Integrating Fault Detection and Diagnosis into Fixed-Wing UAVs

PhD Research Documentation

Georgios Zogopoulos - Papaliakos

National Technical University of Athens

School of Mechanical Engineering
Control Systems Laboratory

e-mail: gzogop@mail.ntua.gr

Athens , March 26, 2015

Contents

1: Why Do UAVs Need FDD Systems

Aerial robotic systems, also known as Unmanned Aerial Vehicles (UAVs) are becoming increasingly popular during the past decade. After seminal research on military applications ran its lifespan, smaller and less expensive systems found their way to research institutes globally. After the imperative and necessary control problems where solved and structural designs where streamlined, UAVs finally found their way to the consumer at the end of the 2000's, as a natural extension of the radio-controlled aircraft market.

The public quickly embraced this new technology and as of 2014, consumer Unmanned Aerial Systems (UASs) are widely used for aerophotography, ground surveys and recreation. However, the fact that UAVs are able to pilot themselves using well-established control algorithms for the nominal case, does not mean that they are able to remain operational in case of faults. Indeed, amateur UAV operators rarely perform Standard Operational Procedures, pre-flight checks or pre-emptive maintenance. As a result, the sight of a UAV falling out of the sky is all too common.

One would argue that operators should be the ones responsible for correct system practices, situation assessment and integrity checks. However, every time a technology gains wide acceptance to the public, eventually the task of safety assurance falls onto the manufacturer. As a comparison, consider how common and indispensable safety systems such as the ABS, ESP and vehicle diagnostics are in the automotive industry. If consumer UAVs are to become a part of our everyday lives, have a respectable degree of reliability and conquer the skies, safety standards must be built at the system level. This is the objective of the discipline of Fault Detection and Diagnosis (FDD)

Let us loosely describe Fault Detection and Diagnosis here and provide a more rigorous, mathematical definition later. Faults are events that throw a system out of its nominal mode of operation. Most of the times, faults, if allowed to go unchecked, will eventually result to system failures, which is an unwanted situation. Hence, it is of our interest to detect faults at their birth and take counter-measures against them, in order to ensure system health and if possible, maintain its operational status. Fault Detection involves setting up mathematical and physical structures that are able to detect when a fault occurs in our

system. Afterwards, Fault Isolation should be performed, in order to identify the system component which is under fault. This is an important piece of information, since it allows us to target the source of a fault and is a necessary prerequisite for reconfigurable, fault-tolerant control (FTC). Finally, Fault Detection methods allow us to assess the magnitude of the fault, be it the drift of a physical parameter value, the flow of a leak or the strength of a disturbance. Oftentimes, Isolation and Detection are mutually referred to as Diagnosis.

2: Historical FDD Overview

Fault Detection and Diagnosis is a subject of research since the end of the 70's, originally intended for application in the petrochemical industry, which was a very innovative, wealthy and large-scale sector at that time. Employing mainly state and parameter estimation techniques, which where very costly computationaly-wise at the time, FDD was barely applicable to the slow-evolving chemical processes.

During the 90's, when adaptive control was a new topic of interest, FDD was re-visited to aid in building fault-tolerant control systems. Especially in the airline industry, which at that time was both booming and employing in-flight computers with system-wide span, efforts where made to design systems which would automatically intervene to the pilot's actions, to detect some basic faults and avert potential (and potentially very expensive and life-threatening) accidents. However, the costly experimental iterations, as well as the inability to guarantee the correct operation of the fault-tolerant control system, which would in turn jeopardize human lives, prevented FDD and FTC from reaching their full potential in airborne systems.

UAVs provide the low-cost and disposable test platform which is needed to verify the efficiency of FDD methods and are getting increasing interest with every successful research milestone.

3: The Current FDD Scope w.r.t. Fixed-Wing UAVs

The most fatal failure on a fixed-wing UAV (which have the traditional "airplane" shape), apart from severe structural damage, is a malfunction of its control surfaces. Indeed, an airplane could perform an emergency landing with malfunctioning undercarriage, with a reduced sensor suite, with a broken propeller or even without an engine. But without properly functioning ailerons and elevators the aircraft is essentially crippled. This is why most research in fw-UAV FDD revolves around detection of faults of control surfaces and their corresponding actuators. Other research topics include detection of Pitot tube malfunctions and icing scenarios. The latter does not directly apply to consumer-grade UAVs, as the icing phenomenon occurs in high altitudes and low temperatures. The former has indeed applicable results to all fw-UAV airframes.

4: The Need for Holistic System Health Monitoring

As is often quoted "The whole is more than the sum of its parts". In the case of fw-UAVs, the low-level system components have attributes and effects which not only affect themselves or the components they are connected to, but also the subsystem they belong to or even the whole airframe. Electrical, mechanical and dynamic interdependencies between components form a mesh of interactions, the knowledge of which is valuable. Each component is not isolated; its faults affect other parts of the system, even beyond its subsystem, and in turn said component may depend on other components working properly for itself to operate nominally. As an example, consider the case of an electric UAV, where a part of the propeller blade gets clipped, as a result of a hardware failure. Immediately, because of the imbalanced load on the motor shaft, the propeller angular inertia is increased and the motor struggles to maintain its commanded RPM. This results in excessive current draw and also reduced thrust power. Moreover, vibrations are generated and transmitted throughout the airframe, cluttering the accelerometer sensor bandwidth and possibly affecting the autopilot performance. Even in simple, single-component failures, the implications on a fast-evolving, tightly-coupled system may be severe and in ways that may not be directly apparent or so complex that are difficult to keep track of.

To showcase the extent of the problem, an indicative list of an UAV system resources and potential failures is presented.

It is apparent that in order to investigate the dependencies between system resources and system faults, with regard to component descriptions, we need to handle the system complexity using a formal, scalable representation method for this kind of dependencies.

System Resources	**System Faults**
Hardware	Hardware
motor	propeller damage
actuators	propeller dislocation
sensors	motor loss of power
video & data tranceivers	motor total failure
antennae	power supply cut-off
fuel	power supply total failure
battery capacity	battery shortened life
processing units (on-board computers)	battery fire
aerodynamics surfaces	actuator failure
structural support	control surface dislocation
Software	control surface break-off
Computational power	sensor elevated noise floor
Computational memory	sensor failure
Other	communication partial/total loss
Airspeed	structural failure partial/total
Altitude	aerodynamic surface alteration
	center of gravity dislocation
	processor damage
	Software
	bugs
	unexpected states

5: Tackling System Complexity

5.1 Component Model

An effort to describe the system structure in a well-defined and scalable way will now be made. The core building block is the component, which will be subsequently defined. To better illustrate our approach, the example of a 1-axis digital MEMS gyroscope will be presented alongside the definition.

Initially, the notion of the service will be introduced. A service is a 6-tuple of the form:

<consumed variables, produced variables, procedure, request, activation condition, resources>

This model describes the service as a process which uses *consumed variables* upon *request* according to a *procedure* in order to return *produced variables*. In order for the service to be carried out *resources* are used up and an *activation condition* must hold true.

Thus, using set theory for every mentioned quantity, we can define a set of services provided by a component k as:

Definition: Service

$$S(k) = \{s_i(k), i \in I_s(k)\}$$
$$s_i(k) = \{cons_i(k), prod_i(k), proc_i(k), rqst_i(k), active_i(k), res_i(k)\}$$

The set $I_s(k)$ contains the service indices which are applicable to the component k. The same holds for the sets $cons(k), prod(k), proc(k), active(k), res(k)$.

In the case of the MEMS accelerometer with the component index 1, we can define:

$$
\begin{aligned}
I_s(1) &= \{1, 2\} \\
cons(1) &= \{\text{angular rate } \omega, \text{temperature } K, \text{measurement noise } w, \text{initial angle input } \theta_{0i}\} \\
prod(1) &= \{\text{measured angular rate } \omega_m, \text{measured angle } \theta_m\} \\
proc(1) &= \left\{\omega_m = \omega + c_1 K + w, \theta_m = \theta_0 + \int \omega_m, \theta_0 = \theta_{0i}\right\} \\
rqst(1) &= \{\text{I2C command}\} \\
active(1) &= \{\text{Power supply present}\} \\
resources(1) &= \{\text{current consumption rate}, \text{gyroscope unit}, \text{I2C channel}\} \\
S(1) &= \{s_1(k), s_2(k)\}
\end{aligned}
$$

Where service $s_1(1)$ is the initial sensor angle specification and $s_2(1)$ is the measurement of angular rate and angle.

Definition: Use Mode We define a set of use-modes

$$M(k) = \{m_i(k), i \in I_m(k)\}$$

where $m_i(k)$ is the set of indices of the available component k services during the use mode i and $I_m(k)$ is the set of indices of available use modes for the component k. The applicability of services during a use mode is a designer choice. In our example we have two use modes:

$$
\begin{aligned}
m_1(1) &= \{s_1(1)\} \quad \text{calibration mode} \\
m_2(1) &= \{s_2(1)\} \quad \text{operational mode}
\end{aligned}
$$

Definition: Component Model We are now finally able to define a component as the aggregation:

$$
\begin{aligned}
&<\text{component k}> \equiv <\text{state transition graph } G(M(k), \tau(k), m^0(k)) > \\
&\quad < M(k) > \\
&\quad < \tau(k) > \\
&\quad < m^0(k) > \\
&\quad < m_i(k) > \\
&\quad < s_i(k) >
\end{aligned}
$$

5.2 Structural Model

Given the component descriptions we have set up and their interconnections, we can "translate" the component behavioural model onto a qualitative model for structural analysis.

The qualitative model retains data only about the information flow and the component interconnections. It is essentially a directional graph which visualizes the evolution of the variables from the known system inputs towards the also known system outputs, via the unknown system internal variables and states.

A vital definition towards this goal is that of the *constraint*. A constraint is a relation between two system variables, which expresses a physical model, a component model or a control law. In essence, the sum of the system constraints constitutes the full set of equations, either algebraic or differential, which make up our system.

Based on the previous component definition, the system constraints are the union of the component procedures

$$\mathcal{C} = \bigcup proc(k)$$

while all the variables involved in $\mathcal{C}$ constitute the variable set $\mathcal{Z}$.

Variables whose values are known (inputs, measurements etc) belong to the known variables set $\mathcal{K}$, whereas the rest of the variables belong to the unknown variables set $\mathcal{X}$. It holds that $\mathcal{Z} = \mathcal{K} \cup \mathcal{X}$.

We define a bi-partite graph $(\mathcal{C}, \mathcal{Z}, \mathcal{A})$, with vertices $\mathcal{C}, \mathcal{Z}$ and edges $\mathcal{A}$ linking constraints with their corresponding variables. It is meaningful to define directionality in the graph, to reflect the fact that a single constraint cannot be solved for any of its variables. Indeed, from the simple differential equation $x_2 = \dot{x_1}$, we cannot extract the values of x_1, since that would require the additional initial condition, which is not given.

