# Inhaltsverzeichnis

# 1. Einleitung

Die Datenübertragung von der Erde zum Mars kann bis zu 26 Minuten dauern (vgl. Carsten u. a. 2009, S.2), welches ein Problem für die Erforschung mit den Rovern ist. Bei einem Fernsteuerungsversuch, würde man mit diesen großen Verzögerungen rechnen müssen und könnte somit nur mit sehr vorsichtigen und langsamen Manövern den riesigen, roten Planeten untersuchen. Zur Lösung dieses Problems, können sogenannte autonome Navigationsprogramme helfen, welche im aktuellen Rover der NASA verwendet werden und auch seit den ersten Mars Missionen in einer Form existierten. Dabei ist vor allem ein fehlerfreier Ablauf wichtig, damit die zeit- und kostenaufwändigen Rover nicht beschädigt werden.
In dieser Arbeit werden im ersten Abschnitt die Gründe, Problemstellungen und Methoden zur Entwicklung autonomer Navigationssoftware erläutert. Der zweite Abschnitt handelt über dessen Funktionsweise bei den verschiedenen Rovern und dessen Weiterentwicklung über den Missionen hinweg. Als nächstes wird der Entstehungsprozess meiner Simulation in der dritten Sektion dokumentiert, welche die Funktionsweisen in einer Praxisumgebung darstellt. Und zum Schluss wird noch ein anschließendes Fazit gezogen.

# 2. Sinnhaftigkeit Navigationssoftware

Es macht nur Sinn an einer Software zu forschen, wenn sie ein einhergehendes Nutzen mit sich bringt und ein zuvor ungelöstes Problem löst. So ist es auch bei der Navigationssoftware, welche versucht die Forschung auf dem Mars so effizient wie möglich zu gestalten, was besonders wichtig bei den begrenzten Energie Ressourcen ist (vgl. Laubach, Sharon, 1999, S.25). Wichtig dabei ist auch ein reibungsloser Ablauf, welcher am ehesten mit Regeln bei der Entwicklung garantiert werden kann. Und um ungewollte Verhaltensweisen zu vermeiden, sollte vor dem Einsatz der Software ausgiebig getestet werden.

## 2.1 Problemstellung der autonomen Software

Bei Missionen ist es vorteilhaft möglichst viel in der kürzesten Zeit wie möglich zu erforschen. Daran gehindert wird man aber durch technische Gegebenheiten. Einmal wären da die oben genannten Probleme der Datenübertragung (vgl. Carsten u. a. 2009, S.2). Durch eine große Verzögerung, würde man entweder nur kleinschrittige

und vorsichtige Manöver durchführen können oder es riskieren den kostenaufwändigen Rover zu beschädigen oder gar zu verlieren.
Da ist aber noch die Limitation der Kommunikation des Deep Space Networks (vgl. Maimone, 2017, 07:32-07:49). Bei einem Versuch der Echtzeit Steuerung, müssten die Bilddaten vom Rover zur Erde übertragen werden, damit ein Mensch die nächsten Schritte planen kann. Diese Daten der nächsten Schritte müssten dann an den Rover gesendet werden, damit er diese ausführt. Da aber das Deep Space Network auch mit allen anderen Missionen geteilt wird (vgl. Laubach, Sharon, 1999, S.), gibt es die Limitation der Kommunikationsmöglichkeit von einmal am Tag (vgl. Maimone, 2017, 06:38-06:50), welche schnell überschritten werden müsste.
Dies würde aber außer achtlassen, dass es nicht zu jeder Zeit eine Verbindung zum Rover plausibel ist. Wenn das Raumfahrzeug keine direkte Sicht auf die Erde hat, kann ein Uplink zum Rover verhindert werden (vgl. Laubach, Sharon, 1999, S.25). Bei der Fernsteuerung, wäre die mobile Forschungseinheit in diesen Zeiträumen nicht in der Lage Befehle zu erhalten und auszuführen.
Eine langsame und verzögerte Übertragung gebündelt mit einer begrenzten Kommunikationsmöglichkeit zum Mars, macht es somit nicht möglich die Rover in Echtzeit zu steuern, weswegen eine autonome Navigationssoftware eine große Bedeutung bei der Erforschung des roten Planeten hat.

