A TECHNICAL REPORT
MODELING AND SIMULATION IN MATLAB/SIMULINK
The Mathematical Modeling, Trimming and Linearisation of an transsonic Missile and a further Control Law Development

School of Aerospace, Transport and Manufacturing
Autonomous Vehicle Dynamics and Control, Msc.
Module: UAS Modelling and Simulation
Cranfield University
Cranfield, United Kingdom

Johannes Autenrieb
December 2018

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Backround

The military defence market is one of the most innovative technology businesses in the technology sector. The area came up with innovations such as the Internet, Virtual reality and Satellite Navigation.[8]
Missiles are an essential entity in modern warfare scenarios. Their ability to protect certain areas against attacks and to destroy targets from long distances makes them to a powerful tool in combat situations, to provide the mentioned abilities in a conflict the weapons need to be equipped with modern sensors and control systems. This assignment discusses the modelling and simulation of a provided missile system in MATLAB/Simulink. This work will present an overview of the whole procedure of the modelling and simulation and details about different approaches in SIMULINK. Each approach and outputted results will be also discussed during this work.

1.2 Problem Statement

The problem presented in this assignment is the modelling and simulation of a provided missile model in MATLAB/Simulink. This work is mainly focusing on modelling all relevant systems to analyse the flight dynamics behaviour and responses of the missile. Furthermore, a control law shall be developed for the modelled system. For that goal suitable algorithm and fitting SIMULINK models should be created and afterwards discussed.

1.3 Aim and objectives

This assignment aims to model and simulate the flight dynamic behaviour of a specified missile by using MATLAB/Simulink. This general aim can be separated into six primary objectives:

- Describing the general model conceptualisation
- Deviate the flight dynamic equations of a missile
- Establishing the SIMULINK model of the missile
- Verification of the created SIMULINK model
- Searching Trimming states of the missile
- Linearisation of flight dynamic equations
- Analysing the Stability behaviour of the missile
- Designing an Autopilot for the missile

Chapter 2

Model Conceptualisation

2.1 Introduction

This section deals with the conceptual arrangement of the general MATLAB/Simulink model. It explains the different subsystems in the model and describes their interactions. The goal of this section is it to deliver an overall overview of the subsystem arrangement and their functional tasks. Therefore is this section introduce each of them separately. Furthermore, the first general system sketch will be presented and explained.

2.2 Theory

The classical control theory delivers the knowledge we need to design the first model sketch of the missile. In general, it can be observed that technical systems which are interacting with their environment are forced to use a control system which gives feedback that can be used to adjust the behaviour. This is called the principle of feedback. In a feedback control system, the states which need to be monitored are looped back and compared to the aspired value. The disparity between the aspired and the actual measured value is used to calculate an improving control action.[4] In general, the control system can be described as a system which commands three bigger subsystems: the controller subsystem, the actuator subsystem and the physical system which needs to be controlled. A sketch of the described architecture is given in figure 2.1.

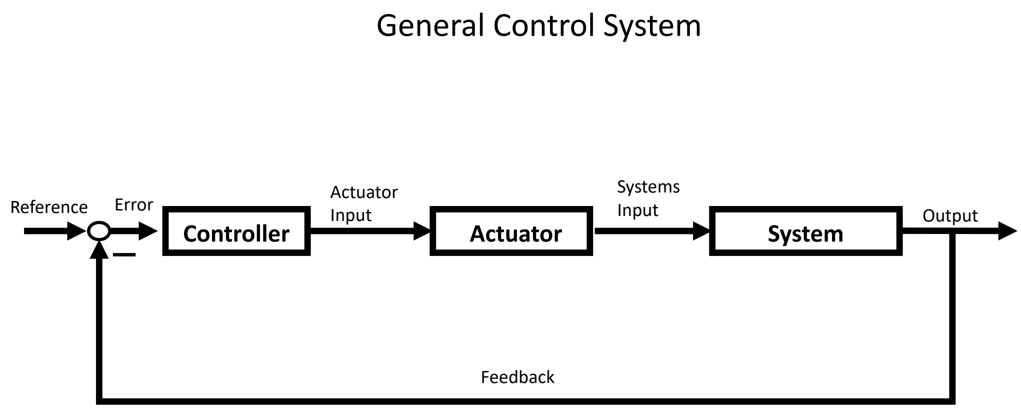

Figure 2.1: A sketch of a general control system architecture [4]

The introduced control system architecture can be utilised and specified for all kind of technical problems. Concerning the problem of flight control, the control system architecture can be used in the same way but can be more specified at some points. For a flight control system, the controller subsystem is called the autopilot. Also, the physical system can be more specified as the flight dynamic model which delivers state values that can be measured. The embedded control system for the missile is shown in figure 2.2.

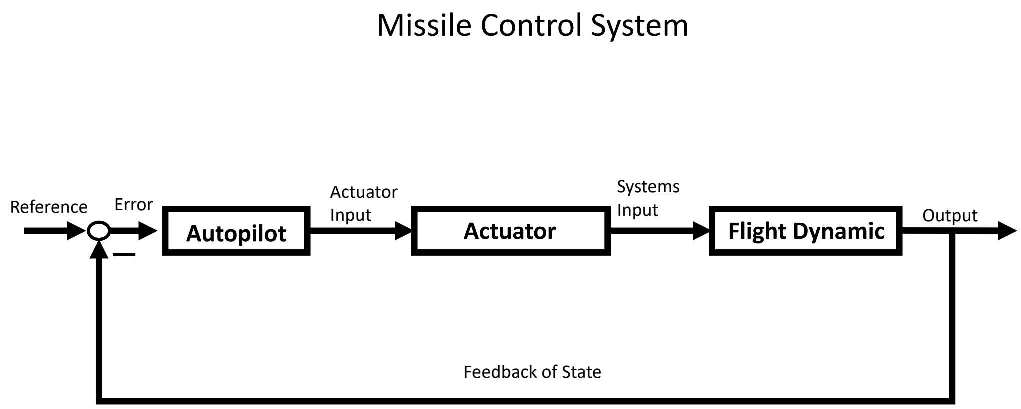

Figure 2.2: A sketch of the control system architecture for a missile system

2.3 The Sub-Systems

2.3.1 Controller

The responsibility of the controller subsystem is to influence the output of the physical system in a way that the system sufficiently reaches a particular state. To achieve that goal the controller evaluates the system behaviour continuously and compare the measured values with the desired values. If both amounts are not equal, the controller takes action to bring the missile in the desired position. A wrongly designed controller can lead to steady state errors or diverging dynamic responses, also called unstable behaviour. To fit the purposes general control systems utilises controller blocks. Four very popular classes are P,- I-,D- and PID-Controller. The most common used controller, in the aerospace sector, is the PID-Controller which combines the features of P-,I -and D-controller blocks. Once the controller processed the everything it creates input values which are used by the actuators.

2.3.2 Actuator

To influence the flight state of a missile the control system utilising the aircraft control surfaces. These surfaces are creating aerodynamical forces and moments around the centre of gravity which is leading to position and attitude changes. Since the control system needs to adjust the control surface position, actuators need to be used. Since a lot of different actuator types are available, the different system behaviour needs to be accounted for the correct calculation of the needed input. That can be for example done with the definition of a transfer function. The output of the actuator system is the control surface angles which can be used as control inputs of the flight dynamic model.

2.3.3 Flight Dyanmic

As already mentioned, the physical system, in this example, is the flight dynamic behaviour of the missile. Since the physical missile is not provided, the demanded physical behaviour need to be modeled in a mathematical construction. For that, the equations of motions of the missile need to be defined and computed. All physical influences on the aircraft and its behaviour need to be considered. The inputs of the actuator system need to be processed and used for the force/moments calculation. The output of this system if the new flight state of the missile which also the input for the controller to compute the next computing step.

Chapter 3

Flight Dynamics of a Missile

3.1 Introduction

To understand the physical system behaviour a missile a flight dynamic model have to create. In general is to say that, flight dynamics models are explaining the movement of a flight vehicle in the environment. As such, they are utilising the fundamental physical tools such as the newtons laws which allows them to describe the dynamic behaviour. These equations are highly depended from the aerodynamics, propulsion, gravitation and the control inputs of the system, which all are influencing the trajectory of the air vehicle. The goal of the section is to discuss at first all relevant areas for the flight dynamic modelling of the missile. Furthermore, this section will address general approaches and assumptions in the modeling of the equation system.

3.2 Systems of Axes

Systems of Axes are used to describe accelerations, velocities and positions in a geometric space. Since there are different areas of application for a coordinate system, they differ depending on the goal. In this part, some important coordinate systems for the aerospace sector are going to be introduced and discussed. Furthermore, tools are provided to transform states from one system to the other.

3.2.1 Earth Axes

The earth-fixed frame is used to describe the motion relative to the earth. So the three axes (x,y,z) are used to describe the aircraft position on earth. The earth is a spheric body which is rotating. That leads to the fact that the coordinate system should rotate with the planet together. However, that would lead to a more complex modulation of the physical problem which needs to be computed. For most of the use-cases, the assumption of a non-rotating flat earth is more than enough to get suitable results in aerospace application. Therefore this assumption is also used for this assignment.[3] The decleration of the axes of this coordiante system are:

- x-axis: x_E
- y-axis: y_E
- z-axis: : z_E

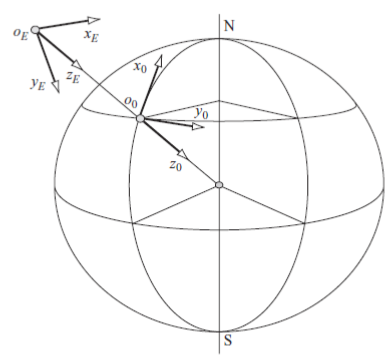

Figure 3.1: The earth-fixed coordinate system

3.2.2 Body-Axes

In the body-frame, the position, velocities and accelerations are described relative to the rigid body of the missile. The origin of this axes-system is the centre of gravity. The x-axis is going in the direction of the fuselage nose, and the z-axis is pointing down. This coordinate-system is mostly used to compute the flight dynamic states, which later on are transformed to the earth-fixed axis-system.[3] The decleration of the axes of this coordiante system are:

- x-axis: x_b
- y-axis: y_b
- z-axis: : z_b

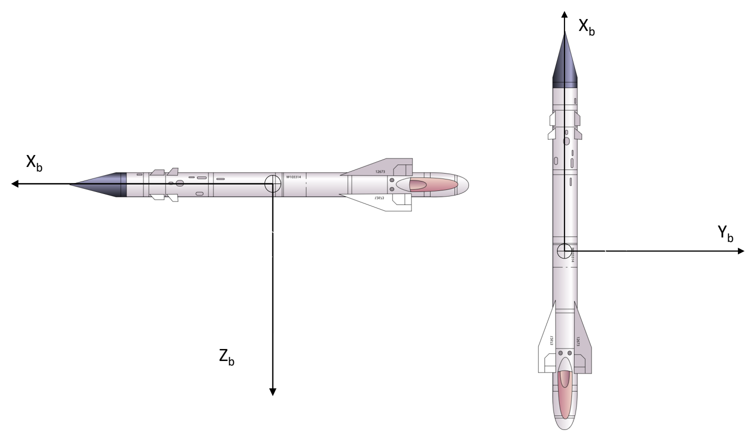

Figure 3.2: The body-fixed frame on a missile

3.2.3 Aerodynamic Axes or Stability Axes

The aerodynamic coordinate system is mainly used to describe aerodynamic outcomes such as the lift and drag. The direction of the aerodynamic coordinate system is determined by the course of the incoming airstream vector. The origin of the coordinate system is the centre of gravity of the missile. The x-axis is pointing parallel to the direction of the airstream. By using the assumption of a calm atmosphere, the airstream vector is the same direction and magnitude as the velocity vector of the aircraft.[3] The decleration of the axes of this coordiante system are:

- x-axis: x_a
- y-axis: y_a

- z-axis: : z_a

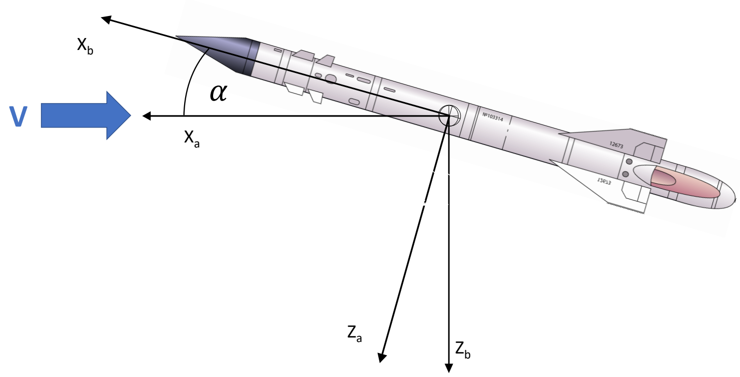

Figure 3.3: The aerodynamic frame of a missile

3.3 Euler Angles

The Euler angles are used to describe the angles the alignment of the aircraft relative to the earth-fixed system. The angles are representing the turns which are required to specify the angular orientation of the body-frame axes with respect to earth axes. [3]Therefore three angles are specified:

- Bank Angle: ϕ
- Pitch Angle: θ
- Yaw Angle: : ψ

3.4 Axes Transformation

Since we have various numbers of coordinate systems which are describing physical states and variables, it is necessary to explain those parameters in other frames. To do that it is needed to transform the states from one coordinate system into the other and vice-versa.

3.4.1 Directional Cosine Matrix (DCM)

The attitude of a body can be described as a sequence of rotations about a given axis-system. Since we know that, it is possible to transform from one coordinate system to another by rotating it around its axes until it is aligned with the other coordinate system. During this procedure, the order of the sequence is essential since different orders of rotating are delivering different results (commutative law is not applicable). The rotation around the axis can be described as matrices and can be multiplicated to a single matrix that allows the direct transformation from one coordinate system to the other. This matrix is called the Direction-Cosine-Matrix (DCM).[3; 9]

- Matrix for roll rotation:

$$M1(\Phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\Phi & -sin\Phi \\ 0 & -sin\Phi & cos\Phi \end{pmatrix} \tag{3.1}$$

- Matrix for pitching rotation:

$$M2(\Theta) = \begin{pmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{pmatrix} \tag{3.2}$$

- Matrix for yawing rotation:

$$M3(\Psi) = \begin{pmatrix} cos\Psi & sin\Psi & 0 \\ -sin\Psi & cos\Psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.3}$$

The DCM matrix is provided by multiplicating the matrices in the rotation order.

$$Mfg = M3(\Psi) \cdot M3(\Psi) \cdot M3(\Psi) \tag{3.4}$$

$$Mfg = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\Phi & -sin\Phi \\ 0 & -sin\Phi & cos\Phi \end{pmatrix} \cdot \begin{pmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{pmatrix} \cdot \begin{pmatrix} cos\Psi & sin\Psi & 0 \\ -sin\Psi & cos\Psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.5}$$

$$Mfg = \begin{pmatrix} cos\Phi cos\Psi & cos\theta sin\Psi & -sin\theta \\ sin\Phi sin\theta cos\Psi - cos\Phi sin\Psi & sin\Phi sin\theta sin\Psi + cos\Phi cos\Psi & sin\Phi cos\theta \\ cos\Phi sin\theta cos + sin\Phi sin\Psi & cos\Phi sin\theta sin\Psi - sin\Phi cos\Psi & cos\Phi \sin\theta \end{pmatrix} \tag{3.6}$$

3.4.2 Angular Velocities Transformation

To transform the angular rates of the missile (p,q,r) from the body frame to the earth-frame system the transformation can be described as a sequence of rotation around different axis.

