Aerospace Mechanics and Controls

A Brief Textbook Presented to the
Student Body of the University of South Alabama

Last Update: October 20, 2025

Current Edition

This manuscript was last updated on October 20, 2025. The latest edition can be found on Github

A note on AI: Note that many sections have been written with the help of Github's Copilot [1] and even Google's Gemini [2]. Originially I had plans to properly cite each section that was written with the help of AI but I lost track given it's seamless integration with VScode and Android smart phones as well as any browser at this point. As such it is important to note that after September of 2025 you can assume that most if not all sections were written with the help of AI.

Manuscript Changes

1. June 10th, 2020 - Magnetic Field Model section updated to reflect the difference between the East North Vertical and the North East Down reference frames. The Figure showing the magnetic field for an example Low Earth Orbit has also been update. Also added this page for manuscript changes and the following pages that list where this file can be found.
2. December 10th, 2020 - Moved to public Github Repo separate from MATLAB
3. December 30th, 2020 - Moved papers.bib to parent directory Added datetime to title page. Added section headers to RC aircraft design. Wrote text all the way through the airfoil selection. Added a references section.
4. December 31st, 2020 - Finished RC aircraft section.
5. September 30th, 2021 - Renamed report to Aerospace Mechanics and Controls
6. December 21st, 2021 - Moved CubeSAT abstract to introduction on CubeSATs. Imported entire Aircraft Mechanics textbook into here
7. December 22nd, 2021 - Added a section on GNC design for CubeSats. Added acknowledgements section.
8. June 2nd, 2022 - Added some items to changes needed and fixed two references
9. July 30th, 2022 - Included a derivation of direct measurement of Euler Angles using an IMU.
10. July 31st, 2022 - Added GPS coordinate conversion to cartesian coodinates as well as heading angle and speed estimation from GPS.
11. October 27th, 2022 - Added computation of lat, lon, alttiude to ECI frame.
12. March 20th, 2024 - Added a Current Edition section above Manuscript Changes. Also added color to hyperlinks
13. May 7th, 2024 - Added a battery sizing section to the aircraft design chapter
14. July 20th, 2024 - separated sections into separate files...Also started adding the quacopter aerodynamic model
15. July 22nd, 2024 - Finished the quadcopter aerodynamics section and made a quick edit to the GNC aircraft PID control scheme
16. December 23rd, 2024 - Added a better description to the gravity model to explain the vector notation of the equations
17. January 6th, 2025 - A title change was performed for the GPS to Cartesian coordinate transform. Also made orbital elements its own section.
18. January 9th, 2025 - Edited the position of different sections so that it's ready for the linear controls section
19. May 13th, 2025 - Added drag equation to aerodynamics
20. September 7-8th, 2025 - Used Github's copilot on the Mobile App to create the first order example of a differential equation. Also used Gemini as well to create the second order example. I also added a citation for Gemini in the papers.bib file. I also edited the foreword to include a note on AI since from here on out I will use it to help write many sections of this manuscript.
21. September 10th, 2025 - Added the equations of motion for a second order mass spring damper system.
22. September 28th, 2025 - Moved the section on numerical integration to be before the LTI section since it makes more sense to have it there. Also added the second order solution showing underdamped, overdamped and critically damped solutions.
23. September 29th, 2025 - Added the time response section for first order systems and finished the derivation of first and second order systems.
24. September 30th, 2025 - added more section headers to the LTI section to make it easier to navigate.
25. October 10th, 2025 - Split the LTI Controls section into its own file and started writing the feedback control section.
26. October 11th, 2025 - Added the attitude dynamics of a satellite/quadcopter section. Also rearranged the sections overall

27. October 12th, 2025 - Finished the rocket dynamics equations and started on the stability section.
28. October 13th, 2025 - Added the pitch dynamics of an aircraft
29. October 14th, 2025 - Finished the dynamics section of LTI systems and am working on the time response section now
30. October 15th, 2025 - Added the week by week schedule for the project based learning section
31. October 18th, 2025 - Added the code for simulating the mass spring damper system
32. October 19th, 2025 - Added stability sections for mass spring damper and the car velocity/position
33. October 20th, 2025 - Added the time response and stability sections for the quadcopter/satellite attitude dynamics

Changes Needed

1. Aircraft Changes

    (a) Include some plots on RC aircraft design
    (b) Add an example RC aircraft design

2. Spacecraft Changes

    (a) Add some work on Kerbal Space Program
    (b) Discuss how to get position and velocity from orbital elements and orbital elements from position and velocity.
    (c) Need to explain the difference between Geodetic and Geocentric coordinates
    (d) Derivation of ground path taking into account orbital precession, rotation of the Earth and swath angle from projection of a satellite onto the Earth. See pdf from the Air Force. A Google search will hopefully turn up the paper I'm thinking of.
    (e) Consider adding the section on pointing analysis

3. General Changes

    (a) Add an abbreviations section
    (b) Add Project Based learning to this manuscript
    (c) Direct the reader to my Instrumentation textbook to build a datalogger to put on an airplane or rocket.
    (d) Direct reader to FASTkit to run dynamics in simulation
    (e) Include some simulation results of aircraft, rocket and satellite from FASTkit.
    (f) Need to add derivation of a complimentary filter using transfer functions
    (g) Need to finish discrete dynamics section
    (h) Need example ID and ADs
    (i) Need to add parallel axis theorem for inertia computation
    (j) LTI Sections for general system dynamics
    (k) Add GNC for all vehicles: spacecraft, rocket, aircraft, quadcopter

Acknowledgements

Contents

1   Introduction

This report represents aerospace mechanics and controls for CubeSats, quadcopters and aircraft. A CubeSAT is a small satellite on the order of 10 centimeters along each axis. A 1U satellite is a small cube with 10 cm sides. These satellites are used for a variety of missions and created by a variety of different organizations. When deployed from a rocket, a CubeSAT may obtain a large angular velocity which must be reduced before most science missions or communications can take place. Maximizing solar energy charging also involves better pointing accuracy. To control the attitude of these small satellites, reaction wheels, magnetorquers and even the gravity gradient are used in low earth orbit (LEO) while reaction control thrusters are typically used in deep space. On a standard LEO CubeSAT, 3 reaction wheels are used as well as 3 magnetorquers. In the initial phase of the CubeSAT mission, the magnetorquers are used to reduce the angular velocity of the satellite down to a manageable level. Once the norm of the angular velocity is low enough, the reaction wheels can spin up reducing the angular velocity to zero. At this point a Sun finding algorithm is employed to find the Sun and fully charge the batteries. In LEO two independent vectors are obtained, the Sun vector and the magnetic field vector, to determine the current attitude of the vehicle which is typically called attitude determination. Other sensors such as horizon sensors, star trackers and even lunar sensors can be used to obtain the quaternion of the vehicle. This paper investigates the necessary mathematics to understand the intricacies of guidance, navigation and control specifically discussing the attitude determination and controls subsystem (ADACS).

2   Nomenclature

x, y, z	components of the mass center position vector in the inertial frame (m)
ϕ, θ, ψ	Euler roll,pitch, and yaw angles (rad)
q_0, q_1, q_2, q_3	quaternions
u, v, w	components of the mass center velocity vector in the body frame (m/s)
p, q, r	components of the mass center angular velocity vector in the body frame (rad/s)
$\vec{\omega}_{B/I}$	angular velocity vector of the vehicle in the body frame (rad/s)
$\mathbf{T}_{IB}$	rotation matrix from frame I to frame B
$\mathbf{H}$	relationship between angular velocity components in body frame and derivative of Euler angles
m	mass (kg)
I	mass moment inertia matrix about the mass center in the body frame $(kg - m^2)$
X, Y, Z	components of the total force applied to CubeSAT in body frame (N)
L, M, N	components of the total moment applied to CubeSAT in body frame (N-m)
$\vec{r}_{A \rightarrow B}$	position vector from a generic point A to a generic point B (m)
$\vec{V}_{A/B}$	velocity vector of a generic point A with respect to a generic frame B (m/s)
$\mathbf{S}(\vec{r})$	skew symmetric matrix operator on a vector. Multiplying this matrix by a vector is equivalent to a cross product
X_i, Y_i, Z_i	components of the total force applied to aircraft i in body frame(N)
L_i, M_i, N_i	components of the total moment applied to aircraft i in body frame(N-m)
X_{Wi}, Y_{Wi}, Z_{Wi}	total weight force applied to aircraft i (N)
L, D	Lift and Drag on Aircraft (N) - Not to be confused with Roll moment
g	gravitational constant on Earth (m/s^2)
ρ	atmospheric density(kg/m^3)
S_i	reference area of wing on aircraft i (m^2)
b_i	Wingspan of aircraft i (m)
$\bar{c}_i$	mean chord of wing on aircraft i (m)
α	Angle of attack (rad)
β	Slideslip angle (rad)
C_L, C_D, C_m	Lift, Drag and Pitch Moment coefficients
$\delta_t, \delta_a, \delta_r, \delta_e$	thrust, aileron, rudder, and elevator control inputs(rad)
$S_B(\vec{r})$	skew symmetric matrix operator on a vector expressed in the body frame.
K_p, K_d, K_I	proportional, derivative, and integral control gains
V	Total airspeed (m/s)
$\hat{p}, \hat{q}, \hat{r}$	Non-dimensional angular velocities
l	Distance from center of mass to aerodynamic center of the tail (m)
l_t	Distance from aerodynamic center of main wing to aerodynamic center of tail (m)
α_0	zero lift angle of attack (rad)
C_{L0}	Zero angle of attack lift coefficient
$C_{m\alpha}$	Pitch moment curve slope versus α
$C_{L\alpha}$	Lift curve slope
C_{mq}	Pitch damping coefficient
$C_{m\delta_e}$	Pitch moment curve slope versus elevator deflection angle
a_∞	Speed of sound (m/s)
μ_∞	Viscosity of Fluid $kg/(m - s)$

3 Particle Dynamics

3.1 Systems of Particles

For this formulation we start with Newton's Second Law with no approximations. Similar dynamic forumlations can be found in [3, 4, 5, 6].

$$\sum_{i=0}^{N} \vec{F}_{ji} = \frac{d\vec{p}_j}{dt} \tag{1}$$

where $\vec{p}_j$ is the momentum of a particle. $\vec{F}_{ji}$ is a force on the particle. The statement above states that sum of all forces on a particle is equal to the time rate of change of momentum. If two particles are then considered the equation can be written for both particles.

$$\sum_{i=0}^{N} \vec{F}_{1i} + \vec{f}_{12} = \frac{d\vec{p}_1}{dt} \qquad \sum_{i=0}^{N} \vec{F}_{2i} + \vec{f}_{21} = \frac{d\vec{p}_2}{dt} \tag{2}$$

Note that the forces $\vec{f}_{12}$ and $\vec{f}_{21}$ are internal forces experienced by each particle exerted on each other since they are rigidly connected. Newton's Third Law states that for every action there is an equal and opposite reaction. That is, $\vec{f}_{12} = -\vec{f}_{21}$. Thus, if both equations are added the following equation is created

$$\sum_{j=0}^{P}\sum_{i=0}^{N} \vec{F}_{ji} = \sum_{j=0}^{P} \frac{d\vec{p}_j}{dt} \tag{3}$$

where P is the number of particles. Typically the double summation in F is written just as $\vec{F}$.

3.2 Rotational Dynamics for Systems of Particles

Note that by construction, a system of particles rigidly connected can now rotate about a center point. The center of mass of a system of particles can be defined using the relationship below

$$\vec{r}_C = \frac{1}{m}\sum_{j=0}^{P} m_j \vec{r}_j \tag{4}$$

where

$$m = \sum_{j=0}^{P} m_j \tag{5}$$

This vector can then be used to create rotational dynamics starting with the linear dynamics.

$$\sum_{j=0}^{P}\sum_{i=0}^{N} \mathbf{S}(\vec{r}_{Cj})\vec{F}_{ji} = \vec{M}_C = \sum_{j=0}^{P} \mathbf{S}(\vec{r}_{Cj})\frac{d\vec{p}_j}{dt} \tag{6}$$

where $\mathbf{S}(\vec{r}_{Cj})$ is the skew symmetric matrix of the vector from the center of mass to the jth particle which results in a cross product. The skew symmetric operator is denoted by $\mathbf{S}()$.

$$\mathbf{S}(\vec{r}_{Cj}) = \begin{bmatrix} 0 & -z_{Cj} & y_{Cj} \\ z_{Cj} & 0 & -x_{Cj} \\ -y_{Cj} & x_{Cj} & 0 \end{bmatrix} \tag{7}$$

4 Rigid Bodies

At this point, many assumptions are made about the system of particles.

1. The mass of each particle or rigid body is constant.

2. An inertial frame is placed at the center of the Earth that does not rotate with the Earth. We assume that the Earth is "fixed" to this point but still rotates. The coordinates of our vehicle though are expressed in this non-rotating inertial frame. This is explained in more detail later.

3. The rigid body is not flexible and does not change shape. That is, the time rate of change of the magnitude of a vector $\vec{r}_{PQ}$ is zero for any arbitrary points P and Q attached to the rigid body.

4.1 Translational Dynamics

Using all of these simplifications, the momentum term on the right can be simplified to

$$\sum_{j=0}^{P} \vec{p}_j = m\vec{v}_{C/I} \tag{8}$$

The derivation of the term above starts by deriving the position of the center of mass as the following equation.

$$\vec{r}_j = \vec{r}_C + \vec{r}_{Cj} \tag{9}$$

Taking one derivative results in the following equation

$$\vec{v}_{j/I} = \vec{v}_{C/I} + \frac{^Bd\vec{r}_{Cj}}{dt} + \mathbf{S}(\vec{\omega}_{B/I})\vec{r}_{Cj} \tag{10}$$

where $\mathbf{S}(\vec{\omega}_{B/I})$ is the skew symmetric matrix of the angular velocity vector which results in a cross product. This equation comes from the derivative transport theorem. Since the body is a rigid body the term $\frac{^Bd\vec{r}_{Cj}}{dt} = 0$ resulting in the equation below

$$\vec{v}_{j/I} = \vec{v}_{C/I} + \mathbf{S}(\vec{\omega}_{B/I})\vec{r}_{Cj} \tag{11}$$

which any dynamicist knows as the equation for two points fixed on a rigid body. This equation can then be substituted into the equation for momentum such that.

$$\sum_{j=0}^{P} \vec{p}_j = \sum_{j=0}^{P} m_j \left(\vec{v}_{C/I} + \mathbf{S}(\vec{\omega}_{B/I})\vec{r}_{Cj}\right) \tag{12}$$

The first term reduces to

$$\sum_{j=0}^{P} m_j\vec{v}_{C/I} = \vec{v}_{C/I} \sum_{j=0}^{P} m_j = m\vec{v}_{C/I} \tag{13}$$

the second term reduces to zero since the sum of all particles from the center of mass is by definition the center of mass and thus zero.

$$\sum_{j=0}^{P} \mathbf{S}(\vec{\omega}_{B/I})m_j\vec{r}_{Cj} = \mathbf{S}(\vec{\omega}_{B/I}) \sum_{j=0}^{P} m_j\vec{r}_{Cj} = 0 \tag{14}$$

Plugging this result for momentum into Newton's equation of motion yields. This is typically called Newton-Euler equations of motion.

$$\vec{F}_C = m \left(\frac{^Bd\vec{v}_{C/I}}{dt} + \mathbf{S}(\vec{\omega})_{B/I}\vec{v}_{C/I}\right) \tag{15}$$

4.2 Rotational Dynamics

Plugging in the expression for two points fixed on a rigid body results in a much different expression. First let's expand the rotational dynamic equations of particles using the assumptions made for a rigid body.

$$\vec{M}_C = \frac{d}{dt} \sum_{j=0}^{P} \mathbf{S}(\vec{r}_{Cj})m_j\vec{v}_{j/I} \tag{16}$$

Then the equation of two points fixed on a rigid body can be introduced to obtain the following equation

$$\vec{M}_C = \frac{d}{dt} \sum_{j=0}^{P} \mathbf{S}(\vec{r}_{Cj})m_j \left(\vec{v}_{C/I} + \mathbf{S}(\vec{\omega}_{B/I})\vec{r}_{Cj}\right) \tag{17}$$

expanding this into two terms yields

$$\vec{M}_C = \frac{d}{dt} \left(\sum_{j=0}^{P} m_j\mathbf{S}(\vec{r}_{Cj})\mathbf{S}(\vec{\omega}_{B/I})\vec{r}_{Cj} + \sum_{j=0}^{P} \mathbf{S}(\vec{r}_{Cj})m_j\vec{v}_{C/I}\right) \tag{18}$$

To simplify this further a useful equality is used for cross products. That is $\mathbf{S}(\vec{a})\vec{b} = -\mathbf{S}(\vec{b})\vec{a}$. The equation above then changes to

$$\vec{M}_C = \frac{d}{dt}\left(\left(-\sum_{j=0}^{P} m_j \mathbf{S}(\vec{r}_{Cj})\mathbf{S}(\vec{r}_{Cj})\right)\vec{\omega}_{B/I} - \mathbf{S}(\vec{v}_{C/I})\sum_{j=0}^{P}\vec{r}_{Cj}m_j\right) \tag{19}$$

Notice, that parentheses were placed around the first term to isolate the angular velocity. This is because the angular velocity is constant across the system of particles. The term on the right has also been altered slightly to isolate the fact that the velocity of the center of mass is independent of the system of particles. With the equation in this form it is easy to see that the term on the right is zero because it is the definition of the center of mass. The equation then reduces to

$$\vec{M}_C = \frac{d}{dt}\left(\sum_{j=0}^{P} m_j \mathbf{S}(\vec{r}_{Cj})\mathbf{S}(\vec{r}_{Cj})^T\right)\vec{\omega}_{B/I} \tag{20}$$

Notice again that minus sign has been removed. The skew symmetric matrix has an interesting property where the transpose is equal to the negative of the original matrix. The term in brackets is a well known value for rigid bodies and is known as the moment of inertia for rigid bodies.

$$\mathbf{I}_C = \sum_{j=0}^{P} m_j \mathbf{S}(\vec{r}_{Cj})\mathbf{S}(\vec{r}_{Cj})^T \tag{21}$$

This results in the kinematic equations of motion for rigid bodies to the simple equation below.

$$\vec{M}_C = \frac{d}{dt}\left(\mathbf{I}_C\vec{\omega}_{B/I}\right) \tag{22}$$

With the equation in this form it is finally possible to carry out the derivative

$$\vec{M}_C = \frac{{}^{B}d(\mathbf{I}_C\vec{\omega}_{B/I})}{dt} + \mathbf{S}(\vec{\omega}_{B/I})\mathbf{I}_C\vec{\omega}_{B/I} \tag{23}$$

The first term requires the chain rule to perform the derivative and can thus result in a time varying moment of inertia matrix and the derivative of angular velocity. Therefore the equation can simply be written as

$$\vec{M}_C = \dot{\mathbf{I}}\vec{\omega}_{B/I} + \mathbf{I}_C\frac{{}^{B}d(\vec{\omega}_{B/I})}{dt} + \mathbf{S}(\vec{\omega}_{B/I})\mathbf{I}_C\vec{\omega}_{B/I} \tag{24}$$

4.3 Inertia Estimation

There are several equations that can be used to compute the moment of inertia depending on the geometry of the vehicle. For this example we will look at a cuboid to demonstrate inertia calculations. Firstly, the total mass m and size (length l, width w, and height h) are required.

$$\begin{aligned} I_x &= \tfrac{m}{12}(l^2 + w^2) \\ I_y &= \tfrac{m}{12}(l^2 + d^2) \\ I_z &= \tfrac{m}{12}(d^2 + w^2) \end{aligned} \tag{25}$$

Where I_x is the cuboid's moment of inertia around the x-axis, It can be seen that the moment of inertia about the y and z-axis are computed similarly, but by using different length parameters. Note that cross products of inertia are obtained by using the parallel axis theorem often caused by solar panels on satellites or payloads on aircraft and quadcopters.

4.4 Aerospace Convention

Aerospace convention involves using the Newton-Euler equations of motion to describe the vehicle[3] as explained in the previous section. Typically the position of the vehicle is written as

$$\mathbf{C}_I(\vec{r}_C) = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} \tag{26}$$

The derivative of the position vector is the velocity vector is then written as

$$\mathbf{C}_I(\vec{v}_{C/I}) = \begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{Bmatrix} \tag{27}$$

However, body frame coordinates are typically used to describe the velocity vector such that

$$\mathbf{C}_B(\vec{v}_{C/I}) = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{28}$$

In order to relate the body frame components of the velocity vector the inertial frame coordinates a transformation matrix is used to give the following equation.

$$\begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{Bmatrix} = [\mathbf{T}_{IB}] \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{29}$$

Note that standard aircraft forces and moments are applied to the body. The forces are typically written as X,Y and Z while the moments are given as L,M and N. They can be written in component form using the equations below.

$$\mathbf{C}_B(\vec{F}_C) = \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = X\hat{I}_B + Y\hat{J}_B + Z\hat{K}_B \tag{30}$$

$$\mathbf{C}_B(\vec{M}_C) = \begin{Bmatrix} L \\ M \\ N \end{Bmatrix} = L\hat{I}_B + M\hat{J}_B + N\hat{K}_B \tag{31}$$

5 Attitude Parameterization of Rigid Bodies

The matrix $\mathbf{T}_{IB}$ is a 3x3 transformation matrix that rotates a vector from the body to the inertial frame. A transformation matrix has the unique property that the inverse of the transformation is just the transpose of the matrix. There are multiple ways to construct this rotation frame and a few ways are discussed in the sections that follow.

5.1 Euler Angles

Euler angle are used to describe 3 unique rotation from the inertial to body frame. They are typically denoted as ψ, θ and ϕ. The order of the rotation can vary however the 3-2-1 sequence is standard for aircraft while the 3-1-3 sequence is standard for spacecraft.

5.1.1 3-2-1 Sequence

The transformation from the inertial frame to the body frame involves three unique rotations. The first is a rotation about the z-axis such that

$$\mathbf{C}_A(\vec{v}_{C/I}) = [\mathbf{T}_{IA}]^T \mathbf{C}_I(\vec{v}_{C/I}) = \begin{bmatrix} cos(\psi) & sin(\psi) & 0 \\ -sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{C}_I(\vec{v}_{C/I}) \tag{32}$$

this rotation is called the yaw or heading rotation. Note that the matrix $[\mathbf{T}_{IA}]$ is a matrix that rotates a vector from the A frame to the inertial frame. The transpose of this matrix rotates a vector from the inertial frame to the A frame. From here the intermediate frame (A frame) is rotated about the y-axis such that

$$\mathbf{C}_{NR}(\vec{v}_{C/I}) = [\mathbf{T}_{ANR}]^T \mathbf{C}_A(\vec{v}_{C/I}) = \begin{bmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{bmatrix} \mathbf{C}_A(\vec{v}_{C/I}) \tag{33}$$

this rotation is called the pitch angle rotation. Finally the (NR) no roll frame is rotated through the x-axis such that

$$\mathbf{C}_B(\vec{v}_{C/I}) = [\mathbf{T}_{NRB}]^T \mathbf{C}_{NR}(\vec{v}_{C/I}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & sin(\phi) \\ 0 & -sin(\phi) & cos(\phi) \end{bmatrix} \mathbf{C}_{NR}(\vec{v}_{C/I}) \tag{34}$$

A Figure of this is shown below.
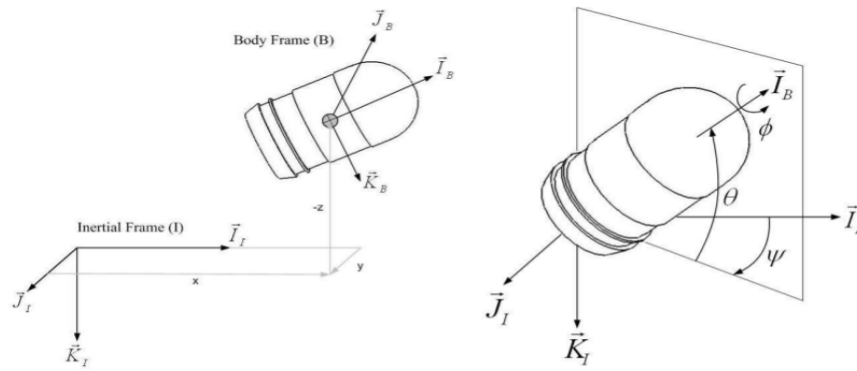
Figure 1: Six Degree of Freedom Schematic

Putting all of these 2-D rotations together creates a transformation matrix from body to inertial.

$$\mathbf{C}_B(\vec{v}_{C/I}) = [\mathbf{T}_{NRB}]^T [\mathbf{T}_{ANR}]^T [\mathbf{T}_{IA}]^T \mathbf{C}_I(\vec{v}_{C/I}) = [\mathbf{T}_{IB}]^T \mathbf{C}_I(\vec{v}_{C/I}) \tag{35}$$

Note again that the inverse of this matrix is given below using the properties of matrix transposes. Standard shorthand notation is used for trigonometric functions: $cos(\alpha) \equiv c_\alpha$, $sin(\alpha) \equiv s_\alpha$, and $tan(\alpha) \equiv t_\alpha$.

$$\mathbf{T}_{IB}(\phi,\theta,\psi) = \mathbf{T}_{IA}\mathbf{T}_{ANR}\mathbf{T}_{NRB} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \tag{36}$$

5.1.2 Derivatives

If Euler angles are used to parameterize the orientation, the derivative of Euler angles is somewhat cumbersome to obtain. The angular velocity of a body is typically written as

$$\vec{\omega}_{B/I} = \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} = p\hat{I}_B + q\hat{J}_B + r\hat{K}_B \tag{37}$$

There are no inertial components for the angular velocity vector. However, a relationship can be derived relating the derivatives of the Euler angles. The angular velocity can be written in vector form such that

$$\vec{\omega}_{B/I} = \dot{\psi}\hat{K}_I + \dot{\theta}\hat{J}_{NR} + \dot{\phi}\hat{I}_B \tag{38}$$

relating the unit vectors $\hat{K}_I$ and $\hat{J}_{NR}$ to the body frame using the planar rotation matrices results in the equation below. Note that NR is denoted as the "No-Roll" frame.

$$\begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} = [\mathbf{H}] \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \tag{39}$$

where

$$\mathbf{H} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi/c_\theta & c_\phi/c_\theta \end{bmatrix} \tag{40}$$

5.1.3 Screw Rotation

It is often useful to extract Euler Angles from a unit vector. A unit vector has two degrees of freedom and thus has two rotations ψ and θ which can be determined using the equation below where $\hat{n}(1)$ denotes the first component of the vector in the body frame.

$$\psi = tan^{-1}\left(\frac{\hat{n}(2)}{\hat{n}(1)}\right); \theta_{Ri} = tan^{-1}\left(\frac{\hat{n}(3)}{\hat{n}(1)^2 + \hat{n}(2)^2}\right) \tag{41}$$

5.1.4 Transformation Matrix to Euler Angles

Besides using unit vectors, sometimes it is beneficial to extract Euler angles from a known tranformation matrix. The equations below can be used to accomplish this where $\mathbf{T}_{BI}(i,j)$ is the ith row and jth column of the $\mathbf{T}_{BI}$ matrix where $\mathbf{T}_{BI} = \mathbf{T}_{IB}^T$

$$\theta = -sin^{-1}(\mathbf{T}_{BI}(1,3)) \quad \phi = tan^{-1}(\mathbf{T}_{BI}(2,3)/\mathbf{T}_{BI}(3,3)) \quad \psi = tan^{-1}(\mathbf{T}_{BI}(1,2)/\mathbf{T}_{BI}(1,1)) \tag{42}$$

5.2 Quaternions

5.2.1 The General Quaternion

It is well known that equations of motion produced by using only three orientation parameters results in a singularity [3]. As such, the orientation of the vehicle can be parameterized using four parameters known as quaternions. Many supplemental equations and explanations can be found for quaternions in [7, 8, 9, 10, 11, 12, 13, 14]. I also

recommend visiting an interactive visualization tool made by popular YouTube star Ben Eater https://eater.net/quaternions. To begin, The standard quaternion is written below.

$$\vec{q} = \begin{Bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{Bmatrix} \tag{43}$$

In this case 4 parameters are used to denote the quaternion. In order to get a physical understanding of what a quaternion is imagine a vector $\vec{\eta}$ in 3-D space. The rotation from the body to the inertial frame is then the rotation of the inertial frame about the unit vector $\vec{\eta}$ through angle γ. The quaternion can then be written as

$$\vec{q} = \begin{Bmatrix} cos(\gamma/2) \\ \vec{\eta} sin(\gamma/2) \end{Bmatrix} \tag{44}$$

In this case it is possible to obtain the individual quaterions as $q_0 = cos(\gamma/2)$ and $\vec{\epsilon} = [q_1, q_2, q_3]^T = \vec{\eta} sin(\gamma/2)$. Furthermore, if given 4 quaternions, the angle γ is simply $cos^{-1}(2q_0)$ and $\vec{\eta} = \vec{\epsilon}/sin(\gamma/2)$. Note that because a quaternion is essentially screw rotation about a known unit vector, there are two identical quaternions for every orientation. That is $\vec{q}(\gamma) = \vec{q}(\gamma - 2\pi)$.

5.2.2 Quaternion Transformations

In order to rotate the inertial frame to the body frame using quaternions, the transformation matrix is shown below. Note that $\mathbf{T}_{BI} = \mathbf{T}_{IB}^T$.

$$\mathbf{T}_{BI}(\vec{q}) = \begin{bmatrix} q_0{}^2 + q_1{}^2 - q_2{}^2 - q_3{}^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0{}^2 - q_1{}^2 + q_2{}^2 - q_3{}^2 & 2(q_0 q_1 + q_2 q_3) \\ 2(q_0 q_2 + q_1 q_3) & 2(q_2 q_3 - q_0 q_1) & q_0{}^2 - q_1{}^2 - q_2{}^2 + q_3{}^2 \end{bmatrix} \tag{45}$$

5.2.3 Euler to Quaternion Transformations

In the event Euler angles are need, converting quaternions to Euler angles is a standard operation and shown below.

$$\begin{aligned} \phi &= tan^{-1}\left(\frac{2(q_0 q_1 + q_2 q_3)}{1 - 2(q_1^2 + q_2^2)}\right) \\ \theta &= sin^{-1}\left(2(q_0 q_2 - q_3 q_1)\right) \\ \psi &= tan^{-1}\left(\frac{2(q_0 q_3 + q_1 q_2)}{1 - 2(q_2^2 + q_3^2)}\right) \end{aligned} \tag{46}$$

It is also possible to convert Euler angles to quaternions using the equations below.

$$\begin{aligned} q_0 &= cos(\phi/2)cos(\theta/2)cos(\psi/2) + sin(\phi/2)sin(\theta/2)sin(\psi/2) \\ q_1 &= sin(\phi/2)cos(\theta/2)cos(\psi/2) - cos(\phi/2)sin(\theta/2)sin(\psi/2) \\ q_2 &= cos(\phi/2)sin(\theta/2)cos(\psi/2) + sin(\phi/2)cos(\theta/2)sin(\psi/2) \\ q_3 &= cos(\phi/2)cos(\theta/2)sin(\psi/2) - sin(\phi/2)sin(\theta/2)cos(\psi/2) \end{aligned} \tag{47}$$

5.2.4 Quaternion Operations

The norm of the quaternions is given by $|\vec{q}| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$. In standard spacecraft applications, the norm of the quaternion is just 1. The conjugate of the quaternion $\vec{q}$ is given below.

$$\vec{q}^* = \begin{Bmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{Bmatrix} \tag{48}$$

The inverse of a quaternion is then just $\vec{q}^{-1} = \vec{q}^*/|\vec{q}|$. Determining the difference between two quaternions is done using the quaternion difference operation as shown below where $|\tilde{q}| = 1.0$ [10].

$$\delta\vec{q} = \vec{q} \oplus \tilde{q}^{-1} = \begin{Bmatrix} q_0 \tilde{q}_0 - \vec{\epsilon}^T \tilde{\epsilon} \\ -q_0 \tilde{\epsilon} + \tilde{q}_0 \vec{\epsilon} - \mathbf{S}(\vec{\epsilon})\tilde{\epsilon} \end{Bmatrix} \tag{49}$$

5.2.5 Quaternion Derivatives

The derivatives of a quaternion are written in shorthand using the equation below.

$$\dot{\vec{q}} = \frac{1}{2}\mathbf{\Omega}(\vec{\omega}_{B/I})\vec{q} = \frac{1}{2}\chi(\vec{q})\vec{\omega}_{B/I} \tag{50}$$

The operators $\mathbf{\Omega}()$ and $\chi()$ are shown below. Note that $\mathbf{\Omega}()$ operates on a 3×1 vector and χ on a 4×1 vector. In this case $\lambda = [\lambda_0, \vec{\kappa}]^T$.

$$\mathbf{\Omega}(\vec{r}) = \begin{bmatrix} 0_{1x1} & -\vec{r}^T \\ \vec{r} & -\mathbf{S}(\vec{r}) \end{bmatrix} \tag{51}$$

$$\chi(\vec{\lambda}) = \begin{bmatrix} -\vec{\kappa} \\ \lambda_0 I_{3x3} + \mathbf{S}(\vec{\kappa}) \end{bmatrix} \tag{52}$$

These vector operators can then be used to expand the kinematic derivatives as shown by equation 53.

$$\begin{Bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{Bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{Bmatrix} \tag{53}$$

where q_i are the four quaternions and p, q, r are the components of the angular velocity vector in the body frame.

6 Aerospace Equations of Motion

The translational equations of motion of the vehicle are written using inertial coordinates using the center of the Earth as a fixed point. For up to cis-lunar orbits this is typically a good approximation for first order analysis. The vehicle is assumed to be a rigid body using quaternions to parameterize orientation.

6.1 Translational Equations of Motion

The translational equations of motion of satellites are fairly simple given that everything is written in the inertial frame. The position vector of the vehicle is $\vec{r} = [x, y, z]^T$ and the velocity is $\vec{V}_{B/I} = [\dot{x}, \dot{y}, \dot{z}]^T$. The acceleration of the vehicle is found by summing the total forces on the body and dividing by the mass of the vehicle. In the equation below $N_{\oplus}$ is the number of planetary bodies acting on the vehicle while $\vec{F}_P$ is the force imparted by thrusters.

$$\vec{a}_{B/I} = \frac{1}{m_s} \left(\sum_{i=1}^{N_{\oplus}} F_i + \vec{F}_P \right) \tag{54}$$

Note that for a spacecraft the magnitude of the gravitational acceleration vector is on the order of $\pm 10\ m/s^2$. Sources point to solar radiation pressure being on the order of 4.5 μPa [15]. For a 1U CubeSat (10 cm x 10 cm) the force would be equal to $0.45mN$. A 1U CubeSat has a nominal mass of 1 kg which would accelerate the CubeSat on the order of $0.45mm/s^2$, which is considerably less than gravitational acceleration. Furthermore, using the standard aerodynamic drag equation ($0.5\rho V^2 S C_D$), where conservative estimates are used, the aerodynamic force at 600 km above the Earth's surface would be about 3.0 nN [16]. This assumes a density equal to $1.03 \times 10^{-14}\ kg/m^3$, a velocity equal to 7.56 km/s, and a drag coefficient equal to 1.0 [17]. A force this small would impart an acceleration of about 3.0 nm/s^2 which is also considerably less than gravitational acceleration. These forces cannot be neglected for longer missions but can be ignored where appropriate. For an aircraft and quadcopter the equations of motion are typically written in the body frame. As such the derivative transport theorem is used and the translational equations of motion are written as the following.

$$\begin{Bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{Bmatrix} = \frac{1}{m} \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} - \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{55}$$

6.2 Reaction Wheel Model

The reaction wheel model must be included before the attitude dynamics because they directly affect the inertia of the vehicle. There are three reaction wheels on this vehicle and each one has it's own angular velocity ω_{Ri} and angular acceleration α_{Ri}. The inertia of each reaction wheel is first written about the center of mass of the reaction wheel and is given by the equation below where the reaction wheel is modeled as a disk with finite radius (r_{RW}) and height (h_{RW}). The subscript R is used to denote that this inertia matrix is about the center of mass of the reaction wheel while the super script R is used to denote the frame of reference.

$$I_{Ri}^R = \begin{bmatrix} m_R r^2/2 & 0 & 0 \\ 0 & (m_R/12)(3r_{RW}^2 + h_{RW}^2) & 0 \\ 0 & 0 & (m_R/12)(3r_{RW}^2 + h_{RW}^2) \end{bmatrix} \tag{56}$$

In order to rotate the inertia matrix into the vehicle body frame of reference an axis of reaction wheel rotation is used. The vector $\hat{n}_{Ri}$ is used to denote the axis about which the reaction wheel rotates. Euler Angles θ_{Ri} and ψ_{Ri} can be extracted from this unit vector as discussed previously in Section 5.1. The rotation matrix $\mathbf{T_{Ri}}(0, \theta_{Ri}, \psi_{Ri})$ can then be generated using equation 36. This matrix can then be used to compute the inertia of the reaction wheel in the vehicle body frame.

$$I_{Ri}^B = \mathbf{T}_{Ri}^T I_R^R \mathbf{T}_{Ri} \tag{57}$$

The parallel axis theorem can then be used to shift the inertias to the center of mass of the vehicle where the subscript RB denotes the reaction wheel inertia taken about the center of mass of the vehicle.

$$I_{RBi}^B = I_{Ri}^B + m_{Ri}\mathbf{S}(\vec{r}_{Ri})\mathbf{S}(\vec{r}_{Ri})^T \tag{58}$$

The vector $\vec{r}_{Ri}$ is the distance from the center of mass of the vehicle to the center of mass of the reaction wheel in the vehicle body reference frame. The total inertia of the entire vehicle-reaction wheel system is then just a sum of

all the reaction wheel inertias.

$$I_S = I_B + \sum_{i=1}^{3} I_{RBi}^{B} \tag{59}$$

The total angular momentum of the vehicle is then equal to the following equation where $\vec{\omega}_{B/I}$ is the angular velocity of the vehicle.

$$\vec{H}_S = I_B \vec{\omega}_{B/I} + \sum_{i=1}^{3} I_{Ri}^{B} \omega_{Ri} \hat{n}_{Ri} \tag{60}$$

In a similar fashion, the total torque placed on the vehicle is equal to the following

$$\vec{M}_R = \sum_{i=1}^{3} I_{Ri}^{B} \alpha_{Ri} \hat{n}_{Ri} \tag{61}$$

It is typically assumed that the angular acceleration of each reaction wheel can be directly controlled. However, as the reaction wheel angular velocity increases, the maximum angular acceleration allowed begins to decrease. Once the reaction wheel reaches its angular velocity limits, the angular acceleration possible drops to zero. This is called reaction wheel saturation and must be dealt with using a method called momentum dumping.

6.3 Attitude Equations of Motion

The attitude equations of motion are formulated assuming the vehicle can rotate about three axes. The derivative of angular velocity is found by equating the derivative of angular momentum to the total moments placed on the vehicle while reaction wheel torques from the vehicle are added.([3]).

$$\dot{\vec{\omega}}_{B/I} = I_S^{-1} \left(\vec{M}_P + \vec{M}_M + \vec{M}_R - \mathbf{S}(\vec{\omega}_{B/I})\vec{H}_S - \dot{I}_S \vec{\omega}_{B/I} \right) \tag{62}$$

The applied moments use subscripts (P) for propulsion, (M) for magnetorquers, and (R) for reaction wheels. The term $\dot{I}_S$ is the change in inertia in the body frame caused by deployment of solar panels and/or antenna. Also, recall that $\vec{H}_S$ is the total angular momentum of the entire vehicle including the reaction wheels if present. For aircraft the rotational dynamic equation can be found as

$$\begin{Bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{Bmatrix} = \mathbf{I}_C^{-1} \left(\begin{Bmatrix} L \\ M \\ N \end{Bmatrix} - \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \mathbf{I}_C \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \right) \tag{63}$$

7 External Models

Many external models are used in simulation to accurately depict the environment. The paper here begins with the Earth Magnetic Field and Gravitational Models. The magnetic field model comes from the Geographic Library model which uses the EMM2015 magnetic field model. The gravitational model comes from the EGM2008 model [18].

