

Angewandte Mathematik und Optimierung Schriftenreihe  
Applied Mathematics and Optimization Series  
AMOS # 65(2017)

Colin Wenzel

Simulierter Luftkampf – Ein umfangreiches  
mathematisches Modell zur Vorhersage von  
Gefechtsausgängen basierend auf dem  
LAMBDA-Tool

Herausgegeben von der  
Professur für Angewandte Mathematik  
Professor Dr. rer. nat. Armin Fügenschuh

Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg  
Fachbereich Maschinenbau  
Holstenhofweg 85  
D-22043 Hamburg

Telefon: +49 (0)40 6541 3540  
Fax: +49 (0)40 6541 3672

e-mail: [appliedmath@hsu-hh.de](mailto:appliedmath@hsu-hh.de)  
URL: <http://www.hsu-hh.de/am>

Angewandte Mathematik und Optimierung Schriftenreihe (AMOS), ISSN-Print 2199-1928  
Angewandte Mathematik und Optimierung Schriftenreihe (AMOS), ISSN-Internet 2199-1936



HELMUT SCHMIDT  
UNIVERSITÄT

Universität der Bundeswehr Hamburg

**Colin Wenzel**

Simulierter Luftkampf - Ein umfangreiches mathematisches Modell zur Vorhersage von Gefechtsausgängen basierend auf dem LAMBDA-Tool

Simulated aerial combat - An extensive mathematical model for the prediction of combat outcomes based on the LAMBDA-Tool

Master-Arbeit

Fakultät für Elektrotechnik

Studiengang:      Wirtschaftsingenieurwesen  
Matr.-Nr.           868830  
Übernahme:       22. Mai 2017  
Betreuer:           Prof. Dr. Armin Fügenschuh  
Weiterer Prüfer: Prof. Dr. Ulrich Tüshaus

## Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbständig verfasst, keine anderen als die im Quellen- und Literaturverzeichnis genannten Quellen und Hilfsmittel, insbesondere keine dort nicht genannten Internet-Quellen benutzt, alle aus Quellen und Literatur wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe und dass die auf einem elektronischen Speichermedium abgegebene Fassung der Arbeit der gedruckten entspricht.

Hamburg,

.....

(Datum)

.....

(Unterschrift)

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
<b>2</b>	<b>Grundlagen</b>	<b>4</b>
2.1	Dynamische Systeme . . . . .	4
2.2	Gesetz von Lanchester . . . . .	6
2.3	LAMBDA-Tool . . . . .	8
2.4	Excel als Simulationstool . . . . .	9
2.5	Vensim als Simulationstool . . . . .	10
2.6	AIMMS als Optimierungstool . . . . .	10
<b>3</b>	<b>LAMBDA Modell</b>	<b>14</b>
3.1	Sets . . . . .	14
3.2	Parameter . . . . .	15
3.2.1	Unabhängige Parameter . . . . .	15
3.2.2	Abhängige Parameter . . . . .	17
3.3	Variablen . . . . .	20
3.3.1	Stützpunktvariablen . . . . .	21
3.3.2	Schwadronenvariablen . . . . .	22
3.3.3	Fraktionsvariablen . . . . .	23
3.4	Erstellen eines Gesamtmodells . . . . .	28
3.4.1	Differentialgleichung für den Status der Stützpunkte . . . . .	28
3.4.2	Differentialgleichung für die Anzahl der Flugzeuge . . . . .	30
3.4.3	Differentialgleichungen für die Anzahl der Boden-Luft-Raketen . . . . .	31
3.4.4	Hilfsvariablen . . . . .	33
3.4.5	Gesamtmodell . . . . .	36
3.5	Analyse des Modells . . . . .	40
<b>4</b>	<b>LAMBDA Modell in Vensim</b>	<b>42</b>
<b>5</b>	<b>Optimierung von Gefechten in AIMMS</b>	<b>46</b>
<b>6</b>	<b>Schlussfolgerungen</b>	<b>49</b>

# Abbildungsverzeichnis

2.1	Systembeispiel Schule . . . . .	5
4.1	Ausschnitt des Vensimmodells. . . . .	42
4.2	Vergleich der Simulationsergebnisse von LAMBDA-Tool und Vensim-Modell. . . . .	43
5.1	Ausschnitt aus AIMMS. . . . .	48

# Abkürzungsverzeichnis

AC	Air Capabilities
AEW	Airborne Early Warning
AI	Air Interdiction
CAS	Close Air Support
CP	Constraint Program
DCA	Defensive Counter-Air
ESC	Escort
ESM	Electronic Support Measures
FU	Fire Unit
KOG	Killed On Ground
LAMBDA	Land-Air-Maritime Battle Determination Algorithms
LP	Linear Program
MCP	Mixed Complementarity Program
MINLP	Mixed Integer NonLinear Program
MIP	Mixed Integer Program
MIQCP	Mixed Integer Quadratically Constrained Program
MIQP	Mixed Integer Quadratic Program
MPCC	Mathematical Program with Complementarity Constraints
NLP	NonLinear Program
OCA	Offensive Counter-Air
QCP	Quadratically Constrained Program
QP	Quadratic Program
SAMS	Surface-to-Air Missile
SEAD	Suppression of Enemy Air Defence
SSM	Surface-to-Surface Missile
TLAM	Tomahawk Land Attack Missile

# 1 Einleitung

Die optimale Strategie für eine Schlacht zu finden, ist schon seit jeher von größtem militärischen Interesse. 1916 lieferte Frederick W. Lanchester erstmals eine mathematische Herangehensweise für diese Thematik. Betrachtet wurde dabei eine Gefechtsituation, bei der sich zwei Seiten gegenüberstehen. In diesem Modell wurden lediglich die Anzahl der Kräfte und deren Feuerkraft miteinbezogen [7]. Das Gesetz wurde von vielen Seiten weiterentwickelt, wie zum Beispiel das Erarbeiten einer Wahrscheinlichkeitsverteilung für die Feuerkraft durch Bonder [3].

Das von Brian Witherden und Christopher Brown für die NATO entwickelte LAMBDA-Tool macht sich das Gesetz von Lanchester zunutze, um auf dessen Grundlage ein weiterentwickeltes Modell zur Berechnung von Gefechtsverläufen darzustellen. Die Abkürzung LAMBDA steht für „Land-Air-Maritime Battle Determination Algorithms“. Ziel dieser Arbeit ist es, mögliche Verbesserungen für die Anwendung des LAMBDA-Tools zu finden. Ein besonderer Fokus liegt dabei auf LAMBDA\_AC. Dabei handelt es sich um eine Abwandlung, die speziell für die Simulation von Luftkämpfen genutzt wird. Das Tool besteht aus einer mehrseitigen Excel-Tabelle und lässt alle Berechnungen im Hintergrund laufen. Hier soll das zugrundeliegende mathematische Modell (im Folgenden als LAMBDA-Modell bezeichnet) extrahiert werden. Dies sollte es ermöglichen, einzelne Zusammenhänge zwischen den Elementen des Systems sichtbar zu machen und auf ein womöglich besser geeignetes Simulationstool wie Vensim zu übertragen. Dies ist insbesondere interessant, da der Nutzen des Tools zuvor darauf limitiert war, mögliche Gefechtsstrategien als Input einzugeben und als Output eine Prognose des Gefechtsausgangs zu erhalten. Ein extrahiertes Modell hingegen erlaubt es, auch Optimierungsrechnungen durchzuführen, um eine Empfehlung für eine erfolgreichere Gefechtsstrategie auszusprechen. Die Allokation der eigenen Kräfte ist dabei der wichtigste Ansatzpunkt. Eine solche Optimierung von Gefechtsstrategien auf Grundlage anderer Modelle war schon Objekt weiterer Arbeiten. Auf die maritime Kriegsführung bezogen, arbeitet Dionne an der Optimierung der Zielzuweisung für Waffensysteme. Gelöst wird das Problem dabei durch sequentielle Optimierung [6]. Auf Grundlage der dynamischen Spieltheorie kann Cruz die Entscheidungen für Truppeneinsätze mittels der Moving-Horizon-Methode optimieren. Die Lösung ist dabei zwar kein globales Optimum, dafür passt sich die Gefechtsstrategie immer wieder dem Schlachtverlauf an [5]. Ebenfalls auf den Luftkampf bezogen, wird bei Popken die Wahrscheinlichkeitsverteilung von möglichen Konsequenzen der eigenen Handlungen untersucht, um durch die stochastische Evaluation dieser Ausgänge eine optimale Entscheidung zu treffen [8]. Deutliche Parallelen zum Optimierungsobjekt dieser Arbeit sind bei Taylor und Berkovitz zu sehen. Berkovitz bezieht sich auch auf den Luftkampf und untersucht nach Prinzipien der Spieltheorie, auf welche Mission die eigenen Truppen optimalerweise entsendet werden sollten, um die jeweilige Aktion des Gegners zu kontern [2]. Taylor wiederum betrachtet den Aspekt der gene-

rellen Kriegsführung auf Basis des Gesetzes von Lanchester. Dabei wird eine Strategie entwickelt, wie das Feuer der eigenen Truppen optimal auf den Gegner verteilt werden sollte [9].

Diese Arbeit liefert erste Ansätze für Verbesserungen des LAMBDA-Tools und zeigt, dass es mit einem Programm wie AIMMS möglich ist, eine optimierte Gefechtsstrategie zu erstellen.

## 2 Grundlagen

In diesem Kapitel werden die Grundlagen, welche für diese Arbeit Voraussetzung sind, erläutert. Sowohl mathematische Methoden, als auch die für diese Arbeit verwendeten Programme werden beleuchtet.

### 2.1 Dynamische Systeme

Dynamische Systeme finden sich überall in unserer Umwelt. Die Schule, auf die wir gegangen sind, unsere Computer oder auch wir Menschen selbst gehören dazu. Doch was macht ein dynamisches System aus und welche Relevanz hat es für uns? Weithin werden einem System drei Merkmale zugesprochen: Der Systemzweck, die Systemstruktur und die Systemintegrität. Besitzt ein Objekt eines dieser drei Merkmale nicht, so ist es nicht als System zu bezeichnen.

Der Systemzweck impliziert, dass ein betrachtetes Objekt eine bestimmte Funktion inne hat, die es zu erfüllen gilt. Beim Beispiel der Schule wäre dies die Ausbildung der Schüler sein. Damit ein Objekt eine Systemstruktur hat, muss es aus Elementen bestehen, die über direkte Wirkungsverknüpfungen miteinander verbunden sind. Zum Beispiel sind Lehrer und Schüler über das Lehrer-Schüler-Verhältnis verknüpft, welches Auswirkung auf den Ausbildungsstand der Schüler hat. Das Merkmal Systemintegrität wird benutzt, um auszudrücken, dass Elemente in dem System existieren, die immanent für die Identität des Systems sind. Das bedeutet, dass durch Verlust dieser Elemente auch der Zweck oder die Funktion des Systems verloren geht. Die Schüler sind zum Beispiel so ein essentielles Element. Ohne Schüler wäre eine Schule keine Schule. Ein Schulhof, welcher im System Auswirkungen auf die Pausenqualität und damit die Konzentration der Schüler hat, wäre beispielsweise kein essentielles Element, da die Schule ohne ihn immer noch eine Schule ist. Wichtig ist, dass in einem System überhaupt ein essentielles Element besteht.

Nach Aufzählung dieser Kriterien fällt auf, auf wie viele Bereiche diese anwendbar und wie weit verbreitet dynamische Systeme sind. Daran lässt sich auch erkennen, welche Bedeutung diese für uns haben und warum es wichtig ist, sie zu verstehen. Dadurch, dass wir die uns umgebenden Systeme verstehen, können wir sie auch verbessern oder zumindest besser nutzen. Der Ausbildungsstand des Systems Schule hat z.B. einen enormen Einfluss auf die Entwicklung und Prosperität unserer Gesellschaft. Die Bedeutung des Verstehens von Systemen steht somit außer Frage, aber wie kann dies erreicht werden? Hier kommt die Modellbildung und die Simulation ins Spiel. Die Modellbildung geht der Simulation voran und stellt die Grundlage für diese dar. Ein Modell ist immer nur eine mehr oder weniger präzise Abbildung der Wirklichkeit. Dabei wird immer nur ein Teil der Elemente der realen Vorlage mit einbezogen, da die Wirklichkeit meist zu komplex ist, um sie genau abzubilden oder zumindest der Arbeitsaufwand dafür unver-

hältnismäßig hoch wäre.

Um ein Modellsystem zu erstellen, muss man sich zuerst der Systemgrenze bewusst werden. Was soll tatsächlich betrachtet werden? Hierbei muss berücksichtigt werden, dass Modellsysteme sich unter Einbeziehung der Systemumwelt der ihnen zugrunde liegenden Vorbilder, zumindest theoretisch, unendlich erweitern lassen. So besteht das System Schule in Bezug zur städtischen Umwelt sowie in Bezug zu seinem Bundesland. Dementsprechend ist zuerst zu klären, welche Aspekte für den Ersteller des Modells von Bedeutung sind. Möchte er nur den Einfluss der Lehrer auf die Schulnoten untersuchen, so sollte das System klein gehalten werden. Soll aber ausführlich der Einfluss von Bruttoinlandsprodukt und Infrastruktur untersucht werden, reicht das System Schule nicht aus. Es können die Einflüsse der Umwelt auf ein System in Form von Systemeintritten mit einbezogen werden, bedingt durch die Systemgrenze jedoch nicht mehr, wie diese zustande kommen.

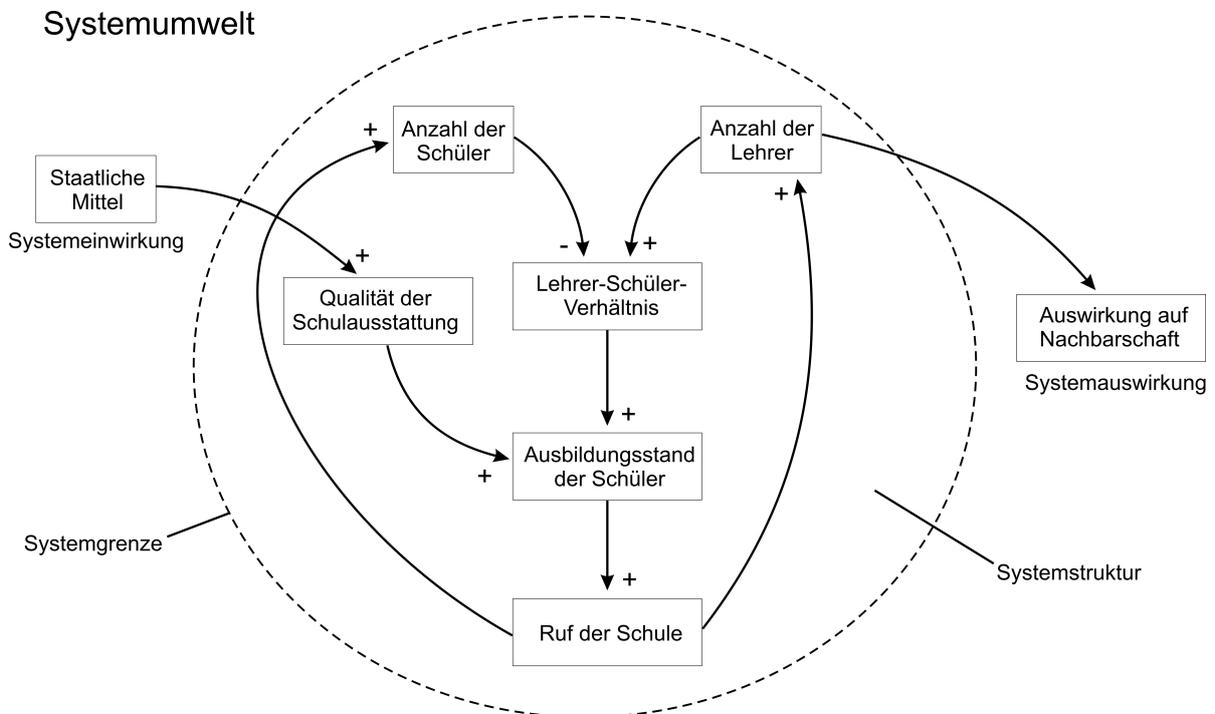


Abbildung 2.1: Systembeispiel Schule

Nach dem Aufstellen der Systemgrenze müssen die einzelnen Elemente bestimmt werden. Hier gilt, je genauer das Modell ausfallen soll, desto mehr Elemente sollten mit einbezogen werden. Mehr Elemente zeigen später mehr Wechselwirkungen auf und haben dank ihrer höheren Genauigkeit auch meist einen größeren Nutzen in der Realität. Weniger Elemente hingegen verringern die Komplexität und erhöhen das Verständnis.

Des Weiteren wird dadurch der Rechenaufwand verringert, bzw. eine Berechnung überhaupt erst ermöglicht. Viele Modelle sind bis dato allein schon durch die zur Verfügung stehende Rechenleistung limitiert.

