Signomial Programming for Aircraft Design

by

Philippe G. Kirschen

B.S.E., University of Michigan (2014)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Author ....
Department of Aeronautics and Astronautics
May 20, 2016

Certified by..
Warren W. Hoburg
Assistant Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by ....
Paulo C. Lozano
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Signomial Programming for Aircraft Design

by

Philippe G. Kirschen

Submitted to the Department of Aeronautics and Astronautics
on May 20, 2016, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Due to the coupled nature of aircraft system design, it is important to consider all major subsystems when optimizing a configuration. This, however, is easier said than done, particularly because each individual subsystem model can be arbitrarily complex. By restricting an optimization problem to have a certain mathematical structure, significantly more effective and tractable solution techniques can be used. Geometric programming, an example of one such technique, guarantees finding a globally optimal solution. Although it has been shown that geometric programming can be used to solve some conceptual aircraft design problems, the required formulation can prove too restrictive for certain relationships. Signomial programming is a closely related relaxation of geometric programming that offers enhanced expressiveness, but without the guarantee of global optimality. Despite this, solution methods for signomial programs are disciplined and effective. In the present work, signomial programming models are proposed for optimal preliminary sizing of the vertical tail, horizontal tail, fuselage, landing gear, and wing of a commercial aircraft with a tube-and-wing configuration. These models are then combined together to produce a full aircraft optimization model. Signomial programming's relaxed formulation allows it to handle some of the key constraints in tail, fuselage, landing gear, and wing design, and therefore an improvement in fidelity over geometric programming models is achieved. The models are readily extensible and easily combined with other models, making them effective building blocks for future work. A primary contribution of this work is to demonstrate signomial programming as a viable tool for multidisciplinary aircraft design optimization.

Thesis Supervisor: Warren W. Hoburg
Title: Assistant Professor of Aeronautics and Astronautics

Acknowledgments

First and foremost I would like to thank my advisor, Woody Hoburg, for the privilege of being his student, and for giving me the opportunity to work on an area of research that I find genuinely exciting. His exceptional intellect, as well as his sense of humour and kindness have made the past two years an overwhelmingly positive experience. I pity anyone who has not had the opportunity to work with him.

I am indebted to Ned Burnell for the amazing work he has done as the lead developer of GPkit. This thesis would simply not have been possible without Ned's diligence and incredible talent. His patience and willingness to help have made him a pleasure to work with.

The idea of coming to graduate school at MIT conjures images of socially secluded students with no sense of camaraderie. Based on my experience in the Aerospace Computational Design Lab (ACDL), this could not be further from the truth. I am grateful to the members of DogeCube (Giulia Pantalone, Carlee Wagner, and Hugh Carson) for the memes and the laughs, to the members of HyperCube (Max Opgenoord, Cory Frontin, and again, Hugh Carson) for the banter, and also to Patrick Blonigan, Ferran Vidal Codina, and Philip Caplan, who ensure the lab is a fun and supportive environment.

My second year at MIT was transformed by being a part of the MIT Hyperloop team. The quality of banter on the team was matched only by the diligence and seemingly endless knowledge of its members. It has been a wonderful, albeit humbling, experience to be part of such a team.

I am grateful to the members of the prestigious AeroAstro intramural ice hockey team for teaching me a new sport and giving me something to look forward to during cold winter months.

I spent the summer between my first and second year in the Bay Area, working at NASA Ames. Mike Aftosmis took me under his wing for a summer internship and challenged me continuously, an experience that I am undoubtedly better for. Parker Vascik was an outstanding road trip companion whose limitless optimism enhances the lives of those around him.

Outside of MIT, I am grateful to Kirsten Bray, for her infallible companionship and for a never ending stream of cute animal videos to brighten even the darkest winter days, and to Bharat Srinivasan, for his unique sense of humour and his advice.

Before coming to MIT, I had the fortune of being mentored by Prof. Krzysztof Fidkowski and Prof. Joaquim Martins, of the University of Michigan, as well as by Nick Nagel, of Triumph Aerospace Systems, and William Chan, of NASA Ames. Their advice and support helped make living the MIT dream possible.

Finally, though it is a cliché to say that I would be nothing without the unconditional love and support of my parents, it does not make it any less true. Their belief in me has never faltered, even when mine has. Among many other things, I am especially grateful to them for teaching me the importance of writing well. I forgive them for strongly encouraging me to go to Stanford.

This work was funded by the NASA Aeronautics Fellowship as well as by the Aurora Flight Sciences corporation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Geometric Programming (GP)[1] is an optimization technique that combines the expressiveness of non-linear objectives and constraints with the mathematical rigor of convex optimization to provide a powerful approach to solving multidisciplinary aircraft design optimization problems. For problems that can be formulated as Geometric Programs (GPs), modern solvers guarantee *globally* optimal solutions, are extremely fast, and return local sensitivities at no extra cost, thanks to the principle of lagrange duality. In previous work, Hoburg [21] shows, firstly, that many models common to aircraft design can be represented directly in GP-compatible form, and, secondly, that there are a number of innovative ways of dealing with models that cannot, including, but not limited to, changes of variables and GP-compatible fitting methods. Finally, it is also shown that such problems can be solved efficiently using a standard laptop computer. The aircraft design problem solved in [21] includes models for steady level flight, range, takeoff distance, landing speed, a sprint flight condition, actuator disk propulsive efficiency, simple drag and weight buildups, and a beam wing box structure.

Due to these promising initial results, there is a strong desire to extend the use of GP for aircraft design both in breadth, by considering more aspects of the aircraft design problem, and in depth, by increasing the fidelity of the models used. Unfor-

[1] The “GP” acronym is overloaded, referring both to geometric programs - the class of optimization problem discussed in this work - and geometric programming - the practice of using such programs to model and solve optimization problems. The same is true of the “SP” acronym.

tunately, the restrictions on the formulation of GPs mean that not all aircraft design constraints can be readily implemented as part of a GP. A generalization of GP called Signomial Programming (SP), helps to address this by allowing constraints with less restrictive formulations [7]. A relatively small relaxation in the restriction on problem formulation means that SP can handle a significantly more general set of problems than GP, but this comes at a cost: SP does not boast the same guarantee of global optimality as GP. Despite this, solution methods remain disciplined and effective by leveraging a *difference of convex program* formulation for SP.

Signomial programming is important for aircraft design for two reasons: it allows a modeller to leverage the speed and reliability of GP on models that are not GP-compatible, and it enables increasing fidelity where it is not possible to do so in a GP-compatible way. From the author's limited experience, only a small proportion of the constraints in aircraft design models require signomials, if any. In many cases, however, omitting these constraints would mean failing to capture an important design consideration. Sometimes, the constraint in question is the only constraint that keeps one or more design variables meaningfully bounded. Thus, a sacrifice of optimization quality and robustness is made in exchange for obtaining dual feasibility and/or higher model fidelity. It is important to stress that the purpose of this work is not to use SP liberally, but rather to use it in a targeted and precise manner, where the marginal cost of introducing a signomial constraint can be justified by an adequate increase in model fidelity or accuracy. Because they result in convex restrictions on the feasible set, monomial and posynomial constraints are still viewed as the preferred approach, wherever possible.

In this work, we develop SP models for design of the vertical tail, fuselage, landing gear, horizontal tail, and wing of a commercial aircraft. These models can be used to determine optimal values for, among other things, the preliminary geometry, positioning and weight of each subsystem. The models created are readily extensible meaning constraints can be added and made more sophisticated with ease. We combine these subsystem models in a full configuration model that captures the highly coupled nature of aircraft design. To the author's best knowledge, this is the first published work on SP applied to aircraft design.

There exists extensive research in Multidisciplinary Design Optimization (MDO) methods for conceptual aircraft design [24, 19, 16, 14, 26]. Of the many different frameworks in the literature, TASOPT [16] is particularly relevant to the present work because of its use of physics based models, medium fidelity analytical models, and multidisciplinary considerations of aircraft subsystems. Common challenges faced in multidisciplinary design optimization include models that are too computationally expensive to be practical for a designer; final results that are sensitive to the choice of baseline design; evaluations of black box functions, about which an optimizer knows little; and coupling of different analysis tools that require delicate wiring between models, and generally another layer of complexity and opacity.

1.1 Introduction to Geometric Programming

First introduced in 1967 by Duffin, Peterson, and Zener [17], a GP is a specific type of constrained, nonlinear optimization problem that becomes convex after a logarithmic change of variables. Modern GP solvers employ primal-dual interior point methods [28] and are extremely fast. A typical sparse GP with tens of thousands of decision variables and one million constraints can be solved on a desktop computer in minutes [7]. Furthermore, these solvers do not require an initial guess and guarantee convergence to a *global* optimum, whenever a feasible solution exists. Being able to find optimal solutions without an initial guess makes the technique particularly useful for conceptual aircraft design, where it is important that results are not biased by preconceptions of how an optimal aircraft should look.

These impressive properties are possible because GPs represent a restricted subset of nonlinear optimization problems. In particular, the objective and constraints can only be composed of *monomial* and *posynomial* functions.

A *monomial* is a function of the form

$$m(\boldsymbol{x}) = c \prod_{j=1}^{n} x_j^{a_j}, \tag{1.1}$$

where $a_j \in \mathbb{R}$, $c \in \mathbb{R}_{++}$, and $x_j \in \mathbb{R}_{++}$. For instance, the familiar expression for lift, $\frac{1}{2}\rho V^2 C_L S$, is a monomial with $\boldsymbol{x} = (\rho, V, C_L, S)$, $c = 1/2$, and $\boldsymbol{a} = (1, 2, 1, 1)$.

A *posynomial* is a function of the form

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} c_k \prod_{j=1}^{n} x_j^{a_{jk}}, \tag{1.2}$$

where $\boldsymbol{a_k} \in \mathbb{R}^n$, $c_k \in \mathbb{R}_{++}$, and $x_j \in \mathbb{R}_{++}$. Thus, a posynomial is simply a sum of monomial terms, and all monomials are also posynomials (with just one term).

In plain English, a GP minimizes a posynomial objective function, subject to monomial equality constraints and posynomial inequality constraints. The standard form of a geometric program in mathematical notation is as follows:

$$\begin{aligned} \text{minimize} \;\; & p_0(\boldsymbol{x}) \\ \text{subject to} \;\; & p_i(\boldsymbol{x}) \leq 1, \;\; i = 1, ..., n_p, \\ & m_i(\boldsymbol{x}) = 1, \;\; i = 1, ..., n_m, \end{aligned} \tag{1.3}$$

where the p_i are posynomial (or monomial) functions, the m_i are monomial functions, and $\boldsymbol{x} \in \mathbb{R}^n_{++}$ are the decision variables.

Although this form may appear restrictive, surprisingly many physical constraints and objectives can be expressed in the necessary form [21]. Many relationships that cannot be formulated exactly as posynomials can be approximated closely, using methods for fitting GP-compatible models to data [22].

1.2 Introduction to Signomial Programming

Geometric programming is a powerful tool, with strong guarantees. As discussed previously, however, the formulation can prove restrictive. While changes of variable present an elegant way of circumventing some formulation obstacles, there may not always exist a suitable variable change. In particular, the restriction $c > 0$ in the definition of a posynomial can be a prohibitive obstacle for a modeler. There are many models where being able to use negative coefficients is necessary to accurately capture a relationship, such as when trying to minimize the difference between two quantities. An example of this is Lock's empirical relationship for wave drag [20] that is commonly used in conjunction with the Korn equation to estimate the drag on a transonic wing.

$$C_{D_{wave}} \geq 20(M - M_{crit})^4 \tag{1.4}$$

A *signomial* is a function with the same form as a posynomial,

$$s(\boldsymbol{x}) = \sum_{k=1}^{K} c_k \prod_{j=1}^{n} x_j^{a_{jk}}, \tag{1.5}$$

except that the coefficients, $c_k \in \mathbb{R}$, can now be any real number. In particular, they can be non-positive. A *signomial program* is a generalization of a geometric program that allows signomial constraints. The 'difference of convex' formulation of a signomial program also permits the objective function to be a ratio of posynomials and is given by:

$$\begin{aligned} \text{minimize} \quad & \frac{p_0(\boldsymbol{x})}{q_0(\boldsymbol{x})} \\ \text{subject to} \quad & s_i(\boldsymbol{x}) \leq 0, \;\; i = 1, ..., n_s, \\ & p_i(\boldsymbol{x}) \leq 1, \;\; i = 1, ..., n_p, \\ & m_i(\boldsymbol{x}) = 1, \;\; i = 1, ..., n_m. \end{aligned} \tag{1.6}$$

The justification for using this formulation is presented in Appendix A.

Although (1.6) is standard form for a signomial program, the majority of signomial constraints presented in this work take the form $p_1(\boldsymbol{x}) \leq p_2(\boldsymbol{x})$ or $s(\boldsymbol{x}) \leq p(\boldsymbol{x})$, because these are often more intuitive, and both can easily be transformed into the standard form $s(\boldsymbol{x}) \leq 0$. This follows the geometric programming convention of using posynomial inequality constraints of the form $p(\boldsymbol{x}) \leq m(\boldsymbol{x})$ and monomial equality constraints of the form $m_1(\boldsymbol{x}) = m_2(\boldsymbol{x})$ [7].

An important point is that *adding just one signomial constraint to a geometric program with arbitrarily many posynomial constraints changes the geometric program to a signomial program.*

The bad news is that the increased expressiveness of signomial programming comes at a price: we can no longer guarantee a *global* optimum, because, unlike with GP, the log transformation of a signomial program is not a convex optimization problem. The good news is that there is a disciplined method for solving Signomial Programs (SPs).

1.3 Signomial Programming Solution Methods

There are a number of different methods for solving SPs, however a common approach is to find a local GP approximation to the SP, solve this GP and then repeat, using the optimal solution as the point about to which take the next GP approximation, until the solution converges [7]. The GP approximation is obtained by approximating each signomial constraint with a posynomial constraint. The first step, if it has not already been done, is to express each signomial, $s(\boldsymbol{x})$, as a difference of posynomials, $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$, and rearrange them to the form of a posynomial less than or equal to another posynomial.

$$\begin{aligned} s(\boldsymbol{x}) &\leq 0 \\ p(\boldsymbol{x}) - q(\boldsymbol{x}) &\leq 0 \\ p(\boldsymbol{x}) &\leq q(\boldsymbol{x}) \qquad (1.7) \end{aligned}$$

Although (1.7) is not a GP-compatible constraint, it can be made into a GP constraint if posynomial $q(\boldsymbol{x})$ is replaced with its local monomial approximation, $\hat{q}(\boldsymbol{x};\boldsymbol{x^0})$, because a posynomial divided by a monomial is also a posynomial.

$$p(\boldsymbol{x}) \leq \hat{q}(\boldsymbol{x};\boldsymbol{x^0}) \qquad (1.8)$$

$$\frac{p(\boldsymbol{x})}{\hat{q}(\boldsymbol{x};\boldsymbol{x^0})} \leq 1 \qquad (1.9)$$

Finding a monomial approximation to a posynomial is equivalent to finding a local affine approximation to a non-linear function in log space. The best-possible local monomial approximation to a posynomial $q(\boldsymbol{x})$ at the point $\boldsymbol{x^0}$ is given by [7]:

$$\hat{q}(\boldsymbol{x})\big|_{\boldsymbol{x^0}} = q(\boldsymbol{x^0}) \prod_{i=1}^{n} \left(\frac{x_i}{x_i^0}\right)^{a_i} \tag{1.10}$$

where x_i are the elements of $\boldsymbol{x}$ and:

$$a_i = \frac{x_i^0}{q(\boldsymbol{x^0})} \frac{\partial q}{\partial x_i}. \tag{1.11}$$

Signomial programming, using formulation (1.6), is an example of 'difference of convex' programming, because the logarithmically-transformed problem can be expressed as

$$\begin{aligned} &\text{minimize} && f_0(\boldsymbol{x}) - g_0(\boldsymbol{x}) \\ &\text{subject to} && f_i(\boldsymbol{x}) - g_i(\boldsymbol{x}) \leq 0, \quad i = 1, ..., m \end{aligned} \tag{1.12}$$

where f_i and g_i are convex. This means that, for the convex (GP) approximation, $\hat{f}(\boldsymbol{x})$, of the non-convex (SP) function, $f(\boldsymbol{x}) - g(\boldsymbol{x})$,

$$\hat{f}(\boldsymbol{x}) \geq f(\boldsymbol{x}) \quad \forall \boldsymbol{x}. \tag{1.13}$$

Because of this, the true feasible set contains the feasible set of the convexified problem, and there is no need for a trust region [6], meaning that there is no need to tune solver parameters for controlling initial trust region sizes and/or update rules. Solving an SP is, therefore, considerably more reliable than solving a general nonlinear program, as there are fewer solver algorithm parameters to tune.

1.4 Structure of this Thesis

The models presented in this work are a compilation of aircraft design constraints and relationships from a variety of references. These models are therefore not necessarily original, but their adaptation to signomial programming is.

Each subsystem model presented begins with assumptions underlying the model followed by a description of the constraints. The intention is to demonstrate the wide range of aircraft design constraints that fit naturally into the signomial programming formulation. Each model is solved using estimates for fixed variables for a reference aircraft, the Boeing 737-800, and a brief discussion of the results is presented. Although the emphasis of this work is not on these solutions, they do allow us to verify that the optimization models have feasible solutions and produce credible results.

The models are then combined together in an aircraft model with fully coupled subsystems as described in chapter 7. In anticipation of further work, three atmospheric models are presented in chapter 8.

To strengthen the claim that signomial programming is a viable tool for aircraft design, we also compare it to alternative optimization techniques. GP and SP are compared with general nonlinear programming techniques in chapter 9, and a discussion of the improvement in fidelity achieved by using SP instead of GP is presented in chapter 10.

All of the geometric and signomial programs presented in this work were composed and solved with GPkit [9], a python package for defining and manipulating geometric programming models, using MOSEK [3] with an academic license as the backend solver.

Chapter 2

Vertical Tail Model

At a conceptual design level, the purpose of an aircraft's vertical tail is two-fold. Firstly, it must provide stability in yaw. Secondly, it must provide adequate yaw control authority in critical flight conditions. For a multi-engine aircraft, the critical flight condition is typically an engine failure at low speeds. The vertical tail must be capable of providing sufficient sideforce in this case [31]. The design of the vertical tail is therefore coupled to the size of the fuselage and the position of the engines.

2.1 Model Assumptions

The high level assumptions for this model are that the aircraft has a single vertical tail, the horizontal tail is mounted in a conventional configuration, so as to not require a reinforced vertical tail structure, and the aircraft has two wing-mounted engines. This model also assumes that the ultimate high level model's objective (e.g. minimizing fuel consumption or operating costs) would apply downward pressure on both weight and drag.

2.2 Model Description

The vertical tail model has 36 free variables and 36 constraints.

Table 2.1: Free variables in the vertical tail model

Free Variables	Units	Description
A_{fan}	$[\mathrm{m}^2]$	Engine reference area
A_{vt}	$[-]$	Vertical tail aspect ratio
$C_{D_{vis}}$	$[-]$	Viscous drag coefficient
$C_{L_{vt}}$	$[-]$	Vertical tail lift coefficient
D_{vt}	[N]	Vertical tail viscous drag, cruise
D_{wm}	[N]	Engine out windmill drag
$L_{max_{vt}}$	[N]	Maximum load for structural sizing
$L_{v_{max}}$	[N]	Maximum load for structural sizing
L_{vt}	[N]	Vertical tail lift in engine out
Re_{vt}	$[-]$	Vertical tail reynolds number, cruise
S	$[\mathrm{m}^2]$	Vertical tail reference area (full)
S_{vt}	$[\mathrm{m}^2]$	Vertical tail reference area (half)
W_{struct}	[N]	Full span weight
W_{vt}	[N]	Vertical tail weight
Δx_{lead_v}	[m]	Distance from CG to vertical tail leading edge
Δx_{trail_v}	[m]	Distance from CG to vertical tail trailing edge
$\bar{c}_{vt}$	[m]	Vertical tail mean aero chord
λ_{vt}	$[-]$	Vertical tail taper ratio
τ_{vt}	$[-]$	Vertical tail thickness/chord ratio
b	[m]	Vertical tail full span
b_{vt}	[m]	Vertical tail half span
$c_{root_{vt}}$	[m]	Vertical tail root chord
$c_{tip_{vt}}$	[m]	Vertical tail tip chord
l_{vt}	[m]	Vertical tail moment arm
p_{vt}	$[-]$	Substituted variable = 1 + 2*taper
q_{vt}	$[-]$	Substituted variable = 1 + taper
$z_{\bar{c}_{vt}}$	[m]	Vertical location of mean aerodynamic chord
WingBox		
A	$[-]$	Aspect ratio
I_{cap}	$[-]$	Non-dim spar cap area moment of inertia
M_r	[N]	Root moment per root chord

W_{cap}	[N]	Weight of spar caps
W_{struct}	[N]	Structural weight
W_{web}	[N]	Weight of shear web
ν	$[-]$	Dummy variable $= (t^2 + t + 1)/(t + 1)$
t_{cap}	$[-]$	Non-dim. spar cap thickness
t_{web}	$[-]$	Non-dim. shear web thickness

Table 2.2: Fixed variables in the vertical tail model

Constants	Units	Description
$C_{D_{wm}}$	$[-]$	Windmill drag coefficient
$C_{L_{vmax}}$	$[-]$	Max lift coefficient
T_e	[N]	Thrust per engine at takeoff
V_1	$[\frac{\mathrm{m}}{\mathrm{s}}]$	Minimum takeoff velocity
V_∞	$[\frac{\mathrm{m}}{\mathrm{s}}]$	Cruise velocity
V_{ne}	$[\frac{\mathrm{m}}{\mathrm{s}}]$	Never exceed velocity
μ	$[\frac{\mathrm{N \cdot s}}{\mathrm{m}^2}]$	Dynamic viscosity (35,000 ft)
ρ_c	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Air density (35,000ft)
ρ_{TO}	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Air density (SL))
$\tan(\Lambda_{vt})$	$[-]$	Tangent of leading edge sweep (40 deg)
$c_{l_{vt}}$	$[-]$	Sectional lift force coefficient (engine out)
d_{fan}	[m]	Fan diameter
e_v	$[-]$	Span efficiency of vertical tail
l_{fuse}	[m]	Length of fuselage
x_{CG}	[m]	x-location of CG
y_{eng}	[m]	Engine moment arm
WingBox		
N_{lift}	$[-]$	Wing loading multiplier
ρ_{cap}	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Density of spar cap material
ρ_{web}	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Density of shear web material
$\sigma_{max,shear}$	[Pa]	Allowable shear stress
σ_{max}	[Pa]	Allowable tensile stress
$f_{w,add}$	$[-]$	Wing added weight fraction
g	$[\frac{\mathrm{m}}{\mathrm{s}^2}]$	Gravitational acceleration
r_h	$[-]$	Fractional wing thickness at spar web

w	$[-]$	Wingbox-width-to-chord ratio

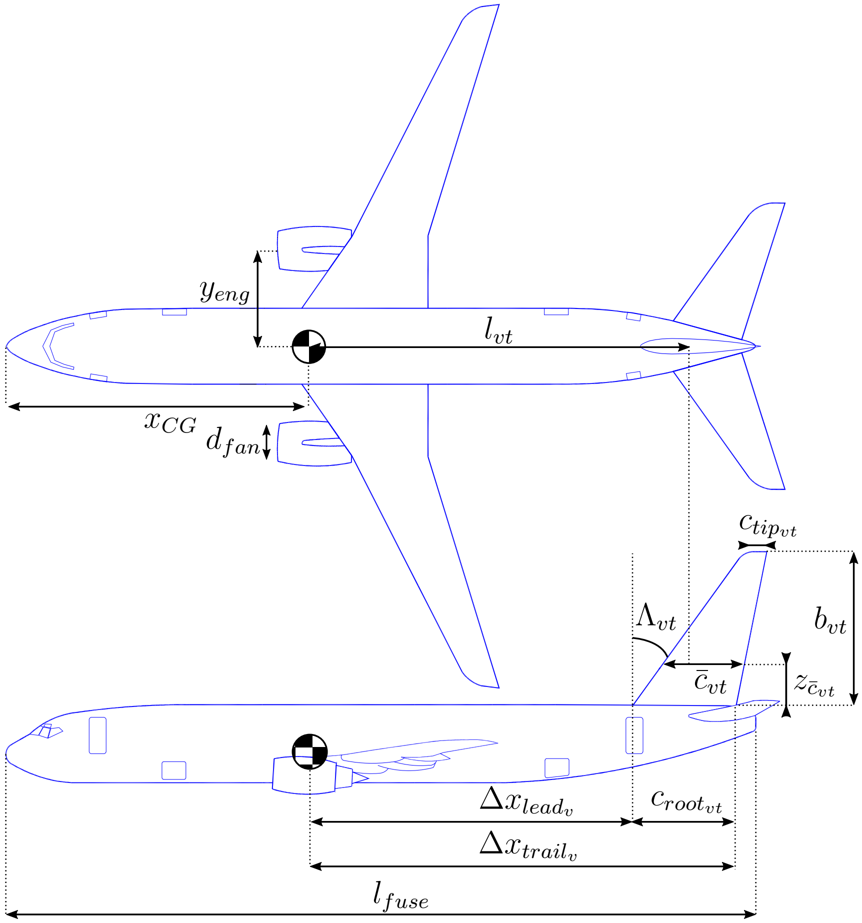

Figure 2-1: Geometric variables[1] of the vertical tail model (adapted from [4])

The first constraint specifies that the maximum moment exerted by the tail must be greater than or equal to the moment exerted by the engines in an engine-out condition. In the worst case scenario, the asymmetric thrust is exacerbated by the windmill drag of the engine that is inoperative [16].

$$L_{vt} l_{vt} \geq D_{wm} y_{eng} + T_e y_{eng} \tag{2.1}$$

The moment arm of the vertical tail is the distance from the aircraft Center of Gravity (CG) to the aerodynamic center of the vertical tail, which is assumed to be at the

[1] Geometric in the sense that they prescribe geometry, not in the sense of geometric programming, which derives its name from the same etymology as the geometric mean

quarter chord. The moment arm is therefore upper bounded by the distance from the CG to the leading edge of the tail at the root, the height of the mean aerodynamic chord above the fuselage, the sweep angle, and the mean aerodynamic chord.

$$l_{vt} \leq \Delta x_{lead} + z_{\bar{c}_{vt}} \tan(\Lambda_{LE}) + 0.25\bar{c}_{vt} \tag{2.2}$$

Note that this is a signomial constraint. A simple GP-compatible approximation to this constraint would be to neglect sweep and to assume that the aerodynamic center is at the leading edge.

