Data Farming und simulationsbasierte Robustheitsanalyse für Fertigungssysteme

Thomas Schulze[1*], Niclas Feldkamp[2], Sören Bergmann[2], Steffen Straßburger[2]

[1] Institut für Technische und Betriebliche Informationssysteme, Otto-von-Guericke Universität Magdeburg, Universitätsplatz 2, 39102 Magdeburg, *thomas.schulze@ovgu.de*

[2] Fachgebiet Wirtschaftsinformatik für Industriebetriebe, Technische Universität Ilmenau, PF 10 05 65, 98684 Ilmenau

Abstract. Diskrete Simulation ist eine etablierte Methodik zur Untersuchung des dynamischen Verhaltens von komplexen Fertigungs- und Logistiksystemen. Konventionelle Simulationsstudien fokussieren auf einzelne Modellaspekte und spezifische Analysefragen. Der Umfang der ausgeführten Szenarien ist häufig gering. Das Konzept des Data-Farming verwendet das Simulationsmodell als Datengenerator für eine breite Skale von Experimenten und ermöglicht unter Nutzung von Data-Mining-Methoden eine wesentlich breitere Untersuchung des simulierten Systems sowie eine höhere Komplexität in den abgeleiteten Erkenntnissen. Anforderungen an Simulationssysteme und –modelle zur Durchführung von Data-Farming werden erläutert. Eine Erweiterung des Ansatzes ist die simulationsbasierte Robustheitsanalyse auf der Basis von Verlustfunktionen nach Taguchi. Beide Vorgehensweisen werden an einer Fallstudie aus dem Fahrzeugbau demonstriert.

1 Einführung

Simulation ist ein etabliertes Werkzeug zur Planung und Steuerung von Fertigungs- und Logistiksystemen. Traditionelle Simulationsstudien versuchen ein vorgegebenes Projektziel zu erfüllen, wie z. B. Verbesserung des Hallenlayouts bzw. Reduzierung von Pufferkapazitäten. Zum Erreichen der Ziele werden häufig einzelne Simulationsexperimente manuell definiert oder es werden simulationsbasierte Optimierungsverfahren verwendet.

Mit der Verfügbarkeit von höherer Rechenleistung durch schnelle Prozessoren und paralleles Abarbeiten sowie der Anwendung von Data-Mining-Verfahren werden die Anwendungsmöglichkeiten von Simulationen erweitert. Somit können sehr viele ausgewählte Simulationsexperimente durchgeführt werden um im Folgenden, bspw. mittels Data-Mining-Methoden, unbekannte Wirkzusammenhänge aufzudecken. Der Simulator agiert hierbei als Datengenerator mit dem Ziel, eine umfangreiche Datenmenge zu erzeugen. In einem Folgeschritt werden diese Datenmengen unabhängig vom Simulator analysiert. Diese Analyse ermöglicht die Ableitung von Erkenntnissen zum simulierten System, die sich aus einem klassischen Simulationsprojekt mit wenigen Experimenten nicht ableiten lassen. [1], [2].

Ein Produktionsprozess wird i.A. als robust bezeichnet, wenn er gegen unerwünschte Einflussgrößen stabil bleibt und die Produktionsziele unter Einhaltung wirtschaftlicher Zielgrößen erreicht. In diesem Beitrag wird unter robustem Fertigungssystem ein System verstanden, welches gegenüber Schwankungen im operatives Produktionsprogramm stabil bleibt.

Allgemein ist festzustellen, dass bei der Planung von Fertigungssystemen die Robustheit der Systeme eine steigende Bedeutung erfährt, häufig wird aber zur simulationsgestützen Planung nur ein vorgegebenes Produktionsprogramm verwendet. [3], [4]. An diesem antizipierten Produktionsprogramm wird die gesamte Planung ausgerichtet. Dagegen sind robuste Systeme in der Lage, auch bei sich verändernden Produktionsprogrammen, die geforderten Leistungsparameter zu sichern. Die Methode des Data Farmings [5] in Kombination mit der Anwendung von Verlustfunktionen nach Taguchi [6] erlaubt eine Bestimmung der Robustheit unterschiedlicher Systementwürfe.

Ziel dieses Beitrags ist das Aufzeigen von ausgewählten Ergebnissen aus einer Data-Farming-Analyse sowie aus der Ermittlung von robusten Systementwürfen an Hand einer Fallstudie aus einem Industrieprojekt. Darüber hinaus werden allgemeine notwendige Anforderungen an Simulationssysteme und –modelle zur Durchführung einer großen Anzahl von Simulationsexperimenten abgeleitet.

Der Beitrag ist wie folgt aufgebaut: Nach der Einleitung erfolgt ein kurzer Abriss zu den Themen Data Farming und simulationsbasierter Robustheitsanalyse. Anschließend werden Anforderungen an Simulationssysteme und -modelle zur Anwendung von Data Farming und Robustheitsanalyse diskutiert. In einer Fallstudie werden die erläuterten Methoden veranschaulicht. Fazit und Ausblick schließen den Beitrag ab.

2 Data Farming

Die Grundidee des ursprünglichen Data-Farming-Konzeptes [7],[8],[9] ist die Verwendung eines Simulationsmodells als Datengenerator, wobei das Ziel ist, mit Hilfe von effizientem Experimentdesign und High Performance Computing, ein möglichst vollständiges Spektrum von Ergebnisdaten zu erzeugen und somit den Informationsgewinn zu verbessern [5],[7]. Die „Farming"-Metapher drückt hierbei aus, dass ähnlich einem Farmer, der sein Land möglichst ertragreich kultiviert, der Datenertrag des Simulationsmodells maximiert werden soll [8]. Die hierbei eingesetzten Ansätze zur Gestaltung von Simulationsexperimenten, z. B. die häufig verwendete Methode Nearly Orthogonal Latin Hypercubes (NOLH) [10], erlauben bei vertretbaren Datenmengen respektive Experimentzahlen die umfassende Abbildung möglicher Wertekombinationen von Eingabeparametern [11]. Ursprünglich wurde Data Farming für militärische Gefechtssimulationen entwickelt, da bei der Anwendung von klassischen Simulationsstudien die gestellten komplexen Fragestellungen nicht ausreichend beantwortet werden konnten [7].