1. first roll rate p:
$$p = \dot{\phi} - \dot{\psi} \sin\theta \tag{3.7}$$

2. pitch rate q:
$$q = \dot{\theta}\cos\phi + \dot{\psi}\sin\phi\cos\theta \tag{3.8}$$

3. yaw rate r:
$$r = \dot{\psi}\cos\phi\cos\theta - \dot{\psi}\sin\phi \tag{3.9}$$

The equations can be placed into the resulting matrix:

$$\begin{bmatrix} \dot{p} \\ q \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{3.10}$$

We need to inverse the matrix because the angular rates in respect to the earth axes are the required states which leads to:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.11}$$

3.5 Equation of Motion

To describe the motion of an aircraft or missile it is needed to utilise newtons second law. The law can be used for rotational as well as for translatory movements.[7]The first step for describing the motion is:

$$\sum \overrightarrow{F} = m\dot{\vec{v}} = \frac{d(m\vec{v})}{dt} \tag{3.12}$$

$$\sum \overrightarrow{M} = \frac{d}{dt}(H) \tag{3.13}$$

That is leading to the following components in and around the three axis:

$$\begin{matrix} F_X = \frac{d(mu)}{dt} & F_Y = \frac{d(mv)}{dt} & F_Z = \frac{d(mw)}{dt} \\ L = \frac{d}{dt}H_X & M = \frac{d(m\vec{v})}{dt}H_Y & N = \frac{d(m\vec{v})}{dt}H_Z \end{matrix} \tag{3.14}$$

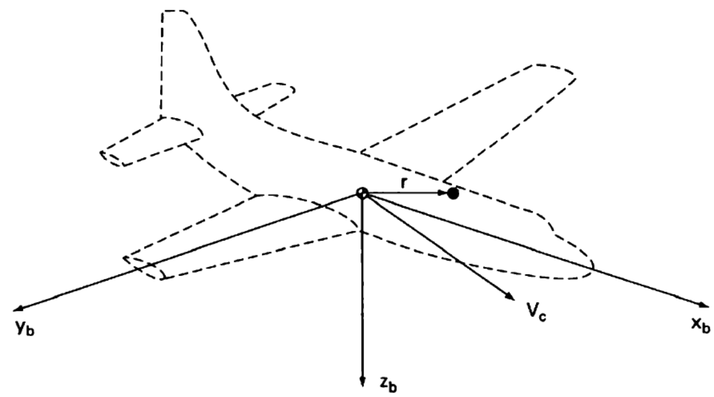

Figure 3.4: An element of mass on an airplane[7]

The forces in the body-frame of a small element of mass on the air vehicle can be described as:

$$\delta F = \delta m \frac{d(v)}{dt} \tag{3.15}$$

$$\delta F = F \tag{3.16}$$

The velocity of the mass element is:

$$v_P = v_c + \frac{dr}{dt} \tag{3.17}$$

$$\sum \delta F = \frac{d}{dt}(v_c + \frac{dr}{dt})\delta m \tag{3.18}$$

With the assumption that the mass of an air vehivle is constant we get:

$$F = m\frac{d}{dt}v_c + \frac{d}{dt}\sum \frac{dr}{dt}\delta m \tag{3.19}$$

$$F = m\frac{d}{dt}v_c + \frac{d^2}{dt^2}\sum r\,\delta m \tag{3.20}$$

The sum of $\sum r\,\delta m$ is equal to zero. With that the equation looks like:

$$F = m\frac{d}{dt}v_c \tag{3.21}$$

The same appraoches as for the translatory movement can be also used for the rotatory movement:

$$\delta M = \frac{d}{dt}\delta H = \frac{d}{dt}(r \times v)\delta m \qquad (3.22)$$

The velocity of the finite mass can be described as:

$$v_P = v_c + \frac{dr}{dt} = \omega \times r \qquad (3.23)$$

The total moment of momentum can be described as:

$$H = \delta H = \sum (r \times v_c)\delta m + \sum [r \times (\omega \times r)]\delta m \qquad (3.24)$$

Since the velocity vc is a constant, it can be taken outside:

$$H = \sum r\,\delta m \times v_c + \sum [r \times (\omega \times r)]\delta m \qquad (3.25)$$

As also in the translatory movement $\sum r\,\delta m$ is equal to zero. The angular rates can be described in vecor components:

$$\omega = pi + qj + rk \qquad (3.26)$$

$$r = xi + yj + zk \qquad (3.27)$$

If we include the velocities as vector components in the moments of momentum equation we get:

$$H = (pi + qj + rk)\sum (x^2 + y^2 + z^2)\,\delta m - \sum (xi + yj + zk)(pi + qj + rk)\delta m \qquad (3.28)$$

Described as scalar components in each axes we get:

$$H_X = p\sum (y^2 + z^2)\,\delta m - \sum xy\,\delta m - \sum r\,\delta m \qquad (3.29)$$

$$H_Y = -p\sum xy\,\delta m + \sum q(x^2+y^2)\delta m - r\sum yz\,\delta m \tag{3.30}$$

$$H_Z = -p\sum xz\,\delta m - q\sum yz\,\delta m + r\sum(x^2+y^2)\,\delta m \tag{3.31}$$

The last step is the summation of all finite elements, which is delivering the mass of inertia of the flight vehicle.

$$\begin{array}{ll} I_x = \int\int\int \quad (y^2+z^2)\,\delta m & I_{xy} = \int\int\int xy\,\delta m \\ I_y = \int\int\int \quad (x^2+z^2)\,\delta m & I_{xz} = \int\int\int xz\,\delta m \\ I_z = \int\int\int \quad (x^2+y^2)\,\delta m & I_{yz} = \int\int\int yz\,\delta m \end{array} \tag{3.32}$$

The equations which are describing the moments of momentum are:

$$H_x = pI_x - qI_{xy} - rI_{xz} \tag{3.33}$$

$$H_y = -pI_{xy} + qI_y - rI_{yz} \tag{3.34}$$

$$H_z = -pI_{xz} - qI_{xy} + rI_z \tag{3.35}$$

The equations which can be use to describe the translatory movement:

$$F_X = m(\dot{u} + qw - rv) \tag{3.36}$$

$$F_Y = m(\dot{v} + ru - pw) \tag{3.37}$$

$$F_Z = m(\dot{w} + pv - qu) \tag{3.38}$$

The equations which can be use to describe the rotatory movement:

$$L = \dot{H}_X + qH_Z - rH_Y = I_x\dot{p} - I_x\dot{r} + qr(I_z - I_y) - I_{xz}pq \tag{3.39}$$

$$M = \dot{H}_Y + rH_X - pH_Z = I_y\dot{q} + rp(I_x - I_z) + I_{xz}(p^2 - r^2) \tag{3.40}$$

$$N = \dot{H}_Z + pH_Y - qH_X = -I_{xz}\dot{p} + I_z\dot{r} + pq(I_y - I_x) + I_{xz}qr \tag{3.41}$$

3.6 Gravitational Forces

The gravity is acting on the missile at its centre of gravity. Because the body axis system got its origin in the centre of gravity, the forces due to gravitation do not produce any moments at the flight vehicle. But it is contributing to the external forces acting on the missile.[7; 3]

$$(F_X)_{gravity} = mg \sin\theta \tag{3.42}$$

$$(F_Y)_{gravity} = mg \cos\theta \sin\phi \tag{3.43}$$

$$(F_Z)_{gravity} = mg \cos\theta \cos\phi \tag{3.44}$$

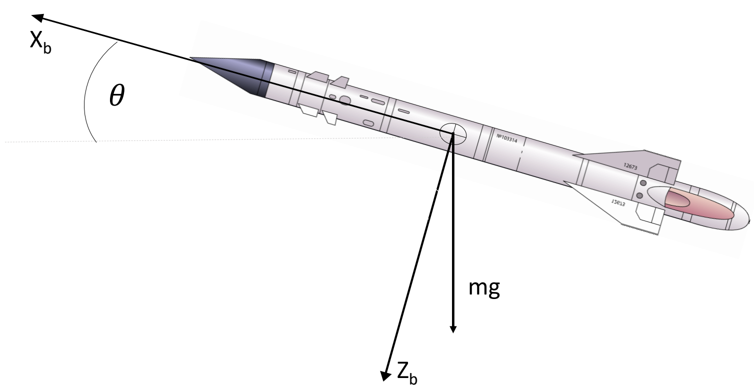

Figure 3.5: The gravitational force attacking on the missile

3.7 Mass of Intertia of a Missile

The mass of inertia is generally used to describe the mass distribution to calculate the rotational behaviour of a missile. Since it is possible to rotate a given body around three axes it is described in a 3x3 matrix.

$$\underline{\underline{I}} = \begin{pmatrix} I_x & I_{xy} & I_{xz} \\ I_{yx} & I_y & I_x \\ X_{zx} & I_{yz} & I_z \end{pmatrix} \tag{3.45}$$

Due to the fact that the missile got symmetries in y and z directions, we are recieving a diagonal matrix.

$$\underline{\underline{I}} = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix} \tag{3.46}$$

3.8 Thrust Forces

The thrust on a general flight vehicle is creating a force and a moment vector attacking the rigid body. In the given use-case the thrust vector is perfectly aligned with the x-axis. That means that in this use-case, the power is only affecting the longitudinal direction. Due to that we also do not get any moments around the centre of gravity.

$$(F_X)_{thrust} = X_T = T \tag{3.47}$$

$$(F_Y)_{thrust} = Y_T = 0 \tag{3.48}$$

$$(F_Z)_{thrust} = Z_T = 0 \tag{3.49}$$

$$(L)_{thrust} = L_T = 0 \tag{3.50}$$

$$(M)_{thrust} = M_T = 0 \tag{3.51}$$

$$(N)_{thrust} = N_T = 0 \tag{3.52}$$

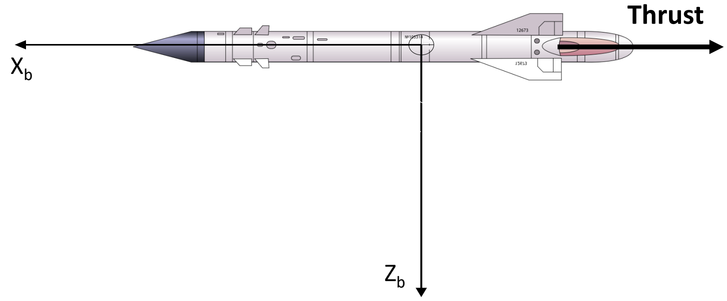

Figure 3.6: A thrust vector attacking on a misiles

3.9 Aerodynamics

Whenever a missile is disturbed from its trimming point, the aerodynamic balance is changing. To describe the aerodynamic behaviour due to aerodynamical disturbances fitting equations need to be defined. The first step is to assume that the aerodynamic force and moment behaviour are dependent on the disturbed motion variables. Mathematically this is conveniently represented as a function including the Taylor series of different variables, each set connecting one motion variable or derivative of a motion variable.[3] The described approach is presented by the example of the taylor series for the lift coefficient. Which would deviate in following way:

$$\begin{gathered} C_L = f(\alpha, \eta, q, \dot{\alpha}, ...) \\ C_L = C_{L_0} + (\frac{\delta X}{\delta \alpha}\alpha + \frac{\delta^2 X}{\delta \alpha^2}\frac{\alpha^2}{2!} + \frac{\delta^3 X}{\delta \alpha^3}\frac{\alpha^3}{3!} + ...) \\ +(\frac{\delta X}{\delta \eta}\eta + \frac{\delta^2 X}{\delta \eta^2}\frac{\eta^2}{2!} + \frac{\delta^3 X}{\delta \eta^3}\frac{\eta^3}{3!} + ...) \\ +(\frac{\delta X}{\delta \eta}q + \frac{\delta^2 X}{\delta q^2}\frac{q^2}{2!} + \frac{\delta^3 X}{\delta q^3}\frac{q^3}{3!} + ...) \\ +(\frac{\delta X}{\delta \dot{\alpha}}\dot{\alpha} + \frac{\delta^2 X}{\delta \dot{\alpha}^2}\frac{\dot{\alpha}^2}{2!} + \frac{\delta^3 X}{\delta \dot{\alpha}^3}\frac{\dot{\alpha}^3}{3!} + ...) \\ +(...) \end{gathered} \tag{3.53}$$

Because we are only focusing on disturbances, the magnitudes of the variables are small. That allows us to only consider the first term in each of the series functions since the higher order terms are negligibly small. The only important higher order derivative terms generally encountered are those including $\dot{w}$.

$$C_L = C_{L_0} + \frac{\delta X}{\delta \alpha}\alpha + \frac{\delta X}{\delta \eta}\eta + \frac{\delta X}{\delta \dot{\alpha}}\dot{\alpha} + ... \tag{3.54}$$

Using the common used coeffiecent derivates style:

$$C_L = C_{L_0} + C_{L_\alpha}\alpha + C_{L_\eta}\eta + C_{L_{\dot{\alpha}}}\dot{\alpha} + ... \tag{3.55}$$

3.9.1 Influences of Mach Number

The Mach number is an aerodynamical variable which allows to make conclusions about the compressibility behaviour of an airflow. The Mach number is a variable which is highly dependent from the gas type and its conditions. Basic equations to describe the Mach number are described in the following :

$$M = \frac{v}{a} \tag{3.56}$$

$$a = \sqrt{\kappa R T} = \sqrt{\kappa \frac{p}{\rho}} \tag{3.57}$$

$$\kappa_{Air} = 1,4$$

$$R_{Air} = 278.058 \frac{J}{kgK}$$

Mach Number on Lift

The lift of a missile is, of course, depends on the angle of attack which means that with an increasing alpha, the lift rises. But a similar coherence is happening due to the increase in Mach number. That means that with a rising Mach number the lift is also increasing. On the other hand, this effects reduces the maximum Lift the wing can attain and also decreases the maximum angle of attack. That means, that a higher Mach number will cause a stall at a higher speed and a lower angle of attack.[5]

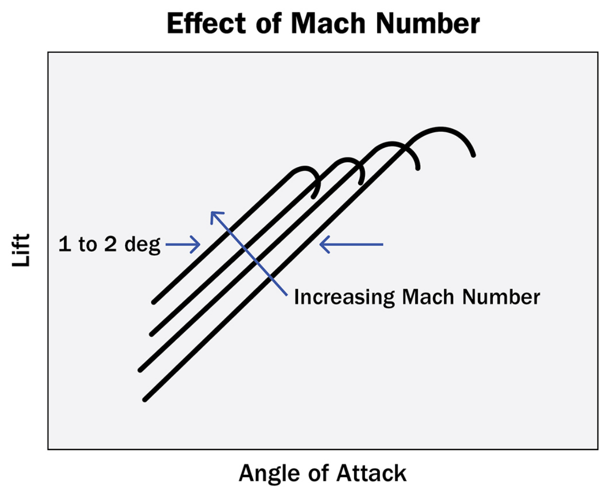

Figure 3.7: The influences of the Angle of Attack and Mach on the Lift

Mach Number on Drag

The change of the aerodynamic drag due to Mach number is given in figure in 3.8. At the point where the Mach Number is equal one is called the sound barrier. Around that locality, the drag shows is an immense change in its behaviour. At first, it is rising tremendously till it reaches the sound barrier. After reaching it, the drag drops in supersonic flow. Mcr is the critical Mach number which is the Mach number in which the airflow reaches the sound of speed at some areas over the wing. The critical Mach number does not influence an raise in drag. MDiv is the divergence Mach number where the drag coefficient starts to increase dramatically. Normally, the divergence Mach number located over the critical Mach number. In a general transition frsom subsonic flight to transonic or supersonic fight, this dramatic increase in drag needs to be considered in the design and operation of a flight vehicle.[1]

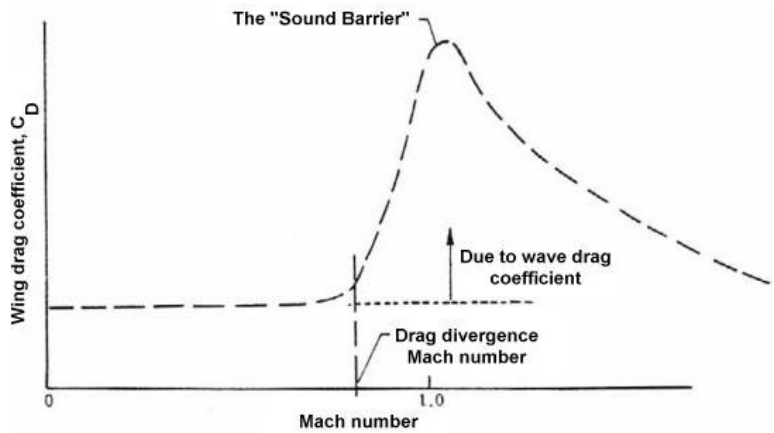

Figure 3.8: Drag development due to Mach increase

3.9.2 Transformation from Body to Aerodynamic Axes

Since the aerodynamic forces are described in the aerodynamic frame, all inputs and values which are coming from another axes-system need to be transformed to the aerodynamic axes-system. The transformation for the angular velocities is described as::

$$\begin{bmatrix} p^a \\ q^a \\ r^a \end{bmatrix} = C_b^a \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.58}$$

The transformation for the controll inputs is given as:

$$\begin{bmatrix} \delta_R^a \\ \delta_P^a \\ \delta_Y^a \end{bmatrix} = C_b^a \begin{bmatrix} \delta_R \\ \delta_P \\ \delta_Y \end{bmatrix} \tag{3.59}$$

$$C_a^b = [C_b^a]^T$$

3.9.3 Aerodynamic Equation for the Missile

The equations to compute the aerodynamic coefficients of the missile is given as:

$$C_X = C_{X_0}(M) + C_{X_{\alpha_T}}(M)\alpha_T + C_{X_{\delta_{eff}}}(M)\delta_{eff}^2 \tag{3.60}$$

$$C_Y = C_{Y_{\alpha_T}}(\alpha_T, M)\sin 4\phi_T + C_{Y_{\delta_Y}}(M)\delta_Y \tag{3.61}$$

$$C_Z = C_{Z_0}C_{Y_{\alpha_T}}(\alpha_T, M) + C_{X_0}C_{Y_{\alpha_T}}(\alpha_T, M)\sin^2\phi_T + C_{Z_{\delta_p}}(M)\delta_p \tag{3.62}$$

$$C_L = C_{L_{\alpha_T}}(M)\alpha_T^2 \sin 4\phi_T + C_{L_p}(M)(\frac{D}{2V})p^a + C_{L_{\delta_R}}(M)\delta_R \tag{3.63}$$

$$C_M = C_{M_0}(\alpha_T, M) + C_{M_{\alpha_T}}(\alpha_T, M)\sin^2 4\phi_T + C_{M_q}(\frac{D}{2V})q + C_{M_{\delta_p}}(M)\delta_p - C_Z(\frac{x_{cg} - x_{ref}}{D}) \tag{3.64}$$

$$C_N = C_{N_{\phi_T}}(\alpha_T, M)\sin 4\phi_T + C_{N_0}(M)(\frac{D}{2V})r + C_{N_{\delta_Y}}(M)\delta_Y + C_Y(\frac{x_{cg} - x_{ref}}{D}) \tag{3.65}$$

The equations are showing that aerodynamic coefficients are mostly dependent on the AoA and the Mach number. Since also other variables are involved in the movement generation, the Taylor series development delivered also the other derivates in the equation.

Chapter 4

Model Building in Simulink

4.1 Introdcution

The following Section will present and discuss the final model building of the missile system in MATLAB/Simulink. For the model building, the theoretical knowledge and the equations of the previous chapters have been utilised to establish a mathematical simulation model. Figure 4.1 shows the high-level assembly of the model. The model is organised like the presented sketch in chapter 2. In the following sections the organisation of the three subsystems: autopilot, actuators and flight dynamics model are going to be exhibited.

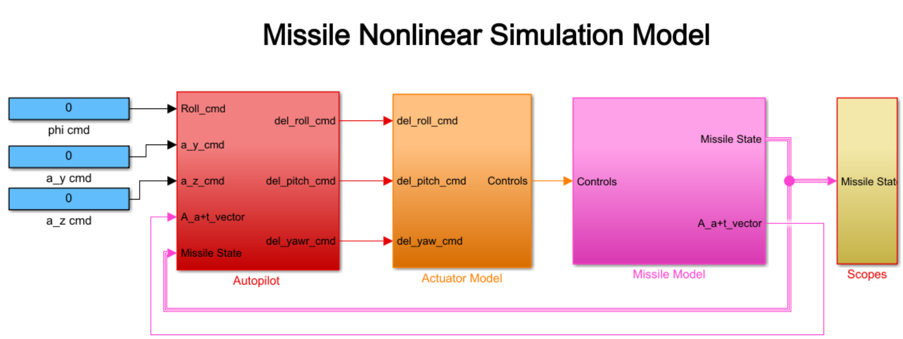

Figure 4.1: System Architecture of the general Simulink model

4.2 Autopilot - Subsystem

The Autopilot is the instance which is interacting as a controller. It receives the needed values from the output of the flight dynamic model and utilises them for a conceivable correction of the state. To guide the missile in a sufficient way a controller controls all three rotational movements.