The worst case engine out condition is likely to occur during takeoff, when the velocity is lowest but the engine force required to safely complete takeoff is highest. The force exerted by the vertical tail in this critical low speed case is determined as the maximum lift coefficient of the tail and vertical tail reference area multiplied by the minimum dynamic pressure, which might typically occur at the rotation speed at sea level. As a conservative estimate, the V_1 speed is used because it is the minimum speed after which a takeoff can be completed, following a critical engine failure. This constraint can be made more demanding by assuming a high altitude runway.

$$L_{vt} = \frac{1}{2}\rho_{TO} {V_1}^2 S_{vt} C_{L_{vt}} \tag{2.3}$$

The 3D lift coefficient is constrained by the airfoil sectional lift coefficient using finite wing theory [1].

$$C_{L_{vt}} \left(1 + \frac{c_{l_{vt}}}{\pi e A_{vt}}\right) \leq c_{l_{vt}} \tag{2.4}$$

Meanwhile, the windmill drag can, to a first approximation, be lower bounded using a drag coefficient and a reference area [16], in this case the area of the engine fan.

$$D_{wm} \geq \frac{1}{2}\rho_{TO} {V_1}^2 A_{fan} C_{D_{wm}} \tag{2.5}$$

$$A_{fan} \geq \pi \left(\frac{d_{fan}}{2}\right)^2 \tag{2.6}$$

A simple formula relates the reference area, span and mean geometric chord, for a trapezoidal tail. The reference area needs to be upper bounded so this becomes a signomial constraint.

$$S_{vt} \leq b_{vt} \frac{c_{root_{vt}} + c_{tip_{vt}}}{2} \tag{2.7}$$

To further define the geometry of the vertical tail, the x-coordinates of the leading and trailing edge at the root are related by the root chord. The tail trailing edge is upper bounded by imposing a constraint that the tail root cannot extend beyond the end of the fuselage. Together these constraints put an upper bound on the moment arm of the tail based on the length of the fuselage.

$$\Delta x_{trail} \geq \Delta x_{lead} + c_{root_{vt}} \tag{2.8}$$

$$l_{fuse} \geq x_{CG} + \Delta x_{trail} \tag{2.9}$$

The mean aerodynamic chord for a trapezoidal wing is defined as:

$$\bar{c}_{vt} = \frac{2}{3} \left(\frac{1 + \lambda_{vt} + \lambda_{vt}^2}{1 + \lambda_{vt}} \right) c_{root_{vt}} \tag{2.10}$$

This can be made into a signomial constraint. Though not absolutely necessary, substitute variables p_{vt} and q_{vt} already appear in the structural model and are therefore convenient to use here too.

$$\bar{c}_{vt} \leq \frac{2}{3} \left(\frac{1 + \lambda_{vt} + \lambda_{vt}^2}{q_{vt}} \right) c_{root_{vt}} \tag{2.11}$$

$$2q_{vt} \geq 1 + p_{vt} \tag{2.12}$$

$$p_{vt} \geq 1 + 2\lambda_{vt} \tag{2.13}$$

$$\lambda_{vt} = \frac{c_{tip_{vt}}}{c_{root_{vt}}} \tag{2.14}$$

These posynomial constraint substitutions can be reused for the constraint that defines the vertical position of the mean aerodynamic chord, which saves us from needing to implement another signomial constraint.

$$z_{\bar{c}_{vt}} = \frac{b_{vt} q_{vt}}{3 p_{vt}} \tag{2.15}$$

It is necessary put a lower bound on taper to avoid unacceptably small Reynolds numbers at the tip [25]. For the purpose of this work, the taper is lower bounded by the taper ratio of the reference aircraft's vertical tail [8].

$$\lambda_{vt} \geq 0.27 \tag{2.16}$$

The viscous drag for the tail in cruise is lower bounded using the familiar expression for drag. A GP-compatible fitted model [22] is used to capture the dependence on airfoil thickness and Reynolds number based on the mean aerodynamic chord. 50 data points were sampled using XFOIL [15] and a posynomial constraint was then fitted to these data points.

$$D_{vt} \geq \frac{1}{2} \rho_c {V_\infty}^2 S_{vt} C_{D_{vis}} \tag{2.17}$$

$$\begin{aligned} {C_{D_{vis}}}^{0.12} \geq & 0.118 {Re_{vt}}^{0.0016} \tau^{0.0082} + 0.198 {Re_{vt}}^{0.0017} \tau^{0.0077} + & (2.18) \\ & 0.19 {Re_{vt}}^{0.0017} \tau^{0.0075} + 1.83 \times 10^4 \frac{\tau^{3.5}}{{Re_{vt}}^{0.49}} & (2.19) \end{aligned}$$

$$Re_{vt} = \frac{\rho_c V_\infty \bar{c}_{vt}}{\mu} \tag{2.20}$$

Finally, the tail model adopts the wing structure model from Hoburg [21], which adds a further 12 constraints. This structural model requires a maximum load, which is constrained by a maximum dynamic pressure and a maximum lift coefficient.

$$L_{v_{max}} = \frac{1}{2} \rho_{TO} V_{ne}^2 S_{vt} C_{L_{vmax}} \tag{2.21}$$

The wing structure model is for a full span wing, whereas, naturally, the vertical tail only has a “half” span. For this reason the vertical tail geometry and weight must be related to the geometry and weight definitions used in the structural model.

$$2S_{vt} = S \tag{2.22}$$

$$2b_{vt} = b \tag{2.23}$$

$$2W_{vt} = W_{struct} \tag{2.24}$$

2.3 Model Results

By using fixed values representative of the reference aircraft and a placeholder objective function, a solution was obtained for this SP. The placeholder objective function is a somewhat arbitrary function of viscous drag in cruise and weight, and is intended to reflect how a higher level objective might apply pressure to the vertical tail model.

$$0.05W_{vt} + D_{vt} \tag{2.25}$$

Solving the SP took 4 GP solves and 0.48 seconds. A comparison of the optimal values with the values for the reference aircraft can be seen in Table 2.3.

Table 2.3: Solution comparison with reference aircraft

Design Variable	Solution Value	Estimate for reference aircraft [8]
A_{vt} [−]	1.95	1.91
b_{vt} [m]	7.16	7.16
S_{vt} [m^2]	26.3	26.4

The sensitivity to a selection of fixed variables is presented in Table 2.4. Each sensitivity is an estimate for how much the objective function would change given a 1% increase in the value used for that variable. For example, the objective function value would decrease by approximately -1.4% if the fuselage length increased by 1%. The convergence of the objective value for the model is show in Figure 2-2.

Table 2.4: The vertical tail model is most sensitive to the decision speed, the cruise velocity and the fuselage length.

Fixed Variable	Value	Sensitivity
V_1 [m/s]	70	-2.4%
V_∞ [m/s]	234	1.5%
l_{fuse} [m]	39	-1.4%

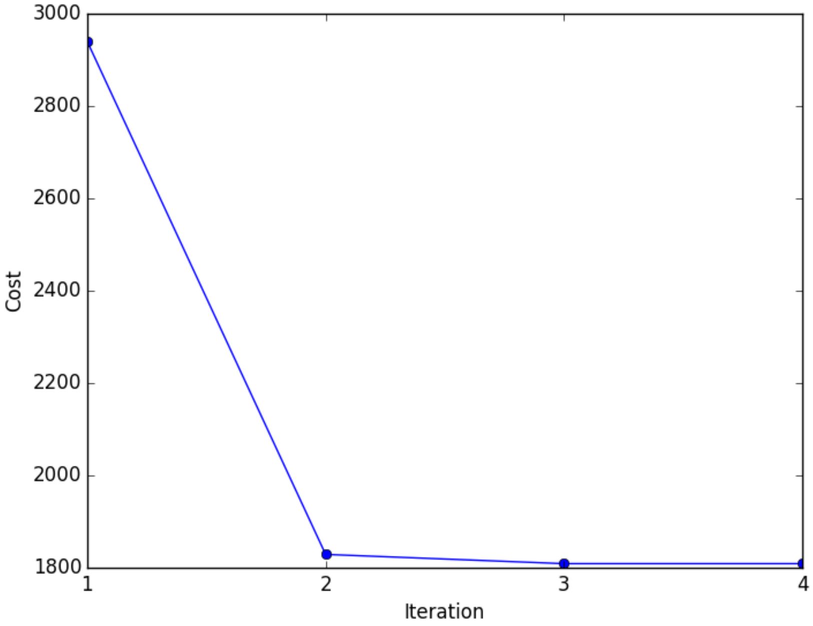

Figure 2-2: The vertical tail model solves in 4 GP iterations

Chapter 3

Fuselage Model

At a high level, the purpose of a conventional commercial aircraft fuselage can be decomposed into two primary functions: integrating and connecting all of the subsystems (e.g. wing, tail, landing gear), and carrying the payload, which typically consists of passengers, luggage, and sometimes cargo. The design of the fuselage is therefore coupled with virtually every aircraft subsystem.

Drela [16] performs a detailed, but still approximate, analysis of fuselage structure and weight, considering pressure loads, torsion loads, buoyancy weight, window weight, payload proportional weights, the floor, the tail cone, and bending loads. The majority of the constraints in this model are adapted directly from these equations.

3.1 Model Assumptions

This model assumes a single circular cross section fuselage. This is slightly inaccurate for narrowbody aircraft like the Boeing 737 and Airbus A320, neither of which have perfectly circular cross sections. The model also assumes a single vertical tail with a conventional configuration, and a single aisle. It also assumes that a higher level model dictates an optimal number of passengers, and again, that drag and weight are both undesirable.

3.2 Model Description

The fuselage model has 61 free variables and 63 constraints.

Table 3.1: Free variables in the fuselage model

Free Variables	Units	Description
A_{floor}	$[\mathrm{m}^2]$	Floor beam x-sectional area
A_{fuse}	$[\mathrm{m}^2]$	Fuselage x-sectional area
A_{hold}	$[\mathrm{m}^2]$	Cargo hold x-sectional area
A_{skin}	$[\mathrm{m}^2]$	Skin cross sectional area
$D_{friction}$	$[\mathrm{N}]$	Friction drag
D_{fuse}	$[\mathrm{N}]$	Total drag in cruise
$D_{upsweep}$	$[\mathrm{N}]$	Drag due to fuse upsweep
FF	$[-]$	Fuselage form factor
M_{floor}	$[\mathrm{N}\cdot\mathrm{m}]$	Max bending moment in floor beams
P_{floor}	$[\mathrm{N}]$	Distributed floor load
R_{fuse}	$[\mathrm{m}]$	Fuselage radius
S_{bulk}	$[\mathrm{m}^2]$	Bulkhead surface area
S_{floor}	$[\mathrm{N}]$	Maximum shear in floor beams
S_{nose}	$[\mathrm{m}^2]$	Nose surface area
V_{bulk}	$[\mathrm{m}^3]$	Bulkhead skin volume
V_{cabin}	$[\mathrm{m}^3]$	Cabin volume
V_{cargo}	$[\mathrm{m}^3]$	Cargo volume
V_{cone}	$[\mathrm{m}^3]$	Cone skin volume
V_{cyl}	$[\mathrm{m}^3]$	Cylinder skin volume
V_{floor}	$[\mathrm{m}^3]$	Floor volume
V_{hold}	$[\mathrm{m}^3]$	Hold volume
V_{lugg}	$[\mathrm{m}^3]$	Luggage volume
V_{nose}	$[\mathrm{m}^3]$	Nose skin volume
W_{apu}	$[\mathrm{N}]$	APU weight
W_{buoy}	$[\mathrm{N}]$	Buoyancy weight
W_{cone}	$[\mathrm{N}]$	Cone weight
W_{floor}	$[\mathrm{N}]$	Floor weight
W_{fuse}	$[\mathrm{N}]$	Fuselage weight
W_{insul}	$[\mathrm{N}]$	Insulation material weight
W_{lugg}	$[\mathrm{N}]$	Passenger luggage weight

W_{padd}	[N]	Misc weights (galley, toilets, doors etc.)
W_{pass}	[N]	Passenger weight
W_{pay}	[N]	Payload weight
W_{seat}	[N]	Seating weight
W_{shell}	[N]	Shell weight
W_{skin}	[N]	Skin weight
W_{window}	[N]	Window weight
λ_{cone}	$[-]$	Tailcone radius taper ratio (xshell2->xtail)
ϕ	$[-]$	Upsweep angle
ρ_{cabin}	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Air density in cabin
σ_x	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$	Axial stress in skin
σ_θ	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$	Skin hoop stress
τ_{cone}	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$	Shear stress in cone
f	$[-]$	Fineness ratio
h_{floor}	[m]	Floor beam height
h_{hold}	[m]	Height of the cargo hold
l_{cone}	[m]	Cone length
l_{floor}	[m]	Floor length
l_{fuse}	[m]	Fuselage length
l_{nose}	[m]	Nose length
l_{shell}	[m]	Shell length
n_{pass}	$[-]$	Number of passengers
n_{rows}	$[-]$	Number of rows
p_{λ_v}	$[-]$	1 + 2*Tail taper ratio
t_{shell}	[m]	Shell thickness
t_{skin}	[m]	Skin thickness
w_{floor}	[m]	Floor width
w_{fuse}	[m]	Fuselage width
x_{shell1}	[m]	Start of cylinder section
x_{shell2}	[m]	End of cylinder section
x_{up}	[m]	Fuselage upsweep point

Table 3.2: Fixed variables in the fuselage model

Constants	Units	Description
LF	$[-]$	Load factor
$L_{v_{max}}$	[N]	Max vertical tail load
N_{land}	$[-]$	Emergency landing load factor
R	$[\frac{\mathrm{J}}{(\mathrm{K}\cdot\mathrm{kg})}]$	Universal gas constant
SPR	$[-]$	Number of seats per row
T_{cabin}	[K]	Cabin temperature
V_∞	$[\frac{\mathrm{m}}{\mathrm{s}}]$	Cruise velocity
W''_{floor}	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$	Floor weight/area density
W''_{insul}	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$	Weight/area density of insulation material
W'_{seat}	[N]	Weight per seat
W'_{window}	$[\frac{\mathrm{N}}{\mathrm{m}}]$	Weight/length density of windows
$W_{avg.pass}$	[lbf]	Average passenger weight
W_{cargo}	[N]	Cargo weight
$W_{carryon}$	[lbf]	Ave. carry-on weight
$W_{checked}$	[lbf]	Ave. checked bag weight
W_{fix}	[lbf]	Fixed weights (pilots, cockpit seats, navcom)
Δh	[m]	Distance from floor to widest part of fuselage
Δp	[Pa]	Pressure difference across fuselage skin
μ	$[\frac{\mathrm{N}\cdot\mathrm{s}}{\mathrm{m}^2}]$	Dynamic viscosity (35,000 ft)
ρ_∞	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Air density (35,000ft)
ρ_{bend}	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Stringer density
ρ_{cargo}	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Cargo density
ρ_{cone}	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Cone material density
ρ_{floor}	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Floor material density
ρ_{lugg}	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Luggage density
ρ_{skin}	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Skin density
σ_{floor}	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$	Max allowable cap stress
σ_{skin}	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$	Max allowable skin stress
τ_{floor}	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$	Max allowable shear web stress
b_{vt}	[m]	Vertical tail span
c_{vt}	[m]	Vertical tail root chord
f_{apu}	$[-]$	APU weight as fraction of payload weight

f_{fadd}	$[-]$	Fractional added weight of local reinforcements
f_{frame}	$[-]$	Fractional frame weight
$f_{lugg,1}$	$[-]$	Proportion of passengers with one suitcase
$f_{lugg,2}$	$[-]$	Proportion of passengers with two suitcases
f_{padd}	$[-]$	Other misc weight as fraction of payload weight
f_{string}	$[-]$	Fractional weight of stringers
g	$[\frac{\text{m}}{\text{s}^2}]$	Gravitational acceleration
n_{seat}	$[-]$	Number of seats
p_s	[in]	Seat pitch
p_{cabin}	[Pa]	Cabin air pressure (8,000ft)
r_E	$[-]$	Ratio of stringer/skin moduli
w_{aisle}	[m]	Aisle width
w_{seat}	[m]	Seat width
w_{sys}	[m]	Width between cabin and skin for systems

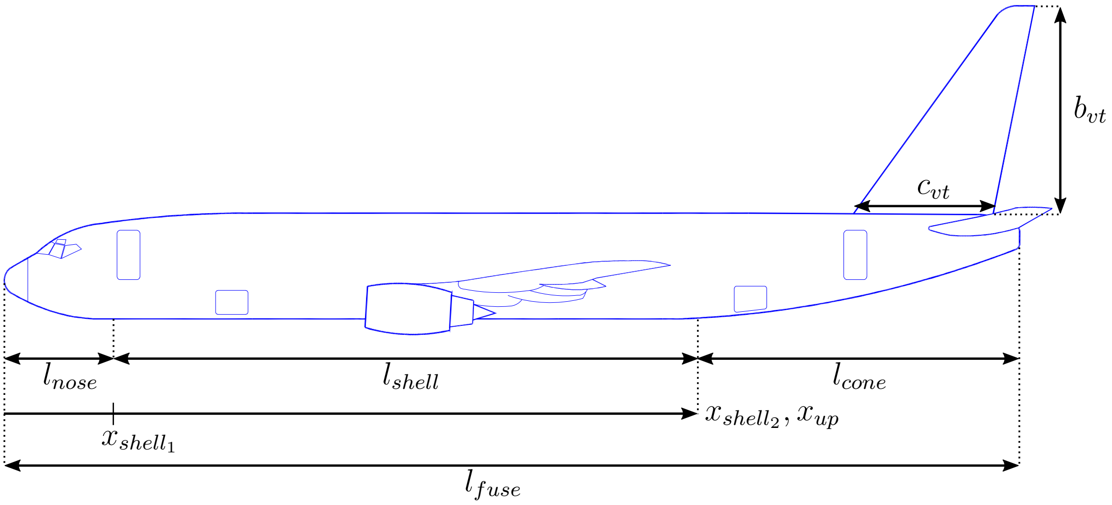

Figure 3-1: Geometric variables of the fuselage model (adapted from [4])

The fuselage must be wide enough to accommodate the width of the seats in a row and the width of the aisle.

$$w_{fuse} \geq (SPR)w_{seat} + w_{aisle} + 2w_{sys} \tag{3.1}$$

The cross sectional area of the fuselage skin is lower bounded using a thin walled

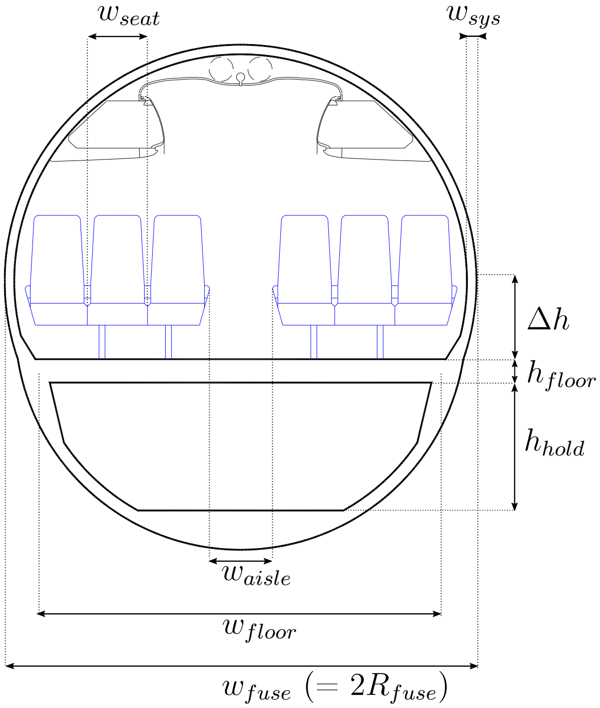

Figure 3-2: Geometric variables (cross-section) of the fuselage model (adapted from [4])

cylinder assumption.

$$A_{skin} \geq 2\pi R_{fuse} t_{skin} \tag{3.2}$$

The cross sectional area of the fuselage is lower bounded using the radius of the fuselage.

$$A_{fuse} \geq \pi {R_{fuse}}^2 \tag{3.3}$$

The effective modulus-weight shell thickness is lower bounded by assuming that only the skin and stringers contribute to bending. This constraint also uses an assumed fractional weight of stringers that scales with the thickness of the skin.

$$t_{shell} \geq t_{skin} \left(1 + f_{string} r_E \frac{\rho_{skin}}{\rho_{bend}}\right) \tag{3.4}$$

The axial and hoop stresses in the fuselage skin are constrained by the pressurization load due to the difference between cabin pressure and ambient pressure at cruise altitude. The thickness of the skin is therefore sized by the maximum allowable stress of the chosen material.

$$\sigma_x = \frac{\Delta p}{2}\frac{R_{fuse}}{t_{shell}} \tag{3.5}$$

$$\sigma_\theta = \Delta p \frac{R_{fuse}}{t_{skin}} \tag{3.6}$$

$$\sigma_{skin} \geq \sigma_x \tag{3.7}$$

$$\sigma_{skin} \geq \sigma_\theta \tag{3.8}$$

With the skin thickness constrained we can assert a lower bound for the weight of the fuselage skin, excluding the tail cone. To do this, we need to constrain the skin surface area, and, in turn, skin volume for the nose, main cabin, and rear bulkhead. The surface area of the nose, which is approximated as an ellipse, is lower bounded using Cantrell's approximation [16]. The surface area of the hemispherical rear bulkhead is also constrained.

$${S_{nose}}^{\frac{8}{5}} \geq \left(2\pi R_{fuse}^2\right)^{\frac{8}{5}} \left(\frac{1}{3} + \frac{2}{3}\left(\frac{l_{nose}}{R_{fuse}}\right)^{\frac{8}{5}}\right) \tag{3.9}$$

$$S_{bulk} = 2\pi {R_{fuse}}^2 \tag{3.10}$$

$$V_{cyl} = A_{skin} l_{shell} \tag{3.11}$$

$$V_{nose} = S_{nose} t_{skin} \tag{3.12}$$

$$V_{bulk} = S_{bulk} t_{skin} \tag{3.13}$$

$$W_{skin} \geq \rho_{skin} g \left(V_{bulk} + V_{cyl} + V_{nose}\right) \tag{3.14}$$

The weight of the fuselage shell is then constrained by accounting for the weights of the frame, stringers, and other structural components, all of which are assumed to scale with the weight of the skin.

$$W_{shell} \geq W_{skin}\left(1 + f_{fadd} + f_{frame} + f_{string}\right) \tag{3.15}$$

The effective buoyancy weight of the aircraft is also constrained using a specified cabin pressure at cruise, the ideal gas law, an approximate cabin volume, and the difference in pressure with respect to the ambient pressure at cruise. This requires a signomial constraint.

$$\rho_{cabin} = \frac{p_{cabin}}{RT_{cabin}} \tag{3.16}$$

$$V_{cabin} \geq A_{fuse} \left(0.67R_{fuse} + 0.67l_{nose} + l_{shell}\right) \tag{3.17}$$

$$W_{buoy} \geq \left(\rho_{cabin} - \rho_{\infty}\right) gV_{cabin} \tag{3.18}$$

The window and insulation weight is lower bounded using assumed weight/length and weight/area densities respectively. It is assumed that only the cabin is insulated and that the cabin takes up approximately 55% of the fuselage cross section.

$$W_{window} = W'_{window} l_{shell} \tag{3.19}$$

$$W_{insul} \geq W''_{insul} \left(0.55 \left(S_{bulk} + S_{nose}\right) + 1.1\pi R_{fuse} l_{shell}\right) \tag{3.20}$$

The APU is assumed to be proportional to the payload weight. Other payload proportional weight is also accounted for using a weight fraction. This category includes flight attendants, food, galleys, toilets, furnishing, doors, lighting, air conditioning, and in-flight entertainment systems. The total seat weight is a product of the weight per seat and the number of seats.

$$W_{apu} = W_{pay} f_{apu} \tag{3.21}$$

$$W_{padd} = W_{pay} f_{padd} \tag{3.22}$$

$$W_{seat} = W'_{seat} n_{seat} \tag{3.23}$$

The floor must be designed to withstand at least the weight of the payload and seats multiplied by a safety factor for an emergency landing.

$$P_{floor} \geq N_{land} W_{pay} + N_{land} W_{seat} \tag{3.24}$$

The maximum moment and shear in the floor are determined based on this design load and the width of the floor, assuming that the floor/wall joints are pinned and there are no center supports.

$$S_{floor} = \frac{P_{floor}}{2} \tag{3.25}$$

$$M_{floor} = \frac{P_{floor} w_{floor}}{8} \tag{3.26}$$

The nose must be long enough to have an aerodynamic profile and to accommodate the cockpit. A reasonable, but arbitrary, lower bound is employed for this work [16].

$$l_{nose} \geq 5.2 \ \ [\mathrm{m}] \tag{3.27}$$

The floor beam cross sectional area is constrained by the maximum allowable cap stress and shear web stress for the beams.

