Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Studienarbeit Anwendungen 2

Jan Ruhnke

Quadruped Rough Terrain Robot, Controller Concepts, related Works

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Jan Ruhnke

ruhnke@enapse.de

Quadruped Rough Terrain Robot, Controller Concepts, related Works

Studienarbeit eingereicht im Rahmen der Veranstaltung Anwendungen 2

im Studiengang Master of Science Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Gunter Klemke
Prüfer : Prof. Dr. rer. nat. Kai von Luck

Abgegeben am 30.08.2012

Jan Ruhnke

Thema der Ausarbeitung

Quadruped Rough Terrain Robot, Controller Concepts, related Works

Stichworte

USAR – Roboter, vierbeiniges Laufsystem, Quadruped Rough Terrain Robot, Kontroller

Kurzzusammenfassung

In dieser Arbeit werden, ergänzend zu vorherigen Arbeiten (AW1), einige Kontrollerkonzepte für Rough Terrain Robots vorgestellt. Zudem werden die Strategien bei der Entwicklung beleuchtet.

Jan Ruhnke

Title of the paper

Quadruped Rough Terrain Robot, Controller Concepts, related Works

Keywords

Quadruped robot inspired by mammals, Search and Rescue Robotics, Quadruped Rough Terrain Robot, Controller

Abstract

This thesis focuses on, in addition to previous works (AW1), controller concepts for Quadruped Rough Terrain Robots. Also some development strategies are described.

Inhaltsverzeichnis

1 Forschungsgegenstand 5

1.1 Wozu laufende Roboter / Fahrzeuge 5

1.2 Interessengruppen 5

2 Konzepte und Ideologien der QRTR Forschung 6

2.1 Embodied Knowledge Ansatz 6

2.2 Theoretischer Ansatz 7

2.3 Rapid Development Ansatz 7

3 Related Works QRTR 8

3.1 Projekt Dante / Dante 2 8

3.2 Honda ASIMO 9

3.3 Projekt ARAMIES 10

3.4 Learning Locomotion Program 10

3.4.1 BigDog 11

3.4.2 LittleDog 12

4 Fazit für das Projekt AMEE 14

4.1 Embodied Knowledge Ansatz 14

4.2 Theoretischer Ansatz 14

4.3 Rapid-Development Ansatz 15

4.4 Zusammenfassung 15

5 Anhang 16

Literaturverzeichnis & Quellen 20

Glossar 23

1 Forschungsgegenstand

Diese Arbeit beschäftigt sich mit Kontrollerkonzepten, die es ermöglichen, vierbeinige Roboter in unebenem Gelände sicher laufen zu lassen. Dazu werden einige realisierte Projekte betrachtet. Zudem werden einige Ideologien der unterschiedlichen Lösungen untersucht, um die Designentscheidungen für das Projekt AMEE [Ruh11] [Ruh121] zu begründen.
Die Betrachtungen dieser Arbeit beziehen sich nur auf das Forschungsumfeld der *Quadruped Rough Terrain Robots (QRTR)*. Getrieben durch das DARPA Programm „Learning Locomotion“ [Def05] hat sich ein eigenständiger Forschungsbereich gebildet, der sich hauptsächlich mit QRTR befasst.

1.1 Wozu laufende Roboter / Fahrzeuge

Von Fachleuten und Laien wird oft die Frage gestellt, wozu wir laufende Roboter überhaupt benötigen. Die Antwort wurde treffend vom *BigDog* Team der Firma Boston Dynamics skizziert:
“Less than half the Earth's landmass is accessible to wheeled and tracked vehicles, yet people and animals can go almost anywhere on Earth. This situation motivates the development of robot vehicles that use legs for their locomotion, thereby embracing nature’s mobility solution. The goal is to achieve animal-like mobility on rough and rugged terrain, terrain too difficult for any existing vehicle.
Whereas the Leg Laboratory robots did a good job of demonstrated the feasibility of dynamically balanced legged systems, they had two primary limitations that would need to be addressed to build practical legged vehicles. One is the need for on-board power so the robot could operate in the field without hoses and wires. Another is the need for control algorithms that provide locomotion and stability on rough terrain. We now turn to BigDog, a self-contained quadruped robot that uses many of the ideas and concepts of the Leg Laboratory robots, but also addresses the practical problems of onboard power and rough-terrain controls in order to move toward practical legged vehicles.” [Bos10], [Marc Raibert, Kevin Blankespoor, Gabriel Nelson, Rob Playter and the BigDog Team]

1.2 Interessengruppen

Die Forschung an QRTR wird von einigen Interessengruppen getrieben und gefördert. Hier sind vor allem Nutzer aus den USAR- und SAR-Umfeld zu nennen [Ruh11] [Ruh121]. Zudem ist ein starkes Interesse aus der Land- und Forstwirtschaft zu beobachten [Ruh121]. Auch die großen Raumfahrtorganisationen benötigen QRTR für Fernerkundungen.

2 Konzepte und Ideologien der QRTR Forschung

Um eine eigene Strategie bei der Entwicklung eines QRTR zu schaffen, werden die Ideologien der einzelnen vorhandenen Konzepte erörtert. Bei der Betrachtung der Veröffentlichungen seit dem Jahr 2005 bis 2012 im Robotik- und speziell im QRTR-Umfeld, können drei unterschiedliche Ideologien bzw. Herangehensweisen identifiziert werden.
Eine solche Klassifizierung ist hilfreich, um eine eigene Entwicklungsstrategie / Ideologie aufzustellen und passende Referenzprojekte zu finden. Natürlich sind die Grenzen der Ideologien fließend, wie bei jeder anderen Forschungsgemeinschaft auch.