7.1 GPS Coordinates to Cartesian Coordinates (Flat Earth Approximation)

All external models below imply a spherical world with an Earth Centered Inertial (ECI) frame at the center of the planet. However, often times for small UAV applications it is useful to convert the GPS coordinates (latitude, longitude, altitude, $(\lambda_{LAT}, \lambda_{LON}, h)$) to a flat earth approximation where the x-axis is pointing North, the y-axis is pointing east and the z-axis is pointing towards the center of the planet. This is similar to spherical coordinates which is explained later on but in this case the axis system is cartesian rather than spherical. This is useful in obtaining the position of the vehicle which can be used to approximate heading and speed which again is explained in another section. The equations to convert LLH (latitude,longitude, altitude) to a cartesian coordinate system are given below. Note that these equations assume that the vehicle creates an origin point to define as the center of the inertial frame which is on the surface of the planet rather than the center of the planet $\lambda_{LAT,0}, \lambda_{LON,0}$. Typically when the vehicle gets its first valid GPS coordinate, that point is set as the origin.

$$
\begin{aligned}
x &= \kappa(\lambda_{LAT} - \lambda_{LAT,0}) \\
y &= \kappa(\lambda_{LON} - \lambda_{LON,0})cos(\tfrac{\pi}{180}\lambda_{LAT,0}) \\
z &= -h
\end{aligned}
\tag{64}
$$

In the equation above $\kappa = 60.0 * (Feet/NauticalMiles) * (Meters/Feet)$ which essentially converts degrees from the LLH coordinates first to nautical miles and then to feet and then to meters. For example, if the vehicle moves North 1 degree that is equivalent to 60 nautical miles on the surface. Vice versa, 1 nautical mile on the surface is equal to one minute or a 60th of a degree in latitude. The conversion from nautical miles to feet is 6076.11548556 and feet to meters is 0.3048. Note that often time is is good to convert the cartesian coordinates back to LLH coordinates. That inversion is shown below.

$$
\lambda_{LAT} = \frac{x}{\kappa} + \lambda_{LAT,0}
\tag{65}
$$

$$
\lambda_{LON} = \frac{y}{\kappa cos(\lambda_{LAT,0}\frac{\pi}{180})} + \lambda_{LON,0}
\tag{66}
$$

$$
h = -z
\tag{67}
$$

7.2 Density Model

The density model is simply given as an exponential model.

$$
\rho = \rho_s e^{-\sigma h}
\tag{68}
$$

where ρ_s is the density at sea-level, h is the altitude above the Earth in kilometers and $\sigma = 0.1354 km^{-1}$ is known as the scale height [19, 20, 21].

7.3 Magnetic Field Model

The Magnetic Field model used in this simulation stems from the Enhanced Magnetic Field Model (EMM2015) ([22]). The Earth's magnetic is a complex superposition of multiple sources including the inner core and outer core of the planet. Models have been created that use spherical harmonics to compute the magnetic field at any location around the Earth. The EMM2015 model uses a 720 order model increasing the spatial resolution down to 56 km. This model was compiled from multiple sources including but not limited to satellite and marine data. It also includes data from the European Space Agency's Swarm satellite mission. In order to include this harmonic mesh data into this simulation, the GeographicLib module written in C++ is included in the simulation ([18]). Note that I take no credit for this model. This section only serves to explain the model. The result of utilizing this model is the ability to provide any position coordinate of the satellite to the module and have the model return the magnetic field strength in East, North, Vertical Coordinates. Specifically, the inputs to the model are the position x, y, z of

the satellite assuming an inertial frame with the z-axis pointing through the north pole and the x axis pointing through the equator at the prime meridian as seen in Figure 2. This is known as the Earth-Centered Inertial (ECI) coordinate system ([23]).
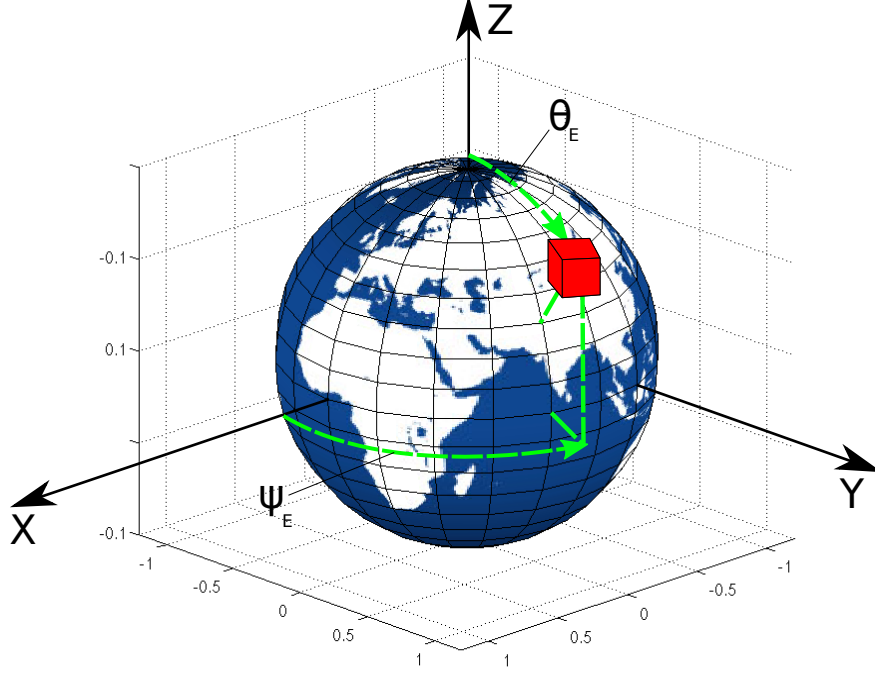
Figure 2: Earth-Centered Inertial Frame and Spherical Coordinate Frame

In order to connect these inertial coordinates (x, y, z) to be used in the EMM2015 model, the latitude, longitude and height above the surface of the Earth are required. To do this, the coordinates are converted into spherical coordinates using the equations below.

$$\rho = \sqrt{x^2 + y^2 + z^2}$$
$$\phi_E = 0$$
$$\theta_E = cos^{-1}\left(\frac{z}{\rho}\right) \quad (69)$$
$$\psi_E = tan^{-1}\left(\frac{y}{x}\right)$$

Note that $\rho, \phi_E, \theta_E, \psi_E$ are related to latitude and longitude coordinates but not quite the same. In order to obtain the latitude and longitude coordinates the following equations are used. The height is simply the distance from the center of the ECI frame minus the reference height from the approximation of Earth as an ellipsoid ($R_\oplus = 6,371,393 \; meters$). Note that the angles from Equation 69 are converted to degrees.

$$\lambda_{LAT} = 90 - \theta_E \frac{180}{\pi}$$
$$\lambda_{LON} = \psi_E \frac{180}{\pi} \quad (70)$$
$$h = \rho - R_\oplus$$

The inputs to the EMM2015 model are the latitude, longitude and height. The inverse of the above two equations are given below. These would be used in the event a latitude and longitude coordinate is given and there is a need to obtain the x,y and z coordinates in the ECI frame. The first step is to convert latitude, longitude and altitude and convert that to standard spherical angles and distance from the center of the planet.

$$\theta_E = (90 - \lambda_{LAT}) \frac{\pi}{180}$$
$$\psi_E = \lambda_{LON} \frac{\pi}{180} \quad (71)$$
$$\rho = h + R_\oplus$$

Once that is complete the extraction of x,y and z are computed by the equation below.

$$\begin{aligned}
x &= \rho sin(\theta_E)cos(\psi_E) \\
y &= \rho sin(\theta_E)sin(\psi_E) \\
z &= \rho cos(\theta_E)
\end{aligned} \tag{72}$$

The output from the EMM2015 model is in the East, North, Vertical (ENV) reference frame where the x-axis is East pointing in the direction of the rotation on the Earth, the y-axis is North pointing towards the North pole and finally the z-axis is the Vertical component that is always pointing radially away from the center of the Earth. In order to get the coordinates into the ECI frame the coordinates must first me converted to the North, East, Down reference frame (NED). In this case the x-axis is pointing North, the y-axis pointing East and the z-axis is always pointing towards the center of the Earth and called Down. The equation to rotate from the ENV frame to NED frame is shown below.

$$\begin{Bmatrix} \beta_x \\ \beta_y \\ \beta_z \end{Bmatrix}_{NED} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{Bmatrix} \beta_x \\ \beta_y \\ \beta_z \end{Bmatrix}_{ENV} \tag{73}$$

Once the magnetic field is in the NED reference frame it can then be rotated to the inertial frame using the following equation where $\vec{\beta}_{NED}$ is the magnetic field in the NED coordinate system and $\vec{\beta}_I$ is the magnetic field in the inertial frame.

$$\vec{\beta}_I = \mathbf{T}_{IB}(0, \theta_E + \pi, \psi_E)\vec{\beta}_{NED} \tag{74}$$

The matrix $\mathbf{T}_{IB}(\phi, \theta, \psi)$ represents the transformation matrix from the spherical reference frame to the inertial reference frame. Note that there is no rotation about the x-axis through ϕ_E and the pitch rotation is augmented by π because of the switch between North, East, Down (NED) and the z-axis of the ECI pointing through the North pole. The result of these equations, is the ability to obtain the magnetic field across an entire orbit. Figure 3 shows an example 56 degree inclination orbit, 600 km above the Earth's surface. The orbit begins with the satellite above the equator and the prime meridian and assumes the Earth does not rotate.
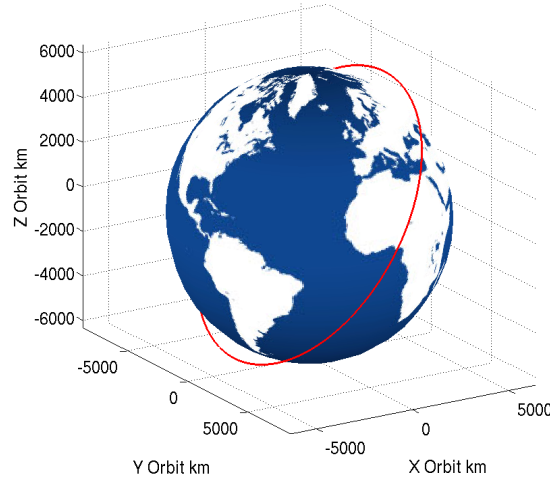
Figure 3: Example 56 Degree Inclination Orbit at 600 km above Earth's Surface

Figure 4 shows the magnetic field during the orbit in the inertial frame. PCI stands for Planet Centered Inertial which in this case is the same as the ECI frame since the planet is Earth.
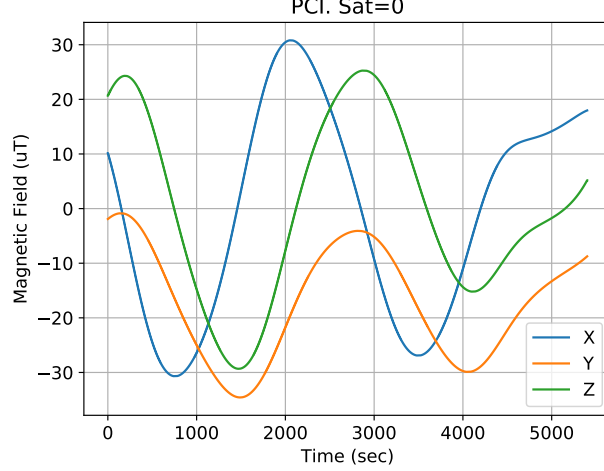
Figure 4: Magnetic Field of Earth in Inertial Frame for 56 Degree Orbit at 600 km Above Surface

For a satellite in LEO, the spacecraft will experience a magnetic dipole moment. The magnetic dipole moment is caused by noting that the structure of the satellite is metal with current that creates its own magnetic field. The magnetic dipole moment torque is given by computing the torque produced by the magnetic field of the Earth interacting with the metallic structure of the satellite. First the dipole constant $d_s = 2.64E - 03\ N - m/T$ is the assumed value for torque as a function of Tesla in LEO. This constant is derived by assuming the torque from this disturbance at 500 km above the surface is the same as the solar radiation torque. Using this constant, the torque is given by the equation below where $\vec{\beta}_I$ is the magnetic field strength of Earth in the inertial frame. The direction of the torque is assumed to be in the same direction of the magnetic field since the structure is not fully modeled. Although not accurate, the goal is to approximate the magnitude as closely as possible.

$$\vec{M}_{MD} = \vec{\beta}_I d_s \tag{75}$$

7.4   Gravitational Models

Three types of gravitational models can be used. The first is the Newtonian gravitational model that assumes all planets are point masses with no volume. The result of the gravitational field vector is then

$$\vec{F}_\oplus = -G\frac{m_\oplus m_s}{r^2}\hat{r} \tag{76}$$

where G is the gravitational constant, $\oplus$ denotes the planet applying the gravitational field, $m_\oplus$ is the mass of the planet, m_s is the mass of the satellite and $\vec{r}$ is a distance vector from the center of the planet to the satellite. The vector $\hat{r}$ is just the unit vector of $\vec{r}$ while r is the magnitude of $\vec{r}$.

$$\hat{r} = \vec{r}/r \tag{77}$$

In component form, $\vec{r} = [x; y; z]^T$. Substituting that component form into the two equations above results in the component form of the gravity model which can be better suited for non-vectorized programming syntax.

$$\begin{Bmatrix} F_{x\oplus} \\ F_{y\oplus} \\ F_{z\oplus} \end{Bmatrix} = -G\frac{m_\oplus m_s}{r^3} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} \tag{78}$$

The second gravitational field model stems from the Earth Gravity Model (EGM2008) [24] which can also be found in the GeographicLib module [18]. This model compute's Earths gravitational field at any point in three dimensional space. The model takes in coordinates in the ECI frame and returns the gravitational acceleration in the ECI frame thus no rotation is required. Just like the EMM2015 model this model uses spherical harmonics and a reference ellipsoid. The reference ellipsoid is then updated with gravity disturbances such as non-uniform geoid heights. This model is an upgrade from EGM84 and EGM96 which only used models of order 180 and 360 respectively. The

EGM2008 model as a comparison uses a model of order 2190. Figure 5 shows the gravitational acceleration vector during a 56 degree orbit at 600 km above the Earth's surface. The x-axis has been non-dimensionalized to represent the entire orbit. Thus when the x-axis is equal to 100 the satellite has completed one orbit.
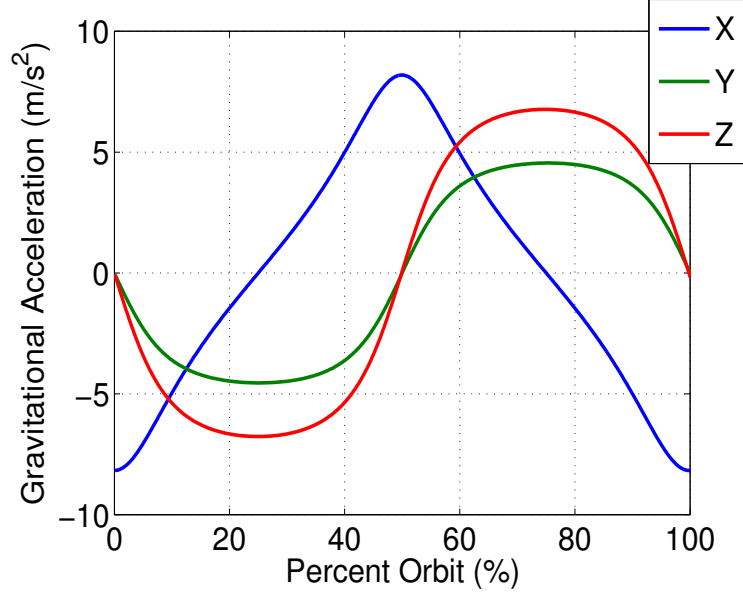
Figure 5: Gravitational Field of Earth in Inertial Frame for 56 Degree Orbit at 600 km Above Surface

For a satellite in LEO, the vehicle will experience a gravity gradient torque. The gravity gradient torque is given by computing the gravitational force at one end of the satellite and the other denoted as $\vec{F}(\vec{r}_B)$ and $\vec{F}(\vec{r}_T)$ for bottom and top respectively. The torques are then crossed with the distance from the center of mass to the top of the satellite. It is assumed that the satellite is symmetric and thus, the torque is just the difference between the two forces crossed with the vector from the CG to one side of the satellite.

$$\vec{M}_G = \mathbf{S}(\vec{F}(\vec{r}_B) - \vec{F}(\vec{r}_T))\vec{r}_{CG-B} \tag{79}$$

The third gravitational field model assumes the vehicle is close to the surface such as a quadcopter or aircraft. In this case the gravitational field is held at a constant and equal to 9.81 m/s^2.

$$\begin{Bmatrix} X_W \\ Y_W \\ Z_W \end{Bmatrix} = mg \begin{Bmatrix} -s_\theta \\ s_\phi c_\theta \\ c_\phi c_\theta \end{Bmatrix} \tag{80}$$

8 External Forces and Moments

In addition to gravity acting on a vehicle, other forces also act on the satellite. For a 1U CubeSat, the gravity gradient over 10 cm is about 0.24 $\mu m/s^2$ using the EGM2008 model. Multiplying this acceleration by a 1 kg mass and applying a 10 cm moment arm yields a moment of about 2.4×10^{-8} $N-m$. Aerodynamic torques could be as large as 1.5×10^{-10} $N-m$ assuming the aerodynamic center is 5 cm away from the center of mass. Typical magnetorquers operate in the vicinity of 3.0×10^{-6} $N-m$, assuming a current of 0.04 A, an area of 0.02 m^2, 84 turns and a magnetic field of $40,000$ nT. Using these calculations, magnetorquers are two orders of magnitude larger than gravity torques and four orders of magnitude larger than aerodynamic torques. It is important to keep these values in mind when neglecting certain parameters [15, 16, 17].

8.1 Solar Radiation Pressure

Solar radiation pressure is relatively constant at 1 AU and thus is simply given as $p_s = 4.5e - 6$ Pa. The force is then found to be just the pressure multiplied by the frontal area of the satellite. The torque, similar to the aerodynamic torque, is the force crossed with a distance vector from the center of mass to the center of pressure of solar radiation. The vector $\hat{s}$ is a unit vector denoting the direction of the sun.

$$\vec{F}_{SR} = p_s S \hat{s} \tag{81}$$

$$\vec{M}_{SR} = \mathbf{S}\left(\vec{F}_{SR}\right)\vec{r}_{CG-CPs} \tag{82}$$

8.2 Propulsion Model

In order for a vehicle to lift off to enter space, engineers must be able to apply a force that is greater than the force acting on the vehicle due to gravity and aerodynamics. The applied force is known as thrust. Thrust can be generated by the propulsion system of the vehicle. Electric and chemical are two well-known methods to produce thrust which take advantage of Newton's third law of motion.

Electric Propulsion Systems typically use electric heating or electric or magnetic fields to accelerate propellants (usually gases). These systems can be very fuel-efficient, however it does not generate enough thrust. These engines are great for deep space exploration where transit times can be very long and rapid maneuvers are not required[25].

Chemical propulsion systems are more effective in our environment. These systems involve the use of chemical reactions to release energy and accelerate gases to produce thrust. Chemical propulsion is a broad category and can be subdivided into liquid propulsion, solid propulsion, and hybrid propulsion.

Liquid propulsion systems can be further subdivided into either a monopropellant (a single propellant fluid) or a bi-propellant (two fluids, which includes fuel and an oxidizer). The simplest form of fuel and oxidizer would be liquid hydrogen and liquid oxygen. Typically, the propellants may be kept on board and fed from high-pressure tanks (pressure-fed) or use turbopumps to move the propellant to the combustion chamber (pump-fed) before the hot exhaust exits the nozzle. Liquid Propulsion systems can produce a wide range of thrust, can have high specific impulse (Isp), and can be easily controlled; but often must be fueled shortly prior to launch.

On the contrary, solid rocket motors (or SRMs) are simple devices. The propellants, the fuel and oxidizer, are mixed together and stored in a cylinder. An electrical signal is sent to the igniter which creates hot gases to ignite the main propellant grain. By converting the high thermal energy of the gases into kinetic energy, therefore thrust is developed. These motors usually have a relatively short burn time. For example, The Thiokol motor using ammonium perchlorate/aluminum as propellant, has a burn time of 75 s with a thrust of 3,300,000 lb.

Even though solid rocket motors are simple and can be ignited in a moment's notice, their Isp (specific impulse) is generally lower than liquid systems. Also, they cannot be readily throttled. Once ignited, the motor will burn to extinction[26].

It is important to note, however, that if propulsion is needed for the spacecraft it is necessary to work with the propulsion team to determine the ΔV, mass flow rate, and attitude control. For this analysis each satellite is equipped with N_P thrusters that have a fixed I_{sp}. The mass flow rate of each thruster is given by the equation below where p is the force of the thruster.

$$\dot{m}_i = \sigma_i \frac{p}{9.81 \, I_{sp}} \tag{83}$$

Each thruster is either on or off as given by the variable σ which is either a 1 or a 0. When the thruster is on, the force applied is equal to p and when the thruster is off the thrust applied is equal to zero. Thus in this fashion to

total mass flow rate per unit time of the entire satellite is just a sum of all the pulses.

$$\dot{m} = \frac{p}{9.81 \ I_{sp}} \sum_{i=1}^{N_P} \sigma_i \tag{84}$$

It is assumed that the time response of the thrusters is instantaneous during power up and power down. There is a delay between pulses and the thrusters only stay on for a fixed time thus the thrusters are pulsed in a square wave fashion with a certain duty cycle. The force applied is simply equal to the force times a unit vector that is aligned with the axis of the thruster. The total force applied to each satellite is then given by the formula below.

$$\vec{F}_P = p \sum_{i=1}^{N_P} \sigma_i \hat{n}_{Pi} \tag{85}$$

The total moment applied to the satellite is simply the force applied crossed with a vector from the center of mass of the satellite to the center of mass of the thruster.

$$\vec{M}_P = p \sum_{i=1}^{N_P} \sigma_i \mathbf{S}(\vec{r}_{Pi}) \hat{n}_{Pi} \tag{86}$$

Note that the total thrust equation is given as

$$T = \dot{m} V_e + A(p_e - p_0) \tag{87}$$

where V_e is the exhaust velocity, A is the area of the nozzle, and p_e and p_0 are the pressures at the exit of the nozzle and the ambient environment. It's easy to see that the thrust at sea-level would be lower than the thrust in a vacuum since p_0 is much higher at sea-level and almost zero in a vacuum. For typical hobbyist rockets it is assumed that $p_e = p_0$ and thus the thrust is simply equal to $\dot{m} V_e$. The exit velocity is given as $I_{sp} g_0$ where g_0 is the standard gravity of 9.81 m/s^2. This means that the mass flow can be given as

$$\dot{m} = \frac{T}{I_{sp} g_0} \tag{88}$$

which is similar to the equation explained above. Now in order to obtain the delta V of the satellite, the rocket equation is used. The rocket equation is given as

$$T = m \frac{dV}{dt} \tag{89}$$

where the sum of the forces is equal to the time rate of velocity. Replacing the left side of the equation with the thrust equation gives

$$-\frac{dm}{dt} I_{sp} g_0 = m \frac{dV}{dt} \tag{90}$$

Note that mathematically the mass flow rate is negative since the mass of the satellite is decreasing as the propellant is being expelled which is why a minus signs was added. We can then combine like terms and cancel out the dt on both sides to obtain

$$-I_{sp} g_0 \frac{dm}{m} = dV \tag{91}$$

Note that mathematicians would scoff if they read that we are dividing dt since dt is infinitesmially small and cancelling out the dt is similar to dividing by zero. The proper way to "cancel" the dt term would be to perform a change of variables but ultimately you arrive at the same solution. Thus an an engineer I would say "close enough. just like an aymptote." Where a mathematician would say "an asymptote never approaches zero" an engineer like myself would say "but it's basically zero" and move on. So, moving on, integrating both sides gives

$$-I_{sp} g_0 \int_{m_i}^{m_f} \frac{dm}{m} = \int_{V_i}^{V_f} dV \tag{92}$$

which reduces to

$$I_{sp} g_0 \ln\left(\frac{m_i}{m_f}\right) = V_f - V_i = \Delta V \tag{93}$$

where the minus sign was removed by inverting the mass ratio which is a property of natural logarithms. The equation above is the standard rocket equation cited in many textbooks and sources [25, 26].

8.3 Magnetorquer Model

The magnetorquer model assumes that three magnetorquers are aligned in such a way that the magnetic moment produced by each magnetorquer is aligned with the principal axes of the body frame of the satellite. Each magnetorquer is controlled independently such that $\vec{i}_M = [i_x, i_y, i_z]^T$ which is the applied current in each magnetorquer. The magnetic moment is then given by the equation below

$$\vec{\mu}_M = nA\vec{i}_M \tag{94}$$

where n is the number of turns in the coil of each magnetorquer and A is the area of the magnetorquer. For simplicity it is assumed that all magnetorquers have the same area and same number of turns. The torque produced by all magnetorquers is then simply found by crossing the magnetic moment with the magnetic field of the Earth in the Body reference frame.

$$\vec{M}_M = \mathbf{S}(\vec{\mu}_M)\mathbf{T_{BI}}(\vec{q})\vec{\beta}_I \tag{95}$$

In order to obtain the magnetic field vector in the body frame, the inertial magnetic field vector must be rotated into the body frame of the satellite. In component form, equation (95) reduces to the following equation using the identity that $\vec{a} \times \vec{b} = -\vec{b} \times \vec{a}$

$$\begin{Bmatrix} L_M \\ M_M \\ N_M \end{Bmatrix} = nA \begin{bmatrix} 0 & \beta_z & -\beta_y \\ -\beta_z & 0 & \beta_x \\ \beta_y & -\beta_x & 0 \end{bmatrix} \begin{Bmatrix} i_x \\ i_y \\ i_z \end{Bmatrix} \tag{96}$$

where $\beta_x, \beta_y, \beta_z$ are the components of the magnetic field in the body frame of the satellite. The moments L, M, N are thus the control torques that rotate the satellite as seen in equation (63).

8.4 Aerodynamics

Aerodynamics are typically written using a taylor series expansion about a trim point[4][16]. That is, the aerodynamic forces are given by

$$\vec{F} = \vec{F}_0 + \frac{\partial \vec{F}}{\partial \vec{x}}(\vec{x} - \vec{x}_0) \tag{97}$$

where $\vec{x} = [x, y, z, \phi, \theta, \psi, u, v, w, p, q, r]^T$. The partial derivative is thus expanded such that

$$\frac{\partial \vec{F}}{\partial \vec{x}} = \begin{bmatrix} \frac{\partial \vec{F}}{\partial x} & \frac{\partial \vec{F}}{\partial y} & \cdots & \frac{\partial \vec{F}}{\partial r} \end{bmatrix} \tag{98}$$

To find all of the partial derivative the forces are first written using a combination of dynamic pressure and coefficients that are functions of geometry and Reynolds number rather than speed, pressure and size. A general lift force can be written using the equation below

$$L = \frac{1}{2}\rho V_\infty{}^2 S C_L \tag{99}$$

where ρ is the atmospheric density, V_∞ is the free-stream velocity, S is the planform area of the lifting surface and C_L is the lift coefficient.

$$V_\infty = \sqrt{u_a{}^2 + v_a{}^2 + w_a{}^2} \tag{100}$$

The subscript 'a' above denotes the velocity of the vehicle plus the atmospheric disturbance.

$$\begin{Bmatrix} u_a \\ v_a \\ w_a \end{Bmatrix} = \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} + \mathbf{T}^T{}_{IB} \begin{Bmatrix} V_x \\ V_y \\ V_z \end{Bmatrix} \tag{101}$$

Note that the dynamic pressure is different for vehicles other than aircraft. Note that the general form for drag is also very similar and can be written as

$$D = \frac{1}{2}\rho V_\infty{}^2 S C_D \tag{102}$$

Specific sections are created below for other flying vehicles. A similar expression can be created for a generic moment such that

$$M = \frac{1}{2}\rho V_\infty{}^2 S \bar{c} C_M \tag{103}$$

where $\bar{c}$ is the mean chord of the lifting surface. The dynamic pressure $q_\infty = \frac{1}{2}\rho V_\infty{}^2 S$ can be used to non-dimensionalize the forces, thus $L/q_\infty = C_L$. This means that the equation involving partial derivatives can be written as

$$\frac{\partial \vec{C}_F}{\partial \vec{x}} = \begin{bmatrix} \frac{\partial \vec{C}_F}{\partial x} & \frac{\partial \vec{C}_F}{\partial y} & \cdots & \frac{\partial \vec{C}_F}{\partial r} \end{bmatrix} \tag{104}$$

If the vector is then expanded to include the components of the vector $\vec{F}$ the partial derivatives expand to

$$\frac{\partial \vec{C}_F}{\partial \vec{x}} = \begin{bmatrix} \frac{\partial C_X}{\partial x} & \frac{\partial C_X}{\partial y} & \cdots & \frac{\partial C_x}{\partial r} \\ \frac{\partial C_Y}{\partial x} & \frac{\partial C_Y}{\partial y} & \cdots & \frac{\partial C_Y}{\partial r} \\ \frac{\partial C_Z}{\partial x} & \frac{\partial C_Z}{\partial y} & \cdots & \frac{\partial C_Z}{\partial r} \end{bmatrix} \tag{105}$$

shorthand can be adopted for the forces above such that $\frac{\partial C_Y}{\partial x} = C_{Yx}$. Using this shorthand the equation above can be written as.

$$\frac{\partial \vec{C}_F}{\partial \vec{x}} = \begin{bmatrix} C_{Xx} & C_{Xy} & \cdots & C_{Xr} \\ C_{Yx} & C_{Yy} & \cdots & C_{Yr} \\ C_{Zx} & C_{Zy} & \cdots & C_{Zr} \end{bmatrix} \tag{106}$$

The coefficients listed above are standard coefficients that all aircraft have. A similar matrix can be formulated for the moments on an aircraft. When system identifying an aircraft all of these coefficients may be determined. However, many of these terms are zero. For example, all coefficients with respect to x y and z are zero. That is, $C_{Xx} = C_{Yx} = ...C_{Nx} = C_{Xy} = ...C_{Nz} = 0$. Other coefficients can be set to zero as well but are not explicitly included in this text.

8.4.1 Aircraft Aerodynamics

For aircraft, some further simplifications are made. Some of the coefficients defined above are combined to be written as functions of the angle of attack(α) and sideslip(β).

$$\alpha = tan^{-1}\left(\frac{w_a}{u_a}\right) \tag{107}$$

$$\beta = sin^{-1}\left(\frac{v_a}{V_\infty}\right) \tag{108}$$

Transforming the equations into these formulations gives rise to coefficients such as $C_{L\alpha}$ which is the change in lift as a function of angle of attack and $C_{Y\beta}$ which is the change in Y-Force as a function of sideslip. Using all of the coefficients defined above taking into account the change to lift and drag, the body aerodynamic force is calculated using the equation below.

$$\begin{Bmatrix} X_A \\ Y_A \\ Z_A \end{Bmatrix} = \frac{1}{2}\rho V_\infty{}^2 S \begin{Bmatrix} C_L s_\alpha - C_D c_\alpha + C_{x_{\delta_t}}\delta_t \\ C_{y\beta}\beta + C_{y\delta_r}\delta_r + C_{yp}\frac{pb}{2V_\infty} + C_{yr}\frac{rb}{2V_\infty} \\ -C_L c_\alpha - C_D s_\alpha \end{Bmatrix} \tag{109}$$

Where the lift and drag coefficients are:

$$\begin{Bmatrix} C_L \\ C_D \end{Bmatrix} = \begin{Bmatrix} C_{L0} + C_{L\alpha}\alpha + C_{Lq}\frac{q\bar{c}}{2V_\infty} + C_{L\delta_e}\delta_e \\ C_{D0} + C_{D\alpha}\alpha^2 \end{Bmatrix} \tag{110}$$

The body aerodynamic moment is also computed using an aerodynamic expansion.

$$\begin{Bmatrix} L_A \\ M_A \\ N_A \end{Bmatrix} = \frac{1}{2}\rho V_\infty{}^2 S\bar{c} \begin{Bmatrix} C_{l\beta}\beta + C_{lp}\frac{pb}{2V_\infty} + C_{lr}\frac{rb}{2V_\infty} + C_{l\delta_a}\delta_a + C_{l\delta_r}\delta_r \\ C_{m0} + C_{m\alpha}\alpha + C_{mq}\frac{q\bar{c}}{2V_\infty} + C_{m\delta_e}\delta_e \\ C_{np}\frac{pb}{2V_\infty} + C_{n\beta}\beta + C_{nr}\frac{rb}{2V_\infty} + C_{n\delta_a}\delta_a + C_{n\delta_r}\delta_r \end{Bmatrix} \tag{111}$$

The aerodynamic coefficients in equations (109), (110) and (111) can be obtained from flight data, aerodynamic modeling and windtunnel tests. Notice that the only coefficients remaining are coefficients from angle of attack, sideslip and angular velocities. Furthermore, the coefficients for angular velocities are also non-dimensionalized by terms such as $b/(2V_\infty)$ where b is the wingspan of the aircraft and $\bar{c}$ is the mean chord of the aircraft. These terms are introduced to fully non-dimensionalize the coefficients. Notice, as well that four extra terms were also introduced. These will be discussed in more detail in the control section however the four terms are the aileron control surface δ_a, the elevator control surface δ_e, the rudder control surface δ_r and the thrust control value δ_t.

8.4.2 Projectile Aerodynamics

To fully define the projectile aerodynamics some more assumptions are made about the projectile.

1. The projectile is axially symmetric

2. The aerodynamic forces are not necessarily formulated at the center of mass

3. The projectile has the potential to be spinning rapidly thus interacting with the surrounding atmosphere

For a projectile the dynamic pressure is written as

$$Q = \frac{\pi}{8}\rho V_\infty^2 d^2 \tag{112}$$

The aerodynamic forces on the projectile are modeled using taylor series ballistic expansions with known coefficients similar to the aircraft model only slightly different assumptions are made given the dynamics of the projectile. The subscripts in the equation below stand for steady and unsteady aerodynamics.

$$\begin{Bmatrix} X_A \\ Y_A \\ Z_A \end{Bmatrix} = \begin{Bmatrix} X_{SA} \\ Y_{SA} \\ Z_{SA} \end{Bmatrix} + \begin{Bmatrix} X_{UA} \\ Y_{UA} \\ Z_{UA} \end{Bmatrix} = Q \begin{Bmatrix} -C_{X_0} - C_{X_2}\frac{v^2+w^2}{V^2} \\ -C_{Y_\beta}\frac{v}{V} \\ -C_{N_\alpha}\frac{w}{V} \end{Bmatrix} + Q \begin{Bmatrix} 0 \\ C_{Y_{p\alpha}}\frac{w}{V}\frac{pd}{2V} \\ C_{Z_{p\alpha}}\frac{v}{V}\frac{pd}{2V} \end{Bmatrix} \tag{113}$$

In this equation, Q is the dynamic pressure, d is the aerodynamic reference area, C_{X_0} is the zero-yaw axial force coefficient, C_{X_2} is the yaw-squared axial force coefficient, C_{N_α} is the normal force derivative coefficient, $C_{Y_{p\alpha}}$ is the Magnus force coefficient, and $V = \sqrt{u^2 + v^2 + w^2}$ is the total velocity of the projectile. The aerodynamic moments acting on the projectile are the pitching, pitch damping, Magnus, and roll damping moments. Pitching and Magnus moments are given by taking the cross product of the normal and Magnus forces given in (113) with the position vector from the center of mass to the center of pressure and location of Magnus force, respectively. The total aerodynamic moments are given in Eqn. (114).

$$\begin{Bmatrix} L_A \\ M_A \\ N_A \end{Bmatrix} = \mathbf{S}_B(\vec{r}_{CG,COP}) \begin{Bmatrix} X_{SA} \\ Y_{SA} \\ Z_{SA} \end{Bmatrix} + \mathbf{S}_B(\vec{r}_{CG,MCOP}) \begin{Bmatrix} X_{UA} \\ Y_{UA} \\ Z_{UA} \end{Bmatrix} + Qd \begin{Bmatrix} C_{l_p}\frac{pd}{2V} \\ C_{m_q}\frac{qd}{2V} \\ C_{n_r}\frac{rd}{2V} \end{Bmatrix} \tag{114}$$

Here, $\mathbf{S}_B(\vec{r}_{CG,COP})$ is the skew-symmetric operator acting on the position vector from the center of mass to the center of pressure expressed in the projectile body frame. Furthermore, $\mathbf{S}_B(\vec{r}_{CG,MCOP})$ is the skew-symmetric operator acting on the position vector from the center of mass to the magnus center of pressure expressed in the projectile body frame. Typically the center of mass is defined from the rear of the projectile such that

$$\mathbf{C}_B(\vec{r}_{CG}) = \begin{Bmatrix} SL_{CG} \\ BL_{CG} \\ WL_{CG} \end{Bmatrix} \tag{115}$$

Similarly, the center of pressure is defined from the rear of the projectile such that

$$\mathbf{C}_B(\vec{r}_{COP}) = \begin{Bmatrix} SL_{COP} \\ BL_{COP} \\ WL_{COP} \end{Bmatrix} \tag{116}$$

The vector $\vec{r}_{CG,COP}$ is then simply the different between both vectors.

$$\vec{r}_{CG,COP} = \vec{r}_{COP} - \vec{r}_{CG} \tag{117}$$

The damping coefficient defined in equation (114) include C_{l_p} which is the roll damping coefficient while C_{m_q} is the pitch damping coefficient. These coefficients are added which essentially inhibit angular motion of the projectile. In addition, to these coefficients, sometimes magnus coefficients are given as pure moments rather than forces acting at a distance. This can be given in the equation below.

$$M_{UA} = Qd(-C_{M\alpha}\frac{v}{V} + C_{N_{p\alpha}}\frac{w}{V}\frac{pd}{2V}) \tag{118}$$

Where $C_{M\alpha}$ replaces the moment produced by $C_{N\alpha}$ and $C_{N_{p\alpha}}$ replaces the moment produced by $C_{Y_{p\alpha}}$. It is possible to derive an equation between the two different representations as given by the equations below.

$$\begin{aligned} C_{M\alpha} &= \frac{(SL_{COP}-SL_{CG})C_{N\alpha}}{d} \\ C_{N_{p\alpha}} &= \frac{(SL_{MAG}-SL_{CG})C_{Y_{p\alpha}}}{d} \end{aligned} \tag{119}$$

8.4.3 Quadcopter Aerodynamics

The aerodynamic model is based on a standard X-frame quadcopter as shown in the Figure below.
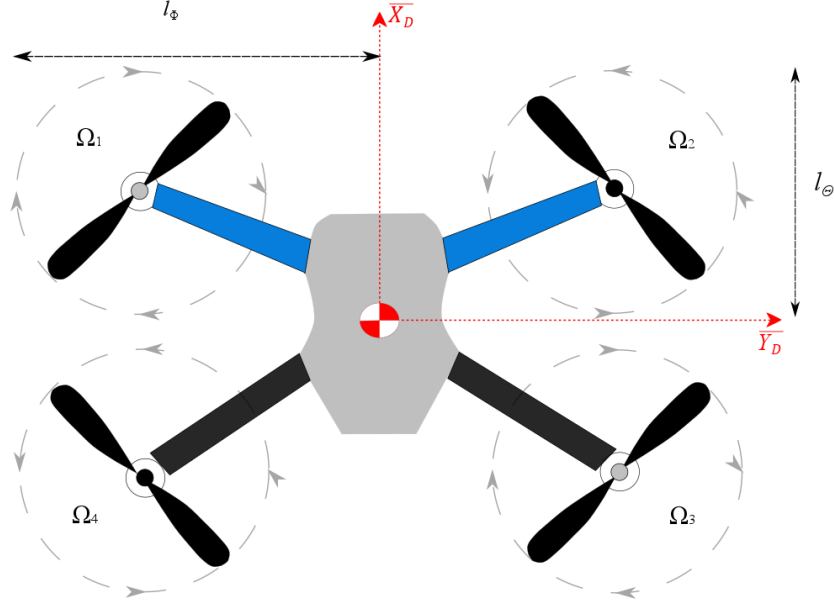
Figure 6: 3DR Iris+ quadrotor model.

