Aerospace Composite Manufacturing Cost Models as Geometric Programs

by

Scott T. Nill

Submitted to the Department of Mechanical Engineering
on May 11, 2018, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

ABSTRACT

The introduction of large, composite transport aircraft, such as the Airbus A350 and the Boeing 787, has been fraught with billions of dollars of production cost overruns. This research develops a novel approach to manufacturing cost modeling during the conceptual design phase using Geometric Programming (GP). A new formulation of a closed queuing network as a GP is presented to capture the crucial cost trade-offs between capacity and inventory. Additionally, GP models are presented for modeling unit processes in composite manufacturing and for modeling cost accounting metrics. Applied to the challenges of conceptual design for composite aircraft, the cost models can be used as a tool to help inform decisions about which manufacturing process to use and what type of supply chain should be deployed. The special sensitivity-analysis properties of the GP solutions can be exploited to explain how different aspects of the design drive manufacturing costs and to find highly sensitive areas of the trade-space that would have a large impact on cost if the design needed to be altered. The framework is demonstrated for fast but informative analyses of process trade-offs in composite fuselage fabrication.

Thesis Committee
Prof. David Hardt (Chair), Department of Mechanical Engineering
Prof. Warren Hoburg (Thesis Advisor), Department of Aeronautics and Astronautics
Prof. Stephen Graves, Department of Mechanical Engineering & Sloan School of Management
# CONTENTS

1. Introduction ........................................................................................................................................................................ 7
   1.1. Motivation: Composites Costs Surprise Aerospace ........................................................................................................ 7
   1.2. Approach: Improved Concept-Phase Cost Models ........................................................................................................ 7
   1.3. Outline .................................................................................................................................................................................. 10

2. Relevant Concepts .................................................................................................................................................................. 12
   2.1. Cost Accounting ................................................................................................................................................................. 12
       2.1.1. Fundamentals of Cost Estimation ................................................................................................................................. 14
       2.1.2. Cost Modeling for Commercial Aircraft ....................................................................................................................... 15
   2.2. Conceptual Aircraft Design: Different Approaches ......................................................................................................... 23
   2.3. Composite Aerospace Production Systems ....................................................................................................................... 26
   2.4. Toward Manufacturing Cost Estimation in Conceptual Aircraft Design ............................................................................... 31

   3.1. Background ............................................................................................................................................................................. 35
       3.1.1. GP Formulation ............................................................................................................................................................... 35
       3.1.2. Solving GPs ....................................................................................................................................................................... 37
       3.1.3. Feasibility, trade-off, and sensitivity analysis with GPs ................................................................................................. 39
   3.2. Opportunity ............................................................................................................................................................................. 42
   3.3. Vision ...................................................................................................................................................................................... 43

4. Production Line Flow Model Development ................................................................................................................................ 44
   4.1. Prior Art: Manufacturing System Models .......................................................................................................................... 44
       4.1.1. Discrete Event Simulations ............................................................................................................................................ 45
       4.1.2. Stochastic Models based on Queueing Theory ............................................................................................................ 45
   4.2. CONWIP Production Line Model ........................................................................................................................................ 45
   4.3. Approximations for CONWIP production line performance ............................................................................................... 49
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.1</td>
<td>W hitt's Queueing Network Analyzer (QNA)</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>4.3.2</td>
<td>Models of CONWIP production lines</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>4.3.3</td>
<td>Approximating multi-server production cells</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td>4.4</td>
<td>Formulating the production line model as a GP</td>
<td>57</td>
<td></td>
</tr>
<tr>
<td>4.5</td>
<td>Evaluating the GP models</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>4.5.1</td>
<td>Comparing Approximations to Simulations</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>4.5.2</td>
<td>Error from the approximations</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>4.6</td>
<td>Summary</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Cost Model Development</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Scope of this Cost Model: Manufacturing Systems</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td>Cost breakdown</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>5.3</td>
<td>GP formulation for total program cost</td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>5.3.1</td>
<td>Non-recurring, initial costs</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>5.3.2</td>
<td>Discounted cash flow model for recurring costs</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>5.3.3</td>
<td>Total cost equation formulation</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>5.4</td>
<td>Summary</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Unit Process Model Development</td>
<td>76</td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>GP Formulations of ACCEM Process Models</td>
<td>76</td>
<td></td>
</tr>
<tr>
<td>6.2</td>
<td>GP Formulations of COSTADE Process Models</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>6.3</td>
<td>Additional Model for Resin Infusion Unit Process</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>6.4</td>
<td>Modeling Process Time Variance</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>6.5</td>
<td>Summary</td>
<td>83</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>GP Optimization: Closing the Loop around the Factory</td>
<td>84</td>
<td></td>
</tr>
<tr>
<td>7.1</td>
<td>Design and Trade Spaces</td>
<td>84</td>
<td></td>
</tr>
<tr>
<td>7.1.1</td>
<td>Design Space</td>
<td>84</td>
<td></td>
</tr>
<tr>
<td>7.1.2</td>
<td>Feasible Subspace</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>7.1.3</td>
<td>Trade Subspace and the Efficiency Frontier</td>
<td>85</td>
<td></td>
</tr>
</tbody>
</table>
12.4.3. Layup Skin.......................................................... 120
12.4.4. Bag and Prep....................................................... 121
12.4.5. Autoclave Cure ................................................. 121
12.5. Resin Infusion Unit Process Models .................... 122
  12.5.1. Load Stringers ............................................... 122
  12.5.2. Layup Skin .................................................. 123
  12.5.3. Locate OML Caul ........................................... 124
  12.5.4. Bag and Prep ............................................... 125
  12.5.5. Infuse and cure ............................................. 126
1. INTRODUCTION

1.1. Motivation: Composites Costs Surprise Aerospace

In the opening years of the 21st century, the introduction of new, composite aircraft for commercial transport has been met with significant delays and cost overruns. The Boeing 787 was introduced in 2004 with an estimated $15B USD development cost. Over the course of the program, the first delivery of a 787 to a customer was delayed by 3 years and is reported to have cost in excess of $30B USD [1]. The comparable Airbus A350, launched two years later in 2006, was reported to have cost $15B USD as well. Many of the issues of high cost and program delays arise from the disconnect of an aircraft’s design process from the complex logistics that govern the performance of the manufacturing systems [1] and the growing complexity of these couplings since the introduction of composites [2]. Design has tended to rigorously focus on aircraft performance without the same regard for manufacturing and supply chain. From an organizational standpoint, the introduction of composites severely disrupted the manufacturing intuition which had been developed based on nearly a century of building metallic aircraft.

Often a disciplined aircraft design program, beginning with a conceptual phase to explore a wide variety of design approaches, is not connected to an equally disciplined approach to designing the production system. In many cases, only after many of the key decisions regarding the aircraft are made, is the production system designed. The net result is often a disconnect between the cost drivers in the aircraft design and the understanding of what these costs will arise during manufacturing.

Tools have been developed for the private sector to improve cost estimates for new composite designs and for better integrating manufacturing cost estimation into the design process. The work presented in this thesis contributes faster estimates of cost, explicit incorporation of the dynamics seen in complex manufacturing systems, and more detailed sensitivity data about the manufacturing cost drivers.

1.2. Approach: Improved Concept-Phase Cost Models

One long-held solution to the problem of unexpected, high production cost is to include manufacturing cost as a consideration during the conceptual phase [3], [4], [5]. During the conceptual design phase, there is little penalty to changing a design or manufacturing plan as the design only exists on paper or in the
computer. However, during the conceptual phase, nearly all of the real manufacturing costs remain uncertain. Figure 1 displays a notional systems engineering perspective of the co-evolution of cost and design throughout a typical design process. Improved cost models can have their greatest impact during early design phases when a significant portion of cost are committed but where the cost of change is minimal.

Figure 1. Cost commitment curve shows the opportunity for cost savings through more informed conceptual design from [2] citing [3], [4]
Figure 2. During the standard aircraft design process, conceptual design corresponds to a high degree of freedom in the design but is accompanied by low actual knowledge of what the final design will be. [5]

This research has produced a method for modeling manufacturing systems as geometric programs (GP) to improve the manufacturing cost modeling throughout the design process. This research further shows the particular suitability of the GP cost estimation to the conceptual design process. The research produced new approximations of CONWIP production lines that are compatible with the required mathematical structure of geometric programs. With the production system dynamics expressed as GPs, the cost models can be solved as convex optimization problems. Convex optimization allows large system models to be solved extremely efficiently on a personal computer. Non-convex optimization problems often require (as opposed to requiring a computing cluster to be solved quickly) and provides detailed sensitivity information on the variables at the optimal solution. The fast solutions and the sensitivity data of the cost model formulations provide a wealth of information during all phases of design but can be especially helpful during conceptual design phases.
When this research is applied to a case study of designing a fuselage for a conceptual airplane, the approach is shown to provide valuable comparison data between production concepts and a clearer picture of how different aspects of the design influence the manufacturing costs. In another case study, GP manufacturing cost models are combined with GP models for the design of an unmanned aircraft system. This demonstrates the applicability to the concurrent design of the aircraft and manufacturing system to minimize manufacturing costs. Together, these case studies demonstrate the improved cost performance from including manufacturing models during the conceptual design process.

The nature of this research necessarily involves discussion regarding costs in manufacturing. This thesis does not attempt to present every cost which may be incurred in the manufacturing process. Equally, it is not about cataloging every type of cost. Rather, the thesis presents the method and framework for coupling costs arising from manufacturing systems design into the early design process of complex systems. Ultimately, the reader must exercise professional judgement regarding which costs should be included in the models.

1.3. Outline

The approach centers on implementing analytical models for the production cost as a Geometric Program based on manufacturing process times. Chapter 2 reviews relevant concepts in cost accounting, cost modeling, composite airframe production, as well as the state of the art in production modeling tools. Chapter 3 gives an introduction to geometric programming (GP) and presents a vision for this dissertation. Chapters 4, 5 and 6 guide the reader through the analytical and numerical derivation of flow models, process models and cost models that go into GP optimizations. Chapter 4 presents a model representation of the factory and new GP approximations of the factory performance. Chapter 5 presents GP models for calculating production costs within the factory models. Chapter 6 presents models of composite fabrication process which predict processing times based on part geometries. Chapter 7 integrates GP models from the previous chapters to solve convex optimization problems to forecast production costs. Finally, Chapter 8 presents case studies where the method is used to inform the design process.

The primary contributions of this thesis are outlined in Table 1:

<table>
<thead>
<tr>
<th>Table 1. Thesis contributions.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GP Models for Composite Manufacturing Cost Estimation</strong></td>
</tr>
<tr>
<td>• GP flow approximations for CONWIP lines</td>
</tr>
<tr>
<td>• GP models for total production cost</td>
</tr>
</tbody>
</table>
• GP process models for composite fabrication

**Deployment of GP Manufacturing Models**

• Performance analysis of existing production line
• Study manufacturing alternatives for new fuselage design

**Generalizable framework**

• Library of open source GP production models
• Application to other types of production systems
• Cloud deployment on python
• Visualization tool for constraint sensitivities
2. **RELEVANT CONCEPTS**

It’s often said, whenever Boeing or Airbus decides to launch a new product, they bet the company. In 2016, the deferred production cost of Boeing’s new, composite 787 ballooned to over $28 Billion USD. Compared to Boeing’s $65 Billion USD in revenue from the commercial airplanes business, the deferred production expense gives some indication of the scale of manufacturing costs in aerospace.

The particular impact of this research is that it brings together many disparate fields, each complex in their own right. This chapter seeks to introduce the reader to the relevant concepts from cost accounting, composite manufacturing, and aircraft design that will be useful for understanding the methods this thesis presents.

2.1. **Cost Accounting**

The goal of a capitalist, commercial manufacturing enterprise, is to generate profit by producing goods and selling them at a higher price than it costs to produce. This section gives an overview of cost accounting: the tools and practices currently in-use for understanding and predicting manufacturing costs.

Many factors influence the cost of goods produced. The enterprise makes decisions based on the knowledge available on how to deploy resources to maximize profit. Operational decisions are made daily, such as allocating labor, tactical decisions are made on an intermediate term, such as determining production schedules and down-time, and strategic decisions are made on a long-term time scale, such as building a new factory, or which series of processes should be used to manufacture the product. On any scale, the core of the decision-making process is to consider the cost-benefit trade-off among the options available to the decision-maker.
In a large enterprise, cost becomes a surprisingly abstract concept. In the US, there are often two lenses under which costs are considered. When a company has to externally communicate its finances, to governmental entities, to the public, or to investors, the company will utilize the Generally Accepted Accounting Principles (GAAP). For internal purposes, a company is not obligated to use any particular standard for accounting and will often develop specialized methods specific to the company, industry, or product.

Internally, manufacturers must consider revenues and costs on a product basis. At this level the cost is based on the resources required to produce the product. The costs of the resources are considered either direct or indirect. Direct costs are those which can be attributed solely to the production of a particular good. For example, if a commercial bakery produces cookies some of the time, the ingredients going into the cookies (flour, sugar, chocolate chips) and the labor required are direct costs.

By contrast, indirect costs are those costs which are required to produce the goods but cannot be assigned to a specific product. If the example bakery above should produce delectable treats other than cookies, costs of the ovens, the building, or the administrative staff are indirect costs to the production of cookies.

Another consideration for production costs are fixed and variable costs. Variable costs are a function of the quantity of the products produced whereas fixed costs will be incurred regardless of the ultimate production quantity. In the example of producing cookies at the bakery, the ingredients are variable costs.
since producing more cookies will require more ingredients, whereas the cost to clean all the equipment at the end of cookie production is a fixed cost since cleaning must happen even if only one cookie is produced.

Another consideration is if costs are represented as an asset or an expense in the financial accounting records. Generally, if a cost results in long-term benefits, such as the purchase as a new piece of manufacturing equipment, it is considered an asset. On the other hand, if a cost only has immediate benefit, such as payment to an external accountant to prepare tax documents, the cost is considered an expense.

Costs can be classified as recurring or non-recurring to capture the nature of the cost in time. For example, labor is a recurring cost because employees are paid on regular intervals. An example of a non-recurring cost would be the expansion of a factory. Classifying something as a non-recurring cost doesn’t necessarily mean that a similar cost won’t be incurred in the future, it rather indicates that the cost is extraordinary and is not expected to occur on a regular basis.

Often business considers the time value of money when calculating costs. This reflects the idea that one dollar today is worth more than a dollar a year from now. This is because money now could be invested and thus would be worth more than a dollar in the future.

Even in the simple cookie example, there is room for interpretation of costs. It is easy to imagine how complexity of cost calculation grows when one considers a product like a commercial airliner. Despite the complexity, it is essential for a business to track and account for production costs.

2.1.1. Fundamentals of Cost Estimation

The fundamental goal of cost estimation is to apply the tools and metrics of cost accounting to production that has not yet occurred. Not only does cost estimation incorporate all of the complexities of cost accounting, it also introduces the difficulties of predicting the future. This uncertainty about the future gives rise to two broad approaches to cost estimation: top-down methods and bottom-up methods. Top-down methods estimate costs based on the cost of similar, past projects. Bottom-up methods, on the other hand, estimate cost by considering all of the resources (material, labor) which will go into the final product. For example, if a company is trying to estimate the cost to produce a new version of software, a top-down estimate may consider how much it cost the last time an update was released. If instead, they consider how many new features and functions need to be written for the new version, how many lines of code will be required, the approximate number of hours required per line of code, and the hourly rate of the programming teams, costs would be estimated by a bottom-up method. [6]

Generally, if the product is considered in its final state and compared to comparable products, costs are estimated using top-down methods. If, on the other hand, the product is decomposed into its most fundamental elements to which cost metrics are applied then summed to achieve a total cost, the estimate
is bottom-up. Bottom-up approaches require detailed system knowledge that can be difficult to obtain in complex organizations. Top-down approaches are unsuitable for representing unprecedented system changes. Usually bottom-up methods arrive to a cost estimate for the final product by simply summing up the cost of the constituent parts. This has the subtle impact, especially when estimating manufacturing costs, of ignoring the cost interaction between different components. Figure 4 summarizes both cost model types.

Figure 4. A comparison of Top-Down vs Bottom-Up cost estimation methodologies.

2.1.2. Cost Modeling for Commercial Aircraft

The lack of hard manufacturing data during the aircraft design phase has motivated the formulations of models which predict the ultimate manufacturing cost of the airplane product based on physical design parameters. Past research produced parametric cost models based on the extrinsic properties of the aircraft design, for example weight, cruise speed, wingspan, etc. The primary tool for these parametric models is the “Development and Production Costs for Aircraft (DAPCA)” program developed by the RAND corporation and first released in 1967. The models comprising the DAPCA system are based on fits of major aircraft properties to historical cost data for other aircraft programs. As technology advances, there have been updates to the DAPCA models. For example, DAPCA III published in 1976, includes a model included below for predicting the total cost of flight testing in USD, $FT$, for a new aircraft based on its
airframe weight in lbs $W$, maximum speed at best altitude in knots $S$, number of flight test aircraft required $Q_{FT}$: [7]

$$FT = 52.08 \cdot W^{0.7095} \cdot S^{0.5856} \cdot Q_{ft}^{0.7160} \cdot 10^{-6}$$

Although the DAPCA models are a useful set of tools to use for initial estimates, they do not reveal information about more specific cost-drivers in the design of the aircraft. For example, two designs of the same weight may have vastly different manufacturing costs. In fact, with advanced manufacturing techniques, weight reducing designs can often lead to higher manufacturing costs. Finally, the reliance of the DAPCA models on past data limit their applicability to new or revolutionary designs. For both Boeing and Airbus, the move to an all-composite passenger aircraft was a revolutionary change which strained the DAPCA models.

The Advanced Composites Cost Estimating Manual (ACCEM) is a bottom-up approach that uses a vast catalog of labor hours for each manufacturing step. [8] The ACCEM models were derived from timed studies in the factory. Example models for labor time estimation are given in Figure 5 and include a fixed estimate of the process setup time and an estimate of the runtime which scales with the indicated input parameter, for example layup area $A$, or interface length $L$. Many of the equations are nonlinear with respect to the cost driver variable. A labor time estimation based on ply area for “manual layup of woven fabric” (a common process in composite construction) is given in Figure 6. Processes for producing composite airframes are covered in Section 2.3.
DETAIL ELEMENTS

CLEAN LAYUP TOOL SURFACE

APPLY RELEASE AGENT TO LAYUP TOOL SURFACE

POSITION TEMPLATE (MYLAR) ON TABLE AND TAPE DOWN

PLY DEPOSITION

MANUAL - 3" TAPE
- 12" TAPE
- WOVEN MATERIAL

HAND-ASSIST - 3" TAPE
- 12" TAPE

CONRAC AUTO. (720 IPM)
- (360 IPM)

TRANSFER PLY FROM TEMPLATE TO STACK OR LAYUP TOOL

TRANSFER STACK TO LAYUP TOOL

CLEAN CURING TOOL SURFACE

APPLY RELEASE AGENT TO CURING TOOL SURFACE

TRANSFER LAYUP TO CURING TOOL

<table>
<thead>
<tr>
<th>DETAIL ELEMENTS</th>
<th>SETUP</th>
<th>RUNTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEAN LAYUP TOOL SURFACE</td>
<td>0.000006A</td>
<td>(L1)</td>
</tr>
<tr>
<td>APPLY RELEASE AGENT TO LAYUP TOOL SURFACE</td>
<td>0.000009A</td>
<td>(L2)</td>
</tr>
<tr>
<td>POSITION TEMPLATE (MYLAR) ON TABLE AND TAPE DOWN</td>
<td>0.000107A</td>
<td>0.7700G</td>
</tr>
<tr>
<td>PLY DEPOSITION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MANUAL - 3&quot; TAPE</td>
<td>0.05</td>
<td>0.00140L</td>
</tr>
<tr>
<td>- 12&quot; TAPE</td>
<td>0.05</td>
<td>0.001454L</td>
</tr>
<tr>
<td>- WOVEN MATERIAL</td>
<td>0.05</td>
<td>0.000751A</td>
</tr>
<tr>
<td>HAND-ASSIST - 3&quot; TAPE</td>
<td>0.10</td>
<td>0.000368L</td>
</tr>
<tr>
<td>- 12&quot; TAPE</td>
<td>0.10</td>
<td>0.001585L</td>
</tr>
<tr>
<td>CONRAC AUTO. (720 IPM)</td>
<td>0.15</td>
<td>0.00063L</td>
</tr>
<tr>
<td>- (360 IPM)</td>
<td>0.15</td>
<td>0.00058L</td>
</tr>
<tr>
<td>TRANSFER PLY FROM TEMPLATE TO STACK OR LAYUP TOOL</td>
<td>0.000145A</td>
<td>0.6711</td>
</tr>
<tr>
<td>TRANSFER STACK TO LAYUP TOOL</td>
<td>0.000145A</td>
<td>0.6711</td>
</tr>
<tr>
<td>CLEAN CURING TOOL SURFACE</td>
<td>0.000006A</td>
<td>(L13)</td>
</tr>
<tr>
<td>APPLY RELEASE AGENT TO CURING TOOL SURFACE</td>
<td>0.000009A</td>
<td>(L14)</td>
</tr>
<tr>
<td>TRANSFER LAYUP TO CURING TOOL</td>
<td>0.000145A</td>
<td>0.6711</td>
</tr>
</tbody>
</table>

WHERE:

A = Area of ply, or greatest ply area of stack or layup, in square inches
L = Length of ply strip, in inches

Figure 5. ACCEM summary of cost models for composite layup operations--basic deposition. [8]
Figure 6. Runtime hour estimation for layup of woven material. “The activities encompassed by this equation are: unroll woven material on layup table, flatten, scribe pattern, position straight edge, cut pattern, move to flat layup tool, and smooth down.” [8]

Recognizing the limitations of DAPCA and ACCEM to estimate revolutionary composite designs, NASA undertook a massive study to develop “Cost Optimization Software for Transport Aircraft Design Evaluation (COSTADE)” [9], [10], which was partly an improvement to ACCEM to include models of the individual manufacturing processes. In a large volume of reports produced by private and academic partners, the project produced a new approach for modeling the cost of aircraft designs to address some of the shortcomings of previous modeling approaches. The COSTADE approach builds a bottom-up cost model by calculating the process time and cost by deconstructing the aircraft design into all of its constitutive manufacturing processes. For example, the cost model for manufacturing a single composite wing panel would include models for model preparation, carbon fiber deposition, curing, demolding, trimming, etc.