## 2.2 Herausforderungen bei der Entwicklung der Software

Die Herausforderungen ändern sich über den Missionen hinweg, doch gibt es auch noch viele Gemeinsamkeiten auf welche bei allen Operationen geachtet werden muss.
Einmal wäre da das unbekannte Terrain, welches die Rover durchqueren müssen. Die gesamte Oberfläche des Mars ist größtenteils nicht bekannt (vgl. Morrison, Jack/ Nguyen, Tam, 1996, S.1), weswegen kein vorher programmierter Weg für die Rover erstellt werden kann. Bei der Operation kann es zu neuen Hindernissen kommen, oder könnte das mobile Oberflächenraumfahrzeug an dem Terrain rutschen und die wirkliche Position verändern (vgl. Laubach, Sharon, 1999, S.25). Somit muss die Software in verschiedenen unvorhersehbaren Situationen einen passenden Weg berechnen und den Rover nach diesen Berechnungen bewegen (vgl. Morrison, Jack/Nguyen, Tam, 1996, S.1).

Dazu kommen noch die Gegebenheiten der Hardware, die von Rover zu Rover unterschiedlich sind. Am größten war diese Hürde bei dem ersten Rover, den Pathfinder Rover oder Sojourner. Der dortige Rover Computer hat nur einen Thread mit 0,5 MB an Speicherplatz und verwendete Solarenergie am Tag und Energie einer nicht aufladbaren Batterie in der Nacht (vgl. Harrison, Reid u. a, 1995, S.3). Wegen den knappen Energiereserven und niedrigeren Performance beim Multitasking, kann er nur eine Hauptfunktion, wie das Fahren, Lenken oder die Kommunikation, gleichzeitig bewerkstelligen (vgl. Harrison, Reid u. a, 1995, S.3) (vgl. Morrison, Jack/Nguyen, Tam, 1996, S.2).

Als letztes ist noch die Verantwortung, die die Programmierer bei der Entwicklung der Software haben. Eine Mission auf einen fremden Planeten einen Rover zu schicken, ist fest geplant mit vielen Investitionen, um das Projekt zu verwirklichen. Würde jedoch ein Programmierfehler aufkommen, könnten damit die Chance über den Mars mehr zu erfahren verschwinden und damit sowohl die Investitionen, als auch das Vertrauen in die Firma (vgl. Gerard J. Holzmann, 2014).

TODO (vgl. Laubach, Sharon, 1999, S.25)

## 2.3 Techniken zur Fehlervermeidung

Um solche Fehler zu reduzieren, werden Strategien und Regeln aufgestellt.

Einmal ist da das "level of compliance (LOC)"(Gerard J. Holzmann, 2014) System, welches je nach LOC des Codes für verschieden wichtige Software zugelassen ist. Beim ersten Level oder "LOC-1" (Gerard J. Holzmann, 2014) wird geachtet, dass der Code nicht auf zusätzliche "Compiler-spezifischen Erweiterungen" (Gerard J. Holzmann, 2014) angewiesen ist, also dass keine Extras verwendet werden, um überhaupt den Code zu kompilieren. Dazu muss es noch den "Compiler und einen guten statischen Source Code Analysator" (Gerard J. Holzmann, 2014) ohne Warnungen bestehen. (Statische Source Code Analysator überprüfen den Source Code auf Schwächen, welche zu Schwachstellen führen könnten.)

"LOC-2" (Gerard J. Holzmann, 2014) enthält hauptsächlich Regeln, die für eine sichere und vorhersehbare Ausführung in einen Embedded System Kontext sorgen.

Eine wichtige Regel bei "LOC-3" (Gerard J. Holzmann, 2014) ist die Verwendung von Assertionen, also eine Behauptung von der man ausgeht in dem Punkt im Programm richtig zu sein. Es sollte eine minimale Assertion Dichte von 2% betragen und bei einer falschen Assertion das System in einen sicheren Zustand bringen,

währenddessen das Programm überprüft wird und später weitergeführt werden kann.