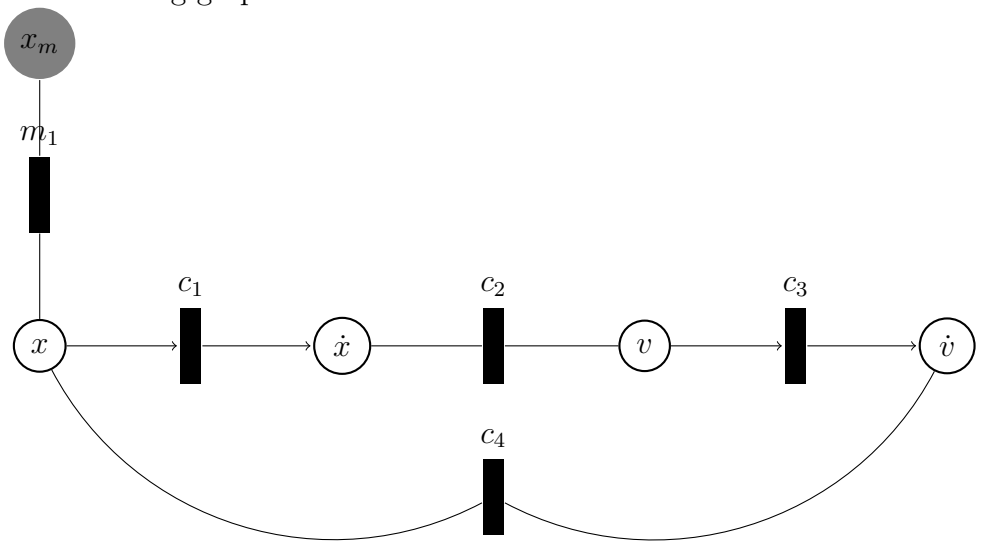
An important segmentation of $\mathcal{C}$ in regard to $\mathcal{Z}$ is splitting it to:
$\mathcal{C}_\mathcal{K}$, which contains the restrictions which refer only to known variables
$\mathcal{C}_\mathcal{X} = \mathcal{C} - \mathcal{C}_K$

To showcase this approach, let us consider another example, a spring-mass system with position measuring.

$$\begin{aligned}
c_1: &\quad \dot{x} = \frac{d}{dt}x \\
c_2: &\quad v = \dot{x} \\
c_3: &\quad \dot{v} = \frac{d}{dt}v \\
c_4: &\quad \dot{v} = \frac{K}{m}x \\
m_1: &\quad x_m = x
\end{aligned}$$

The constraints set is $\mathcal{C} = \{c_1, c_2, c_3, c_4, m_1\}$
The known variables set is $\mathcal{K} = \{x_m\}$

The unknown variables set is $\mathcal{X} = \{x, \dot{x}, v, \dot{v}\}$
The resulting graph is:



Naturally, this graph could be represented as an incidence matrix D, where

$$d_{ij} = \begin{cases} 0 & \text{if } (z_i, c_i) \in \mathcal{A} \\ 1 & \text{if } (z_i, c_i) \notin \mathcal{A} \end{cases}$$

We could extend this definition to include the graph directionality if we wanted to, by setting the vertices of $\mathcal{A}$ to not be bi-directional but rather two separate vertices. This would make the matrix D non-symmetrical.

Having set up the necessary mathematical tools (graphs and adjacency matrices) to explore the system structure, we are able to gain much more insight on our system. For example, we can use the existence of paths connecting nodes to explore whether one variable $x_1$of our system is related to another x_2, in other words, if $\frac{\partial}{\partial x_1} x_2 \neq 0$. Moreover, we can discuss about whether an internal variable is **observable**, by finding connected paths originating from it and reaching known variables. Finally, we can even find if variables of our system are **controllable**, by looking for connected paths to input variables. This will be especially useful in cases where we want to design reconfigurable controllers for our system.

6: The Matching Problem

In the previous chapter, we saw how the system model structure can be represented in a qualitative way by using bi-partite graphs. Using this information, we need to acquire a way, ie a computation sequence, which will allow us to calculate the unknown variables in $\mathcal{X}$ from the inputs and measurements in $\mathcal{K}$. It goes without saying that such a calculation sequence cannot be derived by hand in large-scale systems. Actually, the problem of calculating each element from the set $\mathcal{X}$ using an element from $\mathcal{K}_{\mathcal{X}}$ only once can be generalized to a common problem in graph theory, called *the matching problem* [16, ch 26].

Given a graph $G = (V, E)$, a **matching** is a subset $M \subseteq E$ such that for all vertices $v \in V$, at most one edge of M is incident on v. We say that v is matched by the matching M if an edge of M is incident on v. A **maximum matching** is a matching with maximum cardinality for a particular graph. A maximum matching is not necessarily unique. We say that a matching is complete in V if $|M| = |V|$.

6.1 Matchings in Bi-Partite Graphs

In the case of a bipartite graph $G = (C, X, E)$, where C is the set of constraints, X the set of unknown variables and the edge set E connects the variables to the constraints they appear in, the matching problem can be elaborated a bit more. For a given bi-partite graph, a matching may be complete either in $\mathcal{C}$, or in $\mathcal{X}$, or both or neither.

The following figure presents various matching for a bi-partite graph:

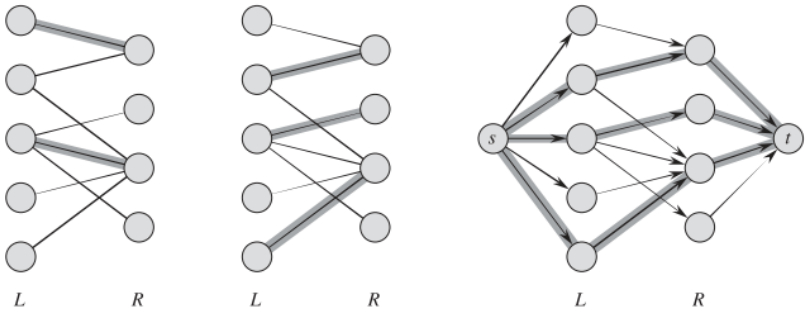


Figure 6.1: Matchings on a bi-parite graph

In the first and second subfigure we see incomplete matchings on the graph, both in terms of L and R. However, the second matching is maximum, since no matching with greater cardinality can be found in this graph.

We can interpret the notion of matching for our need to solve the structural graph as follows: if a variable $x \in \mathcal{X}$ is matched with a vertex $e \in M$, then we also say that it is matched to the constraint $c \in \mathcal{C}$ which e is incident onto, ie $(c, x) \in M$. This can be further extended to say that we can **calculate** x by using c. Hence, no other variable can be calculated using c and no other constraint can produce x.

The above interpretation applies a directionality in the bipartite graph. Let E' be a new edge set where

$$E' = \{(c, x) \in E : (c, x) \in M\} \cup \{(x, c) \in E : (x, c) \notin M\} \tag{6.1}$$

Through the procedure of matching, a new directed graph $G' = (C, X, E')$ is produced whose direction dictates the calculation order of the model structure and visualizes the information flow.

- if an edge connects a constraint to a variable, then this variable is calculated using this constraint.
- if an edge connects a variable to a constraint, then this variable is used by this constraint.

Paths are created, which connect variables as they contribute to the calculation of other variable, down the path.

6.2 Constraints in Matchings of Graphs Representing Real-World Systems

The matching process works very efficiently and without problems in undirected graphs. However, in bi-partite structural graphs resulting from real-world systems many problems arise, which restrict the feasibility of a complete matching in $\mathcal{X}$ and make the matching process difficult.

Notice how in (6.1) the directionality of each edge $e \in E$ is respected. We cannot match c to x ($(c, x) \in M$) if $(c, x) \notin E$. Similarly, we cannot expect a variable x to be used by a constraint c if $(x, c) \notin E$.

Another problem arises from the fact that there is no guarantee that the calculation process created by a matching and the corresponding directional graph can be back-tracked only to known and or measured variables, ie in the transpose graph $G'\top$ there is always a path from $x \in \mathcal{X}$ to $\mathcal{K}$. This leaves us with an incomplete calculation sequence which has unknown variables as input. In the general approach, the only way to verify the feasibility of the calculation sequence is to test if $\mathcal{X}$ is **reachable** from $\mathcal{K}$ in G'.

Moreover, G' is not guaranteed to be acyclic. The existence of cycles in G' is equivalent to the presence of systems of equations in the calculation sequence which have to be solved simultaneously. Since the constraints involved are not necessarily linear and may also be differential, systems of **ordinary differential equations** or even **differential algebraic equations** may need to be solved in real-time in order for the sequence to be calculable. Depending on our assumptions, this may or may not be feasible. Initial conditions may be unknown or the mathematical and numerical tools to solve such systems may not be available to the monitoring system. Hence, we may want to mark such loops as incalculable and restrict the calculation sequence beyond them.

Finally, we may want to favour some matchings over others. Reasons to do this are to avoid too many differentiations in the calculation sequence which enhance the noise or avoid solving a constraint in terms of a variable to which the constraint has small sensitivity or high cost of implementation.

Examples for the second case are the constraints

$$e_1 : \quad x_1 + \sin x_2 = 0 \tag{6.2a}$$

$$e_2 : \quad x_1^2 + x_2 = 0 \tag{6.2b}$$

$$e_3 : x_1 + 0.0001x_2 + 10x_3 = 0 \tag{6.2c}$$

- We would prefer solving e_1 for x_1 rather than for x_2, since the sensitivity of e_1 to x_2 for x_2 around $\pi/2$ becomes 0.
- We would prefer solving e_2 for x_2 for x_2 rather than from x_1, since calculating the power of 2 is cheaper than calculating a square root.

- Assuming comparable variable orders of magnitude, we would prefer solving e_3 for x_1 or x_3 rather than for x_2, since e_3 is much more sensitive to noise in x_1 and x_3 in contrast to noise from x_2.

The following example demonstrates how the above constraints affect the selection of the appropriate matching.

We are given the system

$$c_1: \quad x_1 - 100u_1 = 0 \tag{6.3a}$$
$$c_2: 3x_4 + 7x_2 - x_1 + x_5 = 0 \tag{6.3b}$$
$$c_3: \quad 5x_2 - x_3 = 0 \tag{6.3c}$$
$$c_4: \quad 10x_3 + 4x_4 = 0 \tag{6.3d}$$
$$c_5: \quad x_8 - 2x_7 - x_5 = 0 \tag{6.3e}$$
$$c_6: \quad x_8 - \frac{d}{dt}x_7 = 0 \tag{6.3f}$$
$$c_7: \quad x_9 \cdot x_1 - 1 = 0 \tag{6.3g}$$
$$c_8: \quad x_6 - x_9 = 0 \tag{6.3h}$$
$$c_9: \quad y_1 - \frac{d}{dt}x_9 = 0 \tag{6.3i}$$
$$c_{10}: \quad x_5 - x_6^3 + 1 = 0 \tag{6.3j}$$
$$c_{11}: \quad x_5 + x_1^{1.5} = 0 \tag{6.3k}$$

The structural graph of the system is depicted below.