$$A_{floor} \geq 1.5 \frac{S_{floor}}{\tau_{floor}} + 2 \frac{M_{floor}}{\sigma_{floor} h_{floor}} \tag{3.28}$$

The seat pitch and the number of rows are fixed and constrain the length of the shell. Meanwhile, the required height from the widest part of the fuselage to the floor constrains the width of the floor. Such a constraint could reflect a passenger comfort requirement or even a window viewing angle requirement. This is another example of a signomial constraint.

$$l_{shell} \geq n_{rows} p_s \tag{3.29}$$

$$l_{floor} \geq 2R_{fuse} + l_{shell} \tag{3.30}$$

$$\left(\frac{w_{floor}}{2}\right)^2 + \Delta h^2 \geq {R_{fuse}}^2 \tag{3.31}$$

The weight of the floor is lower bounded by the density of the floor beams multiplied by the floor beam volume, in addition to an assumed weight/area density for the planking.

$$V_{floor} \geq A_{floor} w_{floor} \tag{3.32}$$

$$W_{floor} \geq V_{floor} \rho_{floor} g + W''_{floor} l_{floor} w_{floor} \tag{3.33}$$

The tail cone needs to be able to transfer the loads exerted on the vertical tail to the rest of the fuselage. The maximum torsion moment imparted by the vertical tail depends on the maximum force exerted on the tail as well as the span and taper ratio of the tail. This torsion moment along with the cone cross sectional area and the maximum shear stress of the cone material determines the necessary cone skin thickness . The cone cross sectional area, which of course varies along the cone, is coarsely approximated to be the fuselage cross sectional area (i.e. the cross sectional area of the cone base).

$$Q_v = \frac{L_{v_{max}} b_{vt}}{3} \frac{1 + 2\lambda_v}{1 + \lambda_v} \tag{3.34}$$

$$t_{cone} = \frac{Q_v}{2 A_{fuse} \tau_{cone}} \tag{3.35}$$

The volume of the cone is a definite integral from the base to the tip of the cone. This integral is evaluated [16] and combined with Equation 3.34 and Equation 3.35 to give a single signomial constraint on the cone skin volume.

$$R_{fuse} \tau_{cone} (1 + p_{\lambda_v}) V_{cone} \frac{1 + \lambda_{cone}}{4 l_{cone}} \geq L_{v_{max}} b_{vt} \frac{p_{\lambda_v}}{3} \tag{3.36}$$

A change of variables is used for compatibility with the tail model, which uses $p_{\lambda_v} = 1 + 2\lambda_v$ to make a structural constraint GP compatible. The same taper lower bound is introduced as in the tail model.

$$p_{\lambda_v} \geq 1.6 \tag{3.37}$$

The cone skin shear stress is constrained to equal the maximum allowable stress in the skin material.

$$\tau_{cone} = \sigma_{skin} \tag{3.38}$$

The tail cone taper ratio, defined as the ratio of the cone radius at the leading edge of the tail and the radius at the base of the cone (i.e. the fuselage radius), constrains the length of the cone relative to the chord of the tail.

$$\lambda_{cone} = \frac{c_{vt}}{l_{cone}} \tag{3.39}$$

As with the shell, the cone weight is bounded using assumed proportional weights for additional structural elements, stringers, and frames.

$$W_{cone} \geq \rho_{cone} g V_{cone} \left(1 + f_{fadd} + f_{frame} + f_{string}\right) \tag{3.40}$$

The passenger component of payload weight depends on load factor.

$$n_{seat} = (SPR) n_{rows} \tag{3.41}$$

$$n_{pass} = (LF) n_{seat} \tag{3.42}$$

$$W_{pass} = W_{avg.pass} n_{pass} \tag{3.43}$$

The weight of luggage is lower bounded by a buildup of 2-checked-bag customers, 1-checked-bag customers, and average carry-on weight. This constraint is disproportionately and unnecessarily detailed, but it serves to demonstrate that constraints of a posynomial form can be made arbitrarily complex for little additional cost, as discussed earlier. Such modelling could be used in future work to analyze how sensitive an aircraft design is to the accuracy of the work airlines perform to model expected baggage loads.

$$W_{lugg} \geq 2 W_{checked} f_{lugg,2} n_{pass} + W_{checked} f_{lugg,1} n_{pass} + W_{carryon} \tag{3.44}$$

The weight of luggage and cargo is of course tied to the volume and average density of luggage and cargo. The volume determined by these constraints can be used to determine required luggage hold volume, which in turn helps to determine the hold cross sectional area.

$$W_{lugg} = V_{lugg} \rho_{lugg} g \tag{3.45}$$

$$W_{cargo} = V_{cargo}\rho_{cargo}g \tag{3.46}$$

$$V_{hold} \geq V_{cargo} + V_{lugg} \tag{3.47}$$

$$V_{hold} = A_{hold}l_{shell} \tag{3.48}$$

Given the required cargo hold cross sectional area, and the fact that the cargo hold shape is approximately a segment of the circular cross section, we can constrain the required hold height using an approximation of the area of a circle segment developed by Harris [18]. This approximation is conveniently in SP form and is guaranteed to be within 0.1% of the true value for arc angles less than 150° and within 0.8% for all other angles.

$$A_{hold} \leq \frac{1}{2}\frac{{h_{hold}}^3}{w_{floor}} + \frac{2}{3}h_{hold}w_{floor} \tag{3.49}$$

This hold height introduces another constraint on the height of the floor beams and the radius of the fuselage cross section.

$$R_{fuse} \geq \Delta h + h_{floor} + h_{hold} \tag{3.50}$$

The total weight of the payload, and in turn the total weight of the fuselage, can be lower bounded by adding all of the constituent weights. The fixed weight incorporates pilots, cockpit windows, cockpit seats, flight instrumentation, navigation and communication equipment, which are expected to be roughly the same for all aircraft [16].

$$W_{pay} \geq W_{cargo} + W_{lugg} + W_{pass} \tag{3.51}$$

$$W_{fuse} \geq W_{apu} + W_{buoy} + W_{cone} + W_{floor} + W_{insul} + W_{padd} + W_{seat} + W_{shell} + W_{window} + W_{fix} \tag{3.52}$$

The drag felt by the fuselage in cruise is lower bounded using a form factor, which is in turn constrained using a function of the fineness ratio [31].

$$l_{fuse} \geq l_{nose} + l_{shell} + l_{cone} \tag{3.53}$$

$$f = \frac{l_{fuse}}{\sqrt{\left(\frac{4}{\pi}\right) A_{fuse}}} \tag{3.54}$$

$$FF \geq 1 + \frac{60}{f^3} + \frac{f}{400} \tag{3.55}$$

A lower bound approximation for the friction drag is then the form factor multiplied by the integral of the turbulent flat plate skin friction coefficient [1] over a cylinder. A simplistic assumption here is the lack of correction for compressibility effects.

$$\begin{aligned} D_{friction} &\geq (FF)\left(2\pi R_{fuse}\right)\left(\frac{1}{2}\rho {V_\infty}^2\right)\left(0.074\frac{\nu}{V_\infty}Re^{0.8}\right) \\ \implies D_{friction} &\geq 0.074\pi(FF)R_{fuse}\mu V_\infty \left(\frac{\rho_\infty V_\infty l_{fuse}}{\mu}\right)^{0.8} \end{aligned}$$

An additional source of fuselage drag can be parameterized by the upsweep of the fuselage cone, which is also coupled with the landing gear model. The upsweep angle can be related to the length of the cone and the radius of the fuselage by the tangent of the angle. Unfortunately, trigonometric functions are not SP-compatible, so we must fit it with a function that is. Fortunately, the tangent function is approximately affine in logarithmic space, so it is well approximated by a monomial function, as shown in Figure 3-3. This allows us to use an equality constraint.

$$1.132\phi^{1.038} = \frac{R_{fuse}}{l_{cone}} \tag{3.56}$$

The drag due to upsweep is lower bounded using a relationship given in [31].

$$D_{upsweep} \geq (3.83A_{fuse})\left(\frac{1}{2}\rho_\infty V_\infty^2\right)\phi^{2.5} \tag{3.57}$$

The total drag is lower bounded by the drag due to friction and the pressure drag due to the upsweep.

$$D \geq D_{friction} + D_{upsweep} \tag{3.58}$$

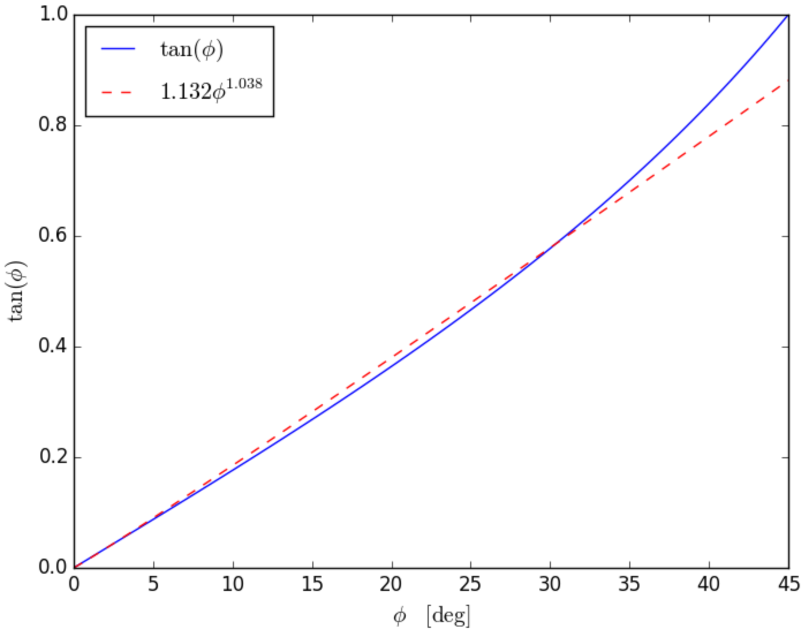

Figure 3-3: A monomial equality constraint closely approximates the tangent function with maximum error less than 6% for all angles up to 30°

3.3 Model Results

By using fixed values representative of the reference aircraft and a placeholder objective function, a solution was obtained for this SP. The placeholder objective is an arbitrary function of drag and weight chosen to reflect the pressure applied by a higher level objective function.

$$0.5W_{fuse} + D_{fuse} \tag{3.59}$$

Solving the SP took 5 GP solves and 0.51 seconds. Even with the arbitrary choice of objective function, there is good agreement between the optimal values and the values for the reference aircraft as can be seen in Table 3.3.

The sensitivity to a selection of fixed variables is presented in Table 3.4 and the convergence of the objective value for the model is show in Figure 3-4.

Table 3.3: Solution comparison with reference aircraft

Design Variable	Solution Value	Estimate for reference aircraft
R_{fuse} [m]	1.86	1.88 [5]
l_{fuse} [m]	39.6	39.1 [5]
W_{fuse} [kg]	15,367	16,300 [16]

Table 3.4: The fuselage model is most sensitive to the number of seats, the seat width, and the load factor.

Fixed Variable	Value	Sensitivity
n_{seat} [−]	186	0.75%
w_{seat} [m]	0.50	0.51%
LF [−]	0.90	0.38%

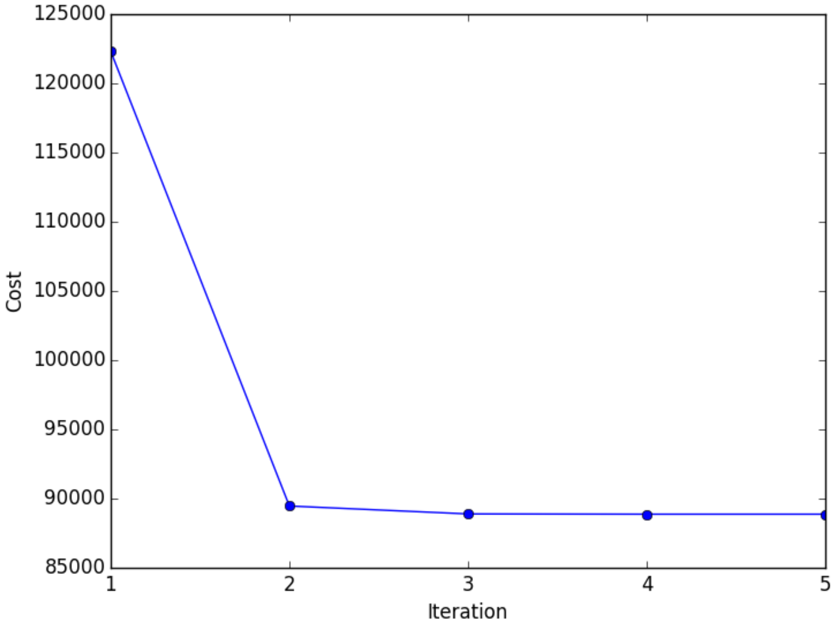

Figure 3-4: The fuselage model solves in 5 GP iterations

Chapter 4

Landing Gear Model

The purpose of the landing gear is to support the weight of the aircraft and allow it to manoeuvre while it is on the ground, including during taxi, takeoff, and landing. The landing gear typically weighs between three and six percent of the maximum aircraft takeoff weight [12]. Many of the constraints imposed on landing gear design are described in [31]. MDO of landing gear is critically important because of how coupled its design is to other subsystems, particularly the fuselage, wings, and engines. Chai [12] proposes a MDO tool for landing gear design, considering the angle of pitch and roll during takeoff and landing, stability at touchdown and during taxi, sideways turnover angle, braking and steering qualities, gear length, landing gear attachment, aircraft turning radius, and centerline-guidance taxiing.

4.1 Model Assumptions

The landing gear model assumes a conventional and retractable tricycle landing gear configuration for narrowbody commerical aircraft such as a Boeing 737 MAX. The nose gear consists of a single strut supported by two wheels. The main gear consists of two struts mounted in the inboard section of the wings, each supported by two wheels. The model only takes one CG location as an input, i.e. it does not consider CG travel. It is also assumed that the main landing gear retracts towards the centerline of the aircraft, rotating about the x axis.

4.2 Model Description

The landing gear model has 44 free variables and 54 constraints.

Table 4.1: Free variables in the landing gear model

Free Variables	Units	Description
B	[m]	Landing gear base
E_{land}	[J]	Max KE to be absorbed in landing
F_{w_m}	[−]	Weight factor (main)
F_{w_n}	[−]	Weight factor (nose)
I_m	[m^4]	Area moment of inertia (main strut)
I_n	[m^4]	Area moment of inertia (nose strut)
L_m	[N]	Max static load through main gear
L_n	[N]	Min static load through nose gear
$L_{n_{dyn}}$	[N]	Dyn. braking load, nose gear
L_{w_m}	[N]	Static load per wheel (main)
L_{w_n}	[N]	Static load per wheel (nose)
S_{sa}	[m]	Stroke of the shock absorber
T	[m]	Main landing gear track
W	[N]	Total aircraft weight
W_{lg}	[N]	Weight of landing gear
W_{mg}	[N]	Weight of main gear
W_{ms}	[N]	Weight of main struts
W_{mw}	[N]	Weight of main wheels (per strut)
W_{ng}	[N]	Weight of nose gear
W_{ns}	[N]	Weight of nose strut
W_{nw}	[N]	Weight of nose wheels (total)
$W_{wa,m}$	[lbf]	Wheel assembly weight for single main gear wheel
$W_{wa,n}$	[lbf]	Wheel assembly weight for single nose gear wheel
Δx_m	[m]	Distance b/w main gear and CG
Δx_n	[m]	Distance b/w nose gear and CG
$\tan(\phi)$	[−]	Angle b/w main gear and CG
$\tan(\psi)$	[−]	Tip over angle
$d_{nacelle}$	[m]	Nacelle diameter
d_{oleo}	[m]	Diameter of oleo shock absorber
d_{t_m}	[in]	Diameter of main gear tires

d_{t_n}	[in]	Diameter of nose gear tires
l_m	[m]	Length of main gear
l_n	[m]	Length of nose gear
l_{oleo}	[m]	Length of oleo shock absorber
r_m	[m]	Radius of main gear struts
r_n	[m]	Radius of nose gear struts
t_m	[m]	Thickness of main gear strut wall
t_n	[m]	Thickness of nose gear strut wall
w_{t_m}	[m]	Width of main tires
w_{t_n}	[m]	Width of nose tires
x_m	[m]	x-location of main gear
x_n	[m]	x-location of nose gear
x_{CG}	[m]	x-location of CG incl. LG
y_m	[m]	y-location of main gear (symmetric)

Table 4.2: Fixed variables in the landing gear model

Constants	Units	Description
E	[GPa]	Modulus of elasticity, 4340 steel
K	[–]	Column effective length factor
N_s	[–]	Factor of safety
$W_{0_{lg}}$	[N]	Weight of aircraft excluding landing gear
η_s	[–]	Shock absorber efficiency
λ_{LG}	[–]	Ratio of max to static load
ρ_{st}	$[\frac{\text{kg}}{\text{m}^3}]$	Density of 4340 Steel
σ_{y_c}	[Pa]	Compressive yield strength 4340 steel
$\tan(\gamma)$	[–]	Dihedral angle
$\tan(\phi_{min})$	[–]	Lower bound on phi
$\tan(\psi_{max})$	[–]	Upper bound on psi
$\tan(\theta_{max})$	[–]	Max rotation angle
d_{fan}	[m]	Fan diameter
$f_{add,m}$	[–]	Proportional added weight, main
$f_{add,n}$	[–]	Proportional added weight, nose
g	$[\frac{\text{m}}{\text{s}^2}]$	Gravitational acceleration
h_{hold}	[m]	Hold height
$h_{nacelle}$	[m]	Min. nacelle clearance

n_{mg}	$[-]$	Number of main gear struts
n_{wps}	$[-]$	Number of wheels per strut
p_{oleo}	$[\frac{\mathrm{lbf}}{\mathrm{in}^2}]$	Oleo pressure
$t_{nacelle}$	[m]	Nacelle thickness
w_{ult}	$[\frac{\mathrm{ft}}{\mathrm{s}}]$	Ultimate velocity of descent
x_{CG_0}	[m]	x-location of CG excl. LG
x_{up}	[m]	Fuselage upsweep point
y_{eng}	[m]	Spanwise loc. of engines
z_{CG}	[m]	CG height relative to bottom of fuselage
z_{wing}	[m]	Height of wing relative to base of fuselage

The difference between the lengths of the main gear and nose gear is constrained by the vertical position of the wing with respect to the bottom of the fuselage, as well as the spanwise location of the main gear and the wing dihedral. This relationship is a signomial constraint.

$$l_n + z_{wing} + y_m \tan(\gamma) \geq l_m \tag{4.1}$$

The landing gear track and base are defined relative to the x- and y-coordinates of the nose and main gear.

$$T = 2y_m \tag{4.2}$$

$$x_m \geq x_n + B \tag{4.3}$$

The geometric relationships between the x-coordinates of the main gear, nose gear and the CG position must be enforced. These relationships are:

$$x_n + \Delta x_n = x_{CG} \tag{4.4}$$

$$x_{CG} + \Delta x_m = x_m \tag{4.5}$$

Because the definition of a signomial program (1.6) does not allow posynomial equality constraints, we must relax these to be inequality constraints. However, (4.4) and (4.5) must be satisfied exactly, meaning the constraints that enforce them must be tight. For each relationship, we have a choice between a posynomial inequality constraint

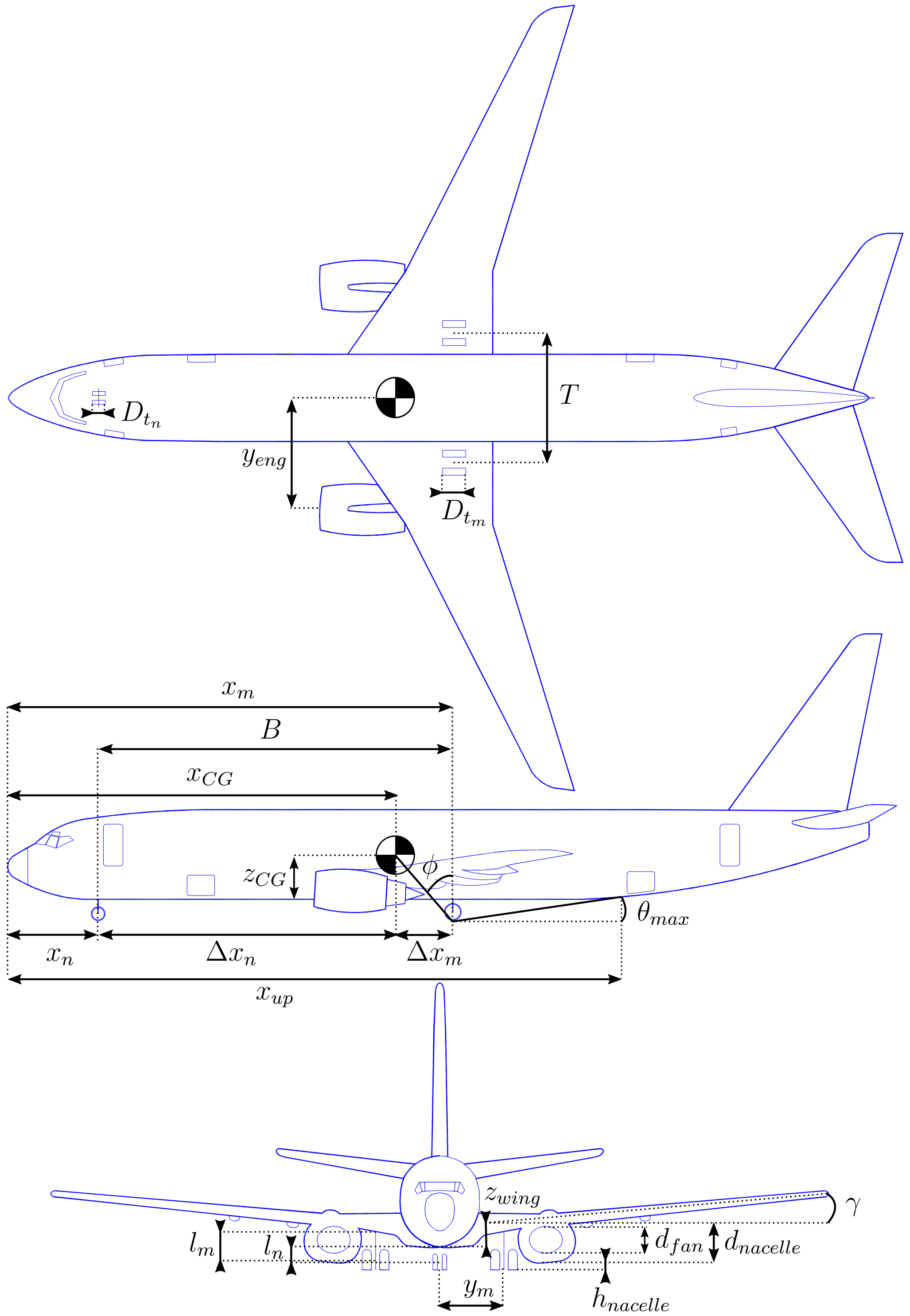

Figure 4-1: Geometric variables of the landing gear model (adapted from [4])

and a signomial inequality constraint. As will be shown below, the load through the nose gear and main gear is proportional to the distance from the CG to the main and nose gear respectively. Because there is downward pressure on these loads - more load generally means heavier landing gear - there is also downward pressure on the distances Δx_n and Δx_m. We must therefore use a signomial constraint for both relationships.

$$x_n + \Delta x_n \geq x_{CG} \tag{4.6}$$

$$x_{CG} + \Delta x_m \geq x_m \tag{4.7}$$

The maximum static load through the nose and main gear is constrained by the weight of the aircraft and the relative distances from the CG to the main and nose gear, respectively.

$$L_n = \frac{W \Delta x_m}{B} \tag{4.8}$$

$$L_m = \frac{W \Delta x_n}{B} \tag{4.9}$$

For the nose gear, there is an additional dynamic load due to the braking condition. A typical braking deceleration of $3m/s^2$ is assumed [31].

$$L_{n_{dyn}} \geq 0.31 W \frac{l_m + z_{CG}}{B} \tag{4.10}$$

The nose gear requires adequate load for satisfactory steering performance. A typical desirable range is between 5% and 20% of the total load [31].

$$\frac{L_n}{W} \geq 0.05 \tag{4.11}$$

$$\frac{L_n}{W} \leq 0.2 \tag{4.12}$$

A longitudinal tip over criterion requires that the line between the main gear and the CG be at least 15° relative to the vertical such that the aircraft will not tip back on its tail at a maximum nose-up attitude [31]. This puts a lower bound on the x-location of the main gear, as measured from the nose of the aircraft. Note that $\tan(\phi)$ is a design variable here, instead of ϕ, to make the constraint GP-compatible.

$$x_m \geq (l_m + z_{CG})\tan(\phi) + x_{CG} \tag{4.13}$$

$$\tan(\phi) \geq \tan(\phi_{min}) \tag{4.14}$$

A lateral tip over constraint is introduced to ensure that an aircraft does not tip over in a turn [12]. The turnover angle is defined as

$$\tan\psi = \frac{z_{CG} + l_m}{\Delta x_n \sin\delta} \tag{4.15}$$

where

$$\tan\delta = \frac{y_m}{B}. \tag{4.16}$$

Using the relationship

$$\cos\left(\arctan\left(\frac{y_m}{B}\right)\right) = \frac{B}{\sqrt{B^2 + y_m^2}}, \tag{4.17}$$

this constraint can, perhaps surprisingly, be rewritten in, not only SP-compatible, but GP-compatible form as

$$1 \geq \frac{(z_{CG} + l_m)^2({y_m}^2 + B^2)}{(\Delta x_n y_m \tan(\psi))^2} \tag{4.18}$$

Typically this angle, ψ, should be no larger than 63° [31].