Als nächstes bleibt noch die Beschreibung der Wirkungszusammenhänge. Hierbei ist es wichtig, nur die direkten Wirkungen in das Modell mit aufzunehmen. Die indirekten Wirkungen ergeben sich, wenn genau gearbeitet wird, aus den Interaktionen der direkten Wirkungen. Wie schon vorher im Modell erwähnt, würden mehr Lehrer das Lehrer-Schüler-Verhältnis verbessern, was wiederum den Ausbildungsstand verbessern würde. Hierbei gibt es zwei direkte Verbindungen zwischen den Elementen, welche mit aufgenommen werden müssen. Durch Betrachten des Systems Schule kann man folgerichtig zu dem Schluss gelangen, dass sich mehr Lehrer positiv auf den Ausbildungsstand auswirken. Diese Verbindung darf aber nicht mit aufgenommen werden, da sie indirekt ist und zu doppelter Einflussnahme führen würde. Indirekte Wirkungen können zu direkten gemachten werden und somit aufgenommen werden, wenn die Zwischenelemente eliminiert werden. Das wäre in diesem Falle aber nicht sinnvoll, da die Schüler auch Einfluss auf das Lehrer-Schüler-Verhältnis nehmen.

Sind all diese Punkte bestimmt, so kann das System beispielsweise wie in Abbildung 2.1 dargestellt werden. Die eben angesprochenen Begriffe sind darin noch einmal verdeutlicht und um ein paar einfache Wirkmechanismen ergänzt. Ist das Modell aufgestellt, ist der nächste Schritt die Simulation eben dieses Modellsystems. Die Simulation eines Systems, sprich das theoretische Durchspielen von Ereignissen, wird heutzutage hauptsächlich mit Computern durchgeführt. Bei einfachen Systemen, wie im obigen Beispiel, ist es als Mensch noch möglich, alle beteiligten Elemente, ihre Interaktionen und die daraus folgenden Konsequenzen in Betracht zu ziehen. Bei komplexeren Modellen hingegen ist ein Computer nicht nur besser geeignet, die hohe Anzahl an Elementen verlässlich miteinzubeziehen, er ermöglicht auch eine direkte Visualisierung der Ergebnisse [4].

Um die Veränderungen der Zustände und die Stärke der Wechselwirkungen in einem System möglichst genau darzustellen, müssen dem Computer auch noch die nötigen Formeln zur Verfügung gestellt werden. Auf Formeln, welche für das in dieser Arbeit behandelte System von besonderer Bedeutung sind, wird im nächsten Kapitel gesondert eingegangen.

## 2.2 Gesetz von Lanchester

Das in dieser Arbeit untersuchte System ist eine Gefechtssituation, in welcher sich zwei Seiten gegenüberstehen und bekämpfen. Um im Vorhinein herauszufinden, wie das Gefecht verläuft, sprich, es zu simulieren, muss genau ausgerechnet werden, in welcher Weise sich der gegnerische Beschuss auf die eigenen Truppen und der eigene Beschuss auf die gegnerischen Truppen auswirkt. Einen Grundstein hierfür hat Frederick W. Lanchester bereits während des Ersten Weltkriegs mit der Aufstellung des nach ihm benannten Gesetz von Lanchester gelegt. Hierbei wird von drei Grundannahmen ausgegangen:

- Es befinden sich nur zwei Parteien im Gefecht und die Truppenstärke ist von beiden Seiten bekannt.

- Beide Seiten feuern gleichzeitig.
- Beide Seiten besitzen eine gleichbleibende Feuerkraft.

Das von Lanchester betrachtete Gefechtsmodell war zwar wesentlich simpler als das System des LAMBDA-Tools, dennoch findet das Gesetz auch dort in abgewandelter Form Anwendung. Ausformuliert sieht das Modell dann wie folgt aus:

**Gesetz von Lanchester:**

$$\frac{dA}{dt} = -\beta B \quad \text{und} \quad \frac{dB}{dt} = -\alpha A. \quad (2.1)$$

Symbole:

- $t$ : Die Zeit in Tagen.
- $A$ : Truppenstärke der Partei A.
- $B$ : Truppenstärke der Partei B.
- $\alpha$ : Feuerkraft von A (Wie viele Soldaten von B pro Soldat von A an einem Tag eliminiert werden.)
- $\beta$ : Feuerkraft von B (Wie viele Soldaten von A pro Soldat von B an einem Tag eliminiert werden.)

Um daraus eine Aussage über die Bedeutung von Truppenstärke und Feuerkraft machen zu können, müssen noch diverse Umformungen vorgenommen werden. Eine Division der beiden Gleichungen, gefolgt von einer Integration, führt zu diesem Ergebnis:

$$\frac{dA}{dB} = \frac{\beta B}{\alpha A} \Rightarrow \int \alpha A \, dA = \int \beta B \, dB \Rightarrow \alpha A^2 = \beta B^2 + c. \quad (2.2)$$

Nun ist ersichtlich, dass wenn A seine Truppenstärke erhöht, B die Möglichkeit hat, seine Feuerkraft oder seine Truppenstärke zu erhöhen, um gleichzuziehen. Allerdings fällt auf, dass sich die Truppenstärke quadratisch auswirkt, während die Feuerkraft nur einen linearen Anteil hat. Lanchester hat daraus geschlossen, dass es viel wichtiger ist, dem Gegner zahlenmäßig überlegen zu sein, als z.B. bessere Waffentechnik zu besitzen[7]. Diese Annahme hat den Militärstrategen in der Vergangenheit gute Dienste geleistet, gilt inzwischen jedoch als veraltet. In der heutigen Zeit stehen sich kaum noch ebenbürtige Truppen gegenüber, die einen Grabenkrieg führen, für den dieses Modell ausgelegt ist. Vielmehr ist heute die asymmetrische Kriegsführung in den Fokus gerückt. Auch ist es schnell einleuchtend, warum neue Waffentechnik, wie zum Beispiel Kampfflugzeuge oder hochmoderne Kriegsschiffe einer größeren Truppenstärke ohne solche Technik überlegen ist. Für eine akkurate Modellierung sind folglich eine Vielzahl an Parametern nötig, um die genannten Asymmetrien abzubilden. Dem LAMBDA-Tool, zum Beispiel, liegt eine solche Komplexität zugrunde.

## 2.3 LAMBDA-Tool

Dieses Kapitel erklärt, was der hohen Komplexität des Tools zugrunde liegt. Bei der Beschreibung dieses Tools und seiner Parameter werden im weiteren Verlauf auch die originalen Bezeichnungen auf Englisch verwendet, um die Nachvollziehbarkeit für jemanden mit Zugriff auf das LAMBDA-Tool zu erhöhen. Im Gegensatz zum Lanchester-Modell werden im LAMBDA-Tool schon vier Streitkräfte betrachtet, welche alle eigene Truppen stellen. Die Namen hierbei sind beliebig wählbar. In diesem Beispiel ist bei den Fraktionen aber von OPFOR und der Alliance die Rede, wobei sich OPFOR noch in Kamon und Stellaria und die Alliance in Friends und Tytan unterteilt. Die vier Parteien werden zwar zu zwei Fraktionen zusammengefasst und entsprechen so wieder dem klassischen Modell, aber jede Streitkraft hat seine eigene Feuerkraft, da die genutzten Parameter individuell sind.

Eine genaue Aufzählung der Einflussfaktoren findet sich in Kapitel 3.2. An dieser Stelle soll aber schon einmal eine Übersicht gegeben werden. Die Parameter umfassen unter anderem auch Effekte des Schlachtfeldes, wie z.B. das Wetter, welches über einen Faktor die Abflugrate der Flugzeuge verringert. Hauptsächlich beinhalten die Parameter aber Faktoren, welche die Effektivität und Stärke der Truppen beschreiben. So gibt es eine Tabelle, die die Parameter der verschiedenen einsetzbaren Flugzeugtypen beinhaltet. Dort wird festgehalten, wie effektiv jedes Flugzeug auf verschiedenen Missionen eingesetzt werden kann und wie wahrscheinlich sein Überleben dabei ist. Auch die Stützpunkte der Fraktionen sind verschieden. Als Parameter werden dabei zum Beispiel die Größe der Stützpunkte oder die Anzahl der Schutzbunker verwendet. Beim Einsatz von Boden-Boden- und Boden-Luft-Raketen wird hauptsächlich unterschieden, wie viele jede Seite davon täglich abfeuern kann.

Den Kern des Tools aber bildet die Allokation der eigenen Kräfte. Die Einheiten, welche in den Kampf geschickt werden, werden Schwadronen (squadrons) genannt. Jede Seite kann seine vorhandenen Flugzeuge beliebig in diese Schwadronen einteilen. Dabei kann ein Schwadron aber nur aus einem Flugzeugtyp bestehen. Zusätzlich zu den Flugzeugen wird jedem Schwadron auch ein Stützpunkt und täglich eine Mission zugeteilt. Ein Schwadron kann einen von acht Missionstypen fliegen. Die offensiven Flüge (OCA) richten hierbei Schaden an den gegnerischen Stützpunkten an. Mit defensiven Flügen (DCA) können gegnerische Offensiven abgefangen und zerstört werden. Die Defensive kann zusätzlich durch luftgestützte Früherkennung (AEW) verbessert werden. Werden allerdings noch Eskorten (ESC) neben den offensiv fliegenden Einheiten mit auf den Weg geschickt, so dezimieren diese wiederum die gegnerische Verteidigung. Des Weiteren kann die Wirkung der Verteidigung durch elektronische Unterstützungsmaßnahmen (ESM) und durch die Abriegelung des Luftraums (AI) verringert werden. Der Einsatz von Boden-Luft-Raketen (SAM) ist ebenfalls defensiv und richtet sich somit gegen die offensiv fliegenden Einheiten. Aber auch dafür gibt es ein Gegenmittel, nämlich die Unterdrückung der feindlichen Luftabwehr (SEAD). An dieser Stelle fällt auf, dass nur sieben von acht Missionstypen erwähnt wurden. Darüber, warum die Luftnahunterstützung (CAS) nicht beschrieben wurde, findet sich mehr in Kapitel 3.5.

Den Daten, welche für die Parameter genutzt werden, liegen unterschiedliche Genauig-

keiten zugrunde. So ist die Anzahl der eigenen Truppen oder die Mission, für welche sie eingesetzt werden, sicher bekannt. Anders verhält es sich bei Parametern über die Effektivität oder die Überlebensfähigkeit. Vor allem das Wetter und seine genauen Auswirkungen sind kaum über mehrere Wochen hinweg vorherzusagen. Folglich muss sich auf Erfahrungs- und Durchschnittswerte verlassen werden. Das LAMBDA-Modell ist hier genauso limitiert wie die meisten Modelle. Auch mit dem momentanen Stand der Technik fehlen die Möglichkeiten, messbare Zahlenwerte für viele in der Realität vorkommende Effekte zu finden.

Werden die Parameter in den Berechnungen verwendet, so ist zu beachten, dass mit vielen theoretischen Werten gerechnet wird. Zum Beispiel wird später Standard Sorties (standardisierte Abflüge) oder SAM FU (Surface-to-Air Missile Fire Unit) verwendet werden. Bei SAM FU ist dies keine tatsächliche Anzahl von Flugkörpern oder Equipment, welches zum Abschuss benötigt wird. Es ist ein theoretischer Wert, welcher widerspiegeln soll, wie viele Objekte gleichzeitig mit einem Raketentyp anvisiert werden können. Trotzdem wird es im Modell als zerstörbares Objekt angesehen. Bei den Standard Sorties wird die Effektivität mit einbezogen und sie stellen somit eine theoretische Anzahl an Abflügen dar, wohingegen Actual Sorties beschreibt, wie viele Flugzeuge tatsächlich den Stützpunkt verlassen.

Das Gefecht wird anschließend über 25 Tage simuliert und die Berechnungen dafür werden mit Excel durchgeführt. Wie gut sich Excel für diese Aufgabe eignet, wird im nächsten Abschnitt besprochen.

## 2.4 Excel als Simulationstool

Bei Excel handelt es sich um ein Tabellenkalkulationsprogramm, welches von Microsoft entwickelt wurde. Daten, die in Tabellenform vorliegen, lassen sich dort einfach verarbeiten und Abhängigkeiten zwischen Werten können leicht erstellt werden. Mehrere Werte können auch durch Formeln in einer weiteren Zellen zusammengeführt werden. Diese Funktionen eignen sich hervorragend für statistische oder kaufmännische Zwecke.

Wird es für Simulationen genutzt, bietet Excel den Vorteil, dass sich In- und Output, beispielsweise in Form verschiedener Diagrammen, übersichtlich darstellen lassen. Dies ist auch bei großen Datenmengen möglich. Die Möglichkeit, Zwischenwerte einfach als Zelleninhalt auszugeben und so zugänglich zu machen, ist ein weiterer Vorteil. Nachteilig hingegen ist die Art und Weise, wie Excel dem User Zwischen- und Endwerte ausgibt. Formeln und Verknüpfungen sind nicht direkt einsehbar, stattdessen verweist das Programm auf die jeweiligen Zellen. Um herauszufinden, wie bestimmte Werte entstanden sind oder was sie genau bedeuten, muss der User die jeweilige Ursprungszelle ausfindig machen. Abhängig von der Komplexität und Anzahl der Zwischenwerte kann sich dies über mehrere Zellen erstrecken und ist deshalb unübersichtlich und zeitintensiv. Auch bedeutet der Bezug auf Zellennummern anstatt tatsächlicher Variablen, dass das Einfügen oder Entfernen von Elementen in bestimmte Zellen die Simulation verfälschen kann. Des Weiteren sind sowohl Simulationszeitraum als auch Zeitschritte relativ statisch, sobald das Modell erstellt ist. Bei Excel werden die Formeln für jeden Tag einzeln aufgestellt.

Jeder Tag muss gesondert beschrieben werden. Somit ist die Möglichkeit, Zeiträume zu ändern, zwar gegeben, aber mit viel Aufwand verbunden.

Zusammenfassend lässt sich sagen, dass Excel nicht als Simulationstool gedacht ist, sich dafür jedoch mit kleinen Abstrichen recht gut eignet.

## 2.5 Vensim als Simulationstool

Vensim ist von Ventana Systems im Gegensatz zu Excel direkt als Simulationsprogramm entwickelt worden. Es ist für die kontinuierliche Simulation gedacht und verwendet hierbei sowohl das explizite Euler- als auch das klassische Runge-Kutta-Verfahren zur Lösung von Differentialgleichungen.

Vensim kann genau wie Excel sowohl Input als auch Output in Tabellenform darstellen. Exceltabellen mit Parametern können anschließend ins Programm importiert und genutzt werden. Auch das Verständnis des Outputs lässt sich visuell durch Grafiken unterstützen. Vorteilhaft bei Vensim ist, dass für jeden Simulationslauf eine Datei mit den Ergebnissen erstellt werden kann, so dass der Vergleich anschließend einfacher fällt. Entgegen der Verweise auf bestimmte Zellen in Excel ist es bei Vensim möglich, die Namen der Variablen direkt in den Formeln zu verwenden. Dadurch lässt sich durch Betrachten einer Variable direkt erkennen, wie sich der Wert zusammensetzt. Des Weiteren werden Einheiten- und Syntaxkontrollen unterstützt, wodurch die Fehlersuche erheblich vereinfacht wird. Der größte Unterschied zu Excel ist die Möglichkeit, durch eine Eingabe in den Modelloptionen die Anzahl und die Länge der Zeitschritte verändern zu können, ohne jeden Zeitschritt neu zu beschreiben. Dies ist möglich, da bei Vensim die Differentialgleichungen für die Zustände zu Beginn aufgestellt werden und dann kontinuierlich ausgeführt werden. Das Programm vermittelt zusätzlich einen guten Überblick über die Abläufe eines Systems, da es bei der Erstellung des Modells automatisch ein Pfeildiagramm entsteht, welches diese verdeutlicht. Dies stößt bei sehr großen Systemen an seine Grenzen, wie beim LAMBDA-Modell in Kapitel 4 zu sehen ist. Wenn viele Verbindungen zwischen allen Elementen des Systems bestehen, haben die vielen Pfeile eher eine verwirrende als verdeutlichende Wirkung. Dies kann umgangen werden, indem das System in kleinere Abschnitte zerlegt wird, was allerdings den Verlust des Überblicks über das Gesamtsystem zur Folge hat.

Vensim eignet sich daher sehr gut für Simulationszwecke und gibt einen guten Einblick in die Vorgänge des zu untersuchenden Modells.

## 2.6 AIMMS als Optimierungstool

Soll ein Teil des Systems optimiert werden, zum Beispiel die Truppenallokation, so ist eine gute Optimierungssoftware notwendig. Für diese Arbeit wird AIMMS verwendet, was eine Modellierungssprache und Entwicklungsumgebung für umfangreiche Optimierungsprobleme beinhaltet. Es ermöglicht nicht nur die Erstellung und anschließende Berechnung dieser Modelle, sondern auch die Erstellung einer grafischen Benutzeroberfläche. Für die Berechnung der Modelle stehen zahlreiche Solver zur Verfügung.

