Competitive Assessment of Aerospace Systems using System Dynamics

A Thesis
Presented to
The Academic Faculty

by

Jens Holger Pfaender

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Aerospace Engineering
Georgia Institute of Technology
December 2006

Competitive Assessment of Aerospace Systems using System Dynamics

Approved by:

Prof. Dimitri Mavris
Committee Chair
School of Aerospace Engineering
Georgia Institute of Technology

Prof. Daniel Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Jim McMichael
Aerospace, Transportation
and Advanced Systems Laboratory
Georgia Tech Research Institute

Dr. Peter Hollingsworth
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Fayette Collier
Langley Aeronautical Research Center
NASA

Date Approved: 17 November 2006

To mom and dad, for your love, patience, and support.

Nunc bibendum est!

ACKNOWLEDGEMENTS

I would like to extend thanks to the many individuals who have helped me through this endeavor. I especially want to thank Prof. Dimitri Mavris, my advisor, for giving me years of support and encouragement and for introducing me to a new way about thinking about aerospace engineering in general. Dr. Daniel Schrage who brought his considerable experience in aerospace systems design to my reading committee and I feel privileged to have had the opportunity to get a different perspective on things from him. I further want to thank Dr. Jim McMichael for his perspective and curiosity in the field of advanced design methods and putting up with my brief set of lectures in Advanced Design Methods III. Dr. Peter Hollingsworth provided me with much needed input on details about aircraft and engine calibration. He also provided me with much needed comments about the method in general. Last but not least, I would like to thank Dr. Fayette Collier for providing guidance on NASA Langley's Aerospace Systems Analysis perspective on important challenges for systems design and analysis, which provided some of the basis of this thesis.

ASDL at Georgia Tech has provided me with a number of contacts, experiences, and most importantly good friends. I would also like to thanks Dr. Michelle Kirby for providing me insights into FLOPS' quirks and peculiarities. Similarly, I also want to thank Dr. Elena Garcia for providing me with insight into the inner workings of ALCCA. In general, everybody that provided me with comments and corrections in the last months also deserves a round of thanks. I also want to thank Simon Briceno and Ismael Fernandez for their sharing of ideas and thoughts about each of our specialty areas and what they mean. Finally, Carl Johnson provided me with insights and tips into neural networks.

I will not attempt to list everybody that I got to know at Georgia Tech and in Atlanta, doing so would take up a lot of space. My time here has been enjoyable and fun. I'm looking forward to spending more time with you.

My family and friends in Germany that have given their encouragement and love all my life no matter where my endeavors took me. I could not have make it without you understanding and support this long way away from home. Mom, Alice, Dad, even if you are not with us anymore...thanks.

Thank you all — without you, I could accomplish nothing.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS OR ABBREVIATIONS

ASDL Aerospace Systems Design Laboratory.

ASM Available Seat Mile.

BRAINN Basic Regression Analysis for Integrated Neural Networks.

CE Concurrent Engineering.

EIA Energy Information Administration.

EIS Environmental Impact Statement.

FAA Federal Aviation Administration.

FAR FAA Regulation.

FLOPS Flight Optimization System.

FPI Fast Probability Integration.

HSCT High Speed Civil Transport.

IDE Integrated Development Environment.

IPD Integrated Product Development.

IPPD Integrated Product and Process Development.

JAR Joint Aviation Regulations.

LCA Large Civil Aircraft.

MADM Multi-Attribute Decision Making.

MFE Model Fit Error.

MRE Model Representation Error.

MTOW Maximum Take-off Gross Weight.

NASA National Aeronautics and Space Administration.

NPV Net Present Value.

OEC Overall Evaluation Criterion.

PDF Portable Document Format.

RDS Robust Design Simulation.

RDT&E Research, Development, Test, and Evaluation.

Re Reynold's Number.

RPM Revenue Passenger Mile.

RSE Response Surface Equation.

RSM Response Surface Methodology.

TIES Technology Identification Evaluation and Selection.

TIF Technology Impact Forecasting.

TOGW Take-off Gross Weight.

US United States.

UTE Unified Tradeoff Environment.

VCR Video Cassette Recorder.

VSLCD Virtual Stochastic Life-Cycle Design.

SUMMARY

Aircraft design has recently experienced a trend away from performance centric design towards a more balanced approach with increased emphasis on engineering an economically successful system. This approach focuses on bringing forward a comprehensive economic and life-cycle cost analysis. Since the success of any system also depends on many external factors outside of the control of the designer, this traditionally has been modeled as noise affecting the uncertainty of the design. However, this approach is currently lacking a strategic treatment of necessary early decisions affecting the probability of success of a given concept in a dynamic environment.

This suggests that the introduction of a dynamic method into a life-cycle cost analysis should allow the analysis of the future attractiveness of such a concept in the presence of uncertainty. One way of addressing this is through the use of a competitive market model. However, existing market models do not focus on the dynamics of the market. Instead, they focus on modeling and predicting market share through logit regression models. The resulting models exhibit relatively poor predictive capabilities.

The method proposed here focuses on a top-down approach that integrates a competitive model based on work in the field of system dynamics into the aircraft design process. Demonstrating such integration is one of the primary contributions of this work, which previously has not been demonstrated. This integration is achieved through the use of surrogate models, in this case neural networks. This enabled not only the practical integration of analysis techniques, but also reduced the computational requirements so that interactive exploration as envisioned was actually possible. The example demonstration of this integration is built on the competition in the 250

seat large commercial aircraft market exemplified by the Boeing 767-400ER and the Airbus A330-200. Both aircraft models were calibrated to existing performance and certification data and then integrated into the system dynamics market model.

The market model was then calibrated with historical market data. This calibration showed a much improved predictive capability as compared to the conventional logit regression models. An additional advantage of this dynamic model is that to realize this improved capability, no additional explanatory variables were required.

Furthermore, the resulting market model was then integrated into a prediction profiler environment with a time variant Monte-Carlo analysis resulting in a unique trade-off environment. This environment was shown to allow interactive trade-off between aircraft design decisions and economic considerations while allowing the exploration potential market success in the light of varying external market conditions and scenarios. The resulting method is capable of reduced decision support uncertainty and identification of robust design decisions in future scenarios with a high likelihood of occurrence with special focus on the path dependent nature of future implications of decisions. Furthermore, it was possible to demonstrate the increased importance of design and technology choices on the competitiveness in scenarios with drastic increases in commodity prices during the time period modeled.

Another use of the existing outputs of the Monte-Carlo analysis was then realized by showing them on a multivariate scatter plot. This plot was then shown to enable by appropriate grouping of variables to enable the top down definition of an aircraft design, also known as inverse design. In other words this enables the designer to define strategic market and return on investment goals for a number of scenarios, for example the development of fuel prices, and then directly see which specific aircraft designs meet these goals.

CHAPTER 1

INTRODUCTION

An era of air and space travel is a millennia old dream of mankind. Looking at predictions of the future of past decades or even centuries, everyday use of flying cars would be common place. There would be floating cities in the sky. Hypersonic flights around the globe would make intercontinental travel as fast (or slow) as a daily commute. There would be vacationing on the moon and have colonies on Mars. This is not how technology developed. Reality is in stark contrast to these visions of the "future". Some of the dreams were made into reality. However, most of them were not.

There is the fact that some of these ideas violate fundamental concepts of physics. However, other ideas also used to be dreams until sudden breakthroughs were achieved in airfoil design and control systems by the Wright brothers and then a few decades later in rocket propulsion. Arguably, rocket propulsion development was not funded out of pocket by the researchers themselves, but rather through years, or even decades, of military and government funding.

Looking at the history of the development of Aeronautics since then, it is a history of gradual optimization and improvement with only a few truly breakthrough developments such as the development of the jet engine and a controllable helicopter. However, this gradual process moved quite rapidly in the first several decades.

Nonetheless, since the introduction of the modern jet transport airliner, nothing has changed fundamentally. This is not to say that no improvements have been made. There have been significant gains in efficiency, reliability, economics, fuel consumption, and noise. This is especially true for important aircraft sub-systems such as engines and especially avionics, whose significant advancements might actually be waiting on other technologies. However, the fundamental design of these aircraft is still essentially the same cylinder with a mid-body low wing configuration with a traditional empennage configuration. This also means that the operational regime, such as flight speed, cruise altitude, landing requirements, is basically the same for this type of aircraft. This is not necessarily bad, since this fundamental design works and works well. It is also been heavily optimized to arguably fly in the most efficient flight regime possible, hence the essentially unchanged flight speed and cruise altitude. This only shows that efficiency is a major driver in aircraft design.

However, one might argue that there could be potential radical new designs that could offer more efficiency, speed than a conventional design. This cannot be easily dismissed, but aircraft design (and building) is a big commercial business, except for military and certain special needs, which are not addressed here. A commercial business defines itself by making profit while minimizing risk. This means that a commercial aircraft design will be very risk averse and venturing outside of the known design space where decades of experience exist is prohibitive due to the enormous financial investment needed and technological risk. A new conventional aircraft design requires large investments that dwarf pretty much any other industry, at least those with a limited market outside of the realm of consumer oriented businesses. A new unconventional, maybe even radically new, aircraft design would be an even larger investment and has a much larger risk of potential failure or at least not performing as well as expected. There have been efforts to reduce the size of the investment and the inherent risk, which can be addressed by government research funding and cross-fertilization from military projects. However, only so much can be done, and it is still the responsibility of the system integrator to bear the integration risk of a new design.

These background dynamics have, over the last several decades, lead commercial aircraft manufacturers down an increasingly narrow path of design choices. Initially a breadth of manufacturers existed in the 1950's and 1960's, that even offered at least marginal variations on the modern jet-liner such as T-tails, podded empennage engines, three engine configurations with integrated tail engines, etc. Basically all but two commercial aircraft manufacturers were forced out of business or into mergers with others. This was facilitated by increased investment needs and the fact that if just a single aircraft is less successful than expected, due to an economical disadvantage compared to competitors, reliability problems or other factors, it can be a heavy financial burden on a company in a competitive environment. This facilitated industry consolidation and an increasingly narrow design and concept space to allow consistent returns on investments at predictable risk levels.

1.1 Motivation

It is clear that the technological development in the commercial aircraft business has been forced down an increasingly narrow path of development. This warrants an investigation into the reasons behind this. First of all, there are a variety of factors influencing the choices of any aircraft designer. These factors range from technological to economic, from regulatory to perceived factors.

1.1.1 Technological Factors

First the technological factors need to be considered. A technological factor is comprised of constraints of physical and technological nature. Physical constraints comprise a set of physical rules that must be followed. This includes conservation of energy, mass and momentum, gravity, Newtonian physics, etc. Even though there are new developments in physics that tend to challenge the validity of these long known rules and could eventually invalidate them, which already happened to Newton's Laws of Physics. This possibility should not be of concern, because the chances of that happening are small and furthermore most likely only concern the very extremes of space and time scales with which we are not concerned with here.

Technological constraints consist of rules that arise from physical limitations, which however can be improved over time as new technologies are developed, albeit mostly quite slowly. The most important of these constraints are arguably material constraints, because they indirectly impose other physical limits through temperature and strength constraints. Material constraint means that the physical properties of any material selected to build any part of an aircraft out of are limited. This especially concerns weight, strength, and thermal properties and combinations thereof. These material constraints have far reaching consequences. They impact nearly all aspects of a vehicle. Either this occurs directly, such as in weight, strength, and durability of structural components that directly limit minimum weight, shape, and use of a specific material. Alternatively, this occurs indirectly such as thermal material limits of propulsion system components that limit thermodynamic cycle parameters, which then in turn affects propulsion sub-system efficiency, weight, and performance, which then translates into effects on the overall vehicle.

Technological constraints such as these have recently been studied extensively. This was made possible by the use of highly integrated computer codes that focus on physics based sizing and synthesis. Such an integrated computer code uses a geometric and operational definition of a plane and then sizes it so that the plane can satisfy the specified operational constraints. Unfortunately, traditionally such a monolithic sizing codes cannot provide the geometry. The geometry has to be specified beforehand and then can be used in the sizing process. This brings up another limitation of such a code, namely, the fact that it is heavily specialized into a very specific category of designs. Namely, FLOPS, as such an example, is limited to general subsonic transport jet aircraft, such as commercial passenger transports. There are a number of other design codes that focus on other areas such as general aviation and military combat aircraft. However, outside of these limited areas of focus no publicly documented code exists. More such codes probably exist in a non-public environment, but it is very likely that they are very specialized and are tailored towards specific needs.

Any given code can be used outside of the range of intended use, however this comes in hand with questionable results that at best suffer from reduced accuracy. Nonetheless, an opportunity to generate some results is better than not being able to analyze a specific problem at all, or having to invest extensive man-hours and money to create such a capability if that is possible at all. Still, any serious project with credible results needs to invest a substantial amount into basic research to reduce uncertainty. Such existing design codes are very limited in scope due to the number of historical data points, each of which represents a plane that exists or has existed and possesses well known properties. This experience and data is very often wrapped into such a design code to improve accuracy by a great amount. The drawback being, that in the absence of such data such as for a new and revolutionary design would be needed, accuracy is greatly reduced and therefore increases risk to design and build such a revolutionary design by a large amount. One example is that if one would try to design and build a supersonic civilian transport, there is very little data available. The only data point that can be used as a reference is the Concorde. Additionally, the Concorde was designed decades ago and there have been improvements in a variety of disciplines, especially materials, propulsion, and manufacturing technology.

It should be mentioned also, that the Concorde design suffered from the very same basic problems mentioned. [1] For example, there was large uncertainty about the performance of the aircraft, mostly stemming from the fact that purely based on the physics of the mission to be performed, small changes in component efficiencies resulted in rather large changes in available payload, range and economics of operations. While members of the project admit that a more thorough up front investigation also involving more wind tunnel testing would have helped to reduce changes necessary at a later stage in the project and probably helped to reduce the number of necessary prototypes and pre-production models, it definitely would still have been necessary to flight test and confirm the aerodynamics, performance, and handling qualities due to the groundbreaking new territory in terms of the flight regime.

Aside from these problems, such integrated design tools allowed, for the first time, a quick analysis of an airplane concept in the very early stages of conceptual design. Furthermore, in combination with advanced design techniques, such computer codes allow the exploration of design constraints. Even more importantly, it was possible to introduce new technologies into a design and study the effects of them onto a design and its constraints. It was further possible to optimize a set of new technologies into a selection of specific combinations to meet certain future challenges and therefore optimize research and development funding to best meet certain goals in the future.

1.1.2 Economical Factors

Economical constraints are a fairly recent addition to aircraft design. Until recently aircraft design was mostly about technological challenges, about bigger, better, higher, faster. This is especially true for military aircraft. Then after a given aircraft was finally built and finished, it was considered the job of the business side of an aircraft company to market and sell the aircraft with little or no interaction between engineers and business executives. Furthermore, the business and marketing divisions would not allow engineers to engage in cost analysis and marketing efforts. Over the years it became painfully apparent to certain companies that an aircraft design has to fit a certain market not just in capability, but also in acquisition cost and operational expenses.

An example of such a project is the Concorde. At the time of introduction it was arguably the most advanced commercial aircraft available. It could fly higher and faster than any other commercial jet. Even up to today there is no other commercial aircraft that can match its performance. However, the research and development expense was enormous. This was due to a variety of technical challenges that directly influenced the economics of the plane. To name just a few of them, the engine inlets of any supersonic aircraft are a challenge to design, especially for flight much above sonic conditions. This requires moving inlet ramps, so that the engine efficiency does not drop off significantly and the better the efficiency the more moving mechanical parts have to be included. Another challenge involves the aerodynamic inefficiency of a delta wing configuration at low speeds. This means that since the plane is optimized for high speed cruise, take-off and landing suffer greatly, which usually means very high speed take-off and landing speeds. Again, this results in added challenges in the landing gear and tire configuration that cause considerable extra expense.

Last but not least, it was necessary to include after-burning engines to achieve the thrust needed only for short periods during takeoff and moving from subsonic to supersonic flight without increasing the size and weight of the engines considerably. This, however, meant that a feature until that point in time only present in military engines, that are maintained very frequently and have low utilization, or flight hours per year, had to be redesigned to be significantly more reliable. Additionally, this, and a super sonic flight regime, also meant that much more complicated and heavy moveable nozzles had to be included. A number of these challenges required considerable engineering effort, which then directly translated into enormous research and development cost.

At the same time the aircraft was much less reliable than normal commercial transports, which meant much more frequent maintenance requirements. Even worse, the Concorde used a lot more fuel per passenger mile due to the bigger energy expenditure of flying faster and the decreased engine efficiency. This directly translated into higher operational cost than any other commercial aircraft, which meant that the Concorde was a commercial failure. Air France and British Airways only put a limited number of planes into service due to the fact that the French and British governments paying for the development and manufacturing expenses and giving the aircraft essentially for free to the airlines, by requiring a revenue sharing on the side of the airlines to help pay for acquisition and development cost. This revenue sharing scheme was dropped later on when both Air France and British Airways failed to produce any profits. Other airlines did not acquire any Concorde aircraft, because at the time of market introduction the fuel price had increased quite dramatically. This especially hurt the Concorde, because it uses a lot more fuel then conventional commercial transport aircraft, which greatly increases the sensitivity of the operating costs to fuel price fluctuations.

Due to the high operational cost of the aircraft, and a number of operational limitations such as range, the Concorde struggled for years to find a niche market, aside from novelty flights and air-shows that had the best potential of at least breaking even. This market turned to be between major European hubs Paris and London and New York. The target clientele was affluent passengers vacationing or business travelers that considered their time very valuable and therefore saved about half the flight time of any regular commercial flight between those cities. This is also the case for certain bank and stock exchange employees that were able to follow the stock market opening in Europe and then catch a Concorde flight to arrive just in time for the opening of the New York Stock Exchange and then be back for dinner in Europe. This, of course, had limited appeal especially considering the high ticket prices. However, in this niche the Concorde was able to barely break even on operational cost.

What contributed to the poor economic performance in this particular case was the fact that airlines tried to operate the Concorde similar to the other conventional transport aircraft they operated. For example, and airline operating a transatlantic passenger service would have flights leaving Europe around noon local time which then arrive mid-afternoon on East Coast destinations. The same aircraft then, after a short turnaround, would then leave for Europe again in the evening to arrive mid-morning back from where it started. These flight times fit well with the operational schedule and overall passenger demand. It also allows the aircraft to be in the air for a good portion of every day, which means high utilization which in turn lowers operating cost by spreading acquisition and a number of other costs over many more flight hours and therefore more passenger miles.

While the Concorde can cut the flight time from Europe to the American East Coast nearly in half, this also means that the aircraft is less in the air. This much lower utilization hurts the operational economics. In most cases, it would be possible to operate the Concorde in such a way that there are two two-way flights per day to increase utilization significantly. However, at least one of the flights would operate at very odd hours, which also would be subject to noise limitations at certain airports due to limits in operating hours for noisy aircraft.

Additionally, the Concorde was forced to fly subsonically over land legs of any trip. This meant that the Concorde would fly no faster than regular aircraft, but still use much more fuel while being able to carry fewer passengers. This shows that the inability to properly make use of the advanced abilities of the Concorde hurt its economic shortcomings even more.

1.1.3 Regulatory Factors

This leads to another set of important factors that influence the success of an aircraft. The aircraft business is a highly regulated environment that regulates a great number of things. These regulations range from very small things such as Federal Aviation Administration (FAA) regulated pens to very large items. All the regulations are split into one of several categories. These categories are in general safety and environmental impact. The official set of FAA regulations (FAR) are split into a significant number of detailed sections that cover all aspects of aeronautics. This includes not only regulations of the aircraft itself, but also regulations on operations, maintenance, and certification.

The Concorde was affected by these regulations because there are very similar regulations in Europe at the time (not yet unified into Joint Aviation Regulations, JAR) and from a quite early stage the primary design mission for the Concorde was transatlantic travel from Europe to the United States and back. Therefore, the Concorde would have to meet regulations in all countries it was to be operated in. The Concorde was affected by a number of issues and an apparent lack of regulations for commercial supersonic operations. Therefore, it was decided at the time of the Concorde program start in 1962 that the Concorde should meet or exceed regulations for subsonic transports at the level specified in 1962. As the program progressed it was obvious that engine noise and emissions could barely be met at the 1962 regulations level.

Since the development program started to run longer than anticipated, and the regulations for subsonic transports became stricter with improving technology, the program was presented with a significant problem because at the time there were no technologies that in actual flight reduced noise significantly. There were some technologies that yielded some improvements in static tests, but those improvements were all but gone in actual flight tests. Due to the Concorde's high sensitivity to weight, it was decided to drop the noise reducing equipment and instead go with the original 1962 targets. This issue then later on became a major problem when the airlines operating the Concorde sought permission for flight operations to New York and Washington.

Due to public opposition, this became a major topic, since the Concorde did not meet any current regulations. Therefore, the airlines sought a special permit that would still allow them to operate limited trial service. This meant that a special inquiry was started with public testimony that the engineers suddenly found themselves thrust into. In the end limited trial operations were allowed by virtue of the small number of flights and the fact that it was considered important to give new technologies a chance to prove themselves and later on improve on them. However, that never really took place. So Air France and British Airways operated over two decades on a limited trial operations special permit. This also lead to an update in the regulations, which now state that any supersonic commercial jet has to meet the same emissions and noise standards as regular subsonic jets.

The other big issue in case of regulations that affected the Concorde was the problem of the sonic boom. A sonic boom is caused by convergence of compression waves around the aircraft during supersonic flight. This convergence causes shock waves to form. These shock waves propagate outward away from the aircraft. Due to the relatively high energy contained in them and depending on a number of factors such as weather, these shock waves can impinge on the ground. These shock waves follow the aircraft around, essentially sweeping ground areas below the flight path. The rapid air pressure disturbance that these shock waves represent, results in a very audible "boom", hence the name sonic boom.

Early tests with military aircraft indicated that this presents a significant problem due to possible side effects in humans and animals, apart from the very high annoyance factor of repeated booms that could exist in high traffic areas. With this in mind, supersonic flight over land was prohibited, with the only exception being certain military operations. Although, supersonic military training flights are conducted mainly over water, this operational regulation led to all over land flights with the Concorde having to be subsonic. This meant that an aircraft optimized for cruise at roughly Mach 2 had to fly a large portion of its overall mission in a sub-optimal point of its overall flight envelope. At the same time it negated the primary benefit of the Concorde, speed. Therefore, the economically feasible range of missions was further reduced to transatlantic flights, because the Concorde's range is too short for the Pacific, and all other transoceanic routes simply do not have the volume of travelers, especially travelers willing to pay a premium for faster transportation.

This illustrates a very close connection of economical and regulatory factors. Just how easily a single, relatively simple regulations can have a dramatic impact on the economic fundamentals of an aircraft.

1.1.4 Perceived Factors

Finally, perceived factors also play a major role. Public perception of almost anything has direct and indirect effects on politicians and therefore public policy-making. This means that how something is perceived in public, in the media, or other avenues for public discourse has potentially significant effects on how laws and regulations are introduced or modified. Equally, public opinion can have significant influence on the economic success. This usually happens by generating "buzz", meaning that informed circles find a given product very worthwhile and then by showing it to others trigger an exponential growth in interest and most likely purchases. This can have a dramatic impact on the market growth of a new product and eventually result in revolutionary shifts in consumer culture in terms of what is the accepted standard. There are examples of this especially in the consumer electronics world. Examples range from video cassette recorders (VCR) to cell phones and a variety of other products. There can also be an inverse relation, meaning that economic success can influence public opinion and public policy making.

However, this can also work the other way and work against a product. If for whatever reason it is perceived as having some positive aspects but with negatives dominating and lacking significant economic success, a product can find itself on the opposing end of this exponential growth, but this time in negative opinion and publicity. This negative opinion can range over a variety of issues such as negative environmental impact, bad safety record, and a variety of other concerns. These objections in general can either be found in scientific data or actually more often than not, not founded in scientific data but rather in second hand opinions that are either completely irrational or based on contradicting evidence from scientific research that was not yet able to form a clearly understood opinion.

In the case of the Concorde, it found itself rather quickly on the negative side of the opinion and also policy making due to real concerns about the impact of noise during landing and especially take off, excessive fuel burn, and also introduction of pollutants directly into the ozone layer, and sonic boom concerns. When the airlines that wanted to operate the Concorde to and from the United States (US) applied for landing permits there were significant objections. Recent changes, at the time, in environmental laws also resulted in the Concorde being the first plane that was required to prepare and Environmental Impact Statement (EIS), which significantly complicated things further. This was especially the case because while there were rules and regulations for subsonic aircraft, none existed for supersonic aircraft.

Furthermore, public hearings also were held to explore the objections to the Concorde certification and operations in the US. During those hearings, Concorde engineers suddenly found themselves in a public platform where they had to defend their project under the eye of the public. This was something completely new and unfamiliar to these engineers.

At the time, several environmental groups significantly influenced public opinion against supersonic transports. The most significant of these groups was probably "Citizens against the Sonic Boom", which actually argued based on scientific objections against the development and operations of supersonic aircraft. They were apart from technical and engineering problems partly responsible for the cancellation of the supersonic transport project awarded to Boeing, also known as 2707. That project was cancelled in the early 1970s after almost half of the program cost of the

Concorde had been spent without construction of any flight hardware. However, this added to the negative public opinion about the Concorde since it was now perceived as a foreign superior technology that could significantly affect the domestic economy and competitiveness.

The result of all this was that supersonic flight over land was banned. Supersonic aircraft starting in 1976 had to meet the same rules and regulations as subsonic aircraft. The existing 16 Concordes were grandfathered in and received Federal Aviation Administration (FAA) approval with minor changes. And finally, the Concorde was granted limited trial operations permission to New York and Washington D.C..

1.1.5 Interconnectedness

As shown here with the example of the Concorde, technical, economical, regulatory, and perceived factors play a major role in the successful fruition of any modern engineering project. This example also shows the interconnectedness of all these factors. Technical challenges will also lead to economical challenges and increased cost. In return, lack of economic success can make overcoming technological challenges much harder due to higher risk in recovering any research and development expenditures.

This shows that there is a clear need to better understand these connections with the goal being an increase in successful engineering projects and to facilitate technological improvement, especially where technology has become locked into certain narrow development paths that were or still are close to optimal but will definitely not be able to deliver in the future or potentially result in a number of problems in the future.

CHAPTER 2

BACKGROUND

With this need for a highly integrated analysis and evaluation very early in the design process to ensure a successful programmatic outcome, it is clear that a look at recent efforts in these areas is warranted. However, current methods are limited in their analysis to specifically address a set of requirements or desired features that very often are only defined qualitatively if at all and are not well understood. This necessity has lead to efforts to integrate methods that further define and expand the understanding of how an aerospace system interacts with a number of complex external systems and is therefore inherently connected to its ultimate success or failure.

2.1 Modern System Design Methods

The primary purpose of modern system design methods is to allow a much greater understanding of a proposed system in the very early stages of a program. This has its roots in Concurrent Engineering (CE) [2] as well as Integrated Product Development (IPD). CE is the idea that other factors outside of pure performance are taken into account in the initial conceptual phases of product design. These factors can range from manufacturability to cost, quality, and maintainability. CE and IPD together with additional concepts form the foundation for Integrated Product and Process Development (IPPD) [3].

2.1.1 Integrated Product and Process Development

Recently, this ongoing effort tried to push knowledge forward in the design process of complex systems. This increased level of information in the early stages of the design sequence through the use of IPPD techniques affords decision makers and designers greater flexibility in choosing the most affordable design and therefore increase the likelihood of a positive programmatic outcome [4, 5, 6].

The fundamental goal of this concept is that by attempting to bring forward information in the design sequence, the most affordable design can be chosen, and the requisite changes made before costs are locked in [7]. The IPPD method and the modified form thereof developed at Georgia Tech allows the engineer and program manager to decompose both the product and process design trade iterations [8]. This implementation of the IPPD methodology allows the engineer to more easily investigate the effect of the uncertainty associated with the design, certification, manufacturing, and operational aspects of the complete life cycle of a complex aerospace system. This uncertainty, together with the highly specific and therefore highly sensitive nature of many optimized designs, which results in a high risk of programmatic failure even if only small changes should become necessary. The result of a number of program failures mainly caused by the non robust nature of a system lead to the desire to produce a methodology that can ensure the robustness of the final system to uncertainty not only during the requirements definition but the whole system life cycle. Robust Design Simulation (RDS) was therefore developed at the Aerospace Systems Design Laboratory (ASDL) as a means to address this specific problem.

2.1.2 Robust Design Simulation

The initial RDS techniques were developed and implemented in ASDL in the early 1990s. The key purpose of the RDS method is to ensure that the final system will meet its goals and satisfy the customer. The best way of achieving this goal "is to deliver a product that performs well not only in the environment for which it was designed, but in all environments." [9] The essential elements of the RDS system are shown in Figure 1.

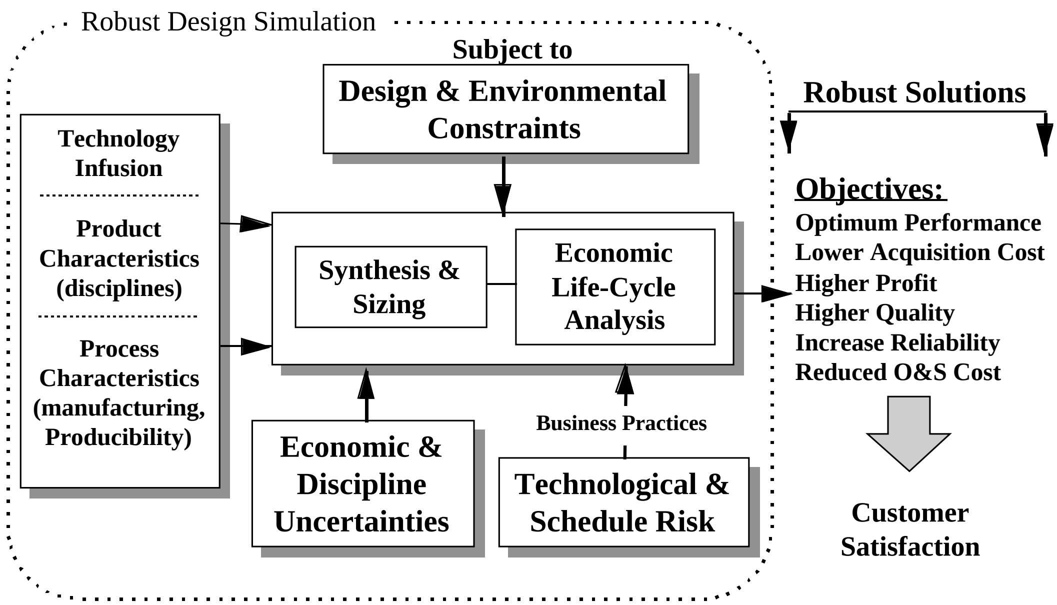


Figure 1: IPPD through Robust Design Simulation [9]

The RDS methods allows the designer to identify 'key product and process characteristics as well as their relative contributions to the chosen evaluation criterion in the presence of risk and uncertainty [9]." A noted example of such risk and uncertainty lies in the effect noise factors have on an aircraft. This is exacerbated in this example by the fact that the cost of fuel is extremely important for an aircraft that consumes a large amount of fuel such as the high speed civil transport (HSCT) studied. It if further shown that in this case reducing the sensitivity of the design to the cost of fuel can be achieved by reducing the consumption of fuel by the aircraft. The issue with that, however, lies in the fact that the way of reducing fuel consumption is by technology development, which adds another form of risk back into the design.

RDS therefore introduces a probabilistic treatment of the noise factors that allow assessment of the feasibility and viability of the design. This treatment is enabled by a design space exploration that concurrently incorporates both performance and economic requirements. This explorations yields insight into which requirements will be limiting the viability or feasibility of a given design ahead of time. Additionally, the amount of necessary improvement will be known too. The probabilistic treatment has generally accepted ten thousand monte carlo runs as sufficient to be able to identify the general shape and location of the resulting distribution. However, this still can represents a significant computational burden especially using full design codes. Therefore, the monte carlo simulation is usually performed on a surrogate model of the desired design. This significantly reduces the computational burden albeit at the expense of accuracy.

2.1.3 Virtual Stochastic Life-Cycle Design Environment

In the same light of trying to make more educated decision that are necessary in the early stages of a design, a Virtual Stochastic Life-Cycle Design (VSLCD) Environment was developed [10, 11]. This environment is designed to comprehensively treat the whole life cycle of a system from design to disposal. This also enables the treatment of variability of a design in time shown in Figure 2

The VSLCD Environment is shown in Figure 3. This shows that RDS is one of the core elements of VSLCD as it provides that foundation for the analysis of the feasibility and viability through the evaluation of uncertainty in a design. This also represents a shift away from design for performance to design for affordability.

The elements required for VSLCD and therefore RDS are shown in further detail below.

2.1.4 Design Space Exploration

Design space exploration is a method of exploring a range of variables and choices

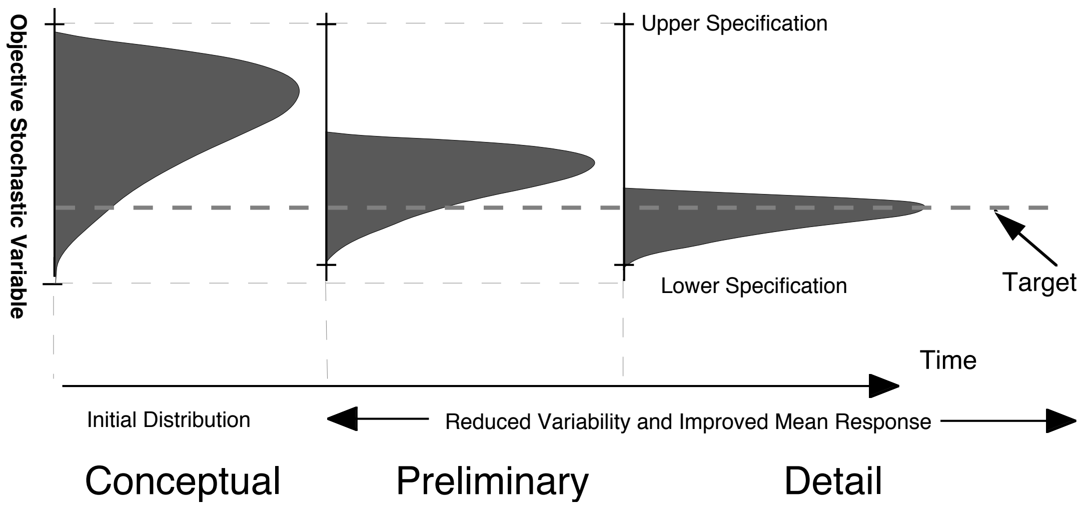


Figure 2: Variability of Design in Time [10]

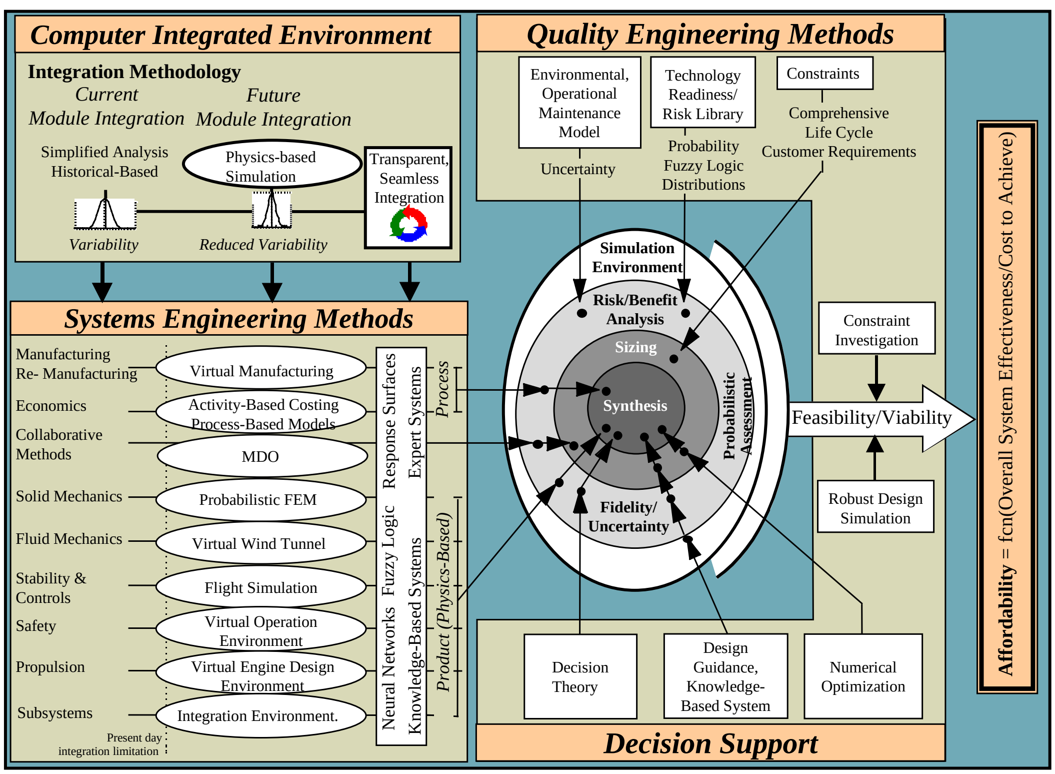


Figure 3: Virtual Stochastic Life-Cycle Design Environment [10]

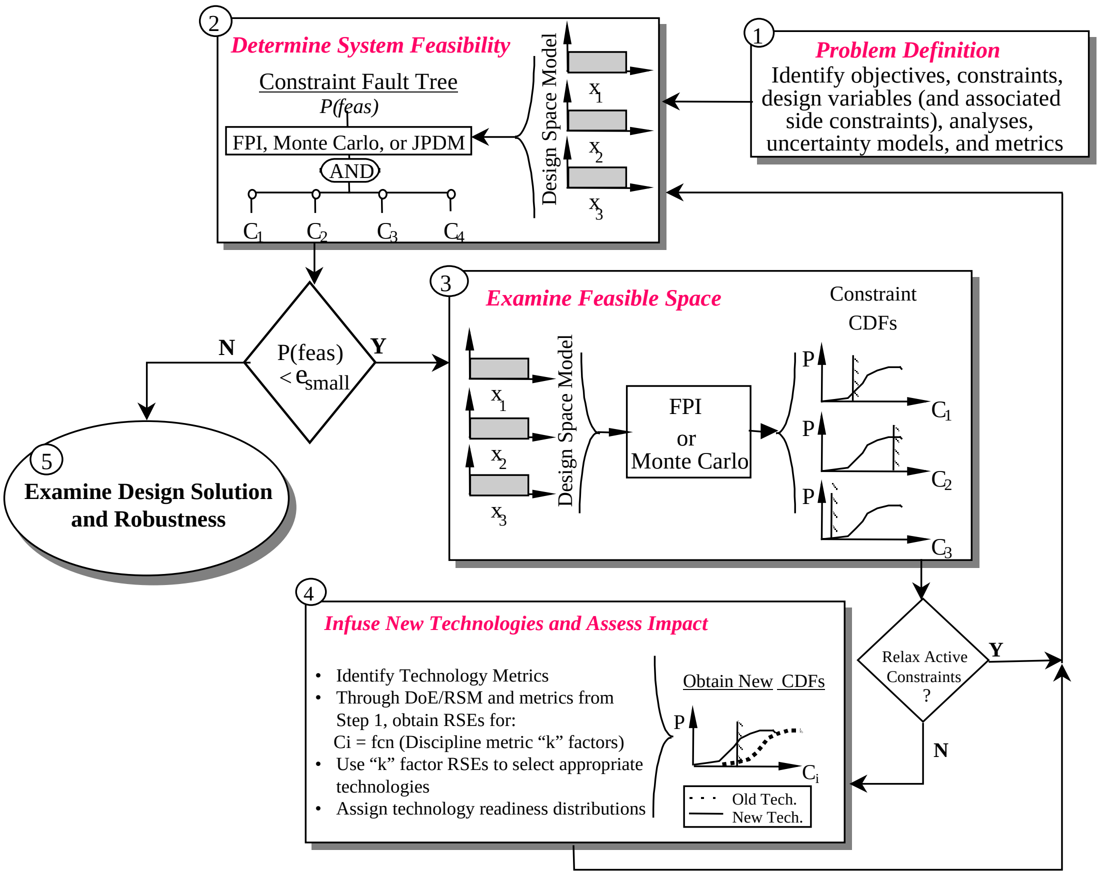


Figure 4: Feasibility and Viability [12]

for a particular system concept. The variables in question can be performance, regulatory, and affordability related. This process involves the systematic evaluation of feasibility and viability of the available design choices. Therefore, a process for evaluating feasibility and viability is required. Such a process is shown in Figure 4. It should be noted that probabilistic methods such as Monte-Carlo, Fast Probability Integration (FPI), or others are required. Monte-Carlo analysis is the computationally most expensive methods. Therefore, it is commonplace to alleviate this by reducing the full models with surrogate models to speed analysis time.

2.1.5 Surrogate modeling

Surrogate modeling in general refers to the practice of building models of models. This comes from the general trend in science to construct ever more complex models trying to more accurately describe the empirical world. However, this increased complexity comes with a price. In general it means that such highly complex models have a very high dimensionality in both inputs and outputs and they require significant effort for execution. Therefore, surrogate modeling offers a methodology to create simplified models — often simple equations — that still adequately represent the more complex model. There exists also a whole field around assessing which parameters are the most important to the variability of the outputs, how to extract the maximum amount of information out of a complex model with a minimum of effort. For a more detailed overview of the subject, the reader is referred to various large volumes [13, 14, 15].

2.1.5.1 Design of Experiments

To extract a maximum of information out of a complex model with a minimum of effort, one must think about the number of variables influencing the model. The number of available variables directly influences the number of required analysis runs to construct a surrogate model of sufficient accuracy. Since the model in question is very likely to require significant computational effort and therefore it is in the best interest to minimize the number of required runs or — more precisely — the scaling of the required runs with the number of available variables.

This is at the core of the effort to “design” experiments in such a fashion that it is possible to create surrogate models from ever more complex models thus allowing greater insight and forecasting power in more complex subject matters.

The simplest set of experiments that can be performed is to simply examine every possible variation of extreme — high and low — settings of all variables and the mid-points. This is commonly referred to as a three level full factorial design and represents one of the least efficient ways to conduct experiments and can become too expensive to carry out with more than a small number of variables. However, it also represents the most accurate method of creating a second order polynomial regression model. To reduce the effort required it is common to leave out some of the mid-point experiments or the corners, which results in a much more favorable scaling with number of available variables. This, however, comes at the price of accuracy and can lead to the confounding of some of the variables. If some information about the relation of variables is available a priori it can be used to arrange the experiments in such a way to avoid confounding and the resulting model representation error. A number of experimental designs are reviewed by Montgomery[16]. It is also possible to utilize non orthogonal designs which are reviewed by Barros[17]. There are also commercial software packages, such as JMP[18], available that help in the automation of the entire surrogate modeling process and also allow the creation of custom designs for very large numbers of design parameters.

It is also important to note that it can be helpful to conduct an initial screening test that simply tests the amount of variability contributed by design variables at their extremes. This can then be used to remove variables from the process to reduce the number of design variables used in the final surrogate model. This can help tremendously, but in some cases where there are many non-linear effects present in the model, it is possible that very important variables are eliminated wrongly. Therefore, any surrogate model has to be tested against a set of space filling points to evaluate the model representation error.

2.1.5.2 Types of Surrogate models

The kind of surrogate model discussed up to now is a linear regression model, commonly referred to as Response Surface Equation (RSE) and the process of developing such models is known as Response Surface Methodology (RSM). This means that a number of points in the model variable space are evaluated and then the coefficients of a prescribed linear equation are calculated by solving the resulting set of linear equations. The type of linear equation is usually limited to a second order polynomial with certain number of interaction terms. This type of model is simple and easy to use and understand. However, it does generally not capture non-linear systems well. This does not mean, however, that it can not be used for non-linear systems. In such cases the applicability of such linear models simply has to be limited to smaller variable ranges such that the resulting variable space more closely approximates the behavior of the assumed polynomials.

Aside from linear statistical regression models there exists also a broad range of non-linear models. A Kriging model is based on global non-linear extensions to a local linear surrogate model[19, 20]. This extends the potential accuracy of the surrogate model. However, the particular nature of this type of model makes it not applicable to certain types of models.

Another type of surrogate model is a neural network. The architecture of a neural network is modeled after neuron behavior and connectivity in brain and other neurological tissues. The models evolved from the idea that artificial intelligence could be achieved by creating a detail model of a brain. Attempts in these direction were not very successful. However, neural networks have since evolved into a class of non-linear Bayesian modeling technique. Fitting the models to data is still achieved by an iterative "training" process[19, 21, 22].

A related type of surrogate model is the Gaussian process[23, 24, 25], which is a more general form of a Kriging model and a variation of a neural network. In a Gaussian process data is treated like a normal distribution and predictions are achieved through a covariance model. However, with very large data sets Gaussian process models become very computationally expensive and therefore have limited use with very large data sets.

Fortunately, the use of neural networks has recently been simplified by a great deal through the availability of tools such as JMP[18] and the Basic Regression Analysis for Integrated Neural Networks (BRAINN)[26]. They provide an integrated and automated method of generating neural network surrogate models with a large number of options available to the user. Furthermore, they allow easy validation and testing of the created surrogate models.

2.1.5.3 Surrogate Model Validation

This brings up the point of how surrogate models can be validated. In general error in surrogate models can be classified into three categories. They are the Model Fit Error (MFE), the Model Representation Error (MRE), and other random error. Both MFE and MRE are due to surrogate modeling process. The random error comes from the noise in the individual observations of each experiment. The subject model here is a computer model and therefore there is no noise inherent in the observation process — at least outside of models with random effects. Therefore, the random error can be assumed to be zero here.

The MFE directly refers to the error in the surrogate model as compared to the observed data. This means that it represents the quality of how well the surrogate model fits the data used to create it. This error should be very small otherwise the surrogate model will be not a very good representation of the known data points. This can occur due to a variety of reasons, but most often is due to errors in the surrogate modeling process or due to confounded variables, meaning that the type of surrogate model selected is inappropriate to fit the data.

On the other hand MRE directly describes the error of the surrogate model as compared to the original model. This means that it describes how well the surrogate model fits the model it is based on. Ideally this should be very low also. However, it can be quite difficult to assess this type of error, especially if the original system model is not well known and difficult to generate data from due to run time. Furthermore, testing MRE depends on the selection of the points used to test the surrogate model. This test set has to be independent of the data set used to create the model in the first place. It is generally agreed upon that the best way of assessing MRE is to use a random data set that is as space spanning as possible. If the surrogate modeling method already depends on random space spanning data it is possible to simply use a larger run and the create the model based on a subset thereof and use the remainder of the data for surrogate model testing and validation to assess both MFE and MRE. Normally, all of these errors are expressed in terms of the R^2 value, which should be as close to one as possible.

2.1.5.4 Uses of Surrogate Modeling

Outside of replacing complex analysis tools in design space exploration, surrogate models can be used for a variety of other uses. For example they can be used to extend the fidelity of a lower fidelity analysis, enable visualization environments, such as JMPs[18] contour profiler environment, and allow inverted prediction of parameters using the system responses. This interactive visualization is a key feature in a number of the techniques described above. A further feature is the ability to represent very complex models with relatively simple easy to understand algebraic equations. This is of tremendous help in understanding the system in question and allows the identification of key trends and features. Furthermore, surrogate models enable the masking of the underlying method of analysis. This means that in today's highly interconnected world, where it is essential to interface with many other partners on a given project, they key intellectual property can be protected through the use of surrogate models.

2.1.6 Technology Identification Evaluation and Selection (TIES)

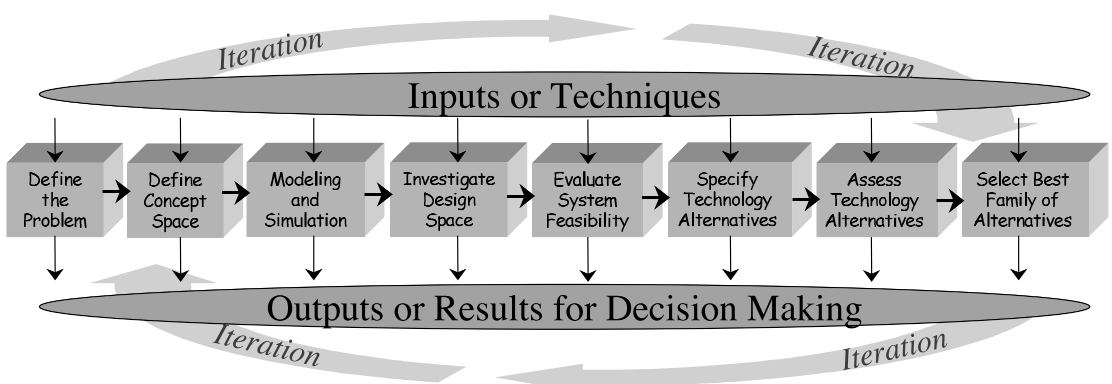


Figure 5: TIES Technical Approach [27]

The process known as TIES was developed at Georgia Tech[6] by attempting to help decision makers understand the effect of technologies — or more importantly a mix of technologies — and their effects on a specific system. Furthermore, TIES includes a structured process that allows decision makers to not only understand which technologies are most appropriate to expand the feasible and viable space of the system[28, 29, 30], but also the uncertainty stemming from the risk associated with the development of new technologies. The process also enables a gap analysis that allows the user to evaluate the required technology improvement to achieve a prescribed target system performance. A overview of the entire process is shown in Figure 5.

The basis for the successful use of TIES is the identification of technologies and their impact on the performance of subsystems. It is therefore important to identify key scaling parameters in the system model that sufficiently capture the impact of subsystem performance parameters and enable the scaling of these impacts onto the overall system performance. Furthermore, it is important to identify the specific impacts of each of the technologies on each of these scaling parameters or "kappa"-factors and the respective compatibility of each of the technologies with others considered in a technology portfolio.

These technology impacts can be assessed with methods like Technology Impact Forecasting (TIF)[32, 33]. This significantly enhances the usability of the process. Furthermore, the limitations imposed by parts of the methodology that limit the number of technologies that can be considered at the same time have also been overcome through the use of genetic algorithms[34]. Both techniques have been successfully integrated into an environment for VSLCD. The combined process is shown in Figure 6.

2.1.7 The Unified Tradeoff Environment

The additional freedom afforded by methods such as TIES made it necessary to be able to investigate the effect of requirements on the design of the system. One possible solution proposed to this problem has been the development of a Unified

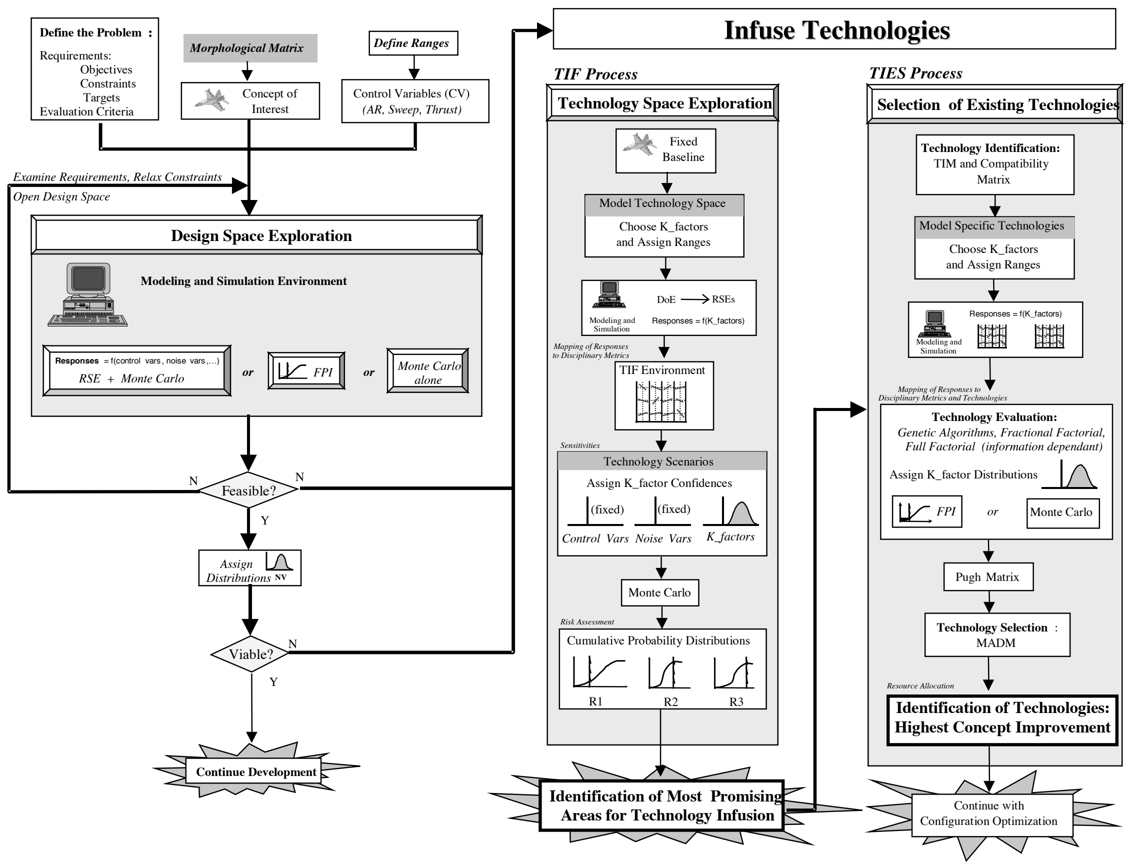


Figure 6: Technology Assessment Process [31]

Tradeoff Environment (UTE). In the original application it was shown how mission requirements, vehicle attributes, and technologies can be combined into one holistic environment[35]. It has since been applied to the US Army's Future Transport Rotorcraft and the development of the F/A18-E/F [36, 37, 38, 39]

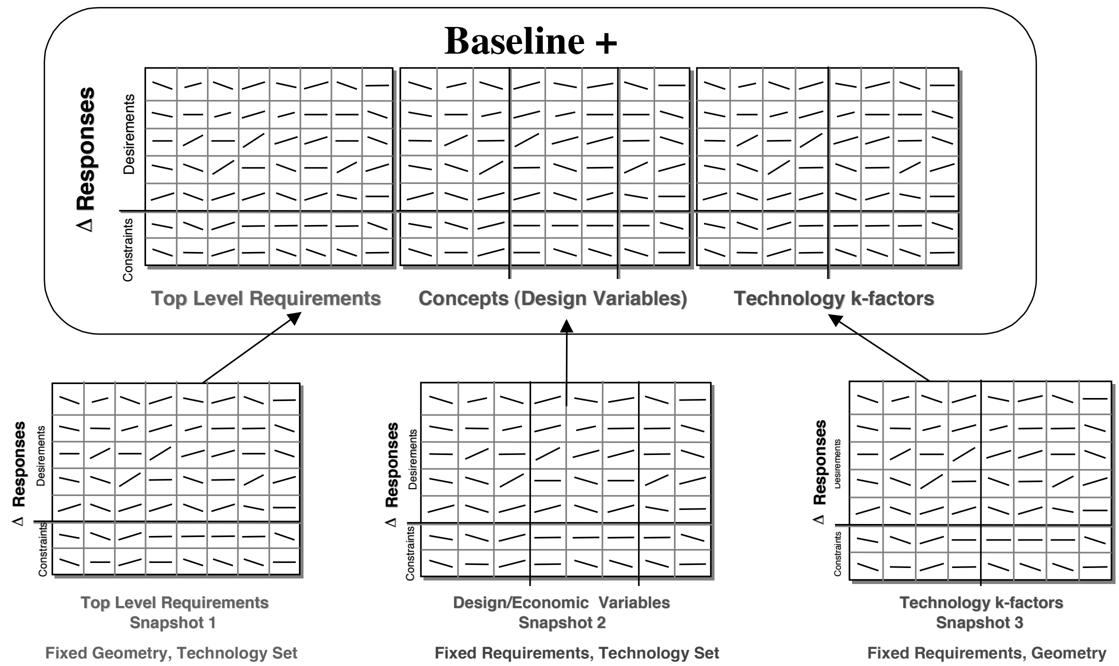


Figure 7: Notional Unified Tradeoff Environment [37]

Figures 7 shows a notional implementation of this environment. It should be noted that it is an application of surrogate modeling into a visualization environment that shows the partial derivatives of the outputs of a system model with respect to all of the system inputs simultaneously. What furthermore distinguishes the UTE is the ability to pull together variables from various types of analyses to in the end represent them as one unified model while allowing interactive exploration of the entire unified environment.

2.2 Aircraft Market

Over the course of the last 30 years Airbus Industrie has captured significant market share in a market that in the 1970s was mainly dominated by US manufacturers. It should be noted that other manufacturers have existed and still do. However, they either compete in the regional aircraft market such as Embraer and British Aerospace, or they are limited to specific national markets such as the Russian manufacturers Tupolev and Ilyushin that so far had little success in the global large civil aircraft (LCA) market. The large commercial aircraft market is economically and strategically important for both Europe and the US [40, 41]. It is therefore helpful to examine the attributes or features of this market.

2.2.1 Market Features

Producers of aircraft have to take into account a number of factors when embarking on the venture of an aircraft program. Generally, companies that can respond rapidly to changes have a competitive advantage. Such an advantage stems from market appeal determined by purchase price, operating cost, commonality with other aircraft types, worldwide support, and meeting international certification standards[41].

Before embarking on creating an aircraft market model it is also prudent to examine previous modeling attempts.

2.2.2 Market Models

In light of the strategic importance of the LCA market several modeling attempts have been made. Especially the US Government has made attempts at this through the US International Trade Commission and the National Aeronautics and Space Administration (NASA) [42, 43]. The model used by the study performed by the US International Trade Commission uses a relatively simple linear logit model based on widely used market share models[44, 45]. According to these sources there are three types of market share models. They follow one of these forms:

$$S_{it} = \alpha_i + \sum_{k=1}^{K} \beta_k X_{kit} + e_{it} \tag{1}$$

$$S_{it} = e^{\alpha_i} \prod_{k=1}^{K} (X_{kit})^{\beta_k} e_{it} \tag{2}$$

$$S_{it} = \frac{e^{\alpha_i} \prod_{k=1}^{K} (X_{kit})^{\beta_k} e_{it}}{\sum_{i=1}^{I} \left[e^{\alpha_i} \prod_{k=1}^{K} (X_{kit})^{\beta_k} e_{it} \right]} \tag{3}$$

where:

t is the time period t

i is the the producer i, $i = 1, 2, ..., I$

K is the total number of predictor variables

S_{it} is the market share of producer i

α_i is the constant term of producer i

β_k is the coefficient for the kth predictor variable

X_{kit} is the kth predictor variable for producer i in period t

e_{it} is the error term for producer i in period t

The linear model, Eq. (1), and the multiplicative model, Eq. (2), specify the market share as a linear or multiplicative function of the predictor variables respectively. The NASA model uses the multiplicative model for its market share prediction. Both models tend to have problems in matching the model to the physical realities of market share. Namely, that the range of market share must be in the interval $[0, 1]$ and the sum of all producers must be equal to 1.0. There are some modifications possible[46, 47] but both types of models do not inherently guarantee these logical consistencies. The third type – shown in Equation (3) – however, does guarantee this consistency, but presents a very difficult problem solving for the coefficients required. A more recent attempt at a modified version of this has been applied to the LCA market[48]. This attempt utilizes a modified model that takes the existing duopoly in the LCA market into account as well as introducing terms for time trend components and general autoregressive distributed lag indicated by previous studies[49, 43]. The final model that was presented takes the form:

$$lnS^*_{US,t} = \alpha^*_{US} + \sum_{k=1}^{K} \beta_k X^*_{kit} + lnS^*_{US,t-1} + \lambda T + e^*_{US,t} \tag{4}$$

where:

$\ln S^*_{US,t} = 0.5 \ln \left(\frac{\ln S_{US,t}}{1 - \ln S_{US,t}} \right)$

$S^*_{US,t} = \frac{S_{US,t}}{\tilde{S}_t}$

$\alpha^*_{US} = \alpha_{US} - \bar{\alpha}$

$e^*_{US,t} = \ln \left(\frac{e_{USt}}{\tilde{e}_t} \right)$

$\tilde{S}_t$ is the geometric mean of S_{it} $\forall i$

$\bar{\alpha}$ is the arithmetic mean of α_i $\forall i$

$\tilde{e}_t$ is the geomteric mean of e_i $\forall i$

λ is the coefficient for the time trend

T is the index for time

With the only two remaining producers — US and Europe — this model does satisfy the requirement of logical consistency. The result of this study were R^2 in the range of $58 - 75\%$. This result is much improved over the results obtained in the linear model study where the R^2 values were in the range of $13 - 61\%$. This is also better than the NASA study using the multiplicative model which was able to achieve an R^2 value of 47%.

Therefore, it is now time to take a closer look at the underlying data and variables used in this latest modeling effort. This study does try to model the overall LCA market. However, it does recognize the differences in the market over the radically different sizes in aircraft. Therefore, the aircraft were differentiated into different size classes as shown in Table 1.

Table 1: List of Aircraft type classifications[50]

Aircraft Type	Seating Capacity
1	< 50
2	50-69
3	70-90
4	91-120
5	121-170
6	171-240
7	241-350
8	> 350

The study however only focuses on the market share of aircraft types 5, 6, and 7, since the two large competitors, Boeing and Airbus Industrie generally do not offer types 1 through 4. Additionally, a number of smaller companies offer regional jets or turbo props of these size classes. Therefore, those classes are excluded since the market does not follow the duopoly assumption and the aircraft types competing in these lower seat classes are more disparate on a technical level.

The market share data shown in Figure 8 for type 7 aircraft was also used in this study. It should be noted that since aircraft sales can vary dramatically year to year due to the nature of the aircraft market, the data was smoothed with a two-period-based centered moving average method[51]. This data shows that Airbus Industrie did effectively break into a market dominated by US aircraft manufacturers.

Boeing serves this segment of the market with a range of 767 variants. Some

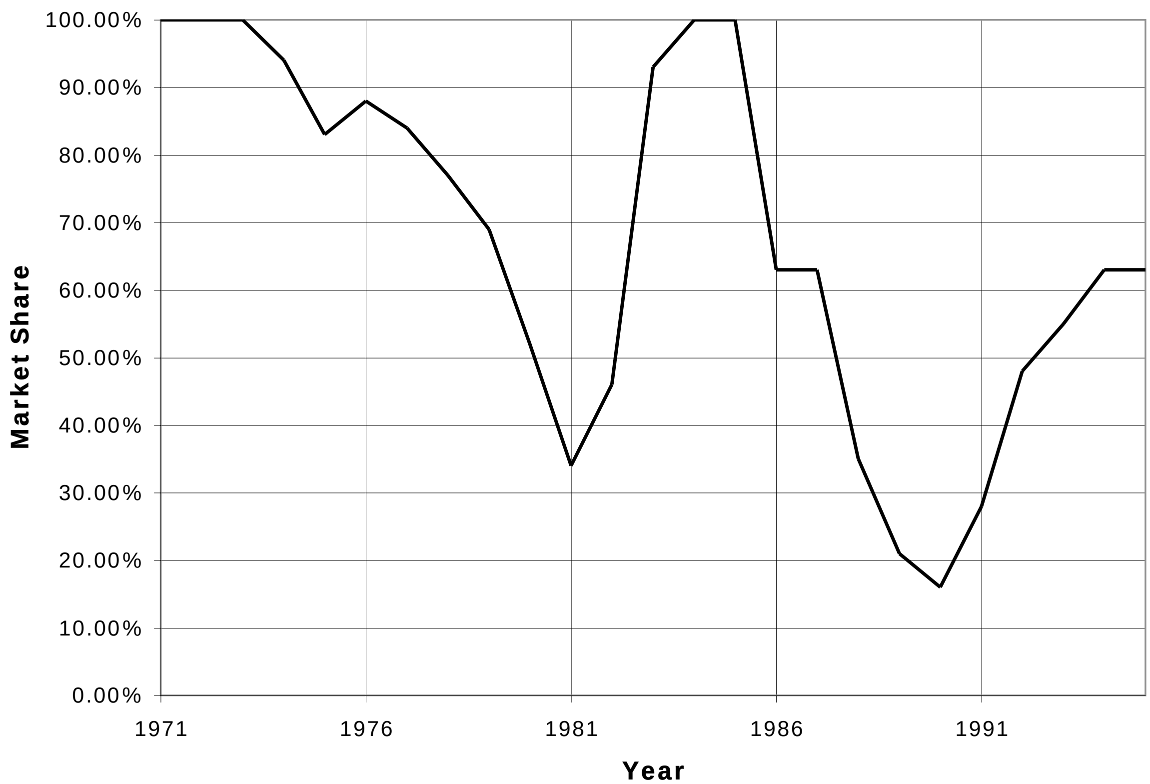


Figure 8: Worldwide Market Share of US Aircraft Manufacturers for 241 to 350 Seat Aircraft. Data Source: [49]

Boeing 757-300s also fit into this size category, depending on the seating layout, but it should be noted that they are no longer produced. At the upper end of this size class some three class seating layouts of the smaller 777 variants do also fit into this class.

