta)*ax + (-sin(psi)*cos(phi) +
cos(psi)*sin(theta)*sin(phi))*ay + ( sin(psi)*sin(phi) +
cos(psi)*sin(theta)*cos(phi))*az);
        vye + Ts*(sin(psi)*cos(theta)*ax + ( cos(psi)*cos(phi) +
sin(psi)*sin(theta)*sin(phi))*ay + (-cos(psi)*sin(phi) +
sin(psi)*sin(theta)*cos(phi))*az);
        vze + Ts*(
            -sin(theta)*ax + (
cos(theta)*sin(phi))*ay + (
            cos(theta)*cos(phi))*az - g
);
        phi + Ts*(p + q*sin(phi)*tan(theta) + r*cos(phi)*tan(theta));
        theta + Ts*(q*cos(phi) - r*sin(phi));
        psi + Ts*(q*sin(phi) + r*cos(phi))*sec(theta)];
end

%Update persistent states
xe= xhat(1);
ye= xhat(2);
ze= xhat(3);
vxe= xhat(4);
vye= xhat(5);
vze= xhat(6);
phi= xhat(7);
theta= xhat(8);
psi= xhat(9);
oldt= gpsTs;

%Update gps clock value

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-----Output Results-----
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Results for body angles
roll= 180/pi*phi;
pitch= 180/pi*theta;
yaw= 180/pi*psi;
pos= xhat(1:3);
vel= xhat(4:6);

end

```

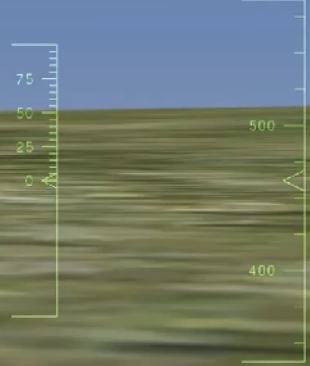
32°52' 46.8N

-114°23' 34.8W

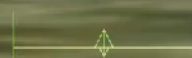
GPS OBS:1 (FROM)
D: 0.9nm TTW:
BRG: 189 XTK: 0.2nm
GS: 0kts



60 50



50 50



40 40



30 30