Um mit AIMMS zu arbeiten, muss zuerst ein Modell definiert werden. Der Aufbau unterteilt sich dabei in Mengen, Parameter, Variablen, Nebenbedingungen und die Zielfunktion. Eine Menge setzt sich dabei zusammen aus Objekten, welche in ihrer Struktur gleichartig sind. Beim LAMBDA-Modell wäre eine Menge zum Beispiel die Menge an Schwadronen einer Fraktion. Parameter sind jene Werte, welche schon zu Beginn gegeben sind. Das könnte beispielsweise die Effektivität eines jeden Schwadrons, eine bestimmte Mission zu fliegen, sein. Die Variablen sind ihrem Namen nach veränderliche Werte. Das Objekt, welches optimiert werden soll, muss als Variable definiert sein. Die Truppenallokation würde sich dafür anbieten, da eine Binärvariable angeben kann, ob ein Schwadron eine bestimmte Mission fliegt. Als Nebenbedingung müssen noch die Grenzen des Modells aufgestellt werden, sodass definiert ist, dass jedes Schwadron nur eine Mission fliegen kann oder wie viele Flüge pro Tag maximal zu bewältigen sind. Die Zielfunktion ist eine Gleichung, welche maximiert oder minimiert wird. Sie stellt in gewisser Weise den zu erzielenden Nutzen oder Schaden dar. Eine Zielfunktion könnte der erzielte Schaden an gegnerischen Stützpunkten und Flugzeugen sein, den es zu maximieren gilt. Ist das Modell erst richtig definiert, so macht es AIMMS sehr leicht, es auch zu lösen. Es muss nur ein passender Solver ausgewählt werden. Die Solver von AIMMS eignen sich unterschiedlich gut für folgende Problemtypen: Linear Program (LP), Mixed Integer Program (MIP), Quadratic Program (QP), Mixed Integer Quadratic Program (MIQP), Quadratically Constrained Program (QCP), Mixed Integer Quadratically Constrained Program (MIQCP), NonLinear Program (NLP), Mixed Integer NonLinear Program (MINLP), Mixed Complementarity Program (MCP), Mathematical Program with Complementarity Constraints (MPCC), Constraint Program (CP). Auch schwankt die Zeit, welche zur Lösung benötigt wird, da bei den Solvern unterschiedliche Algorithmen zur Anwendung kommen. Im Weiteren soll eine kleine Übersicht der bei AIMMS verfügbaren Solver gegeben werden, welche direkt der Internetseite von AIMMS [1] entnommen wurde.

- AOA: Das AOA (AIMMS Outer Approximation Modul) verwendet einen Outer Approximation Algorithmus, um MINLP Probleme zu lösen. Outer Approximation ist eine weitgehend verbreitete Herangehensweise für solche Probleme. Der Algorithmus setzt sich dabei aus zwei Solvern, einem für gemischt lineare Probleme und einem für nicht lineare Probleme, zusammen.
- BARON: Der BARON (Branch-And-Reduce Optimization Navigator) ist ein globaler Optimierungssolver. Das System ist dafür gedacht, das globale Optimum von nicht konvexen Optimierungsproblemen zu finden. Der Name BARON kommt von seiner Kombination aus Restriktionspropagation, Intervallanalyse und Verbindung von Branch and Bound mit Reduzierungskonzepten.
- CBC: Der CBC ist ein Open Source Solver für LP und MIP Probleme. Für das Lösen von LP Problemen kann zwischen dem Primal und Dual Simplex Algorithmus gewählt werden. Der Solver für MIP Probleme kann ebenfalls beide Algorithmen verwenden.

- CONOPT: CONOPT ist ein effizienter Solver für für NLP Probleme mit einer hohen Größenordnung. AIMMS kann exakte zweite Ableitungen bilden, welche von CONOPT genutzt werden können, um NLP Modelle viel effizienter zu lösen.
- CPLEX: CPLEX ist ein Hochleistungssolver für LP, MIP, QP, QCP, MIQP und MIQCP Probleme. Für LP Probleme stehen der Primal oder Dual Simplex Algorithmus, der Barrier Algorithmus und der Network Algorithmus zur Verfügung. Die MIP Probleme bewältigt der Solver mit dem Branch and Bound Verfahren, welcher moderne Verfahren wie Ebenenschnitte und Heuristik anwendet.
- CP Optimizer: Der CP Optimizer (CPOPT oder CPO) ist ein moderner Solver für CP Probleme. Die CP Technologie des Solvers ist sowohl für Bedingungserfüllungsprobleme, Optimierungsprobleme als auch Zeitplanung gedacht.
- GUROBI: Gurobi ist ein moderner Solver für LP, MIP, QP, QCP, MIQP und MIQCP Probleme. Für LP Probleme verwendet Gurobi fortschrittliche Algorithmen, wie den Primal und Dual Simplex Algorithmus, den Barrier Algorithmus und den Sifting Algorithmus. Für die MIP Probleme verwendet der Solver deterministische, parallele Branch and Cut, heuristische, Ebenenschnitt- und Symmetrienerkennungsverfahren.
- IPOPT: IPOPT ist ein Open Source Solver für nicht lineare Probleme. Der Solver ist ein Interne Punkt Optimierer für großangelegte nicht lineare Optimierung.
- KNITRO: Knitro ist ein Solver für NLP Probleme, wobei MPCC, MCP, konvexe oder nicht konvexe Nebenbedingungen und Grenzbedingungen mit eingeschlossen sind. MINLP Probleme können ebenfalls mit Knitro gelöst werden. Für die NLP Probleme werden ein Barrier Algorithmus und eine sequentielle QP Methode verwendet. Die MINLP Probleme mit einer nicht linearen Branch and Bound Methode und der Hybrid Quesada-Grossman Methode gelöst.
- MINOS: Der MINOS Solver ist für NLP Probleme gedacht. MINOS ist besonders effektiv für Probleme mit nicht linearer Zielfunktion und wenigen linearen Nebenbedingungen (z.B. QP). Die nicht linearen Funktionen sollten differenzierbar, müssen aber nicht konvex sein, damit MINOS sie bewältigen kann.
- PATH: Der PATH Solver ist ein Newton-basierter Solver zum Lösen von MCP Problemen. Der PATH Solver wird ausgiebig von Ökonomen verwendet, um generelle Gleichgewichtsprobleme zu lösen. Für die meisten MCP Probleme ist PATH ein robuster und effizienter Solver.
- SNOPT: SNOPT ist ein Solver für NLP Probleme. Besonders effektiv ist er dabei für NLP Probleme, deren Funktionen und Gradienten aufwendig zu lösen sind. Die Funktionen sollten differenzierbar sein, aber es ist nicht nötig, dass sie auch konvex sind.

Mit seiner Vielzahl an Solvern und seiner einfachen Bedienbarkeit eignet sich AIMMS hervorragend als Optimierungstool.

## 3 LAMBDA Modell

Um aus den Excel Tabellen ein mathematisches Modell aufzubauen, wurden alle Zellen auf die dahinter liegenden Formeln untersucht. Im folgenden Kapitel wird zuerst auf die für das Modell definierten Mengen und deren Indizes eingegangen. Zweitens wird auf die Parameter eingegangen, welche vom Bediener vor der Ausführung des Programms bestimmt werden und somit keine Berechnung beinhalten. Drittens werden jene Variablen beschrieben, welchen eine Formel zu Grunde liegt. Zuletzt werden die einzelnen Variablen noch Schritt für Schritt zusammengesetzt, um ein Gesamtmodell zu formen.

### 3.1 Sets

Die Sets bilden die Basis für dieses Modell. Sie werden gebraucht, um nicht jede Formel für jedes Objekt einzeln aufstellen zu müssen. So können an sich unterschiedliche, aber in ihrer Struktur gleichartige Objekte, zu Mengen, den sogenannten Sets, zusammengefasst werden. Den einzelnen Sets werden noch Indizes zugewiesen, welche im Folgenden erörtert werden:

- $t$ : Die Zeit. Im LAMBDA-Tool werden Tage als Zeiteinheit verwendet und im weiteren Verlauf der Arbeit wird der Anschaulichkeit halber auch häufig von Tagen gesprochen. Natürlich können die Zeitschritte beliebig groß gewählt werden.
- $o$ : Die Menge von Schwadronen, welche für die OPFOR fliegen.
- $a$ : Die Menge von Schwadronen, welche für die Alliance fliegen.
- $o_k$ : Eine Teilmenge von  $o$ . Die Schwadronen, welche Kamon zugehörig sind.
- $o_s$ : Eine Teilmenge von  $o$ . Die Schwadronen, welche Stellaria zugehörig sind.
- $a_f$ : Eine Teilmenge von  $a$ . Die Schwadronen, welche Friends zugehörig sind.
- $a_t$ : Eine Teilmenge von  $a$ . Die Schwadronen, welche Tytan zugehörig sind.
- $b$ : Die Menge aller Stützpunkte.
- $b_o$ : Eine Teilmenge von  $b$ . Die Menge der Luftstützpunkte, welche von den OPFOR Kräften genutzt werden.
- $b_k$ : Eine Teilmenge von  $b_o$ . Die Menge der Luftstützpunkte, welche von Kamon genutzt werden.

- $b_s$ : Eine Teilmenge von  $b_o$ . Die Menge der Luftstützpunkte, welche von Stellaria genutzt werden.
- $b_a$ : Eine Teilmenge von  $b$ . Die Menge der Luftstützpunkte, welche von der Alliance genutzt werden.
- $b_f$ : Eine Teilmenge von  $b_a$ . Die Menge der Luftstützpunkte, welche von Friends genutzt werden.
- $b_t$ : Eine Teilmenge von  $b_a$ . Die Menge der Luftstützpunkte, welche von Tytan genutzt werden.
- $c$ : Die verfügbaren Flugzeugtypen.
- $d$ : Die möglichen Missionstypen, für welche die Schwadronen eingesetzt werden können.

## 3.2 Parameter

Die Parameter werden in unabhängige und in abhängige Parameter unterteilt.

### 3.2.1 Unabhängige Parameter

Diese Parameter wirken global auf alle Akteure des Gefechts mit denselben Werten. Es handelt sich hierbei hauptsächlich um Faktoren, welche mit einem Wert zwischen 0 und 1 die Effektivität einer Angriffs- oder einer Verteidigungsaktion darstellen.

- $p_1$ : Effect of one Standard OCA Sortie  
Dieser Parameter gibt die Anzahl der Luftstützpunkte an, welche mittels eines OCA Sortie zerstört werden.
- $p_2$ : Effect of one Standard DCA Sortie  
Dieser Parameter gibt die Anzahl der Flugzeuge auf Penetrator Missionen an, welche mittels eines DCA Sortie zerstört werden.
- $p_3$ : Effect of one Standard ESC Sortie  
Dieser Parameter gibt die Anzahl der Flugzeuge auf DCA Missionen an, welche mittels eines ESC Sortie zerstört werden.
- $p_4$ : Effect of one Standard SEAD Sortie  
Dieser Parameter gibt die Anzahl der SAM FU's an, welche mittels eines SEAD Sortie zerstört werden.
- $p_5$ : Attrition from one SAM FU

Dieser Parameter gibt die Anzahl der Flugzeuge auf Penetrator Missionen an, welche mittels einer SAM FU zerstört werden.

- $p_6$ : Multiplicative Effect of one AEW Sortie  
Dieser Parameter gibt an, um das wievielfache sich die DCA Missionen durch die Nutzung von Früherkennung verbessern.
- $p_7$ : Multiplicative Effect of one ESM Sortie  
Dieser Parameter gibt an, um das wievielfache sich die eigenen Verluste von Flugzeugen auf Penetrator Missionen verringern, wenn elektronische Unterstützungsmaßnahmen eingesetzt werden.
- $p_8$ : Multiplicative Effect of one AI/Counter C2 Sortie  
Dieser Parameter gibt an, um das wievielfache sich die eigenen Verluste von Flugzeugen auf Penetrator Missionen verringern, wenn AI Missionen geflogen werden.
- $p_9$ : Maximum Penetrator or DCA Attrition  
Dies ist die maximale Abnutzung, welche innerhalb eines Tages auf die Flugzeuge auf Penetrator und DCA Missionen wirken kann.
- $p_{10}$ : Fraction of Penetrator Losses on Ingress  
Dieser Parameter wird genutzt, um den Anteil der Penetrator, welche im Anflug zerstört werden, von denjenigen zu trennen, welche erst auf dem Rückweg zerstört werden. Dies ist wichtig, da im Anflug zerstörte Flugzeuge keinen Schaden bei den Gegnern anrichten können.
- $p_{11}$ : Airbases killed by one TLAM  
Dieser Parameter gibt die Anzahl der Luftstützpunkte an, welche mittels einer Tomahawk Rakete zerstört werden.
- $p_{12}$ : Multiplicative Factor for Point Defence SAM Use  
Dieser Parameter gibt an, wie wirksam sich Luftstützpunkte vor direkten Angriffen im Nahbereich mittels Raketen schützen können.
- $p_{13}$ : Multiplicative Factor for Self Defence Virtual Sorties  
Dieser Parameter gibt an, wie wirksam sich die Selbstverteidigung der Flugzeuge auf die Verluste der gegnerischen Flugzeuge auf DCA Missionen auswirkt.
- $p_{14}$ : Multiplicative Factor to Airbase Attrition appropriate to aircraft KOG  
Dieser Parameter gibt an, mit welchem Anteil sich die Abnutzung von Luftstützpunkten auf die Verluste von Flugzeugen auswirkt, welche bei Angriffen am Boden waren.

### 3.2.2 Abhängige Parameter

Die abhängigen Parameter können unterschiedliche Werte annehmen. Die meisten sind abhängig von der Zeit und können jeden Tag einen anderen Wert zugewiesen bekommen. Parameter, welche von mehr als einem Set abhängig sind, bekommen für jede Kombination einen Wert zugeordnet.

- $\alpha_o(t)$ : Weather Factor OPFOR

Dieser Parameter ist abhängig von der Zeit und gibt an, wie weit sich die Abflugrate der OPFOR Flugzeuge aufgrund des Wetters verringert.

- $\alpha_a(t)$ : Weather Factor Alliance

Dieser Parameter ist abhängig von der Zeit und gibt an, wie weit sich die Abflugrate der Alliance Flugzeuge aufgrund des Wetters verringert.

- $\beta_o(t, j)$ : Surge Factor OPFOR

Dieser Parameter ist sowohl abhängig von der Zeit als auch davon, auf welchen Missionstypen er angewendet wird. Er wird genutzt, um tägliche Veränderungen für die Effektivität von den unterschiedlichen Missionen zu berücksichtigen.

- $\beta_a(t, j)$ : Surge Factor Alliance

Dieser Parameter ist sowohl abhängig von der Zeit als auch davon, auf welchen Missionstypen er angewendet wird. Er wird genutzt, um tägliche Veränderungen für die Effektivität von den unterschiedlichen Missionen zu berücksichtigen.

- $\gamma_o(t)$ : SSM launched by OPFOR

Dieser Parameter gibt an, wie viele Boden-Boden-Raketen täglich von den OPFOR Kräften gestartet werden.

- $\gamma_a(t)$ : SSM launched by Alliance

Dieser Parameter gibt an, wie viele Boden-Boden-Raketen täglich von den Alliance Kräften gestartet werden.

- $\delta_o(t)$ : CAS Attrition OPFOR

Dieser Parameter gibt den Verbrauch von Flugkörpern der OPFOR bei der Luftnahunterstützung an.

- $\delta_a(t)$ : CAS Attrition Alliance

Dieser Parameter gibt den Verbrauch von Flugkörpern der OPFOR bei der Luftnahunterstützung an.

- $\epsilon(c, d)$ : Sortie Rate  
Dieser Parameter gibt die Flugrate an, mit der eine Mission von einem bestimmten Flugzeugtyp geflogen werden kann.
- $\eta(c, d)$ : Relative Effectiveness  
Dieser Parameter ist abhängig vom Flugzeug- und Missionstyp, da er aussagt, mit welcher Effektivität ein Flugzeug eine bestimmte Mission fliegen kann.
- $\theta(c)$ : Relative Survivability  
Dieser Parameter gibt die Überlebenschance eines Flugzeuges auf seiner Mission an (quasi ein Wert für die Panzerung oder Ausweichfähigkeit).
- $\lambda(c)$ : Self Defence Factor  
Dieser Parameter gibt an, in welchem Maße sich ein Flugzeug zur Wehr setzen kann (quasi ein Wert für die Bewaffnung).
- $\mu(b)$ : Shelters  
Dieser Parameter gibt an, wie viele Unterstände es auf einem Luftstützpunkt für die Flugzeuge gibt. In diesem Modell wird angenommen, dass ein Unterstand ein Flugzeug absolut vor Beschuss schützt.
- $\pi(b)$ : Size  
Dieser Parameter gibt die Größe eines Luftstützpunktes an. Dies hat Einfluss auf die Reparaturfähigkeit, da Schäden bei einem großen Stützpunkt schwächer ins Gewicht fallen.
- $\sigma_k(t)$ : AREA SAM % effective this day Kamon  
Dieser Parameter gibt an, wie effektiv die flächendeckenden Boden-Luft-Raketen jeden Tag von den Kamon Kräften eingesetzt werden können.
- $\sigma_s(t)$ : AREA SAM % effective this day Stellaria  
Dieser Parameter gibt an, wie effektiv die flächendeckenden Boden-Luft-Raketen jeden Tag von den Stellaria Kräften eingesetzt werden können.
- $\sigma_f(t)$ : AREA SAM % effective this day Friends  
Dieser Parameter gibt an, wie effektiv die flächendeckenden Boden-Luft-Raketen jeden Tag von den Friends Kräften eingesetzt werden können.
- $\sigma_t(t)$ : AREA SAM % effective this Day Tytan  
Dieser Parameter gibt an, wie effektiv die flächendeckenden Boden-Luft-Raketen jeden Tag von den Tytan Kräften eingesetzt werden können.