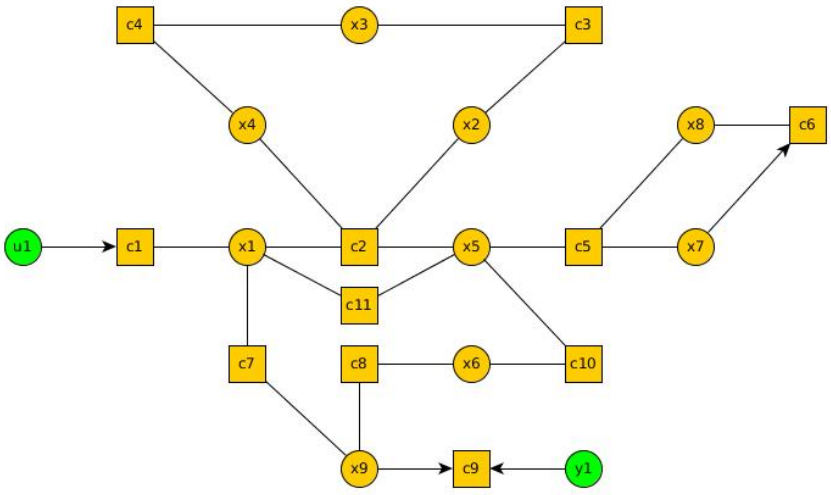


Figure 6.2: The bi-parite graph of the previous system

Notice how the causality of some equations pre-enforces directionality on some of the edges:

- Known variables (inputs and measurements, shown in green) can only be used as inputs in the constraints, not calculated by them.
- Variables differentiated in time can only be used as input to constraints, due to derivative causality (eg in c_6).

A good matching on the above graph is shown below. The edges which belong to the matching are emphasized and the rest of the edges are directed in such a way so as the matching is respected. We verify that at most one edge leaves a constraint vertex and at most one edge enters a variable vertex.

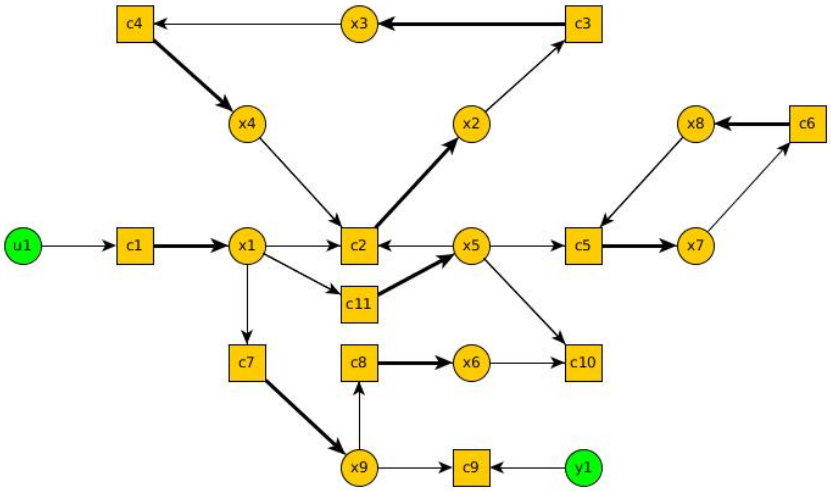


Figure 6.3: A matching of the structural graph with desirable properties

The above matching has some very desirable properties:

- It is a maximum matching on the varables set.
- The each edge of the resulting directed graph respects the directionality of the original graph.
- When possible, the least expensive calculation is selected, as in the case of x_6, which is calculated through c_8, rather than c_{10}

Naturally, the existence of loops in the resulting directed graph cannot be avoided, since this is a property of this particular, actual, real-world system. Whether it is possible to solve them depends on the available analytical and numerical tools we have.

7: Uncovering Analytical Redundancy

A common tool which Fault Tolerant Control uses in order overcome faults and failures is *redundancy*. System components redundant to each other offer the same functionality with regard to an objective. They may operate simultaneously and cooperatively, or they might only perform a hand-over of the objective in the case where one of them fails. *Hardware redundancy* is about including multiple hardware components to assure that within the pre-specified Mean Time Between Failures (MTBF) the overall system will have at least one component to carry out each necessary objective. Hardware redundancy in fw-UAVs is most prevalent in the form of duplicate sensors or actuators. However, there is another form of redundancy, *analytical redundancy*. This is about surplus of signal information. If our system structure contains more constraints than variables, ie $|\mathcal{C}| < |\mathcal{Z}|$, then the system is called **over-constrained**. In the case of sensor redundancy, there is more than one way (path) to observe (connect) some variables from already known variables, thus removing the need for an extra hardware sensor. In the case of actuator redundancy, uncovering redundancy of an output signal against input signals makes it obvious that there is more than one way to affect the specific output variable, thus alleviating the need for a redundant actuator. Given the nature of small-scale UAVs, it is very desirable to keep their cost and weight to a minimum. Hence, our goal is to find ways to replace hardware redundancy with analytical redundancy, as much as possible.

7.1 Solving the System Graph

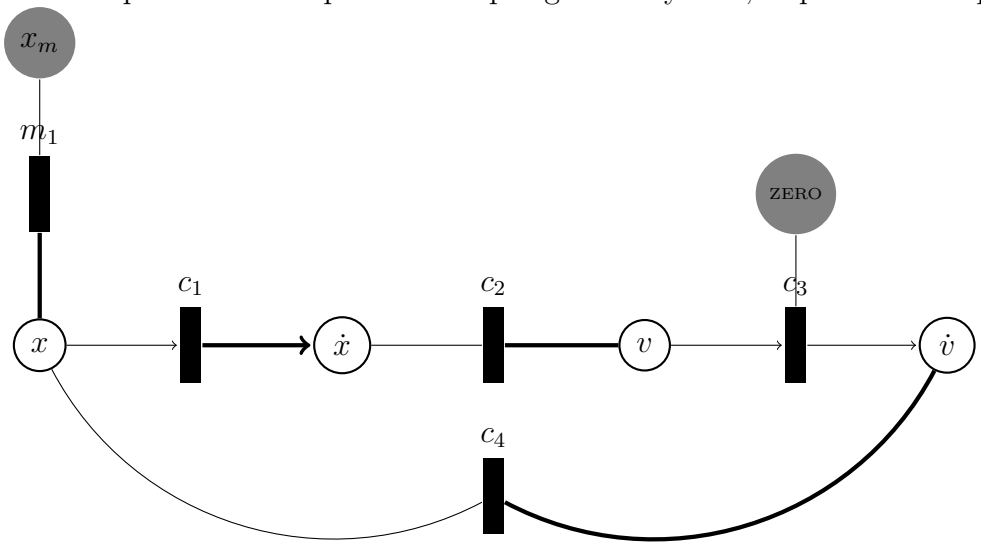
One of the most useful ways one can employ system structural analysis, is indeed to uncover methodically such redundancies within a large-scale system. Instrumental to this goal is the notion of graph matching.

Definition: Matching
A matching $\mathcal{M}$ is a subset of $\mathcal{A}$ such that the restrictions of the projections $p_{\mathcal{C}}$ and $p_{\mathcal{Z}}$ to $\mathcal{M}$ are injective.
$\forall e_1, e_2 \in \mathcal{A} : e_1 \neq e_2 \Rightarrow p_{\mathcal{C}}(e_1) \neq p_{\mathcal{C}}(e_2) \wedge p_{\mathcal{Z}}(e_1) \neq p_{\mathcal{Z}}(e_2)$

This means that given the set of edges $\mathcal{A}$, we select edges so that no two edges have a common vertex. If the cardinality of $\mathcal{M}$ is the maximum possible for a given $\mathcal{A}$, then the matching is called **maximal**. If $\mathcal{M}$ covers all the variables, then it is called **complete** with respect to $\mathcal{Z}$

By using the notion of matching, given a system graph, we can reveal analytical redundancy locations, in the form of unmatched constraints. These can be used to "double-check" the correct operation of our system. The difference of the LHS and RDS of an unmatched constraint equation is called a **residual**. Under nominal system operation all residuals evaluate to zero. If a system constraint is altered, then the unmatched constraint will no longer hold and the associated residual will deviate from zero.

For the previous example of the spring-mass system, a possible complete matching is:



The matching set is $\mathcal{M} = \{(m_1, x), (c_1, \dot{x}), (c_2, v), (c_4, \dot{v})\}$. The matched edges are denoted in the graph in bold.

Notice that the matching is complete in respect of the variables and incomplete in respect to the constraints. That leaves room for one residual generator at the node of c_3. The way we can take advantage of the matching is to evaluate the variables adjacent to the unmatched constraint through the given matching and then evaluate the residual.

In our case, the order of operations would be:

1. Evaluate x from x_m using m_1
2. Evaluate $\dot{x}$ from x using c_1
3. Evaluate v from x using c_2
4. Evaluate $\dot{v}$ from x using c_4
5. Evaluate the residual of c_3

Should everything function as expected the residual should always be zero (or close to the noise floor, for realistic systems). If not, this is an indication that the system has changed and our constraints no longer model it properly.

7.2 Residuals and Monitorability

In the last example, it is easy to understand that a deviation of the residual from zero would undoubtedly be caused by a change of the spring-mass constant $\frac{K}{m}$. The rest of c_1 and c_2 cannot be "faulty" since they represent the mathematical function of derivation. Any fault detected by the residual is isolated at c_4. However, let's suppose that c_1 and c_2 could represent part of the system model and we would like to be able to isolate the fault either to the system or the differentiators.

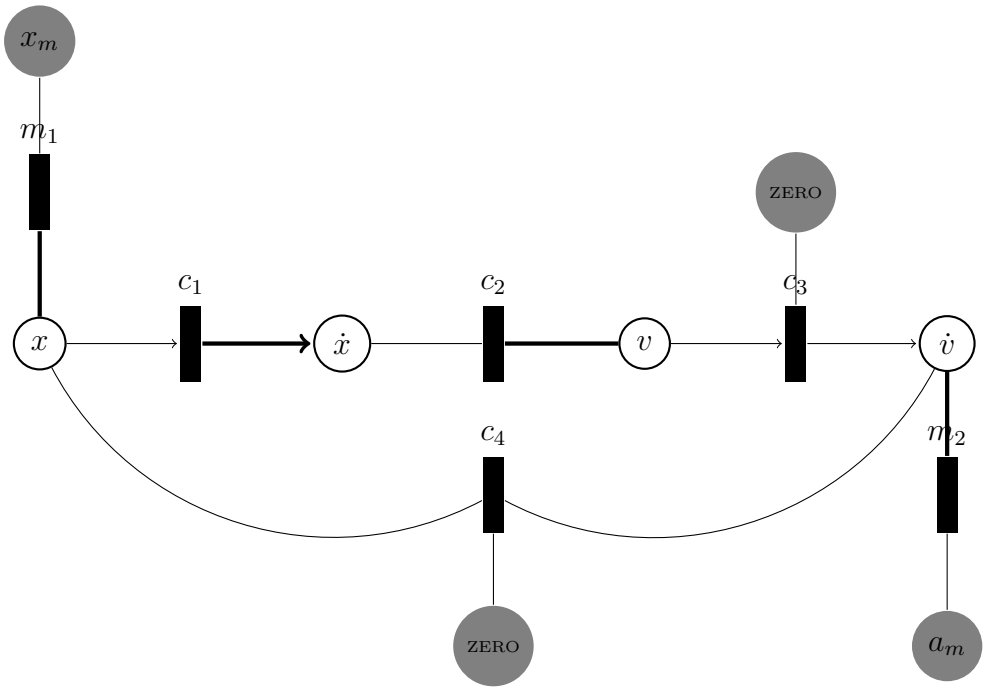
A systematic way to check for such possibility can be through, once again, Boolean mappings from the system constraints to the residuals they end up contributing to. If r is a residual generator in our system, which is affected by specific constraints, then we can create a mapping $F : r \leftarrow M$

M has as many rows as the residuals of our system and as many columns as the constraints. It is the "signature" of the effect the constraints have onto the residuals. If each column of M is unique, then any constraint violation is isolable through the system residuals, ie $c_i \in \mathcal{C}_{isolable} \Leftrightarrow \forall j \neq i : m_i \neq m_j$

For our example, we have

	m_1	c_1	c_2	c_3	c_4
r_1	1	1	1	1	1

Clearly no fault is isolable in this configuration; the columns of M are dependent. If we wanted a fault in c_4 to be isolable, we would need to modify our system, adding an additional accelerometer sensor to give us the necessary analytical redundancy:



The new Boolean mapping of the constraints onto the residuals is

	m_1	m_2	c_1	c_2	c_3	c_4
r_1	1	1	1	1	1	0
r_2	1	1	0	0	0	1

For demonstration purposes, if we assume that our sensors (m_1 and m_2) cannot fail, then the residual mapping will change to the one shown below and the spring-mass time constant fault can be isolated from differentiator faults.

	c_1	c_2	c_3	c_4
r_1	1	1	1	0
r_2	0	0	0	1

Using this structured method, we can specify components-constraints whose faults we are interested to isolate, check their isolability properties formally and if there constraint is not isolable then possibly locate injection points where new sensors are needed to achieve isolability.

