



TECHNISCHE UNIVERSITÄT
ILMENAU

Department of Computer Science and Automation
Systems and Software Engineering Group

Bachelorarbeit

Konzeptionierung und Implementierung einer intelligenten Ladestation in einer diskreten eventbasierten Simulationsumgebung

Registration Date: XX. January 20XX

Submission Date: XX. June 20XX

Supervisor: Prof. Responsible

Submitted by: Ludwig Breitsprecher

Matriculation Number 54131

ludwig.breitsprecher@tu-ilmenau.de

Acknowledgments

I would like to thank ...

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation	1
1.2. Problemstellung	1
1.3. Zielsetzung	2
2. Grundlagen	3
2.1. Unbemannte Luftfahrzeuge	3
2.1.1. Multicopter	3
2.2. Projektumgebung	4
2.2.1. C++ und Framework OMNeT++	5
2.2.2. Projekt	5
2.2.3. Implementierung	6
3. Ladeverhalten	7
3.1. Energiespeicher	7
3.1.1. Bleisäure Akkumulatoren	8
3.1.2. Nickel Akkumulatoren	9
3.1.3. Lithium-Ionen Akkumulatoren	9
3.2. Ladeverfahren	9
3.3. Beispielmessungen	9
3.4. Fazit	9
4. Konzeptionierung	11
4.1. Anforderungsanalyse	11
5. Results and Comparative Analysis	13
5.1. Multi-Trajectory Results	13

A. Appendix	I
CD Structure	III
Tabellenverzeichnis	V
Abbildungsverzeichnis	VI
List of Source Code	VII
Literaturverzeichnis	VIII■

1. Einleitung

1.1. Motivation

Energie ist eine allgegenwärtige physikalische Größe. Elektrische Energie steht in den Industriestaaten durch einfachen Steckdosenzugriff in großer Menge zur Verfügung. Der sparsame Umgang mit diesem Gut schont die finanziellen Ressourcen und reduziert die negativen Auswirkungen der Stromerzeugung auf unsere Umwelt.

Die unbemannte Luftfahrt ist unter anderem für ihren moralisch bedenklichen Einsatz für militärische Zwecke bekannt. Davon abgesehen werden Drohnen auch in zivilen Bereichen immer verbreiteter. Große Logistikunternehmen wie DHL oder Amazon nutzen Prototypen für die unbemannte Auslieferung von Paketen. Die Suche nach Opfern eines Lawinenunglücks ist eine äußerst gefährliche Aufgabe. Weitere Lawinen können ausgelöst werden oder andere Gefahren werden von der Schneedecke verborgen.

Während die Nutzungsmöglichkeiten von unbemannten Luftfahrzeugen grundverschieden sind, wird Energie benötigt. Die Speicherung größerer Mengen von Energie ist ein anhaltendes Problem, neue Akkumulatoren verbessern die Situation, doch das Grundproblem bleibt bestehen. Energie ist knapp und sie zu transportieren zieht Aufwand, oft in Form von Gewicht, nach sich. Gerade für mobile Geräte ist das Gewicht ein ausschlaggebender Punkt. Für jede, nicht in kürzester Zeit lösbare, Aufgabe muss die Stromversorgung der flugfähigen Helfer gesichert werden.

1.2. Problemstellung

Ladestationen sind ein wichtiger Bestandteil von Elektromobilität jeglicher Art.

1.3. Zielsetzung

2. Grundlagen

Im folgenden Kapitel wird auf die nötigen Grundlagen und Hintergründe eingegangen. Die Ladestation wird in ein bestehendes Projekt eingepflegt. Dieses bestehende Projekt beschäftigt sich mit der Simulation von Missionen für unbemannte Luftfahrzeuge. Einleitend wird das angewendete Verständnis eben jener Luftfahrzeuge erläutert. Im Anschluss wird die Projektumgebung grob skizziert.

2.1. Unbemannte Luftfahrzeuge

Ein Unbemanntes Luftfahrzeug (engl. Unmanned aerial vehicle, UAV) ist ein Luftfahrzeug, welches ohne Besatzung an Bord betrieben wird. Für die Steuerung können Fernsteuersysteme oder an Bord befindliche Computer zum Einsatz kommen. [ICA11] Der von der ICAO definierte Begriff schränkt wenig ein. Abgesehen vom Ausschluss von Flugmodellen für Freizeit- und Luftsportaktivitäten werden keine Flugobjekte ausgeschlossen. Sowohl moralisch umstrittene militärische Drohnen, als auch die Prototypen der automatisierten Paketzustellung fallen unter die Definition. Im Rahmen des bestehenden Projektes wird ein UAV als Multicopter, der zu zivilen Zwecken eingesetzt wird, verstanden.

