State estimation of Quadcopter using EKF with stronger emphasis on position

Ludwig Horvath

January 2024

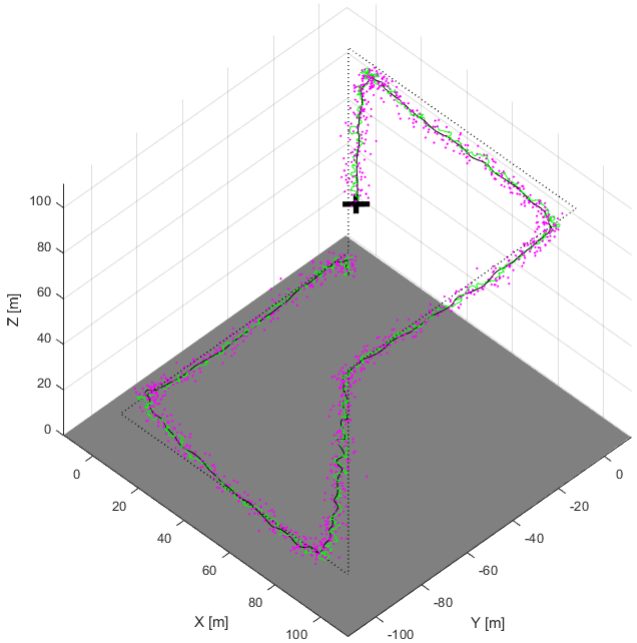


Figure 1: A simulation of the quadcopter following a 100x100x100 [m] "Box" path for a duration of 200 seconds using the EKF for estimating its position in the inertial frame.

Applied Estimation EL2320
Robotics System & Control
Date: 2024-1-14
Examiner: John Folkesson
Project Colleague: Martin Petre

1 Introduction

The objective of this project is to design and tune an Extended Kalman Filter (EKF) tailored for Unmanned Aerial Vehicles (UAVs), specifically focusing on a quadcopter. The successful implementation demonstrates the filter's capability to 'quite accurately' estimate the quadcopter's position relative to an inertial frame, considering various influencing factors. EKF's adaptability to nonlinear dynamic processes is showcased, assuming noise, especially around the operating point, where it in theory is to be considered as a unimodal Gaussian distribution.

In this application, the EKF fuses sensor data to estimate the quadcopter's position and velocity in an inertial frame, conceptualized as a local tangential plane (LTP). Although all twelve states are estimated and filtered, some are considered "perfect." Building upon a prior unsuccessful project conducted in spring 2023 by the author and his colleague [1], this project specifically addresses issues related to states targeted aggressively by the EKF. The filter corrects the errors produced during the propagation of certain states thus removing drift. Several other states are directly measured, introducing simplicity within the measurement model.

While the GPS provides latitude and longitude, unreliable altitude measurements necessitate the use of an altimeter. Altitude estimation relies on a built-in sensor system utilizing pressure measurements to determine altitude relative to an initialization point where also the temperature is measured. Assuming an existing conversion algorithm, longitude and latitude are obtained with acceptable accuracy. However, analytically converting these coordinates poses challenges, prompting the adoption of a parameter estimation technique alongside existing WGS84 spheroid-based conversions, particularly in local contexts.

To address the conversion complexity, GPS measurements are synthetically generated by sampling the quadcopter's true position, eliminating bias, and introducing Gaussian noise. These converted measurements are incorporated into the filter, granting them an identity within the measurement model, not because they are inherently perfect, but due to previously stated project constraints.

Although not all challenges were resolved, significant theoretical advancements were achieved. The states are more accurately estimated, expanding the scope of viable control strategies. Rigorous efforts were invested in visualization and quantification of performance, enabling an iterative and expeditious tuning process. The project's findings and software, available on a public GitHub repository, are referenced at the end of this report, providing valuable insights for future work or as educational material.

2 Mathematical Formulation

The challenge in terms of mathematical derivations is outlined by the EKF algorithm as stated below [5].

Algorithm 1: EKF Algorithm(μ_{t-1}, Σ_{t-1}, u_t, z_t)

1: $\bar{\mu}_t = g(u_t, \mu_{t-1})$
2: $\bar{\Sigma}_t = G_t \Sigma_{t-1} {G_t}^T + R^T$
3: $K_t = \bar{\Sigma}_t {H_t}^T (H_t \bar{\Sigma}_t {H_t}^T + Q_t)^{-1}$
4: $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$
5: $\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$
6: **return** Σ_t, μ_t

This algorithm is restated from the displayed reference. Further details are provided in [5], to understand the algorithm the reader is adviced to read about the Kalman Filter which has a closed form deterministic solution, then by inferring that the distribution of any state at the vicinity of the current state can be approximated as a Gaussian, it becomes quite clear why this algorithm could work.

The challenge is now outlined, the displayed mathematical components must be obtained in order for the algorithm to be realized for the specific use case. The derivation is conducted in the following steps:

1. Make assumptions prior to the derivation of the dynamic equations.

2. Decide control variables, state variables and measurement variables. $\rightarrow u$, μ, z.

3. Derive the dynamic model for the quadcopter, obtaining the system of NLODEs $\rightarrow g_t(u_t, \mu_{t-1})$, G_t, R_t.

4. Decide sensor configuration followed by the measurement model. $\rightarrow h_t(u_t)$, H_t, Q_t.

The derivation of the dynamic model among with most of the steps are included in Appendix A. The final result however in terms of the dynamic model is restated here. Please see table 1 to get acquainted with the variables.

$$
\begin{cases}
\dot{\Phi} &= p + t_\Theta S_\Phi q - t_\Theta C_\Phi r \\
\dot{\Theta} &= C_\Phi q + S_\Phi r \\
\dot{\Psi} &= -\frac{S_\Phi}{C_\Theta} q + \frac{C_\Phi}{C_\Theta} r \\
\dot{p} &= \frac{\tau_x}{I_{xx}} - qr\frac{(I_{zz}+I_{yy})}{I_{xx}} \\
\dot{q} &= \frac{\tau_y}{I_{yy}} - pr\frac{(I_{xx}-I_{yy})}{I_{yy}} \\
\dot{r} &= \frac{\tau_z}{I_{zz}} - pq\frac{(I_{yy}-I_{xx})}{I_{zz}} \\
\dot{u} &= rv - qw - S_\Theta g \\
\dot{v} &= pw - ru + S_\Phi C_\Theta g \\
\dot{w} &= qu - pv + C_\Phi C_\Theta g - \frac{f_t}{m} \\
\dot{x} &= C_\Theta C_\Psi u + (S_\Phi S_\Theta C_\Psi - C_\Phi S_\Psi)v + (C_\Phi S_\Theta C_\Psi + S_\Phi S_\Psi)w \\
\dot{y} &= C_\Theta S_\Psi u + (S_\Phi S_\Theta S_\Psi + C_\Phi C_\Psi)v + (C_\Phi S_\Theta S_\Psi - S_\Phi C_\Psi)w \\
\dot{z} &= -S_\Theta u + S_\Phi C_\Theta v + C_\Phi C_\Theta w
\end{cases}
\tag{1}
$$

The quadcopter configuration can be better understood from figure 2.