It would also be interesting to assign weights to each graph edge, in order to model the value of direct quantity measurements against quantities derived through extensive manipulation of the already known variables.

8: Kinematic Equations

Considering the UAV as a rigid body, the standard kinematic equations will be used. Due to the scope of this analysis, the **Flat-Earth model** equations will be used, instead of a round Earth (eg WGS-84 model). This option was made since the intended UAV area of operations will be constrained over a small area. GPS coordinates will be used as in a Cartesian grid.

8.1 Position

We express the position vector as

$$\boldsymbol{p} = [n \; e \; d]^T \tag{8.1}$$

The variables n, e and d correspond to the North, East and Down direction, which constitute the primary axis of the NED (as it is called) coordinate system. We shall denote this frame as $\mathcal{F}_F$. The origin of the NED frame is arbitrarily located at the home of operations (or launch point) of every mission and placed on the surface of the Eearth. In contrast, the origin of the so-called body-axes $\mathcal{F}_B$ is the center of gravity point of the aircraft. Its x-axis is placed on the line of longitudinal symmetry, its y-axis starboard and the z-axis downwards, producing a right-handed frame, which can be seen in the following figure.

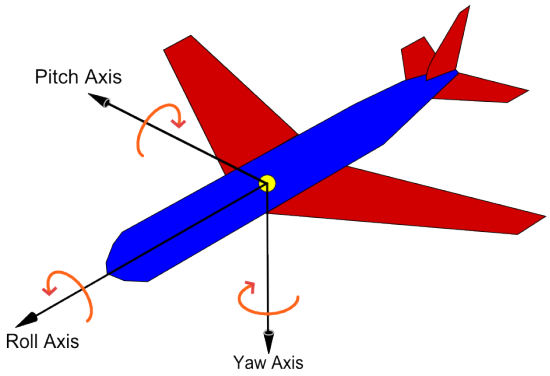


Figure 8.1: Aircraft body axes

The derivative of the position is

$$\dot{\boldsymbol{p}} = \boldsymbol{R}_b^T \boldsymbol{v_b} \tag{8.2}$$

We denote with R_b the transformation matrix **from the Inertial frame to the Body frame**. Its elements are:

$$\boldsymbol{R}_b = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & \sin\phi\cos\theta \\ \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi & \cos\phi\cos\theta \end{bmatrix} \tag{8.3}$$

Equation (8.2) can be broken down onto its three elements as

$$\begin{aligned} \dot{n} &= (\cos\theta\cos\psi)u + (-\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi)v \\ &\quad + (\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi)w && \text{(8.3a)} \\ \dot{e} &= (\cos\theta\sin\psi)u + (\cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi)v \\ &\quad + (-\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi)w && \text{(8.3b)} \\ \dot{d} &= (-\sin\theta)u + (\sin\phi\cos\theta)v + (\cos\phi\cos\theta)w && \text{(8.3c)} \end{aligned}$$

We see that the rotation matrix $\boldsymbol{R_B}$ is dependent upon three variables, ϕ , θ and ψ . These are explained below.

8.2 Orientation

We use Euler angle notation to express the orientation of the aircraft. Roll, pitch and yaw are denoted as ϕ, θ and ψ respectively.

$$\boldsymbol{\Phi} = [\phi\ \theta\ \psi]^T \tag{8.4}$$

Figure 8.2 has a visual representation of those angles. Take a note at the order of rotations: the standard order for aircraft applications, in order to come up with the body frame $\mathcal{F}_B$, starting from the NED frame is:

1. Rotate around NED z-axis by angle ψ, producing frame $\mathcal{F}_1$
2. Rotate around $\mathcal{F}_1$ y-axis by angle θ, producing frame $\mathcal{F}_2$
3. Rotate around $\mathcal{F}_2$ x-axis by angle ϕ, producing the $\mathcal{F}_B$ frame

This convention is called the *Tait-Bryan angles* [6]

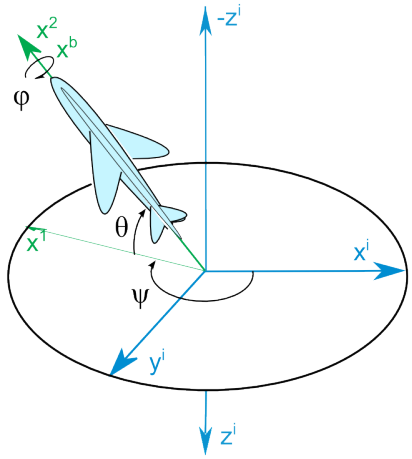


Figure 8.2: Euler angles

The time derivative of the Euler angles is

$$\dot{\mathbf{\Phi}} = \mathcal{E}(\mathbf{\Phi})\boldsymbol{\omega_b} \tag{8.5}$$

where $\mathcal{E}(\mathbf{\Phi})$ is a rotation matrix, dependent upon the current orientation.

The individual angle propagation equations are

$$\dot{\phi} = p + \tan\theta \sin\phi q + \tan\theta \cos\phi r \tag{8.5a}$$
$$\dot{\theta} = \cos\phi q - \sin\phi r \tag{8.5b}$$
$$\dot{\psi} = \frac{\sin\phi}{\cos\theta} q + \frac{\cos\phi}{\cos\theta} r \tag{8.5c}$$

and equivalenty, $\mathcal{E}(\mathbf{\Phi})$ is written as

$$\mathcal{E}(\mathbf{\Phi}) = \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix} \tag{8.6}$$

8.3 Angular velocity

We define the angular velocity vector as

$$\boldsymbol{\omega} = [p \; q \; r]^T \tag{8.7}$$

and its time derivative is

$$\dot{\boldsymbol{\omega}} = \frac{1}{\boldsymbol{J}} \left(\boldsymbol{T_b} - \boldsymbol{\omega} \times (\boldsymbol{J}\boldsymbol{\omega_b}) \right) \tag{8.8}$$

This equation incorporates the inertia matrix $\boldsymbol{J}$ and the input torque, expressed in the body axes T_b.

By spreading out the lines of the above vector equation we get

$$\dot{p} = \frac{1}{\Gamma} \left[J_{xz}(J_x - J_y + J_z)pq - (J_z(J_z - J_y) + J_{xz}^2)qr + J_z T_x + J_{xz} T_z \right] \tag{8.8a}$$

$$\dot{q} = \frac{1}{J_y} \left[(J_z - J_x)pr - J_{xz}(p^2 - r^2) + T_y \right] \tag{8.8b}$$

$$\dot{r} = \frac{1}{\Gamma} \left[((J_x - J_y)J_x + J_{xz}^2)pq - J_{xz}(J_x - J_y + J_z)qr + J_{xz} T_x + J_x T_z \right] \tag{8.8c}$$

To facilitate the calculation of cross products we may use this identity:

$$\boldsymbol{\omega} \times \boldsymbol{v} = \begin{bmatrix} 0 & -r & p \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \boldsymbol{v} \tag{8.9}$$

Where $\boldsymbol{v}$ is an arbitrary vector. Similarly,

$$\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{v}) = \begin{bmatrix} -r^2 - q^2 & pq & pr \\ pq & -p^2 - r^2 & qr \\ pr & qr & -p^2 - q^2 \end{bmatrix} \boldsymbol{v} \tag{8.10}$$

The inertia matrix is defined as

$$\boldsymbol{J} = \begin{bmatrix} \int (y^2 + z^2)\, dm & -\int xy\, dm & -\int xz\, dm \\ -\int xy\, dm & \int (x^2 + z^2)\, dm & -\int yz\, dm \\ -\int xz\, dm & -\int yz\, dm & \int (x^2 + y^2)\, dm \end{bmatrix} \tag{8.11}$$

Commonly, the inertia matrix in fixed-wing aircraft is considered to have zero elements in the x-y and y-z direction, since they are symmetric about the x-z plane.

$$\boldsymbol{J} = \begin{bmatrix} J_x & 0 & -J_{xz} \\ 0 & J_y & 0 \\ -J_{xz} & 0 & J_z \end{bmatrix} \tag{8.12}$$

$$\Gamma = J_x J_z - J_{xz}^2 \tag{8.13}$$

8.4 Linear Velocity

We define linear velocity and its components as

$$\boldsymbol{v_b} = [u\ v\ w]^T \tag{8.14}$$

Its time derivative is

$$\dot{\boldsymbol{v_b}} = -\boldsymbol{\omega_b} \times \boldsymbol{v_b} + \frac{\boldsymbol{F_b}}{m} \tag{8.15}$$

which is equivalent to

$$\dot{u} = rv - qw + \frac{F_b x}{m} \tag{8.15a}$$

$$\dot{v} = -ru + pw + \frac{F_b y}{m} \tag{8.15b}$$

$$\dot{w} = qu - pv + \frac{F_b z}{m} \tag{8.15c}$$

We need to define the wind velocity, in the body frame. This is the velocity of the air-mass, moving above the ground, expressed in the body axes.

$$\boldsymbol{v_w} = [u_w\ v_w\ w_w]^T \tag{8.16}$$

The resulting relative (air)speed of the aircraft is

$$\boldsymbol{v_r} = \boldsymbol{v}_B - \boldsymbol{v}_w \tag{8.17}$$

Relative velocity is a very important quantity in aeronautics, as every aspect of the aircraft's aerodynamic response depends on it, rather than the inertial speed.