2.1 Embodied Knowledge Ansatz

Der Ausdruck „Embodied Knowledge“ stammt in seinem Ursprung aus der „Theoretical Psychology“ und beschreibt ein implizites Wissen, das in *Körpern* gebunden ist. Der Begriff *Körper*, betrifft auch Maschinen. Shogo Tanaka beschreibt dies in seiner Arbeit wie folgt:

„A simple and general example is riding a bicycle. Most of us know how to ride a bicycle, and we are able to do it without any deliberation. There is no need to verbalize or represent in the mind all the procedures required. The knowledge seems to be imprinted in one’s body. The knowing-subject here is the body itself, not the mind. Or more precisely, the knowing subject is the minded-body or embodied-mind.” [Tan11]

Für die Informatik stellt sich die Frage, wie man dieses implizite Wissen auf eine Maschine bzw. einen Roboter übertragen kann. Dabei ist die Kernfrage, wie genau erlernen Menschen beispielsweise das Fahrradfahren. Mit diesen Fragestellungen beschäftigt sich die sogenannte *Brain-like Intelligence* [Jan10].
Dieser Ansatz wird stark von japanischen Firmen (z.B. HRI / Honda Research Institute) und Universitäten verfolgt. Seit dem Jahr 2007 wird beim HRI an einem *Brain-like-OS* gearbeitet und ist absolute Grundlagenforschung. Hier ist auch der Unterschied zum Embodiment zu erkennen. Es wird zwar auch davon ausgegangen das Intelligenz einen Körper und physikalische Interaktion braucht, damit die Maschine lernt mit ihm umzugehen. Aber es wird nicht direkt einen Lernalgorithmus gesucht. Vielmehr wird versucht, die Abläufe im menschlichen Gehirn zu verstehen und daraus Algorithmen abzuleiten. Dies Betrifft natürlich nur primitive Grundfunktionen des Gehirns[1]. *Brain-like Intelligence* ist sinngemäß der Top-

[1] Da das HRI einen kommerziellen Hintergrund hat, sind nicht alle Details einsehbar. Aber es kann davon ausgegangen werden, dass nicht versucht werden soll, ein „Ich-Bewusstsein“ in einer Maschine zu erzeugen.

Down-Ansatz des Embodiment. *Brain-like Intelligence* und *Embodiment* sind auch der Bruch mit der klassischen technischen Informatik in der Robotik, bei dem beispielweise Gliederwinkel genau oder auch ungenau[2] berechnet werden.

2.2 Theoretischer Ansatz

Dieser Ansatz fußt in der theoretischen Informatik und ist stark im Forschungsumfeld der QRTRs in Europa und vor allem in Deutschland ausgeprägt. Dabei wird ein relativ strenger Top-Down Ansatz bei der Modellierung verfolgt. Ein Beispiel ist hier das Projekt ARAMIES [Spe07] und das Nachfolgeprojekt SpaceClimber [Bar10] des DFKI, bei dem größtenteils so verfahren wurde.
Die Grundlage ist eine theoretische Modellierung mit der Kernfrage (bei QRTR), „was ist laufen?“. Dieses Modell wird dann nach Abstraktionsebenen feingranularer. Ergänzt wird dieses Modell durch einen Embodiment-Ansatz für das *Motion-planning* und *-controlling*. Dabei handelt es sich oft um ein angepasstes Verfahren aus dem Reinforcement Learning und steht im Gegensatz zur modellbasierten Entwicklung [Spe07].
Ein theoretischer Ansatz bietet einige Vorteile, da er von den Zwängen der realen Entwicklung befreit ist. Hierdurch kann eine übergeordnete Sicht erzielt werden, die oft zu abstrakten und nicht offensichtlichen Lösungen führt. Dabei besteht aber auch die Gefahr, physikalische und laufzeittechnische Zwänge zu übersehen. Beispielsweise benutzt ARAMIES feste Laufmuster, die sich in den letzten Jahren als nachteilig herausgestellt haben [Kol09]. In einem theoretischen Modell sind dynamische Laufmuster, wie von BigDog / LittleDog (3.4) verwendet, schwer zu modellieren.

2.3 Rapid Development Ansatz

Der Ausdruck Rapid-Development stammt aus dem Maschinenbau (CAD/CAM) und ist stark mit dem Vorgehen des Rapid-Prototyping in der Informatik verwandt. Bei der Entwicklung von QRTRs wird ein Bottom-Up-Ansatz verwendet. Die Grundfrage ist hier **nicht** abstrakt: „was ist Laufen?“, sondern konkreter „wie läuft ein Hund im Gelände?“. Dabei werden zum Beispiel Laufstudien und Anatomie von Tieren, in mechanische Lösungen übertragen (Bionik [Nac10]). Es wird stark interdisziplinär gearbeitet und bei allen Entwicklungsschritten sind alle Fachbereiche beteiligt. Beispielsweise arbeiten Informatiker direkt an mechanischen Lösungen mit den Bionikern, Maschinenbauern und Elektrotechnikern zusammen. Im

[2] Mit speziellen Algorithmen können beispielsweise Gliederwinkel sehr schnell, aber ungenau berechnet werden [Ruh11].

Fachbereich Informatik wird agil gearbeitet. Bedingt durch die interdisziplinäre Arbeit wird oft mit Extreme-Programming Methoden gearbeitet, da sich die Rollenverteilung nach Fachbereichen von allein ausprägt.
Dieses Vorgehen wird verstärkt in den USA und besonders im *Learning Locomotion* Programm angewandt. Dies zeigten auch die meisten Veröffentlichungen aus diesem Programm [Adu07] [Bos10] [Kim] [Kol09] [Reb08] [Shk101] [Shk10]. Interessant ist auch, dass die unterschiedlichen Standpunkte bei den Kontrollerkonzepten im „West-Ostküstenstreit“[3] der USA beigelegt scheinen. Da es sich beim *Learning Locomotion* Programm um ein *Academic-DARPA* Projekt handelt, gibt es eine Vielzahl von unterschiedlichen Teillösungen. Fast alle Lösungen verfolgen aber das *Deliberative- / Reactive-Layer* Konzept mit einer monolithischen Architektur [Ruh121]. Dabei werden Planungs- und Regelaufgaben direkt über Automaten (Kontroller) kontrolliert. Es wird im Gegensatz zu anderen modernen Konzepten (z.B. ARAMIES 2.2) auf mathematische Algorithmen gesetzt. Eine KI (Neuronale-Netze / Reinforcement Learning) wird als beratende „Funktion“ für diese Kontroller eingesetzt. Das heißt: Alles bekannte, wie beispielsweise die inverse Kinematik, wird berechnet und nicht gelernt bzw. trainiert. Die nicht berechenbaren Teile der Kontroller werden als „Berater“ im Deliberative-Layer eingesetzt [Bet12] [Ruh121] [Kol09] [Kim].
Trotz der vielen Risiken beim Rapid-Development-Ansatz, sind hier die größten Durchbrüche im Bereich der QRTRs erzielt worden (Stand Aug. 2012). [Bos10] [Kol09] [Reb08] [Shk10].

