

Using time & frequency domain data analytics for optimal sensor fusion design

Dario Antonio Quintero Dominguez

February 5, 2020

Abstract

Multi-dimensional data analysis was performed to characterize and define the optimal configuration of a multi-sensor heading reference system. A complementary sensor system of a compass and gyro systems was analysed by varying its cutoff frequencies and simulation-time parameters with time-domain statistical analysis. Operational limitations for all components are described. The optimal cutoff frequency was selected at 1 *rad/s* since this is the "steady state" frequency after which all larger cutoff frequencies do not alter the response even when varying simulation time. A combination of standard deviation and RMS is suggested as the error metric for the sensors.

1 Sensor Heading System

The design of a heading directional sensor system is inherently related to the aircraft in which it is installed. Mechanical loads, vibrations, and operational frequencies interfere with electro-mechanical sensors and the outputted electronic signals accordingly. Small or medium sized UAVs also require a an independent heading system when telecommunication is not possible. The fundamental configuration of the integration of a gyroscope and a magnetic compass will be analysed for this scenario and optimal system design metrics are proposed.

In order to design a low cost system, off-the-shelf components are selected. The magnetic compass is lightly damped due to fluid viscosity, and has a rise time $t_{rise} = 0.4s$ and settling time $t_{settling} = 18s$. Levi et al. [4] describe how traditional magnetic compasses are subject to errors from both local magnetic fields and accelerations due to motion. These can also contribute to further dynamic controls.

Given the step response parameters above, the compass system can be idealized as a second order system in the form of Equation 2 in the time domain and Equation 3 in the Laplace domain as further where w_n is natural frequency, ζ is the damping ratio, and λ is the response amplitude gain described by [7]. The under-damped equation condition of a damping ratio $\zeta \ll 1$ is tested and determined to be valid in Equation 1. The derived transfer function in Equation 3 can be analysed in MATLAB in the frequency domain using bode plots. It provides an indication of the response for several frequencies excitation as observed in Figure 1.

$$\zeta \approx \frac{3}{w_n t_{settling}} \approx \frac{3t_{rise}}{1.02t_{settling}} \approx 2.94 \frac{t_{rise}}{t_{settling}} = 0.0666 \quad (1)$$

$$\ddot{y}(t) + 2\zeta\dot{y}w_n(t) + w_n^2 y(t) = \lambda w_n^2 u(t) \quad (2)$$

$$\frac{Y(s)}{U(s)} = \frac{\lambda w_n^2}{s^2 + 2\zeta w_n s + w_n^2} = \frac{6.25}{s^2 + 0.333s + 6.25} \quad (3)$$

Mathematically, it is known from the bode plots that the compass system operates linearly at low mechanical frequencies. To analytically derive this linear range of the response below 3dB, the slopes between the plot sample points were taken to determine their limit closer to 0. All the green points within the magnitude analytic bode plot have a slope between the points of 0.15 in Figure 1, and the maximum linear response at this position occurs at 1.29 *rad* with a magnitude of 2.59 *dB*. If the linear slope restriction would be lowered to 0.12, for example, the maximum response in this line would be at the sampled point of 1.04 *rad* with a maximum magnitude of 1.4969 *dB*. The frequency response of the compass becomes nonlinear since it has a resonance at the natural frequency w_n . The resonant point is achieved at 2.55 *rad* with a magnitude of 16.7 *dB*. The poles are derived as $p_{1,2} = \zeta w_n \pm jw_n \sqrt{1 - \zeta^2}$ for an under-damped system [7] with the damping ratio approximation described in Equation 1. The two horizontal lines in Figure 1 indicate the -3 *dB* and + 3 *dB* range of the response for amplification comparison reference. For the purposes of this sensor model, a transfer function block with these parameters is used in Simulink.

In order to accurately model the gyro, its mechanical operation needs to be considered. The physical design of a rate gyro is described in Morris & Langari [5], and consists of mainly consists of a spinning wheel attached

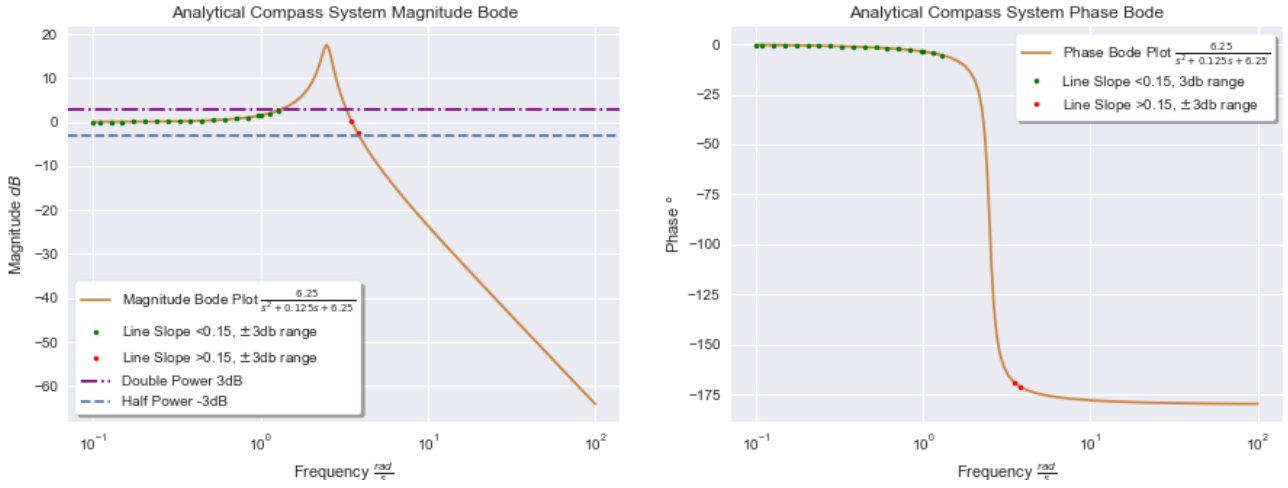


Figure 1: Compass Bode Plot

to a gimbal rotational support that rotates around a fixed static frame. The idealised physical behaviour of how this gyro is described in the time domain and Laplace domain is shown in Equation 4. If the spinning mass has large enough angular momentum h , it would mean that large, torques T exerted on the sensor would force small rotation Θ_m that could be measured by a transducer. Assuming a burst of wind generates torques within the gyro frequency range, destabilizing friction, material stiffness and a limited measurement range could provide inaccurate signals. Hence, low turning rates might not be accurately measured, whereas higher turning rates would.

$$T(s) = \mathcal{L}\{h\dot{\theta}_m\} = sh\theta_m \quad (4)$$

$$\frac{\theta_m}{T} = \frac{1}{sh} \quad (5)$$

To model these imperfections within Simulink, a random number with mean of 0.02 and standard deviation of 0.1 would be the equivalent of $\frac{1}{h}$ as observed in Figure 2b. A value for angular momentum h cannot be explicitly determined since this modelling includes random error considerations. This multiplies an integrator block $\frac{1}{s}$. Purely analytically, it is known that the bode plot of an integrator is a line with a negative slope as observed in Figure 2a, and it would also have jitter when considering the random mean gain. A discrete Fast Fourier transform could also have been performed on the integrator output to obtain this diagram. However, for the purposes of idealistic design, it can already be observed that for a range of frequencies of 10^{-2} to 10^2 rad/s the gyro system provides a linear magnitude response with attenuation at faster frequencies. After the 10^2 rad/s point, the gyro would not respond to higher frequencies.

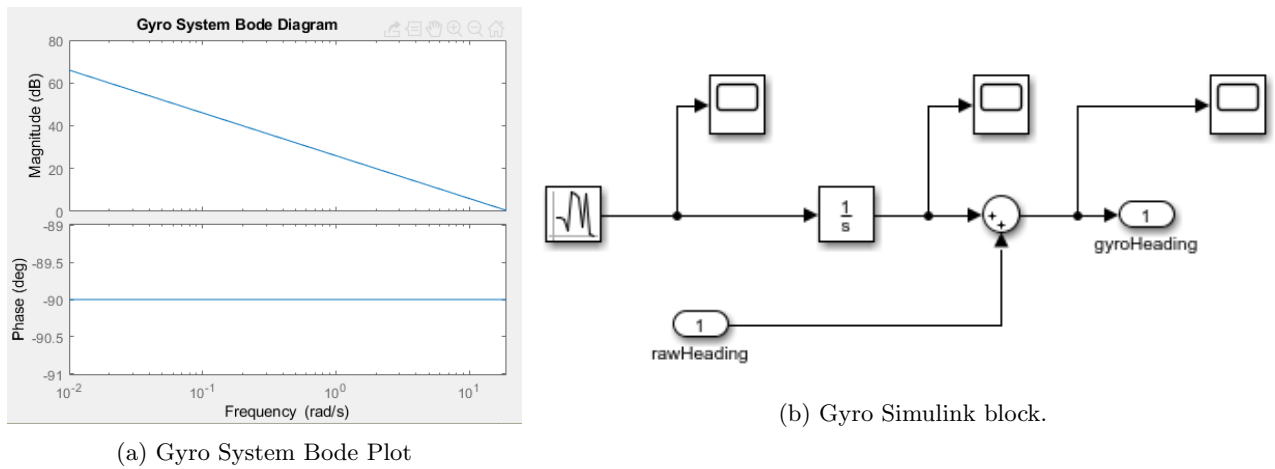


Figure 2: Gyro Simulink Block

Fox et al. [3] demonstrate how sensor fusion through complementary filters integrates responses to minimize individual sensor errors through Kalman algorithms. A simpler, non-probabilistic approach, will be tested

by creating complementary filters with the mathematical relationship in Equation 6 where the time constant $\tau = \frac{1}{w_c}$ is inverse to the cutoff frequency w_c in rad/s . Since filters have the same cutoff frequency, when the low pass $L(s)$ compass filter response begins diminishing, the high pass $H(s)$ gyro filter magnitude increases. There is no amplification gain from this sensor configuration since its absolute value is unity ($K = 1$). The optimal cutoff frequency would reduce the levels of oscillations of the full system signal by making each sensor respond within its ideal operational range.

$$L(s) + H(s) = \frac{s}{s + \frac{1}{\tau}} + K \frac{1}{\tau s + 1} = 1 \quad (6)$$

Using all the numerical parameters described previously, sensor fusion was performed and analysed on a magnetic compass and gyroscope as observed in Figure 3 using Simulink. The selected solver configuration was the variable step ODE-45 with 0.1 sample time was selected since it consistently provided expected results. Although larger sampling times also provided accurate results, computational efficiency was not completely needed for the algorithms in this model. The standard simulation time selected was 350 seconds.