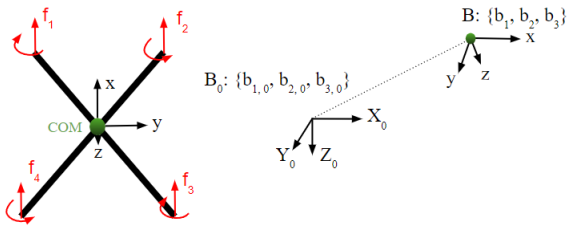


Figure 2: Illustration of the quadcopter configuration, its body fixed frame as related to the inertial frame from which Euler angles are defined.

Most symbols used in the mathematical expressions presented in this section and Appendix A are collected in table 1.

B_0 :	$\{b_{1,0}, b_{2,0}, b_{3,0}\}$	Initial body frame	p	Angular velocity about x	[rad/s]
B :	$\{b_1, b_2, b_3\}$	Body fixed frame	q	Angular velocity about y	[rad/s]
S :	$\{e_1, e_2, e_3\}$	Earth LTP	r	Angular velocity about z	[rad/s]
Ω_1	Angular velocity rotor 1	[rad/s] CW	u	Velocity in x direction	[m/s]
Ω_2	Angular velocity rotor 2	[rad/s] CW	v	Velocity in y direction	[m/s]
Ω_3	Angular velocity rotor 3	[rad/s] CCW	w	Velocity in z direction	[m/s]
Ω_4	Angular velocity rotor 4	[rad/s] CCW	x	Position in X	[m]
Φ	Roll angle, angle about X	[rad/s]	y	Position in Y	[m]
Θ	Pitch angle, angle about Y	[rad/s]	z	Position in Z	[m]
Ψ	Yaw angle, angle about Z	[rad/s]	f_t	Total thrust in z direction	[N]
R_x	Rotation matrix about X	[-]	τ_x	Roll torque about x	[Nm]
R_y	Rotation matrix about Y	[-]	τ_y	Pitch torque about y	[Nm]
R_z	Rotation matrix about Z	[-]	τ_z	Yaw torque about z	[Nm]
$R = R_{ZYX}$	Rotation matrix (3-2-1)	[-]	P	Ambient pressure	[hPa]
m	Mass of drone	[kg]	T_b	Reference temperature	[K]
l	Dist. between BLDC and *COM*	[m]	P_b	Reference pressure	[hPa]
I_{xx}	Moment inertia about x	$[kg \cdot m^2]$	R_t	Process Noise Covariance	[-]
I_{yy}	Moment inertia about y	$[kg \cdot m^2]$	Q_t	Measurement Noise Covariance	[-]
I_{zz}	Moment inertia about z	$[kg \cdot m^2]$			

Table 1: The table contains most of the definitions which are used and referred to during the mathematical derivations to come.

Control variables stored in u
The control input is realized by controlling the produced torques and forces on the quadcopter, as (1) shows, this implies that the control input vector is defined to be defined as (2).

$$\mathbf{u}^T = \begin{bmatrix} f_t & \tau_x & \tau_y & \tau_z \end{bmatrix} \tag{2}$$

State variables stored in μ
The state vector is consisting of the twelve states, to which there is one equation each, see (1). μ_t is the state estimate at time t. Generally, independent of time it assumes the structure displayed by (3).

$$\mu^T = \begin{bmatrix} \Phi & \Theta & \Psi & p & q & r & u & v & w & X & Y & Z \end{bmatrix} \tag{3}$$

Measurement variables stored in z
The measured variables are collected in $\mathbf{z}$, defined by (4)

$$\mathbf{z} = \begin{bmatrix} \Phi & \Theta & \Psi & \text{p} & \text{q} & \text{r} & \text{X} & \text{Y} & \text{P} \end{bmatrix} \tag{4}$$

These are quantities that are directly measured in the model.

2.1 State prediction model

As the derivative of each state in μ is obtained, one can derive the non-linear state transition model $g(\mu_{t-1}, u_t)$. With the previous definitions in mind, if one simply integrates the derivative for a small time step Δt, then the state can be propagated from the previous state (Eulers Method).

$$\bar{\mu_t} = g(u_t, \mu_{t-1}) \tag{5}$$

$$g(u_t, \mu_{t-1}) = \mu_{t-1} + \Delta t \left[\begin{array}{c} p + t_\Theta S_\Phi q - t_\Theta C_\Phi r \\ C_\Phi q + S_\Phi r \\ -\frac{S_\Phi}{C_\Theta} q + \frac{C_\Phi}{C_\Theta} r \\ \frac{\tau_x}{I_{xx}} - qr\frac{(I_{zz}+I_{yy})}{I_{xx}} \\ \frac{\tau_y}{I_{yy}} - pr\frac{(I_{xx}-I_{yy})}{I_{yy}} \\ \frac{\tau_z}{I_{zz}} - pq\frac{(I_{yy}-I_{xx})}{I_{zz}} \\ rv - qw - S_\Theta g \\ pw - ru + S_\Phi C_\Theta g \\ qu - pv + C_\Phi C_\Theta g - \frac{f_t}{m} \\ r_{11}u + r_{12}v + r_{13}w \\ r_{21}u + r_{22}v + r_{23}w \\ r_{31}u + r_{32}v + r_{33}w \end{array} \right] \Bigg|_{\mu_{t-1}, u_t} \tag{6}$$

$$r_{ij} = {}^{B_0}_{B}R_i, \tag{7}$$

This predicts the state and captures the non-linear dynamics, to obtain the Kalman gain however it is necessary to obtain the jacobian, which allows oneself to map the basic Kalman filter to the non-linear case, in the vicinity of the current state estimate. By differentiating each row by each respective state variable in μ, hence not with respect to a control input variable - one then in general terms mean to obtain.

$$G_t = \left[\frac{\partial g_i}{\partial \mu_j}\right]_{i,j=1:12} = \left[\begin{array}{cccc} \frac{\partial g_1}{\partial \mu_1} & \frac{\partial g_1}{\partial \mu_2} & \cdots & \frac{\partial g_1}{\partial \mu_{12}} \\ \frac{\partial g_2}{\partial \mu_1} & \frac{\partial g_2}{\partial \mu_2} & \cdots & \frac{\partial g_2}{\partial \mu_{12}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_{12}}{\partial \mu_1} & \frac{\partial g_{12}}{\partial \mu_2} & \cdots & \frac{\partial g_{12}}{\partial \mu_{12}} \end{array} \right] \Bigg|_{\mu_{t-1}, u_t} \tag{8}$$

Computing all the partial derivatives inside (8) can be a tedious process and an appropriate strategy is to use a software with a symbolic calculator, in this case it was done in MatLab. The non-trivial elements of G_t are tabulated in Appendix B.