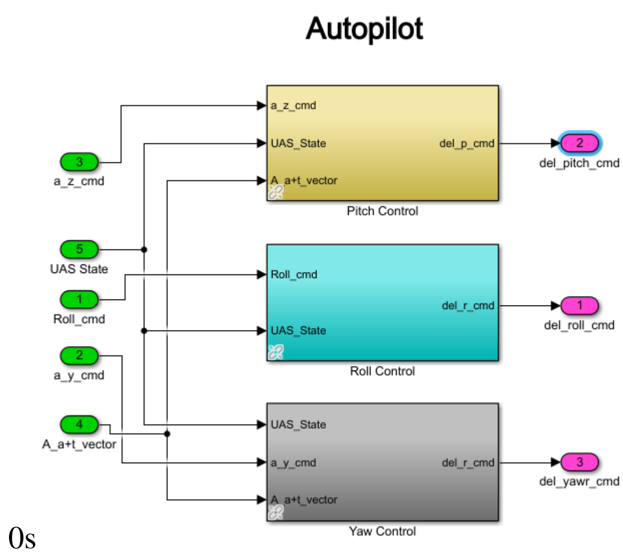

Figure 4.2: The Moduls of the Autopilot Sub-System

4.2.1 Roll Controller

The roll controller is used to control the bank angle of the missile. For that, it is comparing the reference roll command with the roll angle phi and the roll rate p. The controller is designed as a controller which is only utilising proportional block for the control.

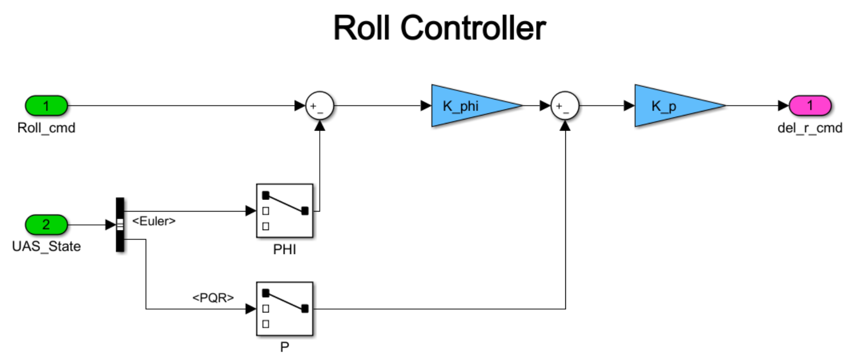

Figure 4.3: The implemented Roll Controller Block

4.2.2 Pitch Controller

The pitch controller is designed and implemented to control the pitch angle of the missile. To do that, the controller compares the reference acceleration in the z-direction with the actual acceleration in the z-direction. This controller is designed as a controller which is utilising proportional and one integrational block for the control.

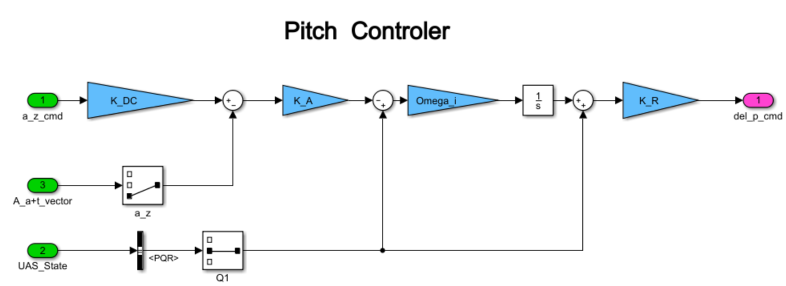

Figure 4.4: The implemented Pitch Controller Block

4.2.3 Yaw Controller

The yaw controller is used to control the yaw angle of the missile. To achieve that goal, the controller is comparing the reference acceleration in the y-direction with the actual acceleration of the missile in the y-direction. As like all the pitch controllers, this controller is designed as a controller which is utilising proportional and an integrational block for the control.

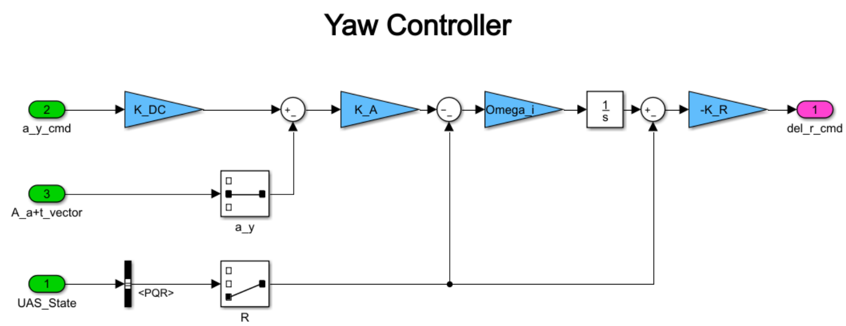

Figure 4.5: The implemented Yaw Controller Block

4.3 Actuator - Subsystem

The actuators are the motors which are moving the control surfaces so they can create moments around the centre of gravity of the missile system. The small engines were not described in more detail, but the transfer function was delivered in the task description..

$$\frac{\delta}{\delta_c} = \frac{\omega_{act}^2}{s^2 + 2\zeta\omega_{act}s + \omega_{act}^2} \tag{4.1}$$

The given transfer function allows it to create a transfer block which can create a suitable output in dependencies to the input. Important is to see that it seems that all control surfaces are using actuators from the type due to the fact that all commands are using the same transfer function.

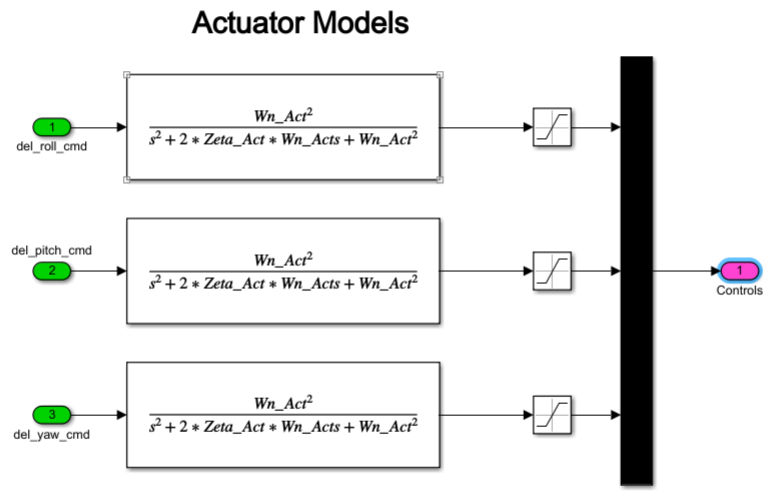

Figure 4.6: The Actuator Sub-System and its moduls

4.4 Missile Dynamic Model - Subsystem

4.4.1 Envioment

During the operations, the flight vehicle is moving through its environment. Due to altitude changes of the missile, the environmental variables such as temperature, density and speed of sound are shifting. Since this variables are influencing the flight dynamics of the missile, the changes have to be taken to account. The International Organization for Standardization (ISO) standardises this behaviour in the International Standard Atmosphere (ISA). The idea of ISO was to provide a standard which allows engineers to compare different approaches and give everyone the same mathematical environment.

The standard atmosphere distinguishes between isotherm layers, in which the temperature is not changing, and layers in which the temperature differs gradually dependent on to the altitude. For both layer types, different equations are provided, which will not be introduced here. Figure 4.7 shows the temperature development related to the elevation.

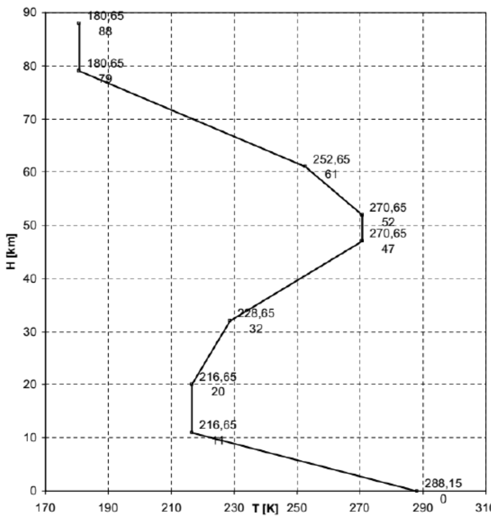

Figure 4.7: The ISA temperature development due to altitude changes

The computing of the density is made by the ISA model block which is provided by the Aerospace Toolbox of Simulink. For the computing of the Speed of sound (SoS), a Look-Up Table is in use. The Look-Up Table is utilising the provided SOS-Table from the university in which the SOS is given as a function of the hight.

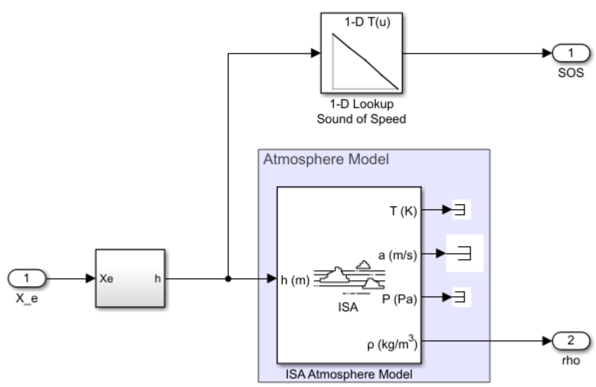

Figure 4.8: The used modules to create the demanded Enviroment

4.4.2 Equation of Motion

As described in chapter 3 the equations of motion are using newtons second law to describe the motion of the missile. The subsystem is using these equations and try to emulate them in the block system. For that, it takes the moments and forces which are attacking the missile and process them to deliver the new position, velocity and alignment of the flight vehicle.

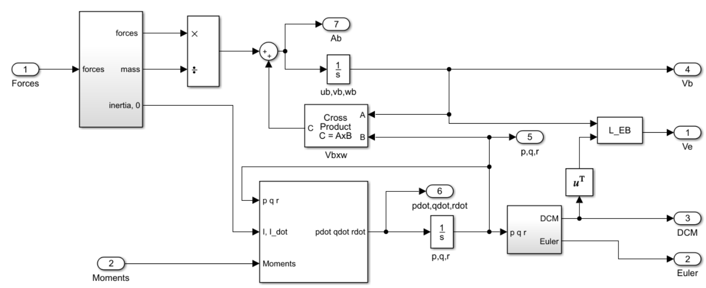

Figure 4.9: The architecture of the Equation of Motion Sub-System

4.4.3 Forces

This subsystem is used to centralise all relevant values for the computing of the total forces and moments of the missile system. It receives the aerodynamic coefficients, the dynamic pressure and the thrust command as an input. The output of the block is the summation of all forces and moment attacking the missile.

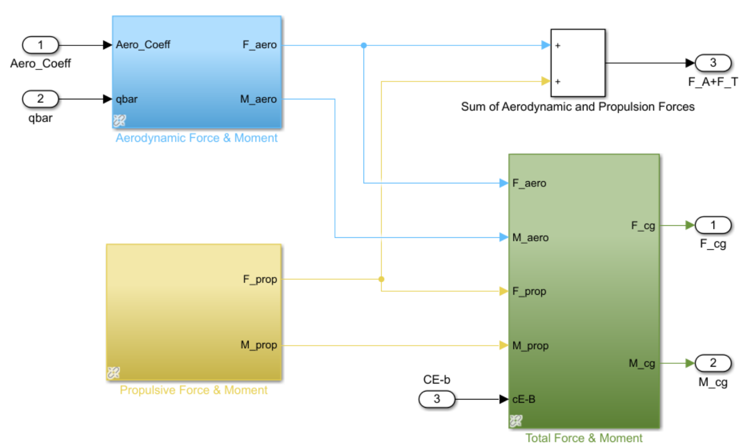

Figure 4.10: The components of the External Forces Sub-System

4.4.4 Aerodynamics

The aerodynamic subsystem is used to compute the needed aerodynamic coefficients which are required to determine the aerodynamic forces attacking the flight vehicle. To calculate the demanded values it needs to receive parameters such as the control inputs, Mach number, the total angle of attack, the total roll angle and the efficient control inputs.

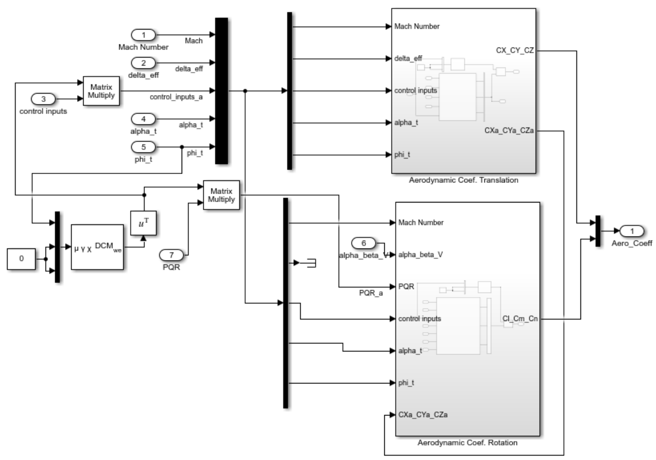

Figure 4.11: The Aerodynamic Coefficent Sub-System

Aerodynamic Coefficients for Translational Movement

The translational aerodynamic block is used to compute the translatory aerodynamic coefficients which are required to determine the aerodynamic forces which are attacking on the flight vehicle. To calculate the coefficients in x-, y- and z-axis the block needs following parameter: control inputs, Mach number, the total angle of attack, the total roll angle and the efficient control inputs.

Aerodynamic Coefficients in X-Direction

The Subsystem calculates the coefficient by emulating the following function:

$$C_X = C_{X_0}(M) + C_{X_{\alpha_T}}(M)\alpha_T + C_{X_{\delta_{eff}}}(M)\delta_{eff}^2$$

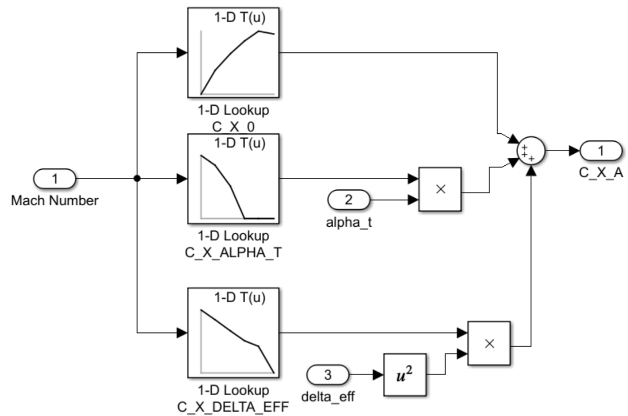

Figure 4.12: The Sub-System to compute the Coefficient C_X

Aerodynamic Coefficientsfor in Y-Direction

The Subsystem calculates the coefficient by emulating the following function:

$$C_Y = C_{Y_{\alpha_T}}(\alpha_T, M)\sin 4\phi_T + C_{Y_{\delta_Y}}(M)\delta_Y$$

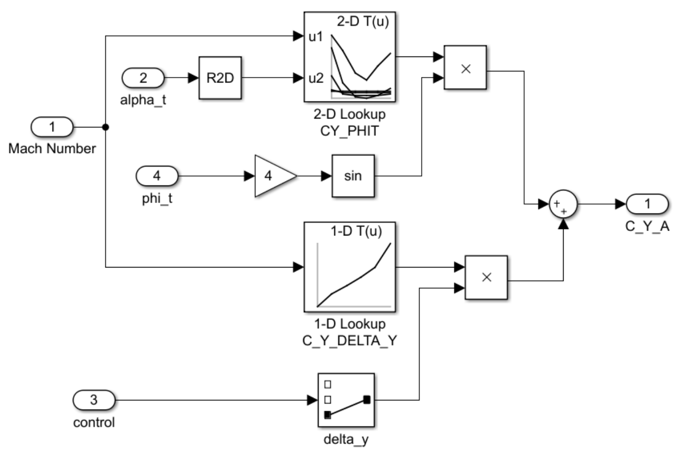

Figure 4.13: The Sub-System to compute the Coefficient C_Y

Aerodynamic Coefficients in Z-Direction

The Subsystem calculates the coefficient by emulating the following function:

$$C_Z = C_{Z_0}C_{Y_{\alpha_T}}(\alpha_T, M) + C_{X_0}C_{Y_{\alpha_T}}(\alpha_T, M)\sin^2\phi_T + C_{Z_{\delta_p}}(M)\delta_p$$

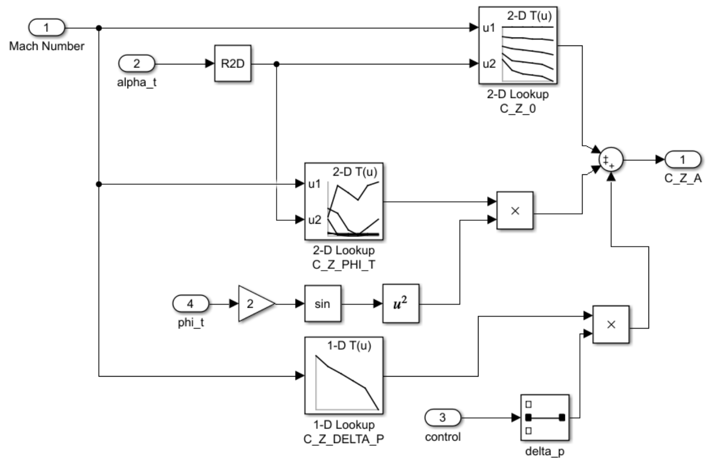

Figure 4.14: The Sub-System to compute the Coefficient C_Z

Aerodynamic Coefficients for Rotational Movement

The rotatory aerodynamic block is used to compute the translatory aerodynamic coefficients which are required to determine the aerodynamic forces attacking on the flight vehicle. To calculate the coefficients the block demands following parameter: control inputs, Mach number, the total angle of attack, the total roll angle and the efficient control inputs.