Whereas COSTADE, DAPCA, and ACCEM are mostly contained in a series of reports and must be implemented in software before they can be used for design, there are some off-the-shelf software packages which provide cost estimates which employ many of these methods. SEER-MFG from Galorath, formerly known as SEER-DFM, is a cost estimation and analysis software, which includes composites tools (http://galorath.com/SEER-Manufacturing-for-Composites-Estimation). It is available for commercial purchase and built upon both public models, including COSTADE, and proprietary models. According to
industry experts, SEER-MFG for Composites uses many of the COSTADE models as the basis for the cost model. Also, according to SEER-MFG users (information acquired in direct conversation), SEER-MFG lacks the ability to take production flow into account, and it requires significant computing time to execute cost estimation in practice.

Of course, many companies have developed internal tools for cost estimation for specific products and industries. Though these tools are not publicly available, they are often derived from existing top-down or bottom-up methods.

Curran et al. from the University of Belfast present what they call the “Genetic Causal Model.” [11] However, Hueber et al. point out of Curran et al.’s work: “Unfortunately, it is impossible for the reader to discern how the model is constructed, how it works, and how the indicated results are obtained, since all the publications only described the model with a few general statements and a simple process sketch.” [2]

Reviews of cost modeling methods for commercial aircraft broadly classify the different methods by top-down or bottom-up. Some authors add further classifications of the top-down methods. [2], [12]

Curran presents a taxonomy for classifying different methods for cost estimation tools into three categories: analogous, parametric, and bottom-up. Hueber et al. follow Curran in classifying cost modeling approaches into bottom-up, parametric and analogous. Weustink [13] by contrast, makes the point that parametric and analogous just represent the same approach to different fidelity levels: parametric methods are a mathematical formalization of analogous methods.

Figure 7 contrasts the different cost modeling approaches as categorized by Hueber et al., based on Curran et al., into analogous, parametric and bottom-up. It can be seen in the figure that analogous and parametric approaches both build on existing product cost information. The analogous methods are some of the easiest to implement and naturally what many designers default to. In a sense, the parametric method is just a mathematical formalization of the analogous method: it simply interpolates between existing products based on important features of the design to arrive at an estimate.
Analogous Cost Estimation:
deriving the product's costs from similarities between known variants and the product.

Parametric Cost Estimation:
deriving the product's costs through mathematic CER (Cost Estimation Relationship) based on historic cases.

Bottom-up Cost Estimation:
deriving the product's costs through summing up all manufacturing steps and their costs.

Figure 7. Schematic description of top-down (left) and bottom-up cost estimation approaches. Reproduced from [2]

Whereas the top-down approach of the DAPCA system is easy to apply at the beginning of the design process, when basic extrinsic data about the aircraft design are available, the bottoms-up approach of COSTADE requires specific design details about each individual part in order to estimate the total construction cost. The advantage of the bottom-up approach is that specific cost drivers can be identified and tied back to specific components. Together, the approaches established by DAPCA IV and COSTADE and the models they provide have been implemented within a number of software tools for manufacturing cost forecasting.

Figure 8 visualizes different cost modeling approaches relative to the classifications.
Figure 8. Classification of estimation models [2]. The division into parametric and analogous was suggested by Curran [11], but rejected by Weustink [13], who summarizes parametric and analogous models as variant-based (and bottom-up as generative.)

Figure 9 weighs the relative merits of the different approaches according to Curran et al. and Hueber et al., which this thesis will expand upon.
Figure 9. Summary of advantages and disadvantages of the three main cost estimation methods, summarized by Hueber [2] largely from [11]

Table 2. Summary of available cost estimation tools

<table>
<thead>
<tr>
<th>Name</th>
<th>Classification</th>
<th>Release Initial / Latest</th>
<th>Availability</th>
<th>Scope</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAPCA [7, 14]</td>
<td>Top-down</td>
<td>1967/ Public</td>
<td>Lifecycle</td>
<td>Regression fits to data from previous aircraft programs.</td>
<td></td>
</tr>
<tr>
<td>SEER-MFG [15]</td>
<td>Top-down and bottom-up</td>
<td>Commercial License</td>
<td>Manufacturing</td>
<td>An easy, off the shelf tool but the models are not available so solutions can be opaque</td>
<td></td>
</tr>
</tbody>
</table>
2.2. Conceptual Aircraft Design: Different Approaches

The challenge of aircraft design is as complex as the machines it produces; modern aircraft like the Boeing 787 have millions of parts, each conceived to perform a certain role and tested, first in a computer then in real life, to assure that it can perfectly perform its role. Before reaching detailed design of these parts, the aircraft goes through conceptual design where different design concepts are proposed and evaluated to select the designs which will be fleshed-out in the detailed design phase. Raymer provides specific guidance to the designer about the order in which the designer should iterate through solving design equations in order to find an aircraft design which feasibly (but not optimally) meets the design criteria.

During the conceptual design phase, designers often will use optimization tools to find the best designs and characterize trade-offs between design decisions. Multidisciplinary design optimization (MDO) is a method to integrate models and simulations of candidate designs and is often used as an optimization tool during conceptual design.

Figure 10 shows the interface of AVID ACS software used to model the physics of an aircraft in the conceptual design phase.

\footnote{http://787updates.newairplane.com/787-Suppliers/World-Class-Supplier-Quality}
Modern CAD tools from Siemens [17] and Dassault Systèmes [18] include built-in workflows or external add-ons that allow the designer to go from part design to FEA simulation to evaluate structural or aerodynamic performance, as well as manufacturing process selection and simulation. This can ultimately lead to tools for designing the layout of the factory to build the product. This integration is relatively new and allows for references to the original design to be easily maintained through the entire design process. Examples include Simulayt Composites Modeler for Solidworks or for Abaqus CAE, VISTAGY with ANSYS, Pro ENGINEER, or CATIA, Anaglyph Laminate Tools for for Solidworks or for ANSYS, and CATIA v5/v6 Composites Design (http://www.digitaleng.news/de/options-for-composites-analysis-and-simulation). Users can define plies using boundary curves on an underlying surface. The tools provide the connectivity between the different evaluation steps, but still rely on the designer to provide inputs to

3 source http://www.avidacrospace.com/software/avid-acs/
improve the design. These CAD tools, advanced as they may be, support only design development in one direction from aircraft configuration to manufacturing planning.

Mikrosam, a manufacturer of automated fiber placement machines used to produce composite aircraft structures, also released a software package that aircraft designers can use to plan and simulate the process. “MikroPlace provides a sophisticated environment for the composite part designer to be able to create composite structures using automated fiber and tape placement (AFP/ATL) techniques, and at the end, to generate the actual program to be executed on AFP or ATL machines. The part designer can perform most of the necessary steps in developing an advanced composite part right in the MikroPlace environment, by being able to draw or import the part’s tool, create ply laminates from numerous strategies, export the composite for analysis to popular Finite Element Analysis (FEA) software packages, analyze and fix production issues, and do custom post-processing modifications of the final layup.” [19] A screenshot is shown in Figure 12.

Figure 11. Interactive Ply Table in Grid Design in CATIA Composites from Dassault Systèmes [18]
2.3. Composite Aerospace Production Systems

Modern airframes, be they composite or traditional metallic construction, are assembled from large modules which are in-turn assembled from smaller submodules. Figure 13 shows how a Boeing 787 is assembled from myriad different sub-assemblies made by Boeing and Boeing’s partners around the world.
Figure 13. The Boeing 787 is assembled from many different large sub-assemblies manufactured by Boeing and its partners from around the world.4

Whereas Boeing's 787 fuselage sections are produced as single-piece "barrels", Airbus' A350 sections are produced as separate pieces that are later assembled into a barrel, as shown in Figure 14.

4 Source: the Boeing Company
Figure 14. A fuselage sub-assembly showing its assembly from major composite components and the fabrication processes used to produce each of the components.\(^5\)

Composite aircraft are built from carbon fiber plies, impregnated with resin. The layers of carbon fiber plies and resin are compressed and cured to form the ultimate part.

The raw carbon fiber material comes in generally two forms: broadgoods (sheets or wide rolls of carbon fiber) and spools of tape or tow. Broadgoods and tapes can be comprised of fibers in a single direction (called unidirectional or UD) where the fibers can be oriented across the axis of the tape or along the axis of the tape. Broadgoods and tapes are also made from woven fibers as well. The material width distinguishes tapes from broadgoods; tapes are generally narrower than 450mm.\(^6\) Tows are spools of unidirectional fibers along the axis of the tow and are not woven. Usually tows are narrower and contain fewer fibers than tape.

Large broadgoods can be cut down into specific shapes and placed (by hand or machine) to molds. Broadgoods can be woven in a number of patterns to obtain different strength properties. Tape products have high strength in the fiber direction. They are stacked up in different orientations (“stacking sequences”) to increase overall stability. The layup is a serial process.

Tapes and tows are usually placed by machine onto a mold or tool. These automated pieces of equipment are often known as automated tape laying (ATL) or automated fiber placement (AFP) machines. Tapes are often better suited for large, nearly flat parts (like wing skins) whereas tows are better to layup


on more contoured surfaces (like fuselages or wing spars). In many cases, the ATL or AFP machine can simultaneously deposit multiple tapes or tows onto the composite tool at a single time.

If the raw materials come pre-impregnated with resin, the production process is known as prepreg. The raw materials must be stored in freezers prior to layup in order to prevent premature curing. After layup, parts are prepared for curing. Most of the resins in commercial production are cured by heating. Some parts are cured in an autoclave which, in addition to heating the part, increases the ambient pressure around the part (like a giant pressure cooker) to compact the plies together and create a stronger part. Once the structure is cured, holes can be drilled (thousands of holes per aircraft), floors and interior can be mounted, etc.

Of course, errors or defects can occur in any of these processes along the way. Correcting such defects has comparatively different effects on overall system performance between metallic and composite construction techniques. With metallic construction, the defect can usually be isolated and work can continue on other components of the assembly. With composite materials where fabrication is highly serial and additive, errors cannot be easily isolated and correcting a defect often interrupts all other work on the part.

In either case, taking time to correct defects add variance to processing times. Timing variance has a more negative impact on the performance of composite production systems than the same variance would have in a metallic production system. The more serial nature of composite fabrication makes it difficult to continue work while addressing a defect. Variation plays a more significant role in composite fabrication than it does when building airframes out of aluminum.

By way of an example, the primary processes for fabricating a fuselage are shown in Figure 15. The fabrication process begins with lamination where the preimpregnated, carbon fiber tows are laid up on the tooling mandrel that forms the inside of the fuselage (or the inner mold line). An automated fiber placement machine (AFP) is used to lay up the composite tows. This machine makes multiple passes for each layer and all of the layers together form a stack. After all of the layers of carbon fiber have been applied, the stack and the tool are prepped and transferred to the autoclave. The autoclave executes a pre-programmed curing recipe that defines required temperature and pressure in time. An example recipe is shown in Figure 16. The high internal pressure in the autoclave compacts all of the layers and the elevated temperature cures the epoxy resin impregnated in the carbon fibers. After curing, the hardened part is removed from the autoclave and placed on a machine for post-processing to make any cutouts and trim the part into its final shape.
Lamination

Carbon fiber prepreg applied to tooling.

Cure

Layers are compacted resin activated in autoclave

Post-Processing

Part is trimmed to final shape. Cutouts are removed.

Figure 15. Depiction of the three primary processes in fabricating a composite fuselage.

Figure 16. A process diagram for a typical autoclave curing cycle for a small composite part. The chart shows the nature of the process beginning with a temperature ramp up to the dwell temperature and pulling a vacuum on the part. The temperature is again ramped up from the dwell temperature to the curing temperature as the atmospheric pressure is increased in the autoclave. As the cure completes, the temperature is returned to ambient, the vacuum on the part is relieved and the atmospheric overpressure in the chamber is released.
2.4. Toward Manufacturing Cost Estimation in Conceptual Aircraft Design

The previous sections have highlighted different approaches to (a) predict the production cost or life cycle cost of an aircraft and (b) facilitate conceptual aircraft design. It was also argued that the important design decisions are made early on and the majority of costs are committed early on in the design process. Thus, there exists a need for cost estimation tools in conceptual aircraft design.

Tools exist (NX and CATIA) for evaluating the structural performance and manufacturability of components during design. They rely on the designer to change the inputs in order to improve design. Multidisciplinary optimization (MDO) seeks to close the loop around the design process to automatically alter the design and evaluate its performance in order to improve given metrics (usually weight).

Usually the performance of the design is evaluated against structural simulations (FEA), aerodynamic simulations (CFD), and other assessments of flight characteristics. Often the objective of the MDO is to minimize the weight of the aircraft. Usually the assumption is that minimizing the weight will minimize the manufacturing and operating costs of the aircraft.

Work done by Karen Willcox and her students at MIT [20]–[22] add models for cost or value to the traditional MDO framework to allow more explicit maximization of value or minimization of cost. In these cases, a weight-based cost estimating relationship (CER) is used to estimate manufacturing cost. Thus, from a manufacturing cost standpoint, these are top-down methods.

Other authors employ bottom-up methods for estimating manufacturing costs. Figure xx below shows the approach by Hueber et al. [2] to combine cost estimation with conceptual design in this MDO framework. They call their tool “ALPHA.”
Mavris et al. have created an MDO framework they call “Manufacturing Influenced Design (MInD)" shown in Figure 18. [23]–[25] There has been significant follow-on work to include an interface for design space exploration (MInD SET) and a version with integration to Simio design and simulation tools (MInD SET PRO). In addition to the traditional structural and aerodynamic simulations, MInD includes calls to SEER-MFG to obtain manufacturing cost estimates to help drive to a cost-minimizing design. The MInD tool also includes a dashboard to allow designers to evaluate trade-studies to see how different design changes influence the cost of the aircraft Figure 20. The MInD tool takes, as input, a concept airframe design and can provide comparative manufacturing trade-studies between the candidate designs.

Recent work from Georgia Tech has integrated factory dynamic simulation tools with concept-phase models [23], [26]. The “Manufacturing Influenced Design (MInD)” environment takes aircraft geometries as input, evaluates structural and aerodynamic performance of the components, predicts process times using off-the-shelf software (SEER-MFG by Galorath), and discrete event simulations of factory dynamics and supply chain to ultimately predict the production cost of the input design. However, since the MInD approach implicitly relies on Monte Carlo methods (see Chapter 4.1 of this thesis), determining the sensitivities between variables is difficult. Furthermore, the approach thus far implemented, requires a design of an aircraft and a supply chain as inputs. Therefore, it is difficult to quickly investigate many different concepts for the production system or to rapidly explore various design trade-offs, known as the “tradespace.”

As Figure 18 and Figure 19 show, MInD and MInD PRO essentially help the designer draw a “black box” around other software packages, such as SEER-MFG, which in itself acts like a “black box”.[27] The
Great advantage is that these are tools many engineers are familiar with. Obvious limitations are: very slow execution speed due to the great number of numerical operations that have to be performed, limited flexibility for use with breakthrough innovations in materials processing not captured by the proprietary software, and limited scientific merit due to the covet proprietary nature of the utilized software.

Figure 18. The Manufacturing Influenced Design (MlnD) tool from Dimitri Mavris et al. from Georgia Tech. [23]
Figure 19. The software components of the MInD multidisciplinary model. Dr. Dimitri Mavris’ group has been working on at Georgia Tech.

Figure 20. Dashboard interface to the Georgia Tech MIND SET PRO tool. 

source: https://www.asdl.gatech.edu/Advanced_Concepts.html
3. GEOMETRIC PROGRAMMING: FRAMEWORK

3.1. Background

This section offers an introduction to the Geometric Programming (GP) form of optimization. The primary source for the background information in this section comes from Boyd et al. “Tutorial on Geometric Programming” [28] and is partially summarized in this thesis for completeness. Geometric programming can solve non-linear optimization problems extremely quickly, with the caveat that the objective function and constraints can be represented in a certain mathematical form.

Geometric programming was originally introduced in the 1960s. It has only been with the recent introduction of efficient solvers that the full power of the GP form is beginning to find use. Geometric programming has helped to solve design problems in several engineering disciplines including aerospace [29], water utility control [30], and communications and electronics [31], among many others.

Recent work from MIT and Berkley has produced new methods for modeling and optimization during the conceptual design phase using Geometric Programming (GP) [20]. Geometric Programs are particularly well-suited for aircraft design optimization because of their ability to handle nonlinear constraints.

3.1.1. GP Formulation

A geometric program is a convex, non-linear, constrained optimization program of the form

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq 1, \quad i = 1, \ldots, p \\
& \quad g_i(x) = 1, \quad i = 1, \ldots, m \\
& \quad x_i \geq 0, \quad \forall x_i \in \bar{x}
\end{align*}
\]

where \( \bar{x} \) is the vector of decision variables

\[
\bar{x} = (x_1, x_2, \ldots, x_n), \quad x_1, x_2, \ldots, x_n > 0
\]
where \( g(\bar{x}) \) is a monomial\(^8\) of the form

\[
g(\bar{x}) = cx_1^{a_1}x_2^{a_2} \cdots x_n^{a_n}, \quad c > 0
\]  

(3)

and where \( f(\bar{x}) \) is a posynomial, a positive sum of monomials, and has the form

\[
f(\bar{x}) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}}, \quad c_k > 0 \ \forall k, \quad a_{nk} \in \mathbb{R} \ \forall n, k
\]  

(4)

An example of a monomial as in (3) could be

\[
\frac{\pi x_1 x_2^3}{\sqrt{x_3}}
\]

and an example of a posynomial as in (4):

\[
2x_1 x_2^3 + x_2^{-0.56}
\]

Boyd also notes that a posynomial that is less than or equal to a monomial

\[
\sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}} \leq cx_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}
\]  

(5)

is a valid constraint for a geometric program. By dividing the posynomial on the left hand side by the monomial on the right hand side, one obtains a posynomial inequality in the canonical form shown in (1). This thesis frequently presents posynomial inequalities in the form of (5).

Developing a “GP model” means finding exact or approximate monomial or posynomial constraints to capture the behavior of the underlying problem. Boyd et al., Hoburg, and others show that many, fundamental physical laws, like \( F = ma \), or well-known approximations can be directly relaxed into the form the GP constraints require. Despite the nonlinearities, constraints of this form are very fast to solve

\(^8\) Note this is different from the normal, algebraic definition of a monomial. This work will refer to this GP definition of a monomial.
with convex optimization methods. This allows for rapid exploration of the aircraft trade-space and gives a wealth of sensitivity data in addition to an optimal solution.

For the models in the context of this thesis, there are three types of variables which are discussed: free and fixed decision variables and parameters. Parameters are values which the user may want to vary but are not part of the geometric program itself. Parameters appear as constants in the GP constraints.

There are many decision variables throughout the GP. Decision variables for which a value is fixed prior to optimization are termed “fixed variables” whereas decision variables that are not fixed and are free to be optimized are appropriately termed “free variable.” Not all combinations of free and fixed variables will produce a feasible GP. If there are free variables which are not bounded, the GP is said to be “dual infeasible” and cannot be solved. On the other hand, a GP is “primal infeasible” if there is no solution which can simultaneously satisfy all constraints.

This thesis distinguishes “parameters” from “fixed variables.” Though both are inputs to a particular problem, “fixed variables” could be turned into “free variables” and, provided the GP is still feasible, optimized. Variables, both fixed and free, form valid GP constraints. Parameters, on the other hand, must be provided as inputs in order for the problem to be a valid GP. Consider the constraint

\[ 1 \geq 3x_1^\alpha + x_2^\beta \]

The constraint is valid as a GP with \( x_1, x_2 \) as free or fixed variables. However, \( \alpha, \beta \) must be “parameters” because they cannot be decision variables in a GP.

Selecting which variables should be free and which should be fixed to properly bound all of the free variables is an important step to contextualize the models to answer a specific question. For example, using the models to predict how much a new airplane with certain performance will cost requires a different context than what the best performing aircraft for a particular budget constraint could be.

Inputs to the model are values for all the parameters and any variables to be fixed. Outputs from the model are the value of the objective function at optimality (often a cost in dollars), the optimal values for the free variables and sensitivities to fixed variables.

### 3.1.2. Solving GPs

As mentioned, a geometric program is a convex optimization problem. The monomial equalities and posynomial inequalities forming a GP are in fact convex in log space. To solve the GP, the problem is transformed by substituting \( x_i = e^{y_i} \). Following this substitution, the natural logarithms of the objective
and each side of each constraint are taken. Since $x_i > 0$ as defined in (2), this logarithm is always defined.

Note that the right-hand-side of the constraints, both monomial equalities and posynomial inequalities, are replaced with $\log(1) = 0$. After transformation, the optimization takes on the form

\[
\begin{align*}
\text{minimize} & \quad \log f_0(e^\bar{y}) \\
\text{subject to} & \quad \log f_i(e^\bar{y}) \leq 0, \quad i = 1, \ldots, p \\
& \quad \log g_i(e^\bar{y}) = 0, \quad i = 1, \ldots, m
\end{align*}
\]

where $\bar{y}$ is a vector $(y_1, y_2, \ldots, y_n)$ and thus $e^\bar{y}$ is a vector formed by component-wise exponentiation

\[e^\bar{y} = (e^{y_1}, e^{y_2}, \ldots, e^{y_n})\]

The trick here, as Boyd et al. point out, is that the original, untransformed GP is non-convex but that its log-transformed problem is convex. This assumption of the form of geometric programs guarantees that they are "log-convex" (by assumption) and are thus transformed to be solved as convex optimization problems. This assumption holds because a monomial with positive terms is log-convex. A posynomial, which is a positive sum of monomials, is also therefore log-convex. The only other assumption of the form of the model is that the variables and coefficients are positive.