Airbus offers two variants of the A330. The A340 in its variations does fit more at the upper end of this size class. Furthermore, currently two new aircraft that also fit into this class are in development by Boeing and Airbus, namely the 787 and the A350. Both are currently in development, with the 787 scheduled to be on the market first in 2008. The A350 is scheduled to be introduced to the market in mid 2012, which represents approximately a four-year delay behind Boeings market introduction.

Older aircraft that fit into this size class include the Lockheed L-1011 and the McDonnell Douglas DC-10. However, the size classes as defined seem to include significant differentiation at the lower end, whereas the large sizes are less differentiated and include multiple aircraft types and market niches in a single class.

The explanatory variables considered in this study are divided into several categories. Most prominently the categories are defined as endogenous and exogenous variables that reflect that manufacturer's capability of influencing the variable or not, respectively. The first exogenous variable used is the exchange rate of dollars to ECU, which was the name of the European currency unit before the introduction of the Euro. The second exogenous variable is the global fuel price. The endogenous variables included are the acquisition cost represented by the purchase price divided by the product of range and seat capacity and the operating cost represented by the ASM divided by fuel capacity. The study cites several sources[52, 43, 49] as source of the data regarding both variables. Additional endogenous variables consist of "dummy" variables that account for market introduction dates. This is used to represent the first mover advantage[41]. Furthermore, the model also uses special control variables to account for the first year that the US manufacturer market share dropped below 1.0 and the autocorrelation of the dependent variables cause by time lag in the market share response. An additional four variables allow the modeling of four world regions in addition to the worldwide market share.

The sole dependent variable is the market share. It is accounted for as world wide market share as well as market share in one of four regions, namely, the United States, Europe, Asia-Pacific, and Rest-of-the-World similarly based on Boeing's Market Outlook[50] as the seat classes. The market share is based on a dollar order basis. This is in contrast to previous market models that used unit orders or deliveries.

The results on this study were that the exchange rate was not a significant factor and as suggested potentially due to currency hedging and long term financing arrangements. The market differentiation across the different regions was also not significant. However, there were significant impacts from a new model launch as well as auto-correlated lag in the market share. The most significant variables were the purchase price, operating cost, and fuel price. Overall, R^2 values were between 58% and 75%, which is not exceptional, especially when using this model for predictions into the future.

Some shortcomings in this model include the use of extra "dummy" variables to account for such effects as the commonality of aircraft and the first mover advantage. The model lacks the inherent structure to account for these effects. Furthermore, due to extreme noise in the underlying order data, it had to be significantly smoothed to even allow the creation of a model. Also, the selected choices for representing the cost variables was not what airlines actually use, but rather artificial variables that may not represent what airlines actually use for their purchasing decisions. Furthermore, data for the cost was taken from a number of sources that tend to only list official list prices and not the actual prices paid.

2.3 Complexity Science

In order to understand the structure of such a market better it is helpful to first examine a branch of science that emerged in the mid 1990s called "Complexity Science". The goal of this branch of science is " the elucidation of a new law, or set of principles, or unified theory, or something that will make it possible to understand and predict the behavior of a wide variety of seemingly dissimilar complex systems" [53]. This now begs the question what a "complex system" is.

2.4 Foundation of System Dynamics

2.4.1 Industrial Dynamics

One method that emerged in the 1960s and 70s to tackle the rising concern about unmanageable complexities in real existing systems and processes was to try to apply control system theory to them. This eventually was then termed "Industrial Dynamics". The system in the name originally referred to a industrial production and distribution system [54]. This was the first effort to model the dynamics of industrial system, hence "Industrial Dynamics". Specifically, this effort pertains mostly to the overall idea of modeling common industry systems. This ranges from supply chain systems to organizational structures. Industrial Dynamics also includes Forrester's first system dynamics publication concerning the role of advertising in industrial dynamics from 1959 [55].

The foundations of these industrial dynamics models are specifically mentioned. The first foundation is information-feedback control theory, which is as previously mentioned the use of control theory that up to that point mostly was used in aiding the design and understanding of engineering systems such as temperature regulators, hydraulic systems, and electrical systems. This control theory was now applied to industrial systems that not necessarily represented real physical systems, but rather represented virtual elements. However, this does not have to be exclusive, it is still possible and sometimes necessary to model the flow of real physical items, such as materials, money, and goods. They, however, do no longer represent primarily the flow of electrical currents and hydraulic fluids.

The second foundation is the modeling of underlying decision-making processes. This means that an industrial dynamics model strives to capture an industrial system in such a way that it includes any relevant decision-making processes. Forrester [54] specifically concentrates on supply chain decisions and the resulting oscillations caused by purchasing decisions that are purely reactive to time delayed material shortages or surpluses caused by the same purchasing decisions. That is the oscillations in the system are purely a cause of the decision making process that is not forward-looking but purely reactive. This represents a great opportunity for building industrial dynamics models that allow decision makers to explore the underlying causes of undesired outcomes of decisions and at the same time modify the decision-making process in such a way that undesired outcomes are avoided as much as possible.

The third foundation is the experimental approach to system analysis. Industrial dynamics tries to take the underlying concepts and make them easily accessible. This is primarily done by representing each element of a system model visually. This facilitates the overall understanding of the connectedness of all elements and their influence on each other. However, since the visual representations used by Forrester [54, 56, 57] are different than the current standard for visual representation, the exact details of the visual modeling are discussed later in section 2.4.4 while the differences are detailed here. Specifically, Forrester used slightly different symbols for the basic elements of a system model. The biggest differences, however, are that he used different line styles to denote the flow of different items such as money or goods whereas the current notation in use does not use differing line styles, and that he did not enforce a minimization of line crossings in the models he presented. The minimization of line crossings is particularly important to facilitate the ease of understanding of each model. The more connection lines cross the harder to read and understand a model is. Additionally, these models support direct experimentation, which is directly facilitated by the final foundation of industrial dynamics, digital computer simulation.

The fourth and final foundation is the use of digital computer simulation. This means that Forrester was the first to make extensive use of computer technology to simulate the system models created. This is important because it allows simulation the system model at hand with sets of differential equations involving key processes and parameters numerically. The comparative speed of calculations is fundamental to be able to experimentally explore system models. In the initial implementation Forrester [54] actually describes the development of a system dynamics compiler called DYNAMO. This compiler was key to being able to specify system dynamics models as sets of equations. However, computer technology at the time was still in its early phases. So the equations had to be specified on sets of punch cards and the output graphs were printouts based on ASCII text symbols. Additionally at the time numerical techniques for solving algebraic or differential equations numerically were not well developed. Forrester spends a good amount of effort at detailing DYNAMO's method of solving the equations of particular system models. He essentially uses a Euler Forward Method [58] of the form

$$y_{n+1} = y_n + hf(x_n, y_n) \tag{5}$$

This solution, however, is only of order $O(h)$ with the remaining term forming an error of $O(h^2)$. Therefore, the solution is very dependent on the size of the increment h. If the increment chosen for a particular solution is too large, the set of solution points will be of poor resolution and additionally can exhibit oscillations or divergence not found in the actual solution. With the results of a simulation being relatively rapidly attainable, the quick turnaround time essentially allows repeated experimentation with the system model. This experimentation then can yield insights into the accuracy and stability of the system model. Furthermore, a number of scenarios, each with specific setting of external parameters or deliberate policy choices, can be simulated in rapid succession. This enables a learning process that yields insights into the overall behavior of the system model and therefore the system if the model is sufficient and has been calibrated.

2.4.2 Application to Social Systems

In the years following the publication of "Industrial Dynamics" [54] Forrester then proceeded to generalize the underlying theory. This then culminated in the publication of "Principles of Systems" [56] where he for the first time introduces the concept of rates and level. Rates essentially represent the rate of change affecting accumulations, that he terms levels. In other words this simply represents the formulation of a fundamental conservation law of a specific system variable. This law then can be used to write equations for all involved variables as necessary to formulate a complete set of equations needed for the solution of the entire system. Forrester also goes on to show how the underlying model structure directly determines the behavior. This is shown for an inventory model that directly links delays in the system to production cycles of the system. This means that he now differentiates between the system structure and behavior. This is also further detailed by Meadows [59] for commodity supply and demand with explicit production capacity, delays, prices, markets with application examples to cattle, hogs, and chicken.

Forrester's next publication [57] then goes on to apply Industrial Dynamics to cities. He specifically builds a generic model of an urban system. The model is then used to show the growth and decay of the system. This proved to be quite controversial at the time [60], but has since been vindicated. Forrester also describes for the first time general characteristics of complex systems such as compensating feedback. The model also implies that the burden of responsibility lies on the intervener directly.

As a logical next step Forrester then proceeded to apply the same methodology on an even bigger, global scale. He correctly termed it "World Dynamics" [61]. The models and ideas presented therein were formulated as the basis for the "Project on the Predicament of Mankind" undertaken by The Club of Rome [62], which also was The Club of Rome's first publication. The Club of Rome is a non governmental organization, which proclaims itself free from any political, ideological, or business interest, that focuses on the solution of the most crucial problems facing humanity [63]. The purpose of that model was to study the human future. The model created for this purpose was very non-technical. It is built mainly on variables such as population, capital investment, natural resources, quality of life, and pollution. The core of the model depends on key relations such as death rates depending on various sources, such as material standard of living, pollution, and food. Consequently birth rates depend on very similar variables. The model also contains equivalent connections for capital investment, pollution, food, and quality of life.

He then proceeds to explore this model further in a section labeled "Limits to Growth" where he studies the sensitivities of the model to various assumptions, but notes that even the baseline is not sufficiently calibrate to serve as an accurate prediction of the future. The model as described contains four inherent forces capable of limiting population. They are depletion of natural resources, rise of pollution, increase in crowding, and decline of food. These are the central concepts in which Forrester explores the changes in his dynamic world model as each of the limiting forces is exacerbated by various scenarios. For example, Forrester's baseline model exhibits a limitation to growth from a high resource utilization, which then eventually limits overall growth and actually leads to a decline in population. Conversely, when the rate of resource consumption is reduced, such as new technology that improves efficiency, that scenario is then limited in growth by a pollution crisis. This crisis marks a very extreme shift in the dynamics of the system where it now exhibits a very violent and rapid reduction in population and quality of life. To explain the higher pollution with reduced resource use, he goes on to remark that a more sophisticated technology might conserve resources, but instead shift to more complex and more polluting forms of resource utilization. Additionally, he answers the argument that the extreme crisis will not happen because it will be realized before when the pressures from pollution become very extreme and therefore human behavior will shift to adjust and prevent the pollution crisis. Forrester's counter argument is that by then population will be so dense that to sustain the population a highly industrialized economy is necessary. This means that adjusting the economy to become less industrialized will trigger a reduction in population because it can no longer be sustained. The alternative would be to do nothing, which will then lead to the pollution crisis. So no matter the choice late in the development, a population crisis will be inevitable at that point. Therefore, it is necessary to understand the dynamic behavior of a dynamic world system very early on, such that long-term sustainable policies can be developed. There has to be a more thorough understanding of the consequences of policies and a shift from merely reacting to immediate developments to long term policy planning. This is where the term sustainable development was coined. Forrester concludes that obvious responses to these challenges do not suffice and exhibit four characteristics. First, obvious solution aimed at fixing problems in the social system lead to solving that particular aspect, but then create a new mode of complex system behavior. Secondly, policies directed at improving short-term outcomes are in direct conflict with long term policies. Thirdly, there exists a conflict between goals of a subsystem and the welfare of the system as a whole. Individual nations or even smaller entities such as cities etc strive for increased population, industrialization, quality of life and supply of food. However, this expansion is directly at odds with the limit of growth existing at a global level. Finally, social systems are inherently insensitive to most policy changes. This means that a look close to the symptoms of trouble will yield apparent causes that usually are no more than a coincident occurrence that is produced by the same dynamics. Additionally, any proposed policy changes that address this apparent cause will then usually lack leverage in the dynamics of the system to actually exact any significant change.

Forrester then explores various policy scenarios addressed to fix the apparent crisis developing in his system dynamic world model. In which he shows that no single measure can prevent this crisis. He then proceeds to try to eliminate the driving forces in positive feedbacks in the system that drive the exponential expansion that eventually causes the global crisis. The result is a set of policies that will achieve a long-term global equilibrium. This set consists of significantly reduced resource use, reduction in pollution, birth rate control, reduced food production, and reduced capital-investment generation. He, however, also admits that these results seem counterintuitive and pose significant implementation challenges that might cause these policies to fail in the end.

A large body of work concerning the application to various other disciplines is what follows after. One of these applications was to management [64], specifically focusing on production, operations, and human resources. This was then gradually extended to cover an entire company [65]. One famous model concerns the rise and fall of the Saturday Evening Post [66]. However, these applications were not limited to micro-economics, management, and business decision making but also covered macro-economics. This ranges from the first attempt to model long term economic cycles [67] to the model of an entire nation [68]. Other applications range from drug policy to environmental models to actual physical systems such as insulin response oscillations in the human body [69].

Further work was also done in trying to better conceptualize, formulate, and validate models. These efforts resulted in a more clearly defined visual representation of the elements of a system dynamics model [70] along with better defined modeling guidelines [71]. Finally, the concept of systems thinking [72] was introduced as a way to achieve organizational learning by means of communicating system structures to a non-technical audience by means of anecdotes and management flight simulators. The methodology of system dynamics was also extended by combining optimization [73], sensitivity analysis [71], and probabilistic methods such as Monte Carlo simulation [] with system dynamics tools.

2.4.3 Mathematical Foundation of System Dynamics

System dynamics models are normally constructed in a very methodical manner. This method is described in detail in Sterman's book [74] and instructor's manual [75] specifically intended for teaching system dynamics. It also includes a process overview on how to create system dynamics models. For purposes of brevity, only a short introduction of the basic modeling foundation is given here. A system dynamics model is constructed out of a small number of standardized elements. Each of these elements is described in detail in the following sections.

2.4.3.1 Stock

A stock is usually used to represent an accumulation of something. This is used to represent a number of real accumulators ranging from warehouses to bank accounts or simple cumulative variables. Usually system dynamics elements are explained with analogies based on water flows. Stocks in a water flow represent any form of tank or bucket that can be filled with or drained of water. More generally a stock represents a state variable. The underlying mathematical equation is simply:

$$S_t = \int_t^{t_0} (F_i - F_o)dt + S_{t-1} \tag{6}$$

where

S_t is the value of the stock at time t,

F_i is the sum of the inflow rates,

f_0 is the sum of the outflow rates, and

dt is the time step.

Figure 9: Typical Graphical Representation of a Stock

As shown in figure 9 a stock is graphically represented by a simple box. The representative variable name is then placed into the center of the box. A stock is then connected to other stocks by any number of inflows or outflows as needed.

2.4.3.2 Flow

A flow provides a basic representation of the inflows or outflows of stocks. A flow is required to be connected on one side as the outflow of one stock and on the other as the inflow into another. Stocks and flows together provide a simple graphical basis to represent conservation equations. Coming back to the water flow analogy, a flow simply represents any water flow or water pipe connecting stocks or containers. Fundamentally a flow represents a rate. Mathematically a flow is simply:

$$\frac{df(t)}{dt} = g(x_1, x_2, \ldots, t) \tag{7}$$

where

$\frac{df(t)}{dt}$ is the rate of change per unit time represented by the flow

g is the function describing the flow

x_n are the variables g depends on

t is time

Figure 10: Typical Graphical Representation of a Flow

As shown in figure 10 a flow is graphically represented as a double wide arrow controlled by a simple valve. Due to the fact that a flow has to connect two stocks with each other to preserve the overall conservation rules, this can become very inconvenient when trying to limit the scope of a specific system dynamics model. Therefore, the notation of a cloud like symbol was introduced specifically to allow flow arrows to connect to them either originating there or ending there. These cloud symbols denote the presence of a flow into or out of the model "universe" and therefore denote the model boundary, which indicates the limits of relevant model elements necessary inside the model to accurately represent reality. Or simply, anything before or beyond is, either rightly or wrongly, outside of the scope of the model and therefore is not considered. The valve represents the function that describes the flow. This function can take any mathematical form necessary and depend on any number of variables and especially time. Influence arrows usually describe this dependency.

2.4.3.3 Influence

An influence is simply used to represent a mathematical dependency of one variable onto another. Specifically, this defines a mathematical relation between one output variable and any number of input variables. Fundamentally it is:

$$y = f(x_1, x_2, \ldots, t) \tag{8}$$

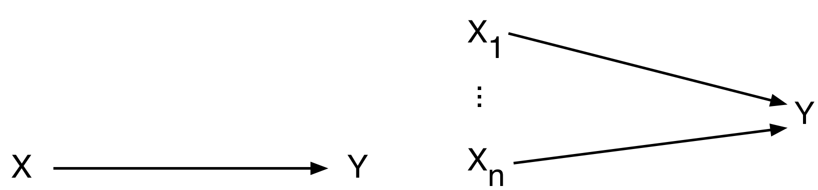
where

y is the resulting output variable

f is the function describing the output variable

x_n are the variables f depends on

t is time



(a) Single variable (b) Influence with more than one Input Variable

Figure 11: Typical Graphical Representation of Influence

Figure 11 shows an example of a typical representation. Often the arrow lines, however, are not straight but rather curved. This facilitates readability especially for more complex system dynamics models. Furthermore, a series of variables connected by influence arrows can be arranged in a circle. Such a circle then represents an algebraic loop that requires special consideration during the solution process. If however such a circle includes a stock and flow anywhere, it instead represents a dynamic feedback loop that is directly responsible for the dynamic behavior of the solution in the time domain. Sometimes such a feedback loop is specifically marked by a directional rotation indicator that shows more directly the general flow of information throughout the loop. Additionally, this can be accompanied by a "+" or "R" to represent a reinforcing loop or a "−" or a "B" to represent a balancing loop. This behavior is usually directly derived from the cumulative sign of all relations around a loop combined. To facilitate the determination of the loop behavior, sometimes a sign indicator is also added to the tip of the each influence arrow.

Another variation in notation is that sometimes it becomes necessary to introduce a time delay into an influence. This delay is the simply indicated by a "D" overlaid onto the influence arrow. Delay means that the influence of one variable on another is not immediate but rather delayed in time. This can happen in a number of ways but very often is linked to a more complex process, which could be represented by a series of stocks and flows. However, in many cases it is much simpler to introduce a straightforward delay in the influence as a somewhat simplistic place holder that nevertheless can serve its purpose. Should such a simple constant delay be insufficient, it is possible to either add influences or make the delay variable. If this is still unsatisfactory, it is then necessary to include a more detailed model of the actual process causing the delay.

2.4.4 Current State of the Art

This section is intended to give an overview over the current state of the art of system dynamics. The focus here will not be on particular applications, as system dynamics has been applied to numerous problems and many different fields. Rather this section will serve as an overview of the core beliefs or paradigms of system dynamics. Furthermore, an overview of learning games that have been developed to illustrate system dynamics and train decision and policy makers. Finally, there is an overview of available software used in system dynamics. This is important because computer simulation is essential to the practice of system dynamics.

2.4.4.1 The System Dynamics Paradigm

The core of scientific knowledge in any particular field of science can be formulated into groups of ideas or paradigms. This particular view of science as pertaining to system dynamics and compared to the history and evolution among other views is detailed by Bell [76]. This view is based on Whewell's theses [77, 78] about the nature of scientific knowledge and advancement. These theses were later advanced by Kuhn [79] to define a paradigm as a set of practices that define a scientific discipline at a particular given time.

Forrester as the founder of system dynamics eventually refined the core of system dynamics into a set of 14 paradigms [80]. Paraphrased they are:

1. Linear analysis is not suitable for industrial and economic systems because almost every factor in these systems is nonlinear.

2. Methods that presuppose inherently stable systems and equilibrium are invalid because industrial and economic behavior shows unstable and nonlinear behavior.

3. The model of a system should be used to predict the character and nature of that system for the design of a more desirable kind of system.

4. Construction of a model must not be limited to variables for which data exists. If data is unavailable best guesses must be substituted until exact data is available.

5. The model must not be limited to generally accepted variables, but also incorporate undefined concepts known to be of major importance and define them.

6. A model must not be limited to formal numerical data but rather try to include descriptive knowledge.

7. There is no distinction between exact and social science. Accuracy must not be measured in terms of numerical measurements but rather in the dynamical behavior of systems.

8. Physical sciences provide the foundation for model building. This refers to information-feedback systems that exist in engineering such as control systems rather than physical laws, which only relate to open systems.

9. Accuracy in the system structure is more important than the accuracy of parameters.

10. Accuracy must not be achieved before precision is useful. A precise statement with assumed values will be able to show the kinds of things that can happen. If these things are important, accuracy can be improved later.

11. It is not necessary to find optimum solutions. Mere improvement is sufficient. Since optimum solutions are generally only possible for simple questions, more is to be gained improving areas of major opportunity than by optimizing areas of minor importance.

12. It is possible to conduct controlled experiments not only with engineering models but also in management and economics.

13. Human decision making can be dealt with relatively few factors while the remaining ones can be relegated to noise and uncertainty.

14. Models should be directed toward policy which governs how decisions are made and not on decision making itself.

2.4.4.2 System Dynamics Games

A number of interactive games have been developed that allow direct interaction between players and a system model while a numeric solution is computed at every time step. These games are usually used in education and training in operations research and management. Some games are meant to be played cooperatively or competitively with other players while others are purely meant to be played alone against a set of external uncontrollable dynamics such as for example a market model.

The Beer Game

The Beer Game was developed at MIT's Sloan School of Management in the early 60's by Jay Forrester. It represents one of a number of management flight simulators that focus on teaching management skills derived from experiences while playing the game. This game focuses on the production and distribution of a given product, which in this case just happens to be beer. The model structure is described in detail in Sterman's system dynamics textbook [74]. There are also several versions available ranging from overhead projector slides [81] to a web based version [82].

STRATAGEM

STRATAGEM is an interactive computer supported board game available from the University of New Hampshire. It is based on a national system dynamics model that was used to show that captial self-ordering is sufficient to generate long waves [67]. A long wave is thought to be a long-term variation in economic growth with a period of fifty to sixty years. Each team in this game manages the development of a nation over a century. Key decision parameters range from population, agriculture, energy, and industry to pollution and trade. Its main use is as an introduction to problems associated with sustainable development.

Fishbanks Game[83, 84]

The Fishbanks LTD. Game is available from the Sustainability Institute and previously from the University of New Hampshire. This game is a computer supported board game concerning the management of renewable resources. In the game every player manages a fishery that competes with all the others but at the same time has to share fish as a renewable resource. Due to the competitive nature it is advantageous for each fishery to try to fish as much as possible. However, this will cause over fishing and lead to a drastic decline in available fish thereby hurting everyone. This game has been widely used in education in a wide range of settings.

People Express Management Flight Simulator

The People Express 2000 Management Flight Simulator is available from Global Strategy Dynamics, which is an updated version of Sterman's People Express Management Flight Simulator. This computer game is based on People Express Airlines. People Express Airlines was a low cost airline that was launched in 1981. The company grew rapidly, even offering international service by 1983. In 1985 People Express bought out Frontier Airlines along with several smaller regional carriers. The aggressive purchasing along with integration problems with Frontier caused by the different cost and operating structures placed People Express into enormous debt and only barely avoiding bankruptcy was forced to be sold off to Texas Air and merged operations with Continental Airlines a subsidiary of Texas Air in 1987. The game is based on quarterly turns where the player has to make decisions concerning strategy, operations, human resources and more. For example, the level of service, that is the amount of service provided to customers, which means that the airline is either a low cost airline or a full service airline, can be changed. The ultimate goal of the game is not to win or loose. It is, however, possible to dominate the industry or go bankrupt. The goal is to learn management of a company without risking a real business. This includes learning how to set prices, how much to advertise and how fast to grow and hire. These decisions influence employee morale, productivity, turnover, demand growth and the competitor's reactions.

2.4.4.3 System Dynamics Software

DYNAMO

System dynamics is based on the concept of numerical computer simulation. Although it is possible to solve a system dynamics model without a computer it is computationally infeasible to do so and really only possible for very simple system models. Therefore, it was imperative that a structured system dynamics solver be available, which is why Jack Pugh, one of Forrester's collaborators at MIT, created such a solver and a formal language in the early 1960s. This language became known as DYNAMO. It is syntactically very similar to FORTAN since it is based on describing a system model with the underlying equations that are then compiled into an executable that is then used to generate simulation results. DYNAMO is available from PA Consulting and there is also an introductory book available [85]. DYNAMO is available on a variety of computer platforms. However, recently PA Consulting developed a successor to DYNAMO, Jitia [86]. Jitia can directly import DYNAMO models but includes many advanced features including multiple integration methods, macros, and an array of other features.

DYSMAP

DYSMAP [87] is a PC-based simulation language that is very similar to DYNAMO, which was originally developed in the 1970s. It includes a optimization capability. DYSMAP is available from the University of Salford, UK.

STELLA/ithink

STELLA and ithink [88] are the academic and professional versions of a system dynamics software packages available from isee systems on both Windows and Mac platforms. STELLA is based on a graphical environment but has only limited data import and export and scripting capability.

Vensim

Vensim[89] is functionally identical to STELLA/ithink except for a wide range of features for analyzing model behavior. It is also similarly limited in its data exchange capability. There is also a free personal learning edition available.

Powersim

Powersim Studio is a system dynamics software package available from Powersim Software AS. It is functionally comparable to STELLA/ithink and Vensim except that there is also an SDK available that allows extension of the core software through a Software Development Kit (SDK). Furthermore, Powersim is also part of SAP's Strategic Enterprise Management (SEM) module.

Anylogic

Anylogic [90] is a Java based class collection and development environment that allows modeling and simulation with a variety of methods from discrete event simulation, agent based modeling, and system dynamics to name a few. Anylogic can create stand alone web applets that contain self contained code and models that then can be run by anyone connected to the internet. It is currently a very popular tool in the system dynamics community due to its flexibility and extensibility in a graphical development environment combined with ease of use.

Exposé

Exposé is a Microsoft Excel plug-in released in July 2005 by Attune Group, Inc. that allows the visual creation of a system dynamics model in Excel by linking cells together. Due to the nature of Excel this allows flexible data import while offering scripting capability through Visual Basic for Applications.

Simile

Simile [91]is a modeling environment derived from the Agroforestry Modeling Environment (AME) [92]. It significantly extends AME especially with external data import and scripting. Simile includes a visual modeling environment that at the same time allows conversion into C++ for compilation and highly improved execution speed and scripting.

Non System Dynamics Specific Software

Fundamentally a software package that is specific to the field of system dynamics is not necessary. In theory implementing some basic numeric differential equations solvers in a programming language of choice along with some data import and export can do the job, albeit without visual modeling capability. However, in reality there is quite some effort involved in creating a stable and robust solver environment. Therefore, it is much simpler to use an available tool with the required functionality. Since only fundamental feature that is required is a flexible differential equations solver preferably with a graphical interface and data import, export and graphing capability. This combination of particular functionality can be found in some smaller or less well known packages such as Microworld Creator from Microworlds, Inc. or some combination of open source simulation and graphics packages. Other software packages or programming languages such as MATLAB/Simulink [93, 94], Modelica [95] do also offer the same capability. It is important, however, to distinguish between these and other software that at first glance is able to produce models the look similar to system dynamics models. Examples of such packages are Analytica [96], which has only limited capability for time based solutions, and PACELab [97], which requires an external plug-in for time based solutions. While very impressive in their own right and with extensive capabilities to graphically link equations, they lack integration of differential equation solvers and primarily rely on non-linear algebraic solvers. Due to the extensive graphing and import and export capabilities present in MATLAB/Simulink while offering a variety of robust differential equation solvers, it will be the software of choice for this analysis. The only minor drawback is that it lacks capability to display models conforming to the visual model elements discussed in section /refMathematical Foundation of System Dynamics. However, this represents only a minor inconvenience since the model can still be represented in one of the system dynamics specific packages such as Vensim.

2.4.5 Existing Aerospace Applications

System dynamics has been applied to topics in aerospace applications before. This particularly has been attempted on the management and business side of the aerospace industry. This includes the governmental acquisition process and the resulting business cycles for the contractors where favorable policies for resource acquisition such as capital investments and hiring processes to minimize risk exposure were explored [98].

Another application in a very similar matter was the analysis of the interaction between government procurement and the aerospace industry for defense projects [99]. This work, however, focused on the identification and quantification of the interactions, specifically in the area of cost overruns and changes in the workforce in a time of decreasing military spending.

A study of externally driven growth of a business, focusing specifically on human resources [100], used system dynamics to develop a process for modeling a company's growth throughout program phases from development to production. This process focuses on variables such as management direction, support systems, and training. It also includes prioritization of activities and the use of technical professionals or contract firms for non-technical duties. Two diverse cases based on two companies were used to create the system dynamics model. One of the companies was a not further specified "aerospace company".

System dynamics was also used to study future resource management strategies within the Air Traffic Control (ATC) system [101]. This was based on the National Aeronautics and Space Administration's (NASA) and the Federal Aviation Administration's (FAA) Small Aircraft Transportation System (SATS). The study focused on the technological development of small aircraft and the management of airspace at airports. The fundamental question to be answered was if a Global Positioning Satellite (GPS) based ATC system can be pursued in the future and the impacts thereof. This was studied with a model including people, facilities, equipment, airports, aircraft, the FAA budget, and the Airport and Airways Trust Fund. With the help of three scenarios, one representing the continuation of current resource management practices, the second representing an emphasis on GPS based ATC system, and the third a combined strategy involving radar systems with GPS systems. The final recommendation was a strategy that focuses primarily on the development of radar based ATC systems.

Furthermore, there is also a series of work on strategic management of complex projects case studies with system dynamics that involve aerospace projects such as the Peace Shield Air Defense System [102]. System dynamics was also used as a tool to support business strategy, particularly the advantage of being able to create accessible and understandable high level models without the models becoming complex, difficult to understand "black boxes". This was demonstrated on a number of case examples also involving aircraft industries [103].

Another very pertinent work concerning the field of aerospace in general was an attempt at forecasting the aircraft market using system dynamics [104]. This attempt focuses specifically on the commercial aircraft market using data covering the period between the years of 1970 and 1987 and attempts to calibrate the system dynamics model, shown in figure 12, using that existing data.

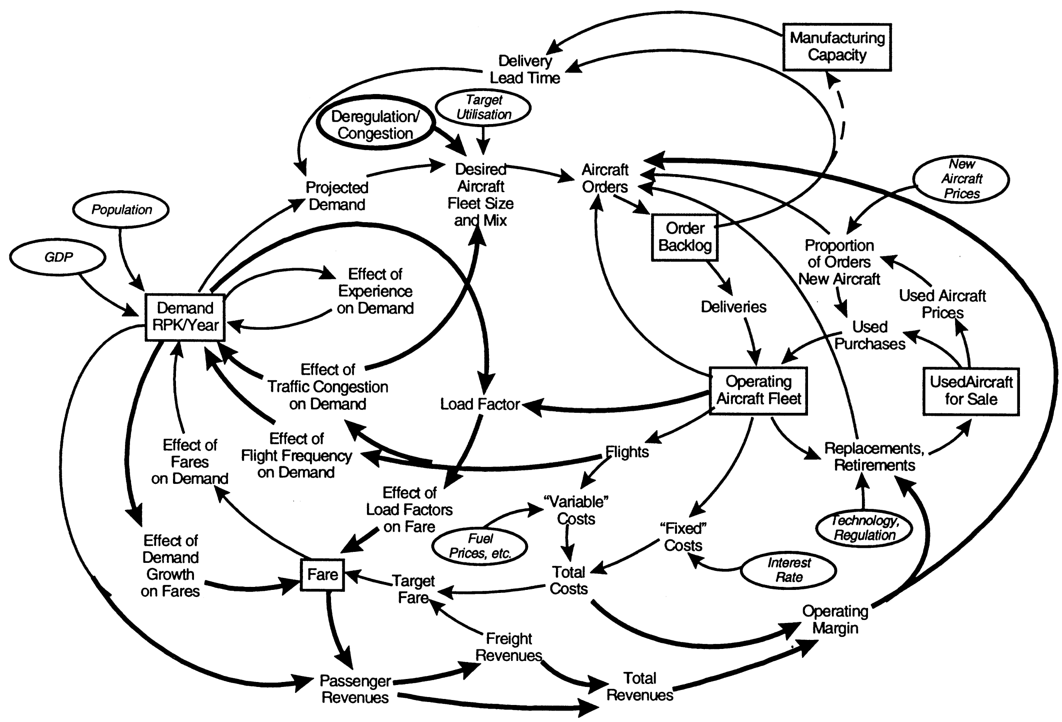


Figure 12: System Dynamics Model for the Commercial Aircraft Market [104]

The system dynamics model is then compared to a regression based model. The regression model is based on a simple regression of aircraft orders based on GDP and fuel price. When compared with the system dynamics model, the regression model clearly misses the order spike in the early 90's while the system dynamics model is able to predict the spike even with some deviation in the magnitude of the maximum and the amount of the following steep decline in aircraft orders.

It is also noted specifically, that when aircraft manufacturers face a decision of whether or not to build a new aircraft, such a model can help manufacturers with more informed decisions that forecasts that lack the ability to incorporate dynamics into the model. Furthermore, while this forecasting and decision making is not part of the traditional system dynamics policy design, it, however, does show promising results. The model particularly succeeds in the prediction of industry upturns and downturns. Finally, Lyneis notes that while it does succeed in providing guidance about industry trends, the model also needs added capability to be able to provide a range of uncertainty of the tends to allow companies a risk guidance in the associated trade-offs between production capacity and the necessary contingencies and buffers during rapid changes in the market. This further exemplifies the need for a more comprehensive statistical treatment beyond sensitivity analysis and parameter importance estimation [105] in system dynamics. In the end the use of system dynamics models for forecasting allows managers to receive an early indication about structural changes within the limits of the modeled dynamics. It also allows the identification of key scenarios and contingencies necessary arising from forecast inaccuracies.

CHAPTER 3

RESEARCH QUESTIONS & HYPOTHESES

After this overview of the background of aerospace systems design methodologies that were recently developed, it is now time to pose a set of questions that will be investigated during the course of the work. The use of system dynamics in combination with aerospace systems design tools and the use of system dynamics theory in conjunction with modern design techniques allow posing the following research questions.

3.1 Research Questions

1. Can an aircraft analysis be integrated directly or indirectly in such a way that the system dynamics model can be calibrated and produces stable solutions while at the same time being computationally feasible?

2. Can this integrated model be used for relevant future scenario forecasting and enable a comprehensive overview in a portfolio of the effects of future decisions or policies?

3. What is the result in overall model behavior of diverging time rates of change in various elements, especially in the face of rapid changes in oil price or accelerating technology development?

3.2 Hypotheses

In order to be able to answer those questions properly a number of hypotheses have to be formulated and then be proven or disproved. These are listed below:

Hypothesis I: Aerospace analyses can be integrated into a system dynamics model showing competitive behavior.

Hypothesis II: The solutions of such a model exist and are computationally feasible.

Hypothesis III: The model can be calibrated.

Hypothesis IV: There exists a multivariate solution space in which each solution represents a specific scenario.

Hypothesis V: This solution space can be explored using design space exploration techniques.

Hypothesis VI: Such a model can be used to generate future probability corridors assuming none of the underlying dynamics change.

Hypothesis VII: The model can be used to demonstrate responses to rapid changes in external drivers.

3.3 *Explanation of Hypotheses*

The hypotheses listed above taken together form a new method of comparing engineering concepts in a much broader context than before. Most importantly, this method is not static but can track changes over time. A more in-depth discussion of these hypotheses follows below.

3.3.1 Hypothesis 1: Engineering Analysis Integration with System Dynamics

Conventionally system dynamics uses relatively simple functional relations, lookup tables or even constants as a basis for dynamic system models. The accuracy of these model elements is often limited to qualitative behavior. It is, however, quite difficult to find the direct functional relation between engineering design choices and, for example, economic variables. This relation, however, is currently available in the form of design analyses, which often come in the form of analysis codes. They provide the direct link needed for a system dynamics model. Therefore, integrating an engineering analysis can link a qualitative System Dynamics model with the engineering and economical data output of a relevant project or scenario.

3.3.2 Hypothesis 2: Computational Feasibility

Since aircraft design and analysis codes depend on a number of discrete variables and the nature of the analysis, the outputs of them tend to be not very smooth or otherwise of particular use for calculating numerical derivatives. Additionally the noise in these analyses created by repeated iterations to arrive at converged solutions tends to increase the difficulty of calculating said numerical derivatives even harder. Therefore, the challenge here is to find a way to integrate the numerical output of aircraft design and analyses codes into a system dynamics model in such a way that the system dynamics model solution process will not be adversely affected. This kind of problem could either arise by the numerical noise that would effectively present the numerical differential equation solver with a "stiff" problem that would result in a significant increase in runs of the aircraft design codes to arrive at a solution with reduced fidelity. Alternatively the large number of evaluations needed due to the numerical derivatives could pose a equally significant challenge due to the execution time of the analysis codes. Even with a fairly fast execution time of these codes on the order of seconds could result in very extensive solution times for the overall system dynamics model. This would then be even more significant if another type of meta-analysis is used on top of the system dynamics model, such as a Monte Carlo analysis for example.

3.3.3 Hypothesis 3: Model Calibration

In order to show the usefulness of such a integrated model, it has to be able to be calibrated and tested on existing sets of data. The challenge here is that the types of data required in this case, engineering data and economic data, are usually not recorded together and even might not be publicly available. Secondly, the system dynamics model needs to be calibrated in order to generate more than just qualitative results. This, however, could pose a significant challenge because if significant dynamics present in the existing data used for calibration, the obviously too simplified system dynamics model cannot be calibrated directly. Instead, the system dynamics model has to be either modified or expanded as needed should such an issue arise. This could also result in significant issues if the underlying metrics and dynamic connections cannot be identified easily, especially if they arise from unexpected sources.

3.3.4 Hypothesis 4: Solution Space

The combined set of engineering and economic variable that feed directly into the design analysis as well as the system dynamics model can represent a scenario that through their time dependencies defines a particular set of time dependent solutions of the integrated model. The challenge here is that all variables exist as a set of time series that represents a large quantity of data. In a simplified analysis the last set of data representing the final state and the end of the forecasting period of the model could initially be used for further analysis.

3.3.5 Hypothesis 5: Solution Space Exploration

The set of time dependent data output from the integrated aircraft design and system dynamics analysis model needs then to be arranged into a more user friendly and useable format to enable a exploration of the set of scenarios. Ideally, if it is arranged in the proper form decision makers could then use this "scenario space exploration" to easily visualize and explore the data and arrive a desirable scenarios. The direct linkage of the integrated model inputs with the scenario outcomes is of particular importance here, so that desirable settings of inputs, or in other words desirable decisions or policy choices can be identified quickly. This can be enabled by the use of advanced surrogate modeling techniques to represent the data sets and to facilitate the visual communication.

3.3.6 Hypothesis 6: Probability Corridors

Once the integrated model of aircraft design and system dynamics has been calibrated and desirable scenarios have been identified, uncertainties in the inputs and particularly decision related inputs could be identified. This could be represented by statistical distributions associated with each of the variables. This in conjunction with a Monte Carlo simulation that includes the integrated design and system dynamics analysis could then result in probability corridors for scenarios that would allow the assessment of how likely certain goals could be achieved with a certain set of decisions and policy choices. Potential problems here are mostly related to the significant computational demand required to achieve at least somewhat credible time dependent output distributions.

3.3.7 Hypothesis 7: Rapid Changes in External Drivers

Finally, the model should be able to allow the investigation of very rapid changes in external drivers, which include technology developments in various disciplines. Another important external driver that needs to be investigated is the potentially sharp rise in fuel prices. This has become more relevant with recent spikes in actual fuel prices. These effects can either cancel each other or one can dominate the other. Therefore, it will be a valuable investigation to examine which levels of technology improvement is required to offset the rise in fuel prices.

CHAPTER 4

SOLUTION APPROACHES TO FORMULATING A MARKET MODEL

Before formulating a market model it will be helpful to explore the make-up of the market. This includes the major players and factors that drive the market and the decision making of each of the players. For the LCA market there are mainly two players involved. On one side there are the aircraft manufacturers that supply aircraft and spare parts at a given price to the operators. On the other side of the users are primarily the airlines that operate aircraft to provide a transportation service to the general public. It is in this environment of conflicting interests of cheap transportation versus profit from operation and sale of aircraft that airlines have to satisfy their passengers as customers and the manufacturers have to satisfy the airlines as their customers but indirectly also have to satisfy passengers as customers of their customers. To gain insight into this environment it will therefore be helpful to investigate the rational behind airline aircraft purchases.

4.1 The Airline Decision Making Process

The process of purchasing aircraft can be very intricate and complex due to a number of considerations that can vary widely depending on a number of factors. A review of basic airline economics suggests that issues such as price as subject to individual negotiation as are financing and credit costs. The most important factor indicated, however, is the cost per seat-mile[106, 107]. The use of fleet planning models through the use of optimization to achieve goals with certain constraints is also common. Most significantly it is stated that:

> Final decisions call forth all the skills of modern management. [...] But the final decision, fraught as it is with possibilities for profit or calamity, is made by top management in a manner that applies a delicate art to a great deal of science.[106]

This demonstrates that although purchasing decisions are based on directly measurable parameters, they are still subject to management discretion. This discretion can lead to deviations from predictions based purely on deterministic methods and can be highly dependent on individual circumstances. Therefore, it is wise to look at the prevailing methods inherent in the airline purchasing process.

4.1.1 Is there a Method to the Chaos?

The aircraft purchasing process can range from a number of obvious things such as cost and factors influencing the attractiveness of the aircraft to various political considerations such as loan guarantees interest rates import/export tariffs and local manufacturing agreements in certain regions and countries. Further complicating things are special discounts granted in a bidding process trying to out compete the competitor and secure a large order with a given customer.

However, as has been suggested by Sterman[74] and Forrester[80] the decision making process can normally be reduced to a relatively small number of variables that will be the most significant underlying variables that are considered. The remaining variables are often simple whimsical factors affecting a decision. These factors can be very dependent of the situation that the decision make was in at the time when the decision was made. Both Sterman and Forrester suggest that these uncontrollable factors behave like noise and are therefore best modeled by a defined uncertainty distribution around the underlying variables of significance. With this, it is now time to take a look at which factors most affect aircraft purchasing decisions.

4.1.2 Factors Affecting Purchasing Decisions

Since the early days of commercial aviation technology had been the primary choice for airlines in making purchasing decisions. This was true for the early propeller driven transports and then even more so when commercial jets became available. The Concorde is probably the last example at the tail end of that era. Since, this has slowly shifted towards replacing technology with operating cost as the primary factor for purchasing decisions. This has been studied in detail by the U.S. International Trade Commission, especially the competition between Boeing and Airbus with very detailed information on the global LCA industry and market, which was based on detailed market information and interviews of aerospace industry officials.

> This shift began after the deregulation of the U.S. airline industry in 1978, when carriers began to institute significant cost reduction and require manufacturers of LCA to produce more affordable and efficient aircraft. [...] Some industry observers believe that the resulting environment has adversely affected the industry: demand pull for technology has been diminished, the decline of airline engineering accelerated. [...] Any potential advantages of incorporating new technology are evaluated alongside airlines' incentives to continue using older aircraft that may be less efficient, but are already depreciated or available at very low prices. [...] An orientation toward technological progress is still critical, but is directed more toward improving the productivity within the production process than in incorporating technological advances in the aircraft[41].

This shows that in recent years there has been an increased focus on the economics of an aircraft in the design phases as well as the purchasing and operational phases.

4.1.3 Fundamental Program Characteristics

This study then goes on to list the most important characteristics an LCA manufacturer has to adhere to when marketing a product. Some of them already have been listed in a brief overview in chapter 2, but for completeness they are reiterated here.

The most important determinant is the net present value (NPV) of the aircraft, which is a discounted cash-flow of the vehicle that includes the purchase price and the operating costs. The operating costs consist of many inputs including maintenance, fuel, salaries and other costs. Furthermore, there are a number of ways to measure operating costs, some of which focus on different aspects of the operations of an aircraft.

First are the direct operating costs that include costs directly related to the operation of an aircraft on a flight-by-flight basis. They include fuel costs and other variable costs. The indirect operating costs mainly include fixed costs such as servicing and administrative costs. The total aircraft related operating costs represent a sum of the direct and indirect operating costs. It should be noted that the direct operating costs include the salaries of the required pilots, while the salaries of the passenger cabin attendants are counted as indirect operating cost. Further complicating this is the way these costs can be accounted for. This can be done either on a per-trip, per-flight-hour, per-block-hour, per-aircraft-mile, or per-available-seat-mile basis. Any of these introduce additional assumptions about average trip length, delays or other factors. While counting cost by available seat mile (ASM) effectively normalizes against aircraft size, it tends to favor larger aircraft since certain costs are independent of aircraft size, for example the pilot salaries are independent of size since any modern LCA is operated by two pilots. However, it could be argued that these salaries are normally defined as size dependent in many pilot union labor contracts, which is true in some cases but not for all airlines.

Another often quoted metric is the yield per revenue passenger mile (RPM) to break even. This represents a number that includes the load factor — the percentage of seats actually taken by paying passengers. It most directly reflects the minimum average ticket price that an airline has to charge to break even for a given route.

Since the airlines have no control over some of the cost components such as taxes, fees, and fuel price. Therefore, they tend to focus on controlling costs such as maintenance, which can be controlled more directly. Therefore a decisive factor of a manufacturer's competitiveness is the direct operating cost of the aircraft. Due to the high cost of incorporating new technologies, they must demonstrate cost-effectiveness to be applied to new aircraft. This requirement tends to limit new technologies to the following the categories[41]:

- Operating cost improvements of the aircraft, usually this refers to lower weight, fuel burn, and reduced maintenance costs

- Environmental performance improvements, usually this refers to lower emissions, noise, and manufacturing waste. This can also directly affect operating costs if any taxes or levies on certain aspects of the environmental performance exist, which is already the case in certain parts of the world or will be in the near future.

- Passenger appeal improvements, usually this refers to increased ride comfort especially as influenced by the interior environment such as seat spacing and internal noise level. This can, however, in some cases influence the operating cost negatively like the seat spacing that when increased will allow fewer passengers to be carried, for example. Therefore, airlines tend to only consider these improvements if they come at no or almost no cost associated with the operating parameters. Examples of this include the increased headroom of certain aircraft types, or entertainment system upgrades that can be relatively inexpensive upgrades, yet increase passenger satisfaction.

Other factors influencing the competitiveness of a manufacturer are the ability to react quickly to changes in the LCA market. This is an ability influenced by the first mover advantage[108, 109] that essentially allows the manufacturer that can bring a product to market first gain an advantage by being able to sell the product without competition for some time before other competing products become available, more on this later.

Another factor that can be very important is the commonality with other aircraft. This “refers to the use of common feature, parts, and systems in an LCA manufacturer’s aircraft that enables an airline to operate as homogeneous a fleet as possible”[41]. This effectively allows airlines to reduce the operating cost by using common maintenance parts and procedures as well as requiring fewer sets of spare parts and reduced training costs. Additionally, such commonality allows the manufacturer reduce development costs as well as manufacturing costs for derivative aircraft as compared to a completely new aircraft.

This also has been a factor in effectively shutting out Russian manufacturers out of the global market, apart from quality considerations, because of the incompatibility with commonly used avionics and engines[41]. On the other hand it is not completely in the interest of airlines to achieve total commonality, because that would effectively mean a monopoly in the LCA market, which would significantly drive up the acquisition costs.

Other considerations such as the presence of a global support network, which is also important for an airline, and certification requirements are important, but once met do not further affect the competitiveness of a given aircraft.

4.1.4 Goals for a Model

Given these characteristics of the LCA market, a model thereof should include the most significant features shown above. These characteristics include the purchase price, operating cost and a number of indicators for the capability of the aircraft such as number of seats and range. Furthermore, it is also extremely important that the model be able to model such factors as the commonality effect as mentioned previously as well as the first mover advantage that a manufacturer gains from being first to market.

4.2 Competition Model

The first element of using System Dynamics in the conceptual phase of aerospace design is to create relevant model. This model proposed here is a model that takes information normally arrived at during the conceptual phase of design and uses it to arrive at further results about the analyzed concept. These further results the hopefully can yield additional insight into the viability of the concept.

4.2.1 Polya Process Models

Specifically, the system dynamics model proposed here is one that models the competing market of two or more products. This model is taken from Sterman's Business Dynamics book [74] and is based on the Polya process [110, 111]. The Polya process is a simple statistical process named after mathematician George Polya. A simple description is as follows. If a container initially contains a black and a white stone each turn either a white or a black stone are added to the container. The choice is random but the probabilities depend on the ratio of the color of the stones already in the container. This will eventually lead to a firm lock-in to a specific ratio of stones in the container. However, over a large number of runs, the distribution across different ratios of stones in the container will be uniform. This process has been used in economics to explain the advantages of international trade even between equal partners, which then over time specialize in certain areas, and the benefits are independent of the areas the specialization occurs in [112]. This process combined with positive feedback dynamics was further used to explain the effects of a company's research and development spending on external entities and the rest of society [113]. An application of this process was also used to model and explain path dependent lock-ins in biology and evolution [114, 115]. Sterman presents this model as one that focuses specifically on the compatibility effect on the market. In his model the compatibility has a major influence on the final outcome and is used to explore the path dependency of a market. This path dependency is the tendency of a market to favor interoperability the larger the installed base is. The installed base is simply the accumulation of all past sales of each of the competing products. The key feature of this model and the Polya process in general is that the factors that decide the ultimate outcome take place in the very beginning [116]. This is consistent with the motivation to bring knowledge forward into the design process and hopefully will enable a better understanding of the decisions needed that will lead a program ultimately to success.

Figure 13 shows a simple competition model with characteristics of a Polya process. Specifically, this model tracks the sales of two competing products. What follows is a description of the underlying equations that form this stochastic process model. Starting with sales, which are accumulated in the stock labeled here as installed base, they follow the following equation:

$$I_i(T) = \int_0^T S_i(t)dt + I_i(0) \tag{9}$$

where:

i is the Product i

t is the time t

T is the time T

I is the Installed Base

S are the Sales

The rate of sales of each product are modeled as being directly dependent on the market share.

$$S_i(T) = D(T) * M_i(T) \tag{10}$$

where:

D is the Total Demand

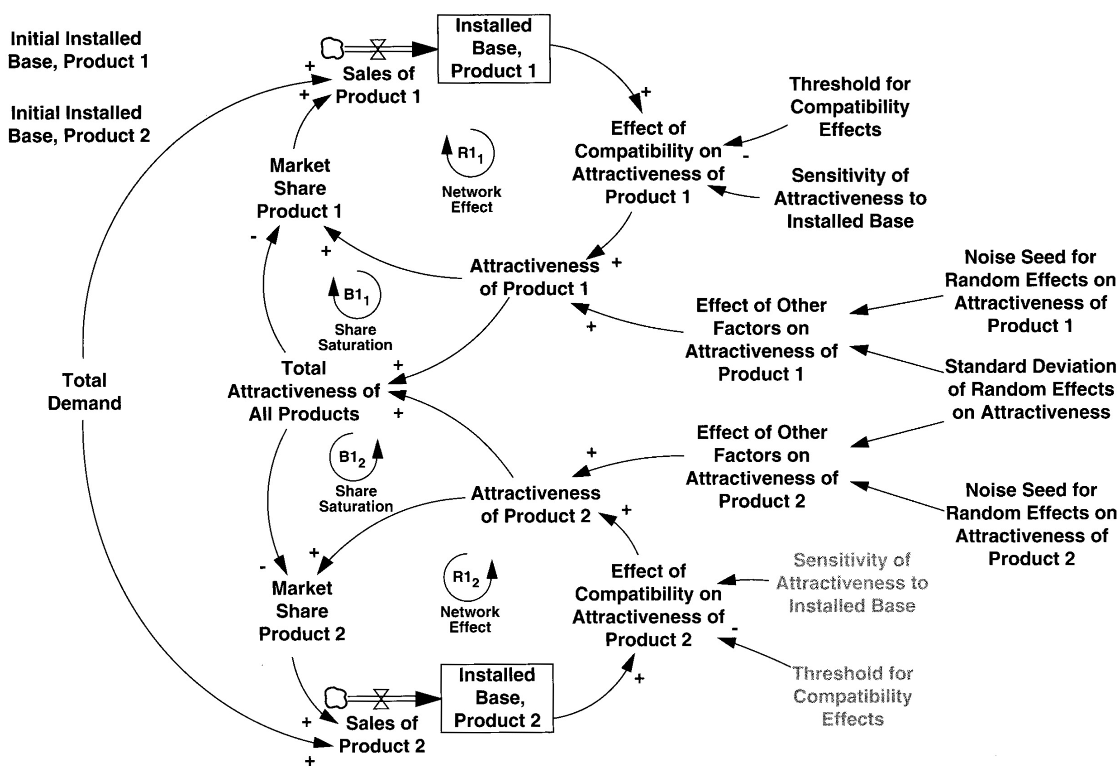


Figure 13: Simple System Dynamics Polya Process Competition Model [74]

M is the Market Share

Market share in turn in based on the relative attractiveness of a product compared to the competing product or products.

$$M_i(T) = \frac{A_i(T)}{O_{AllProducts}(T)} \tag{11}$$

where:

A is the Attractiveness of a Product

O is the Total Attractiveness

and

$$O_{All\ Products}(T) = \sum_{j=1}^{n} A_j(T) \tag{12}$$

The attractiveness of a product is based on the compatibility effect of the installed base of a product and other factors that represent additional factors not explicitly modeled in this model.

$$A_i(T) = E_i(T) * F_i(T) \tag{13}$$

where:

j is the Product j

E is the Effect of Compatibility on Attractiveness

F is the Effect of Other Factors on Attractiveness

The compatibility effect is modeled with a non-linear function of exponential nature that uses several constants, which in of itself can become parameters to allow finer control of this effect.

$$E_i(T) = e^{N(T)*(\frac{I_i(T)}{H(T)})} \tag{14}$$

where:

N is the Sensitivity of Attractiveness to Installed Base

H is the Threshold for Compatibility Effects

This exponential relationship models the increasing effect of compatibility on the product attractiveness as the installed base increases. However, this is a simplified modeling assumption that does not take into account the eventual diminishing effect for a very large installed base.

Sterman also applies this model to the context of the VHS vs. Betamax competitive market of video cassette recorders (VCR) and the competition that took place in the early 80's.

In the late 70's Sony introduced the first VCR to the consumer market, the Betamax system. Only over a year later a consortium of Matsushita, JVC and RCA introduced their competing VCR system, VHS. At that time Sony had sufficient time to introduce its system and essentially have a monopoly market. This is commonly referred to the first mover advantage [108, 109], which means that the first product to market has an advantage due to the fact that it can build a name and a significant installed base during the time until a competing product is released to market. However, if this significant advantage in sales cannot be realized, which is often a problem in revolutionary products, then much of the first mover advantage is trivialized. The reason for this especially in revolutionary products is that very often it takes significant effort to inform and educate the market about the product and its advantages. The market for a completely new product first has to be opened up. This can take significant effort to realize. If the first mover advantage is to be exploited and successful, then initial sales have to be strong and a strategy of making the product brand name synonymous with the product itself can prevent future competitors from realizing their second mover advantage.

Second mover advantage refers to the advantage gained in a particular market by coming to market with a product second. This might not seems as much of an advantage, however, it is two-fold. First, coming to market later can give the developers and engineers more time to perfect and improve a product. This means that being second to market can and should mean that the product is superior to the product of the first mover. Second, the first mover had to spend significant effort on educating the market and opening it up to a particular kind of new product. The second mover, however, can focus most of the effort on differentiating himself from the first mover and highlighting the superiority of the product. Furthermore, the second mover can also attempt to overcome any shortcomings the first mover had and prevented him from realizing the first mover advantage.

Coming back to the Betamax vs. VHS example, this meant that while Betamax had superior picture quality, its shortcoming was short tape running time. This meant that Betamax was not particularly suited to recording or storing full-length movies. VHS on the other hand, offered reduced quality at a much longer tape running time. So while Betamax was useful in home taping of TV shows, VHS was much better tailored for the sales and rental of movies. This additional business attracted further investments in mass production of VHS tapes and other accessories, which then lead to rapid commoditization of this market. This helped VHS to rapidly gain an additional advantage by being cheaper. These combined advantages lead to VHS rapidly eclipsing Betamax sales, which initially had held significant market share. By the mid 80's VHS was dominating the market, while Betamax was pushed to insignificance by high prices, very few accessories, and low availability of movies in the retail and rental market. Sony exited the market with Betamax in 1987 and started selling its own line of VHS VCR's the following year[117]. The history of this competition is shown in Figure 14.

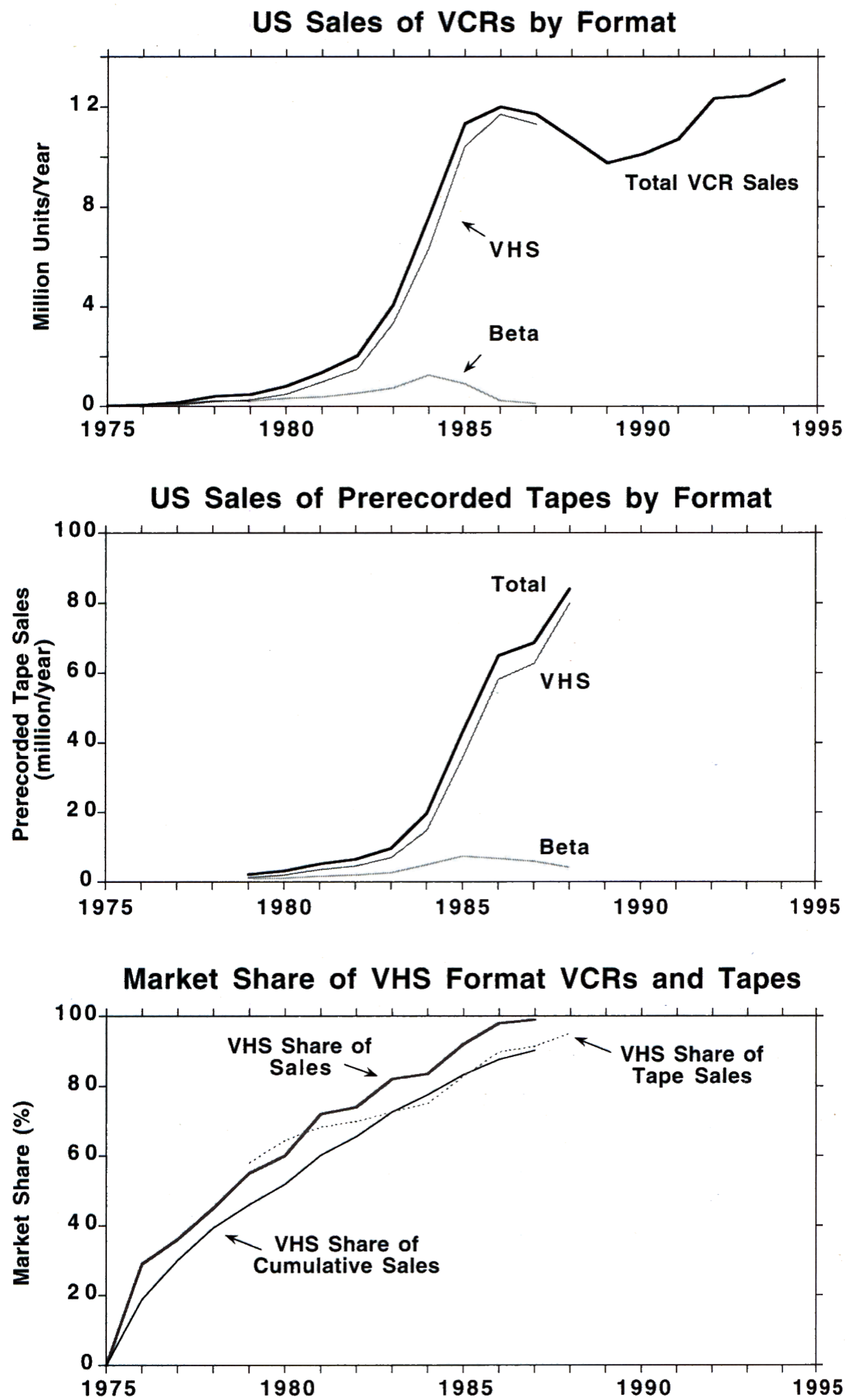


Figure 14: VHS vs Betamax Competition [118]

While the aerospace industry and market is significantly different in the customers it caters to, the size of the market, the commoditization possible, and very often the absence of competition. However, the commercial aircraft market is currently dominated by two major players along with several smaller players that mainly cater to niche markets. The two major players, Boeing and Airbus, compete with their products in a very similar way. The first mover or second mover advantages and realization thereof are equally present. The major influence of product compatibility is not as strong as in the case of commodity consumer electronics product standards. However, there still is an advantage of compatibility that the operators, which in this case are the airlines, can derive from exclusively using a certain product and manufacturer. This is especially the case here, because many new aircraft merely represent derivatives or evolutions of previously existing aircraft, which means that entire lines of either manufacturer share significant compatibility in parts and procedures. Therefore, airlines can achieve cost savings by limiting themselves to one manufacturer, even though a competing aircraft might otherwise be a superior choice. The effect of this is especially visible in low cost airlines such as Southwest and Jet Blue that both limit themselves to one manufacturer or even one specific line of models from one manufacturer.

Additionally, Sterman [74] makes a number of suggestions about extensions to the core competitive market model. Specifically, the inclusion of effects such as what is regularly termed learning curve [119, 120]. The learning curve is a model of the effect of learning while during the repeated production of a certain product. This means that during the successive production of a particular product, a learning process takes place. This learning then usually manifests itself in reduced production cost due to process improvements cause by increased experience. This causes a reduced unit cost in each successive unit produced. However, this takes place only up to a certain point, but it is a very significant effect in a market that is relatively limited in numbers, such as the commercial aircraft market.

The advantage of the inclusion of the learning curve effect is two-fold. A model including a learning curve feedback effect will allow the dynamic tracking of the projected and actual cash flows of the aircraft in question. This is of value because a shortfall in sales or later reduction in production numbers will yield a significant increase in the per-unit-cost of that aircraft. Similarly, an increase will reduce the per-unit-cost. Therefore, it is expected that small changes in sales numbers can significantly affect the validity of initial projections and the competitiveness in the market and with that the ability to produce a ultimate net positive cash flow. Secondly, this tracking then allows the exploration of the relationship between the initial assumptions and the final programmatic outcome. This can involve a sensitivity analysis thereof, which could yield insights as to how important these usually highly uncertain assumptions are to the success of a particular program. Furthermore, this then can result in improved guidance into necessary market conditions to launch a program that shows a higher potential of success.

Another aspect of this is that this time based forecast of the aircraft market for a particular design in the conceptual phase allows the tracking of the cash flow of the entire program throughout its entire life cycle. While similar cash flow tracking currently takes place during the conceptual design phase, this extension means that the predicted sales are no longer arbitrary guesses, but rather based on a virtual competitive market. This market not only includes information about potential competitors but also about potential customers.

4.2.2 Measures of Merit

The first consideration in building the competition model is to clearly define the measures of merit used to compare the competitors. In this case the original VCR market model uses the term market share to compare both competing products and eventually declare the ultimate winner. The number of sales generated with respect to the overall market in this case defines the market share. However, as shown in Figure 14 the bottom two graphs use the number of units to define market share whereas the graph at the top uses a dollar value to do the same. This begs the question what is the appropriate measure to use here. First, it is valuable to look at the number of options that present themselves. With respect to aircraft it is possible to use number of units or aircraft or their respective dollar value. Additionally it is also possible to use the number of equivalent seats as represented by each of the respective aircraft sold. Furthermore, it is not exactly clear what is meant by the term sale. Since aircraft represent investments of significant value the overall purchase is split into a number of phases and is not at all comparable to a simple purchasing transaction as would be the case in the purchase of a VCR at a retail store. Therefore, it is of value to more clearly define the way aircraft purchases usually take place. An airline or leasing company usually chooses to place aircraft orders at certain times with the official announcement thereof usually timed to coincide with certain political or public relations friendly events. These orders are usually either firm orders or order options or a mix of both. Firm orders represent firm commitments to buy a new aircraft from the manufacturer. An order option is a real option bought by the respective airline from the manufacturer that represents the right to have the option converted into a firm order at some time in the future without having to place a new order starting at the end of the production backlog. So essentially an order option represents more or less then reservation of a production queue spot at some point in the future. This is advantageous to airlines because it allows them to carry the flexibility of making purchasing decisions into the future and thus allows them to either convert the options to firm orders or simply let the options expire depending on the airlines economic and market conditions at that point in the future without having to make a potentially costly commitment to an aircraft purchase possibly years ahead. The value in this lies in significant risk reduction in the purchasing decisions of the airlines. However, this uncertainty is therefore shifted to the manufacturers since now a certain amount of orders do not represent real orders but merely options. This means that there is now a certain "quality" to each of the orders, which depends on the economic outlook and stability of the purchaser. Due to this orders by leasing companies and also orders from lesser-known airlines - especially from politically or economically less stable countries - are often considered less reliable than from major carriers from major industrialized nations.

