Feedback Linearization for Model Agnostic Aircraft Simulation Control

Fredrik Horn

LUND UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6219
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2023

Abstract

This thesis presents a method, based on feedback linearization, for controlling a general 3-degree-of-freedom simulation model of an aircraft. Only aircraft models that utilize Taylor approximations of the aerodynamic coefficients are considered. The intent is to be able to simulate the same mission with different aircraft models without having to tune the control system specifically for each model.

The method is based on feedback linearization and generates inverse models from the original aircraft model. These are then used to control different variables using a linear controller. Special care was taken to be able to switch the controlled variables during a flight and smooth out the transitions.

Two test missions were successfully implemented based on the constant altitude cruise and cruise climb flight strategies respectively. After some tuning of the linearized controller, the desired trajectories were followed well. However, additional tuning and analysis of the linear controller could be performed to ensure more specific performance targets are met. Some work should also be done to make the method easier to use with regards to the mission specification the user provides. A better mission segment handler and state event based segments would reduce the requirements on the mission specification and thus make it easier to define missions that work for multiple aircraft.

Acknowledgements

First and foremost, I would like to extend my sincere appreciation to Ivar Torstensson, my supervisor at Modelon. Ivar's assistance, valuable insights, and engaging discussions have played a pivotal role in shaping my work.

I would also like to thank my supervisor Bo Bernhardsson and examiner Yiannis Karayinidis for their willingness to offer their expertise and ideas. Their input and suggestions have widened my theoretical understanding of the subject and I am thankful for their continuous support and encouragement.

I would like to acknowledge Jim Claesson at Modelon for his helpfulness and responsiveness. Jim's willingness to address my questions and provide guidance has been instrumental in overcoming obstacles and refining my work.

Lastly, I would like to thank Modelon as a whole for giving me the opportunity to work with an interesting and applicable subject.

To those mentioned above and many others who have supported me in various ways, I extend my deepest appreciation. Thank you!

Fredrik Horn
Lund, Sweden
June 2023

Contents

1

Introduction

With the recent push towards sustainable flight many novel aircraft designs are being researched which creates a need for efficient performance simulations of different designs. For example, suppose you have a number of different proposed motors, the same with different proposed wing designs, cargo and fuel solutions. In the search for the most promising combination of these parts, a huge number of simulations need to be performed to evaluate the performance of the designs.

Traditionally, flight simulations are performed by first creating one or several mission specifications. These specify a how the airplane should fly, prescribing for example the target altitude and airspeed. Then a model of the aircraft and its aerodynamics is made. The model takes e.g. control surface angles and throttle as inputs and outputs the relevant states of the aircraft, e.g. altitude, pitch angle etc. Then a control system is designed and tuned to ensure a flight profile will be followed. This is an incredibly tedious step if many different aircraft models are to be simulated.

When simulating, the states of the aircraft can be easily accessed which could yield a unique opportunity to use feedback linearisation with high accuracy even when the model is highly nonlinear. This thesis aims to investigate if this approach is feasible for controlling aircraft in simulations without the need to perform manual tuning of the control system for each design. This would drastically reduce the manual work.

1.1 Aim

The aim of this thesis is to investigate the feedback linearization approach for controlling aircraft in simulation with minimal manual tuning, creating an easy way to simulate and compare many different designs.

1.2 Limitations

The work will be limited to 3 degree of freedom [DOF] aircraft models. It will also be limited to Taylor approximations for the aerodynamic coefficients (C_L, C_D, C_M) meaning no models involving lookup tables or other methods of determining these coefficients are evaluated.

1.3 Modelica

Modelica is an equation-based language that is widely used for modeling and simulation of complex systems. Developed and maintained by the Modelica Association, it is an open standard that allows for the creation of reusable models and libraries. The versatility of Modelica allows for the modeling of a wide range of systems, including mechanical, electrical, thermal, control systems and in this case flight dynamics. The ability to simulate Modelica models using various simulation tools, such as Modelon Impact, makes performing and building simulations easy compared to writing simulation tools from scratch. Modelica is used in a wide range of applications and research projects, particularly in the field of system dynamics.

1.4 Modelon Impact

Modelon is a company specializing in Modelica solutions, Modelica compilers and their cloud service Modelon Impact. Modelon Impact is a tool that equips the user with a graphical interface, simplifying the process of writing, connecting, compiling, and running Modelica models. This is the tool that was used throughout the thesis work.

1.5 Mission Specification

An example mission specification is shown in Table 1.1, which is a sequence of mission segments. It describes how the aircraft should fly a mission and needs to be translated into a form the controller can understand. As can be seen different segments require different modes of control with different controlled variables.

Table 1.1 Example mission, the ID describes the order of the segments and the description describes the segment.

ID	Description
1	Takeoff acceleration at full thrust
2	Takeoff rotation: Increase lift to equal weight at full thrust
3	Climb to final altitude of first segment, 35 ft: Start increasing climb rate at full thrust
4	Climb while reducing to part thrust, acceleration up to 250 kts CAS
5	Climb to FL 100 at constant climb rate and CAS, 1600 ft/min
6	Accelerate to 300 kts CAS at constant climb rate, 1500 ft/min
7	Climb to FL 35 at reduced climb rate, constant CAS and max thrust
8	Cruise at initial cruise altitude (before step-up), accelerate to cruise Mach
9	Cruise at initial cruise altitude (before step-up), hold cruise Mach
10	Step-up cruise altitude at max thrust, hold cruise Mach
11	Cruise at final cruise altitude (after step-up), hold cruise Mach
12	After cruise initiate descent up to 300 kts CAS, 1700 ft/min
13	Descent at 300 kts CAS, 1700 ft/min
14	Deccelerate to 250 kts CAS at constant descent rate, 1300 ft/min
15	Descent at 250 kts CAS and low descent rate, 300 ft/min
16	Descent at 250 kts CAS and low descent rate
17	Braking

1.6 Previous Work

M. Hardt and R. Höpler makes an aircraft model follow specialized commands in [1]. Inspiration for the mission specification structure was taken from this work.

Fabrizio Re implements a feedback linearizing controller in Modelica similar to the one constructed in this work for aircraft taxi systems in [2].

2

Theory

2.1 Aircraft Dynamic Model

To be able to simulate an aircraft a dynamic model of it is required. As mentioned earlier, this work will be limited to 3-DOF models meaning the aircraft position and orientation is described using 3 variables: altitude, distance and pitch attitude. Figure 2.1 illustrates these variables where the altitude is the height above the ground plane (H in Figure 2.1), distance is the distance traveled relative to ground (X_{earth} in Figure 2.1) and the pitch attitude is the angle of the aircraft's x-axis (where the nose points) to the plane parallel to ground (θ in Figure 2.1). This means that the aircraft is constrained to a plane where no lateral dynamics occur.

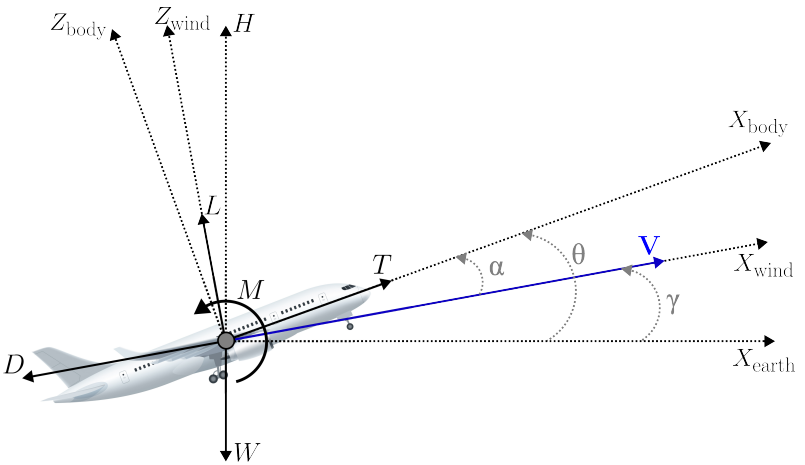


Figure 2.1 Fundamental coordinate systems, angles, forces and moments of a 3-DOF aircraft model.

Figure 2.1 furthermore shows the angle of attack [AoA] α, which is the angle between the aircraft's x-axis (where the nose points) and the current velocity vector

$\mathbf{V}$ (shown in blue), and the flight path angle γ which is the angle the aircraft is currently traveling in with respect to the plane parallel to ground. The forces and the pitching moment about the aircraft center of mass are also shown in solid black.

Forces

The fundamental forces on an aircraft are: lift, drag, thrust and weight. These must all be calculated to be able to perform a dynamic simulation. Figure 2.1 shows these forces, where the thrust T is acting in the direction of the aircraft x axis (not necessarily true in the general case, but a reasonable assumption). The weight vector W is parallel to gravity, the drag D is acting in the opposite direction of the current velocity vector $\mathbf{V}$ and the lift L is perpendicular to $\mathbf{V}$. In the simplest case all these forces cancel out and steady state flight is achieved. Note that for steady state flight it is not required that the thrust and weight are orthogonal to each other or to the aerodynamic forces, only that all forces cancel out. Figure 2.2 shows the simplest case.

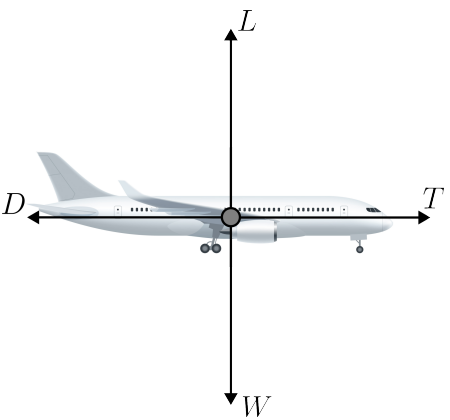


Figure 2.2 Steady state equilibrium forces on an aircraft.