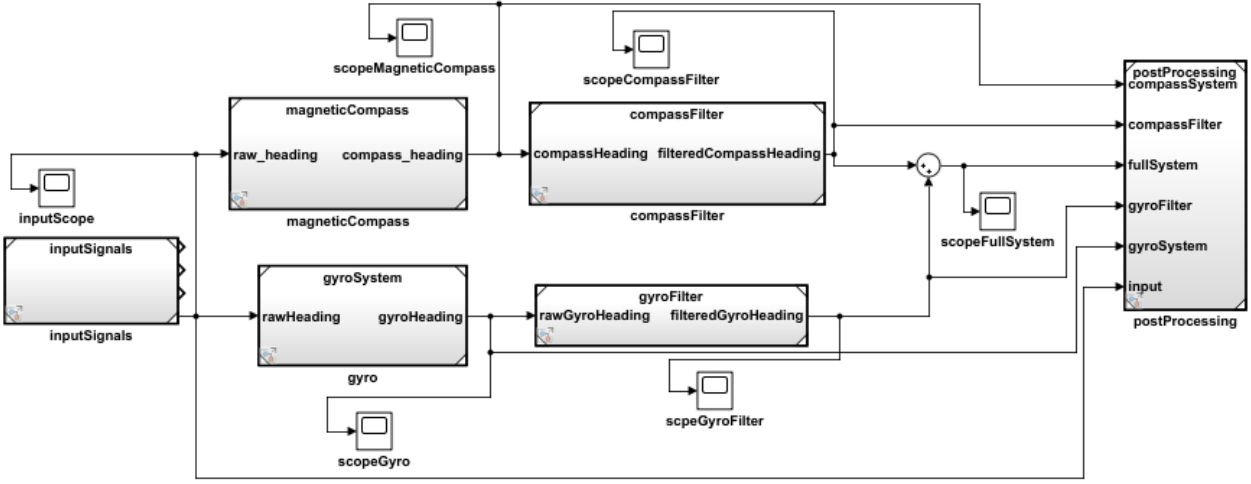


Figure 3: Heading system Simulink model

Several improvements could be done to this sensor fusion design to improve its accuracy, and possible future work could be performed analysing its behaviour. Within traditional aircraft navigation, once the rotational motion is negligible, the zero offset of the gyroscope is generally re-tuned, adding some degree of error. Micro-electrically modulated gyroscopes also reduce the zero-offset error, and Kalman auto-correction algorithms could be implemented as described by Levi et al. [4] and Carque et al. [1]. Integrating accelerometers into the heading system can further improve the accuracy of the signal since inertial changes can be considered alongside mobility changes of the gyro at higher frequencies; as described by Young Quist [8] and Foxlin [3]. Lastly, adding extra gyro sensors could potentially add redundancy for the random error in case no other complementary sensors can be added.

2 True Heading Signal

The heading signal aims to comprehensively test the response of the sensor system for a range of real frequencies. The heading changes modelled were realistically tuned for a small UAV, such as a DJI Mavic 2 [6] with a turning rate capability of $200^\circ/s$ since it helps characterize the sensor system combination. For slower UAVs, the sensor system response limitations do not change, although the heading signal differentials would be slower. Dorobantu et al. [2] identify a number of flight dynamic frequencies of small, low cost fixed-wing UAVs that are considered in this model. The applicable frequency range that describes the majority of their relevant aircraft dynamics is suggested to be between 10^{-1} and $10^1 rad/s$ for their aircraft, although potentially faster UAVs prone to influence by higher frequencies. For reference, spiral, dutch roll, and rolling flight lateral/directional movements within the yaw angle X-Y plane introduce frequencies of 0.021, 6.102 and 14.912 rad/s respectively. The DJI Mavic 2 flies differently and experiences other forces not considered by this model, and this is why the heading signal should mainly excite directional movements within these frequencies. Figure 4 describes the selected heading signal.

The heading signal can be separated into two analysis sections. From seconds 0 to 70, higher frequency changes aim to describe the transient response at the very limit of the DJI Mavic 2 physical operation. Part of this region is illustrated in Figure 5, although zoomed in to provide more detail. Drastic changes from 100

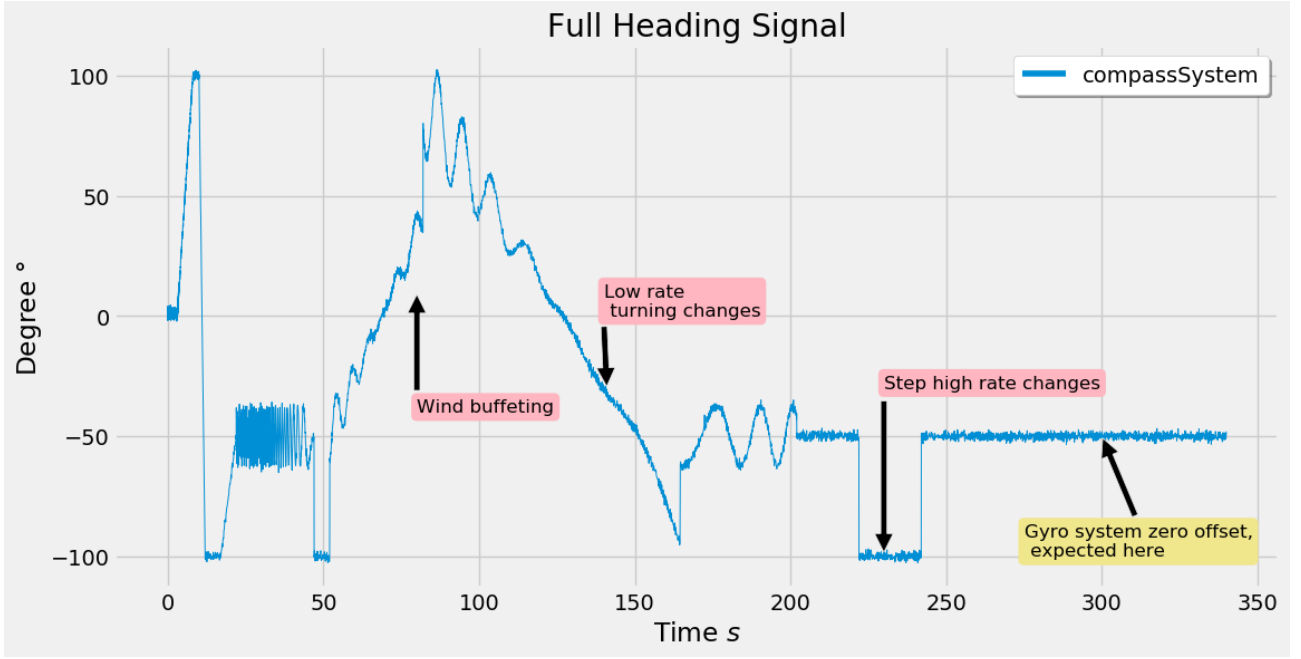


Figure 4: Complete heading signal.

to -100 degrees are observed in a space of 3 seconds at the start, close to the rated turning rate of the DJI Mavic of $200 \frac{^\circ}{s}$. Random noise jitter with a mean of 0.05° has been added to the complete heading signal to the minor imperfections in directional flight as can be individually noticed within the step signal or straight flight towards the end. The chirp sinusoidal wave between seconds 22 to 47 decreases frequency from 20 Hz to 0.01 Hz, within the range of most flight dynamics vibrations suggested by [2]. Mainly steep change slopes are observed at seconds 10, 20, 47 and 52 aim to demonstrate high rate turns.

The second section aims to test slower changes from second 70 onwards. Low rate turns with wind buffeting is described between seconds 50 to 120. Between seconds 160 and 210, sinusoidal waves also aim to demonstrate further low frequency behaviour. The rate gyro zero offset is expected to be observed towards the straight end of the heading flight.

During the design of the heading signal, several standard input signals such as the saw-tooth, sine, step, and delta waves with varying frequencies were independently inputted to understand the specific response of the system and design the heading signal for several frequency behaviours. Most of these independent signal responses are displayed within this heading signal.

3 Results Analysis

Multidimensional data analysis was performed to determine accurate statistical metrics to optimize the sensor fusion design. A set of signals comprised of the raw time signals of the compass system, compass filter, full system combined, gyro system, gyro filter, and input heading signal can be observed in Figure 6. In order to quantify the error of each sensor system output, the difference δ_k was determined for each signal y_k with the input heading signal y_i . A set of new sensor data was created compounding by these differences δ_n . To adequately quantify the error of each sensor output, analysis must be performed throughout a range of cutoff frequencies since this could drastically change the response of the full system. It is not expected that the independent system outputs for the gyro and compass systems would change over a range of cutoff frequencies since they are decoupled from this variation. However, the filter's response and full system response would change since they depend on the cutoff frequencies. Before describing the data analysis performed, the time responses of the signal's response to the heading input is observed in Figures 6 to 8b for a 350 second simulation.