$$\tan(\psi) \leq \tan(\psi_{max}) \tag{4.19}$$

The aircraft must be able to rotate on its main wheels at takeoff without striking the tail of the fuselage and, similarly, must be able to land on its main gear without striking the tail [31]. This constrains the location of the main gear. More specifically, the horizontal distance between the main gear and the point at which the fuselage sweeps up towards the tail must be sufficiently small, relative to the length of the main gear, such that the angle relative to the horizontal from the main wheels to the upsweep point is greater than the takeoff/landing angles. The result is a signomial constraint that imposes a lower bound on the length of the gear and the x-location of the main gear.

$$\frac{l_m}{\tan(\theta_{max})} \geq x_{up} - x_m \tag{4.20}$$

The length of the main gear is also constrained by the engine diameter, because the engines must have sufficient clearance from the ground. A signomial constraint provides another lower bound on the length of the main gear.

$$l_m + (y_{eng} - y_m)\tan(\gamma) \geq d_{nacelle} + h_{nacelle} \tag{4.21}$$

$$d_{nacelle} \geq d_{fan} + 2t_{nacelle} \tag{4.22}$$

The main gear position in the spanwise (y) direction is, on one side, lower bounded by the length of the gear itself and, on the other side, upper bounded by the spanwise location of the engines. Both of these constraints are necessary to allow the landing gear to retract in the conventional manner for typical narrowbody commercial aircraft.

$$y_m \geq l_m \tag{4.23}$$

$$y_m \leq y_{eng} \tag{4.24}$$

Oleo-penumatic shock absorbers are common to landing gear for large aircraft. Their purpose is to reduce the vertical load on the aircraft at touchdown, and they are typically sized by a hard landing condition. The maximum stroke of the shock absorber can be determined by considering the aircraft's kinetic energy, and the target maximum load [33].

$$E_{land} = \frac{W}{2g} w_{ult}^2 \tag{4.25}$$

$$S_{sa} = \frac{1}{\eta_s} \frac{E_{land}}{L_m \lambda_{LG}} \tag{4.26}$$

As a preliminary model, the oleo size can be estimated using historical relations that are conveniently in monomial form [31]. The length of the main gear must be greater than the length of the oleo and the radius of the tyres.

$$l_{oleo} = 2.5 S_{sa} \tag{4.27}$$

$$d_{oleo} = 1.3 \sqrt{\frac{4 \lambda_{LG} L_m / n_{mg}}{p_{oleo} \pi}} \tag{4.28}$$

$$l_m \geq l_{oleo} + \frac{d_{t_m}}{2} \tag{4.29}$$

The wheel weights and sizes can be estimated using historical relations from [13, 31], which are, again, conveniently in monomial form. The nose gear tyres are assumed to be 80% of the size of the main gear tyres.

$$F_{wm} = L_{w_m} d_{t_m} \tag{4.30}$$

$$W_{wa,m} = 1.2 F_{w_m}^{0.609} \tag{4.31}$$

$$L_{w_m} = \frac{L_m}{n_{mg} n_{wps}} \tag{4.32}$$

$$F_{wn} = L_{w_n} d_{t_n} \tag{4.33}$$

$$W_{wa,n} = 1.2 F_{w_n}^{0.609} \tag{4.34}$$

$$L_{w_n} = \frac{L_n}{n_{wps}} \tag{4.35}$$

$$d_{t_m} = 1.63 L_{w_m}^{0.315} \tag{4.36}$$

$$w_{t_m} = 0.104 L_{w_m}^{0.480} \tag{4.37}$$

$$d_{t_n} = 0.8 d_{t_m} \tag{4.38}$$

$$w_{t_n} = 0.8 w_{t_m} \tag{4.39}$$

$$W_{mw} = n_{wps} W_{wa,m} \tag{4.40}$$

$$W_{nw} = n_{wps} W_{wa,n} \tag{4.41}$$

The weight of each strut for both the main and nose struts is lower bounded by assuming a thin walled cylinder with constant cross sectional area.

$$W_{ms} \geq 2\pi r_m t_m l_m \rho_{st} g \tag{4.42}$$

$$W_{ns} \geq 2\pi r_n t_n l_n \rho_{st} g \tag{4.43}$$

The cross-sectional area is itself constrained by the compressive yield of the landing gear material and the maximum load exerted on each strut.

$$2\pi r_m t_m \sigma_{y_c} \geq \frac{\lambda_{LG} L_m N_s}{n_{mg}} \tag{4.44}$$

$$2\pi r_n t_n \sigma_{y_c} \geq (L_n + L_{n_{dyn}}) N_s \tag{4.45}$$

Another, typically more restrictive, structural constraint ensures the struts will not buckle. This constrains the area moment of inertia of the strut cross section, which in turn puts upward pressure on the radius and thickness of the struts. Again, a thin walled cylinder approximation is made. The buckling constraint assumes that no side force is exerted on the cylinder, which is perhaps a weak assumption due to forces exerted in braking, for example, and due to the fact that aircraft do not typically land with the main gear struts perfectly normal to the runway surface.

$$L_m \leq \frac{\pi^2 E I_m}{K^2 {l_m}^2} \tag{4.46}$$

$$I_m = \pi {r_m}^3 t_m \tag{4.47}$$

$$L_n \leq \frac{\pi^2 E I_n}{K^2 {l_n}^2} \tag{4.48}$$

$$I_n = \pi {r_n}^3 t_n \tag{4.49}$$

A machining constraint is used to ensure that the strut walls are not too thin to be fabricated [12].

$$\frac{2r_m}{t_m} \leq 40 \tag{4.50}$$

$$\frac{2r_n}{t_n} \leq 40 \tag{4.51}$$

In addition, simple retraction space constraints are used to ensure that the struts are not too wide for the gear to fit inside the fuselage.

$$2w_{t_m} + 2r_m \leq h_{hold} \tag{4.52}$$

$$2w_{t_n} + 2r_n \leq 0.8 \text{ [m]} \tag{4.53}$$

Finally, the total landing gear system weight is lower bounded by accounting for the weights of each assembly. We use an additional weight fraction to account for weight that is proportional to the weight of the wheels [13].

$$W_{mg} \geq n_{mg} \left(W_{ms} + W_{mw}(1 + f_{add_m}) \right) \tag{4.54}$$

$$W_{ng} \geq W_{ns} + W_{nw}(1 + f_{add_n}) \tag{4.55}$$

$$W_{lg} \geq W_{mg} + W_{ng} \tag{4.56}$$

Of course, the aircraft weight and the CG location are both also affected by the weight and position of the landing gear, and this adds two more constraints, one posynomial and one signomial.

$$W \geq W_{0_{lg}} + W_{lg} \tag{4.57}$$

$$W x_{CG} \leq W_{0_{lg}} x_{CG_0} + W_{ng} x_n + W_{mg} x_m \tag{4.58}$$

4.3 Model Results

Using fixed values representative of the reference aircraft and a placeholder objective function, a solution was obtained for this SP. The placeholder objective function for this model is simply the total landing gear weight, W_{lg}. This is based on the fact that having heavy landing gear directly impacts fuel consumption by increasing the empty weight of the aircraft.

Solving the SP takes 7 GP solves and 1.05 seconds. The result shows good agreement with estimates for the reference aircraft, particularly with geometric quantities such as the gear track and main wheel diameter. A selection of the optimal values are presented in Table 4.3

Table 4.3: Solution comparison with reference aircraft

Free Variable	Solution Value	Estimate for reference aircraft [8]
B [m]	13.4	15.6
T [m]	5.7	5.8
d_{t_m} [in]	44.8	44.5

The sensitivity to a selection of fixed variables is presented in Table 3.4. As can be seen, the landing gear model is sensitive to the proportional added weight fraction, which suggests that there is value in developing a more sophisticated model to replace this low fidelity approach to modelling landing gear weight.

Table 4.4: The landing gear model is most sensitive to the aircraft weight, the proportional added weight fraction of the main gear, and the number of wheels per strut.

Fixed Variable	Value	Sensitivity
$W_{0_{lg}}$ [N]	8.04e+05	0.82%
$f_{add,m}$ [−]	1.50	0.49%
n_{wps} [−]	2	0.18%

The convergence of the objective value for the model is show in Figure 4-2. No objective value is given for the first three solves, because these are all feasibility finding solves, for which the objective value is a slack variable that does not correspond to the objective function of the model.

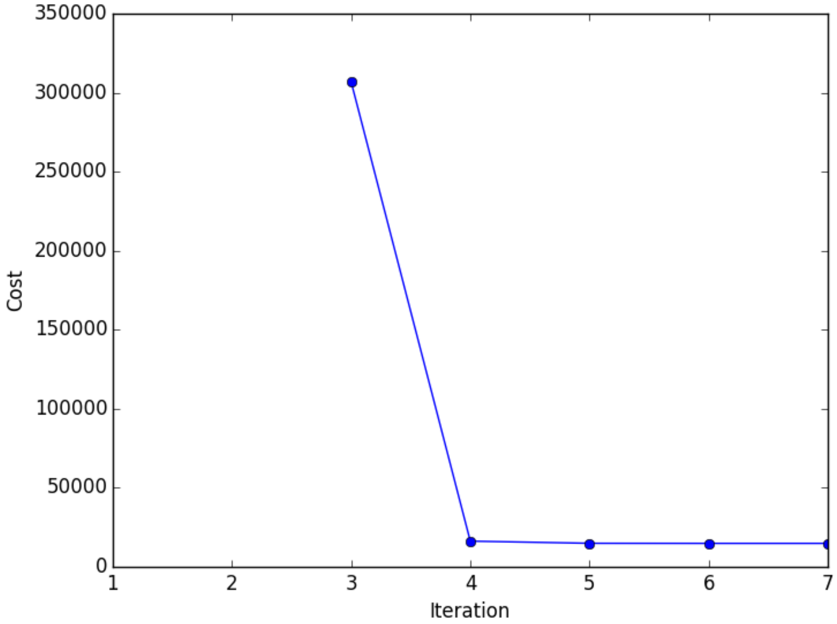

Figure 4-2: The landing gear model solves in 7 GP iterations

Chapter 5

Horizontal Tail Model

At a conceptual design level, the purpose of the horizontal tail is threefold: to trim the aircraft such that it can fly in steady level flight, to provide longitudinal stability, and to give the pilot control authority over a range of flight conditions.

5.1 Model Assumptions

The horizontal tail model assumes a conventional tail configuration with a low horizontal tail. As with the landing gear model, a single fixed CG location is assumed.

5.2 Model Description

The horizontal tail model has 38 free variables and 40 constraints. It shares many constraints with the vertical tail model, including the structural model from [21], basic aerodynamic constraints, and constraints that enforce geometric relationships. To avoid duplication, we do not repeat any constraints here, but we do describe the constraints that distinguish the two models.

Table 5.1: Free variables in the horizontal tail model

Free Variables	Units	Description
AR_h	$[-]$	Horizontal tail aspect ratio
C_{D_h}	$[-]$	Horizontal tail drag coefficient
$C_{D_{0_h}}$	$[-]$	Horizontal tail parasitic drag coefficient
C_{L_h}	$[-]$	Lift coefficient (htail)
$C_{L_{ah_0}}$	$[-]$	Isolated lift curve slope (htail)
$C_{L_{ah}}$	$[-]$	Lift curve slope (htail)
D_{ht}	[N]	Horizontal tail drag
K_f	$[-]$	Empirical factor for fuselage-wing interference
L_h	[N]	Horizontal tail downforce
L_{max_h}	[N]	Maximum load
Re_{c_h}	$[-]$	Cruise Reynolds number (Horizontal tail)
$S.M.$	$[-]$	Stability margin
S_h	$[\mathrm{m}^2]$	Horizontal tail area
V_∞	$[\frac{\mathrm{m}}{\mathrm{s}}]$	Freestream velocity
W_{ht}	[N]	Horizontal tail weight
Δx_{lead_h}	[m]	Distance from CG to horizontal tail leading edge
Δx_{trail_h}	[m]	Distance from CG to horizontal tail trailing edge
α	$[-]$	Horizontal tail angle of attack
$\bar{c}_{ht}$	[m]	Mean aerodynamic chord (ht)
λ_h	$[-]$	Horizontal tail taper ratio
τ_h	$[-]$	Horizontal tail thickness/chord ratio
b_{ht}	[m]	Horizontal tail span
c_{root_h}	[m]	Horizontal tail root chord
c_{tip_h}	[m]	Horizontal tail tip chord
e_h	$[-]$	Oswald efficiency factor
$f(\lambda_h)$	$[-]$	Empirical efficiency function of taper
l_{ht}	[m]	Horizontal tail moment arm
p_{ht}	$[-]$	Substituted variable = 1 + 2*taper
q_{ht}	$[-]$	Substituted variable = 1 + taper
x_w	[m]	Position of wing aerodynamic center
$y_{\bar{c}_{ht}}$	[m]	Spanwise location of mean aerodynamic chord

WingBox		
I_{cap}	[−]	Non-dim spar cap area moment of inertia
M_r	[N]	Root moment per root chord
W_{cap}	[N]	Weight of spar caps
W_{web}	[N]	Weight of shear web
ν	[−]	Dummy variable $= (t^2 + t + 1)/(t + 1)$
t_{cap}	[−]	Non-dim. spar cap thickness
t_{web}	[−]	Non-dim. shear web thickness

Table 5.2: Fixed variables in the horizontal tail model

Constants	Units	Description
AR_w	[−]	Wing aspect ratio
C_{L_w}	[−]	Lift coefficient (wing)
$C_{L_{aw}}$	[−]	Lift curve slope (wing)
$C_{L_{hmax}}$	[−]	Max lift coefficient
$C_{m_{fuse}}$	[−]	Moment coefficient (fuselage)
M	[−]	Cruise Mach number
$S.M._{min}$	[−]	Minimum stability margin
S_w	$[\mathrm{m}^2]$	Wing area
V_{ne}	$[\frac{\mathrm{m}}{\mathrm{s}}]$	Never exceed velocity
Δx_w	[m]	Distance from aerodynamic centre to CG
$\alpha_{max,h}$	[−]	Max angle of attack, htail
$\bar{c}_w$	[m]	Mean aerodynamic chord (wing)
η_h	[−]	Lift efficiency (diff between sectional and actual lift)
η_{ht}	[−]	Tail efficiency
μ	$[\frac{\mathrm{N \cdot s}}{\mathrm{m}^2}]$	Dynamic viscosity (35,000 ft)
ρ	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Air density (35,000 ft)
ρ_0	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Air density (0 ft)
$\tan(\Lambda_{ht})$	[−]	tangent of horizontal tail sweep
a	$[\frac{\mathrm{m}}{\mathrm{s}}]$	Speed of sound (35,000 ft)
l_{fuse}	[m]	Fuselage length
w_{fuse}	[m]	Fuselage width
x_{CG}	[m]	CG location
$\lvert C_{m_{ac}} \rvert$	[−]	Moment coefficient about aerodynamic centre (wing)

WingBox		
N_{lift}	$[-]$	Wing loading multiplier
ρ_{cap}	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Density of spar cap material
ρ_{web}	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Density of shear web material
$\sigma_{max,shear}$	[Pa]	Allowable shear stress
σ_{max}	[Pa]	Allowable tensile stress
$f_{w,add}$	$[-]$	Wing added weight fraction
g	$[\frac{\mathrm{m}}{\mathrm{s}^2}]$	Gravitational acceleration
r_h	$[-]$	Fractional wing thickness at spar web
w	$[-]$	Wingbox-width-to-chord ratio

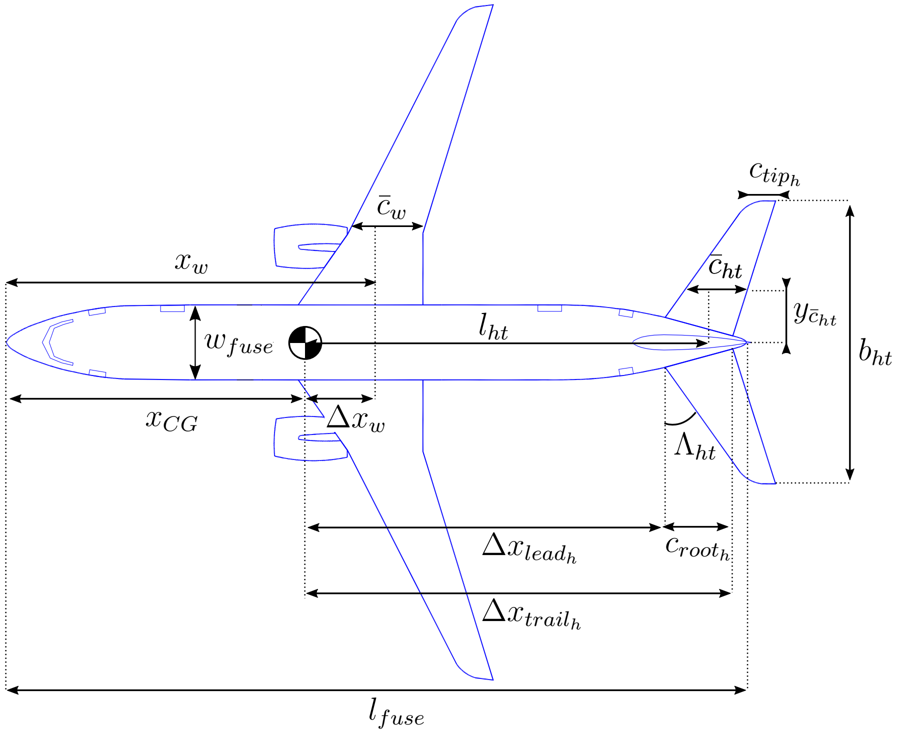

Figure 5-1: Geometric variables of the horizontal tail model (adapted from [4])

The most significant difference is that the vertical tail's engine out constraint is replaced by two sizing constraints for trim and stability. The trim condition requires that the full aircraft moment coefficient, C_m, be zero.

$$C_m = C_{m_{a.c.}} + C_{L_w}\frac{\Delta x_w}{\bar{c}_w} - C_{L_h}\frac{S_h}{S_w}\frac{l_h}{\bar{c}_w} + C_{m_{fu}} = 0$$

Depending on whether $C_{m_{fu}}$ is positive or negative, the resulting constraint is either a posynomial or a signomial constraint.

$$C_{m_{a.c.}} + C_{L_w}\frac{\Delta x_w}{\bar{c}_w} + C_{m_{fu}} \leq C_{L_h}\frac{S_h}{S_w}\frac{l_h}{\bar{c}_w} \tag{5.1}$$

From [25], an equation for longitudinal stability is given by

$$\frac{\Delta x_w}{\bar{c}_w} - \frac{C_{L_{\alpha_h}} S_h l_h}{C_{L_{\alpha_w}} S_w \bar{c}_w} + \frac{K_f w_{fuse}^2 l_{fuse}}{C_{L_{\alpha_w}} S_w \bar{c}_w} \approx -S.M.$$

We can use this relationship as a posynomial inequality constraint in combination with a minimum requirement on the stability margin.

$$S.M. + \frac{\Delta x_w}{\bar{c}_w} + \frac{K_f w_{fuse}^2 l_{fuse}}{C_{L_{\alpha_w}} S_w \bar{c}_w} \leq \frac{C_{L_{\alpha_h}} S_h l_h}{C_{L_{\alpha_w}} S_w \bar{c}_w} \tag{5.2}$$

$$S.M. \geq 0.05 \tag{5.3}$$

This stability constraint depends on the lift curve slopes of both the wing and the horizontal tail. The DATCOM formula [25] is an analytic function for estimating the lift curve slope of a wing or tail, based on empirical results.

$$C_{L_\alpha} = \frac{2\pi AR}{2 + \sqrt{(AR/\eta)^2(1 + \tan^2\Lambda - M^2) + 4}} \tag{5.4}$$

This relationship can be rearranged into a signomial inequality to constrain the lift curve slope, although some algebraic manipulation is needed to see this.

$$\begin{aligned} C_{L_\alpha} &\leq \frac{2\pi AR}{2 + \sqrt{(AR/\eta)^2(1 + \tan^2\Lambda - M^2) + 4}} \\ (AR/\eta)^2(1 + \tan^2\Lambda - M^2) + 4 &\leq \left(\frac{2\pi AR}{C_{L_\alpha}} - 2\right)^2 \\ (AR/\eta)^2(1 + \tan^2\Lambda - M^2) &\leq \frac{4\pi^2 AR^2}{C_{L_\alpha}^2} - \frac{8\pi AR}{C_{L_\alpha}} \\ \frac{C_{L_\alpha}^2}{\eta^2}\left(1 + \tan^2\Lambda - M^2\right) + \frac{8\pi C_{L_\alpha}}{AR} &\leq 4\pi^2 \end{aligned} \tag{5.5}$$

The horizontal tail's lift curve slope is reduce by downwash, ϵ, due to the wing and fuselage [25].

$$C_{L_{\alpha h}} = C_{L_{\alpha h_0}} \left(1 - \frac{\partial \epsilon}{\partial \alpha}\right) \eta_h \tag{5.6}$$

The downwash can be approximated by the downwash far behind an elliptically loaded wing.

$$\epsilon \approx \frac{2 C_{L_w}}{\pi \text{Æ}_w} \tag{5.7}$$

$$\implies \frac{\partial \epsilon}{\partial \alpha} \approx \frac{2 C_{L_{\alpha w}}}{\pi \text{Æ}_w} \tag{5.8}$$

We therefore use (5.5) to constrain the isolated lift curve slope, and introduce an additional posynomial constraint on the corrected lift curve slope.

$$C_{L_{\alpha h}} + \frac{2 C_{L_{a w}}}{\pi \text{Æ}_w} \eta_{ht} C_{L_{\alpha h_0}} \leq C_{L_{\alpha h_0}} \eta_{ht} \tag{5.9}$$

As can be seen in Equation 5.2, the fuselage also affects the stability constraint. The empirical factor, K_f, depends on the position of the wing quarter-chord on the fuselage as a fraction of fuselage length. This relationship is captured using a GP-compatible fit of data [25].

$$K_f \geq 1.5012 \left(\frac{x_w}{l_{fuse}}\right)^2 + 0.538 \left(\frac{x_w}{l_{fuse}}\right) + 0.0331 \tag{5.10}$$

5.3 Model Results

By using fixed values representative of the reference aircraft and a placeholder objective function, a solution was obtained for this SP. The placeholder objective function is a somewhat arbitrary function of cruise drag and weight, and is intended to reflect how a higher level objective might apply pressure to the horizontal tail model.

$$0.1 W_{ht} + D_{ht} \tag{5.11}$$

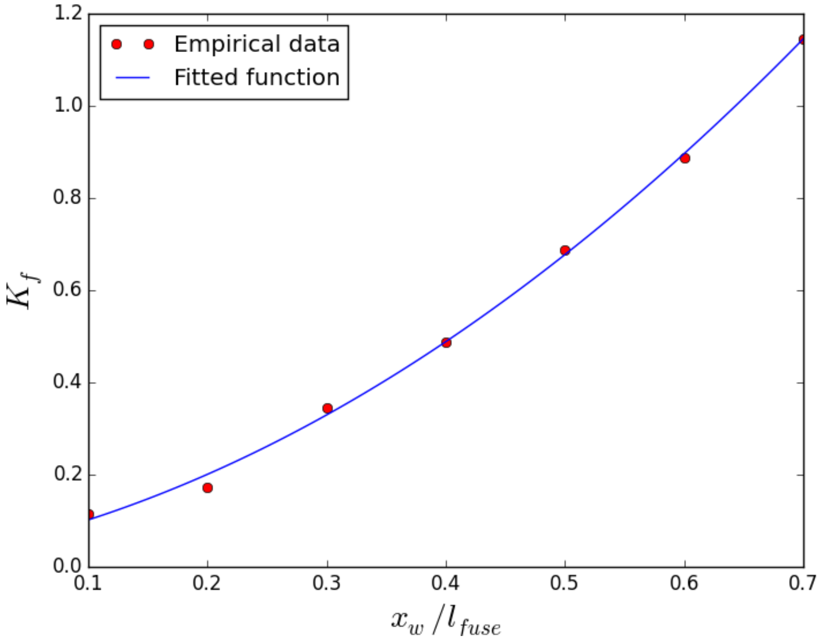

Figure 5-2: GP-compatible fit of K_f data for estimating the effect of wing position on longitudinal stability

The standalone model takes 5 GP solves and 0.53 seconds to solve and the optimal values for a selection of key design variables are presented in Table 5.3.

Table 5.3: Solution comparison with reference aircraft

Free Variable	Solution Value	Estimate for reference aircraft [5]
$Æ_{ht}$ [−]	6.2	6.2
b_{ht} [m]	14.9	14.4
S_{ht} [m^2]	35.8	32.8

The sensitivity to a selection of fixed variables is presented in Table 5.4 and the convergence of the objective value for the model is show in Figure 5-3.

Table 5.4: The horizontal tail model is most sensitive to the fuselage length, tail efficiency factor, and fuselage width.