Based on the relative velocity, we define three more quantities:

- the angle of attack, α
- the angle of sideslip, β
- the airspeed, V_a

$$\alpha = \tan^{-1}\left(\frac{w_r}{u_r}\right) \tag{8.18a}$$

$$\beta = \sin^{-1}\left(\frac{v_r}{V_a}\right) \tag{8.18b}$$

$$V_a = \|[u_r\ v_r\ w_r]^T\| \tag{8.19}$$

Using these angles, a new frame of reference can be constructed, the Stability frame $\mathcal{F}_S$. The relative speed components (expressed in the body frame) can be constructed from the airspeed (expressed in the stability frame) using the following rotation:

$$\boldsymbol{v}_r = \boldsymbol{S}^T \begin{bmatrix} V_a \\ 0 \\ 0 \end{bmatrix} \tag{8.20}$$

$$\boldsymbol{S} = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{8.21}$$

8.5 Mass Distribution

It is useful to model the mass distribution of our aircraft. It has a nominal mass m_{nom}, placed at the center of gravity, the origin of the body axes. Any extra weights, such as payloads, debris, or component detachment can be modeled with extra masses m_i. Under this definition, m_i is allowed to be negative.

$$m = m_{nom} + \sum m_i \tag{8.22a}$$

$$\boldsymbol{p}_{m,nom} = [0\ 0\ 0]^T \tag{8.22b}$$

$$\boldsymbol{p}_{CG} = \frac{1}{m}\left(\sum \boldsymbol{p}_i m_i\right) \tag{8.23}$$

Naturally, mass additions also affect the matrix of inertia of the aircraft. A mass m_e planed at the point $p_e = [x_e, y_e, z_e]^T$ perturbs the matrix of inertia by

$$\Delta\boldsymbol{J} = \begin{bmatrix} y_e z_e m_e & -x_e y_e m_e & -x_e z_e m_e \\ -x_e y_e m_e & x_e z_e m_e & -y_e z_e m_e \\ -x_e z_e m_e & -y_e z_e m_e & x_e y_e m_e \end{bmatrix} \tag{8.24}$$

However, this perturbation in the matrix of inertia has not been incorporated into the angular velocity derivatives equations (8.8), for reasons of simplicity.

8.6 Wind Disturbances

Another interesting model, primarily for simulation of autopilot software, is the one describing wind disturbances. It incorporates two sub-models: one for describing the constant

wind velocity, as a function of altitude and one for producing wind turbulence. Static wind is intuitively expressed in the inertial frame, while turbulence is usually expressed in the body frame, so care is required in order to avoid confusion and errors in notation and coding.

$$\boldsymbol{v_w} = \boldsymbol{v_{ws}} + \boldsymbol{v_{wg}} \tag{8.25}$$

We denote with $\boldsymbol{v_{ws}}$ the constant wind vector, with components.

$$\boldsymbol{v_{ws}} = \boldsymbol{R}_B^T [v_{ws,n}, v_{ws,e}, v_{ws,d}]^T \tag{8.26}$$

Usually we consider the vertical component of the wind to be zero ($v_{ws,d} = 0$)

Another useful expression for constant wind is:

$$\boldsymbol{v_{ws}} = \boldsymbol{R}_B^T V_{ws} [-\cos\theta_w, -\sin\theta_w, 0]^T \tag{8.27}$$

where V_{ws} is the wind overall magnitude (in m/s) and θ_w is the wind direction (0degrees for North wind)

This allows us to introduce an altitude model for the wind magnitude, commonly known as the Power Law:

$$V_{ws} = V_{ws,h_r} \left(\frac{h}{h_r} \right)^{\alpha} \tag{8.28}$$

This formula describes the increase of the wind magnitude as an exponential function of the altitude, given a measurement of wind magnitude V_{ws,h_0} at altitude h_0. α is the Hellmann exponent (roughness exponent), which describes the wind shear effect. Commonly, surface wind magnitude measurements are available for the altitude of 10m, so the above formula becomes.

$$V_{ws} = V_{ws,h_{10}} \left(\frac{h}{h_{10}} \right)^{\alpha} \tag{8.29}$$

Values for α vary with terrain morphology and wind turbulence. Some indicative values can be found in references [3], [24], [32], [33], but it is also claimed that the "1/7th power law" (using a value $\alpha = 1/7$) is an adequate approximation for most purposes. It should be noted, however, that most of the reviewed studies where oriented towards wind farm applications and wind models where validated up to a few hundred meters above ground.

The following table is pulled from [3]

Landscape type	Friction coefficient α
Lakes, ocean and smooth hard ground	0.1
Grasslands (ground level)	0.15
Tall crops, hedges and shrubs	0.20
Heavily forested land	0.25
Small town with some trees and shrubs	0.3
City areas with high rise buildings	0.4

Table 8.1: Hellmann Exponent Values over Various Terrain

An alternative model for wind shear can be found in [21]

Wind gusts can be modeled using Dryden transfer functions, as presented in [21], [4], [17] and [5] . Time responses can be generated by feeding unit variance white noise in the following filters. Note that airspeed is a parameter for these functions, but it can be replaced with the mean or cruise airspeed of the aircraft.

$$\boldsymbol{V_{wg}}(s) = \begin{bmatrix} \sigma_u \sqrt{\frac{2V_a}{\pi L_u}} \frac{1}{s+\frac{V_a}{L_u}} \\ \sigma_v \sqrt{\frac{3V_a}{\pi L_v}} \frac{s+\frac{V_a}{\sqrt{3}L_v}}{(s+\frac{V_a}{L_u})^2} \\ \sigma_w \sqrt{\frac{3V_a}{\pi L_w}} \frac{s+\frac{V_a}{\sqrt{3}L_w}}{(s+\frac{V_a}{L_w})^2} \end{bmatrix} \tag{8.30}$$

Typical values for the transfer function parameters for a small UAV can be found in [15] and are copied below.

Description	altitude (m)	L_u (m)	L_w (m)	σ_u (m/s)	σ_w (m/s)
low altitude, light turbulence	50	200	50	1.06	0.7
low altitude, moderate turbulence	50	200	50	2.12	1.4
medium altitude, light turbulence	600	533	533	1.5	1.5
medium altitude, moderate turbulence	600	533	533	3.0	3.0

Table 8.2: Gust Field Properties

9: Dynamic Equations

In this chapter the forces and moments exerted on the aircraft will be presented. The main dynamic components are gravity, aerodynamic reactions and propulsion (also referred to as thrust).

$$\boldsymbol{F}_b = \boldsymbol{F}_g + \boldsymbol{F}_a + \boldsymbol{F}_t \tag{9.1}$$

Before proceeding to any definitions, it is worth mentioning the control input variables. In normal airplane configurations, there are four control dimensions which the pilot can affect. These correspond to three sets of control surfaces (aileron, elevator and rudder) and the throttle command. The notation for these control variables is $\delta_a, \delta_a, \delta_t, \delta_r$. Essentially, these are the input variables to the aircraft model.δ_a, δ_e, and δ_r are usually expressed in degrees of deflection from the neutral (zero) angle and δ_t is normalized in the (0,1) range. Conventions for the positive direction of each control surface are not always consistent. In this text, the proposed positive direction of a control surface is the one which results in a positive moment around each primary axis (aileron - roll right, elevator - pitch up, rudder - yaw right).

9.1 Gravity

In this take, we assume a constant value for the acceleration of gravity, $g = 9.805416 m/s^2$ [1, p. 8]. The magnitude of the acceleration of gravity normally varies with altitude and geodetic coordinates, but a constant value is a valid approach for low altitude flight; unmodeled components are negligible to an adequate accuracy. The orientation of the gravity vector is always vertical, towards the center of the Earth, thus in the z-direction in the NED frame.

$$\boldsymbol{F}_g = \boldsymbol{R}_b \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{9.2}$$

$$F_{gx} = -\sin\theta \; mg \tag{9.2a}$$

$$F_{gy} = \sin\phi\cos\theta \; mg \tag{9.2b}$$

$$F_{gz} = \cos\phi\cos\theta \; mg \tag{9.2c}$$

$$g = g_0 \tag{9.3}$$

9.2 Aerodynamics

This is the defining part of the flight characteristics of an airplane. The aerodynamic response of the airplane is crucial for its stability as well as its maneuverability. Aerodynamic forces are presented in two frames of reference. The first is the body frame, whose corresponding forces, $\boldsymbol{F_a}$ are used for kinematic calculations.

$$\boldsymbol{F_a} = \begin{bmatrix} F_{ax} \\ F_{ay} \\ F_{az} \end{bmatrix} \tag{9.4}$$

Aerodynamic forces are considered to be exerted, under normal operation, at the *center of lift* ($\boldsymbol{p}_{CoL}$), a point located on the symmetry plane of the aircraft at one-quarter of the main wing chord.

However, for both historical and practical reasons, the flight characteristics of the airplane and the resulting forces (lift, drag and sideforce) are expressed in the stability frame (see eq.8.21). Models of aircraft aerodynamics are also parametrized so as to predict the aerodynamic forces in the stability frame as well.

$$\boldsymbol{F_s} = \begin{bmatrix} -F_D \\ F_Y \\ -F_L \end{bmatrix} \tag{9.5}$$

Thus, we can obtain $\boldsymbol{F}_a$ via the equation

$$\boldsymbol{F}_a = \boldsymbol{S}^T \boldsymbol{F}_s \tag{9.6}$$

$$F_{ax} = -\cos\alpha F_D - \cos\alpha \sin\beta F_Y + \sin\alpha F_L \tag{9.6a}$$
$$F_{ay} = -\sin\beta F_D + \cos\beta F_Y \tag{9.6b}$$
$$F_{az} = -\sin\alpha\cos\beta F_D - \sin\alpha\sin\beta F_Y - \cos\alpha F_L \tag{9.6c}$$

Aerodynamic moments need not be modeled in the stability frame. However, forces are generally not applied on the center of gravity of the aircraft, thus we need to model the moment they apply on it separately. Sources of such additional moments are the stability margin (how much forward than the center of lift lies the center of mass), height difference between center of mass and center of lift (for example in a high-wing aircraft) and mass additions and subtractions to the airframe.

$$\boldsymbol{T_a} = \begin{bmatrix} T_{ax} \\ T_{ay} \\ T_{az} \end{bmatrix} + (\boldsymbol{p}_{CoL} - \boldsymbol{p}_{CG}) \times \boldsymbol{F}_a \tag{9.7}$$

$$T_{ax,tot} = T_{ax} - dzF_{ay} + dyF_{az} \tag{9.7a}$$
$$T_{ay,tot} = T_{ay} + dzF_{ax} - dxF_{az} \tag{9.7b}$$
$$T_{az,tot} = T_{az} - dyF_{ax} + dxF_{ay} \tag{9.7c}$$

The next step in exploring the aerodynamic model is the definition of dynamic pressure. This quantity expresses the pressure exerted by the air moving around the airplane per unit of surface. It is proportional with the density of the air and with the square of the airspeed. Dynamic pressure is a crucial scaling factor for the rest of the aerodynamic equations

$$\bar{q} = \frac{1}{2}\rho V_a^2 \tag{9.8}$$

The air density, ρ, is dependent upon altitude but its sea level value is adequate for numeric calculations ($\rho_0 = 1.225 kg/m^3$)

At this point we are able to express the force components as the product of dynamic pressure, the wing surface S, and the aerodynamic coefficients of lift, drag and sideforce, C_L, C_D, C_Y. It is very important to appreciate that these coefficients are not constant values; instead, they are functions of the state of the flying body, albeit their dependence from some state variables is more dominant than others. A more detailed explanation is given further down.