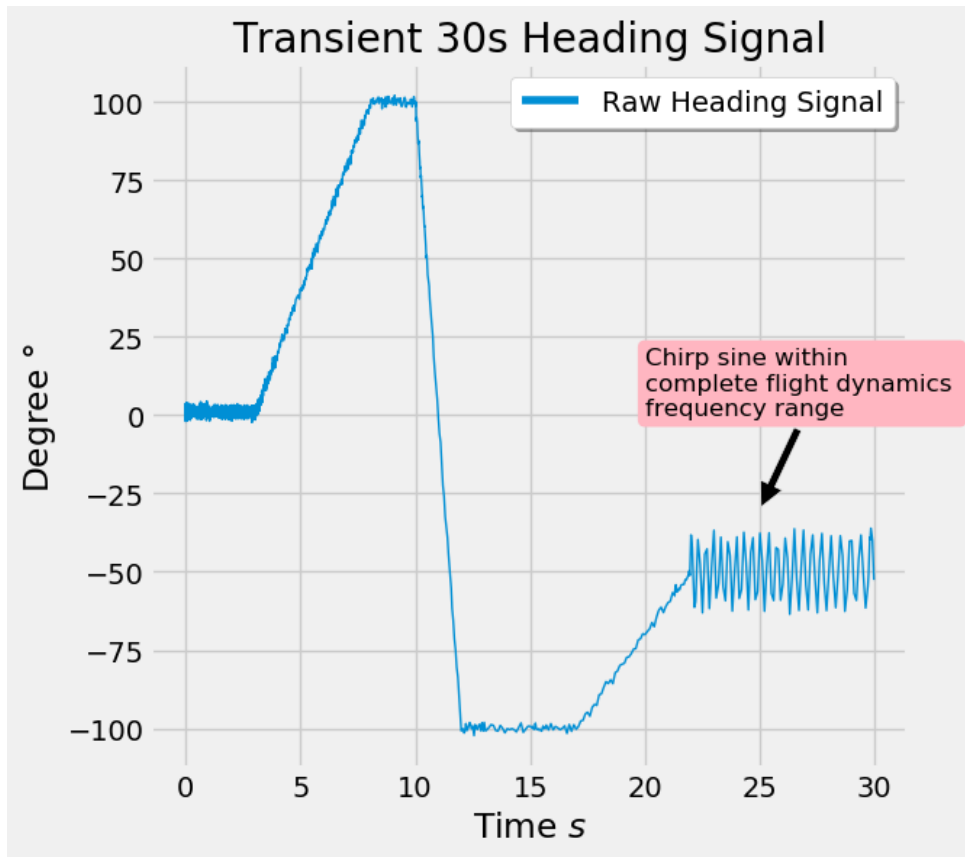


Figure 5: Initial 30s of the heading signal.

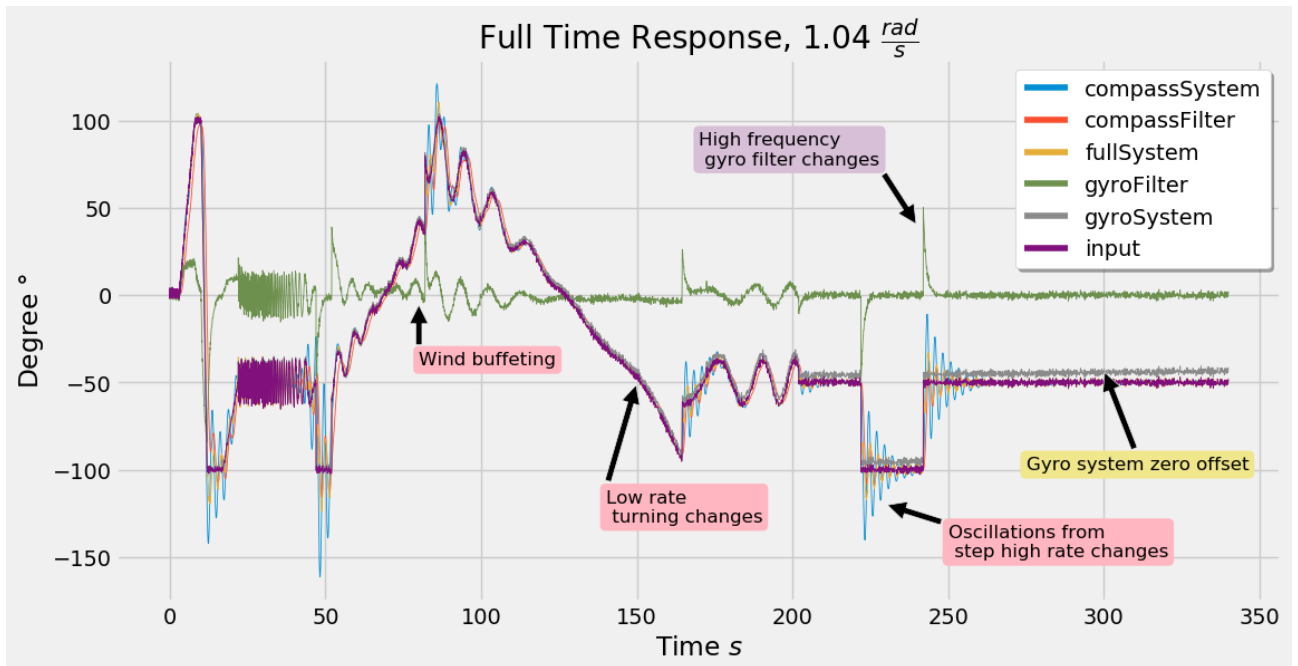


Figure 6: All signals response to the heading input signal.

From Figure 7 it can be noticed that the full system error is considerably smaller than the error of each individual sensor due to the complementary filters. It can even be noticed that at very high frequencies such as near second 50, even the gyro filter begins oscillating potentially due to magnitude attenuation described in Figure 2a.

The gyro filter output error mostly trails the inverse of the full system response in Figure 7. At most points within the heading flight, the gyro filter output value in Figure 6 is near zero. Its magnitude suddenly increases to respond to higher frequency changes such as observed by the 40s chirp sine, 100s wind buffeting, 175s turning

rates, or 225s step changes. This is expected due to the complementary filter mechanism that transfers all higher frequency excitation above the modelled $1.04 \frac{rad}{s}$ cutoff frequency.

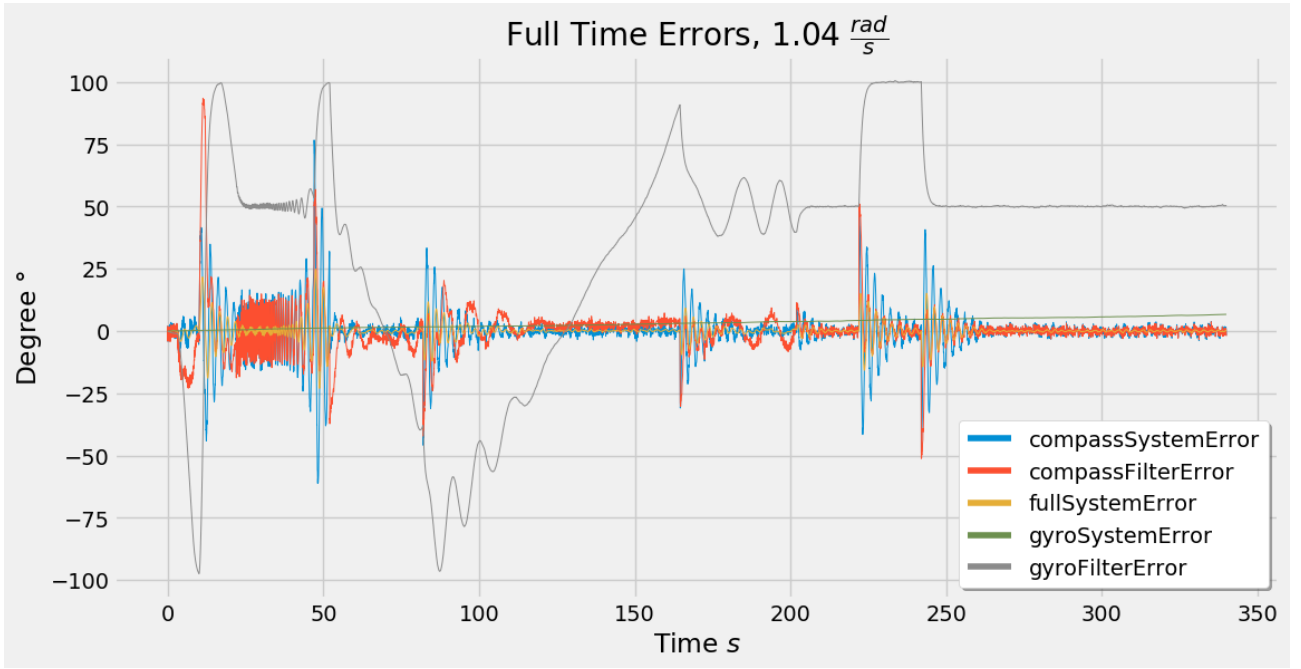
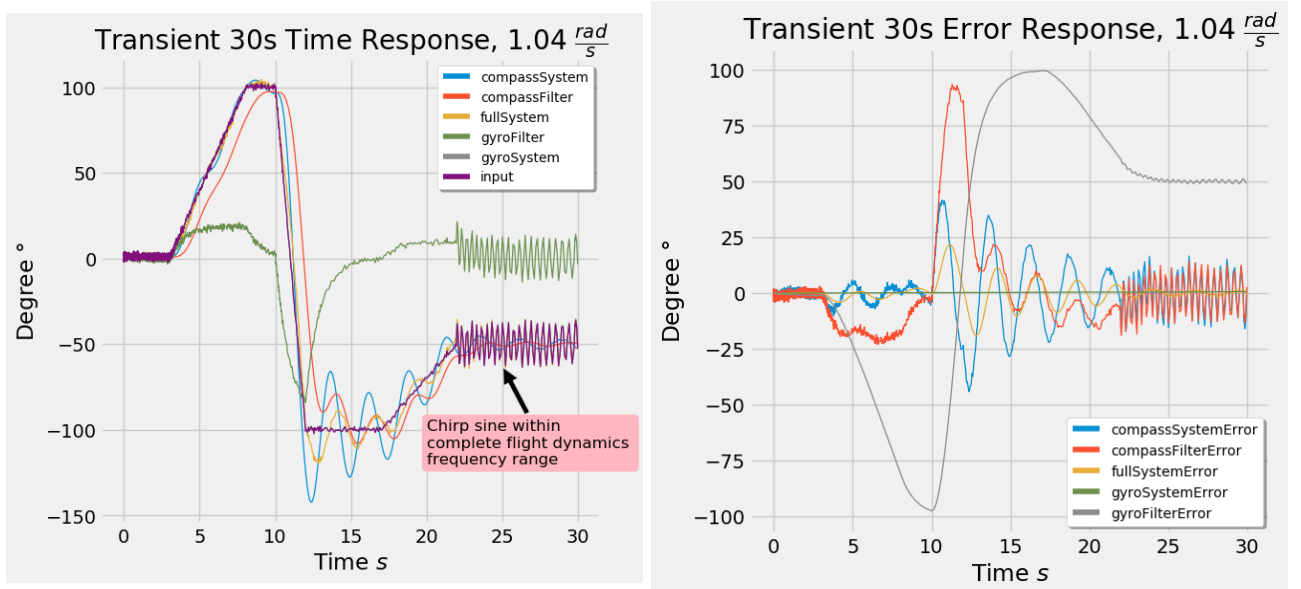