Fixed Variable	Value	Sensitivity
l_{fuse} [m]	40	-1.5%
η_{ht} [−]	0.9	-1.1%
w_{fuse} [m]	6.0	0.8%

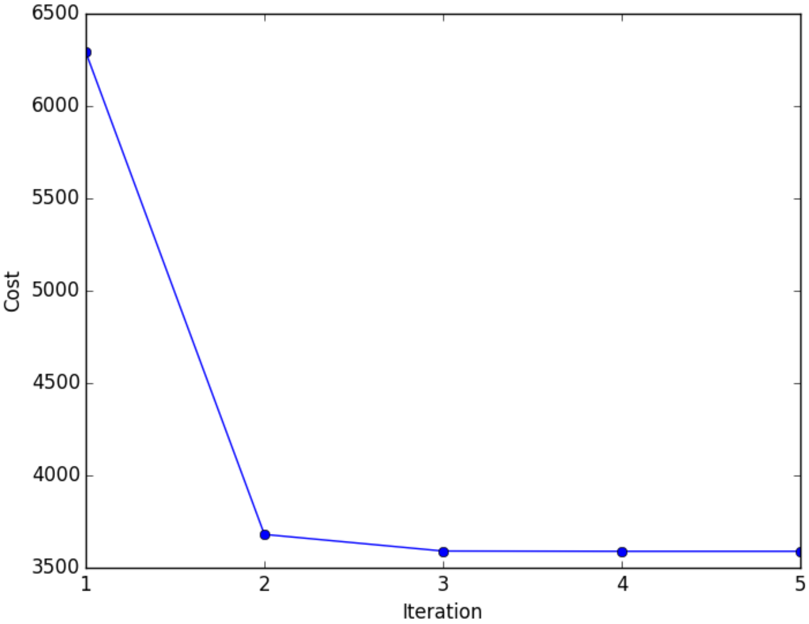

Figure 5-3: The horizontal tail model solves in 5 GP iterations

Chapter 6

Wing Model

The overarching purpose of a commercial aircraft wing is to generate sufficient lift such that the aircraft can takeoff, climb, cruise, descend, and land safely. In conventional commercial aircraft, the wings also carry the fuel tanks and the engines.

6.1 Model Assumptions

The wing model assumes a conventional single low-wing configuration. It does not currently consider wing twist, wing dihedral, or the weight of the engines. It also does not consider roll or yaw stability, or any control surfaces.

6.2 Model Description

The wing model has 34 free variables and 36 constraints. The majority of these constraints are also constraints in the vertical tail and/or horizontal tail models, including the structural model from [21], basic aerodynamic constraints, and constraints that enforce geometric relationships.

Table 6.1: Free variables in the wing model

Free Variables	Units	Description
AR_w	$[-]$	Wing aspect ratio
C_{D_w}	$[-]$	Drag coefficient, wing
$C_{D_{pw}}$	$[-]$	Wing parasitic drag coefficient
C_{L_w}	$[-]$	Lift coefficient, wing
$C_{L_{aw}}$	$[-]$	Lift curve slope, wing
D_{wing}	[N]	Wing drag
L_w	[N]	Wing lift
L_{max_w}	[N]	Maximum load
Re_w	$[-]$	Cruise Reynolds number (wing)
S_w	$[\mathrm{m}^2]$	Wing area
V_∞	$[\frac{\mathrm{m}}{\mathrm{s}}]$	Freestream velocity
$V_{fuel,max}$	$[\mathrm{m}^3]$	Available fuel volume
W	[N]	Aircraft weight
W_{wing}	[N]	Wing weight
α_w	$[-]$	Wing angle of attack
$\bar{A}_{fuel,max}$	$[-]$	Non-dim. fuel area
$\bar{c}_w$	[m]	Mean aerodynamic chord (wing)
λ	$[-]$	Wing taper ratio
τ_w	$[-]$	Wing thickness/chord ratio
b_w	[m]	Wing span
c_{root}	[m]	Wing root chord
c_{tip}	[m]	Wing tip chord
e	$[-]$	Oswald efficiency factor
$f(\lambda_w)$	$[-]$	Empirical efficiency function of taper
p_w	$[-]$	Substituted variable = 1 + 2*taper
q_w	$[-]$	Substituted variable = 1 + taper
$y_{\bar{c}_w}$	[m]	Spanwise location of mean aerodynamic chord
WingBox		
I_{cap}	$[-]$	Non-dim spar cap area moment of inertia
M_r	[N]	Root moment per root chord
W_{cap}	[N]	Weight of spar caps

W_{web}	[N]	Weight of shear web
ν	$[-]$	Dummy variable $= (t^2 + t + 1)/(t + 1)$
t_{cap}	$[-]$	Non-dim. spar cap thickness
t_{web}	$[-]$	Non-dim. shear web thickness

Table 6.2: Fixed variables in the wing model

Constants	Units	Description
$C_{L_{wmax}}$	$[-]$	Max lift coefficient, wing
M	$[-]$	Cruise Mach number
V_{ne}	$[\frac{\text{m}}{\text{s}}]$	Never exceed velocity
W_0	[N]	Weight excluding wing
W_{fuel}	[N]	Fuel weight
$\alpha_{max,w}$	$[-]$	Max angle of attack
$\cos(\Lambda)$	$[-]$	Cosine of quarter-chord sweep angle
η_w	$[-]$	Lift efficiency (diff b/w sectional, actual lift)
μ	$[\frac{\text{N}\cdot\text{s}}{\text{m}^2}]$	Dynamic viscosity (35,000 ft)
ρ	$[\frac{\text{kg}}{\text{m}^3}]$	Air density (35,000 ft)
ρ_0	$[\frac{\text{kg}}{\text{m}^3}]$	Air density (0 ft)
ρ_{fuel}	$[\frac{\text{kg}}{\text{m}^3}]$	Density of fuel
$\tan(\Lambda)$	$[-]$	Tangent of quarter-chord sweep angle
a	$[\frac{\text{m}}{\text{s}}]$	Speed of sound (35,000 ft)
g	$[\frac{\text{m}}{\text{s}^2}]$	Gravitational acceleration
w	$[-]$	Wingbox-width-to-chord ratio
WingBox		
N_{lift}	$[-]$	Wing loading multiplier
ρ_{cap}	$[\frac{\text{kg}}{\text{m}^3}]$	Density of spar cap material
ρ_{web}	$[\frac{\text{kg}}{\text{m}^3}]$	Density of shear web material
$\sigma_{max,shear}$	[Pa]	Allowable shear stress
σ_{max}	[Pa]	Allowable tensile stress
$f_{w,add}$	$[-]$	Wing added weight fraction
g	$[\frac{\text{m}}{\text{s}^2}]$	Gravitational acceleration
r_h	$[-]$	Fractional wing thickness at spar web
w	$[-]$	Wingbox-width-to-chord ratio

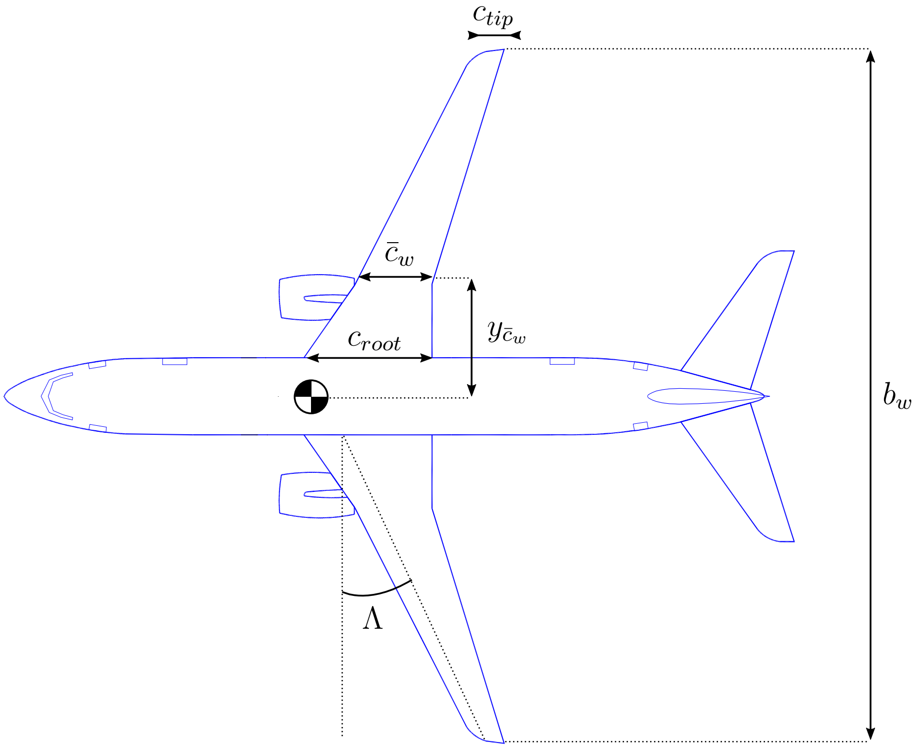

Figure 6-1: Geometric variables of the wing model (adapted from [4])

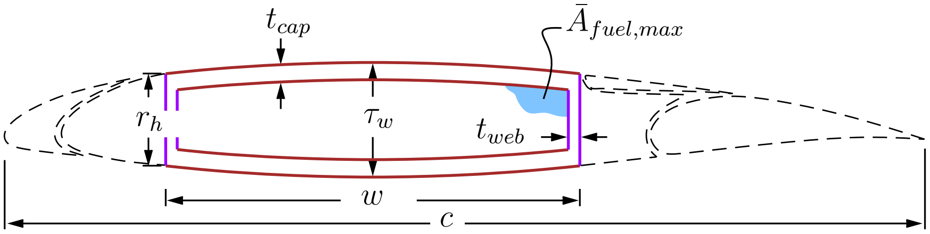

Figure 6-2: Geometric variables of the wingbox cross-section (adapted from [16])

One model that was not presented in previous chapters is a model for Oswald efficiency, which has a large impact on the drag of a wing. The Oswald efficiency is constrained by a relationship from [29], in which the authors fit a polynomial function to empirical data. Given that all polynomials are signomials, this can easily be used in the SP framework.

$$e \leq \frac{1}{1 + f(\lambda) Æ_w} \tag{6.1}$$

$$f(\lambda) \geq 0.0524\lambda^4 - 0.15\lambda^3 + 0.1659\lambda^2 - 0.0706\lambda + 0.0119 \tag{6.2}$$

An analogous pair of constraints is also used as part of the horizontal tail model.

The fuel tanks are typically located inside the wingbox. The wingbox therefore needs

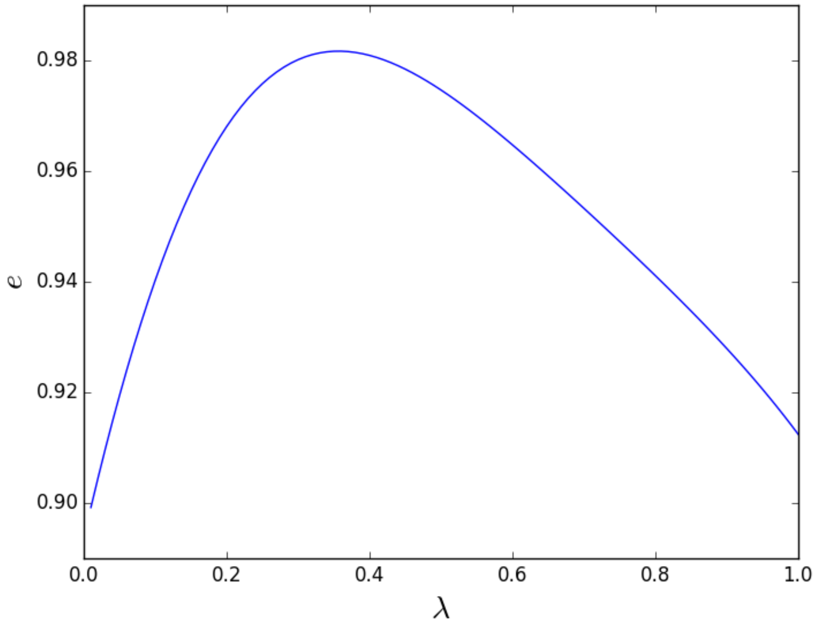

Figure 6-3: Empirical relationship for Oswald efficiency as a function of taper for a wing with Æ=10

to be large enough to carry the required fuel for a given mission, a quantity which itself depends on the wing geometry. To ensure the fuel volume does not exceed the available wingbox volume, three relationships from [16] are adapted as constraints, including one signomial constraint.

$$\bar{A}_{fuel,max} \leq 0.92 w\tau, \tag{6.3}$$

$$V_{fuel,max} \leq c_{root}^2 \frac{b_w}{6}(1+\lambda+\lambda^2)\cos\Lambda \tag{6.4}$$

$$W_{fuel} \leq \rho_{fuel}\bar{A}_{fuel,max}V_{fuel,max}g \tag{6.5}$$

A model for parasitic drag as a function of lift coefficient, Reynolds number, and airfoil thickness is used from [21]. The model is an Implicit Softmax Affine (ISMA) fit [22] of data from XFOIL [15] for a series of NACA 24XX airfoils and has an RMS error of approximately 2%.

$$1 \geq 0.0022\frac{Re_w^{0.14}\tau_w^{0.033}}{C_{D_{pw}}^{0.73}C_{L_w}^{0.01}} + 1.19\times 10^4\frac{C_{L_w}^{9.8}\tau_w^{1.8}}{C_{D_{pw}}^{0.91}Re_w} + 2.56\frac{C_{L_w}^{5.9}}{C_{D_{pw}}^{2.6}Re_w^{1.5}\tau_w^{3.3}}$$

$$+ 3.8 \times 10^{-9} \frac{\tau_w^{6.2}}{C_{D_{pw}}^{9.6} C_{L_w}^{0.92} Re_w^{1.4}} + 6.14 \times 10^{-6} \frac{C_{L_w}^{6.5}}{C_{D_{pw}}^{5.2} Re_w^{0.99} \tau_w^{0.52}} \tag{6.6}$$

Finally, the wing is tasked with lifting the weight of the aircraft, however, this quantity also depends on the weight of the wing. This recursive relationship is captured in a pair of simple constraints.

$$L_w = W \tag{6.7}$$

$$W \geq W_0 + W_w + W_{fuel} \tag{6.8}$$

6.3 Model Results

By using fixed values representative of the reference aircraft and a simple objective of minimizing drag, a solution was obtained for this SP. The standalone model takes 5 GP solves and 0.55 seconds to find a solution and the optimal values for a selection of key design variables are presented in Table 6.3.

Table 6.3: Solution comparison with reference aircraft

Free Variable	Solution Value	Estimate for reference aircraft [8]
$Æ_w$ [−]	9.3	9.5
b_w [m]	36.3	35.9
c_{root} [m]	6.5	7.8
c_{tip} [m]	1.3	1.3

The sensitivity to a selection of fixed variables is presented in Table 6.4 and the convergence of the objective value for the model is show in Figure 6-4.

Table 6.4: The wing model is most sensitive to cruise Mach number, aircraft weight, and the maximum dynamic pressure never-exceed speed.

Fixed Variable	Value	Sensitivity
M [−]	0.8	-1.5%
W_0 [N]	5e+05	1.0%
V_{ne} [m/s]	144	0.99%

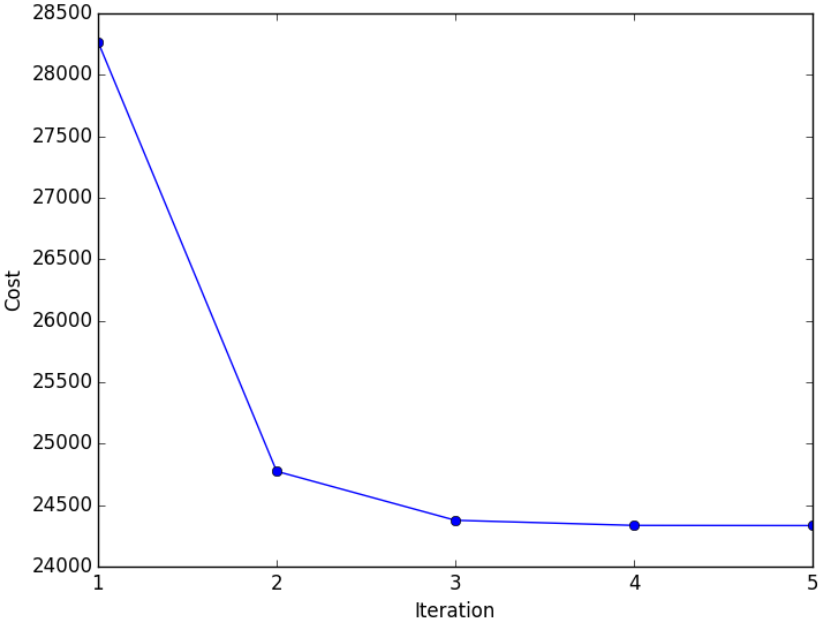

Figure 6-4: The wing model solves in 5 GP iterations

Chapter 7

Combined Aircraft Model

Due to the coupled nature of aircraft system design, it is important to consider all of the major subsystems when trying to optimize a configuration. Combining all of the previously described subsystem models into one large full-aircraft optimization problem allows us to capture the coupled nature of aircraft design.

From a practical perspective, the procedure of combining the subsystem models together can be expressed in three steps:

1. **Create a system-level model** A high level objective must be chosen to govern the combined models. Additionally, any constraints that may not have been relevant in the context of subsystem modelling need to be introduced.

2. **Concatenate the lists of constraints** As mentioned previously, each model is fundamentally just a list of constraints. This means that putting models together essentially just involves a concatenation operation. No wiring between models is needed.

3. **Identify shared variables** To couple these constraints into a single monolithic optimization problem, the optimizer needs to know which variables are in fact the same variable. In many cases this means freeing fixed variables. For example, whereas the fuselage length is fixed in the context of vertical tail design, it can and should be a free variable in a full aircraft context.

7.1 The System-level Model

When fully combined, the aircraft model has 242 unique free variables and 264 constraints. The free and fixed variables that make up the high level model and/or are shared by the sub-models are listed in Table 7.1 and the free variables that are common to two or more of the subsystem models are illustrated in Figure 7-2.

Table 7.1: Free variables in the high level aircraft model and/or shared by sub-models

Free Variables	Units	Description
AR_w	$[-]$	Wing aspect ratio
C_D	$[-]$	Drag coefficient
C_L	$[-]$	Lift coefficient
C_{L_w}	$[-]$	Lift coefficient, wing
$C_{L_{aw}}$	$[-]$	Lift curve slope, wing
D	[N]	Total aircraft drag (cruise)
D_{fuse}	[N]	Fuselage drag
D_{ht}	[N]	Horizontal tail drag
D_{vt}	[N]	Vertical tail drag
D_{wing}	[N]	Wing drag
L_w	[N]	Wing lift
$L_{v_{max}}$	[N]	Maximum load for structural sizing
S_w	$[\mathrm{m}^2]$	Wing reference area
V_{TO}	$[\frac{\mathrm{m}}{\mathrm{s}}]$	Takeoff speed
V_∞	$[\frac{\mathrm{m}}{\mathrm{s}}]$	Cruise velocity
W	[N]	Total aircraft weight
W_{fuel}	[N]	Fuel weight
W_{fuse}	[N]	Fuselage weight
W_{ht}	[N]	Horizontal tail weight
W_{lg}	[N]	Weight of landing gear
W_{pay}	[N]	Payload weight
W_{vt}	[N]	Vertical tail weight
W_{wing}	[N]	Wing weight
W_{zf}	[N]	Zero fuel weight
$\bar{c}_w$	[m]	Mean aerodynamic chord (wing)
$\frac{L}{D}$	$[-]$	Lift/drag ratio

ξ	$[-]$	Takeoff parameter
b_{vt}	[m]	Vertical tail span
c_{vt}	[m]	Vertical tail root chord
h_{hold}	[m]	Hold height
l_{fuse}	[m]	Fuselage length
p_{λ_v}	$[-]$	1 + 2*Tail taper ratio
w_{fuse}	[m]	Fuselage width
x_w	[m]	Position of wing aerodynamic center
$x_{CG_{eng}}$	[m]	x-location of engine CG
$x_{CG_{fu}}$	[m]	x-location of fuselage CG
$x_{CG_{ht}}$	[m]	Horizontal tail CG location
$x_{CG_{lg}}$	[m]	x-location of landing gear CG
$x_{CG_{vt}}$	[m]	x-location of tail CG
$x_{CG_{wing}}$	[m]	x-location of wing CG
x_{CG}	[m]	x-location of CG
x_{TO}	[m]	Takeoff distance
x_{up}	[m]	Fuselage upsweep point
y	$[-]$	Takeoff parameter
z_{bre}	$[-]$	Breguet parameter

Table 7.2: Fixed variables in the high level aircraft model and/or shared by sub-models

Constants	Units	Description
$C_{L_{max}}$	$[-]$	Max lift coefficient
M	$[-]$	Cruise Mach number
N_{lift}	$[-]$	Wing loading multiplier
$Range$	[nmi]	Range
T_e	[N]	Engine thrust at takeoff
V_{ne}	$[\frac{\mathrm{m}}{\mathrm{s}}]$	Never exceed velocity
W_{eng}	[N]	Engine weight
μ	$[\frac{\mathrm{N \cdot s}}{\mathrm{m}^2}]$	Dynamic viscosity (35,000 ft)
ρ	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Air density (35,000 ft)
ρ_0	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Air density (0 ft)
ρ_{cap}	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Density of spar cap material
ρ_{web}	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Density of shear web material
$\sigma_{max,shear}$	[Pa]	Allowable shear stress

σ_{max}	[Pa]	Allowable tensile stress
a	$[\frac{\text{m}}{\text{s}}]$	Speed of sound (35,000 ft)
c_T	$[\frac{\text{lb}}{(\text{hr}\cdot\text{lbf})}]$	Thrust specific fuel consumption
d_{fan}	[m]	Fan diameter
$f_{w,add}$	[−]	Wing added weight fraction
g	$[\frac{\text{m}}{\text{s}^2}]$	Gravitational acceleration
l_r	[ft]	Runway length
r_h	[−]	Fractional wing thickness at spar web
w	[−]	Wingbox-width-to-chord ratio
y_{eng}	[m]	Engine moment arm

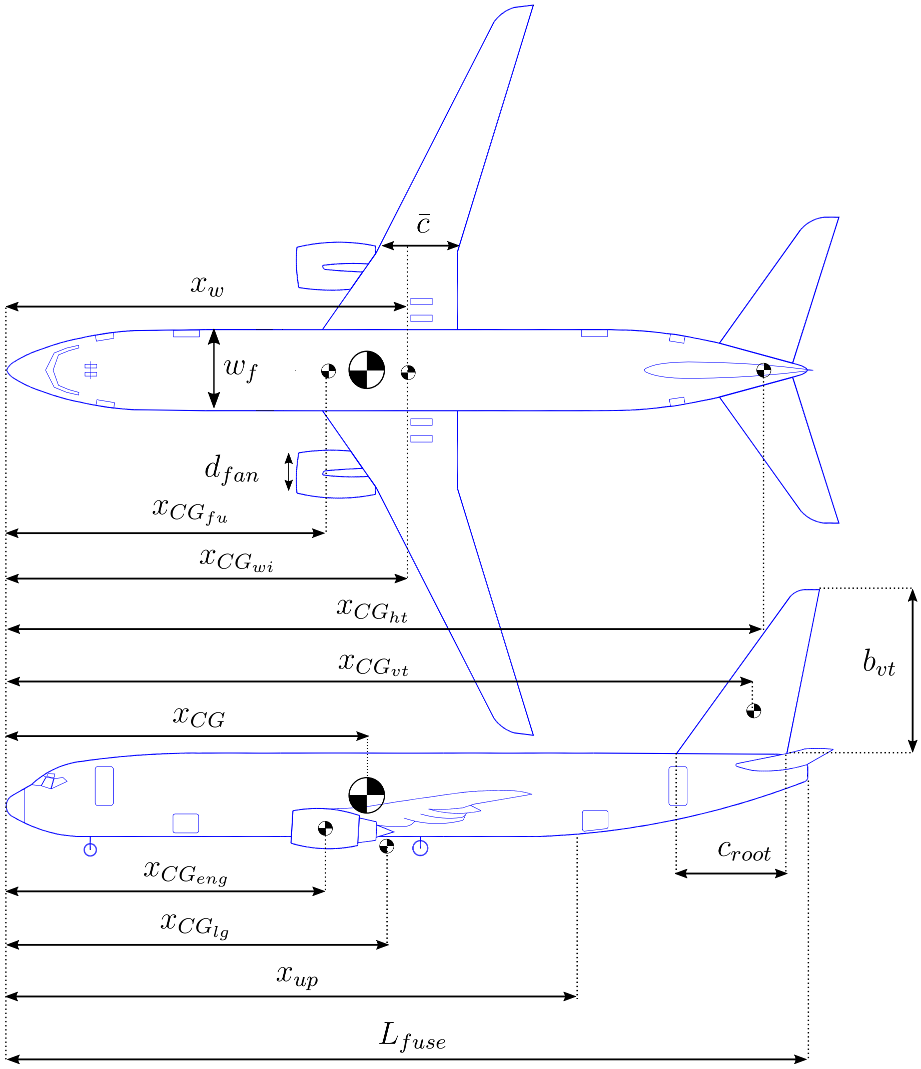

Figure 7-1: Geometric variables that link subsystem models in the full aircraft model (adapted from [4])

The objective for the full aircraft model is to minimize fuel weight, using the same GP-compatible Breguet range formulation introduced in [21].

$$W \geq W_{zf} + W_{fuel} \tag{7.1}$$

$$\frac{W_{fuel}}{W_{zf}} \geq z_{bre} + \frac{{z_{bre}}^2}{2} + \frac{{z_{bre}}^3}{6} + ... \tag{7.2}$$

$$Range \leq \frac{V_\infty}{c_T g}\frac{L}{D} z_{bre} \tag{7.3}$$

We calculate the total empty weight and drag of the aircraft using simple buildup of each component's weight and drag.

$$W_{zf} \geq W_{vt} + W_{fuse} + W_{lg} + W_{wing} + W_{ht} + W_{eng} + W_{pay} \tag{7.4}$$

$$D \geq D_{vt} + D_{fuse} + D_{wing} + D_{ht} \tag{7.5}$$

We also use the same takeoff model as proposed in [21].

$$\xi \geq \frac{1}{2}\frac{\rho_0 {V_{TO}}^2 S_w C_D}{T_e} \tag{7.6}$$

$$1 + y \leq 4\frac{g x_{TO} T_e}{{V_{TO}}^2 W} \tag{7.7}$$

$$1 \geq 0.0464\frac{\xi^{2.7}}{y^{2.9}} + \frac{\xi^{0.3}}{y^{0.049}} \tag{7.8}$$

$$V_{TO} \geq 1.2\sqrt{\frac{2W}{C_{L_{max}} S_w \rho_0}} \tag{7.9}$$

$$x_{TO} \leq l_r \tag{7.10}$$

Finally, the constraint for the aircraft center of gravity is also GP-compatible.

$$\begin{aligned} W x_{CG} \geq\ & W_{eng} x_{CG_{eng}} + W_{fuel} x_{CG_{wing}} + W_{fuse} x_{CG_{fu}} + W_{ht} x_{CG_{ht}} \\ & + W_{lg} x_{CG_{lg}} + W_{pay} x_{CG_{fu}} + W_{vt} x_{CG_{vt}} + W_{wing} x_{CG_{wing}} \end{aligned} \tag{7.11}$$

This in turn requires constraints to model the CG of each subsystem. The landing gear CG is constrained by the moment of each set of gear about the nose.

$$W_{lg} x_{CG_{lg}} \geq W_{mg} x_m + W_{ng} x_n \tag{7.12}$$

The vertical and horizontal tail CGs are bounded by the midpoint of their respective root chords.

$$x_{CG_{h/vt}} \geq \frac{\Delta x_{lead_{h/v}} + \Delta x_{trail_{h/v}}}{2} + x_{CG} \tag{7.13}$$

The fuselage CG buildup constraints are adopted directly from [16] and the wing CG is assumed to coincide with the aerodynamic center.