The forces exerted on the quadrotor are the total thrust and aerodynamic drag.

$$\begin{Bmatrix} X_A \\ Y_A \\ Z_A \end{Bmatrix} = -T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{Bmatrix} X_{A_D} \\ Y_{A_D} \\ Z_{A_D} \end{Bmatrix} \tag{120}$$

In Eq. (120), T is the total thrust from the rotors and $\{X_{A_D}, Y_{A_D}, Z_{A_D}\}^T$ is the total drag on the body. The total thrust exerted on the quadcopter is made possible due to the counter-rotating blades and allows for easy maneuverability by deviating the rotor angular velocities. The thrust can be simply defined as the sum of the exerted rotor forces f_i such that

$$T = \sum_{i=1}^{4} f_i = f_1 + f_2 + f_3 + f_4 \tag{121}$$

The thrust model of each rotor can then be equated to angular velocity of each rotor (Ω_i). Where k_t is a positive constant.

$$f_i = k_t \Omega_i^2 \tag{122}$$

The total drag caused by the thrusting of the motors is a combination of the induced drag and rotor flapping [27]. The induced drag acting on the platform is caused by an imbalance in the thrust produced by advancing and retreating blades. This behavior can be modeled as a linear function of multirotor translational velocity in the quadcopter frame. Additionally, drag due to rotor flapping is caused by rotor flexibility, which is function of advance ratio and dependent on the rotor blades and hub design[28]. Thus, the total drag vector on the quadcopter is expressed as

$$\begin{Bmatrix} X_{A_D} \\ Y_{A_D} \\ Z_{A_D} \end{Bmatrix} = -T \begin{Bmatrix} \left(\frac{A_{1c}}{\eta} + d_x \right) u - \frac{A_{1s}}{\eta} v \\ \frac{A_{1s}}{\eta} u + \left(\frac{A_{1c}}{\eta} + d_y \right) v \\ 0 \end{Bmatrix} \tag{123}$$

where d_x, d_y are the induced drag coefficients and A_{1s}, A_{1c} are positive constants that describe the rotor flapping

response as a function of advance ratio in Eq. (124)

$$\mu = \frac{\sqrt{u^2 + v^2}}{\eta}$$

$$\eta = \sum_{i=1}^{4} \Omega_i R \tag{124}$$

The quadcopter moment vector will consider additional aerodynamics created by the rotor torque.

$$\{L_A, M_A, N_A\}^T = \vec{\Gamma} + \vec{\tau_A} \tag{125}$$

For Eq. (125), $\vec{\Gamma}$ is the gyroscopic moment and $\vec{\tau_A}$ is the aerodynamic torque vector from the rotors. Note that $\vec{\tau_A} = \{\tau_{A_\phi}, \tau_{A_\theta}, \tau_{A_\psi}\}^T$ is the expanded form of the torque vector. The gyroscopic moments due to the rotors are defined as

$$\vec{\Gamma} = I_r \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ \Omega_r \end{bmatrix} = \begin{bmatrix} I_r \Omega_r q \\ -I_r \Omega_r p \\ 0 \end{bmatrix} \tag{126}$$

$$\Omega_r = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4 \tag{127}$$

where I_r is the moment of inertia about the rotor axis and Ω_r is the relative rotor speed. In order to rotate the quadcopter to change its Euler angles, the angular velocitiis of each rotor can be altered. Due to the X-Frame design of the quadcopter, this text's convention is used to enumerate each rotor blade where rotor 1 is the forward-left rotor and the numbers continue in a clockwise fashion. The result is that the Euler angles are changed in a positive direction by noting the inequalities below.

$$\phi^+ : \Omega_1 \Omega_4 {>} \Omega_2 \Omega_3$$
$$\theta^+ : \Omega_1 \Omega_2 {>} \Omega_3 \Omega_4 \tag{128}$$
$$\psi^+ : \Omega_1 \Omega_3 {>} \Omega_2 \Omega_4$$

Based on Eq. (128), the torques are modeled accordingly as

$$\tau_{A_\phi} = l_\phi[(f_1 + f_4) - (f_2 + f_3)]$$
$$\tau_{A_\theta} = l_\theta[(f_1 + f_2) - (f_3 + f_4)] \tag{129}$$
$$\tau_{A_\psi} = \tau_1 - \tau_2 + \tau_3 - \tau_4$$

In Eq. (129), l_ϕ and l_ψ are the lengths from the rotor axis to the positive platform frame axes for roll and pitch, respectively. Positive yaw torque is achieved by reducing the angular velocity in rotors 2 and 4. The torque (τ_i) of each rotor is also modeled as a fucntion of the angular velocity.

$$\tau_i = b{\Omega_i}^2 \tag{130}$$

The constant b in the above equation denote the blade drag factor. The constants are modeled using blade element momentum theory[29].

$$k_t = C_T \frac{4\rho R^4}{\pi^2} \tag{131}$$

$$b = C_\tau \frac{4\rho R^5}{\pi^3} \tag{132}$$

In Eqs. 131 and 132, ρ is the air density, C_T is the thrust coefficient, and C_Q is thetorque coefficient.

8.4.4 Spacecraft Aerodynamics

The aerodynamic force is computed using aerodynamic coefficients and dynamic pressure where $\vec{V}$ is the velocity of the satellite and V is the magnitude of the velocity vector. Furthermore, S is the surface area of the satellite and C_D is the drag coefficient. The torque on the satellite is then given by the cross product between the aerodynamic force and a distance vector representing the distance between the center of mass and the center of pressure $\vec{r}_{CP-CG}$.

$$\vec{F}_A = \frac{1}{2}\rho V \vec{V} S C_D \tag{133}$$

$$\vec{M}_A = -\mathbf{S}\left(\vec{F}_A\right)\vec{r}_{CP-CG} \tag{134}$$

9 Planetary Positions

Assuming that the Sun is the central inertial reference point, it is possible to obtain the position of Earth and all the other planets at any point in time using well documented orbital elements of our solar system. This formulation follows the derivation by JPL and can be found at [30].

9.1 Julian Day

In order to obtain the position of the planets, the Julian Day must be obtained. The Julian Day of January 1st, 2019 is 2,458,485. The Julian Day of January 1st, 2000 (which is the day of the last inertial frame update) is 2,451,545. In order to obtain the Julian Day of the current day, you simply need to count the number of calendar days from January 1st of 2000. Again I have listed the Julian day of January 1st, 2019 to help with this calculation. To compute the orbital elements of the Earth you must then compute the number of centuries from January 1st, 2000 which is given by the equation below where J is the Julian day and C is the number of centuries since 1/1/2000.

$$C = (J - 2,451,545)/36,525.0 \tag{135}$$

9.2 Orbital Elements

This number is then used in the equations below to obtain the current semi-major axis, eccentricity, inclination, mean longitude, longitude of perihelion and the longitude of the ascending node respectively. The subscript 0 denotes the orbital element in the year 2000.

$$\begin{aligned}
a &= (a_0 + \dot{a}C)AU \\
e &= e_0 + \dot{e}C \\
i &= i_0 + \dot{i}C \\
L &= L_0 + \dot{L}C \\
\bar{w} &= \bar{w}_0 + \dot{\bar{w}}C \\
\Omega &= \Omega_0 + \dot{\Omega}C
\end{aligned} \tag{136}$$

The parameters in the equation above for every planet can be found at [30]. Also, The term AU is an astronomical unit which is equal to 149,597,870,700 meters. For reference though the parameters for Earth are shown below. Just in case you are reading this in the not so distant future, these parameters are only valid until the year 2050 (I may be dead who knows!). Also, the parameters below are for the Earth-Moon barycenter which is the center of mass of the Earth and Moon.

Table 1: Orbital Elements of Earth-Moon Barycenter

a	e	i	L	long.peri. ($\bar{w}$)	long.node. (Ω)
AU, AU/Cy	rad, rad/Cy	deg, deg/Cy	deg, deg/Cy	deg, deg/Cy	deg, deg/Cy
1.00000261	0.01671123	-0.00001531	100.46457166	102.93768193	0.0
0.00000562	-0.00004392	-0.01294668	35999.37244981	0.32327364	0.0

In the table, the first row is the value in the year 2000 and the second row is the rate per century (Cy). Using these parameters, compute the argument of the perihelion $w = \bar{w} - \Omega$ and the mean anomaly $M = L - \bar{w}$. Note that for planets Jupiter, Saturn, Uranus and Neptune, the mean anomaly has a different form. Basically anything past the asteroid belt. With the mean anomaly compute you must modulus this value such that M is between plus or minus 180 degrees. Once that's done you must solve for the eccentric anomaly (E) using the Kepler equation below where e^* is the eccentricity in degrees $e^* = 180e/\pi$.

$$M = E - e^* sin(E) \tag{137}$$

Solving this numerically is pretty simple and only requires a few iterations of the loop below using the C++ programming language. This loop can easily be adapted to any language on modern computers. C++ is shown here in the event this is used for embedded processors in future satellite systems.

```
E = M + e*180.0/PI*sin(M*PI/180.0);
dM = 1;
dE = 0;
while (abs(dM) > 1e-6) {
   dM = M - (E - e*180.0/PI*sin(E*PI/180.0));
   dE = dM/(1.0-e*cos(E*PI/180.0));
   E += dE;
}
```

9.3   Sun Centered Inertial Coordinates

At this point the spatial coordinates can be obtained in the planet's orbital plane where the semi-latus rectum or sometimes simply called the parameter is $p = a(1 - e)$.

$$\begin{Bmatrix} x' \\ y' \\ z' \end{Bmatrix} = \begin{Bmatrix} a(cos(\pi E/180) - e) \\ a\sqrt{1 - e^2}sin(\pi E/180) \\ 0 \end{Bmatrix} \tag{138}$$

Notice that the value z' is zero. This is because orbits are all two dimensional. In order to obtain the coordinates of the planet in the J2000 ecliptic plane, the equation below is used which is similar to the standard Euler angle transformation matrix only the 3-1-3 rotation sequence is used rather than 3-2-1.

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{J2000} = \begin{bmatrix} c_w c_\Omega - s_w s_\Omega c_i & -s_w c_\Omega - c_w s_\Omega c_i & 0 \\ c_w s_\Omega + s_w c_\Omega c_i & -s_w s_\Omega + c_w c_\Omega c_i & 0 \\ s_w s_i & c_w s_i & 0 \end{bmatrix} \begin{Bmatrix} x' \\ y' \\ z' \end{Bmatrix} \tag{139}$$

Running through this formulation for all the planets in the Solar System including Pluto it is possible to plot the position of all planets. The figures below are for January 1st, 2019.
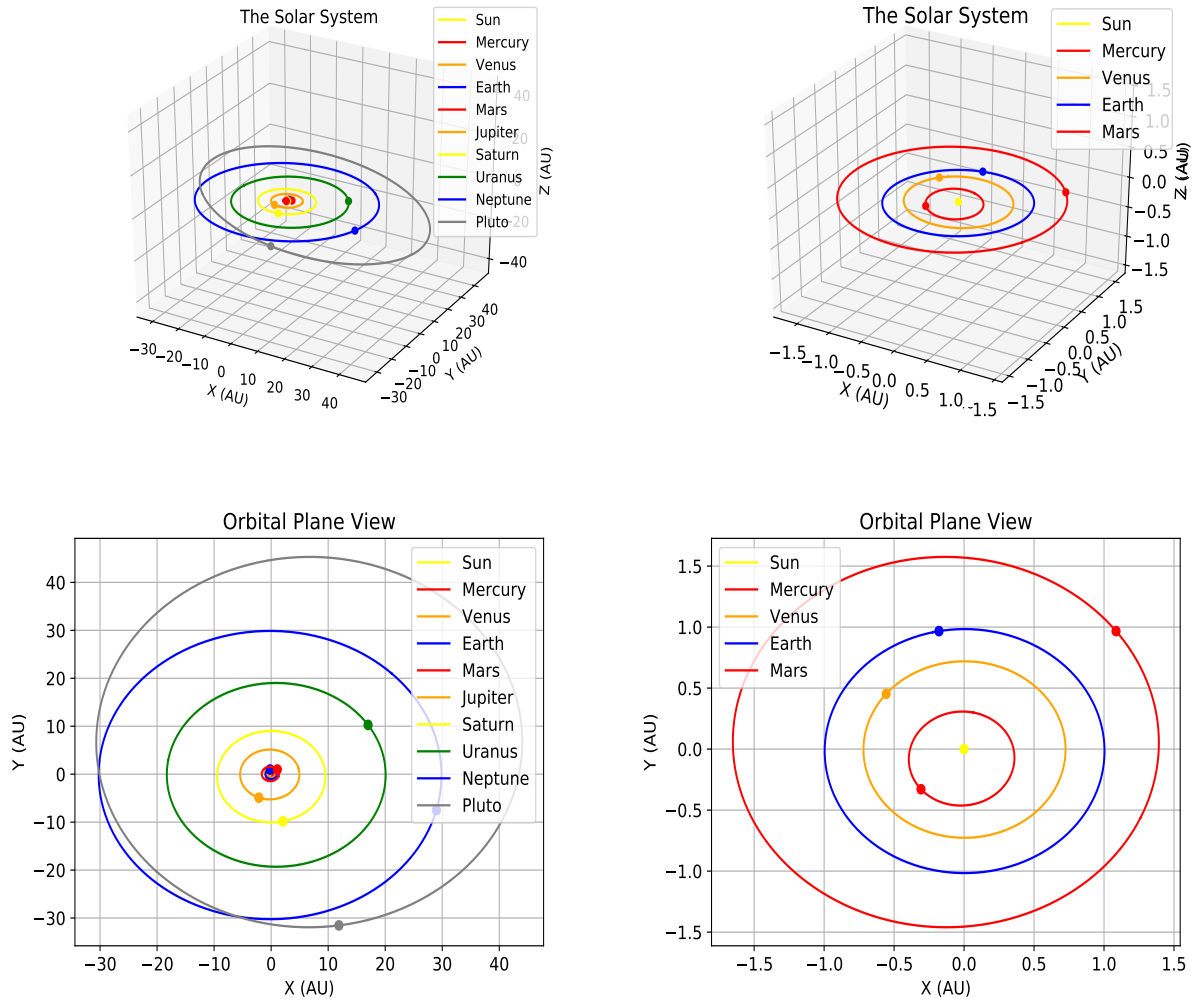
Figure 7: Position of Planets using Orbital Elements

10   Numerical Integration Techniques

10.1   Euler's Method

The equations of motion presented in this text as well as any differential equation can be integrated using Euler's method which is a crude first order method to approximate the time series solution [31]. Note that this method is prone to a significant amount of instability unless the timestep is very small.

$$
\begin{aligned}
\vec{x}_{k+1} &= \vec{x}_k + \dot{\vec{x}}(t_k, \vec{x}_k)\Delta t \\
\dot{\vec{x}}(t_k, \vec{x}_k) &= \vec{f}(\vec{x}_k) + \vec{g}(\vec{x}_k)\vec{u}_k
\end{aligned}
\tag{140}
$$

10.2   Runge-Kutta-4

The RK4 algorithm is the standard in numerical integration and is given in the equation below [31]. The derivative of the quaternions is the same in RK4 as it is in Euler's method. This method is superior to RK4 in that it will converge faster as a function of timestep.

$$
\begin{aligned}
\vec{k}_1 &= \dot{\vec{x}}(t_k, \vec{x}_k) \\
\vec{k}_2 &= \dot{\vec{x}}(t_k + \Delta t/2, \vec{x}_k + \vec{k}_1\Delta t/2) \\
\vec{k}_3 &= \dot{\vec{x}}(t_k + \Delta t/2, \vec{x}_k + \vec{k}_2\Delta t/2) \\
\vec{k}_4 &= \dot{\vec{x}}(t_k + \Delta t, \vec{x}_k + \vec{k}_3\Delta t) \\
\vec{k} &= \tfrac{1}{6}(\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4) \\
\vec{x}_{k+1} &= \vec{x}_k + \vec{k}\Delta t
\end{aligned}
\tag{141}
$$

10.3   Discrete Dynamics

It is often useful for modern computers to write the equations of motion in discrete form....

11 Linear Time Invariant Systems

There is a branch of controls called Linear Time Invariant (LTI) Systems that is often taught at the undergraduate level. Although almost every system encountered in standard applications whether aerospace or not are non-linear, it is still beneficial and more simple to learn about control system dynamics when the system parameters are constant and linear.

11.1 Linearization of Non-Linear Systems

Standard nonlinear dynamics can be placed into standard nonlinear affine form as shown below after much simplification of terms

$$\dot{\vec{x}} = \vec{f}(\vec{x}) + \vec{g}(\vec{x})\vec{u} \tag{142}$$

where $\vec{u}$ is the control input which could be the forces and moments from reaction wheels or thrusters for a spacecraft or lift and drag for an airplane. The vectors f and g represent the systems dynamics which is dependent on the system itself. Note that if the dynamics cannot be put into affine form, the system is highly nonlinear and the control system designed for that would require more sophisticated analysis like Lyapunov design or sliding mode control. That will be discussed in a later section. For the affine form however, the equation can be linearized to give the equation below.

$$\Delta\dot{\vec{x}} = \mathbf{A}\Delta\vec{x} + \mathbf{B}\Delta\vec{u} \tag{143}$$

where $\Delta\vec{x} = \vec{x} - \vec{x}_e$ and $\vec{x}_e$ is an equilibrium point. In this formulation $\mathbf{A} = \partial\vec{f}/\partial\vec{x}$. and $\mathbf{B} = \partial\vec{g}/\partial\vec{x}$ which are partial derivatives of the state matrices. Note that an equilibrium point is a point where the system is at rest and not changing. Mathematically this means that $\dot{x}_e = 0$. When linearizing a system it is important to choose an equilibrium point that is relevant to the system. For example, for an aircraft the equilibrium point could be straight and level flight at a constant velocity. For a spacecraft it could be a circular orbit at a constant altitude. The linearized equations of motion are only valid for small perturbations around the equilibrium point. If the system deviates too far from the equilibrium point, the linearized equations may no longer accurately represent the system's dynamics.

11.2 General Formulation of Differential Equations

The linear systems of equations derived above are in state space form. That is the vector $\vec{x}$ contains all the states of the system. However, it is often easier to understand the dynamics of a system when it is in the form of a single differential equation with all states and derivatives shown in the same equation. Practical examples also help the undergraduate controls engineer as math with practical context is the best way to learn in my humble opinion. The sections below will discuss first and second order systems which are the most common types of systems encountered in engineering applications. Note that higher order systems can be reduced to first and second order systems by using techniques such as root locus or pole placement since higher order systems typically have high frequency dynamics that can be ignored for control purposes. For the most part, first and second order systems are all that is needed to understand the basics of control systems and represent the majority of systems encountered in practice.

11.2.1 First Order Systems

A first order system undergoing free motion will have dynamics that look like this

$$\dot{q} + \sigma q = \sigma f \tag{144}$$

where q is a generalized coordinate, σ is the inverse of the time constant τ, and f is the forcing function. Examples of these types of systems in include thermistors, servos, and velocity equations for system dynamics.

11.2.2 Second Order Systems

A second order system undergoing free motion will have dynamics that look like this

$$\ddot{q} + 2\zeta\omega_n\dot{q} + \omega_n{}^2 q = \omega_n{}^2 f \tag{145}$$

where q is a generalized coordinate, σ is the inverse of the time constant τ, f is the forcing function, ω_n is the natural frequency of the system and ζ is the damping ratio. Examples of these types of systems include mass, spring dampers in linear translation as well as torsional systems and penduluums. Anything that oscillates will exhibit this behavior.

11.3 Equation of Motion Formulation for Practical Systems

The sections below will discuss practical examples of first and second order systems. Many of these examples are classic examples found in many dynamics and controls textbooks and ones that I typically teach budding control systems aerospace/mechanical engineers. Each one offers a slightly different set of dynamics and challenges that will help the undergraduate controls engineer understand the concepts of linear systems as well as how to control them and also determine stability.

11.3.1 Position and Velocity of a Car

Consider a car moving in one dimension. The primary forces acting on the car are the driving force from the tires, F, and a linear viscous drag force, $D = -cv$, where c is the viscous drag coefficient, x is the position of the car and v is the velocity of the car.
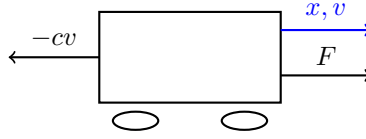
Figure 8: Free body diagram of a car

Applying Newton's second law, the sum of forces is equal to mass times acceleration:

$$m\dot{v} = F - cv \tag{146}$$

where m is the mass of the car, F is the force generated by the tires, and $-cv$ is the linear drag. In this case to make the system first order we simply leave acceleration as $\dot{v}$ instead of $\ddot{x}$ where x would be the position of the vehicle. Rearranging, we obtain the standard first-order linear differential equation:

$$\dot{v} + \frac{c}{m}v = \frac{F}{m} \tag{147}$$

This equation describes the velocity dynamics of the car under a linear drag assumption. In this case $\sigma = c/m$ and $F = cf$. With these substitutions you arrive at the same equation as 144. Note that if v is replaced with $\dot{x}$ and $\dot{v}$ is replaced with $\ddot{x}$ the equation describes the position of car and is second order as shown below.

$$\ddot{x} + \frac{c}{m}\dot{x} = \frac{F}{m} \tag{148}$$

This is another general second order system where the natural frequency $\omega_n{}^2 = 0$. Using the same substitions as above the general formulation is

$$\ddot{q} + \sigma\dot{q} = \sigma f \tag{149}$$

where x is replaced by q. The result is similar to the first order system but still slightly different. given the absence of the constant term. It will be shown later that this system is marginally stable and that the difference between the first and second order transfer function is simply a factor of s given the difference from velocity to position.

11.3.2 Position of a Mass Spring Damper

A mass-spring-damper system is a classic example of a second-order system. The system consists of a mass m attached to a spring with spring constant k and a damper with damping coefficient c. The mass is free to move along a single axis x, and the system is subject to an external force f as shown in Figure 9.
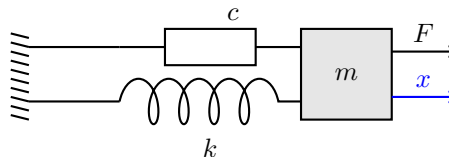
Figure 9: Mass-spring-damper system

Consider the mass-spring-damper system depicted above. The spring exerts a restoring force $-kx$ that is always directed opposite to the displacement x of the mass from equilibrium, while the damper exerts a resistive force $-c\dot{x}$ proportional and opposite to the velocity $\dot{x}$. The external force $F(t)$ acts in the direction of motion. According to Newton's second law, the sum of these forces acting on the mass equals the mass times its acceleration, leading to the equation

$$m\ddot{x} = F - c\dot{x} - kx \tag{150}$$

Rearranging terms, we obtain the standard form of the equation of motion for the mass-spring-damper system.

$$\ddot{x} + \frac{c}{m}\dot{x} + \frac{k}{m}x = \frac{F}{m} \tag{151}$$

This second-order linear differential equation fully describes the dynamic response of the system to any external force F. Looking at equation 145 we can see that $\omega_n = \sqrt{k/m}$ and $\zeta = c/(2\sqrt{mk})$. Furthermore, if we let $F = kf$ we arrive at the same equation as 145.

11.3.3 Attitude Dynamics of a Satellite or Quadcopter

For a satellite undergoing free 1-D rotational motion the dynamics are quite interesting from a controls perspective given the somewhat marginal stability of the satellite. The same can be said for a quadcopter undergoing slow rotational motion. In this case, it is assumed that the spacecraft or quadcopter rotates through the angle θ about its center of mass. The spacecraft/quadcopter has a moment of inertia J and is subject to an external torque τ as shown in Figure 10. In this case ,the torque is created by reaction control thrusters or propellors with a force F at a distance d from the center of mass. Note that in the case of the satellite there are 4 thrusters but only 2 operate at a time. If the top right and bottom left thrusters fire the torque is negative and if the top left and bottom right thrusters fire the torque is positive. In this case we let $F_1 = F_2 = F_3 = F_4 = F$. If $F > 0$ it means it is a positive torque and $F_1 = F_4$ are non-zero while $F_2 = F_3 = 0$. If $F < 0$ the torque is negative and $F_1 = F_4 = 0$ while $F_2 = F_3$ are non-zero. For a quadcopter the 4 propellors are offset by a nominal thrust to keep the quadcopter aloft. Furthermore, since the quadcopter is symmetric about the rotational axis it can be assumed that the quadcopter only has 2 propellors F_3 and F_4. The torque is created by increasing the thrust on one side and decreasing it on the other side. In this case $F_3 = F_H + \Delta F$, $F_4 = F_H - \Delta F$ where F_H is the thrust required for hovering. Adding the effect of F_3 and F_4 yields $2\Delta F$ which can just be replaced with $2F$ and the formulation can continue.
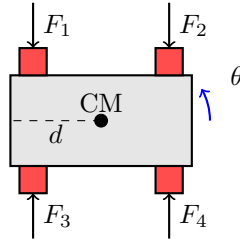
Figure 10: Satellite with reaction control thrusters

The equation of motion for the satellite can be derived from Newton's second law for rotational motion, which states that the sum of torques acting on a body equals the moment of inertia times its angular acceleration as derived in Section 4.2. The full equation is repeated here for clarity.

$$\vec{M}_C = \dot{\mathbf{I}}\vec{\omega}_{B/I} + \mathbf{I}_C\frac{^Bd(\vec{\omega}_{B/I})}{dt} + \mathbf{S}(\vec{\omega}_{B/I})\mathbf{I}_C\vec{\omega}_{B/I} \tag{152}$$

In this section however the inertia is constant, thus the entire first term goes to zero, and since the motion of the satellite is planar the third term is also zero. Thus, the equation reduces to

$$\vec{M}_C = \mathbf{I}_C\frac{^Bd(\vec{\omega}_{B/I})}{dt} = J\vec{\alpha}_{B/I} = J\ddot{\theta} \tag{153}$$

where $\vec{\alpha}_{B/I}$ is the angular acceleration of the satellite. The torque generated by the thrusters is given as $\vec{M}_C = \tau = 2Fd$. The multiplication by 2 is due to 2 thrusters firing at a time or in the case of the quadcopter it is due to the

symmetry of the rotation axis as explained above. The final equations of motion for the satellite/quadcopter are then given as

$$\ddot{\theta} = \frac{2Fd}{J} \tag{154}$$

This is a classic second order system where $\omega_n = 0$ and $\zeta = 0$. This means that the system is marginally stable and will not return to equilibrium after a disturbance. Stability will be discussed in a later section. Letting $F = f\frac{J}{2d}$ results in $\ddot{\theta} = f$ which is almost the same equation as equation 145 but where the natural frequency and damping ratio are both zero.

11.3.4 Pitch Dynamics of an Aircraft

For an aircraft we assume that the aircraft is symmetric about the longitudinal axis and that the aircraft is only pitching about its center of mass. The pitch dynamics of an aircraft are similar to a spring mass damper since there is a constant restoring moment given by $C_{m\alpha}$ due to the aerodynamic forces acting on the aircraft as well as a damping term C_{mq} also created by aerodynamic forces. Using the aerodynamic equations given in Equation 111, the pitch dynamics of an aircraft are given as

$$I_{yy}\ddot{\theta} = \vec{M}_C = \frac{1}{2}\rho V^2 S\bar{c}(C_{m\alpha}\theta + C_{mq}\frac{\bar{c}}{2V}\dot{\theta} + C_{m\delta_e}\delta_e) \tag{155}$$

Note that in this case we assume that $\theta = \alpha$ the angle of attack which is not necessarily true but for quick motion about the pitch axis this can be assumed to be true. Again the moment of inertia is I_{yy} and the pitch angle is θ. The aerodynamic force is given by the dynamic pressure $q = (1/2)\rho V^2$ where ρ is the density of air, V is the velocity of the aircraft (assumed to be constant for pure pitching), S is the reference area and $\bar{c}$ is the mean aerodynamic chord. The moment coefficients are $C_{m\alpha}$ which is the restoring moment coefficient, C_{mq} which is the damping moment coefficient and $C_{m\delta_e}$ which is the control moment coefficient due to elevator deflection δ_e. Rearranging the equation into standard form yields

$$\ddot{\theta} + \left(-\frac{\rho V^2 S\bar{c}^2 C_{mq}}{4I_{yy}V}\right)\dot{\theta} + \left(-\frac{\rho V^2 S\bar{c}C_{m\alpha}}{2I_{yy}}\right)\theta = -\frac{\rho V^2 S\bar{c}C_{m\delta_e}}{2I_{yy}}\delta_e \tag{156}$$

which is in the standard form of second order systems with a natural frequency and damping ratio given as

$$\omega_n{}^2 = -\frac{\rho V^2 S\bar{c}C_{m\alpha}}{2I_{yy}}, \quad \zeta = -\frac{\rho V^2 S\bar{c}^2 C_{mq}}{8I_{yy}V\omega_n} \tag{157}$$

Note that $C_{m\alpha}$ and C_{mq} are typically negative which makes both ω_n and ζ positive. However, in the case of the F-16 $C_{m\alpha}$ is actually negative. In this case the natural frequency squared becomes negative and the solution quite different. In The control term is the elevator deflection δ_e which means performing the final substitution below

$$\delta_e = f\frac{C_{m\alpha}}{C_{m\delta_e}} \tag{158}$$

results in the same equation as equation 145 provided you subsitute θ with the generalized q coordinate.

11.3.5 Pitch Dynamics of a Rocket

The pitch dynamic of a rocket are interesting in that the system is also marginally stable much like the satellite/quadcopter systems explained above. The difference of course is that the system is in the presence of the atmosphere and thus the aerodynamic forces create a damping effect which is proportional to the square of the velocity. This damping term is also nonlinear. However, in order to make the system linear it is assumed that the velocity is constant. For the sake of clarity the full equations of motion will be shown and then simplified to attain a simple linear form. First, the free body diagram of the rocket is shown in the Figure below.
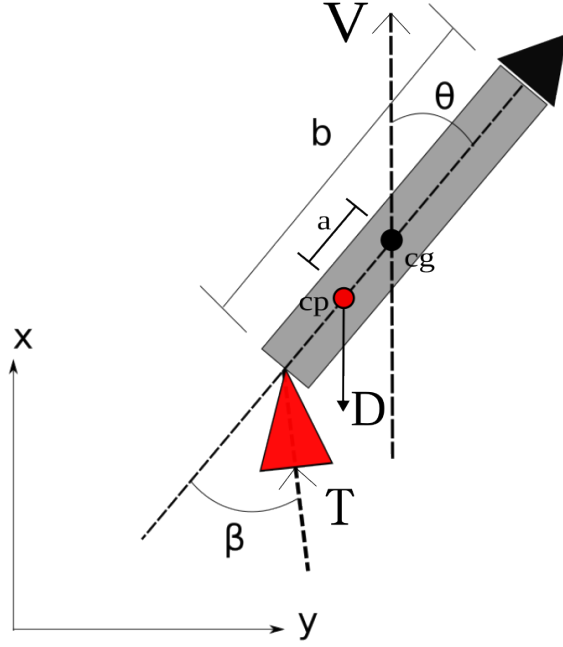
Figure 11: Rocket Pitch Dynamics Free Body Diagram

Here thrust acts at an angle β with respect to the centerline of the rocket with a thrust equal to T. The length of the rocket is b and thus the moment arm of the thrust vector is $b/2$. The aerodynamic force D acts at the center of pressure which is a distance c from the center of mass. It is assumed that the rocket is constantly traveling upwards and thus the Drag force D is always pointed down at a distance of a from the center of mass. The moment of inertia of the rocket is J and the pitch angle is θ. The equations of motion are similar to the quadcopter/satellite system undergoing 1-D rotational motion with a few subtle differences.

$$J\ddot{\theta} = \vec{M_C} = -Da\sin\theta + \frac{Tb}{2}\sin(\beta) \tag{159}$$

First you'll see that the aerodynamic force which is restoring is nonlinear due to the $\sin\theta$ term. Furthermore, the thrust force T acts through the angle β leading to a problem where the thrust force T could change in addition to the angle β. Two control terms would represent a multi-input system which requires the use of optimization to determine the best course of action. Since the focus of this chapter is on Single-Input-Single-Ouput (SISO) systems, the force T will be represented as a constant. Thus, the control term for this system will actually be the angle β. However, note that the control term is also nonlinear. Both $sin()$ terms can be linearized for small angles such that $\sin\theta \approx \theta$ and $\sin\beta \approx \beta$. Note that $\theta = 0$ is the equilibrium position in this case. The equation of motion then becomes

$$J\ddot{\theta} + Da\theta = T\frac{b}{2}\beta \tag{160}$$

Which is linear...almost. The second issue in this equation is that the aerodynamic force is a function of velocity squared. The drag force is given in Equations 112 and 113. As stated above it is assumed that the velocity is constant and thus D is constant as well. The final equation of motion for the rocket pitch dynamics is then given as

$$\ddot{\theta} + \frac{\pi}{J8}\rho V^2 d^2 a\theta = \frac{Tb}{2J}\beta \tag{161}$$

If $(\pi/(J8))\rho V^2 d^2 a = \omega_n{}^2$, $Tb/(2J)\beta = \kappa\omega_n{}^2 f$ and $\theta = q$, the equations of motion are identical to the general form of second order systems except that κ is a gain applied to the forcing function.

$$\ddot{q} + \omega_n{}^2 q = \kappa\omega_n{}^2 f \tag{162}$$

In addition, it is clear that the damping ratio ζ is zero.

11.3.6 Pitch Dynamics of a Pendulum

The dynamics of a pendulum are interesting for two reasons. First, the system is non-linear in the angle θ and the system has an infinite number of equilibrium points. Half of the equilibrium points are also unstable. For the sake of this problem though we will restrict the system to be between $-\pi$ and π which means the system will only have 3 equilibrium points. The first is $\theta = 0$ and the other two are $\theta = -\pi$ and $\theta = \pi$. Obviously the second two equilibrium points are the same. To formulate the dynamics we start with the figure below
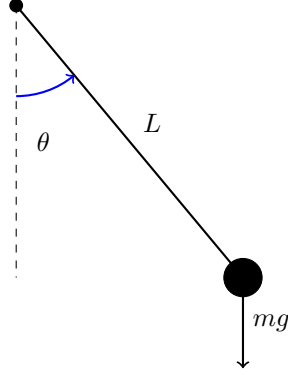
Figure 12: Pendulum Free Body Diagram

The pendulum consists of a mass m attached to a rigid, massless rod of length L. The pendulum swings about a pivot point under the influence of gravity, which exerts a downward force mg on the mass. An external torque τ acts at the pivot point itself. The angle θ represents the angular displacement of the pendulum from the vertical equilibrium position. The equation of motion for the pendulum can be derived just as before with the sum of torques acting on a body equaling the moment of inertia times its angular acceleration. The moment of inertia J for a point mass at a distance L from the pivot is given by $J = mL^2$. The torque generated by the gravitational force is $\tau_g = -mgL \sin \theta$, where the negative sign indicates that gravity acts to restore the pendulum to its equilibrium position. Thus, the equation of motion is given as

$$J\ddot{\theta} = \vec{M}_C = -mgL \sin \theta + \tau \tag{163}$$

Rearranging and substituting the moment of inertia yields

$$\ddot{\theta} = \frac{\tau}{mL} - \frac{g}{L} \sin \theta \tag{164}$$

This equation is close to the second order equations where the damping ratio is zero but it is nonlinear due to the $\sin \theta$ and $\cos \theta$ terms. In this case the system can be linearized about the equilibrium point $\theta = 0$ but can also be linearized about the equilibrium point $\theta = \pi$ if the pendulum is inverted. In order to linearize the equations of motion you need to use equation 143 from the linearization section. Note that in this case the matrices A and B are expanded out instead of using matrix notation.

$$\Delta\ddot{\theta} = \frac{\partial f}{\partial \dot{\theta}} \Delta\dot{\theta} + \frac{\partial f}{\partial \theta} \Delta\theta + \frac{\partial g}{\partial \tau} \Delta\tau \tag{165}$$

where $f = -(g/L) \sin \theta$ and $g = \tau/(mL)$. Calculating the partial derivative and plugging into the equation above yields

$$\Delta\ddot{\theta} = 0 - \frac{g}{L} cos(\theta_e) \Delta\theta + \frac{1}{mL} \Delta\tau \tag{166}$$

Rearranging and dropping the Δ terms yields

$$\ddot{\theta} + \frac{g}{L} cos(\theta_e) \theta = \frac{1}{mL} \tau \tag{167}$$

This will now yield two separate and quite different equations depending on the equilibrium point chosen. First, if the equilibrium point is $\theta_e = 0$ the equation becomes

$$\ddot{\theta} + \frac{g}{L} \theta = \frac{1}{mL} \tau \tag{168}$$

which is in the standard form of second order systems where $\omega_n{}^2 = g/L$ and $\zeta = 0$. The control term is $\tau = fmg$ which results in the same equation as 145. If $\theta_e = \pi$ the equation becomes

$$\ddot{\theta} - \frac{g}{L}\theta = \frac{1}{mL}\tau \tag{169}$$

which is again in the standard form of second order systems except that the constant term is negative which means the system is actually unstable. More on this system will be discussed in the stability section. Furthermore, the solution is also quite different and will be explored in the sections that follow.

11.4 Characteristic and Particular Solutions to Differential Equations

The differential equations presented in the previous section can be solved using a variety of methods. The most common methods are the characteristic and particular solution method and the Laplace transform method. The sections below will discuss both methods starting with the characteristic and particular solution method.