Starting by defining the dynamic pressure $\bar{q}$ on the aircraft according to equation (2.1) [3], where ρ is the air density and v_{TAS} is the true airspeed of the aircraft (the magnitude of $\mathbf{V}$ in Figure 2.1), meaning the speed of the aircraft relative to the local wind, or the incoming wind speed relative to the aircraft depending on which reference frame is used. Note that this is different from the aircraft speed relative to ground as e.g. a headwind will increase this quantity even if the speed relative to ground remains the same [3].

$$\bar{q} = \rho \frac{v_{\text{TAS}}^2}{2} \tag{2.1}$$

Using the dynamic pressure we can define expressions for the lift (L) and drag (D) according to equations (2.2) and (2.3). A is the wing area and C_L, C_D are the

coefficients of lift and drag respectively [3]. Drag is always defined to be in the opposite direction as the current velocity vector (relative to local wind) and lift is defined to be perpendicular to the velocity vector, see Figure 2.1.

$$L = C_L A \bar{q} \tag{2.2}$$

$$D = C_D A \bar{q} \tag{2.3}$$

Note that the coefficients C_L and C_D are not constants, but rather (often highly nonlinear) functions of the current aircraft state (usually depending on air temperature, Mach number, angle of attack α and control surface deflections) [3]. This is usually where the specific aerodynamic modelling takes place for an aircraft and can be done in multiple different ways. One common way is to find coefficients to the Taylor expansion of C_L and C_D around $\alpha = 0$ (and other variables depending on how complex the model is). This is the way the model used in this thesis works and this approximation works well as long as the aircraft stays sufficiently close to $\alpha = 0$. Another way to model these functions is to simply use tabulated values for C_L and C_D, which can be acquired from testing or simulations, and interpolate between them. By increasing the size of the model it can become very accurate even for high α.

Pitching Moment and Elevator

The pitching moment on the aircraft is calculated in a similar manner to the aerodynamic forces as shown in equation (2.4) [3]. This time however, the constant $\bar{c}$ is multiplied by the pitching moment coefficient C_M, the wing area A and the dynamic pressure $\bar{q}$. $\bar{c}$ is the mean aerodynamic chord of the wing which is constant for fixed wing aircraft.

$$M = C_M \bar{c} A \bar{q} \tag{2.4}$$

Controlling a 3-DOF model is done via the elevator and throttle. The elevator has a great effect on the pitching moment coefficient which enables pitch control of the aircraft [3]. In reality all other aerodynamic coefficients change with elevator deflection but in most cases it is reasonable to assume that it will have a comparatively small effect on the other coefficients [3]. The model used in this thesis applies this assumption.

Thrust and Fuel

Engine and thrust models can of course be of varying fidelity, and these are not the focus of this thesis. In simpler models, the trust is often modeled as a first order system with the throttle position (u_{throttle}) as the input [3]. In this work however, it is simply modeled as a proportionality variable depending on the true airspeed, see equation (2.5).

$$T = f(v_{\text{TAS}}) \cdot u_{\text{throttle}} \tag{2.5}$$

A related variable of great importance is the mass of the aircraft, specifically how the mass changes as fuel is burnt. This is of course also subject to great modeling variety. A simple model is to use the thrust specific fuel consumption [TSFC] and that way calculate the fuel consumption based on the output thrust of the engine at each instant [4]. See equation (2.6) where m_{fuel} is the fuel mass and T is current thrust of the engine. Note that in reality the TSFC is not constant but rather a function of parameters such as throttle, true airspeed, air density etc. This is a simplification that holds close to the state of the aircraft when it was measured.

$$\dot{m}_{\text{fuel}} = -\text{TSFC} \cdot T \tag{2.6}$$

2.2 Feedback Linearization

Feedback linearization, also known as nonlinear dynamic inversion [NDI] or just dynamic inversion [DI], is a method of exactly linearizing (different from linearizing around a point) a system using information about the model and states. Consider controlling a general nonlinear system such as the one in equation (2.7) where f and g are some nonlinear functions, x is the state vector and u is the control signal.

$$\begin{aligned} \dot{x} &= f(x, u) \\ y &= h(x, u) \end{aligned} \tag{2.7}$$

Aircraft systems can often naturally be described using the less general control signal affine form [3], shown in equation (2.8), and if not there are strategies to transform the system to this form [5].

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{2.8}$$

Taking the derivative of the output y of equation (2.8) yields equation (2.10) where h_x is defined according to equation (2.9) and n is the number of states of the model.

$$h_x = \left(\frac{\partial h}{\partial x_1}, \ldots, \frac{\partial h}{\partial x_n} \right) \tag{2.9}$$

$$\dot{y} = h_x(x)\dot{x} = h_x(x)(f(x) + g(x)u) = h_x f(x) + h_x g(x)u \tag{2.10}$$

If $h_x g(x) \neq 0$ equation (2.10) clearly shows that $\dot{y}$ can be controlled exactly if h, f, g and the full state vector x are known by selecting u according to equation (2.11) where $\dot{y}_{\text{cmd}}$ is the commanded $\dot{y}$.

$$u = \frac{\dot{y}_{\text{cmd}} - h_x f(x)}{h_x g(x)} \tag{2.11}$$

If however $h_x g(x) \equiv 0$ the output y needs to be differentiated again and $\ddot{y}$ will be the new variable to command. To write this in a concise manner the Lie derivatives L_f and L_g are introduced in equation (2.12) [6].

$$\begin{aligned} L_f &= f_1 \frac{\partial}{\partial x_1} + \cdots + f_n \frac{\partial}{\partial x_n} \\ L_g &= g_1 \frac{\partial}{\partial x_1} + \cdots + g_n \frac{\partial}{\partial x_n} \end{aligned} \tag{2.12}$$

By differentiating $\dot{y}$ and using the Lie derivative notation we obtain equation (2.13) (if $h_x g(x) \equiv 0$) where repeated Lie derivatives are denoted L^v with v being the number of Lie derivatives taken.

$$\begin{aligned} \ddot{y} = (h_x(x)f(x))_x f(x) + (h_x(x)f(x))_x g(x)u = \\ L_f(h_x f(x)) + L_g(h_x f(x))u = L_f^2(h) + L_g L_f(h)u \end{aligned} \tag{2.13}$$

The number of differentiations needed to find the control signal u explicitly in the expression is called the relative degree of the system and will also be the order of the resulting linear system [6]. Assuming $L_g L_f^w(h) \equiv 0 \quad \forall\ w < v - 1$ and $L_g L_f^{v-1}(h) \neq 0$ the general equation (2.14) holds.

$$y^{(v)} = L_f^v(h) + L_g L_f^{v-1}(h)u \tag{2.14}$$

It now follows that the system can be controlled using equation (2.15) especially since $L_g L_f^{v-1}(h) \neq 0$.

$$u = \frac{y_{\text{cmd}}^{(v)} - L_f^v(h)}{L_g L_f^{v-1}(h)} \tag{2.15}$$

The case $L_g L_f^w(h) \equiv 0 \quad \forall\ w < v - 1$ but $L_g L_f^{v-1}(h) = 0$ for some x is a more complicated one, but can be dealt with with a lot of extra care [3]. There is no problem however if this only occurs in parts of phase space that are never reached.

This allows for the final controller structure shown in Figure 2.3. The states x are fed back to the inverse which linearizes the system, hence the name feedback linearization.

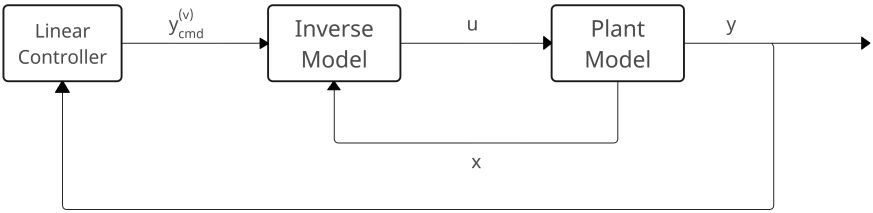


Figure 2.3 Feedback linearization controller structure.

Since this is intended to be used in a simulation, the aircraft model ($f(x)$, $g(x)$ and $h(x)$) is known exactly and all states (x) can be easily accessed. The Modelica compiler is exceptionally good at differentiating expressions and will automatically perform the differentiations needed to linearize the model and compute the inverse.

MIMO Systems

The theory on feedback linearization above is so far restricted to SISO [Single Input Single Output] systems. The type of aircraft model used in this work is a MIMO [Multiple Input Multiple Output] system however, with two inputs (elevator angle and throttle position) and two outputs (the two controlled variables e.g. altitude and true airspeed). In the MIMO case for a system with n states, m inputs and l outputs, we now have a nonlinear system (in control signal affine form) shown in equation (2.16). Here u, y and $h(x)$ are now vectors of sizes m, l and l respectively and $G(x)$ is now a $n \times m$ matrix [7].

$$\begin{aligned} \dot{x} &= f(x) + G(x)u \\ y &= h(x) \end{aligned} \tag{2.16}$$

The relative degree of the system is now replaced by the vector relative degree $\{r_1, ..., r_l\}$, where r_i is the smallest relative degree of each output to any of the inputs [7]. With these relative degrees the $l \times l$ decoupling matrix $A(x)$ can now be formed, shown in equation (2.17). This is the MIMO generalisation of $L_g L_f^{v-1}(h)$, where g_1 - g_m are the columns of $G(x)$. It both linearizes the maps between inputs and outputs and decouples the controlled variables, meaning each variable can be controlled independently. The condition that $L_g L_f^{v-1}(h) \neq 0$ is now replaced with a requirement that the decoupling matrix is nonsingular [7]. Note that for the decoupling matrix to be invertible it has to be square meaning the number of inputs have to equal the number of outputs $m = l$.

$$A(x) = \begin{bmatrix} L_{g_1} L_f^{r_1-1}(h_1(x)) & \dots & L_{g_1} L_f^{r_l-1}(h_l(x)) \\ \vdots & \ddots & \vdots \\ L_{g_l} L_f^{r_1-1}(h_1(x)) & \dots & L_{g_l} L_f^{r_l-1}(h_l(x)) \end{bmatrix} \tag{2.17}$$

Let us also define the vector $B(x)$ according to equation (2.18).

$$B(x) = \begin{bmatrix} L_f^{r_1}(h_1(x)) \\ \vdots \\ L_f^{r_l}(h_l(x)) \end{bmatrix} \tag{2.18}$$

By differentiating each output until an input appears, i.e. r_i times for each y_i, equation (2.19) is obtained [8].

$$\begin{bmatrix} y_1^{(r_1)} \\ \vdots \\ y_l^{(r_l)} \end{bmatrix} = B(x) + A(x)u \tag{2.19}$$

Now if the decoupling matrix is nonsingular in the relevant part of phase space the control law in equation (2.20) can be used, where $y_{\text{cmd}}^{(\mathbf{r})}$ is the vector of commanded derivatives $y_{\text{cmd}}^{(\mathbf{r})} = \begin{bmatrix} y_{1\text{ cmd}}^{(r_1)} & \dots & y_{l\text{ cmd}}^{(r_l)} \end{bmatrix}^T$.

$$u = A(x)^{-1}(y_{\text{cmd}}^{(\mathbf{r})} - B(x)) \tag{2.20}$$

Zero Dynamics

Some thought needs to be given to the controlled variables of a feedback linearized system. Some dynamics of a feedback linearized system might not be visible in the output y, these are called the zero dynamics [6]. Even if the linearized system is stable these dynamics might be unstable, and since the calculated control signals depend on the full state vector x the control signals may be unstable as well, yielding arbitrarily large u which are not realistic. These hidden dynamics might also take the plant to a place in phase space where $L_g L_f^{v-1}(h) = 0$ (or the decoupling matrix is singular), further complicating the control method.