Quellen

2.1.1. Multicopter

Multicopter sind Flugobjekte mit mehreren Rotoren, die für Auftrieb sorgen. Einer der bekannten Vertreter ist der Quadrocopter (mit vier Rotoren), darüber hinaus werden unter anderem auch Tricopter, Hexacopter und Octocopter eingesetzt. Die Anzahl der Rotoren beeinflussen in erster Linie die Stabilität in der Luft. Darüber hinaus können

Quellen

Ausfälle einzelner Rotoren bei größerer Anzahl ausgeglichen werden. Eine höhere Rotorenanzahl birgt nicht nur Vorteile. Das Fluggewicht und die nötige Leistung steigen. Daraus folgt, dass entweder größere Energiespeicher nötig sind, welche das Gewicht weiter erhöhen würden oder die Flugdauer mit steigender Rotorenanzahl abnimmt.

Seit einigen Jahren finden UAVs auch in privaten Bereichen Einsatzmöglichkeiten. Sie werden als Spiel- und Spaßwerkzeug zum Beispiel in Verbindung mit Augmented Reality (AR) verwendet. Dafür wird unter anderem eine einfache Anwendung benötigt. Die Geräte müssen nach dem Auspacken, Zusammenbauen und Aufladen ohne Fachkenntnisse benutzbar sein. [BCVP11, S. 1477] Diese Vereinfachung für Endanwender wird durch Bordelektronik, die Steuerung und Stabilisierung unterstützt ermöglicht.

Nachfolgend wird beispielhaft näher auf den Quadrocopter eingegangen. Viele der Informationen sind auf die anderen Multicopterarten übertragbar.

Antrieb

Die Rotoren des Multicopters sind üblicherweise an einen Elektromotor angeschlossen. Die benötigte Energie wird von einem integrierten Energiespeicher geliefert. Der Antrieb eines Multicopters ist üblicherweise ein Elektromotor. Die benötigte Energie wird von einem Energiespeicher geliefert. Nahezu ausschließlich kommen hier Lithium-Polymer-Akkumulatoren zum Einsatz, weitere Details dazu können im Kapitel nachgelesen werden.

Kapitel einfügen

Steuerung und Stabilisierung

2.2. Projektumgebung

Dieses Kapitel stellt die Projektumgebung dar. Zu Beginn wird auf die verwendete Programmiersprache und das benutzte Framework eingegangen. Anschließend wird das bestehende Projekt abseits der zu implementierenden Ladestation vorgestellt.

2.2.1. C++ und Framework OMNeT++

C++ ist eine Erweiterung der Programmiersprache C. Hinzugefügt werden Eigenschaften, die das objektorientierte Paradigma ermöglichen. Durch den Sprachursprung bleibt C++ effizient und mächtig. Beispielsweise eine automatische Speicherbereinigung findet nicht statt. Der Anwender hat alles in der Hand. Dessen ungeachtet kann die Abstraktion der Objektorientierung eingesetzt werden, um übersichtliche und wartbare Software zu schreiben.

Frameworks werden eingesetzt, um den Entwicklungsaufwand zu reduzieren. Dies geschieht unter anderem durch die Bereitstellung von häufig genutzten Funktionen für den jeweiligen Anwendungsbereich. Das im Projekt eingesetzte Framework nennt sich OMNeT++, es basiert auf der Programmiersprache C++. Es beschreibt sich selbst als *extensible, modular, component-based C++ simulation library and framework primarily for building network simulators.* Die Nutzung ist für wissenschaftliche Arbeiten gebührenfrei. Die Installation ist durch eine mitgelieferte Entwicklungsumgebung und Anleitung einfach. Eine Tutorialreihe und die online Dokumentation vereinfachen den Einstieg. Das Framework ermöglicht die Erstellung von Objekten, welche miteinander und der Umwelt interagieren können. Darüber hinaus bietet es eine optionale grafische Darstellung, welche den Ablauf insbesondere für Außenstehende veranschaulicht.