Für allen Mission kritische Code sollte mindestens “LOC-4" (Gerard J. Holzmann, 2014) erreicht werden. Auf diesem Level wird das begrenzt das vorprozessieren als auch Funktionen Pointer und Pointer Indirektionen.

Die letzten Level “LOC-5 und LOC-6” (Gerard J. Holzmann, 2014) sind das Ziel für Sicherheitskritische und "human-rated” (Gerard J. Holzmann, 2014) Software. Hier werden noch die restlichen Regeln vom bekannten MISRA C coding Guidelines eingeführt.

Doch selbst bei den strengsten Regeln ist es möglich, dass bestimmte Prozesse ungewollte Verhalten aufzeigen. Um diese noch rechtzeitig vor der Mission zu identifizieren, sind Testungen die letzte Hürde.

Tool based code review????? TODO

## 2.4 Testungen zur Erkennung letzter Mängel

Um möglichst sicherzustellen, dass alle Funktionen der Rover einwandfrei laufen, muss der Rover unter verschiedenen Bedingungen getestet werden und bei unerwünschten Verhalten überarbeitet werden. Getestet wird hierbei auf alle verschiedenen Komponenten.

Um die Navigationssoftware in verschiedenen Mars nahen Situationen zu testen, verwendet man Testung auf einem Mars-analogen Terrain. Um erstmal möglichst Mars ähnliche Landschaft zu erstellen, werden Bilder und andere Messungen von anderen Missionen, wie den Viking Lander, verwendet, woraus das Mooresche Model abgeleitet wurde (vgl. Harrison, Reid u. a., 1995, S. 20). Dieses kann die Frequenz und Größe der Steine vorhersagen, welche an einem Standort auf dem Mars vorzufinden sind (vgl. Harrison, Reid u. a., 1995, S. 20), womit ein Testbereich für den Pathfinder Rover mit den Maßen 4x12m angefertigt wurde (vgl. Harrison, Reid u. a., 1995, S. 6). Als Rover, wird ein analog gebauter Rover wie der Sojouner verwendet. Die einzigen Unterschiede hier ist das größere Gewicht, da es Gummiräder statt Metallräder verwendet, und das Fehlen ein paar Sensoren (vgl. Harrison, Reid u. a., 1995, S. 5-6). Für die Mars Exploration Rovers, wurde auch ein Testbereich mit einer 9 Meter Breite und 22 Meter Länge angefertigt, so wie ein Rover namens “Surface System TestBed (SSTB)” (Carsten, Joseph u. a., 2009, S. 15). Dieser weicht von den

Opportunity und Spirit Rover ab, da dieser keine Solar Anlage besitzt und stattdessen die Energieversorgung mit einem Kabel erfolgt. Auch einige Elektronik sind in einem sterilen Raum, welche ebenfalls mithilfe des Kabels zum Rover verbunden sind (Carsten, Joseph u. a., 2009, S. 15).
Dann gibt es MarsYard (NASA, 2013)

So können in einem kleinen, Mars-analogen Raum verschiedene Tests durchgeführt werden, doch für längere Strecken, wird der Rover im Freien getestet. Dafür verwendet NASA ein Gebiet namens Arroyo Seco, welches neben dem Jet Propulsion Laboratory liegt (vgl. Harrison, Reid u. a., 1995, S. 6).
Diese Teststätten werden hauptsächlich verwendet, um die Leistung der einzelnen Navigationsfunktionen zu bestimmen (vgl. Harrison, Reid u. a., 1995, S. 6).
Echte Position vs. Vermutete Position; Formeln zu Error Vorhersage (Harrison, Reid u. a., S.7-11) TODO

Es wird auch ein PC-Emulator verwendet, welcher es ermöglicht das Programm direkt an den UNIX Arbeitsstationen laufen zu lassen und so nicht immer an den Prototyp Rover laufen muss. (vgl. Morrison, Jack/Nguyen, Tam, 1996, S.6)
Um dann auch die Schnittstellen zu anderen Komponenten wie den Lander des Rover zu überprüfen wird hierbei ein Simulator eingesetzt. Dieser kann auf einem PC oder Laptop als auch auf den UNIX Arbeitsstationen laufen, mit welchem man Kommunikation zwischen den Rover und den Lander nachstellt. Über den Kommunikationskanal zum Radio Empfänger des Rovers, können beispielsweise Operationsbefehle gesendet werden und so das Verhalten getestet werden. (vgl. Morrison, Jack/Nguyen, Tam, 1996, S.6-7)
Mit diesen verschiedenen Einrichtungen und Methoden kann die Software in verschiedenen Situationen getestet und dann je nach Ergebnis überarbeitet und verbessert werden.