3 Related Works QRTR

Einige der folgenden Projekte sind nicht im QRTR Umfeld angesiedelt, haben aber entscheidende Schlüsseltechnologien geliefert, die in QRTRs eingesetzt werden. Diese Projekte sind nur eine minimale Auswahl und vernachlässigen einige herausragende Leistungen.

3.1 Projekt Dante / Dante 2

Der Roboter des Projekt Dante (Abbildung 5-1, Seite 16) scheint der erste Rough Terrain Robot zu sein, der außerhalb eines Labors agiert. Der Dante 2 Roboter wurde 1994 erstmals zur Erkundung eines aktiven Vulkans eingesetzt. Dabei handelt es sich um einen

[3] Hierbei handelt es sich um grundlegende Konzepte der Softwarearchitektur. Die unterschiedlichen Ansätze sind: Deliberativer- / Reactive-Layer, Eventsystem, Realisierung über Timeslots, monolithisches System, verteiltes System etc., mit allen Graustufen zwischen den Konzepten.

achtbeinigen statischen Läufer mit festen Laufmustern. In der Weiterentwicklung wurden erste dynamische Kontrolleransätze realisiert. Dieser Roboter ist teilweise autonom und wurde durch eine mobile Konsole gesteuert. Dabei beschreibt David Wettergreen schon im Jahr 1995 [Wet95] die Grundzügen des heutigen Deliberative / Reactive-Layer Konzepts (BigDog/LittleDog Abs. 3.4) und die Auswahl bzw. „Umschaltung“ der Laufmuster. Dieses Projekt ist Grundlagenforschung.

3.2 Honda ASIMO

Beim Honda ASIMO Roboter (Abbildung 5-2, Seite 16) handelt es sich **nicht** um einen Rough Terrain Robot. Er ist ein autonomer bipedaler dynamischer Läufer [Hon07] und die elfte Evolution von Honda Laufrobotern seit 1994. Der Roboter ASIMO selbst, wurde bis heute (2012) in circa 5-7 unterschiedlichen Versionen weiterentwickelt. Bei ASIMO wurden Grundlagenverfahren realisiert und weiterentwickelt. Beispielsweise wurde das ZMP-Konzept (Zero Moment Point) weiterentwickelt und es konnte auf Drucksensoren in den Füßen verzichtet werden [Jan10]. Der ZMP wird nur über Gyroskope und Drehmomentschwankungen in den Beinen berechnet. ASIMO verwendet teilweise ein eventgetriebenes verteiltes System. Die Antriebe im Roboter werden über „eine Vielzahl [Jan10]“ von MCU/ECU gesteuert und entlasten den Haupt-Laufsystem-Rechner um zeitkritische Regelaufgaben. Die bis zu sechzehn Hauptprozessoren sind jeweils einer Aufgabe zugeordnet (Stand 2010). Beispielweise belegt die Pfadplanung einen genau definierten Prozessor. Die einzelnen Prozesse bzw. Layer, Prozessoren und ECUs kommunizieren über Nachrichten miteinander. Ein automatischer Lastausgleich mit einem parallelen verteilten System findet zurzeit wohl nicht statt[4].
Der Schwerpunkt der momentanen Arbeit beim HRI (Honda Research Institut) ist das sogenannte „*Brain-like-OS*“ und damit ein Beispiel für den *Embodied Knowledge* Ansatz (2.1).

Auch wenn ASIMO nur kleine Unebenheiten im Boden mit seinem Laufsystem überwinden kann, bietet er viele innovative Konzepte für einen QRTR. Der Honda ASIMO realisiert Grundlagenforschung.

[4] Aufgrund des kommerziellen Hintergrunds von Honda ASIMO sind genauer Details nicht bekannt. Auch ein persönlicher Kontakt zum HRI wollte sich dazu nicht äußern. Leider wurde bis heute keine komplette Softwarearchitektur veröffentlicht. Wissenschaftliche Veröffentlichungen in Bezug auf Honda ASIMO enthalten nur kleine Teillösungen.

3.3 Projekt ARAMIES

Beim Projekt ARAMIES (Abbildung 5-3, Seite 17) handelt es sich um einen QRTR. Das Entwicklerteam vom DFKI Bremen wurde nach eignen Angabe von Laufstudien einer Katze inspiriert. Der ARAMIES Roboter hat eine Länge von 700mm und eine Schulterhöhe von 600mm bei einem Gewicht von 28kg [Spe05]. Er ist ein statischer Läufer. Das Hardwaresystem basiert auf einem phyCORE Microcontroller Board mit einem XCV600E FPGA. Ergänzt wird das Hardware Konzept mit einem PC-104 für die höhere Logik.
Das softwareseitige Konzept besteht aus einem eigens entwickelten Micro Kernel, der als M.O.N.S.T.E.R. [Spe05] bezeichnet wird. Das Kontrollerkonzept basiert auf verhaltensbasierter Programmierung und soll sich nach Meinung der Entwickler besonders für die Nachbildung „biologischer Systeme“ eigenen, da viele Verhaltensweisen des Laufsystems parallel ablaufen. In der Arbeit von Dirk Spenneberg und Team [Spe05] werden ausführlich die Abwägungen zwischen *RTOS* und *PDL* (Process Description Language) bzw. der *Behavior Language* auf einem Singlecore System diskutiert. Das ARAMIES Team begründet hiermit die Entwicklung des eigenen Kernels, der zwar loopbasierend ist aber trotzdem feste Zeitfenster (ähnlich Timeslots) festlegt. Dabei werden den Verhaltensweisen individuelle Prioritäten zugeordnet. Abläufe die ihr Zeitfenster überschritten haben und von minderer Priorität sind, werden vom System in der Idle Time der CPU abgearbeitet.
Die Motion Control wird als „rhythmic Motion Control“ bezeichnet und basiert auf dem Modell der CPG (Central Pattern Generation) und des sogenannten „Reflex-Models“ [Spe05]. Dabei werden die Schritte als rhythmische Trajektorien der Gelenke betrachtet. Die Bewegungsabläufe basieren auf Bezier-Splines. Beispielsweise wird die Länge, zwischen dem Startpunkt des Splines und dem Hilfspunkt, gleichzeigt auch zur Beschleunigung der Antriebe genutzt. Die eigentliche Regelung der Antriebe erfolgt aber über einen PID-Regler [Spe05]. In den meisten Tests wurden nur die Knie- und Hüftantriebe mit diesem Prinzip gesteuert.