Quelle (Webseite)

Quellen

2.2.2. Projekt

Die Arbeit baut auf ein bestehendes Projekt auf. In diesem Projekt "multiUAV-simulation" werden Missionen für UAVs simuliert. Die Missionen setzen sich aus einzelnen Kommandos zusammen, welche in der selben Form auch in der realen Welt genutzt werden können. Eine einfache Mission könnte aus einer Abfolge von mehreren Bewegungskommandos, die unendlich wiederholt werden sollen, bestehen. Im Laufe der Mission wird entleert sich der Akkumulator des eingesetzten UAV. Vor jedem Kommando führt ein UAV eine Ressourcenkalkulation durch und prüft, ob die Energie noch für das nächste Kommando und einen anschließenden Flug zur nächstgelegenen Ladestation reicht. Falls dieser Test ein negatives Resultat liefert, wird das Flugobjekt ausgetauscht und das vorher im Einsatz befindliche fliegt zur nächsten Ladestation.

??

Ein komplexeres Szenario könnte mehrere simultan genutzte UAVs enthalten und zusätzlich eine Begrenzung der vorhandenen Ersatz-UAVs.

2.2.3. Implementierung

Das Projekt nutzt die objektorientierten Möglichkeiten der Sprache C++. Im folgenden Klassendiagramm wird ein Überblick über die vorhandene Struktur zu Beginn der Arbeit gezeigt. Wie in vielen Softwareprojekten wird nicht ein Teil nach dem anderen entwickelt, sondern unterschiedliche Personen arbeiten an unterschiedlichen Teilen des Projektes. Während der Arbeit hat sich auch die Struktur des restlichen Projektes stetig weiterentwickelt.

Klassendiagramm

Ausführung
Klassendiagramm

3. Ladeverhalten

Im folgenden Kapitel wird das Ladeverhalten der Ladestation entworfen. Begonnen wird mit dem Vergleich mehrerer Energiespeicher. Dieser Vergleich soll herausarbeiten, aus welchen Gründen in mobilen Anwendungen vorzugsweise Lithium-Ionen-basierte Akkumulatoren eingesetzt werden. Darauf aufbauend werden übliche Ladeverfahren und deren Spezifika untersucht. Nach dieser von Literatur angetriebenen Untersuchung, werden Beispielwerte eines Ladegerätes und Akkumulators gemessen. Abschließend wird ein Fazit gezogen, welches die Grundlage für den zu implementierenden Ladealgorithmus liefert.

3.1. Energiespeicher

Im einleitenden Kapitel über Multicopter , wurde festgestellt, dass diese üblicherweise von Elektromotoren angetrieben werden. Elektromotoren benötigen eine elektrische Energiezufuhr, welche von Akkumulatoren bereit gestellt wird.

verweis

Akkumulatortypen lassen sich in Gruppen basierend auf ihren chemischen Bestandteilen einteilen. Drei häufig verwendete Typen basieren auf Lithium-Ionen, Nickel und Bleisäure. Durch die gleiche Basis haben Akkumulatoren innerhalb einer Gruppe ähnliche Merkmale. Für die Auswahl eines Energielieferanten sind neben den funktionalen Eigenschaften Energiedichte (Wh/kg), Kapazität (Ah), Spannung pro Zelle (V) auch weitere nicht funktionale Eigenschaften wie Langlebigkeit, Sicherheit und Ladegeschwindigkeit relevant. In der Anwendung sind die wirtschaftlichen Aspekte ebenfalls ein relevanter Faktor. Für die rein wissenschaftliche Betrachtung ergeben sich hier viele Probleme. Unterschiedliche Marktpreise basieren nicht zwingend auf der Wertigkeit der verwendeten Güter, sondern können auch durch Skaleneffekte der Massenproduktion, Gesetze oder Wettbewerb bezüglich der Technologie entstehen. Diese

Effekte wiederum basieren unter anderem auf der Nachfrage nach dem Produkt und damit auch auf der Bekanntheit. Dessen ungeachtet soll die Ladestation für aktuelle Quadropten konzeptioniert werden und nicht für eine Utopie.