Aerodynamic Coefficients for Roll Moment

The Subsystem calculates the coefficient by emulating the following function:

$$C_L = C_{L_{\alpha_T}}(M)\alpha_T^2 \sin 4\phi_T + C_{L_p}(M)(\frac{D}{2V})p^a + C_{L_{\delta_R}}(M)\delta_R$$

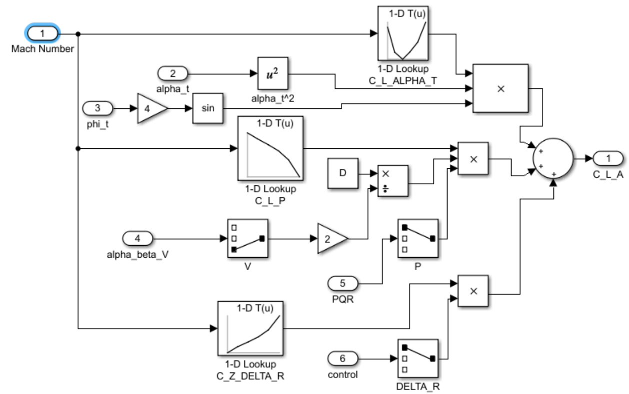

Figure 4.15: The Sub-System to compute the Coefficient C_L

Aerodynamic Coefficients for Pitch Moment

The Subsystem calculates the coefficient by emulating the following function:

$$C_M = C_{M_0}(\alpha_T, M) + C_{M_{\alpha_T}}(\alpha_T, M)\sin^2 4\phi_T + C_{M_q}(\frac{D}{2V})q + C_{M_{\delta_p}}(M)\delta_p - C_Z(\frac{x_{cg} - x_{ref}}{D})$$

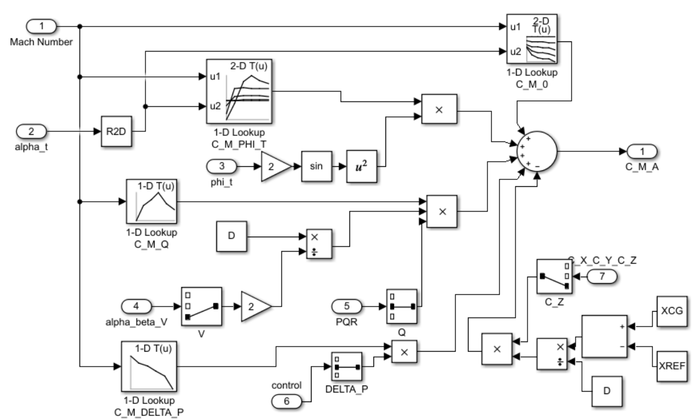

Figure 4.16: The Sub-System to compute the Coefficient C_M

Aerodynamic Coefficients for Yaw Moment

The Subsystem calculates the coefficient by emulating the following function:

$$C_N = C_{N_{\phi_T}}(\alpha_T, M)\sin 4\phi_T + C_{N_0}(M)(\frac{D}{2V})r + C_{N_{\delta_Y}}(M)\delta_Y + C_Y(\frac{x_{cg} - x_{ref}}{D})$$

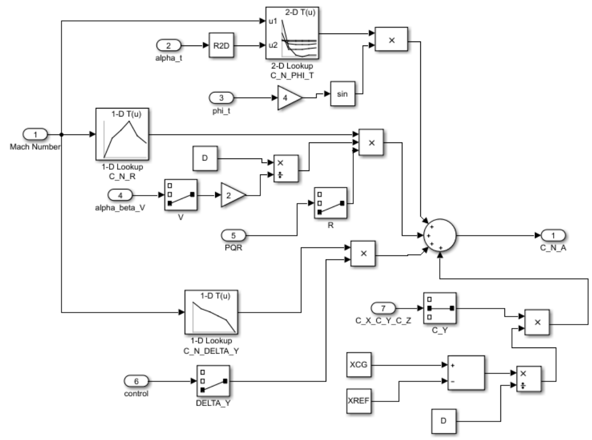

Figure 4.17: The Sub-System to compute the Coefficient C_N

4.5 Effective Control Input - Subsystem

The parameter of the effective control is computed in a separate subsystem shown in 4.18. The block is imitating the following equation:

$$\delta_{eff} = \frac{\delta_R + \delta_P}{2} \tag{4.2}$$

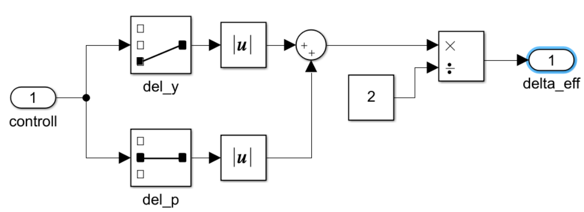

Figure 4.18: The Sub-System to compute the parameters δ_{eff}

4.6 Total Angle of Attack and Aerodynamic Roll

The total angle of attack and the aerodynamic roll are calculated in the block shown in 4.19. The block is imitating the following equation:

$$\alpha_T = \cos^{-1}(\cos\alpha\cos\beta) \tag{4.3}$$

$$\phi_T = \tan^{-1}\left(\frac{\tan\beta}{\tan\alpha}\right) \tag{4.4}$$

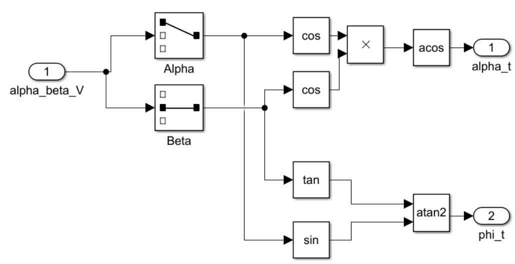

Figure 4.19: The Sub-System to compute the parameters α_T and t ϕ_T

Chapter 5

Model Verfication

In in the discipline of model and simulation, verification is the process of proving that the model is precisely implemented compared to the mathematical representation.
To do that tests of the model are made to find errors and bugs inside of the simulation paradigm. The objective of model verification, therefore, is to ensure that the implementation of the model is correct. There are many approaches to verify a given model. A very popular approach is to compare the model with some test cases in which the solutions will be compared with a required one. Another approach is to analyse the response and dynamic behaviour and compare it with the expected dynamic behaviour of a general flight vehicle. This section uses both approaches to verify the established model correctly.

5.1 Equation Verfication of Aerodynamic Subsystem

The aerodynamic subsystem has been verified by using the given equations. For each sub-block, the equations have been solved for significant values and compared to hand-calculated outcomes. If they both had the same results, the block has been declared as correct.

5.1.1 Equation Verification of Unconnected Calculation for Coefficient in X-Direction

The implemented function is given as:

$$C_X = C_{X_0}(M) + C_{X_{\alpha_T}}(M)\alpha_T + C_{X_{\delta_{eff}}}(M)\delta_{eff}^2$$

Verfication-Case 1

- Input: $M = 0.2$; $\alpha_T = 0°$; $\delta_{eff} = 0°$
- Expcted Value: $C_X = C_{X_0}(0) + C_{X_{\alpha_T}}(0) \times 0 + C_{X_{\delta_{eff}}}(0) \times 0 = -0.3840 + 0.0788 \times 0 - 16.8080 \times 0 = -0.3840$
- Output of the Block: $C_X = -0.3840$

Verfication-Case 2

- Input: $M = 0.2$; $\alpha_T = 4°$; $\delta_{eff} = 4°$
- Expcted Value: $C_X = C_{X_0}(0.2) + C_{X_{\alpha_T}}(0.2) \times 4° \times \frac{\pi}{180} + C_{X_{\delta_{eff}}}(0.2) \times 0 = -0.3840 + 0.0788 \times 4° \times \frac{\pi}{180} - 16.8080 \times 4° \times \frac{\pi}{180} =$
- Output of the Block: $C_X =$

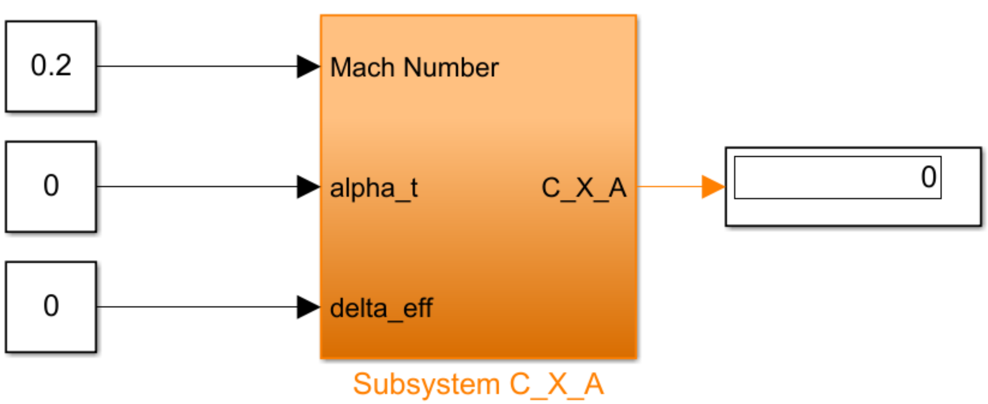

Figure 5.1: The Test-Architecture of the C_X Sub-System

5.1.2 Equation Verification of Unconnected Calculation for Coefficient in Y-Direction

The implemented function is given as:

$$C_Y = C_{Y_{\alpha_T}}(\alpha_T, M)\sin 4\phi_T + C_{Y_{\delta_Y}}(M)\delta_Y$$

Verfication-Case 1

- Input: $M = 0.2$; $\alpha_T = 0$; $\phi_T = 0°$
- Expcted Value: $C_Y = C_{Y_{\alpha_T}}(0, 0.2) \times \sin(4 \times 0) + C_{Y_{\delta_Y}}(0.2) \times 0 = 0 + 5.3056 \times 0 = 0$
- Output of the Block: $C_Y = 0$

Verfication-Case 2

- Input: $M = 0.2$; $\alpha_T = 4°$; $\phi_T = 0°$
- Expcted Value: $C_Y = C_{Y_{\alpha_T}}(0, 0.2) \times \sin(4 \times 4° \times \frac{\pi}{180}) + C_{Y_{\delta_Y}}(0.2) \times 0 = +5.3056 \times 0 = 0.001929$
- Output of the Block: $C_Y = 0.001929$

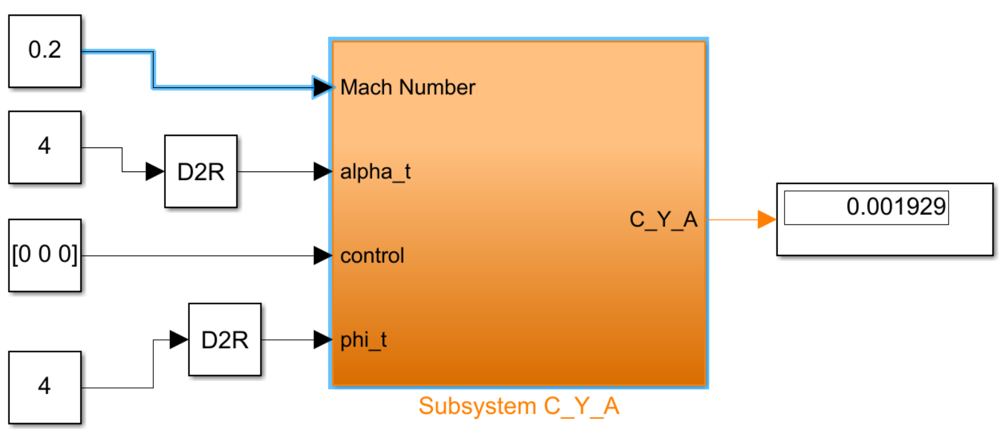

Figure 5.2: The Test-Architecture of the C_Y Sub-System

5.1.3 Equation Verification of Unconnected Calculation for Coefficient in Z-Direction

The implemented function is given as:

$$C_Z = C_{Z_0}(\alpha_T, M) + C_{X_0}(\alpha_T, M)\sin^2\phi_T + C_{Z_{\delta_p}}(M)\delta_p$$

Verfication-Case 1

- Input: $M = 0.2$; $\alpha_T = 0°$;$\phi_T = 0°$; $\delta_p = 0°$
- Expcted Value: $C_Z = C_{Z_0}(\alpha_T, M) + C_{X_0}(\alpha_T, M)\sin^2\phi_T + C_{Z_{\delta_p}}(M)\delta_p = 0+0+0 = 0$
- Output of the Block: $C_Z = 0$

Verfication-Case 2

- Input: $M = 0.2$; $\alpha_T = 4°$; $\delta_{eff} = 0°$
- Expcted Value: $C_Z = C_{Z_0}(4°, 0.2) + C_{X_0}(4°, 0.2)\sin^2(4° \times \frac{\pi}{180}) + C_{Z_{\delta_p}}(0.2) \times 0 = +5.3056 \times 0 = -2.0680 + 0.0089 \times)\sin^2(4° \times \frac{\pi}{180}) - 5.3056 \times 0 = -2.0678$
- Output of the Block: $C_Z = -2.068$

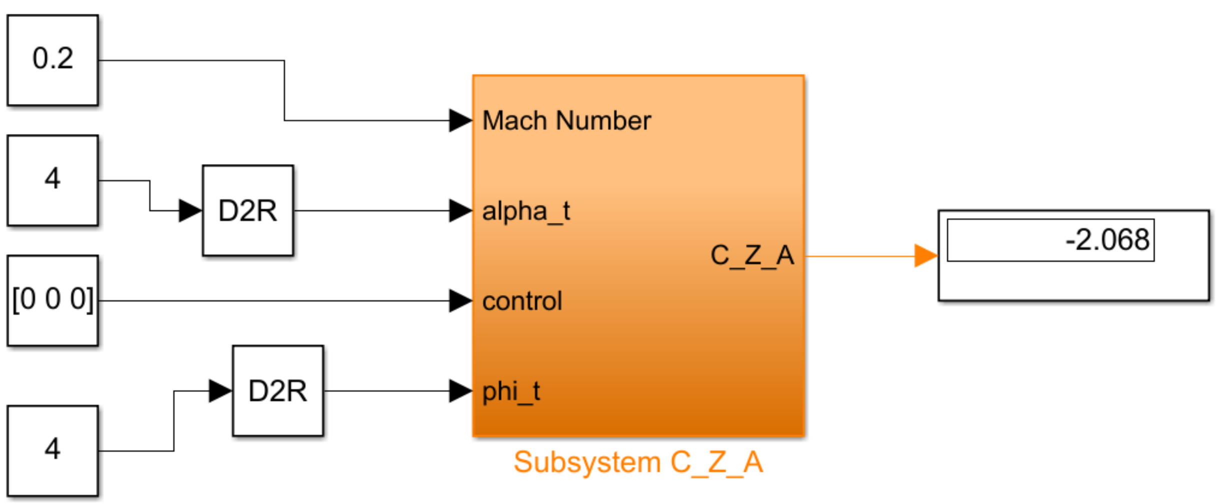

Figure 5.3: Sketches for the differnt Vertical Wing Positions

5.1.4 Equation Verification of Unconnected for Coefficient around X-Axes

The implemented function is given as:

$$C_L = C_{L_{\alpha_T}}(M)\alpha_T^2 \sin 4\phi_T + C_{L_p}(M)(\frac{D}{2V})p^a + C_{L_{\delta_R}}(M)\delta_R$$

Verfication-Case 1

- Input: $M = 0.2$; $\alpha_T = 0°$;$\phi_T = 0°$; $\delta_p = 0°$; $\begin{pmatrix} \alpha \\ \beta \\ V \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$; $\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$
- Expcted Value: $C_L = C_{L_{\alpha_T}}(0.2)\alpha_T^2 \sin 4\times 0 + C_{L_p}(0.2)(\frac{D}{2V})\times 0 + C_{L_{\delta_R}}(0.2)\times 0 = 0$
- Output of the Block: $C_L = 0$

Verfication-Case 2

- Input: $M = 0.2$; $\alpha_T = 4°$;$\phi_T = 4°$; $\delta_p = 0°$; $\begin{pmatrix} \alpha \\ \beta \\ V \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$; $\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$
- Expcted Value: $C_L = C_{L_{\alpha_T}}(0.2) \times \alpha_T^2 \sin(4 \times 4° \times \frac{\pi}{180}) + C_{L_p}(0.2)(\frac{0.15m}{2V}) \times 0 + C_{L_{\delta_R}}(0.2)\times 0 = -0.0988\times(0.0698)^2\times 0.275 + 0 + 0 = -0.001327$
- Output of the Block: $C_L = -0.001327$

5.1.5 Equation Verification of Unconnected Calculation for Coefficient around Y-Axes

The implemented function is given as:

$$C_M = C_{M_0}(\alpha_T, M) + C_{M_{\alpha_T}}(\alpha_T, M)\sin^2 2\phi + C_{M_q}(\frac{D}{2V})q + C_{M_{\delta_p}}(M)\delta_p - C_Z(\frac{x_{cg}-x_{ref}}{D})$$

Verfication-Case 1

- Input: $M = 0.2$; $\alpha_T = 0°$;$\phi_T = 0°$; $\delta_p = 0°$; $\begin{pmatrix} \alpha \\ \beta \\ V \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$; $\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$;
  $C_Z = 0$
- Expcted Value: $C_M = C_{M_0}(0, 0.2) + C_{M_{\alpha_T}}(0, 0.2)\sin^2(2 \times 0°) + C_{M_q}(\frac{0.15}{2\times 1}) \times 0 + C_{M_{\delta_p}}(0.2) \times 0 - 0 \times (\frac{0}{0.15}) = 0$
- Output of the Block: $C_M = 0$