To demonstrate this transformation, consider a simple monomial function:

\[3x^{0.5}\]  

(8)

Substituting $x = e^y$ and taking the log-transformed version of (8) is

\[
\log 3(e^y)^{0.5} = \log 3 + \log(e^{0.5y}) = \log 3 + 0.5y
\]

(9)

Figure 21 shows plots of the untransformed function (a) and the log-transformed function (b). The original function (8) is concave (and thus non-convex) but the transformed function (9) is linear and affine (which is convex.)
Figure 21. Plot of an untransformed, concave monomial function (a) and the same function after log-transformation (b) demonstrating the log-convexity of the form.

The transformed geometric programs are efficiently solved by interior point solvers. Two examples are:

- **MOSEK** [https://www.mosek.com/] is a commercial program that can solve linear, convex quadratic, conic, and mixed integer problems, including geometric programs. It is used in finance, forestry, engineering and supply chain optimization.

- **CVXOPT** [http://cvxopt.org/] is a free software package for convex optimization.

**GPkit**, an open-source python library was created to help designers implement GP models in a more natural way and assist in parsing the problems into the format required by the convex solvers. [32]

### 3.1.3. Feasibility, trade-off, and sensitivity analysis with GPs

Beyond finding the optimal point by solving the GP, Boyd et al. brings up three types of analyses easily performed with geometric programs and that are useful to the design process: feasibility analysis, trade-off analysis, and sensitivity analysis.

Feasibility analyses check that problem is solvable and that there is at least one solution for $\tilde{x}$ for which all of the constraints are satisfied. Feasibility analyses inform the designer if the problem and concept, as
modeled can actually be solved. If analysis shows a particular problem is infeasible, further feasibility analysis techniques are available to highlight which constraints are causing the GP to be infeasible. This information is used by the modeler to refine the definition of the model or to rule that there is not a design which solves the problem.

One of the features of GP as a convex optimization problem is the mathematic certainty that if solver deems the problem infeasible, there is in fact no solution. In some types of non-linear optimization, there is no guarantee that since the solver didn’t find a solution, one doesn’t exist. If a designer discovers that the problem is infeasible, Boyd et al. present a method by which a second GP is formed that relaxes the constraints to find a feasible solution and help inform the designer of how the original GP was over constrained.

Boyd et al. also demonstrate the utility of GPs for performing trade-off analysis between the objective of the optimization and one or more of the constraints. Multiple geometric programs are solved each one varying the constraints. Hoburg uses this technique to produce Pareto frontiers useful for rapidly exploring different aircraft designs. [29]

Boyd et al. discuss the use of sensitivity analysis to understand how small perturbations of the constraints affect the optimal cost of the objective. The sensitivity data come from simultaneously solving the primal and dual versions of the GP.

The primal geometric program to minimize a posynomial has a corresponding dual to maximize a product. The duality properties are presented explored by Duffin et al. in their foundational work on geometric programming.[33] They present the first duality theorem of geometric programming which is repeated here:

Given the primal geometric program A

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq 1, \quad i = 1, \ldots, p
\end{align*}
\]  

(10)

**Theorem:** Suppose that the primal program A is superconsistent that the objective function \(f_0(x)\) attains its constrained minimum value at a point \(x'\) that satisfies the primal constraints. Then:

(i) The corresponding dual problem B is consistent and the dual objective function \(v(\bar{u})\) attains its constrained maximum at a point \(\bar{u}'\) which satisfies all the dual constraints.
(ii) The constrained maximum value of the dual function is equal to the constrained minimum value of the primal function.

(iii) If $\bar{x}$ is a minimizing point for primal program $A$, there is a vector of non-negative Lagrange multipliers $\bar{\lambda} = (\lambda_1', \lambda_2', ... , \lambda_p')$, such that the Lagrange function

$$L(\bar{x}, \bar{\lambda}) = f_0(\bar{x}) + \sum_{k=1}^{p} \lambda_k [f_k(\bar{x}) - 1]$$

(11)

has the property

$$L(\bar{x}', \bar{\lambda}') \leq f_0(\bar{x}') = L(\bar{x}', \bar{\lambda}') \leq L(\bar{x}, \bar{\lambda}')$$

(12)

for arbitrary $x_i > 0$ and arbitrary $\lambda_k \geq 0$.

(Note the boldface lambda $\lambda$ is used here to represent a Lagrange multiplier. This is not to be confused with production rate $\lambda$.)

Each dual variable $u_i$ in the vector $\bar{u} = (u_1, u_2, ..., u_p)$ corresponds to a posynomial inequality constraint $f_1, f_2, ..., f_p$ in the primal problem and to a Lagrange multiplier $\lambda_i$. Boyd shows, by the construction of the Lagrangean dual, a Lagrange multiplier $\lambda_i$ encodes the change of the optimal cost for a perturbation of the posynomial constraint $f_i$ corresponding to the dual variable $u_i$ around the optimal point $\bar{u}'$ such that:

$$\frac{\partial \log v(\bar{u})}{\partial \log u_i} \bigg|_{\bar{u} = \bar{u}', u_i = u_i'} = \lambda_i'$$

(13)

Boyd et al. point out that although the information from sensitivity analyses is similar to that obtained from trade-off analysis, the sensitivity analysis can be performed with only a single solve of the GP.

Hoburg showed how to perform sensitivity analyses for every fixed variable. [34] (The sensitivity to free variables at optimality is zero.) He shows that the overall influence on the optimal cost by perturbing variable $x_i$ is found by a sum of partial derivatives:

$$\frac{\partial \log v(\bar{u})}{\partial \log x_i} \bigg|_{\bar{u} = \bar{u}', x_i = x_i'} = \frac{\partial \log f_0(\bar{x})}{\partial \log x_i} + \sum_{j=1}^{p} \frac{\partial \log v(\bar{u})}{\partial \log u_j} \frac{\partial \log u_j}{\partial \log x_i}$$

(14)
Combining (13) and (14) gives

\[
\frac{\partial \log p^*}{\partial \log x^*_i} = \frac{\partial \log f_0}{\partial \log x_i} + \sum_{j=1}^{m} \lambda_j \frac{\partial \log u_j}{\partial \log x_i}
\]  

(15)

Given the log form of the optimization, the dual costs and thus sensitivities are fractional. If the sensitivity of the optimal cost to variable \( x_i \) is reported as \(-5\), this means increasing \( x_i \) by 1%, one would expect the optimal cost to decrease by 5% (and vice-versa). These fractional sensitivities, by definition, are non-dimensional sensitivities which makes comparison across fixed variables of different magnitude very convenient. Most of the modern solvers use interior point methods which calculate the vector of Lagrange multipliers during the solve. This means sensitivity information is available for every constraint and variable with only a single solve of the GP!

### 3.2. Opportunity

Geometric programming has helped solve the complex problems inherent to aircraft conceptual design. However, there has not been a formulation that allows the manufacturing process to be represented as a GP, let alone the inherent complexity of production flow. This thesis contributes this important piece.

Concept-level cost models of composite aircraft manufacturing processes can drive sub-system specifications of the resulting aircraft by calculating manufacturing costs of different design alternatives. Furthermore, early cost modeling can allow for earlier specification and development of key manufacturing technologies which can help to control costs and drive value later in the life of the program. The sensitivity data can be used to identify inputs which can have a large impact on cost. Knowing the cost drivers can help highlight areas where higher-fidelity modeling may be required, to focus improvement efforts, or insert technology.

The overall approach can be made top-down by first modeling the production system with general, though conservative models. Sensitivity results from the initial model can be used to selectively use more resolution in subsequent models where sensitivities are high.

This results of this thesis provide a framework for using GP as a tool for integrating the design of product and production system for composite aircraft.
3.3. Vision

This thesis proposes to solve the production cost estimation problem using a novel formulation as a Geometric Program. The production cost is estimated by solving the factory design problem, or resource allocation problem, respectively, depending on particular boundary conditions. To achieve this, GP-compatible models of production flow, process flow, and production cost will be analytically developed, and synthesized into a Geometric Program that describes the factory, and which can be solved for a certain variable, such as production cost.

The models presented in this thesis primarily focus on the aircraft production costs specifically related to constructing and operating the factories required to produce a particular aircraft design. The models capture how constraints on the production system, such as minimum rate, interact with the cost of the system. A candidate design is principally characterized by its geometry (or shape), often called the Outer Mold Line (OML) and its manufacturing processes. The design is represented by a series of models that approximate the manufacturing processes specified in the design and approximate the behavior of these individual processes combined into a production line. Finally, the production line is optimally sized subject to the high-level constraints. The framework allows the designer to compare different OML and process combinations by considering the costs of different design combinations subjected to different high-level constraints such as production rate.

The models are formulated as optimization problems with top-level design goals implemented as the objective function and constraints. For example, for a given capital budget, maximize the throughput of a factory or find a cost minimizing design which exceeds a certain production rate target. The governing equations of the model are implemented as constraints as well. Solving the GP optimization problem which contains the factory model will give an estimated cost for implementing a production system under the constraints the modeler has included. Perhaps more useful in the conceptual design phase, however, is the sensitivity of the cost to the different constraints and parameters the modeler has specified. These sensitivities can help guide the maturation of the design by highlighting the significant areas for improving the cost or performance of the production system.
4. PRODUCTION LINE FLOW MODEL DEVELOPMENT

This chapter addresses the development of geometric program (GP) formulations of factory flow models to help understand the (often complex) way in which processes in a manufacturing system interact with one another during production. This work derives an approximation for single-product production lines operating under Constant Work in Progress (CONWIP) policies. CONWIP production lines are a type of “pull” production line and are found in many places in industry, not just in composite aircraft construction. Establishing “pull” in a production system is one of the principle tenets of “lean.” The critical piece of the cost modeling approach presented here is to be able to predict the performance of the factory in a way which is compatible with the mathematical structure of the Geometric Programs.

4.1. Prior Art: Manufacturing System Models

Production system modeling methods generally build upon process cost models and simply sum the modeled costs of each component to predict the cost of the complete product. In reality, the supply chain and manufacturing system required to produce a complex product such as an aircraft is significantly more complex than can be represented by summing component process times and real systems include multiple links in the supply chain with process variances, factory floor controls, and other dynamic factors that add additional cost. The trend has been toward more complex supply chains where the relationships between dynamic factors have become increasingly complex and important to model and manage [35], [36].

The randomness and variance present in most real manufacturing systems complicate the modeling efforts. Traditionally there have been two principal approaches to modeling the behavior of a factory. The first approach uses Monte Carlo methods to simulate the behavior of the factory by building Discrete Event Simulations (DES). [37], [38] In the second, analytical models based on queueing theories are constructed to explicitly capture the stochastic dynamics of the system [39]. Regardless of the approach, the question these models try to answer is usually the same: for a given arrangement of capacity and inventory which form a factory, what is the average performance expected from the factory. Commonly, a factory’s performance is measured by the production rate which can be achieved by a given investment or, conversely, the required investment to achieve a goal production rate. A designer may also wish to consider the lead (flow or cycle) time of an individual part particularly if the production system is set up to fulfill specific orders. This production lead time is equivalent to the time it would take a product to traverse the production system from the beginning of production to completion.
4.1.1. Discrete Event Simulations

Simulations are often fast and easy to construct as there are many commercially-available software tools available specifically for building DES models for factories and supply chains [40], [41]. These simulations, like all Monte Carlo methods, rely on random draws from probability distributions so there is no guarantee that the behavior of the simulation will be the same from one run to the next. The usual approach is to run the simulation many times, often thousands of times, for the estimates of the model to converge to the real behavior of the system. Sensitivities to particular parameters are estimated by perturbing the inputs and running simulations again to observer changes in the output.

4.1.2. Stochastic Models based on Queueing Theory

Models based on queueing theory, on the other hand, seek to build analytical relationships between the model variables. One of the more common approaches is to represent the production process with Markov chains [42] and to solve for the steady-state to find the behavior of the factory. If not constructed carefully, the Markov chain may not be computationally tractable and may have too large of a state-space [39]. For larger systems, there are iterative approaches which approximate the original Markov Chain but in a more computationally robust form. [39]

Markov models of queueing systems were not suitable to this application due to the computational complexity when the models were scaled to reasonable size.

4.2. CONWIP Production Line Model

The composite fabrication production process is modeled as a single-product, serial process. The production line, as a consequence, is modeled as a series of production cells. The production line is assumed to be operating under a constant work in progress (CONWIP) policy. Each station in the production line executes a process which has random processing time $T$. Each cell has one or more parallel, servers each executing identical processes. Figure 23 shows the layout of the production line and introduces the important variables. If all of the servers in a particular cell are busy when a piece of WIP should arrive, the arriving work piece will queue in front of the cell and wait for the next server to be available. It is assumed these queues are infinitively large. These production models do not include the effects of process learning. Process learning is discussed in Chapter 6.

An important assumption is that the conceptual production line will be operating under a Constant Work in Progress (CONWIP) policy. This assumption is based on consultation with experts in the industry.
and direct observations the operational practices in composite fabrication of commercial aircraft. Often, the
CONWIP operating policy is not explicitly stated but rather implicitly imparted through the use of “tool
carriers” of tooling which must remain with the piece as it flows through the production line. Figure 22
shows the aft section of a 787 on a tool carrier. There is significant capital cost in purchasing tools or tool
carriers so the practice is to limit the number of each and to ensure that these tooling resources are well
utilized. In such cases, the overall behavior behaves much like a CONWIP policy.

CONWIP is often used as strategy for implementing lean principles since it is a system that
fundamentally limits the amount of inventory in the system [35], [43]–[46]. In the language of lean,
CONWIP is a type of kanban control system where the kanbans are counted and circulated through the
entire production line. In fact, some implementations of CONWIP do not use separate cards (either physical
or digital) but use the tools or tool carriers themselves as the signaling medium for the production control
system.

An important feature of the CONWIP policy is that it prevents the inventory in the system from
increasing without bound. Therefore, the model can, without the risk of numerical instability, have queues
of infinite size in front of each cell. Though the queues are said to be “ininitely large” the amount of
inventory in any single queue will never exceed the amount of inventory the CONWIP policy allows in the
system. The infinite queue size means that a completed part can never be blocked from leaving a cell by a
lack of space in a downstream queue. This is to say, there is no chance of blocking phenomena in the
system.
Figure 23. Production Line model showing serial cells (stations) and the parallel workstations (servers) in each cell.

In general, the main cost drivers in a production system are the amount of inventory in the system (work in progress) and the amount of capacity (machines, labor). In the model presented here, the inventory cost driver is the CONWIP count $L$ and the capacity is the number of parallel servers (workstations) at each cell $m_i$. Generally, increasing the amount of WIP in the system will increase the production rate. However, there is cost to carrying WIP which arises from the infrastructure costs for carrying the WIP and the opportunity costs based on the value of the WIP pieces themselves. In many cases, the production system must be designed to meet a particular production rate that is established system-wide. For example, the fuselage fabrication facility must match the rate of the wing manufacturing and of the assembly production lines. Adding capacity to a factory will also generally increase the throughput of the factory. In the presented models, the factory is composed of cells in series which each perform a unique work package. In an aircraft manufacturing example, a cell work package may comprise the process of attaching the wings and engines to a fuselage.

The specific case modeled in this section makes the following assumptions about the production line.

**Key assumptions about the production line model:**

- The production line produces a single product;
- The production line is a serial arrangement of cells each with parallel servers and a random processing time characterized by two moments;
The process times distributions are assumed to time-invariant (e.g. learning effects are not included);

- A single server will hold only a single piece of WIP at a time;
- Cells process parts on a first-come, first-served basis;
- The production line is operated with a constant work in progress (CONWIP) policy;
- The models will approximate the steady-state behavior of production line;
- The models implement the relaxed (non-integer) form of the production line;
- There is always a backlog of jobs to complete so that the line is never starved of work.

Capacity is added through the addition of servers (workstations) to the cells. The workstations in each cell are parallel positions for executing identical work packages. Each workstation can only handle a single piece of WIP at a time and each individual piece of WIP will only visit one single workstation in each of the cells in the production line.

Within a workstation in each cell $i$, the workpiece (work-in-progress) undergoes processing for a random amount of time $T_i$. Each process $i$ is represented by its two statistical moments: the average processing time $t_i$ and the standard deviation of processing time $\sigma_i$. Both of these measures are frequently collected in industrial practice. Furthermore, the first two moments are often sufficient to adequately capture the behavior of the production system [47]. In the models, the process is also characterized by the squared coefficient of variation $c_{2i}^2$ where:

$$c_{2i}^2 = \frac{\sigma_i^2}{t_i^2}$$  \hspace{1cm} (16)

For the purposes of this model, it is assumed that there is always a backlog of production orders so that the production line will not be starved by a lack of work. In real production systems in aerospace, there is usually a large backlog of customers’ orders.

The GP implementation allows for fractional numbers of workstations. In reality, the number of workstations is a whole number. Thus, the model implements a continuous relaxation for capacity.

The goal of the flow models is to produce estimates of the line production rate and piece flow time through the line. Inventory $L$, flow time $W$, and production rate $\lambda$ are related by Little’s Law. Under the CONWIP policy, $L$ is known. Knowing either the flow time or the production rate, one can find the other using Little’s Law:
For a given factory, the goal of the optimization is to find the best balance of WIP and capacity to achieve the overall design goals. Often the goals pertain to rate. This thesis presents new flow models, which analytically relate rate, WIP, capacity, and other characteristics of the production line.

4.3. Approximations for CONWIP production line performance

A set of GP constraints are obtained to represent the serial CONWIP production line presented previously. The GP models draw on models of queueing networks developed by Whitt [48] as a starting point. From this starting point, additional models are developed to approximate the behavior of a CONWIP production line and production cells with multiple servers in parallel. The important variables that the model considers are the production rate \( \lambda \), CONWIP capacity \( L \), and production capacity expressed through the number of parallel servers in each cell \( m_i \).

The GP formulation of these models is shown in section 4.4.

4.3.1. Whitt’s Queueing Network Analyzer (QNA)

The Queueing Network Analyzer (QNA) was developed by Whitt as a method to aid in the design of telephone networks [48], [49]. The approach applies well-known approximations of flow time through a single cell (including waiting time and processing time) and approximates the interaction between cells by modeling their arrival and departure processes. The method analyzes the queue behavior at each cell individually and then characterizes the arrival and departure processes of each cell. The QNA provides approximations for the steady-state behavior of the queueing network.

Flow time through a cell

The method divides the cell flow time \( W \) into the time the part spends queued waiting to enter the cell \( W_{q,i} \) and the time spent processing the part which is the expected processing time \( t_i \).
\[ W_i = W_{q,i} + t_i \]  \hspace{1cm} (18)

To approximate the queueing time \( W_q \), Whitt employs the Kingman approximation [50], [51] which depends on the squared coefficient of variation (SCV) of the arrival process to the cell \( \sigma_a^2 \), the processing time at the cell \( t \), the SCV of the processing time in the cell \( \sigma_s^2 \) and the utilization of the cell \( \rho \).

\[ W_q = t \left( \frac{\rho}{1 - \rho} \right) \left( \frac{\sigma_a^2 + \sigma_s^2}{2} \right) \]  \hspace{1cm} (19)

Combining (18) and (19) for an individual cell \( i \) gives:

\[ W_i = t_i \left( \frac{\rho_i}{1 - \rho_i} \right) \left( \frac{\sigma_{a,i}^2 + \sigma_{s,i}^2}{2} \right) + t_i \]  \hspace{1cm} (20)

The QNA uses equation (20) to analyze the total flow time (including queueing time and processing time) of a part through a cell with a single workstation (server). Throughout this section, we assume that each cell has a single server. Section 4.3.3 extends the approximations to multi-server cells.

The Kingman approximation is based on a heavy-traffic assumption which means the approximation assumes that the queue is operating with a utilization close to unity. As the actual utilization of the cell moves away from unity, Kingman's approximation will predict longer queueing times than would be found in the real system. [52], [53] It is shown that not only is the Kingman approximation conservative, it is the true upper bound on the waiting time in a G/G/1 queue. Therefore, the approximation is conservative.

**Arrival and departure processes**

To characterize the departure process from a cell, Whitt refines an approximation from Marshall [54] that relates the squared coefficient of variation of the arrival process \( \sigma_a^2 \), the squared coefficient of variation of the processing times \( \sigma_s^2 \), and the squared coefficient of variation of the departure process \( \sigma_d^2 \)

\[ \sigma_d^2 = (1 - \rho)\sigma_a^2 + \rho \sigma_s^2 \]  \hspace{1cm} (21)

\(^9\) "G/G/1" is "Kendall's Notation" for a cell where the inter-arrival times are generally-distributed, the processing times are generally-distributed, there is a single workstation in the cell, and there is no restriction on the number of pieces waiting in the queue in front of the cell.
Marshall originally derived this model from a diffusion approximation of the queuing process. Nonetheless, the intuition for this model is fairly straightforward: if a cell is not highly utilized (which is to say it's not very busy) the departure process tends to look like the arrival process. As the utilization of the cell increases, the departure looks more like a mix of the arrival process and the service process. Finally, as the utilization approaches unity, the departure process mirrors the behavior of the service process.

Whitt designed the model for an open network of queues wherein routing between the cells is random. Since the model of the composite production line has deterministic routing, the departure process for cell \( i \) is the arrival process of the immediately following cell \( i + 1 \) and the departure SCV of one process is the arrival SCV of the immediately subsequent process. This behavior is expressed

\[
c^2_{d,i} = c^2_{a,i+1}
\]  

(22)

The coupling from one process to the next is precisely how the variation propagates.