- $\phi_k(t)$ : AREA SAM Reinforcements Kamon  
Dieser Parameter gibt an, wie hoch die Verstärkung an Boden-Luft-Raketen jeden Tag ausfällt.
- $\phi_s(t)$ : AREA SAM Reinforcements Stellaria  
Dieser Parameter gibt an, wie hoch die Verstärkung an Boden-Luft-Raketen jeden Tag ausfällt.
- $\phi_f(t)$ : AREA SAM Reinforcements Friends  
Dieser Parameter gibt an, wie hoch die Verstärkung an Boden-Luft-Raketen jeden Tag ausfällt.
- $\phi_t(t)$ : AREA SAM Reinforcements Tytan  
Dieser Parameter gibt an, wie hoch die Verstärkung an Boden-Luft-Raketen jeden Tag ausfällt.
- $\psi(b)$ : Point Defence Score  
Dieser Parameter gibt die Verteidigungsfähigkeit eines Stützpunktes gegen direkte Angriffe im Nahbereich an.
- $\omega(t, b)$ : Repair Capacity  
Dieser Parameter gibt an, wie viel Prozent eines Stützpunktes an einem Tag wieder hergestellt werden kann.
- $\xi(t, b)$ : Enemy OCA WT  
Dieser Parameter ist ein Prozentsatz und gibt an, welcher Teil der gegnerischen Angriffe sich auf jeden Stützpunkt richtet. Die Summe dieser Werte über alle Stützpunkte einer Fraktion muss somit 1 ergeben.
- $\zeta(t, o/a)$ : Mission of Squadron  
Dieser Parameter gibt an, welche Mission jedes Schwadron jeden Tag fliegen soll.
- $\tau(o/a)$ : Airbase of Squadron  
Dieser Parameter gibt an, auf welchem Stützpunkt das Schwadron stationiert sein soll.
- $\rho(o/a)$ : Aircraft of Squadron  
Dieser Parameter gibt an, aus welchem Flugzeugtyp das Schwadron bestehen soll.

Die folgenden Parameter sind die Missionsindikatoren. Sie werden jedem Schwadron zugewiesen. Es handelt sich hierbei um binäre Parameter, die angeben, ob ein Flugzeug einen Missionstypen fliegt oder nicht. Von den ersten acht Parametern darf nur einem der Wert 1 zugewiesen werden, wobei 1 bedeutet, dass die entsprechende Mission geflogen wird. Die Missionsindikatoren müssen auf den Parameter  $\zeta(t, o/a)$  abgestimmt sein.

- $i_{AEW}(t, o/a)$ : AEW Indicator.
- $i_{DCA}(t, o/a)$ : DCA Indicator.
- $i_{OCA}(t, o/a)$ : OCA Indicator.
- $i_{ESC}(t, o/a)$ : ESC Indicator.
- $i_{ESM}(t, o/a)$ : ESM Indicator.
- $i_{SEAD}(t, o/a)$ : SEAD Indicator.
- $i_{AI}(t, o/a)$ : AI Indicator.
- $i_{CAS}(t, o/a)$ : CAS Indicator.
- $i_{PwESC}(t, o/a)$ : PEN Indicator without ESC  
Dieser Indikator wird auf 1 gesetzt, wenn das Schwadron entweder eine OCA, ESM, SEAD oder AI Mission fliegen soll.
- $i_{PEN}(t, o/a)$ : PEN Indicator  
Dieser Indikator wird auf 1 gesetzt, wenn das Schwadron entweder eine OCA, ESC, ESM, SEAD oder AI Mission fliegen soll.

### 3.3 Variablen

Das folgende Unterkapitel widmet sich dem Teil des LAMBDA-Tool, in dem die tatsächlichen Berechnungen für die Simulation durchgeführt werden. Für die Berechnungen werden eine Vielzahl von Variablen benutzt, die hier aufgezählt und beschrieben werden. Die genutzten Formeln sind für alle Fraktionen dieselben, allerdings sind manche Werte der einen Fraktion von Werten von der anderen Fraktion abhängig, weshalb nicht die Gesamtheit aller Schwadronen, sondern immer nur die Teilmenge der OPFOR Schwadronen oder der Alliance Schwadronen betrachtet werden kann. Zum Schluss werden alle Formeln für beide Seite aufgestellt. Der Einfachheit halber werden bei der Aufzählung aber nur jene Variablen betrachtet, welche nötig sind, um die Verluste der eigenen Truppen aus Sicht der OPFOR darzustellen. Durch Vertauschen der Indizes sind diese aber genauso für die Alliance gültig.

### 3.3.1 Stützpunktvariablen

Zuerst werden jene Variablen betrachtet, welche sich auf die Stützpunkte beziehen.

- $e(t, b_o)$ : Status Prior Repair

Diese Variable beschreibt den Status eines Stützpunktes am Anfang des Tages, bevor er repariert wurde. Der Status eines Stützpunktes kann nicht unter 0 Prozent fallen, weshalb hier eine Maximumsfunktion benutzt wird:

$$e(t, b_o) = \max \left( 0; f(t-1, b_o) - \left( \frac{\xi(t-1, b_o) \cdot 100}{\pi(b_o)} \cdot v_o(t-1) \right) \right). \quad (3.1)$$

- $f(t, b_o)$ : Status After Repair

Diese Variable beschreibt den Status eines Stützpunktes, nachdem er repariert wurde. Der Status eines Stützpunktes kann nicht über 100 Prozent steigen, weshalb hier eine Minimumfunktion verwendet wird:

$$f(t, b_o) = \min \left( 100; e(t, b_o) + \omega(t, b_o) \right). \quad (3.2)$$

- $g(t, b_o)$ : Aircraft on Base

Diese Variable beschreibt die Summe der Flugzeuge, welche auf dem Stützpunkt stationiert sind:

$$g(t, b_o) = \sum_{o|\tau(o)=b_o} k(t, o). \quad (3.3)$$

- $h(t, b_o)$ : Aircraft in Open

Diese Variable gibt an, wie viele Flugzeuge auf dem Stützpunkt keinen Unterstand zur Verfügung haben. Da dieser Wert nicht unter 0 sinken kann, wird eine Maximumsfunktion verwendet:

$$h(t, b_o) = \max \left( 0; g(t, b_o) - \mu(b_o) \right). \quad (3.4)$$

- $i(t, b_o)$ : Potential Aircraft KOG Previous Day

Diese Variable beschreibt die Anzahl der Flugzeuge, welche am Vortag durch Angriffe am Boden zerstört wurden. Diese Zahl muss sich zwischen 0 und der Anzahl der Flugzeuge im Freien befinden, weshalb die Minimums- und Maximumsfunktion verwendet wird:

$$i(t, b_o) = \max \left( 0; \min \left( h(t-1, b_o); h(t-1, b_o) \cdot \frac{p_{14}}{100} \cdot (f(t-1, b_o) - f(t, b_o)) \right) \right). \quad (3.5)$$

- $j(t, b_o)$ : Percentage Aircraft on Base KOG

Diese Variable ist das Verhältnis von zerstörten Flugzeugen zur Anzahl von auf dem Stützpunkt stationierten Flugzeugen:

$$j(t, b_o) = \begin{cases} \frac{i(t, b_o)}{g(t-1, b_o)} & ; \text{wenn } g(t-1, b_o) > 0, \\ 0 & ; \text{sonst.} \end{cases} \quad (3.6)$$

### 3.3.2 Schwadronenvariablen

Eine weitere Gruppe von Variablen sind jene, welche sich direkt auf die Schwadronen beziehen.

- $k(t, o)$ : Current

Diese Variable gibt die Anzahl der Flugzeuge an, welche sich momentan in einer Schwadron befinden. Da diese Anzahl nicht negativ ausfallen kann, wird hier eine Maximumsfunktion benutzt:

$$k(t, o) = \max \left( 0; l(t, o) \cdot (1 - j(t, \tau(o))) \right). \quad (3.7)$$

- $l(t, o)$ : Current before KOG

Diese Variable auch die Anzahl der Flugzeuge in einem Schwadron an, allerdings wurden hier die Verluste von am Boden zerstörten Einheiten noch nicht mit eingerechnet:

$$l(t, o) = \max \left( 0; k(t-1, o) \cdot \left( 1 - m(t, o) \cdot p(t-1, o) \cdot \frac{\alpha_o(t-1)}{\theta(\rho(o))} \right) \right). \quad (3.8)$$

- $m(t, o)$ : Base Attrition previous Day

Diese Variable gibt die grundlegende Abnutzung eines Schwadrons am Vortag an.

$$m(t, o) = n(t-1, o) \quad (3.9)$$

- $n(t, o)$ : Base Attrition this Day

Diese Variable gibt die grundlegende Abnutzung einer Schwadron an einem bestimmtem Tag an. Es wird dabei die Abnutzung übernommen, die der Mission der Schwadron entspricht, da nur einer der Missionsindikatoren gleich 1 ist:

$$n(t, o) = i_{PEN}(t, o) \cdot A_o(t) + i_{CAS}(t, o) \cdot \delta_o(t) + i_{DCA}(t, o) \cdot M_o(t). \quad (3.10)$$

- $p(t, o)$ : Actual Sortie Rate

Diese Variable passt die Sortie Rate eines Flugzeuges mit Hilfe des Surge Factors an:

$$p(t, o) = \epsilon(\rho(o), \zeta(o)) \cdot \beta_o(t, \zeta(o)). \quad (3.11)$$

- $q(t, o)$ : Actual Sorties

Diese Variable gibt an, wie viele Sorties pro Schwadron tatsächlich geflogen werden. Dabei wird neben der Actual Sortie Rate auch das Wetter, der Status des Stützpunktes und die Anzahl der verfügbaren Flugzeuge in Betracht gezogen:

$$q(t, o) = k(t, o) \cdot f(t, \tau(o)) \cdot \frac{p(t, o) \cdot \alpha_o(t)}{100}. \quad (3.12)$$

- $r(t, o)$ : Standard Sorties

Mit dieser Variable wird eine theoretische Anzahl von Sorties errechnet, welche die relative Effektivität der Flugzeuge auf ihrer Mission mit einbezieht:

$$r(t, o) = \eta(\rho(o), \zeta(o)) \cdot q(t, o). \quad (3.13)$$

- $u(t, o)$ : Self Defence Virtual Sorties

Dies ist ebenfalls eine theoretische Anzahl von Sorties. Sie gibt an, mit welchem Anteil eine hohe Selbstverteidigungsfähigkeit als ESC Mission gerechnet werden kann:

$$u(t, o) = i_{PwESC}(t, o) \cdot q(t, o) \cdot \lambda(\rho(o)) \cdot p_{13}. \quad (3.14)$$

### 3.3.3 Fraktionsvariablen

Neben den ersten beiden Gruppen von Variablen gibt es noch eine weitere, welche die tatsächliche Verbindung zwischen der Truppenstärke der einen Fraktion und der Abnutzung der anderen Fraktion herstellen. Diese Variablen beziehen sich immer nur auf eine Fraktion und nicht auf einzelne Schwadronen oder Stützpunkte.

- $v_o(t)$ : Airbase Losses OPFOR

Diese Variable relativiert die Stützpunktverluste mittels  $p_{10}$ , da hier berücksichtigt wird, dass Angreifer, welche im Anflug schon zerstört wurden, keinen Schaden angerichtet haben:

$$v_o(t) = w_o(t) \cdot (1 - A_o(t)) \cdot p_{10}. \quad (3.15)$$

- $w_o(t)$ : Total Number of OPFOR Bases Killed

Diese Variable gibt an, wie viele Stützpunkte durch Flugkörper und Flugzeugangriffe zerstört wurden:

$$w_o(t) = x_o(t) + z_o(t). \quad (3.16)$$

- $x_o(t)$ : OPFOR Bases Killed by OCA

Diese Variable gibt an, wie viele Stützpunkte durch von der Alliance geflogene Flugzeugangriffe zerstört wurden:

$$x_o(t) = p_1 \cdot S_{OCA_a}(t). \quad (3.17)$$

- $z_o(t)$ : OPFOR Bases Killed by SSM

Diese Variable gibt an, wie viele Stützpunkte durch von der Alliance abgefeuerte Boden-Boden-Raketen zerstört wurden:

$$z_o(t) = p_{11} \cdot \gamma_a(t). \quad (3.18)$$

- $A_o(t)$ : Basic Penetrator Attrition OPFOR

Diese Variable beschreibt das Verhältnis zwischen der Anzahl der Flugzeuge, welche auf ihrer Penetrator Mission zerstört wurden und der totalen Anzahl an geflogenen Penetrator Missionen:

$$A_o(t) = \min \left( p_9; \frac{B_o(t) + F_o(t)}{S_{PEN_o}(t)} \right). \quad (3.19)$$

- $B_o(t)$ : Potential Number of OPFOR Penetrators Killed by DCA

Diese Variable gibt an, wie viele Flugzeuge auf PEN Mission von gegnerischen Flugzeugen auf DCA Mission zerstört wurden. Die Effektivität eines DCA Fluges wird dabei mit einbezogen:

$$B_o(t) = p_2 \cdot C_a(t). \quad (3.20)$$

- $C_a(t)$ : Number of Alliance Standard DCA Sorties (Mod for AI)

Diese Variable verringert die geflogenen DCA Missionen, je nachdem wie viele gegnerische AI Missionen geflogen wurden:

$$C_a(t) = p_8^{(S_{AI_o}(t))} \cdot D_a(t). \quad (3.21)$$

- $D_a(t)$ : Number of Alliance Standard DCA Sorties (Mod for ESM)

Diese Variable verringert die geflogenen DCA Missionen, je nachdem wie viele gegnerische ESM Missionen geflogen wurden:

$$D_a(t) = p_7^{(S_{ESM_o}(t))} \cdot E_a(t). \quad (3.22)$$

- $E_a(t)$ : Number of Alliance Standard DCA Sorties (Mod for AEW)

Diese Variable erhöht die geflogenen DCA Missionen, je nachdem wie viele eigene AEW Missionen geflogen wurden:

$$E_a(t) = p_6^{(S_{AEW_a}(t))} \cdot S_{DCA_a}(t). \quad (3.23)$$

- $F_o(t)$ : Potential Number of OPFOR Penetrators Killed by SAM

Diese Variable gibt die Anzahl der Flugzeuge auf PEN Mission an, welche von gegnerischen Boden-Luft-Raketen zerstört wurden. Mit einbezogen werden dabei die Effektivität eines Flugkörpers und wie stark eigene AI und ESM Missionen die gegnerischen Abschüsse verringern:

$$F_o(t) = G_a(t) \cdot p_5 \cdot p_8^{(S_{AI_o}(t))} \cdot p_7^{(S_{ESM_o}(t))}. \quad (3.24)$$

- $G_a(t)$ : Number of Alliance Standard SAM FU's Available and in Action

Diese Variable gibt an, wie viele Boden-Luft-Raketen effektiv einsetzbar sind. Davon wird die Anzahl an zerstörten Raketen an diesem Tag abgezogen. Da dieser Wert bei weiteren Rechnungen als alleiniger Nenner vorkommt, wird hier eine Maximumsfunktion mit Wert 0,1 benutzt, um den Wert 0 zu verhindern:

$$G_a(t) = \max \left( 0, 1; I_f(t) + I_t(t) - H_a(t) \right). \quad (3.25)$$

- $H_a(t)$ : Number of Alliance SAM FU's killed by SEAD

Diese Variable beschreibt die tatsächliche Anzahl an zerstörten Flugkörpern durch gegnerische SEAD Missionen:

$$H_a(t) = p_4 \cdot S_{SEAD_o}(t). \quad (3.26)$$

- $I_f(t)$ : Total Friends SAM Effective this Day

Diese Variable gibt an, wie viele Boden-Luft-Raketen durch Friends effektiv einsetzbar sind. Im Unterschied zu  $G_a(t)$  sind die an diesem Tag zerstörten Raketen noch nicht mit einbezogen. Die Effektivität wird dabei durch die tatsächliche Anzahl an Raketen und die Effektivität für flächendeckenden und gezielten Einsatz beeinflusst:

$$I_f(t) = J_f(t) \cdot \sigma_f(t) + L_f(t). \quad (3.27)$$

- $I_t(t)$ : Total Tytan SAM Effective this Day

Diese Variable gibt an, wie viele Boden-Luft-Raketen durch Tytan effektiv einsetzbar sind. Im Unterschied zu  $G_a(t)$  sind die an diesem Tag zerstörten Raketen noch nicht mit einbezogen. Die Effektivität wird dabei durch die tatsächliche Anzahl an Raketen und die Effektivität für flächendeckenden und gezielten Einsatz beeinflusst:

$$I_t(t) = J_t(t) \cdot \sigma_t(t) + L_t(t). \quad (3.28)$$

- $J_f(t)$ : AREA Friends SAM FU's Available at Day Start

Diese Variable beschreibt die tatsächliche Anzahl an vorhandenen Flugkörpern von Friends. Es werden sowohl Verluste durch Beschuss, als auch neu eintreffende Verstärkung berücksichtigt. Da dieser Wert bei weiteren Rechnungen als alleiniger Nenner vorkommt, wird hier eine Maximumsfunktion benutzt, um den Wert 0 zu verhindern:

$$J_f(t) = \max \left( 0, 1; J_f(t-1) - K_f(t-1) + \phi_f(t) \right). \quad (3.29)$$