2.3 Flight Control Strategies

There exist several common methods to try to maximize the range or minimize the fuel consumption for a specific flight. A commonly applied method to achieve this is to fly at the highest possible lift-to-drag ratio $\frac{C_L}{C_D}$ [9]. This means that the most amount of lift possible is generated while minimizing the thrust needed to retain the airspeed. These strategies usually result in controlling the lift coefficient C_L and thereby the angle of attack α to some optimal value [9]. Two of these strategies will be presented that are implemented in this work to allow flight profiles of this nature.

Constant Altitude Cruise

The constant altitude cruise strategy works by flying at an optimal angle of attack α while staying at a constant altitude H [9]. However, as fuel is burned off and

the aircraft becomes lighter the airspeed is decreased to match the decrease in lift needed to stay at the commanded altitude. Equations (2.1) and (2.2) show that lift is heavily dependent on airspeed so decreasing it will result in the desired drop in lift.

Cruise Climb

The cruise climb strategy also works by flying at an optimal α but this time keeps the airspeed constant [9]. This will generate a certain amount of lift, but as fuel is burned off and the aircraft becomes lighter this lift will stay the same, meaning the aircraft will start to accelerate upwards. As it ascends however the air density will drop, looking at equations (2.1) and (2.2) it is clear that the lift will drop with altitude, so a new equilibrium will be achieved at a higher flight level.

3

Method

3.1 Aircraft Model

The main aircraft model used was the 3-DOF model proposed by Stevens and Lewis [3]. It was chosen for its relative simplicity while still capturing the most important aspects of flight dynamics. With a simple atmosphere model and traditional aerodynamic equations for calculating the forces on the vehicle it was deemed a good baseline model.

The model does not however take into account fuel consumption and the resulting loss of mass. To be able to simulate e.g. cruise climb missions, the aircraft mass was added as a state and fuel consumption was modeled using the simple thrust-specific fuel consumption model presented in the theory section. The value was taken from [4] for general large aircraft engines, but it was scaled up to see the effect of fuel burn in the 3800 second simulations performed in this work.

Equations of Motion

Equations (3.1)-(3.4) show the equations of motion derived for the 3-DOF aircraft model by Stevens and Lewis [3]. Here again m is the aircraft mass, α the angle of attack, γ the flight path angle and θ is the pitch angle, see Figure 2.1. L, T, D, mg, M and v_{TAS} are lift, thrust, drag, weight, pitching moment and true airspeed respectively (shown in Figure 2.1 as well), where mg (W in Figure 2.1) is the mass times the gravitational constant. Finally Q is the pitch rate and I is the moment of inertia. Note that γ is not a state but simply $\gamma = \theta - \alpha$.

$$\dot{mv_{\text{TAS}}} = T\cos(\alpha) - D - mg\sin(\gamma) \tag{3.1}$$

$$m\dot{\alpha}v_{\text{TAS}} = -T\sin(\alpha) - L + mg\cos\gamma + mv_{\text{TAS}}Q \tag{3.2}$$

$$\dot{\theta} = Q \tag{3.3}$$

$$\dot{Q} = \frac{M}{I} \tag{3.4}$$

Figure 3.1 shows the Modelica implementation of these equations, with the addition of the differential equation $\dot{H} = v_{\text{TAS}} \sin(\gamma)$ to keep track of the altitude.

```
dvt = (THR*CALP - QS*CD)/AM - Modelica.Constants.g_n*SGAM "2.4-23a";
dalpha = (-THR*SALP - QS*CL + AM*(vt*Q + Modelica.Constants.g_n*CGAM))/(AM*vt + QS*CLADOT);
dtheta = Q;
dQ = (QS*CBAR*(CM + D) + THR*ZE)/AIYY;
dH = vt*Modelica.Math.sin(gamma);

// Time derivatives
der(alpha) = dalpha;
der(vt) = dvt;
der(theta) = dtheta;
der(Q) = dQ;
der(H) = dH;
```

Figure 3.1 Modelica code for the equations of motion of the aircraft model, in the bottom the relevant derivative statements are shown. Note that this is not the complete code for the aircraft model, as many of the variables used in these lines are calculated elsewhere.

Atmosphere Model

To be able to calculate the dynamic pressure $\bar{q}$ for the aerodynamic forces and moment (shown in equation (2.1)), the air density ρ is needed as can be seen in equation (2.1). This is heavily dependent on the altitude of the aircraft, so a model of how the air density ρ depends on the altitude is needed. In reality it is dependent on other variables as well but the simple atmosphere model in the 3-DOF model by Stevens and Lewis in [3] was used for simplicity. This model is shown in equation (3.5), where ρ is the air density, ρ_0 is the air density at sea level and H_{ft} is the altitude in feet.

$$\rho = \rho_0(1 - 0.00000703H_{ft})^{4.14} \tag{3.5}$$

Figure 3.2 shows the Modelica code for calculating the dynamic pressure. Note that this is in imperial units since the atmosphere model was implemented that way in [3]. The appropriate unit conversions were made to control the aircraft using metric units.

```
TFAC = 1.0 - 0.703e-5*alt_ft;
RHO = 2.377e-3 * TFAC^4.14;

qbar = 0.5*RHO*vt_ftPerS*vt_ftPerS;
```

Figure 3.2 Modelica code for calculating the air density and the dynamic pressure, this is done in imperial units and later converted to metric-units.

Fuel and Thrust

The thrust from the engine(s) are modeled using a simple proportionality variable as mentioned in Section 2.1. This variable is a linearization of how the maximum thrust depends on the true airspeed at $v_{\text{TAS}} = 0$, as can be seen in equation (3.6).

Here T is the thrust, T_{static} is the thrust of the engine at zero true airspeed ($v_{\text{TAS}} = 0$) and $\frac{\partial T}{\partial v_{\text{TAS}}}\Big|_{v_{\text{TAS}}=0}$ is the partial derivative of the maximum thrust with respect to the true airspeed evaluated at $v_{\text{TAS}} = 0$. T_{static} and $\frac{\partial T}{\partial v_{\text{TAS}}}\Big|_{v_{\text{TAS}}=0}$ are parameters of the aircraft and therefore constant in each simulation run. u_{throttle} is the throttle position in percent, meaning a value of 0 means 0% throttle and a value of 1 means 100% throttle.

$$T = \left(T_{\text{static}} + v_{\text{TAS}} \left. \frac{\partial T}{\partial v_{\text{TAS}}} \right|_{v_{\text{TAS}}=0} \right) u_{\text{throttle}} \tag{3.6}$$

Figure 3.3 shows the Modelica code for calculating the thrust and fuel consumption. The fuel consumption is modeled using the thrust specific fuel consumption presented in Section 2.1. The equation for the mass fuel consumption is shown again in equation (3.7), where $\dot{m}_{\text{fuel}}$ is the fuel mass consumption, TSFC is the constant parameter *thrust specific fuel consumption* and T is the thrust.

$$\dot{m}_{\text{fuel}} = -\text{TFSC} \cdot T \tag{3.7}$$

```
// Thrust and fuel
THR = (TSTAT + vt*DTDV)*THTL;
der(m_fuel) = -TSFC*THR;
```

Figure 3.3 Modelica code for calculating the thrust and fuel consumption.

3.2 Dynamic Inversion

The method of inverting the aircraft model is based on the feedback linearization approach presented in the theory chapter. The model is not inverted analytically however, as that would introduce a lot of manual work and defeat the purpose of the thesis by not making the process more efficient. By a straightforward transformation it is possible to create an inverse block from the aircraft model itself and let the Modelica compiler differentiate the equations to compute the required inverse. This is done by exchanging the relevant derivative statements for inputs and redefining the control signals to outputs. These changes are shown in the Modelica code in Figure 3.4, to be compared with the equations of motion in Figure 3.1.

```
//Commanded derivatives
dvt = xdot_cmd[2];
dH = xdot_cmd[5];

dvt = (THR*CALP - QS*CD)/AM - Modelica.Constants.g_n*SGAM "2.4-23a";
dalpha = (-THR*SALP - QS*CL + AM*(vt*Q + Modelica.Constants.g_n*CGAM))/(AM*vt + QS*CLADOT);
dtheta = Q;
dQ = (QS*CBAR*(CM + D) + THR*ZE)/AIYY;
dH = vt*Modelica.Math.sin(gamma);

// Time derivatives
//der(alpha) = dalpha;
//der(vt) = dvt;
//der(theta) = dtheta;
//der(Q) = dQ;
//der(H) = dH;
```

Figure 3.4 Modelica code showing the changes made of the original aircraft model to create an inverse block. Note how the derivative statements are exchanged for the commanded derivatives of the controlled variables, in this case the true air speed v_{TAS} and the altitude H.

This can then be fed with a commanded derivative "xdot_cmd" in Figure 3.4, calculated by a controller. The required inputs to the aircraft model are thus calculated for it to follow the prescribed mission.