CHAPTER 5

IMPLEMENTATION OF AN INTEGRATED MODEL

This chapter will address the approaches used to try and implement an integrated model. This model ideally should integrate the aircraft design process into a system dynamics model. This will also address the first hypothesis that it is possible to integrate two disparate methods of analysis.

5.1 Feasibility of Integration

The goal is to prove that it is indeed possible to somehow integrate the system dynamics method and aircraft design methods into a single analysis that then enables designers and decision makers to gain more information about a potential design solution and potentially the requirements for such a solution. Such integration can fundamentally take place in several different ways. The first approach is analytical whereas the other approaches rely on various numerical approaches.

5.2 Feasibility of an Analytical Integration

The analytical approach that will be explored here relies simply on identifying the underlying equations of both methods and then trying to obtain a solution without resorting to numerical methods such as what would be used on a computer. The knowledge gained in the background and literature search presented in Chapter 2 will be the foundation of this approach.

5.2.1 General Comments on the Feasibility of an Analytical Integration

The first step will be to take a closer look at how an analytical solution for an aircraft design is traditionally obtained. This involves a large number of assumptions that will also be presented. The fundamental idea to aircraft design is to split the process into two phases. The first phase is to define the characteristics of a new design. This means that the shape and propulsive characteristics have to be defined. These two characteristics are generally the most important because they define the aerodynamic and propulsive performance of a design, which are essential to defining the physical capabilities of a given design. This is done by calculating drag polars and thrust lapses by analyzing the shape of wings and fuselage and the thermodynamic cycle of the engine.

The most notably absent characteristic in this first phase is the weight. This is partly possible because both the aerodynamics and the propulsive performance can be defined independently of weight and size, at least to some extent. The limitation is that the drag polars and the thrust lapses can only be defined reasonably well for certain scales compared to properties of air. Most importantly flows can highly depend on the Reynolds Number (Re). This is an issue for both the flow of the wings and fuselage, which define the drag polars for the aircraft, as well as the engine, where the thermodynamic cycle highly depends on the efficiencies obtained by the flows over compressor and turbine blades. Therefore, the assumption here is that the drag polars as well as the thrust lapses are calculated with some a priori knowledge of the size scale of the resulting aircraft. This limitation is normally not significant since the aircraft design normally starts out by assuming a certain size, which then is adjusted iteratively over the entire process.

The result of the first phase normally is a constraint plot showing various flight conditions plotted on a thrust-to-weight versus wing loading graph. Such a graph is obtained by using the "master equation" [121]. The "master equation" is of the form:

$$\frac{T_{SL}}{W_{TO}} = \frac{\alpha}{\beta}\left\{\frac{qS}{\beta W_{TO}}\left[K_1\left(\frac{n\beta}{q}\frac{W_{TO}}{S}\right)^2 + K_2\left(\frac{n\beta}{q}\frac{W_{TO}}{S} + C_{D0} + C_{DR}\right)\right] + \frac{P_S}{V}\right\} \quad (15)$$

where:

$\frac{T_{SL}}{W_{TO}}$ is the thrust-to-weight ratio at sea level, take-off

$\frac{W_{TO}}{S}$ is the wing loading at take-off

α is thrust lapse of the engine in the given flight situation

β is the instantaneous weight fraction

q is the dynamic pressure

K_1, K_2, C_{D0}, C_{DR} are coefficients defining the drag polar

P_S is the weight specific excess power

V is the speed

Substituting the drag polar and thrust laps information along with knowledge about each of the flight conditions required to be flown by the aircraft will result in a number of equations relating thrust-to-weight and wing loading. These equations plotted on a thrust-to-weight versus wing loading plot result in a number of lines representing the limitations imposed on the design by the flight conditions. Normally, the result is the designer choosing a particular point on the chart to define a specific thrust-to-weight ratio and a wing loading.

This is followed directly by the second phase, where the goal is to define the weight of the aircraft. Weight along with the design point chosen in the previous

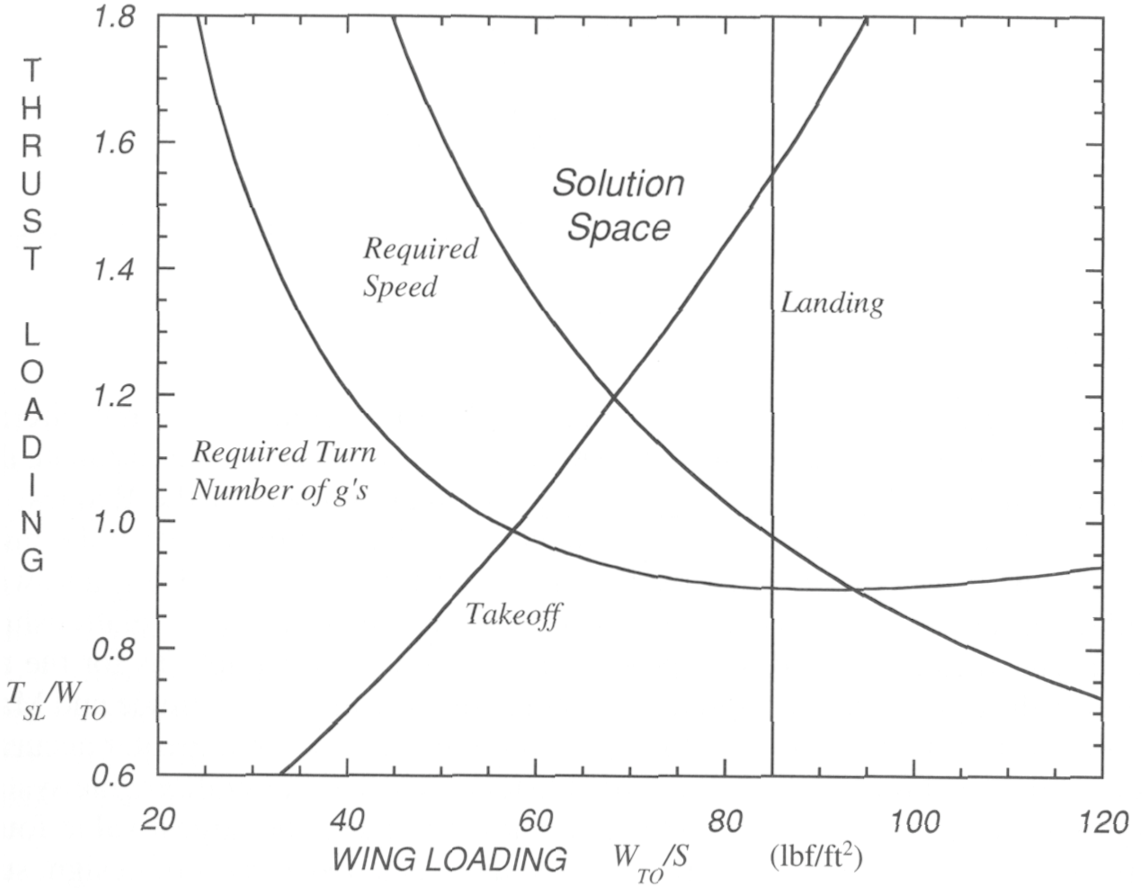


Figure 15: Constraint analysis – thrust loading vs. wing loading [121]

phase then results in the proper scale of the aircraft because now the thrust required from the engines as well as the size of the wings is known. The fundamental equation used is known as the Breguet Range Equation. It relates the initial weight and the final weight as weight fraction to the propulsive and aerodynamic efficiencies and the range [122]. This equation along with some knowledge about the weight fraction of fuel used during taxi, take-off, and landing and information about historical weight regressions for certain types of aircraft is all that is required to estimate the take-off weight of the design. It should be noted that both equations can be readily derived from fundamental conservation laws and force balancing [123].

This process has several implied assumptions that reduce its usefulness. The biggest assumption here is the knowledge of historical weight regressions, which could be very inaccurate or even non-existent for an entirely new class of vehicles. Furthermore, the process relies on manual expert user choice of the design point to be able to finish the weight and balance phase at the end of the process that ultimately determines the size of the design. Additionally, there are a number of uncertain parameters at this early stage in the design process — such as incomplete geometry definition and many simplifying assumptions — that introduce uncertainty into the results of this process. This is especially the case with designs that represent large departures from existing vehicle types and shapes or a set of operating conditions outside of the common ranges.

The result of this is that the design point has to be slightly offset from the most optimum corner of the design space as shown in figure 15.

5.2.2 Formulation of an Analytical Integration

An analytical solution therefore can be simply obtained by defining all the flight conditions an aircraft will be exposed to, from take-off to turns to cruise to landing.

Each of these flight conditions results in a functional relation between thrust-to-weight ratio and wing loading. This set of equations represents a set of constraints that limit the choice of design points in the thrust-to-weight ration and wing loading domain. In other words this can be formulated as a simple optimization problem of the form:

$$\min_{\frac{T_{SL}}{W_{TO}},\frac{W_{TO}}{S}} f\left(\frac{T_{SL}}{W_{TO}},\frac{W_{TO}}{S}\right)=\frac{T_{SL}}{W_{TO}}-\frac{W_{TO}}{S} \tag{16}$$

subject to:

$$\begin{array}{rcl}
\left.\dfrac{T_{SL}}{W_{TO}}\right|_{Flight\,Condition\,1} & > & f\left(\dfrac{W_{TO}}{S}\right)\\
\left.\dfrac{T_{SL}}{W_{TO}}\right|_{Flight\,Condition\,2} & > & f\left(\dfrac{W_{TO}}{S}\right)\\
\left.\dfrac{T_{SL}}{W_{TO}}\right|_{Flight\,Condition\,3} & > & f\left(\dfrac{W_{TO}}{S}\right)\\
 & . & \\
 & . & \\
 & . &
\end{array}$$

It should be noted that the objective function is a very simple formulation. It simply represents the desire to minimize the thrust-to-weight ratio while at the same time maximize wing loading. This represents the ultimate desire of the designer for a small and efficient engine and small and efficient — meaning low drag — wings. Plotting the constraints on a thrust-to-weight ration versus wing loading domain will result in a plot that generally will look like the on shown in figure 15.

The objective function is fairly trivial. However, the constraints are not and represent a significant hurdle. There are a number of ways to handle constrained optimization problems. However, most of them are outside the scope of analytical analysis. The three candidates for constrained optimization are Linear Programming, Sequential Quadratic Programming, and Penalty Methods. Both Linear Programming and

Sequential Quadratic Programming represent methods that involve iterative matrix manipulations and are well suited to numeric implementation. However, they do not allow for analytical solutions. The only candidate for analytical solution is the use of penalty functions. However, these functions can introduce severe non-linearities and possibly even discontinuities into the objective function, depending on the actual form of penalty functions used. Such an example of a penalty function formulation is an interior penalty function given the standard optimization problem with constraints:

$$\min_{x} f(x) \tag{17}$$

subject to:

$$g_i(x) \leq 0, i = 1, 2, ..., m$$

A simple inverse function can then be defined as:

$$\phi(x, \mu) = f(x) + \mu \sum_{i=1}^{m} \frac{-1}{g_i(x)} \tag{18}$$

In this formulation μ is the penalty number which is normally positive. Instead of minimizing f we now minimize ϕ which is essentially the same function as f with added large positive terms near the constraints g_i. This artificially introduces penalty terms that become very large at the constraints and in this particular formulation is known as interior penalty functions since the terms guarantee that the solution always occurs in the feasible region[124].

It is possible to create a function ϕ for a given aircraft design. However, the resulting function will have to accommodate usually four or more constraints for various flight conditions. This will make finding a general minimum difficult. Furthermore, μ has to be decreased iteratively to gain the correct constrained solution as $\mu \to 0$.

This iterative solution that is required makes it generally not feasible to obtain general analytic solutions to this problem. The goal was to potentially obtain generalized equations that could be integrated directly as a functional relation into a design model formulation in a system dynamics model. This, however, seems impossible now.

5.2.3 Implications

The reason for attempting to obtain an analytic solution of the standard aircraft design process was to accomplish the proposed integration into the system dynamics method by simple functional integration of analytical equations. This could have been possible if there were general solutions or underlying equations that are solvable by the system dynamics method. The equations solvable by the system dynamics method were shown earlier but generally have to be either ordinary differential equations or simple functional relations of the form:

$y = f(x_1, x_2, ..., x_n)$

Since y can, however, be on both sides of the equation, this means that it has to be separable. When moving everything to one side yielding the standard form for general systems of equations:

$F(\vec{x}) = 0$

This however still means that the system of equations has to be separable with respect to each component x_i with the remaining $f(x_j)$ where $j = 1, 2, ..., n$ and $j \neq i$ so that they are independent of x_i.

Such as system of equations can be very complicated to solve and there is no guarantee that in general solutions even exist. Therefore, it is customary in system dynamics to keep functional relations as simple as possible. This is achieved by using multiple intermediate variables to express sub-elements of otherwise larger and more complicated expressions. This furthermore aid in the creation of more expressive models at the expense of more variables. In many cases this serves to simplify the equations so much that many of them become simple linear relations. However, generally there still remain some non-linear relations such that it is impossible to use linear algebra methods for even special cases of system dynamics models.

Now that it has been shown that aircraft design equations of this form do not generally exist and the process involves optimization, it is clear that a simple analytical integration of aircraft design into system dynamics is not feasible. Therefore, integration will have to be achieved by numerical means. Numerical methods are one of the foundations of system dynamics because otherwise analytical methods would only allow solutions for very simplified special cases. Therefore, it is expected that the integration of aircraft design with system dynamics by numerical methods has much better potential.

5.3 Feasibility of a Numerical Integration

Integration by numeric means involves completely different pit-falls compared to an analytical integration. It is therefore necessary to first consider the potential problems of a numerical integration.

5.3.1 General Comments on the Feasibility of a Numerical Solution

As shown in section 5.2, the aircraft design problem generally involves an optimization problem. There are a number of techniques to solve constrained optimization problems such as this. All of them involve various numeric schemes and are of varying efficiency and capability. One of the most robust optimization schemes is called sequential quadratic programming, which involves an iterative quadratic programming technique that locally approximates a quadratic function space with constraints and is able to find optimum points in such a space in a single iteration. However, most problems are not generally linear or quadratic in nature and therefore this technique is applied repeatedly to a localized approximation thereof.

Such an optimization technique has to be wrapped around the previously stated aircraft design method. This can be done in various ways through the use of existing optimization libraries. Alternatively, there are already existing aircraft design codes that more or less follow the method described here. In general it would be preferable to use such a code to minimize the work required. However, such design codes usually were originally created for very specific purposes. This means care has to be taken to not exceed the capability and calibration of such tools, unless they implement a way to re-calibrate against additional data points.

Additionally, such tools often represent the legacy work of designers often decades back. This introduces complications since it means that they often exist only on very specific computer platforms and programming languages and the source might not be available. The unavailability of source code due to various reasons, ranging from proprietary concerns to export restrictions to age, introduces a significant limitation in the integration efforts since it means the integration is limited to the platforms of availability of the design code. Furthermore, design codes usually rely on some form of input files and produce output files at the end of their run. This automatically excludes a direct code integration through linked libraries, which means that the execution of a large number of design code runs is directly limited by disk input and output data rate limits.

Since solution of system dynamics models involves a numeric ordinary differential solver, a significant number of function evaluations have to take place which directly means that any design code will have to be run many times. This can become a potential barrier to a direct integration because this automatically limits the execution speed of these codes and leads to longer solution times of the system dynamics model. However, further evaluation is needed to test if such potentially much longer time to generate a solution to such a model is in fact unacceptable.

5.3.2 Formulation of a Numerical Integration

Numerical integration can simply be achieved by directly passing numerical values on a computer from one code to another. At first glance this might seem simple but in practice there can be a number of potential problems that can make a numerical integration very difficult if not impossible.

One of these potential issues that can arise from direct numeric integration – aside from code linking and execution speed concerns – is the stability of these design codes. More specifically, the numeric stability of design codes is limited due to two factors.

First, the allowable input ranges for the various parameters are not always guaranteed to produce converged results. This happens because some parameter settings do in fact not produce planes that can meet the given constraints and are therefore not physically possible and due to code limitations in the form of historic regressions and the like that limit the ranges.

Secondly, numeric optimization schemes rely on iterative schemes to arrive at their optima. This can introduce a significant source of numerical noise in the evaluation of numeric derivative approximations. Small changes in the input parameters can converge to slightly different solutions thereby introducing gradients into the numeric derivatives that in reality might not exist. This means that in some situations the accuracy of numeric ordinary differential equation solvers can be degraded significantly and special care has to be taken to limit the effects thereof.

To test this, a design code named the Flight Optimization System (FLOPS) was used for verification of this claim. More details on this code will follow in the next section. The code was setup to run with one of the baselines used in the application problem in the following chapters. One of the inputs to this code is a thrust-to-weight ratio that depending on certain flags is either used as the actual thrust-to-weight ratio of the design or as an initial point for an optimization. In this test the analysis mode was used that primarily performs aircraft sizing and does not optimize the thrust-to-weight ratio or wing loading or more specifically in the case of FLOPS the wing area. This means that the results as shown in figure 16 are not the result of an optimization as described above. However, the analysis performed still does contain an iterative solution for the aircraft weight. The response of interest chosen here was the fuel weight since it represents the most direct output of this iterative solution. Other results from FLOPS or even a cost analysis that uses the results of the sizing process could have been chosen, but most likely do not include as much noise due to the formulation of the cost calculations that rely on adding many component costs to reduce overall error in the analysis.

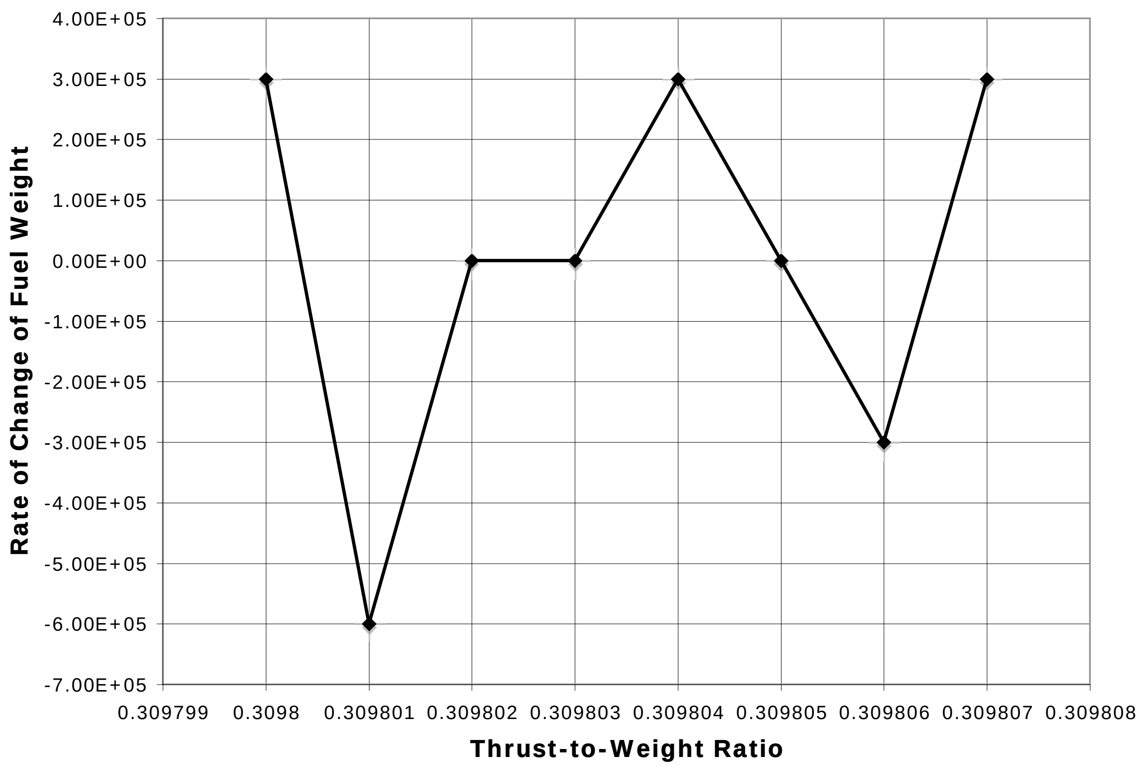


Figure 16: Noise in Numeric Design Codes

This brings up an important issue, namely accuracy. The error in such a conceptual aircraft design tools can be significant, on the order of several percent or more is possible. In the light of such possible error is the error in the numeric derivative shown in figure 16 even significant? To answer this question it is first necessary to examine how the results were obtained. The rate of change of the fuel weight was simply obtained by a simple first order forward differencing scheme and as such represents a simple and fast algorithm that is robust yet at the same time provides the worst case scenario of the potential error. The specific numeric derivative — shown here — represents how much the fuel weight changes with respect to minute changes in thrust-to-weight ratio all else being equal. The relative change in the numeric derivative obtained by comparing the absolute values to the rate of change in this case is on the order of one and a half times the absolute value. This means that a small change on the order of 10^{-6} in thrust-to-weight ratio hints at a change in fuel weight that could mean two and a half time the amount of fuel. This is clearly significant.

However, to realize this error, this means that the system dynamics model has to directly rely on this partial numeric derivative in the model. In such a case the error could accumulate quickly, especially if very small step sizes are used in the solver. Fortunately, this is not very likely since the system dynamics method relies directly on a time-based evolution of the studied system. This means that the solvers used mostly depend on time derivatives. For the error that was shown to be relevant it would mean that the thrust-to-weight ratio or another input has to be in the model and the resulting output would have to be based on the fuel weight or other derived metrics based thereon. Furthermore, it would then be necessary to make the input time variant such that the derivative used in this example would enter into the time based solution. This directly implies that the thrust-to-weight ratio changes over time. Such a model could in theory be possible, however, in reality once an aircraft model and engine combination is designed and in production the thrust-to-weight ratio is fixed not only for the entire production run, but also for the lifetime of the aircraft. Therefore, it is not conceivable that such a situation could arise where the noise in the numeric derivative of a design code directly influences the solution of a system dynamics model.

However, the absolute error in the model as mentioned earlier could become a major source of error in a system dynamics model and will have to be examined further once a specific model has been proposed.

Additional problems with the design code could arise from discrete settings inherent in the problem. For example an aircraft can only posses a whole number of engines and not fractions thereof. This could cause major issues similar to the noise in the code. However, again the aircraft design will be fixed over time and should therefore not pose a problem.

The next major issue is the implementation of automated runs of a design code. This means that it is now important to examine how such a code is executed.

5.3.3 Code Execution

Traditional design codes often have their roots in codes often developed decades ago. Such design codes therefore were developed initially with the computer systems and tools of that era in mind. This means that they were developed using programming languages such as FORTRAN [125] or others that were the most accessible to engineers not necessarily well versed in computer science.

On the other hand, early computers were not user friendly and lacked many ease-of-use tools available today. Therefore, the effort of using an integrated design code such as FLOPS was significant. Even today such legacy codes depend on a quite arcane interface based on command line interfaces with text file inputs and outputs.

Furthermore, a number of versions for various platforms are available. However, the very latest trend was to move from more traditional UNIX based platforms to standard PCs on a Windows platform.

Unfortunately, a number of tools designed to automate the execution of such design codes is only available for UNIX platforms. Therefore, it was necessary to develop a new way of automating such execution on a PC. Recently, a number of software packages such as Modelcenter [126] and iSIGHT [127] have become available that are designed to precisely automate such execution and aid in the general integration of a number of disparate codes into an integrated model.

However, these tools do not directly include differential equation solvers. Rather they normally include a number of numeric algebra solvers and a variety of optimizers and post processing tools designed to analyze and better understand a particular problem setup. Additionally, such tools are very expensive. In a educational setting this can be overcome through educational licenses but still can be a major hurdle. Furthermore, such tools add an additional layer of processing that can be detrimental to the speed of execution. In speed critical applications such as is the case here, this can mean the difference between feasibility and unfeasibility.

Therefore, it is necessary to create a simple relatively easy to use method of automating the execution of the design code. Such a tool is described in the following sections.

5.3.4 Surrogate Modeling

A potential way to avoid both — problems of numerical stability and integration — is the use of surrogate models in place of the actual design code. The surrogate model is created by running the design code at various setting of its input or design variable. The observed responses are then used to statistically analyze the variability and effects caused by changes in design variables. This is then used to create a simplified model that if created carefully for certain limited ranges can with sufficient accuracy represent the actual design code. Special care has to be taken to create a valid model that has sufficient accuracy for the required task. An overview of this process is given in Chapter 2.1.5.

Such a surrogate model — depending on the type used — is typically comprised of polynomial or similar standard functions. This means that such a model is now independent of the original analysis tool such that it can be integrated into other analyses, like the one proposed here, with relative ease. Furthermore, the simplicity of the surrogate model equations means that they are very cheap to evaluate with respect to the computational power required, since no complicated algorithms have to be followed nor any iterative optimization schemes be converged on.

Precisely this simplification also means that the evaluation of numeric derivatives, which will be required here, no longer exhibits any noisy behavior as shown above. This can be of considerable advantage since it means that the numeric solution process used to solve the sets of ordinary differential equations that make up the system dynamics model can obtain not only a more accurate solution, but also operate more efficiently.

5.4 Analysis Code Integration

The next major step is to integrate a design code into the system dynamics model. The design code of choice here is FLOPS/ALCCA. FLOPS is an integrated monolithic design code that allows the analysis and optimization of a given aircraft design. The aircraft design is defined by the use of either an internal or external set of aerodynamic information and external engine performance information. Additional parameters include the design mission parameters and which variables are used in the optimization. FLOPS then produces detailed outputs based on these settings. ALCCA is a life cycle cost analysis tool that then takes the FLOPS output and with additional input variables such as fuel cost, labor rates, economic mission, etc produces a detailed manufacturer cash slows as well as some airline cost analysis, which outputs economic parameters that are important to airlines.

These two codes are either available as stand-alone or combined versions. The input and output data is provided in simple text files and the overall run-times on current computers are only on the order of a fraction of a second. However, the executables are limited to a command line style system. This means that any analysis including these codes will have to be implemented on the same servers or server cluster as necessary. The command line interface will facilitate the necessary scripting. While is helpful the necessary file input and output directly limits overall performance to hard disk throughput.

The next step that is essential here is that the aircraft intended for inclusion into this model are modeled accurately in FLOPS/ALCCA. This means that a series of baseline input files would have to be created and calibrated to match the intended models of aircraft. This potentially involves matching geometry, aerodynamics, weights, cost, and performance in the input and output of FLOPS to existing data. Once that is established the baseline input files can then be used by scripts to be modified as required by changes of variable values in the system dynamics model.

The system dynamics software reviewed earlier for the most part has severe limitations on the data import and export allowed by the software. However, a flexible architecture for data input/output is essential here, because the main goal is to integrate system dynamics into conceptual aircraft design, or rather integrate aircraft design information into a system dynamics market model. The only choices that allow easy integration as necessary here at no incremental cost are the use of MATLAB/Simulink, or the implementation of simple numeric integration algorithms. The implementation of numeric integration into a programming language of choice does not present a significant challenge in itself. However, implementation in a robust manner that also allows flexible use of various algorithms - be it fixed or variable time step methods - represents a significant amount of work that does not directly advance the purpose outlined here. Therefore, Simulink currently represents the best choice for a system dynamics integration environment available here. Simulink does include a wide variety of numeric integration method implementations that are quite robust. Furthermore, it also includes a quite powerful visualization suite that will definitely simplify plotting the anticipated insights. However, one unfortunate side effect is that Simulink's interface is focused on control system simulation, especially for electronics and physical systems. This means that the visuals used by Simulink do not match the standardized visual elements used in system dynamics as described earlier. Therefore, to avoid confusion it is probably best to limit the visual references to models in Simulink format, except for comparison purposes, and represent the models used here in the standard system dynamics format, which can easily be recreated with freely available system dynamics software such as Vensim PLE.

Therefore, the immediate next step is to integrate FLOPS/ALCCA into the system dynamics model. This integration has to be achieved in two ways. First, the mechanical integration of code has to take place. This can be achieved by creating a wrapper that produces input files for FLOPS/ALCCA based on a baseline library of aircraft, which are then modified by certain input parameters in the system dynamics model. This wrapper will then execute FLOPS/ALCCA and parse the generated output for the desired output variables. These variables then are passed along to the rest of the system dynamics model. Second, the conceptual aircraft design analysis has to take place inside the model where it is warranted and supports the overall model. The proposed place in this model is the directly at the variable named effects of other factors on attractiveness of a product. Since the objective here is to model the commercial aircraft market, the variables of concern are the applicable variables of interest to airlines. Furthermore, there is usually more than a single variable that affects the attractiveness and as shown in the model presented earlier the purchasing decisions and therefore market share. This means that various variables concerning aircraft are used to feed into the attractiveness function of the product.

As described earlier it was necessary to develop a quick and simple way of executing FLOPS/ALCCA to quickly obtain values for inclusion into the system dynamics model. This was achieved by creating a simple program that reads in a comma separate value file containing information which variables to change to which values for every run contained in a line. This change occurs in a copy of a baseline file that contains all the necessary information of a specific aircraft. The design code is then executed on the modified copy of the baseline file and the resulting output is then parsed according to another parse information file containing simple information such as the string sequence at the beginning of the line containing the desired output along with the number of the string containing the value as split by empty space and finally the number of occurrence of the beginning of the string sequence at the beginning of the line.

```
FUEL WT, 4, 1
TOGW, 3, 1
Final Aircraft Price, 7, 1
Average Yield/RPM, 4, 1
```

Figure 17: Parse Information Example

Such an example of the parse information definition is shown in Figure 17. The baseline files used are shown in Appendix A. The file format for defining the runs and which variables to switch is shown in Figure 18.

This simple tool helped greatly in executing the required runs as will be described in the following sections.

```
RTRTN,NV
9.36,846
4.71,1048
4.81,1135
```

Figure 18: Run Definition Example

5.4.1 Direct Integration

Obviously, these multiple variable or attributes of the aircraft in question have to be reduced to a single variable, here called attractiveness. This reduction does not necessarily have to take place as shown in the model through attractiveness. Nevertheless, the aircraft attributes still have to be reduced to a single variable. The latest this can take place in the model is at the point where sales are determined. For a model of the commercial aircraft market, it would be prudent to eventually create a more flexible model that allows differentiation for different airlines or at least different prototypes of airlines, but for now this will be implemented in a simplified form that hopefully can be calibrated and is useful for examining the general dynamic behavior of the market model. Multiple aircraft attributes will be combined with certain preferences that are representative for each of the airline prototypes. This final combination of attributes then has to take place using multi-attribute decision-making techniques (MADM). While there are numerous techniques available [128, 129, 130, 131], it should be sufficient to integrate a technique that allows automated execution without manual interaction while still producing sufficient separation in ranking between competing products or aircraft such as TOPSIS [132]. Once this has been achieved the model will then use this information to generate sales data.

This approach however can exhibit certain potential issues. Therefore, it is wise to define potential solutions or alternate strategies as required. The first potential problem deals directly with the integration of a design code into a system dynamics environment. As shown earlier, inputs and therefore also the outputs of such analysis codes tend to be not very smooth or have discontinuities or even discrete values. This can significantly affect the ability to calculate numeric derivatives accurately or even the overall stability of the solution or the algorithm. There is no simple answer on how to alleviate this issue. However, stability can be achieved by using lower order or fixed time step as opposed to variable time step algorithms or both. While this represents a trade-off between stability and solution efficiency, the resulting increase in necessary analysis code evaluations should not be drastically higher and still be feasible given current computer technology.

This brings up another issue. The numeric methods mentioned require a significant number of function evaluations for every time step. This is needed only for the function values but also their derivatives. The higher order the method is the more higher order derivatives have to be calculated. The improvement gained is higher accuracy and therefore larger time steps are possible for solutions with similar accuracy. However, no matter what method is used, a very significant number of function evaluations of the proposed monolithic design codes will be required. Relatively simple system dynamics models, such has the one proposed, do not require an excessive amount of function evaluations due to the limited number of simultaneous differential equations modeled. Unless the intended simulated time period is large along with a very fine necessary computational resolution, it is estimated that existing computational resources, if necessary parallel computing clusters that are available, are adequate.

5.4.2 Indirect Integration

The alternative to brute force computational power in this case would be the use of surrogate models in place of the full analysis codes. This could potentially also alleviate some of the other problems mentioned. The well-defined functional nature of most common surrogate modeling techniques automatically reduces the numerical stability and accuracy problems mentioned earlier. Furthermore, very large numbers of function evaluations are possible due to the very small resources required in the evaluation due to the simplicity of the surrogate models. Although this is not necessarily true for some more advanced surrogate modeling techniques. As long as the surrogate models used in the system dynamics model are properly constructed and accurately represent the analysis code outputs, the potential benefits warrant further investigation into this potential method.

Finally, it will also serve the illustration of this proposed methodology and the competitive market model to not only allow scripted runs for sensitivity and probabilistic studies, but also an easy to use "flight simulator" interface that allows the rapid execution of single scenarios without sophisticated programming knowledge.

5.5 Competition Space Exploration

Once the integrated market model has been completed and calibrated, it will allow the automated execution of various scenarios, each represented by different settings of various parameters in the model or the design code. Since the number of potentially significant parameters is enormous and the subsequent search for favorable and desirable scenarios involves a highly dimensional variable space. This, however, can be reduced to a manageable effort by applying the design space exploration techniques mentioned earlier. However, this model does not directly involve the exploration of a design space, but rather various scenarios. One could term this a scenario space, which is the variable space formed by the various outcomes of scenarios as defined by the scenario parameter space. Therefore, the resulting analysis could then be termed scenario space exploration, or in the case of the proposed competition model competition space exploration.

For this analysis, however, it is important to be able to create surrogate models of the competitive market model proposed here. This can be achieved with the standard surrogate modeling techniques with one exception. The competitive market model is time dependent. This normally would not prevent standard surrogate modeling techniques from being applied. Specifically, time can be treated as another variable among the rest and the surrogate models can then be created as normal. However, the time dependent nature of the competition model, especially the dynamic nature that with certain parameter settings can exhibit a wide variety of behaviors, only some of which lend themselves to easy surrogate modeling. Furthermore, the behavior can change its fundamental character over time. This means that for example a model can show exponential growth for a certain parameter, but then approach a limit and oscillate around that target value. This behavior cannot easily be captured by some of the simpler surrogate modeling methods such as response surface equations without previous knowledge of the type of behavior in certain areas of the variable space. One possible solution is to treat time as a discrete variable and essentially produce independent surrogate models at certain given discrete time steps. The resulting set of surrogate models then can be used to evaluate the outcomes after which the time dependent variations then can be extracted and interpolated to recreate a continuous curve to create the appearance of time as continuous variable. Also some more advanced methods such as neural networks could be suitable for creating a single non-linear surrogate model of a system dynamics model. Alternatively, only the end state of the system can be considered which was demonstrated by Kleijnen [133] with a small model with three input factors and one output variable. The eight runs necessary were conducted manually. However, Kleijnen also defined three policy choices that allow the demonstration of a trade-off between the three policies. Therefore, a total of 24 runs were necessary to be able to decide between policies. This interesting choice of discrete policy definitions also warrants further explorations outside of the continuous scenario space exploration.

Another way to attempt to capture the time dependent behavior of dynamic systems is to resort to time series forecasting methods. This means that rather than treating time as just another independent variable it is treated separately. Namely often this is done by dividing time into equally spaced small time intervals that then form the foundation of formulating a set of equations that can then be used to solve for the desired responses at an advanced time step as a function of that same responses' value at previous time step or several time steps.

For most cases, however, the most important responses will be those at a selected final time. Meaning that the final outcome of a certain model will be of much higher importance than intermediate solutions. For example, in the proposed competitive market model a response with high significance is the market share or return of investment. The market share of a product of interest is of most interest at the end of the selected time period, for example after a number of years. The intermediate behavior is of lesser importance, but can still serve a purpose by exploring its time dependent behavior for what-if scenarios for sudden changes in market conditions or as management training tool for exploring the dynamic behavior of the market for example for exploring different attempts to capture an increased market share.

The final outcomes of such a time-based analysis can then also serve as an additional set of system responses. This automatically then can also enable the integration of the system dynamics market model into probabilistic design. This has the advantage of then being able to track the probability of success not only purely based on engineering and cost responses but also additionally on the outcomes of the market model. The result of that is that not only the probability of successfully meeting performance and emissions and cost targets can be tracked, but also the probability of successfully meeting a market share or sales goal. Furthermore, it is then also possible to track individual designs or ranges of designs meeting target goals concurrently and which goals represent the most difficult challenge.

The integration into probabilistic design can take place in a number of ways. The most commonly used method is to create a surrogate model of the analysis and then use that surrogate model to perform a Monte-Carlo analysis. This involves defining individual distribution for each of the tracked design parameters. These distributions are then used to generate randomly distributed input settings as defined by the distributions. The surrogate model of the analysis will then generate response outputs that can then be arranged into output distributions which then allow tracking the probability of success. This overall process is very well understood, but poses some significant challenges for application on a system dynamics model.

First, there is the issue of surrogate models of time dependent analyses. As shown earlier it is possible to create surrogate models of such an analysis, however, it is significantly harder to do so. This could mean that in this case it would be simpler to not resort on surrogate models for the Monte-Carlo simulation especially if the surrogate models are incomplete due to the time dependent nature of the system dynamics model. Instead it could be performed directly on the analysis as long as the computational effort for each analysis code run is small enough so that the Monte-Carlo simulation runs required can be executed in an adequate timeframe. In case that this would take too long, there are some more advanced statistical techniques that allow the generation of output distributions with much less analysis code runs. However, that comes at the price of reduced accuracy.

Second, due to the time variant nature of the system dynamics model, it is important to define properly what is meant by random distribution in this context. A random distribution for a given variable could simply be defined as the distribution of the initial setting of that variable at the beginning of the simulated time period. However, certain variables can also exhibit randomness in the time domain such as the fuel price for example. This means that a time dependent random distribution has to be defined for such a variable. This means that at every given time step a new random number has to be generated. The problem this can incur is that due to the variable time stepping solvers available the random number generation does not take place at predetermined points in time, but rather on solution and solution accuracy dependent times in the simulated time domain that can furthermore vary as the solution changes shape with changes in input parameters. This is clearly not acceptable. The result is that this means that in this case the solvers have to be limited to fixed time step methods that then use predetermined time steps as required by the simulated time domain. This could be, for example, month-to-month or day-to-day time steps for successive solutions as required by the model, which for the proposed example would be months. The drawback of this is that this will in most cases increase the computational effort by a significant amount, since very often variable time step solvers can arrive at solutions quicker than fixed time step solvers, especially when the time step is not driven by the required solution accuracy but rather other external factors such as is the case here.

Furthermore, simple generation of random numbers based on defined distributions at every given time step may not be accurate for all variables. Generating new random numbers in that way means that at any given time step is completely random and independent of previously generated random numbers at previous time steps. This type of time dependent distribution is generally referred to as white noise in an analogy to white light that has an equal power distribution over all frequency bands. While this might be accurate for some variables, it is not the case for all variables. A specific price, for example, assuming a normal distribution with a mean of $2 and a standard deviation of $1, typically does not jump from one day to the other, or even month to month, from $1 to $3. While this can happen, the likelihood of it happening is smaller than a small change in a given time period. White noise by definition has an equal likelihood of small and large changes from one time period to the next, which is definitely not the case in some variable of interest here. This means that a certain time dependency has to be introduced into the random number generation. The solution is to introduce a frequency-based attenuation so that large changes over a given time period have a smaller chance of occurring. One specific way is to use what is commonly referred to as pink noise. Pink noise introduces a proportional relation of power density and frequency so that each octave contains the same amount of power meaning that in an audio signal all frequencies would appear approximately equally loud. This is why pink noise is commonly used in audio engineering as reference signal. The specific relation is a power density proportional to 1/f, which then amounts to a decrease of 3dB per octave. This means that by using a pink noise random number generator a certain time domain dependency is introduced that will allow a more accurate model of variable such as fuel price. Specifically, pink noise emphasizes small short-term changes and de-emphasizes large short term changes. There are several ways of implementing a pink noise generator, however, most focus on modifying a white noise generator [134].

A variation of this is Brown noise generator, which is equivalent to simulating Brownian motion. The power density there is proportional to $1/f^2$ or a decrease of 6dB per octave. Specifically commodity and also the oil price has been shown to follow a Brownian motion closely [135]. It is, however, under contention which approach should be used to model commodity prices. Some economists argue for the Brown noise, while others argue for the pink noise. In this case Sterman argues for the pink noise, which is what is used here.

5.6 Probability Corridors

Once meaningful distributions of all of the relevant inputs of a specific model have been defined, a Monte Carlo simulation can be run with the model and all of its associated sub-analyses. This means that the system dynamics model is run repeatedly with randomly generated inputs as defined by the distributions on them. The result is a number of probability density functions for each of the tracked model outputs. These probability density functions can then be used to assess the likelihood of a specific outcome. A significant number of model runs is required to simply define the basic shape of the output probability density functions. A commonly used number is in the tens of thousands of runs as long as the tails of the probability density functions are not of specific interest. Should that be the case a much larger number of runs, usually in the millions, can be warranted. To reduce the computational effort and therefore run time it is convenient to run this Monte Carlo simulation with the surrogate models created earlier instead of on top of the actual system dynamics model. Should, however, the computational effort be acceptable the Monte Carlo simulation can also be run on top of the actual system dynamics model instead. This can have the advantage of increased accuracy.

Since the system dynamics model outputs are all defined as time dependent variables, the resulting probability density functions also exist as time dependent functions. This somewhat complicates matters since the probability density functions can now no longer shown as simple line or bar graphs. Instead it is now necessary to either plot them as three dimensional surface plots or alternatively as contour plots mapping them again onto a two dimensional plot showing the output variable value versus time with the overlaid contours showing the probability. It is also possible to reverse the axes and show the probability versus time with the overlaid contours showing the output variable value. In either case the predicted outcome of the system dynamics model can be seen. Due to the time and path dependency of specifically the model proposed here, it will be possible to see "probability corridors". These corridors are literally paths of increased probability linking initial, intermediate, and final states of the system dynamics model while showing paths of high likelihood that the modeled system could take. The value here is that not only end states visible but also the dynamic behavior before achieving them. To facilitate the visualization of these "probability corridors" it might be of value to create yet another surrogate model. This time the surrogate model should be created from the time dependent probability density functions. The purpose of that is that this will enable importing and then direct interaction in a prediction or contour profiler that allows direct and immediate interaction with the surrogate model created from the time dependent probability functions. The value of that lies in the possibility of "what-if" games that can directly change the input parameters that as such define certain policies and scenarios and then immediately observe the change in the "probability corridors" and therefore not only the change in end states of the system dynamics model but also the change in likely paths the system can and will take.

5.7 Process Overview

The proposed overall process is depicted in Figure 19.

The left side of Figure 19 shows the core of the proposed integration of engineering analysis with system dynamics. This can be either achieved by direct integration or by means of surrogate models. The system dynamics model along with well-defined scenarios then can be used to generate future outcomes. However, these outcomes are initially deterministic and therefore of no great value. The integrated system dynamics environment can still be used to generate a scenario space by defining ranges on each of the pertinent scenario variables. These inputs then can be used to rapidly trade off various scenarios or policies and their outcomes, which is enabled by the rapid trade-off environment possible in a prediction profiler. The underlying surrogate models can then be used alongside scenario parameter uncertainty probability distributions to create probability density functions of the end state of the simulated system or as time dependent probability distributions, which is here shown as a contour plot. The ability to rapidly trade off the various outcomes and the likelihood of achieving certain desirable or undesirable paths in the system in question can provide a valuable tool for policy and decision makers because it allows them to very quickly explore different scenarios and visualize the associated risk thereof. An example of such a scenario and how it is defined is shown in Figure 20.

As can be seen there, two competitors have products in a competitive market. Both products are distinguished by a difference in a certain disciplinary metric. This is only specific metric and does not cover any other metrics or a overall comparison between products. For this scenario both competitors decide to invest into the development of a new technology that will improve both products. However, this improvement takes place in two different manners. Product A will be first to market with the new technology. The improvement is significant for this specific metric.

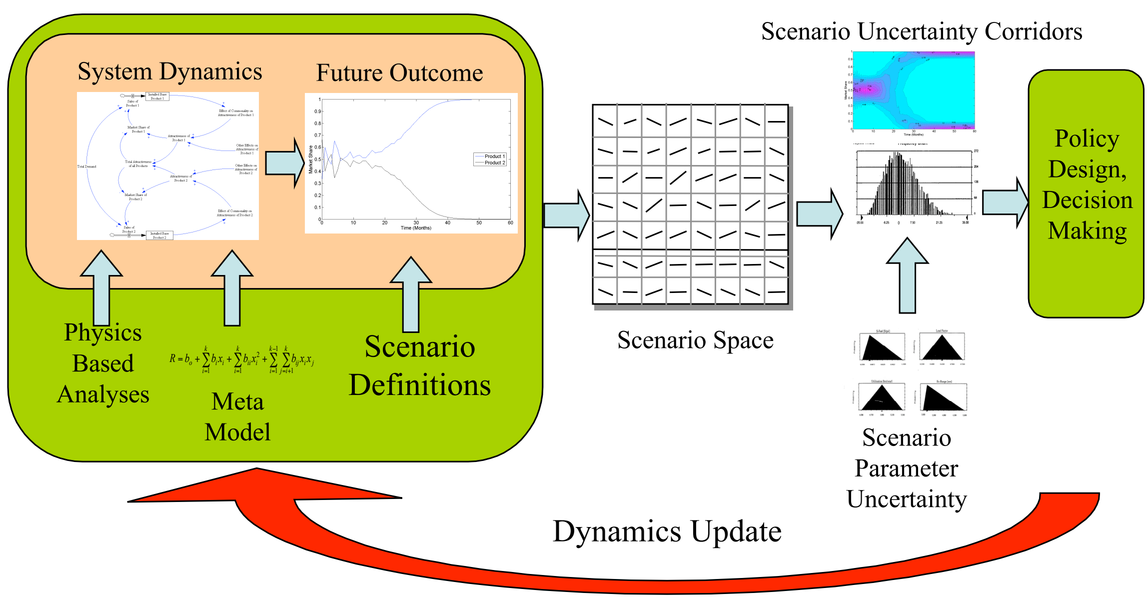


Figure 19: Overview of the Proposed Process

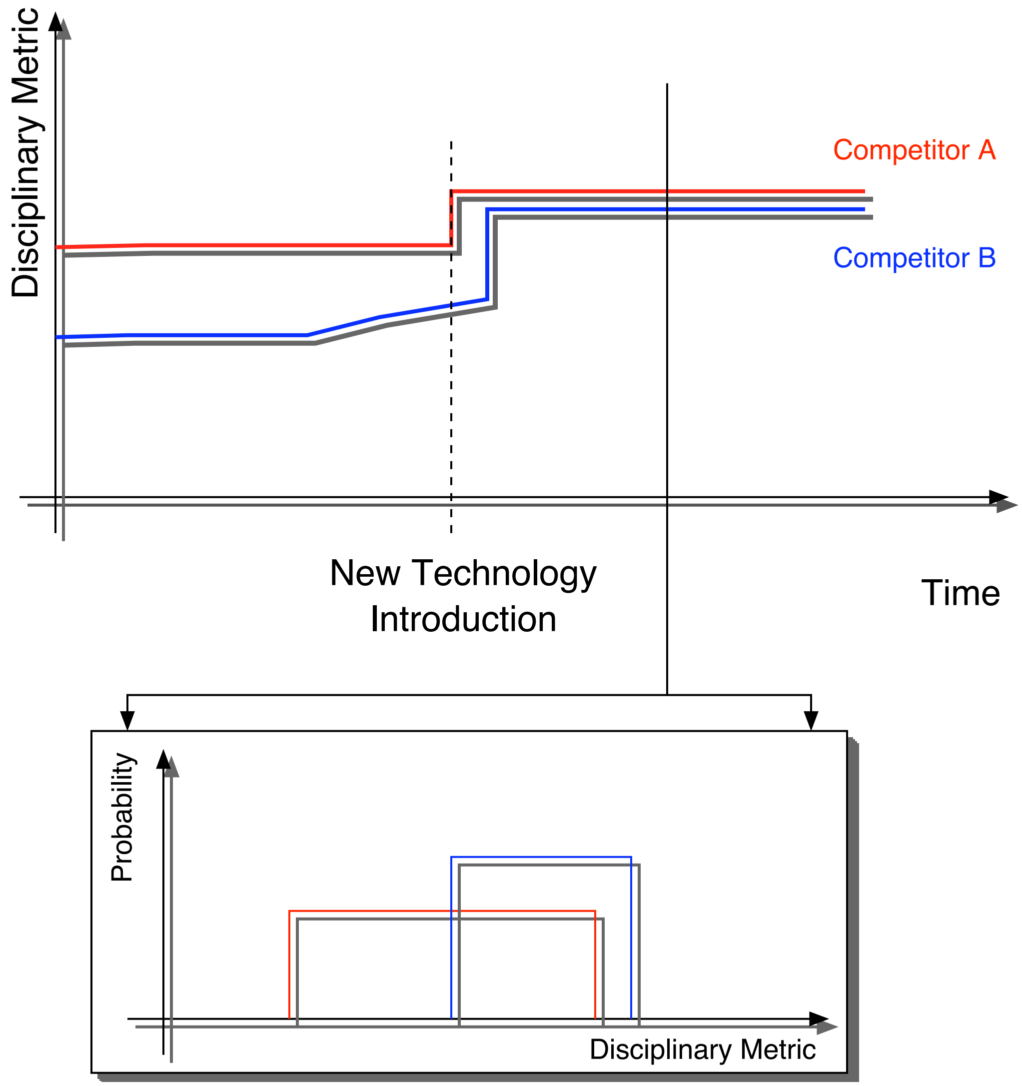


Figure 20: Scenario Definition for New Technology Introduction

However, Competitor B is late to market with the new technology. The improvement on product B is, however, much more profound than and product A, bringing product B very close in this specific disciplinary performance to product A. Furthermore, Competitor B was able to derive some small synergies with new technology development that translated into early and incremental improvement of the existing old technology product. Additionally, when treated probabilistically, each of the expected performances at any given time can be defined as a time dependent probability distribution, shown in the bottom part of Figure 20 as a slice at a specific time. This shows that Competitor B expects to be able to develop the new technology with much higher confidence in a much narrower confidence interval than Competitor A. This means that Competitor A is taking a higher risk by being first to market with a new technology. Each of the pertinent technologies and affected metric require such a definition. This combined with definitions for external factors such as oil price, growth, and inflation with uncertainty definitions as detailed earlier form a scenario definition. This scenario can then be explored with the process detailed before.

The system dynamics model then will translate this scenario definition into an explorable scenario for all parameters of interest. Figure 21 shows such an output deterministically and probabilistically. Shown here are the initially increased sales after the new technology introduction as well as the possible advantage Competitor A has gained by potentially almost eliminating a weakness in the product competitiveness and is therefore able to outsell Competitor B.

Finally, once a decision has been made the system dynamics model created should not be discarded. It rather should be retained and then re-calibrated and updated as necessary as indicated by the feedback arrow at the bottom of Figure 19. Forecasting is an inherently uncertain process and therefore contains unknown uncertainty. The proposed process only contains known uncertainty, that is uncertainty generated by the inherent variability of known model parameters and model elements. Therefore,

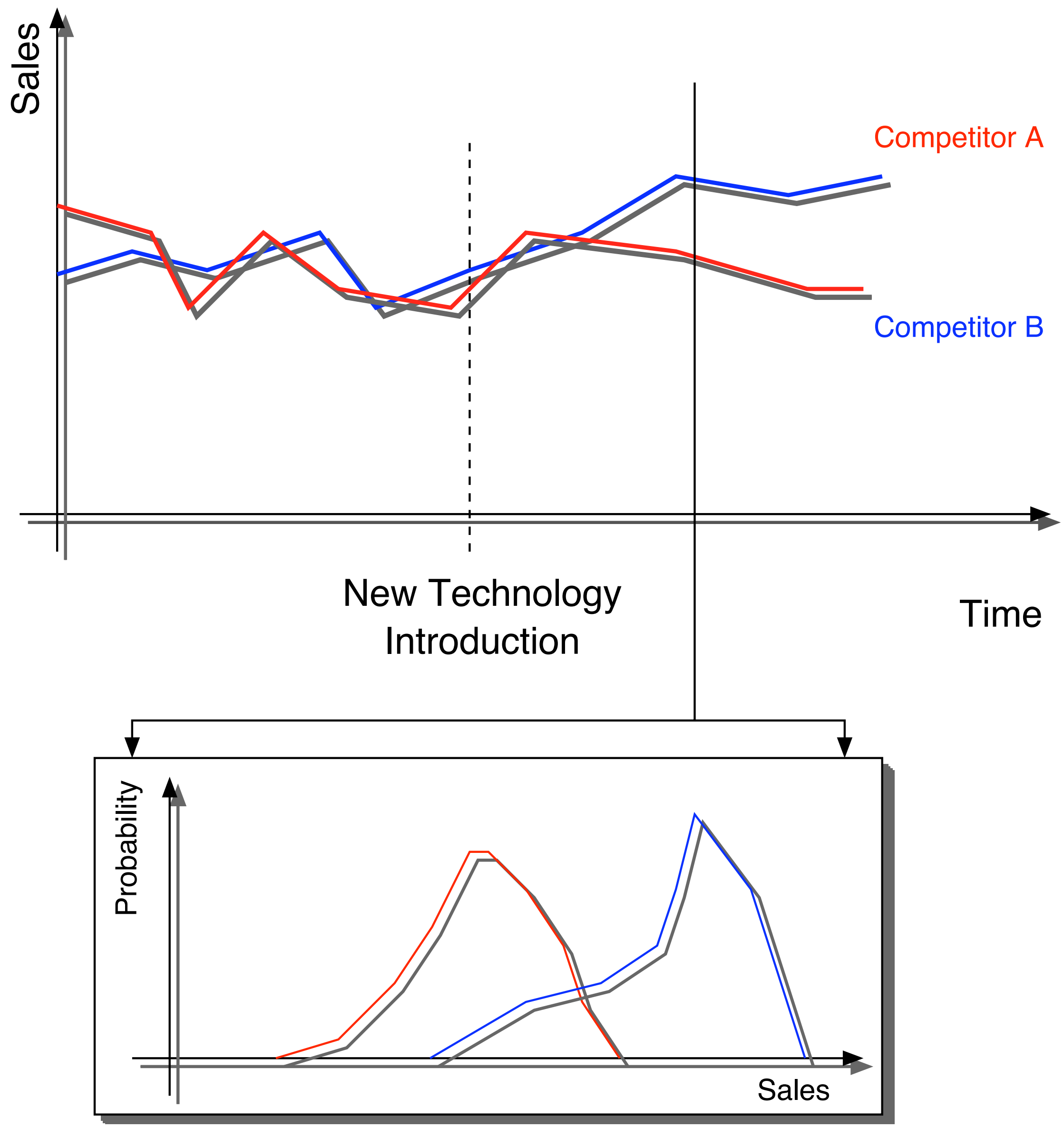


Figure 21: Scenario Outcome for New Technology Introduction

the accuracy of this kind of forecasting process proposed here is limited to a fixed structure of the underlying system. Should this structure change at any given time any of the results produced are no longer indicative of future system behavior. Once this occurs it is therefore necessary to return to the system model and add hereto-unknown elements to the system model and re-calibrate with the newly available historical data. This update should then again bring any future forecasts more into line with reality.

This update is probably best illustrated with an example from the aircraft market forecast [104]. In this study the authors initially created a system dynamics model of the worldwide aircraft market. This model was based on historical data from 1970 to 1987. The resulting model was then used to predict the aircraft market into the 1990s. The model and the actual outcomes were then compared again in 1994. The models predictions were mostly successful, especially because the model was able to accurately forecast a spike in aircraft orders in the early 1990s, which was completely missed by regular regression models. However, the model failed to accurately predict the order levels after the spike. The model's prediction was lower than the actual orders. Upon further analysis it was shown that this was due to a fundamental change in how airlines ordered aircraft, especially through aircraft leasing companies that the model originally had only incompletely accounted for. The model was updated and now included an extended section of the aircraft leasing business. Other effects updated included a much lower negative impact on airlines from congestion at airports. After re-calibration it was now able to match historical data including the increased aircraft order levels after the spike in the early 1990s. The forecasts generated in 1994 were then again compared to the actual market in 1998. The result was that the effect of early retirements again was the forecast was below actual behavior mainly due to forced early retirement of aircraft due to noise regulations. The model was yet again updated and re-calibrated. The final forecast period resulted in the model overestimating aircraft orders. The cause of that was the Asian economic crisis.

This example shows that there is at least short-term value in using a system dynamics model for forecasting for decision making. Even though one of the paradigms of system dynamics states that absolute forecasts should be avoided and rather models simply used to find robust policies that avoid unfavorable outcomes. Since then it has been shown that trying to produce an accurate forecast of a system can provide additional insights into the overall structure and its changes over time, especially when done repeatedly. Still, such forecasts have to be taken cautiously. The known uncertainty can be treated comprehensively with probabilistic methods. However, two places of unknown uncertainty can still occur in system dynamics models. One is the fundamental structure of the model. This structure can change over time and therefore produce invalid forecasts. The other source stems from the necessary model boundaries. The model inexplicably has to focus on a fairly narrow system and basis for the model. The model boundaries are then necessarily replaced by external data. This data implicitly assumes a certain structure of the world outside of the model. Furthermore, this outside structure is fixed and not dynamic in nature. If the relation of the model and elements outside of its scope changes in a significant manner, the overall model will produce false results due to the assumed outside structure. By updating the model structure and inclusion of previously external elements this can be remedied. Therefore, while it is possible to produce short-term forecasts with system dynamics models, they have to be studied carefully and still produce false results when the assumed structure of the model changes appreciably. However, the model structure itself and the resulting forecasts can still produce valuable guidance for decision-making and policy selection.

CHAPTER 6

VALIDATION

6.1 Demonstration of Design Integration

The first step here is to recreate the system dynamics competition model shown earlier in Figure 13 in MATLAB/Simulink. The model created is shown in 22.

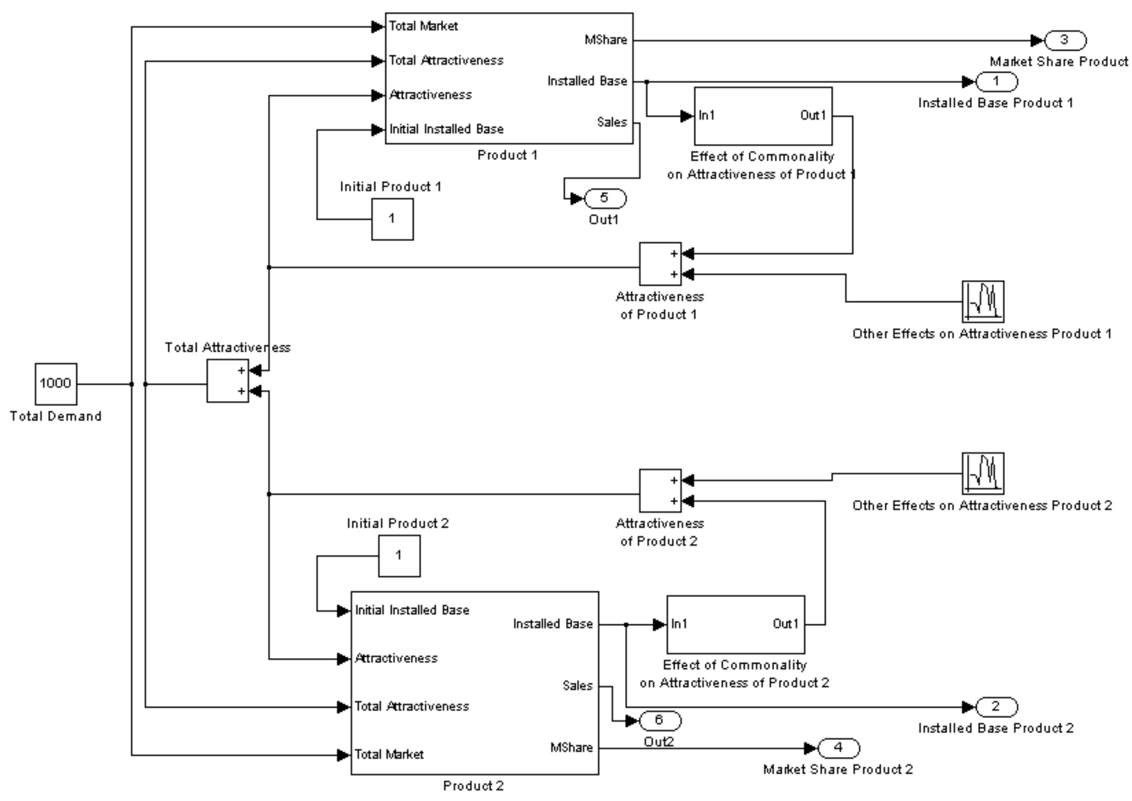


Figure 22: Top Level View of the Competition Model in Simulink

The difference in the graphical notation is obvious nonetheless both models are mathematically equivalent. This specific implementation directly defines a random number generator for the other effects on attractiveness for each product. Since Simulink's random number generator does not directly allow inputs for the seed and the standard deviation, the input parameters are omitted here as they are represented directly inside the random number generator. The same is true for the effect of commonality on attractiveness for each product. This model utilizes a custom Simulink block that here simply uses a direct MATLAB function with hard coded values for the threshold and sensitivity parameters in the exponential function used here. These two variables are then combined to form the attractiveness. This could be any form of multi attribute ranking system with various weightings but for now it is simply implemented as a simple sum of both values with no weights associated to either of them. The individual attractiveness of each product then feeds directly into another summation to compute the total attractiveness, which is then used to compute the market share fraction of the individual products. This calculation was for simplicity reasons moved into a custom block representing each block whose detail implementation is shown in Figure 23.

The custom block directly shows the computation of market share and then sales based on the fraction of the total market which in this case is given by a total demand forecast, which for now was simply left as a constant number per time unit. Sales are then fed into an integrator to compute the installed base. This particular implementation was chosen for ease of use with the Signal & Scope Manager in Simulink that allows direct tracking and graphing the time dependent behavior of variables without the modification of the model by having to insert scopes into the model as needed. This meant much simplified trouble shooting and debugging during the process of model implementation in Simulink.

The integration of analysis codes or meta models as described earlier into a Simulink model such as this involves the use of embedded MATLAB functions. These embedded functions allow the direct integration of MATLAB code directly

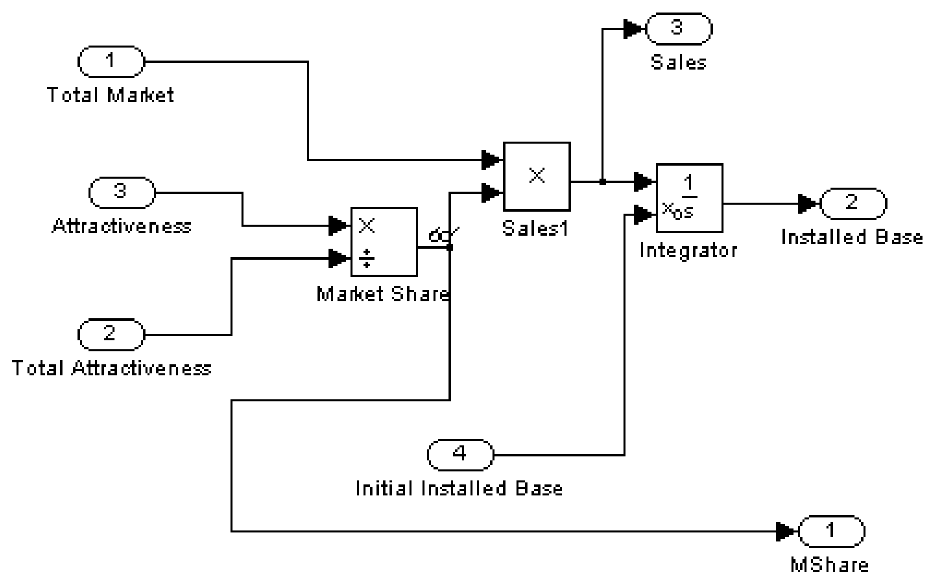


Figure 23: Detail View of the Installed Base Stock in Simulink

into Simulink. There are however several drawbacks associated with them. First, these functions do only exist inside the Simulink model and not outside. This means that testing has to take place either inside the Simulink model or in a separate copy outside of Simulink. Furthermore, these functions are first translated to C code when Simulink embedds them into the model. This has various consequences. The numerical behavior might not always be consistent with MATLAB especially when rounding. Additionally, this also means that the mathematical functions supported in embedded functions are much reduced when compared to regular MATLAB functions. This can create additional problems especially when the debugging took place outside of Simulink and then as a result some functions are not supported when the function is moved into the Simulink model. Finally, the embedded functions are interpreted at runtime as needed and therefore are not as fast as compiled code.