11.4.1 General First Order System

First the first order system defined above will be solved first. For a first order linear differential equation of the form presented in equation 144, the general solution is the sum of the homogeneous (characteristic) solution and a particular solution. The particular solution corresponds to the steady-state response when the forcing function f is constant. In this case assume f is a step function which is just a constant f_0. The particular solution $q_p(t) = C$ where C is a constant and thus $\dot{q}_p = \dot{C} = 0$. Plugging this into equation 144 gives

$$0 + \sigma q_p = \sigma f_0 \tag{170}$$

Thus, the particular solution for a constant input is simply $q_p = f_0$. If the forcing function is not constant then the particular solution will be a function of time and solving for the particular solution will be more difficult. Examples of this include sinusoidal forcing functions or exponentially decaying forcing functions. The solutions to these types of forcing functions can be found in many differential equations textbooks and will not be discussed here for the time being. The homogeneous solution is found by setting the forcing function to zero and solving the resulting equation assuming that $q_h(t) = Ae^{st}$. Thus, the homogeneous equation is

$$\dot{q}_h + \sigma q_h = Ase^{st} + \sigma Ae^{st} = Ae^{st}(s + \sigma) = 0 \tag{171}$$

In the equation above the result yields 3 equations that could be equal to zero. The first is $A = 0$ which is a trivial solution and thus not considered. The second is $e^{st} = 0$ which is never true for any real or complex value of s. The third is $s + \sigma = 0$ which is the charactertic equation for the first order system. The solution to this equation yields the characteristic root $s = -\sigma$. Thus, the homogeneous solution is

$$q_h(t) = Ae^{-\sigma t} \tag{172}$$

where A is a constant determined by the initial conditions. The general solution is the sum of the homogeneous and particular solutions

$$q(t) = q_h(t) + q_p(t) = Ae^{-\sigma t} + f_0 \tag{173}$$

The constant A can be determined by the initial condition $q(0) = q_0$ which yields

$$q(0) = A + f_0 = q_0 \implies A = q_0 - f_0 \tag{174}$$

Thus, the final solution to the first order system is

$$q(t) = q_0 - f_0 + Ae^{-\sigma t} \tag{175}$$

or more simply

$$q(t) = f_0 + (q_0 - f_0)e^{-\sigma t} \tag{176}$$

In the case of the car example the solution would be

$$v(t) = \frac{F_0}{c} + \left(v_0 - \frac{F_0}{c}\right)e^{-\frac{c}{m}t} \tag{177}$$

noting that $\sigma = c/m, F_0 = cf_0$ and v_0 is the initial velocity of the car. Typically for these types of systems the initial velocity is zero so the equation simplifies to

$$v(t) = \frac{F_0}{c}\left(1 - e^{-\frac{c}{m}t}\right) \tag{178}$$

which shows that the car will asymptotically approach a steady state velocity of F_0/c. This will be plotted later in the time response section for open loop analysis.

11.4.2 General Second Order Systems

The solution to second order systems is quite complex given the nature of solving a second order differential equation. The general solution is again the sum of the homogeneous and particular solutions and assume the initial conditions are zero. The major difficulty with second order systems is that the solution to the homogenous characteristic eqaution can be real, repeated or imaginary roots. The roots effect the overall solution and will be discussed later. To start the particular solution is very simple and is again found by assuming a constant forcing function f_0 such that $q_p(t) = C$ where C is a constant and thus $\dot{q}_p = \dot{C} = 0$ and $\ddot{q}_p = \ddot{C} = 0$. Plugging this into equation 145 gives the eqaution below:

$$0 + 0 + \omega_n{}^2 q_p = \omega_n{}^2 f_0 \tag{179}$$

Thus, the particular solution for a constant input is simply $q_p = f_0$. If the forcing function is not constant then the particular solution will be a function of time and solving for the particular solution will be more difficult. Examples of this include sinusoidal forcing functions or exponentially decaying forcing functions. The solutions to these types of forcing functions can be found in many differential equations textbooks and will not be discussed here for the time being. The homogeneous solution is found by setting the forcing function to zero and solving the resulting equation assuming that $q_h(t) = Ae^{st}$. Thus the homogeneous equation is

$$\begin{aligned} \ddot{q}_h + 2\zeta\omega_n\dot{q}_h + \omega_n{}^2 q_h &= 0 \\ As^2 e^{st} + 2\zeta\omega_n Ase^{st} + \omega_n{}^2 Ae^{st} &= 0 \\ Ae^{st}(s^2 + 2\zeta\omega_n s + \omega_n{}^2) &= 0 \end{aligned} \tag{180}$$

In the equation above the result yields 3 equations that could be equal to zero. The first is $A = 0$ which is a trivial solution and thus not considered. The second is $e^{st} = 0$ which is never true for any real or complex value of s. The third is $s^2 + 2\zeta\omega_n s + \omega_n{}^2 = 0$ which is the charactertic equation for the second order system. The solution to this equation yields the characteristic roots which as stated above can have real, repeated or imaginary roots. The characteristic roots are found using the quadratic formula and are given as

$$s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \tag{181}$$

Note that if $\omega_n{}^2$ is negative, as is the case with the unstable $C_{m\alpha}$ of the F-16 or the inverted pendulum, the characteristic equation becomes $s^2 + 2\zeta\omega_n s - \omega_n^2 = 0$ which results in roots of

$$s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 + 1} \tag{182}$$

which is almost the same but notice the plus sign in front of the ζ^2 term. In this case the roots are always real and distinct. It is possible to have complex roots if the damping ratio ζ is negative but this is not a common case and will not be discussed here. The nature of the roots depends on the value of the damping ratio ζ as well as the sign of ω_n^2. The three cases for $\omega_n{}^2 > 0$ are underdamped ($0 \leqslant \zeta < 1$), critically damped ($\zeta = 1$), and overdamped ($\zeta > 1$). Each case will be discussed below. Note that for the case where $\omega_n{}^2 < 0$ the roots are always real and distinct which has a similar solution to the overdamped case.

- **Underdamped** ($0 \leqslant \zeta < 1$): In this case, the roots are complex conjugates ($s_{12} = -\zeta w_n \pm \omega_d j$) where $\omega_d = \omega_n\sqrt{1 - \zeta^2}$, leading to oscillatory behavior. Note that this case also applies to the edge case where the damping ratio ζ is zero. The homogeneous solution is given by:

$$q_h(t) = e^{-\zeta\omega_n t}\left(A_1\cos(\omega_d t) + A_2\sin(\omega_d t)\right) \tag{183}$$

  where $\omega_d = \omega_n\sqrt{1 - \zeta^2}$ is the damped natural frequency, and A_1 and A_2 are constants determined by initial conditions.

43

- **Critically Damped ($\zeta = 1$):** In this case, the roots are real and repeated ($s_{12} = -\omega_n$). The homogeneous solution is given by:

$$q_h(t) = (A_1 + A_2 t)e^{-\omega_n t} \tag{184}$$

- **Overdamped ($\zeta > 1$ or ${\omega_n}^2 < 0$):** In this case, the roots are real and distinct ($s_{12} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$) or ($s_{12} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 + 1}$). The homogeneous solution is given by:

$$q_h(t) = A_1 e^{s_1 t} + A_2 e^{s_2 t} \tag{185}$$

The general solution is the sum of the homogeneous and particular solutions

$$q(t) = q_h(t) + q_p(t) = q_h(t) + f_0 \tag{186}$$

The constants A_1 and A_2 can be determined by the initial conditions $q(0) = q_0$ and $\dot{q}(0) = \dot{q}_0$. The final solution will depend on the damping ratio ζ and the initial conditions. For the case where the initial conditions are zero the final solutions for each case are shown below.

- **Underdamped ($0 \leqslant \zeta < 1$)** Note the solution is still valid if $\zeta = 0$:

$$q(t) = f_0 \left(1 - e^{-\zeta\omega_n t} \left(\cos(\omega_d t) + \frac{\zeta}{\sqrt{1-\zeta^2}} \sin(\omega_d t) \right) \right) \tag{187}$$

- **Critically Damped ($\zeta = 1$):**

$$q(t) = f_0 \left(1 - (1 + \omega_n t)e^{-\omega_n t} \right) \tag{188}$$

- **Overdamped ($\zeta > 1$ or ${\omega_n}^2 < 0$):**

$$q(t) = f_0 \left(1 - \frac{s_2 e^{s_1 t} - s_1 e^{s_2 t}}{s_2 - s_1} \right) \tag{189}$$

In the case of the mass-spring-damper example, recall that $f_0 = F_0/k$, $\omega_n = \sqrt{k/m}$, and $\zeta = c/(2\sqrt{mk})$.

11.4.3   Special Case of Natural Frequency Equal to Zero

As was shown in the special case of the car where no restoring force existed, the natural frequency was equal to zero. In this case the general equation of motion was $\ddot{q} + \sigma\dot{q} = \sigma f$. The solution to this equation is different given the characteristic equation is $s^2 + \sigma s = 0$ which has two roots of $s = 0$ and $s = -\sigma$. The homogeneous solution in this case is given as

$$q_h(t) = A_1 + A_2 e^{-\sigma t} \tag{190}$$

For the particular solution again assume a constant forcing function f_0 such that $q_p(t) = Ct$ where C is a constant and thus $\dot{q}_p = C$ and $\ddot{q}_p = 0$. Plugging this into the equation of motion gives

$$0 + \sigma C = \sigma f_0 \tag{191}$$

and then $C = f_0$. Thus, the particular solution is $q_p(t) = f_0 t$ and the general solution is

$$q(t) = A_1 + A_2 e^{-\sigma t} + f_0 t \tag{192}$$

The constants A_1 and A_2 can be determined by the initial conditions $q(0) = q_0$ and $\dot{q}(0) = \dot{q}_0$. For the case where the initial conditions are zero the final solution is given as

$$q(t) = f_0 t - \frac{f_0}{\sigma}(1 - e^{-\sigma t}) \tag{193}$$

In the case of the car example, recall that $f_0 = F_0/c$ and $\sigma = c/m$. Plugging those values into the equation yields

$$x(t) = \frac{F_0}{c}t - \frac{F_0 m}{c^2}(1 - e^{-c/mt}) \tag{194}$$

This result is intuitive because it states that if a constant force to the car is applied, the car will initially accelerate but then reach a constant velocity due to the drag force. This is different than the case of the quadcopter/satellite example where the angle will continue to accelerate. Note that $\dot{x} = v$. Applying a derivative to the equation above yields the same result as equation the equation for velocity

$$\dot{x}(t) = v(t) = \frac{F_0}{c} - 0 - \frac{c}{m}\frac{F_0 m}{c^2}e^{-c/mt} = \frac{F_0}{c}\left(1 - e^{-\frac{c}{m}t}\right) \tag{195}$$

This is a good check for the budding controls engineer to ensure that the math is correct.

44

11.4.4   Special Case of Natural Frequency and Damping Ratio Equal to Zero

As was shown in the special case of the satellite/quadcopter where no restoring moment or damping moment existed, the natural frequency and damping ratio were both equal to zero. In this case the general equation of motion was $\ddot{q} = f$. The solution to this equation is different given the characteristic equation is simply $s^2 = 0$ which has a repeated root of $s = 0$. The homogeneous solution is then given as

$$q_h(t) = A_1 + A_2 t \tag{196}$$

since $e^{0t} = 1$. The particular solution is again $q_p(t) = (A_0/2)t^2$ where A_0 is a constant. Plugging this into the equation of motion gives

$$\ddot{q}_p = A_0 = f_0 \tag{197}$$

Thus, the particular solution is $q_p(t) = f_0 t^2/2$ and the general solution is

$$q(t) = A_1 + A_2 t + \frac{f_0}{2} t^2 \tag{198}$$

The constants A_1 and A_2 can be determined by the initial conditions $q(0) = q_0$ and $\dot{q}(0) = \dot{q}_0$. For the case where the initial conditions are zero the final solution is given as

$$q(t) = \frac{f_0}{2} t^2 \tag{199}$$

In the case of the satellite/quadcopter example, recall that $f_0 = \frac{2F_0 d}{J}$. This result is intuitive because it states that if a constant force to the thrusters/propellors is applied, the angle of the satellite/quadcopter will increase quadratically over time. That is, the system will continue to accelerate until it runs out of fuel or the thrusters are turned off. In the case of the quadcopter, if the propellors return to symmetric thrust the quadcopter will stop accelerating and maintain a constant angular velocity. In practice of course the quadcopter will have some damping due to air resistance and the satellite will have some damping due to magnetic torques and other effects. However, these effects are typically small and can be ignored for control purposes.

11.5   Laplace Solutions

Another method to solve linear differential equations is the Laplace transform method. The Laplace transform converts a time-domain differential equation into an algebraic equation in the s-domain, which can be easier to manipulate and solve. The Laplace domain and algebraic equations are very useful for control system dynamics and every undergraduate controls engineer should learn its fundamentals. The Laplace transform of a function $f(t)$ is defined as

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^\infty f(t) e^{-st} dt \tag{200}$$

where s is a complex variable similar to the s in the Ae^{st} equations discussed earlier. The inverse Laplace transform is given by

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi j} \lim_{T \to \infty} \int_{\gamma - jT}^{\gamma + jT} F(s) e^{st} ds \tag{201}$$

where γ is a real number such that all singularities of $F(s)$ are to the left of the line $Re(s) = \gamma$. Note that these integrals are very complex and not typically solved by hand but rather looked up in a table of Laplace transforms. Some common Laplace transforms are shown below

Time Domain $f(t)$	Laplace Transform $\mathcal{L}\{f(t)\}$	Description
$u(t)$	$\frac{1}{s}$	Unit step
t	$\frac{1}{s^2}$	Ramp
e^{at}	$\frac{1}{s-a}$	Exponential
$sin(at)$	$\frac{a}{s^2+a^2}$	Sine
$cos(at)$	$\frac{s}{s^2+a^2}$	Cosine
t^2	$\frac{2}{s^3}$	Quadratic
te^{at}	$\frac{1}{(s-a)^2}$	Repeated Root
$e^{at}sin(bt)$	$\frac{b}{(s-a)^2+b^2}$	Underdamped Sine
$e^{at}cos(bt)$	$\frac{s-a}{(s-a)^2+b^2}$	Underdamped Cosine
$\int f(t)dt$	$\frac{F(s)}{s}$	Integral
$\dot{f}(t)$	$sF(s) - f(0)$	First derivative
$\ddot{f}(t)$	$s^2F(s) - sf(0) - \dot{f}(0)$	Second derivative

Table 2: Common Laplace transforms

where $u(t)$ is the unit step function where $u(t) = 0$ for $t < 0$ and $u(t) = 1$ for $t \geqslant 0$. The undergraduate student may look at these equations and wonder how exactly these functions are helpful. To illustrate their usefulness the first and second order systems defined above will be solved using the Laplace transform method.

11.5.1 General First Order Systems

Taking the Laplace transform of equation 144 and assuming zero initial conditions gives

$$sQ(s) + \sigma Q(s) = \sigma F(s) \tag{202}$$

where $Q(s)$ is the Laplace transform of $q(t)$ and $F(s)$ is the Laplace transform of $f(t)$. Rearranging gives the equation below

$$\frac{Q(s)}{F(s)} = G(s) = \frac{\sigma}{(s + \sigma)} \tag{203}$$

The function $G(s)$ is called the transfer function of the system and is a very important function in control systems. The transfer function relates the output of the system $Q(s)$ to the input of the system $F(s)$ in the Laplace domain. The transfer function can be used to analyze the stability and performance of the system. The transfer function can also be used to design controllers for the system which will be discussed later. Let's now assume that the forcing function is a step function such that $f(t) = f_0 u(t)$ where $u(t)$ is the unit step function. The Laplace transform of a step function is $F(s) = f_0/s$. Plugging this into the equation above gives

$$Q(s) = G(s)F(s) = \frac{\sigma}{s + \sigma}\frac{f_0}{s} = \frac{\sigma f_0}{s(s + \sigma)} \tag{204}$$

The solution to the equation above can be found by taking the inverse Laplace transform. However, the equation above is not in the standard table of Laplace transforms. Thus partial fraction decomposition is needed to get $Q(s)$ into a form that can be used for inverse Laplace. The equation can be rewritten using partial fractions as

$$Q(s) = \frac{\sigma f_0}{s(s + \sigma)} = \frac{A}{s} + \frac{B}{s + \sigma} \tag{205}$$

where A and B are constants to be determined. Multiplying both sides by the denominator on the left side gives

$$\sigma f_0 = A(s + \sigma) + Bs \tag{206}$$

46

Setting $s = 0$ gives $A = \sigma f_0/\sigma = f_0$. Setting $s = -\sigma$ gives $B = -\sigma f_0/\sigma = -f_0$. Thus, the equation can be rewritten as

$$Q(s) = \frac{f_0}{s} - \frac{f_0}{s + \sigma} \tag{207}$$

Taking the inverse Laplace transform gives

$$q(t) = f_0 - f_0 e^{-\sigma t} = f_0(1 - e^{-\sigma t}) \tag{208}$$

substituting back in the values for σ and f_0 gives

$$v(t) = \frac{F_0}{c}(1 - e^{-\frac{c}{m}t}) \tag{209}$$

which is the same solution as derived using the characteristic and particular solution method assuming zero initial conditions as given by equation 178.

11.5.2 General Second Order Systems

Taking the Laplace transform of equation 145 and assuming zero initial conditions gives

$$s^2 Q(s) + 2\zeta\omega_n s Q(s) + \omega_n{}^2 Q(s) = \omega_n{}^2 F(s) \tag{210}$$

Rearranging just as before gives the equation below

$$\frac{Q(s)}{F(s)} = G(s) = \frac{\omega_n{}^2}{s^2 + 2\zeta\omega_n s + \omega_n{}^2} \tag{211}$$

where again $G(s)$ is the transfer function of the system. Let's again assume that the forcing function is a step function such that $f(t) = f_0 u(t)$. Plugging the Laplace transform of $F(s)$ just as before gives

$$Q(s) = G(s)F(s) = \frac{\omega_n{}^2}{s^2 + 2\zeta\omega_n s + \omega_n{}^2}\frac{f_0}{s} = \frac{\omega_n{}^2 f_0}{s(s^2 + 2\zeta\omega_n s + \omega_n{}^2)} \tag{212}$$

The solution to the equation above again can be found by taking the inverse Laplace transform. However, again the equation above is not in the standard table of Laplace transforms. Thus partial fraction decomposition is needed to get $Q(s)$ into a form that can be used for inverse Laplace. The equation can be rewritten using partial fractions as

$$Q(s) = \frac{\omega_n{}^2 f_0}{s(s^2 + 2\zeta\omega_n s + \omega_n{}^2)} = \frac{A}{s} + \frac{Bs + C}{s^2 + 2\zeta\omega_n s + \omega_n{}^2} \tag{213}$$

where A, B and C are constants to be determined. Multiplying both sides by the denominator on the left side gives

$$\omega_n{}^2 f_0 = A(s^2 + 2\zeta\omega_n s + \omega_n{}^2) + (Bs + C)s \tag{214}$$

Setting $s = 0$ gives $A = \omega_n{}^2 f_0/\omega_n{}^2 = f_0$. Setting $s = j\omega_n(-\zeta + \sqrt{\zeta^2 - 1})$ and $s = j\omega_n(-\zeta - \sqrt{\zeta^2 - 1})$ gives two equations that can be solved simultaneously to find B and C. The final results are $B = -f_0$ and $C = -2\zeta\omega_n f_0$. Thus, the equation can be rewritten as

$$Q(s) = \frac{f_0}{s} - \frac{f_0 s + 2\zeta\omega_n f_0}{s^2 + 2\zeta\omega_n s + \omega_n{}^2} \tag{215}$$

The first term of the equation is simply the Laplace transform of a step function. The second term can be solved using the table of Laplace transforms. The final solution depends on the value of the damping ratio ζ just as before. The final solutions for each case assuming zero initial conditions are shown below. It is left to the reader to verify that these solutions match those derived using the characteristic and particular solution method. For the case when $\zeta = 0$ the inverse laplace is simply a sine or cosine while the the general underdamped solution is solved by completing the square in the denominator and using the Laplace transform of the underdamped sine and cosine functions. The critically damped and overdamped solutions are solved using the Laplace transforms of repeated roots and real distinct roots respectively.

- **Underdamped** $(0 \leqslant \zeta < 1)$:

$$q(t) = f_0\left(1 - e^{-\zeta\omega_n t}\left(\cos(\omega_d t) + \frac{\zeta}{\sqrt{1 - \zeta^2}}\sin(\omega_d t)\right)\right) \tag{216}$$

- **Critically Damped ($\zeta = 1$):**

$$q(t) = f_0\left(1 - (1 + \omega_n t)e^{-\omega_n t}\right) \tag{217}$$

- **Overdamped ($\zeta > 1$ or $\omega_n{}^2 < 0$):**

$$q(t) = f_0\left(1 - \frac{s_2 e^{s_1 t} - s_1 e^{s_2 t}}{s_2 - s_1}\right) \tag{218}$$

Notice again that the solutions are the same as those derived using the characteristic and particular solution method assuming zero initial conditions. It is possible that the undergraduate engineer still does not see the usefulness of the Laplace transform method. The true power of the Laplace transform method is in the transfer function. The transfer function can be used to analyze the stability and performance of the system as well as design controllers for the system which will be discussed in the sections below.

11.5.3 Special Case of Natural Frequency Equal to Zero

As was shown in the special case of the car where no restoring force existed, the natural frequency was equal to zero. In this case the general equation of motion was $\ddot{q} + \sigma\dot{q} = \sigma f$. Taking the laplace transform of this equation and assuming zero initial conditions gives

$$s^2 Q(s) + \sigma s Q(s) = \sigma F(s) \tag{219}$$

Rearranging gives the equation below

$$\frac{Q(s)}{F(s)} = G(s) = \frac{\sigma}{s(s + \sigma)} \tag{220}$$

where again $G(s)$ is the transfer function of the system. Notice that this is identical to the first order transfer function multiplied by an additional $1/s$ term. This is because the second order system with zero natural frequency is simply the integral of the first order system. That is $x(t) = \int v(t)dt$. In the Laplace domain this means that $X(s) = \frac{1}{s}V(s)$ or if we let $\dot{x} = v(t)$ the lapace transform would result in $sX(s) = V(s)$. This is where the Laplace transform method really shines. The transfer function can be manipulated easily in the Laplace domain to find relationships between different variables. Differential equations in the time domain can be difficult to manipulate but in the Laplace domain they are simply algebraic equations. Let's again assume that the forcing function is a step function such that $f(t) = f_0 u(t)$. Plugging the Laplace transform of $F(s)$ just as before gives

$$Q(s) = G(s)F(s) = \frac{\sigma}{s(s + \sigma)}\frac{f_0}{s} = \frac{\sigma f_0}{s^2(s + \sigma)} \tag{221}$$

The solution to the equation above can be found by taking the inverse Laplace transform. However, again the equation above is not in the standard table of Laplace transforms. Thus partial fraction decomposition is needed to get $Q(s)$ into a form that can be used for inverse Laplace. The equation can be rewritten using partial fractions as

$$Q(s) = \frac{\sigma f_0}{s^2(s + \sigma)} = \frac{A}{s} + \frac{B}{s^2} + \frac{C}{s + \sigma} \tag{222}$$

where A, B and C are constants to be determined. Multiplying both sides by the denominator on the left side gives

$$\sigma f_0 = As(s + \sigma) + B(s + \sigma) + Cs^2 \tag{223}$$

Setting $s = 0$ gives $\sigma f_0 = \sigma B$ and $B = f_0$. Setting $s = -\sigma$ gives $C\sigma^2 = \sigma f_0$ and thus $C = f_0/\sigma$. Setting $s = 1$ gives

$$
\begin{aligned}
\sigma f_0 &= A(\sigma + 1) + f_0(\sigma + 1) + f_0/\sigma \\
\sigma^2 f_0 &= A\sigma(\sigma + 1) + f_0\sigma(\sigma + 1) + f_0 \\
(\sigma^2 - 1)f_0 = (\sigma + 1)(\sigma - 1)f_0 &= A\sigma(\sigma + 1) + f_0\sigma(\sigma + 1) \\
\sigma f_0 - f_0 &= A\sigma + f_0\sigma \\
A &= -f_0/\sigma
\end{aligned} \tag{224}
$$

Plugging in these values for A, B, and C gives

$$Q(s) = -\frac{f_0}{\sigma s} + \frac{f_0}{s^2} + \frac{f_0}{\sigma}\frac{1}{s + \sigma} \tag{225}$$

Taking the inverse Laplace transform gives

$$q(t) = -\frac{f_0}{\sigma} + f_0 t + \frac{f_0}{\sigma} e^{-\sigma t} = f_0 t - \frac{f_0}{\sigma}(1 - e^{-\sigma t}) \tag{226}$$

which is the same result shown previously. Taking the derivative of the equation above gives the same result as the velocity equation shown previously. The budding controls engineer may think that the method of partial fractions and inverse Laplace transform is tedious and not worth the effort. However, the true power of the Laplace transform method is in the transfer function. The transfer function can be used to analyze the stability and performance of the system as well as design controllers for the system which will be discussed in detail in Chapter 13.

11.5.4   Special Case of Natural Frequency and Damping Ratio Equal to Zero

As was shown in the special case of the satellite/quadcopter where no restoring moment or damping moment existed, the natural frequency and damping ratio were both equal to zero. In this case the general equation of motion was $\ddot{q} = f$. Taking the Laplace transform of this equation and assuming zero initial conditions gives

$$s^2 Q(s) = F(s) \tag{227}$$

Rearranging gives the equation below

$$\frac{Q(s)}{F(s)} = G(s) = \frac{1}{s^2} \tag{228}$$

where again $G(s)$ is the transfer function of the system. Let's again assume that the forcing function is a step function such that $f(t) = f_0 u(t)$. Plugging the Laplace transform of $F(s)$ just as before gives

$$Q(s) = G(s)F(s) = \frac{1}{s^2}\frac{f_0}{s} = \frac{f_0}{s^3} \tag{229}$$

The solution to the equation above can be found by taking the inverse Laplace transform. The equation above is in the standard table of Laplace transforms and the inverse Laplace transform is given as

$$q(t) = \frac{f_0}{2} t^2 \tag{230}$$

which is the same solution as derived using the characteristic and particular solution method assuming zero initial conditions. In the case of the satellite example, recall that $f_0 = \frac{2F_0 d}{J}$.

11.6   Open Loop Time Response

Now that the solutions to the first and second order systems have been derived using both the characteristic and particular solution method as well as the Laplace transform method, the time response of these systems can be analyzed. The time response of a system is the output of the system as a function of time given a specific input. The time response can be used to analyze the stability and performance of the system.

11.6.1   Position and Velocity Response of a Car

First, let's plot the time response of the first order system given by equation 178 as well as the second order solution given by equation 195. The parameters for the system are chosen to be $F_0 = 1000\ N$, $c = 50\ Ns/m$, and $m = 1000\ kg$. The initial velocity and position are assumed to be zero. In order to show the time response however, 3 different curves will be placed on the graph. The first will be the analytic solution to the first and second order system, the second is the Laplace transform solution using the control systems toolbox in Python and the last is the numerical integration of the equations of motion. The python code used to generate this is shown in the Figure below and also available on GitHub.

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  from scipy.integrate import odeint
4  import control as ctl
5  # System parameters
6  F0 = 1000      # N
7  c = 50         # Ns/m
8  m = 1000       # kg
9  # Time vector
10 t = np.linspace(0, 200, 1000)
11 # Analytic solution from
12 v_analytic = (F0/c) * (1 - np.exp(-c/m * t))
13 x_analytic = (F0/c)*t + (F0*m/c**2)*(np.exp(-c/m*t)-1)
14 # Numerical integration
15 def eoms(z, t, F0, c, m):
16     x = z[0]
17     v = z[1]
18     xdot = v
19     xddot = (F0 - c*v)/m
20     return [xdot, xddot]
21 numeric = odeint(eoms, [0, 0], t, args=(F0, c, m))
22 v_numeric = numeric[:, 1]
23 x_numeric = numeric[:, 0]
24 # Laplace (transfer function) solution
25 num = [c/m]
26 den = [1, c/m]
27 v_system = ctl.tf(num, den)
28 x_system = ctl.tf(num, [1, c/m, 0]) #notice the extra zero for position

29 t_laplace, v_laplace = ctl.step_response(v_system, T=t)
30 t_laplace, x_laplace = ctl.step_response(x_system, T=t)
31 v_laplace *= F0/c  # scale for step input magnitude
32 x_laplace *= F0/c  # scale for step input magnitude
33 # Plotting
34 plt.figure(figsize=(8,5))
35 plt.plot(t, v_analytic, label='Analytic Solution', lw=2)
36 plt.plot(t, v_numeric, '--', label='Numerical Integration', lw=2)
37 plt.plot(t_laplace, v_laplace, ':', label='Laplace Solution', lw=2)
38 plt.xlabel('Time [s]')
39 plt.ylabel('Velocity [m/s]')
40 plt.title('First Order System Response')
41 plt.legend()
42 plt.grid(True)
43
44 plt.figure(figsize=(8,5))
45 plt.plot(t, x_analytic, label='Analytic Solution', lw=2)
46 plt.plot(t, x_numeric, label='Numerical Integration', lw=2)
47 plt.plot(t_laplace, x_laplace, ':', label='Laplace Solution', lw=2)
48 plt.xlabel('Time [s]')
49 plt.ylabel('Position [m]')
50 plt.title('Second Order System Position Response (ω_n=0)')
51 plt.grid(True)
52 plt.show()
```

Figure 13: Python code to generate the time response of a car

Note the code is commented to help the reader understand what is happening. Note the benefit of using the control systems toolbox in Python. The time response can be generated by first creating the transfer function and then applying a step to the transfer function. This will come in handy later when it is necessary to apply a control system to the transfer function before inversing back to the time domain. The numerical integration is also useful when the equations of motion are known and the control system designer wants to work in the time doman rather than the laplace domain. It's also very helpful when the system is non-linear since transfer functions only work for linear systems. All three methods have their pros and cons. The numerical integration is done using the scipy.integrate.odeint function which is a simple way to numerically integrate ordinary differential equations. The results of the time response are shown in Figure 14.
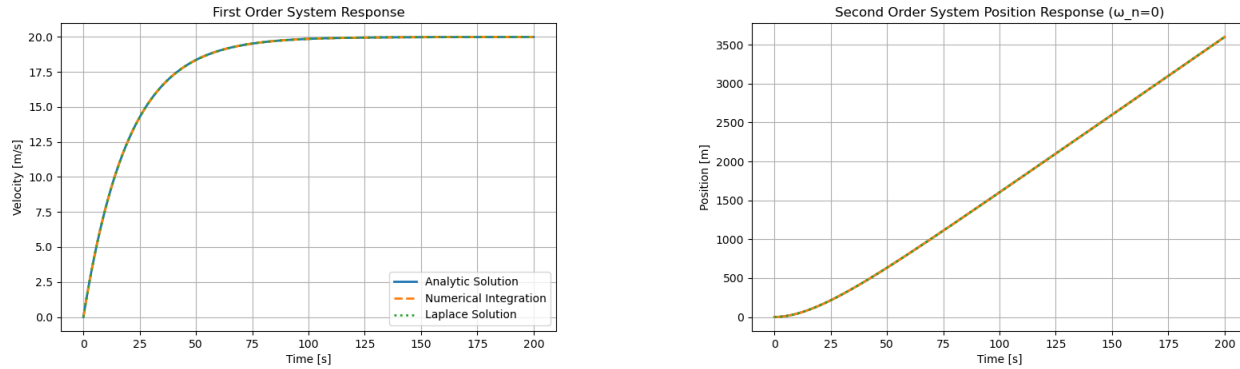
Figure 14: Time response of the position (right) and velocity (left) of a car

The time response shows that the velocity of the car asymptotically approaches a steady state value of $F_0/c = 20 \ m/s$. The time constant of the system is $\tau = m/c = 20 \ s$. The time constant is the time it takes for the system to reach 63.2% of its steady state value. The settling time $T_s = 4\tau$ is the time the system takes to get within 2% of its final value. In this case that is $80 \ s$. The time response can be used to analyze the stability and performance of the system. In this case, the system is stable and the performance is acceptable given the parameters chosen. However, examining the position of the car shows that the position will continue to increase linearly over time. This is because the car is moving at a constant velocity. The position of the car will not reach a steady state value but rather continue to increase linearly over time. This is an example of a marginally stable system. The velocity of the car is stable while the position of the car is marginally stable.

11.6.2 Position of a Mass Spring Damper

Simulating the mass spring damper is very similar to the car example. The parameters for the system are chosen to be $F_0 = 1000 \ N$, $k = 2000 \ N/m$, $c = 50 \ Ns/m$, and $m = 100 \ kg$. The initial position and velocity are assumed to be zero. The python code used to generate this is shown in the Figure below and also available on GitHub.

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
import control as ctl
# System parameters
F0 = 1000        # N
k = 2000         # N/m
c = 50           # Ns/m
m = 100          # kg
# Time vector
t = np.linspace(0, 20, 1000)
# Analytic solution from
A1 = np.cos(np.sqrt(4*m*k - c**2)/(2*m)*t)
A2 = (c/np.sqrt(4*m*k - c**2))*np.sin(np.sqrt(4*m*k - c**2)/(2*m)*t)
x_analytic = (F0/k)*(1 - np.exp(-c/(2*m)*t)*(A1 + A2))
# Numerical integration
def eoms(z, t, F0, c, m,k):
    x = z[0]
    xdot = z[1]
    xddot = (F0 - c*xdot - k*x)/m
    return [xdot, xddot]
numeric = odeint(eoms, [0, 0], t, args=(F0, c, m,k))
x_numeric = numeric[:, 0]

# Laplace (transfer function) solution
num = [k/m]
den = [1, c/m, k/m]
system = ctl.tf(num, den)
t_laplace, x_laplace = ctl.step_response(system, T=t)
x_laplace *= F0/k  # scale for step input magnitude
# Plotting
plt.figure(figsize=(8,5))
plt.plot(t, x_analytic, label='Analytic Solution', lw=2)
plt.plot(t, x_numeric, '--', label='Numerical Integration', lw=2)
plt.plot(t_laplace, x_laplace, ':', label='Laplace Solution', lw=2)
plt.xlabel('Time [s]')
plt.ylabel('Position [m]')
plt.title('Second Order System Position Response')
plt.legend()
plt.grid(True)
plt.show()
```

Figure 15: Python code to generate the time response of a mass spring damper

Note the code is commented to help the reader understand what is happening. Again, the simulation is run by integrating the equations of motion, applying a step response to the transfer function and also plotting the analytic solution. The results of the time response are shown in Figure 16.
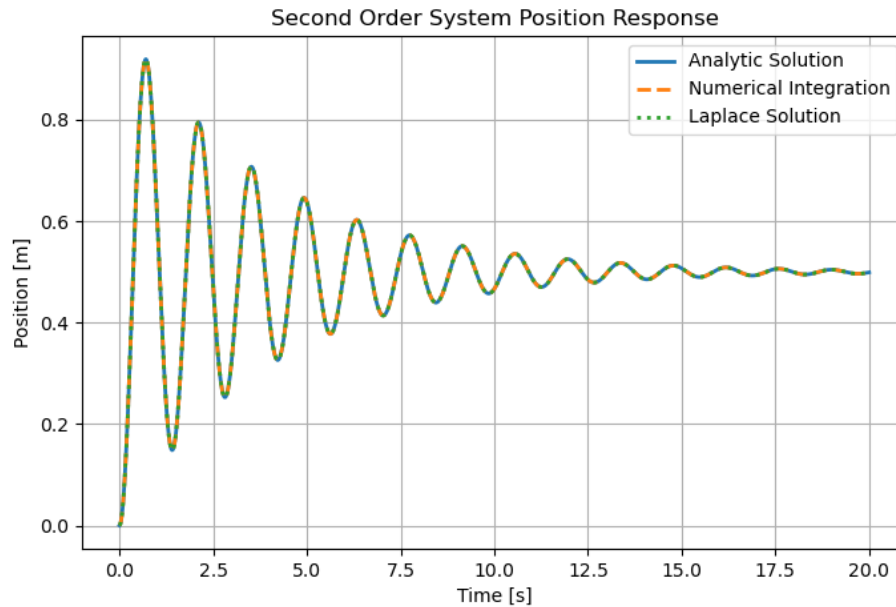
Figure 16: Time response of the position of a mass spring damper

The time response shows that the position of the mass spring damper reaches a steady state value of $F_0/k = 0.5 \ m$. The time constant of the system is $\tau = 2m/c = 4 \ s$. The natural frequency of the system is $\omega_n = \sqrt{k/m} = 4.47 \ rad/s$. The damping ratio of the system is $\zeta = c/(2m\omega_n) = 0.056 (underdamped)$. The settling time $T_s = 4/(\zeta\omega_n)$ is the time the system takes to get within 2% of its final value. In this case that is 16 s. Notice that $T_s = 4\tau$ where again $\tau = 4 \ s$. The time response can be used to analyze the stability and performance of the system. In this case, the system is stable and the performance is acceptable given the parameters chosen. Here are some useful equations

for natural frequency, settling time and other second order parameters.

$$\begin{array}{ccc}
\omega_n = \sqrt{\frac{k}{m}} & T_s = \frac{4}{\zeta\omega_n} & \tau = \frac{2m}{c} \\
\zeta = \frac{c}{2m\omega_n} & \omega_d = \omega_n\sqrt{1-\zeta^2} & T_s = 4\tau \\
\omega_n = 2\pi f & f = \frac{1}{T} & \tau = \frac{1}{\zeta\omega_n}
\end{array} \tag{231}$$

11.6.3   Attitude of a Satellite or Quadcopter

Recall for the satellite/quadcopter example the equation of motion was given as $\ddot{\theta} = \frac{2Fd}{J}$ and the transfer function was given as $\frac{\Theta(s)}{F(s)} = G(s) = \frac{2d}{Js^2}$. The solution to this system due to a step function was shown in the general solution to second order systems with the special case that the natural frequency and damping ratio was zero. The solution then given as $\theta(t) = \frac{F_0 d}{J}t^2$ which increases quadratically over time. The python code used to generate this is also available on Github. The code is not shown here as it is very similar to the previous two examples. Again the code is commented to help the reader understand what is happening. The simulation is run by integrating the equations of motion, applying a step response to the transfer function and also plotting the analytic solution. The results of the time response are shown in Figure 17. For this specific case $F_0 = 0.1 \ N$, $d = 0.5 \ m$ and $J = 10 \ kg - m^2$.
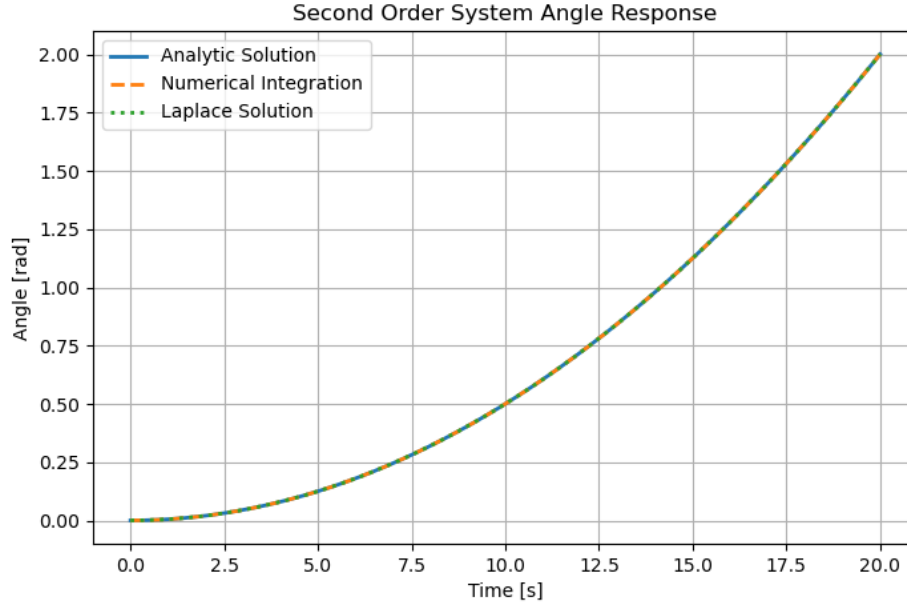
Figure 17: Time response of the attitude of a satellite/quadcopter

The time response shows that the attitude of the satellite/quadcopter increases quadratically over time. This is because there is no restoring moment or damping moment acting on the system. The system is marginally stable as the attitude will continue to increase over time and this will be discussed in more detail in the Stability section.