3.1.1. Bleisäure Akkumulatoren

In einem Bleisäure Akkumulator besteht die Elektrode aus Blei und Bleioxid, der dazugehörige Elektrolyt besteht aus verdünnter Schwefelsäure. Dieser Akkumulator-type gilt als der älteste Entwurf und reicht bis in das 19. Jahrhundert zurück. Die Nennspannung einer Zelle beträgt 2 V und ist damit ebenso wie die Energiedichte mit ungefähr 30-50 Wh/kg vergleichsweise gering. Die Lebensdauer kann grob in einer Zeitspanne angegeben werden, während die anderen Gruppen vor allem durch die Nutzung abgebaut werden. Je nach Qualität ist eine Lebensdauer von 4-15 Jahren zu erwarten. Durch die lange Bekanntheit der Technologie, sind diese Akkumulatoren relativ günstig.

wirklich?

Einsatzgebiete sind aufgrund des hohen Gewichts vor allem stationäre Anwendungen, beispielsweise Notstromversorgung und Energiespeicher für Photovoltaikanlagen. In kleinerem Ausmaß gibt es auch mobile Anwendungsgebiete, unter anderem als Starterbatterie in Autos werden häufig auf Bleisäure basierende Akkumulatoren eingesetzt.

Quelle

3.1.2. Nickel Akkumulatoren

3.1.3. Lithium-Ionen Akkumulatoren

3.2. Ladeverfahren

3.3. Beispielmessungen

3.4. Fazit

4. Konzeptionierung

Im folgenden Kapitel wird die Planung der Entstehung einer intelligenten Ladestation beleuchtet. Die Reihenfolge der Abschnitte orientiert sich an den nötigen Arbeiten. Begonnen wird mit einer Anforderungsanalyse. Anschließend wird die Verbindung mit dem restlichen Projekt beschrieben. Zum Abschluss wird das Konzept der Ladestation und der zugehörigen Komponenten vorgestellt.

4.1. Anforderungsanalyse

5. Results and Comparative Analysis

In this chapter an analysis and evaluation of the TimeNET implementation of the multi-trajectory algorithm is presented. First, an analysis of its parameters is conducted and for this purpose a model is designed. Execution of this model with all possible values of different multi-trajectory parameters are shown, and the best combination of parameters is found. Afterwards, a comparative analysis of the implementation with the other three TimeNET implemented simulation algorithms is presented. This analysis is based on a previous research conducted by the author of the current work, *Rare-Event Algorithms Analysis and Simulation* [?]. Its goals were to evaluate RE-START TimeNET implementation and the implemented heuristic for its importance function. The multi-trajectory simulation results are compared together with the results obtained from the earlier work, followed by an analysis of how variations of the model properties exert different impact on the execution time of the algorithms. Finally, a non-Markovian model is analyzed and different simulations executed on it are presented.

5.1. Multi-Trajectory Results

In this section the results from the multi-trajectory simulations are compared with the results from the previous work. The multi-trajectory algorithm has been executed with the default parameters over all variations of the models, but only the most significant results are presented here.

5.1. Multi-Trajectory Results

Model Variations					Standard Simulation Results			
Mod	Pla	Tok	Del	Math Res	CPU Time	Overall T.	Value	Error
1	5	1	10	9.00E-05	0:00:13	0:00:03	9.20E-05	2.22%
1	5	1	100	9.90E-09	2:41:08	0:28:59	9.19E-09	7.17%
1	5	1	1000	9.99E-13		» 24 hs		
1	5	3	10	2.70E-04	0:00:11	0:00:03	2.71E-04	0.37%
1	5	3	100	2.97E-08	2:45:31	0:29:44	2.83E-08	4.71%
1	5	3	1000	3.00E-12		» 24 hs		
1	5	10	10	9.00E-04	0:00:29	0:00:06	9.23E-04	2.56%
1	5	10	100	9.90E-08	2:48:56	0:32:47	9.67E-08	2.32%
1	5	10	1000	9.99E-12		» 24 hs		
2	5	1	10	6.83E-05	0:00:30	0:00:06	6.92E-05	1.32%
2	5	1	100	9.61E-09	2:41:24	0:28:34	8.95E-09	6.87%
2	5	1	1000	9.96E-13		» 24 hs		
2	5	3	10	2.05E-04	0:00:11	0:00:03	2.04E-04	0.49%
2	5	3	100	2.88E-08	2:49:14	0:29:59	2.63E-08	8.68%
2	5	3	1000	2.99E-12		» 24 hs		
2	5	10	10	6.83E-04	0:00:17	0:00:04	6.79E-04	0.59%
2	5	10	100	9.61E-08	3:09:25	0:34:05	9.32E-08	3.02%
2	5	10	1000	9.96E-12		» 24 hs		

Tabelle 5.1. – Standard Simulation Results [?]]