Die Entwicklung bei ARAMIES ist stark modellgetrieben und ein gutes Beispiel für den theoretischen Ansatz (2.2). Das gesamte Konzept enthält viele neue Abwandlungen bekannter Konzepte und geht hier einen neuen und eigenen Weg in der QRTR Entwicklung.

3.4 Learning Locomotion Program

Das „Learning Locomotion Program“ der DARPA (Förderprogramm des US-Verteidigungsministeriums) wurde im Jahr 2005 gestartet und hat den Schwerpunkt der QRTRs [Ruh121]. Das Projekt wurde im Jahr 2009 und 2012 erneut verlängert. Mit jeder

Verlängerung des Förderprogramms wurde der Forschungsauftrag um neue Themen erweitert. Die Programmerweiterung im Jahr 2012 weitet das QRTR Forschungsgebiet auf die Suche nach autarken Stromversorgungen für QRTRs aus.
Gesteuert wird dieses Projekt durch den akademischen Spinoff „Boston Dynamics“. Die Firma „Boston Dynamics“ versorgt die US-Universitäten mit der Mechanik und dem SDK des LittleDog (3.4.2) Roboters, als Basis für die Forschung. Die Ergebnisse der Forschung werden in den Robotern BigDog(3.4.1) und AlphaDog[5] verwertet [Ruh121].

3.4.1 BigDog

BigDog (Abbildung 5-5, Seite 18) ist ein vollwertiger QRTR. Er ist ein dynamischer Läufer und wird hydraulische über einen Verbrennungsmotor angetrieben. Durch die hydraulischen Beine ist BigDog zu sehr schnellen Bewegungen fähig und ermöglicht sogar das Springen und Galoppieren [Bos10]. Im Gegensatz zu allen anderen oben betrachteten Robotern stellten hier die Trägheitsmomente der Beine einen zu berechnenden Faktor dar.
Das Kontroller-Konzept wurde mit einem H-RTOS (Hard-Real Time OS) realisiert. In BigDog wird das Deliberative- / Reactive-Layer Konzept verwendet, wie es in der AW1-Arbeit [Ruh121] bereits beschrieben wurde. Hardwareseitig handelt es sich um ein Multiprozessor System. Das gesamte Softwaresystem ist in einzelne logische Kontroller feingranular gegliedert. Jedem dieser Kontroller wird ein Timeslot zugeordnet. Dabei wurde Bottom-Up gearbeitet und es wurde sichergestellt, dass die Kontroller ihre Aufgabe in diesem Timeslot erledigen können. Dies ist möglich, da die Kontroller auf einfachen berechenbaren Automaten basieren. Die gesamte Softwarearchitektur kann im Anhang (Abbildung 5-7, Seite 19) inkl. Zeitvorgaben eingesehen werden. Leider sind auch hier nicht alle Details veröffentlicht worden. Diese können aber weitestgehend aus den Veröffentlichungen von LittleDog abgeleitet werden.
Der Roboter BigDog ist aber auf das schnelle Laufen im Gelände ausgelegt. Auf stark unebenem Untergrund sinkt die Laufgeschwindigkeit beim Steigen stark ab [Bos10]. Dieses absinken der Laufgeschwindigkeit kann kaum mit einem dynamischen Läufer verhindert werden [Ruh121].

[5] Leider sind über den QRTR AlphaDog (Abbildung 5-4, Seite 17) kaum Informationen bekannt. Dieser Roboter ist für militärische Einsätze konzipiert. „Er ist größer, stärker und schneller als BigDog [Bos10]“. Software Konzepte und verwendete Technologien sind nicht veröffentlicht worden.

3.4.2 LittleDog

LittleDog (Abbildung 5-6, Seite 18) ist ein statischer / pseudodynamischer Läufer und ein QRTR. Auch er basiert auf dem Deliberative- / Reactive-Layer (D/R-Layer) Konzept [Ruh121]. Aufgrund der geringen Größe von LittleDog sind keine Rechner im Roboter integriert. Die Hauptrechner sind extern und kommunizieren per WLAN mit dem Roboter. Dies hat aber auch den Seiteneffekt, dass handelsübliche PC mit Windows® oder Linux als Kontroller verwendet werden. Damit sind fast alle Konzepte nicht auf eine bestimmte Hardwareplattform abgestimmt, was sehr positiv gesehen werden kann.
LittleDog wurde als Academic-DARPA Projekt konzipiert. Dadurch gibt es eine Vielzahl erstaunlich simpler Lösungen für QRTRs. Da das Grundkonzept von LittleDog bereits in vorherigen Arbeiten erörtert wurde, werden nachfolgend einige interessante Arbeiten beschrieben. Die Auswahl erfolgte nach dem Gesichtspunkt der Verwendbarkeit im Projekt AMEE.
In der Arbeit „***A Control Architecture for Quadruped Locomotion Over Rough Terrain***“ [Kol09] beschreiben Kolter und Team sehr praxisnah ein Kontrollerkonzept für einen QRTR (Abbildung 5-8, Seite 19). Für die Pfadplanung wurde mit einer Stereo-Kamera das Gelände von ober erfasst. Bedingt durch die Bildverarbeitung wurde der Deliberative Teil des D/R-Layer Konzepts nochmals aufgeteilt. Es wird in einem sog. High-Level-Planner die Extraktion der Bodenbeschaffenheit mit dem Body-Path-Planner und dem Footstep-Planner zusammengelegt. Neu ist auch, dass der Body-Path-Planner und der Footstep-Planner getrennt implementiert wurden und nur über eine Schnittstelle kommunizieren. Dies hat den Vorteil, dass im ersten Schritt der Planung geprüft wird ob der Körper des Roboters überhaupt die Steigung bewältigen kann, ohne umzukippen. Parallel dazu wird für jede geplante Position ein Footstep-Plan erstellt, um zu überprüfen ob die Füße im Bereich des Körpers Halt finden werden. Diese Planung wird relativ grob und fortlaufend berechnet. Der Low-Level-Planner (auch Deliberative-Layer) erstellt parallel dazu die nötigen Bewegungen für alle vier Beine, um die nächstliegende Position zu erreichen. Der Low-Level-Controller (Reactive-Layer) kontrolliert nun die Antriebe direkt und überwacht die Stabilität des Roboters. Auch wenn der Roboter dabei noch in einem stabilen Zustand ist, berechnet ein Kontroller ständig wie er stabilisiert werden kann. Eine Besonderheit ist auch, dass erst im Reactive-Layer die genaue Fußposition berechnet wird. Dies geschieht in Wechselwirkung mit dem Gyroskop und den Messwerten der Antriebe in den Beinen.
Die Arbeit „***Stereo Vision and Terrain Modeling for Quadruped Robots***“ [And09] ebenfalls von Kolter und Team ist die Fortsetzung der obigen Arbeit und beschreibt einen Kontroller