The source for error or randomness in the state propagation originates from the process noise which is modeled by R_t. The random errors in the states are assumed to be decoupled, therefore in this model R_t is generally of the form.

$$R_t = \left[\begin{array}{cccc} \sigma_{\mu_1} & 0 & \cdots & 0 \\ 0 & \sigma_{\mu_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{\mu_{12}} \end{array} \right] \tag{9}$$

In this model, the unpredictable process noise arises from the air resistance which is modeled to apply a force anti-parallel to the velocity, scaled by its squared speed and a constant, with superimposed Gaussian noise. The model incorporates shape complexity-induced flow which acts as superimposed Gaussian noise. To enhance realism, noise is low-pass filtered, focusing on sub-50 Hz frequencies. While not a precise model, it does carry unforeseen effects which is an important characteristic about reality. Due to highly nonlinear plant dynamics, noise distribution is confidently non.-Gaussian, also realize that this noise is biased during movement. This can be easily understood from (10) which in its entirety is a disturbance, with a non-zero mean.

$$\mathbf{f_d} = -C(\|\mathbf{v_B}\|)^2 \frac{\mathbf{v_B}}{\|\mathbf{v_B}\|} + \delta_{\mathbf{noise}} \tag{10}$$

For clarity, C is the air resistance coefficient, $\delta_{\mathbf{noise}}$ represents the Gaussian superimposed noise in the form of a 3x1 vector.

Emphasizing practicality over precise quantification, the initial process noise covariance matrix is pragmatically set to what (9) describes where each diagonal element $\sigma_{\mu 1:12} = 10e-6$, thus probably undermining the process noise.

The process noise is produced in Simulink as figure 4 describes. The produced disturbance is a force, which is then added to the resultant in the body fixed frame.

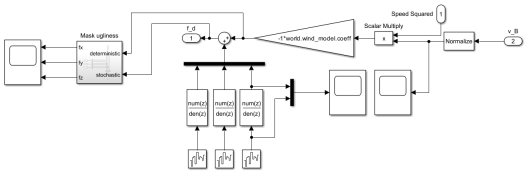


Figure 3: A screenshot of the block scheme for generating process noise.

2.2 Measurement prediction model

As previously stated, the measured units are { Φ, Θ, Ψ, p, q, r, X, Y, P }. Measuring X, Y is done by assuming that they can be obtained directly, but in reality that would be done through a conversion from longitude- and latitude measurements. Z is measured with the Barometric formula [7] by solving for Z and assuming an ambient state using the appropiate associated constants, an assumed reference temperature T_b and reference pressure P_b. These can be estimated during a calibration phase using recursive least squares at the site. The strategy for predicting the measurement is quite trivial but presented regardless as an algorithm.

The measurement model predicts the obtained measurement with the prior estimate of the required state variables, $\bar{Z}$ is used to calculate the prediction in terms of pressure by using the Barometric formula [7].

$$\bar{\mathbf{z}}_\mathbf{t} = h(u_t, \bar{\mu}_t) = \begin{bmatrix} \bar{\Phi}_t \\ \bar{\Theta}_t \\ \bar{\Psi}_t \\ \bar{p}_t \\ \bar{q}_t \\ \bar{r}_t \\ \bar{X}_t \\ \bar{Y}_t \\ P_b \left(\frac{T_b - (-\bar{Z}_t + h_b) \cdot L_b}{T_b} \right)^{\frac{g \cdot M}{R \cdot L_b}} \end{bmatrix} \tag{11}$$

The jacobian $\bar{H}_t$ is acquired by differentiation of (11) with respect to the states.

$$H_t = \left[\frac{\partial h_i}{\partial \bar{\mu}_j} \right]_{i=1:9, j=1:12} = \begin{bmatrix} \frac{\partial h_1}{\partial \bar{\mu}_1} & \frac{\partial h_1}{\partial \bar{\mu}_2} & \cdots & \frac{\partial h_1}{\partial \bar{\mu}_{12}} \\ \frac{\partial h_2}{\partial \bar{\mu}_1} & \frac{\partial h_2}{\partial \bar{\mu}_2} & \cdots & \frac{\partial h_2}{\partial \bar{\mu}_{12}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_9}{\partial \bar{\mu}_1} & \frac{\partial h_9}{\partial \bar{\mu}_2} & \cdots & \frac{\partial h_9}{\partial \bar{\mu}_{12}} \end{bmatrix} \Bigg|_{\bar{\mu}_{\mathbf{t-1}}} \tag{12}$$

Compared to the state propagation model this jacobian produces a much more condense analytical result.

$$H = \begin{bmatrix} I_{8\times 8} & 0_{8\times 4} \\ 0_{1\times 1} & J_{1\times 8} \end{bmatrix}$$

where $I_{8\times 8}$ is the 8x8 identity matrix and $J_{1\times 8}$ is a 1x8 matrix with the last element defined as:

$$J_{1\times 8}(1,8) = \frac{M \cdot Pb \cdot g \cdot ((Tb - Lb \cdot (hb - \bar{Z}_t)/Tb)^{\frac{M \cdot g}{Lb \cdot R} - 1}}{R \cdot Tb}$$

and the rest of them being equal to 0.

The source for error or randomness in the measurement model comes from noise which has its variances collected in Q_t. The random errors in the measurements are assumed to be decoupled, therefore assuming the following structure (13).

$$Q_t = \begin{bmatrix} \sigma_{z_1} & 0 & \cdots & 0 \\ 0 & \sigma_{z_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{z_{12}} \end{bmatrix} \tag{13}$$

An initial standpoint in quantifying the variances can be made from inspecting the accuracies of the different sensors, which is provided in their respective datasheets [4], [2], [6]. Relevant parameters are tabulated in table 2

Table 2: The table summarizes the sensors and their respective defining parameters

Name	Output	Freq.	Accuracy
BNO055	{ Φ, Θ, Ψ, p, q, r }	100 Hz	$\mp$ 1 deg, $\mp$ 1 deg/s [-]
DPS310	{ P }	10 Hz	$\mp$ 0.06 hPa, $\mp$ 0.5 [K]
MAX-M10S	{ X, Y }	10 Hz	$\mp$ 2 [m]

The measurement noise model which is represented by Q_t stores the variances of the measurements in its diagonal. The accuracies in table 2 are interpreted as standard deviations from the assumed unbiased measurement, hence (14).

$$Q_t = \text{diag}(\left[17.453e-2, 17.453e-2, 17.453e-2, 17.453e-2, 17.453e-2, 17.453e-2, 4, 4, 0.036\right]) \tag{14}$$

The block scheme for synthesizing the GPS/GNSS and altimeter measurements is shown in figure 4. The zero-order-hold block limit the update frequency to 10 Hz. The "Barometer" block masks its associated zero-order-hold block, it is also set to 10 Hz. This is in accordance with the sensors specifications that were previously tabulated.

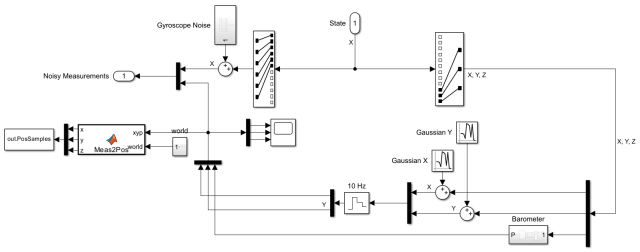


Figure 4: A screenshot of the "generate_measurement" sub-system.

3 MATLAB Implementation

Below is a high-level figure depicting the outermost layer of the system to convey the overall implementation of the system. Most subsystems of lower level have their functionalities masked but overall structure is revealed which should allow the reader to identify a feedback loop with some applied references, a controller, a plant and an observer.