11.6.4   Pitch Response of an Aircraft

11.6.5   Pitch Response of a Rocket

11.6.6   Angle of an Inverted Pendulum

11.7 Stability

Stability of open or even closed loop systems is an important concept in control systems. An open loop system is stable if the output of the system remains bounded for a bounded input. In other words, if the input to the system is a step function, the output of the system will reach a steady state value and not diverge to infinity. The stability of a system can be analyzed using the characteristic equation of the system which was shown in the solution to multiple differential equations. The astute reader will have also noticed that the denominator of the transfer function is identical to the characteristic equation. The roots of the characteristic equation are often called the poles of the system and can be used interchangeably. Thus, the benefit of a transfer function is that stability can be ascertained immediately by looking at the denominator of the transfer function. In the Chapter on feedback control it will be shown that control systems in the laplace domain can be applied by simply multiplying different control systems and examining the resulting closed loop transfer function for stability. When designing controllers in the time domain it is almost impossible to gaurantee closed loop stability. However, with transfer functions, after a bit of algebraic manipulation stability can be ascertained.

So what do the poles or roots of the characteristic equation tell you? Remember that the general soltion to a differential equation is $q(t) = Ae^{st}$. The value of s, the roots, the poles, tell whether the system will oscillate, decay or extend off to infinity. The location of the poles in the complex plane determines the stability of the system. If all poles have negative real parts, the system is stable. This would results in an equation like e^{-st} where that term tends to 0 as $t \to \infty$. If any pole has a positive real part, the system is unstable. If any pole has a real part equal to zero, the system is marginally stable. Marginal stability is defined as a system that neither grows nor decays but rather oscillates indefinitely or increases but not faster than a linear term. This would result in an equation like $e^{j\omega t}$ or $f_0 t$ which is a sine/cosine function or linear term. The imaginary part of the pole determines the frequency of oscillation. The real part of the pole determines the rate of decay or growth. A pole with a large negative real part will result in a fast decay while a pole with a small negative real part will result in a slow decay. A pole with a large positive real part will result in a fast growth while a pole with a small positive real part will result in a slow growth. The sections that follow examine the transfer functions of the systems examined earlier and determine their stability. The equations below summarize the stability criteria.

- **Stable**: All poles have negative real parts. $Re(s) < 0$

- **Unstable**: Any pole has a positive real part. $Re(s) > 0$

- **Marginally Stable**: Any pole has a real part equal to zero. $Re(s) = 0$

Note that for the sections that follow code has been added to all the python scripts in the Time Response section such as satellite.py or mass_spring_damper.py. The code added plots the poles of each system on a real-imaginary plot to visually show the stability of each system. For clarity on the code, the Figure below shows the relevant function used to plot the poles of each system which again has been added to Github.

```python
def plot_pz(sys, title=None):
    p = ctl.poles(sys)
    z = ctl.zeros(sys)
    plt.figure(figsize=(6,6))
    # plot zeros (o) and poles (x)
    if z.size:
        plt.plot(np.real(z), np.imag(z), 'o', ms=10, label='Zeros')
    if p.size:
        plt.plot(np.real(p), np.imag(p), 'x', ms=10, label='Poles')
    plt.axhline(0, color='k', lw=0.5)
    plt.axvline(0, color='k', lw=0.5)
    plt.xlabel('Real')
    plt.ylabel('Imag')
    if title:
        plt.title(title)
    plt.legend()
    plt.grid(True)
```

Figure 18: Pole-Zero Plotting Code

11.7.1 Position and Velocity of a Car

As derived earlier the transfer function for the position of the car is given as

$$\frac{X(s)}{F(s)} = G(s) = \frac{\sigma}{s(s + \sigma)} \tag{232}$$

recall also that the time domain solution to this system due to a step function was given as

$$x(t) = f_0 t - \frac{f_0}{\sigma}(1 - e^{-\sigma t}) \tag{233}$$

Plotted in the time domain the system accelerated and then attained a constant velocity. The poles of the system are located at $s = 0$ and $s = -\sigma$. The pole at $s = -\sigma$ has a negative real part and thus is stable. This is due to the $1/(s + \sigma)$ term which translates to the $e^{-\sigma t}$ term in the time domain. The pole at $s = 0$ has a real part equal to zero and thus is marginally stable as given by the $1/s$ term in the laplace domain and the $f_0 t$ term in the time domain. The marginal stability is apparent because the position of the car will continue to increase linearly over time due to the constant velocity. However, the velocity of the car will reach a steady state value and not diverge to infinity. Thus, the system is considered stable for velocity. Remember the transfer function for velocity is found by simply multiplying the position transfer function by s or taking the derivative in the time domain. The transfer function for velocity is given as

$$\frac{V(s)}{F(s)} = G(s) = \frac{\sigma}{s + \sigma} \tag{234}$$

while the time domain solution to this system due to a step function was given as

$$v(t) = f_0(1 - e^{-\sigma t}) \tag{235}$$

The pole of the velocity system is located at $s = -\sigma$ again from the $1/(s + \sigma)$ term in the laplace domain and the $e^{-\sigma t}$ term in the time domain. The pole has a negative real part and thus is stable. This is because the velocity of the car will reach a steady state value and not diverge to infinity. Examining both systems in the laplace domain is much easier than examining the time domain solutions because you can simply look at the poles of the transfer function. It is also beneficial to plot the poles of the system on a real-imaginary plot as shown in Figure 19.
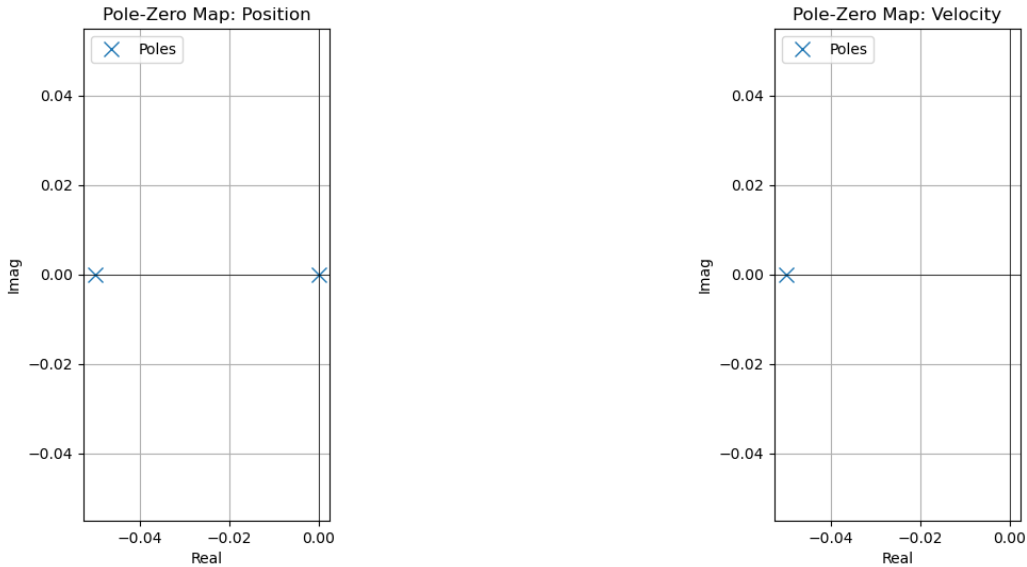
Figure 19: Poles of Car System (Position (left) and Velocity (right))

Notice that the poles of the position system are located at $s = 0$ and $s = -\sigma$ while the pole of the velocity system is located at $s = -\sigma$. The position system has one marginally stable pole and one stable pole while the velocity system has one stable pole. Thus, the position system is marginally stable while the velocity system is stable. Remember if at least one pole is marginally stable and none are unstable the system is considered marginally stable. If all poles are stable the system is considered stable.

11.7.2 Position of a Mass Spring Damper

As derived earlier the transfer function for the position of the mass spring damper is given as

$$\frac{X(s)}{F(s)} = G(s) = \frac{\omega_n{}^2}{s^2 + 2\zeta\omega_n s + \omega_n{}^2} \tag{236}$$

recall also that the time domain solution to this system due to a step function was given as

$$x(t) = f_0 \left(1 - e^{-\zeta\omega_n t} \left(\cos(\omega_d t) + \frac{\zeta}{\sqrt{1-\zeta^2}} \sin(\omega_d t) \right) \right) \tag{237}$$

for the underdamped case. Plotted in the time domain the system oscillated and then attained a steady state value. The poles of the system are located at

$$s = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} \tag{238}$$

The real part of the poles is $-\zeta\omega_n$ which is negative for all positive values of ζ and ω_n. Thus, the system is stable for all positive values of ζ and ω_n. Examining the system in the laplace domain is much easier than examining the time domain solutions because you can simply look at the poles of the transfer function. It is also beneficial to plot the poles of the system on a real-imaginary plot as shown in Figure 20.
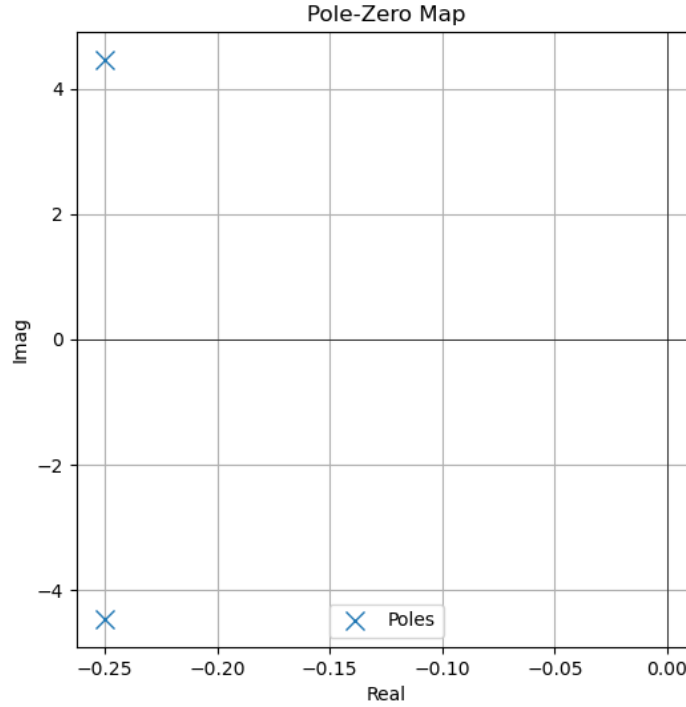
Figure 20: Poles of Mass Spring Damper System

Notice that the poles of the mass spring damper system are located in the left half of the complex plane. The real part of the poles is negative and thus the system is stable. The imaginary part of the poles determines the frequency of oscillation while the real part of the poles determines the rate of decay. The further left the poles are located in the complex plane, the faster the system will decay to its steady state value. Thus again the transfer function shows it's usefulness in determining the stability of a system as well as whether it oscillates or not. In this case the system is stable (left half plane) and oscillatory (imaginary part non-zero).

11.7.3 Attitude Dynamics of a Satellite or Quadcopter

As derived earlier, the transfer function of this system is given as $\frac{\Theta(s)}{F(s)} = G(s) = \frac{2d}{Js^2}$. The poles of the system are located at $s = 0$ and $s = 0$. Both poles have real parts equal to zero and thus the system is marginally stable

because they are on the y-axis of the real-imaginary plane. Physically, this marginal stability is because the angle of the satellite/quadcopter will continue to increase quadratically over time due to the constant angular acceleration from a constant force from the propellor/thrusters. The marginal stability can be seen in the time domain with the system increasing as t^2. However, examining the system in the laplace domain is much easier than examining the time domain solutions because you can simply look at the poles of the transfer function. Again, it is beneficial to plot the poles of the system on a real-imaginary plot as shown in Figure 21.
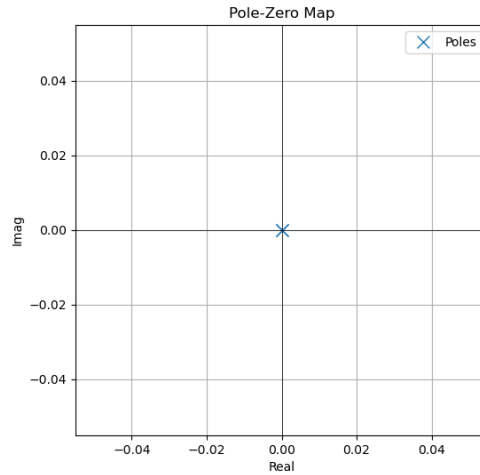

Figure 21: Poles of Satellite/Quadcopter System

Notice that the poles of the satellite/quadcopter system are located at $s = 0$ and $s = 0$. Both poles have real parts equal to zero and thus the system is marginally stable and increases quadratically over time. A single pole at zero would increase linearly but two poles at the origin increases quadratically.

11.7.4 Pitch Dynamics of an Aircraft

11.7.5 Pitch Dynamics of a Rocket

11.7.6 Angle of an Inverted Pendulum

12 Aerospace State Estimation

12.1 Sensor Measurement

During the standard estimation procedure, it is assumed that measurements are made that relate to the state or the state is directly measured. If the state is directly measured like star trackers no special formulation need to made. However, other sensors such as Sun sensors, magnetometers and horizon sensors measure a vector in 3-D space. In general a measurement $\bar{y}_k$ can be expressed by the nonlinear equation shown below where $\vec{x}$ is the state vector.

$$\bar{y}_k = \vec{h}(\vec{x}_k) + \vec{\nu}_k \tag{239}$$

The vector $\vec{\nu}_k$ is noise associated with the sensor [12, 11]. If the system is linearized about some equilibrium point the measurement equation can be written as

$$\bar{y}_k = \mathbf{h}_k \vec{x}_k + \vec{\nu}_k \tag{240}$$

where $\mathbf{h}_k = \partial \vec{h} / \partial \vec{x}$. It's easy to see here that in the case of the star tracker the matrix $\mathbf{h}_k$ is just the identity matrix. The noise vector $\vec{\nu}_k$ is assumed to be gaussian white noise while the covariance $cov()$ is given by the equation below using the expectation operator $\mathbf{E}()$.

$$cov(\vec{\nu}_k) = \mathbf{E}(\vec{\nu}_k \vec{\nu}_k^T) = \mathbf{R}_k \tag{241}$$

If a measurement is made by a Sun sensor or similar where a vector in 3-D space can be compared to a known inertial reference vector the measurement update can be given as

$$\bar{r}_k^B = \mathbf{T}_{BI}(\vec{q}_k) \vec{r}_k^I + \vec{\nu}_k \tag{242}$$

where $\bar{r}_k^B$ is a measurement in the body reference frame at time t_k. The angular velocity measurement in particular can be denoted as $\bar{\omega}_k$. Measurements are typically polluted with bias and white noise. For example, the angular velocity measurement can be given as

$$\bar{\omega} = \vec{\omega} + \vec{b} + \vec{\eta}_g \tag{243}$$

where $\vec{b}$ is a bias that has dynamics given by $\dot{\vec{b}} = \vec{\eta}_b$. The vectors $\vec{\eta}_g$ and $\vec{\eta}_b$ are standard Gaussian white noise vectors. Typically white noise can be filtered out using lowpass filters, complimentary filters or even Kalman Filters while bias can just be substracted. Thus, the estimate for the angular velocity can be written as

$$\tilde{\omega} = \bar{\omega} - \tilde{b} \tag{244}$$

where $\tilde{b}$ is the estimate of the bias.

12.2 Linear Least Squares

In order to understand the nature of a Kalman filter, the linear least squares solution is shown below. Assume for the moment that M independent measurements are made such that $\bar{Y} = [\bar{y}_1, ..., \bar{y}_M]^T$.

$$\bar{Y} = \mathbf{H}\vec{x} + \vec{V} \tag{245}$$

In this case $\mathbf{H} = [\mathbf{h}_1, ..., \mathbf{h}_M]^T$ and $\vec{V} = [\nu_1, ..., \nu_M]^T$. The vector $\vec{V}$ is a vector of error values between your measurements and the actual truth signals $Y = \mathbf{H}\vec{x}$. Absent of all measurement and model noise there would be a unique solution to this problem to solve for the vector $\vec{x}$. The matrices $\bar{Y}$ and $\mathbf{H}$ are known and are the measurements and the output equation relating the measurements to the state values in $\vec{x}$ respectively. Because of measurement and model noise, a unique solution is not possible. That is, the problem is overconstrained since typically the number of measurements is larger than the number of unknowns. Take the linear example as shown in the figure below.
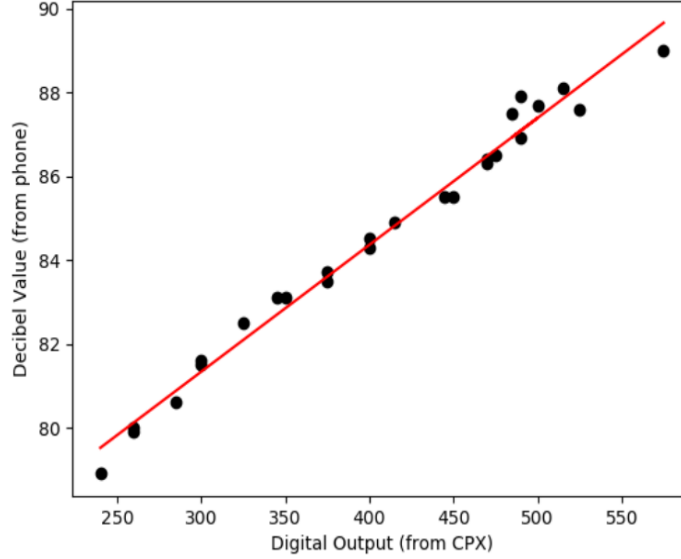
Figure 22: Linear Regression Example

In this case the ordinate axis is the output Y and the abscissa is the independent variable that characterizes the matrix $\mathbf{H}$. The black dots then are the measurements $\bar{Y}$ while the trend line is the estimate $\tilde{Y} = \mathbf{H}\tilde{x}$. In this case the residuals $\hat{Y} = \tilde{Y} - \bar{Y}$ is the distance between the trend line in red and the black dots (the measurements). For this linear example, the unknowns would be the slope and intercept. It is clear here that there exists no linear solution $\vec{x}$ that goes through all black data points. Thus, the equation below can be constructed.

$$\bar{Y} = \mathbf{H}\tilde{x} + \hat{Y} \tag{246}$$

This implies that the trendline $\tilde{Y}$ would go through all data points if $\hat{Y}$ were zero. Thus the solution to this problem was originally found by Gauss [32] and involved minimizing the residuals between $\bar{Y}$ and $\tilde{Y}$ (the estimated Y values). To do this, a cost function is generated such that

$$J = \frac{1}{2}\hat{Y}^T\hat{Y} \tag{247}$$

Substituting in the equation $\hat{Y} = \bar{Y} - \mathbf{H}\tilde{x}$ and minimizing the cost function $\partial J/\partial \tilde{x} = 0$ results in the solution below.

$$\tilde{x} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\bar{Y} \tag{248}$$

Note that the equation above only works if the number of measurements M is greater than or equal to the number of unknowns N. If not, the solution will always be rank deficient and no solution will be found. This is called an under constrained problem. In this there are an infinite number of solutions that satisfy $\bar{Y} = \mathbf{H}\vec{x}$ even in the presence of modeling errors. In order to get around this issue Lagrange's method of optimization is used [33]. For problems like this the residuals between the estimate $\tilde{Y}$ and the measured signals $\bar{Y}$ can be easily made to be zero. Thus minimizing the residuals is trivial since the solution will still be an infinite number of solutions. Therefore a constraint can be placed where $\bar{Y} = \tilde{Y} = \mathbf{H}\tilde{x}$. In order to find a unique solution then the requirement is placed to minimize the estimate $\tilde{x}$. In this case, the cost function to be minimized is given by Lagrange's extension to optimization as shown below

$$L = \frac{1}{2}\tilde{x}^T\tilde{x} + \lambda^T(\bar{Y} - \mathbf{H}\tilde{x}) \tag{249}$$

The cost function above utilizes the method of Lagrange multipliers in order to satisfy the constraint that the solution must pass through all measurements again only if the number of measurements M is less than the number of unknowns N. In the equation above the vector $\tilde{x}$ must be solved and so must the Lagrange multipliers λ. The solution to the problem above requires $\partial L/\partial \tilde{x} = 0$ and $\partial L/\partial \lambda = 0$. Carrying out the partial derivatives and solving for the estimate yields the following equations.

$$\tilde{x} = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}\bar{Y} \tag{250}$$

58

Note, it is standard practice in state estimation to have at least as many measurements as unknowns. In this case $M = N$ and Gauss' solution is sufficient.

12.3 Weighted Least Squares

The weighted least squares solution is found by setting the cost function equal to $J = \frac{1}{2}\hat{Y}^T\mathbf{W}\hat{Y}$ where $\mathbf{W}$ is a positive definite and symmetric weighting matrix. The solution then is shown below.

$$\tilde{x} = (\mathbf{H}^T\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{W}\bar{Y} \tag{251}$$

In the standard Kalman Filter approach, the weighting matrix is given by the inverse covariance of the error $\mathbf{r} = \mathbf{E}[\vec{v}\vec{v}^T]$. Placing this into a matrix yield $\mathbf{W} = \mathbf{R}^{-1}$ where $\mathbf{R} = diag([\mathbf{r}_1, ... \mathbf{r}_M])$. The weighted least squares solution then reduces to

$$\tilde{x} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\bar{Y} \tag{252}$$

12.4 A Priori Knowledge of the State Vector

If a priori knowledge is obtained via other means or in the case of the standard Kalman Filter from integration of the state, it is possible to obtain an updated estimate of the state based on the previous state estimate and the new sensor measurements. First, the a priori estimate $\tilde{x}^-$ is written as

$$\tilde{x}^- = \vec{x} + \vec{w} \tag{253}$$

where $\vec{w}$ is model noise associated with the error in the state estimate. The covariance of this noise is also denoted as a matrix and defined below.

$$cov(\vec{w}) = \mathbf{E}(\vec{w}\vec{w}^T) = \mathbf{q} \tag{254}$$

In this case it is desired for the updated measurement to be some linear combination of the a priori equation and the measurements such that

$$\tilde{x} = \mathbf{\Lambda}\bar{Y} + \mathbf{\Gamma}\tilde{x}^- \tag{255}$$

The matrices $\mathbf{\Lambda}$ and $\mathbf{\Gamma}$ have an added constraint which can be shown by assuming the a priori measurement is perfect $\tilde{x}^- = \vec{x}$ and the measurements $\bar{Y} = Y = \mathbf{H}\vec{x}$. In this case, we must have the updated estimate equal the truth signal. $\tilde{x} = \vec{x}$. Rearranging the equation above yields

$$\vec{x} = \mathbf{\Lambda}\mathbf{H}\vec{x} + \mathbf{\Gamma}\vec{x} \tag{256}$$

which means that $(\mathbf{\Lambda}\mathbf{H} + \mathbf{\Gamma}) = \mathbf{I}$ Again using the method of lagrange multipliers the cost function to be minimized is given as

$$L = \mathbf{E}[\frac{1}{2}\hat{x}^T\hat{x} + \lambda^T(\mathbf{I} - \mathbf{\Lambda}\mathbf{H} - \mathbf{\Gamma})] \tag{257}$$

where $\hat{x} = \tilde{x} - \vec{x}$ and again $\mathbf{E}$ is the expectation operator. Remembering that $\mathbf{q}$ is the covariance of the model noise and $\mathbf{r}$ is the covariance of the measurement noise, the solution to the minimization problem is given by the equation below.

$$\tilde{x} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{q}^{-1})^{-1}(\mathbf{H}^T\mathbf{R}^{-1}\bar{Y} + \mathbf{q}^{-1}\tilde{x}^-) \tag{258}$$

Note that this solution assumes that $\mathbf{E}(\vec{w}\vec{v}^T) = 0$. Measurement and model noise are uncorrelated.

12.5 Complimentary Filter

Looking at the equation for the A priori knowledge it is possible to formulate the complimentary filter. First, the measurements are assumed to be identical to the state vector such that $\mathbf{h}_k = \mathbf{I}$. From here a few extremes are shown below. First, assume that the measurement error is very low such that the $cov(\vec{\nu}) << 1$ while the model noise $\vec{w}$ is very large approaching infinity. In this case, $\mathbf{q}^{-1} = 0$. Substituting this into the weighted apriori equation yields

$$\tilde{x} = average(\bar{Y}) \tag{259}$$

which essentially states that the estimate completely believes the sensor measurement. If instead we assume that the model noise is perfect such that $cov(\vec{w}) << 1$ and the sensor noise is approaching infinity, then $\mathbf{R}^{-1} = 0$. This yields the following equation.

$$\tilde{x} = \tilde{x}^- \tag{260}$$

Thus it can be seen that there is a sliding bar between believing the apriori estimate or the sensor measurement. As such it is possible to develop a much simpler filter. First a constraint is placed on $\mathbf{q}$ and $\mathbf{R}$ such that

$$(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{q}^{-1}) = \mathbf{I} \tag{261}$$

This causes the update law to reduce to the following

$$\tilde{x} = \mathbf{H}^T\mathbf{R}^{-1}\bar{Y} + \mathbf{q}^{-1}\tilde{x}^- \tag{262}$$

If only one measurement is investigated the equation collapses to the following.

$$\tilde{x} = \mathbf{r}^{-1}\bar{y} + \mathbf{q}^{-1}\tilde{x}^- \tag{263}$$

The constraint also collapses to

$$\mathbf{r}^{-1} + \mathbf{q}^{-1} = \mathbf{1} \tag{264}$$

If $\mathbf{q}^{-1} = \mathbf{s}$ and $\mathbf{r}^{-1} = \mathbf{1} - \mathbf{s}$ the update law simplifies to

$$\tilde{x} = (\mathbf{1} - \mathbf{s})\bar{y} + \mathbf{s}\tilde{x}^- \tag{265}$$

Here it is clear that if $\mathbf{s} = \mathbf{1}$ the new estimate will be equal to the old estimate meaning that the sensor noise is approaching infinite. If $\mathbf{s} = 0$ it means that the new estimate is equal to the sensor measurement meaning the model noise is approaching infinity. This is a simple crude first order filter that can be used when only a simple understanding of covariance is known.

12.6   Sequential Linear Estimator

In the above two scenarios, it is assumed that all measurements from 1 to M are known at the same time instant t and thus the least squares estimate can be done "all at once". For discrete time sensors on board a spacecraft this is not possible. For example, if we take the weighted least squares solution assuming we have a *0th* batch of measurements, the estimate of $\tilde{x}$ would be

$$\tilde{x}_0 = (\mathbf{h}_0^T\mathbf{w}_0\mathbf{h}_0)^{-1}\mathbf{h}_0^T\mathbf{w}_0\bar{y}_0 \tag{266}$$

If we then waited Δt seconds for a new set of measurements we would have to obtain a new estimate of $\tilde{x}$ which could be done using the equation below

$$\tilde{x}_1 = (\mathbf{h}_1^T\mathbf{w}_1\mathbf{h}_1)^{-1}\mathbf{h}_1^T\mathbf{w}_1\bar{y}_1 \tag{267}$$

This solution however would only take into account the new measurements. Thus, if larger matrices were constructed like $\mathbf{H} = [\mathbf{h}_0, \mathbf{h}_1]$ the solution for $\tilde{x}$ becomes the same as it was in Equation 251. This process would be tedious if these matrices were computed over and over again. This is because the matrices would continue to grow larger and larger over time and eventually overflow the memory management system on the computer. Thus, a method for updating the state vector every time a new measurement is obtained must be derived. To do this the two equations are substituted into equation 251. Then a covariance matrix is used such that $\mathbf{p} = (\mathbf{h}^T\mathbf{wh})^{-1}$ which never grows in size. Using that simplification and making use of a estimation gain matrix $\mathbf{k}$, the estimation algorithm is as follows:

1. The first measurement is obtained $\bar{y}_0$
2. Compute the matrix $\mathbf{p_0} = (\mathbf{h}_0^T\mathbf{w_0}\mathbf{h_0})^{-1}$
3. Obtain the estimate for $\tilde{x}_0 = \mathbf{p_0}\mathbf{h}_0^T\mathbf{w_0}\bar{y}_0$ (Notice that if you use the equation above this is the same solution as the weighted least squares estimate)
4. Every time a new measurement, $\bar{y}_k$, is obtained use the recursive least squares update law shown in the equation below.

$$\begin{aligned} \mathbf{k}_{k+1} &= \mathbf{p}_k\mathbf{h}_{k+1}^T[\mathbf{h}_{k+1}\mathbf{p}_k\mathbf{h}_{k+1}^T + \mathbf{w}_k^{-1}]^{-1} \\ \mathbf{p}_{k+1} &= [\mathbf{1} - \mathbf{k}_{k+1}\mathbf{h}_{k+1}]\mathbf{p}_k \\ \tilde{x}_{k+1} &= \tilde{x}_k + \mathbf{k}_{k+1}(\bar{y}_{k+1} - \mathbf{h}_{k+1}\tilde{x}_k) \end{aligned} \tag{268}$$

In the special case where the weighting matrix $\mathbf{w}_k$ is equal to a constant $\mathbf{w}$ and the state vector is directly measured such that $\mathbf{h}_k$ is also identity, the sequential linear estimator gives the following simplified steps.

1. The first measurement is obtained $\bar{y}_0$
2. Compute $\mathbf{p}_0 = \mathbf{w}^{-1}$
3. Obtain the estimate for $\tilde{x}_0 = \bar{y}_0$ (this is a fault of $\mathbf{h}_k$ being identity)
4. Every time a new measurement, $\bar{y}_k$, is obtained use the recursive least squares update law shown in the equation below.

$$\begin{aligned} \mathbf{k}_{k+1} &= \mathbf{p}_k[\mathbf{p}_k + \mathbf{w}^{-1}]^{-1} \\ \mathbf{p}_{k+1} &= [\mathbf{1} - \mathbf{k}_{k+1}]\mathbf{p}_k \\ \tilde{x}_{k+1} &= \tilde{x}_k + \mathbf{k}_{k+1}(\bar{y}_{k+1} - \tilde{x}_k) \end{aligned} \tag{269}$$

12.7 The Continuous Time Complimentary Filter

In the above section a discrete sequential least squares update law was formulated. In that derivation it is assumed that the state estimate is held constant in between state measurements. It is possible however to integrate a model of the state dynamics and use that estimate in between state measurements. The is the start of a Kalman Filter. To formulate the Continuous Time Complimentary Filter the dynamics of the system are written such that

$$\dot{\vec{x}} = \mathbf{f}\vec{x} + \mathbf{g}u + \mathbf{m}\vec{w}$$
$$\vec{y} = \mathbf{h}\vec{x} \tag{270}$$

where the initial conditions are $\vec{x}_0$ and $\vec{w}$ is a modeling noise term where $\mathbf{E}[\vec{w}\vec{w}^T] = \mathbf{q}$ just as was defined in the a priori estimation section. The model dynamics are set up such that

$$\dot{\tilde{x}} = \tilde{\mathbf{f}}\tilde{x} + \tilde{\mathbf{g}}u + \vec{\gamma}$$
$$\tilde{y} = \mathbf{h}\tilde{x} \tag{271}$$
$$\bar{y} = \mathbf{h}\vec{x} + \vec{v}$$

where again $\bar{y}$ is the state measurement and $\vec{v}$ is noise associated with the sensor where $\mathbf{E}[\vec{v}\vec{v}^T] = \mathbf{r}$. The term $\vec{\gamma}$ is added as a psuedo control which can be whatever we want. The idea is for u to be the control input to drive $\vec{x} \to \vec{x}_c$ while the psuedo control is for the observer dynamics to drive $\tilde{y} \to \bar{y}$. The model dynamics are going to deviate in between sensor measurements so if the observer dynamics are designed properly the estimate can converge to the measurement. Of course, this means your estimate is only as good as your measurement noise but it is a start. To design the psuedo control law, measurement feedback is used in the same form as standard unity feedback control laws such that $\vec{\gamma} = bfk\hat{y}$ where $\hat{y}$ is the difference between the estimate and the measurement. The closed loop dynamics can then be written as

$$\dot{\tilde{x}} = (\tilde{\mathbf{f}} - \mathbf{kh})\tilde{x} + \tilde{\mathbf{g}}u + \mathbf{k}\bar{y} \tag{272}$$

Looking at this equation it's hard to see the effect of the observer. Thus the error dynamics must be investigated where $\hat{x} = \tilde{x} - \vec{x}$. For the simple case it is assumed that $\mathbf{f} = \tilde{\mathbf{f}}$ and $\mathbf{g} = \tilde{\mathbf{g}}$. The closed loop error dynamics can then be written as

$$\dot{\hat{x}} = (\mathbf{f} - \mathbf{kh})\hat{x} + \mathbf{k}\vec{v} \tag{273}$$

in this case the solution to this equation is

$$\hat{x}(t) = \hat{x}_0 e^{(\mathbf{f}-\mathbf{kh})t} + \vec{\eta} \tag{274}$$

where the term $\vec{\eta}$ is a function of the noise term $\mathbf{k}\vec{v}$. In this case, if $\mathbf{k}$ is chosen to be large, the error dynamics will be very fast but the noise term will be very large. If $\mathbf{k}$ is chosen to be very small the error dynamics will be slow but the error term will not be a prevalent. The issue with this filter of course comes with how to tune the gain matrix $\mathbf{k}$ which is what the Kalman filter seeks to address.

12.8 The Continuous Discrete Kalman Filter

In the case of the continuous discrete Kalman Filter, the model dynamics are integrated just as in the complimentary filter. The only difference is instead of using a continuous observer the state estimate is updated every time a new measurement is obtained much like the sequential least squares technique. First, let's write the model dynamics as before without the observer and the measurement equations are written such that the measurement is taken at timestep t_k and thereafter every Δt.

$$\dot{\tilde{x}} = \tilde{\mathbf{f}}\tilde{x} + \tilde{\mathbf{g}}u$$
$$\tilde{y} = \mathbf{h}\tilde{x} \tag{275}$$
$$\bar{y}_k = \mathbf{h_k}\vec{x}(t_k) + \vec{v}_k$$

The update equation is written using the continuous observer dynamics used for the complimentary filter only in this case the update is discrete.

$$\tilde{x}_k^+ = \tilde{x}_k^- + \mathbf{k}_k(\bar{y}_k - \mathbf{h_k}\tilde{x}_k^-) \tag{276}$$

In this case $\tilde{x}_k^+$ is the estimated state after the update while $\tilde{x}_k^-$ is the estimate before the update. The equation for the covariance update and the Kalman Gain matrix are identical in that the derivation is formulated just as it was before. The equations are shown below again only $+$ and $-$ is used to denote the matrices before and after update.

$$\mathbf{k}_k = \mathbf{p}_k \mathbf{h}_k^T [\mathbf{h}_k \mathbf{p}_k^- \mathbf{h}_k^T + \mathbf{r}]^{-1}$$
$$\mathbf{p}_k^+ = [\mathbf{1} - \mathbf{k}_k \mathbf{h}_k]\mathbf{p}_k^- \tag{277}$$

In the sequential linear estimator however, the covariance matrix was set using a weighted least squares approach. In this case the covariance matrix is set such that $\mathbf{p} = \mathbf{E}[\hat{x}\hat{x}^T]$. Taking a derivative of this equation and substituting in the closed loop error dynamics yields the covariance propagation equation shown below.

$$\dot{\mathbf{p}} = \tilde{\mathbf{f}}\mathbf{p} + \mathbf{p}\tilde{\mathbf{f}}^T + \mathbf{mqm}^T \tag{278}$$

The final Continuous Discrete Kalman Filter then goes like this.

1. Integrate the model dynamics in Equation 275 and the covariance dynamics in equation 278

2. When a measurement is received, the Kalman Gain matrix is computed using equation 277.

3. Equation 277 is also used to update the covariance matrix

4. Finally, equation 276 is used to update the state vector estimate and then the process repeats.

An example figure is shown below for a first order system. In this figure the blue stars represent discrete sensor measurements with some noise. Everytime the sensor is updated the model performs and update and instantaneously changes to a new value. The model then integrates (incorrectly due to model mismatch) until a new sensor measurement is obtained. In this case the model is so inaccurate it makes more sense to update the sensor more frequently or perform some sort of adaptive control algorithm to estimate the plant dynamics.
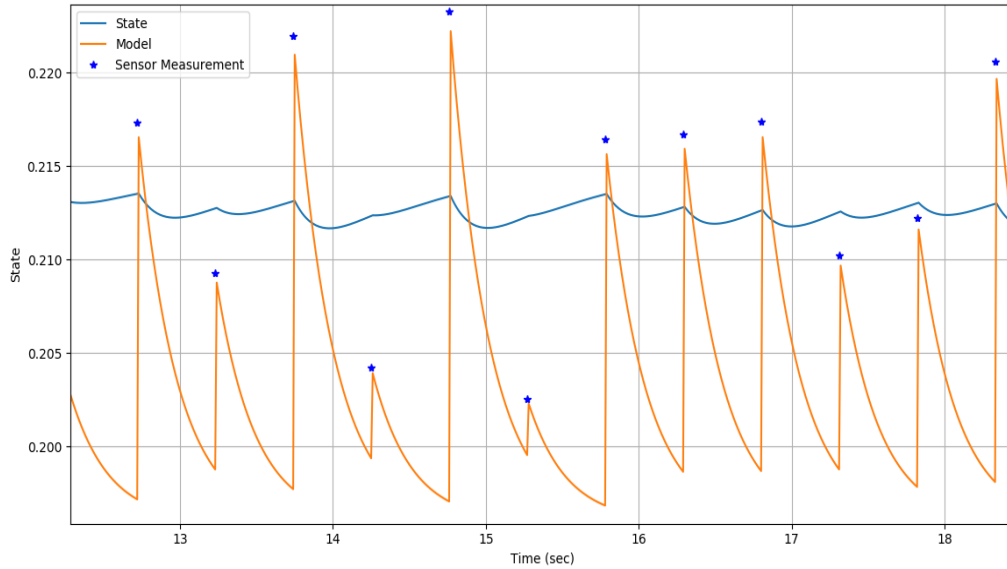
Figure 23: First Order Kalman Filter Example

12.9 Kalman Filter for Spacecraft Dynamics

Attitude estimation involves a combination of attitude determination and state estimation. Assuming at time $t = t_0$ the attitude estimation algorithm is performed and an estimate of the quaternion is obtained as $\tilde{q}_0$. If discrete regular angular velocity ($\bar{\omega}_k$) measurements are made every Δt seconds, the quaternion can be estimated by simply integrating the attitude equations of motion. Even if perfect sensor measurements are made, it is possible to integrate these equations of motion over time and the quaternion $\vec{q}$ will be much different than the estimated quaternion $\tilde{q}$. Thus, the attitude estimation algorithm can run again to obtain a new absolute quaternion measurement. The equations of motion are integrated and when a new sensor measurement is obtained the estimated state is updated based on the estimated covariance combined with and estimate of model errors and sensor errors. Finally, it is possible to create an Extended State Kalman Filter (EKF) which can estimate sensor inaccuracies simply by finding the least squares solution between the sensor measurements and state estimates. The sections that follow details the Kalman Filter for Spacecraft Dynamics as well as the extended state version which estimate bias values in the rate gyro.