zur Bodenerfassung. Dabei wurde die Stereo-Kamera auf dem Roboter LittleDog befestigt. Schwerpunkt dieser Arbeit ist die Erweiterung des vorhandenen Kontrollerkonzepts. Es wird ein Verfahren vorgestellt, wie die fehlenden Bodenprofilinformationen (bedingt durch den Winkel der Kamera) einfach geschätzt bzw. berechnet werden können.

Die Veröffentlichung „***Robot Motion for Obstacle Negotiation***“ [Adu07] von Adukuzhiyil und Team stellt eine Erweiterung vor, wie LittleDog Hindernisse erkennen und überwinden kann. Es wird ein Verfahren basierend auf einem neuronalen Netz vorgestellt, bei dem markante und oft vorkommende Hindernisse trainiert werden. Das Training umfasst komplette Verhaltensweisen. Beispielweise wurde für Stufen, die höher sind als LittleDog selbst, springende Bewegungen trainiert, um das Hindernis zu überwinden. Dabei lag die Hauptschwierigkeit in der Merkmalsextraktion der Hindernisse. Auch hier wurde mit einer Stereo-Kamera gearbeitet.

Alexander Shkolnik und Team beschreiben in „***Bounding on Rough Terrain with the LittleDog Robot***“ [Ale11] einen schnellen Pfadplanungsalgorithmus am Beispiel von RRTs (Rapidly Exploring Random Trees). Das Team beschreibt sehr genau wie dieser Algorithmus implementiert wurde. Das Team konnte zeigen, dass es sich bei der Pfadplanung „nur“ um einen 16 dimensionalen Zustandsraum handelt. Es wurden folgende Algorithmen vorgestellt: Ein Algorithmus der dynamisch die Menge der erfassten Daten reduziert, je nach Nähe und Relevanz für die Pfadplanung. Weiterhin einen rechenzeitstabilen Kontroller für die Pfadplanung, der auch bei Störungen des Kamerasystems stabil bleibt. Der Schwerpunkt dieser Arbeit beschäftigt sich mit der Zustandsraumverkleinerung und die Auswirkungen auf das Kontrollerkonzept.

Eine Grundlagenarbeit für die Entwicklung von QRTR-Kontrollern ist die Arbeit von Rebula und Team, „***A Controller for the LittleDog Quadruped Walking on Rough Terrain***“ [Reb08]. Hier werden Kontroller vorgestellt, wie sie auch in anderen Arbeiten zum Projekt AMEE und dem BigDog bereits evaluiert wurden. Es werden genaue Details zum D/R-Layer Konzept vorgestellt.

Dies ist leider nur ein kleiner Ausschnitt aus den Veröffentlichungen des *Learning Locomotion* Programms. Auffällig an diesen Arbeiten ist die praktische Umsetzbarkeit und Realisierung. Fast jede Arbeit enthält Code-Beispiele und einfache Algorithmen. Sie basieren alle auf dem vorgestellten Kontrollerkonzept und größtenteils auf hierarchischen Automaten.

Diese Arbeiten inkl. BigDog sind typische Vertreter des Rapid Development (2.3) Ansatzes.

4 Fazit für das Projekt AMEE

Die folgende Beurteilung bezieht sich strikt auf die Gegebenheiten und Ressourcen (Manpower, Finanzen und Zeit) des AMEE Projekt Teams. Intension dieses Fazits ist es, eine Entwicklungsstrategie für dieses Projekt zu finden und Begründungen zu skizzieren.

4.1 Embodied Knowledge Ansatz

Das Projekt AMEE hat das Ziel einen QRTR möglichst schnell und kostengünstig zu realisieren. Das AMEE Projektteam ist relativ klein und auf eine möglichst simple Entwicklung angewiesen. Der Embodied Knowledge Ansatz (2.1) befindet sich zurzeit (heute 2012) noch in der Grundlagenforschung und ist mit einem erfahrenen Team (z.B. beim HRI) besetzt. Zudem werden für diesen Ansatz eine hohe Manpower und viel Zeit benötigt, die dem AMEE Team nicht zur Verfügung steht. Dieser Ansatz ist *nicht* für das AMEE Projekt geeignet.

4.2 Theoretischer Ansatz

Der theoretische Ansatz wird durch viele Modellierungssprachen und Tools unterstützt. Damit ist es möglich, neuartige Lösungen zu erarbeiten. Leider sind auch die Gefahren sehr hoch, technische und physikalische Gegebenheiten zu übersehen. Das AMEE Team muss einen funktionsfähigen Prototyp des kompletten Roboters realisieren. Hierbei erscheint dem AMEE Team das Risiko des theoretischen Ansatzes zu groß.
Beispiel: Bei den meisten QRTRs, die durch einen Top-Down und besonders durch einen theoretischen Ansatz entstanden sind, werden die Beine bzw. die Fußspitzen auf Splines bewegt (z.B. [Spe05]). Dies ist die theoretische und mathematische Lösung des Problems. Erst in einem Bottom-Up Ansatz (ausgehend von der mechanischen Konstruktion) fällt auf, das die Präzision von Splines für ein Laufsystem völlig unnötig ist. Wird ein Bein ohne Bodenkontakt über drei bis vier Zwischenpunkte zum Zielpunkt bewegt, ergeben sich zwangläufig Splines. Die Mechanik zwingt die Glieder auf Kreisbahnen [Ruh11]. Zudem sind die Bewegungen wesentlich Energieeffizienter, da die Antriebe der „natürlichen“ Bewegung der Mechanik folgen [Ruh11].
Dieses simple Beispiel mag nur für den Roboter AMEE gelten, zeigt aber deutlich die Gefahren eines theoretischen Ansatzes. *Dies soll aber nicht heißen, dass theoretische Modelle abgelehnt werden.* Das AMEE Team setzt diese Modelle und Methoden ein, um beispielsweise Sicherheitsaspekte zu überprüfen. Weiterhin werden auch Teilprobleme mit diesem Ansatz erarbeitet, da vor allem parallele Abläufe einfach geprüft werden können.