Due to the associated drawbacks it is only of limited utility to use embedded functions for the integration of analysis codes, especially due to the slower execution. Instead, Simulink offers other methods of integrating external functions into a model. This is done by way of S-Functions. These functions are compiled code in MEX format that follow a specific programming interface that allow Simulink to dynamically link them into MATLAB when needed. MEX files are MATLAB's interface to precompiled code created in other languages such as C/C++, Fortran or others. This has the particular advantage that this code executes very fast, even when compared to MATLAB. It is, however, not appropriate for all applications due to the much lower level programming required and subsequent increased development time. S-Functions can also be created from MATLAB code directly. The wrapper will then execute flops with the modified input file and parse the output file for variables of interest. The wrapper function then returns an array of these variable back to the Simulink environment.

The system dynamics model in Simulink is where the selected differential equation solver will call the wrapper repeatedly to obtain the values of the function output and also to calculate numerical derivatives as required by the selected solver.

The overall model shown in Figure 22 also has a number of output sink elements connected to various signal lines in the model. The purpose of these is to provide direct access to the variables represented by the signal lines in MATLAB for further analysis. For this purpose a wrapper is required to be written in MATLAB. This wrapper should automatically open a Simulink model and then replaces certain parameter values as needed. After that it should execute the Simulink model and collects the time, state, and output vectors. Due to the unknown size of time steps caused by potentially using variable time step solvers in Simulink, a new set of time and output vectors have to be created by interpolating the existing data between defined time steps. Alternatively, one can insure to simply use fixed time step solvers with time steps matching the analysis. The drawback of that is that fixed time step solvers are very commonly slower in solving sets of differential equations. Finally, the wrapper allows the output or solution vectors to be analyzed further or used in creating graphs and plots or utilized further for surrogate modeling as proposed.

A preliminary output of the process described is shown in Figure 24 and 25. The underlying data is based on a unmodified competition model implemented as described earlier. For simplicity the attributes influencing the product attractiveness have been assumed to be equal. The underlying time scale used was a month with the simulated time period stretching over 6 years or 72 months. Also the total market size in units per month along with the exponent in the commonality effect on product attractiveness were chosen appropriately to result in lock in after about half of the simulated time period or 36 months. The random number generator placed as the other effects on product attractiveness produce a new random number for each time step in the solver. The wrapper also assists in varying other parameters such as the initial installed base and the seed of the random number generators. However, one has to distinguish between randomly chosen initial conditions and random numbers created at each time step in the solver. Since the random number generator blocks used here have no correlation between the numbers created at each successive time step, they simply produce white noise. They have to be replaced with custom random number generators as needed for certain kinds of external factors such as for example oil prices.

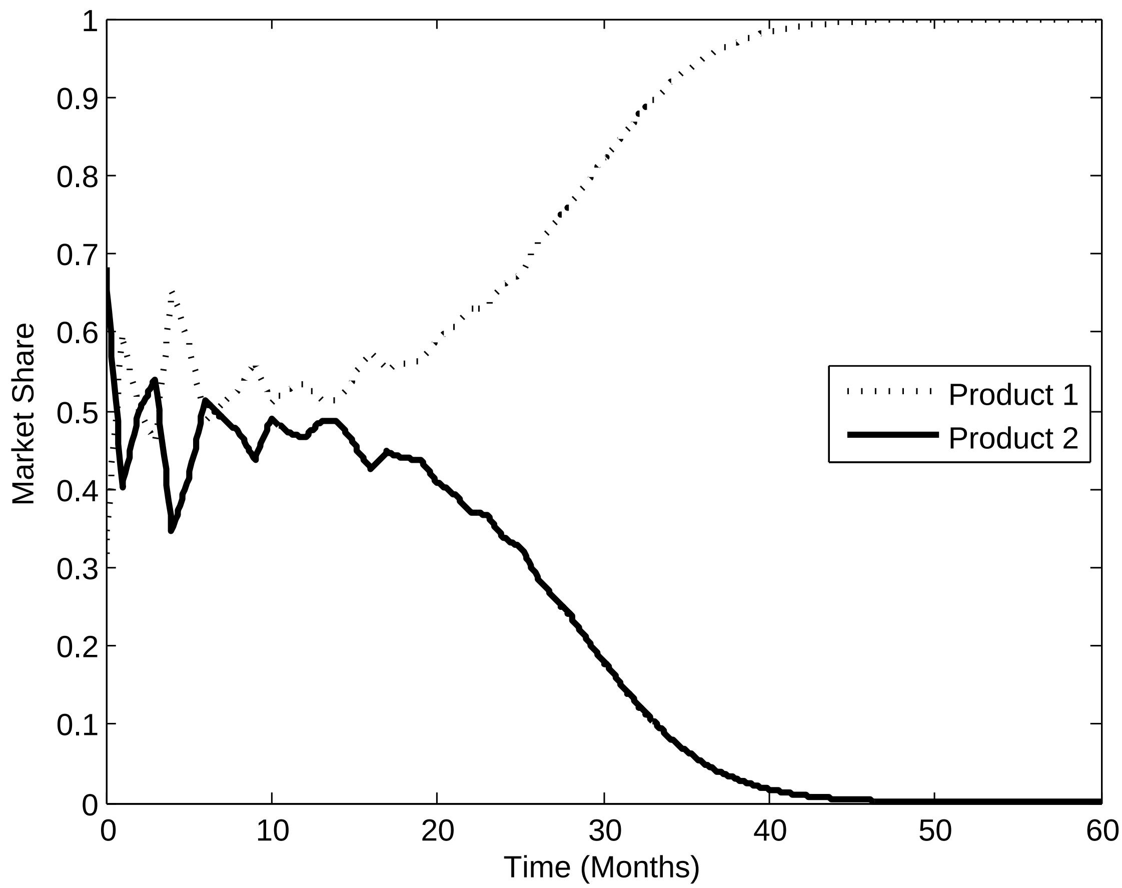


Figure 24: Deterministic Result

Figure 24 shows the result of a simple single run. The output focused on here is the product market share. One interesting item to note is that Product 2 initially starts out with the highest market share. Due to the successive random variations of the other effects on product attractiveness, however, Product 1 gains a slight advantage in installed base. This small advantage then produces a firm lock-in once the commonality effect starts to strongly dominate the product attractiveness. This means here that Product 1 goes on to dominate the market whereas Product 2 is essentially driven out of the market. What this illustrates is the importance of early market share that then goes on to create a larger installed base which in turn drives a large part of the product attractiveness in a mature market. Therefore it is crucially important to explore policies and scenarios that allow a certain competitor to gain this advantage.

This means that the model has to be extended to account for certain sets of scenarios. One example of such a scenario would be that one competitor lowers the price of its product and thereby increase the relative attractiveness potentially leading to increased sales and hopefully dominating the market long term. This means that new model elements and variables such as price discount and discount timing have to be introduced to the model. These and other policy levers then allow detailed analysis of favorable policies and their cost and effectiveness.

The competitive market example currently focuses on a fairly high volume market with a strong commonality effect on product attractiveness. This is not necessarily the case in the commercial aircraft market or at least likely only present in a weaker form. This means that the existing model will have to be re-calibrated to specific data on the aircraft market as outlined before. This also involves a more detailed model of the other effects currently only represented by a random number generator. The model can then also be extended such that products leave the installed base after they have outlived their useful life. This is readily implemented by introducing an outflow to the installed base regulated by measures such as life cycle duration of a specific aircraft, or more specifically a statistical distribution thereof.

Figure 25 shows the result of repeated runs of the same model as before. This time, however, the seed of the random number generators used for the other effects on product attractiveness have been changed every run to produce a new set of random numbers every run instead of the repeatable pseudo random numbers produced with a fixed seed. The resulting output in this case the market share of Product 1 at every time step was then divided into one hundred bins to produce a histogram of all runs at every time step. This number of bins was chosen so that a sufficient resolution for the ordinate axis. Logically it follows that a sufficiently greater number of overall runs must take place such as to provide sufficient resolution for all histograms. The number of overall runs was chosen to be one thousand due to the rapid increase in run time. A set of one thousand runs was shown to take approximately 25 minutes on a standard desktop computer.

The contour plot of probabilities for market share of Product 1 versus time shows some very distinct features. The initial noisiness in the first months is due to the high variability of the product attractiveness introduced by the fairly wide distribution placed on the other effects. However, after about 15 months the distribution spreads

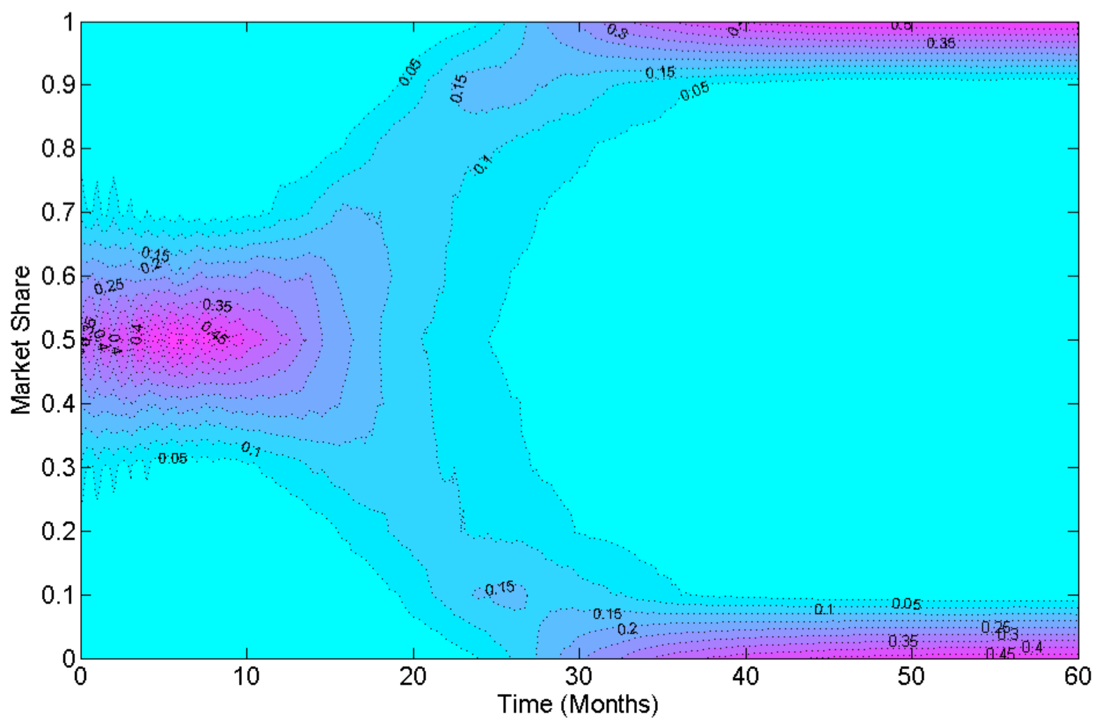


Figure 25: Probabilistic Contour of Market Share

out significantly and then settles in a final state where there is about 50% probability of either completely dominating the market or 50% probability of loosing all but a minor market share. This is an important feature of this model because it was setup to be symmetric with respect to identical products. Nevertheless, one product will go one to dominate the market even with both being equal. Once this model has been extended to allow for policy choices and integrated into a meta model, it will be of great interest to explore what policies result in a positive outcome and also the path dependency thereof.

The drawback of this method of integration is that even without calling FLOPS directly the execution is very slow and not interactive at all. Just the simple baseline of the competition model with identical products takes on the order of half an hour execution time. Additionally any interactivity like in the Unified Tradeoff Environment (UTE) is not only not available in MATLAB/Simulink and would therefore have to be developed with some effort, but the execution time is unacceptable for the envisioned environment.

6.2 Visualisation of Model Output

Such an environment is described next. This effort of a unified and interactive environment is of a great advantage. Therefore, a notional description of such an environment was developed. Termed extended Unified Tradeoff Environment (UTE), it is shown in Figure 26.

This environment includes the standard prediction profiler known from the unified tradeoff environment. However, it adds an additional axis of time shown on the left to the environment. Furthermore, it includes definitions of each of the input variables versus time shown deterministically on the top left as well as a probabilistic definition above the prediction profiler. The idea of this is that the right side depicts a time slice

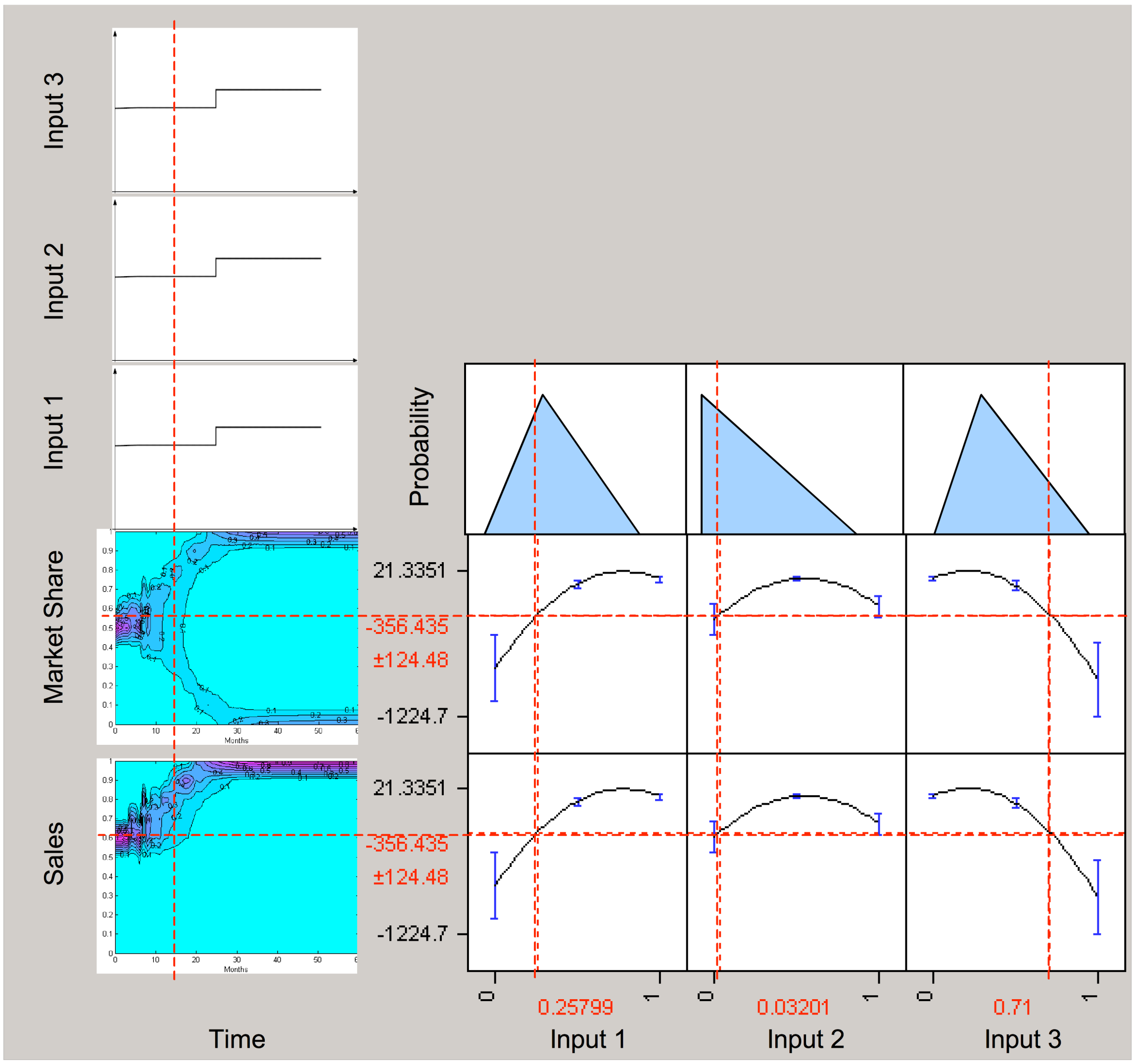


Figure 26: Extended Unified Tradeoff Environment

of the entire system at the time selected on the left side. The entire environment is envisioned to be interactive where the input and output slide bar lines are moveable and update the entire display in real time. This will be able to show the changes taking place when one or more of the input variables are changed and its effect on the future probabilistic distributions of the output variables in time shown in the lower left. Alternatively, it is also possible to show the change in input and output variables changing when the time axis slide bar is moved.

This entire environment will be able to serve as a unified environment enabling the direct visualization of the outputs of a system dynamics model such as the one defined previously and its implications on not only the design decision as implied by the input variables but also on the figures of merit such as market success. The integration of time variance into this is an important factor that creates the link between a system dynamics model and the decision environment as shown.

6.3 Example Application

This section focuses on the application of the described methodology on an example application. This application will be the wide body aircraft market that is currently a very relevant issue with the on-going competition in this market between Boeing and Airbus.

6.3.1 Wide Body Aircraft Market

The wide body aircraft market has been dominated since the late 1970s by Boeing with its 767 aircraft. It was not until the late 1980s that Airbus emerged as a competitor and launched the Airbus A330 program. Since then the 330 turned out to be a very successful program for Airbus. One special feature of the A330 is the commonality of many parts with the A340. The wing and some fuselage sections are mostly identical between both programs. This had the effect of Airbus being able to leverage economies of scale and learning curve effects much better. This was especially important in the light of Boeing's 767 sales over an additional decade previously that enabled Boeing to be very competitive at least on the manufacturing cost side.

The recent developments in the form of Boeings launch of the 787 program and the recent successes in pre-orders thereof add a current interest in the question of who will be able to dominate this market. This is even more relevant in the light of Airbus' A350 launch and the subsequent problems and delays.

All four programs are not just simply single aircraft. They all consist of a variety of variants. These variants range from shorter versions to longer higher capacity versions to long range variants with additional fuel tanks and reinforced structure for a higher maximum take-off gross weight (MTOW). Additionally, all four programs do not cover exactly the same seat classes and ranges. In the interest of simplicity and trying to compare similar aircraft it is therefore a prudent approach to limit this example to a very specific seat class. The problem with this is that even the same aircraft can be configured differently with respect to seating and interior layout. Furthermore, the minimum and maximum seat ranges do not necessarily match either.

The solution to this problem is to use the closest matches for a specific cabin layout. This is the case for the three class default seating arrangement for the Boeing 767-400ER and the Airbus A330-200. What makes this example further more attractive is the overlap of the time during which both aircraft were offered.

This is of value here because existing Boeing and Airbus sales numbers are publicly available [136, 137]. This data can serve as the foundation to calibrate the proposed competition model. Additionally, pre-order numbers for the Boeing 787 and the Airbus 350 are also available. This should enable the same model - once calibrated - to explore future market scenarios and what-if games for this currently relevant environment. The results of these forecasts then should yield valuable information about each product's strategy concerning technology, time to market and pricing that yield a success in the market.

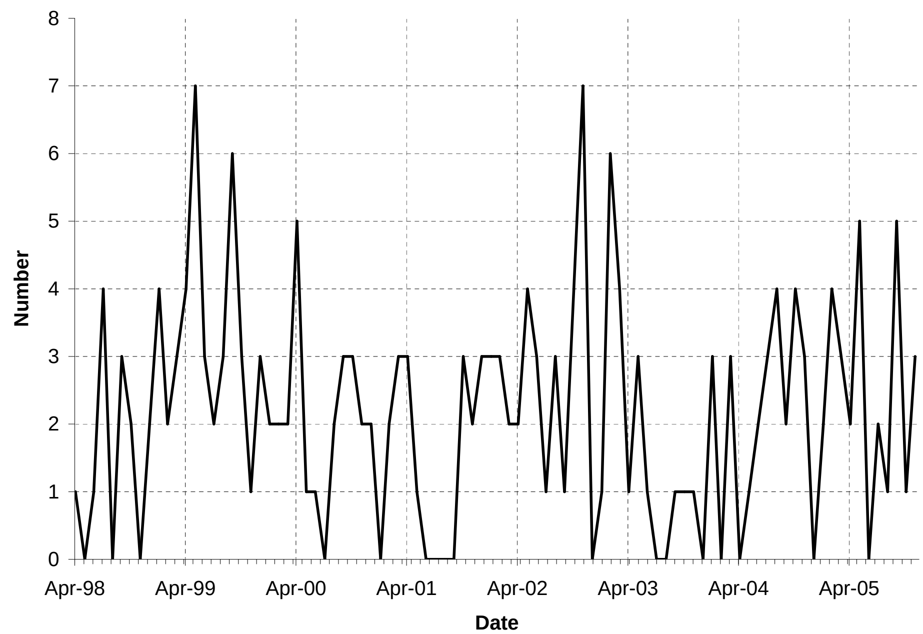


Figure 27: Monthly Sales of the Airbus A330-200

The market data shown in Figures 27 and 28 represents the monthly sales of each of the respective aircraft. This along with the monthly market share data shown in Figures 29 and 30 is the data set that will be used to calibrate and validate the system dynamics market model. It should be noted that the time range represents a 92 month time frame between April 1998 and December 2005.

The other important parameters used for calibration and validation will be the total market share and sales as shown in table 2.

The other important aspect of the market model is that the demand is assumed to be an external driver. This means that the model is driven by the external data

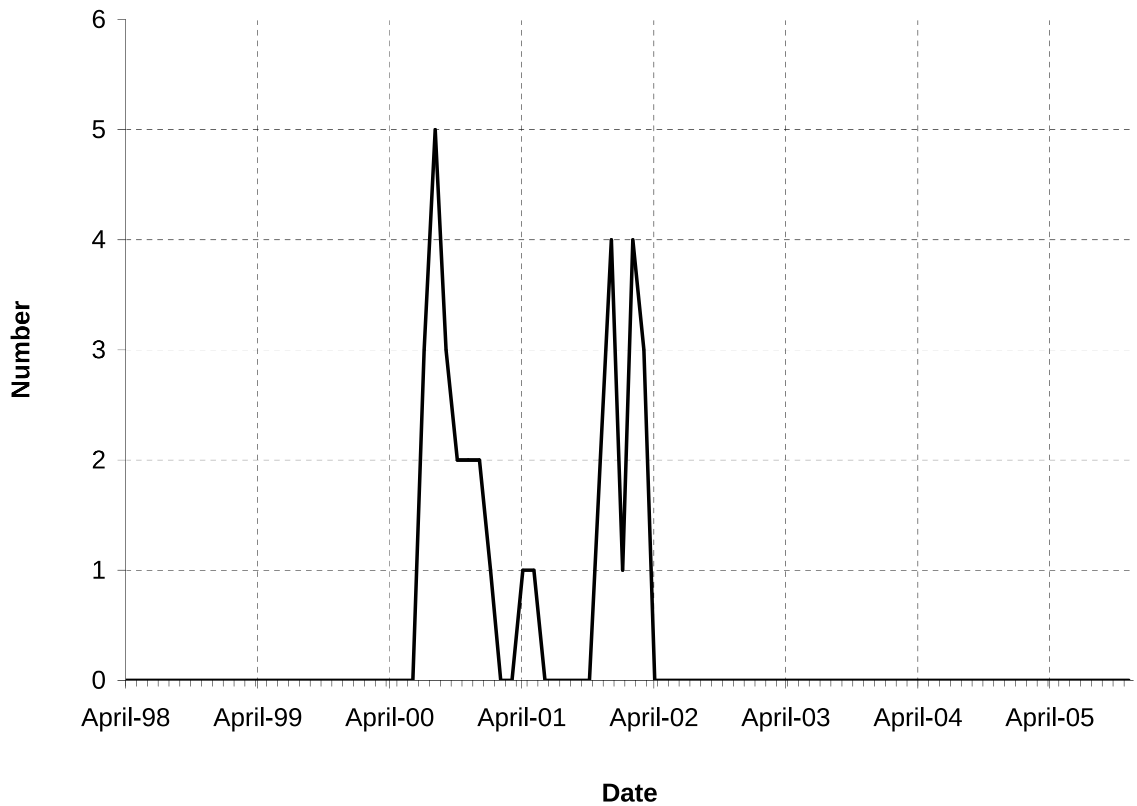


Figure 28: Monthly Sales of the Boeing 767-400ER

Table 2: List of Total Market Values

Total Market Size	236 Aircraft
Overall Airbus A330-200 Market Share	85.6%
Overall Boeing 767-400ER Market Share	14.4%
Total Airbus A330-200 Sales	202 Aircraft
Total Boeing 767-400ER Sales	34 Aircraft

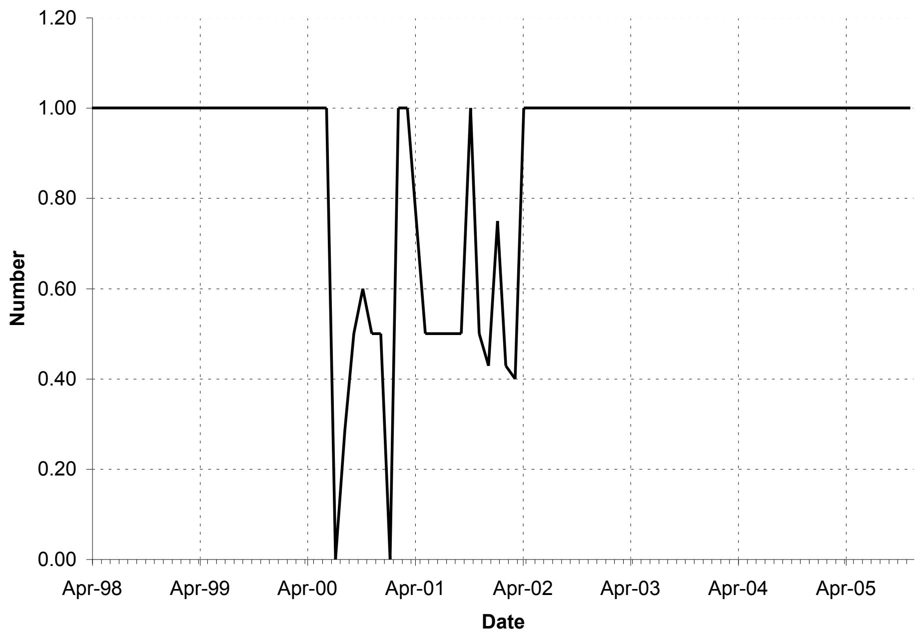


Figure 29: Monthly Market Share of the Airbus A330-200

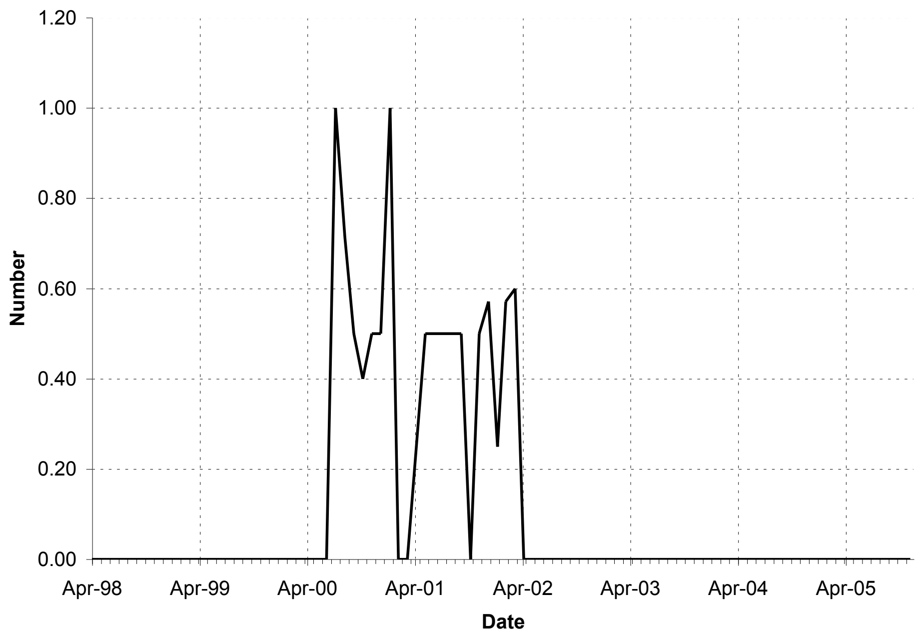


Figure 30: Monthly Market Share of the Boeing 767-400ER

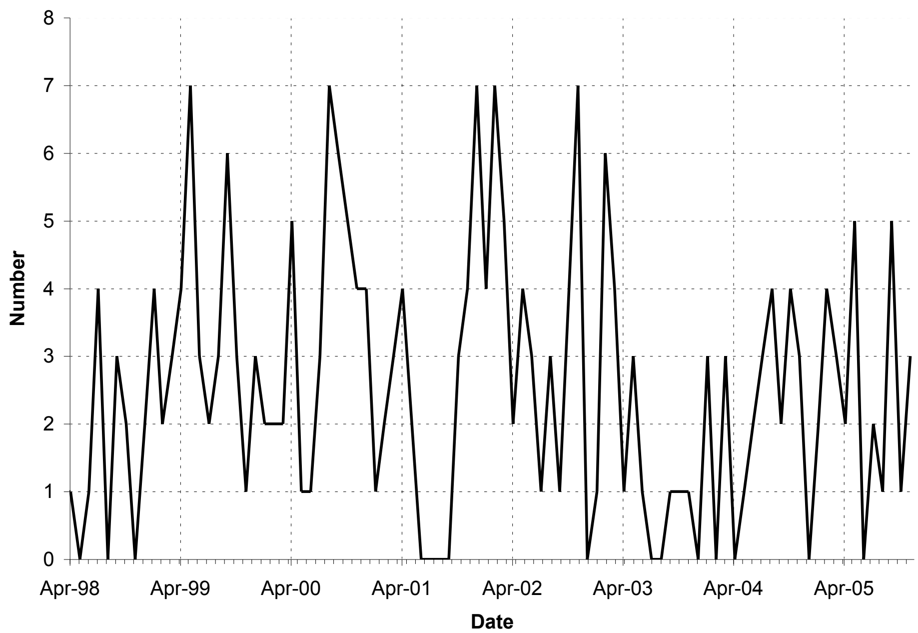


Figure 31: Monthly Total Sales

set shown in Figure 31. This is due to the previously shown work in system dynamics modeling of the total aircraft market. Such a model could then be used to drive the market model proposed here. The model then ideally should be able to predict the individual sales and market share of each of the respective aircraft.

6.3.2 Aircraft Definition

It is now time to define the individual aircraft that were selected. The aircraft were both calibrated with FLOPS. The final input files are shown in Appendices A.1 and A.2. The engine deck used for both aircraft is a NASA Lewis generated deck for a generic 225 seat commercial aircraft in the same thrust class used by both the Airbus A330-200 and the Boeing 767-400ER. Small adjustments had to be made by using the scaling parameter in FLOPS. The data used to calibrate both aircraft was taken from Jane's[138]. Additional data about the range payload information and weights was taken from the airport planning documents[139, 140]. The pricing information was taken from the aircraft values analysis companies and instead of the list price represents an estimate of actual paid prices by the airlines. It should be noted that the actual prices can vary widely and are subject to individual negotiations. However, in general the actual prices represent a discount of around 35% from the official list prices, which is consistent with industry expert estimates.

6.3.3 Calibration

Table 3 shows the most important parameters used in calibrating the FLOPS models. This is not comprehensive due to many small details in the geometry such as the tail parameters etc. However, the parameters shown are the most important. Furthermore the table shows the actual model outputs after calibration. As can be seen, the accuracy is quite good, since the error in all cases stays below one percent.

The Airbus A330-200 was relatively easy to calibrate against and only required

Table 3: Aircraft Calibration Parameters

Reference Parameter [138, 139, 140]	A330-200	767-400ER
Number of Vehicles Produced	830	1049
Passengers	253	243
Fuselage Weight Factor	1.038	1.2
Fuel Flow Adjustment Factor	1.032	0.85
Design Range	6650	5645
Aspect Ratio	10.1	9.3
Taper Ratio	0.267	0.23
Sweep	31.5	31.5
Payload (lbs)	56000	52100
Takeoff Gross Weight Design Mission (lbs)	513675	438000
Operating Weight Empty (lbs)	233818	227400
Aircraft Price ($Million, 2005)	98.2	105.5
Zero Fuel Weight (lbs)	291818	279500
Model Values (FLOPS Results)		
Takeoff Gross Weight Design Mission (lbs)	513384.7	438223.7
Operating Weight Empty (lbs)	233940.3	226791.4
Aircraft Price ($Million, 2005)	98.338	105.541
Zero Fuel Weight (lbs)	289853	280494
Error		
Takeoff Gross Weight Design Mission (lbs)	0.06%	0.05%
Operating Weight Empty (lbs)	0.05%	0.27%
Aircraft Price ($Million, 2005)	0.14%	0.04%
Zero Fuel Weight (lbs)	0.67%	0.36%

minor adjustments in the weights and the fuel flow to match all the target weights. On the other hand the 767-400ER required significant adjustment in the fuel use and the weight. The fuel use had to be lowered by 15% and the weight increased by 20% to match the actual aircraft data. This is potentially due to the fact that the engine used on the A330-200, the GE CF6-80E1A4, is a better match to the reference engine deck than the engine used on this particular version of the 767-400ER, the GE CF6-80C2B8. The weight discrepancy is potentially explained by the fact that the FLOPS weight estimating relations are matched better by the A330-200 than the 767-400ER, which might be due to the airframe potentially representing an older technology airframe being noticeably heavier.

One particular problem presented itself in the economics. The production runs of both variants were relatively small. ALCCA cannot handle small variant production runs. Therefore, the decision was made to base the economics on the entire production runs of the complete 767 and the A330/A340 programs. In the case of the A330 the A340 production numbers were also included because they represent significant synergies that exist due to the component sharing of both programs. Calibration was achieved by the modification of the learning curve parameters and the rate of return on investment for the manufacturer. To keep the learning curve in the accepted range of around 82% the rate of return had to be adjusted to only 5% for the A330-200 as compared to 10% for the 767-400ER. In conclusion, a quite good match was possible for both aircraft. It should also be noted that that required yield per revenue passenger mile showed the correct trends for the most common missions for the A330-200 and the 767-400ER, which are 3092nm and 1500nm respectively.

Furthermore, it should be noted that the economic parameters used to match the aircraft price were simply the rate of return of the manufacturer (RTRTN) and the primary learning curve factor (LEARN1). There are a much larger number of variables available that could also be used to equally achieve calibration. This is possible due to the fact that the actual numbers are either unknown to the author or very difficult to measure in reality and thus might even be unknown to the manufacturers themselves.

Table 4: List of Economic Calibration Variables

ALCCA Variables	Description
RTRTN	Rate of return for the Manufacturer
API	Average annual inflation factor
FACI	Production facilities cost
LEARN1	Airframe learning curve factor for first lot
NV	Number of vehicles demanded
RE	Engineering labor rate
RT	Tooling labor rate
PRDYR	Number of years before production start
PVIRMAN	Percent of first unit cost savings due to virtual manufacturing
FMGT	Project management factor
RLOG	Logistics rate per man hour
RPMGT	Rate per management man hour

In addition to the variables listed in Table 4 there is a large number of complexity factors. These complexity factors are generally divided into groups by splitting the program into two distinct phases. The first phase is the research, development, test, and evaluation (RDT&E) phase. The second phase is the production phase. Thus the complexity factors exist for both phases. Furthermore, the complexity factors are then split into groups by aircraft component, such as fuselage or wing, which are then further split into complexity factors by material type such as aluminum, titanium, or composites.

These complexity factors are set in ALCCA to some default values that are generally accepted for certain components made out of specific materials. This means that there has to be a very specific reason to change them. This could be the case if it is known that a particular manufacturer has some development or production technology that allows the simplification — or inversely does not have the technology

— of a particular components development or production. Since this is beyond the scope of this work and this information is not publicly available easily, the complexity factors were left at their default settings. This is even more the important, because in actuality it would be the difference between both manufacturers that matters the most. This information about which particular component is easier to develop or produce for a particular manufacturer versus the competition can be a very difficult assessment. Therefore, the large number of complexity factors is not considered here and are simply left at their default settings.

Another economic factor that could play a role is the material cost. In this case material cost is expressed as a factor compared to man-hours required to develop or produce a certain component. This means that there is no direct materials cost adjustment possible only a multiplier. Furthermore, material cost should not differ significantly between manufacturers, at least assuming an open unrestricted global market in equilibrium. Therefore, this was also not considered in this analysis.

Table 5: Economic Calibration Variables for the 767-400ER

ALCCA Variables	Default	Minimum	Maximum
RTRTN	10	0	15
LEARN1	82.5	80	85
NV	1049	500	1500
RE	89.68	85	95
RT	54.68	50	60
PRDYR	5	2	10
PVIRMAN	0	0	0.2
FMGT	0.033	0.01	0.05
RLOG	82.29	75	85
RPMGT	94.95	90	100

There is a very extensive list of variables that could potentially be used to achieve the calibration of the ALCCA model to the existing aircraft. However, Table 5 shows a list of ones most likely to differ significantly and allowing the most variability.

To further explore this issue of achieving a calibrated cost model, a set of runs was performed. The variables shown in Table 5 were chosen at random in the intervals shown. The resulting set of data was then used to perform an effect-screening test in JMP, as statistical analysis software package, to determine the individual importance of each variable to the final aircraft price.

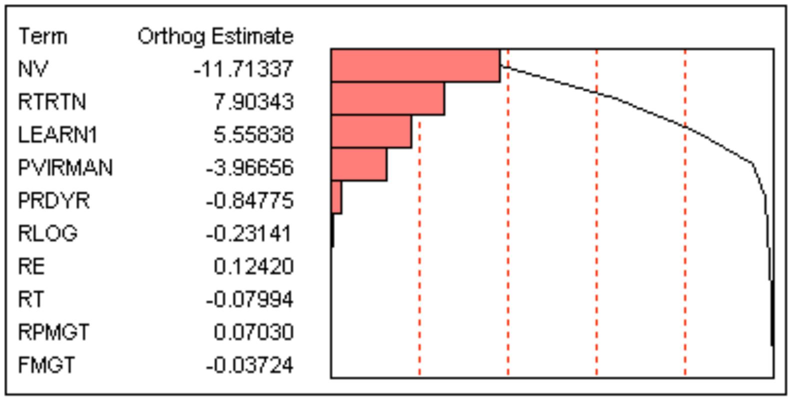


Figure 32: Pareto Plot of Transformed Estimates of Effects on Aircraft Price

The result of this analysis is shown in Figure 32. This shows that the most of the variability of the aircraft price is caused by the number of vehicles produced, the rate of return, the learning curve, and the percent of savings due to virtual manufacturing.

The same data used in the effect-screening test can then be used to create a surrogate model of the aircraft price as a function of the ten input variables shown earlier. The resulting surrogate model can then be used to create contour plots showing two particular input variables on the horizontal and vertical axes, while showing the known price as a contour line in the graph. This is shown in Figure 33.

Shown in this plot is the aircraft price with respect to the rate of return of the manufacturer and the initial learning curve. The line for the price represents the actual aircraft price. The hairlines represent the default settings used in the initial calibration. However, these hair lines can be moved along the constant actual aircraft price to reflect more realistic settings in the rate of return or learning curve as desired without altering the aircraft price output of the ALCCA model.

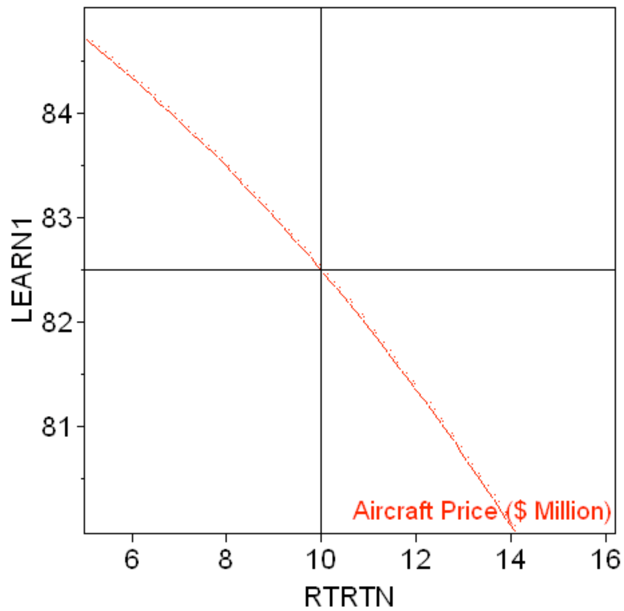


Figure 33: Rate of Return and Learning Curve Calibration Trade-off

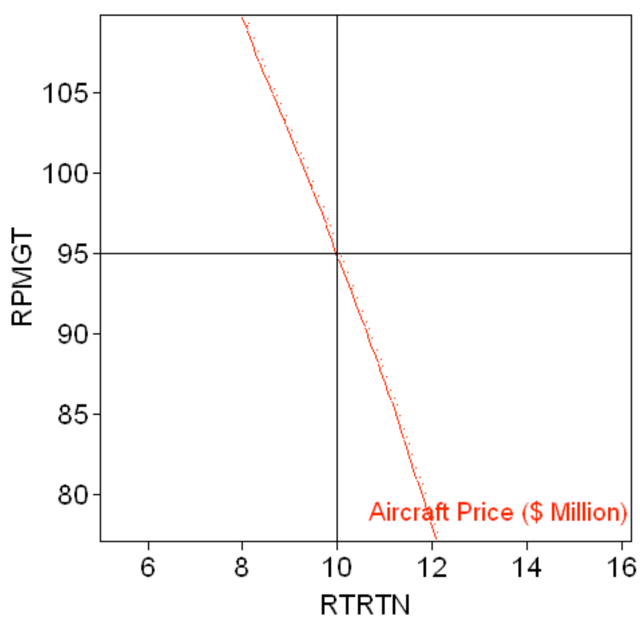


Figure 34: Rate of Return and Labor Rate Trade-off

Another example of the trade-off between labor rates — in this case the management labor rate — and the rate of return is shown in Figure 34. A similar graph can be shown for every particular combination of variables easily by selecting different variables as shown in Figure 35. The numbers shown are the default calibration settings and can be changed by moving the cross hairs in the contour plots as shown previously. The horizontal and vertical axis can be chosen as desired.

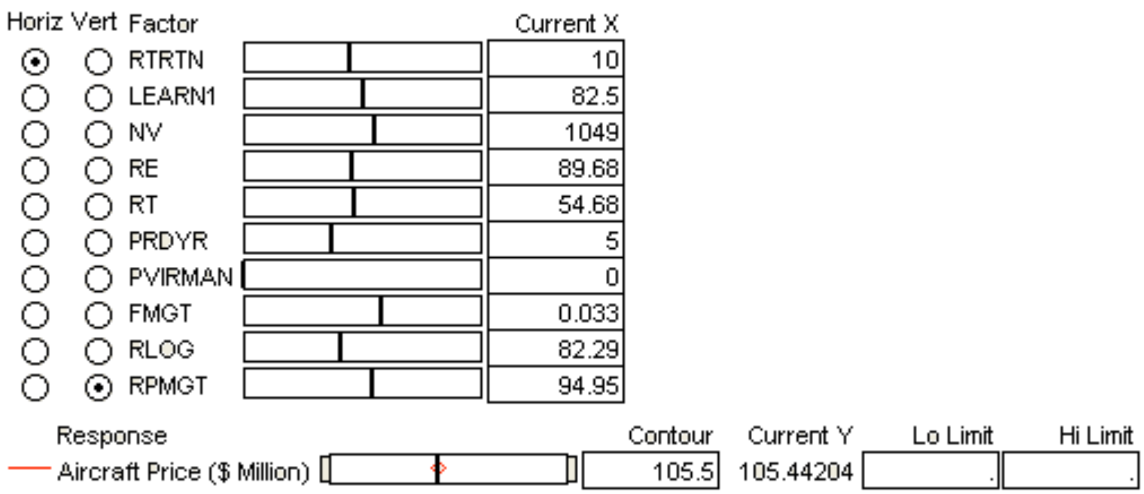


Figure 35: Overview of Contour Profiler Settings

This illustrates that no one particular setting of variables can be considered the correct one without much more in-depth knowledge of the development and manufacturing processes of each of the manufacturers. Therefore, these variables can be chosen as desired in necessary by the final user to achieve a match of the model outputs with the actual values.

6.3.4 Creation of Surrogate Models

These calibrated baselines were then used to create surrogate models with the help of the parsing tool shown in Appendix C.1. These runs were then imported into the neural network creation tool described earlier [26]. The total number of runs was set to 1000 because a representative design of experiments with eleven variables would normally contain 256 runs. This leaves enough room to use data for testing and validation of the surrogate model. The reason that a neural network was used here in place of a response surface was due to several factors.

Table 6: List of Surrogate Model Variables

Model Variables	330-200	767-400ER	min	max
RTRTN	5	10	0	15
NVEH	830	1049	500	1500
PAX	253	243	230	270
FUCOMP	0	0	0	1
CLTOM	2.3	2.3	2	3
FACT	1	1	0.8	1.2
DESRNG	6650	5645	5000	7000
AR	10.1	9.3	8	12
TR	0.267	0.23	0.2	0.3
SWEEP	31.5	31.5	30	35
COFL	0.865	0.865	0.5	7

The first factor is the fact that the surrogate model uses eleven variables. This means that custom designs have to be created, which can be time consuming. Furthermore, available eleven variable designs were tested, but the aliasing structure did not work with the particular setup of variables used here. Therefore, it would have been necessary to create a design from scratch while making sure that it still can produce good results. Additionally, the model verification with a set of random data would involve a significant additional effort. Even then a response surface equation can still produce an insufficient fit to the data due to non-quadratic effects or interactions.

Therefore, it was much less effort to use a neural network surrogate model. The design of experiments used was a random domain spanning set created with uniform random distributions on all variables over the entire range. The list of variables including the defaults and the ranges for the surrogate model are shown in table 6. Only four of the cases failed in FLOPS/ALCCA due to a relatively conservative choice of ranges.

The surrogate models were created with half of the runs to optimize the neural networks and half of the remainder to test the model and the other half of the remainder to validate the model. The results of this are shown in figures shown in Appendix B due to the number and size of the figures. The results shown were obtained with only five hidden nodes in the neural net. This low number worked well due to the fact that most of the data is linear or quadratic. A higher number of hidden nodes is not recommended due to the over fitting that will take place. A Levenberg-Marquardt training algorithm scheme was used for the neural network creation because it seemed to produce the most consistent results with good fits. However, due to the relative simplicity of the problem other techniques work well also and there is no significant difference in speed due to the relatively small number of variables.

The results shown in the appendix show an excellent fit with a small absolute and relative error with no underlying patterns in the residuals. Furthermore, the MFE and MRE are very small and the Testing and Validation R^2 is above 0.999. For further testing the Takeoff Gross Weight (TOGW) and the fuel weights were included on top of the two economic parameters of interest. This was done to ensure the physical correctness of the sizing results not just the economic results.

The next step was to verify the representativeness of the created surrogate models. This was accomplished by entering the surrogate models that were created into JMP and exploring the trends of changes in the prediction profiler.

The results of this are shown in Figure 36 for the Airbus 330-200 and in Figure 37 for the Boeing 767-400ER. As can be seen in both Figures, the trend lines for changes in design variables is consistent with expectations. For example, decreasing the fuel burn of the engine results in a visible drop in fuel weight on the aircraft and also a decrease in the required yield for the airline.

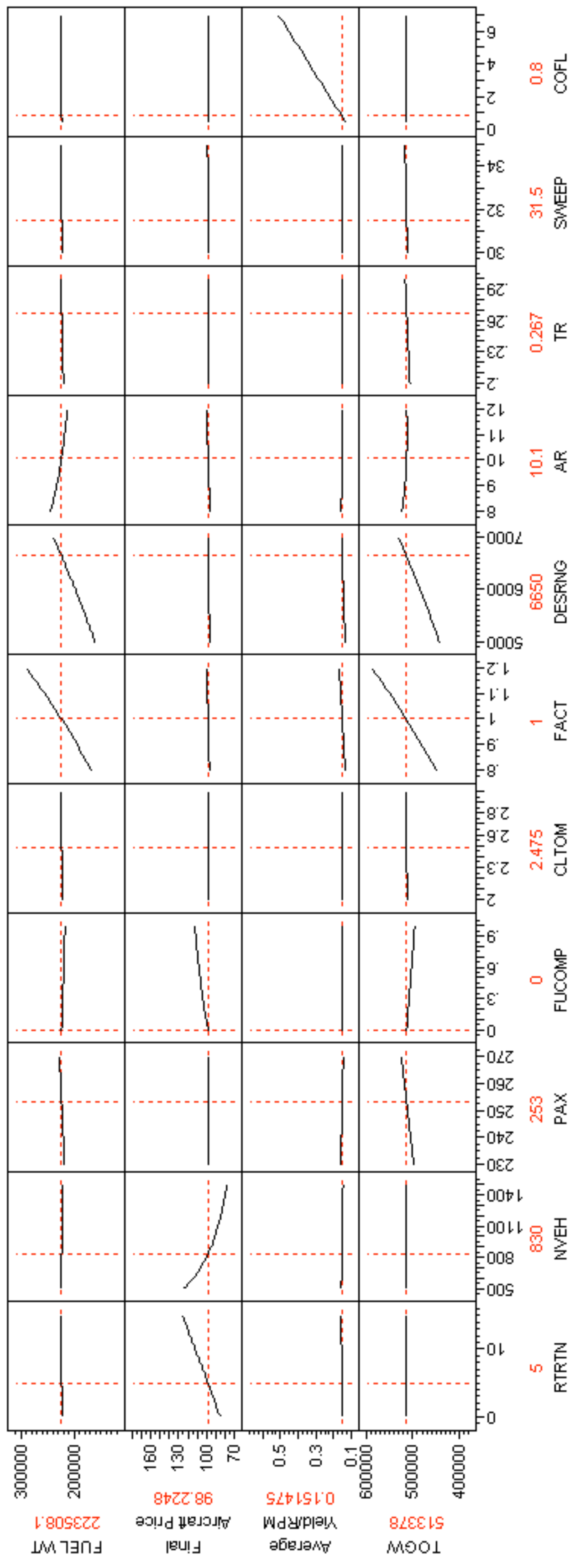


Figure 36: A330 Surrogate Model in Prediction Profiler

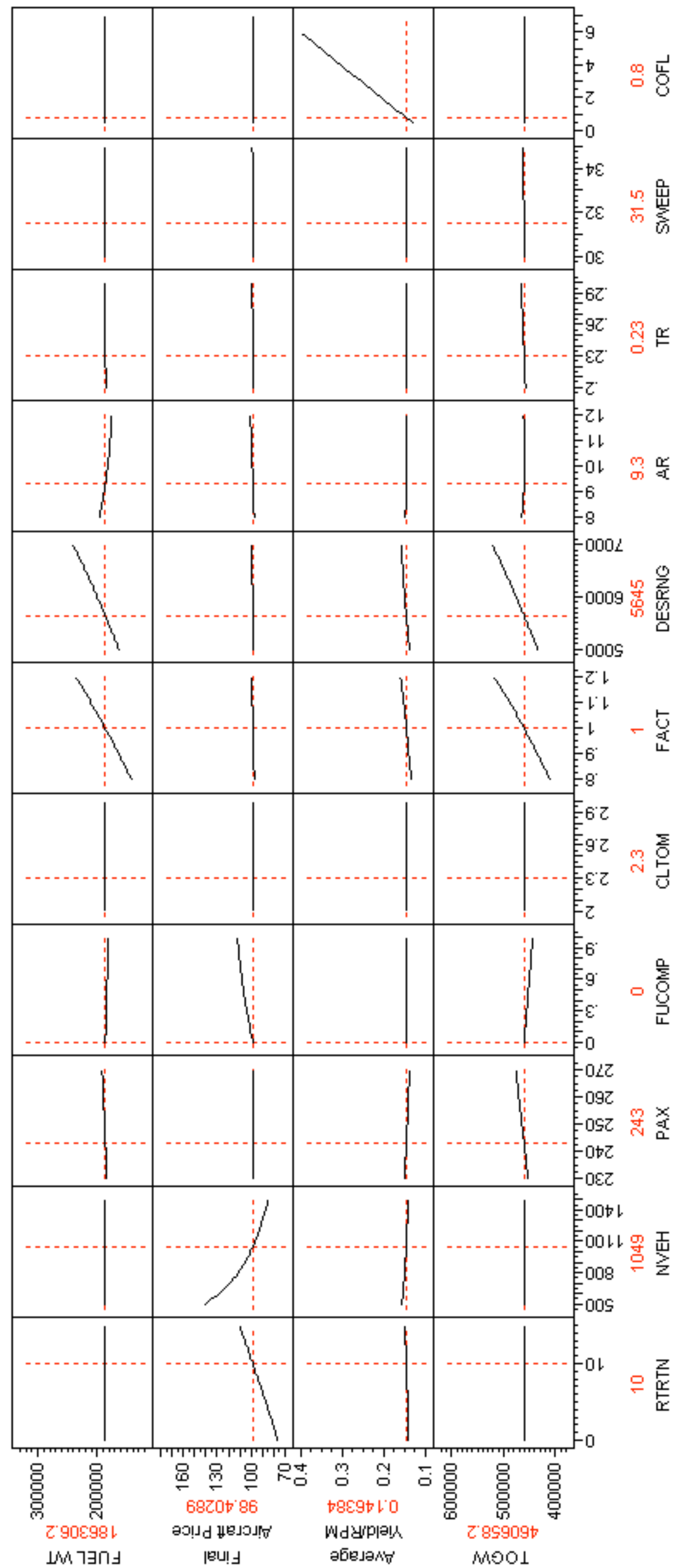


Figure 37: 767 Surrogate Model in Prediction Profiler

6.3.5 Market Model

Once these surrogate models were created, they were then inserted into the Vensim version of the system dynamics market model. This model consists of three views. The first view, which is the main view, contains the core of the system dynamics market model. This model is derived from the Polya process model described earlier in Section 4.2.1. The underlying model is directly based on this process model but is adjusted by using a modified formulation for the attractiveness. Specifically, a Multi-Attribute Decision Making (MADM) formulation known as the Overall Evaluation Criterion (OEC) is used to combine all relevant parameters into a single value used to determine market share. The flows tracking sales and the converters tracking the market share and the MADM formulation is shown in this view, which is shown in Figure 38.

The second view consists of the aircraft surrogate models. To be able to represent the aircraft in this system dynamics model, the surrogate model equations had to be copied into the Vensim equation editor. Thankfully, this was no problem because the default MATLAB notation output by BRAINN matches the Vensim equation format. However, the input variable had to be setup manually. This took some effort to setup, especially the default values and the minimum and maximum range limits. This view is shown in Figure 39.

The third view consists of the fuel price model, which is shown in 40. As described previously this is based directly on the pink noise model described by Sterman[74]. One particular feature of this is the implementation of various shape functions to allow the pink noise to be shaped in various ways. This includes a pulse function, a ramp function, and a sine function. Each of these shape function has a number of parameters associated with it, including the magnitude and start and end times or in the case of the sine function the amplitude and period. For this model the ramp

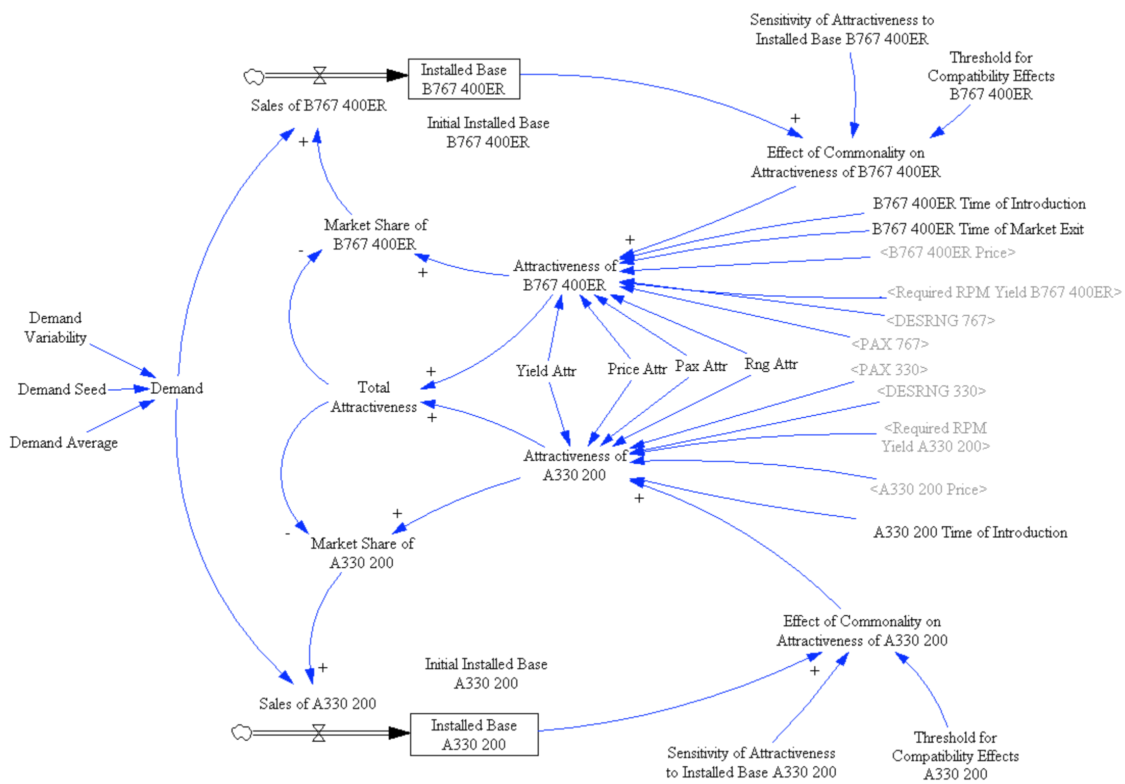


Figure 38: Main View of the Market Model in Vensim

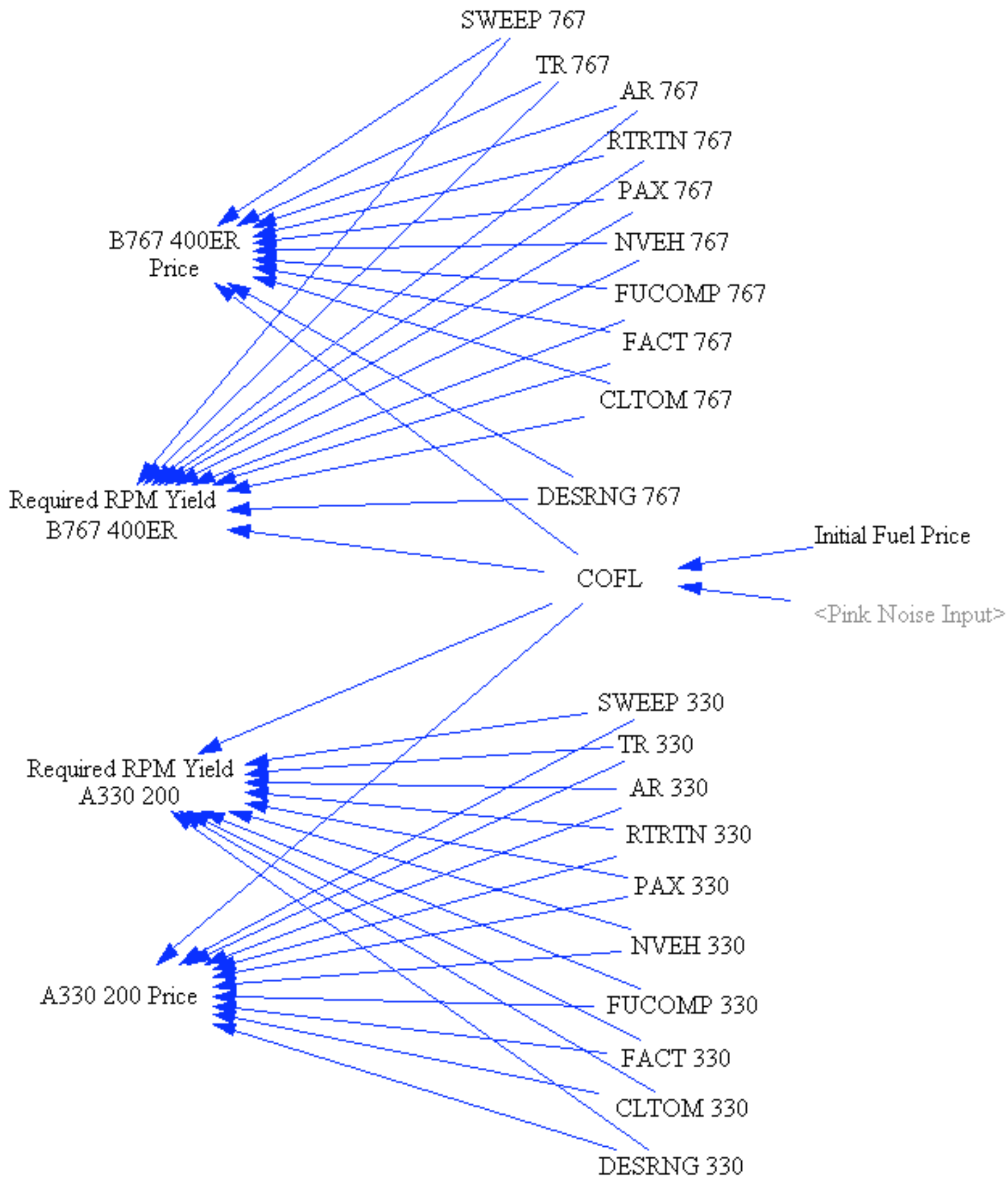


Figure 39: Aircraft Model View of the Market Model in Vensim

function is the most pertinent one since it allows the definition of a slow ramp in the fuel price trend. This means that it is possible to represent a slow creep upwards in the trend in fuel price.

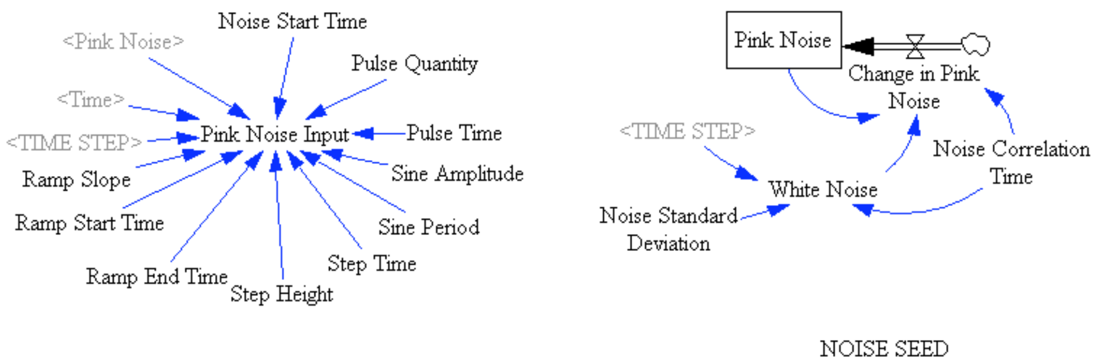


Figure 40: Fuel Price Model View of the Market Model in Vensim

The U.S. Department of Energy's Energy Information Administration (EIA) publishes the daily spot market fuel prices[141]. The current available data ranges from April 1990 to October 2005. As shown in Figure 41, the price increased from 65 cents per gallon to over 240 cents per gallon in the span of just over 15 years. This is equivalent to a ramp of about 117 cents per decade. This is the value used in the ramp slope as constant. Each setting of this ramp constant represents a distinct scenario of fuel price ranges when using this model for future predictions.

The other variable of pertinence is the noise correlation time that is used to control how strongly the pink noise is correlated with respect to time. Here the generally accepted value of twelve months for fuel commodity pricing is used.

After completing this model the next step is to attempt to calibrate the model against the market data shown previously. This is done by mainly adjusting the preference scales in the multi attribute decision making formulation, where for simplicity a simple overall evaluation criterion was used. An overview of this is shown in Table 7.