Model Variations					RESTART Simulation Results			
Mod	Pla	Tok	Del	Math Res	CPU Time	Overall T.	Value	Error
1	5	1	10	9.00E-05	0:00:15	0:00:04	8.87E-05	1.44%
1	5	1	100	9.90E-09	0:02:20	0:00:25	9.74E-09	1.62%
1	5	1	1000	9.99E-13	0:47:54	0:08:01	1.10E-12	10.11%
1	5	3	10	2.70E-04	0:00:17	0:00:05	3.45E-04	27.78%
1	5	3	100	2.97E-08	0:03:27	0:00:35	3.06E-08	3.03%
1	5	3	1000	3.00E-12	0:59:34	0:11:24	3.00E-12	0.00%
1	5	10	10	9.00E-04	0:00:22	0:00:04	1.06E-03	17.78%
1	5	10	100	9.90E-08	0:13:53	0:02:21	1.17E-07	18.18%
1	5	10	1000	9.99E-12	2:20:50	0:23:54	9.84E-12	1.50%
2	5	1	10	6.83E-05	0:00:17	0:00:05	6.74E-05	1.32%
2	5	1	100	9.61E-09	0:00:35	0:00:07	9.75E-09	1.46%
2	5	1	1000	9.96E-13	0:21:25	0:03:38	9.52E-13	4.42%
2	5	3	10	2.05E-04	0:00:17	0:00:04	2.14E-04	4.39%
2	5	3	100	2.88E-08	0:00:47	0:00:10	3.05E-08	5.90%
2	5	3	1000	2.99E-12	0:36:54	0:06:19	3.49E-12	16.72%
2	5	10	10	6.83E-04	0:00:17	0:00:04	7.18E-04	5.12%
2	5	10	100	9.61E-08	0:01:15	0:00:13	9.51E-08	1.04%

A. Appendix

This appendix presents the details of the stationary analysis and multi-trajectory simulation executions of the model introduced in Figure ?? (Section ??, page ??).

Interconnected Model Analysis Output

Interconnected Model Analysis Output

Analysis Output:

STEADY STATE SOLUTION OF NET Interconnected_Model

STRUCTURAL ANALYSIS ...

GENERATING THE REDUCED REACHABILITY GRAPH USING TIME EFFICIENT ALGORITHM ...

The reduced reachability graph of this eDSPN
cannot be generated because of lacking main memory.

ERROR occurred while derive_SMC execution.
SOLUTION OF MODEL Interconnected_Model FAILED.