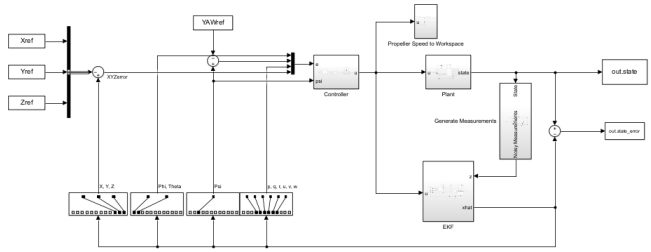


Figure 5: A high-level depiction of the Simulink model. Zooming in can reveal informative labels of different subsystems which are soon explained.

The core of this project is the "EKF" block, it deserves an explanation and some important remarks. In preparation for those remarks however, a lower level insight for the controller is given. As depicted, there is a zero-order hold right after the input. It was discussed among the group members, how is the discrete behavior of the quadcopter separated from the continuous behaviour of the plant? It was agreed upon that there should be a zero-order hold right before the input enters the plant, this makes sense because the control input remains until it is given a new command which happens every 10 milliseconds, see figure 6.

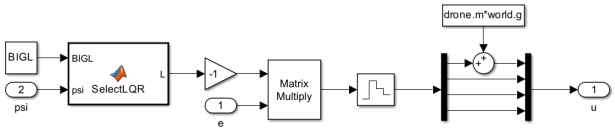


Figure 6: A screenshot of the "controller" sub-system, notice the added force to input f_t, a necessity as the LQR controller assumes a stationary point of equilibrium meaning input is to be centered about gravity. BIGL, a 3-dimensional matrix, stores several LQR solutions, one for each yaw angle in the resolution of 1 degree.

The "EKF" subsystem uses several masked functions and therefore has an additional level, figure 7 shows the expanded subsystem.

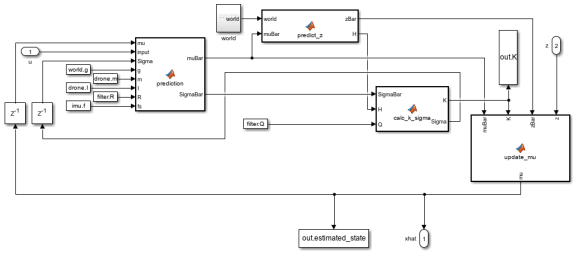


Figure 7: A screenshot of the "EKF" sub-system, notice the recursive dependency between the blocks. Another important thing to highlight are the unit delays which delay the assigned signal by an inherited time-step. The "world" class stores attributes about the modeled physical world.

The block "prediction" predicts the next state vector, hence it propagates the state based on the previous updated state and its updated covariance estimate. Thus, this block implements (Alg1: step 1:2), conducted operations are utilizing (6) and (8). The output quantities are $\bar{\mu}_t$ and $\bar{\Sigma}_t$ allowing computation of the Kalman gain K_t (Alg 1: step 3) and the updated state covariance Σ_t (Alg 1: step 5). A prediction of the measurement $h(\bar{\mu}_t)$ is acquired in the same tick using (11), which is then compared to $\bar{z}_t$ as the measurement has been sampled. The innovation is then passed to "update_mu" which outputs the updated state estimate μ_t (Alg 1: step 4). The output of Algorithm 1 is then sent to be compared to the assigned reference signal.

4 Tuning the EKF

Tuning the EKF is not a straight forward process, it requires repetitive but also diverse simulations. Different trajectories/tests highlight different attributes. It is also important to understand the intimate relationship between the controller and the filter, a controller with a reference in terms of position puts even tougher requirements on the filter to avoid destabilizing effects. The mental model of the tuning process is captured by the following algorithm. Stronger emphasis is put on the position X, Y, Z.

Algorithm 3: Tuning EKF Mental Model

$Q_t = Q_t$ proposed from (14);

for *i=1:5* **do**
  TEST "Hover" $\rightarrow$ RESULT(i);
  CHANGE Q_t(7:9,7:9)

COMPARE RESULTS AND SAVE;

for *i=1:5* **do**
  TEST "Box" $\rightarrow$ RESULT(i);
  CHANGE Q_t(7:9,7:9)

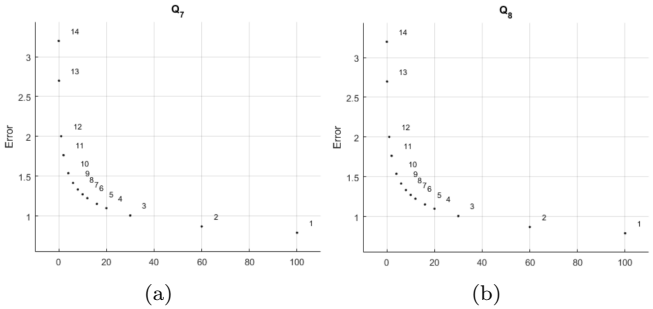
COMPARE RESULTS AND SAVE;
COMPARE SETS AND DECIDE ON Q_t;

return Q_t;

The first five iterations will put extra emphasis on estimation of X, Y and Z in a stationary state when it hovers. The ability for the quadcopter to remain in a small region is highly desired both in manual- and automatic control. It is also an easy process to analyze qualitatively. However, this case is believed to almost exclusively favour a weaker correction since the state is very static and does not unveil the performance of the state propagation model, hence the second set of tests. The second set of tests will test that the propagated state estimate stays close to the ground truth, neither slacking behind or foregoing it, the latter is however thought of as unavoidable considering the non-Gaussian disturbance which is unaccounted for in the state propagation model.

5 Simulation & Results

During the process of tuning the filter (occasionally the controller), a set of results were obtained that are hereby presented. Figure 8 shows the mean square error (MSE) in terms of X and Y against the corresponding noise covariance in Q_t. The 'ID':s, when emphasized can be cross referenced to different settings in table 3



(a) (b)

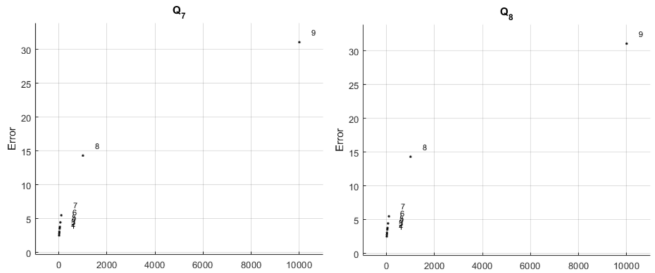
Figure 8: (Hover case) (a) MSE X (b) MSE Y

Table 3: Table collecting the simulations and the corresponding label and setting for the "Hover" case.