Figure 41: U.S. Jet Fuel Spot Market Prices[141]

Table 7: List of Market Model Calibration Variables

Variable Name	Value
Sensitivity of Attractiveness to Installed Base 767	0.01
Sensitivity of Attractiveness to Installed Base A330	0.04
Threshold for Installed Base 767	10.0
Threshold for Installed Base A330	10.0
Yield Attractiveness	$0.15
Price Attractiveness	$Million100.0
Passenger Capacity Attractiveness	250
Range Attractiveness	6000nm

The values shown represent the results that were obtained calibrating the overall number of aircraft sold for each of the competitors. Consequently, Table 8 shows how well the calibrated model fits the actual market data.

Table 8: List of Market Model Calibration Results

	Actual Market	Market Model	% Error
767 sold overall	34	34.86	0.02467
A330 sold overall	202	200.13	0.009344

Table 9: Goodness of Market Model Fit

Predicted Variable	R^2
767 Sales	0.687
A330 Sales	0.853
767 Market Share	0.471
A330 Market Share	0.471
Recalibration with Market Exit Date for the 767	
767 Sales	0.857
A330 Sales	0.945
767 Market Share	0.716
A330 Market Share	0.716

The final table, Table 9, then compares the actual market data as shown before when compared with the predicted market data from the calibrated market model. As can be seen, the predictive power of the model is not overwhelming with R^2 values between 47% and 85%. This is about on the same quality of predictive power that the previously shown Logit market models exhibit.

Therefore, it was decided to go back and slightly modify the model. The variable that was introduced was the 767 market exit date. The model was predicting some 767 sales up to very recently. However, with some discussion with Boeing representatives the point was brought up that the 767 probably would have sold some more units, but Boeing stopped offering the 767-400ER in lieu of shifting the orders on an equivalent variant of the 787, the 787-9. Therefore, it was a prudent approach to introduce another variable into the model to be able to mimic this behavior. The same table shows the re-calibrated results of this improved model. Now the R^2 values are in the range of 72% to 95%, which is a significant improvement over the currently existing models, which generally only show R^2 values of $58 - 75\%$.

6.3.6 Extended Unified Trade-off Environment

The problem with this model in Vensim is that while it can perform a rapid trade-off analysis when changing some of the scenario parameters or aircraft design variables or it can perform a sensitivity simulation where it is possible to define probability distributions for variables and then generate time dependent probability functions on any variable of interest. However, it cannot do both at the same time.

This means that a new kind of environment that can do both at the same time had to be used. The previous examples of the integrated MATLAB/Simulink environment was able to provide some aspects of the required features, but in the end was much too slow to allow an interactive environment.

At this time it is probably a good idea to review the list of requirements for the environment to be used. A short overview of these requirements:

- High speed robust ordinary differential equation solver
- Time dependent modeling of all variables
- Easy integration of surrogate models
- Prediction profiler capability
- Monte-Carlo simulation capability
- Customized graphing capability

- Non-linear algebra equation solver (optional)

A special note has to be given to the non-linear algebra equation solver requirement. The fundamental structure of system dynamics models does allow coupled sets of non-linear algebra equations to exist in a system dynamics model. However, most of the time it is possible to change the model structure to an equivalent model containing stock variables instead of purely converter variables. This breaks circular references in the algebra portion of the solution because now parts of the model are fed into the differential equation solver. The result is a decoupling of the algebraic equations in the model so that no special solver is required for the algebra equations.

This is also consistent with the functionality of the existing system dynamics software. Most of the existing software packages do not implement such advanced non-linear solvers at all and therefore show errors if circular references exist in the model. Only some select high end packages that expose a significant portion of the internal functionality possess this functionality. For the purpose of this work this functionality is therefore also not required.

Some or most of the requirements can be met by JMP. However, it would be necessary to solve the differential equations beforehand and then import the data either as time dependent surrogate models, which could be difficult due to the potentially very non-linear behavior of some of the variables. Furthermore, while it is possible to create custom graphs with the JMP Scripting Language, at the time of review the feature set of the scripting language was not feature complete and lacking in documentation and stability.

Other ways of achieving this had to be researched. The goal was to not to resort to a programming language due to the fact that all the differential equation solving and graphing capabilities had to be developed, which is out of the scope of this work.

However, significant research did not show any existing software that was able to achieve what is required for this Extended Unified Trade-off Environment to work as intended. The only remaining option was to reluctantly go down the path of custom software development. The one caveat that should be noted is that all the requirements can be satisfied by commercially available software separately but not all in one unified environment. Furthermore, custom software can be developed rapidly with the use of existing libraries.

With that being said, it was relatively easy to find a library that satisfies the first requirement of the robust ordinary differential equation solver, because a number of reference implementations exist. The one used here is the well known "Numerical Recipes in C" implementation which is available free of charge online [142]. This implementation shows a fast fourth order Runge-Kutta fixed time step method that is a quite robust implementation. The actual code of this implementation can be found in Appendix C.2.

The other library that was needed was to display the prediction profiler while being able to display time dependent probability distributions. This is possible with a number of libraries but one that stood out for its easy of use and capabilities was the JFreeChart library[143], an open source graphing library that is well documented with reference implementations of various functionalities.

This library is only available in Java, however, and therefore the entire development had to take place in a Java environment for which the Eclipse Integrated Development Environment (IDE)[144] was selected.

The main task that had to be completed was the creation of an interactive profiler panel in which the vertical hairlines had to be synchronized for each column like in a prediction profiler environment. The other task was the high level of speed for displaying all information. The key achievement for this was to not resort to a generalized contour plot for the time dependent probability distributions but rather resort to a set of difference plots showing confidence intervals. Furthermore, the integration of the differential integration solver with the Monte-Carlo simulation with the graphing library was very important. For this reason the graphing capabilities were separated from the computational capabilities into a separate computational model representing the entirety of the surrogate models integrated into the system dynamics market model yet again integrated into a Monte-Carlo simulator. The source code of the result is shown in Appendix C.2. The graphing capability source code is shown in Appendix C.3. It should be noted that the source reflects the final state of the complete competition model. One of the key advantages of using these readily available libraries was that code development was kept at a minimum. The final full capability code is just over 2000 lines, which is remarkably small for a full graphical environment including significant functionality.

The first implementation was purely for testing the environment and accomplishing a match in the results between the reference competition model in Vensim and the custom solution in Java. A screen shot of this is shown in Figure 42. It should be noted that both models match results to six significant digits for various settings of the calibration variables shown.

In order to aid the creation of these high-resolution graphs another library was used. The iText library [145] provides a very simple capability to reuse the graphical user interface objects in Java and "print" them to PDF (Portable Document Format) [146]. This enables the output of the entire environment in a vector format into a document format without the loss of resolution associated with normal screenshots. This means it is possible to zoom into all the required detail of the resulting figures and not be subject to "pixelation". This functionality is simply provided by a simple right click and a "Save as PDF" option. This has been extremely helpful in the generation of outputs from the unified environment.

Another advantage of the implementation in Java — as compared to the other attempts in MATLAB — is the speed of execution. A complete Monte-Carlo simulation with ten thousand runs of the basic market model while already involves 18

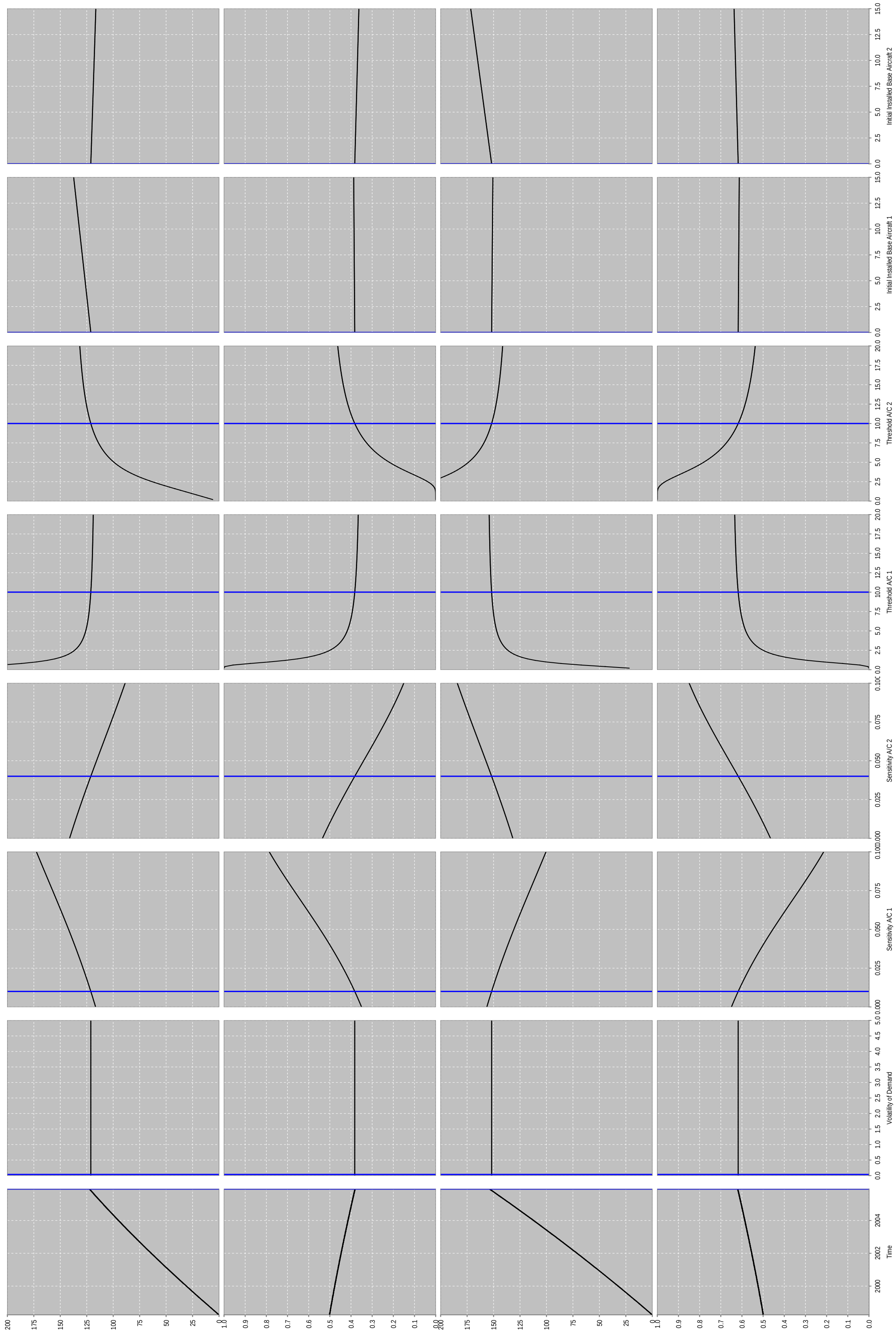


Figure 42: Reference Implementation of the Unified Environment

variables and 92 time steps including the subsequent plotting is able to complete its run in about five seconds, even on an older computer. This means that every time one of the vertical hairlines is moved a complete analysis will run and there will be very little delay in displaying the results immediately. This gives the user immediate feedback about the effects the change in scenario, market conditions, or vehicle design had. Furthermore, it is possible to explore the change over time or alternatively explore the effects of changes at any given instant.

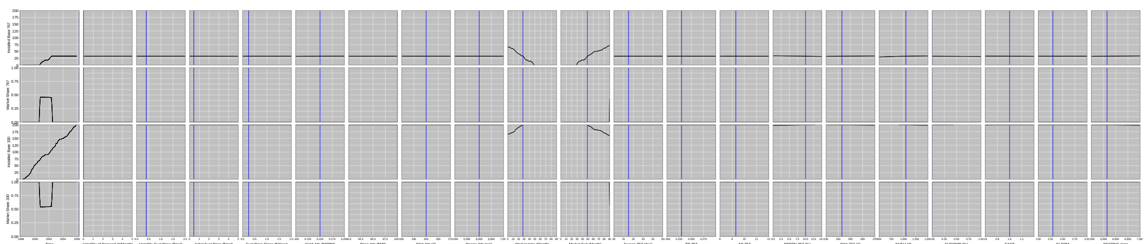


Figure 43: Full Implementation of the Unified Environment

The next step was then to implement the full environment, which is shown in Figure 43. It should be noted that all of the cross hairs are fully moveable. The environment is partitioned into several areas. To the left the time dependent probability distributions are shown for the most relevant variables here, which are market share and sales for each of the two aircraft. The next columns represent a number of volatility variables that define the width of the uncertainty distributions on variables such as the demand and the fuel price. The next variables that follow are the market scenario variables such as the fuel price slope and the attractiveness strength variables for various attributes and the market entrance and exit times. Then farthest to the right are the aircraft design variables of the 767. This represents a compromise solution since in the interest of space it was not possible to construct a graph that shows all 82 model variables at once. Therefore, a scenario where a particular designer is trying to design a particular aircraft trying to compete successfully in a market had to be chosen. Due to this the only variables that are shown are for a single aircraft not all of them.

Due to the nature of the plot it is difficult to reproduce it on standard size paper. Therefore, an in-depth discussion of certain excerpts follows. Furthermore, it should be noted that all variable settings are defaulted to match the Vensim model shown previously. These settings represent a close match to what happened in the actual market. Therefore, the uncertainty distributions are set to zero, except in the case of the fuel price, where the volatility is set to match the volatility of the fuel price during the time period of the simulation.

Starting at the left of the environment, shown in Figure 44, the most important feature is the time axis, where the most important variables in this analysis, namely market share and total overall sales are being tracked as time dependent probability distributions. Shown in those particular graphs are the mean and the 50% confidence intervals. This is in contrast to the full contour plots generated in MATLAB earlier. However, showing the confidence intervals allows a significantly improved execution speed, which is critical in achieving an interactive environment. The other variables that follow are used are the volatility of demand and fuel price. Both variables define the standard deviation in the input probability distributions that enter the market competition model through the external demand and the pink noise model for the fuel price. The next variables define the scenario for the fuel price directly. It is possible to select an initial fuel price, which represents the price of aviation fuel in Dollars per gallon. The second variable is the rate of change in to fuel price in Dollars per gallon per year. The values that were selected are representative of the fluctuation for the duration of the model, which is April 1998 to December 2005. The value of the slope is approximately equal to two Dollars per gallon per decade in price increase. If any particular value is to be selected it should represent the mean of the range and then the volatility will have to be adjusted accordingly. However, if it is desired to determine the possible ranges the volatility should be set to zero and then the slope

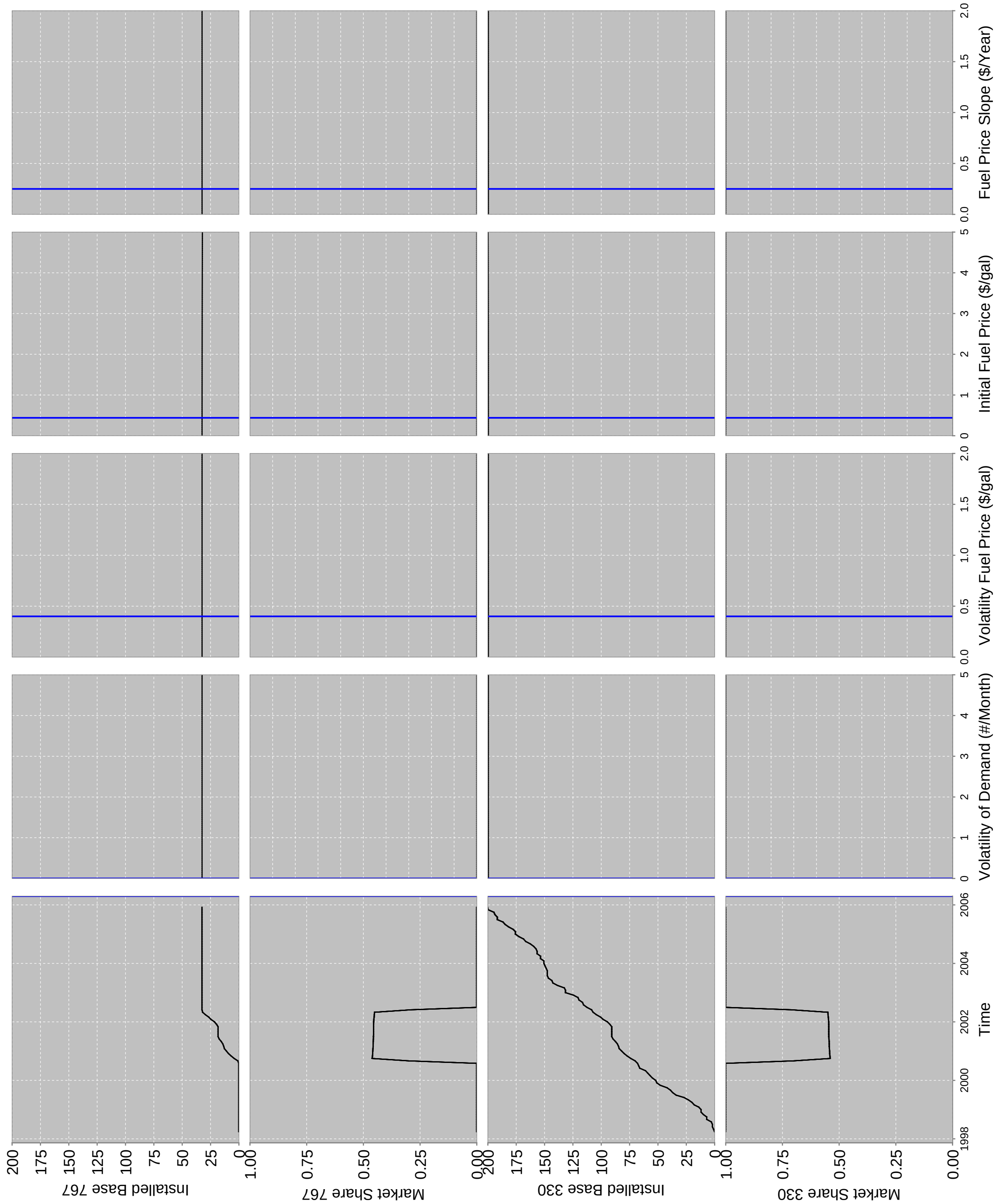


Figure 44: Scenario Definition in the Unified Environment

modified accordingly. This will allow the user to determine the potential variability in the model due to extremes in fuel price change scenarios.

What follows to the right of the scenario variables are a collection of variables that allow changing the preferences of the airlines. Even further to the right are the market entry and exit dates for the Boeing 767-400ER. This is shown in Figure 45. What is significant here is that the plane was offered at a significant later date than the A330-200 and then was stopped being offered in favor of a new variant of the 787. Changing the entry and exit dates can significantly change the overall market as evident by the partial derivatives shown in the respective column.

The speed of execution of the complete environment is obviously reduced as compared to the basic market competition model, especially since now the model also includes a number of surrogate models, which are significantly longer than the rest of the model relations. This leads to a trade-off that has to be performed between accuracy and speed. The issue is that a certain number of Monte Carlo runs have to be performed to gain an acceptable accuracy of the resulting probability distributions. However, the full model executes significantly slower than the simpler model introduced earlier. Therefore, the number of Monte Carlo runs can be lowered to preserve the relative immediacy of the feedback with even the full model. However, this potentially means that the accuracy of the probability distributions in the results is lowered. For purposes of generating screenshots however, the number of runs can easily be increased to improve the accuracy of the results.

Finally, the results as shown in Figure 43 represent the same results that match the calibrated Vensim model as shown earlier. The main difference is that the size of the probability distributions is set to zero for the demand and to the same value as the pink noise in Vensim. The results also represent a complete match of the data.

The details shown in Figure 47 show the effects of the 767 being introduced much earlier and also not exiting the market mid way. Again, this figure only represents

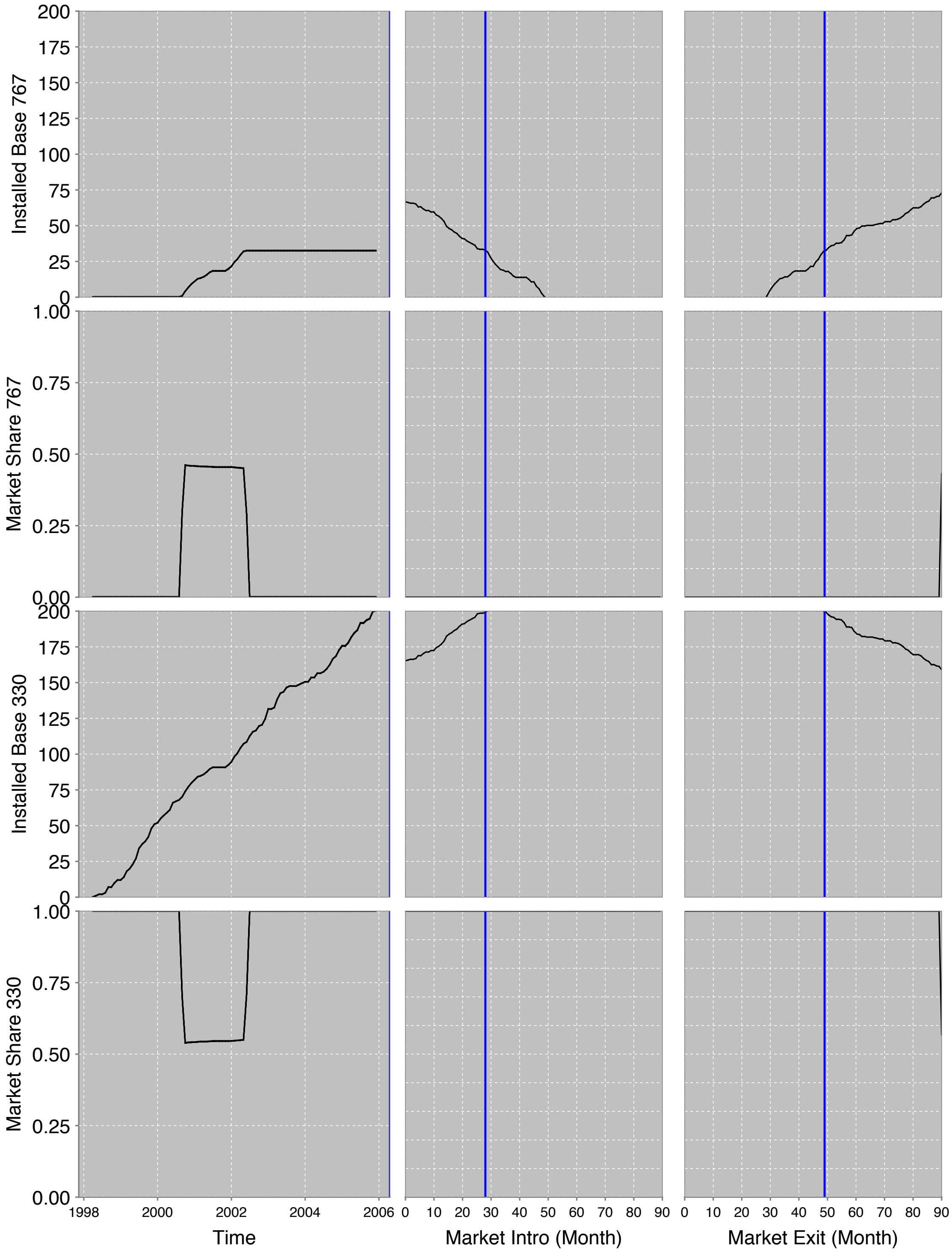


Figure 45: Market Entry and Exit in the Unified Environment

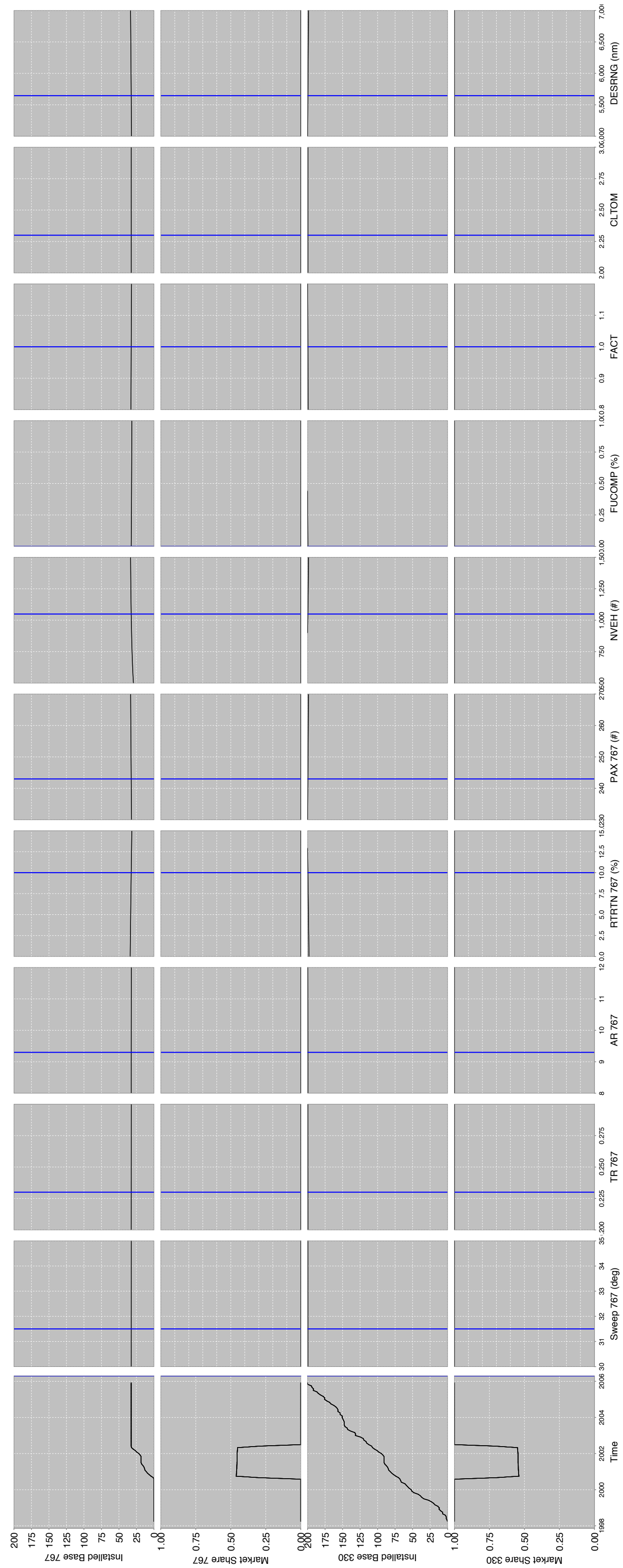


Figure 46: Aircraft Design in the Unified Environment

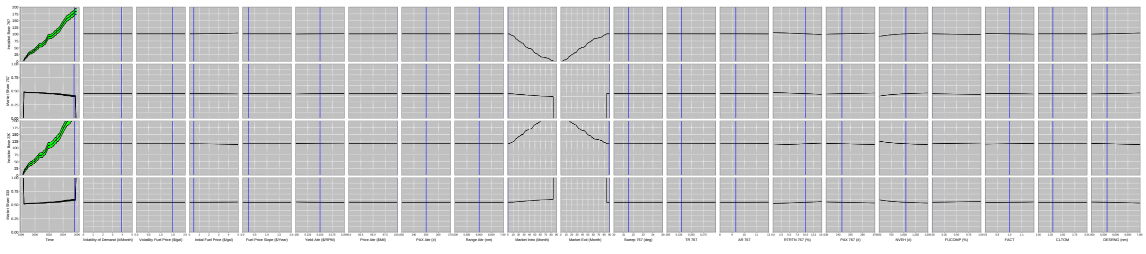


Figure 47: Effects of Market Conditions in the Unified Environment

a rough overview of the entire environment. More detail is shown in the following figures.

What can be seen in Figure 48 are the effects of much larger choices for the volatility of the demand and the fuel prices. This means that significant distributions are now visible. The probability distributions shown in the left most column again show the mean of the values and the 50% confidence intervals as calculated from each the top and the bottom of the range of the individual variables.

The trends that are shown for each of the tracked variables represents the deterministic partial derivative of that variable with respect to that particular choice holding all other variables constant. In Figure 49 it is clearly visible that the market entry and exit dates have a very large effect on the overall sales and market share. The settings chosen for this particular scenario show that if the Boeing 767-400ER would haven been on the market at about the same time and continued to be offered, the overall developments in this particular market would have been quite different.

Figure 50 shows the effects changes in the aircraft design of the Boeing 767-400ER would have in the same situation while holding the A330-200 fixed. The most predominant effects in trying to capture more of the market come from important variables such as the overall size of the program production numbers as well as the rate of return of the manufacturer. The most significant impacts from the remaining variables stem not from geometric variables such as the aspect ratio, the taper ratio, or the wing sweep, but rather from the technology variables such as the composite

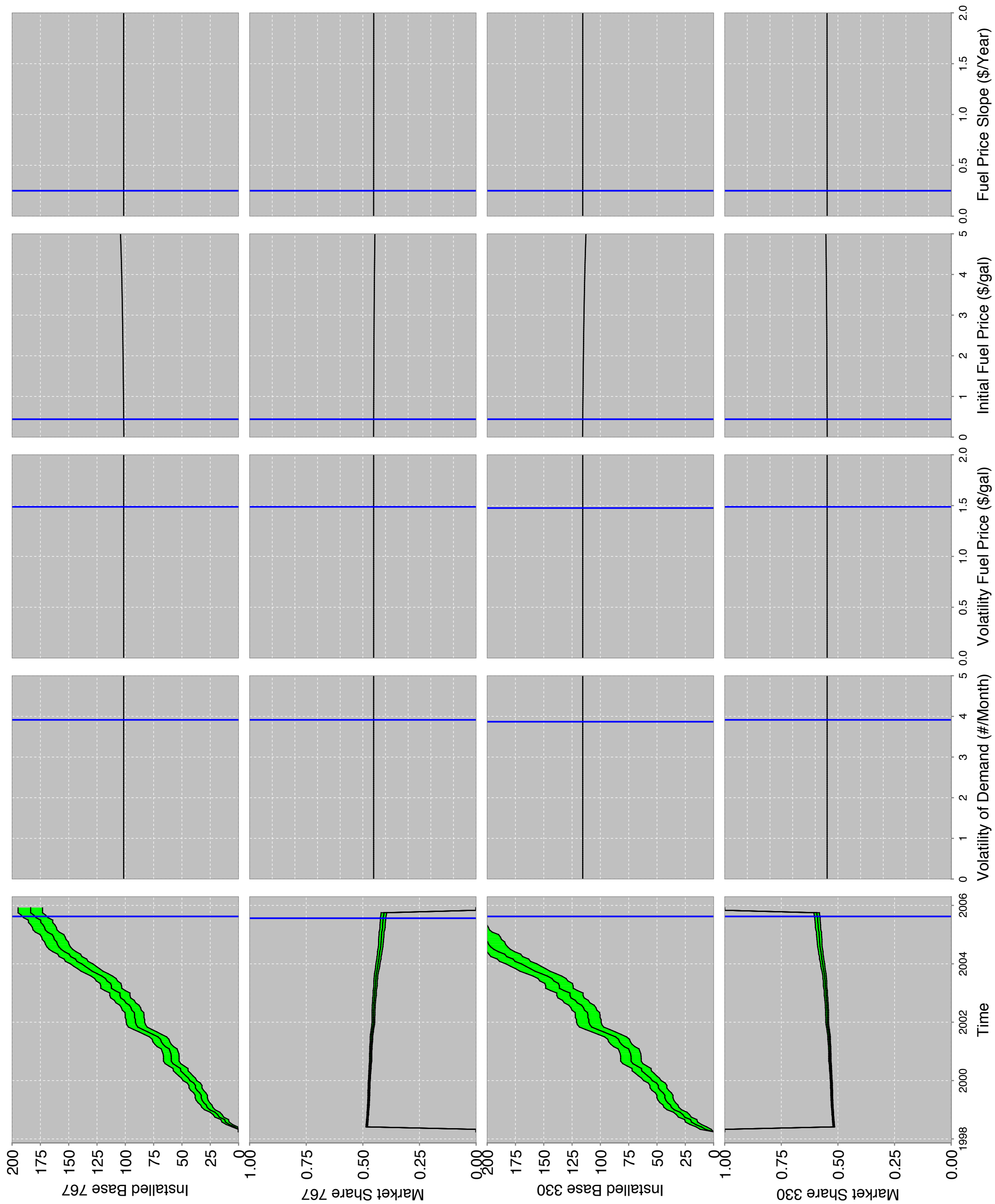


Figure 48: Scenario Definition in the Unified Environment with equal Market Offerings

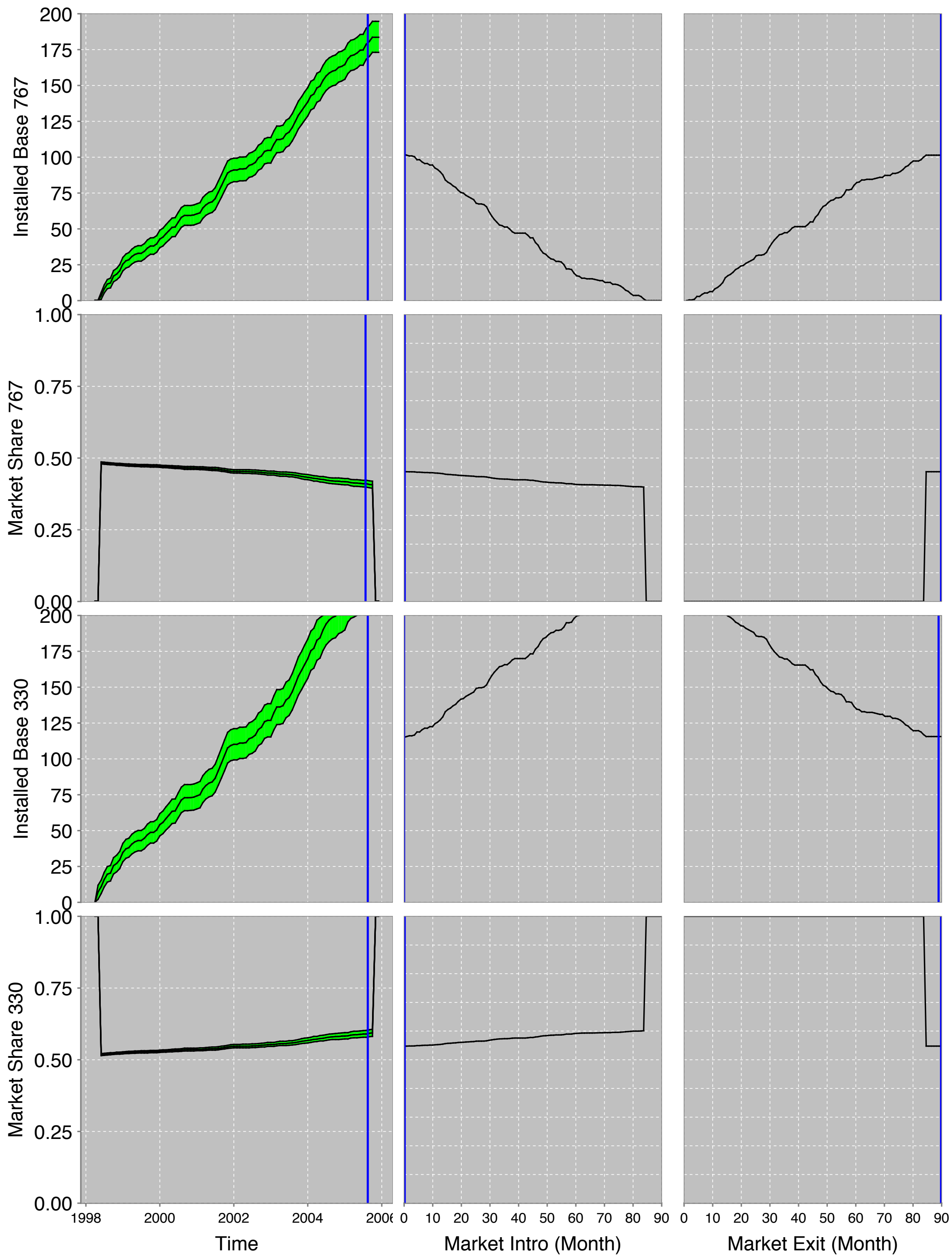


Figure 49: Market Entry and Exit in the Unified Environment with equal Market Offerings

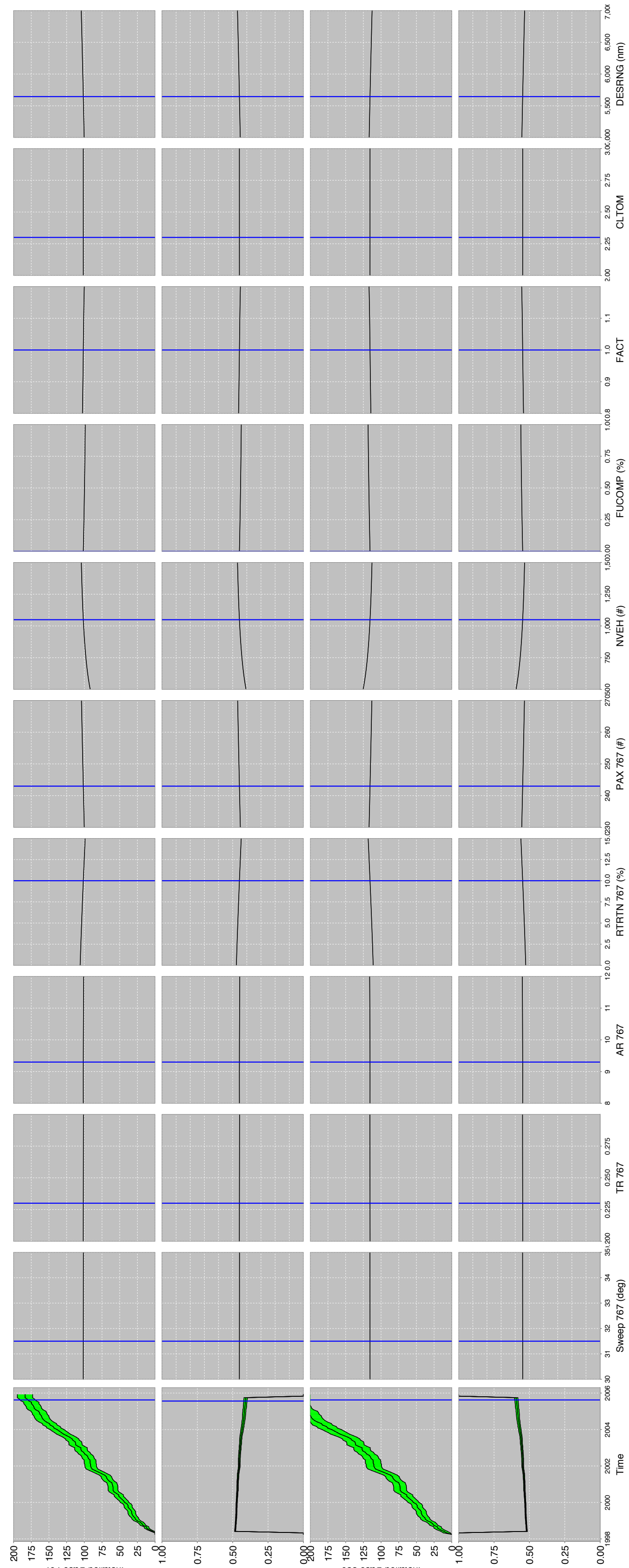


Figure 50: Aircraft Design in the Unified Environment with equal Market Offerings

fuselage switch and the fuel burn improvement factor. Of course the number of passengers and the design range also have some effect due to the changes in aircraft productivity. These changes in technology are consistent with the claimed features of the new Boeing 787-9 that will feature a composite fuselage and a much reduced fuel consumption. If the final aircraft will deliver these improvements as promised it is then definitely possible as can be seen in Figure 50 that a technology improved version of the 767-400ER could definitely be competitive with the existing A330-200. Furthermore, market entry dates would play a major role, as would the fuel price scenarios applicable, which could easily offset or amplify improvements in technology.

All of this is shown with specific corridors of probability which are shown as bounded by the 50% confidence intervals. They are shown only for the market share and the total accumulated aircraft sales, because these variables were deemed the most important to the subject at hand.

These corridors can change size and shape significantly depending on the individual choices performed by the user of the interactive environment. They each represent areas of likely future outcomes as defined by the particular parameter choices, which in turn each define a particular scenario.

As a final comment it should be noted that the environment allows the display of all variables, even simultaneously. However, it is not practical to do so outside of the largest displays such as large projection screen. It is also not enough to simply increase the size of the display, but it has to go hand in hand with an increase in resolution. Furthermore, it then becomes very easy to become overwhelmed by the amount of information accessible at once. It is much more practical — as shown above — to define a specific application or role for the intended user and to hide all but the most important parameters. This enables a much more manageable environment that is easier to understand.

CHAPTER 7

MARKET VIABILITY BASED DESIGN

7.1 Inverse Design

Inverse design is a very recent development in the field of system engineering[147, 148]. The idea of inverse design is to completely reverse the order of thinking in the process of systems design. The conventional approach in systems design has been to first define a concept or more recently a morphological matrix showing choices of alternatives for different system architecture specifications, subsystems, and capabilities. The result of this is that a set of choices for each of the rows in the matrix represents a single concept definition. This has the result that a very large number of concepts can be imagine. The drawback is that most of the possible concepts cannot be analyzed due to the lack of analysis capability or lack of resources and time to analyze more than a few concepts at a time.

The key point of this is that historically engineers have always define a concept first and then analyzed the concept to learn more about its performance and other measures of merit such as cost. However, this also means that before defining a concept engineers always needed system requirements. These requirements often prescribed in great detail what a system's measures of merit were required. This leads to a very detailed specification about shape and performance requirements in a very narrow range of possibilities.

There is a very significant chance that a much better solution might exist outside the narrowly defined concept space. Furthermore, the lack of understanding how low level system choices impact the overall system measures of merit leads to a very time consuming iterative process where the design choices are iteratively altered until most or all of the capability-level measures of effectiveness are met.

Inverse design represents a reversal of this process by providing a large number of concepts defined by surrogate models to reduce the effort required to generate this large number of concepts. The concepts are generated in a process of creating a large number of Monte-Carlo runs of surrogate models. In order to guarantee a good coverage of the entire design space a space filling random design of experiments is used with uniform distributions on design choices, signifying no preference over the range provided.

Mapping the top level measures of effectiveness against all the system variables in one large multivariate scatter plot matrix, it is then possible to explore system designs that meet certain capability-level measures of effectiveness by simply selecting the system design that do not meet the requirements and hide them. This capability is provided in JMP. A high level conceptual view of this new approach is shown in Figure 51.

This process can then be repeated with a number of the capability-level measures of effectiveness until only a small number of desirable systems remain. These remaining systems are then guaranteed to meet the measures of effectiveness selected. Furthermore, it is then also possible to identify which design variable choices are associated with the respective systems. In JMP it is furthermore possible to identify which of the design experiments represent the remaining system designs, which can then be used to refer to the detail design definition and even its visual shape.

This methodology is a good fit for the system dynamics market model with integrated aircraft design models because it enables a new way of thinking about aircraft design. In essence it allows "Market Viability Based Design". This means that instead of the traditional approach of assessing the viability of a system — that is the ability to recover the development costs of a project — it is now possible to define

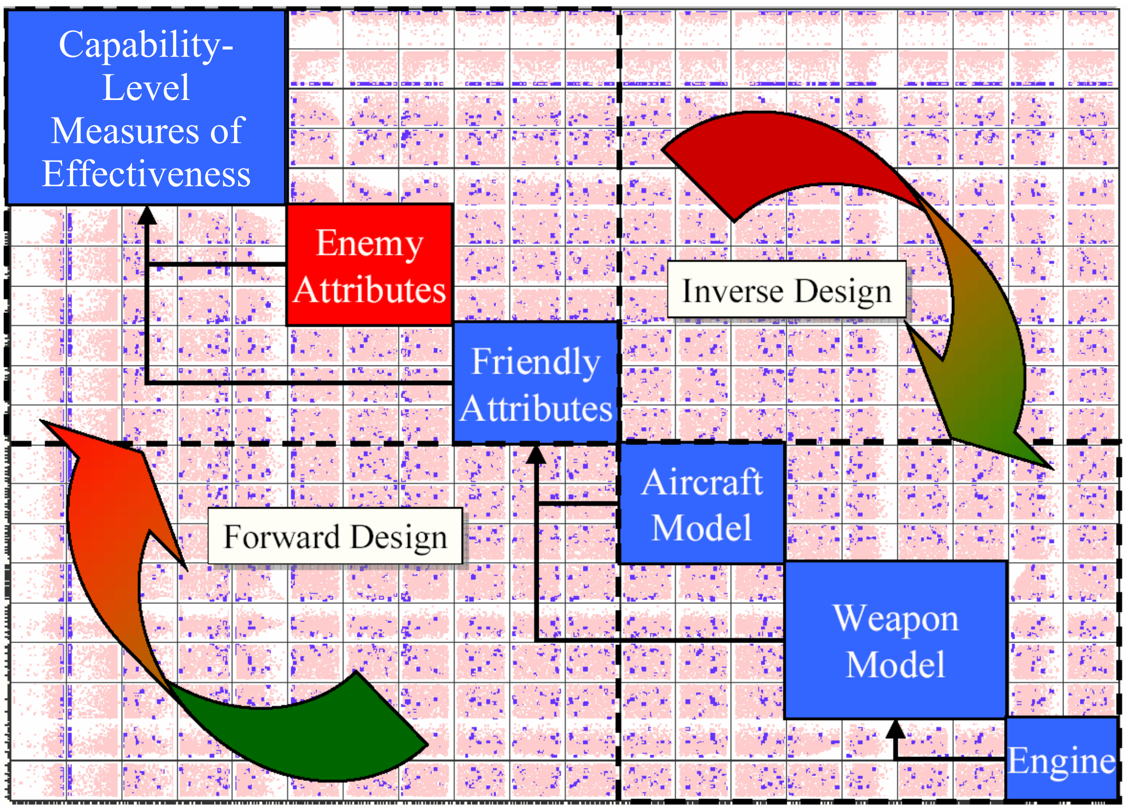


Figure 51: Multivariate View of the Top-Down Decision Making Process for Inverse Design[148]

the desired market outcome and the associated viability of the system and then explore which type of aircraft designs meet these requirements. This has the potential to significantly increase design freedom because it does not require detailed upfront specifications of a system, but rather very high-level specifications about the market success and the return on investment. Therefore, Figure 52 shows how this process can be adopted to achieve just that.

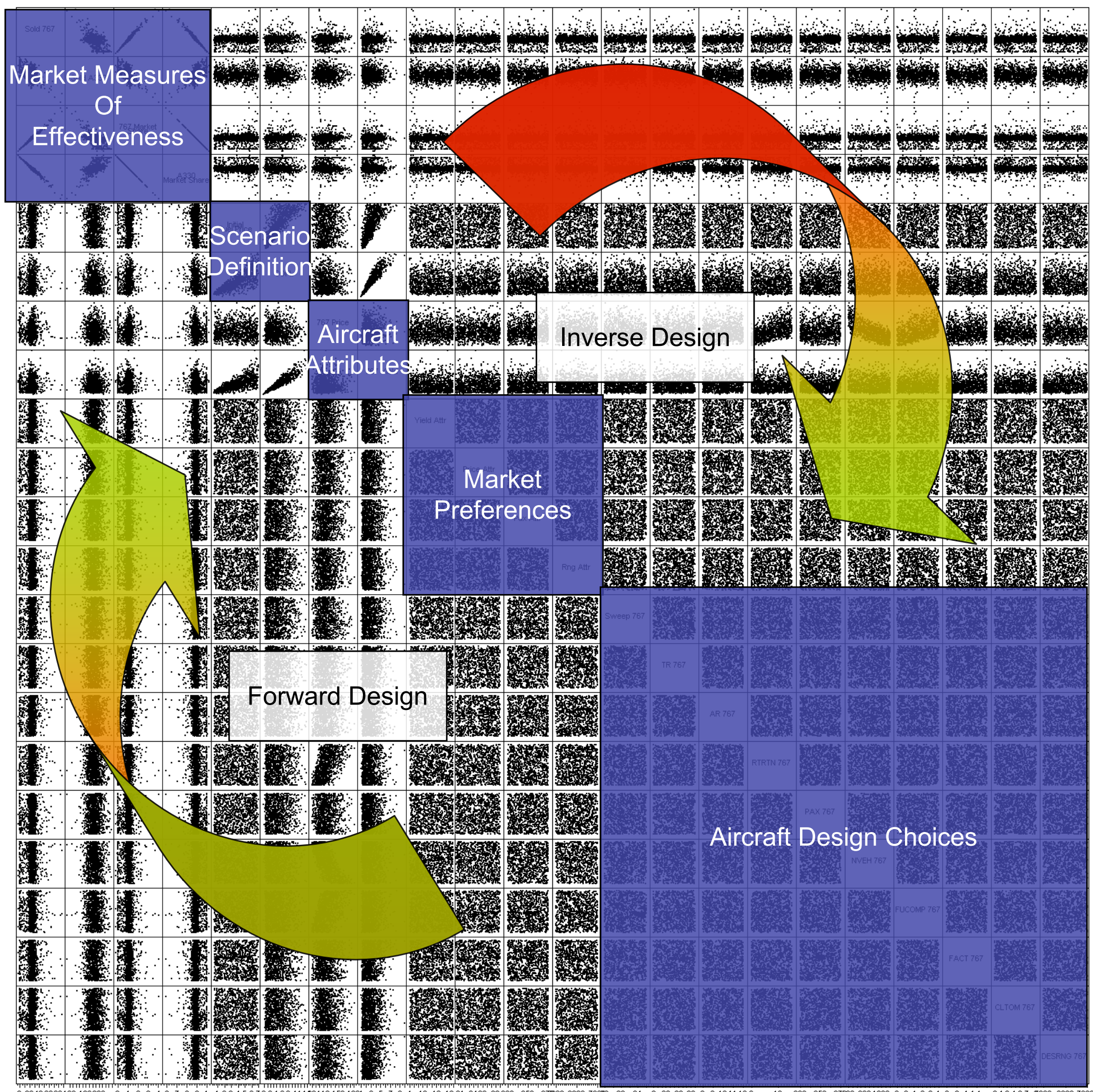


Figure 52: Market Based Viability Top-Down Decision Making Process

7.2 Market Exploration

After this brief introduction to the basics behind the inverse design methodology, it is now possible to utilize the Java version of the market model to create the necessary Monte Carlo runs for the multivariate scatter plot. The Java code was simply extended to be able to save out a comma separated value file containing all the runs, which include all 82 variable settings for each run.

This was then imported into JMP to generate the multivariate scatter plot. At that point some problems were encountered with the speed of the computer. Due to the nature of the plot JMP essentially has to display $81^2 * Runs$ points, which in this case is equal to over 6.5 million individual points. This meant that the computer was no longer able to function properly, including basic functionality like scrolling. Therefore, the number of variables in the scatter plot had to be substantially reduced before a viable scatter plot could be produced.

7.2.1 Market Definition

The first step in this process was to eliminate all the market calibration and preference variables because they are most likely outside of the control of an aircraft designer or even a manufacturer and therefore are of lesser importance for a decision maker trying to determine a successful aircraft design. Although these excluded variables can still be changed if needed, but it was decided that they should stay fixed for the purpose of this demonstration. If necessary, it is easily possible to include them yet again in another iteration of the method presented here should that be deemed necessary.

Another step taken in order to significantly reduce the number of variables that had to be considered, was to select the role of just one aircraft manufacturer. This means that the user takes on the role of one manufacturer and can only actively control the configuration of one aircraft and not of any of the other aircraft in the same market. This assumption is quite reasonable from the perspective of an aircraft manufacturer trying to decide how to best compete against a known competitor. This might not always be desirable, however, but can again be changed easily in any future iterations of this process, should that be deemed necessary. Finally, the result of this exercise of reducing the overall number of variables to just 22 is what is shown in Figure 53.

This figure shows the output of the Monte Carlo simulation that was collected from the EUTE environment. This was accomplished with a simple output to a comma separated value text file containing one line for each of the aircraft and market combinations including all the model variables. Once this data was imported into JMP, this data was then displayed on a multivariate scatter plot. Such a plot shows all of the selected variables on a vertical and horizontal axis against all the remaining variables simultaneously.

This means that all displayed variables can immediately be traded off against each other. Furthermore, some variables — for example the design choices shown in the lower right corner of the plot — evenly cover the entire range of their domain and show no correlation whatsoever between the other design choices next to them. This behavior is as expected, since the design choices available should not be dependent on each other and uniform distributions were selected for them to signify that no particular preferential setting exists within that domain. Some of the market and the scenario variables, however, show some correlation with each other. For example, the market share of both competitors are inversely linked with each other, which makes sense because the simulated market only allows two players that directly have to exchange market share between them since they make up the entire market.

Now that the complete data set and the available variables have been established,

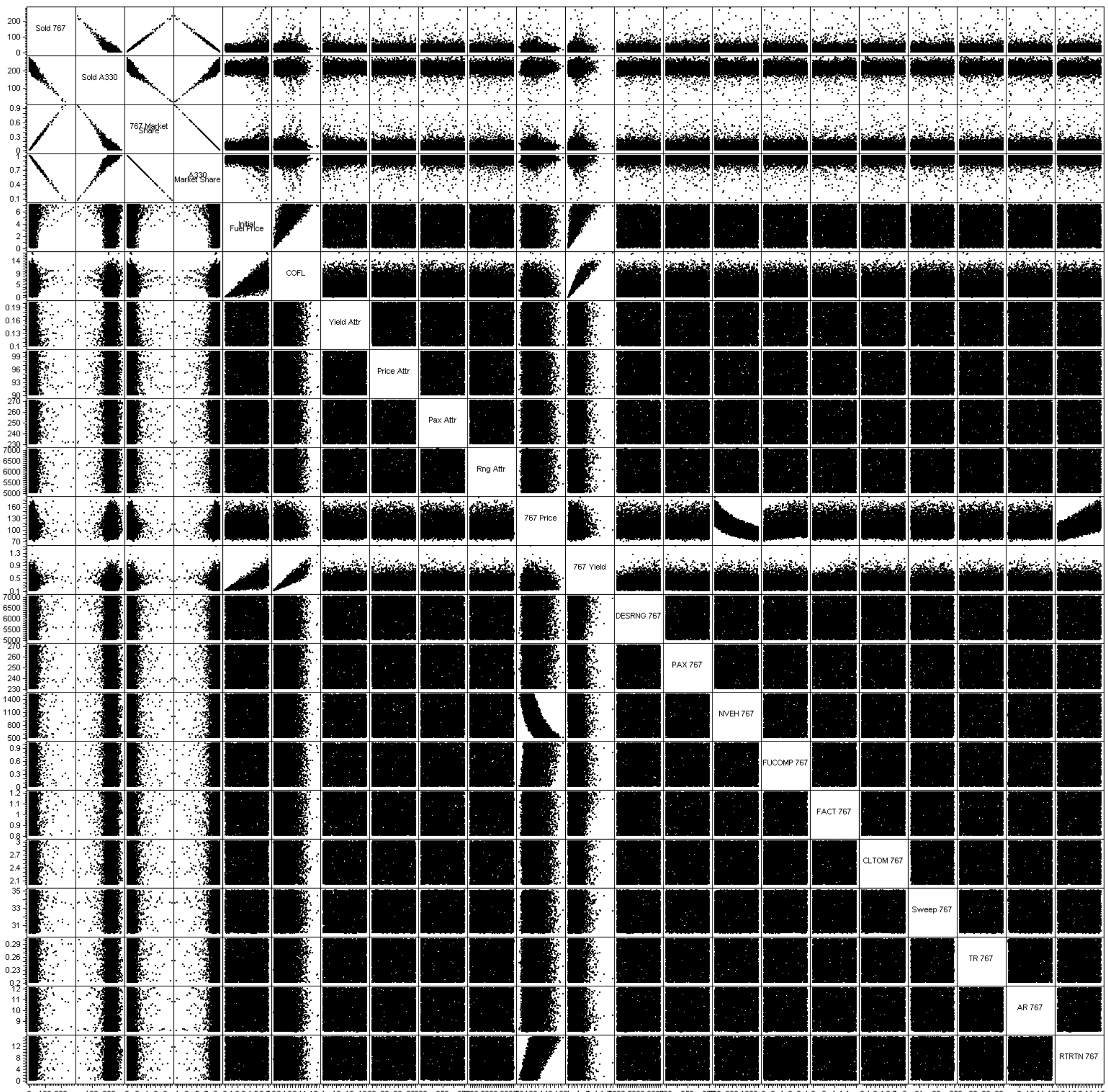


Figure 53: Complete Market Based Inverse Design View

it is time to start making some choices about the type of desired system. Each of the displayed points represents a unique aircraft and market combination. Therefore, it is now possible to select points with certain desirable or undesirable attributes and either highlight them or exclude them. Furthermore, it is also possible to then identify which aircraft design and market situation is represented by each point.

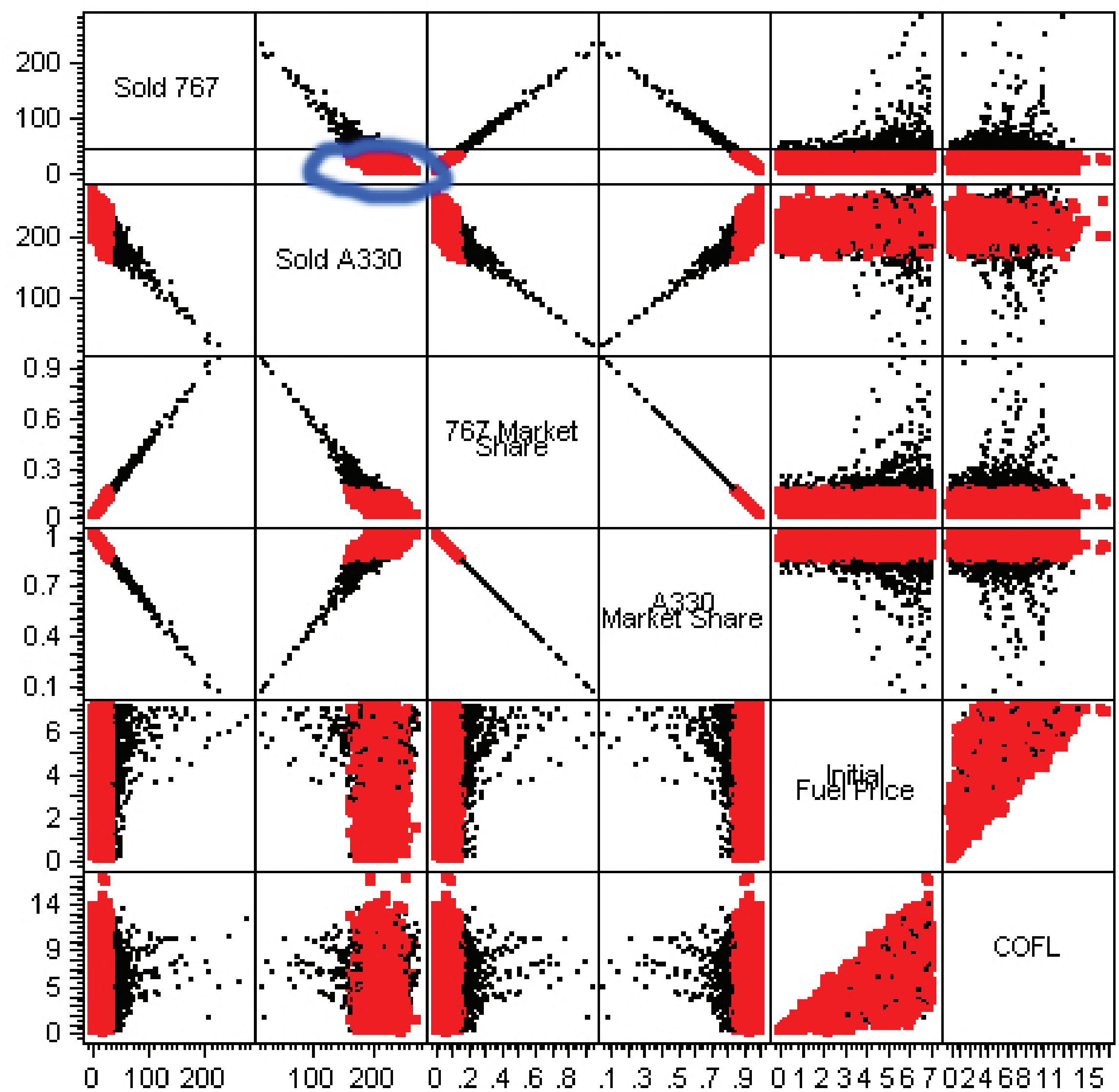


Figure 54: Selection of Minimum Number of Sold Aircraft

First, let's suppose that the first such choice a decision maker or designer makes is to exclude all the designs that sold less than 30 aircraft. The result of this can be seen in Figure 54. Shown in this figure are all the points where the undesirable points are highlighted in red and the remaining points still in black. These points can be easily selected with the lasso tool in JMP. After these points have been selected they can then be specially marked such as changing the point shape and color, which is what was done here. Alternatively, the selected points can also be hidden. The process of hiding these points implies that these points are not desirable. Therefore, hiding these points automatically implies a certain minimum market share for remaining aircraft and market combinations. The repeated process of selecting and then hiding and eliminating points is akin to eliminating undesirable designs and market situations from the pool of choices provided in the multivariate scatter plot environment.

The next step is to define a certain type of scenario for the expected fuel price ranges. Since the scenario variables provided here represent the fuel price at the beginning and the end point of the simulation period, it is possible to exclude certain ranges of price fluctuations. For example, it is only logical to eliminate points representing scenarios where the final fuel price is lower than the initial fuel price. This means first of all to limit the initial fuel price of the model to a maximum and then select a range of final fuel prices at the end of the scenario. These steps are shown in step two and three, which are shown in Figures 55 and 56.

These steps essentially follow the same pattern, where the undesirable points are first selected with the lasso tool in JMP. After they have been selected the user then can change to color to make them more visible, which was done in Figures 55 and 56. After the user is satisfied that the selected points should not be carried forward in this process, it is a simple task to simply hide these points. They are therefore then excluded from the further steps taken here.