Aufbauend auf dem originären Data-Farming-Konzept wurden weitere Methoden zur Analyse der Daten entwickelt bzw. angewendet. Eine grundlegende Methodik, welche das Auffinden von versteckten, potenziell nützlichen Wirkzusammenhängen in Ergebnisdaten von nicht-militärischen Simulationsmodellen, insbesondere im Kontext von Produktion und Logistik, erlaubt, wurde von Feldkamp et al. unter dem Namen „Knowledge Discovery in Simulation Data" entwickelt [12],[1]. Hierbei kommen zur Verarbeitung der vom Simulator generierten Daten Methoden des Data Mining sowie Methoden der interaktiven, visuellen Analyse zum Einsatz. Diese von Feldkamp [12] entwickelte Methodik strebt nach der konsequenten Verzahnung von Datenanalyse und -Visualisierung, wobei deren Verbindungsglied die menschliche Fähigkeit zur Interpretation und Schlussfolgerung darstellt. Gefördert wird dies durch ein hohes Maß an Interaktivität des Gesamtprozesses der Datenauswertung [13]. Diese Methodik zur Wissensentdeckung in Simulationsdaten wurde in Industriefallstudien erprobt.

3 Simulationsbasierte Robustheitsanalyse

Im Allgemeinen versteht man unter Robustheit eines Systems, in wie weit sich nicht direkt vom Systembetreiber kontrollierbare Störgrößen auf die interessieren Leistungskennzahlen eines Systems auswirken. Hierbei ist ceteris paribus die Robustheit hoch, wenn die Auswirkungen der Störgrößen möglichst gering ist, vice versa ist diese gering wenn die Auswirkungen hoch sind [15]. Des Weiterem wird oft, wenn auch von der eigentlichen, ursprünglichen Definition abweichend, eine hohe Leistung der als robust gekennzeichneten Systeme unterstellt; d. h. im engeren Sinn sind auch Lösungen mit sehr geringer Leistung robust, wenn diese interessierenden Leistungskennzahlen unter verschiedenen Werten für die Störgrößen nicht oder nur schwach schwanken. Im Folgenden sollen aber nur Lösungen als robust bezeichnet werden, die neben dieser nötigen Bedingung auch ein definiertes Leistungskennzahlmaß, z. B. Durchsatz und Durchlaufzeiten ausreichend erfüllen. Ziel ist es, die durch den Planer kontrollierbaren Systemeigenschaften, auch als Systemkonfiguration bezeichnet, so zu setzen, dass die mittlere Systemleistung über alle zu berücksichtigen Störungen möglichst hoch ist und die Varianz der Systemleistung möglichst klein ist.

Zur Bewertung der Robustheit wird unter anderen die Methode der Verlust-Funktionen (Loss Functions) nach Taguchi [6] genutzt. Taguchis Idee resultiert aus seinen umfangreichen Erfahrungen aus dem Bereich des Qualitätsmanagements und hierbei insbesondere auf der Beobachtung, dass jede Abweichung von einem Zielwert einen monetären Verlust nach sich zieht. Zur Bewertung einer Systemkonfiguration sind verschiedene Verlustfunktionen abhängig der Rahmen- und Zielparameter möglich, die in Tabelle 1 aufgezeigt sind.

Art der Verlustfunktion	Formel
Nominal-the-best	$\bar{L} = k[\sigma^2 + (\bar{y} - \tau)^2]$
Smaller-the-better	$\bar{L} = k[\bar{y}^2 + \sigma^2]$
Larger-the-better	$\bar{L} = k\left[\sum(1/y^2)\right]/n$

Tabelle 1: Verlustfunktion für verschieden Zwecke [16].

$\bar{L}$ steht jeweils für den durchschnittlichen Verlust einer bestimmten Systemkonfiguration über alle Konfigurationen von Störgrößen, wobei $\bar{y}$ und σ^2den Mittelwert bzw. die Varianz der betrachteten Zielgröße einer Systemkonfiguration darstellen. Bei der „nominal-the-best“-Verlustfunktion, welche Abweichungen von einem vorgegebenen Zielwert sowohl nach oben als auch nach unten bestraft, wird dieser Zielwert als τ angegeben. Zusätzlich kann die sog. Qualitätsverlustkonstante k verwendet werden, um den Qualitätsverlust in geeigneter Weise umzurechnen und monetär bewerten zu können.

Taguchis Ansatz wurde in mehreren Publikationen auf Simulationsexperimente angewendet. So skizzierte Sanchez einen Ansatz zur Integration des Konzepts der Robustheit mit der Response-Surface-Metamodellierung zur Optimierung von Modellen [17]. Dellino et al. nutzte einen ähnlichen Ansatz zur simulationsbasierten Optimierung von Robustheitsproblemen unter Verwendung von Response-Surface-Methodik und Kriging-Metamodellen [18]. Auch im Bereich der Gefechtsfeldsimulation wurden die Potentiale der Verlustfunktion Taguchis erkannt [19].

In der hier vorgestellten Methodik wird das Ziel verfolgt, eine Robustheitsanalyse basierend auf den Verlustfunktionen von Taguchi mit einem großangelegten Experimentdesign und visuell unterstützten Methoden zur Wissensentdeckung für Fertigungssimulationen zu kombinieren [14]. Hierzu kommen gekreuzte Experimentpläne zum Einsatz.

Der Experimentplan zur Variation der Systemkonfigurationen wird gekreuzt mit einem Experimentplan zur Abbildung der Störszenarien. Dies erlaubt neben der Analyse der Konfigurationen je Störszenario auch Bias frei die Analyse der Robustheit einer Systemkonfiguration gegenüber den Störfaktoren mittels den beschrieben Verlustfunktionen. Abschließend können mittels Methoden des Knowledge Discovery in Simulation Data Rückschlüsse bzgl. der relevanten Systemfaktoren und deren Ausprägungen gezogen werden.