4.3 Rapid-Development Ansatz

Das Projekt AMEE beginnt mit einer eigenen Hardwareplattform und erfordert interdisziplinäres Arbeiten. Es müssen Probleme des Maschinenbaus, Mechatronik, Elektrotechnik und Informatik gelöst werden. Aufgrund der Personalressourcen muss sich jeder im AMEE Team auch mit fachfremden Fragen beschäftigen. Dies ist ein erstes Indiz für das Rapid-Development.
Der technische Vorsprung des „Learning Locomotion Programs“(3.4) und die Vielzahl an Vorarbeiten bestätigen bei anderen Teams (z.B. [Shk10] [Shk101] [Reb08]) den Erfolg des Ansatzes. Beeindruckend ist vor allem die Geschwindigkeit in der diese Lösungen realisiert wurden.
Abgesehen vom eigentlichen Ansatz der Entwicklungsstrategie sind die Lösungen aus diesem Programm so stark modular strukturiert, das die bereits aufgestellten Konzepte und Realisierungen des AMEE Teams [Ruh11] [Ruh12] [Bet10] [Bet12], leicht integriert werden können. Zudem sind die Lösungsstrategien für AMEE ähnlich.

4.4 Zusammenfassung

Für das Projekt AMEE wird nach dem Rapid-Development (2.3) Ansatz mit agilen Methoden Bottom-Up vorgegangen. Die Realisierung des ersten AMEE Beins [Ruh11] hat erfolgreich gezeigt, dass dieser Ansatz gut geeignet ist. Auch die Ergebnisse des „Learning Locomotion Programs“ (3.4) werden als Leitlinie und Basis für das Projekt AMEE betrachtet. Diese Konzepte werden mit eigenen erweitert. Bei AMEE wird das Konzept, um ein eventbasierendes System erweitert [Bet10] [Ruh11]. Zudem werden zeitkritische Aufgaben in physikalischen Kontrollern bearbeitet [Ruh11]. Das Footstep-Planning wird um eine beratende KI, mit reinforcement Learning, erweitert [Bet12].
Fraglich ist nur, ob mit wachsender Teamgröße das AMEE Team diese Strategie weiterverfolgen kann oder ob eine stärkere Führung in der Entwicklung nötig wird.

5 Anhang

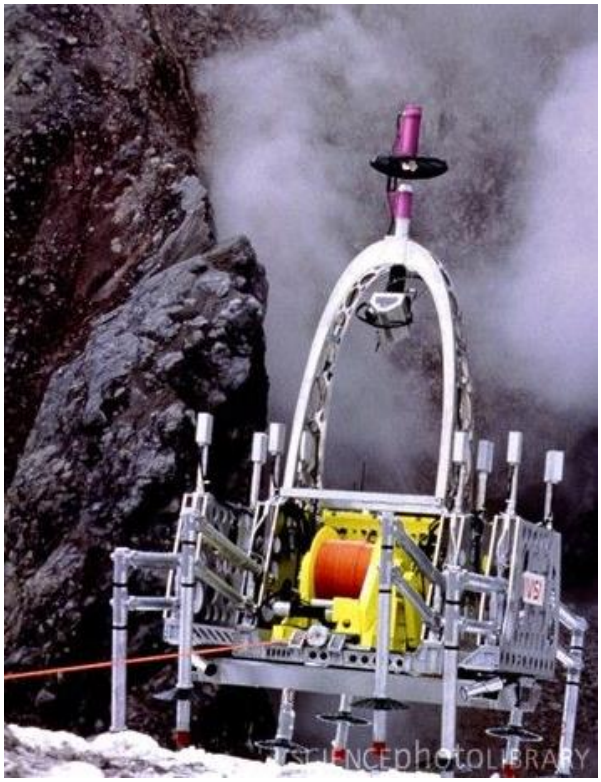

Abbildung 5-1 Projekt Dante, Roboter Dante 2 [Wet95]

Abbildung 5-2 Honda ASIMO [Hon07]

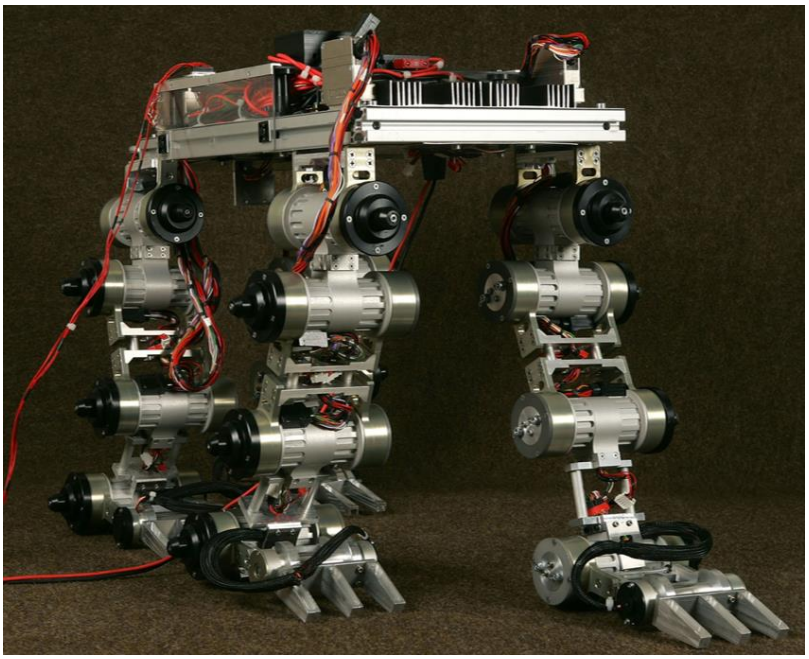

Abbildung 5-3 Projekt ARAMIES [Spe07]