- $J_t(t)$ : AREA Tytan SAM FU's Available at Day Start

Diese Variable beschreibt die tatsächliche Anzahl an vorhandenen Flugkörpern von Tytan. Es werden sowohl Verluste durch Beschuss, als auch neu eintreffende Verstärkung berücksichtigt. Da dieser Wert bei weiteren Rechnungen als alleiniger Nenner vorkommt, wird hier eine Maximumsfunktion benutzt, um der Wert 0 zu verhindern:

$$J_t(t) = \max \left( 0, 1; J_t(t-1) - K_t(t-1) + \phi_t(t) \right). \quad (3.30)$$

- $K_f(t)$ : AREA Friends SAM Losses to SEAD

Diese Variable berechnet den Anteil der zerstörten Alliance SAM FU's, welcher Friends betrifft:

$$K_f(t) = H_a(t) \cdot \frac{I_f(t)}{I_f(t) + I_t(t)}. \quad (3.31)$$

- $K_t(t)$ : AREA Tytan SAM Losses to SEAD

Diese Variable berechnet den Anteil der zerstörten Alliance SAM FU's, welcher Tytan betrifft:

$$K_t(t) = H_a(t) \cdot \frac{I_t(t)}{I_f(t) + I_t(t)}. \quad (3.32)$$

- $L_f(t)$ : Friends Point SAM effective vs. OPFOR

Diese Variable kombiniert die Effektivität von einem Flugkörpereinsatz im Nahbereich mit der Verteidigungsfähigkeit jedes Stützpunktes und dessen Status, um eine Gesamteffektivität gegen gegnerische Flugzeugangriffe zu erzeugen:

$$L_f(t) = \frac{p_{12}}{100} \cdot \sum_{b_f} \left[ \psi(b_f) \cdot f(t, b_f) \cdot \xi(t, b_f) \right]. \quad (3.33)$$

- $L_t(t)$ : Tytan Point SAM effective vs. OPFOR

Diese Variable kombiniert die Effektivität von einem Flugkörpereinsatz im Nahbereich mit der Verteidigungsfähigkeit jedes Stützpunktes und dessen Status, um eine Gesamteffektivität gegen gegnerische Flugzeugangriffe zu erzeugen:

$$L_t(t) = \frac{p_{12}}{100} \cdot \sum_{b_t} \left[ \psi(b_t) \cdot f(t, b_t) \cdot \xi(t, b_t) \right]. \quad (3.34)$$

- $M_o(t)$ : Basic DCA Attrition OPFOR

Diese Variable beschreibt das Verhältnis zwischen der Anzahl der Flugzeuge, welche auf ihrer DCA Mission zerstört wurden und der totalen Anzahl an geflogenen DCA Missionen:

$$M_o(t) = \text{Min} \left( p_9; \frac{N_o(t)}{S_{DCA_o}(t)} \right). \quad (3.35)$$

- $N_o(t)$ : Potential Number OPFOR DCA killed by ESC

Diese Variable gibt an, wie viele Flugzeuge auf DCA Mission von gegnerischen Flugzeugen auf ESC Mission und durch Selbstverteidigung zerstört wurden. Die Effektivität einer ESC Mission wird dabei mit einbezogen:

$$N_o(t) = p_3 \cdot (S_{ESC_a}(t) + S_{PwESC_a}(t)). \quad (3.36)$$

- $S_{AEW_a}(t)$ : Number of Alliance Standard AEW Sorties

Diese Variable gibt die Summe aller von der Alliance geflogenen standardisierten AEW Missionen an:

$$S_{AEW_a}(t) = \sum_a \left[ i_{AEW}(t, a) \cdot r(t, a) \right]. \quad (3.37)$$

- $S_{DCA_o}(t)$ : Number of OPFOR Standard DCA Sorties

Diese Variable gibt die Summe aller von OPFOR geflogenen standardisierten DCA Missionen an:

$$S_{DCA_o}(t) = \sum_o \left[ i_{DCA}(t, o) \cdot r(t, o) \right]. \quad (3.38)$$

- $S_{DCA_a}(t)$ : Number of Alliance Standard DCA Sorties

Diese Variable gibt die Summe aller von der Alliance geflogenen standardisierten DCA Missionen an:

$$S_{DCA_a}(t) = \sum_a \left[ i_{DCA}(t, a) \cdot r(t, a) \right]. \quad (3.39)$$

- $S_{OCA_a}(t)$ : Number of Alliance Standard OCA Sorties

Diese Variable gibt die Summe aller von der Alliance geflogenen standardisierten OCA Missionen an:

$$S_{OCA_a}(t) = \sum_a \left[ i_{OCA}(t, a) \cdot r(t, a) \right]. \quad (3.40)$$

- $S_{ESC_a}(t)$ : Number of Alliance Standard ESC Sorties

Diese Variable gibt die Summe aller von der Alliance geflogenen standardisierten ESC Missionen an:

$$S_{ESC_a}(t) = \sum_a \left[ i_{ESC}(t, a) \cdot r(t, a) \right]. \quad (3.41)$$

- $S_{ESM_o}(t)$ : Number of OPFOR Standard ESM Sorties

Diese Variable gibt die Summe aller von OPFOR geflogenen standardisierten ESM Missionen an:

$$S_{ESM_o}(t) = \sum_o \left[ i_{ESM}(t, o) \cdot r(t, o) \right]. \quad (3.42)$$

- $S_{SEAD_o}(t)$ : Number of OPFOR Standard SEAD Sorties

Diese Variable gibt die Summe aller von OPFOR geflogenen standardisierten SEAD Missionen an:

$$S_{SEAD_o}(t) = \sum_o \left[ i_{SEAD}(t, o) \cdot r(t, o) \right]. \quad (3.43)$$

- $S_{AI_o}(t)$ : Number of OPFOR Standard AI Sorties

Diese Variable gibt die Summe aller von OPFOR geflogenen standardisierten AI Missionen an:

$$S_{AI_o}(t) = \sum_o \left[ i_{AI}(t, o) \cdot r(t, o) \right]. \quad (3.44)$$

- $S_{PwESC_a}(t)$ : Number of Alliance Self Defence Virtual Standard Sorties

Diese Variable gibt die Summe aller von der Alliance geflogenen theoretischen Selbstverteidigungsmissionen an:

$$S_{PwESC_a}(t) = \sum_a \left[ u(t, a) \right]. \quad (3.45)$$

- $S_{PEN_o}(t)$ : Number of Actual OPFOR Penetrator Sorties

Diese Variable gibt die Summe aller von OPFOR geflogenen standardisierten Penetrator Missionen an:

$$S_{PEN_o}(t) = \sum_o \left[ i_{PEN}(t, o) \cdot q(t, o) \right]. \quad (3.46)$$

### 3.4 Erstellen eines Gesamtmodells

Nachdem alle notwendigen Mengen, Parameter und Variablen aufgestellt wurden, geht es darum, die Gleichungen zusammenzufügen, um eine ganzheitliche Formel des Modells und einen Überblick zu erhalten. Da bekannt ist, dass es sich hierbei um ein abgewandeltes Lanchester Modell handelt, müssen die Differentialgleichungen identifiziert werden. Es fällt auf, dass zum Beschreiben der OPFOR Verluste vier Differentialgleichungen notwendig sind. Sowohl der Status der Stützpunkte ( $e(t, b_o)$ ), die Anzahl der Flugzeuge ( $k(t, o)$ ), als auch die Anzahl der SAM FU's ( $J_f(t)$  und  $J_t(t)$ ) sind jeweils direkt oder indirekt (über eine andere Variable) von ihrem Zustand am Vortag abhängig. Diesen vier Differentialgleichungen wird sich im Folgenden einzeln gewidmet. Wie schon vorhergehend erwähnt, wird im LAMBDA-Tool die Zeiteinheit "Tag" verwendet. Da diese Zeiteinheit aber auch infinitesimal klein gewählt werden kann, wird hier der Differentialoperator verwendet. Zusätzlich werden noch Variablen (welche hier als Hilfsvariablen kategorisiert werden) als Platzhalter verwendet, um die Übersichtlichkeit zu wahren.

#### 3.4.1 Differentialgleichung für den Status der Stützpunkte

Zuerst wird sich mit dem Status der Stützpunkte befasst. Mit einem Blick auf  $e(t, b_o)$  und  $f(t, b_o)$  lässt sich erkennen, dass der Status nach der Reparatur in Verbindung mit dem Status vor der Reparatur von sich selber abhängt. Setzt man  $e(t, b_o)$  in  $f(t, b_o)$  ein, so erhält man die Ausgangsfunktion für weitere Umformungen:

$$f(t, b_o) = \min \left( 100; \max \left( 0; f(t-1, b_o) - \left( \frac{\xi(t-1, b_o) \cdot 100}{\pi(b_o)} \cdot v_o(t-1) \right) \right) + \omega(t, b_o) \right). \quad (3.47)$$

Die Bedingung der Maximumsfunktion wird weiterhin eingehalten, wenn der erste Summand herausgezogen wird und die Bedingung aufgestellt wird, dass der zweite Summand niemals größer wird als der erste. Es folgt:

$$f(t, b_o) = \min \left( 100; \omega(t, b_o) + f(t-1, b_o) - \min \left( f(t-1, b_o); \left( \frac{\xi(t-1, b_o) \cdot 100}{\pi(b_o)} \cdot v_o(t-1) \right) \right) \right). \quad (3.48)$$

Um die Veränderung von einem Tag auf den nächsten Tag zu erhalten, wird auf beiden Seiten der Status vom Vortag subtrahiert:

$$f(t, b_o) - f(t-1, b_o) = \min \left( 100 - f(t-1, b_o); \omega(t, b_o) - \min \left( f(t-1, b_o); \left( \frac{\xi(t-1, b_o) \cdot 100}{\pi(b_o)} \cdot v_o(t-1) \right) \right) \right). \quad (3.49)$$

Ersetzt man jetzt die Differenz durch den Differentialquotienten und verschiebt die Formel auf der Zeitachse eine Zeiteinheit weiter, so erhält man folgende Gleichung:

$$\frac{df(t, b_o)}{dt} = \min \left( 100 - f(t, b_o); \omega(t+1, b_o) - \min \left( f(t, b_o); \left( \frac{\xi(t, b_o) \cdot 100}{\pi(b_o)} \cdot v_o(t) \right) \right) \right). \quad (3.50)$$

Durch Einsetzen von  $v_o(t)$  erweitert sich die Gleichung wie folgt:

$$\frac{df(t, b_o)}{dt} = \min \left( 100 - f(t, b_o); \omega(t+1, b_o) - \min \left( f(t, b_o); \left( \frac{\xi(t, b_o) \cdot 100}{\pi(b_o)} \cdot w_o(t) \cdot (1 - A_o(t) \cdot p_{10}) \right) \right) \right). \quad (3.51)$$

Ersetzt man nun  $x_o(t)$  und  $z_o(t)$  in  $w_o(t)$  und setzt dieses wiederum in die Gleichung ein, erhält man Folgendes:

$$\frac{df(t, b_o)}{dt} = \min \left( 100 - f(t, b_o); \omega(t+1, b_o) - \min \left( f(t, b_o); \left( \frac{\xi(t, b_o) \cdot 100}{\pi(b_o)} \cdot (p_1 \cdot SOCA_a(t) + p_{11} \cdot \gamma_a(t)) \cdot (1 - A_o(t) \cdot p_{10}) \right) \right) \right). \quad (3.52)$$

### 3.4.2 Differentialgleichung für die Anzahl der Flugzeuge

Als nächstes wird die Anzahl der Flugzeuge betrachtet. Auch diese Variable ist indirekt von sich selbst abhängig, sodass durch Einsetzen von  $l(t, o)$  in  $k(t, o)$  eine Grundformel gebildet wird:

$$k(t, o) = \max \left( 0; \max \left( 0; k(t-1, o) \cdot \left( 1 - m(t, o) \cdot p(t-1, o) \cdot \frac{\alpha_o(t-1)}{\theta(\rho(o))} \right) \right) \cdot (1 - j(t, \tau(o))) \right). \quad (3.53)$$

Da  $k(t, o)$  laut seiner Definition zu jedem Zeitpunkt größer als 0 ist, hat es innerhalb der zweiten Maximumsfunktion keinen Einfluss darauf, welcher der beiden Werte gewählt wird.  $k(t-1, o)$  kann deshalb auch nach Ausführen der Maximumsfunktion hinzu multipliziert werden. Des Weiteren wird wieder die Anzahl der Flugzeuge des Vortages von beiden Seiten subtrahiert, um die Differenz von einem Tag auf den nächsten darzustellen:

$$k(t, o) - k(t-1, o) = \max \left( 0; k(t-1, o) \cdot \max \left( 0; 1 - m(t, o) \cdot p(t-1, o) \cdot \frac{\alpha_o(t-1)}{\theta(\rho(o))} \right) \cdot (1 - j(t, \tau(o))) \right) - k(t-1, o). \quad (3.54)$$

Die Differenz wird jetzt wieder durch den Differentialquotienten ersetzt.  $k(t-1, o)$  kann nun unter denselben Bedingungen wie zuvor auch aus der ersten Maximumsfunktion herausgezogen werden. Wenn  $(1 - j(t, \tau(o)))$  auch immer größer 0 wäre, könnte dies hierfür ebenso durchgeführt werden. Dafür müsste  $j(t, \tau(o))$  immer kleiner als 1 bleiben. Dies ist auch der Fall, denn es gilt:

$$\text{if} \left( g(t-1, b_o) > 0; j(t, b_o) = \frac{i(t, b_o)}{g(t-1, b_o)}; j(t, b_o) = 0 \right). \quad (3.55)$$

In einem Fall ist  $j(t, \tau(o))=0$ , betrachtet man jetzt noch den anderen Extremfall mit Blick auf die Definition von  $i(t, b_o)$ , so folgt:

$$j(t, b_o) = \frac{i(t, b_o)}{g(t-1, b_o)} \quad (3.56)$$

$$\text{mit } i(t, b_o) < h(t-1, b_o) \quad (3.57)$$

Da die Anzahl Flugzeuge, die sich auf dem Stützpunkt im Freien befindet niemals größer sein wird als die Anzahl der Flugzeuge, welche sich insgesamt auf dem Stützpunkt befindet, wird der Bruch in (3.56) immer kleiner als 1 sein.

Führt man die zuvor angesprochenen Schritte durch und verschiebt die Zeitachse wieder um 1, erhält man für die Differentialgleichung Folgendes:

$$\begin{aligned} \frac{dk(t, o)}{dt} = & k(t, o) \cdot (1 - j(t + 1, \tau(o))) \cdot \max \left( 0; \right. \\ & \left. \max \left( 0; 1 - m(t + 1, o) \cdot p(t, o) \cdot \frac{\alpha_o(t)}{\theta(\rho(o))} \right) \right) - k(t, o). \end{aligned} \quad (3.58)$$

Zwei Maximumsfunktionen direkt hintereinander sind redundant und eine kann weggestrichen werden. Außerdem kann  $k(t, o)$  ausgeklammert werden:

$$\begin{aligned} \frac{dk(t, o)}{dt} = & k(t, o) \cdot \left( (1 - j(t + 1, \tau(o))) \cdot \right. \\ & \left. \max \left( 0; 1 - m(t + 1, o) \cdot p(t, o) \cdot \frac{\alpha_o(t)}{\theta(\rho(o))} \right) - 1 \right). \end{aligned} \quad (3.59)$$

Zuletzt wird noch  $p(t, o)$  in die Gleichung eingesetzt:

$$\begin{aligned} \frac{dk(t, o)}{dt} = & k(t, o) \cdot \left( (1 - j(t + 1, \tau(o))) \right. \\ & \left. \cdot \max \left( 0; 1 - m(t + 1, o) \cdot \frac{\epsilon(\rho(o), \zeta(o)) \cdot \beta_o(t, \zeta(o)) \cdot \alpha_o(t)}{\theta(\rho(o))} \right) - 1 \right). \end{aligned} \quad (3.60)$$

### 3.4.3 Differentialgleichungen für die Anzahl der Boden-Luft-Raketen

Bei dieser Gleichung ist die Variable direkt von sich abhängig. So wird als erstes  $J_f(t)$  aufgestellt:

$$J_f(t) = \max \left( 0, 1; J_f(t - 1) - K_f(t - 1) + \phi_f(t) \right). \quad (3.61)$$

Nach dem Bilden der Differenz zwischen zwei Tagen ergibt sich diese Gleichung:

$$J_f(t) - J_f(t - 1) = \text{Max} \left( 0, 1 - J_f(t - 1); \phi_f(t) - K_f(t - 1) \right). \quad (3.62)$$

Nun wird wieder der Differentialquotient eingesetzt und die Zeitachse verschoben:

$$\frac{dJ_f(t)}{dt} = \max \left( 0, 1 - J_f(t); \phi_f(t + 1) - K_f(t) \right). \quad (3.63)$$

Als nächstes wird noch  $K_f(t)$  in die Gleichung eingesetzt:

$$\frac{dJ_f(t)}{dt} = \max \left( 0, 1 - J_f(t); \phi_f(t+1) - H_a(t) \cdot \frac{I_f(t)}{I_f(t) + I_t(t)} \right). \quad (3.64)$$