Figure 7: Error difference of each system signal for the full input heading signal.

The gyro system's zero offset from mechanical flight imperfections can be observed around 300 s, although for longer simulations that were run, it continuously increased which meant that high frequency oscillations were not accurately captured.

The compass filter and compass system trail the full system response at low frequencies, and these sensors oscillate at high frequency changes as observed in time 10s in Figure 8a. Note that the full system oscillations are much lower since the complementary filter design compensates at high frequencies and reduces the compass system input into the response.



(a) Initial 30s response of all system signals. Note that the input legend refers to the heading signal.

(b) Error with the heading input of all the system signals. Note that the input legend refers to the heading signal.

Figure 8: All signals initial 30s response.

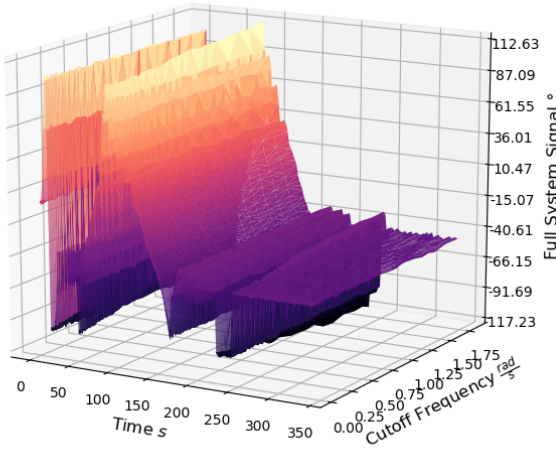
4 Error Quantification & Cutoff Frequency Variations

The cutoff frequency was then varied within a frequency range of 10^{-2} to $10^2 \frac{rad}{s}$ and the above statistics were generated for all of them as described in Appendix A.2. With the aim of characterizing the error of each sensor output, the statistical analysis was mainly interested in seeing the point where the full system signal error is minimal. However, varying the simulation time changed this optimal value. Hence, further analysis was done to practically determine the practical optimal cutoff frequency.

Since multidimensional data needed to be analysed, the Pandas Python data science package was used alongside Matplotlib 3D. Note that within the following analysis, exact values are always given since the random noise inputted in each heading signal simulation changes does not provide an identical response, but overall value ranges are noticed. This was done for the accuracy of attempting a real world heading signal simulation.

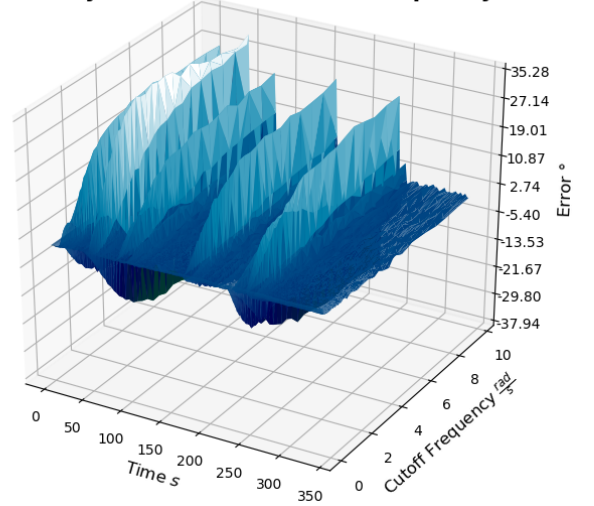
The three main sensor outputs that vary with the cutoff frequency are the gyro filter, compass filter and full system accordingly. Figures 9a to 10 describe provide a 3D plot representing how the time response and respective errors change with the variation of cutoff frequency. Whilst making the figures, a small section of the varied cutoff frequencies was selected to specifically demonstrate the changing behaviour. Figure 9a shows how the overall time response of the full system varies. Within the range of 0 to $1.75 \frac{rad}{s}$, it can be observed how response oscillations increase by comparing the initial amplitude of the time response to the final amplitude. To visualize this clearer, Figure 9 demonstrates the difference between the full response and the input heading within a larger cutoff frequency range and the increase is noticeable. This is expected since it would mean that the compass, whose bode plot starts becoming nonlinear between 1 and $2 \frac{rad}{s}$, begins responding to higher frequency oscillations.

Full System Response vs Cutoff Frequency



(a) Full system amplitude time response vs cutoff frequency.

Full System Error vs Cutoff Frequency

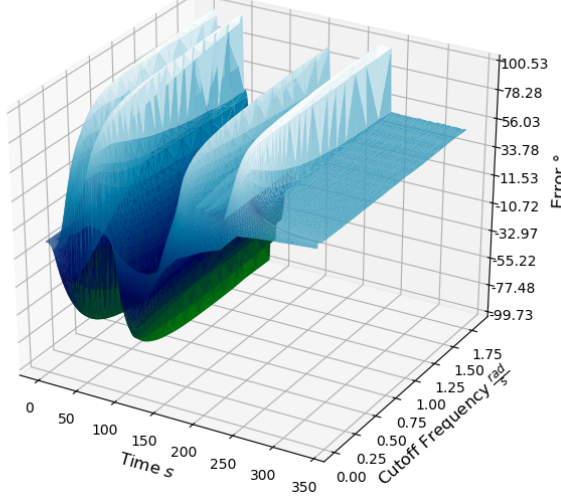


(b) Full system error variation in time

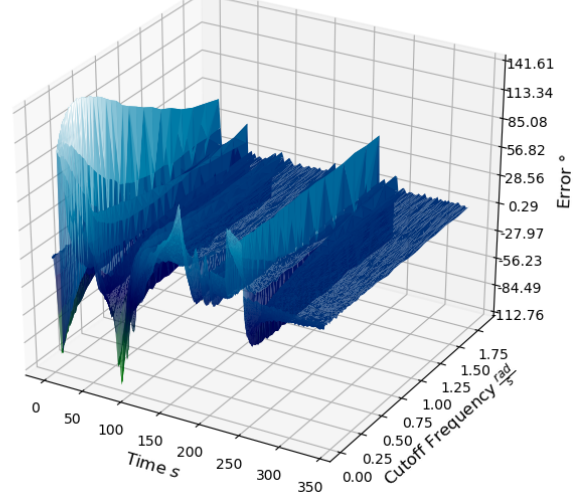
Figure 9: Full system error time response vs cutoff frequency. Note the difference in scales to adequately represent changes.

Figure 10a also demonstrates an interesting behaviour within a small cutoff frequency range of 0 to $1.75 \frac{rad}{s}$. The gyro filter error response magnitude reaches steady state from $1.75 \frac{rad}{s}$ to the rest of the cutoff frequencies, which is why only this range was modelled. The transient variation of the magnitude of the error in this graph within 0 and $1.25 \frac{rad}{s}$ could be considered as differential change due to cutoff frequency change in the form $\frac{d(GyroError)}{d(CutoffFrequency)}$. The steady state value is reached around $1 \frac{rad}{s}$, and from this point on-wards the $\frac{d(GyroError)}{d(CutoffFrequency)} \approx 0$. Considering this as a uni-variate optimization problem, it could be said that this is the optimal cutoff point since the compass is designed to operate adequately at lower frequencies. Also observing Figure 10, a $1 \frac{rad}{s}$ compass cutoff frequency has relatively low error values in comparison to the rest of its error magnitude response after exponentially decreasing from a maximum near $0 \frac{rad}{s}$. Hence, even in 3D space, the behaviour of complementary filter errors are demonstrated also when steady state values are reached. However, to fully discuss this, statistical time domain analysis needs to be performed.