4 Anforderungen an Simulationssysteme und –modelle zur Durchführung von Data-Farming

Die verwendeten Simulationssysteme und -modelle müssen bestimmten Anforderungen genügen, die im Folgenden erläutert werden. Darüber hinaus wird ein Data-Farming-Management-Tool mit den folgenden Basisfunktionen benötigt:

- Erstellung des Experimentdesigns unter Verwendung geeigneter Methoden wie NOLH, Automatisches Verteilen, Starten und Beenden der Experimente sowie deren Monitoring und
- Erfassen und Aufbereiten der Resultate aus den einzelnen Experimenten. Fehlerhafte Experimente müssen erkannt und in Abhängigkeit von der Fehlerart gegebenenfalls automatisch neu gestartet werden.

Die zu verwendeten Simulationssysteme müssen die Modellgenerierung bzw. -modifikation und das Setzen von Eingabeparametern ohne Nutzereingriffe erlauben. Schnittstellen müssen vorhanden sein, um die Eingabedaten aus den definierten Experimentplänen zu importieren. Es sind Datenschnittstellen zu verwenden, die ein laufzeiteffizientes Einlesen sicherstellen. Des Weiteren müssen die Simulationssysteme durch das Data-Farming-Management-Tool steuerbar sein.

Existierende kommerzielle Simulationssysteme zur Simulation von Fertigungssystemen, wie Plant Simulation [20] oder FlexSim [21] verfügen nicht über alle benötigten Basisfunktionen eines Data-Farming-Management-Tools. Für das Data-Farming in der Fallstudie wurde daher ein proprietäres Softwarewerkzeug entwickelt.

Eine der Limitationen in der Anwendung der Data-Farming-Methodik ist durch die Validität der Simulationsmodelle begründet. Für klassische Simulationsstudien mit wenigen Szenarien sind entsprechende Validierungsverfahren entwickelt worden [22]. Im Ergebnis des Data-Farming-Experimentdesigns werden Wertekombinationen für Eingabedaten verwendet, die unter Umständen den validierten Eingabedatenraum aus der klassischen Studie überschreiten. Bei der Analyse der Ergebnisdaten ist dieser Umstand zu berücksichtigen.

Zur Durchführung von Data-Farming muss das existierende Simulationsmodell aus der klassischen Simulationsstudie angepasst werden. Zum einen muss der Umfang der interessierenden Zusammenhänge an das Spektrum der Ausgabewerte angepasst werden. Das bedeutet ggf. weitere Ausgabewerte zu modellieren oder auch nicht benötigte Werte aus dem Modell zu entfernen. Des Weiteren müssen Schnittstellen zur Kommunikation mit dem Data-Farming-Management-Tool zum Einlesen der Eingabedaten und der Ergebnisausgabe implementiert werden.

Eine Modellanalyse zur Erhöhung der Abarbeitungsgeschwindigkeit wird empfohlen. Nur die Modellvariablen, die als Ergebnisgrößen für das Data Farming verwendet werden, sind zu berücksichtigen. Werteaufzeichnungen, statistische Berechnungen, grafische Aufbereitungen und das Speichern auf externen Geräten von nicht benötigten Modellvariablen erhöhen in unnötiger Weise die Rechenzeit. Bei der Durchführung von Tausenden von Experimenten beim Data-Farming hat die Rechenzeit eines einzelnen Experimentes einen Einfluss auf die Anzahl der auszuführenden Experimente, wenn für die Durchführung der Untersuchungen nur ein definierter Zeitrahmen zur Verfügung steht.

5 Fallstudie

Für eine Demonstration der Vorgehensweisen zum Data-Farming und der simulationsbasierten Robustheitsanalyse wird ein Simulationsmodell aus einem Industrieprojekt verwendet, welches die Montage von Fahrzeugen nachbildet. Das „legacy"-Simulationsmodell wurde vom Industriepartner erstellt und in einer klassischen Simulationsstudie verwendet. Dieses Modell wurde an die Anforderungen zum Data-Farmings angepasst. Die ursprüngliche Rechenzeit für ein Experiment konnte dabei signifikant gesenkt werden.

Das zu simulierende System umfasst über 50 Montagestationen mit über 50 Werkern und 14 Puffer. Die Stationen werden in 5 verschiedenen Zonen logisch zusammengefasst. Das Produktspektrum enthält 720 verschiedene Fahrzeugtypen, die an den einzelnen Stationen unterschiedliche Montagezeiten aufweisen. Die Ergebnisgrößen für die klassische Simulationsstudie sind u. a. der mittlere tägliche Durchsatz und die Auslastung der Werker.

Im ersten Schritt des Data-Farming-Konzeptes werden die Eingabefaktoren definiert. Diese Faktoren werden klassifiziert in Entscheidungs- und Störfaktoren. Entscheidungsfaktoren lassen sich bei der Planung des zu untersuchenden Systems durch den Planer beeinflussen. Störfaktoren hingegen sind durch den Planer nicht steuerbar. Tabelle 2 gibt einen Überblick zu den verwendeten Eingangsfaktoren und ihren Wertebereichen. Die Eingabefaktoren mit den entsprechenden Wertebereichen wurden gemeinsam mit dem Industriepartner ausgewählt.

Zwischen 13 ausgewählten Montagestationen der Hauptlinie sind Puffer vorgesehen. Mit der Faktorengruppe „Pufferkapazität" wird die Kapazität eines einzelnen Puffers bezeichnet, wobei insgesamt 13 Faktoren dieser Gruppe zugeordnet sind.

Faktorname	Anzahl	Typ	Wertebereich
Pufferkapazität	13	E	1 - 10
Skill Level Werker	5	E	1 oder 2
Normzeitkoeffizient	5	E	1,0 – 1,3
Produktmix	30	S	30 Varianten

Tabelle 2: Eingabefaktoren ; Erläuterung der verwendeten Faktoren (Typ E: Entscheidungs-; Typ S: Störfaktor).