Some care has to be taken to make sure the inverses have access to the correct equations needed to find a unique solution. The Modelica compiler is very particular about this and performs non-trivial transformations to the system so that the flow of information and causality is not always obvious. This was one of the more tricky things to get an understanding of, especially when changing the controlled variables during simulation.

3.3 Control Modes

The choice of what variables to control defines what type of flight profiles our simulated aircraft will be capable of following. Looking at Table 1.1 most example mission segments involve controlling the airspeed and the altitude. To limit the amount of work 3 different control modes were created that could be used to simulate the most common flight profiles, other flight profiles were neglected. These are shown in Table 3.1. It must also be possible to switch between these control modes during a simulation.

Table 3.1 Implemented control modes. H is the altitude, v_{TAS} is the true airspeed and α is the angle of attack.

Control Mode	Controlled Variables	Useful for:
1	H and v_{TAS}	Ascent/descent
2	H and α	Constant altitude cruise type missions
3	α and v_{TAS}	Cruise climb type missions

3.4 Implementation

The overall implemented structure is similar to the general feedback linearization of Figure 2.3. Figure 3.5 shows the final controller configuration, where the rightmost block named "3-dof model" is the actual aircraft model. This structure that allows for any mission block along with switching and smooth transitioning between control modes is the main novel contribution of this thesis.

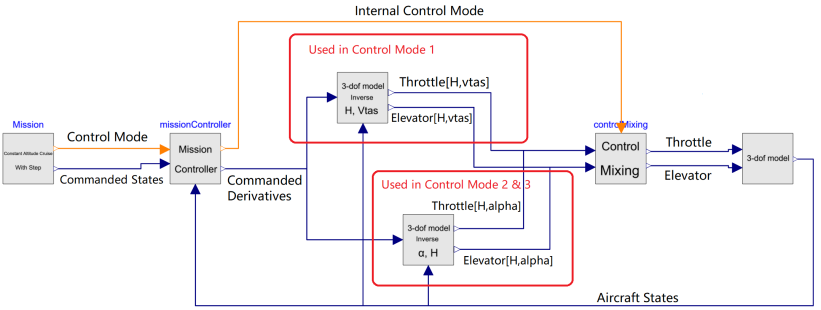


Figure 3.5 The final controller configuration. Orange arrows are integer signals and blue ones are real signals. The red boxes show the inverses and denote in which control mode they are used.

Starting from the left in Figure 3.5, the mission block contains the mission specification and outputs the integer "Control Mode" which can take three values (1, 2 or 3 corresponding to Table 3.1) specifying which control mode should be active. It also outputs a vector of commanded states for the mission. Both of these signals are fed into the "Mission Controller" block, which has a few purposes. Its main purpose is to perform the control of the linearized system, but also to do some filtering to make sure the commanded derivative signals are continuously differentiable. The "Mission Controller" block performs additional control in control mode 3 as well, which will be discussed further in the mission controller subsection. The mission controller outputs the commanded derivatives to the inverse models, which both continuously calculate the required throttle and elevator positions for their respective control mode. These are then all fed into the "Control Mixing" block, where the actual switching and selection of the inputs to the aircraft model occur, with some smoothing of the transitions that occur when the control mode changes. The throttle and elevator commands are then sent to the aircraft model and the aircraft states are fed back to the inverses and mission controller.

Note that there is no restriction on the throttle and elevator signals in the simulation model, so the mission controller has to be tuned to generate commands that

keep these signals reasonable. The mission specification also has to be reasonable for the parametrisation of the aircraft. If a simulation yields control signals out of their respective range, the current aircraft parametrisation must be deemed unable to perform the mission.

Mission Controller and Control Mode Switching

The mission controller block is used for mode switching and linear feedback of the controlled states. It also passes commanded state inputs through a first order filter. This makes the commanded trajectory continuous in the worst case but often even smoother, depending on the mission specification. This helps smooth out parts of the trajectory that are dynamic and prevents the inverses from requiring exceptionally large control signals. Figure 3.6 shows the Modelica code for the linear controller part of the "Mission Controller" block. $K[i]$ are the feedback gains and $D[i]$ are the dampening factors. Note that the dampening factors are not needed for a good result and shouldn't be needed for controlling a first order system, why they were left is discussed in Section 3.5.

```
//Derivative commands with dampening factors. (Most are 0)
 xdot_cmd[1] = K[1].*(alpha_cmd_filt - x[1])  - D[1]*der(x[1]); //dalpha
 xdot_cmd[2] = K[2].*(vt_cmd_filt - x[2]) - D[2]*der(x[2]); //dvt
 xdot_cmd[5] = K[5]*(H_cmd_filt-x[5]); //dH
```

Figure 3.6 Modelica code showing the linear controller part of the "Mission Controller" block. "xdot_cmd[]" are the commanced derivatives, alpha_cmd_filt is the filtered commanded state α (vt-v_{TAS} and H-H respectively) and x[] are the current actual states in the aircraft model.

This block is also used to control the third control mode (row 3 in Table 3.1), the reason for this is discussed in Section 5.5. The controller works by using the second control mode internally, meaning it is using the inverse block for control mode 2 (row 2 in Table 3.1). The "Internal Control Mode" signal sent to the control mixing block in Figure 3.5 is thus either 1 or 2 (1 for control mode 1 and 2 for control modes 2 and 3). Control mode 3 thus controls the altitude and angle of attack, but instead of controlling the altitude using a commanded value from the mission specification, it is using the following structure. For any given α the air pressure is inversely proportional to the true air speed if the lift should remain constant, shown in equations (2.1) and (2.2). Because the air pressure decreases monotonically with altitude, going up in altitude thus means increasing the true airspeed if α is held constant. It is therefore possible to control v_{TAS} using the altitude according to equation (3.8) where $\dot{H}_{\text{cmd}}$ is the commanded climb rate (sent through the "Commanded Derivatives" signal in Figure 3.5) and k is the feedback gain, a tunable parameter.

$$\dot{H}_{\text{cmd}} = k(v_{\text{TAS cmd}} - v_{\text{TAS}}) \tag{3.8}$$

When a mode switch is triggered, the mission controller switches the internal control mode. For control mode 1 and 2 the internal control mode is equivalent to the control mode. For control mode 3 the internal control mode is 2, meaning the Altitude and α inverse block is used (denoted "3-dof model inverse α, H" in figure 3.5). To make this mode switch happen as smoothly as possible and prevent spikes in the control signals, control mode 3 is phased in and out using a smooth step function described below.

Smooth Step. The smooth step function used is defined according to equation (3.9), where t is the time, t_{switch} is the time the switch starts, and t_{dur} is a tunable parameter that controls the switch duration.

$$S(t) = \begin{cases} 0 & t < t_{\text{switch}} \\ 3\left(\frac{t-t_{\text{switch}}}{t_{\text{dur}}}\right)^2 - 2\left(\frac{t-t_{\text{switch}}}{t_{\text{dur}}}\right)^3 & t_{\text{switch}} < t < t_{\text{switch}} + t_{\text{dur}} \\ 1 & t > t_{\text{switch}} + t_{\text{dur}} \end{cases} \tag{3.9}$$

By creating two scale factors using $S(t)$ in equation (3.9) where one is $S(t)$ and the other $1 - S(t)$, it is possible to smoothly transition the output of different control modes. Equation (3.10) shows how two signals from different control modes can be smoothly switched between using $S(t)$, here w_1 is the current signal and w_2 is the signal that should be switched to and w is the final signal.

$$w = (1 - S(t)) \cdot w_1 + S(t) \cdot w_2 \tag{3.10}$$

Figure 3.7 shows how the different control modes are smoothly transitioned in and out. Scale factor 1 is $S(t)$ and Scale factor 2 is $1 - S(t)$. Two switches occur at $t = 2$ s and $t = 20$ s with a switch duration $t_{\text{dur}} = 10$ s.

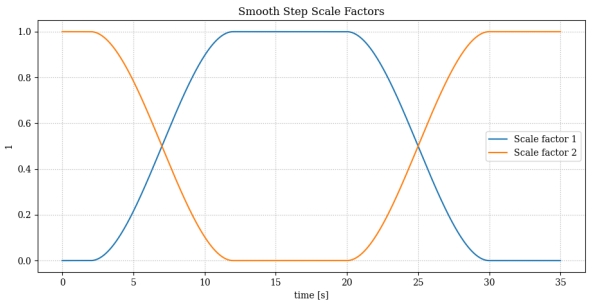


Figure 3.7 Example smooth step scale factors transitioning between two control modes.

The commanded climb rate $\dot{H}_{\text{cmd}}$ is switched using this function to switch to and from control mode 3 (row 3 in Table 3.1). H_{cmd} is then filtered and sent through

the proportional controller in Figure 3.6 which finally outputs xdot_cmd to the signal "Commanded Derivatives" in Figure 3.5. Figure 3.8 shows how this is done in Modelica code. Here smoothStep[1].Out is one of the scale factors using the smooth step function and smoothStep[2].Out is the other. The second term on the right hand side is for control mode 3 and can be identified to be the right hand side of equation (3.8) where x[2] is the true airspeed v_{TAS}, vt_cmd is the commanded true airspeed $v_{\text{TAS cmd}}$ and k_Hvt is the feedback gain (k in (3.8)). The first term on the right hand side is simply the derivative of the commanded altitude ("der(x_cmd[5])"), i.e the commanded climb rate from the mission specification.

```
//Control mode 3 switch
der(H_cmd) = smoothStep[1].Out*der(x_cmd[5]) + smoothStep[2].Out*k_Hvt*(vt_cmd-x[2]);
```

Figure 3.8 Modelica code for switching to and from control mode 3 by using the smooth step functionin the mission controller block. "smoothStep[1].Out" is one of the scale factors using the smooth step function and "smoothStep[2].Out" is the other. The second term on the right hand side correspond to control mode 3 and the first term correspond to the other control modes.

The smooth step function is also used in mission specifications to smoothly transition between two different commanded values, e.g. if the commanded true airspeed v_{TAS} should change from 150 to 160 in a smooth manner.