Des Weiteren wird noch  $H_a(t)$  hinzugefügt:

$$\frac{dJ_f(t)}{dt} = \max \left( 0, 1 - J_f(t); \phi_f(t+1) - p_4 \cdot S_{SEAD_o}(t) \cdot \frac{I_f(t)}{I_f(t) + I_t(t)} \right). \quad (3.65)$$

Für  $I_f(t) + I_t(t)$  lohnt sich eine gesonderte Betrachtung. Addiert man die Formeln der beiden Terme und stellt man die Gleichung auf, erhält man Folgendes:

$$I_f(t) + I_t(t) = J_f(t) \cdot \sigma_f(t) + L_f(t) + J_t(t) \cdot \sigma_t(t) + L_t(t). \quad (3.66)$$

Jetzt werden noch  $L_f(t)$  und  $L_t(t)$  mit eingefügt:

$$\begin{aligned} I_f(t) + I_t(t) = & J_f(t) \cdot \sigma_f(t) + \frac{p_{12}}{100} \cdot \sum_{b_f} \left[ \psi(b_f) \cdot f(t, b_f) \cdot \xi(t, b_f) \right] \\ & + J_t(t) \cdot \sigma_t(t) + \frac{p_{12}}{100} \cdot \sum_{b_t} \left[ \psi(b_t) \cdot f(t, b_t) \cdot \xi(t, b_t) \right]. \end{aligned} \quad (3.67)$$

Klammert man  $\frac{p_{12}}{100}$  aus so sieht man zwei Summen, einmal über die Stützpunkte von Friends und einmal über die Stützpunkte von Tytan. Addiert man diese beiden, ist dies nichts anderes als die Summe über alle Stützpunkte der Allianz. Zusammengefasst sieht dies wie folgt aus:

$$I_f(t) + I_t(t) = J_f(t) \cdot \sigma_f(t) + J_t(t) \cdot \sigma_t(t) + \frac{p_{12}}{100} \sum_{b_a} \left[ \psi(b_a) \cdot f(t, b_a) \cdot \xi(t, b_a) \right]. \quad (3.68)$$

Zuletzt wird nun diese Gleichung (3.68) und  $I_f(t)$  in die Differentialgleichung eingesetzt:

$$\begin{aligned} \frac{dJ_f(t)}{dt} = & \max \left( 0, 1 - J_f(t); \phi_f(t+1) - p_4 \cdot S_{SEAD_o}(t) \right. \\ & \left. \cdot \frac{J_f(t) \cdot \sigma_f(t) + \frac{p_{12}}{100} \sum_{b_f} \left[ \psi(b_f) \cdot f(t, b_f) \cdot \xi(t, b_f) \right]}{J_f(t) \cdot \sigma_f(t) + J_t(t) \cdot \sigma_t(t) + \frac{p_{12}}{100} \sum_{b_a} \left[ \psi(b_a) \cdot f(t, b_a) \cdot \xi(t, b_a) \right]} \right). \end{aligned} \quad (3.69)$$

Für die Boden-Luft-Raketen von Tytan gilt das Pendant von der Formel für Friends, wobei die Indizes  $f$  und  $t$  miteinander vertauscht werden:

$$\frac{dJ_t(t)}{dt} = \max \left( 0, 1 - J_t(t); \phi_t(t+1) - p_4 \cdot S_{SEAD_o}(t) \cdot \frac{J_t(t) \cdot \sigma_t(t) + \frac{p_{12}}{100} \sum_{b_t} [\psi(b_t) \cdot f(t, b_t) \cdot \xi(t, b_t)]}{J_f(t) \cdot \sigma_f(t) + J_t(t) \cdot \sigma_t(t) + \frac{p_{12}}{100} \sum_{b_a} [\psi(b_a) \cdot f(t, b_a) \cdot \xi(t, b_a)]} \right). \quad (3.70)$$

#### 3.4.4 Hilfsvariablen

Die vier Differentialgleichungen enthalten vier Variablen, welche noch näher beschrieben werden müssen, da sie noch nicht komplett auf Parameter zurückgeführt wurden. Es geht dabei um  $j(t, b_o)$ ,  $m(t, o)$ ,  $A_o(t)$  und  $S_{SEAD_o}(t)$ . Schrittweise werden alle Variablen in den Formeln ersetzt, bis sie nur noch durch bekannte Parameter definiert sind. Die Summen der geflogenen Missionen werden dabei zum Schluss betrachtet. Beginnend mit  $A_o(t)$  sieht die Gleichung folgendermaßen aus.

$$A_o(t) = \min \left( p_9; \frac{B_o(t) + F_o(t)}{S_{PEN_o}(t)} \right). \quad (3.71)$$

$B_o(t)$  und  $F_o(t)$  sollen dabei einzeln betrachtet werden, da sie sehr ausführlich sind. Setzt man direkt  $C_a(t)$  in  $B_o(t)$  ein, entsteht diese Gleichung:

$$B_o(t) = p_2 \cdot p_8^{(S_{AI_o}(t))} \cdot D_a(t). \quad (3.72)$$

Durch Einsetzen von  $D_a(t)$  folgt:

$$B_o(t) = p_2 \cdot p_8^{(S_{AI_o}(t))} \cdot p_7^{(S_{ESM_o}(t))} \cdot E_a(t). \quad (3.73)$$

Durch Einsetzen von  $E_a(t)$  folgt:

$$B_o(t) = p_2 \cdot p_8^{(S_{AI_o}(t))} \cdot p_7^{(S_{ESM_o}(t))} \cdot p_6^{(S_{AEW_a}(t))} \cdot S_{DCA_a}(t). \quad (3.74)$$

Betrachtet wird nun  $F_o(t)$ . Direktes Einsetzen von  $G_a(t)$  in  $F_o(t)$  liefert folgende Gleichung:

$$F_o(t) = p_5 \cdot p_8^{(S_{AI_o}(t))} \cdot p_7^{(S_{ESM_o}(t))} \cdot \max \left( 0, 1; I_f(t) + I_t(t) - H_a(t) \right). \quad (3.75)$$

$I_f(t) + I_t(t)$  wurde schon ausführlich in Gleichung (3.68) beschrieben. Setzt man diese zusammen mit  $H_a(t)$  ein, ergibt sich folglich dieses Endprodukt:

$$F_o(t) = p_5 \cdot p_8^{(S_{AI_o}(t))} \cdot p_7^{(S_{ESM_o}(t))} \cdot \max \left( 0, 1; J_f(t) \cdot \sigma_f(t) + J_t(t) \cdot \sigma_t(t) \right. \\ \left. + \frac{p_{12}}{100} \sum_{b_a} \left[ \psi(b_a) \cdot f(t, b_a) \cdot \xi(t, b_a) \right] - p_4 \cdot S_{SEAD_o}(t) \right). \quad (3.76)$$

Da  $A_o(t)$  jetzt bis zum Ende verfolgt wurde, wird sich nun  $j(t, b_o)$  gewidmet. Da in der Differentialgleichung für die Flugzeuge statt  $j(t, b_o)$  dieser Wert  $j(t+1, b_o)$  gebraucht wird, soll damit zum Zwecke der Kontinuität auch weitergearbeitet werden:

$$j(t+1, b_o) = \begin{cases} \frac{i(t+1, b_o)}{g(t, b_o)} & ; \text{wenn } g(t, b_o) > 0, \\ 0 & ; \text{sonst.} \end{cases} \quad (3.77)$$

Im Weiteren soll nur der Fall  $g(t, b_o) > 0$  betrachtet werden. In einem Schritt lässt sich  $i(t+1, b_o)$  und  $g(t, b_o)$  einsetzen:

$$j(t+1, b_o) = \frac{\max \left( 0; \min \left( h(t, b_o); h(t, b_o) \cdot \frac{p_{14}}{100} \cdot (f(t, b_o) - f(t+1, b_o)) \right) \right)}{\sum_{o|\tau(o)=b_o} k(t, o)}. \quad (3.78)$$

Es fällt auf, dass  $(f(t, b_o) - f(t+1, b_o))$  dem negativen Differentialquotienten für den Status der Stützpunkte entspricht. Dieser wird eingesetzt und ebenso wird  $h(t, b_o)$  aus der Minimumsfunktion rausmultipliziert:

$$j(t+1, b_o) = \frac{\max \left( 0; h(t, b_o) \cdot \min \left( 1; \frac{p_{14}}{100} \cdot \left( -\frac{df(t, b_o)}{dt} \right) \right) \right)}{\sum_{o|\tau(o)=b_o} k(t, o)}. \quad (3.79)$$

Da  $h(t, b_o)$  laut Definition immer größer als 0 ist, kann es ohne Probleme auch aus der Maximumsfunktion herausgezogen werden. Setzt man im selben Zug auch die Formel für  $h(t, b_o)$  ein, ergibt sich dieses Ergebnis:

$$j(t+1, b_o) = \frac{\max \left( 0; g(t, b_o) - \mu(b_o) \right)}{\sum_{o|\tau(o)=b_o} k(t, o)} \cdot \max \left( 0; \min \left( 1; \frac{p_{14}}{100} \cdot \left( -\frac{df(t, b_o)}{dt} \right) \right) \right). \quad (3.80)$$

Zuletzt bleibt jetzt nur noch ein zweites Mal  $g(t, b_o)$  zu ersetzen:

$$j(t+1, b_o) = \frac{\max\left(0; \sum_{o|\tau(o)=b_o} [k(t, o)] - \mu(b_o)\right)}{\sum_{o|\tau(o)=b_o} k(t, o)} \cdot \max\left(0; \min\left(1; \frac{p_{14}}{100} \cdot \left(-\frac{df(t, b_o)}{dt}\right)\right)\right). \quad (3.81)$$

Als nächstes wird die Aufschlüsselung von  $m(t, o)$  in Angriff genommen. Auch hier wird in der Differentialgleichung  $m(t+1, o)$  verwendet, weshalb ab jetzt damit weiter gearbeitet wird. Setzt man  $n(t, o)$  in  $m(t+1, o)$  ein, ergibt sich folgende Gleichung:

$$m(t+1, o) = i_{PEN}(t, o) \cdot A_o(t) + i_{CAS}(t, o) \cdot \delta_o(t) + i_{DCA}(t, o) \cdot M_o(t). \quad (3.82)$$

$A_o(t)$  wurde schon im oberen Abschnitt beschrieben und  $M_o(t)$  lässt sich folgendermaßen ersetzen:

$$m(t+1, o) = i_{PEN}(t, o) \cdot A_o(t) + i_{CAS}(t, o) \cdot \delta_o(t) + i_{DCA}(t, o) \cdot \min\left(p_9; \frac{N_o(t)}{S_{DCA_o}(t)}\right). \quad (3.83)$$

Jetzt bleibt nur noch  $N_o(t)$  als Unbekannte übrig und durch Einsetzen ergibt sich Folgendes:

$$m(t+1, o) = i_{PEN}(t, o) \cdot A_o(t) + i_{CAS}(t, o) \cdot \delta_o(t) + i_{DCA}(t, o) \cdot \min\left(p_9; \frac{p_3 \cdot (S_{ESC_a}(t) + S_{PwESC_a}(t))}{S_{DCA_o}(t)}\right). \quad (3.84)$$

Die letzten zu untersuchenden Hilfsvariablen sind diejenigen, die jeweils die Summe der Flüge eines Missionstypen angeben. Für die Missionstypen AEW, DCA, OCA, ESC, ESM, SEAD, und AI ist es der gleiche Fall, abgesehen von den Missionsindikatoren. Deshalb wird hier beispielhaft nur  $S_{SEAD_o}(t)$  betrachtet:

$$S_{SEAD_o}(t) = \sum_o [i_{SEAD}(t, o) \cdot r(t, o)]. \quad (3.85)$$

$r(t, o)$  lässt sich leicht ersetzen und es folgt:

$$S_{SEAD_o}(t) = \sum_o [i_{SEAD}(t, o) \cdot \eta(\rho(o), \zeta(o)) \cdot q(t, o)]. \quad (3.86)$$

$q(t, o)$  wird noch einmal einzeln aufgestellt, da es für alle Missionstypen die gleiche Formel ist:

$$q(t, o) = k(t, o) \cdot f(t, \tau(o)) \cdot \frac{p(t, o) \cdot \alpha_o(t)}{100}. \quad (3.87)$$

Durch Einsetzen von  $p(t, o)$  folgt:

$$q(t, o) = k(t, o) \cdot f(t, \tau(o)) \cdot \frac{\epsilon(\rho(o), \zeta(o)) \cdot \beta_o(t, \zeta(o)) \cdot \alpha_o(t)}{100}. \quad (3.88)$$

$S_{PEN_o}(t)$  ist hiermit fertig untersucht, da nur noch  $q(t, o)$  als Unbekannte vorhanden ist und diese soeben aufgeschlüsselt wurde. Somit steht nur noch die Variable  $S_{PwESC_a}(t)$  aus:

$$S_{PwESC_a}(t) = \sum_a [u(t, a)]. \quad (3.89)$$

Durch Einsetzen von  $u(t, o)$  folgt:

$$S_{PwESC_a}(t) = \sum_a [i_{PwESC}(t, a) \cdot q(t, a) \cdot \lambda(\rho(a)) \cdot p_{13}]. \quad (3.90)$$

Auch hier bleibt jetzt nur noch  $q(t, o)$  übrig. Somit sind jetzt alle Differentialgleichungen und Hilfsvariablen ausreichend beschrieben, um das Modell in seiner Gesamtheit aufzustellen.

### 3.4.5 Gesamtmodell

Im vorangegangenen Teil wurden alle notwendigen Gleichungen aufgestellt, um das Modell zusammenzufassen. Für die Zusammenfassung werden jetzt alle Gleichungen für beide Seiten herangezogen. Um dies zu bewerkstelligen, müssen für die Formeln der Alliance bei den bisher gezeigten Formeln folgende Indizes miteinander vertauscht werden:  $o$  mit  $a$ ,  $k$  mit  $f$ , und  $s$  mit  $t$ . Das Modell setzt sich aus den folgenden Gleichungen zusammen: Der Differentialgleichung für den Status der Stützpunkte (3.52), der Differentialgleichung für die Anzahl der Flugzeuge (3.60), den Differentialgleichungen für die Anzahl an SAM FU's (3.69, 3.70) und den Hilfsvariablen Basic Penetrator Attrition (3.71), Potential Number of Penetrators Killed by DCA (3.74), Potential Number of Penetrators Killed by SAM (3.76), Percentage Aircraft on Base KOG (3.77, 3.81), Base Attrition Previous Day (3.84), Summe der Sorties (3.86, 3.90) und Actual Sorties (3.88). Möchte man Formeln von über zwei Zeilen vermeiden, so sieht das LAMBDA-Modell in seiner Gesamtheit und in seiner komprimiertesten Form wie folgt aus:

$$\begin{aligned} \frac{df(t, b_o)}{dt} = & \min \left( 100 - f(t, b_o); \omega(t+1, b_o) - \min \left( f(t, b_o); \right. \right. \\ & \left. \left. \left( \frac{\xi(t, b_o) \cdot 100}{\pi(b_o)} \cdot (p_1 \cdot S_{OCA_a}(t) + p_{11} \cdot \gamma_a(t)) \cdot (1 - A_o(t) \cdot p_{10}) \right) \right) \right), \end{aligned} \quad (3.91)$$

$$\begin{aligned} \frac{df(t, b_a)}{dt} = & \min \left( 100 - f(t, b_a); \omega(t+1, b_a) - \min \left( f(t, b_a); \right. \right. \\ & \left. \left. \left( \frac{\xi(t, b_a) \cdot 100}{\pi(b_a)} \cdot (p_1 \cdot S_{OCA_o}(t) + p_{11} \cdot \gamma_o(t)) \cdot (1 - A_a(t) \cdot p_{10}) \right) \right) \right), \end{aligned} \quad (3.92)$$

$$\begin{aligned} \frac{dk(t, o)}{dt} = & k(t, o) \cdot \left( (1 - j(t+1, \tau(o))) \right. \\ & \left. \cdot \max \left( 0; 1 - m(t+1, o) \cdot \frac{\epsilon(\rho(o), \zeta(o)) \cdot \beta_o(t, \zeta(o)) \cdot \alpha_o(t)}{\theta(\rho(o))} \right) - 1 \right), \end{aligned} \quad (3.93)$$

$$\begin{aligned} \frac{dk(t, a)}{dt} = & k(t, a) \cdot \left( (1 - j(t+1, \tau(a))) \right. \\ & \left. \cdot \max \left( 0; 1 - m(t+1, a) \cdot \frac{\epsilon(\rho(a), \zeta(a)) \cdot \beta_a(t, \zeta(a)) \cdot \alpha_a(t)}{\theta(\rho(a))} \right) - 1 \right), \end{aligned} \quad (3.94)$$

$$\begin{aligned} \frac{dJ_f(t)}{dt} = & \max \left( 0, 1 - J_f(t); \phi_f(t+1) - p_4 \cdot S_{SEAD_o}(t) \right. \\ & \left. \cdot \frac{J_f(t) \cdot \sigma_f(t) + \frac{p_{12}}{100} \sum_{b_f} \left[ \psi(b_f) \cdot f(t, b_f) \cdot \xi(t, b_f) \right]}{J_f(t) \cdot \sigma_f(t) + J_t(t) \cdot \sigma_t(t) + \frac{p_{12}}{100} \sum_{b_a} \left[ \psi(b_a) \cdot f(t, b_a) \cdot \xi(t, b_a) \right]} \right), \end{aligned} \quad (3.95)$$