Whitt's models are designed to approximate the behavior of open queueing systems wherein arrival processes characterize how new parts (or jobs) enter the system. In most use cases covered by Whitt’s QNA, the average arrival rate and squared coefficient of variation of the arrival rate are required inputs to the model. Furthermore, the average process time \( t_i \) and squared coefficient of variation of processing time \( c^2_{s,i} \) for each cell \( i \) are assumed inputs.

Note the appearance of \( \rho, c^2_{a,i}, c^2_{s,i} \) in (20), (21) and (22). These non-linear equations need to be solved simultaneously to find the predicted behavior of the queueing network. Whitt describes an algorithm for iteratively calculating the simultaneous equations to converge on values for \( \rho_t, c^2_{a,i} \) to find the predicted performance of the system including throughput \( \lambda \), and average inventory \( L \).

4.3.2. Models of CONWIP production lines

This section uses Whitt’s QNA as a starting point and develops a model for a CONWIP production line. A CONWIP line is considered a closed network of queues because the WIP count is kept constant (there are no arrivals into or departures from the network of queues which must be considered). There are two primary features which distinguish a CONWIP production line (a closed queueing network) from the open queueing network of Whitt’s QNA: first, the feedback mechanism of the closed network (wherein a new job at the first cell can only begin once a job at the last cell is completed) couples the departure process of the last cell in the line to the arrival process of the first cell in the line; second, the sum of all the inventory in the system is constant.
The coupling of the last cell to the first cell which defines the closed network is accomplished by adding an additional equation which equates the departure process of the final cell in the series to the arrival process of the first cell in the series. If there are \( M \) cells in series in the production line numbered 1 through \( m \), the additional equality added to couple the first and last cells is:

\[
C_{a,1}^2 = C_{d,M}^2
\]  

(23)

For a given cell in the CONWIP line, WIP arrives from only one cell and the WIP departs the cell to only one other cell. Therefore, the rate for every arrival and departure process must be the same and equivalent to the production rate of the line \( \lambda \). Since we assume every cell is served by a single workstation (server), for every cell

\[
\lambda = \frac{\rho_i}{t_i}
\]  

(24)

The CONWIP policy is enforced by equating the total WIP count, by Little’s Law (17), to the production rate \( \lambda \) and the total flow time through all \( M \) cells:

\[
L = \lambda \sum_{i=1}^{M} W_i
\]  

(25)

4.3.3. Approximating multi-server production cells

Whitt’s original models for the Queueing Analyzer and the extensions to closed queueing networks for CONWIP systems presented previously, do not yet consider the behavior of production cells with multiple-parallel servers as the model of the production system requires (recall the importance of having parallel servers as the mechanism by which capacity is altered.)

Any approximation used in formulating the GP models ought to be conservative which is to say the GP models should under-predict production rate and over-predict flow time. If the flow models are conservative in this way, the associated costs should also be conservative in the sense that the capacity and inventory costs predicted during optimization should be higher for the conceptual phase than what more detailed modeling later in the design process should predict.

The model begins with the approximation from Kingman as stated in (19). Another way Kingman’s approximation is stated is that the behavior of a queue with general distributions for arrival intervals and
processing times is a modification of the corresponding Markovian queue by the average squared coefficient of variation of the service process and the arrival process. For the queueing time $W_q$ of a G/G/I with SCV of the arrival process $c^2_a$ and SCV of the service $c^2_s$, the corresponding Markovian queue is an M/M/1 queue\(^{10}\) where the expected queueing time is $W_q(M/M/1)$. Kingman’s approximation states:

$$W_q(G/G/1) \leq \left(\frac{c^2_a + c^2_s}{2}\right) W_q(M/M/1) \tag{26}$$

In fact, this approximation has been shown to be an upper bound on the queueing time for a queueing system with generally-distributed inter-arrival and process times. [47]

Queueing theory gives an exact solution for $W_q(M/M/1)$ [55]

$$W_q(M/M/1) = t \left(\frac{\rho}{1 - \rho}\right) \tag{27}$$

Combining (26) and (27) gives the Kingman approximation as seen before.

To approximate the waiting time for the multi-server case of the general queue $W_q(G/G/m)$, where $m$ is the number of parallel servers, one can use the same form for the approximation with $c^2_a$ and $c^2_s$:

$$W_q(G/G/m) \leq \left(\frac{c^2_a + c^2_s}{2}\right) W_q(M/M/m) \tag{28}$$

The calculation of utilization at each cell $\rho_i$ is altered to reflect the increased capacity of multiple parallel servers:

$$\rho_i = \frac{\lambda}{m_i} \frac{t_i}{m_i} \tag{29}$$

where $\lambda$ is the production rate of the serial line, the number of parallel servers (workstations) at the cell $m_i$, and the average processing time $t_i$. This definition of utilization for the multi-server case is standard in queueing systems analysis.

\(^{10}\) "M/M/1" is the notation for a queue where the inter-arrival times are poisson-distributed (Markovian), processing times are exponentially-distributed (Markovian) and the cell has one workstation.
In this multi-server case, the expected queueing time for the Markovian queue is a function of the probability that all of the servers are busy. This probability is expressed as the probability that the total number of pieces in the cell (both in the queue and in service) \( N \) is greater than the number of servers \( m \), \( P(N \geq m) \). If not all of the servers are busy, an arriving part can immediately find an open server and won’t have to wait in a queue. If all of the servers are occupied, an arriving part is expected to wait in the queue for time \( \frac{t}{m(1-\rho)} \). Therefore:

\[
W_q(M/M/m) = \frac{t}{m(1-\rho)} P(N \geq m)
\]

(30)

For a Markovian queue, the probability \( P(N \geq m) \) is modeled exactly by the Erlang-C formula [56] and is a function of the cell utilization \( \rho \) and the number of parallel servers \( m \):

\[
P(N \geq m) = \frac{(m\rho)^m}{m! (1-\rho) \zeta}
\]

(31)

where

\[
\zeta = \left[ \frac{(m\rho)^m}{m! (1-\rho)} + \sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} \right]^{-1}
\]

(32)

To check this formulation consider the case of a single-server for which the exact solution is known and shown in equation (27). For \( m = 1 \), \( P(N \geq m) = \rho \) from equations (31) and (32). Thus, for \( m = 1 \) (27) and (30) are equal.

Figure 24 shows a plot of the Erlang-C function for \( 0 < \rho < 1 \) and \( 0 \leq m \leq 5 \) which are typical values for the types of composite production lines designed here. If \( m = 1 \), \( P(N \geq m) = \rho \) as has been shown. The plot also shows \( P(N \geq m) < \rho, \ m > 1 \).
Figure 24. The Erlang-C function for $0 < \rho < 1$ and $1 \leq m \leq 5$

This inequality means selecting $\rho$ as an approximation for $P(N \geq m)$ as shown in (33) will be conservative since a higher likelihood of having to wait in the queue will have a higher average waiting time which will result in a prediction of lower production rate than will be achieved by the corresponding real system. An over-estimation of waiting time, by Little’s Law, is an underestimation of the production rate. Furthermore, the approximation is exact when $m = 1$.

$$P(N \geq m) \approx \rho$$

(33)

As $\rho$ is always greater than or equal to the Erlang-C function, combining (33) with (30) gives an approximation for the queueing time that is, in fact, an upper bound on the actual queueing time.

$$W_q(M/M/m) \leq \left( \frac{t}{m} \right) \left( \frac{\rho}{1 - \rho} \right)$$

(34)
This upper bound is noted with the $\leq$ operator to distinguish it from the posynomial inequalities which are constructed as constraints in a GP. The inequality constraints in a GP are contrived to perform the optimization.

Now combining (34) with (28) gives the approximation for the queuing time for a G/G/m queue:

$$W_q(G/G/m) = \left(\frac{t}{m}\right)\left(\frac{\rho}{1-\rho}\right)\left(\frac{c_a^2 + c_x^2}{2}\right)$$  \hspace{1cm} (35)

Since the expected process time is not effected by the queueing time, the approximation for the flow time through a multi-server queue with general arrival and service processes is

$$W = \left(\frac{t}{m}\right)\left(\frac{\rho}{1-\rho}\right)\left(\frac{c_a^2 + c_x^2}{2}\right) + t$$  \hspace{1cm} (36)

The approximation in equation (36) is what is used for the GP formulation.

The model of the serial line only allows for parts to travel between from one cell to the next. Therefore, by conservation, the flow rate of parts from one cell to the next is the same between every pair of cells and is the production rate. The CONWIP policy, as shown in section Error! Reference source not found., forms a closed queuing network so there are not arrivals into, or departures from the queueing network itself.

In the GP, the production rate is bounded by the set of monomial constraints from applying the multi-server utilization equation (29) at every cell. The production rate $\lambda$ is also bounded in the GP by the application of Little’s Law to the total flow time and inventory count as the posynomial inequality from (25).

In reality, the number of parallel workstations $m$ will be an integer value. Note how the Erlang-C formula (31) and (32) require the factorial of $m$ and require a summation where $m$ is used as an index which require $m$ to be integer. The GP formulation based on (36) can use a continuous value for $m$ and so the GP solves a relaxed version of the problem.
4.4. Formulating the production line model as a GP

To summarize to this point, section 4.2 introduced the type of production system to be modeled and section 4.3 introduced the model equations themselves. This section will derive the GP formulation of the models from section 4.3.

For the GP formulation which follows, there is assumed to be downward pressure on flow time. To say there is “downward pressure” on a flow time simply means that in order to minimize the objective function the optimization will minimize flow time. Downward pressure on flow time will make at least one constraint which places a lower bound on flow time tight at optimality. The downward pressure on flow time is balanced by upward pressure applied by other constraints in the optimization. The process of deriving the GP formulation for the production line models is all about placing the balancing pressure on flow time through other constraints.

To summarize all of the equations which will form the GP relaxation:

\[
W_i = \left( \frac{t_i}{m_i} \right) \left( \frac{\rho_i}{1 - \rho_i} \right) \left( \frac{c_{a,i}^2 + c_{s,i}^2}{2} \right) + t_i \quad \forall i \in [1, M] \quad (36)
\]

\[
c_{a,i}^2 = (1 - \rho_i)c_{a,i}^2 + \rho c_{s,i}^2 \quad \forall i \in [1, M] \quad (21)
\]

\[
c_{a,i}^2 = c_{a,i+1}^2 \quad \forall i \in [1, M - 1] \quad (22)
\]

\[
c_{a,M}^2 = c_{a,1}^2 \quad \forall i \in [1, M] \quad (23)
\]

\[
\lambda = \frac{\rho_i m_i}{t_i} \quad \forall i \in [1, M] \quad (29)
\]

\[
L = \lambda \sum_{k=1}^{M} W_k \quad (25)
\]

equation (36) approximates flow time through each cell, (21) is the diffusion approximation for the departure process, (22) couples the successive departure and arrival processes for the cells, (23) couples the departure process of the last cell to the arrival process of the first cell, (29) ensures the production rate is consistent through the entire line, and (25) finds the total WIP count for the CONWIP policy.

Recall from Chapter 3 that a geometric program comprises monomial equalities and posynomial inequalities. As written equations (22), (23) and (29) are monomial equalities and are thus, already in a GP-compatible form. To put these monomials into the canonical form, the right hand side is divided through. For example, (29) becomes \( \frac{\lambda t_i}{\rho_i m_i} = 1 \).

Equation (25) is relaxed to the polynomial inequality
During optimization where rate is maximized for a fixed value of $L$ or where a minimum $L$ is found for a target $\lambda$ this constraint will be tight and the GP formulation will be exact.

The approximation for cell flow time (36) is first relaxed to provide the lower bound on $W$

$$W_i \geq \left( \frac{t_i}{m_i} \right) \left( 1 - \rho_i \right) \left( \frac{c_{a,i}^2 + c_{s,i}^2}{2} \right) + t_i$$

However, there is still a problematic $(1 - \rho)$ in the denominator. First $\alpha$ is substituted for the $(1 - \rho)$ term.

$$W_i \geq \left( \frac{t_i}{m_i} \right) \left( \frac{\rho_i}{\alpha_i} \right) \left( \frac{c_{a,i}^2 + c_{s,i}^2}{2} \right) + t_i$$

As the optimizer tries to minimize $W$, it will try and make every $\alpha_i$ as large as possible so an additional constraint is added to both put an upper bound on $\alpha_i$ and relate it to $\rho_i$

$$1 \geq \rho_i + \alpha_i$$

As before, downward pressure on $W_i$ will drive both inequalities (39) and (40) to be tight at optimality and the GP formulation is exact. Furthermore, the $\alpha_i$ substitution can be made into (21) for an exact GP formulation

$$c_{a,i}^2 \geq \alpha_i c_{a,i}^2 + \rho_i c_{s,i}^2$$

It may appear that $c_{a,i}^2$ is not properly bounded from below and would happily go to zero during optimization. However the arrival-departure process coupling provides the necessary constraint. Note the absence of bounds on $t_i$ and $c_{s,i}$. These variables are either specified as input parameters or additional constraints (models) are included as will be shown in Chapter 6.

The beauty of the GP formulation is that there is no forced distinction between which variables must be specified as input parameters and which can be left as decision variables. This is the power of the non-linear formulation and partially why large GP models can be modularly constructed from lower-level
models. It must be noted that the number of cells in the line $M$ must be specified as the problem is defined and cannot be optimized with this particular formulation. Furthermore, this model allows for fractional numbers of workstations. In reality, the number of workstations must be integer when workstations are defined by monolithic tooling, or large pieces of equipment.

The full GP formulation for a CONWIP production line of $M$ cells is shown as Table 3, below. In this model, only the number of cells in the serial line $M$ must be specified as an input parameter. Any of the other variables may be free or fixed as inputs. If variables are free, they must be properly bounded by the GP as explained in Chapter 3.

Table 3. GP formulation of the CONWIP production line model

| $W_i \geq \left( \frac{t_i}{m_i} \right) \left( \frac{\rho_i}{\alpha_i} \right) \left( \frac{c_{a,i}^2 + c_{s,i}^2}{2} \right) + t_i$ | $\forall i \in [1, M]$ |
| $1 \geq \rho_i + \alpha_i$ | $\forall i \in [1, M]$ |
| $c_{d,i}^2 \geq \alpha_ic_{a,i}^2 + \rho_ic_{s,i}^2$ | $\forall i \in [1, M]$ |
| $\lambda = \frac{\rho_i m_i}{t_i}$ | $\forall i \in [1, M]$ |
| $c_{d,i}^2 = c_{a,i+1}^2$ | $\forall i \in [1, M - 1]$ |
| $c_{d,M}^2 = c_{a,1}^2$ |
| $L \geq \lambda \sum_{k=1}^{M} W_k$ |

Table 4. Listing of variables in the production line model.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>Line production rate</td>
</tr>
<tr>
<td>$L$</td>
<td>Constant WIP inventory level</td>
</tr>
<tr>
<td>$M$</td>
<td>Total number of cells in the production line</td>
</tr>
<tr>
<td>$m_i$</td>
<td>Number of parallel servers in cell $i$</td>
</tr>
<tr>
<td>$t_i$</td>
<td>Average process time for service at cell $i$</td>
</tr>
<tr>
<td>$c_{a,i}^2$</td>
<td>Squared coefficient of variation of process times in cell $i$</td>
</tr>
<tr>
<td>$c_{a,i}^2$</td>
<td>Flow time through cell $i$</td>
</tr>
<tr>
<td>$\rho_i$</td>
<td>Utilization of cell $i$</td>
</tr>
<tr>
<td>$c_{a,i}^2$</td>
<td>Squared coefficient of variation of the arrival process to cell $i$</td>
</tr>
</tbody>
</table>
Table 5 shows an example GP in canonical form which implements the CONWIP production line model derived in the previous section. In this example, the objective is to maximize production rate $\lambda$. The GP requires the objective function to be minimized so the objective is to minimize $\lambda^{-1}$ which will maximize the production rate.

Table 5. CONWIP production line model in GP canonical form with the objective to maximize throughput.

\[
\begin{align*}
\min & \quad \frac{1}{\lambda} \\
\text{subject to} & \quad \frac{1}{W_i} \left( \frac{t_i}{m_i} \right) \left( \frac{\rho_i}{\alpha_i} \right) \left( \frac{c_{d,i}^2 + c_{s,i}^2}{2} \right) + \frac{t_i}{W_i} \leq 1 \quad \forall i \in [1, M] \\
& \quad \rho_i + \alpha_i \leq 1 \quad \forall i \in [1, M] \\
& \quad \frac{\alpha_i c_{a,i}^2}{c_{d,i}^2} + \frac{\rho_i c_{s,i}^2}{c_{d,i}^2} \leq 1 \quad \forall i \in [1, M] \\
& \quad \frac{\rho_i m_i}{t_i \lambda} = 1 \quad \forall i \in [1, M] \\
& \quad \frac{c_{d,i}^2}{c_{a,i+1}^2} = 1 \quad \forall i \in [1, M - 1] \\
& \quad \frac{c_{d,M}^2}{c_{a,1}^2} = 1 \\
& \quad \frac{\lambda}{L} \sum_{k=1}^{M} W_k \leq 1
\end{align*}
\]

### 4.5. Evaluating the GP models

A GP constraint formulation of the production line model has been derived. In this section, the predictions of performance for the factory by the GP models is compared to results obtained from discrete event simulations of the corresponding production line. Though the GP model has been developed to be conservative, the degree to which it is conservative or the relative error to the simulations is discussed along
with approximations for the expected error of the GP models. In general, the model error decreases as utilization increases.

4.5.1. Comparing Approximations to Simulations

Simio, a commercially-available discrete event simulation software package, is used to approximate the performance of a real system and benchmark the GP models. The simulation results shown in this chapter were obtained using the simulation parameters in Table 6.

Table 6. Simulation run parameters.

<table>
<thead>
<tr>
<th>Simulation Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run length</td>
<td>2000 Hrs</td>
</tr>
<tr>
<td>Number of runs per point</td>
<td>100</td>
</tr>
<tr>
<td>Warm up period</td>
<td>100 Hrs</td>
</tr>
<tr>
<td>Confidence interval</td>
<td>95%</td>
</tr>
</tbody>
</table>

The first study examines a line comprised of five identical cells, each with a processing time of 5 hours, a process SCV of 0.25 (which corresponds to a CV of 0.5), and each cell with 5 servers in parallel. The line is about the same length as what was seen in the field. The number of servers was chosen to be slightly higher than what was observed in the field as was the SCV to explore a larger potential source of error. For the simulation, processing times are sampled from a Gamma distribution with shape parameter \( k = 4 \) and scale parameter \( \theta = \frac{5}{4} \) [hours]. A Gamma distribution with these parameters has a mean of 5 hours and squared coefficient of variation of 0.25 as desired. With a processing time of 1, note that the production rate of the system is equivalent to the utilization at each of the cells.

Table 7. Production line parameters for the rate and inventory experiment.

| Number of cells in the line \( M \) | 5     |
| Number of servers at each cell \( m_i \) | 5     |
| Process time \( t_i \)                | 5 hours|
| Process SCV \( c_s^2 \)               | 0.25  |

The goal of this first experiment is to compare GP model predictions of production rate against the simulation results as the WIP quantity is varied. Generally, rate is expected to asymptotically approach \( \frac{1}{\lambda} \) unit/hour as the amount of inventory is increased.

To obtain the production rate from the GP models, the constraints from Table 3 are put into an optimization where the objective is to maximize rate (or to minimize \( 1/\lambda \)). The WIP inventory \( L \), process
time \( t_i \), and process squared coefficient of variation \( c_{s_i}^2 \), are specified as input parameters for the optimization. The GP is re-optimized for different values of \( L \) so that the result is a set of \( \lambda, L \) pairs. The simulation is only evaluated for integer values for the WIP count whereas the GP models allows continuous values of WIP count. Therefore in the resulting plot in Figure 25, the results for the simulated line are shown as points whereas the GP model results are a line formed from 100 values for \( L \) between 5 and 40.

![Comparison of Simulation and GP Model Results](image)

Figure 25. Comparison of simulated line to a GP-modeled production line showing production rate as WIP count varies.

The estimate for the production rate obtained from the GP model is always less than what is modeled owing to the conservative nature of the approximations as previously shown in section 4.3. In this example, the GP model under-predicts production rate by about 4% relative to the simulation results.

The second comparison considers the effect that process time variance has on the production rate for the same line of 5 cells each with 5 parallel servers. The WIP count is kept at a constant 20 for this second study. For the GP optimization, the objective remains to maximize production rate (which puts the proper downward pressure on the part flow time.) Each cell is assumed to have the same processing time squared coefficient of variation \( c_{s_i}^2 = c_s^2 \). To investigate the influence of processing time variation, \( c_s^2 \) is varied over the interval \( 0 \leq c_s^2 \leq 1 \). The results of the second comparison are shown below in Figure 26.
Figure 26. A second comparison between simulations and GP models showing the effect of process time variance on production rate.

As the variance of the process time increases, the production rate decreases which also means the utilization decreases as well by equation Error! Reference source not found.. Additionally, the error of the GP model increases as the variance increases. The error ranges from 0% when there is no variance (this is akin to when the process times are deterministic) to about 4% where SCV is 1 (corresponding to process times being exponentially-distributed random variables). The next section will discuss the sources and nature of these errors.

The lines here, while meant to test the models are not necessarily indicative of what is found in real production systems. More realistic production lines are investigated as part of the case studies in Chapter 8.

4.5.2. Error from the approximations

In section 4.3, two conservative approximations are employed to build the model: Kingman’s approximation (19) and the approximation for the Erlang-C function (33). Each approximation, though conservative, has some error from over-estimating the queueing time.