Gyro Filter Error vs Cutoff Frequency



Compass Filter Error vs Cutoff Frequency



(a) Gyro filter error variation with cutoff frequency over time. (b) Compass filter error variation with cutoff frequency over time.

Figure 10: Compass filter error variation with cutoff frequency over time.

A statistical metric cannot be chosen at random to quantify this error since all metrics provide different information about the signal data distribution. A comprehensive overall error metric might be a combination of different data statistics. In order to adequately select the error metric, or combined error metric, the following analysis were performed for both the raw signals and error sets: maximum, minimum, mean, standard deviation, variance, skewness, kurtosis, moments from 1 to 4, RMS, and 3dB power bandwidth (not for errors).

Figures 12 provides a sample of statistical metrics for all the signal errors whilst changing the cutoff frequency. In order to minimize the magnitude of the full system error, ideally its magnitude should be regularly near zero. However, realistically, because the incremental zero offset errors introduced into the gyro compass design that cannot be corrected via frequency filtering, the ideal value of the full system is when it matches the gyro system offset error within this model. It was noted that by varying the simulation time, the time-domain statistical optimal parameters vary; given the larger effect of the gyro system offset error. Hence, even if the time domain statistical parameters varied, what is required is selecting a cutoff frequency that is in steady state of the slope $\frac{d(\text{SignalResponse})}{d(\text{cutoffFrequency})} = 0$. At that point, the compass filter output is still at its maximum and the gyro filter output will have its standard zero-offset expected error. Hence, the response of the compass filters is not minimized specially when this is used to calibrate the gyro in flight during steady conditions. However, further time-domain statistical analysis will be performed in order to compare the optimal low statistic points for a specific simulation flight length with the steady cutoff frequency that is independent of this. This is because whilst the error magnitude may increase the overall zero offset rate is constant since it is not coupled mathematically as a function of the change of the cutoff frequency seen in Figure 9, only the filters are.

The effect of varying cutoff frequencies on the amplitude of the response can be primarily observed in Figures 11a and 11b. If the cutoff frequency induces more oscillations, this will lead to larger differences within the response. It can be observed between the 10^1 to 10^2 regions that the full system response error stabilizes at a minima of around -60 and maxima of around 80. By the shape of the graphs, it can be inferred that oscillations differences are larger when the cutoff frequency is larger. This is expected since the compass filter would mainly respond to most flight oscillations, despite being optimized for lower frequencies.

The first aspect that will be analysed from the error differences are the standardized data moments which basically identify the crucial peak points within the data.

The mean ($\propto 1^{st} \text{Moment}$) is $\mu = \frac{1}{N} \sum_{i=1}^N x_i$, where N is the time domain signal magnitude sample points. The mean is represented in Figure 11c, but does not fully account for transient responses. However, it provides a general indication that the overall response averaged error $\frac{d(\text{FullSystemError})}{d(\text{CutoffFrequency})}$ stabilizes to 0.1° at a 0.13 rad/s . However, the mean has already reached a near zero value before the 10^{-1} . Potentially because the system has not run for long enough, the mean is near 0 and has not stabilized yet to the gyro system error. It is also worth noting that there is a near-inflection point for both the compass (blue) and gyro (green) filter error plots at this point, which is related to a cutoff frequency change. Hence, the mean might not be an accurate error metric for the overall system error minimization response.

The standard deviation ($\propto 2^{nd} \text{Moment}$) for each cutoff frequency is $\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$ where X_i is a time domain sample point, and this illustrates how large the distribution of overall errors is from the averaged

value in Figure 11d. Hence, for a larger σ is, it means that X_i varied errors can be drastically different from the overall averaged error. It could also mean that oscillations could be very large but quick. When the objective is to minimize both transient and overall errors, it is desired that the overall error magnitude value and distribution of errors to be near zero. The full system has an minimum point where the distribution is near 0 at 0.048 rad/s . Again there appears to be an inflection point at this value for the compass and gyro filters. However, since it is known that practically that minimum is not the ideal value due to the increasing zero-offset observed during large simulation times, it can also be observed that at about 1 rad/s the gyro filter error and compass filter errors reach a steady state value. The full system plot intersects also with the gyro system error plot. This was expected from observing their time responses in Figure 10.

Again regarding the full system signal, the skewness ($\propto 3^{rd} \text{Moment}$) and kurtosis ($\propto 4^{th} \text{Moment}$) in Figures 11e & 11f provide understanding of the asymmetrical accentuation of the probabilistic distribution of data around a overall mean value. The interesting aspect is that both signals have a maximum at the values around 0.048 rad/s . They also seem to reach steady state values slightly before or after 10^1 rad/s as expected from previous moments.

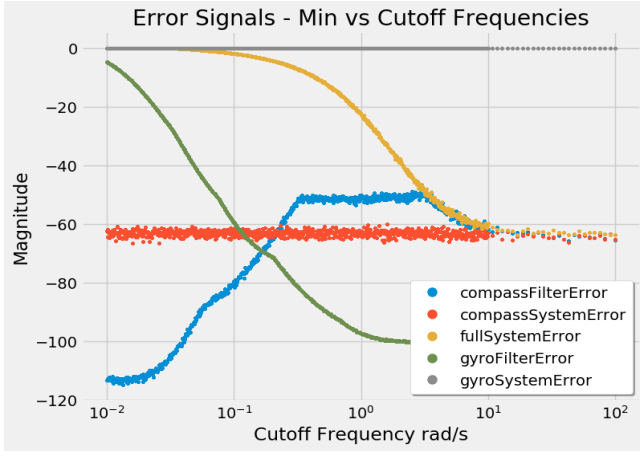
When characterizing the gyro system error, it is known that there will be zero offset as mechanical forces destabilize it during the heading flight. At its latest position, it has a 8.66° difference for 350s heading flight. Assuming a linear slope, this offset error will only increase with longer flights, its slope in this simulation is $\approx 0.025^\circ/\text{s} \approx 1.5^\circ/\text{min}$ as seen in Figure 6 which validates our design goal. Its standard deviation remains constant around 2.13 since the only changes are from a few high frequency turning rates that lead to higher errors (as observed) in Figure 7.

The root mean squared RMS of a function $g(t)$ is $g_{RMS} = \lim_{T \rightarrow \infty} \sqrt{\frac{1}{T} \int_0^T [g(t)]^2 dt}$. where T is the number of sample points of the response function $g(t)$. Because the mean of all the squared magnitudes of the signal are considered, it provides a general indication of the “energy” value of the wave. It can also be noted that there is a minima around 0.048 rad/s . This full system error has an inflection point around 1 rad/s where it intersects with the gyro system error probably due to the inverse stabilization of the the gyro and compass filter magnitudes.

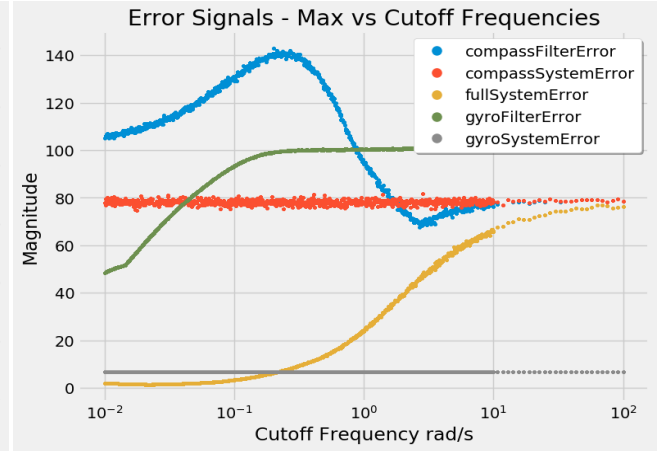
Hence, an error metric for each sensor is required to be derived. It is known that the RMS and standard deviation are the most accurate statistics to determine the optimal cutoff frequency at the intersection of the gyro system error and the full system error. The error with any other cut-off frequency can be determined by the absolute difference between the full system error and gyro system error accordingly. Also, because the gyro filter and compass filter behave nearly inversely, as in Figure 11d, the difference between the two can be compared to an optimal difference in magnitude of 32.52 for the standard deviation and 37.63 for the RMS. Since there is an inflection point for both error signals, this can be related to the ideal minimum for that specific flight time and this point will move towards the right as the flight time is increased. Hence, a geometrical relationship can be derived from these graphs. Integrating this analysis together constraining it to give a value between 0 and unity of error can be written in Equation 7, where E_m is the error metric, $\Delta\sigma$ is the difference in the standard deviation for that cut-off frequency w_c , and $\Delta\kappa$ is the difference in RMS values. The closer it is to 0, the cutoff frequency is closer to the ideal one.

$$E_m(w_c) = \frac{32.52\Delta\sigma}{37.63} + \frac{37.63\Delta\kappa}{32.52} \quad (7)$$

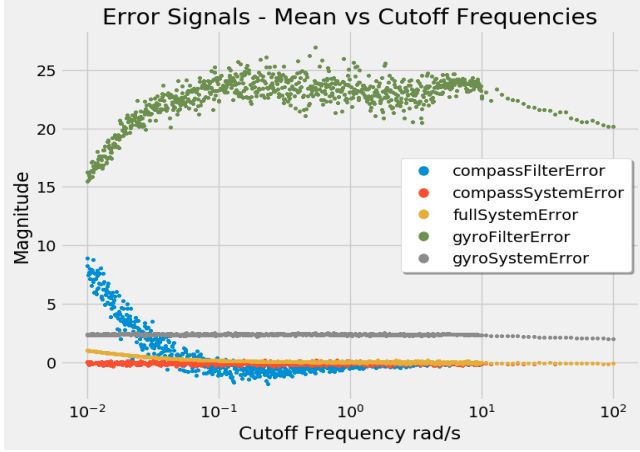
This metric can be compared to different heading flights and possibly the ideal difference values can be calibrated after testing over a further range of frequencies and simulation time. In summary, multi-dimensional Big Data time statistical analysis was used to determine the optimal design parameters of a sensor fusion system.