Mit dem Faktorengruppe „Skill Level Werker" wird die Fähigkeit der Werker beschrieben, an mehr als an einer Station arbeiten zu können. In dieser Studie wurden zwei diskrete Ausprägungen 1 oder 2 verwendet. Der Wert 1 bedeutet hierbei, dass ein Werker nur an einer ihm explizit zugewiesenen Station arbeiteten kann. Ein Wert von 2 erlaubt eine höhere Flexibilität, indem Werker die Fähigkeiten besitzt, sowohl an seiner direkt zugewiesenen Station, sowie an den vor- bzw. nachgelagerten Stationen zu arbeiten.

Die Skill-Level-Faktoren werden hierbei nicht für einzelne Werker, sondern insgesamt je Zone vergeben. Für die fünf Zonen ergeben sich somit fünf verschiedene Faktoren aus dieser Gruppe. Die einzelnen Faktoren dieser Gruppe werden mit Skill_Zone* bezeichnet, wobei für * die Werte 1 bis 5 gelten.

Die Faktorengruppe „Normzeitkoeffizient" (Real Time Calibration Coefficient – RTCC) beschreibt, in welchem Verhältnis die geplante zur tatsächlichen Montagezeit steht. Ein Wert von 1 bedeutet hierbei das die tatsächliche Bearbeitungszeit der geplanten entspricht. Ein höherer Wert zeigt an, dass die tatsächlichen Montagezeiten kürzer als die geplanten Montagezeiten sind. Der Faktor „Normzeitkoeffizient" kann ebenfalls nur je Zone differenziert werden. Auch hier ergeben sich somit für die fünf Zonen die entsprechenden einzelnen Faktoren, die mit RTCC_Zone* bezeichnet werden.

Der Faktor „Produktmix" ist ein qualitativer Faktor, d. h., die numerischen Werte für diesen Faktor dienen nur der Kennzeichnung von verschiedenen Produktspektren, auch als Produktmixe bezeichnet. Unter einem Produktmix wird eine Liste prozentualer Anteile

aller zu produzierenden Produkttypen aufgefasst, die in einem bestimmten Zeitraum zu produzieren sind. Das verwendete Simulationsmodell verwendet einen derartigen Produktmix als Eingabeparameter und generiert aus selbigen ein Produktionsprogramm, welches die Reihenfolge der zu fertigenden Fahrzeuge nachbildet. Es werden 30 verschiedene Produktmixe verwendet und der Wert für den Faktor „Produktmix“ kennzeichnet die laufende Nummer des Produktmixes.

Die Fallstudie wurde in zwei Phasen durchgeführt. Phase 1, als Data-Farming-Analyse bezeichnet, verwendet nur die Entscheidungsfaktoren. In Ergänzung dazu werden in Phase 2 auch die Störfaktoren mit einbezogen. Diese Phase wird als Robustheitsanalyse bezeichnet.

5.1 Data-Farming-Analyse

In Phase 1der Fallstudie wurden zunächst nur die 23 (13 + 5+ 5) verschiedenen Entscheidungsfaktoren verwendet. Die Einbeziehung der Störfaktoren, d.h. unterschiedlicher Produktmixe, erfolgt in der zweiten Phase.

Der Experimentplan für diese Entscheidungsfaktoren wurde auf Basis der Nearly-Orthogonal-Latin-Hypercube-Methode (NOLH) erstellt [10]. Basierend auf dieser Methode wurden 510 verschiedene Kombinationen über die verschiedenen Entscheidungsfaktoren erzeugt. Eine einzelne Kombination wird auch als Konfiguration bezeichnet. Nach der Durchführung der 510 Experimente können erste Auswertungen vorgenommen werden. Unter der Festlegung der Simulationszeit mit 5 Wochen betrug die Rechenzeit für jeden einzelnes Experiment 260 bis 280 sec.

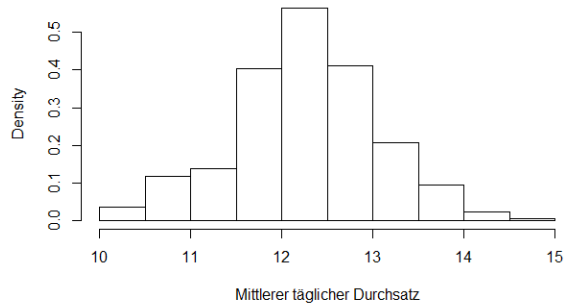

Bild 1: Histogramm der Ergebnisgröße “Mittlerer täglicher Durchsatz” über 510 Systemkonfigurationen.

Eine erste Analyse untersucht den Einfluss der Entscheidungsfaktoren auf die Ergebnisgrößen. Für diesen Beitrag wird sich auf die Ergebnisgröße „Mittlerer täglicher Durchsatz“ beschränkt. Bild 1 zeigt ein Histogramm über die 510 Simulationsexperimente für die Ergebnisgröße „Mittlerer täglicher Durchsatz“, wobei jedes Experiment eine Systemkonfiguration beschreibt. Der Wertebereich erstreckt sich von minimal 10,35 bis maximal 14,6 Fahrzeuge pro Tag. Diese Ergebnisgröße wird von den Konfigurationen und somit von den Entscheidungsfaktoren beeinflusst.

Weitere Analysen untersuchen, welche der 23 Entscheidungsfaktoren den stärksten bzw. einen starken Einfluss auf die Ergebnisgröße „Mittlerer täglicher Durchsatz“ haben. Erste Hinweise gibt eine Korrelationsanalyse zwischen der Ergebnisgröße und den Entscheidungsfaktoren. Hohe Korrelationskoeffizienten weisen auf einen tendenziell starken Einfluss hin. Die Faktoren der Gruppe „Pufferkapazität“ weisen allesamt sehr niedrige Korrelationswerte auf. Damit wird ein sehr geringer Einfluss dieser Faktorengruppe auf den Durchsatz vermutet. Die fünf Entscheidungsfaktoren mit den höchsten Korrelationskoeffizienten werden als Hauptentscheidungsfaktoren bezeichnet und sind in Tabelle 3 aufgelistet. Es ist abzulesen, dass die Faktorengruppe „Pufferkapazität“ keinen signifikanten Einfluss auf die Ergebnisgröße hat.