Verfication-Case 2

- Input: $M = 0.2$; $\alpha_T = 4°$;$\phi_T = 4°$; $\delta_p = 0°$; $\begin{pmatrix} \alpha \\ \beta \\ V \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$; $\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$;
  $D = 0.15m$;
- Expcted Value: $C_M = C_{M_0}(4°, 0.2) + C_{M_{\alpha_T}}(4°, 0.2)\sin^2(2 \times 4° \times \frac{\pi}{180}) + C_{M_q}(\frac{0.15}{2\times 1}) \times 0 + C_{M_{\delta_p}}(0.2) \times 0 - 0 \times (\frac{0}{0.15}) = -2.2250 - 0.0024 \times \sin(2 \times 4° \times \frac{\pi}{180}) + 0 - 0 = -2.225$
- Output of the Block: $C_M = -2.225$

5.1.6 Equation Verification of Unconnected Calculation for Coefficient around Z-Axes

The implemented function is given as:

$$C_N = C_{N_{\phi_T}}(\alpha_T, M)\sin 4\phi_T + C_{N_0}(M)(\frac{D}{2V})r + C_{N_{\delta_Y}}(M)\delta_Y + C_Y(\frac{x_{cg}-x_{ref}}{D})$$

Verfication-Case 1

- Input: $M = 0.2$; $\alpha_T = 0°$;$\phi_T = 0°$; $\delta_p = 0°$; $\begin{pmatrix} \alpha \\ \beta \\ V \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$; $\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$; $C_Y = 0$
- Expcted Value: $C_N = C_{N_{\phi_T}}(0°, 0.2)\sin 4 \times 0° + C_{N_0}(0.2)(\frac{0.15}{2\times 1}) \times 0 + C_{N_{\delta_Y}}(0.2) \times 0 + 0 \times (\frac{0}{0.15}) = 0$
- Output of the Block: $C_N = 0$

Verfication-Case 2

- Input: $M = 0.2$; $\alpha_T = 4°$;$\phi_T = 4°$; $\delta_p = 0°$; $\begin{pmatrix} \alpha \\ \beta \\ V \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$; $\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$; $D = 0.15m$
- Expcted Value: $C_N = C_{N_{\phi_T}}(4°, 0.2)\sin(4 \times 4° \times \frac{\pi}{180}) + C_{N_0}(0.2) \times (\frac{0.15}{2\times 1})r + C_{N_{\delta_Y}}(0.2) \times 0 + C_Y \times (\frac{0}{0.15}) = 0.0040 \times \sin(2° \times 4° \times \frac{\pi}{180}) - 82.9643 \times (\frac{0.15}{2\times 1}) \times 0 = 0.001103$
- Output of the Block: $C_N = 0.001103$

5.1.7 Equation Verfication of overall Effective Control Input Subssystem

The implemented function is given as:

$$\delta_{eff} = \frac{\delta_R + \delta_P}{2}$$

Verfication-Case 1

- Input: $\delta_R = 0$; $\delta_P = 0$
- Expcted Value: $\delta_{eff} = \frac{\delta_R + \delta_P}{2} = \frac{0+0}{2} = 0$
- Output of the Block: $\delta_{eff} = 0$

Verfication-Case 2

- Input: $\delta_R = 1$; $\delta_P = 1$
- Expcted Value: $\delta_{eff} = \frac{\delta_R + \delta_P}{2} = \frac{1+1}{2} = 1$
- Output of the Block: $\delta_{eff} = 1$

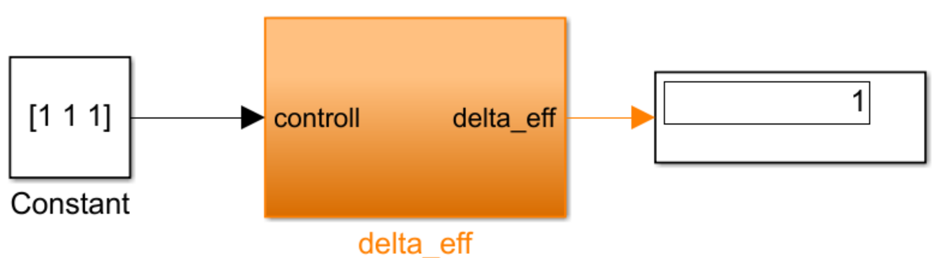

Figure 5.4: The Test-Architecture of the δ_{eff}Sub-System

5.1.8 Equation Verfication of overall Total Angle of Attack and Aerodynamic Roll Subssystem

The implemented function is given as:

$$\alpha_T = \cos^{-1}(\cos\alpha\cos\beta)$$

$$\phi_T = \tan^{-1}(\tfrac{\tan\beta}{\tan\alpha})$$

Verfication-Case 1

- Input: $\alpha = 0^\circ$; $\beta = 0^\circ$
- Expcted Value:
  - $\alpha_T = \cos^{-1}(\cos 0^\circ \cos 0^\circ) = 0$ and $\phi_T = \tan^{-1}(\frac{\tan 0^\circ}{\tan 0^\circ}) = 0$
- Output of the Block:
  - $\alpha_T = 0$ and $\phi_T = 0$

Verfication-Case 2

- Input: $\alpha = 45^\circ$; $\beta = 45^\circ$
- Expcted Value:
  - $\alpha_T = \cos^{-1}(\cos 45^\circ \cos 45^\circ) = 60$ and $\phi_T = \tan^{-1}(\frac{\tan 45^\circ}{\tan 45^\circ}) = 54.74$
- Output of the Block:
  - $\alpha_T = 60^\circ$ and $\phi_T = 54.74^\circ$

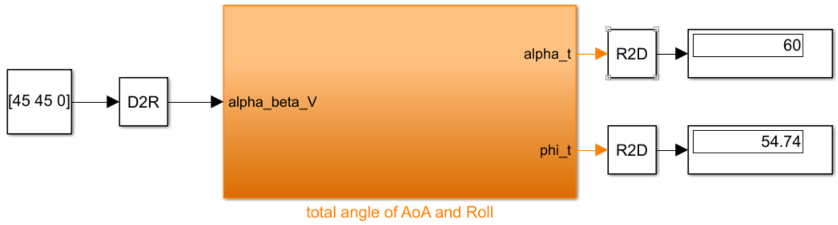

Figure 5.5: The Test-Architecture of the α_T and t ϕ_T Sub-System

5.2 Model behaviour due to Thrust increase

The thrust is a force which is attacking on the missile. As long as the thrust force is high enough the speed of a missile should increase. This Verification step uses that knowledge to verify the model behaviour. The Thrust, as a system input, was set manually to 3000 Newton. The chosen force magnitude is quite high, which allows seeing an increase over a more extended period until the drag will occur gradually. The Diagram got the graph development as it should have, which shows that the model is correct.

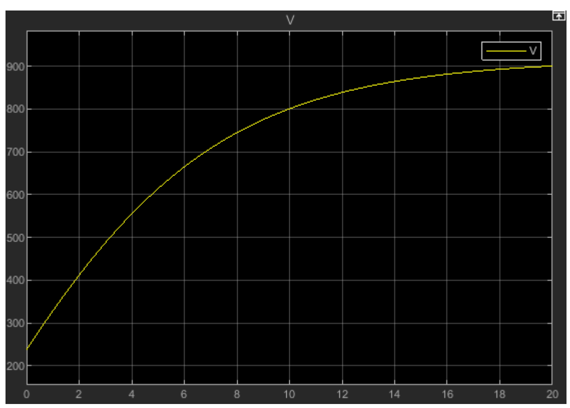

Figure 5.6: The development of the Speed under the influence of a constant Thrust

5.3 Model behaviour due to roll command input

The rolling behaviour of the aircraft can be validated by evaluating the model behaviour and explain by the delivered equations and values. The equation of the aerodynamic roll coefficient is:

$$C_L = C_{L_{\alpha_T}}(M)\alpha_T^2 \sin 4\phi_T + C_{L_p}(M)(\frac{D}{2V})p^a + C_{L_{\delta_R}}(M)\delta_R$$

The interesting components,in terms of a validation, are the damping influences C_{L_p} and the command input influences with $C_{L_{\delta_R}}$. The second addend of the equation is the damping of the function, which means that this part is working agains the roll motion of the missile. When we look at the given values of the aerodynamic coefficient C_{L_p} we can see that this value is relatively very high. Due to that, the behaviour leads to a very high damping when a roll motion is induced to the missile. The third addend is showing the influences of the roll command to the aerodynamic roll coefficient. This summand rises when the command input is given. However, the given coefficients for C_{L_p} are dramatically smaller than the ones for the damping of the function. That means that the slope of the function during an input would be not that steep as when the missile tries to stop the motion after the input went back to zero. Figures 5.7 shows the function roll rate with a time discrete roll command input (1° for 0.1 seconds). As expected is the slope of the function during the rising phase, not that steep as after the control input went back to zero. That shows that the damping is much stronger than the influences depending on the command input. So that shows the expected behaviour.

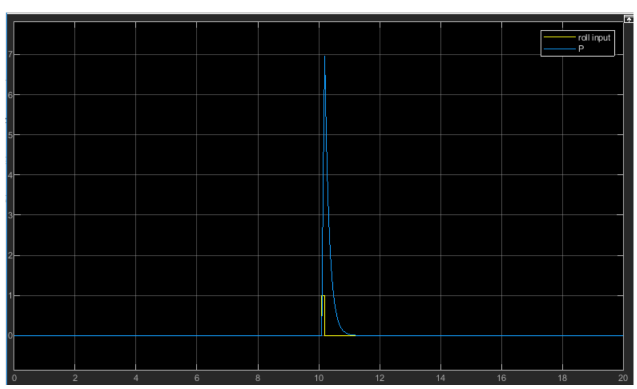

Figure 5.7: The roll response due to a roll command impulse input

5.4 Model behaviour due to pitch command input

Figure 5.8 shows the model behaviour in the longitudinal direction. This can be used to verify the pitching behaviour of the missile. The diagram shows the reaction of the missile due to a time discrete pitch command input (1°for 0.1 sec). The general Simulation time is set to 20 seconds. At the beginning of the simulation, the system induced some oscillation due to the fact that the missile has an initial speed of 238 m/s, but set all attitude angles to zero. This fact leads to a longitudinal oscillation called phugoid mode. After this mode damped out the missile finds its trimming angle of attack and keep that angle as a constant, also called equilibrium. This behaviour shows that the missile owns a dynamic longitudinal stability and since the static stability is a necessary condition for the dynamic stability, the missile also possesses that. After 10 seconds a command input of 1°for 0.1 seconds gets induced to the system. It can be observed that the first reaction of the system is a negative angle development due to the command. That is also a evidence of the static longitudinal stability. The response of the system after the command went back to zero is a damped oscillation around the equilibrium which decreases in amplitude until the missile goes back to a non-oscillation behaviour in its equilibrium angle.

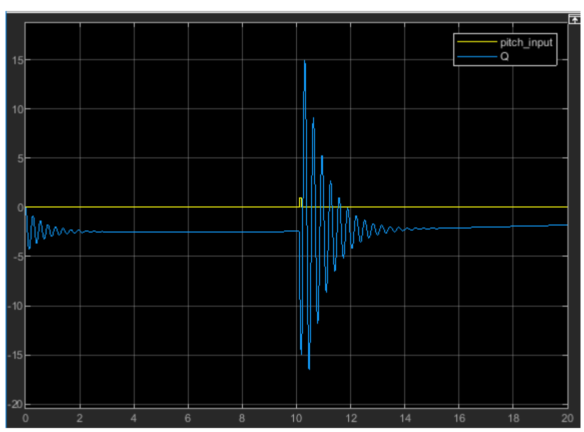

Figure 5.8: The pitch angle overtime and the response due to a pitch command input

5.5 Model behaviour due to yaw command input

Figure 5.9 shows the model behaviour in the vertical direction. This can be used to verify the yawing behaviour of the missile. A yaw command input impulse of 1° was induced to the system right at the beginning of the simulation. After that, the system ossilated without any external input parameters. It can be observed that the behaviour of the yawing rate is increasing at the beginning of the simulation since an input impulse is given to the system. During the free oscillation, the missile is showing a damped quivering behaviour which leads to a decrease of the magnitudes over the time. This behaviour proofs the directional stability of the missile system.

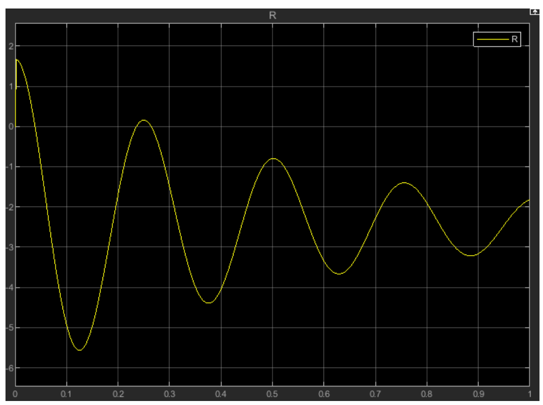

Figure 5.9: The yaw angle overtime due to a yaw command input

5.6 Dutch-Roll mode

Figure 5.10 shows the model behaviour in the lateral direction. The response behaviour can be used to verify the dutch-roll response of the missile. The diagram is showing the speed of the rocket in the body-frame y-axis direction. This speed behaviour is highly coupled to the roll behaviour since a roll angle of the system leads to a movement to the side direction due to the fact that the lift vector tilts around the x-axis. That means an induced roll angle always leads to a side force and in our example, the side speed becomes zero when the missile has no roll angle. To prove that the behaviour a horizontal speed and a roll angle was introduced to the projectile as initial values. The simulation time was set to 7 seconds.

It can be observed that with the begin of the simulation the missile starts to oscillating around its equilibrium which is the roll angle of zero. Since the oscillating is damped, the projectile reaches its trimming point after 5 seconds at a static roll angle of zero. This behaviour shows that the system owns lateral stability.

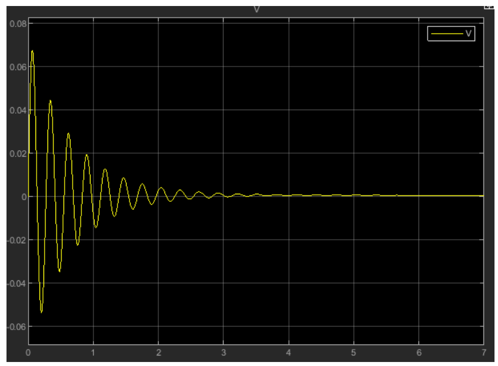

Figure 5.10: The ploting of the behaviour of the lateral speed v overtime

Chapter 6

Trimming of the Missile

6.1 Introduction

The trimming function is one of the most important functions a general flight vehicle can have. In piloted flight vehicle the trim function is essential to relieve the pilot from the duty to hold constant pressure on the sticks. Trimming means that out all forces and moments attacking on the vehicle are balanced which is leading to an steady-state flight. Besides the advantages for the pilot the trimming feature also provides benefits for the control system. Since a trimming point is defined, the provided functions can be simplified and linearized. That allows it to reduce the complexity of high non-linear systems in control and saves a lot of time, money and computing power. This chapter presents the theory behind the trimming approach and presents the Simulink model and its results.

6.2 Theory

6.2.1 Hartman- Grobman Theorem

To study dynamical systems is necessary to find ways of simplifying problems. Most dynamics systems are highly non-linear, and it is not permissible to solve these multidimensional problems analytically. Despite that complexity, it is sometimes necessary to gain a deep understanding of the behaviour of a specific dynamic system. In that case, the Hartman- Grobman Theorem delivers a tool that could help to simplify the non-linear models. The theorem says that the response of a dynamical system in an area close to a hyperbolic equilibrium spot is qualitatively the identical as the behaviour of its linearization near this equilibrium location. Hyperbolicity means that no eigenvalue of the linearization has a real part corresponding to zero. Accordingly, when dealing with highly complex systems, it is easier to utilise the more simplified linearisation of a system to analyse its behaviour around steady state points.[6; 2]

6.2.2 Equilibrium

For a flight vehicle in any random initial state and preset control settings, the body alignment and the flight track alternating about an equilibrium point. If the flight vehicle is unstable, the velocity and the attitude of the vehicle deviate from the equilibrium. If the vehicle is stable, it will converge with the equilibrium point and transit to a steady state flight. In general is an aircraft trimmed or in equilibrium flight when the velocity is constant, and the pitch and roll angles are steady. So the obstacle in the trim point search is to find the control settings that produce a stable flight situation.[9]

$$\dot{x_e} = f(x_e(t), u_e(t), \sigma_e(t)) = 0 \tag{6.1}$$

In order to find a state in which all derivatives of the equations are equal to zero, various approaches can be used. It is , for example, possible to compute the needed deviates in an iteration or to find some specific constraints equations which are approximating certain values during the computation. The second mentioned approach has shown to be very effective in the use of solving aerospace engineering problems.

Coordinate-Turn Constrain (CTC)

The Coordinate-Turn is a turn in which the missile perfectly takes a manoeuvre without any slipping or skipping angle. Such an turn should always be approached by a missile or aircraft and is therefore a constrains.

$$CTC = \sin\phi - G_{turm}\cos\beta(\sin\alpha\tan\theta + \cos\alpha\cos\phi) = 0 \tag{6.2}$$

$$G_{turn} = \frac{\dot{\psi_{trim}}V}{g} \tag{6.3}$$

$$\dot{\psi_{trim}} = 0 \tag{6.4}$$

Rate-of-Climb Constrain (ROC)

The Rate-of-Climb of the aircraft should be zero, which means that the aircraft should approach a steady level flight without any increase of the altitude.

$$ROC = \sin\gamma_{trim} - a\sin\theta + b\cos\theta = 0 \tag{6.5}$$

$$a = \sin\alpha\sin\beta \tag{6.6}$$

$$b = \sin\phi\sin\beta + \cos\phi\sin\alpha\cos\beta \tag{6.7}$$

Speed Constrain (ROC)

In a steady flight, the missile shall not have any changing of the velocity over the time and should, therefore, stay in the trimmed velocity. The speed constraint describes this constraint.

$$SC = V_{trim} - V = 0 \tag{6.8}$$

Pull-Up and Turn-Rate Constraints

This constraint is dealing with the pitch rate of the aircraft. In order to be classified as trimmed an aircraft should have a no pitch and yaw rates. This constrain is called the Pull-Up and Turn-Rate Constraints.