The error of the Kingman approximation has been well studied. [52] [53] [57] A simple approximation for the typical upper bound on the error from the Kingman approximation depends on the utilization $\rho$ and the squared coefficient of variation of the arrival process $c_a^2$. The typical bound is for random processing times sampled from distributions with light to moderate tails and finite variance. Extensive simulation
studies by Whitt showed that a good estimate for the typical upper error bound of the Kingman approximation relative to the expected queueing time is

$$E_{\text{Kingman}} \leq (0.04) \frac{c_2^2}{\rho}$$ (42)

The inverse proportionality of the error to the utilization makes intuitive sense as the Kingman approximation is derived from heavy-traffic assumptions that utilization is near unity. Thus, it follows that error should increase as the utilization decreases.

For an M/M/1 queue with $\rho \rightarrow 1$, the Kingman approximation is exact. If $\rho = 1$ and $c_2^2 = 1$ are substituted into (42) to approximate an M/M/1 queue approaching full utilization, the error model predicts a 4% relative error instead of 0%. Nonetheless, this model generally works well when utilization is not unity based on extensive simulation experiments carried out in literature. The relative error arising from the approximation of the Erlang-C formula by the utilization at a queue is more precisely defined. The error from this approximation can become extremely large, especially as the utilization nears zero as shown in Figure 27.

Waiting time approximation error for multi-server queue

![Figure 27. Error in queueing time of approximating the Erlang-C function by $\rho$ relative to the true queueing time.](image)

64
With even moderate utilization, the error can be quite large. The relative error also gets larger as the number of servers in parallel grows. Consider, however, that this is the relative error for the queueing time only. Lower utilization and more parallel servers usually decrease queue waiting time as a fraction of total flow time. If the Erlang-C approximation error is normalized not by the waiting time but rather by the total flow time, as is shown in Figure 28 as a function of utilization for different numbers of servers in parallel \( m \), a different story emerges.

![Waiting time approximation error as a fraction of total flow time for different \( m \)](image)

Figure 28. Error from Erlang-C approximation relative to the processing time.

For low utilization, the waiting time error may be large but it is extremely small compared to the service time. The relative error rises for moderate utilizations then falls back to nearly negligible levels. Also interesting is how for increasing numbers of parallel servers, the relative error decreases.

A model for the typical upper bound on the over-estimation error of the queueing time relative to the process time is derived by combining the error from the Kingman approximation and the approximation for the Erlang-C function (43). The error is given relative to the processing time. The new, combined error model is a function of utilization \( \rho \), number of parallel servers \( m \), arrival coefficient of variation \( c_a^2 \), and the Erlang-C function \( f_{Erlang}(\rho, m) \) from equations (31) and (32)
\[ E_{\text{combined}} \leq \frac{1}{m} \left[ \rho + 0.04c_a^2 \right] \frac{1}{f_{\text{Erlang}}(\rho, m) - 1} \]  

Figure 29 and Figure 30 show this combined relative error plotted as a function of \( \rho \) for different numbers of parallel servers for a moderate squared coefficient of variation.

Waiting time error relative to total flow time assuming \( c_a^2 = (0.75)^2 \)

Figure 29. Typical upper bound on wait-time over-estimation error relative to processing time.

Around 50\% utilization, the over-estimation of the waiting time by the Kingman approximation dominates. In the combined error at lower utilizations, increasing the number of parallel servers makes the over-prediction worse relative to the processing time. The more zoomed-in version in Figure 30 shows that the expected error is more manageable. Recall that the error shown here is based on the typical upper-bound of the relative error. During this research, GP model error relative to simulations of the equivalent system rarely approached this bound for models representative of real systems.
Waiting time error relative to total flow time assuming $c^2_d = (0.75)^2$

Figure 30. Additional graph of wait-time over-estimation relative to flow time with a focus on $\rho > 70\%$.

The figures show how the error improves as $\rho$ increases. Most times, the modeler doesn’t directly influence the utilization. Generally increasing the WIP inventory or decreasing variability in the system will increase utilization (and thus improve the error of the approximation). At lower utilizations, the error when adding more servers in parallel will increase the error for a given utilization.

The conservative nature of the error means that the real system may benefit beyond what the models predict by going to lower utilization.

Given the sometimes large errors that can occur with low utilization, the modeler should keep the above error graphs in mind in the case where the models are operating in regions of low utilization. These models are not appropriate for modeling low-utilization systems given the heavy-traffic approximations on which the models were built. In practice, very low utilizations in composite fabrication production systems was not observed with average utilizations generally above 75\%.
4.6. Summary

Provided the overall optimization problem is formulated to place downward pressure on flow time, there is a GP formulation of a CONWIP, serial, single product manufacturing line which can model the effect of multiple servers and random processing times (represented with a two-moment approximation). The GP constraints and variables are shown below.

Though the models are guaranteed to yield conservative estimates for the production rate, the modeler should be wary using these models to approximate a production system with low utilization as the error, while still conservative, can quickly grow large.

The modeler should also beware the fact the GP model derived in this chapter allows for fractional values for the WIP inventory and the workstation count. In reality, WIP inventory and counts of parallel workstations must be integer numbers. Therefore, the GP solves a relaxed version of the integer optimization. As the design process progresses to more detail, designers can explore the integer solutions to the models. Here again, sensitivities aid the designer to which integer solutions are the most significant to include.

Table 8. GP formulation of the CONWIP production line model

\[
W_i \geq \left( \frac{t_i}{m_i} \right) \left( \frac{c_{a,i}^2 + c_{s,i}^2}{2} \right) + t_i \quad \forall i \in [1, M] \\
1 \geq \rho_i + \alpha_i \quad \forall i \in [1, M] \\
c_{a,i}^2 \geq \alpha_i c_{a,i}^2 + \rho_i c_{s,i}^2 \quad \forall i \in [1, M] \\
\lambda = \frac{\rho_i m_i}{t_i} \quad \forall i \in [1, M] \\
c_{d,i}^2 = c_{a,i+1}^2 \quad \forall i \in [1, M - 1] \\
c_{d,M}^2 = c_{a,1}^2 \\
L \geq \lambda \sum_{k=1}^{M} W_k
\]
Table 9. Listing of variables in the production line model.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>Line production rate</td>
</tr>
<tr>
<td>$L$</td>
<td>Constant WIP inventory level</td>
</tr>
<tr>
<td>$M$</td>
<td>Total number of cells in the production line</td>
</tr>
<tr>
<td>$m_i$</td>
<td>Number of parallel servers in cell $i$</td>
</tr>
<tr>
<td>$t_i$</td>
<td>Average process time for service at cell $i$</td>
</tr>
<tr>
<td>$c_{s,i}^2$</td>
<td>Squared coefficient of variation of process times in cell $i$</td>
</tr>
<tr>
<td>$W_i$</td>
<td>Flow time through cell $i$</td>
</tr>
<tr>
<td>$\rho_i$</td>
<td>Utilization of cell $i$</td>
</tr>
<tr>
<td>$c_{a,i}^2$</td>
<td>Squared coefficient of variation of the arrival process to cell $i$</td>
</tr>
<tr>
<td>$c_{d,i}^2$</td>
<td>Squared coefficient of variation of the departure process from cell $i$</td>
</tr>
<tr>
<td>$\alpha_i$</td>
<td>Dummy variable for the GP formulation</td>
</tr>
</tbody>
</table>
5. **Cost Model Development**

The ultimate goal of this thesis is to create a concise tool for predicting the production costs of a composite airframe components based on an aircraft's design, the scale at which it is produced, and the different degrees of freedom in the production system. To understand trade-offs among all of these parameters, analytical formulations are derived to describe the various types of costs incurred in production lines, that allow for these different degrees of freedom in factory design, and that take into account the exact effect of the biggest source of error: variation.

5.1. **Scope of this Cost Model: Manufacturing Systems**

Production systems for composite airframe components involve significant costs to initially build-out the production lines and procure equipment, and to operate the production systems to actually build the airframe components. The new Boeing Composite Wing Center cost about $1 Billion USD to construct and equip.\(^{11}\) Once the system is producing parts, costs are incurred for every aircraft built as production continues. Labor and material costs for an aircraft like the 787 or A350 are around $100 Million USD per aircraft.

The models derived in this chapter focus on capturing the recurring and non-recurring costs typical of manufacturing activities. The goal is to understand the financial implications of the capacity (allocating of workstations at different cells) and inventory (setting the CONWIP policy) design decisions characterized by the production line models in Chapter 4.

5.2. **Cost breakdown**

The different types of cost are explained in detail in *Chapter 2.1: Cost Accounting*. A few important concepts are reviewed here for convenience.

*Non-recurring vs. recurring costs*

Examples for non-recurring costs are equipment costs and other capital expenditure, whereas recurring costs are incurred repeatedly, like material costs and consumables.

*Fixed vs. variable costs*

---

Similar to non-recurring and recurring costs, fixed and variable costs differ by when they are incurred. Fixed costs are incurred no matter the production rate, including when no goods are produced. Variable costs are a function of production rate and are only incurred if goods are produced.

*Direct vs. indirect costs*

Direct costs are costs that can be assigned to the production of a single item, like direct labor or materials. Indirect costs are costs that are required for production but serve many purposes, such as overhead.

*Inventory holding costs*

There are costs to the manufacturer for holding parts inventory, work in progress, or finished product yet to be sold.

*Total costs vs unit costs*

Costs can be calculated for an entire program or they can be calculated on a per-unit basis. Considering total program cost is simpler because many of the complex issues of absorbing indirect costs into the unit cost are avoided. Some industries may also report total costs on a yearly basis.

*Capacity costs*

The flow models use capacity as an important lever for the factory sizing problem. For each cell, capacity is added by adding additional workstations in parallel. The addition of that capacity can look different from a cost standpoint. If the cell is automated, adding an additional workstation would show up primarily as a capital expense for the acquisition of the capital equipment. On the other hand, if the workstation is primarily manual labor, the costs will include more operational expenses of the additional labor needed to perform the work in the cell. Furthermore, the flexibility of the labor will determine if it is a variable cost or fixed cost for the workstation.

5.3. GP formulation for total program cost

This section will show the formulation of a model for the total cost of initially building a production line and the recurring costs of operating the production line in perpetuity. If there is downward pressure on cost in the optimization, the following GP formulations are exact. Since the objective is to minimize costs,
this places downward pressure on all the cost equations. Therefore, relaxing the cost equations to
posynomial constraints is straightforward since the optimization will drive the inequalities to equality.

5.3.1. Non-recurring, initial costs

The model for non-recurring costs $C_0$ include the initial capital investment for inventory and equipment
to construct the production line as designed as a function of the inventory capacity $L$, the capital cost of
inventory capacity $H$, for each cell $i$ in the line of $M$ cells, the number of parallel workstations in the cell
$m_i$, the initial capital cost of the workstation $\omega_{0,i}$, and any additional initial costs for the cell (which do not
scale with the number of workstations) $\omega_i$.

$$C_0 \geq LH + \sum_{i=1}^{M} (m_i\omega_{0,i} + \omega_i)$$

5.3.2. Discounted cash flow model for recurring costs

The recurring costs, since they are incurred periodically throughout time, are modeled considering the
time value of money by using discounted cash flow (DCF) analysis. The concept these models capture is
that a dollar is worth more today than it will be worth 10 years from now because a dollar could be invested
today and yield future returns. Similar logic applies to costs: one would rather incur the same monetary
amount as a cost sometime in the future rather than today because money can be invested today and earn
returns in excess of the cost. The “time value of money” is a principle well-known to finance professionals
but may be unknown to the expert in aerostructures.

The DCF estimates the present value $PV$ of a cash flow $C_t$, that occurs $\tau$ periods into the future and is
discounted at per period rate $r$.

$$PV = \frac{C_t}{(1 + r)^\tau}$$

As cash flows occur in different periods, their respective present values $PV$ can be summed over $N$
periods to find the net present value $NPV$.

$$NPV = \sum_{t=0}^{N} \frac{C_t}{(1 + r)^\tau}$$
If the cash flow recurs perpetually, (46) where $N \to \infty$ and $C_t = C$ for every period, the resulting infinite series converges to:

$$NPV = \frac{C}{r}$$ (47)

Three types of recurring costs are considered: inventory holding costs, fixed, and variable costs. Holding cost for a period $t$ is determined by the amount of inventory $L$ and the holding rate inventory holding rate $h_r$. Fixed and variable costs for a period $\tau$ are calculated per cell $i$, in a production line of $M$ cells. These costs are based on the number of workstations in the cell $m_i$, the utilization of the cell $\rho_i$ the fixed, recurring costs (per period per workstation) $\omega_F,i$, and the recurring, variable costs (per period per workstation) $\omega_V,i$. In each workstation, some costs $\omega_V,i$ can vary as a function of utilization. If the workstation variable costs include materials which are consumed only when the workstation is building a part, take paint for example, the cost is only incurred when there is actively work in the station, hence it is multiplied by the utilization $\rho$. For each period $\tau$, the total cost is

$$C = Lh_r + \sum_{i=1}^{M} m_i(\omega_F,i + \rho_i\omega_V,i)$$ (48)

If $C_\tau$ does not vary from period to period, ($C_\tau = C$), combining (46) and (48) and relaxing to an inequality gives

$$NPV \geq \left[Lh_r + \sum_{i=1}^{M} m_i(\omega_F,i + \rho_i\omega_V,i)\right] \sum_{\tau=0}^{N} \frac{1}{(1 + r)^\tau}$$ (49)

If $r$ and $N$ are decision variables, (49) is not a posynomial inequality. If the number of periods the discount rate $r$ are known input parameters, (49) becomes a posynomial inequality because the summation becomes a constant

$$\sum_{\tau=0}^{N} \frac{1}{(1 + r)^\tau} = \frac{1}{r} \left[1 - (1 + r)^{N+1}\right]$$ (50)
If the recurring cost $C_t$ does change from period to period and $r$ and $N$ are input parameters, (46) defines a valid geometric program. If the term $(1 + r)$ is replaced with a single variable, (46) is a GP where only $N$ is required as an input parameter.

If the factory is assumed to be operated in perpetuity, (47) and (48) are combined to give

\[ NPV \geq \frac{1}{r} \left[ Lh_r + \sum_{i=1}^{M} m_i \left( \omega_{F,i} + \rho_i \omega_{V,i} \right) \right] \tag{51} \]

In (51) note that $N$ no longer appears and that the inequality is a posynomial inequality even if $r$ remains a free variable. Though the production which is ultimately built will not operate in perpetuity, using (51) can be useful for a relative comparison among different options without having to specify the length of time for which the factory will operate.

5.3.3. Total cost equation formulation

The total cost is the sum of the recurring and non-recurring costs of production. For the calculation of the recurring cost, the modeler selects either the model for the finite time-horizon discounted cash flow (46) or the model for the production line in perpetual operation (51) to obtain the $NPV$ of the recurring costs. The total cost $C_{total}$ is therefore a posynomial inequality

\[ C_{total} \geq C_0 + NPV \tag{52} \]

5.4. Summary

This chapter presented GP-compatible models which capture the different types of costs associated with manufacturing and the framework by which the costs are combined to find a total program cost. The GP models are adapted from standard cost accounting equations. Table 10 summarizes the models that capture variable and fixed costs as well as recurring and non-recurring costs. Total costs are calculated for the production program operated in perpetuity.

Table 10. Summary of cost accounting constraints.

| $C_{total} \geq C_0 + NPV$ |
\[ C_0 \geq LH + \sum_{i=1}^{M} (m_i \omega_{0,i} + \omega_i) \]

\[ NPV \geq \frac{1}{r} \left[ L h_r + \sum_{i=1}^{M} m_i (\omega_{F,i} + \rho_i \omega_{V,i}) \right] \]
6. **UNIT PROCESS MODEL DEVELOPMENT**

The production line models introduced in Chapter 4 approximate a CONWIP production line in part based on two-moment approximations of the distributions of the processing time at each cell. For a conceptual design, data for all the production processes may not be available. For a GP model to be of use during conceptual design, the production line performance should reflect design changes made to the conceptual aircraft.

The process models relate part geometries to processing times in the cells of the production line models. This link allows a designer to input higher-level information about candidate aircraft designs rather than having to input specific process time data directly into the production line models.

This chapter introduces unit process models, derived from literature and formulated as GP constraints, that can be used to bound process time based on input design geometry. Methods for modeling variability and learning rates when using unit process models are also discussed.

As demonstrated in chapter 2.1.2, a number of cost models exist that provide detailed benchmarks for composite airframe production cost and timing.

If process times are known, for example if they are measured from existing processes, the two-moment representation of the process time distribution can be estimated directly from the first two statistical moments of sampled processing time data. If the cost modeling activity is for conceptual design, process time data will not yet exist for all of the different designs which will be considered. Unit process models are used to predict processing times given input geometry.

The two main sources for approximations for the expected time to complete certain common processes in composite fabrication are the Advanced Composite Cost Estimating Manual (ACCEM) [8] and the Cost Optimization Software for Transport Aircraft Design Estimation (COSTADE). [10] The principles presented in both of these sources are used in industry for estimating process times in composite fabrication; however, often times the parameters in the models are fine-tuned based on proprietary data from within the companies where they are implemented.

6.1. **GP Formulations of ACCEM Process Models**

The ACCEM models are fits to experimental data where the process was performed repeatedly varying geometries and recording the time to establish the “cost estimating relationships” (CERs).
The Advanced Composite Cost Estimating Manual presents a number of equations which estimate process times based on time and motion studies carried out for a variety of different processes in composite fabrication. The models focus primarily on manual processes but does include limited models of automated processes. The ACCEM models are all either monomials or posynomials and can be used directly in a GP model. For example, the ACCEM approximation for the time required \( t \) (in hours) for hand layup of a woven charge with area \( A \) with units of square inches is approximated by

\[
t \approx 0.05 + 0.000751A^{0.6295} \quad (53)
\]

By dividing through by \( t \) \((53)\) can be written as a posynomial inequality for use as a constraint in a GP

\[
\frac{0.05}{t} + \frac{0.000751A^{0.6295}}{t} \leq 1 \quad (54)
\]

Some of the processes modeled in the ACCEM are outdated. Furthermore, changing the existing processes or modeling new processes using the ACCEM framework would require a new set of time and motion studies so the framework is not particularly scalable.

### 6.2. GP Formulations of COSTADE Process Models

As discussed in chapter 2.1.2, the bottom-up cost model framework COSTADE gives models which predict process times based on input geometries. Unlike the ACCEM unit process models, the models in COSTADE are extensible based on first-principles physics models. \[58\], \[59\] COSTADE contains models of both automated and manual processes.

All of the geometry-dependent process time approximations in COSTADE use a variant of one of two equations

\[
t \approx \left[ \left( \frac{\text{Setup Time}}{\text{Run}} \right) + \left( \frac{\text{Operations}}{\text{Run}} \right) \right] \left( \left( \frac{\text{Delay}}{\text{Operation}} \right) + \sqrt{\frac{x_0}{v_0}} + \left( \frac{2r}{v_0} \right) x_0 \right) \quad (55)
\]

and

\[
t \approx \left[ \left( \frac{\text{Setup Time}}{\text{Run}} \right) + \left( \frac{\text{Operation}}{\text{Run}} \right) \right] \left( \left( \frac{\text{Delay}}{\text{Operation}} \right) x_4 + \frac{x_1}{v_{0,1}} + \frac{x_2}{v_{0,2}} + \frac{x_3}{v_{0,3}} \right) \quad (56)
\]
is used mostly to model automated layup processes whereas (55) or a variant thereof is used to model all other processes. In these equations $(\frac{\text{Setup Time}}{\text{Run}})$, $(\frac{\text{Operations}}{\text{Run}})$, $(\frac{\text{Delay}}{\text{Operation}})$, $v_{0,i}$, $\tau$ are parameters of the COSTADE models and $x_j$ are the geometry inputs.

Equation (56) is already a posynomial and becomes a posynomial inequality for a GP constraint:

$$t \geq \left[ (\frac{\text{Setup Time}}{\text{Run}}) + (\frac{\text{Operations}}{\text{Run}}) \right] \left[ (\frac{\text{Delay}}{\text{Operation}}) x_4 + \frac{x_1}{v_{0,1}} + \frac{x_2}{v_{0,2}} + \frac{x_3}{v_{0,3}} \right]$$

(57)

Equation (55), however, can’t as easily be turned into a posynomial inequality due to the addition term underneath the square root. Boyd et al. [28] refer to this form as a “generalized posynomial.” Specifically they note, “A function is a generalized posynomial if it can be formed from posynomials using the operations of addition, multiplication, positive (fractional) power, and maximum.” Since the problematic term

$$\left[ \left( \frac{x_0}{v_0} \right)^2 + \left( \frac{2\tau}{v_0} \right) x_0 \right]^{\frac{1}{2}}$$

(58)

is itself a posynomial raised to a fractional power, (55) is a generalized posynomial.

Boyd et al. show that generalized posynomials satisfy the same convexity property that posynomials satisfy and therefore can be represented exactly as geometric programs. The exact GP formulation splits the generalized posynomial into several posynomial constraints. (58) becomes

$$w_1 \geq \left( \frac{x_0}{v_0} \right)^2 + \left( \frac{2\tau}{v_0} \right) x_0$$

(59)

$$t \geq \left[ \left( \frac{\text{Setup Time}}{\text{Run}} \right) + \left( \frac{\text{Operations}}{\text{Run}} \right) \left\{ (\frac{\text{Delay}}{\text{Operation}}) + \sqrt{w_1} \right\} \right]$$

(60)

using the helper variable $w_1$.

The COSTADE process models are archived in [60]. Figure 31 shows how the processes are archived in the COSTADE documentation. For each process, there is a short, narrative description (Explanation) of what behavior the process model captures. To form the model, the appropriate equation is identified using the EquationID and the listed coefficients for the process are substituted into the corresponding equation.
The COSTADE models assume lengths are input in inches and produce time estimates in minutes. Equation 2 in the COSTADE framework is formulated as a GP using (59) and (60).