Faktorname	Korrelationskoeffizient
RTCC_Zone1	0,532
Skill_Zone1	0,507
RTCC_Zone4	0,409
RTCC_Zone3	0,165
RTCC_Zone2	0,098

Tabelle 3: Korrelationskoeffizienten für die Hauptentscheidungsfaktoren.

Ohne weitere Betrachtung von Interaktionseffekten zwischen einzelnen Faktoren scheint es, dass der Entscheidungsfaktor RTCC_Zone1, d.h. der Normzeitkoeffizient für Zone 1, den größten Einfluss auf die Ergebnisgröße „Mittlerer täglicher Durchsatz“ hat. Der Wert des Korrelationskoeffizienten von 0,532 lässt vermuten, dass noch weitere Entscheidungsgrößen bzw. Interaktionseffekte den täglichen Durchsatz beeinflussen. Das Streudiagramm zwischen diesem Entscheidungsfaktor und der Ergebnisgröße in Bild 2 indiziert die Tendenz, dass ein hoher Wert für RTCC_Zone1 zu einem hohen täglichen Ausstoß führt.

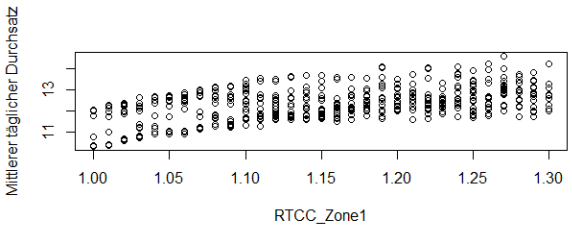

Bild 2: Scatterdiagramm für den Entscheidungsfaktor RTCC_Zone1 und der Ergebnisgröße „Mittlerer täglicher Durchsatz“.

Aus der Liste der Korrelationskoeffizienten in Tabelle 3 lässt sich weiter ableiten, dass die Entscheidungsfaktoren RTCC_Zone1 und Skill_Zone1 den höchsten Einfluss auf die Ergebnisgröße „Mittlerer täglicher Durchsatz“ haben. Für den Planer ergibt sich die Erkenntnis, dass zur Erreichung eines hohen Durchsatzes die Zone1 von besonderer Bedeutung ist und ggf. sogar einen Engpass darstellt.

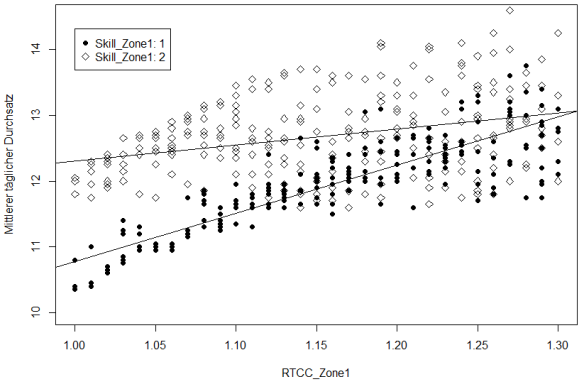

Bild 3: Scatterdiagramm für die Ergebnisgröße „Mittlerer täglicher Durchsatz“ in Abhängigkeit von den Entscheidungsfaktor RTCC_Zone1 und Skill_Zone1.

In einer weiteren Analyse wird untersucht, ob es eine Interaktion zwischen beiden Entscheidungsfaktoren auf diese Ergebnisgröße gibt. Dabei ist zu berücksichtigen, dass der Entscheidungsfaktor Skill_Zone 1 nur die Werte 1 oder 2 hat. Bild 3 zeigt ein Streudiagramm zwischen der Ergebnisgröße „Mittlerer täglicher Durchsatz“ und den beiden Einflussfaktoren. Die Linien sind Regressionsgeraden für die Ergebnisgröße in Abhängigkeit vom Wert des Entscheidungsfaktors Skill_Zone1. Aus diesem Bild wird geschlossen, dass bei niedrigen Werten von bis zu bis 1,15 für den Einflussfaktor RTCC_Zone1 der Durchsatz gesteigert wird, wenn der Entscheidungsfaktor Skill_Level1 den Wert gleich 2 hat. Des Weiteren kann für die Planer abgeleitet werden, dass eine höhere Flexibilität der Werker in Zone 1 vorteilhaft ist, wobei sich dieser Effekt bei hohen Werten für den Normzeitkoeffizienten wieder relativiert.

Entscheidungsbäume sind eine weitere Methodik zur Analyse von Einflussfaktoren und ggf. sogar Interaktionen zwischen diesen. Das Anwendungsziel eines Entscheidungsbaumes, hier eines Regressionsbaumes ist die Vorhersage eines Zielgrößenwertes in Abhängigkeit von Eingabewerten. Für die Ergebnisgröße „Mittlerer täglicher Durchsatz“ wurde ein Regressionsbaum in Abhängigkeit von den identifizierten Hauptentscheidungsfaktoren erstellt (vgl. Bild 4). Die Prozentzahlen an den Knoten geben an, wieviel Prozent der untersuchten Fälle durch diesen Knoten beschrieben werden.

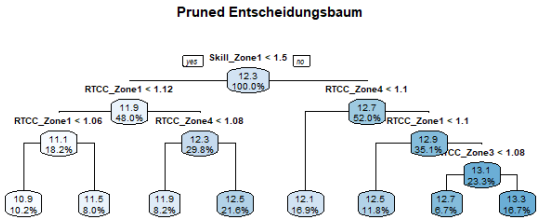

Bild 4: Pruned Entscheidungsbaum für die Ergebnisgröße „Mittlerer tägliche Durchsatz“.