$$F_D = \bar{q}SC_D \tag{9.9a}$$
$$F_Y = \bar{q}SC_Y \tag{9.9b}$$
$$F_L = \bar{q}SC_L \tag{9.9c}$$

$$C_D = C_D(\alpha, q, \delta_e) \tag{9.10a}$$
$$C_Y = C_Y(\beta, p, r, \delta_a, \delta_r) \tag{9.10b}$$
$$C_L = C_L(\alpha, q, \delta_e) \tag{9.10c}$$

The aerodynamic moments are defined in a similar fashion, with the additional inclusion of wingspan b and wing chord c. The notation for the moment coefficients may vary from source to source.

$$T_{ax} = \bar{q}SbC_l \tag{9.11a}$$
$$T_{ay} = \bar{q}ScC_m \tag{9.11b}$$
$$T_{az} = \bar{q}SbC_n \tag{9.11c}$$

$$C_l = C_l(\beta, p, r, \delta_a, \delta_r) \tag{9.12a}$$
$$C_m = C_m(\alpha, q, \delta_e) \tag{9.12b}$$
$$C_n = C_n(\beta, p, r, \delta_a, \delta_r) \tag{9.12c}$$

Finally, a model for the aerodynamic coefficients is provided. Undoubtedly, the most common and established approach is to use multivariable polynomials which are linear with respect to their coefficients. This representation lends itself nicely for parameter identification techniques, such as least-squares flavours, very common in real-world aircraft design and testing.

$$C_D = C_{D0} + C_{D\alpha}\alpha + C_{Dq}\frac{c}{2V_a}q + C_{D\delta_e}\delta_e \tag{9.13a}$$
$$C_Y = C_{Y0} + C_{Y\beta}\beta + C_{Yp}\frac{b}{2V_a}p + C_{Yr}\frac{b}{2V_a}r + C_{D\delta_a}\delta_a + C_{Y\delta r}\delta_r \tag{9.13b}$$
$$C_L = C_{L0} + C_{L\alpha}\alpha + C_{Lq}\frac{c}{2V_a}q + C_{L\delta_e}\delta_e \tag{9.13c}$$

$$C_l = C_{l0} + C_{l\beta}\beta + C_{lp}\frac{b}{2V_a}p + C_{lr}\frac{b}{2V_a}r + C_{l\delta_a}\delta_a + C_{l\delta_r}\delta_r \tag{9.14a}$$
$$C_m = C_{m0} + C_{m\alpha}\alpha + C_{mq}\frac{c}{2V_a}q + C_{m\delta_e}\delta_e \tag{9.14b}$$
$$C_n = C_{n0} + C_{n\beta}\beta + C_{np}\frac{b}{2V_a}p + C_{nr}\frac{b}{2V_a}r + C_{n\delta_a}\delta_a + C_{n\delta_r}\delta_r \tag{9.14c}$$

The coefficients which correspond to state variables ($C_{D\alpha}$,C_{Dq},$C_{Y\beta}$,C_{Yp}, C_{Yr},$C_{L\alpha}$,C_{Lq},$C_{l\beta}$, C_{lp},C_{lr},$C_{m\alpha}$,C_{mq},$C_{n\beta}$,C_{np},C_{nr}) in most cases have values such that the resulting forces and moments tend to revert the aircraft to straight and level flight. Hence they are called *stability derivatives.*

On the other hand, coefficients which correspond to the input variables ($C_{D\delta e}$,$C_{D\delta a}$,$C_{Y\delta r}$, $C_{L\delta e}$,$C_{l\delta_a}$,$C_{l\delta_r}$,$C_{m\delta_e}$,$C_{n\delta_a}$,$C_{n\delta_r}$) describe how the aircraft responds to control inputs and are hence called *control derivatives*

Not all of the above derivatives need to be used in the formulation of the aerodynamic model of a plane. Some of them might be statistically insignificant for certain airframes ([14]). On the other hand, longitudinal quantities usually do not incorporate coefficients of coefficients of lateral quantities and vice versa. This is in-line with the effort of decoupling the aircraft dynamics into a longitudinal and lateral plane, as much as possible, but there also are physical reasons, stemming from the aircraft geometry.

9.3 Propulsion

In the case of fixed-wing aircraft, most of the times the thrust is provided by a single propeller, acting upon the longitudinal axis of the aircraft, placed either in the front or the back of the fuselage. This discussion mainly draws from [2, p. 127] but extends its contents from other sources to adapt to an electric power plant. Propeller placement does not have an impact on the set of equation which form the model, even in the case of twin-propeller configurations, mounted on the wings.

Usually, thrust is considered to be produced only along the body x-axis, however, this approach may lack realism: it is not uncommon for the propeller to be mounted slightly tilted right and downwards, in order to minimize the counter-moment it produces and which affects the aircraft negatively.

$$\boldsymbol{F}_t = \begin{bmatrix} F_{tx} \\ F_{ty} \\ F_{tz} \end{bmatrix} \tag{9.15}$$

In case the propeller axis is aligned with the x-axis of the body frame, the following is also true:

$$F_{ty} = 0 \tag{9.15a}$$
$$F_{tz} = 0 \tag{9.15b}$$

Naturally, the motor and propeller produce moments. Similarly to the aerodynamic moment analysis, we also have to take into account the displacement of the trust vector from the x-axis of the body frame.

$$\boldsymbol{T_t} = \begin{bmatrix} T_{tx} \\ T_{ty} \\ T_{tz} \end{bmatrix} + (\boldsymbol{p}_{prop} - \boldsymbol{p}_{CG}) \times \boldsymbol{F}_p \tag{9.16}$$

$$T_{tx,tot} = T_{tx} - dzF_{ty} + dyF_{tz} \tag{9.16a}$$
$$T_{ty,tot} = T_{ty} + dzF_{tx} - dxF_{tz} \tag{9.16b}$$
$$T_{tz,tot} = T_{tz} - dyF_{tx} + dxF_{ty} \tag{9.16c}$$

We opt to consider in our model only the aforementioned counter-moment produced in the x-axis by the propeller and motor, and not gyroscopic effects caused by the spinning propeller disc:

$$T_{ty} = 0 \tag{9.16d}$$
$$T_{tz} = 0 \tag{9.16e}$$

With the aforementioned assumptions, the moment equations can be simplified into

$$T_{ax,tot} = T_{tx} \tag{9.16f}$$
$$T_{ay,tot} = dzF_{ax} \tag{9.16g}$$
$$T_{az,tot} = -dyF_{ax} \tag{9.16h}$$

In order to build a detailed and useful thrust model for our aircraft, it is common to break it down into two main components: The propeller, along with its aerodynamic characteristics (after all, it is an airfoil operating inside the atmosphere), and the power plant. In this document, due to the author's research interests, a model for an electric power plant will be presented, comprised of an electric brushless motor, an Electronic Speed Controller (ESC) and a battery.

9.3.1 Propeller Model

Much like the aerodynamics of the airplane lifting surfaces, propeller thrust is dependent upon a coefficient function (C_T), the air density (d) and its scale (propeller diameter D). Additionally, it is proportional to the square of its rotational speed (n, in revolutions per second).

$$F_{tx} = C_T \rho n^2 D^4 \tag{9.17}$$
$$n = \frac{\omega_p}{2\pi} \tag{9.18}$$

This time, the coefficient of thrust function has only one variable, the *advance ratio* (J). This quantity is the ratio of the aircraft airspeed to the speed the propeller "screws" itself into the air volume. It is a means to quantify how efficiently the propeller forces itself into the air volume. C_T generally doesn't have a simple shape. It is usually given by propeller manufacturers as a data set, which peaks at the nominal advance ratio and falls off at either side of the advance ratio range. This function can adequately model propeller stall, as well as the windmilling effect.

$$C_T = C_T(J) \tag{9.19}$$
$$J = \frac{V_a}{nD} \tag{9.20}$$

In a similar fashion, there is an expression for the power that is consumed by the propeller in order to produce thrust and it uses a coefficient of power (C_P).

$$P_p = C_P \rho n^3 D^5 \tag{9.21}$$
$$C_P = C_P(J) \tag{9.22}$$

A detail included in this model, not mentioned in most resources, is the equation which describes how the rotational speed of the propeller changes transiently in time. This is a differential equation of the rotational speed as a function of the difference between the available power from the motor (P_e) and the power consumed by the propeller. A necessary parameter is the moment of inertia along the x-axis of the whole engine-rotor and propeller system, which is summed up in the term J_p.

$$\dot{n} = \frac{1}{J_p} \frac{P_e - P_p}{n} \tag{9.23}$$

Finally, the expression of propeller moment is simply derived from the ratio of the propeller power over the propeller rotational speed.

$$T_{tx} = \frac{P_p}{\omega_p} \tag{9.24}$$

9.3.2 Electric Motor Model

The most wide-spread motor used in small UAVs is brushless outrunner permanent magnet electrical motor. Usually a 3-constant model is used, but for the sake of completeness, a 4-constant model will be used here, as presented in [10].

The input to the motor is the driving voltage, V_{in} and the output is the output power, P_e. Naturally, if there is a direct-drive configuration between the motor and the propeller, the motor rotational speed (n, in revolutions per second) is the same as that of the propeller.

$$n = K_v E_i \tag{9.25}$$
$$E_i = V_{in} - I_m R_m \tag{9.26}$$
$$P_e = E_i I_i \tag{9.27}$$
$$I_i = I_m - I_0 - \frac{E_i}{R_1} \tag{9.28}$$
$$P_{in} = V_{in} I_m \tag{9.29}$$

9.3.3 Battery and Electronic Speed Controller

Finally, an expression for the overall combination of the battery and ESC is given.

- V_{bat} is the internal battery voltage
- R_{bat} is the battery internal resistance
- R_S is the ESC in-line resistance
- δ_t is the throttle command, essentially modulating the output voltage

$$V_{in} = (V_{bat} - I_o(R_{bat} + R_S))\delta_t \tag{9.30}$$

10: Sensory Equipment Equations

In this chapter, the equations that describe the measurements of the various quantities and states via hardware sensors are laid out. Sensors often times use complicated physical phenomena to obtain a measurement of the desired quantity and sometimes the conversion process isn't straightforward. An effort to document relevant information is done here. No effort to model measurement errors is done here. Most of the following device descriptions correspond to their MEMS versions.

10.1 Inertial Measurements

The first and perhaps most used family of sensors in robotics are the inertial sensors. Fusing them allows the extraction of instantaneous orientation and acceleration data which in turn enables dead-reckoning. This is a sensor suite with large cumulative error, but with very high bandwidth, suitable for attitude control. In this chapter, we assume that the sensor axes are aligned with the body axes perfectly. A good source of relevant information is [28, p.25].

10.1.1 Accelerometer

An accelerometer installed onboard the aircraft measures the body-frame accelerations **except** the acceleration of gravity. As a result, when the aircraft is laid to rest on the ground the sensor records acceleration with magnitude of 1G, even though the aircraft isn't moving at all. This is similar to the apparent feeling a human has, feeling the force of his weight pulling him down when resting but feeling weightless while in free-fall.