Abbildung 5-4 Boston Dynamics Alpha Dog [Bos10]2012

Abbildung 5-5 Boston Dynamics BigDog [Bos10]

Abbildung 5-6 Boston Dynamics LittleDog [Bos10]

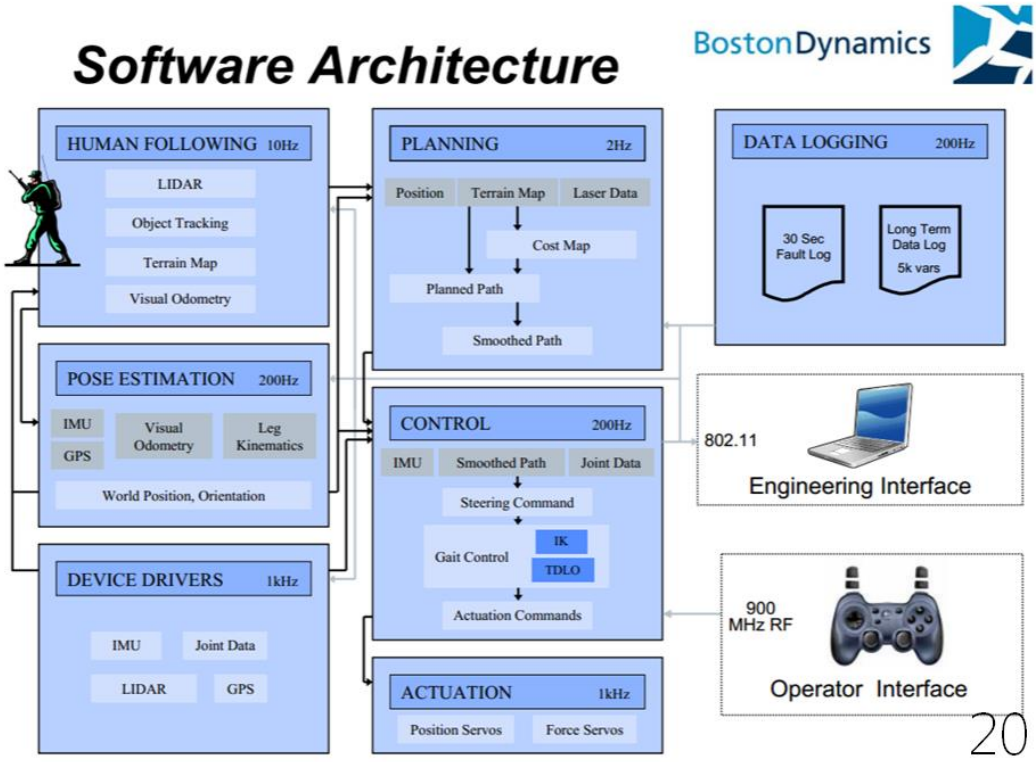

Abbildung 5-7 Software Architektur BigDog [Bos10]

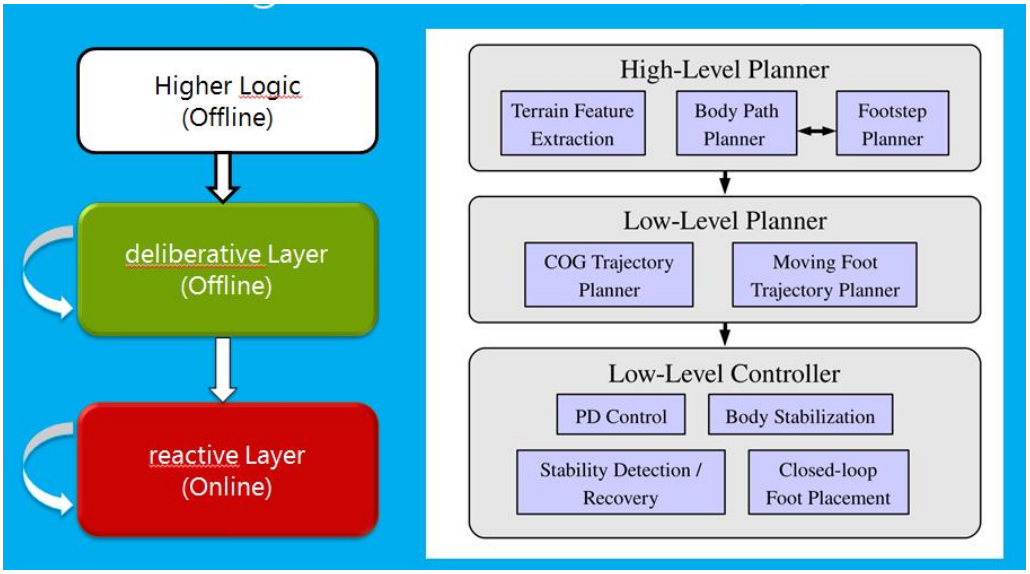

Abbildung 5-8 Grobe Software Architecture Little Dog, Stanford University [Kol09]

Literaturverzeichnis & Quellen

[Adu07] **Adukuzhiyil, Anish, Singh, Harshit and Vantimitta, Pavani. 2007.** *Robot Motion for Obstacle Negotiation.* [PDF] Stanford USA : Stanford University, 2007. IEEE Int. Conf. Robotics and Automation.

[Ale11] **Alexander Shkolnik, Michael Levashov, Ian R. Manchester and Russ Tedrake. 2011.** *Bounding on Rough Terrain with the LittleDog Robot.* [PDF] MASSACHUSETTS,USA : MASSACHUSETTS INST OF TECH, 2011. DOI: 10.1177/0278364910388315.

[Ani09] **Anish Adukuzhiyil, Harshit Singh, Pavani Vantimitta. 2009.** *Robot Motion for Obstacle Negotiation.* [PDF] USA : Stanford University , 2009.