$$\begin{aligned} \frac{dJ_t(t)}{dt} = & \max \left( 0, 1 - J_t(t); \phi_t(t+1) - p_4 \cdot S_{SEAD_o}(t) \right. \\ & \left. \cdot \frac{J_t(t) \cdot \sigma_t(t) + \frac{p_{12}}{100} \sum_{b_t} \left[ \psi(b_t) \cdot f(t, b_t) \cdot \xi(t, b_t) \right]}{J_f(t) \cdot \sigma_f(t) + J_t(t) \cdot \sigma_t(t) + \frac{p_{12}}{100} \sum_{b_a} \left[ \psi(b_a) \cdot f(t, b_a) \cdot \xi(t, b_a) \right]} \right), \end{aligned} \quad (3.96)$$

$$\begin{aligned} \frac{dJ_k(t)}{dt} = & \max \left( 0, 1 - J_k(t); \phi_k(t+1) - p_4 \cdot S_{SEAD_a}(t) \right. \\ & \left. \cdot \frac{J_k(t) \cdot \sigma_k(t) + \frac{p_{12}}{100} \sum_{b_k} \left[ \psi(b_k) \cdot f(t, b_k) \cdot \xi(t, b_k) \right]}{J_k(t) \cdot \sigma_k(t) + J_s(t) \cdot \sigma_s(t) + \frac{p_{12}}{100} \sum_{b_o} \left[ \psi(b_o) \cdot f(t, b_o) \cdot \xi(t, b_o) \right]} \right), \end{aligned} \quad (3.97)$$

$$\frac{dJ_s(t)}{dt} = \max \left( 0, 1 - J_s(t); \phi_s(t+1) - p_4 \cdot S_{SEAD_a}(t) \right. \\ \left. \cdot \frac{J_s(t) \cdot \sigma_s(t) + \frac{p_{12}}{100} \sum_{b_s} \left[ \psi(b_s) \cdot f(t, b_s) \cdot \xi(t, b_s) \right]}{J_k(t) \cdot \sigma_k(t) + J_s(t) \cdot \sigma_s(t) + \frac{p_{12}}{100} \sum_{b_o} \left[ \psi(b_o) \cdot f(t, b_o) \cdot \xi(t, b_o) \right]} \right), \quad (3.98)$$

$$A_o(t) = \min \left( p_9; \frac{B_o(t) + F_o(t)}{S_{PEN_o}(t)} \right) \quad \text{und} \quad A_a(t) = \min \left( p_9; \frac{B_a(t) + F_a(t)}{S_{PEN_a}(t)} \right), \quad (3.99)$$

$$B_o(t) = p_2 \cdot p_8^{(S_{AI_o}(t))} \cdot p_7^{(S_{ESM_o}(t))} \cdot p_6^{(S_{AEW_a}(t))} \cdot S_{DCA_a}(t), \quad (3.100)$$

$$B_a(t) = p_2 \cdot p_8^{(S_{AI_a}(t))} \cdot p_7^{(S_{ESM_a}(t))} \cdot p_6^{(S_{AEW_o}(t))} \cdot S_{DCA_o}(t), \quad (3.101)$$

$$F_o(t) = p_5 \cdot p_8^{(S_{AI_o}(t))} \cdot p_7^{(S_{ESM_o}(t))} \cdot \max \left( 0, 1; J_f(t) \cdot \sigma_f(t) + J_t(t) \cdot \sigma_t(t) \right. \\ \left. + \frac{p_{12}}{100} \sum_{b_a} \left[ \psi(b_a) \cdot f(t, b_a) \cdot \xi(t, b_a) \right] - p_4 \cdot S_{SEAD_o}(t) \right), \quad (3.102)$$

$$F_a(t) = p_5 \cdot p_8^{(S_{AI_a}(t))} \cdot p_7^{(S_{ESM_a}(t))} \cdot \max \left( 0, 1; J_k(t) \cdot \sigma_k(t) + J_s(t) \cdot \sigma_s(t) \right. \\ \left. + \frac{p_{12}}{100} \sum_{b_o} \left[ \psi(b_o) \cdot f(t, b_o) \cdot \xi(t, b_o) \right] - p_4 \cdot S_{SEAD_a}(t) \right), \quad (3.103)$$

$$j(t+1, b_o) = \begin{cases} \frac{\max \left( 0; \sum_{o|\tau(o)=b_o} \left[ k(t, o) \right] - \mu(b_o) \right)}{\sum_{o|\tau(o)=b_o} k(t, o)} & ; \text{wenn } g(t, b_o) > 0, \\ \cdot \max \left( 0; \min \left( 1; \frac{p_{14}}{100} \cdot \left( -\frac{df(t, b_o)}{dt} \right) \right) \right) & \\ 0 & ; \text{sonst,} \end{cases} \quad (3.104)$$

$$j(t+1, b_a) = \begin{cases} \frac{\max \left( 0; \sum_{a|\tau(a)=b_a} \left[ k(t, a) \right] - \mu(b_a) \right)}{\sum_{a|\tau(a)=b_a} k(t, a)} & ; \text{wenn } g(t, b_a) > 0, \\ \cdot \max \left( 0; \min \left( 1; \frac{p_{14}}{100} \cdot \left( -\frac{df(t, b_a)}{dt} \right) \right) \right) & \\ 0 & ; \text{sonst,} \end{cases} \quad (3.105)$$

$$\begin{aligned}
m(t+1, o) &= i_{PEN}(t, o) \cdot A_o(t) + i_{CAS}(t, o) \cdot \delta_o(t) \\
&\quad + i_{DCA}(t, o) \cdot \min \left( p_9; \frac{p_3 \cdot (S_{ESC_o}(t) + S_{PwESC_o}(t))}{S_{DCA_o}(t)} \right), \tag{3.106}
\end{aligned}$$

$$\begin{aligned}
m(t+1, a) &= i_{PEN}(t, a) \cdot A_a(t) + i_{CAS}(t, a) \cdot \delta_a(t) \\
&\quad + i_{DCA}(t, a) \cdot \min \left( p_9; \frac{p_3 \cdot (S_{ESC_a}(t) + S_{PwESC_a}(t))}{S_{DCA_a}(t)} \right), \tag{3.107}
\end{aligned}$$

$$S_{AEW_o}(t) = \sum_o \left[ i_{AEW}(t, o) \cdot \eta(\rho(o), \zeta(o)) \cdot q(t, o) \right], \tag{3.108}$$

$$S_{AEW_a}(t) = \sum_a \left[ i_{AEW}(t, a) \cdot \eta(\rho(a), \zeta(a)) \cdot q(t, a) \right], \tag{3.109}$$

$$S_{DCA_o}(t) = \sum_o \left[ i_{DCA}(t, o) \cdot \eta(\rho(o), \zeta(o)) \cdot q(t, o) \right], \tag{3.110}$$

$$S_{DCA_a}(t) = \sum_a \left[ i_{DCA}(t, a) \cdot \eta(\rho(a), \zeta(a)) \cdot q(t, a) \right], \tag{3.111}$$

$$S_{OCA_o}(t) = \sum_o \left[ i_{OCA}(t, o) \cdot \eta(\rho(o), \zeta(o)) \cdot q(t, o) \right], \tag{3.112}$$

$$S_{OCA_a}(t) = \sum_a \left[ i_{OCA}(t, a) \cdot \eta(\rho(a), \zeta(a)) \cdot q(t, a) \right], \tag{3.113}$$

$$S_{ESC_o}(t) = \sum_o \left[ i_{ESC}(t, o) \cdot \eta(\rho(o), \zeta(o)) \cdot q(t, o) \right], \tag{3.114}$$

$$S_{ESC_a}(t) = \sum_a \left[ i_{ESC}(t, a) \cdot \eta(\rho(a), \zeta(a)) \cdot q(t, a) \right], \tag{3.115}$$

$$S_{ESM_o}(t) = \sum_o \left[ i_{ESM}(t, o) \cdot \eta(\rho(o), \zeta(o)) \cdot q(t, o) \right], \tag{3.116}$$

$$S_{ESM_a}(t) = \sum_a \left[ i_{ESM}(t, a) \cdot \eta(\rho(a), \zeta(a)) \cdot q(t, a) \right], \tag{3.117}$$

$$S_{SEAD_o}(t) = \sum_o \left[ i_{SEAD}(t, o) \cdot \eta(\rho(o), \zeta(o)) \cdot q(t, o) \right], \tag{3.118}$$

$$S_{SEAD_a}(t) = \sum_a \left[ i_{SEAD}(t, a) \cdot \eta(\rho(a), \zeta(a)) \cdot q(t, a) \right], \tag{3.119}$$

$$S_{AI_o}(t) = \sum_o \left[ i_{AI}(t, o) \cdot \eta(\rho(o), \zeta(o)) \cdot q(t, o) \right], \tag{3.120}$$

$$S_{AI_a}(t) = \sum_a \left[ i_{AI}(t, a) \cdot \eta(\rho(a), \zeta(a)) \cdot q(t, a) \right], \quad (3.121)$$

$$S_{PwESC_o}(t) = \sum_o \left[ i_{PwESC}(t, o) \cdot q(t, o) \cdot \lambda(\rho(o)) \cdot p_{13} \right], \quad (3.122)$$

$$S_{PwESC_a}(t) = \sum_a \left[ i_{PwESC}(t, a) \cdot q(t, a) \cdot \lambda(\rho(a)) \cdot p_{13} \right], \quad (3.123)$$

$$S_{PEN_o}(t) = \sum_o \left[ i_{PEN}(t, o) \cdot q(t, o) \right], \quad (3.124)$$

$$S_{PEN_a}(t) = \sum_a \left[ i_{PEN}(t, a) \cdot q(t, a) \right], \quad (3.125)$$

$$q(t, o) = k(t, o) \cdot f(t, \tau(o)) \cdot \frac{\epsilon(\rho(o), \zeta(o)) \cdot \beta_o(t, \zeta(o)) \cdot \alpha_o(t)}{100}, \quad (3.126)$$

$$q(t, a) = k(t, a) \cdot f(t, \tau(a)) \cdot \frac{\epsilon(\rho(a), \zeta(a)) \cdot \beta_a(t, \zeta(a)) \cdot \alpha_a(t)}{100}. \quad (3.127)$$

### 3.5 Analyse des Modells

Zuerst fällt auf, dass keine Summe der CAS Missionen in den Formeln vertreten ist. Der CAS Indikator  $i_{CAS}(t, o/a)$  taucht nur in Kombination mit der CAS Attrition  $\delta_o(t)$  an den Stellen (3.106) und (3.107) auf, an welchen die Abnutzung für die jeweilig geflogene Mission mit in die Gleichungen einfließt. Dies bedeutet, dass die eigenen Truppen durch geflogene CAS Missionen nur eine Abnutzung erfahren, jedoch keine Effekte dadurch erzielt werden. Es sei gesagt, dass an anderer Stelle im LAMBDA-Tool jedoch vermerkt ist, welcher Schaden durch CAS Missionen beim Gegner verursacht wird. Jedoch stehen diese Werte für sich und werden an keiner Stelle mit einbezogen, was die Veränderung der Truppenstärke angeht. Dies legt die Vermutung nahe, dass es sich bei den CAS Missionen um ein Relikt eines vorher vorhandenen Mechanismus oder einen noch nicht fertig implementierten Einflussfaktor handelt.

Des Weiteren ist in den Formeln für die Abnutzung der Stützpunktzustände (3.91) und (3.92) zu erkennen, dass die eigenen Penetratorverluste  $A_o(t)/A_a(t)$  Einfluss auf den Schaden an den eigenen Stützpunkten haben. Eigentlich soll der Formelteil  $A_o(t) \cdot p_{10}$  den Anteil der zerstörten OCA Sorties angeben, welche schon im Anflug zerstört werden und somit keinen Schaden mehr anrichten. Der Ausdruck  $(1 - A_o(t) \cdot p_{10})$  gibt somit den überlebenden Anteil an, welcher Schaden anrichtet. Bei dem jetzigen Stand der Formel richten gegnerische OCA Sorties an den eigenen Stützpunkten mehr Schaden an, je mehr

eigene OCA Flieger bis zum Eintreffen überleben. Dies würde bedeuten, dass man seine komplette Offensive zerstören lassen könnte, um seine eigenen Stützpunkte zu schützen. Es ist offensichtlich, dass dieser Effekt so nicht beabsichtigt ist. An dieser Stelle müssten  $A_o(t)/A_a(t)$  in den beiden Formeln vertauscht werden, um einen richtigen Einfluss von  $p_{10}$  zu gewährleisten.

Da das in Vensim implementierte Modell, wie in Kapitel 4 zu sehen, auch dieselben Werte liefert wie das LAMBA-Tool, ist auszuschließen, dass es sich bei den oben angesprochenen Punkten um Auswertungsfehler handelt.

Nach dem Gesetz von Lanchester ist bei einer Schlacht die Truppenstärke der entscheidendere Faktor als die Feuerkraft, da die Anzahl der Truppen einen quadratischen Einfluss hat. Das LAMBDA-Modell ließe sich auch darauf reduzieren, welcher Aspekt den größeren Einfluss hat, wobei sich die Feuerkraft in diesem Fall aus einer Vielzahl von Parametern zusammensetzen würde. Allerdings müssten dabei einige Fallunterscheidungen getroffen werden, je nachdem, welche Missionen geflogen werden, da diese unterschiedlich in die Formeln eingehen. Diese Fallunterscheidungen sind zugleich auch ein Indiz dafür, was einen noch viel größeren Einfluss als die Truppenstärke und die Feuerkraft hat. Die Allokation der Truppen bringt die Entscheidung. Wenn der Gegner offensiv fliegt, kann der eigene Schaden an den Stützpunkten minimiert werden, indem eine gute Defensive aufgestellt wird, da, wie oben angesprochen, die zerstörten Angreifer keinen Schaden mehr anrichten. Der dennoch angerichtete Schaden könnte durch die Repair Capacity  $\omega(t, b)$  ausgeglichen werden, sodass die eigene Seite diesen Tag unbeschadet übersteht, während der Gegenseite stark zugesetzt wurde. Ersetzt der Gegner aber viele der offensiven Flüge durch Eskorten und elektronische Unterstützungsflüge, so wird die eigene Defensive dezimiert. An diesem Punkt könnten die eigenen Truppen aber wieder offensiv eingesetzt werden, da der Gegner keine Defensive besitzt und die geringe Offensive kaum Schaden anrichten würde. Es lässt sich erkennen, dass dieses Szenario somit auf ein klassisches Stein-Schere-Papier Prinzip hinaus läuft. Welche Mission dabei gegen welche andere Mission wirksam ist, wurde schon in Kapitel 2.3 besprochen. Entscheidend ist also nicht mehr, dass möglichst viele der eigenen Truppen eingesetzt werden, sondern wie die Truppen eingesetzt werden, die zur Verfügung stehen. Dem Wissen über gegnerische Aktivitäten wird somit eine sehr hohe Bedeutung zugeordnet.



Dies ist wichtig, um die Ergebnisse für alle Situationen mit den Ergebnissen des LAMBDA-Tools vergleichen zu können. Dabei werden OPFOR die Schwadronen 1 und 2 zugewiesen, deren Namen entsprechend Stützpunkt 1 und 2 zugeteilt werden. Für die Alliance sind das demzufolge Schwadronen 3 und 4. Zum anderen sind die Parameter des Modells über die Zeit nicht veränderlich. Dies ist hauptsächlich dem Umstand geschuldet, dass für diese Arbeit Vensim PLE, eine Version für akademische Verwendung, benutzt wird, welche eine solche Funktion für veränderliche Parameter nicht enthält. Mit einer Vollversion von Vensim ist dies jedoch leicht zu implementieren, ohne an dem Modell etwas zu verändern.

### Anzahl der Flugzeuge in Schwadron 1

	LAMBDA-Tool	Explizites Euler Verfahren			Klassisches Runge-Kutta-Verfahren		
Schrittweite	1	1	0,5	0,25	1	0,5	0,25
Zeit (Tage)							
0	80	80	80	80	80	80	80
0,25				75,92474			76,14371
0,5			71,84949	72,2887		72,66962	72,66964
0,75				69,00682			69,5094
1	63,69899	63,69899	65,34397	66,01514	66,60992	66,6102	66,61022
1,25				63,26454			63,93069
1,5			59,91567	60,71664		61,43779	61,43782
1,75				58,34094			59,105
2	53,19704	53,19704	55,23685	56,1129	56,91005	56,91046	56,91048
2,25				54,0125			54,83636
2,5			51,10325	52,02324		52,86765	52,86768
2,75				50,13138			50,9919
3	45,23596	45,23596	47,37999	48,32537	49,19784	49,19834	49,19837
3,25				46,59543			47,47803
3,5			43,97391	44,93322		45,82304	45,82307
3,75				43,33154			44,22675
4	38,65945	38,65945	40,81823	41,78419	42,68257	42,68316	42,6832
4,25				40,28577			41,18729
4,5			37,86363	38,83155		39,73448	39,73452
4,75				37,41735			38,32089
5	32,9298	32,9298	35,07272	36,03951	36,94217	36,94286	36,94289
5,25				34,69476			35,59737
5,5			32,41655	33,3802		34,28148	34,28151
5,75				32,09321			32,9928
6	27,75509	27,75509	29,87228	30,83146	31,72855	31,72887	31,72891
6,25				29,59311			30,48725
6,5			27,42233	28,37547		29,2652	29,26526
6,75				27,17607			28,06062
7	22,96067	22,96067	25,04959	25,99265	26,87061	26,87113	26,8712

Abbildung 4.2: Vergleich der Simulationsergebnisse von LAMBDA-Tool und Vensim-Modell.