## 3. Entwicklung der autonomen Software

Die Navigationssoftware für Rover wurde nicht auf einem Mal fertig geplant und dann umgesetzt. Eine Software durchläuft meistens mehrere Versionen wobei immer wieder Iteriert und Getestet wird. Es werden immer wieder neue Funktionen ausprobiert und auch welche gestrichen. So auch bei der Navigationssoftware,

dessen Entwicklung man besonders an den verschiedenen Mars Missionen sehen kann. Aber auch während den Operationen gab es Neuerungen, die erst nachträglich zum Rover gesendet und verwendet wurden.
Hier wird die Entwicklung der verschiedenen Versionen, als auch deren Funktionsweisen beschrieben.

## 3.1 Mars Pathfinder Mission

Die erste Mars Mission war eine besondere Herausforderung für die Entwickler, da bis zu diesem Zeitpunkt, den 4. Juli 1997, noch kein anderer Rover auf dem Mars gelandet und eingesetzt wurde. Es war auch das ursprüngliche Missionsziel, ein Rover auf den roten Planeten zu bringen. (vgl. NASA, Mars Pathfinder)
Wegen der limitierten Elektronikleistung, dem Missionszielen und des kurzen Entwicklungszyklus, wurde die Software Architektur von vier zentralen Eigenschaften motiviert: der Zuverlässigkeit, der Flexibilität, der Simplizität und der Visibilität. Zuverlässigkeit soll garantiert sein, da es bei einem Fehler kein menschliches Eingreifen möglich ist. Flexibilität um sich den Gegebenheiten der Hardware und der Umgebung anzupassen. Simplizität hat die Vorteile, dass die Lösungen zuverlässig, flexibel und schnell zu entwickeln sind, sowohl auch leicht zu testen sind. Visibilität ist auf die Zustände der Elektronikkomponenten und des Programms bezogen, dass diese möglichst oft übermittelt werden. (vgl. Morrison, Jack/Nguyen, Tam, 1996, S.1-2)

Um den Rover ein Ziel festzulegen, muss man mithilfe der Stereo Bilder Wegpunkte festlegen. Die Funktionsweise der autonomen Navigation, um ein Wegpunkt zu erreichen, erfolgt mit einer einfachen Kontrollschleife. Der Mars Pathfinder Rover fährt ein Reifenradius von 6,5 cm (vgl. Harrison, Reid u. a., 1995, S.16) bei 0,67cm/s (vgl. Harrison, Reid u. a., 1995, S.19), hält dann an und scannt die Umgebung. Dies geschieht mit Infrarotsensoren und CCD Kameras, welche eine Karte mit den ungefähren Höhen des Terrains erstellt. (vgl. Morrison, Jack/Nguyen, Tam, 1996, S.2) Falls ein Hindernis identifiziert wird, dreht sich der Rover um dessen Mittelpunkt in die Richtung, die einen kleineren Drehwinkel besitzt oder zum Wegpunkt, wenn es keinen Vorzug gibt. Kommt kurz darauf ein weiteres Objekt in den Weg, dreht er sich in dieselbe Richtung weiter um ein hin und her Drehen zu

vermeiden (vgl. Morrison, Jack/Nguyen, Tam, 1996, S.3). Wenn der Weg frei ist, fährt der Sojourner erst ein paar Segmente gerade aus und nimmt dann wieder in einem Bogen den Kurs zum Wegpunkt auf. Dieser Zyklus kann durch eine davor festgelegte Zeitüberschreitung, Sensoranzeige außerhalb der sicheren Grenzen, unmittelbare Nähe des Landers oder einem physischen Kontakt unterbrochen werden (vgl. Morrison, Jack/Nguyen, Tam, 1996, S.2). Bei diesen Vorkommnissen, wird die Route zum Wegpunkt abgebrochen. Wurde davor aber spezifisch aufgetragen selbst klar zu kommen, fährt der Rover erst zurück, dreht sich und versucht es nochmal (vgl. Morrison, Jack/Nguyen, Tam, 1996, S.3).