Of particular note is that the default price drift rate guarantees that even the high initial fuel price scenarios, which were selected and then subsequently excluded, when viewed on the final fuel price axis shown that these particular scenarios consistently

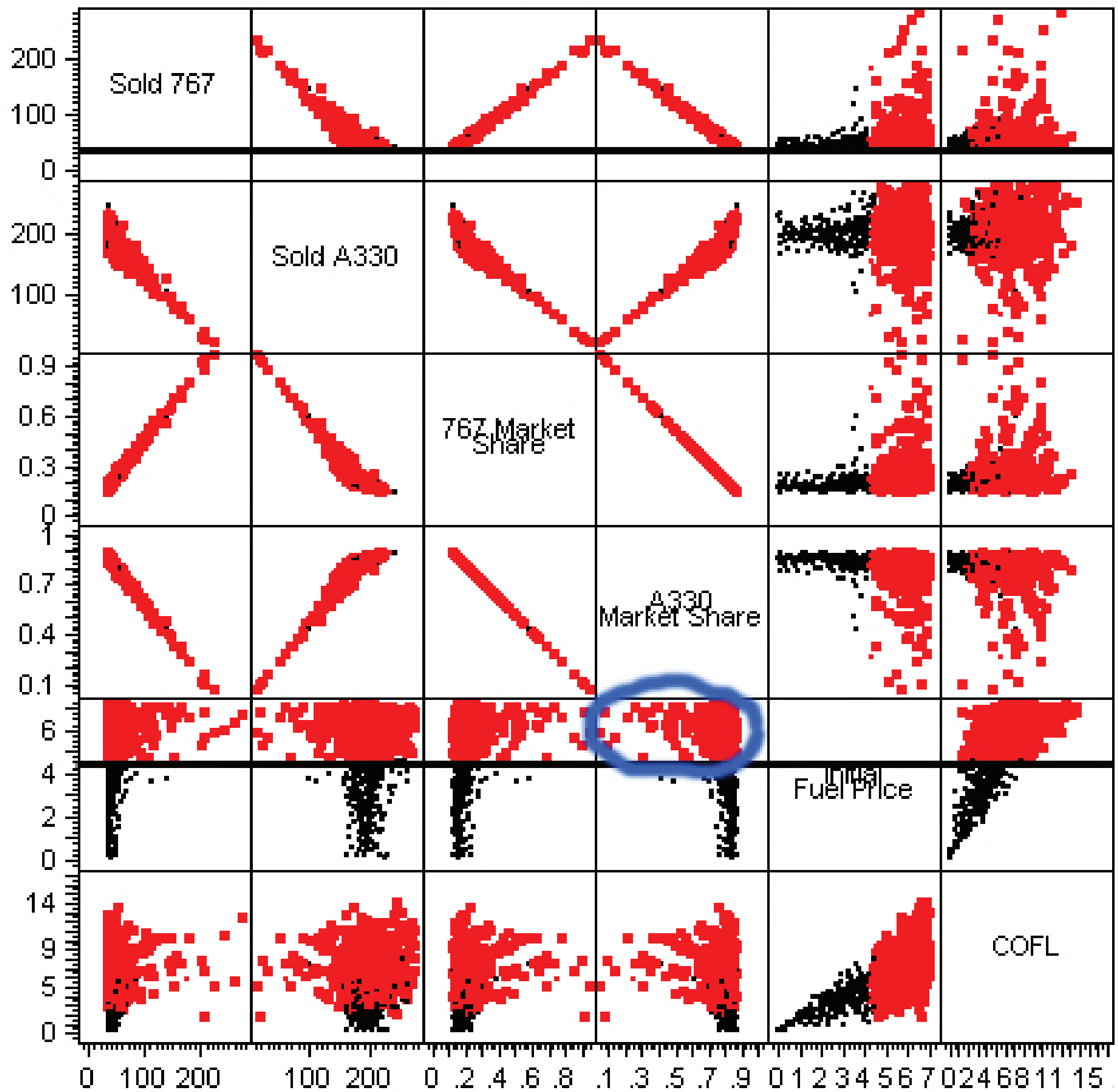


Figure 55: Selection of Initial Fuel Price Scenario Setting

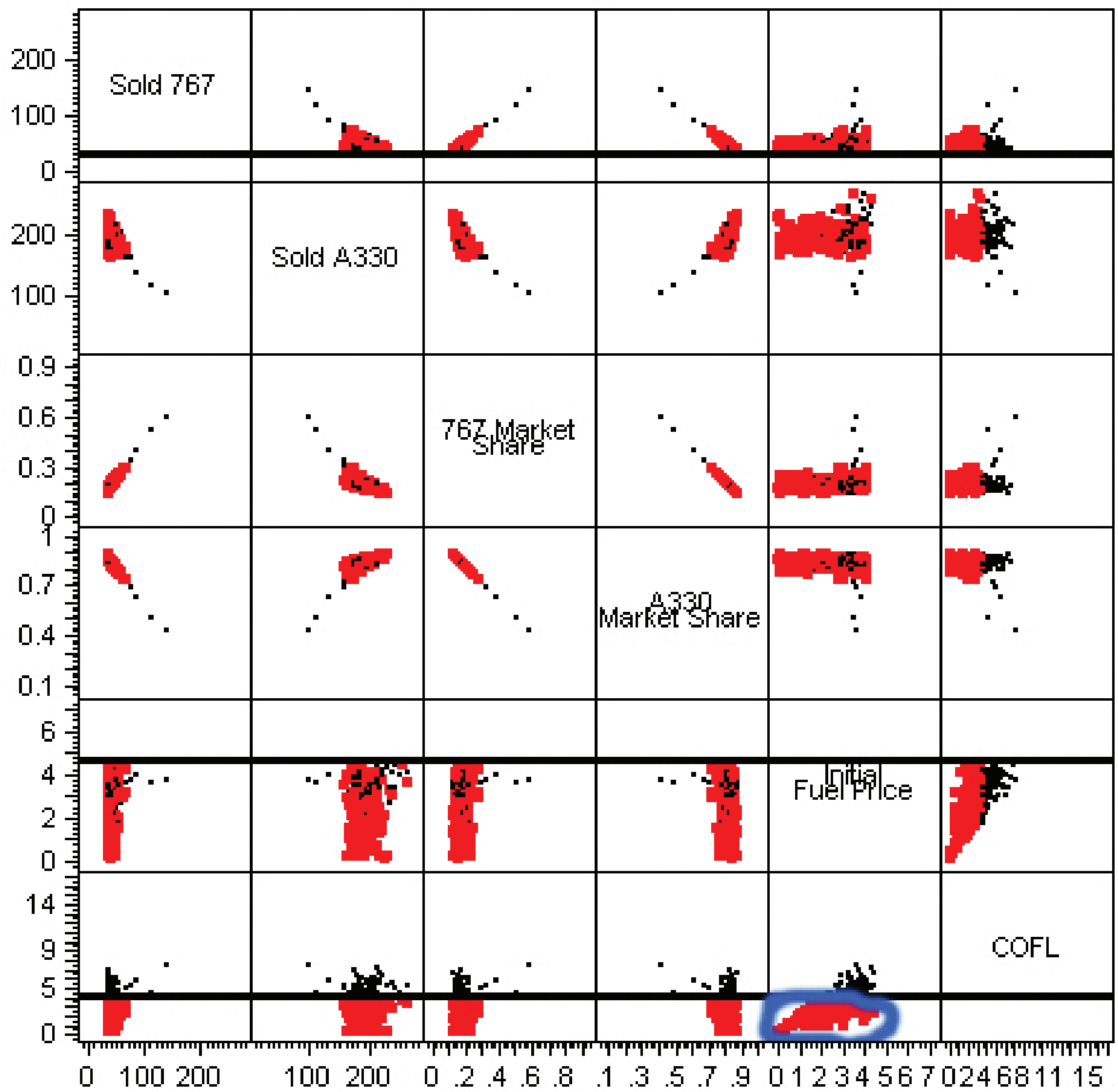


Figure 56: Selection of Fuel Price End Scenario Setting

correlate with high final fuel price scenarios. This shows that the general trend of fuel prices in this data set is generally upwards, which is exactly what was specified earlier.

Another important factor in the decision making for aircraft purchases by airlines is the average required yield per revenue passenger mile. As elaborated in more detail earlier, this represents the nominal ticket price an airline has to charge per passenger independent of trip length. This measure also includes a nominal airline return on investment assumption as well as a large number of items on the cost side involving direct and indirect operating costs as well as aircraft cost of ownership. Therefore, airlines have — depending on their cost structure and market they serve — a certain yield that they can achieve. This means that for an airline it will be favorable to operate an aircraft that requires a yield below a certain limit. The less the better off the airline is. Therefore, the next step is then to eliminate any required yield over $0.15 per revenue passenger mile. This is a very likely assumption to what airlines require. Again, the points representing aircraft above the given limit are selected by the lasso tool and colored in red. The result of which is shown in Figure 57.

After eliminating aircraft likely not considered competitive to operate by airlines, it is now time to look at manufacturer economic decisions. The rate of return for the manufacturer is a very important aspect of any vehicle program. Therefore, it is prudent to select only points that represent a rate of return of more than a certain desired minimum. In this example this was set to 5%, but it can be chosen as desired within the range limits of this variable shown earlier, which was set between 0% and 15%. Again, the points below this selected limit are selected and then colored in red, shown in Figure 58.

These points were then hidden, just like in the previous iterations discussed earlier. In the final step aircraft that cost more than $ 94 Million are eliminated. Again, this was done to represent choices by airlines that put an upper limit on the aircraft

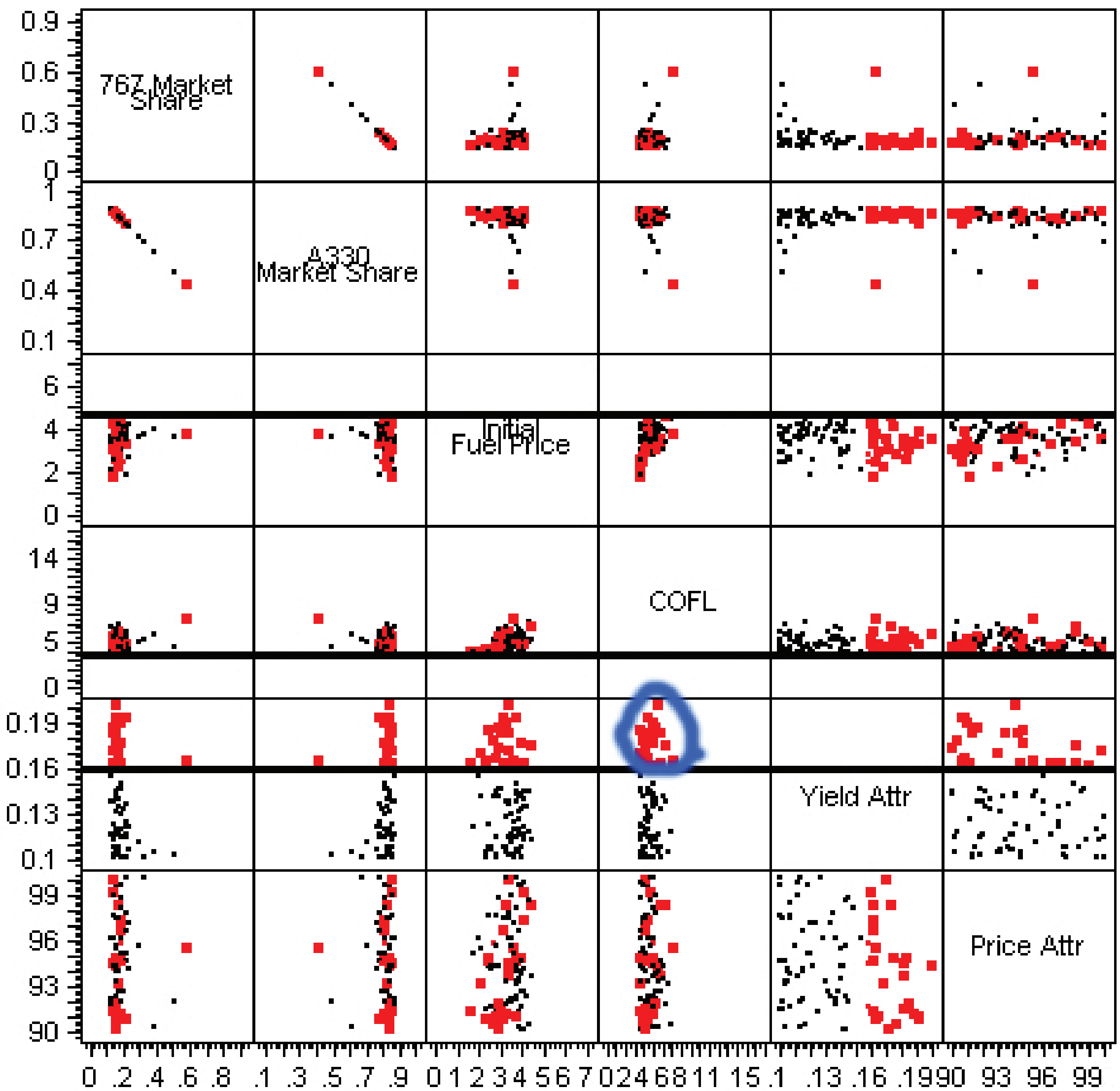


Figure 57: Selection of Maximum Market Required Yield

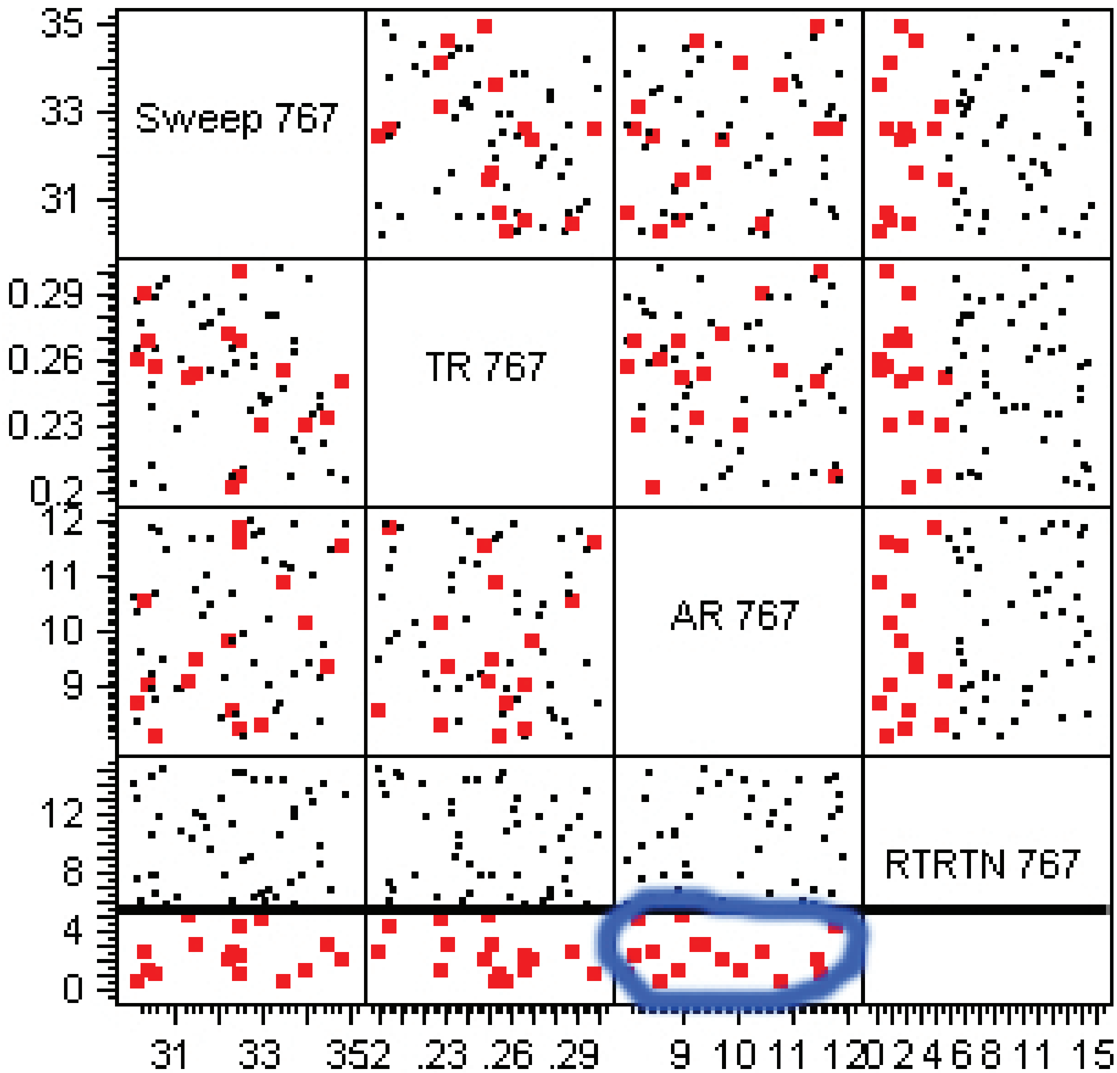


Figure 58: Selection of Minimum Desired Manufacturer Return on Investment

prices. Even though the aircraft price represents only part of the cost of ownership as represented by the required yield described earlier, nonetheless the price is a very important factor for the airlines and the manufacturer because it is directly linked to the required amount of capital and can also be subject to substantial deals and discounts, which are highly dependent on the individual negotiations between airline and manufacturer.

The result of this final elimination is shown in Figure 59. This figure also represents the final state of the multivariate scatter plot environment. This is significant because only relatively few points are left. These remaining points directly represent a particular aircraft and market. Due to the significantly reduced number of choices a decision maker could now go back and lookup what each of these points represents and then come to further conclusions about the accuracy, feasibility, and viability of each of the designs in a given market scenario. Furthermore, it is also possible to discover common properties of these remaining points, which can lead to common design rules or process policies that can be implemented to increase the successful outcome of a program.

Now that the market and preference variables have been constricted in ranges, there are only relatively few aircraft design points left in the scatter plot. This means that the remaining aircraft and market combinations can finally be explored in more detail. This can be included into an interactive decision making process that can not only focus on the aircraft design decisions, but also about the market success and the requirements all at the same time. The simplicity of iterating this process with adjusted preferences and settings can further yield insight into the competitiveness of a particular aircraft. Iterations of this process can also include changes to the underlying market model and the technical details of each aircraft if desired.

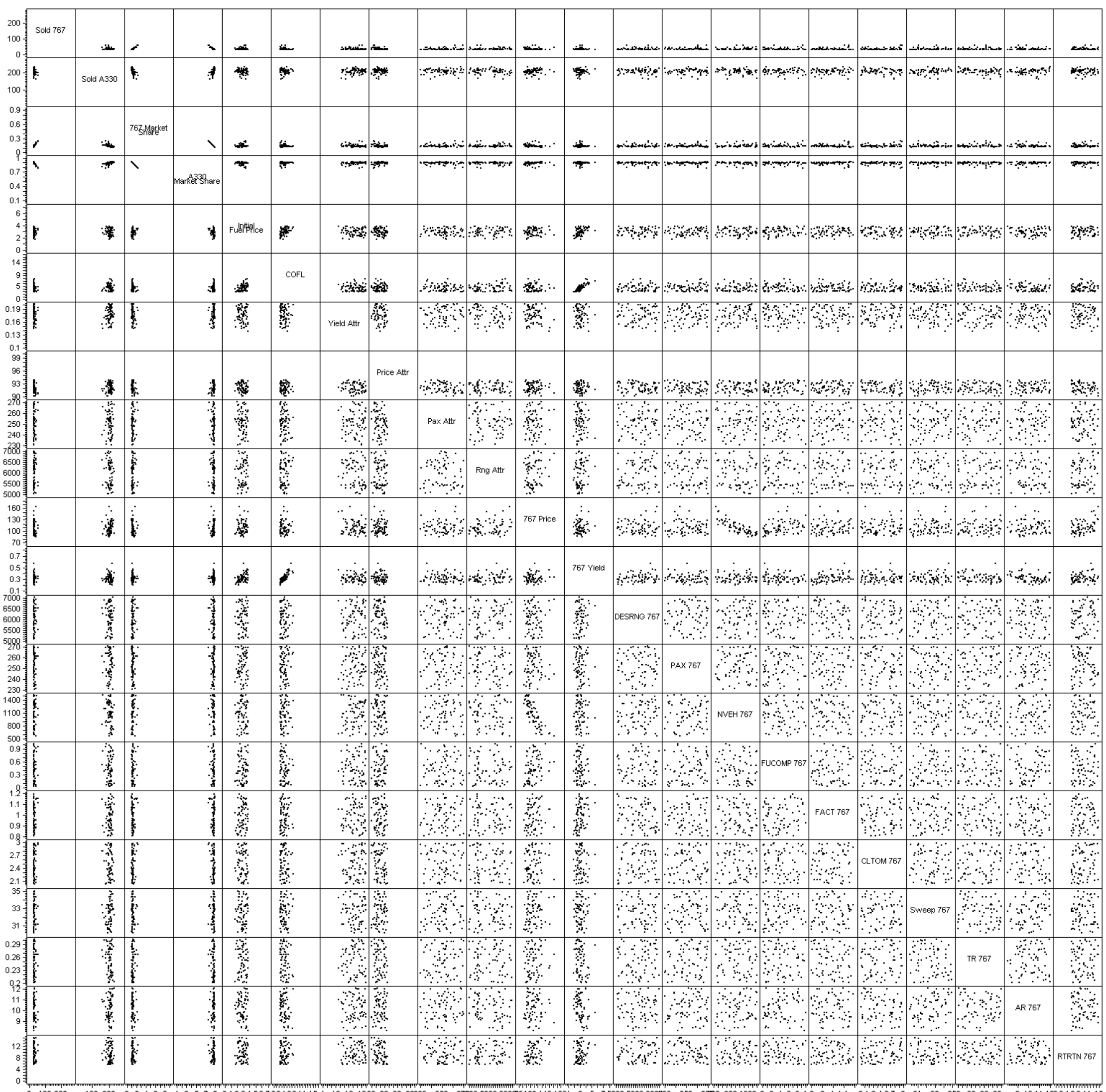


Figure 59: Final Multivariate Scatter Plot Environment

7.2.2 Viable Aircraft

Therefore, it is now time to discuss any pertinent features of the remaining design points. This is accomplished by looking at the lower right ten variables of the final scatter plot in Figure 59. The trends that are visible show that there seems to be a trend in wing aspect ratio of around ten and a sweep around 31 degrees, which agrees with the original designs that were used as the basis of this study. Furthermore, the design ranges of the remaining aircraft are all scattered around a middle range of 6000nm. These trends are most likely due to the inherent optimizations in the underlying aircraft represented in the surrogate models. It should be noted that the remaining design points can be selected in JMP and the specific number of the point be identified. This point then can be easily looked up in the underlying table of cases and specific numeric values can be identified.

It can also be observed that the remaining market scenarios between the A330-200 and the 767-400ER generally follow the trend of the actual market data. This means that the market model is consistent with reality. Furthermore, it shows that the decisions made in selecting and excluding certain aspects of the market were consistent. Additionally, the points representing the upper end of the market share for the 767-400ER are those modified designs that include significant technology improvements such as the composite fuselage and the reduced fuel burn. This is again consistent with the results shown in the EUTE earlier.

7.2.3 Insights

Creating a large set of variant aircraft designs, each with an according market situation, was achieved by exporting the results of the Monte Carlo simulation performed in the EUTE environment. Only small changes had to be made in the assumptions driving the input probability distributions. Especially, affected were the uniform probability distributions assigned to the aircraft design variables. A uniform probability distribution on design choices represents the equal desirability of design choices. This assumption should be true in most cases, at least in the strictly limited ranges prescribed here.

The process that was demonstrated here shows how step-by-step undesirable designs and scenarios can be easily hidden because they are not currently of significance or are to be explored later. This process of elimination is useful for identifying aircraft designs that can be successful under a number of market conditions. This is especially helpful in being able to identify which designs have the highest chance of being successful in the market against a competitor. The result is that this process enables decision makers identify which choices to make to have the highest chance at a successful program thereby enabling increased understanding about the potential market success already at the concept stage.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

After demonstrating the proposed method on a specific example and illustrating the potential uses, it is now time to draw some conclusions from the results and present contributions to the field.

8.1 Contributions to Aerospace Systems Design

This thesis identified the need for a market-oriented approach to aircraft design. This is exemplified by the focus on conventional design approaches in recent history and the focus on continual improvement of well-known systems, which means that a large number of potential choices in aircraft design are removed simply because these choices venture into unknown territory with no guarantee of success. This premature rejection of many revolutionary concepts has the potential to leave out a great deal of promising ideas.

The key idea is that the fear of the unknown is driven by simple economic considerations that place a negative value on risk due to the high risk of failure. A comparison to capital investment shows that there is a direct relation between risk and the potential rate of return. When the risk is high the rate of return can take a broad range of values. The maximum potential payoff is high and the minimum potential payoff is low or even negative. At the other end, when the risk is low the maximum potential payoff is significantly reduced, but at the same time the minimum potential payoff is much higher, to the point that it is possible to guarantee that it remains positive.

The same is true for revolutionary aircraft designs. An incremental improvement of an existing design guarantees a higher minimum rate of return, whereas a truly revolutionary design has a high potential for a negative rate of return, but also has a much higher potential maximum rate of return.

Therefore, this market-based approach can be used to differentiate aircraft designs that have the chance of performing exceptionally in a competitive market. Furthermore, it is possible to identify the impact of scheduling risks making a direct trade-off of technology inclusion or exclusion and market introduction dates possible.

At the core of this thesis are a number of research questions and hypotheses that should now be able to be answered. For this purpose an example problem was introduced throughout the text to demonstrate the methodology of the process. The problem was selected such that it represents a currently relevant aircraft design problem while at the same time having enough publicly available data to be able to calibrate and validate the model. This example problem was chosen to be the competing market entries between Boeing and Airbus in the 250 seat category wide-body aircraft market represented by the 767-400ER and the A330-200 in the late 1990s and the early 21st century. This market was modeled and analyzed with existing data about each aircraft and delivery numbers. The creation of this model and the subsequent exploration of its results and capabilities lead to answers to the questions posed at the beginning of this endeavor. Which now finally brings us to the answers of the research questions and the hypotheses posed earlier.

Question 1 asked *Can an aircraft analysis be integrated directly or indirectly in such a way that the system dynamics model can be calibrated and produces stable solutions while at the same time being computationally feasible?* This question is based on the literature review of available market models. After it was found that most of these models are lacking in accuracy and predictive power a further review of differing types of analyses, which allow essential feedback mechanisms to be modeled in a comprehensive way, was conducted. This found that system dynamics is the best method of modeling such competitive market models with even a number of existing models available. This directly leads to **Hypothesis 1** that proposed that aerospace analyses such as aircraft design can be integrated into a system dynamics competitive market model. This was not possible in an analytic method. However, indirect integration through the use of surrogate models proved successful. In the direct follow-on to this **Hypothesis 2** proposed that such a model has solutions and they are computationally feasible. This was proven to be the case with standard system dynamics tools. Again this was only possible due to the well-behaved nature and the algebraic form of the surrogate models representing the aircraft design analysis. An important caveat, however, is that the ranges of the design parameters used to create the surrogate models also have to be observed in the system dynamics model. This means that much care has to be taken to impose strict limits on the ranges of said variables. The selected Polya process overall exhibits a very stable behavior over all the possible ranges in the model variables of interest. A large number of system dynamics models that can exhibit very unstable behavior due to rapid changes in key process rates, which means that they have to be extensively calibrated and mechanically checked for stability and consistency of behavior at extreme settings. However, this was not the case here since the Polya process model automatically guarantees market share numbers between 0% and 100%.

The integration of surrogate models was therefore demonstrated not only hypothetically, but also accomplished by building a functional market model. The next hypothesis, **Hypothesis 3**, proposed that the resulting market model could be calibrated to actual market data. The specific market that was chosen was the competition between the Airbus A330-200 and the Boeing 767-400ER. The available delivery data was then used to calibrate the available market model constants. The result was that the model could successfully infer market share and individual sales, given an external demand. The demand was left as an external driver due to the fact that high fidelity models already exist in that field due to the significant interest of airplane manufacturers in future market outlooks.

Question 2 asked *Can this integrated model be used for relevant future scenario forecasting and enable a comprehensive overview in a portfolio of the effects of future decisions or policies?* After trying to identify which future decisions and policies would be available for an aircraft manufacturer, it became clear that these mostly consist of market entrance and exit timing as well as design decisions regarding shape and performance as well as technological improvements. To be able to answer this question a whole new approach was needed. Therefore, **Hypothesis 4** proposed a new multivariate solution to this problem where different scenarios can be easily explored and evaluated. This was addressed with the Extended Unified Trade-off Environment, which was first constructed as a concept and then later implemented. Initially it was thought that this would be possible in a commercial software tool, but this turned out not to be the case. Therefore, a custom environment was assembled relatively rapidly through the use of existing libraries that were tied together to provide a seamless environment. This environment, however, remains a prototype and is far from being a complete product. However, it served its purpose as a demonstration environment that was able to show that every specific position of one of the hairlines of one of the scenario defining variables such as the fuel price ramp slope in its prediction profiler represents a specific scenario. **Hypothesis 5** proposed that this solution space can be explored using existing techniques. This was shown to be possible because the required underlying analysis to generate the Extended Unified Trade-off Environment produced all the required data to generate a multivariate scatter plot. This in return enables the capabilities afforded by the inverse design methodology shown previously. Furthermore, the same interactive environment also contains information about the future likelihood of different potential outcomes. These are shown as confidence intervals around the mean value. The result is a "corridor" of probable outcomes for each of the tracked measures of merit, which were total overall sales and market share. This is exactly what was proposed by **Hypothesis 6**.

Question 3 asked *What is the result in overall model behavior of diverging time rates of change in various elements, especially in the face of rapid changes in oil price or accelerating technology development?* This is a meaningful question to ask in a current environment where increasing changes in external developments and drivers pose more and more strain on what is required of a new system, especially in aviation. Therefore, **Hypothesis 7** proposed that it would be possible to use this methodology to analyze the response to rapid changes in external drivers. The inclusion of technology factors such as the fuel efficiency and the introduction of new materials to entire components of an aircraft allow this type of analysis to be conducted. In the Extended Unified Trade-off Environment it its possible to directly see the partial derivatives of each of the variables, including the external drivers. This enables the user of the environment to select specific values corresponding to rapid increases in fuel price. After this it is then possible to find values of technological improvements that counteract the effect of the increased fuel price on the market.

This lead to some potential lessons that were learned during the progression of this research effort and some potential improvements that can be achieved with further improvements to the methodology.

8.2 Lessons and Potential Improvements

One of the lessons learned is the difficulty of constructing a meaningful aircraft model that is useful for further analysis in the market model. Additionally, this methodology also depends on being able to model the competing aircraft performance with sufficient accuracy. This leads to another difficulty that was encountered during the development of this process. The availability of design variable for multiple aircraft very quickly leads to an enormous number of variables. While the available methods can in principle deal with this large number of variables, the problem becomes a practical problem due to the effort that has to be undertaken in controlling and managing the appropriate ranges for such large amounts of data.

Furthermore, there exists also a computational limitation due to this. The high number of data points visible at any given time on the screen quickly leads to very unacceptable delays in the display of the required plots. Additionally, the number of plots shown quickly overwhelms not only the user but also the size and resolution of common displays and printers. Only very large and high-resolution screens are capable of viewing and using the entire environment at the same time.

As an intermediate solution it was chosen to pose a specific scenario of presenting only the variables of a single aircraft and only a limited number of variables associated with the market model. However, depending on the requirements of a specific user there might be a need to be able to quickly change these settings. There is a definite potential for improvement in this direction.

Another further improvement would be to, instead of having demand as an external model driver, incorporate the system dynamics models for the demand that already exist into this environment. With this combined environment it would then be possible to analyze the direct impact of revolutionary vehicles on the demand and vice versa. This could then further improve the credibility of such an analysis, because it would be able to show if a new and revolutionary vehicle would be able to be a commercial success or not.

8.3 *New Research Paths*

After presenting some ways of improving the methodology that was presented, it is now time to focus on how entirely new research paths can be opened up by the proposed method. For example, it will now be possible to evaluate directly how revolutionary new vehicle concepts will perform in a competitive environment. At the same time this methodology serves as an enabler for concept selection. One caveat to this, however, is the need for detail analysis that is required to be able to narrow the uncertainty especially on new and revolutionary concepts.

There is also the possibility to further explore the evaluation of technologies and technology portfolios that can be added to the competing aircraft. The timing and cost of development together with the technology impacts on the system it would then be possible to explore questions about the cost-benefit ratio for each technology or a set of technologies. This knowledge could then be used to make more informed technology investment decisions, helping decision makers focus resources on the most promising and cost effective solutions.

Furthermore, this market model could also be integrated with a manufacturer production model. Such a model would replicate the production line capabilities and also be able to capture design and engineering efforts on new programs. This would then enable a decision maker to explore not only the effect of vehicle technologies, but also the impact of production and process technologies as well as changes in the company structure and improvements in cycle reduction time. The result of this could be concepts for improving the corporate organization and overall organization of a company, helping to improve the overall aircraft development process.

In summary, there are a number of highly important areas that this market model can be integrated. This is especially the case in business processes and organizational structure. Applications such as the ones shown can be used to further extended the market model presented here and explore some of the most pressing questions for the leaders of the aerospace industry while still staying firmly rooted in the technical aspects of aerospace design and engineering efforts.

APPENDIX A

AIRCRAFT BASELINES

A.1 Airbus A330-200 Baseline

Listing A.1: A330-200.in

```
A330-200 253pax w/NASA Lewis Engine
 $OPTION
  MPRINT=1,
  IOPT=1,
  IANAL=3,
  INENG=0,
  ITAKOF=0,
  ILAND=0,
  NOPRO=0,
  NOISE=0,
  ICOST=2,
  IFITE=0,
  IPOLP=0,
 $END
 $WTIN
  MYWTS=0,
  VMMO=0.86,
  DIH=6.0,
  XL=193.6,
  WF=18.5,
  DF=18.5,
  WLDG=396825.0,
  SHT=960.8,
  SWPHT=30.0,
  ARHT=4.2,
  TRHT=0.25,
  TCHT=0.11,
  SVT=571.1,
  SWPVT=40.0,
  ARVT=1.6,
  TRVT=0.32,
  TCVT=0.09,
  NEW=2,
  THRSO=58468.7,
  WENG=11545.5,
  XNAC=15.13,
  DNAC=12.1,
  FULFMX=86512.32,
  NPF=12,
  NPT=241,
```

```
 NFLCR=2,
 WPPASS=177.,
 FRFU=1.038,
 WTHR=0.0,
 WPMSC=0.0,
 MLDWT=1,
$END
$CONFIN
 DESRNG=6650.0,
 HTVC=1.,
 VTVC=1.,
 OFG=1.,
 OFF=0.,
 GW=507060.0,
 AR=10.1,
 TR=0.267,
 SWEEP=31.5,
 TCA=.1175,
 VCMN=0.82,
 CH=41100.0,
 THRUST=64530.0, 1.0,
 SW=3875.05, 1.0,
$END
$AERIN
 CAM=2.0,
 AITEK=2.00,
 E=0.80,
 VAPPR=145.0,
 FLTO=12000.0,
 FLLDG=12000.0,
 CLTOM=2.3,
 CLLDM=3.0,
$END
$ENGDIN
 IGENEN=-1,
 EIFILE="225pax.flops-engdata",
 IDLE=1,
 MAXCR=1,
 NOX=1,
$END
$IWGT
 IPRINTE= 1,
 IACOST= 0,
 FENGQ= 2000.,
 PWINGAL= 0.9,
 PWINGTI= 0.0,
 PWINGCO= 0.1,
 PWEMPAL= 0.7,
 PWEMPTI= 0.0,
 PWEMPCO= 0.3,
 PWBODYAL= 0.9,
 PWBODYTI= 0.0,
 PWBODYCO= 0.1,
 PWLGAL= 0.5,
 PWLGTI= 0.5,
```

```
  PWLGCO=  0.0 ,
  PWNACAL=  0.1 ,
  PWNACTI=  0.0 ,
  PWNACCO=  0.9 ,
  AKRDTE=  0.0 ,
  AKOANDS=  0.0 ,
  AKPRICE=  0.0 ,
 $END
 $CMAN
  IRDTE=  1 ,
  YEAR=   2005 ,
  PYEAR=   1987 ,
  API=   0.03 ,
  RTRTN=  5 ,
  RTRTNA=  3. ,
  CFENG=  1.0 ,
  CFAVON=  0.36 ,
  ENSPAO=  .23 ,
  FEE=  0.0 ,
  IAIRROI=  1 ,
  ICSHFLW=  1 ,
  IAIRPL=  3 ,
  ICONFG=  6 ,
  IPROD=  1 ,
  IENGS=  1 ,
  IOPS=  1 ,
  REVFLAG=  0 ,
  NEWTECHF=  0 ,
  HMVFLAG=  0 ,
  LEARN1=  79. ,
  LEARN2=  85.0 ,
  LEARNA1=  78.5 ,
  LEARNA2=  85.0 ,
  LEARNAS1=  78.5 ,
  LEARNAS2=  81. ,
  LEARNFE1=  78.5 ,
  LEARNFE2=  85. ,
  LEARNP1=  100. ,
  LEARNP2=  100. ,
  PUNITS=   830. ,
  NV=  830. ,
  NVEH=   1. ,  830. ,
  RATE=   18*1 ,
  RE=  89.68 ,
  RT=  54.68 ,
$END
$COPER
  RL=  25.0 ,
  IYIELD=    0 ,
  YFACT=    0.01 ,
  CLF=    0.716 ,
  COFL=    0.865 ,
  CFAFRM=   1.0 ,
  FINSUR=    .35 ,
  BDMAIN=  200.0 ,
```

```
  FLF=     0.75,
  ECLIFE=     20.,
  HSUB=     38000.,
  DWNPYM=     0.,
  NSL=     2,
  RESDVL=   10.0,
  SL=   3092.0,1500.,
  U=     3900.0,
  SUBMACH=   0.82,
  SUBL=     1.0,
  SUPL=     0.0,
  RINRST=   8.0,
  GRNDTM=     2.,
  IAFMANT=     0,
  IEMAINT=     0,
  IBLOCK=     0,
$END
$IMAINT
  MTBF=   10000,
  MTTR=   1,
  CF1=     1.6,
  PNAC=     2.,
  IOXG=     2,
$END
$RDTE
  CFWAL= 1.0,
  IRDTEPRT=   0,
  WLFCLE= 0.,
  WLFCMAN= 0.,
  WLFCPN= 0.,
  WFLTPROV= 482.,
  WMISPROV= 0.,
  CFBODYTF= 1.0,
  CFFUSMAT= 1.0,
  CFBODBFL= 1.0,
  CFIACBFL= 1.0,
$END
$MISSIN
  NPCON=1,
  FACT=1.032,
  IFLAG=2,
  IRW=1,
  ITTFF=1,
  TAKOTM=2.0,
  TAXOTM=9.0,
  APPRTM=4.0,
  TAXITM=5.0,
  NCLIMB=1,
  CLAMIN=0.0,
  FWF=-.001,
  RCIN=300.0,
  NCRUSE=2,
  CRMACH=0.82, 0.6,
  CRALT=38000.0, 25000.0,
  IOC=1, 4,
```

```
  IVS=1,
  DEAMIN=0.0,
  IRS=1,
  TIMMAP=2.0,
  ALTRAN=150.0,
  NCLRES=1,
  NCRRES=2,
  HOLDTM=45.0,
  NCRHOL=1,
  IHOPOS=2,
  ICRON=0,
  IATA=0,
$END
START
CLIMB
CRUISE
DESCENT
END
```

A.2 Boeing 767-400ER Baseline

Listing A.2: 767-400ER.in

```
767-400ER 243pax w/NASA Lewis Engine
 $OPTION
  MPRINT=1,
  IOPT=1,
  IANAL=3,
  INENG=0,
  ITAKOF=0,
  ILAND=0,
  NOPRO=0,
  NOISE=0,
  ICOST=2,
  IFITE=0,
  IPOLP=0,
 $END
 $WTIN
  MYWTS=0,
  VMMO=0.84,
  DIH=6.0,
  XL=197.11,
  WF=16.5,
  DF=16.5,
  WLDG=350000.0,
  SHT=818.4,
  SWPHT=38.0,
  ARHT=4.57,
  TRHT=0.235,
  TCHT=0.11,
  SVT=545.8,
  SWPVT=40.0,
```

```
  ARVT=1.71,
  TRVT=0.273,
  TCVT=0.09,
  NEW=2,
  THRSO=58468.7,
  WENG=11545.5,
  XNAC=14.592,
  DNAC=9.17,
  FULFMX=50890.0,
  NPF=16,
  NPT=227,
  NFLCR=2,
  WPPASS=177.,
  FRFU=1.2,
  FRWI=1.45,
  FRHT=1.2,
  FRVT=1.1,
  FRLGN=1.1,
  FRLGM=1.1,
  FRNA=1.3,
  WTHR=0.0,
  WPMSC=0.0,
  MLDWT=1,
$END
$CONFIN
  DESRNG=5645.0,
  HTVC=1.0,
  VTVC=1.0,
  OFG=1.,
  OFF=0.,
  GW=451000.0,
  AR=9.3,
  TR=0.23,
  SWEEP=31.5,
  TCA=.1409,
  VCMN=0.8,
  CH=41000.0,
  THRUST=63500.0,  1.0,
  SW=3129.0,  1.0,
$END
$AERIN
  CAM=2.0,
  AITEK=2.00,
  E=0.80,
  VAPPR=145.0,
  FLTO=9000.0,
  FLLDG=9000.0,
  CLTOM=2.3,
  CLLDM=3.0,
$END
$ENGDIN
  IGENEN=-1,
  EIFILE="225pax.flops-engdata",
  IDLE=1,
  MAXCR=1,
```

```
 NOX=1,
$END
$IWGT
 IPRINTE= 1,
 IACOST= 0,
 FENGQ= 2100.,
 PWINGAL= 0.9,
 PWINGTI= 0.0,
 PWINGCO= 0.1,
 PWEMPAL= 0.9,
 PWEMPTI= 0.0,
 PWEMPCO= 0.1,
 PWBODYAL= 0.9,
 PWBODYTI= 0.0,
 PWBODYCO= 0.1,
 PWLGAL= 0.5,
 PWLGTI= 0.5,
 PWLGCO= 0.0,
 PWNACAL= 0.1,
 PWNACTI= 0.0,
 PWNACCO= 0.9,
 AKRDTE= 0.0,
 AKOANDS= 0.0,
 AKPRICE= 0.0,
$END
$CMAN
 IRDTE= 1,
 YEAR=  2005,
 PYEAR=  1978,
 API=  0.03,
 RTRTN= 10,
 RTRTNA= 3.,
 CFENG= 1.0,
 CFAVON= 0.36,
 ENSPAO= .23,
 FEE= 0.0,
 IAIRROI= 1,
 ICSHFLW= 1,
 IAIRPL= 3,
 ICONFG= 6,
 IPROD= 1,
 IENGS= 1,
 IOPS= 1,
 REVFLAG= 0,
 NEWTECHF= 0,
 HMVFLAG= 0,
 LEARN1= 82.5,
 LEARN2= 85.0,
 LEARNA1= 82.,
 LEARNA2= 85.0,
 LEARNAS1= 82.5,
 LEARNAS2= 79.,
 LEARNFE1= 82.,
 LEARNFE2= 85.,
 LEARNP1= 100.,
```

```
  LEARNP2= 100.,
  PUNITS=  1049.,
  NV= 1049.,
  NVEH=  1., 1049.,
  RATE= 15*1,
  RE= 89.68,
  RT= 54.68,
$END
$COPER
  RL= 25.0,
  IYIELD=   0,
  YFACT=   0.01,
  CLF=   0.716,
  COFL=   0.865,
  CFAFRM=  1.0,
  FINSUR=   .35,
  BDMAIN= 200.0,
  FLF=   0.75,
  ECLIFE=   20.,
  HSUB=   38000.,
  DWNPYM=   0.,
  NSL=   2,
  RESDVL=  10.0,
  SL=  3092.0,1500.,
  U=   3900.0,
  SUBMACH=  0.8,
  SUBL=   1.0,
  SUPL=   0.0,
  RINRST=  8.0,
  GRNDTM=   2.,
  IAFMANT=   0,
  IEMAINT=   0,
  IBLOCK=   0,
$END
$IMAINT
  MTBF=  10000,
  MTTR=  1,
  CF1=   1.6,
  PNAC=   2.,
  IOXG=   2,
$END
$RDTE
  CFWAL= 1.0,
  IRDTEPRT=  0,
  WLFCLE= 0.,
  WLFCMAN= 0.,
  WLFCPN= 0.,
  WFLTPROV= 482.,
  WMISPROV= 0.,
  CFBODYTF= 1.0,
  CFFUSMAT= 1.0,
  CFBODBFL= 1.0,
  CFIACBFL= 1.0,
$END
$MISSIN
```

```
  NPCON=1,
  FACT=0.85,
  IFLAG=2,
  IRW=1,
  ITTFF=1,
  TAKOTM=2.0,
  TAXOTM=9.0,
  APPRTM=4.0,
  TAXITM=5.0,
  NCLIMB=1,
  CLAMIN=0.0,
  FWF=-.001,
  RCIN=300.0,
  NCRUSE=2,
  CRMACH=0.8, 0.6,
  CRALT=38000.0, 25000.0,
  IOC=1, 4,
  IVS=1,
  DEAMIN=0.0,
  IRS=1,
  TIMMAP=2.0,
  ALTRAN=200.0,
  NCLRES=1,
  NCRRES=2,
  HOLDTM=45.0,
  NCRHOL=1,
  IHOPOS=2,
  ICRON=0,
  IATA=0,
$END
START
CLIMB
CRUISE
DESCENT
END
```

A.3 Baseline Engine Deck

Listing A.3: 225pax.flops-engdata

```
 0.00      0.0    23407.1     0.0    7484.9          10.905   3967.5
 0.00      0.0    35089.5     0.0   11569.7          17.591   3967.5
 0.00      0.0    46781.7     0.0   16006.2          25.754   3967.5
 0.00      0.0    49710.2     0.0   17153.0          27.989   3967.5
 0.00      0.0    52630.5     0.0   18325.9          30.335   3967.5
 0.00      0.0    55545.0     0.0   19523.1          32.783   3967.5
 0.00      0.0    58468.7     0.0   20760.7          35.401   3967.5
 0.00   2000.0    22459.7    37.9    7138.6          10.399   3967.5
 0.00   2000.0    33669.2    46.0   11059.0          16.842   3967.5
 0.00   2000.0    44883.3    52.8   15302.4          24.629   3967.5
 0.00   2000.0    47689.8    54.3   16406.1          26.774   3967.5
 0.00   2000.0    50501.3    55.8   17545.7          29.058   3967.5
 0.00   2000.0    53298.5    57.2   18696.3          31.418   3967.5
```

0.00	2000.0	56097.0	58.6	19914.1	34.038	3967.5
0.00	5000.0	21007.3	34.7	6615.3	9.638	3967.5
0.00	5000.0	31485.3	42.1	10280.0	15.683	3967.5
0.00	5000.0	41978.9	48.2	14236.7	22.913	3967.5
0.00	5000.0	44597.5	49.6	15271.1	24.928	3967.5
0.00	5000.0	47219.1	51.0	16332.7	27.034	3967.5
0.00	5000.0	49844.3	52.3	17453.0	29.384	3967.5
0.00	5000.0	52469.5	53.5	18657.1	32.035	3967.5
0.00	10000.0	18699.0	29.7	5806.4	8.478	3967.5
0.00	10000.0	28024.2	36.0	9065.7	13.859	3967.5
0.00	10000.0	37359.8	41.3	12567.1	20.242	3967.5
0.00	10000.0	39701.4	42.5	13505.3	22.065	3967.5
0.00	10000.0	42043.2	43.6	14504.4	24.135	3967.5
0.00	10000.0	44369.5	44.7	15571.9	26.440	3967.5
0.00	10000.0	46712.2	45.7	16678.2	28.910	3967.5
0.10	0.0	25357.8	4176.5	7935.6	11.601	3967.5
0.10	0.0	36750.9	4988.9	11960.3	18.295	3967.5
0.10	0.0	48014.0	5664.8	16261.8	26.264	3967.5
0.10	0.0	50829.7	5820.1	17373.3	28.439	3967.5
0.10	0.0	53637.6	5968.5	18507.5	30.720	3967.5
0.10	0.0	56422.3	6111.6	19653.0	33.083	3967.5
0.10	0.0	59204.1	6251.8	20840.5	35.589	3967.5
0.10	2000.0	24307.8	3941.2	7566.1	11.060	3967.5
0.10	2000.0	35242.4	4708.3	11436.2	17.510	3967.5
0.10	2000.0	46066.5	5347.0	15556.9	25.131	3967.5
0.10	2000.0	48771.6	5492.3	16627.8	27.235	3967.5
0.10	2000.0	51454.1	5632.9	17722.0	29.433	3967.5
0.10	2000.0	54128.2	5766.1	18815.8	31.685	3967.5
0.10	2000.0	56784.8	5897.6	19989.7	34.203	3967.5
0.10	5000.0	22699.3	3600.6	7011.3	10.254	3967.5
0.10	5000.0	32936.6	4301.9	10636.1	16.305	3967.5
0.10	5000.0	43067.3	4885.0	14478.3	23.393	3967.5
0.10	5000.0	45585.3	5018.1	15481.0	25.358	3967.5
0.10	5000.0	48100.0	5146.0	16506.1	27.423	3967.5
0.10	5000.0	50603.5	5268.7	17579.7	29.662	3967.5
0.10	5000.0	53108.6	5384.9	18730.4	32.201	3967.5
0.10	10000.0	20157.9	3081.9	6153.1	9.022	3967.5
0.10	10000.0	29283.5	3683.1	9367.3	14.413	3967.5
0.10	10000.0	38324.1	4182.0	12796.2	20.699	3967.5
0.10	10000.0	40551.0	4295.2	13696.0	22.462	3967.5
0.10	10000.0	42791.0	4400.4	14661.1	24.466	3967.5
0.10	10000.0	44996.3	4502.6	15678.6	26.676	3967.5
0.10	10000.0	47231.6	4599.4	16744.7	29.063	3967.5
0.20	0.0	28184.5	8806.8	8337.3	12.273	3967.5
0.20	0.0	39381.9	10334.8	12331.3	19.008	3967.5
0.20	0.0	50339.7	11609.8	16556.7	26.930	3967.5
0.20	0.0	53067.9	11901.8	17647.8	29.084	3967.5
0.20	0.0	55777.6	12184.5	18755.9	31.329	3967.5
0.20	0.0	58478.8	12454.3	19879.4	33.666	3967.5
0.20	0.0	61162.4	12719.3	21028.2	36.108	3967.5
0.20	2000.0	26968.0	8304.2	7950.5	11.705	3967.5
0.20	2000.0	37728.6	9747.2	11793.7	18.200	3967.5
0.20	2000.0	48255.2	10952.5	15847.1	25.783	3967.5
0.20	2000.0	50874.8	11228.6	16898.4	27.863	3967.5
0.20	2000.0	53480.5	11494.1	17970.3	30.039	3967.5

0.20	2000.0	56067.0	11747.4	19043.0	32.267	3967.5
0.20	2000.0	58637.9	11996.0	20174.7	34.702	3967.5
0.20	5000.0	25120.6	7577.9	7368.7	10.854	3967.5
0.20	5000.0	35207.3	8899.7	10976.5	16.958	3967.5
0.20	5000.0	45079.6	10002.8	14764.2	24.031	3967.5
0.20	5000.0	47527.0	10255.4	15752.5	25.978	3967.5
0.20	5000.0	49970.1	10496.7	16759.0	28.035	3967.5
0.20	5000.0	52388.7	10729.1	17798.9	30.209	3967.5
0.20	5000.0	54803.9	10949.3	18908.8	32.663	3967.5
0.20	10000.0	22203.8	6470.0	6461.0	9.541	3967.5
0.20	10000.0	31190.2	7604.7	9668.1	14.966	3967.5
0.20	10000.0	39997.7	8553.4	13047.0	21.250	3967.5
0.20	10000.0	42191.7	8765.3	13940.7	23.030	3967.5
0.20	10000.0	44347.7	8966.3	14874.8	24.970	3967.5
0.20	10000.0	46485.7	9159.0	15866.5	27.136	3967.5
0.20	10000.0	48637.9	9342.0	16901.0	29.462	3967.5
0.30	0.0	31929.8	14066.8	8727.2	13.000	3967.5
0.30	0.0	43023.8	16217.6	12714.7	19.827	3967.5
0.30	0.0	53759.4	18012.6	16916.3	27.801	3967.5
0.30	0.0	56412.9	18424.1	17989.6	29.970	3967.5
0.30	0.0	59045.7	18822.2	19078.3	32.217	3967.5
0.30	0.0	61664.4	19203.8	20184.5	34.543	3967.5
0.30	0.0	64269.9	19577.5	21305.3	36.953	3967.5
0.30	2000.0	30503.4	13252.0	8326.7	12.403	3967.5
0.30	2000.0	41145.0	15282.4	12165.2	18.983	3967.5
0.30	2000.0	51466.1	16982.8	16193.2	26.647	3967.5
0.30	2000.0	54018.6	17369.7	17235.8	28.736	3967.5
0.30	2000.0	56546.3	17745.8	18287.2	30.904	3967.5
0.30	2000.0	59064.4	18105.3	19348.6	33.136	3967.5
0.30	2000.0	61568.6	18456.9	20446.6	35.513	3967.5
0.30	5000.0	28361.2	12080.5	7727.3	11.514	3967.5
0.30	5000.0	38343.6	13942.9	11338.9	17.717	3967.5
0.30	5000.0	48036.7	15502.5	15117.8	24.889	3967.5
0.30	5000.0	50431.9	15859.2	16096.0	26.855	3967.5
0.30	5000.0	52815.5	16202.7	17089.8	28.893	3967.5
0.30	5000.0	55173.4	16523.5	18094.8	31.023	3967.5
0.30	5000.0	57492.5	16840.6	19174.1	33.417	3967.5
0.30	10000.0	24913.7	10284.5	6768.1	10.103	3967.5
0.30	10000.0	33818.7	11886.8	9982.1	15.633	3967.5
0.30	10000.0	42458.2	13224.2	13354.3	22.010	3967.5
0.30	10000.0	44582.1	13526.9	14228.8	23.764	3967.5
0.30	10000.0	46691.9	13816.4	15146.5	25.663	3967.5
0.30	10000.0	48788.9	14088.7	16122.8	27.817	3967.5
0.30	10000.0	50876.5	14347.2	17135.1	30.110	3967.5
0.30	15000.0	18807.1	8115.7	4745.9	6.094	3967.5
0.30	15000.0	25392.6	9361.9	6929.5	9.068	3967.5
0.30	15000.0	31778.8	10407.2	9208.1	12.417	3967.5
0.30	15000.0	33354.3	10645.3	9795.4	13.321	3967.5
0.30	15000.0	34918.0	10875.9	10392.8	14.254	3967.5
0.30	15000.0	36471.6	11094.8	10988.6	15.200	3967.5
0.30	15000.0	38013.2	11310.1	11620.6	16.232	3967.5
0.40	0.0	36792.7	20137.0	9143.6	13.860	3967.5
0.40	0.0	47778.3	22807.4	13130.6	20.784	3967.5
0.40	0.0	58336.0	25045.0	17311.7	28.928	3967.5
0.40	0.0	60918.0	25557.8	18384.0	31.113	3967.5

0.40	0.0	63487.2	26050.2	19479.1	33.371	3967.5
0.40	0.0	66042.2	26529.4	20566.3	35.723	3967.5
0.40	0.0	68584.6	26998.4	21674.8	38.152	3967.5
0.40	2000.0	34998.3	18930.6	8705.5	13.188	3967.5
0.40	2000.0	45570.3	21466.4	12554.4	19.897	3967.5
0.40	2000.0	55733.1	23589.3	16576.6	27.728	3967.5
0.40	2000.0	58224.6	24074.2	17616.9	29.839	3967.5
0.40	2000.0	60703.5	24541.9	18668.9	32.032	3967.5
0.40	2000.0	63165.2	24994.6	19727.3	34.295	3967.5
0.40	2000.0	65617.2	25438.2	20806.7	36.661	3967.5
0.40	5000.0	32452.6	17236.7	8089.9	12.254	3967.5
0.40	5000.0	42379.0	19561.9	11711.1	18.603	3967.5
0.40	5000.0	51925.5	21513.8	15507.8	25.938	3967.5
0.40	5000.0	54277.1	21957.5	16488.5	27.943	3967.5
0.40	5000.0	56609.5	22389.5	17481.0	30.025	3967.5
0.40	5000.0	58921.7	22799.5	18474.7	32.153	3967.5
0.40	5000.0	61213.2	23200.8	19526.7	34.484	3967.5
0.40	10000.0	28357.1	14636.1	7085.4	10.742	3967.5
0.40	10000.0	37207.8	16642.2	10326.9	16.413	3967.5
0.40	10000.0	45741.1	18320.3	13717.7	22.964	3967.5
0.40	10000.0	47840.4	18703.2	14598.8	24.770	3967.5
0.40	10000.0	49921.4	19070.2	15507.3	26.683	3967.5
0.40	10000.0	51981.1	19413.0	16463.0	28.769	3967.5
0.40	10000.0	54022.5	19739.0	17452.8	31.022	3967.5
0.40	15000.0	21562.6	11589.4	4965.8	6.457	3967.5
0.40	15000.0	28093.1	13141.8	7153.5	9.472	3967.5
0.40	15000.0	34378.0	14448.0	9436.2	12.888	3967.5
0.40	15000.0	35924.2	14746.4	10020.9	13.805	3967.5
0.40	15000.0	37458.6	15035.4	10614.2	14.749	3967.5
0.40	15000.0	38983.1	15313.5	11211.6	15.712	3967.5
0.40	15000.0	40498.9	15585.1	11830.8	16.732	3967.5
0.40	20000.0	18702.6	9733.9	4322.2	5.629	3967.5
0.40	20000.0	24511.7	11063.2	6275.2	8.329	3967.5
0.40	20000.0	30111.6	12178.6	8306.8	11.362	3967.5
0.40	20000.0	31490.4	12434.3	8832.7	12.178	3967.5
0.40	20000.0	32857.8	12678.6	9373.7	13.044	3967.5
0.40	20000.0	34210.3	12909.7	9938.5	13.968	3967.5
0.40	20000.0	35550.9	13128.5	10532.5	14.981	3967.5
0.50	0.0	42678.1	27108.9	9567.3	14.846	3967.5
0.50	0.0	53563.9	30214.8	13549.9	21.901	3967.5
0.50	0.0	63959.7	32818.5	17743.9	30.219	3967.5
0.50	0.0	66508.4	33424.2	18833.7	32.464	3967.5
0.50	0.0	69056.5	34004.7	19923.1	34.824	3967.5
0.50	0.0	71570.2	34565.5	21016.0	37.232	3967.5
0.50	0.0	74061.3	35113.4	22121.4	39.713	3967.5
0.50	2000.0	40535.3	25469.1	9121.2	14.141	3967.5
0.50	2000.0	51018.3	28417.3	12973.3	20.984	3967.5
0.50	2000.0	61026.3	30884.0	17011.5	29.010	3967.5
0.50	2000.0	63468.4	31455.5	18075.5	31.157	3967.5
0.50	2000.0	65927.2	32005.1	19122.6	33.441	3967.5
0.50	2000.0	68342.5	32536.9	20172.8	35.773	3967.5
0.50	2000.0	70742.9	33058.8	21235.6	38.148	3967.5
0.50	5000.0	37484.3	23156.3	8482.8	13.149	3967.5
0.50	5000.0	47346.1	25869.6	12121.9	19.656	3967.5
0.50	5000.0	56764.3	28145.2	15939.2	27.243	3967.5

0.50	5000.0	59071.8	28667.4	16928.6	29.296	3967.5
0.50	5000.0	61368.2	29171.7	17925.6	31.421	3967.5
0.50	5000.0	63646.6	29660.2	18920.3	33.586	3967.5
0.50	5000.0	65915.6	30141.2	19941.2	35.859	3967.5
0.50	10000.0	32623.7	19629.4	7446.9	11.532	3967.5
0.50	10000.0	41466.5	21978.5	10736.9	17.416	3967.5
0.50	10000.0	49932.6	23948.1	14183.7	24.246	3967.5
0.50	10000.0	52008.9	24395.9	15057.5	26.076	3967.5
0.50	10000.0	54055.6	24833.6	15951.3	27.977	3967.5
0.50	10000.0	56086.2	25247.2	16871.7	29.992	3967.5
0.50	10000.0	58099.9	25650.2	17842.3	32.202	3967.5
0.50	15000.0	24944.3	15579.3	5202.4	6.890	3967.5
0.50	15000.0	31434.1	17391.8	7400.0	9.969	3967.5
0.50	15000.0	37633.5	18913.4	9688.7	13.483	3967.5
0.50	15000.0	39146.7	19262.6	10278.7	14.417	3967.5
0.50	15000.0	40653.4	19600.6	10874.6	15.378	3967.5
0.50	15000.0	42149.3	19927.3	11472.6	16.359	3967.5
0.50	15000.0	43640.8	20250.2	12078.8	17.370	3967.5
0.50	20000.0	21559.0	13063.1	4543.5	6.018	3967.5
0.50	20000.0	27352.2	14620.2	6522.6	8.806	3967.5
0.50	20000.0	32902.1	15926.6	8585.6	11.954	3967.5
0.50	20000.0	34265.2	16225.6	9112.7	12.794	3967.5
0.50	20000.0	35614.6	16517.2	9647.7	13.654	3967.5
0.50	20000.0	36949.4	16793.0	10192.8	14.548	3967.5
0.50	20000.0	38269.6	17063.1	10770.0	15.527	3967.5
0.50	25000.0	18447.2	10853.4	3928.4	5.210	3967.5
0.50	25000.0	23563.3	12176.3	5676.4	7.683	3967.5
0.50	25000.0	28467.9	13285.9	7504.0	10.462	3967.5
0.50	25000.0	29667.7	13539.3	7979.5	11.215	3967.5
0.50	25000.0	30853.3	13780.0	8477.4	12.031	3967.5
0.50	25000.0	32030.3	14011.0	8993.0	12.897	3967.5
0.50	25000.0	33202.5	14234.2	9517.8	13.805	3967.5
0.50	30000.0	15623.9	8930.2	3355.9	4.473	3967.5
0.50	30000.0	20075.7	10042.5	4875.8	6.618	3967.5
0.50	30000.0	24348.2	10970.2	6488.0	9.062	3967.5
0.50	30000.0	25389.7	11175.1	6925.5	9.785	3967.5
0.50	30000.0	26415.6	11373.5	7375.8	10.527	3967.5
0.50	30000.0	27439.2	11564.6	7836.9	11.314	3967.5
0.50	30000.0	28456.5	11750.5	8309.7	12.153	3967.5
0.60	15000.0	29013.5	20159.5	5461.7	7.409	3967.5
0.60	15000.0	35451.3	22177.9	7665.0	10.552	3967.5
0.60	15000.0	41576.4	23881.3	9980.3	14.182	3967.5
0.60	15000.0	43080.6	24281.4	10579.5	15.126	3967.5
0.60	15000.0	44595.4	24675.6	11173.2	16.122	3967.5
0.60	15000.0	46088.2	25060.4	11767.1	17.131	3967.5
0.60	15000.0	47567.7	25437.8	12370.1	18.154	3967.5
0.60	20000.0	24944.5	16862.6	4778.9	6.472	3967.5
0.60	20000.0	30726.7	18605.1	6785.8	9.353	3967.5
0.60	20000.0	36253.6	20092.5	8876.6	12.619	3967.5
0.60	20000.0	37622.6	20445.6	9408.9	13.488	3967.5
0.60	20000.0	38976.9	20788.5	9946.2	14.369	3967.5
0.60	20000.0	40317.2	21118.2	10487.5	15.262	3967.5
0.60	20000.0	41649.9	21442.2	11049.1	16.210	3967.5
0.60	25000.0	21248.2	13980.0	4135.6	5.604	3967.5
0.60	25000.0	26369.6	15467.9	5927.7	8.191	3967.5

0.60	25000.0	31286.6	16752.4	7775.4	11.059	3967.5
0.60	25000.0	32492.6	17053.4	8249.8	11.815	3967.5
0.60	25000.0	33687.2	17342.3	8741.6	12.622	3967.5
0.60	25000.0	34868.7	17615.1	9249.4	13.473	3967.5
0.60	25000.0	36040.3	17882.3	9773.4	14.379	3967.5
0.60	30000.0	17941.9	11485.2	3547.5	4.820	3967.5
0.60	30000.0	22423.9	12743.9	5116.6	7.090	3967.5
0.60	30000.0	26752.5	13841.0	6750.4	9.602	3967.5
0.60	30000.0	27801.4	14087.7	7185.6	10.317	3967.5
0.60	30000.0	28838.0	14324.7	7639.2	11.061	3967.5
0.60	30000.0	29872.7	14552.1	8097.6	11.871	3967.5
0.60	30000.0	30900.7	14784.6	8567.0	12.689	3967.5
0.60	35000.0	14932.0	9322.9	2989.0	4.081	3967.5
0.60	35000.0	18783.5	10376.2	4325.8	6.010	3967.5
0.60	35000.0	22492.8	11285.0	5757.7	8.212	3967.5
0.60	35000.0	23399.7	11487.7	6151.1	8.859	3967.5
0.60	35000.0	24304.1	11687.6	6549.0	9.547	3967.5
0.60	35000.0	25198.8	11882.7	6958.6	10.273	3967.5
0.60	35000.0	26081.6	12065.9	7397.0	11.079	3967.5
0.60	39000.0	12425.5	7730.4	2501.8	3.639	3967.5
0.60	39000.0	15649.2	8610.6	3618.5	5.355	3967.5
0.60	39000.0	18750.9	9363.5	4821.2	7.339	3967.5
0.60	39000.0	19508.1	9531.5	5149.2	7.926	3967.5
0.60	39000.0	20257.4	9695.5	5482.5	8.540	3967.5
0.60	39000.0	20999.8	9857.7	5826.0	9.194	3967.5
0.60	39000.0	21734.8	10007.6	6200.2	9.936	3967.5
0.70	15000.0	33832.9	25387.5	5751.0	8.034	3967.5
0.70	15000.0	40252.6	27586.9	7980.6	11.287	3967.5
0.70	15000.0	46402.4	29516.3	10312.3	15.015	3967.5
0.70	15000.0	47922.7	29979.8	10906.2	16.010	3967.5
0.70	15000.0	49429.4	30428.6	11503.5	17.016	3967.5
0.70	15000.0	50921.7	30866.2	12104.2	18.048	3967.5
0.70	15000.0	52404.1	31294.9	12712.7	19.098	3967.5
0.70	20000.0	28927.4	21179.7	5035.2	7.016	3967.5
0.70	20000.0	34716.7	23096.1	7071.4	9.994	3967.5
0.70	20000.0	40290.3	24804.4	9172.5	13.374	3967.5
0.70	20000.0	41650.9	25203.8	9710.3	14.257	3967.5
0.70	20000.0	43002.7	25592.3	10259.5	15.153	3967.5
0.70	20000.0	44354.9	25975.0	10803.6	16.083	3967.5
0.70	20000.0	45697.7	26350.3	11358.3	17.028	3967.5
0.70	25000.0	24547.4	17528.2	4371.5	6.079	3967.5
0.70	25000.0	29723.7	19195.1	6200.3	8.778	3967.5
0.70	25000.0	34718.6	20681.0	8088.5	11.778	3967.5
0.70	25000.0	35936.5	21025.8	8567.4	12.559	3967.5
0.70	25000.0	37142.4	21361.4	9056.7	13.360	3967.5
0.70	25000.0	38336.2	21681.8	9552.6	14.190	3967.5
0.70	25000.0	39520.4	21999.5	10072.6	15.091	3967.5
0.70	30000.0	20626.3	14367.1	3750.3	5.220	3967.5
0.70	30000.0	25185.2	15803.5	5356.3	7.595	3967.5
0.70	30000.0	29582.3	17072.5	7024.2	10.213	3967.5
0.70	30000.0	30651.8	17362.9	7456.6	10.918	3967.5
0.70	30000.0	31707.7	17644.4	7906.5	11.652	3967.5
0.70	30000.0	32765.7	17919.3	8371.5	12.467	3967.5
0.70	30000.0	33823.6	18196.1	8846.9	13.312	3967.5
0.70	35000.0	17153.8	11660.8	3180.6	4.440	3967.5