Nr	Sim. ID	Q_t(7:8,7:8)	MSE
13	1	100, 100	0.78895
12	2	60, 60	0.86794
11	3	30, 30	1.006
10	4	20, 20	1.0977
9	5	16, 16	1.1513
8	6	12, 12	1.2239
7	7	10, 10	1.2718
6	8	8, 8	1.3326
5	9	6, 6	1.4149
0	10	4, 4	1.5374
1	11	2, 2	1.7639
2	12	1, 1	2.0018
3	13	0.1, 0.1	2.699
4	14	1e-4, 1e-4	3.1994

As figure 8 and table 3 suggests, the error is reduced in what mimics an exponential decay as $Q_t(7,7)$ and $Q_t(8,8)$ are increased. Thus suggesting that the accuracy improves as the mentioned covariances are increased. The error reduction is initially significant but then followed by an increasingly weak continuation.

The second test forces the propagation model to be used extensively, allowing for better judgement of its performance while also diversifying the tests thus allowing for more robust tuning. Figure 9 shows the mean square error (MSE) in terms of X and Y against the corresponding noise covariance in Q_t. The 'ID':s, when emphasized can be cross referenced to different settings in table 4 which tabulates the MSE in position.



(a) (b)

Figure 9: (a) MSE X (b) MSE Y

Table 4: Table collecting the simulations and the corresponding label and setting for the "Box" case.

Nr	Sim. ID	Q_t(7:8,7:8)	MSE
23	1	10, 10	2.4982
22	2	15, 15	2.7873
21	3	20, 20	3.058
20	4	30, 30	3.5121
19	5	30, 30	3.7672
18	6	40, 40	4.431
17	7	100, 100	5.4826
16	8	1000, 1000	14.306
15	9	10000, 10000	31.079

As figure 9 and table 4 shows there is a varying but consistent positive correlation between the MSE and the covariances $Q_t(7,7)$ and $Q_t(8,8)$. Given the scale of the axes, the reader is advised to look at the table for a more precise understanding. A bit of tweaking was necessary during these tests, the settings in terms of the LQR are however collected in the 'Results' folder available at GitHub[3], the folder nr containing the result can be cross referenced with the value in the 'Nr' column.

6 Discussion

Tuning the filter was not a straight forward experience, there are several negative behaviours encircling the region for the domain of acceptable parameters. To choose a set of parameters for Q_t it helps to overlap or at least mentally portray the two results in one plot, in figure 10 is a stylized graph depicting the mental model in defence for the choice of parameters.

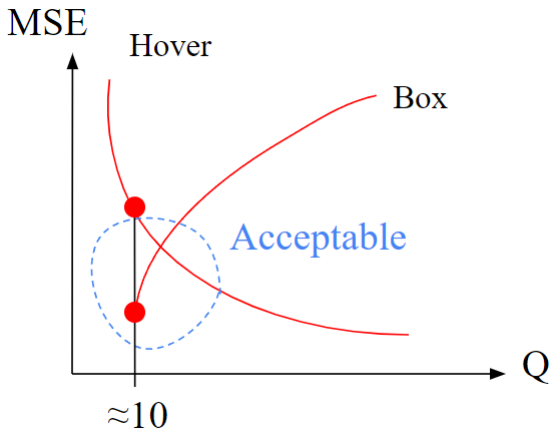


Figure 10: Mental model for choosing Q_t(7:8,7:8)

Therefore, the extreme was chosen from the set of results associated with the Box case, see table 4. It is evident from table 3 that it is not the 'best' alternative in case of hovering, however in the case of this vehicle if you will, there is of course a strong bias towards performance in dynamic situations.

To further expand on what was previously stated, that the acceptable region is surrounded by several unwanted behaviours, the reader is asked to inspect the illustration below in figure 11.

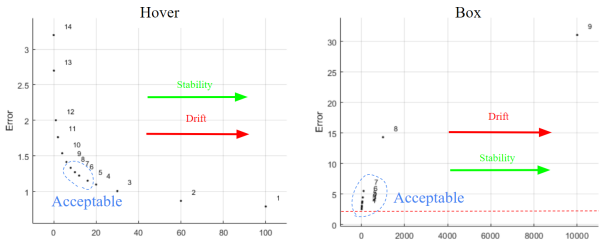


Figure 11: Illustration for qualitative analysis of filter performance

When tuning the filter it was primarily a trade between improving the stability and increasing the drift and vice versa. Reducing the drift could be attained by either overstating or rightfully stating the accuracy of the GPS/GNSS to amplify its corrective effect. This however would make the quadcopter more prone to destabilize itself, this phenomena would occur every time the quadcopter was to accelerate upward in the Box trajec-

tory, meaning that the actuators increased their RPM in order to satisfy the applied reference in Z. The finite resource in terms of input together with the faster dynamics and stronger corrections in X and Y would then collaboratively trigger and in some occasions amplify an oscillatory behaviour that in the case of being amplified would lead to the system diverging. This implied a lower border for the acceptable in terms of $Q_t(7{:}8,7{:}8)$.

The upper border is implied by the drift phenomena, in the Box case when the propagation model is used extensively, drift becomes an important factor. Result Nr 15:18 in table 4 lets the drift problem clearly manifest itself. Of course, given enough time, any position estimate can eventually converge to the acceptable vicinity, for a very small choice of $Q_t(7{:}8,7{:}8)$ - but time is off the essence and in what proportion is largely decided by how fast the dynamics are.

Figure 12 is shown below to convey the normal and expected behaviour of the EKF.

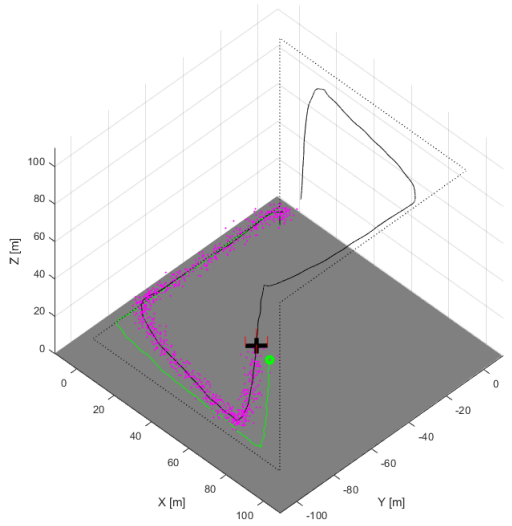


Figure 12: Screenshot of simulation for Q(7:8,7:8)=1000, demonstrating the drift phenomena.

The reader might at this point agree that the filter behaves as a typical EKF and that the typical phenomena arise and scale with parameters in a typical way. What is less typical perhaps is the lower constraint of the $Q_t(7:8,7:8)$ values, it is believed to rather be a consequence of the controller. The controller is not very adaptive and can't be fine-tuned for every situation due to its inherent simplicity. Indeed, at this point it is believed that an improved controlled would be the single most significant improvement for the overall capability of the quadcopter model. For now, the controller uses a model linearized about a stationary point - the set of stationary points are currently $\{0, 0, \Psi, 0, 0, 0, 0, 0, 0, X, Y, Z\}$ meaning that any deviation from a totally horizontal state with an arbitrary yaw angle at an arbitrary position is counteracted. For the quadcopter to travel between to points it must induce a non-zero velocity which implies the necessity for a tilted quadcopter. The result upon forced translation in the horizontal plane is a 'jig-saw' behaviour in the uncontrollable states. By introducing the air resistance into the model in an appropriate way the region of stationary points can expand vastly allowing for controllability of velocity as many tilted states are contained in the subspace of stationary points.