First, the 4-dimensionality of the quaternion renders the above Kalman filter formulation to be impossible mostly because the quaternion derivative is a 4 by 1 matrix while the angular velocity vector is a 3 by 1. Furthermore, the quaternion derivative is not linear and cannot be expressed as the linear matrices in the previous section. As such the Kalman Filter must be updated somewhat. The derivative of the state $\vec{q}$ is cumbersome and follows the reference in [14]. First the angular velocity measurement is substituted into the derivative of quaternions where the $\mathbf{\Omega}()$ and $\chi()$ identity is used to separate out the white noise parameter.

$$\dot{\vec{q}} = \frac{1}{2}\mathbf{\Omega}(\vec{\omega})\vec{q} = \frac{1}{2}\mathbf{\Omega}(\bar{\omega} - \vec{b} - \vec{\eta}_g)\vec{q} = \frac{1}{2}\mathbf{\Omega}(\bar{\omega} - \vec{b})\vec{q} - \frac{1}{2}\chi(\vec{q})\vec{\eta}_g \qquad (279)$$

At this point an error quaternion is created using the difference between $\vec{q}$ and $\tilde{q}$. Recall that the error quaternion is given by the equation below. The full equation is shown in 49.

$$\delta\vec{q} = \vec{q} \oplus \tilde{q}^{-1} \qquad (280)$$

The derivative of this difference quaternion is beyond the scope of this report but can be found in [34].

$$\dot{\delta\vec{q}} = \left\{ \begin{matrix} 0 \\ -\mathbf{S}(\tilde{\omega})\delta\vec{\epsilon} \end{matrix} \right\} + \frac{1}{2}\mathbf{\Omega}(\delta\vec{\omega})\delta\vec{q} \qquad (281)$$

where $\delta\vec{\omega} = \vec{\omega} - \tilde{\omega}$ and $\delta\vec{\epsilon} = \vec{\epsilon} - \tilde{\epsilon}$. Recall that $\tilde{\omega} = \bar{\omega} - \tilde{b}$. The second term in the equation above can be expanded using the equations in Section 12.1. Note that $\delta\vec{\omega}$ simplifies to $-\delta\vec{b} - \vec{\eta}_g$ and $\delta\vec{q} = [\delta q_0, \delta\vec{\epsilon}]^T$.

$$\dot{\delta\vec{q}} = \left\{ \begin{matrix} 0 \\ -\mathbf{S}(\tilde{\omega})\delta\vec{\epsilon} \end{matrix} \right\} - \frac{1}{2} \left\{ \begin{matrix} -\delta\vec{\epsilon}^T\delta\vec{b} \\ \delta q_0 \delta\vec{b} + \mathbf{S}(\delta\vec{\epsilon})\delta\vec{b} \end{matrix} \right\} - \frac{1}{2} \left\{ \begin{matrix} -\delta\vec{\epsilon}\vec{\eta}_g \\ \delta q_0\vec{\eta}_g + \mathbf{S}(\delta\vec{\epsilon})\vec{\eta}_g \end{matrix} \right\} \qquad (282)$$

In order to proceed further, small angle approximations are made such that $|\delta\vec{q}| << 1$. The latter 3 variables in the quaternion are further approximated as $\delta\vec{\rho} = \delta\vec{\epsilon}$. In order to fit in with the standard Kalman filter, the state vector $\vec{x} = \vec{\rho}$ and thus the state dynamics $\dot{\delta\vec{x}}$ can then be written as

$$\dot{\delta\vec{x}} = \dot{\delta\vec{\rho}} = -\mathbf{S}(\tilde{\omega})\delta\vec{d} - \frac{1}{2}\delta\vec{b} - \frac{1}{2}\vec{\eta}_g \qquad (283)$$

In order to extract the attitude quaternion from the approximated state the following equations are used.

$$\delta\vec{\epsilon} = \frac{\delta\vec{\rho}}{\sqrt{1 + \delta\vec{\rho}^T\delta\vec{\rho}}} \quad q_0 = \frac{1}{\sqrt{1 + \delta\vec{\rho}^T\delta\vec{\rho}}} \qquad (284)$$

12.10 Extended State Kalman Filter

As shown in the previous section, a Kalman filter can be used to estimate the state. The standard Kalman filter however can be extended to include the bias of the angular velocity measurement. Thus the state vector is augmented to be $\vec{x} = [\vec{q}, \vec{b}]^T$. Since the derivative of the bias is the white noise vector, the difference state vector after much simplification is shown below.

$$\dot{\delta\vec{x}} = \left\{ \begin{matrix} \dot{\delta\vec{\rho}} \\ \dot{\delta\vec{b}} \end{matrix} \right\} = \begin{bmatrix} -\mathbf{S}(\tilde{\omega}) & -\frac{1}{2}\mathbf{I}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} \end{bmatrix} \left\{ \begin{matrix} \delta\vec{\rho} \\ \delta\vec{b} \end{matrix} \right\} + \left\{ \begin{matrix} -\frac{1}{2}\vec{\eta}_g \\ \vec{\eta}_b \end{matrix} \right\} \qquad (285)$$

In this formulation $\delta\vec{b} = \vec{b} - \tilde{b}$. The derivative is then $\dot{\delta\vec{b}} = \vec{\eta}_b - 0$. It is assumed that the derivative of the estimate is zero and thus is only updated when sensor measurements are made. The states equation above can be reduced to the state space form shown below.

$$\dot{\delta\vec{x}} = \mathbf{A}\delta\vec{x} + \vec{\eta} \qquad (286)$$

12.11 Euler Angle Estimation via IMU

Using an IMU it is possible to obtain Euler Angles assuming a Flat Earth Approximation. Recall that Euler angles are a 3D transformation from the Inertial frame to the Body Frame (See Section 5.1). The angle ϕ and θ can directly be measured via the accelerometer by creating a relationship between the gravity vector in the inertial and body frames. The heading angle can be measured by creating a relationship between the magnetic field in the body frame and the inertial frame using a magnetometer. The rate gyro can be used to integrate the angular

velocity to obtain Euler angles as well but is prone to drift. The accelerometer though is prone to errors when the vehicle experiences large acceleration loads. Thus, typically the Euler angles from the rate gyro are fused with the estimates from the magnetometer and the accelerometer. Still, some errors can still exist and the Euler angles can be fused with estimates from GPS but that will be explored in a separate section. First, let's examine the direct estimation of roll and pitch using the accelerometer.

12.11.1   Direct Measurement of Roll and Pitch

Understand that the gravity vector in the inertial frame can be written as $\mathbf{C}_I(\vec{g}) = [0, 0, g]^T$. However, since the first rotation in the Euler angle sequence is about the z-axis, the gravity vector in the A frame and Inertial (I) frames are identical. That is, $\mathbf{C}_A(\vec{g}) = \mathbf{C}_I(\vec{g})$. Normalizing the gravity vector yields $\mathbf{C}_A(\bar{g}) = [0, 0, 1]^T$. The measurement from the accelerometer must also be normalized such that $\bar{a}_{B/I} = \hat{a}_{B/I}/\|\hat{a}_{B/I}\|$. Since the aircraft is always experiencing gravity, and the accelerometer is measuring the acceleration vector a relationship can be obtained between the gravity vector in the A frame and the acceleration vector in the body frame. Note that an assumption is being made here. It is assumed that the only acceleration being experienced is gravity. Therefore, if any external accelerations are experienced by the vehicle via thrust or aerodynamics, this equation is not valid. Still, for small UAV applications these equations can be accurate if fused properly with the rate gyro measurements.

$$\mathbf{C}_B(\bar{a}_{B/I}) = \mathbf{T}_{NRB}^T \mathbf{T}_{ANR}^T \mathbf{C}_A(\bar{g}) = \begin{Bmatrix} -s_\theta \\ s_\phi c_\theta \\ c_\phi c_\theta \end{Bmatrix} \tag{287}$$

The equation above takes the normalized gravity vector in the A frame and rotates it to the body frame through the no roll frame. Since the rotation is from the A frame to the body frame, only two rotations are required. Notice also that the first row can be used to obtain the pitch angle.

$$\theta = -sin^{-1}(\bar{a}_x) \tag{288}$$

The roll angle can then be obtained by taking the second two rows and dividing them together to get a tangent function.

$$\phi = tan^{-1}\left(\frac{\bar{a}_y}{\bar{a}_z}\right) \tag{289}$$

Note that this equation is only valid if $c_\theta \neq \pi/2$. This means the vehicle cannot fly straight up. For quadcopters and airplanes this is pretty typical for standard and level flight. For rockets however, either the IMU must be placed in an orientation that doesn't result in this singularity at launch or quaternions must be used. For spacecraft an entirely different algorithm is needed and is explained in a different section.

Note that the pitch angle equation is written using the inverse sine function. Often times it is beneficial to compute the pitch angle using the inverse tangent function so that the *atan2* function may be utilized on a microcontroller which determines the quadrant of the angle more robustly. To do this the gravity vector must be written in the no roll frame.

$$\mathbf{C}_{NR}(\bar{g}) = \mathbf{T}_{ANR}^T \mathbf{C}_A(\bar{g}) = \begin{Bmatrix} -s_\theta \\ 0 \\ c_\theta \end{Bmatrix} \tag{290}$$

Then the acceleration vector is rotated to no roll frame as well from the body frame

$$\mathbf{C}_{NR}(\bar{a}_{B/I}) = \mathbf{T}_{NRB}\mathbf{C}_B(\bar{a}_{B/I}) = \begin{Bmatrix} \bar{a}_x \\ \bar{a}_y c_\phi - \bar{a}_z s_\phi \\ \bar{a}_y s_\phi + \bar{a}_z c_\phi \end{Bmatrix} \tag{291}$$

Setting the two equations above equal to each other and diving the first row by the second row results in a tangent equation for pitch. This result is shown in the equation below.

$$\theta = -tan^{-1}\left(\frac{\bar{a}_x}{\bar{a}_y s_\phi + \bar{a}_z c_\phi}\right) \tag{292}$$

Notice that these equations for pitch can be constructed by drawing a right triangle with the gravity vector as the hypotenuse. The sine function is the opposite side of the triangle divided by the hypotenuse which is 1 since the gravity vector was normalized while the inverse tangent function is the opposite side over the adjacent side.

12.11.2 Direct Measurement of Yaw

In order to obtain the yaw angle of the vehicle through a direct measurement, the magnetometer is used. First it is assumed that the magnetic field strength is a constant through the flight of the vehicle and that it is oriented along the x-axis in the inertial frame of the Flat Earth Approximation. Remember that the x-axis is North using the Flat Earth Approximation. The magnetic field vector of the Earth is then normalized to unity.

$$\mathbf{C}(\bar{\beta}_{\oplus}) = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \tag{293}$$

Again, in order to get a tangent function for the yaw angle estimation, the magnetic field of the Earth is written in the A frame.

$$\mathbf{C}_A(\bar{\beta}_{\oplus}) = \mathbf{T}_{IA}^T \mathbf{C}_I(\bar{\beta}_{\oplus}) = \begin{Bmatrix} c_\psi \\ -s_\psi \\ 0 \end{Bmatrix} \tag{294}$$

The magnetic field measurment of the magnetometer is then written in the A frame as well. However, the magnetometer measures the magnetic field in the body frame. Thus 2 rotations are requires to get from the body frame to the A frame. Again the magnetometer measurement is normalized.

$$\mathbf{C}_A(\bar{\beta}_B) = \mathbf{T}_{ANR}\mathbf{T}_{NRB}\mathbf{C}_B(\bar{\beta}_B) = \begin{Bmatrix} \bar{\beta}_x c_\phi + \bar{\beta}_y s_\theta s_\phi + \bar{\beta}_z c_\phi s_\theta \\ \bar{\beta}_y c_\phi - \bar{\beta}_z s_\phi \\ -\bar{\beta}_x s_\theta + \bar{\beta}_y s_\phi c_\theta + \bar{\beta}_z c_\phi c_\theta \end{Bmatrix} \tag{295}$$

The two equations for magnetic field in the A frame can then be equated. In this case, the second row is divided by the first row to obtain a tangent relationship for yaw. The result for yaw is shown below.

$$\psi = tan^{-1}\left(\frac{\bar{\beta}_z s_\phi - \bar{\beta}_y c_\phi}{\bar{\beta}_x c_\phi + \bar{\beta}_y s_\theta s_\phi + \bar{\beta}_z c_\phi s_\theta} \right) \tag{296}$$

12.12 Low Earth Orbit Attitude Estimation

In LEO the main algorithm begins with obtaining the magnetic field in the body frame using magnetometers $\vec{\beta}_B$. Using the IGRF model the locally measured magnetic field can be compared with the known magnetic field for any given location within its orbit. Using the true data and the measured data, the spacecraft can compute its actual position to the measured position and make the correct adjustments. A Sun measurement is then taken using a Sun sensor $\vec{S}_B$. Once those two independent body frame measurements are taken the inertial reference vectors must be obtained from a database. Startrackers have this database built in; however, for the magnetic field and the Sun vector these must be obtained from a separate database as discussed in Section 9. The idea is that if the position of the Earth is known then the position of the Sun with respect to the Earth is also known. The magnetic field vector can be obtained from the IGRF model as discussed in Section 7.3. The magnetic field vector in the inertial frame is given as $\vec{\beta}_I$. Note that the IGRF model requires the latitude and longitude to be known. Thus, in LEO a GPS is required to feed into the database. The inertial Sun vector $\vec{S}_I$ only requires the Julian time which can be obtained from GPS as well. The julian time is based on the julian day as explained in Section 9.

The initial attitude determination algorithm itself requires two independent vectors. As stated previously, startrackers provided a large enough aperture and enough stars to produce the full quaternion by obtaining multiple unique vectors to unique stars. Multiple solar sensors or multiple magnetometers unfortunately do not obtain non-unique vectors and the algorithm fails. In LEO this is typically done with solar sensors and magnetometers but it can be done with star trackers. In deep space it is typically done with startrackers but it could be possible to obtain a Moon vector that would require a Moon sensor.

The derivation below is done for the LEO case with a Sun and magnetic field measurement. The derivation is identical for the deep space case with a Moon sensor simply by substituting the magnetic field measurment with a Moon measurement. Every vector is first normalized to obtain $\hat{\beta}_B, \hat{\beta}_I, \hat{S}_B, \hat{S}_I$. A triad is then created from body frame vectors using the equations below.

$$\hat{f}_1 = \hat{S}_B \quad \hat{f}_2 = \hat{f}_1 \times \hat{\beta}_B \quad \hat{f}_3 = \hat{f}_1 \times \hat{f}_2 \tag{297}$$

The matrix $\mathbf{F}$ is then created using the triad as an orthonormal basis $F = [\hat{f}_1, \hat{f}_2, \hat{f}_3]$. Similar equations are used for the inertial measurements.

$$\hat{g}_1 = \hat{S}_I \quad \hat{g}_2 = \hat{g}_1 \times \hat{\beta}_I \quad \hat{g}_3 = \hat{g}_1 \times \hat{g}_2 \tag{298}$$

The matrix $\mathbf{G}$ is then created just as the $\mathbf{F}$ matrix such that $\mathbf{G} = [\hat{g}_1, \hat{g}_2, \hat{g}_3]$. The transformation from inertial to body frame is then created using the formula below.

$$\mathbf{T}_{BI} = \mathbf{F}\mathbf{G}^T \tag{299}$$

This matrix above is similar to the matrix in equation 36 and thus the Euler angles can be extracted from the matrix itself using the formulation defined in Section 5.1. Euler can then be converted to quaternions if needed. Note that it is relatively easy to extract Euler angles from the $\mathbf{T}_{IB}$ matrix, it is not so simple to extract quaternions. This is due to the fact that for every orientation there exists two quaternions that represent this space. Thus, it is more ideal to obtain Euler angles from the transformation matrix and then convert them to quaternions.

12.13   Spacecraft Position Estimation using a Ground Station Network (GSN)

There are several types of ground stations depending on the spacecraft's distance from Earth. Ground services may be either Direct-to-Earth (DTE) or space relay. DTE ground stations are located on the Earth's surface. They provide direct point-to-point access with antennas at ground stations. DTE services are great for missions needing frequent, short-duration contacts with high data transfer.

Space relay services involve an intermediate satellite that communicates with a ground station on the Earth's surface. Relay communication satellites for low-Earth orbit spacecraft can be in Geosynchronous Equatorial Orbit (GEO), roughly 36,000 km from the surface of Earth, or in low-Earth orbit. Relays are important for providing communication and tracking when direct-to-ground communications are not feasible due to physical asset visibility constraints. Space-based relay assets give missions full-time coverage and continuous access to communication and tracking services.

Finally, deep space communication is also possible. The Deep Space Network (DSN) is developed to conduct telecommunication and tracking operations with space missions in GEO. This includes missions at lunar distances, the Sun-Earth LaGrange points, and in highly elliptical Earth orbits, and even missions to other planets[35]. The DSN network consists of three ground stations placed around 120 degrees apart on Earth which provides 360 degrees coverage[36].
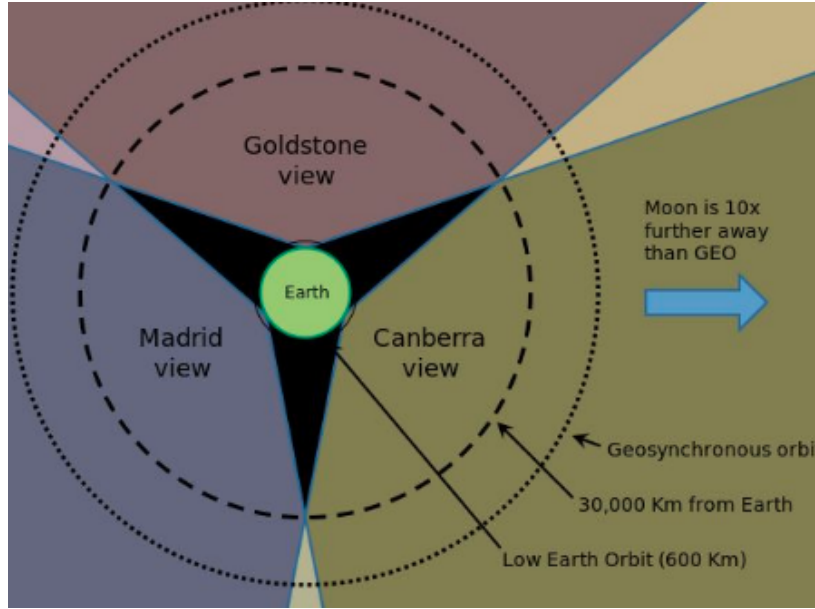
Figure 24: Deep Space Network Satellite Coverage [37]

12.14   Heading Angle and Speed Estimation using GPS

On Earth there is no need for a DSN because the vehicle is within the GPS constellation. Assuming the vehicle has the necessary GPS sensors a full NMEA (National Marine Electronics Association) can be obtained. However, in this example it is assumed that only the latitude, longitude and altitude coordinates are obtained in a discrete

fashion. In order to get heading and speed it is assumed that consecutive measurements are obtained at i and $i + 1$ timestamps. Let's assume that the vehicle is traveling in a specific direction or heading and obtains a GPS coordinate $(\lambda_{LAT,i}, \lambda_{LON,i}, h_i)$ at time t_i. A few seconds later or whenever the update period may be the vehicle moves and the GPS returns a new GPS coordinate $(\lambda_{LAT,i+1}, \lambda_{LON,i+1}, h_{i+1})$ at time t_{i+1}. First, the coordinates are converted to a cartesian coordinate system. This is explained in the External Model section 7.1. This results in x_i, y_i, z_i at time t_i and $x_{i+1}, y_{i+1}, z_{i+1}$ at time t_{i+1}. First, the speed estimate is given by using a simple first order differentiation as given by

$$\begin{aligned} v_x &= (x_{i+1} - x_i)/\Delta t \\ v_y &= (y_{i+1} - y_i)/\Delta t \end{aligned} \tag{300}$$

where $\Delta t = t_{i+1} - t_i$. Note that it is not recomended to compute the velocity in the z-axis as the altitude estimate of GPS is often not very good. Finally, the estimate for heading can follow from the speed estimate and is given as

$$\psi = tan^{-1}\left(\frac{v_y}{v_x}\right) \tag{301}$$

Note that it is recommended to filter these estimates as GPS on its own is only accurate to around 3 meters.

13 Feedback Control

Feedback control is a large and complex topic. This section will introduce some of the basic concepts and methods used in feedback control. The sections that follow will provide many examples for linear time invariant systems as defined in the previous section. However, more complex examples will be discussed in later subsections.

==include graphic of Feedback.png==

13.1 Controllability

Before beginning to talk about control there is a necessary discussion about controllability. Controllability is formally stated as a system where any initial state $x(0) = x_0$ and final state $x_1, t_1 > 0$, there exists a piecewise continuous input $u(t)$ such that $x(t_1) = x_1$. What this means is that a system is controllable if it can be driven from any initial state to any final state in a finite time. This is an important concept because if a system is not controllable then no matter how good the control system is it will never be able to drive the system to the desired state.

For a fixed wing aircraft the system has 12 states with 8 dynamic modes and 4 zero or rigid body modes. For a fixed wing aircraft the system has 12 states with 8 dynamic modes and 4 zero or rigid body modes. A conventional aircraft has 4 controls to control these 12 modes. The easiest way to test the controllability of a system is to compute the controllability matrix. However, the controllability matrix must be computed using a linearized model such that $\dot{\vec{x}} = A\vec{x} + B\vec{u}$. In order to do this the aircraft must be in equilibrium. For this example the aircraft is set with an initial velocity of 20 m/s at an altitude of 200 m. The altitude command is set to 200 m and the heading command is set to zero. Given the zero heading angle command and the symmetry of the configurations investigated the rudder and aileron commands are set to zero. Thus, only the thrust and elevator controls are activated for the trimming procedure. Each configuration is simulated for 200 seconds or until the derivatives of all states except $\dot{x}$ are within a required tolerance. Using this equilibrium point a linear model can be computed by using forward finite differencing assuming that the aircraft model is put in the form $\dot{\vec{x}} = F(\vec{x}, \vec{u})$.

$$\dot{\vec{\delta x}} = \frac{F(\vec{x_0} + \Delta\vec{x_0}, \vec{u_0}) - F(\vec{x_0}, \vec{u_0})}{\Delta\vec{x}}\vec{\delta x} + \frac{F(\vec{x_0}, \vec{u_0} + \Delta\vec{u}) - F(\vec{x_0}, \vec{u_0})}{\Delta\vec{u}}\vec{\delta u} \tag{302}$$

This linear model is the classic linear model where $\dot{\vec{\delta x}} = A\vec{\delta x} + B\vec{\delta u}$. Using this linear model, the controllability matrix can be computed as

$$W_C = [B \ AB \ A^2B \ A^3B \ ... \ A^{N-1}B] \tag{303}$$

where N is the number of states in the system. With the controllability matrix formulated, the rank of the matrix is computed. If the $rank(W_C) = N$ the system is said to be controllable.

13.2 Bang Bang Control of a Satellite

13.3 Proportional Control of a Satellite or Quadcopter

13.4 Proportional Derivative Control of a Satellite or Quadcopter

13.5 Proportional Control of a Car

13.6 Proportional Integral Control of a Car

13.7 Proportional Derivative Integral Control of a Spring Mass Damper System

13.8 Proportional Derivative Control of an Inverted Pendulum

14 Conventional Aerospace Controls

As explained in the previous section, PID control is a common method of controlling mutliple systems. For aerospace systems, PID control is often used as a starting point for feedback but more complex methods are often needed to achieve the desired performance. This section will discuss some of the more advanced control strategies used in aerospace applications.

14.1 Spacecraft Attitude Control Schemes

Many control schemes are needed to orient a satellite and all depend on the application. In LEO magnetorquers can be used to detumble a satellite while thrusters must be used in deep space. In addition reaction wheels can be used to detumble a satellite anywhere in space provided the angular momentum in the satellite does not saturate the reaction wheels. Sections that follow detail the control schemes typically utilized on small sats. A section on PID control for aircraft is also in this section.

14.1.1 B-dot Controller

In LEO, the standard B-dot controller reported in many sources ([38],[39],[40],[41]) can be used to de-tumble a satellite. The standard B-dot controller requires the magnetorquers to follow the control law shown below

$$\vec{\mu}_B = k\mathbf{S}(\vec{\omega}_{B/I})\mathbf{T}_{BI}(\vec{q})\vec{\beta}_I \tag{304}$$

where k is the control gain. Using equation (94) it is possible to write the current in component form again using the identity that $\vec{a} \times \vec{b} = -\vec{b} \times \vec{a}$

$$\begin{Bmatrix} i_x \\ i_y \\ i_z \end{Bmatrix} = \frac{k}{nA} \begin{bmatrix} 0 & \beta_z & -\beta_y \\ -\beta_z & 0 & \beta_x \\ \beta_y & -\beta_x & 0 \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \tag{305}$$

This equation can then be substituted into equation (96) to produce the total torque on the satellite assuming that the magnetorquers can provide the necessary current commanded by equation (305).

$$\begin{Bmatrix} L \\ M \\ N \end{Bmatrix} = -k \begin{bmatrix} \beta_y^2 + \beta_z^2 & -\beta_x\beta_y & -\beta_x\beta_z \\ -\beta_x\beta_y & \beta_x^2 + \beta_z^2 & -\beta_y\beta_z \\ -\beta_x\beta_z & -\beta_y\beta_z & \beta_x^2 + \beta_y^2 \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \tag{306}$$

The goal of the controller here is to drive $\vec{\omega}_{B/I} \to 0$. The literature will show that this is not completely achieved [42]. There are multiple explanations for this. For starters, equation (95) assumes that the magnetic moment is not co-linear with the magnetic field of the Earth. If it is, the result is zero torque applied to the satellite. Furthermore, equation (305) results in zero current if the angular velocity vector of the satellite is co-linear with the magnetic field. Thus, if the magnetic field vector, angular velocity vector or the magnetic moment vector are co-linear, the torque applied to the satellite will be zero. If a new operator is defined such that

$$\mathbf{W}(\mathbf{T}_{BI}(\vec{q})\vec{\beta}_I) = \begin{bmatrix} \beta_y^2 + \beta_z^2 & -\beta_x\beta_y & -\beta_x\beta_z \\ -\beta_x\beta_y & \beta_x^2 + \beta_z^2 & -\beta_y\beta_z \\ -\beta_x\beta_z & -\beta_y\beta_z & \beta_x^2 + \beta_y^2 \end{bmatrix} \tag{307}$$

it is easy to see that the torque applied to a satellite is then simply the angular velocity vector multiplied by this transition matrix. If this transition matrix is put into row-reduced-echelon form it is easy to see that the determinant of this matrix is equal to zero ([43]).

$$rref(\mathbf{W}(\mathbf{T}_{BI}(\vec{q})\vec{\beta}_I)) = \begin{bmatrix} 1 & 0 & -\beta_x/\beta_z \\ 0 & 1 & -\beta_y/\beta_z \\ 0 & 0 & 0 \end{bmatrix} \tag{308}$$

A zero determinant means that there exists a vector $\vec{\omega}_{B/I}$ that will result in zero torque for a given value of the magnetic field. This is typically avoided since the magnetic field of the Earth is time and spatially varying which results in a transition matrix that changes over time due to orientation changes in the satellite as well as changes in the satellite's orbit. However, for low inclination orbits, it's possible for the magnetic field to stay relatively constant with $\beta_x \approx \beta_y \approx 0$. If the satellite is tumbling about the yaw axis such that $p = q = 0$, the yaw torque on the satellite (N) will be zero. Using this simple controller, there is no way to remove the remaining angular velocity from the satellite unless reaction wheels are used.

14.1.2 Reaction Wheel Control

Assuming each reaction is aligned with a principal axis of inertia the control scheme is extremely simple. When the wheels are not aligned the derivation will proceed similar to the reaction control thruster section. The derivation here will just be for the aligned case. In this analysis it is assumed that a torque can be applied to the reaction wheel and thus the angular velocity of the reaction wheel α_{Ri} can be directly controlled. Assuming this a simple PD control law can be used to orient the satellite at any desired orientation using Euler angles for this control law since the satellites are aligned with the principal axes of rotation [3].

$$\alpha_{Ri} = -k_p(\epsilon_i - \epsilon_{desired}) - k_d(\omega_i - \omega_{desired}) \tag{309}$$

In the equation above ϵ denotes either roll ϕ, pitch θ or yaw ψ depending on which reaction wheel is being used. The Euler angles in this case would be obtained by converting the quaternions to Euler angles as defined in Section 5.2.

Often times however your reaction wheels are not pointed on the principal axis of inertia. In this case a Least Squares Regression model is needed. In this case the equation above is used to compute the desired torque to be placed on the satellite such that

$$\vec{M}_{desired} = -k_p(\epsilon_i - \epsilon_{desired}) - k_d(\omega_i - \omega_{desired}) \tag{310}$$

This equation is then equated to the equation for torque placed on the satellite where the angular accelerations are placed into a vector.

$$\vec{M}_{desired} = \vec{M}_R = \sum_{i=1}^{NR} I_{Ri}^B \alpha_{Ri} \hat{n}_{Ri} = \begin{bmatrix} I_{R1}^B \hat{n}_{R1} & \cdots & I_{RNR}^B \hat{n}_{RNR} \end{bmatrix} \left\{ \begin{array}{c} \alpha_1 \\ \vdots \\ \alpha_{NR} \end{array} \right\} = \mathbf{J}\vec{\alpha} \tag{311}$$

Since $\mathbf{J}$ is a $3 \times N_{RW}$ matrix its impossible to simply invert the matrix and solve for the vector of angular accelerations $\vec{\alpha}$. In this case there are an infinite number of solutions. As such a minimization routine is required where the solution found also happens to be the lowest amount of angular acceleration. In this case, Lagrange's method was used to find the vector of angular accelerations [33].

$$\vec{\alpha} = \mathbf{J}^T \left(\mathbf{J}\mathbf{J}^T \right)^{-1} \vec{M}_{desired} \tag{312}$$

14.1.3 Reaction Control Thrusters

The control law for the thrusters is a bit complex if the location of thrusters is not know apriori. If the location *is* known then simple PID control laws can be generated by applying pure couples to the correct thrusters that activate the correct axes. If the location *is not* known then the following derivation will suffice. There are N_P thrusters and only 3 degrees of freedom that need to be controlled; thus, the system is an overactuated system. Using equation 86, the equation can be written in matrix form as given by the equation below where $\vec{M}_p$ is replaced by $\vec{M}_{desired}$. The equation for $\vec{M}_{desired}$ is generated using a similar PD control law as the reaction wheels.

$$\vec{M}_{desired} = p[\mathbf{S}(\vec{r}_{P1})\hat{n}_{P1} \quad \mathbf{S}(\vec{r}_{P2})\hat{n}_{P2} \quad \cdots \quad \mathbf{S}(\vec{r}_{PN_p})\hat{n}_{PN_p}]\vec{\sigma} = \mathbf{M}\vec{\sigma} \tag{313}$$

Since $\mathbf{M}$ is a $3 \times N_P$ matrix its impossible to simply invert the matrix and solve for the vector of pulses $\vec{\sigma}$. In this case there are an infinite number of solutions. As such a minimization routine is required where the solution found also happens to be the least amount of pulses. In this case, Lagrange's method was used to find the vector of pulses [33].

$$\vec{\sigma} = \mathbf{M}^T \left(\mathbf{M}\mathbf{M}^T \right)^{-1} \vec{M}_{desired} \tag{314}$$

Note that a similar equation can be derived for $\vec{F}_{desired}$. The solution to the equation above results in values of σ that are bigger than 1 and sometimes negative. If a value in this vector is bigger than 0 the value is set to 1 and if the value is negative the value is set to 0. Thus, the solution does not yield an exact solution but it does allow for flexibility in the number of thrusters and their respective orientations. Sizing of the thrusters depends on many independent variables including the thrust T and the I_{sp}. Using the I_{sp} the exit velocity of the thruster can be obtained by using the equation below

$$v_e = I_{sp}g_0 \tag{315}$$

where g_0 is the gravitational acceleration of the Earth at sea-level. Then the mass flow rate of the thruster can be obtained using the equation below.

$$\dot{m}_P = T/v_e \tag{316}$$

Using this mass flow rate total propellant mass required can be computed assuming a certain duty cycle.

14.1.4 Cross Products of Inertia Control

An interesting form of control is to take advantage of momentum dumping. Looking at the equation for angular acceleration again (Eqn 63) this equation can be simplified for certain cases. For example, if the roll rate of the satellite is set to be non-zero while the pitch rate and yaw rates are set to zero it is easy to see that if the inertia is diagonal the derivative of angular velocity is zero. However, if the cross products of inertia are given by the matrix below

$$\mathbf{I}_s = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \tag{317}$$

the derivative of angular velocity becomes

$$\dot{\vec{\omega}}_{B/I} = I_S^{-1} \left(\left\{ \begin{array}{c} 0 \\ -I_{xz}p^2 \\ I_{xy}p^2 \end{array} \right\} \right) \tag{318}$$

again assuming the roll rate is non zero and the pitch rate is zero. This result shows that momentum can be transferred to different axes provided the cross products of inertia are non-zero.

14.2 Inner and Outer Loop Control of an Aircraft

14.2.1 Pitch Control

For a conventional PID controller of an aircraft, the rudder, elevator and aileron commands are set to

$$\begin{aligned} \delta_r &= -K_v v \\ \delta_e &= K_p(\theta - \theta_c) + K_d \dot{\theta} \\ \delta_a &= K_p(\phi - \phi_c) + K_d \dot{\phi} \end{aligned} \tag{319}$$

The Euler angle commands ϕ_c and θ_c are set using the following relationships:

$$\begin{aligned} \phi_c &= K_p(\psi - \psi_c) + K_d \dot{\psi} \\ \theta_c &= K_p(z - z_c) + K_d \dot{z} + K_I \int z - z_c dt \end{aligned} \tag{320}$$

The control scheme defined above is a conventional inner loop-outer loop control of a fixed wing aircraft using a PID tracking controller.

14.2.2 Velocity Control

14.2.3 Altitude Control

14.2.4 Roll Angle Control

14.2.5 Heading Angle Control of a Car

14.2.6 Heading Angle Control of an Airplane

14.2.7 Waypoint Control of a Car

14.2.8 Waypoint Control of an Airplane

15 Nonlinear Control Techniques

15.1 Lyapunov Control

15.2 Sliding Mode Control

15.3 Adaptive Control

16 Project Based Engineering and the Systems Engineering Life Cycle

Systems engineering is a branch of engineering used to help in the design of complex systems. The general V&V diagram for a systems life cycle can be seen below.
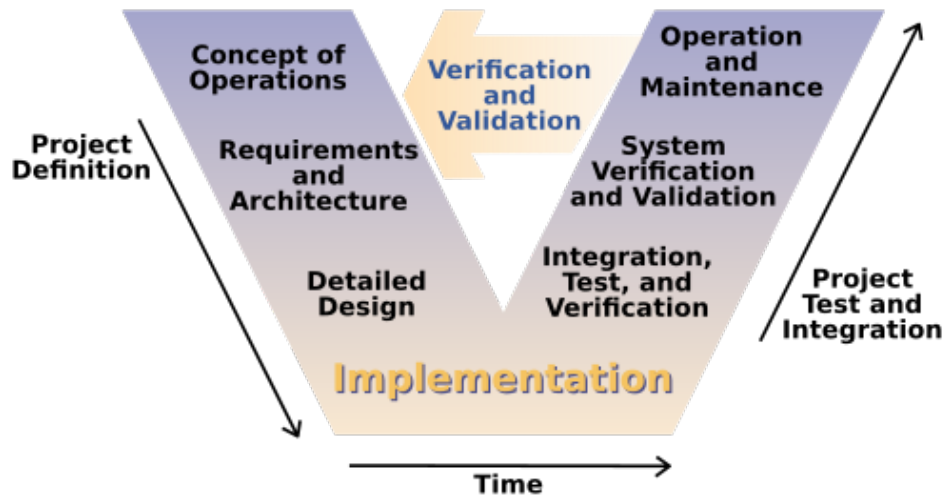

Figure 25: Verification and Validation Diagram for a Systems Engineering Life Cycle [44]

The left side of the V&V diagram highlights all of the required design while the right side details the build. For undergraduate engineering courses the Design, Build, Test (DBT) framework is often used and has been used at the University of South Alabama for many years with much success. The systems engineering life cycle is a more detailed version of the DBT framework. The systems engineering life cycle is used in many industries including aerospace, automotive, and software development. At the undergraduate level the following steps are used for a standard 16 week curriculum.

1. **Week 1:** *Title Page and Group Development* - Students are given an explanation of the project scope, rubric are asked to split into teams to create a title page and team name. In aircraft design the project is to design, build and test a radio controlled aircraft. In spacecraft design the project is to design, build and test a model rocket. In Instrumentation and Experimental Methods the project is to design, build and test a data acquisition system with at least 3 electrical components not counting the CPU, flash memory, or power supply. This also includes mechanical subsystems so that students practice both hardware and software integration. The teams are typically 1-3 students each depending on class size.
2. **Week 2:** *Concept of Operations (ConOps)* - Students are given an explanation of the ConOps and asked to create a ConOps for their project as well as include a list of requirements which is based on the project scope given by the instructor as well as any additional requirements the students feel are necessary.
3. **Week 3:** *Conceptual Design* - Students are asked to create a hand sketch of their project as well as a brief list of materials they will need to complete the project. This also includes a list of subsystems and components that will be needed to complete the project which requires the students to create a Block Definition Diagram (BDD) as well as an Internal Block Diagram (IBD).
4. **Week 4:** *Preliminary Design* - For the preliminary design phase the task is different depending on the course. In Aircraft Design the students are asked to perform a feasibility analysis and look at overall wingspan, chord and an initial weight estimate. In spacecraft design the students need to look at creating rockets in OpenRockt and perform a preliminary analysis of the rocket's stability. In Instrumentation and Experimental Methods the students need to create more detailed BDDs and IBDs for the mechanical, electrical and software subsystems as well as create an initial interface diagram (ID) for the project.
5. **Weeks 5-7:** *Detailed Design* - The detailed design phase is broken into 3 separate weeks. For aircraft design the students are asked to design the aerodynamic surfaces (wing and tail), the airframe and finally the propulsion system. For spacecraft design the students need to design the structure, propulsion system and perform a detailed trajectory analyis of their rocket. For Instrumentation and Experimental Methods the students need to design the mechanical, electrical and software subsystems creating schematics, wiring

diagrams and a list of all software needed to run their system. For every course the students also need to create a detailed bill of materials with prices, weights and links. At this point the system is ready to be built.

6. **Week 8:** *Build Plan, Purchase materials + Break* - In the Fall semester, Fall break typically lands on Week 8 and in the spring Week 8 is typically spring break. As such Students are asked to create a build plan which is not something to be turned in but just something helpful to keep students on track as well as purchase all parts necessary while they are on holiday. If parts are purchased the Friday before break that typically gives the retail stores over 5 business days or 9 calendar days to ship the parts to the students before they return to class. This way students can immediately start building and testing during Week 9 when they return

7. **Weeks 9-11:** *Build and Subsystem Testing* - The build process includes building everything component by component and testing each subsystem as it is built. This is important to ensure that each subsystem works properly independently before integrating everything together. For aircraft design the students will first build the airframe, then the propulsion system and finally the payload which is typically just a datalogger that measures acceleration and GPS location. For spacecraft design the students will first build the structure, then the propulsion system and finally the payload which for spacecraft design is a datalogger that measures acceleration and pressure. For Instrumentation and Experimental Methods the students will test each electrical component one by one and test the software for that specific component. The mechanical system is something to be built and integrated in Week 12.