Der Entscheidungsfaktor Skill_Zone1 bildet die Wurzel dieses Baumes, d.h. dieser Faktor hat den größten Einfluss im Zusammenspiel mit den anderen Entscheidungsfaktoren. Ein Wert von 1 für diesen Faktor bedeutet, dass dann der Mittelwert für die Ergebnisgröße „Mittlerer täglicher Durchsatz“ gleich 11,9 über alle Kombinationen der restlichen Einflussfaktoren ist. Wird hingegen für den Einflussfaktor ein Wert von 2 verwendet, so ist der Mittelwert über die restlichen Wertekombinationen gleich 12,7. Aus diesem Baum lässt sich an die Planer des Montagesystems folgende Empfehlung für einen hohen Durchsatz ableiten:

$$(\text{Skill\_Zone1} = 2) \;\&\&\; (\text{RTCC\_Zone4} > 1{,}4) \;\&\&\; (\text{RTCC\_Zone1} > 1{,}1)$$

5.2 Robustheitsanalyse

Die Robustheit des Montagesystems steht im Zentrum von Phase 2 der Fallstudie. Bei der Planung von Fertigungssystemen spielt die Robustheit der Systeme gegenüber Störeinflüssen, wie z. B. Schwankungen der Produkteanteile im geplanten Produktionsprogramm, eine große Rolle. Mit der Robustheitsanalyse soll unter-

sucht werden, wie robust die 510 verschiedenen Systemkonfigurationen gegenüber solchen unterschiedlichen Produktmixen sind. Bei diesem Simulationsmodell kann sich auf Produktmixe als Eingabeparameter beschränkt werden, da die Reihenfolge der Produkte durch das Simulationsmodell erzeugt wird.

Eine Robustheitsanalyse kann sich auf eine oder mehrere Ergebnisgrößen beziehen. Für diese Fallstudie wird sich auf die Analyse bezüglich der Ergebnisgröße „Mittlerer täglicher Durchsatz“ beschränkt. Ziel dieser Robustheitsanalyse ist es, die Systemkonfigurationen zu identifizieren, welche eine hohe Robustheit hinsichtlich dieser Ergebnisgröße aufweisen. Zur Durchführung der Analyse müssen die notwendigen Produktmixe erzeugt werden. Diese Produktmixe sind als Werte für den Störfaktor Produktmix (Tabelle 1) zu betrachten.

Der im klassischen Simulationsprojekt verwendete Produktmix wird als originärer Produktmix bezeichnet. Dieser Mix ist der Ausgangspunkt zur Generierung der benötigten ähnlichen Produktmixe, die Derivate genannt werden. Der originäre Produktmix enthält 720 verschiedene Produkttypen. Die Anteile der Produkttypen am originären Produktmix sind unterschiedlich. Auf der Basis der folgenden Schrittfolge werden die Derivate abgeleitet.

1. Festlegung eines relativen Bereiches, z. B. 20 %, für alle Anteilswert der Produkttypen im originären Produktmix.
2. Ermittlung der neuen Anteilswerte der Produkttypen aus den in Schritt 1 bestimmten Bereichen mittels stochastischer Experimente. Dabei muss die Bedingung eingehalten werden, dass die Summe der prozentualen Anteile 100 ergibt.

Schritt 2 muss für jeden neuen Produktmix wiederholt werden. In dieser Fallstudie wurden 29 zusätzliche Produktmixe generiert.

Die Anzahl der zu verwendenden Produktmixe wird in dieser Fallstudie stark durch die benötigte Rechenzeit für die Simulationsexperimente bestimmt, denn beim Experimententwurf für die Robustheitsanalyse werden die 510 Systemkonfigurationen mit den unterschiedlichen Produktmixen gekreuzt. Bei insgesamt 30 Produktmixen ergibt sich ein Experimentumfang von 30 x 510 = 15300 Experimenten.

Die Experimentbedingungen in Abschnitt 5.1 für die Analysen in Phase 1, mit 5 Wochen Simulationszeit, führten zu einer mittleren Rechenzeit von 5 min pro Experiment. Unter Beibehaltung dieser Bedingung ergibt sich zur Durchführung der 15300 Experimente ein Rechenzeitbedarf von schätzungsweise 76500 min, das entspricht 1275 Stunden oder 53 Tagen. Der Rechenzeitaufwand für die Robustheitsanalyse ist unter diesen Bedingungen zu hoch.

Zur Reduzierung des Rechenzeitaufwandes wurden zwei Techniken eingesetzt:

1. Reduzierung der Rechenzeit für ein Experiment auf 1 min durch Verkürzung der Simulationszeit.
2. Parallelisierung der Experimente durch die Anwendung des proprietären Data-Farming-Management-Tools.

Durch die Verkürzung der Simulationszeit auf eine Woche konnte die Rechenzeit für ein Experiment auf 1 min verkürzt werden. Daraus ergibt sich für die Ergebnisgrößen der Simulation eine stärkere Beeinflussung durch den Initialisierungs-Bias. Als Konsequenz daraus sind beispielsweise die ermittelten Werte für die Ergebnisgröße „Mittlerer täglicher Durchsatz“ kleiner als die Werte aus der Analyse mit einer Simulationsdauer von 5 Wochen. Da das Ziel der Robustheitsanalyse in einem Vergleich der Konfigurationen besteht, wurde diese Einschränkung akzeptiert, um alle 15300 geplanten Experimente durchführen zu können.

Durch die Anwendung eines proprietären Data-Farming-Management-Tools konnte die Experimentdurchführung automatisiert und parallelisiert werden. Die Parallelisierung kann wahlweise auf den Prozessorkernen eines PCs oder auf einer virtuellen Maschine mit mehreren Prozessoren oder Kernen erfolgen. Bei der Durchführung der Experimente auf 8 Kernen einer leistungsstarken virtuellen Maschine wurde ein Durchsatz von 360 Experimenten pro Stunde erreicht. Die theoretische Dauer beträgt annähernd 42 Stunden. Dieser Wert wurde bei der praktischen Durchführung der Experimente überschritten, da eine gewisse Anzahl von Experimenten wiederholt werden musste. Die reale Dauer für die 15300 Experimente betrug annähernd 60 Stunden.