Es gibt noch die Möglichkeit sich für eine bestimmte Distanz durch schmalere Wege "durchzuquetschen" (Morrison, Jack/Nguyen, Tam, 1996, S.3), wobei man dafür nicht mehr den Platz einberechnet um sich auf der Stelle zu drehen. Wenn der Sojourner doch auf ein Hindernis stößt, so fährt er zurück und sucht nach einer Alternative. Ein weiterer Modus der Operation ist "Rock finding" (Morrison, Jack/Nguyen, Tam, 1996, S.3), wobei bei einem außergewöhnlichen Felsen der Mars Pathfinder Rover zum Stein zentriert wird und ihn mit dem Spektrometer scannt. Falls als erstes das Ziel erreicht wird, sucht es in einer Spirale nach einem Gestein. (vgl. Morrison, Jack/Nguyen, Tam, 1996, S.3)

## 3.2 Mars Exploration Rover (MER)

Der Sprung vom Mars Pathfinder Rover zu den Mars Exploration Rover bringt die meisten Änderungen und neue Features hervor. Dies ist aufgrund der weit weniger limitierten Hardware von Spirit und Opportunity als die des Vorgängers (vgl. NASA, Rover "Brains"), wobei nun komplexere Berechnungen und parallele Prozesse, wie das gleichzeitige fahren und scannen, möglich sind.

Das neue System hat nun die Fähigkeit Wissenschaftszielorte und das Ziel festzulegen, wobei von einem Programm namens "CASPER (Continuous Activity Scheduling, Planning, Execution, and Replanning)" (Balaram, J. u. a., 2000, S.2) Zwischenziele angelegt werden, die sich je nach neuen Informationen dynamisch nach einer möglichen Route anpasst. Wenn es bei den Eingabebedingungen der Wissenschaftszielorte, den Zuständen des Rovers oder der Position zum Konflikt kommt, wird der Plan mithilfe eines sogenannten "iterative repair" (Daun, Brian u. a., 1994, S.1) angepasst. CASPER hat auch die Möglichkeit auf eine globale Karte,

unteranderem generiert von Bildern des Lander, zuzugreifen und damit einen effizienteren Weg zu planen. (vgl. Balaram, J. u. a., 2000, S.2) Durch das dynamische neuplanen auf unerwartete Ereignisse, kann der Operator Befehle auf einem höheren Level auftragen, wo bei dem Mars Pathfinder Rover eine genaue und arbeitsintensivere Beschreibung der Ausführung gegeben werden musste. (vgl. Balaram, J. u. a., 2000, S.1) (vgl. Baumgartner, Eric u. a., 2000, S.3)

Zu planen des Weges wird eine Wegplaner mit dem Namen "Rover-Bug" (Baumgartner, Eric u. a., 2000, S.3) verwendet, welches speziell für Rover mit geringen Sichtfeld und eingeschränkten Prozessleistung entwickelt wurde, um die kleinste Nummer an Scans zu erheben (vgl. Balaram, J. u. a., 2000, S.2). Der Planer verwendet dabei die lokalen Sensoren des Rovers und geht je nach Situation einen von zwei Operationsmodi über. Einmal gibt es "motion-to-goal" (Baumgartner, Eric u. a., 2000, S.3), was standartmäßig eingesetzt wird. In dem Modus werden von der Höhenkarte Hindernisse abgelesen und in "convex hulls" (Baumgartner, Eric u. a., 2000, S.3) eingehüllt.

(Balaram, J. u. a, Cheng, Yang u. a, Carsten, Joseph u. a)

### 3.3 Mars Science Laboratory (MSL)

(Angelova, Anelia u. a)

### 3.4 Mars 2020 Mission: Neuerungen autonomer Navigation

(Jet Propulsion Laboratory, o. J.)