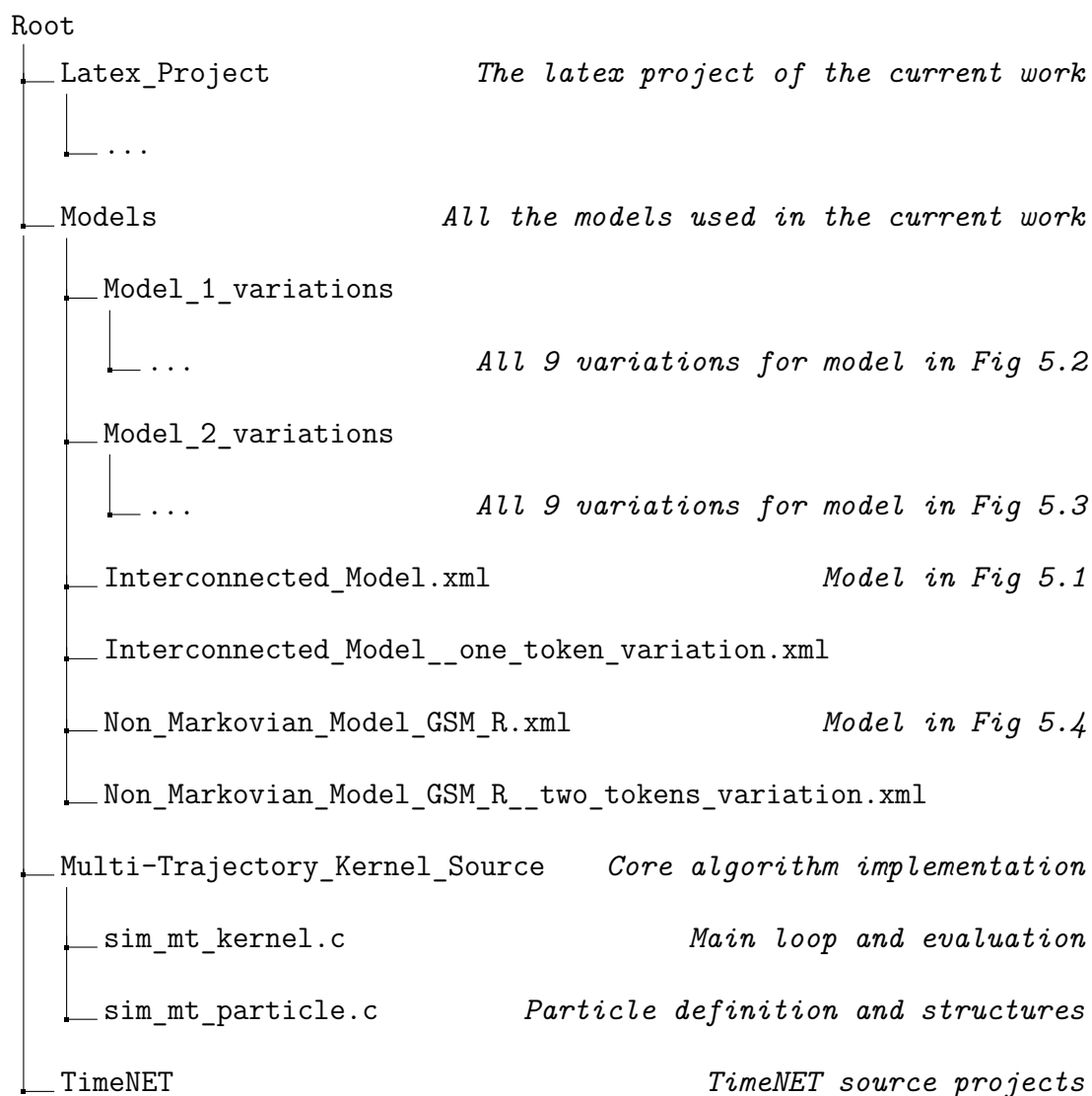
Removing temporary files.

Analysis error, see above.

c@FancyVerbLinee-----

CD Structure

A soft version of the current work is submitted. Following, the CD structure is presented.



└ ...

including all files added by this work

└ Multi-Trajectory Rare-Event Simulation of Stochastic Petri Nets.pdf

Tabellenverzeichnis

5.1. Standard Simulation Results [?]	15
5.2. RESTART Simulation Results [?]	15

Abbildungsverzeichnis

List of Source Code

Literaturverzeichnis

- [BCVP11] BRISTEAU, Pierre-Jean ; CALLOU, François ; VISSIÈRE, David ; PETIT, Nicolas: The Navigation and Control technology inside the AR.Drone micro UAV. In: *IFAC Proceedings Volumes* 44 (2011), Nr. 1, S. 1477–1484. <http://dx.doi.org/10.3182/20110828-6-IT-1002.02327>. – DOI 10.3182/20110828-6-IT-1002.02327. – ISSN 14746670 2.1.1
- [ICA11] ICAO: *ICAO circular*. Bd. 328: *Unmanned aircraft systems: (UAS)*. Montréal : International Civil Aviation Organization, 2011. – ISBN 978-92-9231-751-5 2.1

Liste der noch zu erledigenden Punkte

■ Quellen	3
■ Quellen	3
■ Kapitel einfügen	4
■ Quelle (Webseite)	5
■ Quellen	5
■ ??	5
■ Klassendiagramm	6
■ Ausführung Klassendiagramm	6
■ verweis	7
■ wirklich?	8
■ Quelle	8

Declaration Of Authorship

I, Ludwig Breitsprecher, Matriculation Number 54131, hereby declare that my Bachelorarbeit with the Title

Konzeptionierung und Implementierung einer intelligenten Ladestation in einer diskreten eventbasierten Simulationsumgebung

was produced by me, independently and without using any other references than those mentioned. This work is submitted in completion of the requirements for the Master Degree Research in Computer and Systems Engineering in Ilmenau University of Technology, and I did not publish nor submit this thesis to another university examination board.

Ilmenau, XX.XX.20XX

LUDWIG BREITSPRECHER