Pull-Up Constrain

$$\dot{\theta} = q\cos\phi - r\sin\phi - \dot{\theta}_{trim} = 0 \tag{6.9}$$

with:

$$\dot{\theta}_{trim} = 0 \tag{6.10}$$

Turn-Rate Constrain

$$\dot{\psi} = q\sin\phi\sec\theta + r\cos\phi\sec\theta - \dot{\psi}_{trim} \tag{6.11}$$

with:

$$\dot{\psi}_{trim} = 0 \tag{6.12}$$

6.3 Approach in Matlab/Simulink

In order to find a position in which the given state values are leading to a trimming in which the missile will remain in that place. The constraints have to be implemented into the Simulink model as a tool to determine that a particular point is in a trimming condition or not.

6.3.1 Coordinate-Turn Constrain

The used constrain has been implemented into the constraints subsystem. Figure 6.1 shows the way the equation has been emulated for computing the trimming. At the end of the assembly, an integrator block, with the initial condition zero is processing the input. The idea behind that is that the integrator will always give an output as long the input values differ from the initial condition in the block. That helps the system to recognise when the trimming point is achieved because than the output will be zero.

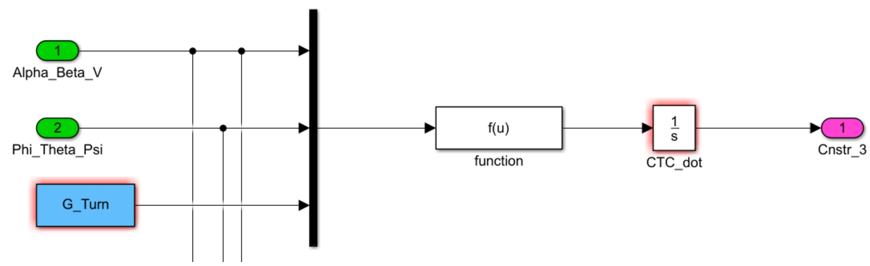

Figure 6.1: The implemented Coordinate-Turn Constrain

6.3.2 Rate-of-Climb Constrain

The presented constrain has been implemented into the constraints subsystem. Figure 6.2 shows the way the equation has been emulated for computing the trimming. Like for the other constraints an integrator is used to locate the point in which the equation becomes zero.

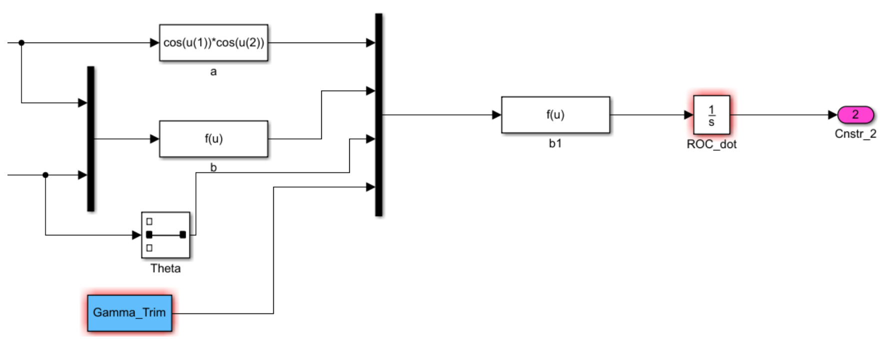

Figure 6.2: The implemented Rate-of-Climb Constrain

6.3.3 Speed Constrain

The in introduced constrain has been implemented into the constraints subsystem. Figure 6.3 shows the way the equation has been emulated for computing the trimming. Like for the other constraints an integrator is used to locate the point in which the equation results become zero.

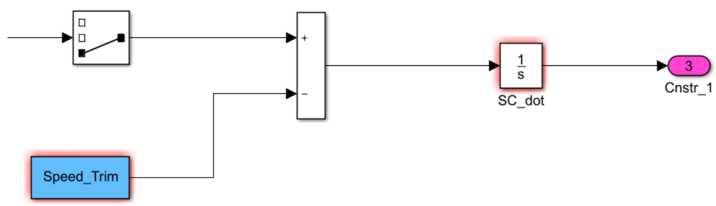

Figure 6.3: The implemented Speed Constrain

6.3.4 Pull-Up and Turn-Rate Constraints

The introduced constraints have been implemented directly into the equation of the motion subsystem. Figure 6.4 shows the way the equation has been emulated for computing the Euler angles and the DCM matrix. But unlike the other constraints, these two are not used as indicators to analyse if the trimming point has been achieved or not. The restrictions had been implemented into the time discrete fight dynamic calculation to achieve a convergence of the flight state with the searched trimming state.

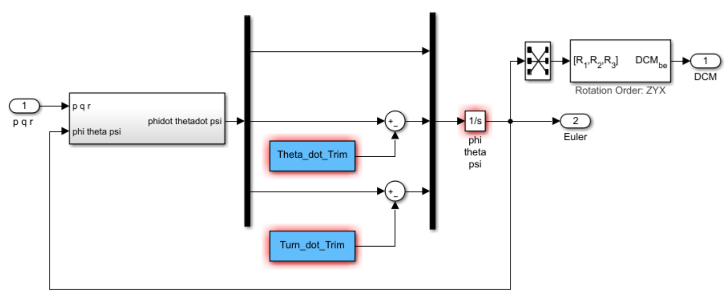

Figure 6.4: The implemented Pull-Up and Turn-Rate Constraints

6.3.5 Matlab Script

To calculate the trimming point for the given flight state values of Mach and Altitude a Matlab script has been created. The script is loading the missile and simulation data information and executes the Simulink model in the background. To find a trimming point for a given state the trim-function of the Matlab library has been utilised. The function uses a sequential quadratic programming algorithm to find the trim point for a given state.

Algorithmus 6.1 Structure and function of the trimming algorithm for the example of Mach=0.7 and Altitude=0

begin
$//Loading\ all\ needed\ data\ from\ Missile\ Data\ and\ SimvParameter\ files$
Input : $Missile\ Data$; $SimvParameter\ files$;
var $MachTrim := 0.7$,$AltTrim := 0m$
var $SpeedTrim := MachTrim \times SpeedOfSoundAsFunctionOfAltitude(AltTrim)$
While $achievedAllConstraints \quad == \quad False$ than **do**
  $achievedAllConstraints = sequatialQuadraticOptimaisation()$
end While
$//Print\ and\ Save\ the\ Results\ ofthe\ Trimming$
Output : $Trim\ State$; $Trim\ Command\ Inputs$;
end

That approach is commonly used for constrained nonlinear optimisation problems and the solving of queries in which a specific value of a function is needed. A generic idea of the programmed MATLAB script and its structure is given in 6.1.

6.4 Results

The Results of the Script are printed out in the MATLAB command window and also as saved in MATLAB files on the computer. The trimming points have been generated for three different cases. Which have following specification:

- For the first case, the trimming point for the following values have to be searched : $M = 0.7$ and $Altitude = 0$
- For the second case, the trimming points for the following values have to be searched : $M = [0.3, 0.5, 0.7, 0.9, 1.1]$ and $Altitude = 0$
- For the first case, the trimming point for the following values have to be searched : $M = 0.7$ and $Altitude = [0m, 1000m, 2000m, 3000m, 4000m]$

6.5 Results for Trimming State for constant Mach and constant Altitude

$$Altitude = 0m; Mach = 0.7$$

$$TrimState = \begin{pmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \\ \alpha \\ \beta \\ V \end{pmatrix} = \begin{pmatrix} 238.1547\,[m/s] \\ 0\,[m/s] \\ 3.9578\,[m/s] \\ 0\,[deg/s] \\ 0\,[deg/s] \\ 0\,[deg/s] \\ 0\,[deg/s] \\ 0.9521\,[deg] \\ 0\,[deg] \\ 0.9521\,[deg] \\ 0\,[deg] \\ -238.1876\,[m/s] \end{pmatrix};$$

$$u_{trim} = \begin{pmatrix} \delta_T \\ \delta_R \\ \delta_P \\ \delta_Y \end{pmatrix} = \begin{pmatrix} 216.0976\,[N] \\ 0\,[deg] \\ -1.4886\,[deg] \\ 0\,[deg] \end{pmatrix}$$

6.5.1 Verfication of the Trimm results for constant Mach and constant Altitude

The approach to verify the trimming solution was to use the trimming solutions for the module and plot the behaviour. The x_trim solution of the trimming was used for initialising the state of the missile and the u_trim values as the inputs for the roll, pitch, yaw and thrust commands. The goal of that approach was to observe a steady state behaviour of the missile over the whole simulation time. Figure 6.5 shows the setup of the verification.

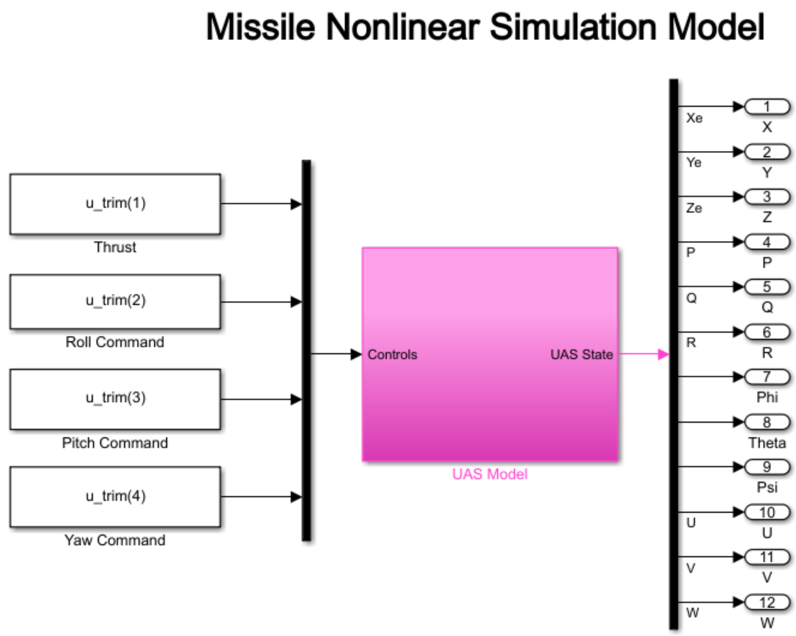

Figure 6.5: The Verfictation Architecture for the Trim-State of Mach=0.7 and h=0

The Verfication of the velocities in the Body-Frame

Figure 6.6 shows the velocities of the aircraft in body-frame axes. The expected behaviour was to see a steady state motion of the missile. The plot diagram shows exactly the demanded behaviour of the flight vehicle. All velocities are constant and do not change over time. That plot gives a good indicator that the aircraft remains in the trim state.

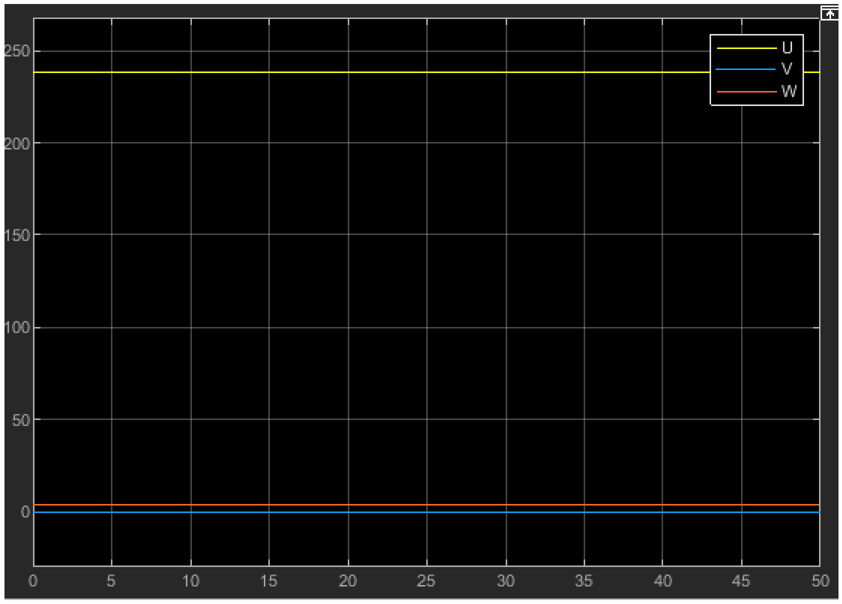

Figure 6.6: The plot of the missile velocities in the body-frame

The Verfication of the angular rates in the Body-Frame

Figure 6.7 shows the angular rates of the aircraft in body-frame axes. The expected behaviour is to see a steady state motion of the aircraft. That means that the rotation rates remain at zero. Despite the fact, the diagram implies that the values are not zero, actually they are. In the top right corner is shown that the displayed values are displayed with the scale '10^{-14}' that means that the values are very small and nearly zero. So it can be assumed that this behaviour also shows the demanded behaviour.

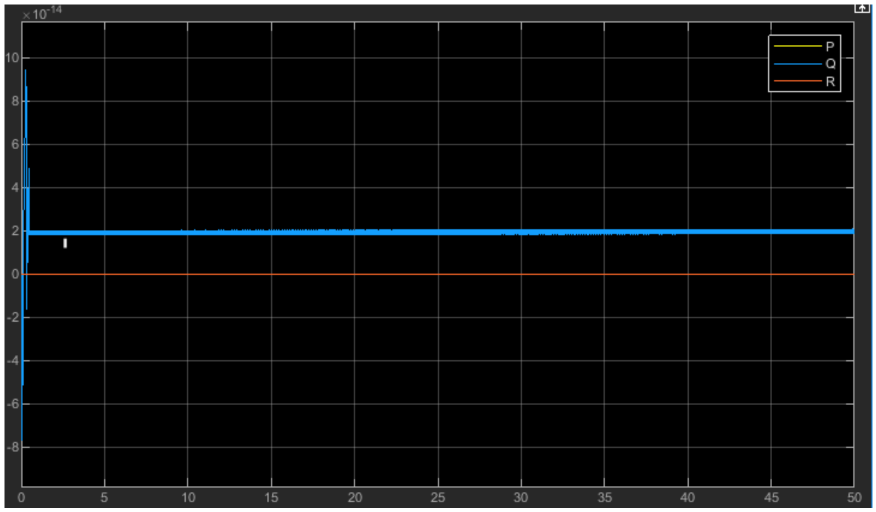

Figure 6.7: The plot of the angular rates of the missile in the body-frame

The Verfication of the Angle of Attack, Side Slip Angle and total Velocity

Figure 6.8 shows the angle of attack, the side slip angle and the net velocity of the aircraft. The expected behaviour is to see steady state angles and velocity. The Diagram shows that the system shows the expected behaviour perfectly and gives a proof that it remains in the trim position at all time.

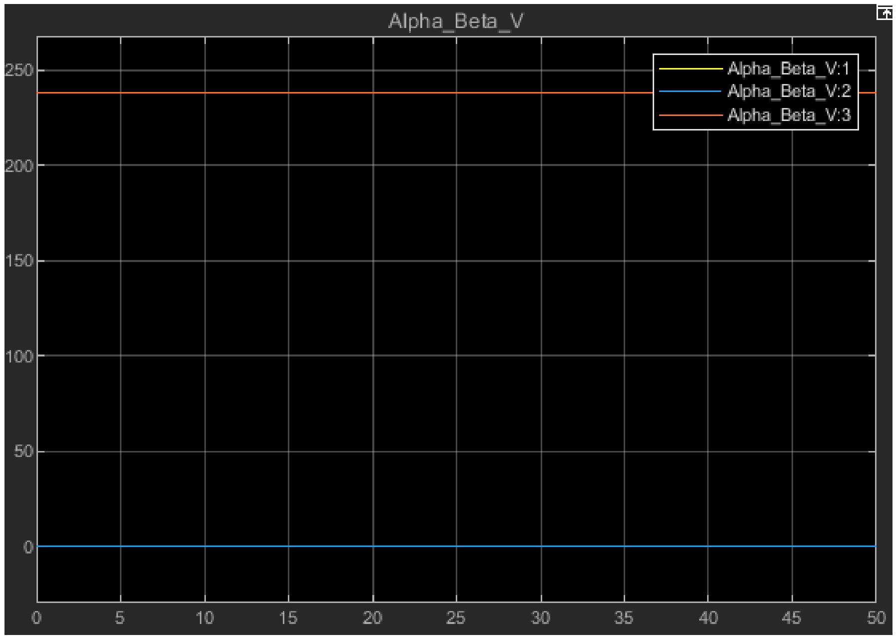

Figure 6.8: The plot of the Alpha, Beta and the net Velocity

The Verfication of the Movement in the Earth-Frame

Figure 3.1 shows the motion of the aircraft int he earths fixed system. Since we have a steady level flight, the system should remain at the same hight. Since we have a constraint that we do not want to have a side motion the system should not have a movement to the y-direction. The only action which should be happening is a move into the x-direction which should grow with a linear behaviour since we have a steady flight with a constant velocity. The expected behaviour is shown in the diagram, and therefore it can be clarified that the computed trim solution is a real trim solution of the model.