In einem weiteren Schritt der Robustheitsanalyse werden für jede Konfiguration über alle Produktmixe die Werte für die Ergebnisgröße „Mittlerer täglicher Durchsatz“ zusammengestellt. Für diese 30 Werte je Konfiguration werden der Mittelwert und die Standardabweichung berechnet. Das Ziel der Robustheitsanalyse für die obige Ergebnisgröße besteht darin, die Konfigurationen zu identifizieren, die einen hohen Mittelwert und eine geringe Standardabweichung aufwei-

sen. Bild 5 zeigt ein Scatterdiagramm für die Mittelwerte und Standardabweichungen über alle 510 Konfigurationen. Der Bereich von Interesse ist eingekreist.

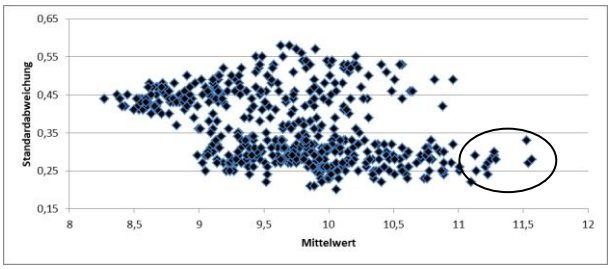

Bild 5: Scatterdiagramm für Mittelwerte und Standardabweichungen der Ergebnisgröße „Mittlerer täglicher Durchsatz“ über alle 510 Konfigurationen.

Zur Quantifizierung der Robustheit wird auf die Anwendung von Verlustfunktionen nach Taguchi [6] zurückgegriffen. Für jede Konfiguration wird ein Verlust ermittelt. Die Berechnung des Verlustwertes für eine Konfiguration erfolgt nach Formel 1

$$L = \sum_{i=1}^{n}(\tau - y_i)^2 \qquad (1)$$

mit L als Wert der Verlustfunktion, τ als Schwellwert, y_i als Wert für die Ergebnisgröße „Mittlerer täglicher Durchsatz“ für den Produktmix i und n der Anzahl der Produktmixe.

Die angewandte Analyse beinhaltet die folgenden Schritte:

1. Definition des Schwellwertes τ für die Ergebnisgröße „Mittlerer täglicher Durchsatz“. Unter Einbeziehung der errechneten Mittelwerte wird ein Schwellwert τ von 15 verwendet. Dieser Wert ist größer als das Maximum der ermittelten Werte.
2. Berechnung des Verlustes für jede Konfiguration.
3. Aufsteigende Sortierung der Liste aus Schritt 2.

Tabelle 4 zeigt die auf Basis der Verlustfunktion errechneten Ergebniswerte der Robustheitsanalyse für die sieben Konfigurationen mit den geringsten Verlusten. Es ist zu erkennen, dass die Werte für die Ergebnisgröße „Mittlerer täglicher Durchsatz“ nicht wesentlich differieren. Ein Systemplaner hat nun die Möglichkeit, diese ausgewählten Konfigurationen auf ihre praktikable Umsetzung hin zu untersuchen.

Konfiguration	Mittl. Durchsatz	Verlust
448	11,57	354,48
373	11,54	361,60
103	11,53	363,60
410	11,29	414,48
237	11,18	417,52
256	11,28	417,84
169	11,25	423,44

Tabelle 4: Liste mit robusten Konfigurationen, dem mittleren täglichen Durchsatz und dem ermittelten Verlust.

Anschließend wurde in einer Korrelationsanalyse untersucht, welche Eingangsgrößen die Robustheit stark beeinflussen. Die Eingangsgrößen mit dem größten Einfluss wurden selektiert. Die ermittelten Werte der signifikanten Eingangsgrößen für robuste Konfigurationen sind in Tabelle 5 aufgelistet.

Die Ergebnisse der Robustheitsanalyse unterstützen den Systemplaner bei der Auswahl geeigneter Konfigurationen. Für robuste Konfigurationen werden die folgenden Werte empfohlen, die Tabelle 5 zusammengestellt sind.

Faktorname	Wertebereich
RTCC_Zone1	1,19 – 1,30
Skill_Zone1	2,0
RTCC_Zone4	1,23 – 1,30
RTCC_Zone3	1,10 – 1,24
RTCC_Zone2	1,16 – 1,28

Tabelle 5: Empfehlungen für die Entscheidungsfaktoren zum Erreichen robuster Konfigurationen.

6 Fazit und Ausblick

Dieser Beitrag präsentiert das methodische Vorgehen zum Data Farming in Verbindung mit simulationsbasierter Robustheitsanalyse und erläutert dieses Vorgehen an einer realen Fallstudie für die Simulation einer Montagelinie. Hierzu werden Anforderungen an die zu verwendenden Simulationssysteme dargestellt, damit diese Systeme im Kontext des Data-Farming operieren können. Des Weiteren werden Hinweise gegeben, wie

existierende Simulationsmodelle aus klassischen Simulationsprojekten für Data-Farming adaptiert werden sollten. In der Fallstudie werden die Ergebnisse ausgewählter Analysemethoden aus dem Data-Farming und Data-Mining für das Simulationsmodell vorgestellt. Für die Robustheitsanalyse hinsichtlich einer Ergebnisgröße wurde beispielhaft eine Verlustfunktion erstellt, die zur Bestimmung robuster Systemkonfigurationen verwendet wurde.

Entwicklungsbedarf besteht sowohl in der Automatisierung des Ablaufs zur Wissensentdeckung durch Data-Farming als auch in der erweiterten Nutzung interaktiver Werkzeuge zur visuellen Analyse. Forschungsbedarfe ergeben sich auch bzgl. des Experimentdesigns von Produktmixen bzw. Produktionsprogrammen unter Beachtung von Qualitätskriterien, wie der Orthogonalität, Ausgeglichenheit sowie der Anzahl der Experimente.