In the ideal case where the accelerometer is installed at the center of gravity of the aircraft, the recorded acceleration is

$$\boldsymbol{a}_{b,m} = \boldsymbol{a}_b - \boldsymbol{R}_b\boldsymbol{g} \tag{10.1}$$
$$= \frac{\boldsymbol{F}_b}{m} - \boldsymbol{R}_b\boldsymbol{g} \tag{10.1a}$$

However, this is an ideal case. Most of the times, the sensor is placed in a point with coordinates $\boldsymbol{r}_{acc}$ relative to the center of gravity. The resulting measurement, affected by the aircraft rotational velocity will be

$$\boldsymbol{a}_{b,m} = \frac{\boldsymbol{F_b}}{m} - \boldsymbol{R}_b\boldsymbol{g} + \dot{\boldsymbol{\omega}}_b \times \boldsymbol{r}_{acc} + \boldsymbol{\omega}_b \times (\boldsymbol{\omega}_b \times \boldsymbol{r}_{acc}) \tag{10.2}$$

which can be expanded into

$$a_{m,x} = \frac{F_x}{m} + g\sin\theta - \dot{r}r_{acc,y} + \dot{q}r_{acc,z} + (-r^2 - q^2)r_{acc,x} + pqr_{acc,y} + prr_{acc,z} \tag{10.2a}$$
$$a_{m,y} = \frac{F_y}{m} - \sin\phi\cos\theta + \dot{r}r_{acc,x} - \dot{p}r_{acc,z} + pqr_{acc,x} + (-p^2 - r^2)r_{acc,y} + qrr_{acc,z} \tag{10.2b}$$
$$a_{m,z} = \frac{F_z}{m} - \cos\phi\cos\theta - \dot{q}r_{acc,x} + \dot{p}r_{acc,y} + prr_{acc,x} + qrr_{acc,y} + (-p^2 - q^2)r_{acc,z} \tag{10.2c}$$

The angular velocity derivatives can be further substituted using eq 8.8.

10.1.2 Gyroscope

Gyroscopes are sensors capable of measuring the rotational velocities of the rigid body they are attached onto. This measurement is unaffected by the location of placement.

$$\boldsymbol{\omega_m} = \boldsymbol{\omega_b} \tag{10.3}$$

10.1.3 Magnetometer

Magnetometers are sensors which measure the magnetic field of the Earth. At any given point on the surface of the Earth the vector field components are known and thus this sensor can be used to extract orientation information. The intensity of the magnetic field (measured usually in nanoTesla) is mostly constant, at about 45 000 nT. However, the direction of the magnetic flux lines changes radically from location to location [30], [31], [11]. If we express the magnetic field intensity as a vector quantity with North, East and Down components, for a given pair of coordinates

$$\boldsymbol{h} = [h_N, h_E, h_D]^T \tag{10.4}$$

then we define as declination the angle

$$dec = \tan^{-1}\left(\frac{h_E}{h_N}\right) \tag{10.5}$$

and the inclination as

$$inc = \tan^{-1}\left(\frac{h_D}{\sqrt{h_N^2 + h_E^2}}\right) \tag{10.6}$$

The measurement of the onboard magnetometer will then be

$$\boldsymbol{h}_m = \boldsymbol{h}_b = \boldsymbol{R}_b \boldsymbol{R}_{inc} \boldsymbol{R}_{dec} \begin{bmatrix} |\boldsymbol{h}| \\ 0 \\ 0 \end{bmatrix} \tag{10.7}$$

10.1.4 AHRS

There also are hardware solutions which incorporate all three aforementioned sensors and extract an estimation of the Euler angles set.

$$\boldsymbol{\Phi_m} = \boldsymbol{\Phi} \tag{10.8}$$

10.2 GPS Measurements

In order to have a reading of the absolute position of the aircraft in the inertial frame, be it the local NED frame or the global frame, defined by latitude and longitude, a GPS sensor is required. This is the only reasonable option, as radio-location is not a viable option for low-cost and low-weight robotic platforms. Whether the system will use a simple GPS receiver, a multi-constellation receiver, differential measurements or even RTK packages improves only the measurement accuracy.

Common quantities reported by a GPS receiver are the longitude, latitude and altitude set. However, these do not correspond to the NED frame. In order to translate them to the north, east and down quantities we need to obtain the data on the curvature of the Earth at the location of interest. The following discussion should be adequate for the needs of a NED frame spanning a few kilometers to each direction of the starting coordinates.

10.2.1 Earth Model

This discussion draws from [22], specifically chapter 3 and equations of p. 7.5. According to the WGS-84, the Earth is modeled as an ellipsoid which is flatter on the poles and

symmetric on the z-axis (the Earth's rotation axis). The semi-major axis is

$$a = 6\ 378\ 137m \tag{10.9}$$

and the flattening, defined as

$$f = \frac{a-b}{a} \tag{10.10}$$

is equal to

$$\frac{1}{f} = 298.257222101 \tag{10.11}$$

b is the semi-minor axis of the ellipsoid and is calculated from the above equations and is equal to

$$b = 6\ 356\ 752.3142m \tag{10.12}$$

The first eccentricity, e, is defined as

$$e^2 = 2f - f^2 \tag{10.13}$$

With these constants available, for any given point on the surface of the Earth with latitude ϕ, we can calculate the radius of curvature in the prime vertical

$$R_N = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}} \tag{10.14}$$

and the curvature in the meridian

$$R_M = \frac{a(1-e^2)}{(1 - e^2 \sin^2 \phi)^{3/2}} \tag{10.15}$$

Finally, a small change of latitude $\Delta\phi$ causes a shift in the north-south direction

$$\Delta n = R_M \sin \Delta\phi \tag{10.16}$$

and a small change of longitude $\Delta\lambda$ causes a shift in the east-west direction

$$\Delta e = R_N \sin \Delta\lambda \tag{10.17}$$

For such small angles (in the order of hundredths of a degree), the simplification $\sin \Delta\phi = \Delta\phi$ (in radians) might be acceptable.

For reference, at a latitude of 45degrees, a change of 0.01degree to the North corresponds to a distance of 1111.318m and the same change Eastwards corresponds to 1115.062m.

Sidenote: As of writing, the Google Earth 7.1.2 application for Linux reports a distance of about 1113m between the coordinates (45.00,00.00) and (45.01,00.00) and (curiously) a

distance of about 788m(!) between the coordinates (45.00,00.00) and (45.00,00.01). Until further investigation, this results seems to be erroneous.

In case simpler calculations are required, one could use the geometric mean of the semi-major and semi-minor axis

$$\bar{R} = \sqrt{a * b} \tag{10.18}$$

and multiply it by the change in angle, regardless of direction.

$$\Delta n = \bar{R}\Delta\phi \tag{10.19}$$

For the previous example, the above equation yields 1111.327m, which is very acceptable for many applications.

Regarding altitude, GPS receivers report their vertical distance from the surface of the ellipsoid. Since the local NED frame is placed at the starting altitude, this altitude (h_0) should be used to initialize the frames. One could take altitude into account when calculating Cartesian displacements in the NED frame. This is as simple as adding altitude to the local Earth radius. This is demonstrated below.

Summing up, we do the following steps to convert from geodetic coordinates to local coordinates:

1. Note the starting coordinates ϕ_0, λ_0 and altitude h_0
2. Calculate the radii of curvature $R_N(\phi_0)$ and $R_M(\phi_0)$

For each GPS reading (ϕ, λ, h)

1. Calculate the differences $\Delta\phi = \phi - \phi_0$, $\Delta\lambda = \lambda - \lambda_0$ and $\Delta h = h - h_0$
2. Calculate $n = (R_M + h) \sin \Delta\phi$
3. Calculate $e = (R_N + h) \sin \Delta\lambda$
4. Calculate $d = -\Delta h$

Having done the above necessary conversions, we can treat the GPS receiver as a sensor which reports displacement in the local NED frame

$$\boldsymbol{p_{gps}} = \boldsymbol{p} \tag{10.20}$$

Additionally, most GPS receivers report ground velocities

$$v_{gps} = \|[\dot{n}\ \dot{e}]'\| \tag{10.21}$$

and the direction of travel, also referred to as the *course angle*

$$\chi_{gps} = \text{atan2}\,(\dot{e}, \dot{n}) \tag{10.22}$$

10.3 Atmospheric Measurements

10.3.1 Atmosphere Model

The primary resource for atmospheric models is [1]. This technical report is the most accessible and adequately complete document among its "competitors". It provides models for all of the components of the atmosphere, ranging from temperature gradients to molecular synthesis of the air. It partitions the atmosphere into several layers, but we are interested only in the first one, spanning from 0m to approximately 11km. This range in adequate for the majority of UAV operations.

Note: The following discussion uses the gravitational model from [1, eq. 17]. This model differs from the one presented in [22] in a noticeable way. [1] uses the *Standard Gravity* for the gravitational acceleration on the surface of the Earth, established by the International Committee on Weights and Measures in 1901.

$$g_0 = 9.80665\ m/s^2 \tag{10.23}$$

This number is used uniformly for all latitudes. It is known that this is a simplification. However, we can proceed to the rest of the calculations and if need arises, switch g_0 for the more detailed model from [22, eq. 4-1].

Additionally, the US Standard Atmosphere gravity model with respect to altitude uses the inverse square law for gravity decay

$$g(Z) = g_0 \left(\frac{r_0}{r_0 + Z} \right) \tag{10.24}$$

where

- Z is the geometric altitude above the surface, as reported by GPS systems
- r_0 is the radius of the Earth at a latitude where g_0 takes its standard value of $9.80665\ m/s^2$

The respective altitude model in WGS-84 is given in equation (4-3), for relatively low altitudes. Again, one can opt to use that model.

Where all this matters, is in the definition of the *geopotential altitude*, H. This is expressed in *geopotential meters* (m'), a unit of altitude, lifting a mass by which, always requires the same amount of work, no matter at which altitude this work is performed. Essentially, this expresses how easier it gets to lift masses in the weakening gravity as altitude increases and, in turn, this affects how less densely atmosphere is packed as altitude increases.

$$H = \frac{1}{g_0} \int_0^Z g \cdot dZ \tag{10.25}$$

Combining equations 10.23, 10.24 and 10.25 we get the following conversion expressions between geometric and geopotential altitude

$$H = \frac{r_0 \cdot Z}{r_0 + Z} \tag{10.26}$$

$$Z = \frac{r_0 \cdot H}{r_0 - H} \tag{10.27}$$

Next, we need to include a temperature model as a function of altitude.

$$T = T_0 + L_0 \cdot (H - H_0) \tag{10.28}$$

where

- H_0 is our initialization altitude
- T_0 is the temperature at our initialization altitude. Its nominal value at sea-level is 288.15 K
- $L_0 = -0.0065\ K/m$ is the temperature gradient

The equation above describes how temperature drops as altitude increases.