Another perhaps less typical phenomena which was observed during the simulations was the delay of the ground truth (the black cross), hence the estimated position (the green dot) was often in front of the quadcopter. Also, the effect seemed to scale with the speed of the quadcopter. This is believed to be due to the air resistance not being modeled in the state propagation model, the effect is profound since the disturbance has a non-zero mean that scales quadratically with the speed. To reduce this effect one simply has to implement this term into Newton-Euler's equation of translational motion, however choosing the constant to scale this effect is not thought to be straight forward. Below in figure 13, the depiction demonstrates the slacking of the estimated position in relation to ground truth.

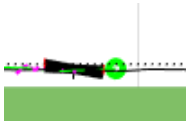


Figure 13: Screenshot of simulation for Q(7:8,7:8)=1000, demonstrating the drift phenomena.

7 Conclusion

The project is deemed as successfull since there is a clear difference in terms of performance in comparison to what was achieved in the previous project [1]. The simulation in the previous project did not induce noise into the model but while working with the hardware it was evident that some measurements had a lot of noise and that it in fact did lead to drift of position while velocity could not be estimated in states of constant velocity. This implies that the quadcopter now is possible to control, this is demonstrated in [3] where one can test the model by controlling the quadcopter in a 3D rendered environment using a joystick, in case the reader does not own a joystick there is also a video provided with some commentary.

The result in terms of accuracy are stated through the tables cross referenced with the choise of $Q_t(7{:}8,7{:}8)$. By evidence the performance depends quite a lot on the situation which infers different dynamics, some are handled better than others but it generally tends to be harder the faster the dynamics are. It is however explored that one can control the quadcopter manually rather successfully.

References

[1] Ludwig Horvath and Martin Petré. "Quadcopter Control System Design and Implementation with LQR assisted by Kalman Filter: An Experimental Study". Independent thesis. KTH, School of Industrial Engineering and Management (ITM), 2023. URL: https://kth.diva-portal.org/smash/record.jsf?aq2=%5B%5B%5D%5D&c=16&af=%5B%5D&searchType=LIST_LATEST&sortOrder2=title_sort_asc&query=&language=en&pid=diva2%3A1775501&aq=%5B%5B%5D%5D&sf=all&aqe=%5B%5D&sortOrder=author_sort_asc&onlyFullText=false&noOfRows=50&dswid=386.

[2] Infineon Technologies AG. *Infineon DPS310 Data Sheet*. 2019. URL: https://www.infineon.com/dgdl/Infineon-DPS310-DataSheet-v01_02-EN.pdf?fileId=5546d462576f34750157750826c42242.

[3] Martin Petré Ludwig Horvath. *Quadcopter-Simulation*. GitHub repository. 1/20/2024. URL: https://github.com/JELH8980/Quadcopter-Simulation.git.

[4] Bosch Sensortec. *BNO055 Intelligent 9-axis absolute orientation sensor datasheet*. 2014. URL: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bno055-ds000.pdf.

[5] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005. ISBN: 978-0262201629.

[6] u-blox AG. *MAX-M10S Data Sheet*. 2018. URL: https://cdn.sparkfun.com/assets/7/5/9/a/a/MAX-M10S_DataSheet_UBX-20035208.pdf.

[7] Wikipedia contributors. *Barometric formula*. Accessed: January 7, 2024. 2022. URL: https://en.wikipedia.org/wiki/Barometric_formula.

A Derivation of Dynamic Equations

In accordance with the steps written in the mathematical formulation section, the first stage is to confine the complexity of the problem by imposing some assumptions.

A.1 Assumptions

1. The quadcopter is not to pitch or roll beyond the $\mp$ 90 $^\circ$ mark, this assumption validates the use of the Euler rotational sequence: 3-2-1 since singularities are avoided during assumed operation.
2. The frame associated with the quadcopters initial position is assumed to be an inertial frame.
3. The COM is assumed to intersect with the volume center, or the center of the cross as displayed in 14, making gravity not exert any torque in the body fixed frame.
4. The body fixed frame is centered in the COM which is the exact location of the IMU.

A.2 Quadcopter configuration

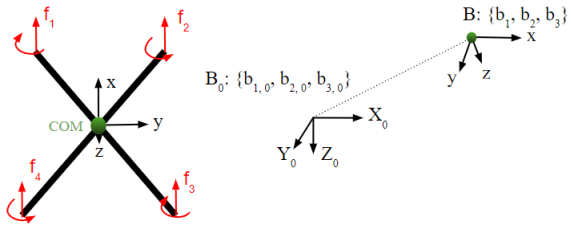


Figure 14: Illustration of the quadcopter configuration, its body fixed frame as related to the inertial frame to which Euler angles are measured.

As figure 14 suggests, the quadcopter assumes a symmetrical X configuration, the four actuators produce one force each. Adjacent actuators spin about the opposing directions while the opposing actuator spins about the same direction, this expands the region of controllability allowing the quadcopter to yaw during stable operation. The pose of the body fixed frame B is totally determined with respect to the inertial frame B_0 by its translation and orientation with respect to B_0:s frame, the orientation is defined in terms of Euler angles Φ, Θ, Ψ. For the purpose of brevity and formality, the reader is reminded to glance at the definitions summarized in table 1. The configuration in regards to the actuators is clarified by the tabulated system parameters, see table 5

Actuator number	1	2	3	4
Direction of rotation	CW	CCW	CW	CCW
Lift coefficient	b	b	b	b
Drag coefficient	d	d	d	d

Table 5: The table contains system parameters which together state the configuration of the quadcopter in terms of its actuators.

Starting with the actuators we have a set of equations relating the produced torques and forces to the angular rate of the propellers.

$$\begin{cases} \tau_x = \frac{\ell}{\sqrt{2}}({f_1}^2 - {f_2}^2 - {f_3}^2 + {f_1}^2) \\ \tau_y = \frac{\ell}{\sqrt{2}}({f_1}^2 + {f_2}^2 - {f_3}^2 - {f_1}^2) \\ \tau_z = -d{\Omega_1}^2 + d{\Omega_2}^2 - d{\Omega_3}^2 + d{\Omega_4}^2 \end{cases} \tag{15}$$

Where $f_i \ \forall \ i = \{1,2,3,4\}$ is given by:

$$f_i = b{\Omega_i}^2 \tag{16}$$

Note that b is of course a constant that must be obtained empirically, one can make LSE from the parameters and result from recorded experiments which are provided online or copy the exact configuration of hardware and expect the same result, for a wide region of operation the slope relating the force and squared angular velocity is close to constant.