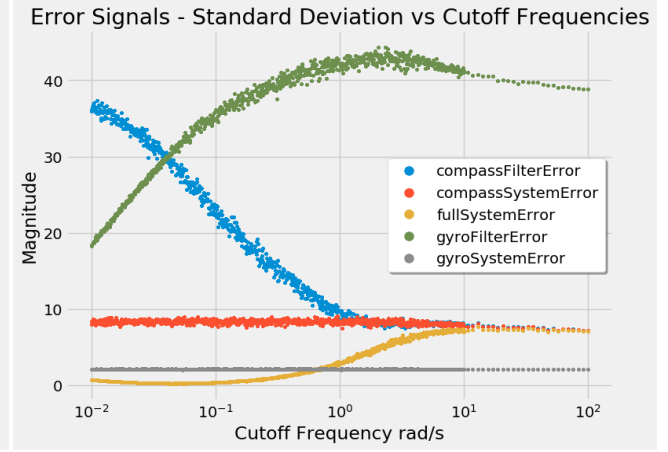
(a) Signal errors minimum value.



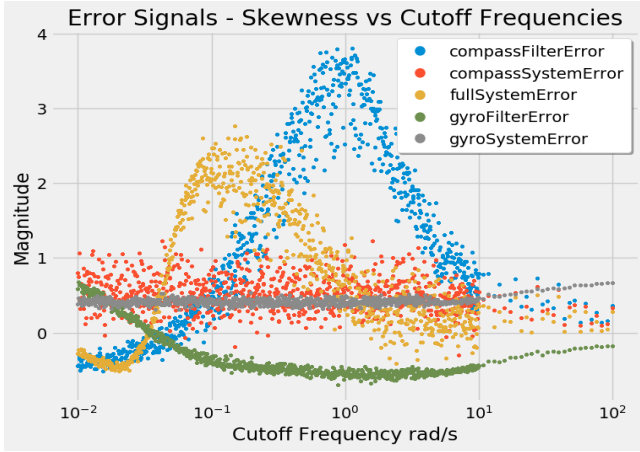
(b) Signal errors maximum value.



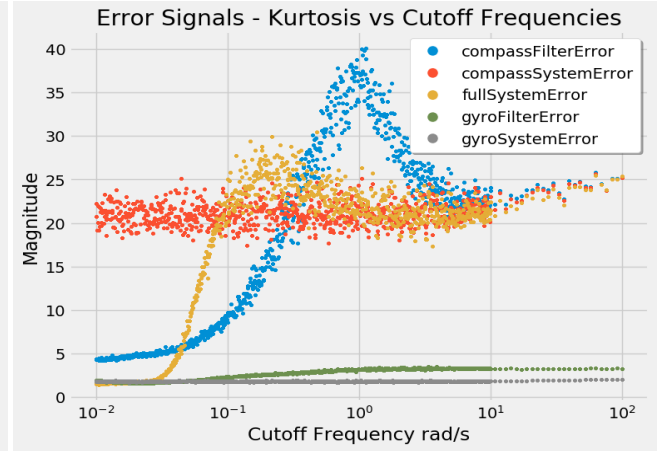
(c) Signal errors mean value ($\propto 1^{st}$ Moment)



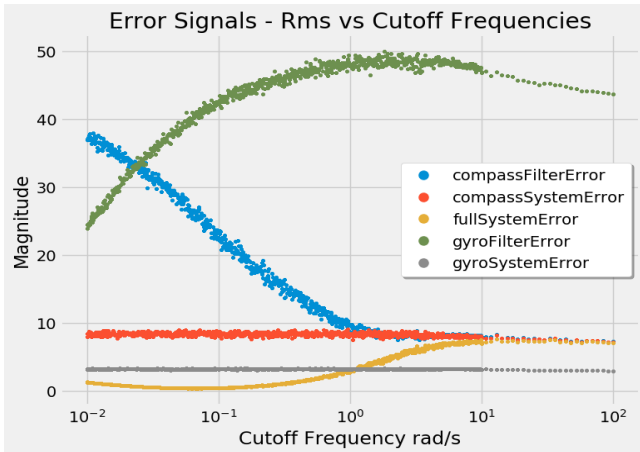
(d) Signal errors standard deviation. ($\propto 2^{nd}$ Moment)



(e) Signal errors skewness. ($\propto 3^{rd}$ Moment)



(f) Signal errors kurtosis ($\propto 4^{th}$ Moment)



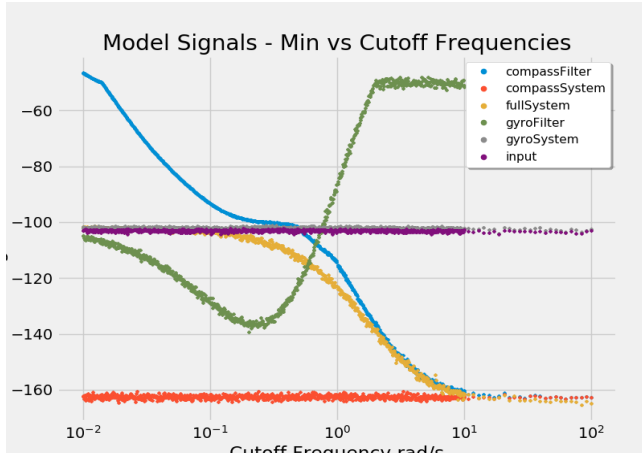
(g) Signal error RMS (Energy).

References

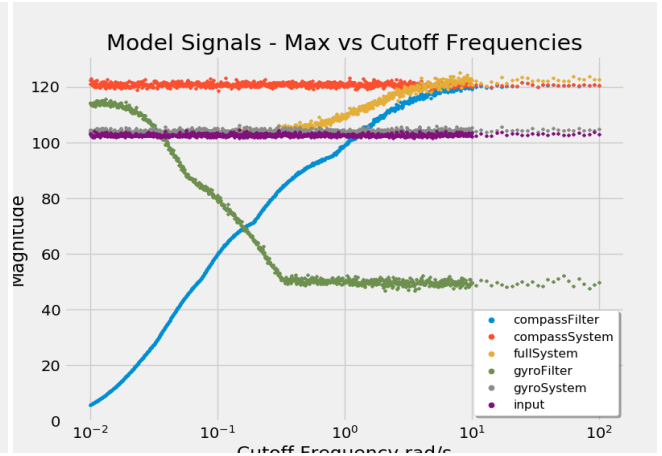
- [1] Sergio De La Parra Carqué. *Sensor fusion system and method for estimating position, speed and orientation of a vehicle, in particular an aircraft*. US Patent 7,970,500. June 2011.
- [2] Andrei Dorobantu et al. “Frequency domain system identification for a small, low-cost, fixed-wing uav”. In: *AIAA guidance, navigation, and control conference*. 2011, p. 6719.
- [3] Eric Foxlin. “Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter”. In: (1996).
- [4] Robert W Levi and Robert R Marshall. *Gyro aided magnetic compass*. US Patent 6,842,991. Jan. 2005.
- [5] Alan S. Morris and Reza Langari. “Chapter 20 - Rotational Motion Transducers”. In: *Measurement and Instrumentation (Second Edition)*. Ed. by Alan S. Morris and Reza Langari. Second Edition. Boston: Academic Press, 2016, pp. 599–632. ISBN: 978-0-12-800884-3. DOI: <https://doi.org/10.1016/B978-0-12-800884-3.00020-4>. URL: <http://www.sciencedirect.com/science/article/pii/B9780128008843000204>.
- [6] *Navigation system design using time-varying complementary filters*. 2020. URL: <https://www.dji.com/uk/mavic-2/info#specs>.
- [7] Katsuhiko Ogata and Yanjuan Yang. *Modern control engineering*. Vol. 17. Pearson Upper Saddle River, NJ, 2010.
- [8] John S Youngquist. *Rate gyro error correction and use as heading source*. US Patent 5,654,906. Aug. 1997.

A Appendix

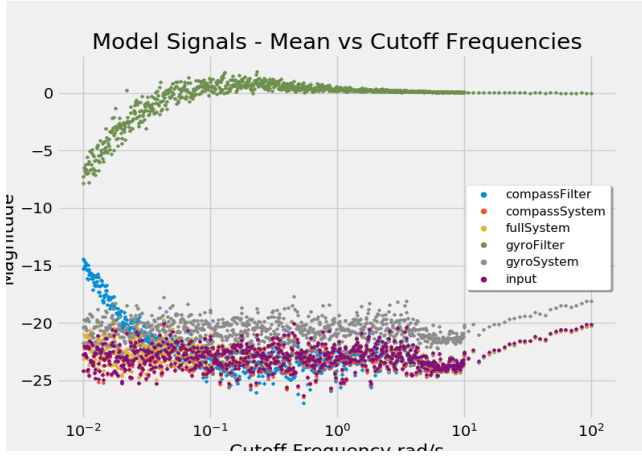
A.1 Signals Time Domain Statistical Analysis



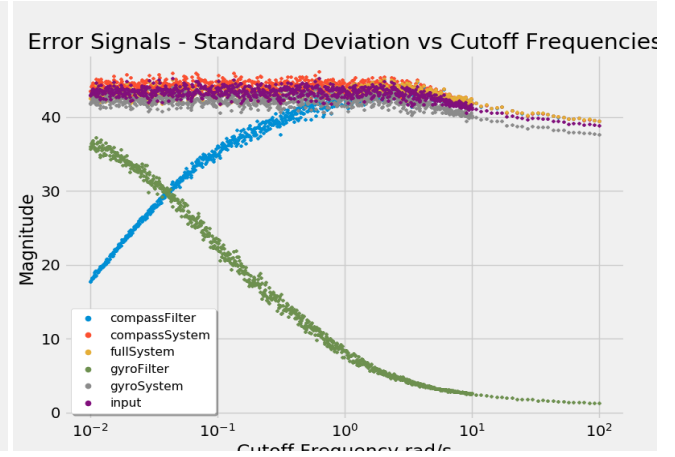
(a) Signals minimum value.