References

[1] Feldkamp N; Bergmann S; Straßburger S; Schulze T. Data Farming im Kontext von Produktion und Logistik. In Wenzel S, Peter T, editors. Simulation in Produktion und Logistik 2017. *Simulation in Produktion und Logistik;*2017. 20.-22. September 2017; Kassel:Kassel:Kassel university press. 169-178.

[2] Feldkamp N; Bergmann S; Strassburger S; Schulze T. Knowledge Discovery in Simulation Data: A Case Study of a Gold Mining Facility. In Roeder TM, Frazier P I; Szechchtman R, Zhou E, Huschka T; Chick S E, editors. Winter Simulation Conference 2016. *Winter Simulation Conference;* 2016 December, Washington D.C.. Piscataway,N.J.: IEEE Inc.1607-1618.

[3] Kleijnen, J P C; Sanchez S M, Lucas Thomas W, Cioppa T M. State-of-the-Art Review: A User's Guide to the Brave New World of Designing Simulation Experiments. *INFORMS Journal on Computing*. 2005; 17 (3): 263–289.

[4] Painter M K, Erraguntla M, Hogg G L, Beachkofski B. Using Simulation, Data Mining, and Knowledge Discovery Techniques for Optimized Aircraft Engine Fleet Management. In: Perrone L F, Wieland F P, Liu J, Lawson B G, Nicol D M, Fujimoto R M, editors. *Winter Simulation Conference*; 2006 December; Monterey. Piscataway, N.J.: IEEE inc. 1250-1257.

[5] Sanches S M. Work Smarter Not Harder: Guidelines for Design Simulation Experiments. In Henderson S G, Biller B, Hsieh M H, Shortle J, Tew J D, Barton R R, editors. Winter Simulation Conference 2007. *Winter Simulation Conference*; 2007 December, Washington D.C.. Piscataway,N.J.: IEEE Inc. 805-816.

[6] Taguchi, G. Quality engineering (Taguchi methods) for the development of electronic circuit technology. *IEEE Transactions on Reliability* 44, 1995. 225–229.

[7] Horne G E; Meyer T E. Data Farming: Discovering Surprise. In Kuhl M E, Steiger N M, Armstrong F B, Joines J A, editors. Winter Simulation Conference 2005. *Winter Simulation Conference;* 2005 December; Orlando FL. Piscataway,N.J.: IEEE Inc. 1082–1087.

[8] Sanchez S M. Simulation Experiments: Better Data, Not Just Big Data. In: Tolk A, Diallo S D, Ryzhov I O, Yilmaz L, Buckley S, Miller J A , editors. Winter Simulation Conference 2014. *Winter Simulation Conference;* 2014 December; Savannah GA. Piscataway,N.J.: IEEE Inc. 805–816.

[9] Brandstein, A G, Horne G E. Data Farming: A Meta-technique for Research in the 21st Century. In Hoffman F G, Horne G E, editors. Maneuver Warfare Science. *Maneuver Warfare Science*; 1998 Dec; Quantico VA. Marine Corps Combat Development Command: 93–99.

[10] Hernandez A S, Lucas T W, Carlyle M. Constructing nearly orthogonal latin hypercubes for any nonsaturated run-variable combination. *ACM Transactions on Modeling and Computer Simulation.* 2012 22, 1–17.

[11] Kleijnen, J P, Sanchez S M, Lucas, T W, Cioppa T M. State-of-the-Art Review: A User's Guide to the Brave New World of Designing Simulation Experiments. INFORMS Journal on Computing 2005; 17: 263–289.

[12] Feldkamp N, Bergmann S, Strassburger S. Knowledge Discovery in Manufacturing Simulations. In Taylor S, editors. Proceedings of the 2015 ACM SIGSIM PADS Conference. *2015 ACM SIGSIM PADS Conference*; 2015 June; London UK. New York: ACM New York. 3 –12.

[13] Keim D A, Mansmann F, Schneidewind J, Thomas J, Ziegler H. Visual Analytics: Scope and Challenges. In: Simoff S, Boehlen M H, Mazeika A, editors. *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*. Heidelberg: Springer; 2008. p 76-90.

[14] Feldkamp N, Bergmann S, Strassburger S, Schulze T. Knowledge Discovery and Robustness Analysis in Manufacturing Simulations. In Chan V, D'Ambrogio A, Zacharewicz G, Mustafee N,editors. Winter Simulation Conference. *Winter Simulation Conference;* 2017 December; Las Vegas. Piscataway, N.J.: IEEE , p 3952-3963.

[15] Park G J, Lee T H, Lee K H, Hwang K H. Robust Design: An Overview. *AIAA Journal.*2006; (44):181–191.

[16] Phadke M S. Quality engineering using robust design. Englewood Cliffs, NJ: Prentice Hall, 1989.

[17] Sanchez S. Robust design: seeking the best of all possible worlds. In Joines R R. Barton K, Kang K, Fishwick P A, editors. Winter Simulation Conference. *Winter Simulation Conference*; 2000 December; Orlando. Piscataway, NJ: IEEE Inc. 69–76.

[18] Dellino G, Kleijnen J P C, Meloni J P C. Robust Simula-

tion-Optimization using Metamodels. In Rossetti M D, Hill R R, Johansson B, Dunkin A, Ingalls R G, editors. Winter Simulation Conference. *Winter Simulation Conference*; 2009 Dec; Austin. Piscataway, NJ: IEEE Inc, 540–550.

[19] Horne G, Åkesson B, Meyer S, Anderson S. Data farming in support of NATO. Final Report of Task Group MSG-088, Neuilly-sur-Seine Cedex: North Atlantic Treaty Organisation, 2014.

[20] Plant Simulation: Siemens PLM Software. Available: https://www.plm.automation.siemens.com/de/products/tecnomatix/manufacturing-simulation/material-flow/plant-simulation.shtml. Accessed:17-July-2018.

[21] FlexSim. Available: https://www.flexsim.com/de/. Accessed:17-July-2018.

[22] Rabe M, Spieckermann S, Wenzel S. Verifikation und Validierung für die Simulation in Produktion und Logistik. Heidelberg: Springer Verlag, 2008. 237.