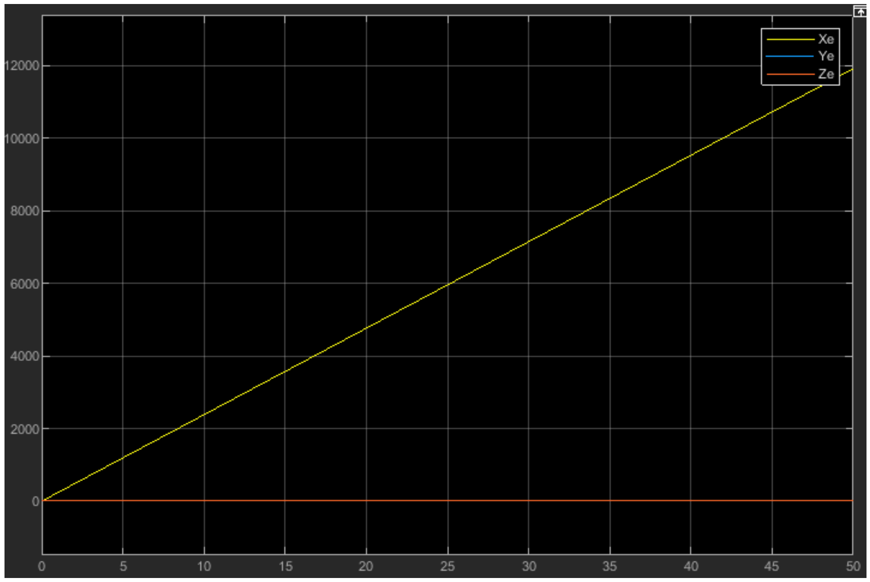

Figure 6.9: The plot of the missile movement in the earth-fixed coordiante system

6.6 Results for Trimming State for different sequences of Mach by having a constant Altitude

$$Altitude = 0m; Mach = [0.3, 0.5, 0.7, 0.9, 1.1]$$

State		Mach=0.3	Mach=0.5	Mach=0.7	Mach=0.9	Mach=1.1
u	[m/s]	101.4478	170.0086	238.1547	306.2257	374.2863
v	[m/s]	0	0	0	0	0
w	[m/s]	11.3467	6.5304	3.9578	3.0787	2.5190
p	[deg/s]	0	0	0	0	0
q	[deg/s]	0	0	0	0	0
r	[deg/s]	0	0	0	0	0
ϕ	[deg]	0	0	0	0	0
θ	[deg]	6.3819	2.1998	0.9521	0.5760	0.3856
ψ	[deg]	0	0	0	0.0197	-0.0024
α	[deg]	6.3819	2.1998	0.9521	0.5760	0.3856
β	[deg]	0	0	0	0	0
V	[m/s]	102.0804	170.1340	238.1876	306.2412	374.2948

Table 6.1: Table of the x_{trim} for a sequence of different Mach numbers at a constant height

Input-Vector		Mach=0.3	Mach=0.5	Mach=0.7	Mach=0.9	Mach=1.1
δ_T	[N]	87.9640	124.6131	216.0976	349.9694	519.4491
δ_R	[deg]	0	0	0	0	0
δ_P	[deg]	-9.8236	-3.4305	-1.4886	-0.9006	-0.6029
δ_Y	[deg]	0	0	0	0	0

Table 6.2: Table of the u_{trim} for a sequence of different Mach numbers at a constant height

6.6.1 Discusions of the results for different sequences of Mach by having a constant Altitude

The trimming solutions for a constant hight by varying the Mach number are going to be discussed in this section. The Mach number increased from the initial 0.3 to 1.1. Since the altitude is constant the ISA model delivers a constant speed of sound. That means that any increase in Mach number can only be related to an actual speed increase of the missile. That connection explains why the horizontal velocity u value in the body-frame and the net velocity V are increasing during the sequence of Mach numbers. Since we implemented the Rate-of-Climb constraints that says that rate of climb shall be zero. That can be only achived when γ_{trim} is zero. The connection between the angles γ,α and $\theta$is:

$$\gamma = \theta - \alpha = 0 \tag{6.13}$$

$$\theta = \alpha \tag{6.14}$$

That conncection leads to the fact that theta and alpha should share the same value in all trim values. Due to the fact of a steady flight, the Lift should not change during the flight. However, because of the case that the increasing speed is influencing the Lift the lift coefficient need to counter that tendency by reducing the lift coefficient.

$$Lift = \frac{\rho}{2} v^2 C_L S = constant \tag{6.15}$$

The reduces of the lift coefficient is achieved by decreasing the angle of attack of the missile. Since the pitch angle and the AoA are coupled the pitch angle also decreases. Over the time also the vertical body-frame velocity w decreased. That can be explained by the fact that the velocity is highly coupled with u and α:

$$w = \tan(\alpha) \times u \tag{6.16}$$

The values of the AoA and the horizontal speed are showing that the AoA decreased relatively in a higher ratio as the speed increased. That explains why the vertical speed increased even though the horizontal speed is increasing. When we leave the state variables and talk about the input parameters, we can also connect the behaviour to the created scenario. Since we increase the speed of the aircraft to reach higher Mach numbers an increasing thrust needs to be provided. Because no roll or yaw movement is allowed, the inputs for these are zero. The pitch command is decreasing over the sequence of increasing speed. That is because the angle of attack and also the pitch angle is decreasing, as already explained. Because of that, the pitch input needs to be reduced to adjust the changing pitch angles. Furthermore, it needs to be said that the results for the Mach Numbers higher as 0.7 are not precisely. Since the provided table stops at 0.7 which leads to the fact that only the dynamic pressure is changing. Because of that aerodynamic effects such as compression shock and wave drags are not considered.

6.7 Results for Trimming State for different sequences of Altitude by having a constant Mach

$$Altitude = [0m, 1000m, 2000m, 3000m, 4000m]; Mach = 0.7$$

State		Altitude= 0m	Altitude= 1000m	Altitude= 2000m	Altitude= 3000m	Altitude= 4000m
u	[m/s]	238.1547	235.4443	232.7006	229.9208	227.1031
v	[m/s]	0	0	0	0	0
w	[m/s]	3.9578	4.4112	4.929	5.522	6.2035
p	[deg/s]	0	0	0	0	0
q	[deg/s]	0	0	0	0	0
r	[deg/s]	0	0	0	0	0
ϕ	[deg]	0	0	0	0	0
θ	[deg]	0.9521	1.0734	1.2134	1.3758	1.5647
ψ	[deg]	0	0	0	0	0
α	[deg]	0.9521	1.0734	1.2134	1.3758	1.5647
β	[deg]	0	0	0	0	0
V	[m/s]	238.1876	235.4856	232.7528	229.9871	227.1878

Table 6.3: Table of the x_{trim} for a sequence of different Altitude numbers while the Mach number is constant

Input-Vector		Altitude= 0m	Altitude= 1000m	Altitude= 2000m	Altitude= 3000m	Altitude= 4000m
δ_T	[N]	216.0976	193.3617	172.9839	154.8275	138.7715
δ_R	[deg]	0	0	0	0	0
δ_P	[deg]	-1.4886	-1.6782	-1.8972	-2.151	-2.4463
δ_Y	[deg]	0	0	0	0	0

Table 6.4: Table of the x_{trim} for a sequence of different Altitude numbers while the Mach number is constant

6.7.1 Discusions of the results for different sequences of Altitude by having a constant Mach

The trimming solutions for a constant Mach number by varying the Altitude are going to will be discussed in this section. Since the Mach number is fixed but the height is changing the speed of the missile is also changing, regarding to the ISA model. The speed of sound is decreasing since the temperature over the altitude is decreasing. That leads to the fact that the horizontal speed (u) in the body-frame and the net speed V are decreasing, while the Mach number constant. Since the velocity is decreasing alpha needs to increase to enable the missile to balance the flight vehicle at the same height with a constant Lift. As already mentioned due tot he Rate-of-Climb constraint theta is behaving like the angle of attack. Because the vertical speed (w) is coupled to the angle of attack, it has to rise due to an increase of alpha. The Input for the thrust is decreasing because the speed of the aircraft needs to decrease to keep the Mach number constant. That leads to fact that the demanded thrust is less high. Since the pitch angle is increasing over the series the input for the pitch command has also to increase to provide enough angle of attack to hold the Lift constant.

Chapter 7

Linearisation of the Flight Dynamic Model

7.1 Introduction

The non-linear equations of motions is a highly complex system. It takes a lot of computational effort to solve them and analyse them. Since the computer performance of the most systems is restricted. A tool is needed which enables us to analyse the system and its behaviour. The tool required in our case is the linearization. The linearization simplifies the equation in a way which allows it to analyse the characteristics with even low-level computer systems. In the following that approach will be introduced and the results of the linearosation for the missile system presented.

7.2 Theory

As discussed in section 7.1 it is possible to use the linerised form of an highly non-linear problem to analyse the stability behavior of the system of interisting. The linearisation of the non-linear function is made around the defined equilibirum point by using a taylor series approximation.

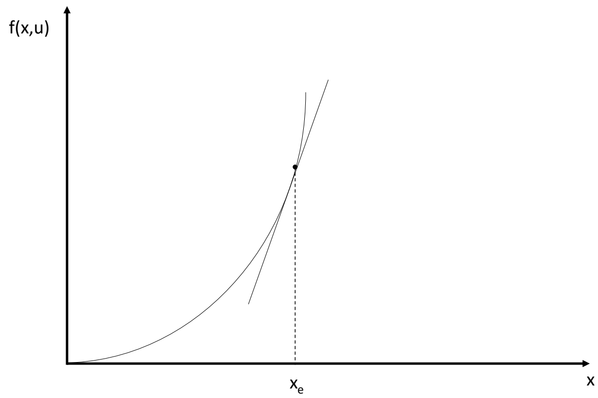

Figure 7.1: A sketch of the appraoch that is used for a linearisation

The non-linear equation can be described as:

$$\dot{x} = f(x)$$

The equalibrium would be:

$$x = x_0$$

Small pertubation around that operations point are:

$$x = x_0 + \delta x$$

The Taylor-Series Approximation around the pertubation is described followed :

$$\dot{x} + \delta\dot{x} = f(x_0 + \delta x) = f(x_0) + \frac{\partial f}{\partial x} + \frac{\partial^2 f}{\partial x^2}\frac{(\partial x)^2}{2!} + ...$$

Higher orders of the equation are not considered since theses are very smal and can be neglected. The resulsts of this step can be described in a linear system.

$$\dot{x} = Ax + Bu$$

$$A = \frac{\partial f(x,u)}{\partial x}$$

$$B = \frac{\partial f(x,u)}{\partial x}$$

7.3 Appraoch in Matlab

The linearization of the flight dynamic model of the missile has been achieved by creating a MATLAB script. The MATLAB script is loading the flight model of the rocket and linearise it. For the linearization, the standard MATLAB function linmod() has been utilised. The function uses the previously calculated trimming points which are saved in a separate MATLAB file and uses them as the equilibrium point from where the Taylor Series expansion starts. For the Taylor Series, only the first order will be considered since it can be assumed that the higher order parts will have negligibly small values. By having only the first order derivatives of the function, the Taylor series can be put into the Jacobian matrix. From that point, the A, B, C and D matrices can be developed.

Algorithmus 7.1 Structure and function of the trimming algorithm for the example of Mach=0.7 and Altitude=0

begin
//Loading Trimming Data
Input : *Trimming Solutions*;
var $X[12] := LoadedTrimStateVector$
var $U[4] := LoadedTrimCommandVector$
While $calculatedAllDerivates\ \ == \ \ False$ than **do**
  $calculatedAllDerivates\ \ = derivateTheGivenFlightModel()$
end While
//Print and Save the Results ofthe Trimming
Output : $Longitudinal(A,B,C,D)$;$Lateral(A,B,C;D)$
end

7.4 Results

The linearization of the flight model had been made for the trimming point of Mach=0.7 and an altitude of MSL. The results of the linearization are presented in this section.

7.4.1 Linearized Longitudinal Model

$$A_{Longitudinal} = \begin{bmatrix} -0.0607 & 0.0039 & -3.9578 \\ -0.0561 & -3.5066 & 238.1547 \\ 0.0450 & -2.7205 & -0.2472 \\ 0 & 0 & 1 \end{bmatrix}$$

$$B_{Longitudinal} = \begin{bmatrix} 6.1351 \\ -155.0421 \\ -413.4210 \\ 0 \end{bmatrix}$$

$$C_{Longitudinal} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D_{Longitudinal} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The longitudinal flight behaviour can be computed with:

$$\dot{x}_{Longitudinal} = A_{Longitudinal} x_{Longitudinal} + B_{Longitudinal} u_{Longitudinal} \tag{7.1}$$

$$y_{Longitudinal} = C_{Longitudinal} x_{Longitudinal} + D_{Longitudinal} u_{Longitudinal} \tag{7.2}$$

7.4.2 Linearized Longitudinal Model

$$A_{Lateral} = \begin{bmatrix} -3.531 & 3.9578 & -238.1547 \\ -0.0064 & -9.8464 & 0 \\ 0.0161 & 0 & -0.2472 \\ 0 & 1 & 0.0166 \end{bmatrix}$$

$$B_{Lateral} = \begin{bmatrix} 0 & 155.0421 \\ 0 & 0 \\ 0 & -7.2156 \\ 0 & 0 \end{bmatrix}$$

$$C_{Lateral} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D_{Lateral} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The lateral flight behaviour can be computed with:

$$\dot{x}_{Lateral} = A_{Lateral} x_{Lateral} + B_{Lateral} u_{Lateral} \tag{7.3}$$

$$y_{Lateral} = C_{Lateral} x_{Lateral} + D_{Lateral} u_{Lateral} \tag{7.4}$$

Chapter 8

Stability

8.1 Introduction

The stability of a flight vehicle is a critical topic in the field of aerospace engineering. The stability of a flight vehicle can be sub-divided into two parts: static stability and dynamic stability. In general, it can be said that stability refers to the tendency of the vehicle response to a perturbation of the equilibrium of flight. If the behaviour of the missile got a tendency which brings it back to the equilibrium we call it a stable system. If the tendency of the missile leads to a divergent behaviour, we call it unstable behaviour. In the aerospace sector, the stable behaviour is the characteristic which is demanded. This chapter analyses the stability characteristics of the missile and presents the results of the numerical analysis.

8.2 Theory

The stability behaviour of an aircraft can be described as an eigenvalue problem. That means that the eigenvalues of the flight dynamic model will deliver a solution which can be used to understand the stability characteristics. Since the system-matrices A of the missile for the longitudinal and lateral direction are already been computed, these matrices can be used to calculate the eigenvalues. The eigenvalues for the longitudinal matrix delivering the characteristics of the short-period oscillation and the phugoid. The approach for that is:

$$\det(\lambda I - A_{lon}) = (\lambda^2 + 2\zeta_{ph}\omega_{ph}\lambda + \omega_{ph}^2)(\lambda^2 + 2\zeta_{sp}\omega_{sp}\lambda + \omega_{sp}^2) = 0 \tag{8.1}$$

The eigenvalues for the lateral matrix delivering the characteristics of the dutch-roll, roll and spiral mode. The approach for that is:

$$\det(\lambda I - A_{lat}) = (\lambda^2 + 2\zeta_{DR}\omega_{Dr}\lambda + \omega_{DR}^2)(\lambda^2 + \omega_{roll})(\lambda^2 + \omega_{spiral}) = 0 \tag{8.2}$$

8.3 Approach in MATLAB/Simulink

For the stability analysis, a MATLAB script has been created. The script loads the previously computed linearised and calculates the eigenvalues of the lateral and the longitudinal system separately. For calculating the needed matrices the MATLAB standard functions 'eig()' and 'damp()' has been utilized. After computing the values the script print out on the command window and save the results as a MATLAB file.

8.4 Results

The stability analysis delivered following the results which are given in the following sections.

8.4.1 Dynamic Modes - Longitudinal

Figure 8.1 is showing the complex plane diagram created by the MATLAB script. The diagram shows the plots of the eigenvalues for the longitudinal motion of the missile. On the diagram, you can see on the left side two solution points with a real value of -1.876 and the imaginary values of +- 25.406i. These solutions are related to the short-period oscillation of the missile. Interpreting the values leads to the conclusion that this motion type got a high natural frequency but is on the other hand highly damped. The plot on the right side of the diagram is the phugoid mode. On figure 8.1 it looks like that it is only one point, but in reality that are two points which are just overlapping due to the scale of the picture. The phugoid is also a longitudinal eigenmode which has a lower frequency and lower damping as the short period oscillation. The real part for this mode is -0.0312 and the imaginary value is +-0.0607.

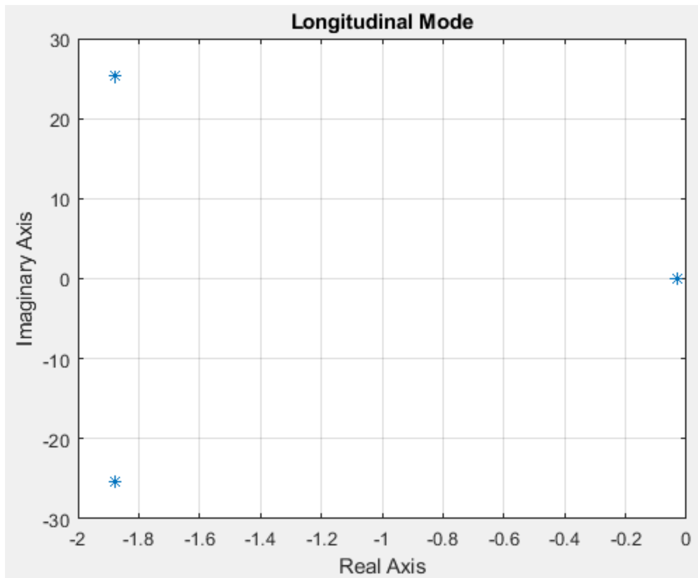

Figure 8.1: The plot of the longitudinal eigen-modes in the z-plane

Short Periode

Eigenvalue: $p = -1.8760 \pm 25.4051i$

Natural Frequecy: $\omega = 0.0736[Hz]$

Damping: $\zeta = 25.5$

Phugoide

Eigenvalue: $p = -0.0312 \pm 0.0607i$

Natural Frequecy: $\omega = 0.458[Hz]$

Damping: $\zeta = 0.0682$

8.4.2 Dynamic Modes - Lateral

Figure 8.2 is showing the complex plane diagram created by the MATLAB script. The graph shows the plots of the eigenvalues of the lateral motion of the missile. On the very left side of the diagram, a plot with a real value of -9. 84 is given. That plot is related to the roll motion of the missile. That motion is an aperiodic stable eigenmode of the flight vehicle. The two plots at the real value of -1.89 are related to the dutch-roll mode of the missile. This mode is oscillating behaviour of the aircraft which is coupling the lateral and directional motion of the aircraft. The plot at the real value 0 is the spiral model of the missile. The value can be used as the conclusion that this flight vehicle does not provide that eigenmode.