### Figure 31. Example process listing from COSTADE report CR-4739. [60]

The coefficients in Figure 31 correspond to the parameters in (59) and (60) $K_1 = \left(\frac{\text{Setup Time}}{\text{Run}}\right) = 1$, $K_2 = \left(\frac{\text{Delay}}{\text{Operation}}\right) = 0$, $K_3 = \nu_0 = 40.04$, $K_4 = \tau = 2.25$. The preform length $PCV_1 = x_0$ measured in inches and $\left(\frac{\text{Operations}}{\text{Run}}\right) = PCV_4$. Therefore the posynomial inequality constraints are

$$w_1 \geq \left(\frac{x_0}{40.04}\right)^2 + \left(\frac{2(2.25)}{40.04}\right)x_0$$  \hspace{1cm} (61)

$$t \geq \left[1.00 + \left(\frac{\text{Operations}}{\text{Run}}\right)\sqrt{w_1}\right]$$  \hspace{1cm} (62)

### 6.3. Additional Model for Resin Infusion Unit Process

Outside of the models in the ACCEM or COSTADE, an additional model was derived to approximate resin infusion time for use in the case studies. The model is based on publicly-available information published by NASA [61]. The infusion time is found to be a function of the resin viscosity $\mu$, the porosity of the dry layup stack $\phi$, the differential pressure driving the infusion $\Delta P$, the permeability of the charges
\( \kappa \), the relevant length of the infusion path \( l_{rel} \), and the thickness of the stack \( h_{stack} \). The dominant physics for the infusion process are captured by Darcy’s law:

\[
\tau_{\text{infusion}} \geq \frac{\mu \phi (l_{rel})(h_{stack})}{\kappa \Delta P}
\]  

(63)

Table 11 shows typical parameters for the infusion process drawn from literature.

<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE [UNITS]</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi )</td>
<td>40%</td>
<td>Porosity of bulk charges ( 1 - F_{fill} )</td>
</tr>
<tr>
<td>( \mu )</td>
<td>350 [mPa \cdot s]</td>
<td>Resin viscosity</td>
</tr>
<tr>
<td>( \Delta P )</td>
<td>67 [kPa]</td>
<td>Pressure drop</td>
</tr>
<tr>
<td>( \kappa )</td>
<td>( 3.63 \times 10^{-11} ) [m(^2)]</td>
<td>Permeability of charges</td>
</tr>
</tbody>
</table>

### 6.4. Modeling Process Time Variance

The ACCEM and COSTADE unit process models estimate process times based on input geometries but do not provide methods for estimating the variability of the process times. The process time, captured by the squared coefficient of variation is required for analysis of the CONWIP production line. This section describes a method for estimating processing time squared coefficients of variation when using unit process models from ACCEM and COSTADE and helps to show the nature of the time predicted by these two modeling methods.

The literature on the ACCEM [8], [62] state that the production time predicted by the ACCEM models is the estimate only for the basic work content. The total process time is the time required to complete this basic work content in addition to any time caused by delays or any other “element of variance.” Error! Reference source not found., taken from the literature, shows how the total process time is comprised of both the basic work content and the elements of variation.
Figure 32. "Elements of Factory Labor" from [8] shows variable sources of additional process time beyond what the standard models predict.

For the ACCEM process models, the total work content for a given cell (or total process time) $T_i$ is a random variable. It is the sum of the deterministic time required for the basic work content (as predicted by the equations in the ACCEM) $\bar{t}_i$ and the time required to address the random elements of variance $N_i$. So for a given cell $i$

$$T_i = \bar{t}_i + N_i$$

(64)

This equation shows that the total, stochastic processing time can be estimated with a linear noise model. Equation (64) in terms of the two moment approximation for processing time in cell $i$ used in the production line models:

$$t_i = E[T_i] = \bar{t}_i + E[N_i]$$

(65)

$$t_i^2 \sigma_{s,i}^2 = Var[T_i] = Var[N_i]$$

(66)

Finding these first two moments shows that all of the variability comes from (perhaps unsurprisingly) from the "elements of variance".

How should this $N_i$ be modeled? First, note that only the first two moments of $N_i$ are used. Also note that as a representation of the delay, $N_i$ should be positive. It doesn’t make sense for variable representing
a delay to the processing time to be negative and therefore reduce processing time. Given, \( N_i \geq 0 \), any variability in the process indicated \( E[N_i] \geq 0 \).

Literature finds that delays in manufacturing systems can be well approximated as being exponentially distributed. [55], [63], [64] Thus, when using models from the ACCEM, this research assumes \( N_i \) is exponentially distributed with rate \( \frac{1}{\beta} \) such that \( E[N_i] = \beta \) and \( Var[N_i] = \beta^2 \). If the designer fixes the \( c_{s,i}^2 \) variable as an input, the variable is calculated from (65) and (66). The input for \( c_{s,i}^2 \) can be selected based on known data for similar processes or explored as a trade-off.

The COSTADE unit process models already include a delay term as seen in equations (56) and (57). Therefore, a process time \( \hat{t}_i \) estimated using COSTADE models is already the approximation for the first moment of \( T_i \); \( \hat{t}_i = t_i = E[T_i] \).

The COSTADE models were developed with the expectation that only the setup and delay portions of the process contribute to the variability of the processing time. That is to say:

\[
Var[T_i] = Var\left[\left(\frac{\text{Setup Time}}{\text{Run}}\right)\right] + \left(\frac{\text{Operations}}{\text{Run}}\right)Var\left[\left(\frac{\text{Delay}}{\text{Operation}}\right)\right]
\]

assuming the delay and the setup are independent.

If the delays and the setup times we assumed to be exponentially distributed again with rate \( \frac{1}{\beta} \), the expected variation for the process can be calculated directly from the COSTADE model parameters. If \( \beta_1 = \left(\frac{\text{Setup Time}}{\text{Run}}\right) \), \( \beta_2 = \left(\frac{\text{Delay}}{\text{Operation}}\right) \)

\[
t_i^2 c_{s,i}^2 \geq \beta_1^2 + \left(\frac{\text{Operations}}{\text{Run}}\right)\beta_2^2
\]

Of course, there is nothing preventing the designer from fixing the \( c_{s,i}^2 \) variables as inputs. These inputs can be based on variability measures of existing processes similar to the one in the model. Or the design can experiment with the effects of process time variability. When used with unit process models to approximate \( t_i \) and with fixed \( c_{s,i}^2 \) as input, the designer explicitly defines the two moments of the underlying distribution for \( T_i \) as: \( E[T_i] = t_i = \hat{t}_i \) and \( Var[T_i] = c_{s,i}^2 \).

The designer is free to choose how to consider process time variation. The point of this section is simply to show the different options for how process time variation can be considered when incorporating unit
process models. Also, this section shows the ramifications of certain choices on the underlying stochastic models.

It is interesting to note, especially in the case of the COSTADE models, despite significant effort to estimate process times, there is not explicit consideration for the variability of those process times. Yet, understanding variability is crucial to understanding the production system and, as a consequence, production costs.

6.5. Summary

This chapter shows how existing unit process models can be used in a GP model to estimate processing times from part geometries. The unit process models are derived primarily from COSTADE and ACCEM literature. Additionally, a model for resin infusion is included. It is shown how the unit process models can be formulated as GP constraints. This chapter also discusses different methods for estimating squared coefficients of variation when using process models.
7. **GP Optimization: Closing the Loop around the Factory**

The previous chapters have given the reader an overview how to represent the stochastic flow of goods and the composite processes in a way that is compatible with geometric programming. This chapter will document some of the ways in which geometric programming can be used to solve for different unknowns and objectives. The foremost optimization problem will be to minimize cost subject to performance.

### 7.1. Design and Trade Spaces

One can easily see that a trade-off exists between design objectives. A more powerful vehicle generally costs more to make. To improve one factor, another factor must become worse. The challenge is to find the optimum.

![Some figure which demonstrates this idea in everyday life](image)

#### 7.1.1. Design Space

The design process often refers to different types of “spaces”. A space in the mathematical sense represents how many different variables relate to one another. In the everyday world, one of the more common spaces we interact with is 3D or cartesian space. In school, we learn that three numbers (x,y,z) can describe where something is in the world around us. As a corollary, all three numbers, (x,y,z) are required to precisely define a point in the space. If one of the numbers is missing, is it not possible to precisely define a point in the space. This is to say, the variables x,y,z define the space.

A mathematical space does not need to just be defined by variables that describe spatial position. A space can be defined by nearly any set of variables. Thus, the design space is the mathematical space defined by (spanned by) the design variables. The design space contains every possible combination of the design variables. As a simple example, imagine we are tasked with designing a cardboard box. The design space would typically include the width, length, and height of the box. Given how much freedom the designer will have, it could also include the thickness of the corrugate to be used, the dimensions of the flaps, or the location of handles. If we include just the short list of the design variables here, the design space is already 6-dimensional!
Even in this simple example, it is easy to see how the design can grow to include a huge number of variables and thus how the design space can quickly grow into several dimensions. Now imagine a design as complex as an airplane. Each of its six million parts is in itself described by several variables!

7.1.2. Feasible Subspace

The design space is huge, because it includes every possible design for the object being studied. Some areas of the design space define designs that are impossible to actually achieve in the real world. In the box example, impossible designs may be those where the length of the box is zero. Or in the context of aircraft design, an impossible design may be that where the wing and engines are too small for the airplane to generate enough lift to actually overcome gravity and fly. These regions of the design space where particular points in the design space define a design which cannot be realized are called “infeasible” regions. The regions where valid designs exist are called “feasible” regions.

7.1.3. Trade Subspace and the Efficiency Frontier

The first requirement for any real design is that it is feasible. However just because a design is feasible does not make it a good design. Consider the design space for cars; two examples of feasible designs would be an F1 car and a family minivan. We know these are feasible because they actually exist in real life. Feasibility is the most basic condition by which a design can be measured. In order to measure the “goodness” of a design, one has to have some metrics by which to evaluate particular designs. These metrics are roughly divided into two broad categories: performance and cost.

Performance and cost metrics are calculated results of design inputs. In the box example, a performance metric could be the total interior volume of the box which results from the length, width, and height inputs we discussed before. An example cost metric for out box example could be a simple formula based on the surface area of the cardboard used to build the box. Thus, performance and cost metrics can be used to compare different designs to one another and judge “goodness”. If we consider two designs for a box which enclose the same volume but one required much more cardboard, and thus costs more, we could use the cost metric to say that one design is better than the other. This comparison assumes that the only metrics of interest are volume and cost.

This example highlights the very contextual nature of the metrics to compare candidate designs. We return to the case of the F1 car and the minivan to demonstrate the importance of contextualizing the design metrics. The context of the design which helps define the overall goal also helps to define the importance given to different metrics. Contextualizing the metrics and thus giving the metrics different priorities is what allows us to say that an F1 car is good for racing and a minivan is good for a family on a road trip.
The process of defining the context of the design and thus which metrics should be prioritized is by no means an easy task. In the business world, for example, many tools for helping define and prioritize these metrics are based on trying to figure out what aspects of product influence customers’ demand.

Let’s consider again the problem of designing a minivan. For simplicity, we will assume that the only two metrics by which the minivan designs are measured are weight and cost in order of increasing importance. A designer first proposes that the new minivan be made of solid gold. After a pause, the same designer suggests perhaps, the minivan should be made from steel instead. The rest of the design team voices their agreement with the change. By changing the material to steel, the designers can improve both cost and weight. Maybe it won’t look as cool but the metrics set forth for the minivan design exclude looks. That the design could be improved on multiple metrics without making any metric worse is a good sign.

As the design is developed and metrics are only improved, the design moves closer to what is called the efficiency or Pareto frontier. Once a design is on this frontier, one metric cannot be improved without making another worse; improving one metric has to be “traded-off” against making another metric worse. If the feasible design space is further restricted to designs which lie on this frontier, this gives rise to what is called the “tradespace”. To say a design is “good” is tantamount to saying that the design is in the tradespace because there are no longer design metrics which can be improved without degrading other metrics. Once the tradespace is defined, the design activity then becomes about moving among designs in the tradespace as the different metrics are given different priorities.

The efficiency frontier and the tradespace are not static. New technologies are constantly being introduced which can substantially change the tradespace. Consider how technological improvements to computer chips which drastically increased speed and decreased power requirements (improvements to two metrics) coupled with new battery technologies (improve power/weight metric) produce a pocket-sized, portable smartphone which has an order of magnitude more computing power than a room-sized supercomputer from forty years ago.

It should be considered how different metrics compare to one another. Metrics can be understood as different representations of reality. We humans have developed different abstractions in description to help us make sense of the realities around us. For example, a book may be described as “rectangular, green, 128 pages” or the very same book could be described more abstractly as a “tragic play wherein ‘a pair of star cross’d lovers take their lives’.” Similarly, metrics can be abstracted. An airplane design can be measured by its wingspan and number of engines or, in a more abstract sense, the same airplane could be measured by its ability to generate profit for an airframer. Generally, the more abstraction can be applied, the fewer metrics are ultimately important but the remaining metrics become increasingly more complex to measure.
7.2.  Optimization

An optimization can generally be described as mathematical decision making. Across an organization, people understand trade-offs, although they may be hard to see and difficult to bring together and weigh against each other. Optimization solvers bring these complex trade-offs together and allow the user to define constraints, degrees of freedom, and an overarching objective, e.g. minimize production cost for certain design and system constraints.

An important observation is that insight can come from anywhere in an organization. The convex approach illuminates these tradeoffs and thus enables everyone across the organization to contribute their insight.

The issue with solving cost functions for minimum cost is the complexity of how the factors tie together. The beauty of GP is how swiftly complex systems can be solved. The condition for using GP, however, is to parameterize the design space with a combination of monomial equalities and posynomial inequalities as outlined in Chapter 3.

Generally, an optimization consists of a single objective function and one or more constraints.

An example for an important objective function could be: minimize total cost, i.e. the sum of fixed and variable cost.

Constraints include the flow model and process model relationships derived in the preceding chapters, for example: the bottleneck rate is the smallest (i.e. slowest) of all the rates across all stations; process times are a function of the COSTADE model times, and so forth. The constraints generally span a multi-dimensional tradespace.

Geometric programs solve convex optimization problems, so the objective function is always written as a minimization function.

In general, geometric programs are written as:

\[
\begin{align*}
\text{minimize} & \quad \text{[objective function]} \\
\text{subject to} & \quad \text{[list of constraints]}
\end{align*}
\]

7.3.  Using GP in Conceptual Design

The mathematical advantages of the non-linear, convex optimization that is geometric programming is in itself very useful but there are ancillary benefits that come from using GP in conceptual design.

The multi-disciplinary nature of the conceptual design activity requires an equally multi-disciplinary team to perform the design. At times, it is difficult for one domain to see how decisions may impact another
domain. Designers, regardless of domain, have a fundamental understanding of trade-offs. Though explicit trade-off analysis and sensitivity analyses with GP, these cross-disciplinary effects are easily revealed. In this sense, GP acts as a sort of trade-off translator and allows design innovation across all of the disciplines involved. Moreover, the high speed of the convex solvers for GPs allows this translation to happen over much shorter time-spans than previously possible (in many cases less than a second of delay and in most cases there is less than a few seconds between solutions). In this sense GP enables real-time translation of the design into trade-offs.

The non-linear formulations of GPs allow easy contextualization of the models. Variables can be fixed as inputs or left free to be optimized depending on the particular exploratory goals of a particular context. Different disciplines can perform different, domain-specific trade studies but all using the common knowledge-base integrated into the model. Design might study different airfoil shapes, operations may study the impact of different manufacturing processes, and finance may be interested in the performance of different financial metrics.

Finally, supports a disciplined design process wherein constraints on the design are added (and in-turn model fidelity increased) incrementally. Beginning with the simplest, conservative, representation of the design space, sensitivities can be used to find the areas of the design space where additional modeling is needed. Hence, the complexity of the design grows in step with the complexity of the underlying models.
8. **Case Studies**

The preceding chapters have demonstrated how the complex relationships of production flow and manufacturing can be represented in a way that is compatible with GP. The following chapter will present key implementations and examples that demonstrate the potential impact of the cost estimation and optimization technique presented in the previous chapters of this dissertation.

Two case studies are presented in this chapter.

### 8.1. GPkit Implementation

The GP formulations of the cost models presented above are solved with convex optimizers as pointed out in chapter 3.1. GPkit is an open-source python library (http://www.github.com/convexengineering/gpkit) which allows design problems, formulated as geometric programs, to be easily formulated and solved inside a python environment. GPkit acts as an interface to convex optimizers such as CVXOPT and Mosek and helps the designer build models in a more intuitive fashion rather than having to deal with the large input and output matrices of the optimizers (see chapter 3.1 for more detail). GPkit is developed by the Convex Engineering Group at MIT (http://convex.mit.edu). The manufacturing cost models presented in Chapter 4, Chapter 5, and Chapter 6 were implemented as python classes in GPkit. The case studies below use this python implementation to evaluate the geometric programs.

### 8.2. Evaluation of Existing Manufacturing Process

The first experiment validates simulation and GP model results against a known factory operating state. The second set of experiments keeps the same configuration of the factory and predicts how increasing WIP inventory would affect production rate. The third study in the case, assumes costs for adding inventory and costs for adding additional workstations to each cell and investigates cost-optimal strategies for expanding a production line to meet higher production rate requirements.

#### 8.2.1. Motivation

This first case study uses the GP flow and cost models to evaluate the performance and opportunities within an existing production system. To protect proprietary data, the production data are normalized without invalidating the methods and conclusions. True cost information is equally (if not more) proprietary.
than production data so representative cost figures are used for studies that consider measures of production cost.

The goals of the study are to verify results from GP model studies and results from simulation studies with the performance of a known factory at a single, known operating point. Once validated against simulation and the real production line, the GP model is used for trade-off and sensitivity analyses.

8.2.2. Data Sources

The production line is a serial flow line producing a single product operating under a CONWIP policy. The production line comprises 8 production cells in series, each with one or more parallel workstations processing the parts as detailed in Table 12. Flow model parameters were estimated from real production data from two sources: historical process time data collected by the company (that spans about five years) and high-fidelity data collected by the author spanning the production of about thirty units. The data collected by the author includes detailed flow times, including records of processing times and queueing times, and a record of inventory position. The detailed data are used to establish the known operating point of the factory.

The average production rate of the factory, which is operated with a CONWIP policy, is calculated using Little’s Law and the average measured flow time. By definition, the amount of work in progress was constant for period over which the author recorded the high-fidelity data. Since the production line was observed for a relatively short while, the bootstrap resampling method was used to estimate the true, long-run average production rate from the samples. The bootstrap method produces a distribution of the likely, long-run average production rate. The reported, measured production rate is the mean of this bootstrapped distribution and is reported along with the standard deviation of the same bootstrapped distribution for production rate.

Table 12 shows the first two moments of the historical process time data. These two-moment fits are used in the production flow models. The number of parallel workstations $m_i$ and a representative cost of each workstation for each cell are also. The representative cost figures were established with input from experts. Additional inputs are listed in Table 13.

Table 12. Cell characteristics.

<table>
<thead>
<tr>
<th>Cell Number ($i$)</th>
<th>Process time characteristics</th>
<th>Fixed cost per workstation</th>
<th>Number of Stations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_i$</td>
<td>$t_i$ [time units]</td>
<td>$c_{x,i}^2$</td>
<td>$\omega_{0,i}$</td>
</tr>
</tbody>
</table>
Table 13. Additional model inputs.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H$</td>
<td>Inventory cost (present value)</td>
<td>20</td>
</tr>
</tbody>
</table>

8.2.3. Model Formulation

GP models for the production line are implemented as shown in Chapter 4. Since the variables for average process time and squared coefficient of variation are fixed, unit process models are not needed for this case to predict the process times. Only the financial model for initial cost (44) is included in this case and the cost variables are fixed as shown in Table 12 and Table 13.

Table 14. GP model implemented for the study of a current factory.

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{\lambda} \\
W_i & \geq \left(\frac{t_i}{m_i}\right) \left(\frac{\rho_i}{\alpha_i}\right) \left(\frac{C_{a,i}^2 + C_{s,i}^2}{2}\right) + t_i & \forall i \in [1, M] \\
1 & \geq \rho_i + \alpha_i & \forall i \in [1, M] \\
c_{d,i}^2 & \geq \alpha_i c_{a,i}^2 + \rho_i c_{s,i}^2 & \forall i \in [1, M] \\
\lambda & = \frac{\rho_i m_i}{t_i} & \forall i \in [1, M] \\
c_{d,i}^2 & = c_{a,i+1}^2 & \forall i \in [1, M - 1]
\end{align*}
\]
The output from the GP is compared to that from a discrete event simulation. The discrete event simulations were run using Simio software. The random processing times were drawn from a Gamma distribution matched to the first two statistical moments of the actual processing time data shown in Table 12. The Gamma distribution is parameterized by the shape $\theta$ and the scale $k$ for each cell $i$ where

$$\theta_i = t_i c_{s,i}^2, \quad k = \frac{1}{c_{s,i}^2}, \quad \forall i \in [1,9]$$

(69)

The simulation parameters are summarized in Table 15. The simulation was allowed to run for enough time that variance of the production rate was small (CV < 0.005) and each experiment was repeated 100 times to achieve better convergence to the true mean of the estimated production rate.

Table 15. Simulation parameters for comparison case study.

<table>
<thead>
<tr>
<th>Simulation Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run length</td>
<td>2000 [time units]</td>
</tr>
<tr>
<td>Number of runs per point</td>
<td>100</td>
</tr>
<tr>
<td>Warm up period</td>
<td>100 [time units]</td>
</tr>
<tr>
<td>Confidence interval</td>
<td>95%</td>
</tr>
</tbody>
</table>

8.2.4. Results

The first study compares the measured production rate of the real system (with error bounds) to the estimates of production rate obtained from simulation and the GP-models. The real system was operated with CONWIP count $L = 12$ which is input to both the simulation and GP models.
To obtain an estimated production rate from the GP model, all of the variables (including $L$) are fixed and the constraints are put into an optimization problem with the objective to minimize $\frac{1}{\lambda}$ which properly bounds all the variables so all the constraints are tight and the GP formulation is exact. The resulting optimization problem is solved to find the production rate. In this case, solving the GP model is simply finding a solution to the simultaneous, non-linear equations. Maximizing rate ensures all the constraints are tight.