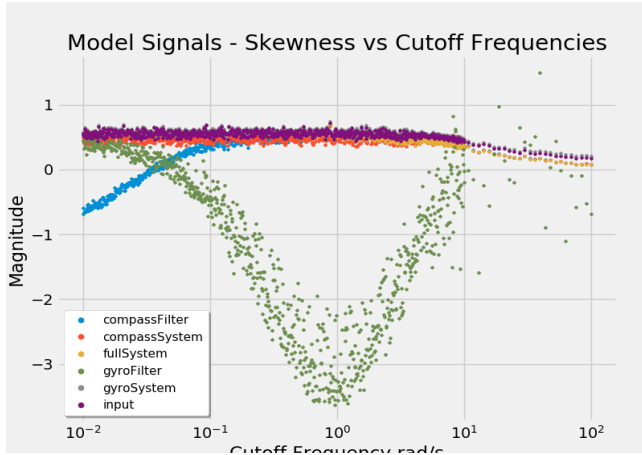
(b) Signals maximum value.



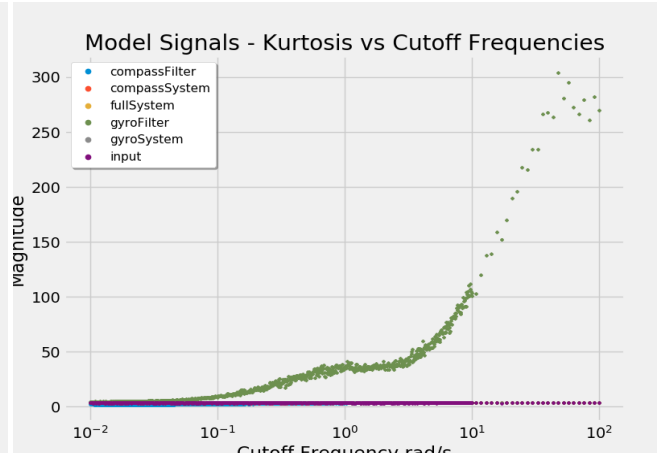
(c) Signals mean value ($\propto 1^{st}$ Moment)



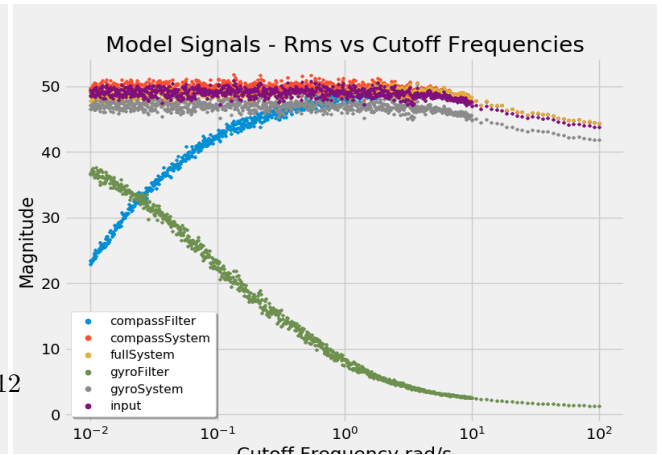
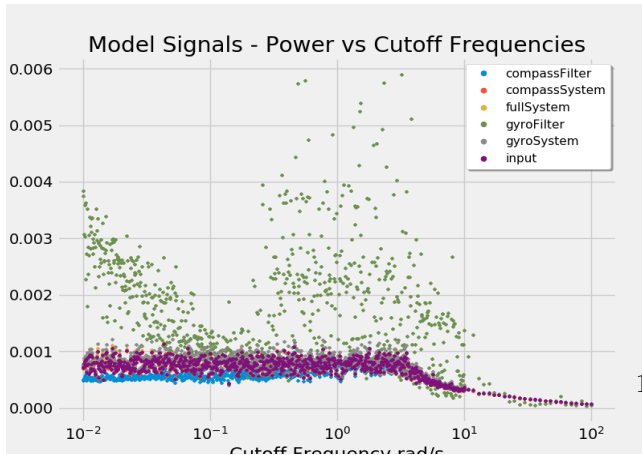
(d) Signals standard deviation. ($\propto 2^{nd}$ Moment)



(e) Signals skewness. ($\propto 3^{rd}$ Moment)



(f) Signals kurtosis ($\propto 4^{th}$ Moment)



A.2 Cutoff Frequency Variation Code

```
% This code models the variation of cutoff frequencies and the respective
% steady state input response (True Heading)

% Create standard model parameters
run("./compassControlParameters.m");
run("./gyroControlParameters.m");
testInput = "trueHeading";

% Data save folder location
saveFolderLocation = "analytics/cutoffVariations/standard_dataset/";
% Vary the cutoff frequencies over the attenuation range of the gyroFiler.
varyingCutoffFrequencies = logspace(-2, 2, 100);

% Compass Constant Parameters
compassFilterGain = 1;

p = 1;
% Iterating Parameters and Simulations
for cutoffFrequencyIteration = varyingCutoffFrequencies
    % Compass Filter Variation
    compassTimeConstant = 1/cutoffFrequencyIteration; % Rad
    compassFilterNumerator = compassFilterGain;
    compassFilterDenominator = [compassTimeConstant 1];
    compassFilterTransferFunction = tf(compassFilterNumerator, compassFilterDenominator);

    % Gyro Filter Variation
    gyroTimeConstant = compassTimeConstant;
    gyroFilterNumerator = [1 0];
    gyroFilterDenominator = [1 1/gyroTimeConstant];
    gyroFilterTransferFunction = tf(gyroFilterNumerator, gyroFilterDenominator);

    %% SIMULATION
    simulation = sim('fullSystem', 'SimulationMode', 'normal');
    % Get results
    simulationData = simulation.get('simulationData');
    assignin('base', 'simulationData', simulationData);

    % Simulation Data
    time = simulationData.time;
    signalsNamesOrdered = ["compassSystem", "compassFilter", ...
        "fullSystem", "gyroFilter", "gyroSystem", "input", ...
        "compassSystemError", "compassFilterError", ...
        "fullSystemError", "gyroSystemError", "gyroFilterError"...
    ].';
    compassSystem = simulationData.signals.values(:, 1);
    compassFilter = simulationData.signals.values(:, 2);
    fullSystem = simulationData.signals.values(:, 3);
    gyroFilter = simulationData.signals.values(:, 4);
    gyroSystem = simulationData.signals.values(:, 5);
    input = simulationData.signals.values(:, 6);
    % Errors
    fullSystemError = fullSystem - input;
    compassSystemError = compassSystem - input;
    compassFilterError = compassFilter - input;
    gyroSystemError = gyroSystem - input;
    gyroFilterError = gyroFilter - input;
    allData = [simulationData.signals.values(:, :), ...
        compassSystemError, ...
        compassFilterError, ...
```

```

        fullSystemError, ...
        gyroSystemError, ...
        gyroFilterError];

rawDataTable = table(...
    time,...
    compassSystem,...
    compassFilter,...
    fullSystem,...
    gyroFilter,...
    gyroSystem,...
    input,...
    compassSystemError, ...
    compassFilterError, ...
    fullSystemError, ...
    gyroSystemError, ...
    gyroFilterError ...
);
writetable(rawDataTable, saveFolderLocation...
+ testInput...
+ regexprep(string(cutoffFrequencyIteration),'\.','_')...
+ 'Raw.csv');

%      % Plot
%      figure
%      plot(time, compassSystem, time, compassFilter, time, fullSystem,...
%           time, gyroFilter, time, gyroSystem, time, input, time, fullSystemError)
%      title(num2str(cutoffFrequencyIteration))
%      legend(signalsNamesOrdered)
%      savefig(saveFolderLocation...
%           + testInput...
%           + regexprep(string(cutoffFrequencyIteration),'\.','_')...
%           + 'Fig.fig')

%% Correlations
compassSystemFullSystemErrorCrossCorrelation = xcorr(fullSystemError, compassSystem);
compassFilterFullSystemErrorCrossCorrelation = xcorr(fullSystemError, compassFilter);
fullSystemFullSystemErrorCrossCorrelation = xcorr(fullSystemError, fullSystem);
gyroFilterFullSystemErrorCrossCorrelation = xcorr(fullSystemError, gyroFilter);
gyroSystemFullSystemErrorCrossCorrelation = xcorr(fullSystemError, gyroSystem);
% Errors
compassSystemErrorFullSystemErrorCrossCorrelation = xcorr(fullSystemError, compassSystemError);
compassFilterErrorFullSystemErrorCrossCorrelation = xcorr(fullSystemError, compassFilterError);
fullSystemErrorFullSystemErrorCrossCorrelation = xcorr(fullSystemError, fullSystemError);
gyroSystemErrorFullSystemErrorCrossCorrelation = xcorr(fullSystemError, gyroSystemError);
gyroFilterErrorFullSystemErrorCrossCorrelation = xcorr(fullSystemError, gyroFilterError);

correlationData = [compassSystemFullSystemErrorCrossCorrelation,...
    compassFilterFullSystemErrorCrossCorrelation,...
    fullSystemFullSystemErrorCrossCorrelation, ...
    fullSystemFullSystemErrorCrossCorrelation,...
    gyroFilterFullSystemErrorCrossCorrelation, ...
    gyroSystemFullSystemErrorCrossCorrelation,...
    compassSystemErrorFullSystemErrorCrossCorrelation, ...
    compassFilterErrorFullSystemErrorCrossCorrelation, ...
    fullSystemErrorFullSystemErrorCrossCorrelation, ...
    gyroSystemErrorFullSystemErrorCrossCorrelation, ...
    gyroFilterErrorFullSystemErrorCrossCorrelation...
    ];