A.3 Kinetics and Kinematics

Using the Newtons-Euler equations of motion, one can obtain differential equations relating the translational- and rotational motion to the forces and moments exerted on the UAV. Newtons equation for translational motion (17) is stated below , together with the kinematic relationship for the acceleration of the center of mass.

$$\sum_{\forall i} \mathbf{F_i} = m\mathbf{a_G} \quad = m\left({}^{B}\frac{\partial}{\partial t}\,{}^{R}\mathbf{v_G} + {}^{R}\omega_B \times {}^{R}\mathbf{v_G} \right) \tag{17}$$

Where $\mathbf{v_G}$, $\omega_\mathbf{B}$ is the velocity of COM and angular velocity of the body fixed frame B_0 respectively, defined in the body fixed reference frame.

$$\begin{aligned} {}^{R}\mathbf{v_G} &= u\mathbf{b_1} + v\mathbf{b_2} + w\mathbf{b_3} \\ {}^{R}\omega_B &= p\mathbf{b_1} + q\mathbf{b_2} + r\mathbf{b_3} \end{aligned} \tag{18}$$

Newton-Eulers equation for the rotational motion of the body is stated in the following way.

$$\sum_{\forall i} \mathbf{M_i} = I_G\,{}^{R}\alpha_\mathbf{B} + {}^{R}\omega_B \times \mathbf{H_G} \tag{19}$$

Where $\mathbf{H_G}$ represents the angular momentum and it is calculated as such.

$$\mathbf{H_G} = I_G\,{}^{R}\omega_B \tag{20}$$

Substitution of (22) into (17) gives.

$$\sum_{\forall i} \mathbf{F_i} = m\left(\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{2} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \right) \tag{21}$$

Substitution of (22) into (19), provided that

$${}^{R}\alpha_\mathbf{B} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{22}$$

produces the following result, completing the RHS of the kinetic equations.

$$\sum_{\forall i} \mathbf{M_i} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \times \left(\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \right) \tag{23}$$

What remains is to complete the LHS of (21) and (24). The LHS term of each equation is written in terms of the body fixed unit vectors $\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$.

Thus, the final representation is obtained for the Newton-Euler equation for rotational motion.

$$\begin{bmatrix}\tau_x\\ \tau_y\\ \tau_z\end{bmatrix} = \begin{bmatrix}I_{xx} & 0 & 0\\ 0 & I_{yy} & 0\\ 0 & 0 & I_{zz}\end{bmatrix}\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} + \begin{bmatrix}p\\ q\\ r\end{bmatrix} \times \left(\begin{bmatrix}I_{xx} & 0 & 0\\ 0 & I_{yy} & 0\\ 0 & 0 & I_{zz}\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix}\right) \tag{24}$$

The sum of the thrust from all four actuators in addition with the effect of gravity constitutes the LHS of (21), see (25).

$$\sum_{\forall i} \mathbf{F_i} = -\sum_{\forall j}^{4} \mathbf{f_j} + m\, {}^{B_0}_{B}R^T \begin{bmatrix}0\\ 0\\ g\end{bmatrix} \tag{25}$$

The final representation for the Newtons equation for translational motion is thus.

$$-\sum_{\forall j}^{4} \mathbf{f_j} + m\, {}^{B_0}_{B}R^T \begin{bmatrix}0\\ 0\\ g\end{bmatrix} = m\left(\begin{bmatrix}\dot{u}\\ \dot{v}\\ \dot{2}\end{bmatrix} + \begin{bmatrix}p\\ q\\ r\end{bmatrix} \times \begin{bmatrix}u\\ v\\ w\end{bmatrix}\right) \tag{26}$$

The rotation matrix ${}^{B_0}_{B}R$ is defined as

$${}^{B_0}_{B}R = \begin{bmatrix} C_\Theta C_\Psi & C_\Theta S_\Psi & -S_\Theta \\ S_\Phi S_\Theta C_\Psi - C_\Phi S_\Psi & S_\Phi S_\Theta S_\Psi + C_\Phi C_\Psi & S_\Phi C_\Theta \\ C_\Phi S_\Theta C_\Psi + S_\Phi S_\Psi & C_\Phi S_\Theta S_\Psi - S_\Phi C_\Psi & C_\Phi C_\Theta \end{bmatrix}^T \tag{27}$$

The matrix transforms coordinates in the initial body frame B_0 to corresponding coordinates in the body frame B, provided that the body fixed frame has Euler angles Ψ, Φ and Θ. Finally, (21) is rewritten in the following form.

$$\begin{cases} \dot{u} = rv - qw - S_\Theta g \\ \dot{v} = pw - ru + S_\Phi C_\Theta g \\ \dot{w} = qu - pv + C_\Phi C_\Theta g - \frac{f_t}{m} \end{cases} \tag{28}$$

On to relating the angular velocities in the body frame to the inertial frame. From three consecutive rotations in a (3-2-1) rotation sequence, one can state that the resulting angular velocity is given by (29).

$${}^{R}\omega_B = \dot{\Psi}\mathbf{b_{3,0}} + \dot{\Theta}\mathbf{N_2}'' + \dot{\Phi}\mathbf{b_1} \tag{29}$$

To complete the kinematic relationship the ambition is to write (29) in terms of body-fixed unit vectors exclusively $(\mathbf{b_1}, \mathbf{b_2}, \mathbf{b_3})$. Before presenting this derivation, realize that the following is true.

$$\begin{Bmatrix}\mathbf{N_1}'\\ \mathbf{N_2}'\\ \mathbf{N_3}'\end{Bmatrix} = R_3^T \begin{Bmatrix}\mathbf{b_{1,0}}\\ \mathbf{b_{2,0}}\\ \mathbf{b_{3,0}}\end{Bmatrix} \quad \begin{Bmatrix}\mathbf{N_1}''\\ \mathbf{N_2}''\\ \mathbf{N_3}''\end{Bmatrix} = R_2^T \begin{Bmatrix}\mathbf{N_1}'\\ \mathbf{N_2}'\\ \mathbf{N_3}'\end{Bmatrix} \quad \begin{Bmatrix}\mathbf{e_1}\\ \mathbf{e_2}\\ \mathbf{e_3}\end{Bmatrix} = R_1^T \begin{Bmatrix}\mathbf{N_1}''\\ \mathbf{N_2}''\\ \mathbf{M_3}''\end{Bmatrix} \tag{30}$$

With the relations stated in (30) one can by substitution into (29) obtain an expression in matrix form.

$$\begin{bmatrix}p\\ q\\ r\end{bmatrix} = T^{-1}\begin{bmatrix}\dot{\Phi}\\ \dot{\Theta}\\ \dot{\Psi}\end{bmatrix} \qquad , \qquad T = \begin{bmatrix}1 & t_\Theta S_\Phi & -t_\Theta C_\Phi\\ 0 & C_\Phi & S_\Phi\\ 0 & -\frac{S_\Phi}{C_\Theta} & \frac{C_\Phi}{C_\Theta}\end{bmatrix} \tag{31}$$

This expression is then inverted and written in component from, thus (32) is obtained.

$$\begin{cases} \dot{\Phi} &= p + t_\Theta S_\Phi q - t_\Theta C_\Phi r \\ \dot{\Theta} &= C_\Phi q + S_\Phi r \\ \dot{\Psi} &= -\frac{S_\Phi}{C_\Theta} q + \frac{C_\Phi}{C_\Theta} r \end{cases} \tag{32}$$

What remains is to first invert (24) and instead write $\dot{p}$, $\dot{q}$, $\dot{r}$ in terms of p, q, r thus obtaining a system of 1:st order NL-ODEs.