Table 16. Comparison of production rate of the system with the production rate predicted by simulation and GP models

<table>
<thead>
<tr>
<th></th>
<th>Production Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual System ($\pm \sigma$)</td>
<td>0.7332 ± 0.024</td>
</tr>
<tr>
<td>Simio Simulation</td>
<td>0.7329</td>
</tr>
<tr>
<td>GP Models</td>
<td>0.7285</td>
</tr>
</tbody>
</table>

Table 16 compares the estimates of the production rate to the likely average production rate as measured from the real production data. Production rate of the real system was calculated using Little’s Law and the measured total flow time of about thirty units. Both estimates are within the one standard deviation error bounds of the real production line. The GP model offers a slightly conservative estimation which is anticipated by the derivation in Chapter 4. Both methods yield extremely good estimations of the performance of the real system.

At this operating point, sensitivity of production rate to the different inputs is evaluated.

Figure 33. The sensitivity of the inverse of the production rate to average processing times.
Figure 33 shows the fractional sensitivity from the GP model of $1/\lambda$ to the expected processing time in each cell $t_i$. Of all the cell production times, the production rate is most sensitive to the process time of Cell 2, as shown in the graph. Bearing in mind that sensitivities reported from GPs are fractional and that the objective function is $1/\lambda$, the reported sensitivity for $t_2$, 0.827 indicates that a 1% decrease in the process time in Cell 2 would yield an 0.827% decrease in the objective function. Since the objective is $1/\lambda$ a 1% decrease in Cell 2 average processing time would lead to a 0.827% increase in production rate. This makes sense as Cell 2 is the bottleneck of the system so improving the bottleneck (making the bottleneck faster) should improve the overall production rate of the system. The low sensitivities of the other process times show that they are not constraining the system.

![Graph showing fractional sensitivities to other selected inputs.](image)

Figure 34. Fractional sensitivities to other selected inputs.

Figure 34 shows sensitivity of the objective function to input variables $t_2, m_2, L$. The influence of $t_2$ was discussed previously. Increasing the number of parallel stations at cell 2 would increase production rate as would increasing the inventory in the CONWIP policy. A graph such as this one can help identify which constraints have the greatest impact on the objective.

This begs the question, what is the trade-off between increasing the inventory and the production rate. This experiment cannot be feasibly performed on the real factory but it was shown that simulation and GP models are good approximates for how the real factory performs. We can reevaluate the simulation for
cases where $L = 10, 11, 12, 15, 20, 25$. The GP model allows fractional WIP counts as input and therefore we can fix $L$ to any value. The objective to minimize $1/\lambda$ is the same and we solve the GP with $L$ for 100 points evenly spaced on the interval [10,25]. The results of the trade-off analysis are shown in Figure 35.

![GP Approximation Compared to Simulation of Real Factory](image)

Figure 35. A trade-off of the rate achievable by only varying the amount of inventory in the production line.

The figure also shows where the real operating point (with $\pm\sigma$ error bars) of the actual factory falls in the trade study. Both the simulation and GP models predict an increase in rate from increasing the CONWIP count $L$. Furthermore, the graph displays the conservative nature of the estimates from the GP model.

The figure also shows how, around 13 or 14 units in the CONWIP count, the production rate quickly plateaus with addition of CONWIP units. This is well-known behavior that increasing inventory without changing the underlying factory, will cause the production rate to asymptotically approach the theoretical maximum production rate. In this example, the theoretical maximum production rate is the bottleneck rate (the rate of the slowest cell) which is Cell 2. The fact that Cell 2 is the bottleneck, also shows up in the sensitivities that overall production rate is most sensitive to the rate in Cell 2.
In a third experiment, we determine how much it would cost to construct a production line where the required production rate is input. The required rate is increased to obtain cost estimates for expanding the future capacity of the factory. In this case all of the variables determining capacity $m_t$, as well as the variable for inventory $L$ are free variables. The production rate variable $\lambda$ is fixed as an input and the objective of the optimization is to minimize $C_0$ where $C_0$ is defined in Table 14. The study is executed for a range of production rate $\lambda \in [0.7, 1.0]$ and the results are displayed below in Figure 36.

![Cost vs rate trade-off analysis using GP models.](image)

With the capacity and inventory variables free to be optimized, the optimization adds capacity and inventory in a cost-minimizing way to achieve the required production rate. Each point along the line represents a different production line design that has been optimized to minimize cost. The cost-production rate trade-off is slightly sub-linear as illustrated in the graph with a line drawn between the minimum and the maximum of the GP model result.

A trade-off like this one could be useful when considering expansion projects to understand the cost required to increase the production rate.
8.3. Manufacturing cost trade-studies for conceptual composite fuselage sections

The GP cost models are used to compare alternative composite manufacturing process for fuselage manufacturing: prepreg and resin infusion. The prepreg process is what is currently employed in fuselage fabrication. However, there are some engineers who believe the infusion process would lower costs. This trade-off is explored using GP models.

8.3.1. Background

The motivation is to compare the costs of building production lines that utilize two different composite fabrication techniques: preimpregnation (prepreg) or resin infusion. The two techniques, whose constituent processes are laid out in Figure 37, differ by how the resin, which is ultimately cured to bond the carbon fiber together, is introduced to the production process. In the prepreg technique, carbon fibers that have already been preimpregnated with epoxy resin arrive as the raw material and are laid up in their slightly-tacky state. The prepreg stack is cured under heat and pressure in an autoclave oven. In the infusion method, carbon fibers are laid up dry and then resin is infused into the dry fiber stack under pressure until the fibers are saturated. The stack is then cured in an oven. The reason some believe the infusion technique to be cheaper is that the layup equipment is faster and cheaper than prepreg and it does not require an extremely expensive autoclave to cure the parts.
The two different techniques are compared for constructing the same conceptual fuselage design. The comparative analyses are performed by implementing two GP cost models, one for each of the production alternatives, and comparing the solutions. The same fuselage geometry inputs and factory inputs are shared between the two, parallel GP models. Figure 38 shows the parallel GP models used to compare the processing techniques.

Figure 38. Implementation of comparative GP cost models considering the identical fuselage designs but manufactured in production lines executing different production processes.

Inputs for the factory flow and fuselage geometry were included, see section 8.3.2 (Data Sources).

Cost was shown as a function of rate for the two alternatives, for producing the same fuselage design, see section 8.3.4 (Results).
8.3.2. Data Sources

The process models for both prepreg and the infusion concepts were obtained from publicly available sources. [10], [61]. The cost data used in the study are representative however. The purpose of presenting this study is not to offer a definitive conclusion for which process should be used but rather to demonstrate the framework for analyzing the decision. In principle, a manufacturer interested in analyzing the trade-off in a real situation should replace the general, publicly-available parameters in the models with specific parameters derived from the company’s proprietary information.

For this study, the fuselage dimensional variables are fixed as inputs and all of the production line design variables are left free. Figure 39 shows the fuselage geometry variables used in the study. Additionally, the production rate is fixed as an input.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value [units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l_{fuselage}$</td>
<td>46 [ft]</td>
<td>Fuselage length</td>
</tr>
<tr>
<td>$d_{fuselage}$</td>
<td>23 [ft]</td>
<td>Fuselage diameter</td>
</tr>
<tr>
<td>$h_{skin}$</td>
<td>0.78 [in]</td>
<td>Skin thickness</td>
</tr>
<tr>
<td>$a_{stringer}$</td>
<td>9 [in]</td>
<td>Stringer average center-to-center spacing</td>
</tr>
<tr>
<td>$w_{stringer}$</td>
<td>16 [in]</td>
<td>Stringer width (flattened)</td>
</tr>
</tbody>
</table>

Figure 39. Fuselage dimension variables as inputs.

8.3.3. GP Formulation

A geometric program is optimized for each production approach: prepreg and infusion. Each of the processes shown in Figure 37 are modeled as their own cell in the production system. In this manner, the outlines of the techniques also serve as diagrams for the two conceptual factories that are modeled. The two formulations use the same models for total cost and for production line performance. The two production techniques are differentiated at the production cell level where there are different recurring and non-recurring costs and different unit processes. As indicated, most of the unit process models are derived from
the COSTADE models. The production line models and the financial models are as presented in Chapter 4 and Chapter 5 respectively. The GP models in their entirety are included as Appendix 3.

8.3.4. Results

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Top-Level Summary</th>
<th>Cell &amp; Process Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rate</td>
<td>10.00</td>
<td></td>
</tr>
<tr>
<td>Diameter</td>
<td>7.00</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>14.00</td>
<td></td>
</tr>
<tr>
<td>Holding Value</td>
<td>20.00</td>
<td></td>
</tr>
<tr>
<td>Holding Rate</td>
<td>0.10</td>
<td></td>
</tr>
<tr>
<td>Invt. CAPEX</td>
<td>3.00</td>
<td></td>
</tr>
</tbody>
</table>

Figure 40. As the designer moves the sliders, the inputs to the corresponding fixed variables are updated and the GP is solved again. Top-level results for comparing cost and performance of the two manufacturing techniques are displayed in addition to detailed results regarding the production system. Solving the model takes about 50 milliseconds on a laptop so the designer is able to get near-instant feedback about how changing the design affects the cost of the production systems.
Figure 40. Point-by-point design space explorer for fuselage manufacturing process trade-off.

To further compare the two manufacturing techniques, sensitivities for selected variables can be analyzed for the different production techniques. Since the objective function is the total cost, the sensitivity data are useful for identifying the cost drivers for the different manufacturing alternatives. Figure 41 shows a chart of selected sensitivities for the two different production processes.

![Sensitivity chart showing total cost and sensitivities for selected inputs for infusion and prepreg processes.]

Figure 41. Sensitivity analyses for selected inputs to help compare two different production processes.

At the particular design point (a particular set of inputs) shown in Figure 41, the GP model predicts a slightly lower cost factory for an infusion process than the corresponding factory for a prepreg process. The designer should not stop and simply choose the apparently cheaper option.

Figure 41 shows, for the same selected fixed variables, the infusion process is more sensitive across the board than the prepreg. This can be used to judge the “volatility” of a concept. Since the sensitivities for the infusion line are higher than the prepreg line, a given perturbation to the inputs will have a more significant impact on the cost of an infusion line. Given that inputs almost always change during the design process, the infusion line appears to be more volatile. If the designers were to select the infusion process only for its lower cost and the fuselage geometry were to change, the cost advantage may quickly be eaten up.
This idea of “volatility” also has important ramifications for later design phases. As more resolution is added and the models are converted to the unrelaxed forms, costs are likely to increase as inputs are changed. At this particular design point, the designer needs to be very wary because the cost difference is relatively small and one of the strategies has a very volatile cost. Even though there is not a clear winning strategy at this design point, all of the information the models provide is helpful for understanding exactly why there is no winning strategy and thus knowing what would have to change for there to be a clearer outcome.

Simply knowing an input is a cost driver is of course helpful, but as design complexity increases, it becomes useful to understand how a top-level input is driving overall cost through its interaction with various lower-level models. Recall from Chapter 3, equation (15) how the sensitivity for a single variable is found by summing together the partial sensitivities of every constraint to the variable of interest. For some variables, it can be very informative to instead display all of these constraint-level sensitivities. The fuselage diameter, for example, is a significant cost driver but it appears in several different constraints in the model. Figure 42 shows the constraint-level sensitivity break down for the fuselage diameter in the resin infusion process. The figure shows that most of the cost sensitivity to the fuselage diameter comes through Layup step in the manufacturing process.

![Fuselage Diameter Sensitivity Breakout](image)

Figure 42. Constraint-level sensitivity breakdown for fuselage diameter.
A cost vs rate trade-off is also a useful tool for comparing the two process alternatives. Figure 43 shows the cost vs rate trade-off between a conceptual production line implementing an infusion process and a conceptual factory utilizing the prepreg process.

![Cost vs Rate Trade-Off Graph](image)

Figure 43. Trade-off analysis showing the effect of required rate on the overall cost of the system for different production methods.

Though the infusion factory starts off less expensive than the prepreg factory, the cost of the infusion factory increases faster than the cost of the prepreg factory as the rate increases until the infusion factory becomes slightly more expensive. This behavior is anticipated by the sensitivity analysis shown in Figure 41 where the sensitivity of the infusion cost to the required rate is higher than that of the prepreg process.
9. CONCLUSION

This thesis presents a novel method for manufacturing cost modeling, based on geometric programming (GP). The motivating application for the work was to build a tool that would be useful during the conceptual design phase of a modern, composite, commercial airliner.

The novel GP formulation of the CONWIP production line approximations allows concurrent design of the product and the production system within the same mathematical framework. Unit process models provide a link between cells in the conceptual factory to the key attributes of a conceptual design. Finally, GP formulations of cost models reflect the fiscal impact of design decisions at an enterprise level by translating the engineering and operations trade-offs into dollars and cents.

The fast solutions and rich sensitivity analysis enabled by the GP formulation of the problem lend support to multi-disciplinary design teams working on complex projects where disciplinary boundaries inhibited communication and innovation.

More generally, this thesis shows how manufacturing cost estimation for a product can be modeled to include the unit processes required to create the product and to include the factories where these unit processes are performed in a production line. Moreover, the thesis demonstrates that these models can be solved with a non-linear, convex optimization approach that offers fast and reliable solutions.

Models are built using the framework for two case studies: one to explore the capabilities of an existing production system and another to study two different manufacturing process concepts in the context of a conceptual aircraft fuselage. Results from the case studies are offered to the reader as a guide for the types of decisions that can be supported with solutions from the GP models.

The GP frameworks developed in the thesis are implemented in Python using GPkit libraries. The models can be incorporated into existing systems modeled with GPkit. The case studies are performed using the GPkit libraries.

The approach, as well as many of the approximations, is applicable to production systems in general and has applications far beyond aerospace: production line models can be used with any CONWIP line. The financial principles that are modeled are common across different industries; the framework for integrating unit process models, since they are derived from fundamental process physics, is extensible to new processes.
Table 17. Significant contributions of the thesis.

<table>
<thead>
<tr>
<th><strong>GP Models for Composite Manufacturing Cost Estimation</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• GP flow approximations for CONWIP lines</td>
</tr>
<tr>
<td>• GP models for total production cost</td>
</tr>
<tr>
<td>• GP process models for composite fabrication</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Deployment of GP Manufacturing Models</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Performance analysis of existing production line</td>
</tr>
<tr>
<td>• Study manufacturing alternatives for new fuselage design</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Generalizable framework</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Reusable python library of GP production models</td>
</tr>
<tr>
<td>• Application to other types of production systems</td>
</tr>
<tr>
<td>• Cloud deployment on python</td>
</tr>
<tr>
<td>• Visualization tool for constraint sensitivities</td>
</tr>
</tbody>
</table>
10. **Future Work**

Future research for application to the field of conceptual aircraft design should be undertaken to integrate the cost modeling methods introduced herein with the GP models developed for conceptual aircraft design by others [65]–[67]. Completing this integration paves the way for simultaneous design and optimization of both product and production. Bringing product and production design into the same mathematical framework would allow for propagation of design requirements through to production costs and for top-level requirements to be captured and understood as the cost drivers through the use of sensitivity analysis.

Future work should also build out the manufacturing models for additional composites subassemblies like the wings and the empennage. Models for complex assembly unit processes (e.g. joining the wing to the fuselage) would complement the fabrication unit process models discussed here. These processes could then be assembled into the tiered supply chain to capture the manufacturing process from raw material through to final assembly. Additional research could be undertaken to address some of the limitations of the current production line models. Namely:

- **Join-lines**: the current model only considers a single part produced on a single line. Future work would develop models for co-optimizing parallel production lines where the products are joined at the end. Models of this nature could be helpful for modeling the fabrication of composite wings as the parallel fabrication of skins and spars which must be fastened together to produce a whole wing.

- **Multi-product lines**: the production line model that this work presents is only for a single part-type. Often, in production, different variants of similar products are produced using the same production line.

- **Integer GP Solutions**: the models implement a relaxed representation of the production line where the number of workstations can be continuous (e.g. non-integer). Future work would investigate integer optimization techniques (such as branch-and-bound) for solving an integer version of the problem.
There are also opportunities to utilize GP models outside of the conceptual design process. Initial work has begun to explore how the knowledge encoded into the GP models during conceptual design can be updated and utilized during more mature phases of design.

As a parting thought and example of what lay ahead, simple labor time models were incorporated with the GP models used to design the Jungle Hawk Owl, a long-endurance UAV that was built at MIT and was entirely sized using GP models.[66] Unit process models from the ACCEM for hand-layup are incorporated with the GP aircraft design models. Figure 44 shows the resulting labor cost estimate for building a single aircraft and also shows the cost sensitivities to the input requirements.

Figure 44. Layup labor cost model for a UAV designed with GP models.
REFERENCES


[26] D. N. Mavris, “A METHODOLOGY FOR PARAMETRIC PRODUCTION PLANNING IN PRELIMINARY AIRCRAFT DESIGN.”
[34] W. W. Hoburg, “Aircraft design optimization as a geometric program,” University of California, Berkeley, 2013.
[40] Simio. Sewickley, PA: Simio LLC.


11. APPENDIX I: THE BUSINESS OF COMMERCIAL AEROSPACE

Commercial aerospace generally refers to the segment of the larger aerospace sector which is not military or private. The average consumer in a developed country will most likely directly interact with commercial aerospace as a passenger. As a passenger, this average consumer will directly interact with the two major types of players in commercial aviation: the airline and the airframer. The airline sells the ticket to the passenger in exchange for the service of transporting the customer safely to their final destination. Worldwide, some airlines the reader may recognize are Air Canada, Delta Airlines, British Airways, Lufthansa, Emirates, Japan Airlines, etc. There are many competitive markets for air travel, each with many players.

11.1. Suppliers and customers

Regardless of what the airline promises to the passenger, no one is going anywhere without an actual airplane. Airframers are the companies that actually build and sell the airplanes. Unlike the market for airlines, the market of airframers is relatively small. As a matter of fact, especially in the markets for larger, transoceanic aircraft, the market is essentially a duopoly dominated by the European airframer Airbus and the Boeing Company, based in the US.
This image of aircraft parked at Los Angeles International Airport succinctly captures the state of the airline industry. The tails of these airplanes show seven different airlines (from near to far: Asiana (South Korea), Lufthansa (Germany), Japan Airlines (Japan), Air France (France), Emirates (United Arab Emirates), China Airlines (China), and British Airways (United Kingdom). All of these same seven aircraft, however, were built by either Airbus or Boeing (Boeing 747, Boeing 747, Boeing 777, Airbus A380, Airbus A380, Boeing 747, Boeing 747).

Generally both airlines and airframers are capitalistic enterprises, though there is constant accusation from airlines and airframers of unfair government subsidy. As businesses with a goal of maximizing profit, there is significant pressure on the price of new airplanes; the airlines as customers want lower prices and the airframers as suppliers would prefer higher prices. Overall, however, the market for commercial airplanes is dominated by high costs and small profit margins. For an airline, a new jet airliner which can carry passengers across the ocean can cost upwards of $100,000,000 USD {ref A320 ETOPS}. In the highly-competitive airline market, average net profit margins were 5.1% in 2017\textsuperscript{12}. For an airframer, the cost to design and develop a new airplane can be on the order of $10 billion USD. In the same year, airframer Boeing reported a net profit margin of 5.2\%.\textsuperscript{13} (Compare this with Apple who reported a net profit margin

\textsuperscript{12}http://www.iata.org/pressroom/pr/Pages/2015-12-10-01.aspx

\textsuperscript{13}https://www.stock-analysis-on.net/NYSE/Company/Boeing-Co/Ratios/Profitability/#Net-Profit-Margin
of 21.3% in 2016\footnote{https://www.marketwatch.com/investing/stock/aapl/financials} These large expenses and narrow margins put a lot of pressure on costs. It is imperative for the airframers to deliver products that deliver features to the airlines at a cost where the airline still can find value in operating them.

11.2. Designing a New Airplane (Product design)

When designing a new airplane, the challenge for the airframer is to provide a product to airline customers that has substantial performance but does not cost too much. A key metric for the airline, which provides a more holistic view of cost and performance, is the . Additionally, the airline operator usually seeks range and cruise speed performance in excess of some threshold values.

Modern commercial airliners are incredibly complex machines. The newest version of the iconic 747, for example, includes some six million parts. It would be very inefficient to design these millions of parts from the very beginning so a phased design approach is used instead. Companies differ in the precise nature of how the design is broken down but generally there are three main phases: conceptual design, preliminary design, and detailed design.

The design usually begins at the conceptual phase wherein it’s decided what the performance and cost goals are for the proposed product. A critical step in the conceptual design phase is to understand the needs of the customer and what value a new product would provide. In the context of an airframer, top-level characteristics of an airplane, such as range, cruise speed, passenger capacity, fuel burn, and target costs are decided during the conceptual phase. It is relatively inexpensive to produce a conceptual design compared to the costs of designing and building an entire airplane. It is also significantly less expensive to make changes to the conceptual design since there is usually very little, if any, physical hardware. There is a lack of detailed knowledge at the conceptual phase so flexibility has to be built into the conceptual design to account for these unknown details. For example, the conceptual design may predict a weight based on the strength of a structural material. However, only through detailed design, testing, and ultimately building the product, will the true, final weight of the aircraft be known. The nature of these unknown details gives rise to one of the most difficult aspects of conceptual design: including the right amount of “wiggle-room” in the design to allow for modifications as the design progresses and detail is added.