8. **Weeks 12-13:** *System Integration and Testing* - The integration process includes integrating all subsystems together and testing the entire system as a whole. For aircraft design the students will integrate the airframe, propulsion system and payload and perform a ground test to ensure everything works properly. For spacecraft design the students will integrate the structure and payload and perform a series of tests to ensure everything works properly. This includes a shock cord test, a drop test, a datalogger test, a motor integration test and a nose cone test. For Instrumentation and Experimental Methods the students will integrate all electrical and software subsystems and perform a test without the mechanical subsystem in the way.

9. **Weeks 14-16:** *Final Testing and Validation* - The final testing and validation phase includes performing the final test of the entire system. For aircraft design the students will perform a flight test of their aircraft and recover the data from the datalogger. For spacecraft design the students will perform a launch of their rocket and recover the data from the datalogger. For Instrumentation and Experimental Methods the students will integrate the mechanical subsystem and perform a final test of the entire system. Note that 3 weeks are alloted for this phase to account for weather delays, part delays and any other unforeseen issues that may arise. Many times especially in aircraft design the system fails sometimes catastrophically as in a crash and the students need time to rebuild and retest before the final presentation.

10. **Week 17:** *Final Report and Presentation* - Although a semester is exactly 16 weeks, the final report and presentation is typically due the first week of finals which ends up being Week 17. The final report includes a summary of the entire project including all diagrams, analyses, test results and conclusions. The final presentation is a 10-15 minute presentation to the class and any invited guests which summarizes the entire project. Note that the final report and presentation is typically worth 30-40% of the overall grade so it is important that students take this seriously and put in their best effort. Furthermore, there are no exams or quizzes in these project based courses so the final report and presentation is the only way to demonstrate what the students have learned throughout the semester. This gives the students ownership of the project and motivates them to do their best work.

16.1   Block Definition Diagrams (BDDs)

16.2   Internal Block Diagrams (IBDs)

16.3   Interface Diagrams

Interface Diagrams (IDs) are important for many reasons. The biggest reason that these diagrams are created and maintained is to create visibility within each subsystem team in a systems engineering7 project. It is crucial that each member working on a project has a general or even an expert understanding of each subsystem. During the design process, it is important that each function of each subsystem's component is documented so that the current design selection is recorded and so each of the member's is familiar with each subsystem and all of its functionalities. The following image is the ID for the attitude control and maneuver electronics subsystem on the Gemini Spacecraft.
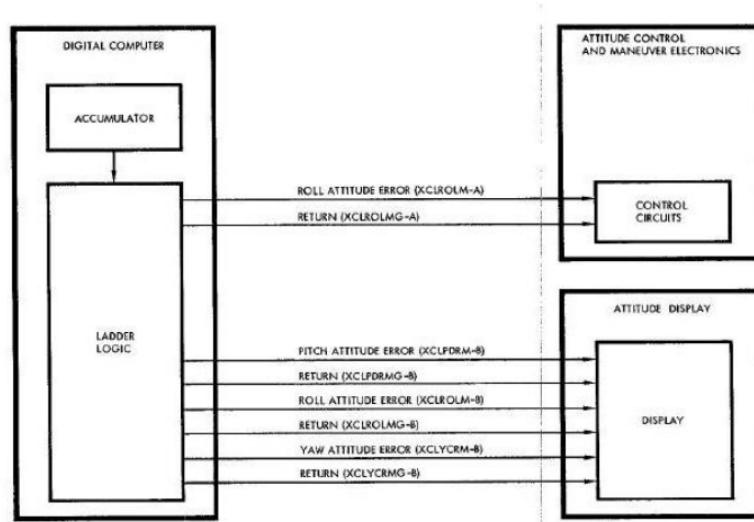
Figure 26: ACME Interface Diagram from the Gemini Spacecraft[45]

This is an example of an interface diagram for a small subsystem of a large systems engineering project. The spacecraft requires several subsystems to work together in the design and fabrication process. Each of these subsystems has their own ID as well. This is a good example to introduce you to what these diagrams are. The left side of the image is the digital computer which has an accumulator which flows to the ladder logic block. From the ladder logic flows several functions into both the attitude control and maneuver electronics and attitude display blocks. This ID is simple since there are only four components with many functions, however, many complicated systems can have extremely complex diagrams. The Figure below is an example of an ID for an entire spacecraft with all of the subsystems included in the diagram.
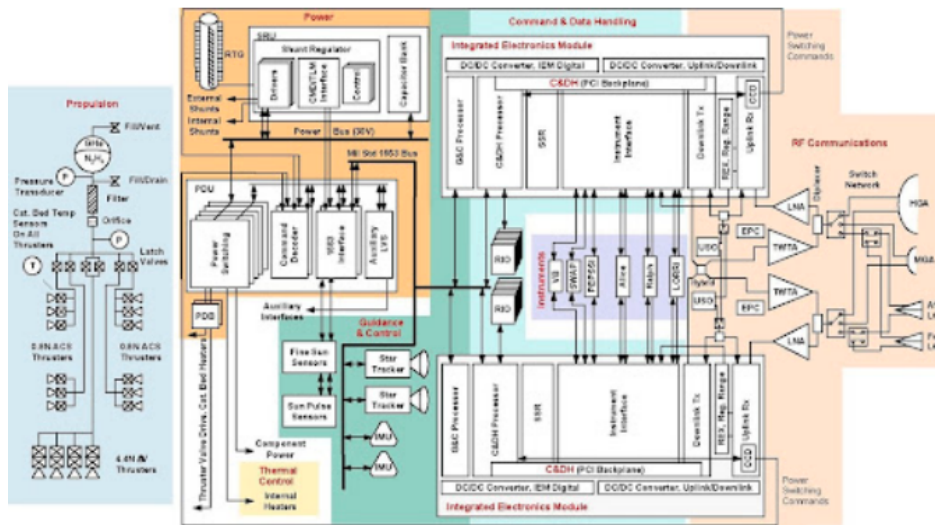
Figure 27: New Horizons Interface Diagram[46]

16.4   Activity Diagrams

Activity diagrams are similar to a flowchart. It is a tool for representing the sequence of Actions that describe the behavior of a Block or other structural element. The sequence of execution is defined using Control Flows. The Actions in an activity diagram can contain Input and Output Pins which act as buffers for items that flow from one action to another. Anything that can be produced, consumed, or conveyed by the system is considered to be an item. Physical materials, energy, power, data, and information are examples of items[47].

Activity diagrams are useful for engineering modeling. It conveys high-level functions and operations to the user. The main purpose of an activity diagram is to draw the activity or action flow of a system. Next, it is used to describe the sequence from one activity to another. Finally, an activity diagram describes the parallel, branched, and concurrent flow of the system [48].

Activity diagrams are important to the design process because they maintain coherence throughout the project. If a system is composed of multiple, intertwining subsystems, a person can simply follow the activity diagrams to understand how each component and subsystem interacts with one another. Activity diagrams also describe the data flow of a system and when or where that data is created, converted or used. Overall, activity diagrams are the roadmap of a system and describe the intricacies of how it operates.

As stated previously, activity diagrams denote how components and subsystems interact with each other. This is accomplished through the use of swimlanes which keep separate the individual activities and object flows of a component or subsystem. Swimlanes are the boxes which contain activities, however control and object flows can traverse swimlanes. Activities which have inputs and outputs will also have a pin attached to them. This small rectangular box denotes when matter, energy or data flow is created or destroyed.

The main purpose of activity diagrams is to show the control flow of the system. This control flow refers to the execution path of an activity. Control flow is indicated by control nodes for which there are seven different kinds. Each control node is described in the table below.

Control Node	Description	Appearance
Initial Node	The beginning of an activity sequence	Black circle
Activity Final Node	The end of an activity sequence	Black circle within a circle
Flow Final Node	The end of one branch of an activity sequence, but not the entire activity diagram.	Circle with an 'X' in it
Decision Node	The splitting point of an activity flow based on the outcome of a boolean operator such as 'isCommandValid ==True'. One flow in comes in, while two flows come out as branching paths.	Diamond
Merge Node	The merging point of two activity paths. Two flows come in, while one flow comes out.	Diamond
Fork Node	The distribution of an activity flow into multiple pathways. When one object or control token comes in, it is duplicated into multiple paths.	Black bar
Join Node	The joining point of multiple synchronized concurrent flows. Two concurrent flows come in, while one flow comes out.	Black bar

17   Guidance Navigation and Control Design for CubeSATs

Every vehicle must be able to maintain a specific position within its flight or orbit as well as point in a specific direction to achieve its mission. Mission goals have specific trajectories or orbits that must be reached and maintained, and this section is intended to explain how this is done from a guidance navigation and controls perspective. The basis for the guidance, navigation, and control (GNC) subsystem are attitude control, attitude estimation, and position estimation. Attitude describes which direction the vehicle is pointing in three-dimensional space. Attitude control is the act of controlling the orientation of the vehicle while attitude estimation is the process of determining the precise direction of the vehicle in order to perform attitude control.

Vehicle State estimation is a fundamental portion of GNC and requires the vehicle to determine it's orientation with respect to an inertial frame as well as its position from an inertial reference point. Some sensors are specific to the vehicle application but here this section will discuss some of the major state estimation sensors.

Position estimation primarily involves integration schemes or GPS in order to determine the position of the vehicle on the surface of the planet. While performing the mission, the vehicle will continually use this subsystem to document the position. The basis of these techniques follows an understanding of spaceflight mechanics and systems engineering. Thus, the following approach will help garner a better understanding of an academic approach to the GNC subsystem and what methods are used to convey each component of the subsystem.

The GNC subsystem is critical for the survival of the vehicle. It is the system that determines the vehicles orientation and position in space. Guidance is task of computing the desired trajectory and orientation of a vehicle. Guidance is completed by using components to determine any changes in position, altitude, or orientation to assist the vehicle in following its projected trajectory. Similar to guidance, navigation is the system's way of leading the vehicle in space and keeping it on its intended path. When people think of navigation they typically refer to Global Positioning Systems (GPS) in their car or on their phones. This same idea applies to aerospace systems as well. A GPS is a common device used for navigation on a vehicle and these components are discussed more further in this section. In order to have a successful flight and achieve the intended mission goal the vehicle needs to be stable and controlled in space. There are many different components different aerospace vehicles use to accomplish this. Satellites use reaction wheels and gimbaled thrusters to name a few while aircraft use aerodynamic surfaces.

17.1   Trajectory Analysis

The trajectory of a spacecraft is crucial to the GNC subsystem. The type of orbit that the spacecraft is designed for plays a large role in determining the components chosen for the attitude control, attitude estimation, and position estimation.

There are five main trajectories that spacecrafts typically follow:

1. LEO (Low Earth Orbit) - Things that must be taken into account in this trajectory are aerodynamics, Earth albedo, and space debris.
2. MEO (Medium Earth Orbit) - Van Allen Belts, which are radiation belts, need to be considered during the design process.
3. GEO (Geosynchronous Orbit) - Most communication satellites are in geostationary orbit since the satellite appears stationary and the communication dishes can be pointed directly at them and do not need to be maneuvered to follow the satellite.
4. HEO (High Earth Orbit or Highly Elliptical Orbit) - In this case you might be well outside of the GPS constellation, thus, constant position must be considered by an alternate means.
5. Deep Space - A deep space orbit is largely outside of any protective atmosphere, therefore radiation is a dominant concern. Also, well beyond the GPS constellation.
6. Propulsion, magnetorquers, and reaction wheels are the three main mechanisms for controlling the attitude of the spacecraft. Reaction wheels can be used universally, however, magnetorquers are only useful within a magnetic field. Thus, within Low Earth Orbit (LEO) magnetorquers are incredibly useful, however in highly elliptical orbit (HEO) they may not be nearly as useful. As for the third option, propulsion, it is really only useful outside of a magnetic field when magnetorquers are not ideal. A middle Earth orbit (MEO) is unique in that it can use everything that is offered in the LEO and HEO but it will continually travel through the Van Allen Belts. The Van Allen Belts are pockets of radiation trapped by the Earth's magnetic field which require additional shielding onboard the spacecraft to mitigate its effects.

In LEO, it is common to use a Global Positioning System (GPS) to estimate the position of the spacecraft. This component is most useful at lower altitudes because it is able to reach the GPS signal, however, as the altitude increases and the spacecraft moves farther away from the signal, it is no longer helpful. This is when an integration

scheme may be the best possibility for position estimation. Integration schemes, such as RK-11, are tedious but are commonly used in HEO. Since HEO's have an enormous apogee compared to perigee, up to a 35:1 ratio, the need for an alternate means to determine position is imperative[11][49].

Much like a HEO, any deep space trajectory will more than likely take the spacecraft outside of the GPS constellation. Therefore, in order to determine position the spacecraft must use NASA's deep space network, a numerical analysis, or another mathematical means. Communication with the spacecraft is another complex variable when in a deep space trajectory. Because of the large amount of latency between the ground and the spacecraft, the spacecraft will need to be able to autonomously operate without a constant uplink from the ground. Lastly, the spacecraft will need additional radiation shielding when in deep space to protect itself. This will further increase its mass and add further constraints to the GNC subsystem.

Attitude estimation can be determined by the use of magnetometers, however, similar to magnetorquers, they are only useful within a magnetic field. Thus, magnetometers are used when the spacecraft is set to be in LEO. There are several methods that can be used for attitude estimation, however, the only other one that greatly depends on the orbit are horizon sensors which must be used in LEO.

17.2 Spacecraft Environment

The environment of space can have an enormous impact on the spacecraft and the GNC subsystem. The environment is dependent on the trajectory of the spacecraft and at what altitude it is set to reach. The environment of space changes as the spacecraft travels farther from the earth and there can be negative repercussions depending on the components and materials chosen for the spacecraft.

There are four main environments that may affect a spacecraft:

1. Earth: Does not affect the design of the spacecraft but it is important to keep in mind that if components are not commercial off the shelf (COTS), they need to be built in a lab
2. Launch: It is crucial to research the vibration requirements before launching. Again, if it is not commercial off the shelf, a vibe test must be run to ensure the components survive.
3. LEO: This environment has many factors that must be considered including effects from the sun, thermal changes, Van Allen Belts which account for a large amount of radiation thus radiation shielding must be researched for the separate components.
4. Deep Space: There are some things such as meteorites that may need to be considered, however, there is not much that can be done. This environment should not heavily affect the design. Note that the spacecraft is going to undergo multiple types of disturbances as explained in the dynamic model section.

17.3 Spacecraft Attitude Control

Attitude control is the means by which the orientation of the spacecraft is maintained within the orbit. Throughout the mission, the spacecraft experiences disturbance torques that are caused by a variety of sources, such as aerodynamics, gravity gradient, and solar radiation pressure discussed previously. These torques impart momentum which rotate the system away from its desired orientation. The purpose of attitude control is to reject these disturbance torques while simultaneously pointing towards a desired orientation. Possible active mechanisms for attitude control are magnetorquers, reaction wheels, thrust vector control, reaction control system, and control moment gyros.

There are several components that can be used for the attitude control of a satellite. Selecting the proper component is dependent on the orbit and its apogee/perigee, the system's requirements, and the risks associated with the component. There are several steps involved with component selection. Typically the team will review the requirements and perform a preliminary risk analysis. This will generally help narrow down which components to use where. To determine the specific component, trade studies are conducted to compare the components from different manufacturers to determine the best one for the project at hand.

17.3.1 Magnetorquers

The first option for attitude control is a magnetorquer. A magnetorquer, or torque rod, is built from electromagnetic coils. These coils produce a magnetic field that interacts with another magnetic field, typically Earth's, which in turn produces a torque upon the satellite[50]. These components are used for changing the angular momentum of the satellite. In addition to attitude control, magnetorquers are also used for detumbling the satellite. Detumbling is a means of stabilizing the satellite after being inserted into the desired orbit[51]. The International Geomagnetic Reference Field (IGRF) is a software that reports the magnetic field of the earth by using latitude, longitude,

and altitude from a GPS. This magnetic field is then used to determine the proper magnetic field required by the magnetorquer to maneuver the satellite.

Figure 28: Example Magnetorquer[52]

An example of magnetorquer effectiveness is its ability to detumble during an orbit, and how many orbits it will take to completely detumble the spacecraft. First, the magnetorquer will have a magnetic moment value associated with the physical parameters of the rod μ. First the average torque is computed by taking the magnetic moment and multiplying it with the average field strength and the point (from IGRF model). Then, this average torque is multiplied by total time of the orbit T_p to determine the magnetorquer's momentum effectiveness over one orbit. Note that the torque from the magnetorquer's $\vec{M}_{MT}$ is substracted by the total disturbance torques $\vec{M}_D$ to obtain the total effectiveness of the magnetorquers. From this, it is possible to calculate the number of orbits it will take to detumble the spacecraft on an axis.

$$N_{orbits} = \frac{||\vec{H_S}||}{T_p(\vec{M}_{MT} - \vec{M}_D)} \tag{321}$$

The magnetorquer effectiveness boils down to the difference in the total moment that the magnetorquers can produce and the total disturbance momentum during the orbit. Then, the number of orbits that it will take for the spacecraft to detumble in an axis is a function of the magnetorquer effectiveness divided by the initial angular momentum in each axis.

17.3.2 Reaction Wheels

Reaction wheels are mechanical disks, or flywheels, that rotate the satellite during orbit. The reaction wheels take in electrical power and provide a torque to the satellite[53]. This electrical power is provided to a DC motor which then spins the reaction wheel. Because momentum must be conserved, angular momentum must also be conserved. Thus, by Newton's third law, when the reaction wheel rotates one direction the satellite rotates in the other. So, a minimum of one reaction wheel is placed on each axis to control the rotation in all directions. The primary role of a reaction wheel is to point the satellite without the use of any fuel.

Figure 29: Example Reaction Wheels[54]

The inertia storage for reaction wheels is a parameter that can be used to seek out the best reaction wheels for the spacecraft. Once the inertia storage is computed, one can search for reaction wheels with the proper amount of

inertia storage required. For example, if a reaction wheel with an inertia storage of 50 mNms is needed, one can go to Blue Canyon Technologies website to find the datasheets for their different reaction wheels. From there, it can be determined that the best fit would be the RWP050 reaction wheels[55].

Once an estimate of the inertia of the satellite is obtained, they can be used to compute the required inertia storage of the reaction wheels. The tip off rate for a spacecraft is the rate at which the spacecraft's angular velocity is altered due to its deployment. The following equation shows how momentum can be calculated due to tip off.

$$||\vec{H}_S|| = max(\mathbf{I}_S)\omega_{TOR}f \tag{322}$$

where ω_{TOR} is typically set at 10 deg/s and f the factor of safety is typically set to 2.

17.3.3 Thrust Vector Control (TVC)

When aerodynamic control surfaces are ineffective, TVC is a solution to maintain the vehicle's correct attitude for the thrust duration. It accomplishes this goal by gimballing (rotating) the thrust chamber or by redirecting the exhaust-gas flow to develop torque [56]. However, thrust vectoring only moves the vehicle in two directions. TVC is typically used to control pitch and yaw attitude on boosters and upper stages.
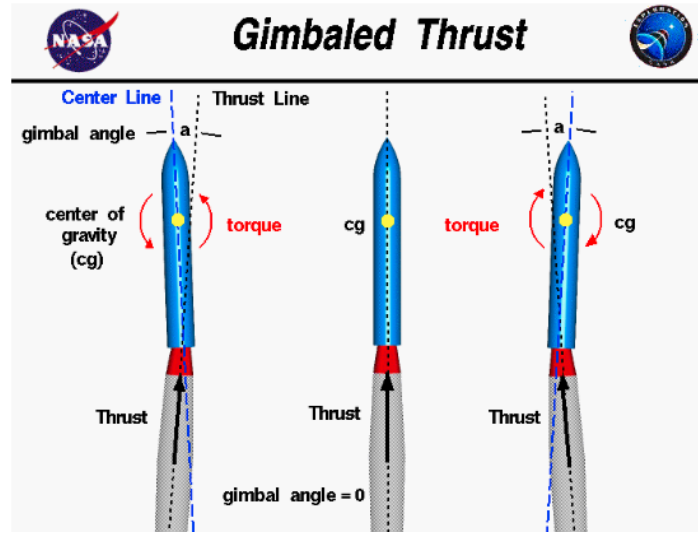
Figure 30: TVC Diagram[57]

17.3.4 Reaction Control System (RCS)

Reaction control systems are small thrusters. These thrusters alter the speed of the spacecraft's rotational or spinning motion. They are good for making quick turns and are used for getting to new orientations quickly. Reaction control thrusters are typically in "couples" or pairs of thrusters that together can spin the spacecraft without changing the lateral velocity. In addition, these thrusters usually work great with reaction wheels. When reaction wheels slow down, they produce a force. The RCS will oppose this force which allows the spacecraft to remain in the intended orientation[58].

Figure 31: Example RCS Thruster [59]

17.3.5 Control Moment Gyro

A CMG is a spinning rotor at a constant speed [25]. Similar to a reaction wheel, a CMG also has a spinning fly-wheel controlled by a brushless motor. However, the spin axis of a CMG can rotate with the h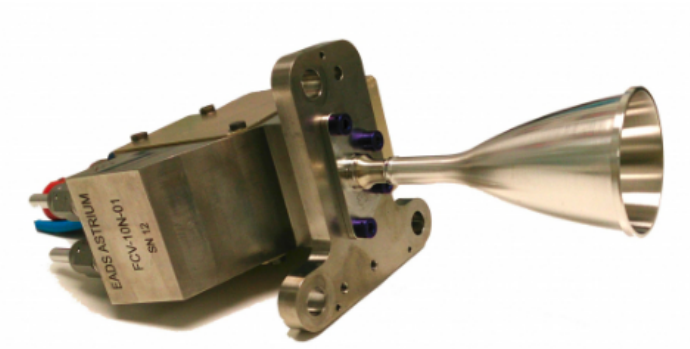elp of a second motor placed on a gimbal axis. If the angular momentum can only rotate in a fixed plane, it is known as a single gimbal control moment gyros (SGCMG). GMCS are usually more efficient and produce higher torque than reaction wheels as the size of the unit increases[60].

Figure 32: Example CMG [61]

17.4 Spacecraft Attitude Determination

Attitude determination is a fundamental portion of the ADACS board and requires the vehicle to determine it's orientation with respect to an inertial frame. The sections that follow detail the sensors and fundamental attitude determination algorithms derived thus far.

17.4.1 Sensor Overview

There are a multitude of sensors that are typically used on board small sats. Note that most satellites use a combination of these sensors rather than using all of them on one single satellite.

1. **Magnetometers:** Only used in LEO, they measure the magnetic field in the body frame $\vec{\beta}_B = [\beta_x, \beta_y, \beta_z]^T$.
2. **Rate Gyros:** These sensors measure the angular velocity of the spacecraft in the body frame p, q, r.
3. **Solar Sensors:** These sensors can be coarse analog sensors with an accuracy of 45 degrees or can be high precision digital sensors that have accuracy down to 1 degree. Sun senors return an azimuth v and declination δ angle which can be then translated into a vector in the body frame $\vec{S}_B$.
4. **Horizon Sensors:** Horizon sensors are typically used in LEO as they find the horizon of the Earth and use that for orientation information. These sensors also return an azimuth and declination angle that can be translated into a body frame vector $\vec{H}_B$.

5. **Startrackers:** Startrackers utilize a large aperture digital camera to photograph a starmap within the field of view of the lens. The photographed stars are then cross referenced with a starmap database and return the full quaternion vector.

Some issues arise with all of these sensors. For example, magnetometers must be activated when magnetorquers are turned off otherwise those artificial magnetic fields will pollute the data. Rate gyros are prone to drift while solar sensors can be quite inaccurate. Startrackers also run the risk of being blinded by the Sun and/or the Moon thus it is possible to design an attitude determination algorithm that utilizes the Sun's ephemeris data along with the Moon's ephemeris data in the event that the startracker is obscured by the Sun/Moon.

17.4.2 Inertial Measurement Unit

An inertial measurement unit (IMU) is a combination of three sensors. An accelerometer, rate gyro and magnetometer. A magnetometer is a device that measures the local magnetic field in the body frame $\hat{\beta}_B = [\hat{\beta}_x, \hat{\beta}_y, \hat{\beta}_z]^T$ [62]. Note that the ˆ implies a measurement rather than the truth signal. Measurements from sensors are prone to bias, drift, scale factor, misalignment, noise and other sources of error that must be accounted for.

Figure 33: Example Magnetometer [63]

A rate gyroscope, commonly referred to as a rate gyro measures the angular velocity also in the body frame $\hat{\omega}_{B/I} = [\hat{g}_x, \hat{g}_y, \hat{g}_z]^T$.

Figure 34: Example IMU with Integrated Rate Gyro [64]

Accelerometers are sensors used to measure acceleration at a point P on a rigid body $\hat{a}_{B/I} = [\hat{a}_x, \hat{a}_y, \hat{a}_z]^T$. For simplicity however, it is assumed that point P on the rigid body is the center of mass point C therefore the accelerometer is measuring the acceleration of the body itself in the body frame with respect to an inertial frame B/I.
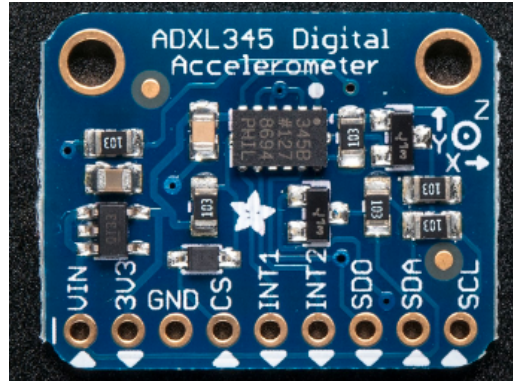
Figure 35: Example Accelerometer [65]

As mentioned before, the IMU consists of 3 sensors all returning 3 measurments. This results in 9 scalar quantities being returned from this sensor which is where the term 9DOF gets it origin. In reality DOF means Degrees of Freedom which is contrary to the standard 6DOF simulation models explained above. However, the sensor community chooses to coin the term 9DOF to highlight the 9 different scalar values returned from IMUs. It is possible to obtain a 10DOF sensor which also returns pressure or temperature data.

17.4.3 StarTracker

A star tracker is another critical component of attitude control. This device is a camera that points out from one face of the satellite. The camera is able to detect the stars in its field of view (FOV) and determine the location of the satellite based on this. Specifically, the star tracker images the "starscape" to identify the known planets and stars, and compares the imaged locations to the known locations using the SPICE Toolkit[66]. NASA Jet Propulsion Laboratory (JPL) has produced the SPICE Toolkit for knowledge of the location of the planetary bodies and stars and specific times[67].
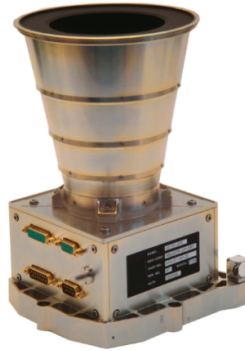
Figure 36: Example StarTracker [68]

17.4.4 Sun Sensors

A sun sensor is a device that senses the sun's direction to measure the position of the sun with respect to the sensor's position. There are three types of sun sensors. The first one is an analog sensor whose output signal is a continuous function of the Sun angle. Next is a sun presence sensor which provides a constant output signal when it senses the sunlight. The last one is a digital sensor that produces encoded discrete output that is measured by the sun angle function.

Sun sensors work based on the entry of light into a thin slit on top of a rectangular chamber with the bottom part lined with a group of light-sensitive cells. The chamber casts an image of a thin line on the chamber bottom. The cells at the bottom of the rectangular chamber measure the distance of the image and the refraction angle by using the chamber height. The cells convert the incoming photons into electrons and develop voltages which are

converted into digital signals. the direction of the sun can be computed when the sensors are perpendicular to each other [69].

Figure 37: Example Sun Sensor [70]

17.4.5 Horizon Sensor

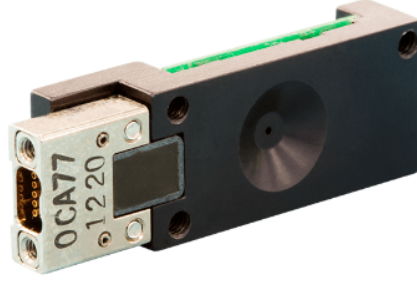The horizon sensor is a sensor used by spacecraft in LEO to determine the location of the Earth's horizon. There are two main types of horizon sensors: statics and scanning. A static horizon sensor is able to detect the infrared radiation emitting from the Earth's surface to locate the horizon of Earth. Scanning horizon sensors are much more complicated and use a system consisting of a spinning mirror to direct light onto a bolometer. This bolometer can sense when the infrared signal is present or lost. So, as the mirror rotates and reflects infrared light into the bolometer, the bolometer is able to detect when the signal is present or lost to determine the edge of the horizon[71].
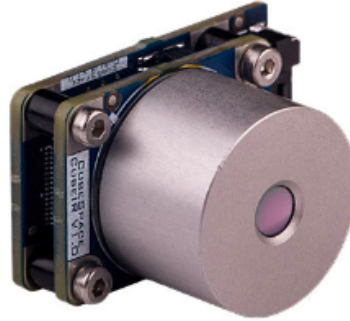
Figure 38: Example Horizon Sensor [72]

17.4.6 Deep Space

As explained earlier, in deep space it is possible to obtain a vector to the Moon to be used in the attitude determination algorithm. The Moon sensor would give a vector to the Moon in the body frame $\vec{M}_B$ while an inertial vector would be needed $\vec{M}_I$. This inertial Moon vector could be obtained via the Moon's ephemeris data which could be loaded onto the satellite's processor and use the orbital elements of the Moon to determine its position relative to the Earth. However, the Moon's ephemeris data would more than likely give the Moon's position relative to the Earth ($\vec{r}_{\oplus \to \mathbb{C}}$). The vector $\vec{M}_I$ would then be given by

$$\vec{M}_I = \vec{r}_{\oplus \to \mathbb{C}} - \vec{r}_B \tag{323}$$

where $\vec{r}_B$ is the satellite's position relative to the Earth. Note however that the position of the satellite relative to the Earth would need to be obtained via the Deep Space Network (DSN) and a combination of state estimation by integrating the orbital equations. The reference paper [12] is a great paper that details all the different kinds of sensors and their algorithms. This section will eventually be supplemented by the material in that reference paper.

17.5 Position Estimation

Position estimation is for determining the placement or location of the spacecraft in orbit. This can often be confused with attitude estimation, however, this can be clarified with an analogy. A human's posture is their attitude while their location, or place in reference to something else, is their position. This analogy can be transferred to spacecraft attitude and position. Position estimation is how mission control determines the location of the spacecraft in space. It is vital for the spacecraft to know its position in order for the on board sensors to work as intended. Position estimation is similar to attitude control in the sense that the orbit plays a large role in determining the proper component. The major options for attitude estimation are Global Positioning Systems and the Ground Station Network.

The Global Positioning System (GPS) was developed in order to allow accurate determination of geographical locations by military and civil users. It works by using satellites in Earth's orbit to transmit data which makes it possible to measure the distance between the satellites and the operator. This form of signal communication is incredibly accurate and used heavily for attitude estimation. Up to 30 GPS satellites are currently in orbit, mostly in MEO, at altitudes around 20,000 km. There will be between four and eight of them above any site on the Earth at any time[73]. These satellites continuously emit coded high-frequency radio signals which may be received by special GPS receivers. These signals contain information about the exact orbits of the satellites and the time of atomic clocks onboard. When signals from three or more satellites are received, the GPS receiver will compute the best possible location of the user by triangulation. Much like when on Earth, a GPS can be used for navigation in space. The GPS receiver on board the satellite will also receive its longitude, latitude, and altitude as long as it is within the GPS constellation.

Figure 39: Example GPS Receiver [70]

17.6 Trade Studies

Tradeoffs and considerations are analysis tools used to compare different components. It is not always obvious which component is best for a subsystem function. As shown in the previous sections, there are several options for attitude control, attitude estimation, and position estimation. It may not stand out at first glance which components are the best, so it is up to the team to perform a proper tradeoff analysis, a.k.a a trade study. What this typically looks like is a comparison of the different parameters of the components. For example, if the GNC team decides to look at an integrated system for attitude estimation, a trade study can be conducted to compare the mass, power, volume, and momentum storage for different Blue Canyon Technologies XACT systems [55]. It is important to keep this trade study organized and regularly updated, so it is recommended that a table is created to compare these parameters.

The name "tradeoff" comes from the fact that for a system it is typical that one feature must be compromised for the other. For example, to get more power, it is probably reasonable to assume that volume will be compromised because the size of the component will need to increase to fit a larger power supply. Another example may be that in order to increase the volume of a component to fit a larger payload, the mass will be compromised and may increase. These compromises especially become issues when they interfere with the subsystem requirements.

17.7   Risks

Risk assessments are arguably the most important task for the GNC team. It is crucial that the entire system is taken into consideration and assessed for possible failures because only one part needs to fail to cripple the whole mission. This includes non physical issues such as increased lead times on parts for example. The purpose of a risk assessment is to identify all possible risks, their severity and likelihood and to come up with mitigation strategies for those risks.

The purpose of risk assessments is to determine mitigation strategies so that the identified risks may be prevented. These mitigations are discussed amongst the team and heavily considered while designing the subsystem. Mitigation strategies can range anywhere from simply creating a preflight checklist to diverting control from one component to another in the event of a failure. These strategies are the main purpose of creating a risk analysis in the first place, so careful deliberation should be used when creating them.

The best way to conduct a risk assessment is by creating a risk table for the GNC subsystem. A risk table will list out the possible risks determined by the team and their subsequent mitigation strategies. It will also list out the likelihood and severity of each individual risk as decided by the team. These can be determined in any way deemed reasonable by the team, but is most often used on a scale of one to five for each category. These values can then be input into a risk criticality matrix which shows which risks are of the greatest threat level.

It is important that risk statements are written in a specific way to avoid any confusion or miscommunication between the different subteams. A risk statement consists of four parts: the condition, departure, asset and consequence. The condition is a single phrase that describes the current key fact-based situation or environment that is causing concern, doubt, anxiety, or uneasiness. The departure describes a possible change from the design or plan. It is an undesired event that is made credible or more likely as a result of the condition. The asset is an element of the system or plan. It represents the primary resource that is affected by the individual risk. The consequence is a single phrase that describes the foreseeable, credible negative impact(s) to meet performance requirements. A proper risk statement should utilize these four parts in a format similar to the following: "Given that [CONDITION], there is a possibility of [DEPARTURE] adversely impacting [ASSET], which can result in [CONSEQUENCE]"

17.8   Conclusions

The GNC subsystem's purpose is to know where the spacecraft is, where it needs to go, and determine how it will get there. These requirements are determined by the overall mission goal of the spacecraft. The system constraints are divided into three main categories: the attitude control, attitude estimation, and position estimation. The attitude control is how the spacecraft will maintain its current orientation during orbit. Magnetorquers, reaction wheels, thrust vector control, reaction control systems, and control moment gyros are all common components used to counteract the disturbance torques and maintain the desired orientation. Attitude estimation is how the spacecraft estimates its current orientation. This is commonly achieved by star trackers, sun sensors, magnetometers, rate gyros, accelerometers, and horizon sensors. Lastly, the position estimation is knowing the precise location of the satellite during its orbit. GPS and the GSN are the two most common methods for determining the satellites position.

Not only are the components important but one must also consider the trade-offs and risks associated with the mission. Risks must be assessed and properly mitigated in order to ensure the safety and success of the spacecraft. This is done by creating risk tables using if-then terminology.

18   Radio Controlled Aircraft Design

Many principles for large aircraft design can be applied to smaller radio controlled aircraft but understand that many of the aerodynamic principles are not quite well defined for slow and small aircraft. These aircraft have low Reynolds number which often exhibits odd phenomena. For example, airfoil selection is really not so much a design point other than ease of manufacturing rather than maximizing lift coefficient. Remember that Reynolds number can be defined from the equation below where ρ is the density at sea-level (1.225 kg/m^3 in SI units), V is the desired flight speed, $\bar{c}$ is the mean aerodynamic chord and μ_∞ is the viscosity of air which in SI units is 1.81e-5 $kg/(m-s)$.

$$Re = \frac{\rho V \bar{c}}{\mu_\infty} \tag{324}$$

You can tell that flying a small aircraft ($\bar{c}$) and flying slow (V) results in a low Reynolds number. Either way the procedure below has produced some great aircraft and the tools you'll learn along the way will help you in your future aerospace engineering career. If you're not an engineer then this text will at least give you an appreciation for what goes into aircraft design. If you'd much rather watch youtube videos than read this document, feel free to watch my Youtube playlist on radio controlled aircraft design[74]

18.1   Vehicle Type Selection and Requirements

In the very beginning of your design you need to decide on the type of aircraft you want to build. Designing a glider versus an aerobatic airplane will result in vastly different engineering design decisions. For example, a glider is going to have very long slender wings while an aerobatic airplane is going to have somewhat shorter wings with large control surfaces. I suggest you select from the following types of aircraft and then move on: Gliders, Trainers, Sport Aerobatic, Racers. If you'd like to build a scale aircraft there isn't much to design since the shape of the aircraft is pretty much built. If you do go with a scale aircraft this textbook isn't really for you since you aren't really building a scratch build aircraft. You're more just copying someone else's design. In that case you may as well just buy a kit or watch some videos on balsa wood construction and an overview of all the electronics required for RC aircraft flight.

If you selected one of the other styles you're ready to move onto to the next stage which is requirements. There is so much literature on Systems Engineering, top level requirements, functional requirements and derived requirements. The bottom line is you need to determine what you want your aircraft to do. Do you want it to fly straight up, upside down? Do you want the aircraft to be hand launched? Land on a runway? Determine what you want the aircraft to do and create a bulleted list of those requirements. Throughout the design you can refer to these requirements and make sure you are satisfying these requirements. If this is your first build then you may just have one requirement and that is to take off and land without crashing. But think a bit deeper. Do you want to turn the vehicle? Do you want full channel control for roll, pitch and yaw or just yaw control? Do you want landing gear? What sort of flying characteristics do you want? Be as specific as possible here.

18.2   Initial Design - Hand Sketch and Aspect Ratio

Once you have an idea of what the aircraft type is and what the requirements are it's time to hand sketch your aircraft. Try and use engineering paper, french curves, a ruler and a compass. Make this hand sketch look nice so you can use it in your future design. Draw your sketch to scale. You might be wondering, "how do I draw the aircraft without knowing what my wing loading or thrust to weight ratio is?". The answer comes from my late aircraft design professor Dr. Mikolowski (RIP). He would always say "If it looks good, it flies good". After designing so many aircraft and seeing so many scratch builds from my students I can honestly say that this is true 100%. If you're reading this now it means that there is already over 100 years of aircraft technology on the internet for you to research and see what other aircraft look like. Make your aircraft look like that but make sure it fits into your aircraft type and make sure it satisfies your requirements from above. If one of your requirements was to hand launch, then make sure your drawing reflects a vehicle without landing gear and a place to grab the aircraft. If you wanted full channel support make sure to include all the control surfaces. Think about where you want the propulsion system to go and how you're going to access the electronics before you fly. Think about what you want the wing to look like. Make it as big as you think it needs to fly. Use your intuition. This is an art. So much of engineering is an art.