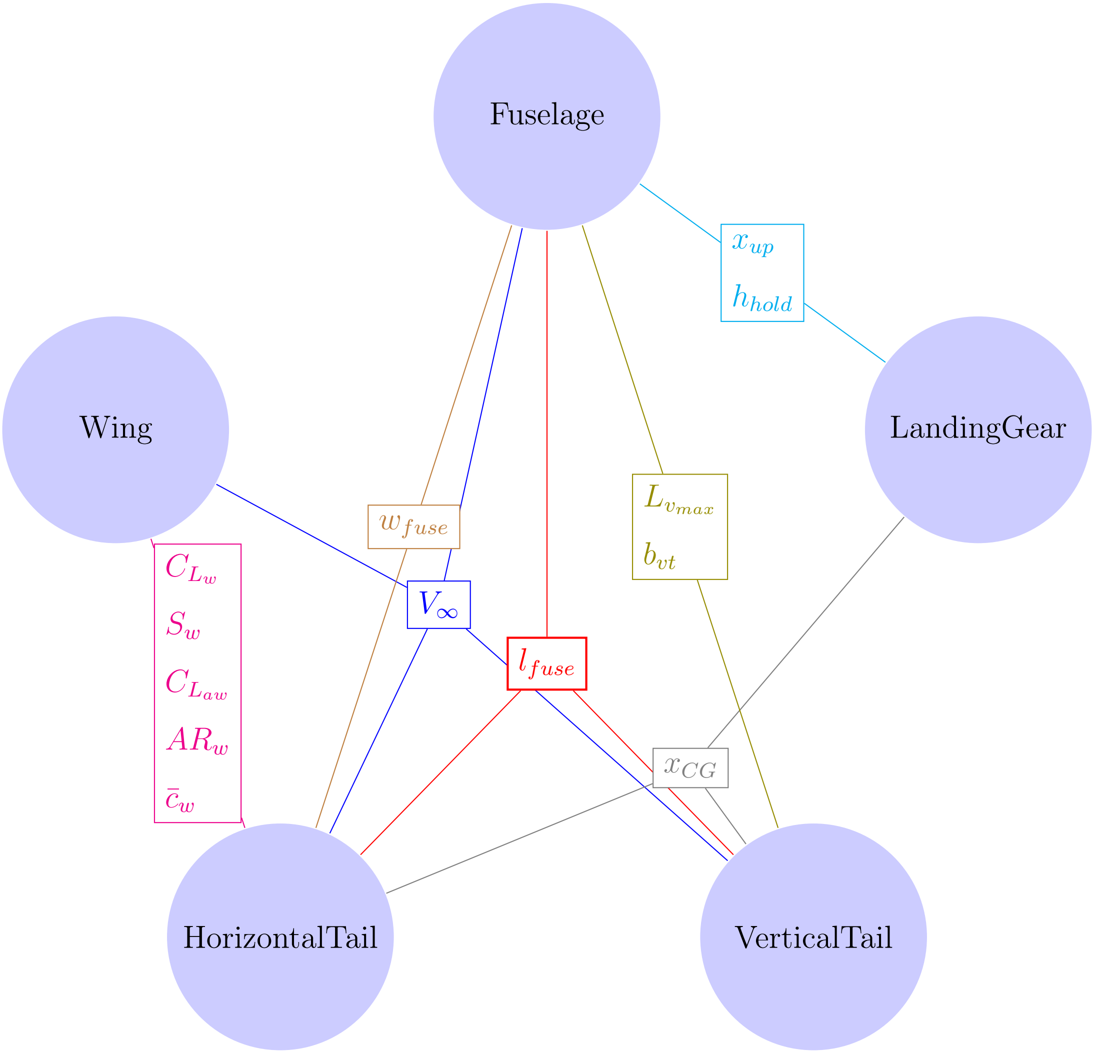

Figure 7-2: An illustration of how free variables link each of the subsystem models

7.2 Model Results

The SP takes 6 GP solves and 3.2 seconds to solve on a standard laptop computer with a 2.4 GHz Intel Core i5 processor. The optimal values for a selection of key design variables are presented in Table 7.3.

Table 7.3: Solution comparison with reference aircraft

Free Variable	Solution Value	Estimate for reference aircraft
Wing		
$Æ_w$ [$-$]	9.3	9.5
b_w [m]	30.2	35.9
S_w [m^2]	98.0	124.6
Horizontal Tail		
$Æ_{ht}$ [$-$]	7.7	6.2
b_{ht} [m]	10.7	14.4
S_{ht} [m^2]	14.8	32.8
Landing Gear		
B [m]	15.7	15.6
T [m]	5.7	5.8
d_{t_m} [in]	39.7	44.5
Fuselage		
R_{fuse} [m]	1.86	1.88
l_{fuse} [m]	52.5	39.1
W_{fuse} [kg]	16,471	16,300
Vertical Tail		
A_{vt} [$-$]	0.71	1.91
b_{vt} [m]	3.71	7.16
S_{vt} [m^2]	19.4	26.4

The optimal solution is most sensitive to the Mach number in cruise (-1.3%) and the range requirement (1.1%).

The convergence of the objective value for the model is show in Figure 7-3. As with the landing gear model, there are no objective values for the first three iterations because these are feasibility finding solves, for which the objective function is a slack variable that does not correspond to the objective function of the model.

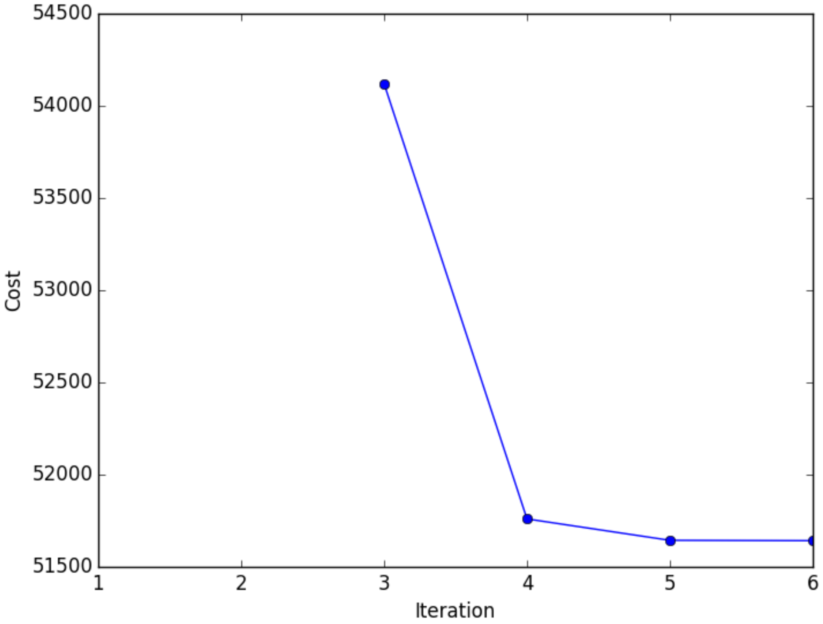

Figure 7-3: The aircraft model solves in 6 GP iterations

Chapter 8

Atmosphere Model

Lift, drag, Reynolds number, and Mach number are all functions of atmospheric quantities, and, in turn, altitude. There is, therefore, value in making cruise altitude a decision variable over which to optimize. To do this, we need to model how atmospheric quantities change with altitude as part of the optimization problem. Conveniently, analytical standard atmosphere models fit naturally into the signomial programming framework. In this chapter we present two such models, one for the Troposphere and one for the Tropopause, along with a model for viscosity that can be used in conjunction with both models. With these three models, the temperature, density, pressure, and viscosity of air can all be modelled for altitudes up to 20 km, sufficiently high to cover all realistic flight altitudes for a commercial aircraft.

Table 8.1: Free variables in the atmospheric models

Free Variables	Units	Description
T	[K]	Temperature
μ	$[\frac{\mathrm{kg}}{(\mathrm{m}\cdot\mathrm{s})}]$	Dynamic viscosity
ρ	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$	Density
h	[m]	Altitude
p	[Pa]	Pressure

Table 8.2: Constants in the atmospheric models

Constants	Value	Units	Description
C_1	1.458e-06	$[\frac{\mathrm{kg}}{(\mathrm{K}^{0.5}\cdot\mathrm{m}\cdot\mathrm{s})}]$	Sutherland coefficient
L	0.0065	$[\frac{\mathrm{K}}{\mathrm{m}}]$	Temperature lapse rate
R	287	$[\frac{\mathrm{J}}{(\mathrm{K}\cdot\mathrm{kg})}]$	Specific gas constant (air)
T_0	288.1	[K]	Temperature at sea level
T_s	110.4	[K]	Sutherland Temperature
g	9.81	$[\frac{\mathrm{m}}{\mathrm{s}^2}]$	Gravitational acceleration
p_0	1.013e+05	[Pa]	Pressure at sea level
p_1	22,630	[Pa]	Pressure at 11 km altitude

8.1 Troposphere

The Troposphere is the layer of atmosphere between sea level and approximately 11 km (36,000 ft) altitude. It is characterized by a temperature profile that decreases linearly with altitude [2].

$$T = T_0 - Lh \tag{8.1}$$

The ideal gas law relates air density to pressure and temperature.

$$\rho = \frac{p}{RT} \tag{8.2}$$

Combining this equation of state with the hydrostatic equation, and Equation 8.1, we obtain a relationship for the pressure variation as a function of temperature [2].

$$\frac{p}{p_0} = \left(\frac{T}{T_0}\right)^{\frac{g}{RL}} \tag{8.3}$$

We now have three equations to model pressure, density, and temperature as functions of altitude. These equations have the potential to form either a geometric program or a signomial program. Because all three are physical relationships, their respective constraints must be satisfied with equality to be valid. Equation 8.2 and Equation 8.3 can clearly be used as monomial equality constraints, assuming g, R, and L are constant. Handling Equation 8.1 is more complicated.

If we can guarantee that there is *exclusively* downward pressure[1] on temperature, then it can be used as the basis for a signomial constraint.

$$T + Lh \geq T_0 \tag{8.4}$$

An example where this might arise is when trying to maximize range for a jet propelled aircraft, where downward pressure on T arises through downward pressure on Mach number for a given flight true air speed.

On the other hand, if there is *exclusively* upward pressure on temperature, the relationship can be used as the basis for a GP constraint.

$$T + Lh \leq T_0 \tag{8.5}$$

This might arise when a model's other constraints want to maximize air density, for example to maximize the endurance of a propellor aircraft.

Unfortunately, cases can easily arise in which the pressure(s) on temperature are not known apriori. In such cases it is necessary to treat the relationship as a signomial *equality* constraint. A comprehensive discussion of implementing signomial equality constraints is not attempted here. Instead, we briefly address possible ways of handling them. Although it may seem like an obvious solution at first glance, simply using two inequalities with opposing $\leq$ and $\geq$ operators is not an effective method for enforcing equality. This is because the feasible set that results from this program is a single point, which interior point method solvers find difficult to handle. A crude, but relatively effective, way of overcoming this hurdle is to multiply one of the sides of one of the conjugate constraints by a number close to, but not exactly, 1 (e.g. 1.01). This makes the feasible set larger than a single point and allows optimizers to per-

[1] pressure in this context refers to optimization pressure, the direction in which a given objective function pushes the decision variables

form more effectively. Using this technique equality can be approximately enforced as follows:

$$T + Lh \leq T_0 \tag{8.6}$$

$$T + Lh \geq 1.01T_0 \tag{8.7}$$

Another, more effective, technique is to successively find the local monomial approximation of the left hand side of the constraint and pose it as a local monomial equality constraint at each iteration of an SP solve.

$$(T + Lh)_{MA} = T_0 \tag{8.8}$$

8.2 Tropopause

The region of atmosphere above the Troposphere is an isothermal layer called the Tropopause that extends to an altitude of 20 km [2].

$$T = 216 \text{ K} \tag{8.9}$$

The ideal gas law is, of course, still valid in this region, but because of the constant temperature, the pressure now decreases according to an exponential function of altitude.

$$\frac{p}{p_{11}} = e^{-\frac{g}{RT}(h - 11000 \text{ [m]})} \tag{8.10}$$

At first glance, this relationship does not appear to be compatible with either GP or SP. However, recall that exponential functions can be well approximated by a Taylor series expansion.

$$e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} ... \tag{8.11}$$

This means that, after some algebraic manipulation, we can obtain a relationship that could be used in a geometric or signomial program.

$$\begin{aligned}
&e^{\frac{11000g}{RT}}\frac{p}{p_{11}} = e^{-\frac{g}{RT}h} \\
\implies &k\frac{p_{11}}{p} = e^{\frac{g}{RT}h} \\
&k\frac{p_{11}}{p} \approx 1 + \frac{gh}{RT} + \frac{1}{2}\left(\frac{gh}{RT}\right)^2 + \frac{1}{6}\left(\frac{gh}{RT}\right)^3
\end{aligned} \tag{8.12}$$

where k is constant. As with the Troposphere model, this relationship needs to be satisfied with equality. Either of the methods described above can be used to achieve this.

8.3 Sutherland Viscosity Model

The other altitude-dependent atmospheric quantity of interest is air viscosity, which affects Reynolds number. The Sutherland model [32] gives dynamic viscosity as a function of temperature and is therefore readily combined with the above atmosphere models.

$$\mu = \frac{C_1 T^{\frac{3}{2}}}{T + T_S} \tag{8.13}$$

Although this relationship could be relaxed to either a signomial or posynomial inequality constraint, as with the previous models, it must be satisfied with equality. As before, there are a number of possible methods for handling this. In this case, because the posynomial term can be well approximated by a monomial, it is possible to take a single monomial approximation about, for example, sea level, and use this as an approximate model, without needing to find a new approximation at each iteration. On its own, this model is therefore GP-compatible.

$$\mu = \frac{C_1 T^{\frac{3}{2}}}{(T + T_S)_{MA}} = \frac{C_1 T^{\frac{3}{2}}}{6.64 T^{0.72}} \tag{8.14}$$

It can be seen in Figure 8-1 that the error for all temperatures within the Troposphere and Tropopause is less than 1%.

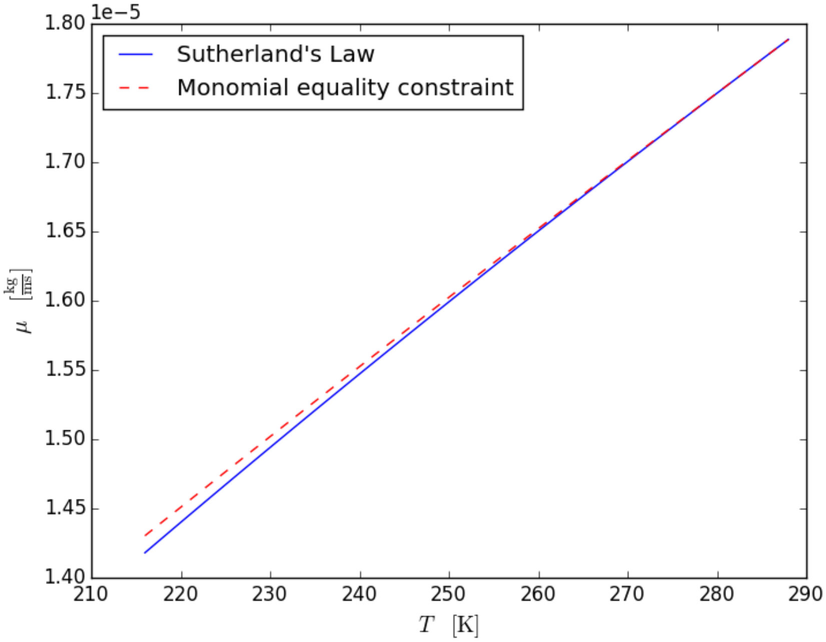

Figure 8-1: A monomial equality constraint closely approximates Sutherland's law for all temperatures within the Troposphere and Tropopause

Chapter 9

Comparison with General Nonlinear Programming Techniques

In [21] the author demonstrates that simple aircraft design problems can be solved quickly, effectively and easily using geometric programming. In the preceding chapters we have demonstrated that more sophisticated conceptual aircraft design problems can be solved quickly, effectively and easily using signomial programming. These claims of speed, effectiveness and ease have limited value, however, without a benchmark or comparison against an existing alternative technique.

We therefore set out to compare geometric and signomial programming with general Nonlinear Programming (NLP) techniques in the context of aircraft design problems. To achieve this, a tool was created that automatically generates the necessary files to solve any geometric or signomial program using MATLAB's ® [27] `fmincon` solver, a popular and commercially available general purpose optimization tool.

Specifically, our goal was to compare the time taken to solve aircraft design problems and the quality of the solution found, as measured by the value of the optimal cost. We were also interested in a less quantifiable metric of how easy it is for a user to obtain a meaningful solution and how sensitive this process is to the information the solver is given.

`fmincon` is a MATLAB function for finding the minima of constrained nonlinear optimization problems. It comprises a number of different solvers, including both an interior point method solver and a Sequential Quadratic Programming (SQP) solver. Interior point methods are a powerful class of algorithm that, as mentioned previously, are the method of choice for solving log-transformed GPs, and therefore, by extension, SPs. Though originally derived for solving linear programs, and most frequently used for convex optimization problems, interior point algorithms have also been developed and successfully applied to solving nonlinear, nonconvex problems [34]. The `fmincon` implementation of an interior point method is based on work presented in [10], [11] and [35].

SQP is one of the most effective methods for nonlinearly constrained optimization [30] and is widely used in many fields, including aircraft design [23]. The `fmincon` implementation of the SQP algorithm is largely based on the algorithm described in [30].

Given the widespread use and popularity of interior point methods and SQP, we chose to compare both algorithms with the techniques of geometric and signomial programming. These techniques (e.g. geometric programming) and algorithms (e.g. interior point method) are not analogous though - after all, one uses the other - so what does this actually mean? Recall that there are three key steps to solving an optimization problem using geometric programming:

1. Formulate the problem as a geometric program

2. Transform the problem into logarithmic space using a change of variables

3. Solve the transformed problem using an interior point method

For the purpose of this discussion, *geometric programming* refers only to steps (2) and (3), because, for all comparisons presented here, we used the same formulation of each problem, and only varied the solution technique. Similarly, in this chapter *signomial programming* refers to the process of solving a signomial program, once it has been formulated.

Therefore, the only fundamental difference between solving a GP using `fmincon`'s interior point method and using a geometric programming package, such as GPkit, is the log transformation used by the latter to make the problem convex.

The comparison between signomial programming and the general NLP methods is slightly different. As discussed previously, the difference between solving a GP and solving an SP is significant when using a geometric programming solver, because the former is convex in log space, whereas the latter is not. On the other hand, solving a signomial program using an NLP solver is, in general, no more difficult that solving a geometric program using an NLP solver, because neither problem is convex without the logarithmic transformation. So, when comparing signomial programming (i.e. GPkit) with an interior point method (i.e. `fmincon`), we are fundamentally comparing using an interior point method on a sequence of convex problems, with using an interior point method on a single non-convex problem.

The measures of computational cost presented in this chapter must be taken with a grain of salt, as the alternatives are not implemented in the same programming language, the overhead cost associated with each implementation is not closely controlled, and the cost is measured in wall clock time, as opposed to CPU time. Although one could argue that a more rigorous comparison would use the same interior point method solver for both the geometric/signomial programming approach and the general NLP approach, the MOSEK interior point solver used by GPkit is heavily tailored for convex problems [3] and users are therefore discouraged from employing it to solve non-convex problems, such as a GP without the change of variables. Meanwhile, `fmincon`'s interior point method implementation is based on algorithms that were designed to be robust to non-convexity [11].

`fmincon` has a number of options designed to improve its performance. For example, it is possible to give gradients of the objective and constraints as inputs. In the results presented below, `fmincon` had access to analytical gradients for the objectives and constraints, computed as follows. The gradient of a signomial,

$$s(\boldsymbol{u}) = \sum_{k=1}^{K} c_k \prod_{j=1}^{n} u_j^{a_{jk}} \tag{9.1}$$

can be computed as

$$\frac{\partial s(\boldsymbol{u})}{\partial u_i} = \sum_{k=1}^{K} c_k a_{ik} u_i^{a_{ik}-1} \prod_{j=1, j\neq i}^{n} u_j^{a_{jk}} \tag{9.2}$$

The gradients were checked using `fmincon`'s `DerivativeCheck` feature.

`fmincon` requires an initial guess for each design variable, regardless of which solution algorithm is used. By contrast, the interior point methods used to solve geometric and signomial programs do not require an initial guess, although the signomial programming heuristic requires initial guesses for variables that appear in signomial constraints.

To ensure a comprehensive comparison, we tested several combinations of the different options, as well as a range of initial guesses, for two different test problems: one geometric program and one signomial program. We began by performing a comparison on a relatively small GP, before comparing the larger landing gear SP model from chapter 4.All tests are performed on a laptop computer with a 2.4 GHz Intel Core i5 processor, using MATLAB 8.5.0.

9.1 Simple UAV Design Geometric Program

As a first test case we used the simple UAV design problem from [21]. This problem is a GP with 10 decision variables and 8 constraints.

$$\begin{aligned}
\text{minimize} \quad & D \\
\text{subject to} \quad & C_D \geq \frac{C_{D_{A0}}}{S} + kC_f S_{wet,ratio} + \frac{C_L^2}{\pi A e} \\
& W_w \geq W_{c_1} S + W_{c_2} \frac{N_{ult} A^{1.5} \sqrt{W_0 W S}}{\tau} \\
& D = \frac{1}{2} \rho V^2 S C_D \\
& Re \leq \frac{\rho V \sqrt{S/A}}{\mu} \\
& C_f \geq 0.074 Re^{-\frac{1}{5}}
\end{aligned}$$

$$
\begin{aligned}
W &\leq \frac{1}{2}\rho V^2 S C_L \\
W &\leq \frac{1}{2}\rho V_{min}^2 S C_{L_{max}} \\
W &\geq W_0 + W_w
\end{aligned}
$$

GPkit solves this problem in 0.015 seconds and finds a guaranteed globally optimal cost of 303.1 N. We tested the exact same problem using `fmincon` with both the interior point method solver and the SQP solver, as well as a variety of different settings: with and without gradients; with and without an explicit distinction between linear constraints; and with different initial guesses.

Table 9.1: Initial guesses used for simple UAV design case

Variable	Optimal value from GP solution	A Ones	B Almost exact solution	C Order of magnitude (floor)	D Order of magnitude (round)	E Order of magnitude (mix)
A	8.46	1	8	1	10	10
C_D	0.02059	1	0.02	0.01	0.01	0.1
C_L	0.4988	1	0.5	0.1	1	1
C_f	0.003599	1	0.004	0.001	0.01	0.01
D	303.1	1	300	100	100	100
Re	3.675×10^6	1	4×10^6	1×10^6	1×10^7	1×10^6
S	16.44	1	20	10	10	10
V	38.15	1	40	10	100	10
W	7341	1	7000	1000	10,000	10,000
W_w	2401	1	2000	1000	1000	1000

The exact initial guesses used are listed in Table 9.1. Option A is an arbitrary initial guess of one for every variable. Without any understanding of a problem, let alone prior knowledge of the solution, this might represent the best initial guess a user can provide. It is also an appealing initial guess because it requires effectively no effort from the user.

Recognizing that the completely naive option A might be asking too much of most optimizers, initial guess B is, at the opposite extreme, the globally optimal solution rounded to one significant figure, where the optimal solution is taken from the GP solution. This is, of course, an unrealistic initial guess, but it tests the optimizer on what should be an easy problem. The intention behind rounding to one significant figure is to try and prevent any gradient based searching from getting stuck in numerical noise.

Initial guesses C, D and E are based on the order of magnitude of the optimal solution. These are intended to be intermediate guesses between options A and B. They are more realistic than initial guess B and more informative than initial guess A, and are intended to represent an initial guess that a user might be able to provide in a practical context.

Two different interpretations of the order of magnitude are used: the “floor” and “round” estimates. Using mathematical notation for the floor and round operators, initial guesses C and D are given by

$$x_C^0 = 10^{\lfloor \log_{10}(x^*) \rfloor} \tag{9.3}$$

and

$$x_D^0 = 10^{\lceil \log_{10}(x^*) \rfloor}, \tag{9.4}$$

respectively, where x^* is the optimal value as obtained from the GP solution.

Initial guess E is a combination of initial guesses C and D, with one extra perturbed value (C_D). These three fairly similar variants are intended to test how sensitive the solution is to slight changes in the assumed order of magnitude of the final solution.