[Bar10] **Bartsch, Sebastian. 2010.** DFKI. [Online] 2010. [Cited: August 02, 2012.] http://robotik.dfki-bremen.de/de/forschung/projekte/weltraumrobotik/spaceclimber.html.

[Bet12] **Bettzüche, Björn. 2012.** *Foothold Selection for Quadruped Locomotion.* [PDF] Hamburg, Germany : HAW-Hamburg, 2012.

[Bet10] **—. 2010.** *Machbarkeitsprüfung zur Entwicklung von SW-Anwendungen mit MS-Robotics Developer Studio für das Robocup Rescue Szenario.* [PDF] s.l., Hamburg : HAW Hamburg, Technische Informatik, Juli 2010.

[Bos10] **Boston Dynamics Inc. 2008, 2010.** *BigDog Overview.* [Internet] 78 Fourth Avenue, Waltham, MA, 02451-7507, US : Boston Dynamics Inc., 2008, 2010. http://www.bostondynamics.com.

[Che09] **Cherouvim, Nicholas and Papadopoulos, Evangelos. 15 June 2009.** *Novel Energy Transfer Mechanism in a Running Quadruped Robot with One Actuator per Leg.* [PDF] 9 Heroon Polytechniou Street, 15780 Athens, Greece : Department of Mechanical Engineering, National Technical University of Athens,, 15 June 2009. Advanced Robotics 24 (2010) 963–978.

[Def05] **Defense Advanced Research Projects Agency. 2005.** Learning Locomotion. USA : Goverment USA, 2005.

[Hon071] **Honda. 2007.** *ASIMO Technical Information.* [PDF] Japan / Germany : Honda Motor Co., Ltd., Public Relations Division, 2007.

[Hon07] **Honda Motor Co., Ltd. September 2007.** *ASIMO Technical Information.* [PDF] s.l. : Honda Motor Co., Ltd., September 2007. kA.

[Jan10] **Janssen, Herbert, Principal Scientist. 2010.** Personal Communication Honda ASIMO. [EMail/Phone]. Honda Research Institute Europe GmbH, Carl-Legien-Strasse 30, 63073 Offenbach/Main, Germany : s.n., 2010.

[Kim] **Kimura, H., et al.** *Towards 3d adaptive dynamic walking of a quadruped robot on irregular terrain by using neural system model.* s.l. : 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems.

[And09] **Kolter, J. Zico, Andrew, Y. and Youngjun, Kim. 2009.** *Stereo Vision and Terrain Modeling for Quadruped Robots.* [PDF] CA 94305, USA : Stanford University, Stanford, 2009.

[Kol09] **Kolter, J. Zico, Rodgers, Mike P. and Andrew, Y. 2009.** *A Control Architecture for Quadruped Locomotion Over Rough Terrain.* [PDF] CA 94305 USA : Computer Science Department, Stanford University, Stanford, 2009.

[Nac10] **Nachtigall, Werner. 2010.** *Bionik als Wissenschaft.* Heidelberg : Springer-Verlag, 2010. ISBN 978-3-642-10319-3 .

[Reb08] **Rebula, John R., et al. 2008.** *A Controller for the LittleDog Quadruped Walking on Rough Terrain.* [PDF] Florida 32502, USA : Florida Institute for Human and Machine Cognition, 2008. jrebula@alum.mit.edu.

[Ruh121] **Ruhnke, Jan. 2012.** *A Walksystem for a Quadruped Rough Terrain Robot, Controller Concepts AW1.* [PDF] Hamburg, Germany : HAW-Hamburg, 2012. AW1.

[Ruh11] **—. 2011.** *Entwicklung und Realisierung eines vierbeinigen USAR-Roboter-Laufsystems.* [PDF] Hamburg, Germany : HAW-Hamburg Dep. Technische Informatik, Juni 02, 2011. Bachelor Arbeit.

[Ruh12] **—. 2012.** *Simulation der Drehmomente in einem Roboter Bein, Ausarbeitung Modellierung Technische Systeme .* HAW-Hamburg : s.n., 2012.

[Shk11] **Shkolnik, Alexander and Tedrake, Russ. 2011.** *Inverse Kinematics for a Point-Foot Quadruped Robot with Dynamic Redundancy Resolution.* [PDF] Cambridge, MA 02139 USA : Massachusetts Institute of Technology, 2011.

[Shk10] **Shkolnik, Alexander, et al. 12.2010.** *Bounding on Rough Terrain with the LittleDog Robot.* [PDF] MASSACHUSETTS USA : MASSACHUSETTS INST OF TECH , 12.2010. DOI: 10.1177/0278364910388315 The International Journal of Robotics Research.

[Shk101] **Shkolnik, Alexander, et al. 2010.** *Motion Planning for Bounding on Rough Terrain with the LittleDog Robot.* [PDF] Cambridge, MA 02139, USA : Computer Science and Artificial Intelligence Lab, MIT, 2010.

[Spe07] **Spenneberg, Dirk . 2007.** http://robotik.dfki-bremen.de. *ARAMIES.* [Online] April 1, 2007. [Cited: Juni 27, 2011.] http://robotik.dfki-bremen.de/de/forschung/projekte/weltraumrobotik/aramies.html.

[Spe05] **Spenneberg, Dirk, et al. 2005.** *ARAMIES: A FOUR-LEGGED CLIMBING AND WALKING ROBOT.* [PDF] D-28359 Bremen, Germany : University of Bremen, Robotics Lab, Faculty of Mathematics and Computer Science, 2005.

[Tan11] **Tanaka, Shogo. 2011.** *The notion of embodied knowledge.* [PDF] Japan : Captus University Publications, 2011.

[Wet95] **Wettergreen, David, Pangels, Henning and Bares, John. August 1995.** *Behavior-based Gait Execution for the Dante II Walking Robot.* [PDF] Pittsburgh, PA 15213-3891, USA : Carnegie Mellon University, August 1995.

Glossar

DARPA: *Defense Advanced Research Projects Agency*, ist eine Behörde des Verteidigungsministeriums der US-Regierung, die Forschungs-Projekte für die US-Streitkräfte durchführt.

USAR: *Urban Search and Rescue*, ist die Bezeichnung für Rettungseinsätze mit Gebäudeschäden in städtischer Bauweise.

SAR: *Search and Rescue,* ist die allgemeine Bezeichnung für Rettungseinsätze.