0.70	35000.0	21121.5	12886.4	4568.6	6.487	3967.5
0.70	35000.0	24934.3	13949.0	6035.6	8.771	3967.5
0.70	35000.0	25864.6	14189.5	6432.5	9.441	3967.5
0.70	35000.0	26787.9	14430.5	6836.6	10.136	3967.5
0.70	35000.0	27710.3	14663.8	7253.0	10.873	3967.5
0.70	35000.0	28626.3	14896.7	7679.1	11.658	3967.5
0.70	39000.0	14279.1	9670.8	2665.6	3.965	3967.5
0.70	39000.0	17596.9	10689.9	3826.6	5.787	3967.5
0.70	39000.0	20784.7	11575.0	5055.5	7.835	3967.5
0.70	39000.0	21565.1	11778.1	5391.3	8.436	3967.5
0.70	39000.0	22341.9	11977.0	5732.5	9.075	3967.5
0.70	39000.0	23113.1	12175.0	6078.2	9.738	3967.5
0.70	39000.0	23882.5	12368.4	6443.6	10.461	3967.5
0.70	43000.0	19596.5	10185.2	5285.8	9.381	3967.5
0.75	25000.0	26363.9	19463.0	4488.3	6.334	3967.5
0.75	25000.0	31596.9	21250.3	6327.9	9.078	3967.5
0.75	25000.0	36633.6	22836.0	8235.6	12.149	3967.5
0.75	25000.0	37867.6	23206.2	8731.1	12.956	3967.5
0.75	25000.0	39095.9	23567.5	9228.3	13.780	3967.5
0.75	25000.0	40311.0	23917.1	9725.1	14.617	3967.5
0.75	25000.0	41519.6	24266.5	10233.7	15.504	3967.5
0.75	30000.0	22128.0	15953.8	3854.3	5.438	3967.5
0.75	30000.0	26751.5	17496.4	5481.4	7.869	3967.5
0.75	30000.0	31190.9	18850.3	7171.6	10.549	3967.5
0.75	30000.0	32276.6	19165.6	7606.5	11.249	3967.5
0.75	30000.0	33355.5	19474.4	8056.9	12.011	3967.5
0.75	30000.0	34423.3	19771.2	8520.3	12.810	3967.5
0.75	30000.0	35493.6	20073.2	8998.2	13.659	3967.5
0.75	35000.0	18401.2	12954.6	3277.9	4.633	3967.5
0.75	35000.0	22436.8	14270.8	4690.6	6.746	3967.5
0.75	35000.0	26302.6	15412.0	6179.4	9.072	3967.5
0.75	35000.0	27252.1	15681.9	6581.6	9.737	3967.5
0.75	35000.0	28202.6	15942.9	6993.2	10.470	3967.5
0.75	35000.0	29135.5	16194.6	7409.2	11.219	3967.5
0.75	35000.0	30066.1	16454.1	7833.3	11.984	3967.5
0.75	39000.0	15325.8	10748.1	2750.5	4.142	3967.5
0.75	39000.0	18707.1	11844.7	3936.1	6.030	3967.5
0.75	39000.0	21947.7	12798.3	5184.8	8.118	3967.5
0.75	39000.0	22739.3	13018.2	5520.3	8.734	3967.5
0.75	39000.0	23524.5	13237.0	5863.9	9.372	3967.5
0.75	39000.0	24309.8	13449.6	6217.3	10.047	3967.5
0.75	39000.0	25094.9	13665.6	6577.2	10.757	3967.5
0.75	43000.0	12592.3	8858.3	2268.7	3.725	3967.5
0.75	43000.0	15354.8	9756.0	3235.5	5.415	3967.5
0.75	43000.0	18003.0	10541.3	4252.5	7.278	3967.5
0.75	43000.0	18652.2	10719.1	4527.2	7.828	3967.5
0.75	43000.0	19291.8	10897.9	4807.1	8.398	3967.5
0.75	43000.0	19932.7	11071.5	5095.3	9.003	3967.5
0.75	43000.0	20572.2	11246.7	5392.8	9.652	3967.5
0.80	25000.0	28346.3	21549.4	4607.6	6.609	3967.5
0.80	25000.0	33655.5	23463.0	6468.1	9.400	3967.5
0.80	25000.0	38731.0	25142.3	8392.9	12.543	3967.5
0.80	25000.0	39964.1	25532.1	8888.5	13.363	3967.5
0.80	25000.0	41192.4	25913.2	9390.9	14.199	3967.5
0.80	25000.0	42418.2	26291.2	9890.2	15.053	3967.5

0.80	25000.0	43644.0	26668.5	10400.5	15.945	3967.5
0.80	30000.0	23758.0	17660.9	3959.4	5.669	3967.5
0.80	30000.0	28454.6	19313.9	5608.3	8.159	3967.5
0.80	30000.0	32944.9	20754.2	7333.0	10.919	3967.5
0.80	30000.0	34057.1	21094.7	7769.5	11.637	3967.5
0.80	30000.0	35150.0	21429.3	8216.3	12.384	3967.5
0.80	30000.0	36232.8	21754.6	8678.0	13.188	3967.5
0.80	30000.0	37313.6	22077.2	9158.5	14.047	3967.5
0.80	35000.0	19740.8	14341.4	3372.8	4.831	3967.5
0.80	35000.0	23838.7	15745.9	4814.7	7.005	3967.5
0.80	35000.0	27763.4	16975.9	6320.9	9.390	3967.5
0.80	35000.0	28736.0	17268.8	6723.7	10.058	3967.5
0.80	35000.0	29698.7	17551.5	7140.6	10.782	3967.5
0.80	35000.0	30650.3	17826.8	7561.4	11.551	3967.5
0.80	35000.0	31601.5	18108.1	7988.4	12.335	3967.5
0.80	39000.0	16433.2	11897.8	2828.3	4.316	3967.5
0.80	39000.0	19869.2	13070.9	4036.2	6.260	3967.5
0.80	39000.0	23170.8	14101.3	5305.1	8.403	3967.5
0.80	39000.0	23988.2	14344.4	5648.4	9.014	3967.5
0.80	39000.0	24798.0	14581.1	6000.7	9.675	3967.5
0.80	39000.0	25596.4	14810.1	6354.5	10.375	3967.5
0.80	39000.0	26392.9	15046.4	6714.5	11.077	3967.5
0.80	43000.0	13523.2	9812.2	2339.0	3.893	3967.5
0.80	43000.0	16338.3	10774.7	3330.2	5.641	3967.5
0.80	43000.0	19040.8	11619.6	4364.8	7.559	3967.5
0.80	43000.0	19703.4	11815.1	4639.9	8.110	3967.5
0.80	43000.0	20352.8	12007.3	4922.3	8.688	3967.5
0.80	43000.0	21001.9	12196.3	5212.4	9.297	3967.5
0.80	43000.0	21655.9	12388.7	5507.9	9.938	3967.5
0.85	30000.0	25554.5	19512.4	4078.2	5.931	3967.5
0.85	30000.0	30337.4	21276.4	5751.0	8.490	3967.5
0.85	30000.0	34884.4	22800.0	7499.2	11.323	3967.5
0.85	30000.0	35993.9	23163.0	7941.0	12.058	3967.5
0.85	30000.0	37102.3	23525.3	8393.1	12.821	3967.5
0.85	30000.0	38206.0	23877.8	8850.2	13.614	3967.5
0.85	30000.0	39303.1	24229.7	9329.1	14.478	3967.5
0.85	35000.0	21197.2	15840.1	3471.9	5.045	3967.5
0.85	35000.0	25362.0	17330.4	4939.1	7.284	3967.5
0.85	35000.0	29346.2	18647.9	6471.1	9.729	3967.5
0.85	35000.0	30337.7	18962.9	6871.3	10.418	3967.5
0.85	35000.0	31307.5	19271.5	7285.7	11.120	3967.5
0.85	35000.0	32278.7	19572.3	7712.5	11.898	3967.5
0.85	35000.0	33247.7	19872.7	8144.5	12.702	3967.5
0.85	39000.0	17646.1	13140.3	2913.1	4.508	3967.5
0.85	39000.0	21142.1	14387.8	4145.9	6.513	3967.5
0.85	39000.0	24498.1	15492.1	5434.5	8.716	3967.5
0.85	39000.0	25332.5	15760.3	5776.9	9.324	3967.5
0.85	39000.0	26157.0	16015.2	6130.2	10.001	3967.5
0.85	39000.0	26966.7	16264.6	6489.1	10.690	3967.5
0.85	39000.0	27781.8	16519.0	6853.0	11.417	3967.5
0.85	43000.0	14521.7	10837.0	2407.9	4.066	3967.5
0.85	43000.0	17385.1	11860.8	3417.5	5.865	3967.5
0.85	43000.0	20129.1	12763.5	4469.0	7.837	3967.5
0.85	43000.0	20805.5	12979.9	4743.5	8.385	3967.5
0.85	43000.0	21472.0	13191.8	5030.8	8.958	3967.5

0.85	43000.0	22140.8	13397.3	5325.6	9.596	3967.5
0.85	43000.0	22807.7	13604.7	5623.6	10.245	3967.5
0.90	35000.0	22762.8	17451.4	3570.5	5.268	3967.5
0.90	35000.0	27000.1	19035.7	5064.2	7.565	3967.5
0.90	35000.0	31055.8	20438.3	6625.5	10.108	3967.5
0.90	35000.0	32062.9	20783.0	7027.6	10.780	3967.5
0.90	35000.0	33062.3	21119.9	7444.4	11.512	3967.5
0.90	35000.0	34048.5	21442.7	7872.1	12.277	3967.5
0.90	35000.0	35034.2	21767.5	8307.3	13.097	3967.5
0.90	39000.0	18948.9	14477.9	2998.3	4.710	3967.5
0.90	39000.0	22507.9	15801.7	4254.9	6.772	3967.5
0.90	39000.0	25922.6	16980.7	5564.9	9.048	3967.5
0.90	39000.0	26769.2	17270.7	5908.5	9.658	3967.5
0.90	39000.0	27614.7	17552.8	6262.2	10.322	3967.5
0.90	39000.0	28446.7	17824.4	6626.8	11.036	3967.5
0.90	39000.0	29278.0	18096.7	6993.6	11.777	3967.5
0.90	43000.0	15596.9	11941.5	2475.3	4.246	3967.5
0.90	43000.0	18504.1	13023.6	3499.4	6.095	3967.5
0.90	43000.0	21291.5	13989.2	4573.7	8.128	3967.5
0.90	43000.0	21989.2	14229.3	4854.4	8.677	3967.5
0.90	43000.0	22685.7	14461.2	5143.2	9.268	3967.5
0.90	43000.0	23366.8	14683.6	5441.1	9.908	3967.5
0.90	43000.0	24046.1	14906.1	5741.5	10.573	3967.5

APPENDIX B

SURROGATE MODEL FIT RESULTS

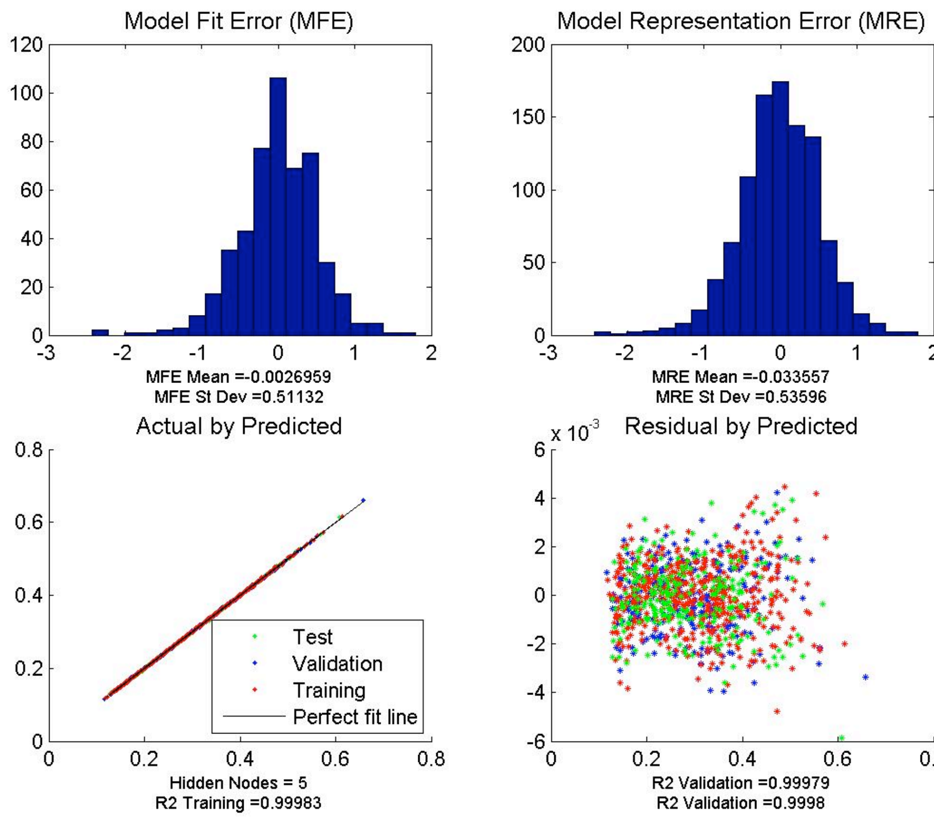


Figure 60: Average Yield for the A330 Surrogate Model Quality

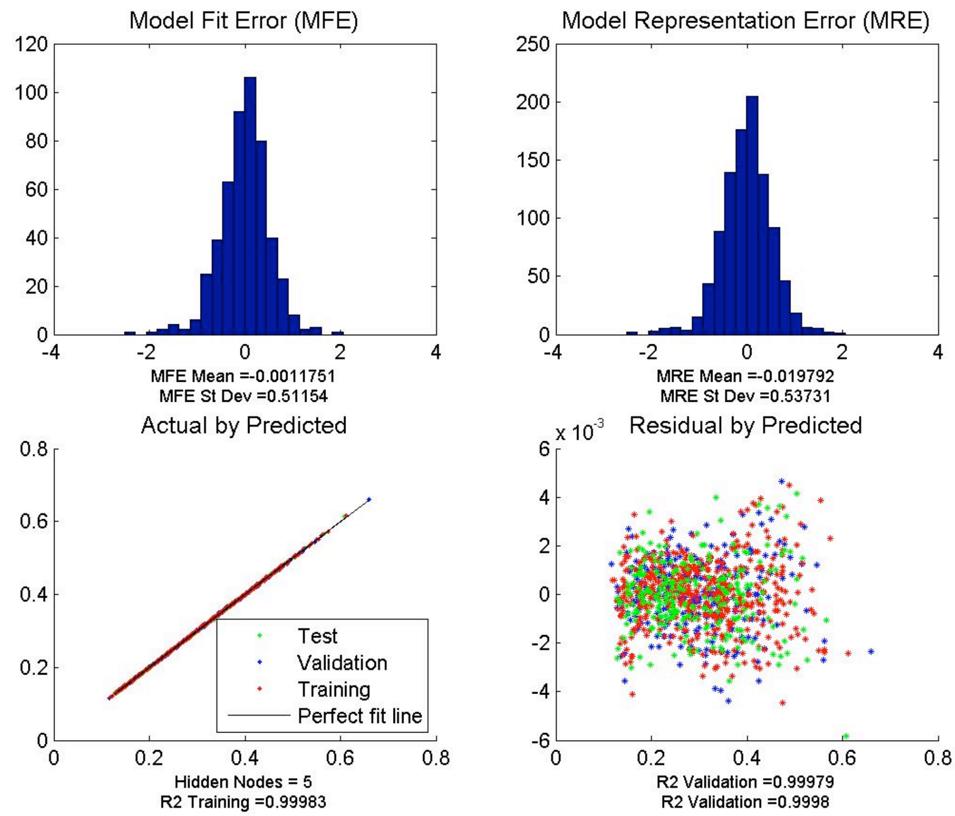


Figure 61: Average Yield for the 767 Surrogate Model Quality

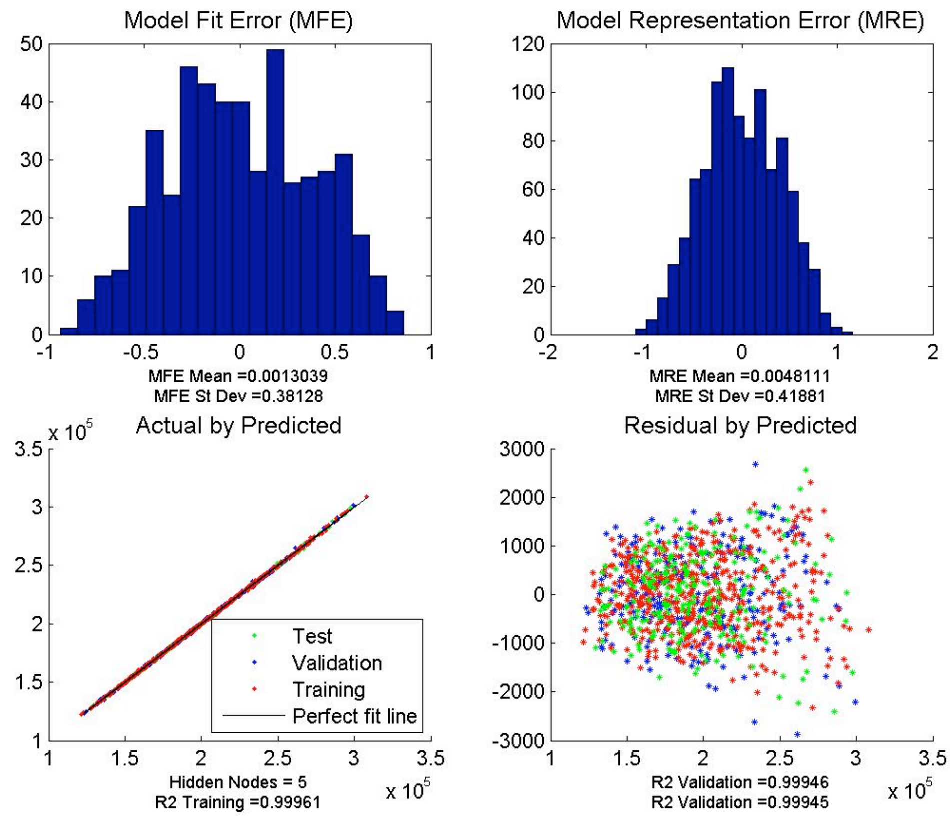


Figure 62: Fuel Weight for the A330 Surrogate Model Quality

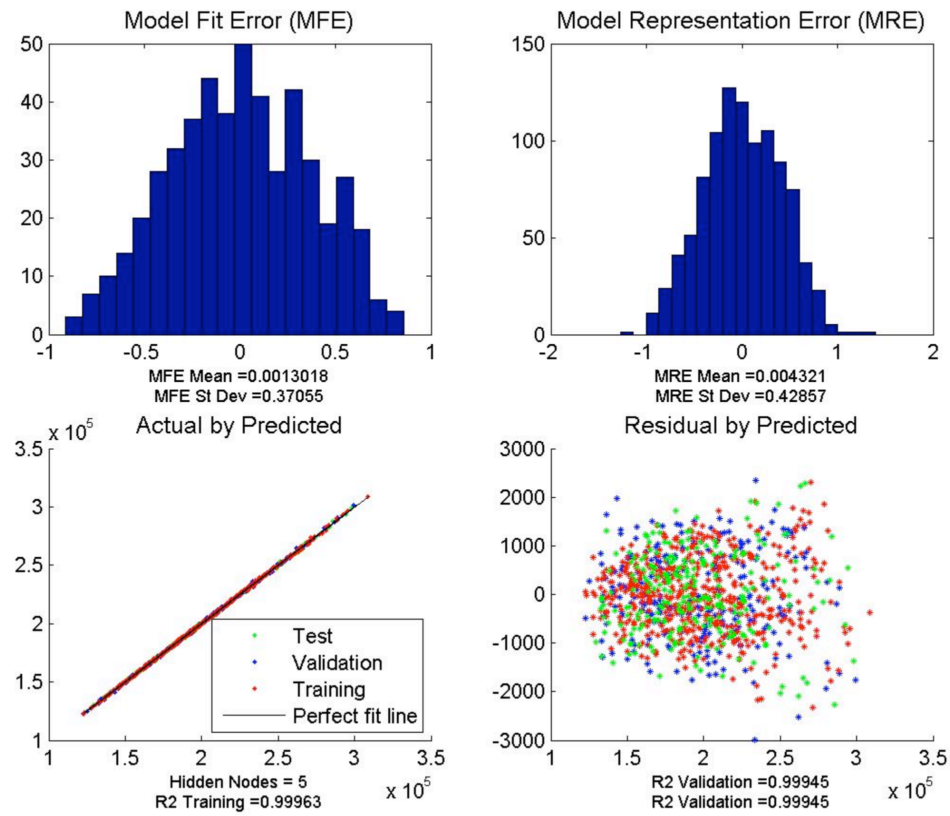


Figure 63: Fuel Weight for the 767 Surrogate Model Quality

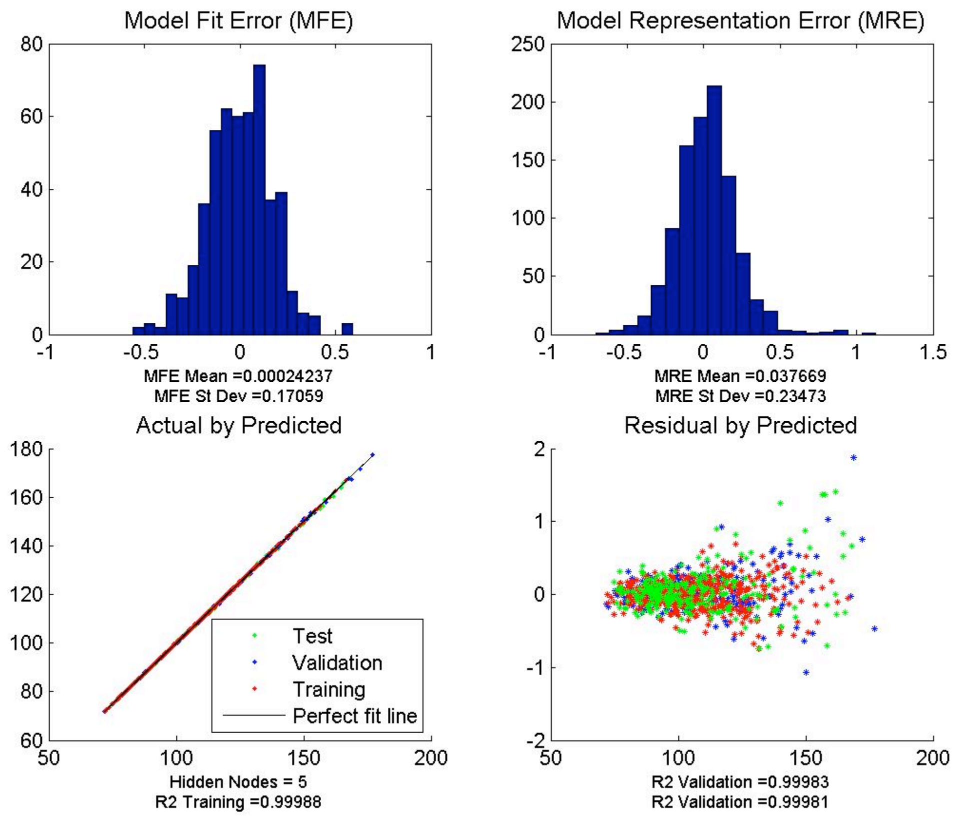


Figure 64: Price for the A330 Surrogate Model Quality

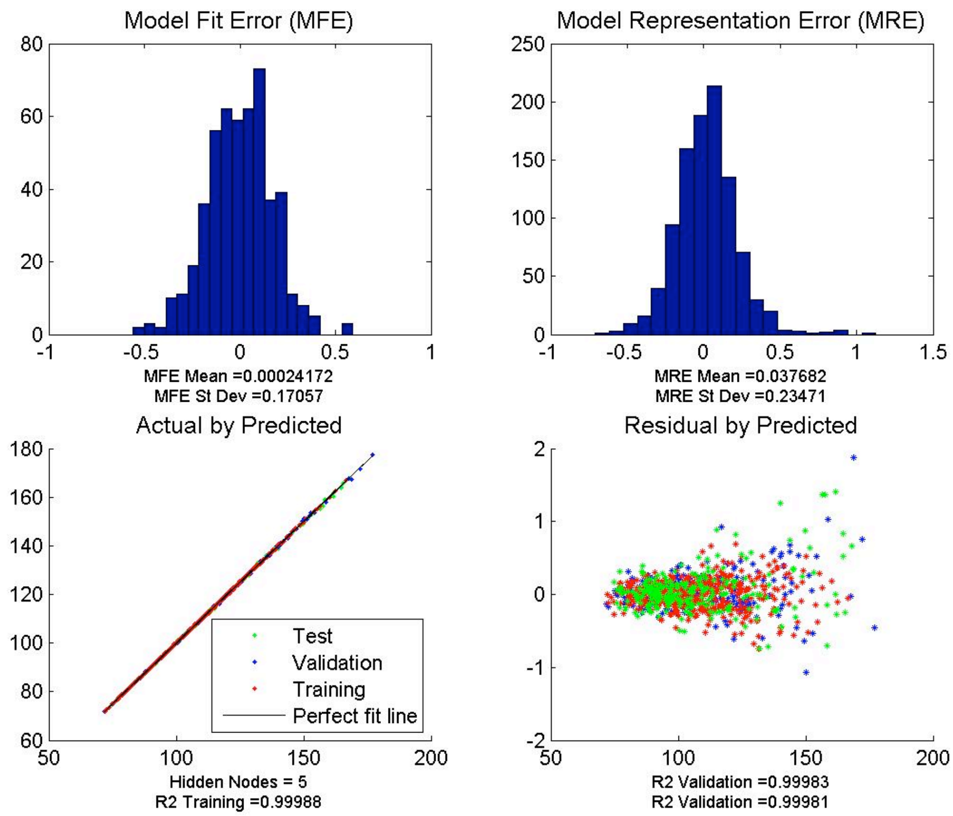


Figure 65: Price for the 767 Surrogate Model Quality

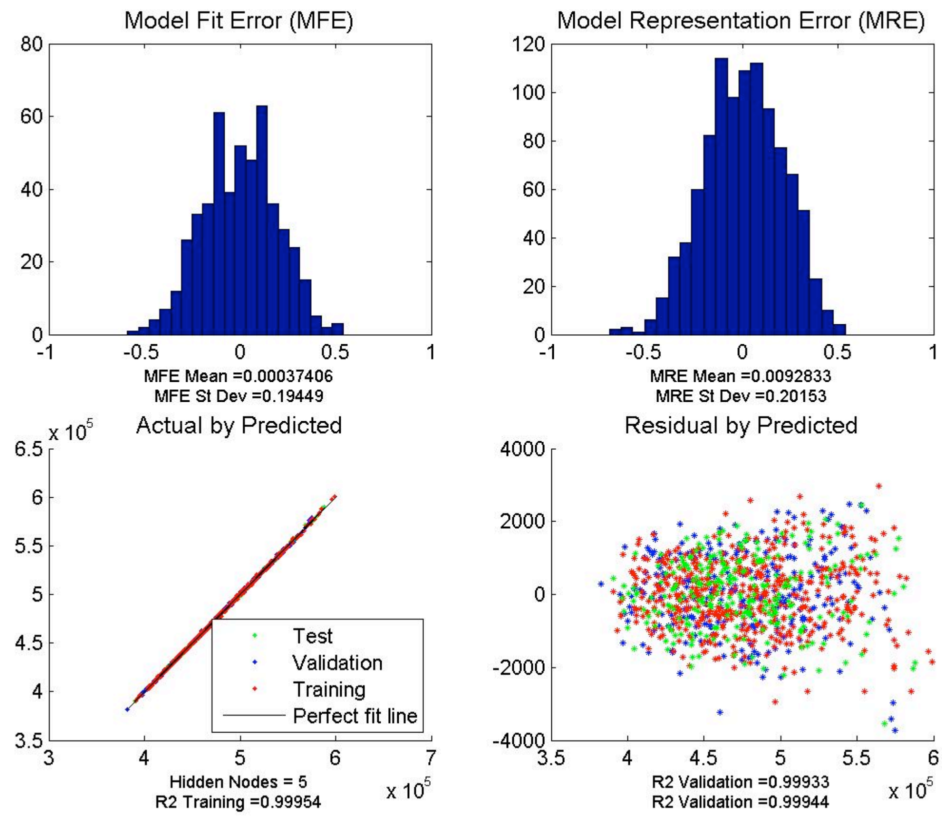


Figure 66: Takeoff Gross Weight for the A330 Surrogate Model Quality

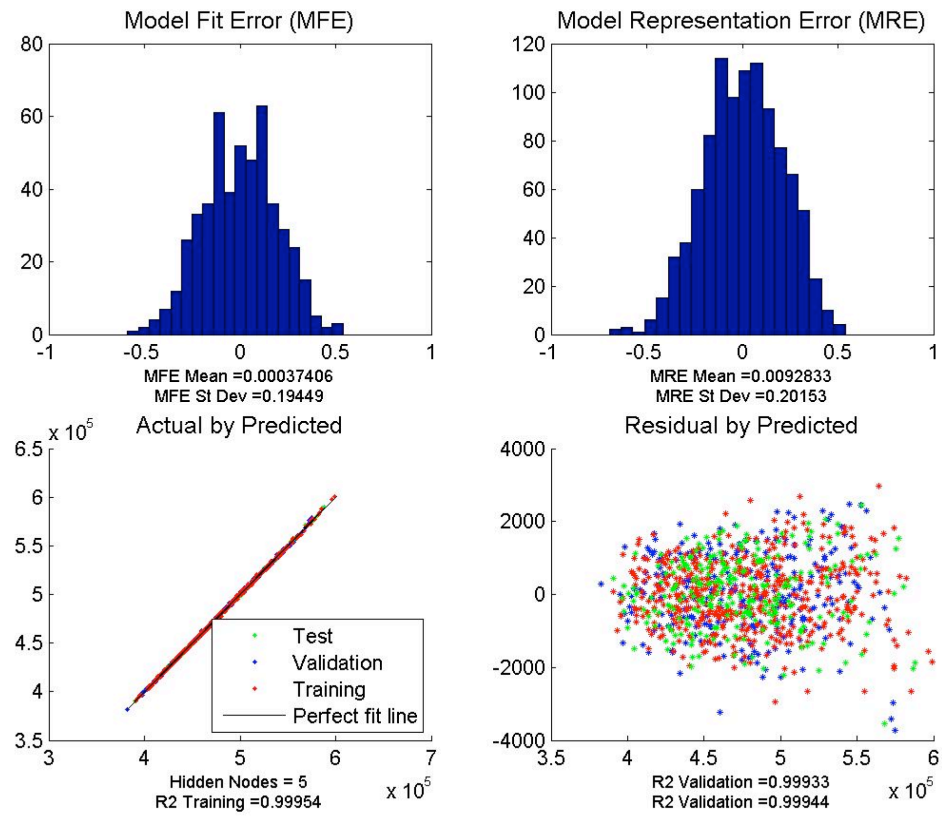


Figure 67: Takeoff Gross Weight for the 767 Surrogate Model Quality

APPENDIX C

SOURCE CODE

C.1 Parser

This code is for an executable command-line program that executes FLOPS/ALCCA while using a baseline case definition file and a second file with a set of cases and variables to be switched for each. It then produces a output file with outputs parsed according to the parse information file that contains results for all cases.

Listing C.1: Parser.cs

```
using System;
using System.Collections.Generic;
using System.Text;
using System.IO;
using System.Diagnostics;
using System.Collections;

namespace Parser
{
    class Parser
    {
        struct ParseInfo
        {
            public int read, occurance;
        }

        string baselinefile;
        Hashtable Inputs,Parseinfo;
        int noCases;

        public Parser(string f)
        {
            baselinefile = f;
        }

        string parseFile(string file, string match, int read, int occurrence)
        {
            string tmp,tmp2;
            string[] tmp3;
            int count=0;
            int i=0,j=0;
```

```
    using (StreamReader fs = File.OpenText(file))
    {
        while ((tmp = fs.ReadLine()) != null)
        {
            tmp2 = tmp.Trim();
            if (tmp2.Contains("NO_WEIGHT"))
            {
                return "FAILED";
            }
            else if (tmp2.Contains("FAILURE_FOR"))
            {
                return "FAILED";
            }
            else if (tmp2.Contains(match))
            {
                count++;
                if (count == occurrence)
                {
                    tmp3 = tmp2.Split(' ');
                    while ((i < tmp3.Length)&&(j<read))
                    {
                        if (!tmp3[i].Equals(""))
                        {
                            j++;
                            if (j == read)
                            {
                                return tmp3[i];
                            }
                        }
                        i++;
                    }
                }
            }
        }
    }
    return "FAILED";
}

public void runFLOPS()
{
    ProcessStartInfo psi = new ProcessStartInfo(@"flops.exe"); // @ ignores slashes as
        special character prefix, i.e. no need for double slashes
    psi.Arguments = "tmp.in tmp.out";
    psi.RedirectStandardOutput = true;
    psi.WindowStyle = ProcessWindowStyle.Hidden; // hide window completely
    psi.UseShellExecute = false; // don't use os shell to start
    Process flops;
    flops = Process.Start(psi);
    StreamReader output = flops.StandardOutput;
    flops.WaitForExit(20000);
    if (flops.HasExited)
    {
        string outstring = output.ReadToEnd();
    }
}
```

```
public void readInputsHoriz(string file)
{
    string tmp, tmp2, name;
    string[] tmp3;
    Hashtable variables = new Hashtable();
    ArrayList numbers;
    int i;

    using (StreamReader fs = File.OpenText(file))
    {
        while ((tmp = fs.ReadLine()) != null)
        {
            numbers = new ArrayList();
            tmp2 = tmp.Trim();
            tmp3 = tmp2.Split(',');
            name = tmp3[0];
            noCases = tmp3.Length - 1;
            for (i = 1; i < tmp3.Length; i++)
            {
                numbers.Add(Double.Parse(tmp3[i]));
            }
            variables.Add(name, numbers);
        }
    }
    Inputs = variables;
}

public void readInputs(string file)
{
    string tmp, tmp2;
    string[] tmp3,names;
    Hashtable variables = new Hashtable();
    ArrayList numbers;
    int i;

    using (StreamReader fs = File.OpenText(file))
    {
        if ((tmp = fs.ReadLine()) != null)
        {
            tmp2 = tmp.Trim();
            tmp3 = tmp2.Split(',');
            names = new string[tmp3.Length];
            for (i = 0; i < tmp3.Length; i++)
            {
                numbers = new ArrayList();
                names[i] = tmp3[i];
                variables.Add(names[i], numbers);
            }

            noCases = 0;
            while ((tmp = fs.ReadLine()) != null)
            {
                tmp2 = tmp.Trim();
                tmp3 = tmp2.Split(',');
```

```
                for (i = 0; i < tmp3.Length; i++)
                {
                    numbers = (ArrayList) variables[names[i]];
                    numbers.Add(Double.Parse(tmp3[i]));
                }
                noCases++;
            }
        }
    }
    Inputs = variables;
}

public void readParseinfo(string file)
{
    string tmp, tmp2, name;
    string[] tmp3;
    ParseInfo pi;
    Hashtable info = new Hashtable();

    using (StreamReader fs = File.OpenText(file))
    {
        while ((tmp = fs.ReadLine()) != null)
        {
            tmp2 = tmp.Trim();
            tmp3 = tmp2.Split(',');
            name = tmp3[0];
            pi.read = Int32.Parse(tmp3[1]);
            pi.occurance = Int32.Parse(tmp3[2]);
            info.Add(name, pi);
        }
    }
    Parseinfo = info;
}

public void switchVariables(string filein, string fileout, Hashtable vars)
{
    string tmp, tmp2, replacement;
    string[] tmp3;
    int i,j;

    using (StreamReader fsin = File.OpenText(filein))
    using (StreamWriter fsout = File.CreateText(fileout))
    {
        while ((tmp = fsin.ReadLine()) != null)
        {
            foreach (string name in vars.Keys)
            {
                if (tmp.Contains(name))
                {
                    i = tmp.IndexOf(name);
                    char c = tmp[i + name.Length];
                    char c2 = tmp[i-1];
                    if ((c == '=') && (c2 == ' '))
                    {
                        i += name.Length;
```

```
                        tmp2 = tmp.Substring(i);
                        replacement = tmp.Substring(0, i);
                        tmp3 = tmp2.Split('=');
                        replacement += "= ";
                        j = 0;
                        while ((j < tmp3.Length) && ((tmp3[j].Equals("")) || tmp3[j].
                            Equals("=")))
                        {
                            replacement += tmp3[j];
                            j++;
                        }
                        replacement += vars[name].ToString();
                        for (j++; j < tmp3.Length; j++)
                        {
                            replacement += tmp3[j];
                        }
                        replacement += ",";
                        tmp = replacement;
                    }
                }
            }
            fsout.WriteLine(tmp);
        }
    }

}

public void runCases()
{
    string result = "";
    int i;

    foreach (string name in Parseinfo.Keys)
    {
        result += name;
        result += ", ";
    }
    result += "\n";

    for (i = 0; i < noCases; i++)
    {
        Console.WriteLine("Running Case..." + i);
        Hashtable test = new Hashtable();
        foreach (string name in Inputs.Keys)
        {
            test.Add(name, ((ArrayList) Inputs[name])[i]);
        }
        switchVariables(baselinefile, "tmp.in", test);
        runFLOPS();
        result += "\n";
        foreach (string name in Parseinfo.Keys)
        {
            result += parseFile("tmp.out", name, ((ParseInfo)Parseinfo[name]).read, ((
                ParseInfo)Parseinfo[name]).occurance);
            result += ", ";
```

```
            }
        }
        using (StreamWriter fsout = File.CreateText("summary.txt"))
        {
            fsout.Write(result);
        }
    }

    static void Main(string[] args)
    {
        Parser parse = new Parser(args[0]);
        parse.readInputs(args[1]);
        parse.readParseinfo("parseinfo.txt");
        parse.runCases();
    }
  }
}
```

C.2 Computational Model Source Code

C.2.1 Differential Equation Solver Source Code

This is the interface definition for an ordinary differential equation which simply defines that any such equation has to return the vector of derivatives as a function of all the variables.

Listing C.2: odeequation.java

```
package edu.gatech.asdl.pfaender.ode;

public interface ODEequation {
        double[] dfxy(double x, double y[]);
}
```

The solver interface defines the requirement that any solver implement a function that solves and ordinary differential equation with the time step, absolute time, and the vector of variables and derivatives.

Listing C.3: solver.java

```
package edu.gatech.asdl.pfaender.ode;

public interface solver {
        double[] solve(double h, double t, int n, double y[], double dydx[], ODEequation eq);
}
```

This following code is the actual implementation of a 4th order Runge-Kutta fixed time step solver. This implementation follows several well known implementations and is quite efficient and fast in obtaining ordinary differential equation solutions.

Listing C.4: rk4.java

```
package edu.gatech.asdl.pfaender.ode;

public class rk4 implements solver{

        public double[] solve(double h, double t, int n, double y[], double dydx[], ODEequation eq
            ) {
                double yout[];
                int i;
                double th,hh,h6;
                double dym[],dyt[],yt[];
                yout = new double[n];
                dym = new double[n];
                dyt = new double[n];
                yt = new double[n];

                hh=h*0.5;
                h6=h/6.0;
                th=t+hh;
                for (i=0;i<n;i++) yt[i]=y[i]+hh*dydx[i];
                dyt=eq.dfxy(th,yt);
                for (i=0;i<n;i++) yt[i]=y[i]+hh*dyt[i];
                dym=eq.dfxy(th,yt);
                for (i=0;i<n;i++) {
                        yt[i]=y[i]+h*dym[i];
                        dym[i]+=dyt[i];
                }
                dyt=eq.dfxy(t+h,yt);
                for (i=0;i<n;i++)
                        yout[i]=y[i]+h6*(dydx[i]+dyt[i]+2.0*dym[i]);

                return yout;
        }
}
```

C.2.2 Competition Model Source Code

The following code is the actual core implementation of the competitive market system dynamics model. It contains the neural network surrogate model representations of both aircraft as well as the definition of all model variables both as initial conditions and continuous equations as well as stochastic and deterministic implementations.

This quadruple implementation was required to satisfy the varying requirements of the EUTE. Furthermore, this also makes use of the interfaces and the solver described earlier.

Listing C.5: competitionmodelcalculator3.java

```
package edu.gatech.asdl.pfaender.competitionmodel;

import java.util.Random;

import edu.gatech.asdl.pfaender.ode.ODEequation;
import edu.gatech.asdl.pfaender.ode.rk4;
import edu.gatech.asdl.pfaender.ode.solver;

// Class that inherits the RK4 solver and implements the ODE equations
// includes complete Monte Carlo solver

public class competitionmodelCalculator3 extends rk4 implements ODEequation {
        solver CompetitionSolver;
        double t0;
        double tf;
        double dt;
        int nstep;
        int nvar=79;
        int nvarstock=3;
        double[] y0;
        double[] y0stock;
        double[] y;
        double[] ystock;
        double[] dy;
        double[][] v;
        double[][] vstock;
        double[] vstockrangel = {0.0,0.0,0.0};
        double[] vstockrangeh = {15.0,15.0,1.0};
        // Range Definitions
        double[] vrangel = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                        30,0.2,8,0,230,500,0,0.8,2,5000,
                        30,0.2,8,0,230,500,0,0.8,2,5000,
                        0,0,0,0,0.1,90,230,5000,0,0,0,0,0,0,0,0,0,0,
                        0,0,0,0,0,30,0.2,8,0,230,500,0,0.8,2,5000,0.1,90,230,5000,0.0};
        double[] vrangeh = {0,10,10,0.1,0.1,20,20,0,0,0,2,20,30,0,0,2,90,0,0,5,0,
                        35,0.3,12,15,270,1500,1,1.2,3,7000,
                        35,0.3,12,15,270,1500,1,1.2,3,7000,
                        0,0,0,0,0.2,100,270,7000,90,90,90,0,0,0,0,0,0,0,
                        5,0.1,0.1,20,20,35,0.3,12,15,270,1500,1,1.2,3,7000,0.2,100,270,7000,7};

        Random rnd = new Random();
        double[][] ib1bins,ib2bins,ms1bins,ms2bins;
        double[] ib1mean,ib1con90u,ib1con90l,ib1con50u,ib1con50l;
        double[] ib2mean,ib2con90u,ib2con90l,ib2con50u,ib2con50l;
        double[] ms1mean,ms1con90u,ms1con90l,ms1con50u,ms1con50l;
        double[] ms2mean,ms2con90u,ms2con90l,ms2con50u,ms2con50l;
```

```
// Constructor that initializes all the variables

        public competitionmodelCalculator3() {
                CompetitionSolver = new rk4();

                t0 = 0;
                tf = 92;
                dt = 1;
                nstep = (int)Math.round((tf-t0)/dt);

                y0 = setupDefaultAlgebraicEquations();
                y0stock = setupDefaultStocks();
                y = new double[nvar];
                ystock = new double[nvarstock];
                dy = new double[nvarstock];

                int i;

                v = new double[nvar][nstep+1];
                for (i=0;i<nvar;i++) {
                        v[i] = new double[nstep+1];
                }
                vstock = new double[nvarstock][nstep+1];
                for (i=0;i<nvarstock;i++) {
                        vstock[i] = new double[nstep+1];
                }

        }

// Setup stock initial values

        double[] setupDefaultStocks() {
                double[] ystock = new double[nvarstock];

                ystock[0]=y0[1]; //Installed Base Aircraft 1
                ystock[1]=y0[2]; //Installed Base Aircraft 2
                ystock[2]=y0[9]; //Pink Noise
                return ystock;
        }

// Setup the default algebra equations initial conditions

        double[] setupDefaultAlgebraicEquations() {
                double[] y = new double[nvar];

                y[59]=0; //range of demand
                y[60]=0; //range of sensitivity 767
                y[61]=0; //range of sensitivity 330
                y[62]=0; //range of threshold 767
                y[63]=0; //range of threshold 330
                y[64]=0; //range of Sweep
                y[65]=0; //range of TR
                y[66]=0; //range of AR
```

```
y[67]=0; //range of RTRTN
y[68]=0; //range of PAX
y[69]=0; //range of NVEH
y[70]=0; //range of FUCOMP
y[71]=0; //range of FACT
y[72]=0; //range of CLTOM
y[73]=0; //range of DESRNG
y[74]=0; //range of Yield Attr
y[75]=0; //range of Price Attr
y[76]=0; //range of Pax Attr
y[77]=0; //range of Rng Attr
y[78]=0; //range of initial fuel price;

y[0]=rndUniform(y[59],vrangel[59],vrangeh[59],Demand(t0));
if (y[0]<0)
        y[0]=0;

y[1]=0; //Initial Installed Base 767
y[2]=0; //Initial Installed Base 330


y[3]=rndUniform(y[60],vrangel[60],vrangeh[60],0.01);
//Sensitivity of Attractiveness to Installed Base 767
y[4]=rndUniform(y[61],vrangel[61],vrangeh[61],0.04);
//Sensitivity of Attractiveness to Installed Base 330

y[5]=rndUniform(y[62],vrangel[62],vrangeh[62],10);
//Threshold for Compatibility Effects 767
y[6]=rndUniform(y[63],vrangel[63],vrangeh[63],10);
//Threshold for Compatibility Effects 330

y[7]=Math.exp(y[3]*y[1]/y[5]); //Effect of Commonality on Attractiveness of 767
y[8]=Math.exp(y[4]*y[2]/y[6]); //Effect of Commonality on Attractiveness of 330

y[9]=0; //Initial Pink Noise
y[10]=0.4; //Noise std
y[11]=0; // Noise seed
y[12]=12; //Noise correlation time
y[13]=y[10]*(Math.sqrt(24*12/dt)*(rnd.nextDouble()-0.5)); //White Noise
y[14]=(y[13] - y[9])/y[12]; //Change in Pink Noise
y[15]=0.25 ; //Ramp Slope $ per decade
y[16]=0; //Ramp Start Time
y[17]=1e9; //Ramp End Time
y[18]=1+ramp(y[15]/120,y[16],y[17],t0)+y[9]; //Pink Noise Input

y[19]=rndUniform(y[78],vrangel[78],vrangeh[78],0.44); //Initial Fuel Price
y[20]=y[19]*y[18]; //COFL

y[21]=rndUniform(y[64],vrangel[64],vrangeh[64],31.5); //Sweep 767
y[22]=rndUniform(y[65],vrangel[65],vrangeh[65],0.23); //TR 767
y[23]=rndUniform(y[66],vrangel[66],vrangeh[66],9.3); //AR 767
y[24]=rndUniform(y[67],vrangel[67],vrangeh[67],10); //RTRTN 767
y[25]=rndUniform(y[68],vrangel[68],vrangeh[68],243); //PAX 767
y[26]=rndUniform(y[69],vrangel[69],vrangeh[69],1049); //NVEH 767
y[27]=rndUniform(y[70],vrangel[70],vrangeh[70],0); //FUCOMP 767
```

```
            y[28]=rndUniform(y[71], vrangel[71], vrangeh[71],1); //FACT 767
            y[29]=rndUniform(y[72], vrangel[72], vrangeh[72],2.3); //CLTOM 767
            y[30]=rndUniform(y[73], vrangel[73], vrangeh[73],5645); //DESRNG

            y[31]=rndUniform(y[64], vrangel[64], vrangeh[64],31.5); //Sweep 330
            y[32]=rndUniform(y[65], vrangel[65], vrangeh[65],0.267); //TR 330
            y[33]=rndUniform(y[66], vrangel[66], vrangeh[66],10.1); //AR 330
            y[34]=rndUniform(y[67], vrangel[67], vrangeh[67],5); //RTRTN 330
            y[35]=rndUniform(y[68], vrangel[68], vrangeh[68],253); //PAX 330
            y[36]=rndUniform(y[69], vrangel[69], vrangeh[69],830); //NVEH 330
            y[37]=rndUniform(y[70], vrangel[70], vrangeh[70],0); //FUCOMP 330
            y[38]=rndUniform(y[71], vrangel[71], vrangeh[71],1); //FACT 330
            y[39]=rndUniform(y[72], vrangel[72], vrangeh[72],2.3); //CLTOM 330
            y[40]=rndUniform(y[73], vrangel[73], vrangeh[73],6650); //DESRNG 330

            y[41]=Price767(y[21],y[22],y[23],y[24],y[25],y[26],y[27],y[28],y[29],y[30],y[20]);
            //767 Price
            y[42]=Yield767(y[21],y[22],y[23],y[24],y[25],y[26],y[27],y[28],y[29],y[30],y[20]);
            //767 Yield

            y[43]=Price330(y[31],y[32],y[33],y[34],y[35],y[36],y[37],y[38],y[39],y[40],y[20]);
            //330 Price
            y[44]=Yield330(y[31],y[32],y[33],y[34],y[35],y[36],y[37],y[38],y[39],y[40],y[20]);
            //330 Yield

            y[45]=rndUniform(y[74], vrangel[74], vrangeh[74],0.15); //Yield Attr
            y[46]=rndUniform(y[75], vrangel[75], vrangeh[75],100); //Price Attr
            y[47]=rndUniform(y[76], vrangel[76], vrangeh[76],250); //Pax Attr
            y[48]=rndUniform(y[77], vrangel[77], vrangeh[77],6000); //Rng Attr
            y[49]=28; //Time of Intro 767
            y[50]=0; //Time of Intro 330
            y[51]=49; //Time of Exit 767

            y[52]=pulse(y[49],y[51]-y[49],t0)*(y[7]+1-y[42]/y[45]+1-y[41]/y[46]+y[30]/y[48]+
                    y[25]/y[47]); //Attractiveness of 767
            y[53]=pulse(y[50],200,t0)*(y[8]+1-y[44]/y[45]+1-y[43]/y[46]+y[40]/y[48]+
                    y[35]/y[47]); //Attractiveness of 330

            y[54]=y[52]+y[53]; //Total Attractiveness

            y[55]=y[52]/y[54]; //Market Share of 767
            y[56]=y[53]/y[54]; //Market Share of 330

            y[57]=y[0]*y[55]; //Sales of 767
            y[58]=y[0]*y[56]; //Sales of 330

            return y;
        }

// Prepare algebra equations for Monte Carlo

        double[] setupDefaultAlgebraicEquationsMC() {
            double[] y = new double[nvar];

            y[59]=1; //range of demand
```

```
y[60]=1; //range of sensitivity 767
y[61]=1; //range of sensitivity 330
y[62]=1; //range of threshold 767
y[63]=1; //range of threshold 330
y[64]=1; //range of Sweep
y[65]=1; //range of TR
y[66]=1; //range of AR
y[67]=1; //range of RTRTN
y[68]=1; //range of PAX
y[69]=1; //range of NVEH
y[70]=1; //range of FUCOMP
y[71]=1; //range of FACT
y[72]=1; //range of CLTOM
y[73]=1; //range of DESRNG
y[74]=1; //range of Yield Attr
y[75]=1; //range of Price Attr
y[76]=1; //range of Pax Attr
y[77]=1; //range of Rng Attr
y[78]=1; //range of initial fuel price;

y[0]=rndUniform(y[59],vrangel[59],vrangeh[59],Demand(t0));
if (y[0]<0)
        y[0]=0;

y[1]=0; //Initial Installed Base 767
y[2]=0; //Initial Installed Base 330

y[3]=rndUniform(y[60],vrangel[60],vrangeh[60],0.01);
//Sensitivity of Attractiveness to Installed Base 767
y[4]=rndUniform(y[61],vrangel[61],vrangeh[61],0.04);
//Sensitivity of Attractiveness to Installed Base 330

y[5]=rndUniform(y[62],vrangel[62],vrangeh[62],10);
//Threshold for Compatibility Effects 767
y[6]=rndUniform(y[63],vrangel[63],vrangeh[63],10);
//Threshold for Compatibility Effects 330

y[7]=Math.exp(y[3]*y[1]/y[5]);
//Effect of Commonality on Attractiveness of 767
y[8]=Math.exp(y[4]*y[2]/y[6]);
//Effect of Commonality on Attractiveness of 330

y[9]=0; //Initial Pink Noise
y[10]=0.4; //Noise std
y[11]=0; // Noise seed
y[12]=12; //Noise correlation time
y[13]=y[10]*(Math.sqrt(24*12/dt)*(rnd.nextDouble()-0.5)); //White Noise
y[14]=(y[13] - y[9])/y[12]; //Change in Pink Noise
y[15]=0.25 ; //Ramp Slope $ per decade
y[16]=0; //Ramp Start Time
y[17]=1e9; //Ramp End Time
y[18]=1+ramp(y[15]/120,y[16],y[17],t0)+y[9]; //Pink Noise Input

y[19]=rndUniform(y[78],vrangel[78],vrangeh[78],0.44); //Initial Fuel Price
y[20]=y[19]*y[18]; //COFL
```

```
y[21]=rndUniform(y[64],vrangel[64],vrangeh[64],31.5); //Sweep 767
y[22]=rndUniform(y[65],vrangel[65],vrangeh[65],0.23); //TR 767
y[23]=rndUniform(y[66],vrangel[66],vrangeh[66],9.3); //AR 767
y[24]=rndUniform(y[67],vrangel[67],vrangeh[67],10); //RTRTN 767
y[25]=rndUniform(y[68],vrangel[68],vrangeh[68],243); //PAX 767
y[26]=rndUniform(y[69],vrangel[69],vrangeh[69],1049); //NVEH 767
y[27]=rndUniform(y[70],vrangel[70],vrangeh[70],0); //FUCOMP 767
y[28]=rndUniform(y[71],vrangel[71],vrangeh[71],1); //FACT 767
y[29]=rndUniform(y[72],vrangel[72],vrangeh[72],2.3); //CLTOM 767
y[30]=rndUniform(y[73],vrangel[73],vrangeh[73],5645); //DESRNG

y[31]=rndUniform(y[64],vrangel[64],vrangeh[64],31.5); //Sweep 330
y[32]=rndUniform(y[65],vrangel[65],vrangeh[65],0.267); //TR 330
y[33]=rndUniform(y[66],vrangel[66],vrangeh[66],10.1); //AR 330
y[34]=rndUniform(y[67],vrangel[67],vrangeh[67],5); //RTRTN 330
y[35]=rndUniform(y[68],vrangel[68],vrangeh[68],253); //PAX 330
y[36]=rndUniform(y[69],vrangel[69],vrangeh[69],830); //NVEH 330
y[37]=rndUniform(y[70],vrangel[70],vrangeh[70],0); //FUCOMP 330
y[38]=rndUniform(y[71],vrangel[71],vrangeh[71],1); //FACT 330
y[39]=rndUniform(y[72],vrangel[72],vrangeh[72],2.3); //CLTOM 330
y[40]=rndUniform(y[73],vrangel[73],vrangeh[73],6650); //DESRNG 330

y[41]=Price767(y[21],y[22],y[23],y[24],y[25],y[26],y[27],y[28],y[29],y[30],y[20]);
//767 Price
y[42]=Yield767(y[21],y[22],y[23],y[24],y[25],y[26],y[27],y[28],y[29],y[30],y[20]);
//767 Yield

y[43]=Price330(y[31],y[32],y[33],y[34],y[35],y[36],y[37],y[38],y[39],y[40],y[20]);
//330 Price
y[44]=Yield330(y[31],y[32],y[33],y[34],y[35],y[36],y[37],y[38],y[39],y[40],y[20]);
//330 Yield

y[45]=rndUniform(y[74],vrangel[74],vrangeh[74],0.15); //Yield Attr
y[46]=rndUniform(y[75],vrangel[75],vrangeh[75],100); //Price Attr
y[47]=rndUniform(y[76],vrangel[76],vrangeh[76],250); //Pax Attr
y[48]=rndUniform(y[77],vrangel[77],vrangeh[77],6000); //Rng Attr
y[49]=28; //Time of Intro 767
y[50]=0; //Time of Intro 330
y[51]=49; //Time of Exit 767

y[52]=pulse(y[49],y[51]-y[49],t0)*(y[7]+1-y[42]/y[45]+1-y[41]/y[46]+y[30]/y[48]+
        y[25]/y[47]); //Attractiveness of 767
y[53]=pulse(y[50],200,t0)*(y[8]+1-y[44]/y[45]+1-y[43]/y[46]+y[40]/y[48]+
        y[35]/y[47]); //Attractiveness of 330

y[54]=y[52]+y[53]; //Total Attractiveness

y[55]=y[52]/y[54]; //Market Share of 767
y[56]=y[53]/y[54]; //Market Share of 330

y[57]=y[0]*y[55]; //Sales of 767
y[58]=y[0]*y[56]; //Sales of 330

return y;
```

```
	}

// Prepare algebra equations for prediction profiler

	double[] setupDefaultAlgebraicEquationsdeterministic() {
		double[] y = new double[nvar];

		y[59]=0; //range of demand
		y[60]=0; //range of sensitivity 767
		y[61]=0; //range of sensitivity 330
		y[62]=0; //range of threshold 767
		y[63]=0; //range of threshold 330
		y[64]=0; //range of Sweep
		y[65]=0; //range of TR
		y[66]=0; //range of AR
		y[67]=0; //range of RTRTN
		y[68]=0; //range of PAX
		y[69]=0; //range of NVEH
		y[70]=0; //range of FUCOMP
		y[71]=0; //range of FACT
		y[72]=0; //range of CLTOM
		y[73]=0; //range of DESRNG
		y[74]=0; //range of Yield Attr
		y[75]=0; //range of Price Attr
		y[76]=0; //range of Pax Attr
		y[77]=0; //range of Rng Attr

		y[0]=Demand(t0);
		if (y[0]<0)
			y[0]=0;

		y[1]=0; //Initial Installed Base 767
		y[2]=0; //Initial Installed Base 330

		y[3]=0.01; //Sensitivity of Attractiveness to Installed Base 767
		y[4]=0.04; //Sensitivity of Attractiveness to Installed Base 330

		y[5]=10; //Threshold for Compatibility Effects 767
		y[6]=10; //Threshold for Compatibility Effects 330

		y[7]=Math.exp(y[3]*y[1]/y[5]); //Effect of Commonality on Attractiveness of 767
		y[8]=Math.exp(y[4]*y[2]/y[6]); //Effect of Commonality on Attractiveness of 330

		y[9]=0; //Initial Pink Noise
		y[10]=0.4; //Noise std
		y[11]=0; // Noise seed
		y[12]=12; //Noise correlation time
		y[13]=y[10]*(Math.sqrt(24*12/dt)*(rnd.nextDouble()-0.5)); //White Noise
		y[14]=(y[13] - y[9])/y[12]; //Change in Pink Noise
		y[15]=0.25 ; //Ramp Slope $ per decade
		y[16]=0; //Ramp Start Time
		y[17]=1e9; //Ramp End Time
		y[18]=1+ramp(y[15]/120,y[16],y[17],t0)+y[9]; //Pink Noise Input

		y[19]=0.44; //Initial Fuel Price
```

```
y[20]=y[19]*y[18]; //COFL

y[21]=31.5; //Sweep 767
y[22]=0.23; //TR 767
y[23]=9.3; //AR 767
y[24]=10; //RTRTN 767
y[25]=243; //PAX 767
y[26]=1049; //NVEH 767
y[27]=0; //FUCOMP 767
y[28]=1; //FACT 767
y[29]=2.3; //CLTOM 767
y[30]=5645; //DESRNG

y[31]=31.5; //Sweep 330
y[32]=0.267; //TR 330
y[33]=10.1; //AR 330
y[34]=5; //RTRTN 330
y[35]=253; //PAX 330
y[36]=830; //NVEH 330
y[37]=0; //FUCOMP 330
y[38]=1; //FACT 330
y[39]=2.3; //CLTOM 330
y[40]=6650; //DESRNG 330

y[41]=Price767(y[21],y[22],y[23],y[24],y[25],y[26],y[27],y[28],y[29],y[30],y[20]);
      //767 Price
y[42]=Yield767(y[21],y[22],y[23],y[24],y[25],y[26],y[27],y[28],y[29],y[30],y[20]);
      //767 Yield

y[43]=Price330(y[31],y[32],y[33],y[34],y[35],y[36],y[37],y[38],y[39],y[40],y[20]);
      //330 Price
y[44]=Yield330(y[31],y[32],y[33],y[34],y[35],y[36],y[37],y[38],y[39],y[40],y[20]);
      //330 Yield

y[45]=0.15; //Yield Attr
y[46]=100; //Price Attr
y[47]=250; //Pax Attr
y[48]=6000; //Rng Attr
y[49]=28; //Time of Intro 767
y[50]=0; //Time of Intro 330
y[51]=49; //Time of Exit 767

y[52]=pulse(y[49],y[51]-y[49],t0)*(y[7]+1-y[42]/y[45]+1-y[41]/y[46]+y[30]/y[48]+
                 y[25]/y[47]); //Attractiveness of 767
y[53]=pulse(y[50],200,t0)*(y[8]+1-y[44]/y[45]+1-y[43]/y[46]+y[40]/y[48]+
                 y[35]/y[47]); //Attractiveness of 330

y[54]=y[52]+y[53]; //Total Attractiveness

y[55]=y[52]/y[54]; //Market Share of 767
y[56]=y[53]/y[54]; //Market Share of 330

y[57]=y[0]*y[55]; //Sales of 767
y[58]=y[0]*y[56]; //Sales of 330
```

```java
            return y;
        }

// Progress algebra solution

        public void evaluateAlgebraicEquations(double t) {
            y[0]=rndUniform(y[59],vrangel[59],vrangeh[59],Demand(t));
            if (y[0]<0)
                y[0]=0;

            y[7]=Math.exp(y[3]*ystock[0]/y[5]); //Effect of Commonality on Attractiveness of
                 767
            y[8]=Math.exp(y[4]*ystock[1]/y[6]); //Effect of Commonality on Attractiveness of
                 330

            y[13]=y[10]*(Math.sqrt(24*12/dt)*(rnd.nextDouble()-0.5)); //White Noise
            y[14]=(y[13] - ystock[2])/y[12]; //Change in Pink Noise

            y[18]=1+ramp(y[15]/120,y[16],y[17],t)+ystock[2]; //Pink Noise Input

            y[20]=y[19]*y[18]; //COFL

            y[41]=Price767(y[21],y[22],y[23],y[24],y[25],y[26],y[27],y[28],y[29],y[30],y[20]);
                 //767 Price
            y[42]=Yield767(y[21],y[22],y[23],y[24],y[25],y[26],y[27],y[28],y[29],y[30],y[20]);
                 //767 Yield

            y[43]=Price330(y[31],y[32],y[33],y[34],y[35],y[36],y[37],y[38],y[39],y[40],y[20]);
                 //330 Price
            y[44]=Yield330(y[31],y[32],y[33],y[34],y[35],y[36],y[37],y[38],y[39],y[40],y[20]);
                 //330 Yield

            y[52]=pulse(y[49],y[51]-y[49],t)*(y[7]+1-y[42]/y[45]+1-y[41]/y[46]+y[30]/y[48]+
                       y[25]/y[47]); //Attractiveness of 767
            y[53]=pulse(y[50],200,t)*(y[8]+1-y[44]/y[45]+1-y[43]/y[46]+y[40]/y[48]+
                       y[35]/y[47]); //Attractiveness of 330

            y[54]=y[52]+y[53]; //Total Attractiveness

            y[55]=y[52]/y[54]; //Market Share of 767
            y[56]=y[53]/y[54]; //Market Share of 330

            y[57]=y[0]*y[55]; //Sales of 767
            y[58]=y[0]*y[56]; //Sales of 330
        }

// progress deterministic algebra solution

        public void evaluateAlgebraicEquationsdeterministic(double t) {
            y[0]=Demand(t);
            if (y[0]<0)
                y[0]=0;

            y[7]=Math.exp(y[3]*ystock[0]/y[5]); //Effect of Commonality on Attractiveness of
                 767
```

```
            y[8]=Math.exp(y[4]*ystock[1]/y[6]); //Effect of Commonality on Attractiveness of
                330

            y[13]=y[10]*(Math.sqrt(24*12/dt)*0); //White Noise
            y[14]=(y[13] - ystock[2])/y[12]; //Change in Pink Noise
            y[18]=1+ramp(y[15]/120,y[16],y[17],t)+ystock[2]; //Pink Noise Input

            y[20]=y[19]*y[18]; //COFL

            y[41]=Price767(y[21],y[22],y[23],y[24],y[25],y[26],y[27],y[28],y[29],y[30],y[20]);
                 //767 Price
            y[42]=Yield767(y[21],y[22],y[23],y[24],y[25],y[26],y[27],y[28],y[29],y[30],y[20]);
                 //767 Yield

            y[43]=Price330(y[31],y[32],y[33],y[34],y[35],y[36],y[37],y[38],y[39],y[40],y[20]);
                 //330 Price
            y[44]=Yield330(y[31],y[32],y[33],y[34],y[35],y[36],y[37],y[38],y[39],y[40],y[20]);
                 //330 Yield

            y[52]=pulse(y[49],y[51]-y[49],t)*(y[7]+1-y[42]/y[45]+1-y[41]/y[46]+y[30]/y[48]+
                          y[25]/y[47]); //Attractiveness of 767
            y[53]=pulse(y[50],200,t)*(y[8]+1-y[44]/y[45]+1-y[43]/y[46]+y[40]/y[48]+
                          y[35]/y[47]); //Attractiveness of 330

            y[54]=y[52]+y[53]; //Total Attractiveness

            y[55]=y[52]/y[54]; //Market Share of 767
            y[56]=y[53]/y[54]; //Market Share of 330

            y[57]=y[0]*y[55]; //Sales of 767
            y[58]=y[0]*y[56]; //Sales of 330
        }

// Complete ODE solution

        public void solveODE() {
            int i,k;
            double t=t0;

            for (i=0;i<nvar;i++) {
                v[i][0]=y0[i];
                y[i]=y0[i];
            }

            ystock[0]=y0stock[0]; //+y[16]*rnd.nextGaussian();
            if (ystock[0]<0.0)
                ystock[0]=0.0;
            vstock[0][0]=ystock[0];
            ystock[1]=y0stock[1]; //+y[16]*rnd.nextGaussian();
            if (ystock[1]<0.0)
                ystock[1]=0.0;
            vstock[1][0]=ystock[1];

            //              progress solution by a time step
            for (k=1;k<=nstep;k++) {
```

```
                evaluateAlgebraicEquations(t);
                dy=dfxy(t,ystock);
                ystock=CompetitionSolver.solve(dt,t,nvarstock,ystock,dy,this);
                for (i=0;i<nvarstock;i++) {
                        vstock[i][k]=ystock[i];
                }
                for (i=0;i<nvar;i++) {
                        v[i][k]=y[i];
                }
                t+=dt;
        }
    }

//Complete deterministic ODE solution

    public void solveODEdeterministic() {
        int i,k;
        double t=t0;

        for (i=0;i<nvar;i++) {
                v[i][0]=y0[i];
                y[i]=y0[i];
        }

        for (i=0;i<nvarstock;i++) {
                vstock[i][0]=y0stock[i];
                ystock[i]=y0stock[i];
        }

        //              progress solution by a time step
        for (k=1;k<=nstep;k++) {
                evaluateAlgebraicEquationsdeterministic(t);
                dy=dfxy(t,ystock);
                ystock=CompetitionSolver.solve(dt,t,nvarstock,ystock,dy,this);
                for (i=0;i<nvarstock;i++) {
                        vstock[i][k]=ystock[i];
                }
                for (i=0;i<nvar;i++) {
                        v[i][k]=y[i];
                }
                t+=dt;
        }
    }

// Rates of change for ODE solver

    public double[] dfxy(double x, double y[]) {
        double[] dy;
        dy = new double[nvarstock];

        dy[0]=this.y[57];
        dy[1]=this.y[58];
        dy[2]=this.y[14];

        return dy;
```

```
        }

// Run complete Monte Carlo of ODE solution

        void runMonteCarlo(double lowLimitBinsib, double highLimitBinsib) {
                int i,k;
                int bins = 1001;
                double lowLimitBinsms = 0.0;
                double highLimitBinsms = 1.0;
                double rangeBinsib = highLimitBinsib-lowLimitBinsib;
                double rangeBinsms = highLimitBinsms-lowLimitBinsms;
                int runs = 10000;
                double invruns = 1.0/runs;
                double bins_over_rangeBinsib=bins/rangeBinsib;
                double bins_over_rangeBinsms=bins/rangeBinsms;

                ib1bins = new double[bins][nstep+1];
                for (i=0;i<bins;i++) {
                        ib1bins[i] = new double[nstep+1];
                }
                ib1mean = new double[nstep+1];
                ib1con50u = new double[nstep+1];
                ib1con50l = new double[nstep+1];

                ib2bins = new double[bins][nstep+1];
                for (i=0;i<bins;i++) {
                        ib2bins[i] = new double[nstep+1];
                }
                ib2mean = new double[nstep+1];
                ib2con50u = new double[nstep+1];
                ib2con50l = new double[nstep+1];

                ms1bins = new double[bins][nstep+1];
                for (i=0;i<bins;i++) {
                        ms1bins[i] = new double[nstep+1];
                }
                ms1mean = new double[nstep+1];
                ms1con50u = new double[nstep+1];
                ms1con50l = new double[nstep+1];

                ms2bins = new double[bins][nstep+1];
                for (i=0;i<bins;i++) {
                        ib2bins[i] = new double[nstep+1];
                }
                ms2mean = new double[nstep+1];
                ms2con50u = new double[nstep+1];
                ms2con50l = new double[nstep+1];

                int currentbin;

                // do all the runs
                for(k=0;k<runs;k++) {
                        solveODE();

                        // count bins
```

```
        for (i=0;i<=nstep;i++) {
            currentbin = (int)Math.round((vstock[0][i]-lowLimitBinsib)*
                bins_over_rangeBinsib);
            if ((currentbin>=0)&&(currentbin<bins))
                ib1bins[currentbin][i]++;
            ib1mean[i]+=vstock[0][i];

            currentbin = (int)Math.round((vstock[1][i]-lowLimitBinsib)*
                bins_over_rangeBinsib);
            if ((currentbin>=0)&&(currentbin<bins))
                ib2bins[currentbin][i]++;
            ib2mean[i]+=vstock[1][i];

            currentbin = (int)Math.round((v[55][i]-lowLimitBinsms)*
                bins_over_rangeBinsms);
            if ((currentbin>=0)&&(currentbin<bins))
                ms1bins[currentbin][i]++;
            ms1mean[i]+=v[55][i];

            currentbin = (int)Math.round((v[56][i]-lowLimitBinsms)*
                bins_over_rangeBinsms);
            if ((currentbin>=0)&&(currentbin<bins))
                ms2bins[currentbin][i]++;
            ms2mean[i]+=v[56][i];
        }
    }

    double[] testconf = new double[bins+1];

    // find confidence intervals
    for (i=0;i<=nstep;i++) {
        ib1bins[0][i]*=invruns;
        testconf[0]=ib1bins[0][i];
        for (k=1;k<bins;k++) {
            ib1bins[k][i]*=invruns;
            testconf[k]=testconf[k-1]+ib1bins[k][i];
        }
        ib1mean[i]*=invruns;
        k=0;
        while ((testconf[k]<0.25)&&(k<bins)) {
            k++;
        }
        k--;
        if (k<0)
            k=0;
        ib1con50l[i]=(k+0.5)*rangeBinsib/(bins-1)+lowLimitBinsib;

        k=bins-1;
        while ((testconf[k]>0.75)&&(k>0)) {
            k--;
        }
        k++;
        if (k>bins-1)
            k=bins-1;
        ib1con50u[i]=(k+0.5)*rangeBinsib/(bins-1)+lowLimitBinsib;
```

```
}

for (i=0;i<=nstep;i++) {
	ib2bins[0][i]*=invruns;
	testconf[0]=ib2bins[0][i];
	for (k=1;k<bins;k++) {
		ib2bins[k][i]*=invruns;
		testconf[k]=testconf[k-1]+ib2bins[k][i];
	}
	ib2mean[i]*=invruns;
	k=0;
	while ((testconf[k]<0.25)&&(k<bins)) {
		k++;
	}
	k--;
	if (k<0)
		k=0;
	ib2con50l[i]=(k+0.5)*rangeBinsib/(bins-1)+lowLimitBinsib;

	k=bins-1;
	while ((testconf[k]>0.75)&&(k>0)) {
		k--;
	}
	k++;
	if (k>bins-1)
		k=bins-1;
	ib2con50u[i]=(k+0.5)*rangeBinsib/(bins-1)+lowLimitBinsib;
}

for (i=0;i<=nstep;i++) {
	ms1bins[0][i]*=invruns;
	testconf[0]=ms1bins[0][i];
	for (k=1;k<bins;k++) {
		ms1bins[k][i]*=invruns;
		testconf[k]=testconf[k-1]+ms1bins[k][i];
	}
	ms1mean[i]*=invruns;
	k=0;
	while ((testconf[k]<0.15)&&(k<bins)) {
		k++;
	}
	k--;
	if (k<0)
		k=0;
	ms1con50l[i]=(k+0.5)*rangeBinsms/(bins-1)+lowLimitBinsms;

	k=bins-1;
	while ((testconf[k]>0.85)&&(k>0)) {
		k--;
	}
	k++;
	if (k>bins-1)
		k=bins-1;
	ms1con50u[i]=(k+0.5)*rangeBinsms/(bins-1)+lowLimitBinsms;
```

```
        }


        for (i=0;i<=nstep;i++) {
                ms2bins[0][i]*=invruns;
                testconf[0]=ms2bins[0][i];
                for (k=1;k<bins;k++) {
                        ms2bins[k][i]*=invruns;
                        testconf[k]=testconf[k-1]+ms2bins[k][i];
                }
                ms2mean[i]*=invruns;
                k=0;
                while ((testconf[k]<0.15)&&(k<bins)) {
                        k++;
                }
                k--;
                if (k<0)
                        k=0;
                ms2con50l[i]=(k+0.5)*rangeBinsms/(bins-1)+lowLimitBinsms;

                k=bins-1;
                while ((testconf[k]>0.85)&&(k>0)) {
                        k--;
                }
                k++;
                if (k>bins-1)
                        k=bins-1;
                ms2con50u[i]=(k+0.5)*rangeBinsms/(bins-1)+lowLimitBinsms;
        }
}

// run Monte Carlo and dump
double[][] runMonteCarloOutput() {
        int i,k;
        int runs = 5000;

        double[][] output = new double[nvar+nvarstock][runs];
        for (i=0;i<nvar+nvarstock;i++) {
                output[i] = new double[runs];
        }

        for(k=0;k<runs;k++) {
                y0 = setupDefaultAlgebraicEquationsMC();
                solveODE();

                for(i=0;i<nvarstock;i++) {
                        output[i][k]=vstock[i][nstep];
                }
                for(i=0;i<nvar;i++) {
                        output[nvarstock+i][k]=v[i][nstep];
                }
        }
        y0 = setupDefaultAlgebraicEquations();

        return output;
```

```
}

// Ramp function
double ramp(double slope, double start, double end, double current) {
        double d=0;
        double x;

        if (current<=end) {
                x=current-start;
                d=slope*x*0.5*(1+Math.signum(x));
        }
        return d;
}

// Pulse Function
double pulse(double start, double width, double current) {
        double x;

        x=current-start;
        return 0.5*(Math.signum(x)-Math.signum(x-width));
}

// Neural Nets

double Price767(double a,double b,double c,double d,double e,
                double f,double g,double h,double i,double j,double k) {
        return 130.2700645668302 + -52.6924500150449 * 1/(1+Math.exp(-1*(-0.4476618755213
            +
                        0.0247341795421 * d + -0.0004323733945 * f +  0.0002646120372 * e
                            +
                        -1.6030774097580 * g +  0.0201035329047 * i + -0.0387918870201 * h
                            +
                        0.0000075005516 * j + -0.0067054780626 * c + -0.1103704879509 * b
                            +
                        -0.0054579659804 * a + -0.0026923199428 * k))) + -72.6671337025737
                            *
                        1/(1+Math.exp(-1*( 5.6285558401151 +  0.0872053738451 * d +
                            -0.0005619096435 * f +
                        -0.0027636845017 * e +  0.0294718380984 * g +  0.0954479630731 * i
                            +
                        -0.9907088931980 * h + -0.0001951648240 * j + -0.0955280221679 * c
                            +
                        -2.0542060064063 * b + -0.0263084900697 * a + -0.0042159321899 * k
                            ))) +
                        244.5028553614710 * 1/(1+Math.exp(-1*(-0.7840973989484 +
                            0.0797434643063 * d +
                        -0.0048200384672 * f +  0.0002101947598 * e +  0.1275565860740 * g
                            +
                        -0.0261204675135 * i +  0.1072523587227 * h +  0.0000091644411 * j
                            +
                        0.0091087542216 * c +  0.0197435302884 * b +  0.0007573142979 * a
                            +
                        0.0002903760220 * k))) + 105.3871453263820 * 1/(1+Math.exp(-1*(
                            3.4204517208660 +
```

```
                    0.1148812734344 * d + -0.0008854177664 * f + -0.0012608188257 * e
                        +
                    0.0935679353372 * g +  0.0628673291564 * i + -0.4919468565841 * h
                        +
                    -0.0001046559937 * j + -0.0442205030687 * c + -1.0683925946769 * b
                        +
                    -0.0154413210036 * a + -0.0025432512885 * k))) + -60.3865644210801
                        *
                    1/(1+Math.exp(-1*( 0.6005394556168 + -0.1252661748348 * d +
                        0.0022704307890 * f +
                    -0.0006080449609 * e + -0.0226722448275 * g + -0.0331260067931 * i
                        +
                    -0.0936789734599 * h + -0.0000217670482 * j + -0.0225291259579 * c
                        +
                    -0.1491469968637 * b + -0.0018815221137 * a +  0.0030904518080 * k
                        )));
}

double Yield767(double a,double b,double c,double d,double e,
        double f,double g,double h,double i,double j,double k) {
    return 3.0693436949679 +  1.4566356364833 * 1/(1+Math.exp(-1*(-5.2863421903429 +
                     -0.0373731261151 * d +  0.0012036038376 * f + -0.0093270280066 *
                         e +
                     -0.0718854047870 * g + -0.0083918775486 * i +  5.5011660620383 *
                         h +
                     -0.0000288517482 * j +  0.0270896487443 * c +  1.8009591750375 *
                         b +
                     -0.0141456621516 * a + -0.0428232954624 * k))) + -3.9493456523727
                         *
                     1/(1+Math.exp(-1*( 4.0485216546828 + -0.0026525710810 * d +
                         0.0000596568225 * f +
                    0.0040794554266 * e +  0.0175864964870 * g +  0.0002928666207 * i
                        +
                    -1.4501369695425 * h + -0.0002084072559 * j +  0.0171050673527 * c
                        +
                    -0.2278874744205 * b +  0.0006629152134 * a + -0.0809770071050 * k
                        ))) +
                    -0.5325717439828 * 1/(1+Math.exp(-1*( 8.7760699448306 +
                        0.0108705351097 * d +
                    -0.0001919064640 * f +  0.0007585171177 * e +  0.2039496556537 * g
                        +
                    0.0083337490571 * i + -3.2858689562975 * h + -0.0005995495802 * j
                        +
                    0.4072268218722 * c + -1.0583486775285 * b + -0.0365106773514 * a
                        +
                    -0.1958041145080 * k))) +  1.4887264196259 * 1/(1+Math.exp(-1*(
                        5.1479206591727 +
                    0.0352553727779 * d + -0.0011223575756 * f +  0.0092424979685 * e
                        +
                    0.0678584745381 * g +  0.0034586673169 * i + -5.5924270699493 * h
                        +
                    0.0000606363431 * j + -0.0245206895962 * c + -1.7948086826103 * b
                        +
                    0.0130865062246 * a +  0.0458422452150 * k))) + -2.8755199176456 *
```

```
                    1/(1+Math.exp(-1*(-4.3424235061584 + -0.0012059024256 * d +
                        0.0000428140376 * f +
                    -0.0037068812514 * e + -0.0369795831814 * g + -0.0050741772924 * i
                         +
                    1.4186067234081 * h +  0.0002910381194 * j + -0.0284269109037 * c
                        +
                    0.2195757321281 * b + -0.0008181056205 * a + -0.1265897196922 * k)
                        ));
}

double Price330(double a,double b,double c,double d,double e,
        double f,double g,double h,double i,double j,double k) {
    return 59.5710573155849 + 71.1908780113579 * 1/(1+Math.exp(-1*(-5.6468795699341 +
                    -0.0870018038178 * d +  0.0005529846854 * f +  0.0027908239535 * e
                         +
                    -0.0293233068538 * g + -0.0963061832671 * i +  0.9962492361890 * h
                         +
                    0.0001960494455 * j +  0.0960264332186 * c +  2.0630224324930 * b
                        +
                    0.0263974738841 * a +  0.0042650063635 * k))) + -52.9313406714092
                        *
                    1/(1+Math.exp(-1*(-0.4536092980619 +  0.0246784343503 * d +
                        -0.0004315971352 * f +
                    0.0002657394471 * e + -1.5993737003602 * g +  0.0199764912908 * i
                        +
                    -0.0383861860337 * h +  0.0000076252749 * j + -0.0066239650187 * c
                         +
                    -0.1097956471142 * b + -0.0054183631768 * a + -0.0026862912129 * k
                        ))) +
                    244.5068062857333 * 1/(1+Math.exp(-1*(-0.7812559424607 +
                        0.0796105961641 * d +
                    -0.0048169720932 * f +  0.0002094090905 * e +  0.1273184062420 * g
                         +
                    -0.0260576982189 * i +  0.1067253105316 * h +  0.0000090765426 * j
                         +
                    0.0090364134375 * c +  0.0189763818784 * b +  0.0007419502967 * a
                        +
                    0.0002791425092 * k))) + 103.5719369120488 * 1/(1+Math.exp(-1*(
                        3.4090409855194 +
                    0.1153512588558 * d + -0.0008846587414 * f + -0.0012605953838 * e
                        +
                    0.0939284716601 * g +  0.0631375587816 * i + -0.4900800650465 * h
                        +
                    -0.0001042638722 * j + -0.0439753740828 * c + -1.0635185307040 * b
                         +
                    -0.0153955769518 * a + -0.0025587242805 * k))) + -60.7072195377685
                         *
                    1/(1+Math.exp(-1*( 0.6033016713352 + -0.1249935716478 * d +
                        0.0022608217211 * f +
                    -0.0006022208404 * e + -0.0220820733166 * g + -0.0330820800556 * i
                         +
                    -0.0931237565603 * h + -0.0000216702264 * j + -0.0224602850076 * c
                         +
                    -0.1480055423557 * b + -0.0018847425880 * a +  0.0030796822675 * k
                        )));
```

```
}

double Yield330(double a,double b,double c,double d,double e,
                double f,double g,double h,double i,double j,double k) {
        return 3.6027996879991 + -0.0467334695758 * 1/(1+Math.exp(-1*( 7.7966192358722 +
                        0.0637671497365 * d + -0.0022364716472 * f +  0.0095741574807 * e
                            +
                        0.1175388265523 * g +  0.0831958036904 * i + -3.4420350386214 * h
                            +
                        -0.0005563954284 * j + -0.0713240302585 * c + -2.0707917995720 * b
                            +
                        0.0281350480579 * a + -0.0301847940991 * k))) +  1.1645502311478 *
                        1/(1+Math.exp(-1*(-8.4739293408969 + -0.0079510739148 * d +
                            0.0001491645202 * f +
                        -0.0012563926543 * e + -0.1553732553647 * g + -0.0189318997145 * i
                            +
                        2.7620393055610 * h +  0.0005424599823 * j + -0.3355547808227 * c
                            +
                        0.9526561815595 * b +  0.0280658505695 * a +  0.1735810810957 * k)
                            )) +
                        -3.5250080022768 * 1/(1+Math.exp(-1*( 3.9541743675685 +
                            -0.0031388817928 * d +
                        0.0000727380914 * f +  0.0041032046117 * e +  0.0164097259083 * g
                            +
                        -0.0016356746488 * i + -1.4334054165356 * h + -0.0002086110487 * j
                            +
                        0.0148800304857 * c + -0.2021361861867 * b +  0.0010935361500 * a
                            +
                        -0.0825609151256 * k))) + -2.7048546930824 * 1/(1+Math.exp
                            (-1*(-4.3155107975333 +
                        -0.0012334955330 * d +  0.0000442388406 * f + -0.0037897981876 * e
                            +
                        -0.0357319875596 * g + -0.0047172021118 * i +  1.4302179738963 * h
                            +
                        0.0002987069152 * j + -0.0294582652590 * c +  0.2628328970787 * b
                            +
                        -0.0014858706048 * a + -0.1282800025353 * k))) +  0.0461690844377
                            *
                        1/(1+Math.exp(-1*( 3.1954449106749 +  0.0052712799265 * d +
                            0.0000258978199 * f +
                        0.0070279828114 * e +  0.0031348712681 * g + -0.1202563717282 * i
                            +
                        -8.1358438979762 * h +  0.0007571515616 * j + -0.0273747922347 * c
                            +
                        -1.3255772794832 * b +  0.0035605732277 * a +  0.1810005820685 * k
                            )));
}

 // external demand function with data
double Demand(double t0) {
        double[] t = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,
        21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,
        41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,
                        61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,
                        82,83,84,85,86,87,88,89,90,91,92};
```

```
        double[] d = {1,1,0,1,4,0,3,2,0,2,4,2,3,4,7,3,2,3,6,3,1,3,2,2,2,5,1,1,
        3,7,6,5,4,4,1,2,3,4,2,0,0,0,0,3,4,7,4,7,5,2,4,3,1,3,1,4,7,0,1,6,4,1,3,
        1,0,0,1,1,1,0,3,0,3,0,1,2,3,4,2,4,3,0,2,4,3,2,5,0,2,1,5,1,3};
        int i;
        for(i=0;i<t.length;i++) {
            if (t[i]>t0)
                break;
        }

        double r = d[i-1]+(d[i]-d[i-1])/(t[i]-t[i-1])*(t0-t[i-1]);
        return  r;
    }

    // scalable uniform random number distribution with clipping
    double rndUniform(double scale, double min, double max, double def) {
        double diff = max-min;
        double diff1 = def-min;
        double diff2 = max-def;
        double lowstep = scale*diff1;
        double highstep = scale*diff2;
        double low = def - lowstep;
        double high = def + highstep;
        double r = low+(high-low)*rnd.nextDouble();
            //min+0.5*diff+diff*(scale*(rnd.nextDouble()-0.5));
        return r;
    }

}
```