It should be noted that none of these initial guesses, besides perhaps option A, are a fair comparison with GP because the starting point for an interior point solver is typically chosen automatically and not informed by the user.

The results of the comparison tests are listed in Table 9.2. As can be seen, the solution obtained is highly dependent on the solver, the settings, and the choice of initial guess.

Table 9.2: Simple UAV design results for `fmincon` using different inputs

Method	Initial	Objective	Constraint	Time	Cost
	guess	gradients	gradients	(s)	
GP	-	-	-	0.015	303.14
IP	A	N	N	> 8 hours	N/A
IP	A	Y	Y	651.1	-6052.5 [1]
IP	B	N	N	10.0	303.14
IP	B	Y	N	18.1	303.14
IP	B	N	Y	131.2	303.14
IP	B	Y	Y	> 8 hours	N/A
IP	C	N	N	3.0	303.14
IP	C	N	Y	7.4	303.14
IP	C	Y	Y	7.3	303.14
IP	D	N	N	15.5	303.14
IP	D	Y	N	6.8	303.14
IP	E	N	N	47	303.14
IP	E	Y	N	75	303.14
IP	E	Y	Y	28	303.14
SQP	A	N	N	0.3	303.14
SQP	A	Y	N	0.3	303.14
SQP	A	Y	Y	0.4	6865.8 [1]
SQP	B	N	N	0.1	304.95
SQP	B	Y	N	0.1	304.95
SQP	B	N	Y	0.1	304.95
SQP	B	Y	Y	0.0	304.95
SQP	C	N	N	0.3	-6.2662e+05 [1]
SQP	C	Y	N	12.6	-8.4531e+16 [1]
SQP	C	N	Y	0.4	-59355 [1]
SQP	C	Y	Y	3.7	1144.7 [1]
SQP	D	N	N	11.0	438.66 [1]
SQP	D	Y	N	0.8	972.45 [1]
SQP	D	N	Y	0.9	2869.2 [1]
SQP	D	Y	Y	0.9	618.98 [1]
SQP	E	N	N	0.2	337.8
SQP	E	Y	N	0.2	337.8
SQP	E	Y	Y	0.1	337.8

A few observations can be made from the results:

1. Giving the initial guess of all ones (choice A) did not work for the interior point solver. The solver either did not converge within a reasonable amount of time, or converged to an infeasible point.

2. With other initial guesses, the interior point solver was typically able to find the optimal solution, although the solution time was highly dependent on the initial guess and whether gradients were used. The fastest interior point solve took 200 times longer than then GP solve.

3. Giving analytic gradients did not necessarily help, and in some cases hindered the solver.

4. The SQP solver typically converged much faster than the interior point solver, but, from the order of magnitude initial guesses, either converged to an infeasible point or converged to a point that had a 10% higher cost than the true optimum.

5. Surprisingly, the SQP solver performed best without analytic gradients and with the most naive initial guess, taking only 20 times as long as the GP solve to find the optimal solution.

6. Despite the fact they are relatively similar, there were significant and unpredictable differences in performance between the three order of magnitude initial guesses for both solver algorithms. For example the interior point solver took up to 15 times longer to solve from initial guess E than from initial guess C. point solver.

They key take-away from this experiment is that, even for a relatively small GP, both of the general NLP methods are very sensitive to the choice of initial guess and are markedly slower than the GP solver.

[1] Infeasible solution

9.2 Landing Gear Signomial Program

We next compare the NLP methods with signomial programming, using the landing gear model presented in chapter 4 as a test case. In addition to being a signomial, rather than geometric, program, this problem is also larger than the UAV design test case with 44 decision variables and 54 constraints.

GPkit is able to solve the SP using 7 GP solves in 1.16 seconds, achieving an optimal cost (weight) of 14,630 N, using an initial guess of one for each design variable that is part of a signomial constraint. The results achieved using the `fmincon` solvers are presented in Table 9.3. We used both the interior point method and SQP solvers with the same types of initial guesses as in the previous example, this time based on the optimal solution of the SP.

We can make a few observations from these results:

1. All but one of the SQP solves converges to an infeasible solution

2. The interior point algorithm converges to a feasible solution in most cases, but is only capable of achieving a solution that has a cost 5% higher than the SP optimal solution, even when starting from near the optimal solution.

3. Once again the solution time is highly dependent on the choice of initial guess. The fastest interior point solution from an order-of-magnitude initial guess is seven times longer than the GPkit solution time, and only attains a cost 10% higher than the SP solution.

Given that the central theme of this thesis is that SP is a viable technique for aircraft design optimization, this experiment is perhaps one of the most valuable contributions, as it compares using signomial programming with using existing general NLP techniques for solving an aircraft design problem. The results demonstrate that, even though signomial programming cannot guarantee a global optimum, it is able to find a better solution in considerably less time than both an interior point solver and a SQP solver.

Table 9.3: Landing gear model results for `fmincon` using different inputs

Method	Initial guess	Objective gradients	Constraint gradients	Time (s)	Cost
SP	-	-	-	1.2	1.463E4
IP	A	N	N	189.5	-1.5333e+05 [2]
IP	A	Y	N	176.3	-1.1736e+05 [2]
IP	A	N	Y	$>$ 8 hours	N/A
IP	A	Y	Y	$>$ 8 hours	N/A
IP	B	N	N	127.1	15472
IP	B	Y	N	39.9	15472
IP	B	N	Y	12.8	15915 [2]
IP	B	Y	Y	9.3	15454
IP	C	N	N	599.8	15472
IP	C	N	Y	75.5	6302 [2]
IP	C	Y	N	$>$ 8 hours	N/A
IP	C	Y	Y	550.7	15511
IP	D	N	N	17.7	16185
IP	D	N	Y	7.5	16101 [2]
IP	D	Y	N	20.3	15448
IP	D	Y	Y	31.5	16781
SQP	A	N	N	0.5	-2438.4 [2]
SQP	A	Y	N	0.8	-3586.8 [2]
SQP	A	N	Y	0.6	-94759 [2]
SQP	A	Y	Y	1.3	-2.2044e+05 [2]
SQP	B	N	N	0.3	2778.2 [2]
SQP	B	Y	N	0.3	33879 [2]
SQP	B	N	Y	1.2	15709 [2]
SQP	B	Y	Y	43.6	15724 [2]
SQP	C	N	N	0.3	-9.323e+05 [2]
SQP	C	Y	N	0.3	-5.241e+06 [2]
SQP	C	N	Y	0.1	9762.8 [2]
SQP	C	Y	Y	0.1	9762.8 [2]
SQP	D	N	N	1.3	19930
SQP	D	Y	N	0.3	19603 [2]
SQP	D	N	Y	1.9	10344 [2]
SQP	D	Y	Y	89.7	10303 [2]

As the results from this chapter demonstrate, taking the time to formulate a problem as a geometric or signomial program has great benefits. These include being able to (i) effectively solve large optimization problems without needing to provide any

[2] Infeasible solution

information beyond the constraints that govern the problem, and (ii) reliably solve problems quickly using a standard desktop computer or laptop.

One thing not taken into consideration is the cost of (i.e. time spent) formulating a problem as a GP or SP in the comparisons performed in this chapter. In previous chapters, however, we have attempted to demonstrate that many aircraft design relationships are expressed naturally in a form compatible with SP, thus making this cost relatively low.

Combining these findings, there is a strong argument for formulating and solving aircraft design problems as geometric and signomial programs wherever possible, due to the speed and robustness advantages they hold over general NLP methods.

Chapter 10

Comparing Geometric and Signomial Programming

In the previous chapter, SP was compared with more general, but somewhat unreliable, nonlinear programming techniques. We now turn to the other side of the spectrum and compare SP with the less general, but very reliable, technique of GP. A key claim of this thesis is that the ability to capture certain relationships using SP enables a marked improvement in modelling fidelity over GP. The price paid for this improvement is a loss in the guarantee of global optimality and an increase in computational cost. To justify using signomial programming instead of geometric programming, we must demonstrate that the improvement in fidelity is substantial enough to warrant this cost.

To achieve this, we return to some of the models presented in previous chapters and investigate what changes need to be made for them to be GP-compatible, what this means from a modelling fidelity standpoint, and how it affects the solution time and optimal solution.

In discussions of SP solution techniques, the GP approximation of a given SP refers specifically to the GP sub-problem that is solved to find the SP solution, and is found by taking the local monomial approximation of a posynomial term in each signomial constraint. For the purpose of this discussion, however, the GP approximation refers more generally to a program where each signomial constraint is replaced by a posynomial constraint that models the same relationship, or a similar one, in a lower fidelity way.

Our primary approach for these GP approximations is to use engineering intuition to find posynomial constraints that suitably replace signomial constraints. Another approach is to use the monomial approximation formula, Equation 1.10, to replace signomial constraints with posynomial constraints, using engineering intuition to choose a suitable point about which to take this approximation. As will be shown, in some cases these two approaches are one and the same.

The choice between geometric and signomial programming is ultimately up to the modeler. There are cases where signomial programming enables capturing a critical relationship, but there are also cases where the additional fidelity is not substantial enough to merit sacrificing the guarantees and speed of geometric programming.

In the following examples, we attempt to qualitatively and quantitatively evaluate this difference in fidelity between a geometric programming and signomial programming approach.

10.1 Examples

10.1.1 Vertical Tail Model

The vertical tail model proposed in chapter 2 has three signomial inequality constraints. The first bounds the tail moment arm by the position of the leading edge, the leading edge sweep, and the vertical position and magnitude of the mean aerodynamic chord.

$$l_{vt} \leq \Delta x_{lead} + z_{\bar{c}}\tan(\Lambda_{LE}) + \frac{\bar{c}}{4} \tag{10.1}$$

The second constrains the mean aerodynamic chord by the taper ratio and the root chord.

$$\bar{c} \leq \frac{2}{3}\left(\frac{1+\lambda+\lambda^2}{1+\lambda}\right)c_{root} \tag{10.2}$$

The third constrains the vertical tail area by the span and root and tip chords.

$$S_{vt} \leq b_{vt}\frac{c_{root} + c_{tip}}{2} \tag{10.3}$$

All of these constraints reflect exact relationships accurately using signomial inequalities. They can each, however, be replaced by constraints that capture the true relationships more crudely. A GP-compatible replacement for Equation 10.1 is:

$$l_{vt} \leq \sqrt{\Delta x_{lead} \Delta x_{trail}}, \tag{10.4}$$

where the moment arm is now bounded by the geometric mean of the tail leading and trailing edge locations.

Similarly the mean aerodynamic chord can be constrained by the geometric mean of the root and tip chord, thus replacing Equation 10.2 with:

$$\bar{c} \leq \sqrt{c_{root} c_{tip}}. \tag{10.5}$$

Not only is the geometric mean of the root and tip chords a fairly simple, if crude, approximation of the mean aerodynamic chord, but it also implicitly corresponds to the best possible monomial approximation about any point where $\lambda = \frac{c_{tip}}{c_{root}} = 1$.

We can then use the mean aerodynamic chord approximation for bounding the tail area too, replacing Equation 10.3 with:

$$S_{vt} \leq b_{vt}\bar{c}. \tag{10.6}$$

Here we have effectively replaced an arithmetic mean with a geometric mean, which again corresponds to the best possible monomial approximation about any point where $c_{tip} = c_{root}$.

Following these constraint substitutions, the vertical tail optimization model is a GP and can be solved as such. The obvious question is: how do these changes affect the optimal solution? To find out, we solve the GP using the same objective function

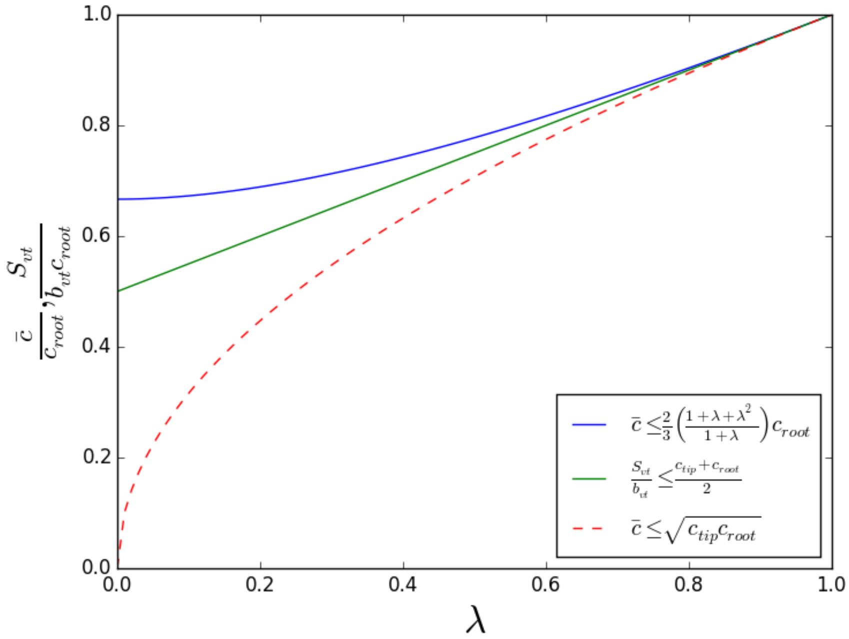

Figure 10-1: The geometric mean approximation corresponds to a monomial approximation of the SP constraints at $\lambda = 1$

and fixed variable values as in chapter 2, and the solution is compared with that of the original SP in Table 10.1.

It should be noted that the substituted relationships - a selection from many possible options - are not necessarily the best approximations, either from an aircraft design perspective or from an optimization perspective. They are however, perfectly reasonable, and perhaps even obvious, alternatives.

Importantly, the second and third approximations are also both strictly conservative

Table 10.1: Solution comparison between the vertical tail SP model and its GP approximation

	SP solution	GP solution	Relative change
Objective	1808.89	1902.64	5.18%
Design Variables			
b_{vt} [m]	7.16	6.67	-6.77%
$\bar{c}_{vt}$ [m]	4.08	4.24	3.84%
l_{vt} [m]	17.88	17.01	-4.91%
S_{vt} [m^2]	26.30	28.27	7.47%

replacements. For example, the geometric mean is always less than or equal to the arithmetic mean, so the constraint on vertical tail area is more restrictive. This is exactly the same property that allows SPs to be solved without requiring a trust region. The conservative property does not apply to the first replacement on its own because x_{trail} does not appear in the original constraint. A strictly conservative replacement could be:

$$l_{vt} \leq \Delta x_{lead}. \tag{10.7}$$

That said, in the context of other constraints that relate the mean chord $\bar{c}$ to Δx_{trail} and Δx_{lead} through the root chord c_{root} (e.g. Equation 10.2), it can be shown that the first replacement would indeed be a conservative substitution as well.

10.1.2 The Wing Model

The standalone wing model has four signomial constraints: the DATCOM formula (5.5), a constraint equivalent to (10.3), a constraint for maximum fuel volume (6.4), and a constraint that relates Oswald efficiency to taper (6.2).

The DATCOM Formula

The DATCOM formula [25] is an analytical relationship for estimating the lift curve slope of a wing or tail.

$$C_{L_\alpha} = \frac{2\pi Æ}{2 + \sqrt{\left(\frac{Æ}{\eta}\right)^2 (1 + \tan^2 \Lambda - M^2) + 4}}$$

As discussed in the previous chapters, it can be used as a signomial inequality constraint, although some algebraic manipulation is needed to see this.

$$\frac{C_{L_\alpha}^2}{\eta^2}\left(1 + \tan^2 \Lambda - M^2\right) + \frac{8\pi C_{L_\alpha}}{Æ} \leq 4\pi^2$$

Removing the $-M^2$ term from the left hand side would clearly make this constraint GP-compatible. Interestingly, this corresponds to a fairly common practice in aerodynamics: assuming that a flow is incompressible and that Mach number effects are therefore negligible.

$$\frac{C_{L_\alpha}^2}{\eta^2}\left(1+\tan^2\Lambda\right)+\frac{8\pi C_{L_\alpha}}{\AE}\leq 4\pi^2$$

So, the DATCOM formula is a GP-compatible constraint for incompressible flow regimes and an SP-compatible constraint for compressible flow regimes.

Making the wing model a GP

To make the wing model a GP, each signomial constraint is replaced with a GP-compatible constraint. The DATCOM formula is replaced with the incompressible form described above. The reference area constraint is replaced with (10.6) as described in the previous section. The fuel volume and Oswald efficiency constraints are made GP compatible by setting the taper ratio to a constant value of 0.2 (the previous optimal value). The results of the original SP are compared with the incompressible SP (only replacing the DATCOM constraint), and the GP. As can be seen in Table 10.2, changing the DATCOM constraint to be a posynomial constraint has a much more significant impact than changing the other three constraints.

Table 10.2: Solution comparison between the original wing model SP, the SP with an incompressible DATCOM constraint, and a GP approximation of the SP

	Original SP	Incompressible SP	GP
Objective	24333.8	27284.8 (+12.1%)	27927.4 (+14.8%)
Design Variables			
$\AE$ [−]	9.3	7.3	7.3
b_w [m]	36.3	37.4	37.3
S_w [m^2]	141.2	191.9	191.8

Chapter 11

Conclusion

In this work, signomial programming has been used to tackle aircraft design problems. More specifically, signomial programming models have been created to find the optimal preliminary sizing of a commercial tube-and-wing aircraft's wing, vertical tail, horizontal tail, fuselage, and landing gear. The subsystem models have also been combined into a single monolithic signomial program that captures the coupled nature of aircraft design.

In doing this work, signomial programming has been demonstrated as a viable approach to multidisciplinary aircraft design optimization, with a wide range of constraints fitting naturally into the required formulation. Though not as rigorous as for geometric programs, the solution method for signomial programs is disciplined and effective. A significant improvement in fidelity over previous geometric programming models has been achieved thanks to the relaxed restrictions on signomial programs. Lagrange multipliers obtained from the solution procedure mean that, in addition to finding an optimal design, the models also give local sensitivities to fixed variables, thus giving insight into the design space.

Comparisons have also been made between signomial programming and general nonlinear programming methods, as well as between signomial programming and geometric programming. The results support the claims that signomial programming is faster and more reliable than general nonlinear programming methods and can achieve marked improvements in fidelity over geometric programming.

Appendix A

The Difference of Convex Formulation of a Signomial Program

An interesting subtlety of signomial programming is how to actually define a signomial program. Despite the importance of choosing an effective formulation, there does not appear to be a consensus in the literature regarding the best definition. The purpose of this appendix is to explain the formulation used in this work. As discussed in chapter 1, the ability to write a signomial program as a 'difference of convex' optimization problem is key to being able to solve it without needing a trust region, and this is the underlying motivation for the chosen formulation, (A.1).

$$\begin{aligned} \text{minimize} \quad & \frac{p_0(\boldsymbol{u})}{q_0(\boldsymbol{u})} \\ \text{subject to} \quad & s_i(\boldsymbol{u}) \leq 0, \;\; i = 1, ..., n_s, \\ & p_i(\boldsymbol{u}) \leq 1, \;\; i = 1, ..., n_p, \\ & m_i(\boldsymbol{u}) = 1, \;\; i = 1, ..., n_m. \end{aligned} \tag{A.1}$$

A.1 Constraints

Recall that the standard form of a GP (1.3) has both monomial and posynomial constraints less than or equal to one. One might therefore assume that the natural definition for a signomial constraint is:

$$s(x) \leq 1. \tag{A.2}$$

Although this form appears fairly common in the literature [7], it can be shown that using the form

$$s(x) < 0 \tag{A.3}$$

results in a more efficient solution procedure. The distinction may seem trivial, after all $s(x) - 1$ is also a signomial, but it has significant implications for the local GP approximation of the constraint. To better see this, consider the approximation procedure for a signomial constraint using both formulations.

$$\begin{array}{rlcrl}
s(x) & \leq 0 & \qquad & s(x) & \leq 1 \\
p(x) - q(x) & \leq 0 & & p(x) - q(x) & \leq 1 \\
p(x) & \leq q(x) & & p(x) & \leq 1 + q(x) \\
 & & & p(x) & \leq r(x) \\
\dfrac{p(x)}{\hat{q}(x)} & \leq 1 & & \dfrac{p(x)}{\hat{r}(x)} & \leq 1
\end{array}$$

where $\hat{q}(x)$ and $\hat{r}(x)$ are the local monomial approximations to $q(x)$ and $1 + q(x)$, respectively. To see how these approximations compare, consider an example signomial constraint.

$$x + y \geq z \tag{A.4}$$

The two possible formulations for this constraint are:

$$\frac{z}{(x+y)_{MA}} \leq 1 \qquad\qquad \frac{z+1}{(1+x+y)_{MA}} \leq 1. \tag{A.5}$$

Using Equation 1.10 to evaluate the monomial approximations, their respective GP compatible forms are:

$$\frac{z}{(a+b)\left(\frac{x}{a}\right)^{\frac{a}{a+b}}\left(\frac{y}{b}\right)^{\frac{b}{a+b}}} \leq 1 \qquad \frac{z+1}{(1+a+b)\left(\frac{x}{a}\right)^{\frac{a}{1+a+b}}\left(\frac{y}{b}\right)^{\frac{b}{1+a+b}}} \leq 1. \tag{A.6}$$

We can clearly see that these approximations are only equivalent where $x = a$ and $y = b$, i.e. the point about which the approximation is being taken. As can be seen in Figures A-1 and A-2, besides this point, the $s(x) \leq 1$ approximation (dotted contours) is always further than the $s(x) \leq 0$ approximation (dashed contours) from the original constraint (solid contours). The first approximation is better because it is a monomial, whereas the second is a posynomial, and the constraint being approximated is log-concave. In log space, monomials are affine, whereas posynomials are strictly convex so a monomial is a strictly better approximation to a log-concave function than a posynomial.

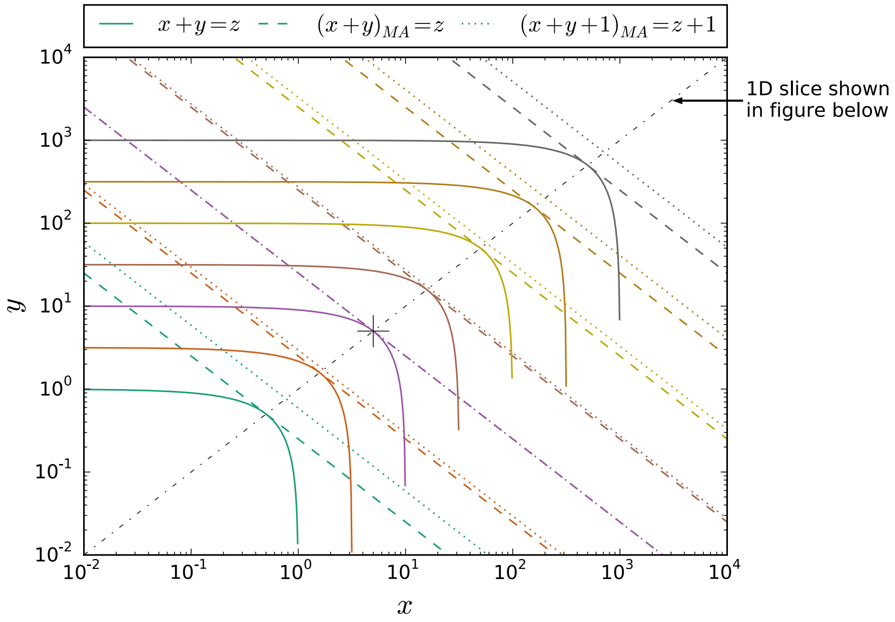

Figure A-1: The $s(x) \leq 0$ formulation (dashed contours) provides a better approximation to the original constraint (solid contours) than the $s(x) \leq 1$ formulation (dotted contours).

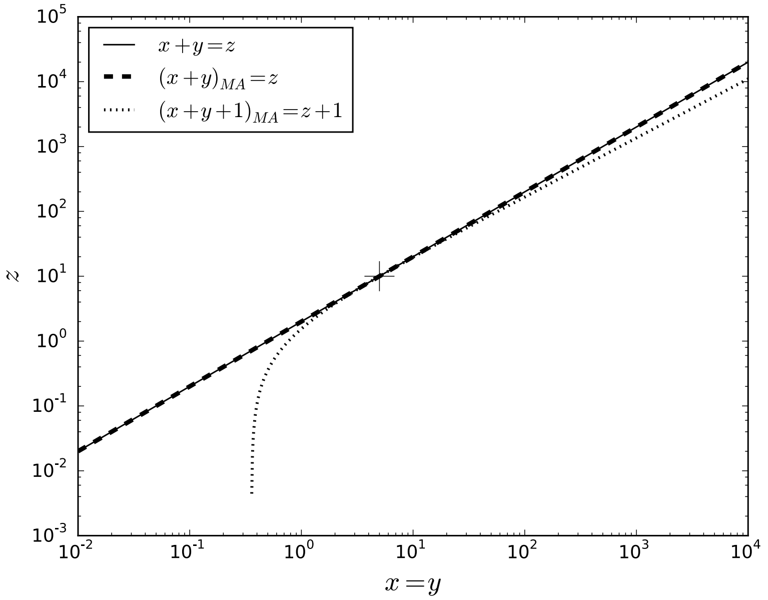

Figure A-2: The $s \leq 0$ approximation matches the original constraint exactly for the 1D slice through $x = y$, whereas the $s(x) \leq 1$ approximation is only tangent at the point about which the approximation was taken.

A.2 Objective

Again, given that the objective of a geometric program is a posynomial, it might seem natural for the objective of a signomial program to be a signomial. This is not the case because a signomial is not a 'difference of convex' function - it is not, in general, possible to write a signomial as the difference of two log-convex functions. A posynomial is log-convex so the objective for a signomial can be, and often is, a posynomial. It is, however, possible to achieve a more general objective function using a ratio of posynomials, given that

$$\log\left(\frac{p_1}{p_2}\right) = \log(p_1) - \log(p_2) \tag{A.7}$$

which is a difference of convex functions. This has an additional elegant feature that the objective function and constraint functions can be interchanged, without transformation, when signomial constraints are written as a ratio of posynomials less than one.