Des Weiteren wird die Annahme getroffen, dass Flugzeuge entsprechend ihrer Befähigung eingesetzt werden und dementsprechend die bestmöglichen Werte wie Effektivität und Überlebenschance besitzen. Dies ermöglicht es, im Modell nur einen Wert für jeden dieser Flugzeugparameter nutzen zu können, da die Werte für alle Schwadronen gleich sind.

An dem Ausschnitt des fertigen Modells in Abbildung 4.1 ist gut zu erkennen, wodurch das LAMBDA-Modell so komplex wird. Die Verluste von Schwadron 1 sind von jedem anderen Schwadron und jedem Stützpunkt abhängig. Da dies auch für den Verlust von jedem anderen Schwadron gilt, entsteht ein sehr verworrenes Netzwerk. Dieser Umstand ist auch der Grund für die getroffenen Annahmen für Vereinfachungen, da jedes zusätzliche Schwadron die Anzahl an Verbindungen quadratisch in die Höhe treibt.

Mit dem fertigen Modell werden anschließend noch Simulationsläufe zum Vergleich mit dem LAMBDA-Modell durchgeführt, um die Auswertungen zu verifizieren. Ein Beispiel ist in der Abbildung 4.2 zu sehen. Aus Übersichtszwecken ist nur die Entwicklung der Anzahl der Flugzeuge von Schwadron 1 abgebildet, da die Grafik die Effekte der unterschiedlichen Lösungsverfahren und Schrittweiten darstellen soll und der dabei zu erkennende Trend für alle Schwadronen der gleiche ist. Für diesen vergleichenden Simulationslauf muss das LAMBDA-Tool an das Vensim-Modell angepasst werden. Die Schwadronen werden dort auf zwei pro Fraktion reduziert und es werden in beiden Programmen folgende Parameter verwendet:

Effect of one standard OCA Sortie	0,05
Effect of one standard DCA Sortie	0,06
Effect of one standard ESC Sortie	0,05
Effect of one standard SEAD Sortie	0,20
Attrition from one SAM FU	0,35
Multiplicative Effect of one AEW Sortie	1,10
Multiplicative Effect of one ESM Sortie	0,99
Multiplicative Effect of one AI/Counter C2 Sortie	0,99
Maximum Penetrator or DCA Attrition	0,30
Fraction of Penetrator Losses on Ingress	0,80
Airbases killed by one TLAM	0,03
Multiplicative Factor for Point Defence SAM Use	0,80
Multiplicative Factor for Self Defence Virtual Sorties	0,40
Weather Factor OPFOR/Alliance	0,75
Surge Factor Sq1/Sq2/Sq3/Sq4	1,00
SSM launched OPFOR/Alliance	1
CAS Attrition OPFOR	0,30
CAS Attrition Alliance	0,10
Sortie Rate	1,50
Relative Effectiveness	0,50
Relative Survivability	1,00
Self Defence Factor	0,50

Shelters Base1/Base2/Base3/Base4	0
Size Base1/Base2/Base3/Base4	2
AREA SAM % effective this day OPFOR/Alliance	0,20
AREA SAM Reinforcements OPFOR/Alliance	0
Point Defence Score Base1/Base2/Base3	10
Point Defence Score Base4	25
Repair Capacity Base1/Base2/Base3/Base4	5
Enemy OCA WT Base1	0,10
Enemy OCA WT Base2	0,90
Enemy OCA WT Base3	0,50
Enemy OCA WT Base4	0,50
Mission of Squadron1	ESC
Mission of Squadron2	OCA
Mission of Squadron3	DCA
Mission of Squadron4	AEW
Initial Number of SAM FU's OPFOR/Alliance	10
Initial Number of Aircraft in Squadron1	80
Initial Number of Aircraft in Squadron2	20
Initial Number of Aircraft in Squadron3	30
Initial Number of Aircraft in Squadron4	50

An dem Simulationslauf ist zu erkennen, was auch für alle anderen Simulationsläufe gilt: Das Vensim Modell liefert mittels des expliziten Euler-Verfahrens dieselben Werte wie das LAMBDA-Tool. Mit kleiner werdender Schrittweite reduziert sich die Anzahl der Flugzeuge auch langsamer. Dies war beim Euler-Verfahren zu erwarten, da bei einer Kurve mit negativer Steigung und positiver Krümmung für zu lange Zeiträume eine zu hohe Steigung angenommen wird. Werden die Zeitschritte kleiner, so wird die Steigung auch früher auf den niedrigeren Wert angepasst. Das wesentlich genauere klassische Runge-Kutta-Verfahren liefert schon bei einer Schrittweite von 1 die höheren Werte und verbessert sich mit kleiner werdender Schrittweite auch nur noch geringfügig.

## 5 Optimierung von Gefechten in AIMMS

Nach der Aufstellung eines Modells ist besonders die Optimierung bestimmter Aspekte von Interesse. Im Falle des LAMBDA-Modells wäre das Wissen über die optimale Truppenallokation ein enormer Gewinn, da sie für das Militär eine hohe strategische Bedeutung hat. Beim LAMBDA-Modell gäbe es dabei mehrere Ansatzpunkte. Welche Schwadronenanzahl wäre optimal, auf welchen Stützpunkten stationiert man die Schwadronen am besten oder wie setzt man seine Truppen optimalerweise ein, um die Schlacht zu gewinnen. Der letzte Punkt ist dabei von größtem Interesse, da er einen entscheidenden Einfluss auf den Verlust der Truppen hat. Im LAMBDA-Modell eignen sich zwei Parameter für diesen Ansatz: Die Missionsindikatoren, die aussagen, für welche Mission eine Schwadron eingesetzt wird, und der Parameter Enemy OCA WT ( $\xi(t, b)$ ), der bestimmt, mit welchem Anteil der eigenen Offensive jeder generische Stützpunkt angegriffen wird. Aus Zeitgründen soll in dieser Arbeit das Optimierungsproblem nur im Ansatz erforscht werden und Ziel ist es zu klären, ob solche Gefechtsoptimierungen mit AIMMS überhaupt möglich sind. Deshalb wurde sich als guter Einstieg für die Optimierung der Verteilung der Feuerkraft entschieden.

Zu diesem Zweck wird ein gesondertes kleines Modell in AIMMS erstellt. In diesem Modell gibt es ebenfalls die zwei Fraktionen OPFOR und Alliance. Die Alliance hat einen wehrlosen Stützpunkt, von dem allerdings im Angriffsfall Abfangjäger gestartet werden können, welche den Stützpunkt verteidigen und gegnerische Angreifer abschießen. Die OPFOR schicken jetzt eine Schwadron zum Angriff los, welches ihr Feuer entweder auf den Stützpunkt oder auf die Jäger richten kann. Dabei sind die Angreifer effektiver gegen den wehrlosen Stützpunkt als gegen die agilen Abfangjäger. Des Weiteren haben die OPFOR nur 60 Minuten Zeit, um ihre Offensive zu fliegen, da nach dieser Zeit Verstärkung der Alliance eintrifft, welche die OPFOR Flieger zum Rückzug zwingt, da diese eindeutig unterlegen sind. Die 60 Minuten werden in diesem Modell in 6 Zeitschritte von 10 Minuten unterteilt. Die für das Optimierungsproblem notwendige Nutzenfunktion soll sich aus verhindertem und ausgeteiltem Schaden zusammensetzen. Hierbei ist es gleich, ob Schaden am Stützpunkt oder an den Abfangjägern ausgeteilt wird. Allerdings soll erlittener Schaden der eigenen Kräfte gravierendere Auswirkungen haben. Mathematisch sieht das Modell wie folgt aus:

$$\text{Maximiere } N(u(t)) = 3 \cdot A(6) - J(6) - S(6), \quad (5.1)$$

$$(5.2)$$

$$\text{u.d.N. } \frac{dA(t)}{dt} = -aJ(t), \quad (5.3)$$

$$\frac{dJ(t)}{dt} = -jA(t) \cdot (1 - x(t)), \quad (5.4)$$

$$\frac{dS(t)}{dt} = -sA(t) \cdot x(t). \quad (5.5)$$

Symbole:

- $t$ : Zeitschritte
- $A(t)$ : Anzahl der angreifenden Flugzeuge.
- $J(t)$ : Anzahl der verteidigenden Jäger.
- $S(t)$ : Status des Stützpunktes in Prozent.
- $a$ : Effektivität der Jäger gegen Angreifer (abgeschossene Flugzeuge pro Zeitschritt).
- $j$ : Effektivität der Angreifer gegen Jäger (abgeschossene Flugzeuge pro Zeitschritt).
- $s$ : Effektivität der Jäger gegen den Stützpunkt (angerichteter Schaden pro Zeitschritt in Prozent).
- $x(t)$ : Anteil des Feuers auf den Stützpunkt.

Wird zu Beginn von 36 Angreifern und 12 Jägern ausgegangen und werden die Werte  $a = 1$ ,  $j = 0,16$  und  $s = 1,5$  vorausgesetzt, so errechnet AIMMS folgendes Ergebnis in Abbildung 5.1. Es ist zu erkennen, dass die optimale Herangehensweise die ist, bei der das Feuer zuerst vollkommen auf die Jäger und anschließend auf den Stützpunkt gerichtet wird. Nur zwischendurch sollte das Feuer einmal aufgeteilt werden, da bei den letzten Jägern nicht mehr die volle Feuerkraft gebraucht wird.

Das Optimierungsproblem des Typ QCP kann dabei von vier der fünf benutzten Solver gelöst werden. CPLEX ist nicht in der Lage eine Lösung zu finden. CONOPT, IPOPT, MINOS und SNOPT hingegen finden das Optimum in so kurzer Zeit, dass von AIMMS eine Berechnungszeit von 0 Sekunden angegeben wird.

Somit lässt sich sagen, dass es möglich ist, Optimierungsprobleme, welchen die Aufteilung der Feuerkraft bei Gefechten zu Grunde liegt, mit AIMMS zu lösen und daraus eine Strategie für den Kampf abzuleiten.

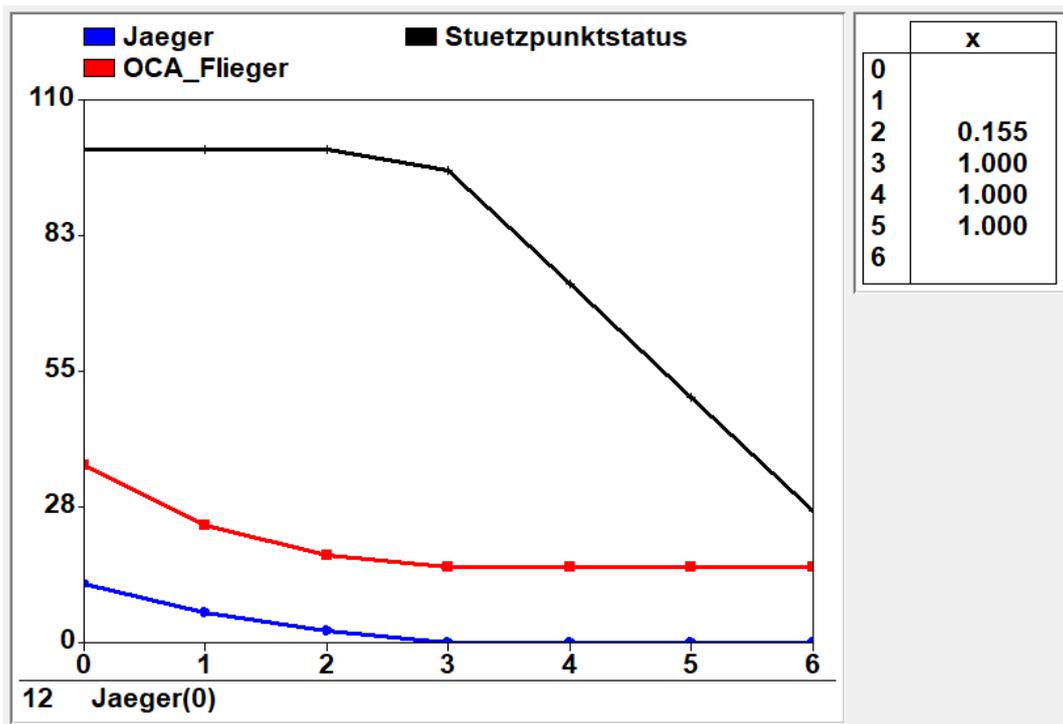


Abbildung 5.1: Ausschnitt aus AIMMS.

## 6 Schlussfolgerungen

Im Zuge dieser Arbeit war es möglich, das dem LAMBDA-Tool zugrunde liegende mathematische Modell zu extrahieren und zusammenfassend aufzustellen. Die Auswertung des Tools wurde dabei durch eine Implementierung des Modells in Vensim verifiziert, da es möglich war, dieselben Ergebnisse wie das LAMBDA-Tool zu produzieren. Generell hat die Nutzung von Vensim auch einen neuen Weg für die Simulation des Modells weg von Excel hin zu einem passenderen Simulationstool gezeigt. Solange eine Optimierung von Gefechtsstrategien für das LAMBDA-Modell noch nicht vollzogen ist, erlaubt Vensim es, schneller und flexibler verschiedene Szenarien durchzuspielen und somit die Effekte von geplanten Strategien zu erfahren. Die Implementierung von mehr als zwei Schwadronen pro Fraktion wäre dabei ein Ansatzpunkt für zukünftige Arbeiten.

Die Auswertung des LAMBDA-Tools hat außerdem gezeigt, dass das Modell um ein Vielfaches komplexer ist als das ursprüngliche Gesetz von Lanchester. Aktuell herrschen zwar keine beidseitig luftgestützten Konflikte, allerdings sind diese in Zukunft auch nicht auszuschließen. In dem Fall würde das LAMBDA-Modell durch seine vielen Parameter die Einbindung von asymmetrischen Ausgangslagen erlauben und somit wertvolle Einblicke in Gefechtsabläufe liefern können. Die Analyse hat aber auch gezeigt, dass es Stellen im Modell gibt, die Ausbesserung verlangen.

Des Weiteren liefert das Modell einen anderen Blickwinkel auf die Strategiewahl bei Gefechten als das Gesetz von Lanchester. Wo es früher von Bedeutung war, seine Truppen möglichst geballt aufzustellen, ist es jetzt viel wichtiger, sie an den richtigen Stellen und auf den Gegner angepasst einzusetzen. Die Arbeit hat an dieser Stelle auch im Ansatz gezeigt, dass es möglich ist, diese Wahl des Truppeneinsatzes mit AIMMS zu optimieren. Die Weiterführung dieser Optimierung des LAMBDA-Modells sollte in anderen Arbeiten durchaus angegangen werden, da sie einen erheblichen Gewinn für Gefechtstplanung darstellen würde. Es wird aber womöglich nie gelingen, eine zu Beginn festgelegte Gefechtsstrategie zu erarbeiten, die einer Seite den Sieg garantiert, da dies nicht nur das Wissen über Parameter wie das Wetter, was in Zukunft vielleicht sogar exakt vorhersagbar ist, sondern auch über das zukünftige Verhalten des Gegners verlangt. Übergeordnetes Ziel für Forschung in diesem Gebiet sollte somit das Erarbeiten einer sich dynamisch anpassbaren Strategie sein.

## Literaturverzeichnis

- [1] *AIMMS Solvers*. <https://aimms.com/english/developers/resources/solvers/>, 09.08.2017.
- [2] BERKOVITZ, L. D. und MELVIN DRESHER: *A Game-Theory Analysis of Tactical Air War*. *Operations Research*, 7(5):599–620, 1959.
- [3] BONDER, SETH: *The Lanchester Attrition-Rate Coefficient*. *Operations Research*, 15(2):221–232, 1967.
- [4] BOSSEL, HARTMUT: *Systeme, Dynamik, Simulation - Modellbildung, Analyse und Simulation komplexer Systeme*. Books on Demand GmbH, Norderstedt, 2004.
- [5] CRUZ, J. B., M. A. SIMAAN, A. GACIC und Y. LIU: *Moving horizon Nash strategies for a military air operation*. *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):989–999, Jul 2002.
- [6] DIONNE, D., E. POGOSSIAN, A. GRIGORYAN, J. COUTURE und E. SHAHBAZIAN: *An optimal sequential optimization approach in application to dynamic weapon allocation in naval warfare*. In: *2008 11th International Conference on Information Fusion*, Seiten 1–6, June 2008.
- [7] LANCHESTER, FREDERICK W.: *Aircraft in Warfare – The Dawn of the Fourth Arm*. Constable and Company Limited, London, 1916.
- [8] POPKEN, DOUGLAS und LOUIS COX: *A Simulation-optimization Approach to Air Warfare Planning*. *The Journal of Defense Modeling and Simulation*, 1(3):127–140, 2004.
- [9] TAYLOR, JAMES G.: *Lanchester-type models of warfare and optimal control*. *Naval Research Logistics Quarterly*, 21(1):79–106, 1974.