$$\begin{cases} \dot{p} = \frac{\tau_x}{I_{xx}} - qr\frac{(I_{zz}+I_{yy})}{I_{xx}} \\ \dot{q} = \frac{\tau_y}{I_{yy}} - pr\frac{(I_{xx}-I_{yy})}{I_{yy}} \\ \dot{r} = \frac{\tau_z}{I_{zz}} - pq\frac{(I_{yy}-I_{xx})}{I_{zz}} \end{cases} \tag{33}$$

The last necessary step in order to control X, Y and Z is a relation between their time derivatives and the velocity in the body fixed frame.

$$\begin{cases} \dot{x} = C_\Theta C_\Psi u + (S_\Phi S_\Theta C_\Psi - C_\Phi S_\Psi)v + (C_\Phi S_\Theta C_\Psi + S_\Phi S_\Psi)w \\ \dot{y} = C_\Theta S_\Psi u + (S_\Phi S_\Theta S_\Psi + C_\Phi C_\Psi)v + (C_\Phi S_\Theta S_\Psi - S_\Phi C_\Psi)w \\ \dot{z} = -S_\Theta u + S_\Phi C_\Theta v + C_\Phi C_\Theta w \end{cases} \tag{34}$$

All twelve equations are now to be collected from (32), (33) and (28). Below is the total system of dynamic equations(35)

A.4 Final result of derivation

$$\begin{cases} \dot{\Phi} = p + t_\Theta S_\Phi q - t_\Theta C_\Phi r \\ \dot{\Theta} = C_\Phi q + S_\Phi r \\ \dot{\Psi} = -\frac{S_\Phi}{C_\Theta}q + \frac{C_\Phi}{C_\Theta}r \\ \dot{p} = \frac{\tau_x}{I_{xx}} - qr\frac{(I_{zz}+I_{yy})}{I_{xx}} \\ \dot{q} = \frac{\tau_y}{I_{yy}} - pr\frac{(I_{xx}-I_{yy})}{I_{yy}} \\ \dot{r} = \frac{\tau_z}{I_{zz}} - pq\frac{(I_{yy}-I_{xx})}{I_{zz}} \\ \dot{u} = rv - qw - S_\Theta g \\ \dot{v} = pw - ru + S_\Phi C_\Theta g \\ \dot{w} = qu - pv + C_\Phi C_\Theta g - \frac{f_t}{m} \\ \dot{x} = C_\Theta C_\Psi u + (S_\Phi S_\Theta C_\Psi - C_\Phi S_\Psi)v + (C_\Phi S_\Theta C_\Psi + S_\Phi S_\Psi)w \\ \dot{y} = C_\Theta S_\Psi u + (S_\Phi S_\Theta S_\Psi + C_\Phi C_\Psi)v + (C_\Phi S_\Theta S_\Psi - S_\Phi C_\Psi)w \\ \dot{z} = -S_\Theta u + S_\Phi C_\Theta v + C_\Phi C_\Theta w \end{cases} \tag{35}$$

This constitutes the dynamic model.

B Jacobian Matrix Elements G_t

Index	Element
(1,1)	$h(\dot{q}C_\Phi t_\Theta + \dot{r}S_\Phi t_\Theta) + 1$
(1,2)	$-h(\dot{r}C_\Phi(t_\Theta^2+1) - \dot{q}S_\Phi(t_\Theta^2+1))$
(1,4)	h
(1,5)	$hS_\Phi t_\Theta$
(1,6)	$-hC_\Phi t_\Theta$
(2,1)	$h(\dot{r}C_\Phi - \dot{q}S_\Phi)$
(2,2)	1
(2,5)	hC_Φ
(2,6)	hS_Φ
(3,1)	$-h((\dot{q}C_\Phi)/C_\Theta + (\dot{r}S_\Phi)/C_\Theta)$
(3,2)	$h((\dot{r}C_\Phi S_\Theta)/C_\Theta^2 - (\dot{q}S_\Phi S_\Theta)/C_\Theta^2)$
(3,3)	1
(3,5)	$-hS_\Phi/C_\Theta$
(3,6)	hC_Φ/C_Θ
(4,4)	1

Index	Element
(4,5)	$h\dot{r}(I_y - I_z)/I_x$
(4,6)	$h\dot{q}(I_y - I_z)/I_x$
(5,5)	1
(5,6)	$-h\dot{p}(I_x - I_z)/I_y$
(6,6)	1
(7,2)	$-ghC_\Theta$
(7,4)	$-hw$
(7,5)	hv
(7,7)	1
(7,8)	$h\dot{r}$
(8,1)	$ghC_\Phi C_\Theta$
(8,2)	$-ghS_\Phi S_\Theta$
(8,4)	hw
(8,5)	$-hu$
(8,6)	$-h\dot{r}$
(8,8)	1
(8,9)	$h\dot{p}$
(9,1)	$-ghC_\Theta S_\Phi$
(9,2)	$-ghC_\Phi S_\Theta$
(9,4)	$-hv$
(9,5)	hu
(9,6)	$h\dot{q}$
(10,1)	$h(v(S_\Phi S_\Psi + C_\Phi C_\Psi S_\Theta) + w(C_\Phi S_\Psi - C_\Psi S_\Phi S_\Theta))$
(10,2)	$h(vC_\Psi C_\Theta S_\Phi - uC_\Psi S_\Theta + wC_\Phi C_\Psi C_\Theta)$
(10,3)	$-h(v(S_\Phi C_\Psi + C_\Phi S_\Psi S_\Theta) - w(C_\Psi S_\Phi - C_\Phi S_\Psi S_\Theta) + uC_\Theta S_\Psi)$
(10,7)	$hC(\Psi)C_\Theta$
(10,8)	$-h(C(\Phi)S_\Psi - C(\Psi)S_\Phi S_\Theta)$
(10,9)	$-h(C(\Phi)S_\Psi - C(\Psi)S_\Phi S_\Theta)$
(10,10)	1
(11,1)	$-h(v(C_\Psi S_\Phi - C_\Phi S_\Psi S_\Theta) + w(C_\Phi C_\Psi + S_\Phi S_\Psi S_\Theta))$
(11,2)	$h(wC_\Phi C_\Theta S_\Psi - uS_\Psi S_\Theta + vC_\Theta S_\Phi S_\Psi)$
(11,3)	$h(w(S_\Phi S_\Psi + C_\Phi C_\Psi S_\Theta) - v(C_\Phi S_\Psi - C_\Psi S_\Phi S_\Theta) + uC_\Psi C_\Theta)$
(11,7)	$hC_\Theta S_\Psi$
(11,8)	$h(C_\Phi C_\Psi + S_\Phi S_\Psi S_\Theta)$
(11,9)	$-h(C_\Psi S_\Phi - C_\Phi S_\Psi S_\Theta)$
(11,11)	1
(12,1)	$h(vC_\Phi C_\Theta - wC_\Theta S_\Phi)$
(12,2)	$-h(uC_\Theta + wC_\Phi S_\Theta + vS_\Phi S_\Theta)$
(12,3)	0
(12,7)	$-hS_\Theta$
(12,8)	$hC_\Theta S_\Phi$
(12,9)	$hC_\Phi C_\Theta$
(12,12)	1

Table 6: Non-zero elements of the matrix G_t