Control Mixing

The control mixing block is there to switch the internal control state, meaning it switches which inverse block controls the actual aircraft model. To make sure the control inputs (throttle and elevator) are continuous this is done using two first order filters (one for each input) that continuously switch between zero and one and thereby scale the two sets of control inputs to switch between them. Figure 3.9 shows the Modelica code for the control mixing block. Here "ctrlStateVec" is a vector that has the value 1 at the index of the current internal control state (see Figure 3.5), and 0 everywhere else. The scaling factors, one for each input, "smoothScale[1]" and "smoothScale[2]" are first order filtered versions of "ctrlStateVec" to continuously switch which input is active when a discrete change in "ctrlStateVec" occurs. "k_filt" is a tunable parameter controlling the speed of the switch. "elevInput[1]" and "THTLInput[1]" are from the control mode 1 inverse ("Throttle[H,vtas]" and "Elevator[H,vtas]" in Figure 3.5) and "elevInput[2]" and "THTLInput[2]" are from the other inverse ("Throttle[H,alpha]" and "Elevator[H,alpha]" in Figure 3.5). "elevOutput" and "THTLOutput" are the control signals sent to the aircraft model ("Throttle" and "Elevator" in Figure 3.5).

```
der(smoothScale) = k_filt.*(ctrlStateVec-smoothScale);

elevOutput = smoothScale[1]*elevInput[1] + smoothScale[2]*elevInput[2];
THTLOutput = smoothScale[1]*THTLInput[1] + smoothScale[2]*THTLInput[2];

algorithm
ctrlStateVec := zeros(2);
ctrlStateVec[ControlState] := 1;
```

Figure 3.9 Modelica code for switching the internal control mode (i.e. which inverse in Figure 3.5 is used). "ctrlStateVec" has the value 1 at the index of the current internal control state (see Figure 3.5), and 0 everywhere else. The scaling factors "smoothScale[1]" and "smoothScale[2]" are filtered versions of "ctrlStateVec" to continuously switch which input is active. "k_filt" is a tunable parameter controlling the speed of the switch. "elevInput[1]" and "THTLInput[1]" are from one inverse ("Throttle[H,vtas]" and "Elevator[H,vtas]" in Figure 3.5) and "elevInput[2]" and "THTLInput[2]" are from the other inverse ("Throttle[H,alpha]" and "Elevator[H,alpha]" in Figure 3.5). "elevOutput" and "THTLOutput" are the control signals sent to the aircraft model ("Throttle" and "Elevator" in Figure 3.5).

Angle of Attack Approximation

The inverse blocks, denoted "3-dof model inverse" in Figure 3.5, are originally copies of the model itself but with the relevant derivative statements exchanged for prescribed values and outputs redefined as mentioned above. There are however some changes made to make the simulation run better. For the control mode 1 inverse, controlling the altitude H and true airspeed v_{TAS}, the simulation is incredibly sensitive towards the start values of the model, and even if the model starts simulating, it is often incredibly slow without lowering the simulation tolerance a lot. This is discussed further in the discussion chapter.

A fix for this problem was to not feed the α state directly from the aircraft model but to calculate it heuristically in each inverse based on a steady-state level-flight approximation. The coefficient of lift is calculated according to equation (3.11) based on the simplification shown in Figure 2.2 and the lift equation, equation (2.2). The main assumption here is that the lift is equal to the aircraft weight. This assumes level flight, $\gamma = \theta = 0$, and steady state, meaning the aircraft is not accelerating in any direction.

$$C_L = \frac{L}{A\bar{q}} \approx \frac{mg}{A\bar{q}} \tag{3.11}$$

This C_L is then fed back into the inverse model to calculate an approximation of α. This makes the simulation run fast and with low tolerance. Other effects of the approximation are discussed in Section 5.3.

3.5 Linear Controller Design

The linear feedback controller controls most variables using simple proportional controllers, located in the mission controller block (see Figure 3.5). The Modelica code is shown in Figure 3.6. Normally dampening factors (derivative terms, D[1] and D[2] in Figure 3.6) makes no sense in first order systems, but because of the AoA approximation the system is not completely linearized and dampening factors on α and v_{TAS} where found to smoothen the dynamic sections marginally. Control mode 3 is a special case that should be calibrated to the atmosphere model (the one used in this thesis is shown in Section 3.1). This is discussed further in Section 5.5.

Table 3.2 shows the controller parameters used. "H (Mode 1 & 2)" is for controlling the altitude directly (control mode 1 and 2) and "H (Mode 3)" is for controlling v_{TAS} using the altitude (control mode 3). These parameters are not optimal in any way, they were found by experimentation but they seem to work well and not generate too large control signals. Too small feedback gains will lead to a slow system, meaning the aircraft will only very slowly approach the commanded values. Too large gains will cause unreasonably large control signals, e.g. a throttle above 1 or elevator deflections above 30°. Small dampening factors will have no effect and large ones will prevent rapid changes in the variable, making the system slower.

Table 3.2 The linear controller parameters used. α and v_{TAS} uses the same feedback gains and dampening factors for all modes were they are controlled. In control mode 3 the altitude H is controlled using the true airspeed error instead of the altitude error, these have different feedback gains. The rightmost column shows where these gains and dampening factors are used in the code.

Variable	Feedback gain	Dampening	in Figure
α	1	5	3.6 as K[1] and D[1]
v_{TAS}	0.1	5	3.6 as K[2] and D[2]
H (Mode 1 & 2)	0.05	-	3.6 as K[5]
H (Mode 3)	1	-	3.8 as k_Hvt

3.6 Missions

Two example missions are presented below. These were fed into the mission controller block to evaluate the performance of the method. The missions are designed to be somewhat realistic representations of real flights, but much shorter to efficiently illustrate what happens when maneuvers are performed and control modes are switched.

Constant Altitude Cruise

This flight profile is based on the constant altitude cruise flight strategy presented in the theory section, it also contains segments of ascent, altitude change and descent. Figures 3.10 and 3.11 show the signals sent to the mission controller block. The blue parts indicate that the plotted variable is actively controlled during that time and the orange parts mean the variable is not controlled and the controller will ignore the value of the variable at these times. This means that when a graph switches between blue and orange, a control mode switch occurs. Figure 3.11 shows that the control mode is switched four times between controlling the altitude together with true airspeed and controlling the altitude with the angle of attack.

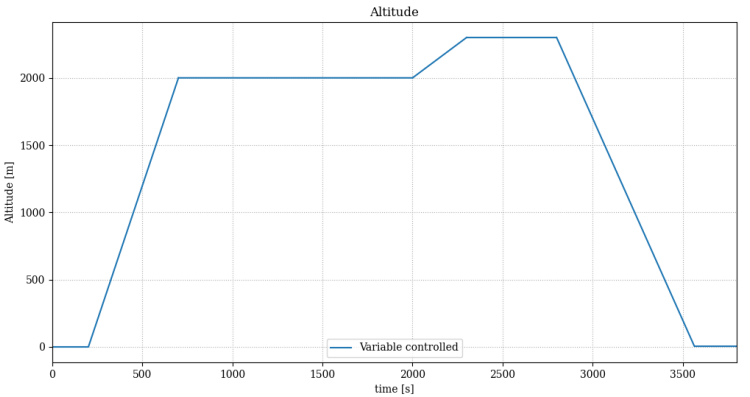
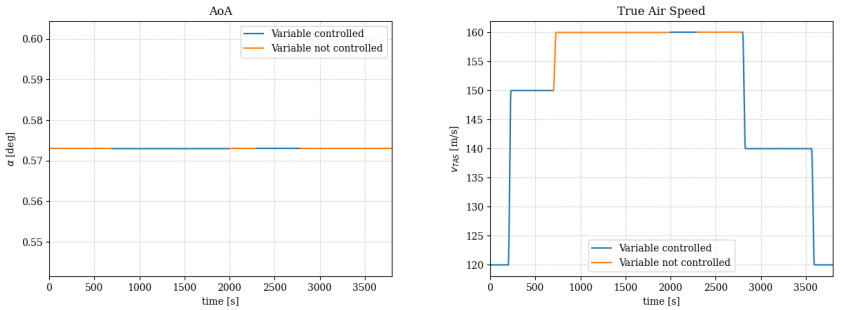


Figure 3.10 Commanded altitude with respect to time sent to the mission controller block. This is the output of the mission block (see figure 3.5) and not the result of a flight simulation. The resulting simulation with this input can be seen in Section 4.1.



(a) **(b)**

Figure 3.11 Commanded angle of attack and true airspeed with respect to time sent to the mission controller block. The blue segments indicate that the variable is actively controlled during that time and the orange segments mean the variable is not controlled. This means that when a graph switches color between blue and orange, a control mode switch occurs. This is the output of the mission block (see figure 3.5) and not the result of a flight simulation. The resulting simulation with this input can be seen in Section 4.1.

Cruise Climb

This cruise climb flight profile is similarly based on the cruise climb flight strategy presented in the theory section, it also contains segments of ascent and descent. Figures 3.12 and 3.13 show the signals sent to the mission controller block ("Commanded States" in Figure 3.5), note that this time the climb rate is specified instead of the altitude. Because the final altitude of the cruise climb is unknown it is undesirable to command a specific altitude immediately since this might lead to large instantaneous change in command signal. This can of course be mitigated by smoothing the transitions a lot. Once again the colors determine what is being controlled, this time only two switches are performed between controlling the altitude and true airspeed and controlling the angle of attack and true airspeed.