Appendix B

Combined Aircraft Model Results

This appendix contains the solution of the combined aircraft model. It consists of three tables: the first contains the optimal values of the free variables, the second contains the values used for fixed variables, and the third contains the sensitivities to the fixed variables. For each table, the variables are grouped by sub-model. The sensitivity table is additionally ranked by values of the sensitivities.

Table B.1: Free variable optimal values

Free Variables	Value	Units
Aircraft		
AR_w	9.298	[−]
C_D	0.02911	[−]
C_L	0.5616	[−]
C_{L_w}	0.5616	[−]
$C_{L_{aw}}$	5.616	[−]
D	2.909e+04	[N]
D_{fuse}	1.038e+04	[N]
D_{ht}	1011	[N]
D_{vt}	944.6	[N]
D_{wing}	1.675e+04	[N]
L_w	5.612e+05	[N]
$L_{v_{max}}$	6.399e+05	[N]
S_w	98	[m^2]
V_{TO}	73.38	[$\frac{\mathrm{m}}{\mathrm{s}}$]
V_∞	231.7	[$\frac{\mathrm{m}}{\mathrm{s}}$]
W	5.612e+05	[N]
W_{fuel}	5.164e+04	[N]
W_{fuse}	1.616e+05	[N]
W_{ht}	5196	[N]
W_{lg}	1.212e+04	[N]
W_{pay}	1.616e+05	[N]
W_{vt}	1963	[N]
W_{wing}	1.204e+05	[N]
W_{zf}	4.729e+05	[N]
$\bar{c}_w$	3.728	[m]
$\frac{L}{D}$	19.29	[−]
ξ	0.07295	[−]
b_{vt}	3.707	[m]
c_{vt}	9.146	[m]
h_{hold}	0.7714	[m]
l_{fuse}	52.47	[m]
p_{λ_v}	1.6	[−]
w_{fuse}	3.71	[m]
x_w	19.6	[m]
$x_{CG_{eng}}$	19.6	[m]
$x_{CG_{fu}}$	17.67	[m]
$x_{CG_{ht}}$	51.32	[m]
$x_{CG_{lg}}$	19.3	[m]
$x_{CG_{vt}}$	47.9	[m]
$x_{CG_{wing}}$	19.6	[m]
x_{CG}	17.6	[m]
x_{TO}	1524	[m]
x_{up}	29.61	[m]
y	0.1318	[−]
z_{bre}	0.1036	[−]
Fuselage, Aircraft		
A_{floor}	0.05549	[m^2]
A_{fuse}	10.81	[m^2]
A_{hold}	1.68	[m^2]
A_{skin}	0.01087	[m^2]
$D_{friction}$	9646	[N]
$D_{upsweep}$	736.7	[N]
FF	1.057	[−]
M_{floor}	4.442e+05	[$\mathrm{N} \cdot \mathrm{m}$]
P_{floor}	1.137e+06	[N]
R_{fuse}	1.855	[m]
S_{bulk}	21.62	[m^2]
S_{floor}	5.686e+05	[N]
S_{nose}	49.82	[m^2]
V_{bulk}	0.02016	[m^3]
V_{cabin}	315	[m^3]

V_{cargo}	6.796	$[\mathrm{m}^3]$
V_{cone}	0.1657	$[\mathrm{m}^3]$
V_{cyl}	0.2653	$[\mathrm{m}^3]$
V_{floor}	0.1734	$[\mathrm{m}^3]$
V_{hold}	41.02	$[\mathrm{m}^3]$
V_{lugg}	18.24	$[\mathrm{m}^3]$
V_{nose}	0.04645	$[\mathrm{m}^3]$
W_{apu}	5657	$[\mathrm{N}]$
W_{buoy}	1517	$[\mathrm{N}]$
W_{cone}	7898	$[\mathrm{N}]$
W_{floor}	9865	$[\mathrm{N}]$
W_{insul}	4307	$[\mathrm{N}]$
W_{lugg}	1.79e+04	$[\mathrm{N}]$
W_{padd}	6.465e+04	$[\mathrm{N}]$
W_{pass}	1.337e+05	$[\mathrm{N}]$
W_{seat}	2.79e+04	$[\mathrm{N}]$
W_{shell}	1.582e+04	$[\mathrm{N}]$
W_{skin}	8791	$[\mathrm{N}]$
W_{window}	1.062e+04	$[\mathrm{N}]$
λ_{cone}	0.4	$[-]$
ϕ	0.07883	$[-]$
ρ_{cabin}	0.8711	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
σ_x	3.831e+07	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$
σ_θ	1.034e+08	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$
τ_{cone}	1.034e+08	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$
f	14.14	$[-]$
h_{floor}	0.08359	$[\mathrm{m}]$
l_{cone}	22.86	$[\mathrm{m}]$
l_{floor}	28.12	$[\mathrm{m}]$
l_{nose}	5.2	$[\mathrm{m}]$
l_{shell}	24.41	$[\mathrm{m}]$
n_{pass}	167	$[-]$
n_{rows}	31	$[-]$

t_{shell}	0.001259	$[\mathrm{m}]$
t_{skin}	0.0009324	$[\mathrm{m}]$
w_{floor}	3.125	$[\mathrm{m}]$
$xVbulk$	0.5969	$[\mathrm{m}^4]$
$xVcyl$	4.617	$[\mathrm{m}^4]$
$xVnose$	0.1208	$[\mathrm{m}^4]$
$xWapu$	2.069e+05	$[\mathrm{N} \cdot \mathrm{m}]$
$xWcone$	3.242e+05	$[\mathrm{N} \cdot \mathrm{m}]$
$xWfix$	2.802e+04	$[\mathrm{N} \cdot \mathrm{m}]$
$xWfloor$	1.717e+05	$[\mathrm{N} \cdot \mathrm{m}]$
$xWfuse$	2.856e+06	$[\mathrm{N} \cdot \mathrm{m}]$
$xWinsul$	7.496e+04	$[\mathrm{N} \cdot \mathrm{m}]$
$xWpadd$	1.125e+06	$[\mathrm{N} \cdot \mathrm{m}]$
$xWseat$	4.856e+05	$[\mathrm{N} \cdot \mathrm{m}]$
$xWshell$	2.543e+05	$[\mathrm{N} \cdot \mathrm{m}]$
$xWskin$	1.413e+05	$[\mathrm{N} \cdot \mathrm{m}]$
$xWwindow$	1.848e+05	$[\mathrm{N} \cdot \mathrm{m}]$
x_{shell1}	5.2	$[\mathrm{m}]$
x_{shell2}	29.61	$[\mathrm{m}]$

HorizontalTail, Aircraft

AR_h	7.707	$[-]$
C_{D_h}	0.006697	$[-]$
$C_{D_{0_h}}$	0.005348	$[-]$
C_{L_h}	0.1785	$[-]$
$C_{L_{ah_0}}$	5.348	$[-]$
$C_{L_{ah}}$	2.963	$[-]$
K_f	0.4434	$[-]$
L_h	2.695e+04	$[\mathrm{N}]$
L_{max_h}	4.702e+05	$[\mathrm{N}]$
Re_{c_h}	1.001e+07	$[-]$
$S.M.$	0.05	$[-]$
S_h	14.81	$[\mathrm{m}^2]$

Δx_{lead_h}	32.57	[m]
Δx_{trail_h}	34.88	[m]
α	0.06026	[−]
$\bar{c}_{ht}$	1.591	[m]
λ_h	0.2	[−]
τ_h	0.15	[−]
b_{ht}	10.68	[m]
c_{root_h}	2.31	[m]
c_{tip_h}	0.462	[m]
e_h	0.9752	[−]
$f(\lambda_h)$	0.0033	[−]
l_{ht}	34.73	[m]
p_{ht}	1.4	[−]
q_{ht}	1.2	[−]
$y_{\bar{c}_{ht}}$	3.052	[m]

LandingGear, Aircraft

B	15.75	[m]
E_{land}	2.657e+05	[J]
F_{w_m}	4458	[−]
F_{w_n}	400.9	[−]
I_m	5.007e-06	$[\mathrm{m}^4]$
I_n	7.204e-07	$[\mathrm{m}^4]$
L_m	4.489e+05	[N]
L_n	1.122e+05	[N]
$L_{n_{dyn}}$	4.834e+04	[N]
L_{w_m}	1.122e+05	[N]
L_{w_n}	5.612e+04	[N]
S_{sa}	0.2959	[m]
T	5.662	[m]
W_{mg}	1.1e+04	[N]
W_{ms}	1048	[N]
W_{mw}	1781	[N]
W_{ng}	1113	[N]
W_{ns}	85.63	[N]
W_{nw}	410.8	[N]
$W_{wa,m}$	200.2	[lbf]
$W_{wa,n}$	46.18	[lbf]
Δx_m	3.149	[m]
Δx_n	12.6	[m]
$\tan(\phi)$	0.2679	[−]
$\tan(\psi)$	1.963	[−]
$d_{nacelle}$	2.05	[m]
d_{oleo}	0.3119	[m]
d_{t_m}	39.72	[in]
d_{t_n}	31.78	[in]
l_m	2.375	[m]
l_n	1.627	[m]
l_{oleo}	0.7399	[m]
r_m	0.04181	[m]
r_n	0.04592	[m]
t_m	0.02181	[m]
t_n	0.002368	[m]
w_{t_m}	0.3439	[m]
w_{t_n}	0.2751	[m]
x_m	20.75	[m]
x_n	5	[m]
y_m	2.831	[m]

VerticalTail, Aircraft

A_{fan}	2.405	$[\mathrm{m}^2]$
A_{vt}	0.7093	[−]
$C_{D_{vis}}$	0.004781	[−]
$C_{L_{vt}}$	0.3905	[−]
D_{wm}	3609	[N]
$L_{max_{vt}}$	1.28e+06	[N]

L_{vt}	2.271e+04	[N]
Re_{vt}	4.099e+07	[−]
S	38.76	$[\mathrm{m}^2]$
S_{vt}	19.38	$[\mathrm{m}^2]$
W_{struct}	3927	[N]
Δx_{lead_v}	25.73	[m]
Δx_{trail_v}	34.88	[m]
$\bar{c}_{vt}$	6.519	[m]
λ_{vt}	0.3	[−]
τ_{vt}	0.09958	[−]
b	7.415	[m]
$c_{root_{vt}}$	9.146	[m]
$c_{tip_{vt}}$	2.744	[m]
l_{vt}	28.2	[m]
p_{vt}	1.6	[−]
q_{vt}	1.3	[−]
$z_{\bar{c}_{vt}}$	1.004	[m]

Wing, Aircraft

C_{D_w}	0.01677	[−]
$C_{D_{p_w}}$	0.005637	[−]
L_{max_w}	3.112e+06	[N]
Re_w	2.344e+07	[−]
$V_{fuel,max}$	93.38	$[\mathrm{m}^3]$
α_w	0.1	[−]
$\bar{A}_{fuel,max}$	0.069	[−]
λ	0.2	[−]
τ_w	0.15	[−]
b_w	30.19	[m]
c_{root}	5.411	[m]
c_{tip}	1.082	[m]
e	0.9702	[−]
$f(\lambda_w)$	0.0033	[−]
p_w	1.4	[−]
q_w	1.2	[−]
$y_{\bar{c}_w}$	8.624	[m]

WingBox, HorizontalTail, Aircraft

I_{cap}	2.377e-05	[−]
M_r	2.114e+05	[N]
W_{cap}	3382	[N]
W_{web}	329.3	[N]
ν	0.8612	[−]
t_{cap}	0.005417	[−]
t_{web}	0.002345	[−]

WingBox, VerticalTail, Aircraft

A	1.419	[−]
I_{cap}	7.456e-07	[−]
M_r	1.21e+05	[N]
W_{cap}	2107	[N]
W_{web}	697.4	[N]
ν	0.8225	[−]
t_{cap}	0.0003581	[−]
t_{web}	0.0007934	[−]

WingBox, Wing, Aircraft

I_{cap}	3.458e-05	[−]
M_r	1.688e+06	[N]
W_{cap}	7.984e+04	[N]
W_{web}	6158	[N]
ν	0.8612	[−]
t_{cap}	0.008251	[−]
t_{web}	0.002828	[−]

Table B.2: Fixed variable values

Constants	Value	Units
Aircraft		
$C_{L_{max}}$	2.5	$[-]$
M	0.78	$[-]$
N_{lift}	2	$[-]$
$Range$	3000	[nmi]
T_e	1.29e+05	[N]
V_{ne}	144	$[\frac{\mathrm{m}}{\mathrm{s}}]$
W_{eng}	1e+04	[N]
μ	1.4e-05	$[\frac{\mathrm{N \cdot s}}{\mathrm{m}^2}]$
ρ	0.38	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
ρ_0	1.225	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
ρ_{cap}	2700	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
ρ_{web}	2700	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
$\sigma_{max,shear}$	1.67e+08	[Pa]
σ_{max}	2.5e+08	[Pa]
a	297	$[\frac{\mathrm{m}}{\mathrm{s}}]$
c_T	0.3	$[\frac{\mathrm{lb}}{(\mathrm{hr \cdot lbf})}]$
d_{fan}	1.75	[m]
$f_{w,add}$	0.4	$[-]$
g	9.81	$[\frac{\mathrm{m}}{\mathrm{s}^2}]$
l_r	5000	[ft]
r_h	0.75	$[-]$
w	0.5	$[-]$
y_{eng}	4.83	[m]
Fuselage, Aircraft		
LF	0.898	$[-]$
N_{land}	6	$[-]$
R	287	$[\frac{\mathrm{J}}{(\mathrm{K \cdot kg})}]$
SPR	6	$[-]$
T_{cabin}	300	[K]
W''_{floor}	60	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$
W''_{insul}	22	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$
W'_{seat}	150	[N]
W'_{window}	435	$[\frac{\mathrm{N}}{\mathrm{m}}]$
$W_{avg.pass}$	180	[lbf]
W_{cargo}	1e+04	[N]
$W_{carryon}$	15	[lbf]
$W_{checked}$	40	[lbf]
W_{fix}	3000	[lbf]
Δh	1	[m]
Δp	5.2e+04	[Pa]
ρ_∞	0.38	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
ρ_{bend}	2700	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
ρ_{cargo}	150	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
ρ_{cone}	2700	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
ρ_{floor}	2700	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
ρ_{lugg}	100	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
ρ_{skin}	2700	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
σ_{floor}	2.069e+08	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$
σ_{skin}	1.034e+08	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$
τ_{floor}	2.069e+08	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$
f_{apu}	0.035	$[-]$
f_{fadd}	0.2	$[-]$
f_{frame}	0.25	$[-]$
$f_{lugg,1}$	0.4	$[-]$
$f_{lugg,2}$	0.1	$[-]$
f_{padd}	0.4	$[-]$
f_{string}	0.35	$[-]$
n_{seat}	186	$[-]$
p_s	31	[in]
p_{cabin}	7.5e+04	[Pa]
r_E	1	$[-]$

w_{aisle}	0.51	[m]
w_{seat}	0.5	[m]
w_{sys}	0.1	[m]
$xapu$	120	[ft]
$xfix$	2.1	[m]

HorizontalTail, Aircraft

$C_{L_{hmax}}$	2.5	[−]
$C_{m_{fuse}}$	0.05	[−]
$S.M._{min}$	0.05	[−]
Δx_w	2	[m]
$\alpha_{max,h}$	0.1	[−]
η_h	0.97	[−]
η_{ht}	0.9	[−]
$\tan(\Lambda_{ht})$	0.5774	[−]
$\lvert C_{m_{ac}}\rvert$	0.1	[−]

LandingGear, Aircraft

E	205	[GPa]
K	2	[−]
N_s	2	[−]
η_s	0.8	[−]
λ_{LG}	2.5	[−]
ρ_{st}	7850	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
σ_{y_c}	4.7e+08	[Pa]
$\tan(\gamma)$	0.08749	[−]
$\tan(\phi_{min})$	0.2679	[−]
$\tan(\psi_{max})$	1.963	[−]
$\tan(\theta_{max})$	0.2679	[−]
$f_{add,m}$	1.5	[−]
$f_{add,n}$	1.5	[−]
$h_{nacelle}$	0.5	[m]
n_{mg}	2	[−]
n_{wps}	2	[−]
p_{oleo}	1800	$[\frac{\mathrm{lbf}}{\mathrm{in}^2}]$
$t_{nacelle}$	0.15	[m]
w_{ult}	10	$[\frac{\mathrm{ft}}{\mathrm{s}}]$
z_{CG}	2	[m]
z_{wing}	0.5	[m]

VerticalTail, Aircraft

$C_{D_{wm}}$	0.5	[−]
$C_{L_{vmax}}$	2.6	[−]
V_1	70	$[\frac{\mathrm{m}}{\mathrm{s}}]$
ρ_c	0.38	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
ρ_{TO}	1.225	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
$\tan(\Lambda_{vt})$	0.8391	[−]
$c_{l_{vt}}$	0.5	[−]
e_v	0.8	[−]

Wing, Aircraft

$C_{L_{wmax}}$	2.5	[−]
$\alpha_{max,w}$	0.1	[−]
$\cos(\Lambda)$	0.866	[−]
η_w	0.97	[−]
ρ_{fuel}	817	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
$\tan(\Lambda)$	0.5774	[−]

Table B.3: Fixed variable sensitivities

Sensitivities	Value	Units
Wing, Aircraft		
$C_{L_{wmax}}$	0.3192	$[-]$
$\tan(\Lambda)$	0.06735	$[-]$
η_w	-0.1465	$[-]$
$\alpha_{max,w}$	-0.2513	$[-]$
VerticalTail, Aircraft		
ρ_c	0.03472	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
$C_{L_{vmax}}$	0.0266	$[-]$
e_v	-0.01605	$[-]$
ρ_{TO}	-0.04466	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
$c_{l_{vt}}$	-0.05721	$[-]$
V_1	-0.1425	$[\frac{\mathrm{m}}{\mathrm{s}}]$
LandingGear, Aircraft		
$f_{add,m}$	0.01242	$[-]$
$\tan(\theta_{max})$	0.01065	$[-]$
n_{mg}	-0.02991	$[-]$
n_{wps}	-0.0339	$[-]$
HorizontalTail, Aircraft		
Δx_w	0.05261	[m]
$C_{L_{hmax}}$	0.01468	$[-]$
$\tan(\Lambda_{ht})$	0.01088	$[-]$
η_h	-0.02993	$[-]$
η_{ht}	-0.04177	$[-]$
Fuselage, Aircraft		
n_{seat}	0.7349	$[-]$
LF	0.5094	$[-]$
$W_{avg.pass}$	0.4494	[lbf]
w_{seat}	0.3223	[m]
ρ_∞	0.3064	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
SPR	0.1623	$[-]$
p_s	0.16	[in]
f_{padd}	0.1481	$[-]$
Δh	0.1033	[m]
W'_{seat}	0.06547	[N]
$W_{checked}$	0.05992	[lbf]
w_{aisle}	0.0548	[m]
$f_{lugg,1}$	0.03995	$[-]$
ρ_{skin}	0.03587	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
Δp	0.03587	[Pa]
W_{cargo}	0.03361	[N]
W_{fix}	0.02683	[lbf]
W'_{window}	0.02433	$[\frac{\mathrm{N}}{\mathrm{m}}]$
ρ_{cone}	0.02152	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
w_{sys}	0.02149	[m]
$f_{lugg,2}$	0.01997	$[-]$
f_{apu}	0.01495	$[-]$
W''_{floor}	0.01208	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$
f_{string}	0.01116	$[-]$
N_{land}	0.01052	$[-]$
ρ_{floor}	0.01052	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$
σ_{skin}	-0.05739	$[\frac{\mathrm{N}}{\mathrm{m}^2}]$
Aircraft		
g	1.435	$[\frac{\mathrm{m}}{\mathrm{s}^2}]$
c_T	1.062	$[\frac{\mathrm{lb}}{(\mathrm{hr}\cdot\mathrm{lbf})}]$
$Range$	1.062	[nmi]
V_{ne}	0.721	$[\frac{\mathrm{m}}{\mathrm{s}}]$
N_{lift}	0.339	$[-]$
ρ_0	0.3339	$[\frac{\mathrm{kg}}{\mathrm{m}^3}]$

ρ_{cap}	0.2748	$[\frac{\text{kg}}{\text{m}^3}]$
$f_{w,add}$	0.0849	$[-]$
y_{eng}	0.07245	$[\text{m}]$
T_e	0.07126	$[\text{N}]$
μ	0.04465	$[\frac{\text{N}\cdot\text{s}}{\text{m}^2}]$
W_{eng}	0.02314	$[\text{N}]$
r_h	0.02241	$[-]$
ρ_{web}	0.02241	$[\frac{\text{kg}}{\text{m}^3}]$
$\sigma_{max,shear}$	-0.02241	$[\text{Pa}]$
w	-0.04182	$[-]$
σ_{max}	-0.3166	$[\text{Pa}]$
ρ	-0.4166	$[\frac{\text{kg}}{\text{m}^3}]$
a	-1.164	$[\frac{\text{m}}{\text{s}}]$
M	-1.312	$[-]$

Bibliography

[1] John D Anderson. Fundamentals of aerodynamics, 2001.

[2] John David Anderson. *Introduction to flight*, volume 199. McGraw-Hill Boston, 2005.

[3] MOSEK ApS. *The MOSEK C optimizer API manual. Version 7.1 (Revision 41).*, 2015.

[4] Boeing Commercial Airplanes. 737 airplane characteristics for airport planning. *Seattle, Washington, USA*, 2007.

[5] Boeing Commercial Airplanes. 737 max airplane characteristics for airport planning. *Seattle, Washington, USA*, 2015.

[6] Stephen Boyd. Sequential convex programming. *Lecture Notes, Stanford University*, 2008.

[7] Stephen Boyd, Seung-Jean Kim, Lieven Vandenberghe, and Arash Hassibi. A tutorial on geometric programming. *Optimization and engineering*, 8(1):67–127, 2007.

[8] Chris Brady. The boeing 737 technical site. `http://www.b737.org.uk`, 2015 (accessed November 1st 2015).

[9] Edward Burnell and Warren Hoburg. Gpkit. `https://github.com/convexopt/gpkit`, 2016. Version 0.4.0.

[10] Richard H Byrd, Jean Charles Gilbert, and Jorge Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–185, 2000.

[11] Richard H Byrd, Mary E Hribar, and Jorge Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

[12] Sonny T Chai and William H Mason. Landing gear integration in aircraft conceptual design. Master's thesis, Virginia Polytechnic Institute and State University, 1996.

[13] Norman S Currey. *Landing Gear Design Handbook*. Lockheed-Georgia Company, 1984.

[14] Olivier de Weck, Jeremy Agte, J Sobieski, Paul Arendsen, Alan Morris, and Martin Spieck. State-of-the-art and future trends in multidisciplinary design optimization. In *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA*, volume 1905, 2007.

[15] Mark Drela. Xfoil: An analysis and design system for low reynolds number airfoils. In *Low Reynolds number aerodynamics*, pages 1–12. Springer, 1989.

[16] Mark Drela. *TASOPT 2.08 Transport Aircraft System OPTimization*, 2011.

[17] Richard James Duffin, Elmor L Peterson, and Clarence Zener. *Geometric programming: theory and application*. Wiley New York, 1967.

[18] John W Harris and Horst Stöcker. *Handbook of mathematics and computational science*. Springer Science & Business Media, 1998.

[19] Ryan P Henderson, JRRA Martins, and Ruben E Perez. Aircraft conceptual design for optimal environmental performance. *Aeronautical Journal*, 116(1175):1, 2012.

[20] William F Hilton. *High-speed aerodynamics*. Longmans, Green, 1951.

[21] Warren Hoburg and Pieter Abbeel. Geometric programming for aircraft design optimization. *AIAA Journal*, 52(11):2414–2426, 2014.

[22] Warren Hoburg, Philippe Kirschen, and Pieter Abbeel. Fitting geometric programming models to data. *Optimization and Engineering*.

[23] I Kroo. Pass, program for aircraft synthesis studies. *Software Package, Desktop Aeronautics, Palo Alto, CA*, 2005.

[24] Ilan Kroo, Steve Altus, Robert Braun, Peter Gage, and Ian Sobieski. Multidisciplinary optimization methods for aircraft preliminary design. *AIAA paper*, 4325:1994, 1994.

[25] Ilan Kroo and Richard Shevell. Aircraft design: Synthesis and analysis. *Desktop Aeronautics Inc., Textbook Version 0.99*, 2001.

[26] Joaquim RRA Martins and Andrew B Lambe. Multidisciplinary design optimization: a survey of architectures. *AIAA journal*, 51(9):2049–2075, 2013.

[27] MATLAB. *version 8.5.0 (R2015a)*. The MathWorks Inc., Natick, Massachusetts, 2015.

[28] Sanjay Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on optimization*, 2(4):575–601, 1992.

[29] Mihaela Niţă and Dieter Scholz. *Estimating the oswald factor from basic aircraft geometrical parameters.* Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV, 2012.

[30] Jorge Nocedal and Stephen Wright. *Numerical optimization.* Springer Science & Business Media, 2006.

[31] Daniel P Raymer. Aircraft design: A conceptual approach, 1992.

[32] William Sutherland. Lii. the viscosity of gases and molecular force. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 36(223):507–531, 1893.

[33] Egbert Torenbeek. *Synthesis of subsonic aircraft design.* 1982.

[34] Robert J Vanderbei and David F Shanno. An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, 13(1-3):231–252, 1999.

[35] Richard A Waltz, José Luis Morales, Jorge Nocedal, and Dominique Orban. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Mathematical programming*, 107(3):391–408, 2006.