Once your aircraft sketch is complete (make sure to do a front view, side view and top view of your aircraft), it's time to take down some wing characteristics. This is why you need to draw your sketch to scale. Measure the length of the wing (wingspan b) and the chord at the root (c_r) and the tip (c_t). Compute the area of the wing (S)

using the area of a trapezoid or rectangle depending on the shape of your wing. Once you have the wingspan and area you can compute the aspect ratio of your aircraft.

$$AR = \frac{b^2}{S} \tag{325}$$

The general rule is that the larger the aspect ratio the more aerodynamically efficient your aircraft will be. This is why gliders have very long and slender wings. At the same time, high aspect ratio wings suffer from larger bending moments and can flex considerably in flight. Finally, it's important to compute the mean aerodynamic chord of the wing. This is basically the average chord of your wing. If you create a rectangular wing your mean aerodynamic chord is just the chord of the wing since it's constant. If not you'll need to integrate over the length of the wing using the formula below where y = 0 is the centerline of the vehicle and y=b/2 is the wingtip on the right side[75]. The parameter $c(y)$ is the chord length as a function of y.

$$\bar{c} = \frac{2}{S} \int_0^{b/2} c(y)^2 dy \tag{326}$$

18.3   Weight Estimate - Tabular Approach

Once you have an idea of the overall shape it's time to estimate how heavy the aircraft will be. First think about the fundamental components of an aircraft. If you're not familiar with any of the components below, just type the item into Google and you'll find numerous articles and Youtube Videos about each component.

1. ESC - Electronic Speed Controller
2. Battery - Assume for now that you'll be using a 1500mAh 3S Battery unless you're building a micro aircraft in which case you might end up using a 600 mAh 2S or even a 300 mAh 1S. Think about the size of your aircraft. You will do more sophisticated battery design in the future.
3. Motor and Propeller - Again select something that is in the ballpark of the aircraft you're building. You'll do a redesign later.
4. Servos
5. Receiver
6. Control Linkages and Servo Horns
7. Fuselage
8. Main Wing
9. Tail both Horizontal and Vertical
10. Payload

For each of the components above you need to estimate the weight of these components. The only way to do that is to either look up the weight of similar aircraft to the one you're designing or find components that you think will work for your aircraft and add up all of the weights. The most difficult part is going to be estimating the empty weight of the aircraft which is just the structure of the aircraft. For these estimates you need to decide what materials you plan on using. Is your aircraft made out of foam, balsa, carbon fiber or some type of combination. Perform an initial material selection and then use that to estimate your weight. Create a table in a spreadsheet type program or a numerical computer program so that you can go back and change weights as your design progresses. Use the spreadsheet or numerical program to compute the total weight of your aircraft. This is your maximum takeoff weight.

18.4   Airfoil Selection and 2D and 3D Lift

The section 8.4 details the aerodynamic forces on the aircraft. In this stage of design we will only be looking at the lift equation.

$$L = \frac{1}{2}\rho V^2 S C_L \tag{327}$$

The coefficient of lift C_L is the lift coefficient of the aircraft. This coefficient is a function of Reynolds number, angle of attack, airfoil shape and wing shape. Recall that the angle of attack is the angle between the zero lift line of the airfoil and the free stream air. Breaking the velocity vector into components yields the equation below where w is the airspeed along the z-axis of the vehicle and u is the velocity along the x-axis.

$$\alpha = tan^{-1}\left(\frac{w}{u}\right) \tag{328}$$

Using the angle of attack as a parameter, the lift coefficient can be expanded to the following:

$$C_L = C_{L\alpha}(\alpha - \alpha_0) \tag{329}$$

The parameter α_0 is the angle of attack that results in zero lift. This is a function of the airfoil shape. The parameter $C_{L\alpha}$ is the lift curve slope and is a function of airfoil shape, wing shape and Reynolds number. In order to remove, wing shape from the design, the aspect ratio is used to convert the wing lift to a sectional airfoil coefficient.

$$C_{L\alpha} = \frac{C_{l\alpha}}{1 + \frac{C_{l\alpha}}{\pi e AR}} \tag{330}$$

The parameter e is an efficiency parameter which is often assumed to be 80-90%. The coefficient, $C_{l\alpha}$ is the airfoil lift curve slope which is different than the wing lift curve slope. The wing lift curve slope will always be smaller than the airfoil lift curve slope. This is because of an effect called wing tip vortices. Wing tip vortices are an aerodynamic effect where high pressure from the bottom of the wing moves around the wingtips to the area of low pressure. The only way to mitigate these vortices is by installing winglets or increasing the aspect ratio. Note that winglets increase drag and weight which is why you don't typically see them on RC aircraft. The coefficient $C_{l\alpha}$ is then simply a function of airfoil shape and Reynolds number. This is what is often referred to as 2D lift. It is the lift of airfoils which are two dimensional rather than 3D lift which is over an entire wing. It is at this point that airfoil selection and design can be used. The website `airfoiltools.com` is a great resource for plotting lift curve slopes of various airfoil shapes as a function of Reynolds number. I also have a great Youtube video on how to use XFLR5 (pronounced X-Flyer Five)[76]. A more comprehensive guide on XFLR5 can be found in [77] while the software itself can be downloaded in [78]. The basics of airfoil selection then break down into the following process.

First, using your max takeoff weight and cruise flight speed, compute the lift coefficient required in cruise. This assumes that lift equals weight.

$$C_L = \frac{2W}{\rho V^2 S} \tag{331}$$

Then using your flight speed and chord length, compute the Reynolds number of your aircraft. Use this Reynolds number and airfoiltools or XFLR5 to compute the sectional lift characteristics of various airfoils. Airfoil selection can be as complicated as you make it but you're looking for the highest lift to drag ratio airfoil. Another simple way is to just select the airfoil with the highest sectional lift coefficient. Another item to consider is manufacturing. Some airfoils may be feasible for full sized aircraft but not for RC flyers. My recommendation is to build an aircraft with a Clark-Y airfoil. They are easy to cut with balsa or shape with foam and have good lift to drag characteristics. Note that this airfoil is cambered. If you want to fly upside down I suggest you use a symmetric airfoil like a NACA 0012 or NACA 0014 is you need a bit more thickness to fit a wing spar through the airfoil. Once you have your airfoil selected, use the lift curve slope to estimate $C_{l\alpha}$ by fitting a linear trend line to the portion of the graph before stall. Also make sure to take note of what the zero lift angle of attack is. You can then compute the wing lift curve slope by using equation 330. Once you have $C_{L\alpha}$, you can compute the angle of attack needed for cruise. Make sure to convert α_0 to radians before you use the equation below.

$$\alpha = \frac{C_L}{C_{L\alpha}} + \alpha_0 \tag{332}$$

The resulting answer will be in radians and needs to be converted to degrees to make sure you are not close to stall. Typically in cruise, the angle of attack is only a few degrees with perhaps 8 degrees on the high end. If the answer you receive is higher than that it can mean a few things which may involve a redesign.

1. The simplest way to get more lift is to fly faster. The problem is you need a bigger motor which will increase your weight which will require more lift and more angle of attack. Flying faster will also change your Reynolds number.
2. You can increase the aspect ratio of your wing which will make your aircraft more efficient. The problem is that will increase the bending moment at the root creating the need for stronger materials at the root which also increases weight. This will also change your mean aerodynamic chord which will change your Reynolds number.
3. You can increase the area of the wing. This will also increase drag and weight but if the material you are using has a high lift to weight ratio then adding more wing area might be a good option. Depending on how you change the aircraft wing shape, the aspect ratio and/or the mean aerodynamic chord might change meaning you'll have to recompute the wing lift curve slope as well as the Reynolds number.

4. If you are using a symmetric airfoil it's possible you could forgo flying upside down and switch to a cambered airfoil which has more lift.

Regardless of what you do make sure you angle of attack in cruise is low which will reduce drag. It's optimal to fly at the aircraft's highest lift to drag ratio but radio controlled aircraft don't typically do that. It is also important to compute the stall speed of your aircraft. You'd like this value to be as small as possible.

$$V_{stall} = \sqrt{\frac{2W}{\rho S C_{Lmax}}} \tag{333}$$

In this case C_{Lmax} is the maximum lift coefficient your aircraft can obtain before stalling. If the stall speed is too high for your design go back and redesign your vehicle. Once you have redesigned your vehicle to fit within tolerable limits it's time to look at some aircraft performance characteristics which include the W/S (wing loading) and the T/W (thrust to weight ratio).

18.5 Wing Loading and Thrust to Weight Ratio

The wing loading is defined as the weight of the aircraft over the main wing area (W/S). Intuitively though, it is the amount of lift required per square foot of wing area for your aircraft. If the wing loading is high it means you have a heavy aircraft with small wings which means you either need very high lift creating devices like flaps and cambered airfoils or the aircraft needs to fly very fast. You can imagine that warbirds and racers have higher wing loading then say a glider which has very low wing loading. In this case the aircraft can fly slow because the aircraft is light with larger wings. At this stage of the design you already know the maximum weight of the aircraft and the main wing area so it's simple to calculate. For larger aircraft, there is a standard wing loading for aircraft as well as another parameter called the wetted wing loading which is the weight divided by the wetted area of the wing. The wetted area is basically the surface area of the wing. For radio controlled aircraft though the wing cube loading is used to ensure the aircraft is of the correct type.

$$WCL = \frac{W}{S^{3/2}} \tag{334}$$

Using the units of ounces for the weight and sqft for the area the table below can be used to ensure that your aircraft is in the correct ballpark[79].

Type of Aircraft	WCL (oz/ft^3)
Gliders	under 4
Trainers	5-7
Sport Aerobatic	8-10
Racers	11-13
Scale	over 15

After computing your WCL it is possible that for the type of aircraft you've designed, your value falls well outside the limits of the table above. In that case you must redesign your vehicle by changing the shape of the wings or choosing a different material to lighten up the aircraft. In my experience, if you are designing a racer or scale aircraft and you wing loading is smaller than above this is typically ok so long as you have enough thrust to fly fast. In this case you may just have a very fast trainer which isn't necessarily a bad thing. The danger is when trying to design a glider with a WCL of over 15. That aircraft will exhibit a very high stall speed and poor lift to drag ratio (L/D) characteristics. Once you are satisfied with your WCL you can move on to computing the thrust to weight ration (T/W).

For full-sized aircraft the T/W can range from 0.6 for passenger aircraft all the way up to 1.2 or higher for get aircraft. For R/C aircraft the same general rule applies. If your aircraft is a trainer with landing gear you can probably get away with a T/W of 0.6 but I would not recommend it. My recommendation would be to go no lower than 0.8 which would mean your maximum take off thrust is 80% of your maximum takeoff weight. In this configuration, the aircraft will accelerate down the runway until just over stall speed at which point the aircraft can takeoff. If you have strict runway requirements, airborne requirements like vertical flight or loops and snap rolls, I recommend increasing the size of your motor, ESC, battery, propeller combination to yield a T/W of at least 1.2. In this case, even if your wings are not very efficient, you can fly on thrust alone. You may not exhibit great

aerodynamic performance and still have a high stall speed but worst case you can land the aircraft like a harrier which I've done before.

Once you've selected your T/W you need to go find a battery/ESC/motor/propeller combination that yields the thrust you need. Tiger Motors website is typically very good at listing the motor, propeller and battery combination to give you a certain amount of thrust. Unfortunately, the hobbyist market is not using standard engineering units and thrust is reported in grams. My recommendation then is to use the following formula to compute your thrust required in grams. This assumes that 4.44 N = 1 lbf and that you are on Earth with 9.81 m/s^2 of gravitational acceleration. It'd be nice if the community just reported thrust in lbf but alas that is not the case.

$$T_{grams} = (T/W)W_{lbf}/453.59 \tag{335}$$

Remember, when selecting a propulsion system, you will need to go back and update your weight estimate with the new values you've obtained. This may effect your wing area slightly and may even require you to choose a new motor if you were very off the first time you estimated the weight. This is an iterative process and every step builds on the previous step. The hope is that each iteration is not very different than the last.

18.6   Battery Sizing

For battery sizing, you need to make sure the battery voltage is not too high for the ESC or the motor. Typically the ESC and motor will have a voltage limit listed on there. For example, an ESC that can handle 3-6S LiPos can handle anywhere from 12.6V to 25.2V. Motors may have different constraints than the ESC. It's also important to make sure that you look up the maximum current in the motor and that number must be less than the maximum current rating of the ESC. To compute flight time at max throttle use the equation below

$$\frac{mAh/1000}{A_{max,motor}} = TOF_{hours} \tag{336}$$

where mAh is the number of milliamp hours of your battery, $A_{max,motor}$ is the max current draw of your motor at full throttle and TOF_{hours} is the time of flight in hours. Note that you can repeat the calculation for different throttle settings provided you know the current through the motor.

You also need to make sure that the battery can output the necessary current your motor needs at full throttle. The equation below is used to ensure the battery can supply the necessary current.

$$\frac{mAh}{1000}C > A_{max,motor} \tag{337}$$

In the equation above, C is the C rating of the battery usually listed on the battery itself or in the product description.

18.7   Stability and Control, Center of Mass, Aerodynamic Center and Static Margin

Stability and Control is a very large section of literature and could be taught over an entire semester. Stability just ensure the aircraft flies steady and level and is typically broken up into lateral (side to side) and longitdutinal (front to back) stability. Lateral stability in my opinion is more complex but to ensure you aircraft is laterally stable, just be sure your aircraft is symmetric about the left and right planes and also be sure that your tail surface provides adequate yaw stability through the use of a vertical tail or a V-tail if you opted for a combined control surface. Longitudinal stability involved two more calculations that must be done before the aircraft can be built. These two parameters are the center of mass and the aerodynamic center. The center of mass is a very simple quantity to compute by just using the center of mass formula shown below.

$$x_{cm} = \sum \frac{x_i W_i}{W} \tag{338}$$

In the equation above, x_i is the distance of a component from a reference point on the aircraft. I typically use the nose of the fuselage as the reference point. For standard aircraft the motor would have a negative distance from the reference point and the receiver and battery would have a positive value. The value W_i is then the weight of each component. Placing servos, receivers and other electronics is a design parameter to move your center of mass while the fuselage is typically a parameter that must be estimated at this stage. My recommendation is to break the aircraft into fuselage, tail boom, tail and main wing components and treat each one as a component. You will notice that moving the main wing and battery drastically changes the center of mass.

The next parameter is the aerodynamic center. The main wing looking from the top is basically a 2D distributed load. As such the center of lift must be computed. Assuming the main wing is symmetric, the aerodynamic center will lie on the center line of the aircraft. In this case the problem reduces to a 1D computation. In order to compute the center of lift along the x-axis (pointing towards the nose) you need to compute the weighted average of the center of lift of each airfoil. In this case if you have a symmetric airfoil, the center of lift is 1/4 of the chord length. If you have a cambered airfoil the center of lift is typically in the 30% range so you can use 25% for a symmetric airfoil and something slightly larger for a cambered airfoil. Once you determined the center of lift for the airfoil you can use the equation below for the aerodynamic center of the entire wing[16, 75].

$$x_{ac} = \frac{2}{S} \int_0^{b/2} x_{af}(y)c(y)dy \tag{339}$$

The parameter $x_{af}(y)$ is the location of the center of lift of the airfoil as a function of y. Once you have the aerodynamic center and the center of mass you can compute the static margin of your aircraft.

$$S_m = \frac{x_{ac} - x_{cm}}{\bar{c}} \tag{340}$$

The value of the static margin is not as important as the sign. If measuring from the nose of the aircraft the aerodynamic center must be behind the center of mass. To explain this think about stable aerodynamic vehicles like darts, or arrows. Notice that darts and arrows have fletching in the rear to create aerodynamic surfaces farther back. Unstable vehicles like frisbees and footballs have aerodynamic centers in front of the center of mass in which case they must spin in order to provide stability just like a bike tire or a dreidel. Using the equation above, if x_{ac} is behind x_{cm} it means that x_{ac} is bigger or more positive than x_{cm} in which case your static margin would be positive.

$$\begin{Bmatrix} S_m > 0, & stable \\ S_m < 0, & unstable \end{Bmatrix} \tag{341}$$

If you perform these two calculations and find your static margin to be negative it means that you need to shift your battery and other components more towards the nose or move your wings backwards. Note that moving your wing backward will also shift your center of mass so try and move some components forward before shifting your wings around.

The final stage of this design is Control. Aircraft in flight require 3 control surfaces to provide roll, pitch and yaw control. These include the aileron, elevator and rudder. It's possible to fly aircraft without a rudder by performing a "bank and yank" maneuever and it's also possible to combine elevators and ailerons into something called elevons. I've even seen some ruddervators. Whatever you decide to do make sure that you can adequately control all three axes or if one axis is uncontrollable be sure that that axis is stable. I've seen some aircraft that only have rudder and elevator and no ailerons. I find these aircraft hard to control but the idea is you move the rudder to yaw the aircraft which also rolls the aircraft allowing you to turn. It creates a very slow aircraft but it also reduces complexity if that is something you're interested in doing.

18.8   Iteration, Detailed Sketch and Final Checks

This section in my opinion is absolutely essential. It involves going back and making sure that your current design satisfies your requirements you originally wrote in the first section of this design. Recompute your aspect ratio, and update your weight estimate based on any calculations you've obtained. This may be finding better estimates for parts or materials. You also need to create a better sketch and determine where EVERY component is going to go and what sort of support you will need. If you're building your aircraft out of balsa you will need detailed sketches on rib, spar and stringer placement. All of these updates will change your weight estimate which will change your WCL and your T/W. Be sure your WCL and T/W are within tolerable bounds. You also need to go back and compute you angle of attack during cruise and be sure you are not in a stall regime. Be realistic with your flight speed as well during cruise. Go back and compute your stall speed. Is it realistic? If not then go back and make some minor changes. Finally, be sure you aircraft is longitudinally and laterally stable and that you can control all 3 axes or at least the uncontrollable axes are stable. Once you are certain the aircraft will fly you can begin purchasing components.

18.9   Computer Aided Design (CAD)

This section is optional but sometimes it's just nice to have a CAD view of your aircraft especially if you are 3D printing parts or perhaps getting some component machined out of aluminum. Some CAD programs are even so

powerful they will compute bending loads, center of mass and even drag. Use whatever tools are at your disposal to help you in the design.

18.10   Purchase Components

I wanted to write an entire section on purchasing component to go over a few common mishaps. First, if you are using a LiPo battery be sure to familiarize yourself with the dangers of LiPo batteries. I've almost caught my entire lab on fire by charging a damaged LiPo but all batteries can technically catch fire. Please be careful. Furthermore, be sure you purchase an ESC with the right current rating. If you overload an ESC with too much current it will also catch on fire. The servos you buy have a torque rating. Be sure your servos can overcome the aerodynamic torque estimated in flight. Generally servos are sized by the size of the aircraft so you can just purchase servos that are designed for your particular size aircraft. The motor you purchase is going to need a mounting point. Consider designing a motor plate or even a firewall depending on the type of aircraft. When purchasing materials make sure to get them from a good brand. Companies like Flite Test sell really good double plated foam that is designed for RC aircraft. Purchasing Dollar Tree foam board is totally acceptable but just understand that after a crash or two the foam board won't work anymore. Also consider what type of glue you are planning on using. Certain types of glue can actually melt foam and hot glue can melt certain types of foam as well. CA glue is very good for balsa but will melt foam. Two part epoxy is strong but it weighs more than CA glue and takes a long time to set compared to other types of glue. Also be sure to be lenient on glue where you can. Glue just adds weight and that will reduce your performance. Finally, make sure your receiver supports the number of servos you plan on using and be sure that your transmitter and receiver are compatible. All of my aircraft use Spektrum technology but you may opt to use a different type of protocol.

   Finally, when all components come in be sure to test them. DOA stands for dead on arrival and so many components come DOA. Before you spend the time to install all your components in your aircraft and then throw the aircraft in the air make sure you test every component and be sure it works. I also suggest weighing each component on a small scale and updating your weight estimate to ensure everything is within tolerable bounds.

18.11   Building

Once you have all necessary resources to build your aircraft I recommend starting with the fuselage or main wing and then installing all componets. I recommend taking pictures of your build in case you need to reference them later. Go slow. If you break something it will be expensive. Also remember that if the aircraft looks good it flies good. This means that gluing all components properly and having the aircraft be as smooth as possible will translate to better flight performance.

18.12   Flying

Before you fly I recommend finding a pilot will more experience than yourself to check out the aircraft and make sure the aircraft has been built properly. You may even want to discuss your initial design before you even begin building to make sure there are no major critical issues. If you want to fly the aircraft yourself I suggest flying an aircraft using a simulator. My recommendation is the free program CRRCSim[80]. It is not a great simulator by any means but it will at least familiarize yourself with aircraft controls. Before you fly make sure to build yourself a pre and post flight check list. I've included my pre and post flight check list that I use before every flight test.

18.12.1   Day Before Flight Checklist

1. Assess the weather to ensure acceptable flight conditions

    (a) No strong winds (anything over half your stall speed is too fast)
    (b) No rain or lighting
2. State and confirm the purpose of the flight test - Set clear goals the aircraft should complete before test
3. Check for damage to the plane and if the moving parts are secured including motor and electronic speed control and all components
4. Check for a full battery charge on plane and controller; charge all electronics if not fully charged
5. Perform Ground Safety Check List
6. Take note of items that need to be repaired even if the flight test is not implemented

18.12.2 Ground Safety Check List

1. Ensure that propellor is off
2. Turn on TX
3. Connect battery to aircraft
4. Ensure all control surfaces are operational and moving the correct way
5. Spin up motor and be sure that motor is spinning the correct way
6. Perform a range check where pilot moves control surfaces with 50% or more throttle while walking away from aircraft. Ensure that pilot can move at least 300 feet away without any dropouts.
7. Remove battery and install propeller
8. Reconnect battery.
9. Using safety glasses, apply full throttle to TX and ensure that adequate thrust is generated to fly aircraft. Leave full throttle applied for at least 30 seconds to be sure no component. fails. Better to fail on the ground than in the air.

18.12.3 Preflight Checklist

1. Perform all "Day Before Flight" Checks
2. Perform Ground Safety Checks
3. Check for any damage to any components including the battery
4. Install Prop
5. Turn on TX
6. Plug in main battery
7. Confirm flight time and range distance
8. Clear obstructions and make sure there is clear space for takeoff and landing
9. Arm TX if the TX has an arm switch
10. Ensure all control surfaces are operational and moving the correct way
11. Apply throttle and fly
12. Upon landing do everything in reverse order

18.12.4 Post Flight checklist

1. Check plane for any damage
2. Check all moving parts are still secured
3. Check for battery overheating, discoloration, warping, or swelling
4. Check battery usage with a voltmeter
5. Check if the plane is able to power on again
6. Have PIC (pilot in command) give a post flight assessment
7. Put batteries in LiPo storage

19 Helpful Aircraft Equations

Mach Number and Reynolds Number

$$M_\infty = \frac{V}{a_\infty}$$

$$Re = \frac{\rho V \bar{c}}{\mu_\infty} \tag{342}$$

Total Velocity

$$V = \sqrt{u^2 + v^2 + w^2} \tag{343}$$

Angle of Attack and Sideslip

$$\alpha = tan^{-1}\left(\frac{w}{u}\right)$$
$$\beta = sin^{-1}\left(\frac{v}{V}\right) \tag{344}$$

Lift Drag and Moment

$$Lift\ (L) = \frac{1}{2}\rho V^2 S C_L$$
$$Drag\ (D) = \frac{1}{2}\rho V^2 S C_D$$
$$Roll\ Moment\ (L) = \frac{1}{2}\rho V^2 S b C_l \tag{345}$$
$$Pitch\ Moment\ (M) = \frac{1}{2}\rho V^2 S \bar{c} C_m$$
$$Yaw\ Moment\ (N) = \frac{1}{2}\rho V^2 S b C_n$$

Lift and Drag Coefficients

$$C_L = C_{L0} + C_{L\alpha}\alpha$$
$$C_L = C_{L\alpha}(\alpha - \alpha_0)$$
$$C_D = C_{D0} + C_{D\alpha}\alpha^2 \tag{346}$$
$$C_D = C_{D0} + kC_L^2$$

Non-dimensional Angular velocities

$$\hat{p} = pb/2V$$
$$\hat{q} = q\bar{c}/2V \tag{347}$$
$$\hat{r} = rb/2V$$

Pitch Moment equation

$$C_m = C_{m0} + C_{m\alpha}\alpha + C_{m\delta_e}\delta_e + C_{mq}\hat{q} \tag{348}$$

$$C_{m0} = C_{MAC} + C_{L0}\bar{x}_{sm}$$
$$\bar{x}_{sm} = \frac{x_{cg}}{\bar{c}} - \frac{x_{acW}}{\bar{c}}$$
$$C_{m\alpha} = \left(C_{L\alpha,W} + \frac{S_t}{S}C_{L\alpha,t}\right)\bar{x}_{sm} - V_H C_{L\alpha,t}$$
$$V_H = \frac{l_t S_t}{S\bar{c}} \tag{349}$$
$$C_{m\delta_e} = \left(C_{Lt\delta_e}\frac{S_t}{S}\right)\bar{x}_{sm} - V_H C_{Lt\delta_e}$$
$$C_{mq} = 2C_{L\alpha t}\frac{l^2}{\bar{c}^2}$$

Max Lift to Drag Ratio (Only valid if $C_{L0} = 0$)

$$\alpha_{max,L/D} = \sqrt{\frac{C_{D0}}{C_{D\alpha}}} \tag{350}$$

Lift to Drag when $T = 0$ (Sum of Forces still zero)

$$\frac{D}{L} = tan(\alpha)$$
$$Lcos(\alpha) + Dsin(\alpha) = W \tag{351}$$

Airfoil and Wing Aerodynamics

$$x_{ac} = c/4 \quad a = \frac{a_0}{1 + \frac{a_0}{\pi e AR}}$$
$$AR = \frac{b^2}{S} \tag{352}$$

Standard Atmosphere

$$\rho = 1.225\ kg/m^3 = 0.00238\ slugs/ft^3$$
$$\mu_\infty = 1.81x10^{-5}\ kg/(m-s) \tag{353}$$
$$a_\infty = 331.3\ m/s$$

General Notes

1. In trim or steady and level or cruise $q = 0$, $C_m = 0$, $L = W$, $T = D$

2. For symmetric airfoil $C_{MAC} = 0$ and $C_{L0} = 0$ thus $C_{m0} = 0$

3. For a flat plate all symmetric properties apply but $a_0 = 2\pi$

4. Tail surfaces are always assumed to be flat plates

5. For longitudinal problems, $\beta = 0$ so $v = 0$ (side velocity)

References

[1] GitHub. GitHub Copilot. https://copilot.github.com/, 2021. Accessed on October 20, 2025.

[2] Google DeepMind. Gemini: Googles multimodal ai model. https://deepmind.google/technologies/gemini/, 2023. Accessed on October 20, 2025.

[3] B. Etkins. *Dynamics of Atmospheric Flight*. Dover, Mineola, New York, 2000. pages 9-13,134-151,196-318.

[4] Warren F. Phillips. *Mechanics of Flight*. John Wiley and Sons Hoboken New Jersey, 2010.

[5] R.C. Nelson. *Flight Stability and Automatic Control*. McGraw-Hill, 2nd edition, 1998.

[6] Jerry E. White Roger R. Bate, Donald D. Mueller. *Fundamentals of Astrodynamics*. Dover, 1971.

[7] Arthur KL Lin and Regina Lee. Attitude control for small spacecraft with sensor errors. In *AIAA SPACE Conferences and Exposition, Pasadena, California*, page 4423, August 2015.

[8] Nuno Filipe and Panagiotis Tsiotras. Adaptive position and attitude-tracking controller for satellite proximity operations using dual quaternions. *Journal of Guidance, Control, and Dynamics*, 38(4):566–577, 2015. doi:10.2514/1.G000054.

[9] 0. A. Bachau. *Flexible Multibody Dynamics*. Springer, 2008.

[10] Karsten Grosekathofer and Zizung Yoon. Introduction into quaternions for spacecraft attitude representation. *Technical University of Berlin Department of Astronautice and Aeronautics*, 2012.

[11] John L. Cassidis and John L. Junkins. *Optimal Estimation of Dynamic Systems*. Chapman and Hall/CRC, 2004.

[12] Sebastian Munoz and E. Glenn Lightsey. A sensor driven trade study for autonomous navigation capabilities. *AIAA*.

[13] John L. Cassidis, F. Landis Markley, and Yang Chen. Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1), 2007.

[14] Bing Liu, Zhen Chen, Xiangdong Liu, and Fan Yang. An efficient nonlinear filter for spacecraft attitude estimation. *International Journal of Aerospace Engineering*, 2014(540235), 2014.

[15] R. M. Georgevic. The solar radiation pressure forces and torques model. *The Journal of the Astronautical Sciences*, 27(1), 1973.

[16] Anderson D. *Fundamentals of Aerodynamics 4th Edition*. McGraw Hill Series, 2007.

[17] Graham Gyatt. The standard atmosphere. *A mathematical model of the 1976 U.S. Standard Atmosphere*, Jan 2006.

[18] Geographic library c++. https://geographiclib.sourceforge.io/. cited Jul 11, 2017.

[19] L. G. Jacchia. Static diffusion models of the upper atmosphere with empirical temperature profiles. *Smithson. Astrophys. Obs. Spec. Rept.*, (170), 1964. Cambridge, Massachusetts.

[20] L. G. Jacchia. Revised static models of the thermosphere and exosphere with empirical temperature profiles. *Smithson. Astrophys. Obs. Spec. Rept.*, (332), 1971. Cambridge, Massachusetts.

[21] L. G. Jacchia. Thermospheric temperature, density, and composition: New models. *Smithson. Astrophys. Obs. Spec. Rept.*, (375), 1977. Cambridge, Massachusetts.

[22] Enhanced magnetic field model. https://www.ngdc.noaa.gov/geomag/EMM/. cited Jul 11, 2017.

[23] Neil Ashby. The sagnac effect in the global positioning system. *Relativity in rotating frames: relativistic physics in rotating reference frames*, page 11, 2004. doi:1-4020-1805-3.

[24] Earth gravity model. https://www.ngdc.noaa.gov/geomag/EMM/. cited Jul 31, 2017.

[25] Chemical propulsion systems. glenn research center — nasa. https://www1.grc.nasa.gov/research-and-engineering/chemical-propulsion-systems/l-propulsion-systems/. Retrieved September 29, 2021.

[26] Solid rocket propulsion. https://engineering.purdue.edu/ propulsi/propulsion/rockets/solids.html. Retrieved September 29, 2021.

[27] Waslander S.L. Hoffman G.M., Huang H. and Tomlin C.J. Quadrotor helicopter flight dynamics and control: Theory and experiment. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 6–15, 2007.

[28] H. Huang, G. Hoffman, C. Talor, S. Waslander, and C. Tomlin. Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. *Proc. IEEE Int. Conference of Robotics and Automation (ICRA)*, pages 3277–3282, May 2009. doi:10.1109/ROBOT.2009.5152561.

[29] Moosavian S.A.A. Sadr S. and Zarafshan P. Dynamics modeling and control of a quadrotor with swing load. *Journal of Robotics*, 2014.

[30] Jet Propulsion Laboratory. Solar system dynamics. https://ssd.jpl.nasa.gov/ Accessed October 4th, 2019.

[31] Steven C. Chapra and Raymond P. Canale. *Numerical Methods for Engineers*. McGraw-Hill Education, 7th edition, 2012.

[32] Stephen M. Stigler. Gauss and the invention of least squares. *Ann. Statist.*, 9(3):465–474, 05 1981. doi:10.1214/aos/1176345451.

[33] 1736-1813; Binet Jacques Philippe Marie 1786-1856; Garnier Jean Guillaume 1766-1840 Lagrange, J. L. (Joseph Louis). Mécanique analytique. *Paris, Ve Courcier*, 1, 1811.

[34] A. B. Younes, D. Mortari, J. D. Turner, and J. L. Junkins. Attitude error kinematics. *Journal of Guidance, Control and Dynamics*, 37(1), 2014.

[35] S. Caldwell. Ground data systems and mission operations. nasa. http://www.nasa.gov/smallsat-institute/sst-soa/ground-data-systems-and-mission-operations. Accessed Nov. 4, 2021.

[36] H. Monaghan. Networks. nasa. http://www.nasa.gov/directorates/heo/scan/services/networks/index.html. Accessed Nov. 4, 2021.

[37] AmericaSpace. Happy birthday, dsn: Nasa's iconic deep space communications network turns 50. https://www.americaspace.com/2013/12/26/happy-birthday-dsn-nasas-iconic-deep-space-communications-network-turns-50/. Accessed Nov. 4, 2021.

[38] Mirko Leomanni. Comparison of control laws for magnetic detumbling. *Research Gate*, 10 2012. accessed: 2/9/2018.

[39] M. Lovera. Magnetic satellite detumbling: The b-dot algorithm revisited. In *2015 American Control Conference (ACC)*, pages 1867–1872, July 2015. doi:10.1109/ACC.2015.7171005.

[40] Farhat Assaad. *Attitude Determination and Control System for a CubeSat*. PhD thesis, Worcester Polytechnic Institute, March 1st 2013.

[41] Amit Sanyal and Zachary Lee-Ho. Attitude tracking control of a small satellite in low earth orbit. Aug 2009. doi:10.2514/6.2009-5902.

[42] Fabio Celani. Robust three-axis attitude stabilization for inertial pointing spacecraft using magnetorquers. *ScienceDirect: Acta Astronautica*, 107:87–96, February–March 2015. doi:10.1016/j.actaastro.2014.11.027.

[43] E. Carlen and M.C. Carvalho. *Linear Algebra: From the Beginning*. W. H. Freeman, 2006.

[44] Wikipedia. V-modle. https://en.wikipedia.org/wiki/V-Model. Accessed December 21st, 2021.

[45] Gemini spacecraft. navigation system. http://geminiguide.com/Systems/navigation.html/Attitude_Control_and_Maneuver_Electronics. Retrieved September 26, 2021.

[46] Spacecraft overview. new horizons. https://spaceflight101.com/newhorizons/spacecraft-overview/. Retrieved September 26, 2021.

[47] Sysml activity diagram — enterprise architect user guide. https://sparxsystems.com/enterprise_architect_user_guide/15.2/model_domains/sysml_activity_diagram.html. Accessed Nov. 28, 2021.

[48] Uml - activity diagrams. https://www.tutorialspoint.com/uml/uml_activity_diagram.htm. Accessed Nov. 28, 2021.

[49] W. Emery and A. Camps. Introduction to satellite remote sensing: Atmosphere, ocean, land and cryosphere applications. *Elsevier Science Publishing*, 2017.

[50] H. Curtis. Magnetorquers: An overview of magnetic torquer products available on the global marketplace for space. https://blog.satsearch.co/2019-08-21-magnetorquers-an-overview-of-magnetic-torquer-products-available-on-the-global-marketplace-for-space. (2021, July 22).

[51] CubeSTAR. Cubestar. attitude determination & control. http://cubestar.no/index.php?p=1_22_Attitude-Determination-Control:~:text=The%20detumbling%20is%20the%20process,derivative%20of%20the%20angular%20rate. Retrieved November 3, 2021.

[52] Nctr-m002 magnetorquer rod. cubesatshop.com. https://www.cubesatshop.com/product/nctr-m002-magnetorquer-rod/. (2020, November 6).

[53] Reaction wheels: An overview of attitude control systems available on the global marketplace for space. https://blog.satsearch.co/2019-07-25-reaction-wheels-an-overview-of-attitude-control-systems-available-on-the-global-marketplace-for-space. Accessed Nov. 3, 2021.

[54] NanoAvionics. Cubesat reaction wheels control system satbus 4rw.

[55] Blue canyon technologies. https://www.bluecanyontech.com/components. Retrieved November 29, 2021.

[56] Thrust vector control - an overview — sciencedirect topics. https://www.sciencedirect.com/topics/physics-and-astronomy/thrust-vector-control. Accessed Nov. 4, 2021.

[57] Gimbaled thrust. https://www.grc.nasa.gov/www/k-12/rocket/gimbaled.html. Accessed Nov. 4, 2021.

[58] Control devices. https://mars.nasa.gov/mro/mission/spacecraft/parts/gnc/controldevices/. Accessed Nov. 4, 2021.

[59] Low thrust, low impulse bit reaction control thruster. https://artes.esa.int/projects/low-thrust-low-impulse-bit-reaction-control-thruster. Accessed Nov. 4, 2021.

[60] Sinclair D. Votel, R. and S. Interplanetary. Comparison of control moment gyros and reaction wheels for small earth-observing satellites.

[61] B. L. Gyroscope.com. Control moment gyroscope platform - mk2. https://www.gyroscope.com/d.asp?product=CMGMK2. Accessed Nov. 4, 2021.

[62] What is a magnetometer? sciencing. https://sciencing.com/about-5397128-magnetometer.html. Accessed Nov. 4, 2021.

[63] Satcatalog. hi-rel magnetometer fgm-a-75 - magnetometer — satcatalog. https://www.satcatalog.com/component/hi-rel-magnetometer-fgm-a-75/. Accessed Nov. 4, 2021.

[64] Sensonor, "sensonor - stim300 sensonor.no.". https://www.sensonor.com/products/inertial-measurement-units/stim300/. Accessed: 25-May-2021.

[65] Industries, a. adxl345 - triple-axis accelerometer (+-2g/4g/8g/16g) w/ i2c/spi. adafruit industries blog rss. https://www.adafruit.com/product/1231. Retrieved November 5, 2021.

[66] Satellite star trackers - the cutting edge celestial navigation products available on the global space marketplace. satsearch blog. https://blog.satsearch.co/2019-11-26-satellite-star-trackers-the-cutting-edge-celestial-navigation-products-available-on-the-global-space-marketplace. Accessed Nov. 3, 2021.

[67] Spice toolkit. https://naif.jpl.nasa.gov/naif/toolkit.html. Accessed Nov. 3, 2021.

[68] Advanced star tracker. https://www.esa.int/Applications/Telecommunications_Integrated_Applications/Alphasat/Advanced_Star_Tracker. Accessed Nov. 3, 2021.

[69] What is a sun sensor? https://www.azosensors.com/article.aspx?ArticleID=223. Accessed Nov. 3, 2021.

[70] Nss cubesat sun sensor. CubeSatShop.com.

[71] Vincent L. Pisacane. Fundamentals of space systems. *New York, Oxford University Press*, 2005.

[72] SatCatalog. Satcatalog. cubeir - earth/horizon sensor. https://www.satcatalog.com/component/cubeir/. Accessed Nov. 4, 2021.

[73] GPS.Gov: Space Segment. https://www.gps.gov/systems/gps/space/. Accessed Nov. 3, 2021.

[74] Carlos Montalvo. Radio controlled aircraft design. accessed:12/30/2020 https://www.youtube.com/watch?v=RjC3Rou0wkA&list=PL_D7_GvGz-v3d8gRDXWjQaVYfvNOPVx97.

[75] Caughey D.A. *Introduction to Aircraft Stability and Control Course Notes for M&AE 5070.* , edition, 2011.

[76] Montalvo C. Getting lift curve slopes and drag polars using xflr5. accessed 12/30/2020 https://www.youtube.com/watch?v=Yz7rmuhGYWO&list=PL_D7_GvGz-v3d8gRDXWjQaVYfvNOPVx97&index=4.

[77] Andre Deperrois. About stability analysis using xflr5 presentation document. June 2008. http://www.xflr5.tech/docs/XFLR5_and_Stability_analysis.pdf.

[78] Xflr5. accessed 12/30/2020 http://www.xflr5.tech/xflr5.htm.

[79] Ken Meyers. Wing cube loading (wcl). Accessed: 12/31/2020 https://www.sefsd.org/general-interest/wing-cube-loading-wcl/.

[80] Crrcsim. accessed: 12/30/2020 https://sourceforge.net/projects/crrcsim/.