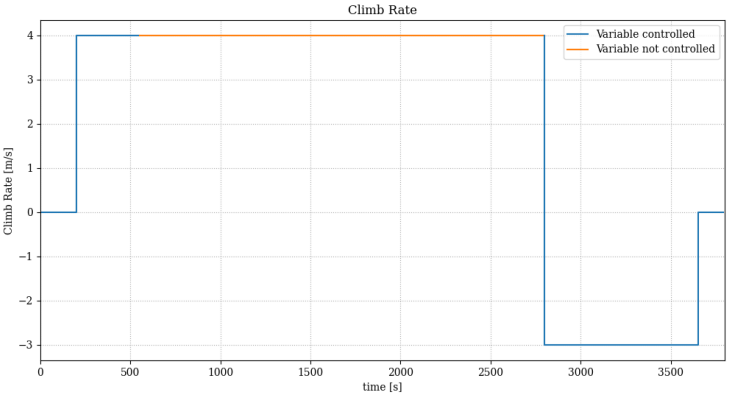
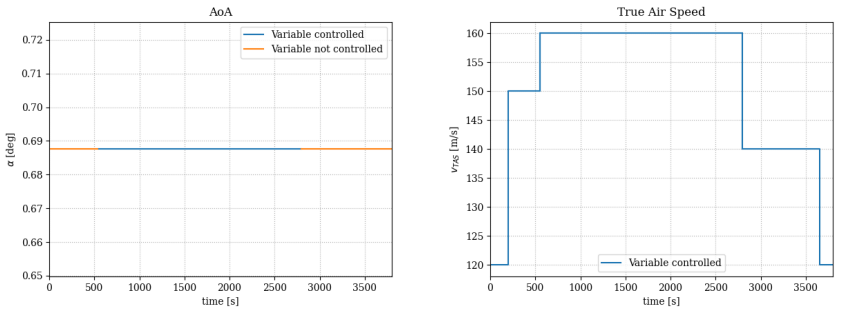


Figure 3.12 Commanded climb rate with respect to time sent to the mission controller block. The blue segments indicate that the variable is actively controlled during that time and the orange segments mean the variable is not controlled. This means that when a graph switches color between blue and orange, a control mode switch occurs. This is the output of the mission block (see figure 3.5) and not the result of a flight simulation. The resulting simulation with this input can be seen in Section 4.2.



(a) **(b)**

Figure 3.13 Commanded angle of attack and commanded true airspeed with respect to time sent to the mission controller block. The blue segments indicate that the variable is actively controlled during that time and the orange segments mean the variable is not controlled. This means that when a graph switches color between blue and orange, a control mode switch occurs. This is the output of the mission block (see figure 3.5) and not the result of a flight simulation. The resulting simulation with this input can be seen in Section 4.2.

3.7 State Event Based Missions

A state event based mission means that the mission profile does not need to be completely determined in terms of time but rather based on aircraft states. Mission segment switching can then be performed based on statements like "Climb with specified climb rate until reaching specified altitude". This is a more natural way of specifying a mission and will greatly simplify the production of missions since the need for translation to a time based form is eliminated.

Some experiments were performed using state event based missions and yielded promising results, but unfortunately the structure of the mission block needs to be improved in order to properly support them.

4

Results

Below the resulting simulations for the two missions are presented. For the variables that were sometimes controlled, a blue color indicates that the variable was controlled during that time, while an orange color indicates that it was not controlled. The green dashed lines show the commanded values of the variables from the mission controller. Note that the fuel burn rate has been greatly exaggerated in both missions to see the effect of the decreased mass.

4.1 Constant Altitude Cruise

Figure 4.1 shows the simulated altitude H with respect to time for the simulated constant altitude cruise mission presented in Section 3.6. It follows the commanded value well. In the parts of the command signal that are ramps the simulated altitude has a stationary error, as is to be expected when controlling the derivative based on the error only.

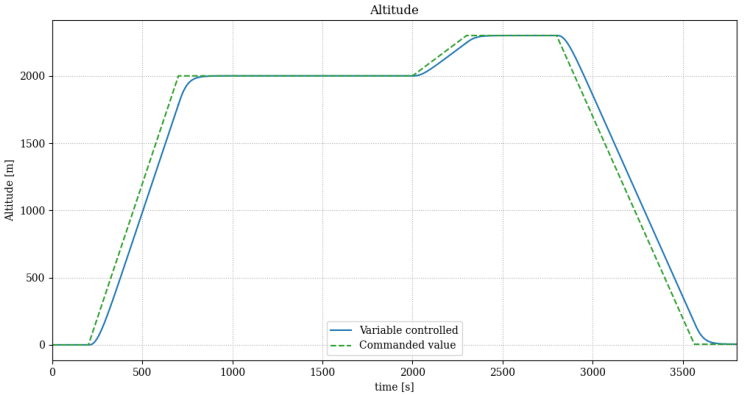
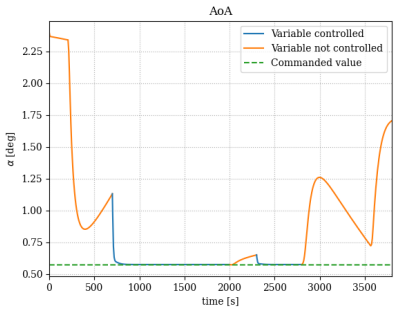


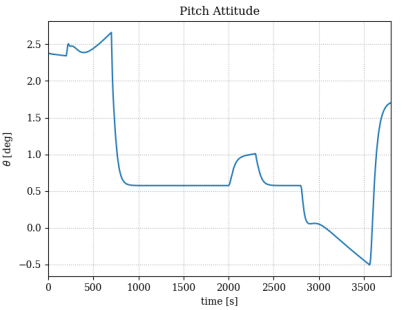
Figure 4.1 Simulated altitude with respect to time for the constant altitude cruise mission. The green dashed line is the the commanded altitude from the mission specification and the blue line is resulting trajectory from he simulation.

Figure 4.2 shows the simulated angle of attack α and pitch attitude θ. The angle of attack is a controlled variable at some time intervals in the simulation, this is shown in Figure 4.2a with a blue line for when the variable was controlled and an orange line for when the variable was not controlled. In the latter case the simulated variable is merely a result of other controlled variables. The pitch attitude is never controlled so the graph is always a result of other controlled variables.

The angle of attack in figure 4.2a stays within a reasonable range and quickly goes to the commanded value when the control mode is switched to control it. The pitch attitude also stays within a reasonable range.



(a) The blue line indicates that the angle of attack was actively controlled, an orange line indicates it was not actively controlled at that time. The green dashed line shows the commanded angle of attack.

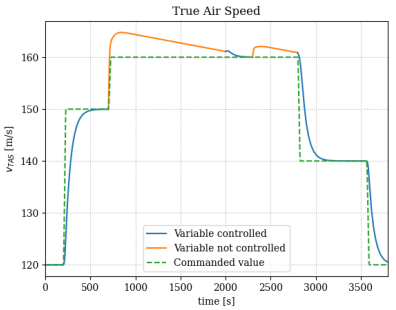


(b) The pitch attitude is never controlled and merely a result of control of other variables.

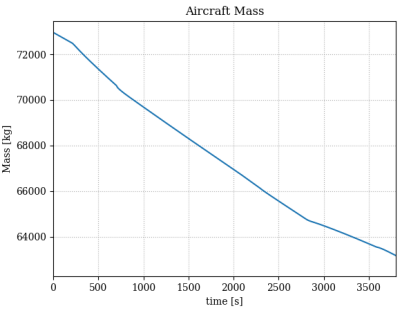
Figure 4.2 Simulated angle of attack and pitch attitude with respect to time for the constant altitude cruise mission.

Figure 4.3 shows the simulated true airspeed v_{TAS} and the simulated aircraft mass m. The true airspeed is a controlled variable at some time intervals in the simulation, this is shown in Figure 4.3a with a blue line for when the variable was controlled and an orange line for when the variable was not controlled. In the latter case the simulated variable is merely a result of other controlled variables. The aircraft mass is never controlled so the graph is always a result of other controlled variables.

The effect of decreasing mass (decrease in required true airspeed) is clearly shown in the parts of Figure 4.3a with an orange line, meaning the aircraft was in control mode 2. As the mass decreases the true airspeed decreases as well to lower the lift, thereby keeping the altitude constant. Note that the fuel burn rate is heavily exaggerated to better see the effect of the decreasing mass.



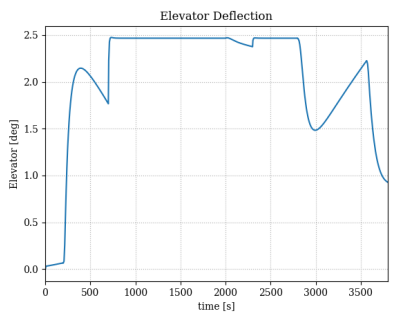
(a) The blue line indicates that the true airspeed was actively controlled, an orange line indicates it was not actively controlled at that time. The green dashed line shows the commanded true airspeed.



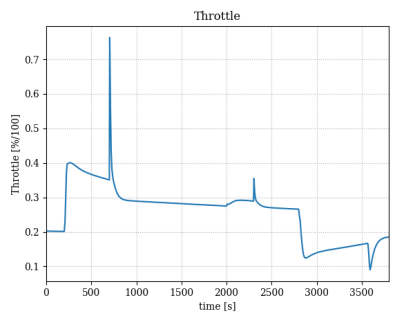
(b) The aircraft mass is never controlled and merely a result of control of other variables.

Figure 4.3 Simulated true airspeed and aircraft mass with respect to time.

Figure 4.4 shows the simulated control inputs to the aircraft model. These are both results of the feedback linearization control scheme. These are within reasonable ranges but the throttle does have a large spike at around 600 seconds, the reason for this is discussed further in the discussion section.



(a)



(b)

Figure 4.4 Simulated control inputs with respect to time.

4.2 Cruise Climb

Figure 4.5 shows the simulated altitude H with respect to time for the simulated cruise climb mission presented in Section 3.6. Here the effect of decreasing mass is

clearly shown as an increase in altitude when in control mode 3, in this case when the line is orange.

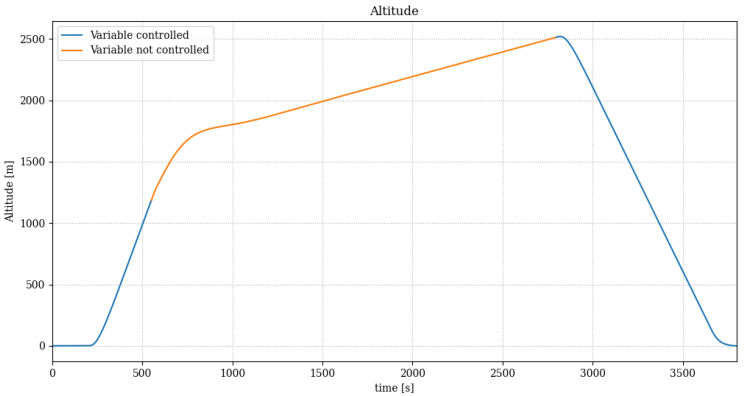


Figure 4.5 Simulated altitude with respect to time for the constant altitude cruise mission. The green dashed line is the the commanded altitude from the mission specification and the blue line is resulting trajectory from he simulation.