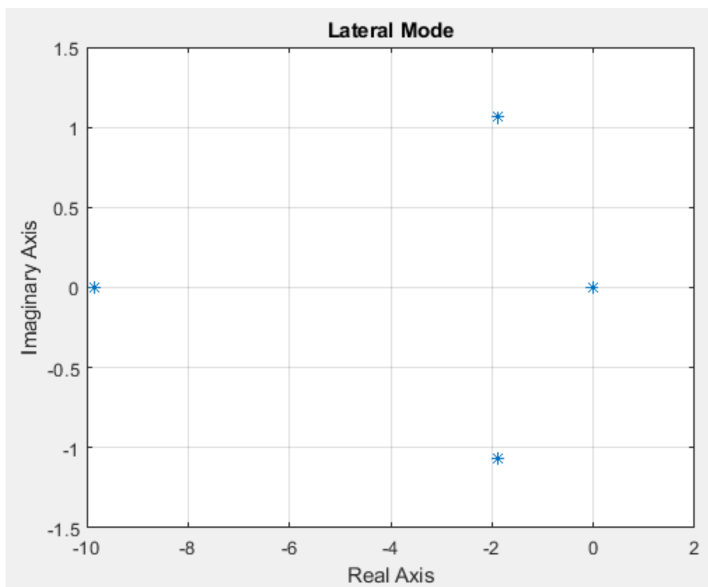

Figure 8.2: The plot of the lateral eigen-modes in the z-plane

Dutch Roll

$$\text{Eigenvalue: } p = -1.89 \pm 1.07i$$

$$\text{Natural Frequecy: } \omega = 2.17[Hz]$$

$$\text{Damping: } \zeta = 0.871$$

Roll

$$\text{Eigenvalue: } p = -9.84$$

Spiral

$$\text{Eigenvalue: } p = 0.222 \times 10^{-3}$$

Chapter 9

Control Laws for a Missile System

9.1 Introduction

Modern military flight vehicle systems rely on complex flight computers to command and protect the missile in flight. These abilities are administered by computational laws which are committed by a flight control system, also called autopilot. Autopilots provide augmentation in normal flight, such as increased protection of the missile from high loads or providing the ability to reach a certain target autonomously. In piloted flight vehicles, control is achieved by the pilot's control inputs s that regularly move cables, pulleys or hydraulic servo valves which in command control surfaces or change engine settings. Since missile systems are unmanned, it is necessary to replace these mechanical controls with autopilot systems. These missiles have flight control computers which send electronic signals to operate control surfaces or engine controls. This chapter will present the implementation of a given autopilot in the missile model and its gains adjustments to fit a certain trimming state.

9.2 Theory

Since we trimmed the aircraft for a specific state the task of the controller is it to hold that state and secure that the flight vehicle will come back to that perturbation events occur. The trim state is a state in which the moments around the centre of gravity are zero, and the attacking forces are balanced. Figure 9.1 shows that behaviour for the pitching moment which becomes zero at the trimming point. That leads to a steady state flight of the missile system. Figure 9.1 also shows that the adjustment of the pitching command can be used to reach a certain trimming point. That behaviour is commonly used to trim an aircraft in a particular state. The pilot or the control system augmented a static command which holds the missile in a demanded state. Additionally, a dynamic behaviour stabilises the air vehicle around that point balances the system against small perturbations.

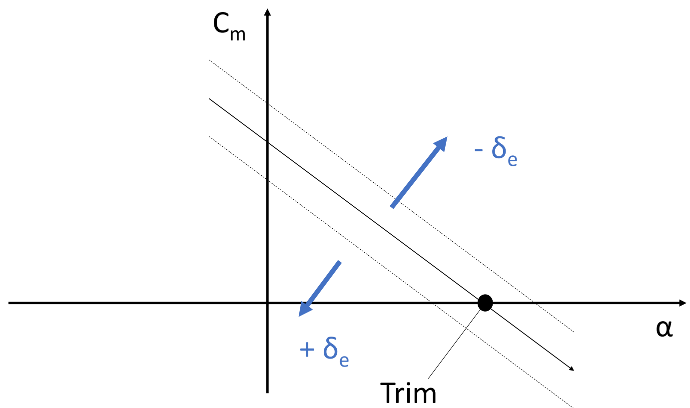

Figure 9.1: The functional behaviour of the pitch moment coefficient and the influences of the controll input

9.3 Appraoch in Matlab/Simulink

The previous section explained behaviour which is used to adjust the control laws of the missile system. The missile needs to be trimmed for $Mach = 0.7$ and $Altitude = 0$. While the reference values which are imposed $a_{y_{cmd}} = 10\frac{m}{s}$, $a_{z_{cmd}} = 10\frac{m}{s}$ and roll command is zero.

The Initial state of the parameters is the earlier computed state vector x_{trim}. These values need to be loaded and implemented into the integrator blocks of the equation of motion subsystem. The calculated r u_{trim} values are the augmented control surface command which needs to be implemented in the actuator model. Figure 9.2 is showing how the additional commands are implemented to the actuator model. The goal of the approach is it to find the best fitting gains of the autopilot to create a control system that follows the r a_z and a_y command with a minimal steady state error.

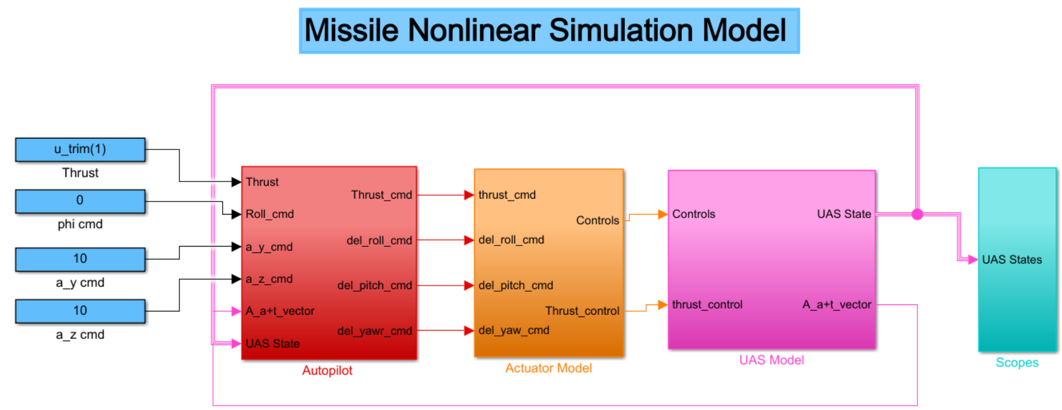

Figure 9.2: The overall of all Systems and the input of the reference values

The u_{trim} values have been added to the output of the actuator system in way that the trimming values are augmented to the computed ones.

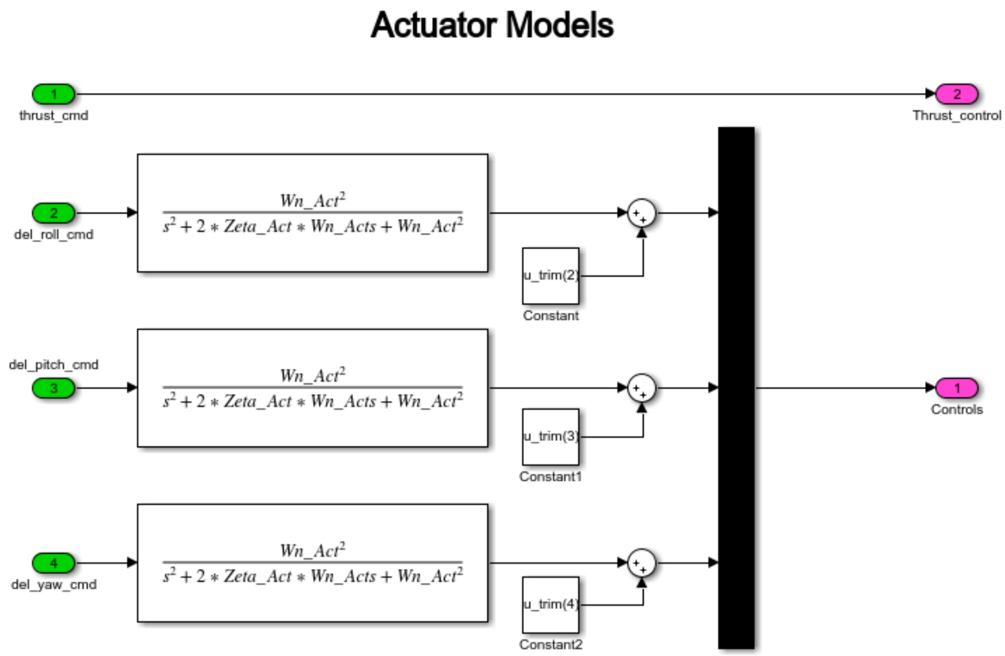

Figure 9.3: The Actuator models with the trim command augmentation

9.4 Results

This section presents two possible solution approaches for the given task. The results had been achieved by using manual iteration steps in which the gains have been adjusted. During the procedure, it became evident that two possible solutions could be used. Both results got their advantages and disadvantages, and in the end, the requirements of the mission should decide which technical answer is the best fitting. In the following, both results are presented and discussed.

9.4.1 Control solution with overshot

This solution is one possible approach to achieve a stable converging behaviour of the command input and the system output. The disadvantage of this strategy is that the controller creates in both, $a_z$and a_y, an overshoot above the reference value. But on the other hand, the two values are converging fast, compared to the other approach. The used values for that are given in table 9.1 and the figures 9.4 and 9.5 are showing the behaviour of the controlled values ay and az.

Gain	K_ϕ	K_p	K_{DC}	K_A	ω_i	K_R
Value	20	0.01	0.957	0.15	-0.83	0.0254

Table 9.1: The Gains for the found Control Approach with overshoting but fast converging behaviours

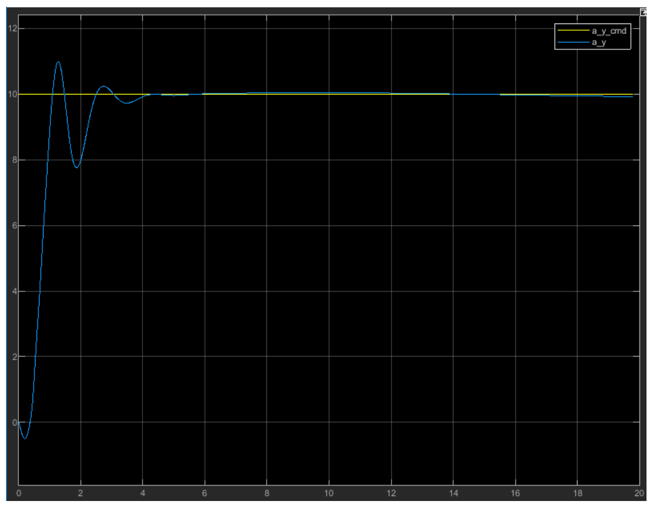

Figure 9.4: The system response of the $a_y$controll with a overshoot

It is interesting that the start values of a_y and a_z are both different. The start value of a_y is zero, while the start value of a_z is around -10. That is because the aerodynamic coefficient C_z start as a negative value which creates a negative force, which can be described as the lift. The main reason for the negative value of C_z are the two coefficient elements C_{z_0} and $C_{z_{\delta p}}$.

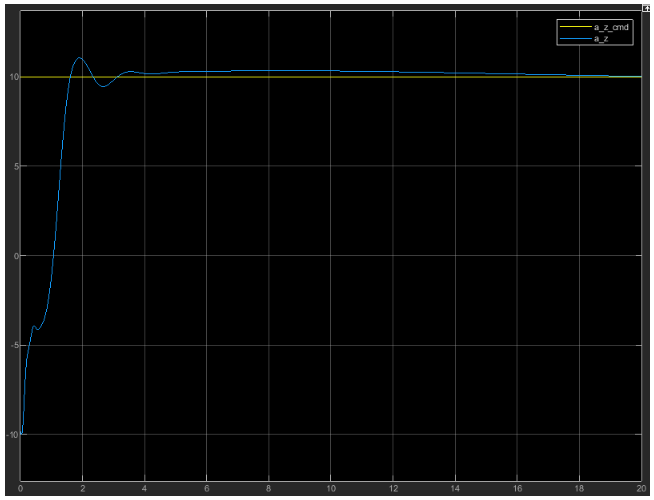

Figure 9.5: The system response of the $a_z$controll with a overshoot

9.4.2 Control solution without overshot

This proposed solution is another possible approach to achieve a stable converging behaviour of the command input and the system output. The disdvantage of this approach, compared to the other solution proposal, is that the controller needs a longer time to make the reference and output values converge. But on the flip side, the output value is not overshooting. That could be an advantage if overshoots are highly restricted. The used values for that are given in table 9.2 and the figures 9.6 and 9.7 are showing the behaviour of the controlled values ay and az.

Gain	K_ϕ	K_p	K_{DC}	K_A	ω_i	K_R
Value	20	0.01	0.96	0.87	-0.11	0.0254

Table 9.2: The Gains for the found Control Approach without overshoting but slow converging behaviour

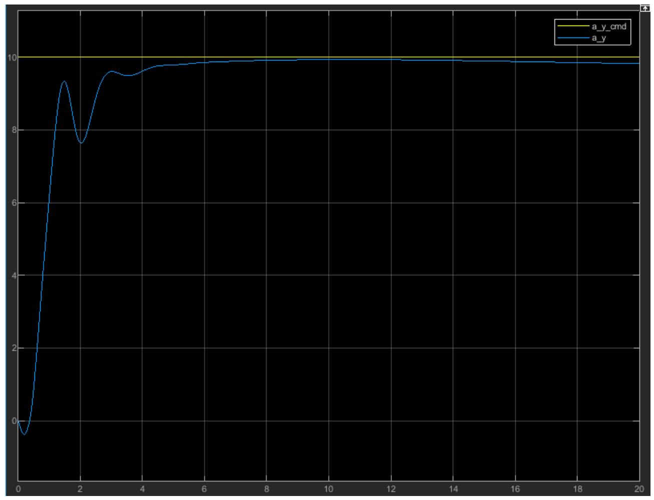

Figure 9.6: The system response of the $a_y$controll without overshoot

It is interesting that the start values of a_y and a_z are both different. The start value of a_y is zero, while the start value of a_z is around -10. That is because the aerodynamic coefficient C_z start as a negative value which creates a negative force, which can be described as the lift. The main reason for the negative value of C_z are the two coefficient elements C_{z0} and $C_{z_{\delta p}}$.

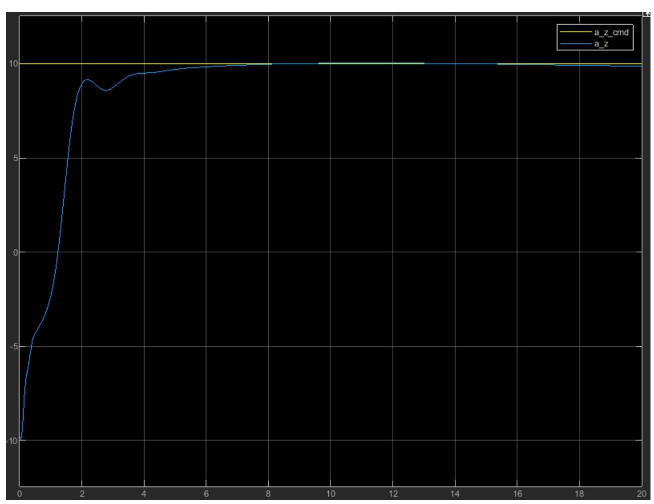

Figure 9.7: The system response of the $a_z$controll without overshoot

Chapter 10

Conclusion

Within this assignment, A Technical Report Modelling And Simulation In MATLAB/Simulink, has been exhibited. Chapter one was used to describe the general model conceptualisation of the created MATLAB model. Chapter two explained the used flight dynamic equations of the missile in detail. Chapter three was used to present the established Simulink model of the missile. Chapter four presented the applied verification approaches to test the correct behaviour of the model. The fifth section is dealing the approaches to find trim states for given flight conditions. In this chapter, the approaches, as well as the results, had been discussed. The sixth part of the assignment presented the work for the linearization of the flight dynamic model for a given trim point and its results. In the seventh chapter stability of the missile system has been analysed. For the previously linearised model has been used. The last chapter presented the approach to adjust the delivered controller for a trimming point.

All the objectives declared in the preface of the work have been successfully achieved. Unfortunately, the model couldn't been verified with more test cases due to a restricted amount of information. Also, the model behaviour got some critical points which should be clarified in the future work. For example, the provided aerodynamic coefficient of the system is only valid till a Mach number of 0.7, but the simulation has been done till a Mach of 1.1. That could deliver a behaviour that is not representing the real missile behaviour in that flight speed. Furthermore, it could be sufficient to exchange the current autopilot system with a PID-Controller which would be possibly able to create a better system control behaviour.

Bibliography

[1] J. D. Anderson. *Fundamentals of Aerodynamics*. McGraw-Hill, 2001.

[2] D. K. Arrowsmith and C. M. Place. *"The Linearization Theorem" - Dynamical Systems: Differential Equations, Maps, and Chaotic Behaviour*. London: Chapman & Hall, 1992.

[3] M. V. Cook. *Flight dynamics principles: a linear systems approach to aircraft stability and control*. Butterworth-Heinemann, 2012.

[4] A. T. John Doyle, Bruce Francis. *Feedback Control Theory*. Macmillan Publishing, 1990.

[5] B. N. N. JOHN E. CASHMAN, BRIAN D. KELLY. Operational use of angle of attack. *Aero magazine*, 2010.

[6] G. LAJOIE. The hartman-grobman theorem and the equivalence of linear systems. Online, 2006.

[7] R. C. Nelson. *Flight stability and automatic control*, volume 2. WCB/McGraw Hill New York, 1998.

[8] T. Owusu. 30 of the most powerful military invoations in history. Technical report, The Street, 2018.

[9] R. F. Stengel. *Flight Dynamics*. Princeton University Press, 2004.