C.3 Extended Unified Trade-off Environment Source Code

This section contains the source code that actually implements the EUTE. Due to the high use of readily available graphing libraries, a significant amount of code purely related to the display of data on the screen with the help of these libraries as well as export to files was omitted for brevity. However, this still represents the complete implementation of this interactive environment.

Listing C.6: competitionmodel6.java

```
package edu.gatech.asdl.pfaender.competitionmodel;

import org.jfree.ui.*;
import org.jfree.chart.*;
import org.jfree.chart.axis.*;
import org.jfree.chart.plot.*;
```

```java
import org.jfree.chart.renderer.xy.*;
import org.jfree.data.time.*;
import org.jfree.data.*;
import org.jfree.data.xy.*;
import java.awt.Color;
import javax.swing.*;
import java.awt.geom.*;
import java.awt.event.*;
import java.awt.*;
import java.util.*;
import java.io.*;
import java.text.*;
import com.lowagie.text.Document;
import com.lowagie.text.DocumentException;
import com.lowagie.text.Rectangle;
import com.lowagie.text.pdf.DefaultFontMapper;
import com.lowagie.text.pdf.FontMapper;
import com.lowagie.text.pdf.PdfContentByte;
import com.lowagie.text.pdf.PdfTemplate;
import com.lowagie.text.pdf.PdfWriter;

// Class that create complete EUTE environment

public class competitionmodel6 extends ApplicationFrame implements ActionListener {

        double lowlimitib = 0.0;
        double highlimitib = 200.0;
        int currentTime;
        int overallFontSize = 14;
        Month startMonth = new Month(4,1998);
        competitionmodelCalculator3 cmc;
        private int CHART_COUNT = 2*4;

        // Class to display time axis

    class competitionPanel extends JPanel implements ChartMouseListener {
                public JFreeChart timechart;
                private ChartPanel timechartPanel;
                private XYPlot[] timeplots = new XYPlot[4];

            public competitionPanel(competitionmodelCalculator3 cmc) {
            // omitted for brevity
            }

            // create dataset for plotting from results in cmc
            private XYDataset createDatasetib1simple(int n) {

                TimeSeries series1 = new TimeSeries(" ");
                TimeSeries series2 = new TimeSeries(" ");

                Month month = new Month(startMonth.getMonth(),startMonth.getYearValue());

                if (n==1) {
                     series1 = new TimeSeries("Mean", Month.class);
                     series2 = new TimeSeries("Lower 50% Confidence", Month.class);
```

```
                    for (int i = 0; i <= cmc.nstep; i++) {
                            series1.add(month, cmc.ib1mean[i]);
                            series2.add(month, cmc.ib1con50l[i]);
                            month = (Month) month.next();
                    }
    }

    if (n==2) {
        series1 = new TimeSeries("Upper 50% Confidence", Month.class);
        series2 = new TimeSeries("Mean", Month.class);
            for (int i = 0; i <= cmc.nstep; i++) {
                    series1.add(month, cmc.ib1con50u[i]);
                    series2.add(month, cmc.ib1mean[i]);
                    month = (Month) month.next();
            }
    }
    TimeSeriesCollection dataset = new TimeSeriesCollection();
    dataset.addSeries(series1);
    dataset.addSeries(series2);
    return dataset;
}

private XYDataset createDatasetib2simple(int n) {
// omitted for brevity
    }

private XYDataset createDatasetms1simple(int n) {
// omitted for brevity
}

private XYDataset createDatasetms2simple(int n) {
// omitted for brevity
}

// refresh Monte Carlo run
public void refreshMC(double x){
        currentTime = timeToPoint(x);
        cmc.runMonteCarlo(lowlimitib, highlimitib);
}

// update all the data sets after a run
public void refreshData() {
        updateDataset(0,0, createDatasetib1simple(1));
        updateDataset(0,1, createDatasetib1simple(2));
        updateDataset(1,0, createDatasetms1simple(1));
        updateDataset(1,1, createDatasetms1simple(2));
        updateDataset(2,0, createDatasetib2simple(1));
        updateDataset(2,1, createDatasetib2simple(2));
        updateDataset(3,0, createDatasetms2simple(1));
        updateDataset(3,1, createDatasetms2simple(2));
}

// format plot
private JFreeChart createChart() {
// omitted for brevity
```

```
        }

            // link cross hairs together in every column and rerun Monte Carlo

            public void chartMouseClicked(ChartMouseEvent event) {
                int mouseX = event.getTrigger().getX();
                int mouseY = event.getTrigger().getY();

                Point2D p = timechartPanel.translateScreenToJava2D(new Point(mouseX, mouseY));
                CombinedDomainXYPlot plot = (CombinedDomainXYPlot) timechart.getPlot();

                PlotRenderingInfo pri = timechartPanel.getChartRenderingInfo().getPlotInfo();
                int subplotindex = pri.getSubplotIndex(p);
                if (subplotindex>=0) {
                        PlotRenderingInfo subplotinfo = pri.getSubplotInfo(subplotindex);
                        Rectangle2D plotArea = subplotinfo.getDataArea();

                        if (plotArea!=null) {
                                    ValueAxis domainAxis = plot.getDomainAxis();
                                RectangleEdge domainAxisEdge = plot.getDomainAxisEdge();
                                double chartX = domainAxis.java2DToValue(p.getX(), plotArea,
                                     domainAxisEdge);

                                this.refreshMC(chartX);
                                cp.refreshData();
                                for (int i = 0; i< pps.length;i++) {
                                    pps[i].refreshData();
                                }

                                for (int i = 0; i< pp.length;i++) {
                                    pp[i].refreshData();
                                }

                                java.util.List ListSubplots = plot.getSubplots();
                                    for (int i = 0; i < ListSubplots.size()-1; i++) {
                                    XYPlot subplot = (XYPlot) ListSubplots.get(i);
                                    subplot.setDomainCrosshairValue(chartX,false);
                                }
                                XYPlot subplot = (XYPlot) ListSubplots.get(ListSubplots.size()
                                    -1);
                                subplot.setDomainCrosshairValue(chartX,true);
                    }
                }
        }

        public void chartMouseMoved(ChartMouseEvent event) {
        }
    }

// class to display stock variables
class profilerPanelstock extends JPanel implements ChartMouseListener {
            private JFreeChart chart;
            private ChartPanel chartPanel;
            private int numpoints = 100;
            private boolean rangeAxisVisible = false;
```

```
            private XYPlot[] profilerplots = new XYPlot[4];
            int stock;
            String name;

        public profilerPanelstock(competitionmodelCalculator3 cmc, int s, String n) {
        // omitted for brevity
        }

        private XYDataset createDatasetib1() {
        // omitted for brevity
        }

        private XYDataset createDatasetib2() {
        // omitted for brevity
            }

        private XYDataset createDatasetms1() {
        // omitted for brevity
        }

        private XYDataset createDatasetms2() {
        // omitted for brevity
            }

        public void updateDataset(int i, XYDataset dataset) {
            profilerplots[i].setDataset(dataset);
        }

        public void refreshMC(double x) {
                cmc.y0stock[this.stock]=x;
                cmc.runMonteCarlo(lowlimitib, highlimitib);
        }

        public void refreshData() {
                updateDataset(0, createDatasetib1());
                updateDataset(1, createDatasetms1());
                updateDataset(2, createDatasetib2());
                updateDataset(3, createDatasetms2());
        }

       // format plot
        private JFreeChart createChart() {
        // omitted for brevity
        }

        // link cross hairs and update Monte Carlo
            public void chartMouseClicked(ChartMouseEvent event) {
        // omitted for brevity
        }

        public void chartMouseMoved(ChartMouseEvent event) {
        }
}

// class to display influence variables
```

```
class profilerPanel extends JPanel implements ChartMouseListener {
            private JFreeChart chart;
            private ChartPanel chartPanel;
            private int numpoints = 100;
            private boolean rangeAxisVisible = false;
            private XYPlot[] profilerplots = new XYPlot[4];
            int var;
            String name;

        public profilerPanel(competitionmodelCalculator3 cmc, int v, String n) {
        // omitted for brevity
        }

        // run deterministic solution hold all but one variable constant
        private XYDataset createDatasetib1() {

                    XYSeries series1 = new XYSeries("_");
                    double temp;
                    int j = var;
                    int k = 0;

                    temp = cmc.y0[var];
             double inc = (cmc.vrangeh[j]-cmc.vrangel[j])/numpoints;
             double x = cmc.vrangel[j];
                    for (int i = 0; i <= numpoints; i++) {
                            cmc.y0[var]=x;
                            cmc.solveODEdeterministic();
                            series1.add(x, cmc.vstock[k][currentTime]); //cmc.v[k].length-1
                            x+=inc;
                    }
                    cmc.y0[var]=temp;

            XYSeriesCollection dataset = new XYSeriesCollection();
            dataset.addSeries(series1);
            return dataset;
        }

        private XYDataset createDatasetib2() {
        // omitted for brevity
            }

        private XYDataset createDatasetms1() {
        // omitted for brevity
        }

        private XYDataset createDatasetms2() {
        // omitted for brevity
        }

        public void refreshMC(double x) {
                cmc.y0[this.var]=x;
                cmc.runMonteCarlo(lowlimitib,highlimitib);
        }

        public void refreshData() {
```

```
        updateDataset(0, createDatasetib1());
        updateDataset(1, createDatasetms1());
        updateDataset(2, createDatasetib2());
        updateDataset(3, createDatasetms2());
    }

    // format chart
    private JFreeChart createChart() {
    // omitted for brevity
    }


        public void chartMouseClicked(ChartMouseEvent event) {
    // omitted for brevity
    }

    public void chartMouseMoved(ChartMouseEvent event) {
    }
}

JPanel mainPanel;
competitionPanel cp;
profilerPanelstock[] pps = new profilerPanelstock[2];
profilerPanel[] pp = new profilerPanel[20];
JMenuItem menuItem,menuItem2;

// setup overall display

public competitionmodel6() {
  super("Competition Model Demo");
  mainPanel = new JPanel();

  cmc = new competitionmodelCalculator3();
  cmc.runMonteCarlo(lowlimitib, highlimitib);

  currentTime = cmc.nstep-1;

  GridLayout layout = new GridLayout(1,4);
  mainPanel.setLayout(layout);

  cp = new competitionPanel(cmc);
  mainPanel.add(cp);

  pp[0] = new profilerPanel(cmc,59,"Volatility of Demand (#/Month)");
  mainPanel.add(pp[0]);
  pp[1] = new profilerPanel(cmc,10,"Volatility Fuel Price ($/gal)");
  mainPanel.add(pp[1]);
  pp[2] = new profilerPanel(cmc,19,"Initial Fuel Price ($/gal)");
  mainPanel.add(pp[2]);
  pp[3] = new profilerPanel(cmc,15,"Fuel Price Slope ($/Year)");
  mainPanel.add(pp[3]);
  pp[4] = new profilerPanel(cmc,45,"Yield Attr ($/RPM)");
  mainPanel.add(pp[4]);
  pp[5] = new profilerPanel(cmc,46,"Price Attr ($Mil)");
  mainPanel.add(pp[5]);
```

```
        pp[6] = new profilerPanel(cmc,47,"PAX Attr (#)");
        mainPanel.add(pp[6]);
        pp[7] = new profilerPanel(cmc,48,"Range Attr (nm)");
        mainPanel.add(pp[7]);
        pp[8] = new profilerPanel(cmc,49,"Market Intro (Month)");
        mainPanel.add(pp[8]);
        pp[9] = new profilerPanel(cmc,51,"Market Exit (Month)");
        mainPanel.add(pp[9]);
        pp[10] = new profilerPanel(cmc,21,"Sweep 767 (deg)");
        mainPanel.add(pp[10]);
        pp[11] = new profilerPanel(cmc,22,"TR 767");
        mainPanel.add(pp[11]);
        pp[12] = new profilerPanel(cmc,23,"AR 767");
        mainPanel.add(pp[12]);
        pp[13] = new profilerPanel(cmc,24,"RTRTN 767 (%)");
        mainPanel.add(pp[13]);
        pp[14] = new profilerPanel(cmc,25,"PAX 767 (#)");
        mainPanel.add(pp[14]);
        pp[15] = new profilerPanel(cmc,26,"NVEH (#)");
        mainPanel.add(pp[15]);
        pp[16] = new profilerPanel(cmc,27,"FUCOMP (%)");
        mainPanel.add(pp[16]);
        pp[17] = new profilerPanel(cmc,28,"FACT");
        mainPanel.add(pp[17]);
        pp[18] = new profilerPanel(cmc,29,"CLTOM");
        mainPanel.add(pp[18]);
        pp[19] = new profilerPanel(cmc,30,"DESRNG (nm)");
        mainPanel.add(pp[19]);

        // add menu items to right click menu

        JPopupMenu jpm = cp.timechartPanel.getPopupMenu();
        jpm.addSeparator();
        menuItem = new JMenuItem("Save PDF");
        jpm.add(menuItem);
        menuItem.addActionListener(this);
        menuItem2 = new JMenuItem("Save MC");
        jpm.add(menuItem2);
        menuItem2.addActionListener(this);

        add(mainPanel);
        mainPanel.setPreferredSize(new java.awt.Dimension(4200, 900));
    }

// event handler for right click menu

    public void actionPerformed(ActionEvent e){
        JMenuItem source = (JMenuItem)(e.getSource());
        if (source.equals(menuItem)) {
                try {
                        File fileName = new File("jfreechart1.pdf");
                        saveChartAsPDF(fileName, 4300, 900, new DefaultFontMapper());
                }
                catch (IOException ioe) {
                        System.out.println(ioe.getMessage());
```

```
                }
        }
        if (source.equals(menuItem2)) {
                        File fileName = new File("mc.csv");
                        writeMC(fileName);
        }

    }

//unit conversion functions
    public int timeToPoint(double t) {
        double l = cp.timechart.getXYPlot().getDomainAxis().getLowerBound();
        double h = cp.timechart.getXYPlot().getDomainAxis().getUpperBound();

        int i = (int) Math.round(cmc.nstep*(t-l)/(h-l));
        if (i<0)
                i=0;
        if (i>=cmc.nstep)
                i=cmc.nstep-1;

        return i;
    }

    public double pointToTime(int i) {
        double l = cp.timechart.getXYPlot().getDomainAxis().getLowerBound();
        double h = cp.timechart.getXYPlot().getDomainAxis().getUpperBound();

        double t = i*(h-l)/cmc.nstep+l;
        if (t<l)
                t=l;
        if (t>h)
                t=h;

        return t;
    }

// function to save EUTE as pdf file
    public void saveChartAsPDF(File file, int width, int height, FontMapper mapper) throws
         IOException {
        OutputStream out = new BufferedOutputStream(new FileOutputStream(file));
        writeChartAsPDF(out, width, height, mapper);
        out.close();
    }

    public void writeChartAsPDF(OutputStream out, int width, int height, FontMapper mapper) throws
          IOException {
        Rectangle pagesize = new Rectangle(width, height);
        Document document = new Document(pagesize, 50, 50, 50, 50);
        try {
            PdfWriter writer = PdfWriter.getInstance(document, out);
            document.addAuthor("JFreeChart");
            document.addSubject("Demonstration");
            document.open();
            PdfContentByte cb = writer.getDirectContent();
            PdfTemplate tp_1 = cb.createTemplate(300, height);
```

```java
            Graphics2D g2_1 = tp_1.createGraphics(300, height, mapper);
            Rectangle2D r2D_1 = new Rectangle2D.Double(0, 0, 300, height);
            cp.timechart.draw(g2_1, r2D_1);
            g2_1.dispose();
            cb.addTemplate(tp_1, 0, 0);
            // .... and repeated
            // rest omitted for brevity

        }
        catch (DocumentException de) {
            System.err.println(de.getMessage());
        }
        document.close();

    }

    // save csv file of Monte Carlo

    public void writeMC(File file) {
        try {
                BufferedWriter out = new BufferedWriter(new FileWriter(file));

                double[][] output = cmc.runMonteCarloOutput();

                for(int i=0;i<output[0].length;i++) {
                        for(int j=0;j<output.length;j++) {
                                out.write(Double.toString(output[j][i])+",");
                        }
                        out.write('\n');
                }

                out.close();
        } catch (IOException ioe) {
                System.out.println(ioe.getMessage());
        }

    }

    // Main program entry point

    public static void main(String[] args) {
                competitionmodel6 cm = new competitionmodel6();

                cm.pack();
                RefineryUtilities.centerFrameOnScreen(cm);
                cm.setVisible(true);

        }
}
```

REFERENCES

[1] Owen, K., *Concorde, new shape in the sky*. Jane's, 1982.

[2] Kusiak, A., *Concurrent Engineering*. John Wiley & Sons, Inc., 1993.

[3] Schrage, D. P. and Mavris, D. N., "Integrated product/process design/development (IPPD) through robust design simulation," in *1st AIAA Aircraft Engineering, Technology, and Operations Congress, Los Angeles, CA*, September 1995.

[4] Fabrycky, W. and Blanchard, B., *Life-Cycle Cost and Economic Analysis*. Englewood Cliffs: Prentice-Hall, 1991.

[5] "DoD guide to integrated product and process development," tech. rep. version 1.0, Office of the Under Secretary of Defense, Department of Defense, February 5, 1996.

[6] Kirby, M. R., *A Methodology for Technology Identification, Evaluation, and Selection in Conceptual and Preliminary Aircraft Design*. PhD Thesis, Georgia Institute of Technology, 2001.

[7] Mavris, D. N., Baker, A., and Schrage, D. P., "IPPD through robust design simulation for an affordable short haul civil tiltrotor," in *American Helicopter Society 53rd Annual Forum*, (Virginia Beach, VA), April-May 1997.

[8] Marx, W. J., Mavris, D. N., and Schrage, D. P., "Integrating design and manufacturing for the high speed civil transport," in *American Institue of Aeronautics and Astronautics, Inc.*, September 1994.

[9] Mavris, D. N. and Bandte, O., "Economics uncertainty assessment using a combined design of experimetns/Monte Carlo simulation approach with application to an HSCT," in *Presented at the 17th Annual Conference of the International Society of Parametric Analysis*, (San Diego, CA), May 1995.

[10] Mavris, D. N., DeLaurentis, D. A., Bandte, O., and Hale, M. A., "A Stochastic Approach to Multi-disciplinary Aircraft Analysis and Design," in *36th AIAA Aerospace Sciences Meeting & Exhibit*, (Reno, NV), January 1998.

[11] Mavris, D. N., DeLaurentis, D. A., Hale, M. A., and Tai, J. C., "Elements of an emerging virtual stochastic life cycle design environment," in *4th World Aviation Congress and Exposition*, (San Francisco, CA), October 1999.

[12] Mavris, D. N., Mantis, G., and Kirby, M. R., "Demonstration of a probabilistic technique for the determination of econominc viability," in *2nd World Aviation Congress and Exposition*, (Anaheim, CA), October 1997.

[13] Box, G. E. P. and Draper, N. R., *Empirical Model-Building and Response Surfaces*. New York: John Wiley & Sons, Inc., 1987.

[14] Myers, R. H. and Montgomery, D. C., *Response Surface Methodology: Process and Product Optimization Using Design Experiments*. New York: John Wiley & Sons, Inc., 1995.

[15] Neter, J., Kutner, M. H., Nactsheim, C. J., and Wasserman, W., *Applied Linear Statistical Models*. Boston, MA: McGraw-Hill, 4th ed., 1996.

[16] Montgomery, D. C., *Design and Analysis of Experiments*. New York: John Wiley & Sons, Inc., 5th ed., 2001.

[17] Barros, P. A., Kirby, M. R., and Mavris, D. N., "Impact of sampling technique selection on the creation of response surface models," in *Society of Automotive Engineers Aerospace Congress and Exhibition 2004-01-3134*, 2004.

[18] SAS Institute. "JMP Software - Data Analysis - Statistics - Six Sigma - DOE," [online]. Available from: `http://www.jmp.com`. Accessed July 12, 2006.

[19] Simpson, T. W., *A Concept Exploration Method for Product Family Design*. PhD Thesis, Georgia Institute of Technology, 1998.

[20] Storkey, A. J., *Efficient Covariance Matrix Methods for Bayesion Gaussian Processes and Hopfield Neural Networks*. PhD Thesis, Imperial College, University of London, London, January 1999.

[21] Daberkow, D., *A Formulation of Metamodel Implementation Processes for Complex Systems Design*. PhD Thesis, Georgia Institute of Technology, June 2002.

[22] Putter, H. and Young, G. A., *On the effect of covariance function estimation on the accuracy of kriging predictions*. 1991.

[23] MacKay, D. J. C., "Introduction to Gaussian Processes," *Neural Networks and Machine Learnging*, vol. 168, pp. 133–165, 1998.

[24] Rasmussen, C. E., *Evaluation of Gaussian Processes and other Methods for Non-Linear Regression*. PhD Thesis, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, 1996.

[25] Gibbs, M. N., *Bayesian Gaussian Processes for Regression and Classification*. PhD Thesis, University of Cambridge, Cambridge, England, 1997.

[26] Johnson, C. and Schutte, J., *Basic Regression Analysis for Integrated Neural Networks (BRAINN) Documentation Version 2.0.* Georgia Institute of Technology, August 10 2006.

[27] Kirby, M. R. and Mavris, D. N., "A Technique for Selecting Emerging Technologies for a Fleet of Commercial Aircraft to Maximize R&D Investment," in *Society of Automotive Engineers Aerospace Congress and Exhibition, 2001-01-3018*, September 2001.

[28] Kirby, M. R. and Mavris, D. N., "Technology Identification, Evaluation, and Selection for Commercial Transport Aircraft," in *58th Annual Conference of the Society of Allied Weight Engineers, San Jose, CA*, May 24-26, 1999.

[29] Kirby, M. R. and Mavris, D. N., "Forecasting Technology Uncertainty in Preliminary Aircraft Design," in *4th World Aviation Congress and Exposition, San Francisco, CA*, October 19-21, 1999.

[30] Kirby, M. R., Mavris, D. N., and Largent, M. C., "A Process for Tracking and Assessing Emerging Technology Development Programs for Resource Allocation," in *1st AIAA Aircraft Engineering, Technology, and Operations Congress, Los Angeles, CA*, October 16-18, 2001.

[31] Mavris, D. N., Soban, D. S., and Largent, M. C., "An Application of a Technology Impact Forecasting (TIF) Method to an Uninhabited Combat Aerial Vehicle," in *4th World Aviation Congress and Exposition, San Francisco, CA*, October 19-21, 1999.

[32] Kirby, M. R. and Mavris, D. N., "Forecasting the Impact of Technology Infusion on Subsonic Transport Affordability," in *3rd World Aviation Congress and Exposition, Anaheim, CA*, September 28-30, 1998.

[33] Mavris, D. N., Kirby, M. R., and Qui, S., "Technology Impact Forecasting for a High Speed Civil Transport," in *3rd World Aviation Congress and Exposition, Anaheim, CA*, September 28-30, 1998.

[34] Roth, B., German, B., Mavris, D. N., and Macsotai, N., "Adaptive Selection of Engine Technology Solution Sets from a Large Combinatorial Space," in *37th Joint Propulsion Conference and Exhibit, Salt Lake City, UT*, July 2001.

[35] Baker, A. P., *The Role of Mission Requirements, Vehicle Attributes, Technologies and Uncertainty in Rotorcraft System Design.* PhD Thesis, Georgia Institute of Technology, March, 2002.

[36] Baker, A. P. and Mavris, D. N., "Assessing the simultaneous impact of requirements, vehicle characteristics, and technologies during aircraft design," in *39th Aerospace Sciences Meeting & Exhibit, Reno, NV*, no. AIAA-2001-0533, January 8-11, 2001.

[37] Mavris, D. N. and DeLaurentis, D. A., "Methodology for Examining the Simultaneous Impact of Requirements, Vehicle Characteristics, and Technologies on Military Aircraft Design," in *22nd Congress of the International Council on the Aeronautical Sciences (ICAS), Harrogate, England*, August 27-31, 2000.

[38] Mavris, D. N., Baker, A. P., and Schrage, D. P., "Simultaneous Assessment of Requirements and Technologies in Rotocraft Design," in *Annual Forum of the American Helicopter Society, Virginia Beach, VA*, May 2-4, 2000.

[39] Baker, A. P., Mavris, D. N., and Schrage, D. P., "Assessing the Impact of Mission Requirements, Vehicle Attributes, Technologies and Uncertainty in Rotorcraft System Design," in *58th Annual Forum of the American Helicopter Society, Montreal, Canada*, June 11-13, 2002.

[40] March, A., "The future of the US aircraft industry," *Technology Review*, vol. 93, no. 1, pp. 27–36, 1990.

[41] US International Trade Commission, "The changing structure of the global large civil aircraft industry and market: Implications for the competitiveness of the us industry." US International Trade Commission, Publication 3143, Washington, DC, 1998.

[42] US International Trade Commission, "Global competitiveness of us advanced-technology manufacturing industries: Large civil aircraft." US International Trade Commission, Publication 2667, Washington, DC, 1993.

[43] Wingrove, E. R., Gaier, E. M., and Santmire, T. E., "The asac air carrier investment model (third generation)." NASA Langley Research Center CR-1998-207656, Hampton, VA, 1998.

[44] Owen, A. D., Chowdury, K., and Garrido, J. R. R., "A market share model for vegetable and tropical oils," *Applied Economic Letters*, vol. 3, no. 2, pp. 95–99, 1996.

[45] Naert, A. and Weverbergh, M., "On the prediction power of market share attraction model," *Journal of Marketing Reseach*, vol. 2, no. 18, pp. 146–153, 1981.

[46] Beckwith, N. E., "Concerning the logical consistency of multivariate market share models," *Journal of Marketing Reseach*, vol. 10, no. 2, pp. 341–344, 1973.

[47] Leeflaung, P. S. H. and Reuyl, J. C., "On the predictive power of market share attraction models," *Journal of Marketing Reseach*, vol. 21, no. 2, pp. 211–215, 1984.

[48] Fernandez, A., Jacobs, D., Keating, C., and Kauffman, P., "A market attraction model for predicting the US market share of large civil aircraft," *Journal of Air Transport Management*, vol. 6, pp. 3–11, January 2000.

[49] Fernandez, A., “Market share study: Commercial aircraft industry.” Unpublished technical report, Logistics Management Institute Contract No. 9719. Old Dominion University. Norfolk, VA. As referenced in [48], 1997.

[50] Boeing Commercial Airplanes, “Current Market Outlook 2005,” 2005.

[51] Bowerman, B. and O’Connell, R., *Time Series Forecasting.* Boston, MA: Duxbury Press, 1987.

[52] *Jane’s All the World’s Aircraft.* Surrey, UK: Jane’s Information Group Limited, 1997.

[53] Horgan, J., “What’s Wrong with Complexity,” *Complexity*, vol. 2, no. 2, pp. 13–15, 1995.

[54] Forrester, J. W., *Industrial Dynamics.* Cambridge, MA: Productivity Press, 1961.

[55] Forrester, J. W., “Advertising: A problem in Industrial Dynamics,” *Harvard Business Review*, vol. 37, no. 2, pp. 100–110, 1959.

[56] Forrester, J. W., *Principles of Systems.* Cambridge, MA: Productivity Press, 1968.

[57] Forrester, J. W., *Urban Dynamics.* Cambridge, MA: Productivity Press, 1969.

[58] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., *Numerical Recipies in FORTAN: The Art of Scientific Computing, 2nd ed.* Cambridge, England: Cambridge University Press, 2 ed., 1992.

[59] Meadows, D. L., *Dynamics of Commodity Production Cycles.* Cambridge, MA: Productivity Press, 1970.

[60] Sterman, J. D. “Desert Island Dynamics: An Annotated Survey of the Essential System Dynamics Literature,” [online]. Available from: `http://web.mit.edu/jsterman/www/DID.html`. Accessed July 12, 2005.

[61] Forrester, J. W., *World Dynamics.* Cambridge, MA: Productivity Press, 2nd ed., 1973.

[62] Meadows, D., Meadows, D., Randers, J., and Behrens, W. W. I., *The Limits to Growth: A Report for the Club of Rome’s Project on the Predicament of Mankind.* New York: Universe Books, 1972.

[63] “The Club of Rome Homepage,” [online]. Available from: `http://www.clubofrome.org`. Accessed July 25th 2005.

[64] Coyle, R. G., *Management System Dynamics.* New York: John Wiley & Sons, 1977.

[65] Lyneis, J. M., *Corporate Planning and Policy Design.* Cambridge, MA: Productivity Press, 1980.

[66] Hall, R. I., "A System Pathology of an Organization: The Rise and Fall of the Old Saturday Evening Post," *Administrative Science Quarterly*, vol. 21, no. 2, pp. 185–211, 1976.

[67] Sterman, J. D., "A Behavioral Model of the Economic Long Wave," *Journal of Economic Behavior and Organization*, vol. 6, no. 1, pp. 17–53, 1985.

[68] Forrester, J. W., "The System Dynamics National Model: Macrobehavior from Microstructure," in *Computer-Based Management of Complex Systems: International System Dynamics Conference* (Milling, P. M. and Zahn, E. O. K., eds.), (Berlin), Springer Verlag, 1989.

[69] Sturis, J., Polonsky, K. S., Mosekilde, E., and Cauter, E. V., "Computer Model for Mechanisms Underlying Ultradian Oscillations of Insulin and Glucose," *American Journal of Physiology*, vol. 260, no. Endocrinol. Metab. 23, pp. E801–E809, 1991.

[70] Randers, J., *Elements of the System Dynamics Method.* Cambridge, MA: Productivity Press, 1980.

[71] Roberts, N., Andersen, D. F., Deal, R. M., Grant, M. S., and Shaffer, W. A., *Introduction to Computer Simulation: A System Dynamics Modeling Approach.* Reading, MA: Addison-Wesley, 1983.

[72] Senge, P. M., *The Fifth Discipline: The Art and Practice of the Learning Organization.* New York: Doubleday Currency, 1990.

[73] Gustafsson, L. and Wiechowski, M., "Coupling DYNAMO and optimization software," *System Dynamics Review*, vol. 2, no. 1, pp. 62–66, 1986.

[74] Sterman, J. D., *Business Dynamics: Systems Thinking and Modeling for a Complex World.* McGraw-Hill, 2000.

[75] Sterman, J. D., *Instructor's Manual for Business Dynamics: Systems Thinking and Modeling for a Complex World.* McGraw-Hill, 2001.

[76] Bell, J. A. and Bell, J. F., "System Dynamics and Scientific Method," in *Elements of the System Dynamics Method* (Randers, J., ed.), pp. 3–22, Cambridge, MA: MIT Press, 1980.

[77] Whewell, W., *The Philosophy of the Inductive Sciences.* New York: Johnson Reprint Corp., 2nd ed., 1847.

[78] Whewell, W., *History of the Inductive Sciences.* New York: P. Appleton Co., 1858.

[79] Kuhn, T., *The Structure of Scientific Revolutions*. Chicago: The University of Chicago Press, 2nd ed., 1970.

[80] Forrester, J. W., "14 "obvious truths"," *System Dynamics Review*, vol. 3, no. 2, pp. 156–159, 1987.

[81] Sterman, J. D. "Flight Simulators for Management Education," [online]. Available from: `http://web.mit.edu/jsterman/www/SDG/beergame.html`. Accessed July 12, 2005.

[82] for Supply Chain Innovation, T. M. F. "The MiT Beer Game," [online]. Available from: `http://beergame.mit.edu/default.htm`. Accessed July 12, 2005.

[83] Meadows, D. L., Fiddaman, T., and Shannon, D., *Fish Banks, LTD. - Game Administrator's Manual*. Hartland, VT: Sustainability Institute, 2004.

[84] Meadows, D. L., Fiddaman, T., and Shannon, D., *Fish Banks, LTD. - Materials Manual*. Hartland, VT: Sustainability Institute, 2004.

[85] Richardson, G. P. and Pugh, A. L., *Introduction to System Dynamics Modeling with Dynamo*. Cambridge, MA: MIT Press, 1981.

[86] Eubanks, K. and Yeager, L., "An Introduction to Jitia: Simulation Software Designed for Developing Large System Models," in *Proceedings of the 2001 System Dynamics Conference*, 2001.

[87] Ratnatunga, A. K. and Stewart, C. J., "Dysmap user's manual." University of Bradford, UK: System Dynamics Research Group, 1977.

[88] Hannon, B. M. and Matthias, R., *Dynamic Modeling*. New York: Springer Verlag, 1994.

[89] "Vensim: Ventana simulation environment." Vensim Version 5 User's Guide, Ventana Systems Inc.

[90] "Anylogic user's manual." XJ Technologies Company Ltd., 2005.

[91] Muetzelfeldt, R. and Massheder, J., "The Simile visual modelling environment," *European Journal of Agronomy*, vol. 18, pp. 345–358, 2003.

[92] "The agroforestry modeling environment." In Agroforestry Modelling and Research Coordination, Annual Report 1996-97, ODA Forestry Research Programme, Project R5652., NERC/ITE Edinburgh, UK, 1997.

[93] "Matlab: The language of technical computing." MATLAB Version 7, The MATHWORKS Inc.

[94] "Simulink: Simulation and model-based design." Simulink Version 6, The MATHWORKS Inc.

[95] "Modelica - a unified object-oriented language for physical systems modeling." Language Specification Version 2.2, The Modelica Association.

[96] "Analytica user guide." Analytica Version 3.1, Lumina Decision Systems Inc.

[97] "Pacelab." PACE America Inc.

[98] Brechtel, D. L., *Design and Analysis of a Simulation Model of the Resource-Acquisition Process for Government Contractors.* PhD Thesis, The Florida State University, December 1981.

[99] Bakkila, M. V., "A System Dynamics Analysis of the Interaction Between the U.S. Government and the Defense Aerospace Industry," Master's Thesis, Massachusetts Institute of Technology, June 1996.

[100] Forbes, J. L., *Externally driven growth: A system dynamics investigation.* PhD Thesis, The Claremont University, Dec 1993.

[101] Galvin, J. J. J., *Air traffic control resource management strategies and the small aircraft transportation system: A system dynamics perspective.* PhD Thesis, Virginia Polytechnic Institute and State University, Feb 2005.

[102] James M. Lyneis, Kenneth G. Cooper, S. A. E., "Strategic management of complex projects: a case study using system dynamics," *System Dynamics Review*, vol. 17, no. 3, pp. 237–260, 2001.

[103] Lyneis, J. M., "System dynamics for business strategy: a phased approach," *System Dynamics Review*, vol. 15, no. 1, pp. 37–70, 1999.

[104] Lyneis, J. M., "System dynamics for market forecasting and structural analysis," *System Dynamics Review*, vol. 16, no. 1, pp. 3–25, 2000.

[105] Mass, N. J. and Senge, P. M., "Alternative tests for selecting model variables," in *Elements of the System Dynamics Method* (Randers, J., ed.), pp. 203–221, Cambridge, MA: MIT Press, 1980.

[106] O'Connor, W. E., *An introduction to airline economics.* Westport, CT, Praeger Publishers, 6th ed., 2001.

[107] James, G. W., ed., *Airline Economics.* Lexington, MA: Lexington Books, 1982.

[108] Porter, M. E., *Competitive Strategy.* New York: Free Press, 1980.

[109] D'Aveni, R., *Hypercompetition: Managing the dynamics of strategic manuevering.* New York: The Free Press, 1994.

[110] Marshall, A., *Principles of Economics.* London: Macmillan, 1891.

[111] Arthur, W., *Increasing Returns and Path Dependence in the Economy.* Ann Arbor, MI: University of Michigan Press, 1994.

[112] Krugman, P. R., "Increasing returns, monopolistic competition, and international trade," *Journal of International Economics*, vol. 9, no. 4, pp. 469–479, 1979.

[113] Romer, P., "Endogenous technological change," *Journal of Political Economy*, vol. 5, pp. Part 2, S71–S102, 1990.

[114] Prigorgine, I., *From Being to Becoming*. San Francisco: Freeman, 1980.

[115] Prigorgine, I. and Stengers, I., *Order Out of Chaos*. New York: Bantam, 1984.

[116] Matthews, R. "Path Dependence and Transition Strategies," [online]. Available from: `http://business.kingston.ac.uk/cibp.php?item=3`. Centre for International Business Policy, Kingston University London, Accessed August 22 2005.

[117] Wielage, M. and Woodcock, R., "The Rise and Fall of Beta," *Videofax*, no. 5, 1988.

[118] Anderson, E., "VCR battle of the formats model." Memo, MIT Sloan School of Management, Cambridge, MA, 1996 as referenced in [74].

[119] Wright, T. P., "Factors Affecting the Cost of Airplanes," *Journal of Aeronautical Science*, vol. 3, no. 2, 1936.

[120] Novick, D., *Weapon System Cost Analysis*. Santa Monica, CA: The Rand Corporation, 1956.

[121] Mattingly, J. D., Heiser, W. H., and Pratt, D. T., *Aircraft Engine Design*. American Institute of Aeronautics and Astronautics, 2002.

[122] Anderson, J. D., *Aircraft Performance and Design*. McGraw-Hill, 1999.

[123] Hale, F. J., *Introduction to Aircraft Performance, Selection, and Design*. John Wiley & Sons, 1984.

[124] Heath, M. T., *Scientific Computing: An Introductory Survey*. New York: McGraw-Hill, 2nd ed., 2002.

[125] Backus, J. and et al., *The FORTRAN Automatic Coding System for the IBM 704 EDPM*. New York: International Business Machines Corporation, 1957.

[126] "ModelCenter 7.0 Data Sheet," [online]. Available from: `http://www.phoenix-int.com`. Accessed September 23 2006.

[127] "iSIGHT Product Overview," [online]. Available from: `http://www.engineous.com/product_iSIGHT.htm`. Accessed September 23 2006.

[128] Hwang, C.-L. and Masud, A. S. M., "Multiple Objective Decision Making: Methods and Applications," in *Lecture Notes in Economics and Mathematical Systems*, (Berlin, Germany), Springer Verlag, 1979.

[129] Yoon, K. P. and Hwang, C.-L., "Multiple Attribute Decision Making: An Introduction," in *Quantitative Applications in the Social Sciences*, (Thousand Oaks, California), Sage Publications, 1995.

[130] Triantaphyllou, E., *Multi-Criteria Decision Making Methods: A Comparative Study*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 2000.

[131] Zeleny, M., *Multiple Criteria Decision Making*. New York: McGraw-Hill, 1982.

[132] Hwang, C.-L. and Yoon, K., *Multiple Attribute Decision Making: Methods and Applications: A State-of-the-Art Survey*. Berlin, Germany: Springer Verlag, 1981.

[133] Kleijnen, J. P. C., "Sensitivity analysis and optimization of system dynamics models: Regression analysis and statistical design of experiments." Center for Economic Research (CentER) and Department of Information Systems, Tilburg University, Tilburg, Netherlands, 1995.

[134] Voss, R. F. and Clarke, J., ""1/f noise" in music: Music from 1/f noise," *Journal of the Acoustical Society of America*, vol. 63, no. 1, pp. 258–163, 1978.

[135] Schwartz, E. S., "The Stochastic Behavior of Commodity Prices: Implications for Valuation and Hedging," *The Journal of Finance*, vol. 52, no. 3, pp. 922–973, 1997.

[136] Roach, J. and Eastwood, A. B., *Jet Airliner Production List Volume 2*. TAHS, 2004.

[137] "Boeing Commercial Aircraft Orders & Deliveries," [online]. Available from: `http://active.boeing.com/commercial/orders/index.cfm`. Accessed August 28 2005.

[138] Jackson, P., ed., *Jane's All the World's Aircraft*. Surrey, UK: Jane's Information Group Limited, 2005-2006.

[139] *A330 Airplane Characteristics For Airport Planning AC*. Airbus S.A.S., 2002.

[140] *767 Airplane Characteristics for Airport Planning*. No. D6-58328, Boeing Commercial Airplanes, September 2005.

[141] U.S. Department of Energy, E. I. A. "Daily Spot Prices of Kerosene-Type Jet Fuel," [online]. Available from: `http://www.eia.doe.gov`. Accessed May 5 2006.

[142] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd ed., 1992.

[143] "JFreeChart," [online]. Available from: `http://www.jfree.org/jfreechart/index.html`. Accessed June 6 2006.

[144] “The Eclipse Integrated Development Environment,” [online]. Available from: `http://www.eclipse.org`. Accessed May 5 2006.

[145] Lowagie, B. “iText, a Free Java-PDF Library,” [online]. Available from: `http://www.lowagie.com/iText`. Accessed June 6 2006.

[146] *PDF Reference, Adobe Portable Document Format Version 1.6*. Adobe Systems Incorporated, 5th ed., 2006.

[147] Mavris, D. N., Biltgen, P., Ender, T., and Cole, B., “Technology Assessment and Capability Tradeoff Using Architecture-Based Systems Engineering Methodologies,” in *CEIAT 2005-0048, 1st International Conference on Innovation and Integration in Aerospace Sciences, Queens University Belfast, Northern Ireland, UK*, August 4-5, 2005.

[148] Biltgen, P. T., Ender, T., and Mavris, D. N., “Development of a Colaborative Capability-Based Tradeoff Environment for Complex System Architectures,” in *AIAA-2006-0728, 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV*, January 9-12, 2006.

VITA

Jens Holger Pfaender was born in Nuernberg, Bavaria, Germany. He resided in his home town Langenzenn just outside of Nuernberg for his first 19 years. He then served in the Luftwaffe as a Patriot radar recorder. He then attended Embry-Riddle Aeronautical University in Daytona Beach, Florida starting in the fall of 1997, where he received his Bachelor's Degree in Aerospace Engineering in April of 2001. He then enrolled at Georgia Tech for graduate school and was awarded a Master's Degree in Aerospace Engineering in August 2003. He passed the Ph.D. qualifying exams in April of 2003 and is on track to be awarded his Ph.D. in Aerospace Engineering in December 2006.

Holger enjoys all things related to aviation, electronics and mechanics. On occasion he enjoys being an amateur rocketeer focusing mainly on small scale hybrids.