Since the cruise climb mission is controlled in terms of climb rate instead of altitude Figure 4.6 shows the simulated climb rate. Once again the dashed green line shows the commanded climb rate, the blue line shows the climb rate when it is actively controlled and the orange line shows the climb rate when it is not controlled. The commanded value is followed well when the variable is controlled.

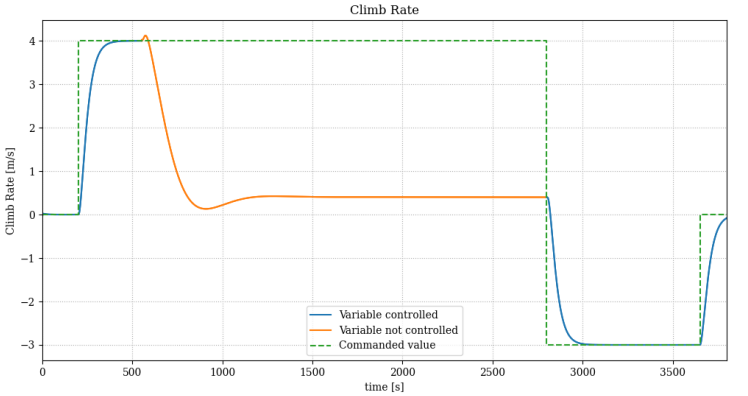
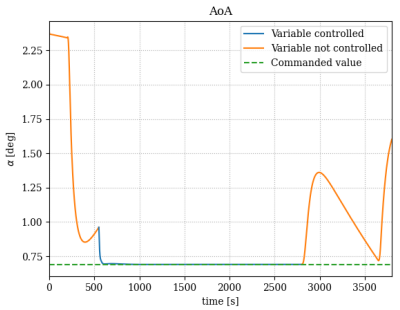


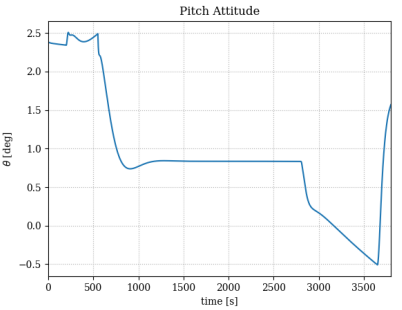
Figure 4.6 Simulated climb rate with respect to time. The green dashed line is the the commanded climb rate from the mission specification and the blue and orange lines show the resulting trajectory from he simulation. When the line is blue the climb rate was actively controlled and when the line is orange the line is merely a result of other controlled variables.

Figure 4.7 shows the simulated angle of attack α and pitch attitude θ. The angle of attack is again a controlled variable at some time intervals in the simulation, this is shown in Figure 4.7a with a blue line for when the variable was controlled and an orange line for when the variable was not controlled. In the latter case the simulated variable is merely a result of other controlled variables. The pitch attitude is never controlled so the graph is always a result of other controlled variables.

The angle of attack in figure 4.7a stays within a reasonable range for this mission as well and quickly goes to the commanded value when the control mode is switched to control it. The pitch attitude also stays within a reasonable range.



(a) The blue line indicates that the angle of attack was actively controlled, an orange line indicates it was not actively controlled at that time. The green dashed line shows the commanded angle of attack.

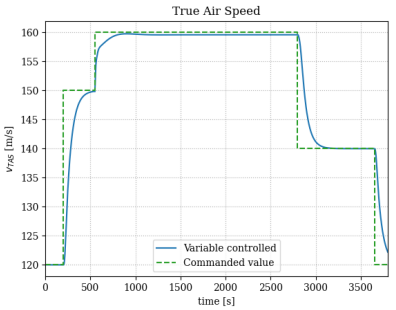


(b) The pitch attitude is never controlled and merely a result of control of other variables.

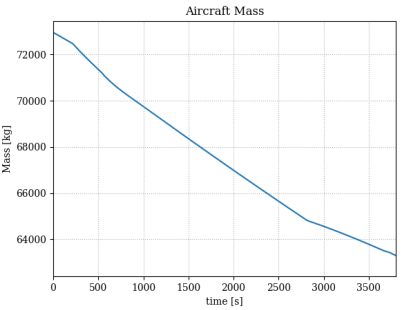
Figure 4.7 Simulated angle of attack and true airspeed with respect to time.

Figure 4.8 shows the simulated true airspeed v_{TAS} and the simulated aircraft mass m. For this mission the true airspeed is always a controlled variable and is shown in Figure 4.8a with a blue line. The aircraft mass is never controlled so the graph is always a result of other controlled variables.

The true airspeed follows the commanded value quite well but there is a stationary error during the part of the mission that uses control mode 3, in Figure 4.8a when the commanded value is 160 m/s. This is a result of the special nature of control mode 3 and is discussed further in Section 5.2.



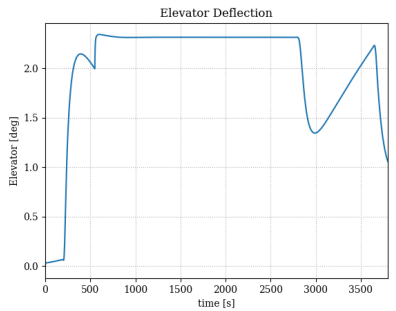
(a) The blue line indicates that the true airspeed was actively controlled, an orange line indicates it was not actively controlled at that time. The green dashed line shows the commanded true airspeed.



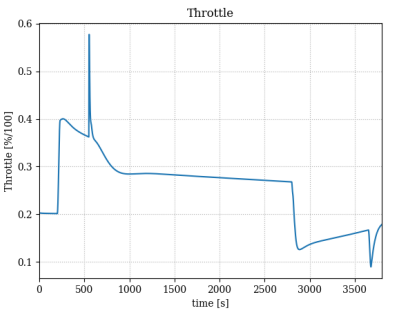
(b) The aircraft mass is never controlled and merely a result of control of other variables. Note that the fuel burn rate is greatly exaggerated to show the effects of decreasing mass better.

Figure 4.8 Simulated true airspeed and aircraft mass with respect to time.

Figure 4.9 shows the simulated control inputs to the aircraft model. These are both results of the feedback linearization control scheme. For this mission both the elevator and the throttle are within reasonable ranges. Once again the throttle has a large spike at around 600 seconds however, this occurs for the same reason as for the constant altitude cruise mission and is discussed further in Section 5.1.



(a)

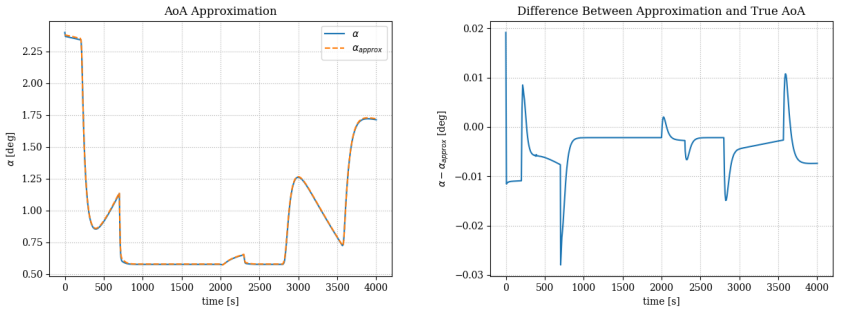


(b)

Figure 4.9 Simulated control inputs with respect to time.

4.3 Angle of Attack Approximation

Figure 4.10 shows both α and α_{approx} and the difference between the true angle of attack and the approximated one during the constant altitude cruise mission. As can be seen, the approximation is excellent.



(a) α approximation in inverse model and true value.

(b) Difference between α approximation and true value.

Figure 4.10 Angle of attack approximation during constant altitude cruise mission.

5

Discussion

5.1 Constant Altitude Cruise

Figures 4.1 to 4.4 in the results chapter show that the commanded signals are followed well. The ramp signal in Figure 4.1 is followed as one would expect from a first order system: with a slight time delay. Figure 4.2 shows the angle of attack command is followed well when it is controlled. Some parts of both the AoA α and the pitch attitude θ graph almost look like they have discontinuous derivatives. This is simply because of the time scale, zooming in they are clearly smoother and the change happens over several seconds.

The true airspeed shown in Figure 4.3 is also followed well when it is commanded. The more interesting thing in this Figure is that the effect of the decreasing aircraft mass is clearly visible in the portion of the mission where the angle of attack α is controlled. It is clearly seen that the true airspeed is decreasing with decreasing mass, and when the altitude goes up again the air speed momentarily goes up to then continue decreasing. This is exactly what is expected since a decreased aircraft mass means that a lower air speed is needed to create a lift force that balances gravity.

The control signals in Figure 4.4 are both always within reasonable ranges, there is however a spike in the throttle signal at around 600 seconds. Once again it looks almost instantaneous but is actually continuous and within a couple of seconds, which could be reasonable depending on the engine used [4]. The spike occurs because the aircraft is at too low speed to quickly go to the lower α while maintaining altitude. There are a few ways to mitigate this. One way is to lower the feedback gain for α, which will help by not requiring the aircraft to reach the commanded α as quickly, this will however change the behaviour everywhere α is controlled. The other way, probably better in this case, is to change the commanded signal for v_{TAS} so that it is closer to the natural speed when flying at that altitude and angle of attack. An alternative mitigation strategy is to phase in the command for alpha slower.

5.2 Cruise Climb

Controlling α and v_{TAS} (i.e. control mode 3) also works well with tuned parameters for the linearized control, but it clearly is not as simple as the other control modes since the dynamics of the atmosphere are not inverted because v_{TAS} is controlled using the altitude H (see equation (3.8)). This is why the slow, slightly oscillatory curve occurs in Figure 4.5 before settling in to a linear climb. Here the effect of the decreasing mass (Figure 4.8b) is clearly shown as the altitude is rising as expected.