```



```

correlationsDataTable = table(...
    compassSystemFullSystemErrorCrossCorrelation,...
    compassFilterFullSystemErrorCrossCorrelation,...
    fullSystemFullSystemErrorCrossCorrelation,...
    gyroFilterFullSystemErrorCrossCorrelation,...
    gyroSystemFullSystemErrorCrossCorrelation,...
    compassSystemErrorFullSystemErrorCrossCorrelation, ...
    compassFilterErrorFullSystemErrorCrossCorrelation, ...
    fullSystemErrorFullSystemErrorCrossCorrelation, ...
    gyroSystemErrorFullSystemErrorCrossCorrelation, ...
    gyroFilterErrorFullSystemErrorCrossCorrelation);
writetable(correlationsDataTable, saveFolderLocation...
+ testInput...
+ regexprep(string(cutoffFrequencyIteration), '\.', '_')...
+ 'Correlations.csv');

% figure
% stackedplot(correlationsDataTable)
% title(num2str(cutoffFrequencyIteration))
% savefig(saveFolderLocation...
% + testInput...
% + regexprep(string(cutoffFrequencyIteration), '\.', '_')...
% + 'CorrelationFig.fig')

%% Signals Processing
% Raw Signals
meanSignals = mean(allData(:, :)).';
standardDeviationSignals = std(allData(:, :)).';
varianceSignals = var(allData(:, :)).';
maxSignals = max(allData(:, :)).';
minSignals = min(allData(:, :)).';
kurtosisSignals = kurtosis(allData(:, :)).';
firstMomentSignals = moment(allData(:, :), 1).';
secondMomentSignals = moment(allData(:, :), 2).';
thirdMomentSignals = moment(allData(:, :), 3).';
fourthMomentSignals = moment(allData(:, :), 4).';
skewnessSignals = skewness(allData(:, :)).';
rmsSignals = rms(allData(:, :)).';
powerSignals = powerbw(allData(:, :)).';
% Correlations
maxCorrelationSignals = max(correlationData).';
meanCorrelationSignals = mean(correlationData).';
standardDeviationCorrelationSignals = std(correlationData).';
varCorrelationSignals = var(correlationData).';
kurtosisCorrelationSignals = kurtosis(correlationData).';
powerCorrelationSignals = powerbw(correlationData).';
rmsCorrelationSignals = rms(correlationData).';

%% Display Analytics
analyticsTable = table(signalsNamesOrdered, maxSignals, minSignals, meanSignals,...
    standardDeviationSignals, varianceSignals, kurtosisSignals,...
    skewnessSignals, firstMomentSignals, secondMomentSignals,...
    thirdMomentSignals, fourthMomentSignals, rmsSignals,...
    powerSignals, maxCorrelationSignals, meanCorrelationSignals,...
    standardDeviationCorrelationSignals, varCorrelationSignals, kurtosisCorrelationSignals,...
    powerCorrelationSignals, rmsCorrelationSignals)
writetable(analyticsTable, 'analytics/cutoffVariations/longHeading/'...
+ testInput...
+ regexprep(string(cutoffFrequencyIteration), '\.', '_')...

```

```

        + 'Analytics.csv');
    p = p + 1
%     if p > 100
%         close all
%         p=0;
%     end
end
! git add * && git commit -am "Ran long heading" && git push origin master
close all

```

A.3 Heading Signal MATLAB System Block

```

classdef HeadingSignals < matlab.System
    % This are the heading signals class.
    % It implements a number of functions to test the response behaviour
    %
    % This template includes the minimum set of functions required
    % to define a System object with discrete state.

    % Public, tunable properties
    properties

    end

    properties(DiscreteState)
    end

    % Pre-computed constants
    properties(Access = public)
    end

    methods(Access = protected)
        function setupImpl(obj)
            % Perform one-time calculations, such as computing constants
        end

        function [step, ramp, sine, trueHeading] = stepImpl(obj, time)
            step = generateStep(obj, time);
            ramp = generateRamp(obj, time);
            sine = generateSine(obj, time);
            trueHeading = generateTrueHeading(obj, time);
            % sawtoothWave = generateSawtooth(obj, time);
            % turn180 = generate180Turn(obj, time, turnTime);
        end

        function resetImpl(obj)
            % Initialize / reset discrete-state properties
        end
    end

    methods(Access = private)
        function step = generateStep(obj, time)
            % Unit step Input
            step = ones(1, length(time));
        end

        function ramp = generateRamp(obj, time)
            % Ramp input
            ramp = time;
        end
    end
end

```

```

function sine = generateSine(obj, time)
    % Ramp input
    sine = sin(time);
end

% function lowRateTurn = generateLowRateTurn(obj, time, period)
%
% end
%
% function sawtoothWave = generateSawtooth(obj, time, period)
%     sawtoothWave = sawtooth(time);
% end

end

methods(Access = public)

function trueHeading = generateTrueHeading(obj, time)
    maximumTurnRate = 200;
    % Long Chirp Period
    longChirpStartFrequency = 0.001;
    longChirpEndFrequency = 1;
    longChirpPeriod = 150;
    longChirpConstant = (longChirpEndFrequency - longChirpStartFrequency) / longChirpPeriod;

    % Short Chirp
    chirpStartFrequency = 0.001;
    chirpEndFrequency = 20;
    chirpPeriod = 25;
    chirpConstant = (chirpEndFrequency - chirpStartFrequency) / chirpPeriod;

    % Time limits
    step1Time = 3;
    transition12EndTime = step1Time + 5; % 8
    step2Time = transition12EndTime + 2; % 10
    transition23EndTime = step2Time + 2; % 12
    step3Time = transition23EndTime + 5; % 17
    transition34EndTime = step3Time + 5; % 22
    step4Time = transition34EndTime + chirpPeriod; % 47
    step5Time = step4Time + 5;
    step6Time = step5Time + longChirpPeriod;

    % State at those specific times
    step1State = 1;
    step2State = maximumTurnRate/2;
    transitionState12 = (time - step1Time) * (step2State - step1State) / (transition12EndTime - step1Time);
    step3State = -maximumTurnRate/2;
    transitionState23 = (time - step2Time - 1) * (step3State - step2State) / (transition23EndTime - step2Time);
    step4State = -maximumTurnRate/4;
    transitionState34 = (time - step3Time - 10) * (step4State - step3State) / (transition34EndTime - step3Time);
    step5State = -maximumTurnRate/2;
    variableState = step4State;

    if time <= step1Time
        trueHeading = step1State;
    elseif (time > step1Time) && (time <= transition12EndTime)
        trueHeading = transitionState12;
    elseif (time > transition12EndTime) && (time <= step2Time)
        trueHeading = step2State;
    elseif (time > step2Time) && (time <= transition23EndTime)
        trueHeading = transitionState23;
    elseif (time > transition23EndTime) && (time <= step3Time)
        trueHeading = step3State;
    elseif (time > step3Time) && (time <= transition34EndTime)
        trueHeading = transitionState34;
    elseif (time > transition34EndTime) && (time <= step4Time)
        trueHeading = step4State;
    elseif (time > step4Time) && (time <= step5Time)
        trueHeading = step5State;
    elseif (time > step5Time) && (time <= step6Time)
        trueHeading = step6State;
    else
        trueHeading = step6State;
    end
end

```

```

elseif (time > transition23EndTime) && (time <= step3Time)
    trueHeading = step3State;
elseif (time >= step3Time) && (time < transition34EndTime)
    trueHeading = transitionState34;
elseif (time >= transition34EndTime) && (time < step4Time)
    trueHeading = step4State/4 * sin(((chirpConstant * (time-step4Time) ^ 2 / 2 +...
        chirpStartFrequency *(time-step4Time)))) + step4State;
elseif (time >= step4Time) && (time < step5Time)
    trueHeading = step5State;
elseif (time >= step5Time) && (time < step6Time)
    if (time < step5Time + 30)
        variableState = step4State + 3 * (time - step5Time);
    elseif (time >= step5Time + 30) && (time <= step5Time + 3/4 * longChirpPeriod)
        variableState = (variableState - 2 * (time - step5Time) + 200);
    end
    trueHeading = variableState/4 * sin(((2 * longChirpConstant * (time-longChirpPeriod) ^
        chirpStartFrequency *(time-step5Time)))) + variableState;
elseif (time >= step6Time + 20 ) && (time < step6Time + 40)
    trueHeading = step5State;
else
    trueHeading = step4State;
end
trueHeading = trueHeading + random("normal", 0, maximumTurnRate * 0.005);
end
end
end

```