Overall, the product design process should be seen as iteration on the steps of collecting information, making decisions, and documenting the decisions. The design of complex things requires a great number
of decisions over a great number of variables. Dividing the design process into phases helps to break the process up into more manageable decisions.
12. **APPENDIX 2: FUSELAGE TRADE-OFF GP MODEL**

This appendix is an archive of the full GP model used to perform the fuselage manufacturing trade-off outlined in section 8.3. The GP model is formulated as an optimization to minimize cost. For ease of presentation, the constraints are broken up into groups where each group of constraints implements a different aspect of the overall model. Though the constraints are presented in groups, they will all be satisfied at optimality. The GPs for each of the two manufacturing process alternatives utilize the same models of the production line and total cost which will be presented first. Second, the model for the fuselage is included. Third, the unit process models which define the two alternative approaches are presented.

For each group of constraints, input parameters, input fixed variables, and free variables are listed. Variables which are free in some constraint groups may be fixed in other groups of constraints. In any case, all free variables must be fixed or fully bounded for the GP to be feasible.

### 12.1. CONWIP Production Line Model

Both production lines are single product, serial lines operating under a CONWIP policy. In both cases, the same number of cells comprise the production line. The CONWIP production line model derived in Chapter 4 is used.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value [units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>5</td>
<td>Number of cells in production line</td>
</tr>
<tr>
<td>$i$ $i \in [1, M]$</td>
<td></td>
<td>Cell index</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fixed Variables</th>
<th>Value [units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>10 [fuselage/month]</td>
<td>Required Production Rate</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Free Variables</th>
<th>[units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L$ [count]</td>
<td></td>
<td>CONWIP inventory</td>
</tr>
<tr>
<td>$m_i$ [count]</td>
<td></td>
<td>Workstation count at cell $i$</td>
</tr>
<tr>
<td>$t_{ij}$ [hours]</td>
<td></td>
<td>Processing time at cell $i$</td>
</tr>
<tr>
<td>$W_i$ [hours]</td>
<td></td>
<td>Total flow time through cell $i$</td>
</tr>
<tr>
<td>$\rho_i$ [unitless]</td>
<td></td>
<td>Utilization of cell $i$</td>
</tr>
</tbody>
</table>
### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value [units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M )</td>
<td>6</td>
<td>Number of cells in production line</td>
</tr>
</tbody>
</table>

### Fixed Variables

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value [units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r )</td>
<td>0.10 [1/year]</td>
<td>Discount rate</td>
</tr>
<tr>
<td>( h_r )</td>
<td>20 [USD/year]</td>
<td>Inventory holding value</td>
</tr>
</tbody>
</table>

### Constraints

\[
W_i \geq \left( \frac{t_i}{m_i} \right) \left( \frac{\lambda (c_{a,i}^2 + c_{s,i}^2)}{2} \right) + t_i \quad \forall i \in [1, M]
\]

\[
1 \geq \rho_i + \alpha_i \quad \forall i \in [1, M]
\]

\[
c_{d,i}^2 \geq \alpha_i c_{a,i}^2 + \rho_i c_{s,i}^2 \quad \forall i \in [1, M]
\]

\[
\lambda = \frac{\rho_i m_i}{t_i} \quad \forall i \in [1, M]
\]

\[
c_{d,i}^2 = c_{a,i+1}^2 \quad \forall i \in [1, M - 1]
\]

\[
c_{d,M}^2 = c_{a,1}^2
\]

\[
L \geq \lambda \sum_{k=1}^{M} W_k
\]

### 12.2. Total Cost Model

The total cost from building a production line and considering inventory holding costs in perpetual operation is used to compare the two processes. The financial models are introduced in Chapter 5.
### Free Variables

<table>
<thead>
<tr>
<th></th>
<th>[units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$</td>
<td>USD</td>
<td>Total cost</td>
</tr>
<tr>
<td>$L$</td>
<td>count</td>
<td>CONWIP inventory</td>
</tr>
<tr>
<td>$m_i$</td>
<td>count</td>
<td>Workstation count at cell $i$</td>
</tr>
<tr>
<td>$\omega_{0,i}$</td>
<td>USD/workstation</td>
<td>Cell $i$ workstation non-recurring cost</td>
</tr>
</tbody>
</table>

### Constraints

\[
C \geq \frac{1}{r} L h_r + \sum_{i=1}^{M} m_i \omega_{0,i}
\]

### 12.3. Fuselage Section Geometric Model

As outlined in section 8.3, the fuselage section geometry is characterized by a few key dimensions included here.

### Fixed Variables

<table>
<thead>
<tr>
<th></th>
<th>Value [units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l_{\text{fuselage}}$</td>
<td>46 [ft]</td>
<td>Fuselage section length</td>
</tr>
<tr>
<td>$d_{\text{fuselage}}$</td>
<td>23 [ft]</td>
<td>Fuselage section diameter</td>
</tr>
<tr>
<td>$h_{\text{skin}}$</td>
<td>0.78 [in]</td>
<td>Fuselage average skin thickness</td>
</tr>
<tr>
<td>$a_{\text{stringer}}$</td>
<td>9 [in]</td>
<td>Average center-to-center stringer spacing</td>
</tr>
<tr>
<td>$w_{\text{stringer}}$</td>
<td>16 [in]</td>
<td>Stringer flattened width</td>
</tr>
</tbody>
</table>

### Free Variables

<table>
<thead>
<tr>
<th></th>
<th>[units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_{\text{stringer}}$</td>
<td>[in²]</td>
<td>Flattened stringer width</td>
</tr>
<tr>
<td>$n_{\text{stringer}}$</td>
<td>[count]</td>
<td>Number of stringers</td>
</tr>
</tbody>
</table>

### Constraints

\[
A_{\text{stringer}} = (w_{\text{stringer}})(l_{\text{fuselage}})
\]

\[
n_{\text{stringers}} = \pi(a_{\text{stringer}})(d_{\text{fuselage}})
\]
12.4. Prepreg Unit Process Models

Each of the cells in the prepreg production has a model for the unit processes executed inside the cell. The unit process models lower bound the processing time. Cell costs are also presented as part of these unit process models. Specific references are cited for each unit process model but most come from COSTADE. A more in-depth description of the derivations for the unit process models is given in Chapter 6.

![Figure 45. Serial process diagram for Prepreg composite fabrication.](image)

12.4.1. Load Stringers

This process model is derived from COSTADE. The process involves inserting pre-formed, prepreg stringer charges into the fuselage mold.

### Fixed Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value [units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1$</td>
<td>1 [min]</td>
<td>Set up time per run</td>
</tr>
<tr>
<td>$K_2$</td>
<td>0.5 [min]</td>
<td>Delay per stringer</td>
</tr>
<tr>
<td>$K_3$</td>
<td>500 [in^2/min]</td>
<td>Dynamic constant</td>
</tr>
<tr>
<td>$K_4$</td>
<td>0.94 [min]</td>
<td>Dynamic constant</td>
</tr>
<tr>
<td>$\omega_{0.1}$</td>
<td>25 [USD workstation]</td>
<td>Workstation non-recurring cost</td>
</tr>
</tbody>
</table>

### Free Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>[units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_i$</td>
<td>[minutes]</td>
<td>Process time</td>
</tr>
<tr>
<td>$w_i$</td>
<td>[minutes^2]</td>
<td>Helper variable</td>
</tr>
<tr>
<td>$n_{stringers}$</td>
<td>[count]</td>
<td>Number of stringers</td>
</tr>
<tr>
<td>$A_{stringer}$</td>
<td>[in^2]</td>
<td>Flattened stringer area</td>
</tr>
<tr>
<td>$w_{stringer}$</td>
<td>[in]</td>
<td>Flatten width of stringer</td>
</tr>
</tbody>
</table>
12.4.2. Assemble Tooling

This process model is derived from COSTADE. The process involves assembling a segmented fuselage mold into a single piece.

### Fixed Variables

<table>
<thead>
<tr>
<th>Value [units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1$</td>
<td>1 [min] Set up time per run</td>
</tr>
<tr>
<td>$K_2$</td>
<td>0.5 [min] Delay per segment</td>
</tr>
<tr>
<td>$K_3$</td>
<td>22.50 [min] Dynamic constant</td>
</tr>
<tr>
<td>$K_4$</td>
<td>0.54 [min] Dynamic constant</td>
</tr>
<tr>
<td>$n_{\text{segment}}$</td>
<td>Number of segments</td>
</tr>
<tr>
<td>$\omega_{0.2}$</td>
<td>40 [USD] Workstation non-recurring cost</td>
</tr>
</tbody>
</table>

### Free Variables

<table>
<thead>
<tr>
<th>[units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_2$ [minutes]</td>
<td>Process time</td>
</tr>
<tr>
<td>$w_2$ [minutes$^2$]</td>
<td>Helper variable</td>
</tr>
</tbody>
</table>

### Constraints

<table>
<thead>
<tr>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_2 \geq K_1 + n_{\text{segments}}(\sqrt{w_1} + K_2)$</td>
</tr>
<tr>
<td>$w_2 \geq \left(\frac{l_{\text{fuselage}}}{K_3}\right)^2 + \frac{2(K_4)(l_{\text{fuselage}})}{K_3}$</td>
</tr>
</tbody>
</table>
12.4.3. Layup Skin

This process model is derived from COSTADE. The process involves layup of the prepreg carbon fiber skin with an automated fiber placement machine.

### Fixed Variables

<table>
<thead>
<tr>
<th>Value [units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_{\text{layers}}$</td>
<td>10 [count]</td>
</tr>
<tr>
<td>$r_{\text{laydown}}$</td>
<td>2848.1 $\text{in}^2/\text{min}$</td>
</tr>
<tr>
<td>$v_{\text{travel}}$</td>
<td>65.67 $\text{in}/\text{min}$</td>
</tr>
<tr>
<td>$w_{\text{tow}}$</td>
<td>0.25 [in]</td>
</tr>
<tr>
<td>$n_{\text{tows}}$</td>
<td>32 [count]</td>
</tr>
<tr>
<td>$z_{\text{positioning}}$</td>
<td>5 [sec]</td>
</tr>
<tr>
<td>$\omega_{0.3}$</td>
<td>$75 \frac{\text{USD}}{\text{workstation}}$</td>
</tr>
</tbody>
</table>

### Free Variables

| Description |
|----------------|-------------|
| $t_3$ [minutes] | Process time |
| $n_{\text{passes}}$ [count] | Number of passes per layer |
| $l_{\text{transverse}}$ [in] | Length dimension orthogonal to the layup direction |
| $l_{\text{travel}}$ [in] | Total head travel distance per layer |
| $A_{\text{layup}}$ [in$^2$] | Total layup area per layer |

### Constraints

\[
\begin{align*}
  t_3 & \geq n_{\text{layers}} \left( \frac{A_{\text{layup}}}{r_{\text{laydown}}} + \frac{l_{\text{travel}}}{v_{\text{travel}}} + \frac{(n_{\text{passes}})(z_{\text{positioning}})}{(n_{\text{tows}})} \right) \\
  n_{\text{passes}} & = \frac{l_{\text{transverse}}}{(w_{\text{tow}})(n_{\text{tows}})} \\
  l_{\text{travel}} & = \frac{A_{\text{layup}}}{(w_{\text{tow}})(n_{\text{tows}})} \\
  A_{\text{layup}} & = \pi(d_{\text{fuselage}})(l_{\text{fuselage}})
\end{align*}
\]
12.4.4. Bag and Prep

This process model is derived from COSTADE. The process involves placing a peel ply and a vacuum bag around the carbon fiber stack.

<table>
<thead>
<tr>
<th>Fixed Variables</th>
<th>Value [units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_{bags} )</td>
<td>1</td>
<td>Number of bags</td>
</tr>
<tr>
<td>( K_1 )</td>
<td>5 [min]</td>
<td>Set up time per run</td>
</tr>
<tr>
<td>( K_3 )</td>
<td>600 [in²/min]</td>
<td>Dynamic constant</td>
</tr>
<tr>
<td>( K_4 )</td>
<td>1.4 [min]</td>
<td>Dynamic constant</td>
</tr>
<tr>
<td>( K_{11} )</td>
<td>4 [min]</td>
<td>Peel ply setup time</td>
</tr>
<tr>
<td>( K_{33} )</td>
<td>347.63 [in²/min]</td>
<td>Dynamic constant</td>
</tr>
<tr>
<td>( \omega_{0.4} )</td>
<td>10 USD/workstation</td>
<td>Workstation non-recurring cost</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Free Variables</th>
<th>[units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_4 )</td>
<td>[minutes]</td>
<td>Process time</td>
</tr>
<tr>
<td>( w_4 )</td>
<td>[minutes²]</td>
<td>Helper variable</td>
</tr>
<tr>
<td>( A_{bag} )</td>
<td>[in²]</td>
<td>Area of the bag</td>
</tr>
</tbody>
</table>

Constraints

\[
t_4 \geq K_1 + \sqrt{w_4 + K_{11} + \frac{A_{bag}}{K_{33}}} \\
w_4 \geq \left(\frac{A_{bag}}{K_3}\right)^2 + \frac{2(K_4)(A_{bag})}{K_3} \\
a_{bag} = \pi(l_{fuselage})(d_{fuselage})
\]

12.4.5. Autoclave Cure

The stack is placed into the autoclave and the curing cycle is run.

<table>
<thead>
<tr>
<th>Fixed Variables</th>
<th>Value [units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_5 )</td>
<td>480 [min]</td>
<td>Autoclave cycle time</td>
</tr>
</tbody>
</table>
12.5. Resin Infusion Unit Process Models

Each of the cells in the resin infusion production has a model for the unit processes executed inside the cell. The unit process models lower bound the processing time. Cell costs are also presented as part of these unit process models. Specific references are cited for each unit process model but most come from COSTADE. A more in-depth description of the derivations for the unit process models is given in Chapter 6.

![Diagram of serial process](image)

Figure 46. Serial process diagram for Resin Infusion composite fabrication.

12.5.1. Load Stringers

This process model is derived from COSTADE. The process involves inserting dry stinger charges into the fuselage mold.

<table>
<thead>
<tr>
<th>Fixed Variables</th>
<th>Value [units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1$</td>
<td>1 [min]</td>
<td>Set up time per run</td>
</tr>
<tr>
<td>$K_2$</td>
<td>1 [min]</td>
<td>Delay per stringer</td>
</tr>
<tr>
<td>$K_3$</td>
<td>9.20 [in/min]</td>
<td>Dynamic constant</td>
</tr>
<tr>
<td>$\omega_{0.1}$</td>
<td>25 [USD/workstation]</td>
<td>Workstation non-recurring cost</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Free Variables</th>
<th>[units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>[minutes]</td>
<td>Process time</td>
</tr>
<tr>
<td>$n_{\text{stringers}}$</td>
<td>[count]</td>
<td>Number of stringers</td>
</tr>
<tr>
<td>$l_{\text{stringer}}$</td>
<td>[in]</td>
<td>Length of the stringers</td>
</tr>
</tbody>
</table>
12.5.2. Layup Skin

This process model is derived from COSTADE. The process involves layup of the prepreg carbon fiber skin with an automated fiber placement machine.

<table>
<thead>
<tr>
<th><strong>Fixed Variables</strong></th>
<th><strong>Value [units]</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_{layers} )</td>
<td>10 [count]</td>
<td>Number of layers which must be layed up</td>
</tr>
<tr>
<td>( n_{laydown} )</td>
<td>2848.1 ( \text{in}^2 ) ( \text{min}^{-1} )</td>
<td>Areal Laydown rate of the AFP machine</td>
</tr>
<tr>
<td>( v_{travel} )</td>
<td>65.67 ( \text{in} ) ( \text{min}^{-1} )</td>
<td>Linear, steady-state travel speed of the AFP machine while laying fiber</td>
</tr>
<tr>
<td>( w_{tow} )</td>
<td>0.25 [in]</td>
<td>Width of individual tow</td>
</tr>
<tr>
<td>( n_{tows} )</td>
<td>32 [count]</td>
<td>Number of tows on the head</td>
</tr>
<tr>
<td>( z_{positioning} )</td>
<td>5 [sec]</td>
<td>Time (delay) to stop and restart a course</td>
</tr>
<tr>
<td>( \omega_{0.2} )</td>
<td>75 ( \text{USD} ) [workstation]</td>
<td>Workstation non-recurring cost</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Free Variables</strong></th>
<th><strong>[units]</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_2 )</td>
<td>[minutes]</td>
<td>Process time</td>
</tr>
<tr>
<td>( n_{passes} )</td>
<td>[count]</td>
<td>Number of passes per layer</td>
</tr>
<tr>
<td>( l_{transverse} )</td>
<td>[in]</td>
<td>Length dimension orthogonal to the layup direction</td>
</tr>
<tr>
<td>( l_{travel} )</td>
<td>[in]</td>
<td>Total head travel distance per layer</td>
</tr>
<tr>
<td>( A_{layup} )</td>
<td>[in(^2)]</td>
<td>Total layup area per layer</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
  t_1 & \geq K_1 + n_{stringers} \left( \frac{l_{stringer}}{K_3} + K_2 \right) \\
  n_{stringers} & = \frac{\pi d_{fuselage}}{a_{stringer}} \\
  l_{stringer} & = l_{fuselage}
\end{align*}
\]
### Fixed Variables

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_2$</td>
<td>5 [min] Delay per caul</td>
</tr>
<tr>
<td>$K_3$</td>
<td>9.20 [in/min] Dynamic constant</td>
</tr>
<tr>
<td>$K_4$</td>
<td>1.4 [min] Dynamic constant</td>
</tr>
<tr>
<td>$n_{cauls}$</td>
<td>6 Number of caul plates</td>
</tr>
<tr>
<td>$\omega_{0.3}$</td>
<td>10 [USD/workstation] Workstation non-recurring cost</td>
</tr>
</tbody>
</table>

### Free Variables

<table>
<thead>
<tr>
<th>[units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_3$ [minutes]</td>
<td>Process time</td>
</tr>
<tr>
<td>$A_{caul}$ [in$^2$]</td>
<td>Area of each caul plate</td>
</tr>
<tr>
<td>$l_{stringer}$ [in]</td>
<td>Length of the stringers</td>
</tr>
</tbody>
</table>

### Constraints

$$t_3 \geq n_{cauls} \left( \frac{A_{caul}}{K_3} + K_2 \right)$$
12.5.4. Bag and Prep

This process model is derived from COSTADE. The process involves placing a peel ply and two vacuum bags around the carbon fiber stack.

<table>
<thead>
<tr>
<th>Fixed Variables</th>
<th>Value [units]</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_{bags} )</td>
<td>2</td>
<td>Number of bags</td>
</tr>
<tr>
<td>( K_1 )</td>
<td>5 [min]</td>
<td>Set up time per run</td>
</tr>
<tr>
<td>( K_3 )</td>
<td>600 ([\text{in}^2]/\text{min})</td>
<td>Dynamic constant</td>
</tr>
<tr>
<td>( K_4 )</td>
<td>1.4 [min]</td>
<td>Dynamic constant</td>
</tr>
<tr>
<td>( K_{11} )</td>
<td>4 [min]</td>
<td>Peel ply setup time</td>
</tr>
<tr>
<td>( K_{33} )</td>
<td>347.63 ([\text{in}^2]/\text{min})</td>
<td>Dynamic constant</td>
</tr>
<tr>
<td>( \omega_{0.4} )</td>
<td>10 (\frac{\text{USD}}{\text{workstation}})</td>
<td>Workstation non-recurring cost</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Free Variables</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_4 )</td>
<td>Process time</td>
</tr>
<tr>
<td>( w_4 )</td>
<td>Helper variable</td>
</tr>
<tr>
<td>( A_{bag} )</td>
<td>Area of the bag</td>
</tr>
</tbody>
</table>

Constraints

\[
\begin{align*}
  t_4 & \geq K_1 + \sqrt{w_4} + K_{11} + \frac{A_{bag}}{K_{33}} \\
  w_4 & \geq \left( \frac{A_{bag}}{K_3} \right)^2 + \frac{2(K_4)(A_{bag})}{K_3} \\
  a_{bag} & = \pi(l_{fuselage})(d_{fuselage})
\end{align*}
\]
12.5.5. Infuse and cure

This process model is derived from Darcy’s laws from fluid dynamics. Darcy’s law is used to estimate how much time it will take the resin to flow into the part. Parameters are derived from NASA publications. The process involves a compaction cycle, followed by the infusion of the resin. Finally, the part is moved into an oven and cured.

<table>
<thead>
<tr>
<th><strong>Fixed Variables</strong></th>
<th><strong>Value [units]</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{oven}$</td>
<td>251 [min]</td>
<td>Time required in oven to cure</td>
</tr>
<tr>
<td>$t_{debulk}$</td>
<td>1 [hr]</td>
<td>Debulking and compaction cycle time</td>
</tr>
<tr>
<td>$\phi$</td>
<td>40%</td>
<td>Porosity of the bulk charges</td>
</tr>
<tr>
<td>$\mu$</td>
<td>350 [mPa⋅s]</td>
<td>Resin Viscosity</td>
</tr>
<tr>
<td>$\Delta P$</td>
<td>67 [kPa]</td>
<td>Pressure differential driving resin flow</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>$3.63 \times 10^{-11}$ [m²]</td>
<td>Permeability of the charges</td>
</tr>
<tr>
<td>$\omega_{0,3}$</td>
<td>$50 \frac{[USD]}{[workstation]}$</td>
<td>Workstation non-recurring cost</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Free Variables</strong></th>
<th><strong>[units]</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_4$</td>
<td>[minutes]</td>
<td>Process time</td>
</tr>
<tr>
<td>$l_{rel}$</td>
<td>[in]</td>
<td>Relevant length to the flow problem</td>
</tr>
</tbody>
</table>

**Constraints**

\[
t_4 \geq t_{debulk} + \frac{\mu \phi (l_{rel})(h_{skin})}{\kappa \Delta P} + t_{oven}
\]

\[
l_{rel}^2 \geq l_{fuselage}^2 + (\pi d_{fuselage})^2
\]