One important note from Figure 4.8 is that the commanded value for the true airspeed is never reached. This is because the mass of the aircraft is constantly decreasing which effectively yields a ramp signal in the commanded altitude, and since we are controlling the altitude without any integral action there will be a stationary error. This can easily be taken care of by adding integral feedback to the controller.

There is a spike in the throttle signal for this mission as well, shown in Figure 4.9. This happens for the same reason as in the previous mission and can be mitigated in the same way.

5.3 AoA Approximation

Figure 4.10 shows how the α approximation performs in the constant altitude cruise mission. It is clearly very close to the true alpha for this mission, which makes sense since the aircraft is always quite close to the assumptions made in the approximation ($\gamma = 0$, $\theta = 0$, steady state). For e.g. a fighter aircraft this approximation would be far off and not viable at all depending on the mission. The difference between the approximated and the true α shows that during dynamic events the approximation gets worse, which of course is expected. When steady state is achieved though the difference between the true α and the approximated one is not zero. This is because even though the flight path angle $\gamma = 0$, the aircraft is not flying at a pitch angle $\theta = 0$, which means a portion of the thrust from the engine(s) will be pointing upwards and therefore lowering the required lift to stay in steady state, thereby overestimating the angle of attack α. No analysis was made as to when this approximation breaks down but climb rates >50 m/s were achievable without any issues.

5.4 Control Signals

As mentioned in the method chapter the control signals, the throttle and elevator, are not restricted in the simulation to their respective practically achievable ranges. This means that the inverse models could output a negative throttle signal, which

would yield a negative thrust from the engine(s), or elevator angles of more than 90 degrees. This is highly non-physical and any simulation were this happens must be deemed invalid. If the aircraft only momentarily experiences unreasonable control signals when switching control modes, the smoothing parameters (e.g. t_{dur} of the smooth step function in equation (3.9)) of the controller can be adjusted to achieve slower transitions to prevent this from happening. If however this happens when a command signal from the mission block changes but the control mode is not switched, in a suitable fashion, the mission must be adjusted with slower change of the command signals (e.g. by passing it through the smooth step function with a high t_{dur}). Alternatively the feedback gains of the linear controller can be adjusted. I would advise against this though since it could make other parts of the mission unnecessarily slow.

5.5 Control Mode 3

Originally the third control mode was supposed to have its own inverse block as well, but after some experimenting this block was found to yield undesirable trajectories. Huge oscillations and even loops occurred if the aircraft was not initialized close to the perfect state for cruise climb. This is an example of unstable zero dynamics, when looping the pitch attitude θ will continue increasing forever but this is not visible in the controlled variables. In this case this is not that big of a problem though, since all equations containing θ has the bounded functions $\sin$ or $\cos$ applied to it. To get around this problem control mode 2 is used instead, meaning the altitude and α are controlled. Because the density of the air decreases monotonically when increasing in altitude, it is possible use feedback of v_{TAS} to control the altitude. In simple terms, if the true air speed is too low increase the altitude and vice versa. This yields much nicer trajectories and seems like a more natural way to express what is actually desired.

This control mode is not entirely independent of the aircraft model however, as the atmosphere model will impact the dynamics of controlling the altitude using v_{TAS}. In the intended use case one would probably use the same atmosphere model for all aircraft models to yield a fair comparison which mitigates the problem this poses. This controller only needs to be tuned once per atmosphere model.

5.6 Multiple Inverses Model Structure

There is a quite unintuative behaviour of Modelica when using multiple inverses. Both inverse blocks in Figure 3.5 control the altitude, but even if they are never active the same time the Modelica compiler throws an error and refuses to compile. My guess is that this is because they are both connected through the aircraft model, even though both are receiving the states as inputs, and the compiler thinks we now

have two redundant equations. The way to get around this is to only connect the commanded derivative for the altitude to one of the blocks. The "information" will then travel through the aircraft model to the other inverse block and that block will calculate the correct control inputs. Note that in reality speaking about information traveling like this is not representative of what actually happens. The Modelica compiler will perform non-trivial transformations to the system so that it cannot be viewed in terms of connectors and blocks. One can see however that some variables appear in equations related to both inverses confirming that they are indeed connected after the transformations.

5.7 Inverse Blocks Without AoA Approximation

As mentioned in the method chapter the control mode 1 inverse without the angle of attack approximation would not even start simulating saying that the initialization problem is unsolvable and showing the residuals of the problem. Changing the start values of some variables had a significant impact on the solution found with the lowest residuals though, and I was able to improve the residuals by a factor of ≈ 50. Note that the start values in Modelica are different from initial values, as start values only tell the solver where to start looking for a solution to the problem. This led me to speculate that there could actually be a proper solution, but the initial problem is ill posed with lots of local minima or weak gradients for the solver to get stuck in. Tweaking the start values a bit and increasing the tolerance a lot, tol $= 10$ instead of 10^{-6}, the model was able to simulate and most simulation results looked very good. Almost identical to what the AoA approximation generated, except the pitch rate Q was continuously 0, indicating that the simulation is not valid in any way. By tweaking the start values further I was able to simulate the model with a tolerance of 1 but the simulation was exceptionally slow.

These findings led me to believe that perfect start values and/or some freedom to allow the the inverse to not be subject to exactly the same equations as the aircraft model (or have equivalent states) is needed for the model to simulate efficiently. Whether it is by approximating the angle of attack or by increasing the tolerance. This raises the question if there is a way of introducing this slack in a controlled way without limiting the method to only work close to some approximated value. An alternative is to find a way to modify the method to get around the problem entirely.

5.8 Future Work

Below some suggestions for future work are presented.

Implement State Event Segment Switching

By implementing state event segment switching the ease of use of the method and especially mission specification will improve greatly.

Try Different Aircraft Models

It would be interesting to see how well the method works with other aircraft models. For example models with higher order Taylor approximations or lookup tables for the aerodynamic coefficients. Since differentiability is required, the lookup tables would need to use some interpolation method facilitating automatic differentiation. Another idea is to use the inverse of a lower fidelity model, such as the one presented in this work, to control more advanced models. If the parameters of the lower fidelity model match the advanced model, the inverse should work well locally. If such more advanced models work well the method could be investigated for 6 degree of freedom models as well.

Investigate Inverses of Ineffective Controlled Variables

As mentioned in Section 5.7 it seems some freedom is needed for the inverse model not to be exact for the method to simulate efficiently for some controlled variable combinations. To be able to control an aircraft where e.g. the angle of attack approximation does not hold this needs to be investigated. Hopefully a way to handle this with some guarantee on how accurate the simulation is can be found, or a way to get around the problem completely.

5.9 Conclusion

The method works well for its intended purpose with some important limitations. With the current state of the method only missions where the angle of attack approximation holds can be flown. However, the parametrisation of the aircraft is general and the method should work with a wide range of parametrisations, i.e. different aircraft. With some work it could provide an easy to use way to simulate many different parametrisations and compare them. More accurate simulations obviously need to be performed in later stages of aircraft development.

Bibliography

[1] R. Höpler and M. Hardt. “Generating aircraft trajectories encoded with the aircraft intent description language using the modeling language Modelica”. In: *5th CEAS Conference on Guidance, Navigation and Control*. 2019.

[2] F. Re. “Automatic control generation for aircraft taxi systems through nonlinear dynamic inversion of object-oriented model” (2013).

[3] B. Stevens, F. Lewis, and E. Johnson. *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. Wiley, 2015. ISBN: 9781118870983.

[4] S. Gudmundsson. “Chapter 7 - selecting the power plant”. In: S. Gudmundsson (Ed.). *General Aviation Aircraft Design*. Butterworth-Heinemann, 2014, pp. 181–234. ISBN: 978-0-12-397308-5. DOI: https://doi.org/10.1016/B978-0-12-397308-5.00007-6.

[5] D. Enns, D. Bugajski, R. Hendrick, and G. Stein. “Dynamic inversion: an evolving methodology for flight control design”. In: *International Journal of Control 59, no. 1*. 1994, pp. 71–91.

[6] T. Glad and L. Ljung. *Control Theory*. Control Engineering. Taylor & Francis, 2000. ISBN: 9780748408788.

[7] M. Henson and D. Seborg. *Nonlinear Process Control*. Prentice Hall PTR, 1997. ISBN: 9780136251798.

[8] T. Fossen and B. Foss. “Sliding control of mimo nonlinear systems”. *Modeling, Identification and Control: A Norwegian Research Bulletin* **12** (1995). DOI: 10.4173/mic.1991.3.3.

[9] A. Filippone. *Advanced Aircraft Flight Performance*. Cambridge Aerospace Series. Cambridge University Press, 2012. DOI: 10.1017/CBO9781139161893.

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER'S THESIS

Date of issue
September 2023

Document Number
TFRT-6219

Author(s)
Fredrik Horn

Supervisor
Bo Bernhardsson, Dept. of Automatic Control, Lund University, Sweden
Yiannis Karayinidis, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle
Feedback Linearization for Model Agnostic Aircraft Simulation Control

Abstract

This thesis presents a method, based on feedback linearization, for controlling a general 3-degree-of-freedom simulation model of an aircraft. Only aircraft models that utilize Taylor approximations of the aerodynamic coefficients are considered. The intent is to be able to simulate the same mission with different aircraft models without having to tune the control system specifically for each model.

The method is based on feedback linearization and generates inverse models from the original aircraft model. These are then used to control different variables using a linear controller. Special care was taken to be able to switch the controlled variables during a flight and smooth out the transitions.

Two test missions were successfully implemented based on the constant altitude cruise and cruise climb flight strategies respectively. After some tuning of the linearized controller, the desired trajectories were followed well. However, additional tuning and analysis of the linear controller could be performed to ensure more specific performance targets are met. Some work should also be done to make the method easier to use with regards to the mission specification the user provides. A better mission segment handler and state event based segments would reduce the requirements on the mission specification and thus make it easier to define missions that work for multiple aircraft.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-46

Recipient's notes

Security classification

http://www.control.lth.se/publications/