## 4. Simulation der Navigationssoftware

(Harrison, Reid u. a., Sektion 7) rock Distribution

## 5. Fazit

## 6. Literaturverzeichnis

### 6.1 Papers


- Angelova, Anelia u. a.: Terrain Adaptive Navigation for Mars Rovers, Journal of Field Robotics 26(4), Volume 26, Issue 4, February 2009.

- Backes, Paul G. u. a.: The Science Activity Planner for the Mars Exploration Rover Mission: FIDO Field Test Results., 2003 IEEE Aerospace Conference, Big Sky, MT., 2003.
- Balaram, J. u. a.: Enhanced Mars Rover Navigation Techniques, IEEE International Conference on Robotics and Automation (ICRA), San Francisco CA, April 24-28 2000.
- Baumgartner, Eric u. a.: Technology Development and Testing for Enhanced Mars Rover Sample Return Operations, IEEE Aerospace Conference, Big Sky, Montana, March 18-25, 2000.
- Cheng, Yang u. a.: Two Years of Visual Odometry on the Mars Exploration Rovers, Journal of Field Robotics 24(3), 169-186, 2007.
- Carsten, Joseph u. a.: Global Planning on the Mars Exploration Rovers: Software Integration and Surface Testing, Journal of Field Robotics, 26(4), 2009.
- Daun, Brian u. A.: "Scheduling and Rescheduling with Iterative Repair". In J. Zweben and M. Fox, editors, Intelligent Scheduling, pages 241-256. Morgan Kaufman, 1994.
- Harrison, Reid u. a.: Mars Microrover Navigation: Performance Evaluation and Enhancement, Autonomous Robots Journal, Special Issue on Autonomous Vehicles for Planetary Exploration, 2(4), 1995.
- Laubach, Sharon: Theory and Experiments in Autonomous Sensor Based Motion Planning with Applications for Flight Planetary Microrovers. PhD thesis, California Institute of Technology, May 1999.
- Morrison, Jack/Nguyen, Tam: On-Board Software for the Mars Pathfinder Microrover, John Hopkins University, Applied Physics Laboratory, Laurel, Maryland, 1996.
- Rusu, Alexandru: Path Planning and Autonomous Navigation for a Planetary Exploration Rover, Université de Toulouse, 2014. ???????TODO ????????
- Volpe, Richard: Navigation Results from Desert Field Tests of the Rocky 7 Mars Rover Prototype, International Journal of Robotics Research, Special Issue on Field and Service Robots, 18(7), July 1999. ??????????

## 6.2 Websites


- Gerard J. Holzmann (2014): Mars Code, https://cacm.acm.org/magazines/2014/2/171689-mars-code/fulltext?mobile=false (Stand: 07.02.2021)
- Jet Propulsion Laboratory (Hrsg., o. J.): Enhanced Autonav for Mars 2020 Rover: Introduction, https://trs.jpl.nasa.gov/bitstream/handle/2014/48231/CL%2317-3124.pdf (Stand: 08.03.2021)
- Jet Propulsion Laboratory (Hrsg., o. J.): The MarsYard II, https://www-robotics.jpl.nasa.gov/facilities/previousFacility.cfm?Facility=1 (Stand: 06.09.2021)
- Jet Propulsion Laboratory (Hrsg., o. J.): The MarsYard III, https://www-robotics.jpl.nasa.gov/facilities/facility.cfm?Facility=14 (Stand: 06.09.2021)
- Maimone, Mark (03.05.2017): The Evolution of Autonomous Capabilities on NASAs Mars Rovers, Southern California Robotics Symposium 2017, [YouTube] https://www.youtube.com/watch?v=u4-4x8GhE6Y (Stand: 13.05.2021)
- NASA (Hrsg., 19.09.2013): Leave the Driving to Autonav, https://mars.nasa.gov/resources/20147/leave-the-driving-to-autonav/ (Stand: 08.03.2021)
- NASA (Hrsg., o. J.): Mars Pathfinder, https://mars.nasa.gov/mars-exploration/missions/pathfinder/ (Stand: 26.09.2021)
- NASA (Hrsg., o. J.): Rover “Brains”, https://mars.nasa.gov/mer/mission/rover/brains/ (Stand: 16.10.2021)