Finally, we can express the variation of barometric pressure as a function of geopotential altitude

$$P = P_0 \left(\frac{T_0}{T(H)} \right)^{\left(\frac{g_0 \cdot M_0}{R^* \cdot L_0} \right)} \tag{10.29}$$

where

- P_0 is the pressure at our initialization altitude. According to the US Standard Atmosphere, at sea-level its value is $P_0 = 101325\ N/m^2 = 1013.25\ mb$
- $M_0 = 0.0289644\ kg/mol$ is the mean molecular weight of the air
- $R^* = 8.31432\ N \cdot m/(mol \cdot K)$ is the gas constant

Most importantly, one can solve this equation for H and obtain the geopotential altitude by measuring the barometric pressure.

$$H = \frac{T_0}{L_0} \left(\left(\frac{P}{P_0} \right)^{\frac{g_0 \cdot M_0}{R^* \cdot L_0}} - 1 \right) + H_0 \tag{10.30}$$

Air density can be obtained from the air pressure model

$$\rho = \frac{P \cdot M_0}{R^* \cdot T} \tag{10.31}$$

10.3.2 Pressure

Having defined the atmospheric model, we can now introduce a barometer sensor, which measures the barometric pressure at the UAV altitude.

$$P_{bar} = P \tag{10.32}$$

10.3.3 Temperature

Also, thermometers are available, which record the atmospheric temperature and are necessary for the extraction of altitude, through pressure readings. Care is needed however, as electronics are known to warm-up and raise the temperature in the enclosed spaces they are placed. However, this discussion belongs to an fault analysis and will not be expanded upon here.

$$T_m = T \tag{10.33}$$

What will be noted here is the fact that thermometers can be used to measure the temperature of components as well, such as batteries, ESCs or motors and provide a health status of those components.

10.4 Air Data

For an efficient and safe aircraft flight, air data measurements are required. These measurements refer to airspeed $\|V_a\|$, angle of attack α and angle of sideslip β.

Airspeed is measured using Pitot tubes, which take into account Bernoulli's equation to convert dynamic air pressure into airspeed readings. Essentially, a barometer is used to perform this reading.

$$P_t = P + \frac{\rho V_a^2}{2} \tag{10.34a}$$

$$P_{t,m} = P_t \tag{10.34b}$$

$$\tag{10.35}$$

Aerodynamic angles α and β can be measured using appropriate instruments, such as wind vanes, but such sensors are seldom found in medium and small UAVs, due to their size and weight.

$$\alpha_m = \alpha \tag{10.36}$$

$$\beta_m = \beta \tag{10.37}$$

10.5 Distance Measurement

Instruments which measure the distance of the UAV from the ground belong to this category. Common types are ultrasonic sensors, laser range finders and LIDARs. Their range varies from a few meters to a few hundred meters and their price range varies accordingly. What all these sensors have in common is that their measurements have a fixed range, varying from $d_{rf,min}$, the minimum reading, up to $d_{rf,max}$, in terms of absolute magnitudes.

$$d_{rf} = \begin{cases} d_{rf,min} & , d > d_{rf,min} \\ d & , h \in [d_{rf,min}, d_{rf,max}] \\ d_{rm,max} & , d < d_{rf,max} \end{cases} \tag{10.38}$$

Some confusion may be caused by the direction of the comparison ranges, but keep in mind that using the NED frame, d is negative when the airplane is above ground altitude.

10.6 Voltage Measurement

Voltage Measurement is a very useful means to assess the remaining energy capacity of an onboard battery. Even though the relation between remaining energy and output voltage is not linear, useful conclusions can be extracted.

In general, any voltage level can be measured by such an instrument

$$V_{i,m} = V_i \tag{10.39}$$

but in the specific case of the battery voltage measurement we have

$$V_{B,m} = V_B - I_o R_B \tag{10.40}$$

10.7 Current Measurement

Similarly to voltage measurements, current measurements are available as well. These are useful to monitor the power consumption of the aircraft and, in extension, are a valid means to approach the thrust produced by the power plant.

Usually, current sensors are placed on the battery leads, since their most common occurrence measures only DC current.

$$I_{o,m} = I_o \tag{10.41}$$

10.8 Motor Angular Velocity Measurement

Finally, a quantity with great informational content is the rotational velocity of the motor shaft. This can be measured with commercial RPM sensors and can provide information on the thrust produced or even the health status of the motor itself.

$$n_m = n \tag{10.42}$$

11: Other Future Directions

11.1 Residual Analysis

Using formal ways to tackle large system complexity was shown to be beneficial for the purposes of Fault Detection and Diagnosis. In this text, the analysis ended with the production of residuals, which contain information on potential system faults. Inevitably, in order to extract final conclusions on system faults, these residuals have to be analyzed to recognize the appearance of fault signatures inside them. A plethora of methods is employed to achieve this goal, ranging from simple thresholding up to statistical tests and pattern recognition.

11.2 Separating Faults from Disturbances

Under the right system model, faults and disturbances can be handled in a similar fashion, and in the same ways that faults are detected and isolated and identified, so can disturbances be. This can prove beneficial for assessing the intensity of environmental disturbances.

11.3 Hybrid Systems

Hybrid systems have recently received a lot of attention in FDD and reconfigurable control applicaitons. Indeed, they offer an intuitive and formal way to handle events in the system, introduce temporal causality and define different system states, which in turn need different handling, in terms of monitoring, control and mission planning. Discrete Event Systems (DES) used along with the results from FMEA can also be used to back-track fault apparitions in time and have better estimates on fault isolation.

11.4 Fault Identification and Reconfigurable/Restructurable Control

If component failures can be precisely isolated, then the task of on-line system identification can be rendered a lot easier. If the system structure has not changed, then on-line identification algorithms can be focused on specific subsets of input/output signals to detect parameter changes with much greater accuracy. Thus adaptive controllers have an easier task to re-tune their gains for effective control. If structural changes are indeed detected, then this information can be used to perform re-structurable control.

11.5 Mission Planning

By having a constantly updated system structure/model, its limits and capabilities can be compared against mission goals to assess their feasibility in real-time. This not only adds an extra layer of safety for the UAV, since it enables it to abort un-achievable missions, but it may also enable it to re-specify its mission automatically in order to complete as much of the original mission as possible while being able to return to base at the same time.

Bibliography

[1] US Standard Atmosphere, 1976. Technical report, National Oceanic and Atmospheric Administration, National Aeronautics and Space Adminisration, United States Air Force, 1976.

[2] D Allerton. *Principles of flight simulation*. Wiley, 2009.

[3] Francisco Bañuelos Ruedas, César Angeles-Camacho, and Sebastián Rios-Marcuello. Methodologies Used in the Extrapolation of Wind Speed Data at Different Heights and Its Impact in the Wind Energy Resource Assessment in a Region. In Gastn Orlando Suvire, editor, *Wind Farm - Technical Regulations, Potential Estimation and Siting Assessment*, chapter 4. InTech, June 2011.

[4] T. R. BEAL. Digital simulation of atmospheric turbulence for Dryden and von Karman models. *Journal of Guidance, Control, and Dynamics*, 16(1):132–138, January 1993.

[5] RW Beard and TW McLain. *Small unmanned aircraft: Theory and practice*. 2012.

[6] Paul Berner and D Ph. Orientation, Rotation, Velocity, and Acceleration and the SRM. Technical Report June, 2008.

[7] M. Blanke. *Diagnosis and Fault-Tolerant Control*. Engineering online library. Springer, 2003.

[8] M. Blanke, M. Staroswiecki, and N.E. Wu. Concepts and methods in fault-tolerant control. In *American Control Conference, 2001. Proceedings of the 2001*, volume 4, pages 2606–2620 vol.4, 2001.

[9] Mogens Blanke and Torsten Lorentzen. *SaTool - a Software Tool for Structural Analysis of Complex Automation Systems*, pages 673–678. Elsevier Science, 2006.

[10] John Carri. A four-constant model for electric motors (Draft No. 2). Technical Report 2, 2007.

[11] NOAA National Geophysical Data Center. Geomagnetism. `http://www.ngdc.noaa.gov/geomag/geomag.shtml`. [Online; accessed 13-January-2015].

[12] Boskovic J.D., Mergstrom S.E., and Mehra R.K. Retrofit reconfigurable flight control in the presence of control effector damage. In *American Control Conference*.

[13] Eliahu Khalastchi, Meir Kalech, and Lior Rokach. Sensor fault detection and diagnosis for autonomous systems. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '13, pages 15–22, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems.

[14] V Klein and EA Morelli. *Aircraft system identification: theory and practice*. AIAA, 2006.

[15] Jack W. Langelaan, Nicholas Alley, and James Neidhoefer. Wind Field Estimation for Small Unmanned Aerial Vehicles. *Journal of Guidance, Control, and Dynamics*, 34(4):1016–1030, July 2011.

[16] Charles E. Leiserson, Thomas H. Cormen, Clifford Stein, and Ronald Rivest. *Introduction to Algorithms*. The MIT Press, 3 edition, 2009.

[17] MathWorks. Aerospace toolbox documentation: Dryden wind turbulence model (continuous). `http://www.mathworks.com/help/aeroblks/drydenwindturbulencemodelcontinuous.html`, 2015. [Online; accessed 8-January-2015].

[18] Peter S Maybeck and Richard D Stevens. Reconfigurable flight control via multiple model adaptive control methods. *Aerospace and Electronic Systems, IEEE Transactions on*, 27(3):470–480, 1991.

[19] N. Meskin, K. Khorasani, and C.A Rabbath. Hybrid fault detection and isolation strategy for non-linear systems in the presence of large environmental disturbances. *Control Theory Applications, IET*, 4(12):2879–2895, December 2010.

[20] Tousi M.M. and Khorasani K. Optimal hybrid fault recovery in a team of unmanned aerial vehicles. *Automatica*, 48(2):410–418, 2012.

[21] David J. Moorhouse and Robert J. Woodcock. Background Information and User Guide for MIL-F-8785C, Military Specification - Flying Qualities of Piloted Airplanes. Technical report, Air Force Wright Aeronautical Laboratories, 1982.

[22] WM Mulaire. Department of Defense: World Geodetic System 1984. Technical report, National Imagery and Mapping Agency, 2000.

[23] T Ozyagcilar. Calibrating an eCompass in the presence of hard and soft-iron interference. Technical report, Freescale Semiconductor, 2012.

[24] EW Peterson and JP Hennessey Jr. On the use of power laws for estimates of wind power potential. *Journal of Applied Meteorology*, 1978.

[25] William Premerlani. Magnetometer Offset Cancellation: Theory and Implementation, revisited. Technical report, 2011.

[26] Marc Rauw. FDC 1.2-A Simulink Toolbox for Flight Dynamics and Control Analysis. Technical report, 2001.

[27] Marc Steinberg. Historical overview of research in reconfigurable flight control. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 219(4):263–275, 2005.

[28] BL Stevens and FL Lewis. *Aircraft control and simulation*. John Wiley & Sons, illustrate edition.

[29] M.M. Tousi and K. Khorasani. Robust observer-based fault diagnosis for an unmanned aerial vehicle. In *Systems Conference (SysCon), 2011 IEEE International*, pages 428–434, April 2011.

[30] Wikipedia. Magnetic declination. `https://en.wikipedia.org/wiki/Magnetic_declination`, 2015. [Online; accessed 13-January-2015].

[31] Wikipedia. Magnetic dip. `https://en.wikipedia.org/wiki/Magnetic_dip`, 2015. [Online; accessed 13-January-2015].

[32] Wikipedia. Wind gradient. `http://en.wikipedia.org/wiki/Wind_gradient`, 2015. [Online; accessed 8-January-2015].

[33] Wikipedia. Wind profile power law. `http://en.wikipedia.org/wiki/Wind_profile_power_law`, 2015. [Online; accessed 8-January-2015].

[34] Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32(2):229 – 252, 2008.